Hitachi Single-Chip Microcomputer

H8/3664 Series

H8/3664
HD6433664

H8/3663
HD6433663

H8/3662
HD6433662

H8/3661
HD6433661

H8/3660
HD6433660

H8/3664F-ZTAT™
HD64F3664

Hardware Manual

HITACHI

ADE-602-202A
Rev. 2.0
9/25/00
Hitachi, Ltd.

Hitachi semiconductor

Cautions

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

Preface

The H8/3664 Series of single-chip microcomputers has the high-speed H8/300H CPU at its core, with many necessary peripheral functions on-chip. The H8/300H CPU instruction set is compatible with the H8/300 CPU.

The H8/3664 Series includes such peripheral functions as four timers, an I^2C bus interface, a serial communication interface, and a 10-bit A/D converter, so that they can be used as an embedded microcomputer for a sophisticated control system.

This manual describes the hardware of the H8/3664 Series. For details on the H8/3664 Series instruction set, refer to the H8/300H Series Programming Manual.

Notes:

When using an on-chip emulator (E10T) for H8/3664 program development and debugging, the following restrictions must be noted.

1. The $\overline{\text{NMI}}$ pin is reserved for the E10T, and cannot be used.
2. Pins P85, P86, and P87 cannot be used. (In order to use these pins, additional hardware must be provided on the user board.)
3. Area H'7000 to H'7FFF is used by the E10T, and is not available to the user.
4. Area H'F780 to H'FB7F must on no account be accessed.
5. When the E10T is used, address breaks can be set as available to the user, or for use by the E10T. If address breaks are set as being used by the E10T, the address break control registers must not be accessed.
6. When the E10T is used, $\overline{\text{NMI}}$ is an input/output pin (open-drain in output mode), P85 and P87 are input pins, and P86 is an output pin.

HITACHI

Main Revisions and Additions in this Edition

Page	Item	Description
4	Figure 1.1 Block Diagram	$\overline{\text{TEST}}$ pin is amended to TEST pin
43	2.9.2 Notes on Bit Manipulation	Example 1 description added
54	3.4.2 Interrupt Edge Select Register 2 (IEGR2)	Bit 5 description amended
79	Figure 5.9 Pin Connection when not Using Subclock	Figure amended
88	Table 6.3 Transition Mode after the SLEEP Instruction Execution and Interrupt Handling	*1 description changed
102	Figure 7.4 User Program Mode	Figure amended
122	7.9 Flash Memory and Power-Down States	Description amended
	Table 7.10 Flash Memory Operating States	
179	Figure 11.2 Increment Timing with Internal Clock	Figure amended
281	14.5.1 Data Transfer Format	1st line, reference figure No. amended
322	Figure 15.5 I^2C Bus Timing	R/W is amended to R/$\overline{\text{W}}$
322 to 324	15.3.2 Master Transmit Operation	Description changed
	Figure 15.6 Example of Master Transmit Mode Operation Timing (MLS = WAIT = 0)	Figure amended
324 to 326	15.3.3 Master Receive Operation	Description changed
	Figure 15.7 Example of Master Receive Mode Operation Timing (1) (NLS = ACKB = 0, WAIT = 1)	Figure amended
	Figure 15.7 Example of Master Receive Mode Operation Timing (2) (NLS = ACKB = 0, WAIT = 1)	
326	15.3.4 Slave Receive Operation	R/W is amended to R/$\overline{\text{W}}$
327	Figure 15.8 Example of Slave Receive Mode Operation Timing (1) (MLS = ACKB = 0)	R/W is amended to R/$\overline{\text{W}}$
328	15.3.5 Slave Transmit Operation	Description amended
329	Figure 15.10 Example of Slave Transmit Mode Operation Timing (MLS = 0)	R/W is amended to R/$\overline{\text{W}}$
332	Figure 15.13 Flowchart for Master Transmit Mode (Example)	Flowchart changed
333	Figure 15.14 Flowchart for Master Receive Mode (Example)	Flowchart changed

Page	Item	Description
339, 340	15.4 Usage Notes • Notes on Start Condition Issuance for Retransmission Figure 15.17 Flowchart and Timing of Start Condition Instruction Issuance for Retransmission	Description added
348	16.2.3 A/D Control Register (ADCR)	Bit 7 Note added
367	Table 18.2 DC Characteristics (2)	Conditions changed
370	Table 18.4 I^2C Bus Interface Timing	Symbol in SCL and SDA output fall time amended
372, 373	Table 18.6 A/D Converter Characteristics	Min Value in AVcc amended Test Condition of Conversion time (single mode) amended
373	Table 18.7 Watchdog Timer Characteristics	Unit amended
376 to 388	18.3 Electrical Characteristics (Mask ROM Version)	Added
411 to 417	A.3 Number of Execution States	Added
423	B.2 Register Bits	Bit name in ABRKSR amended

Contents

Section 1 Overview

1.1 Features

Table 1.1 Features

Item	Description
CPU	H8/300H CPU (upward compatibility with H8/300 CPU at object level) • General-register machine — Sixteen 16-bit registers (also usable as eight 16-bit registers plus sixteen 8-bit registers or eight 32-bit registers) • High-speed operation — Max. operation speed: 16 MHz — Add/subtract: 0.125 µs — Multiply/divide: 0.875 µs • Address space: 64 kbytes • Instruction features — 8/16/32-bit data transfer, arithmetic, and logic instructions — Signed and unsigned multiply instructions (8 bits × 8 bits, 16 bits × 16 bits) — Signed and unsigned divide instructions (16 bits ÷ 8 bits, 32 bits ÷ 16 bits) — Bit accumulator function — Bit manipulation instructions with register-indirect specification of bit positions
Interrupts	• 11 external interrupt sources ($\overline{\text{NMI}}$, $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$, $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$) • 20 internal interrupt sources
Clock pulse generators	• System clock pulse generator: 1 to 16 MHz • Sub-system clock pulse generator: 32.768 kHz (for watch)
Power-down modes	Transition possible between five modes • Active mode • Sleep mode • Standby mode • Subsleep mode • Subactive mode Gear function • Module standby function

Item	Description			
Memory				
	Type No.		**ROM**	**RAM**
	HD64F3664	(Flash memory version)	32 kbytes	2,048 bytes
	HD6433664	(Mask ROM version)	32 kbytes	1,024 bytes
	HD6433663	(Mask ROM version)	24 kbytes	1,024 bytes
	HD6433662	(Mask ROM version)	16 kbytes	512 bytes
	HD6433661	(Mask ROM version)	12 kbytes	512 bytes
	HD6433660	(Mask ROM version)	8 kbytes	512 bytes
I/O ports	• 29 I/O pins, including 8 large current ports (I_{OL} = 20 mA, @ V_{OL} = 1.5 V • 8 input pins (also used for analog input)			
Timers	• Timer A: 8-bit timer Count-up timer with selection of eight internal clock signals divided from the system clock and four clock signals divided from the watch sub-clock • 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 — Externally triggerable • Timer W: 16-bit timer — Counts any of four internal clock signals or external events — Maximum of four types of pulses can be input or output and processed — Output compare/input capture (4 output pins) — Output compare/input capture operation can be buffered — PWM mode can be set (maximum of three synchronous outputs) • Watchdog timer: 8-bit timer — Reset signal generated by counter overflow — Operates independent from system clock by internal oscillation circuit			
Serial communication interface	• Selectable between asynchronous mode or 8-bit clock synchronous mode • Incorporate baud rate generator • Multi-processor communication function (asynchronous)			

Item	Description
I^2C bus interface	• Conforms to I^2C bus interface proposed by Philips Electronics • Selectable between single master mode and slave mode • Supports two slave addresses
A/D converter	• 10-bit resolution • 8-channel analog input pins (selectable between single mode and scan mode) • Conversion time: 7 µs • Sample and hold function
Package	

Code	Body Size	Pin Pitch
QFP-64 (FP-64E)	10.0 × 10.0 mm	0.5 mm
QFP-64 (FP-64A)	14.0 × 14.0 mm	0.8 mm
SDIP-42 (DP-42S)	14.0 × 37.3 mm	1.78 mm

1.2 Internal Block Diagram

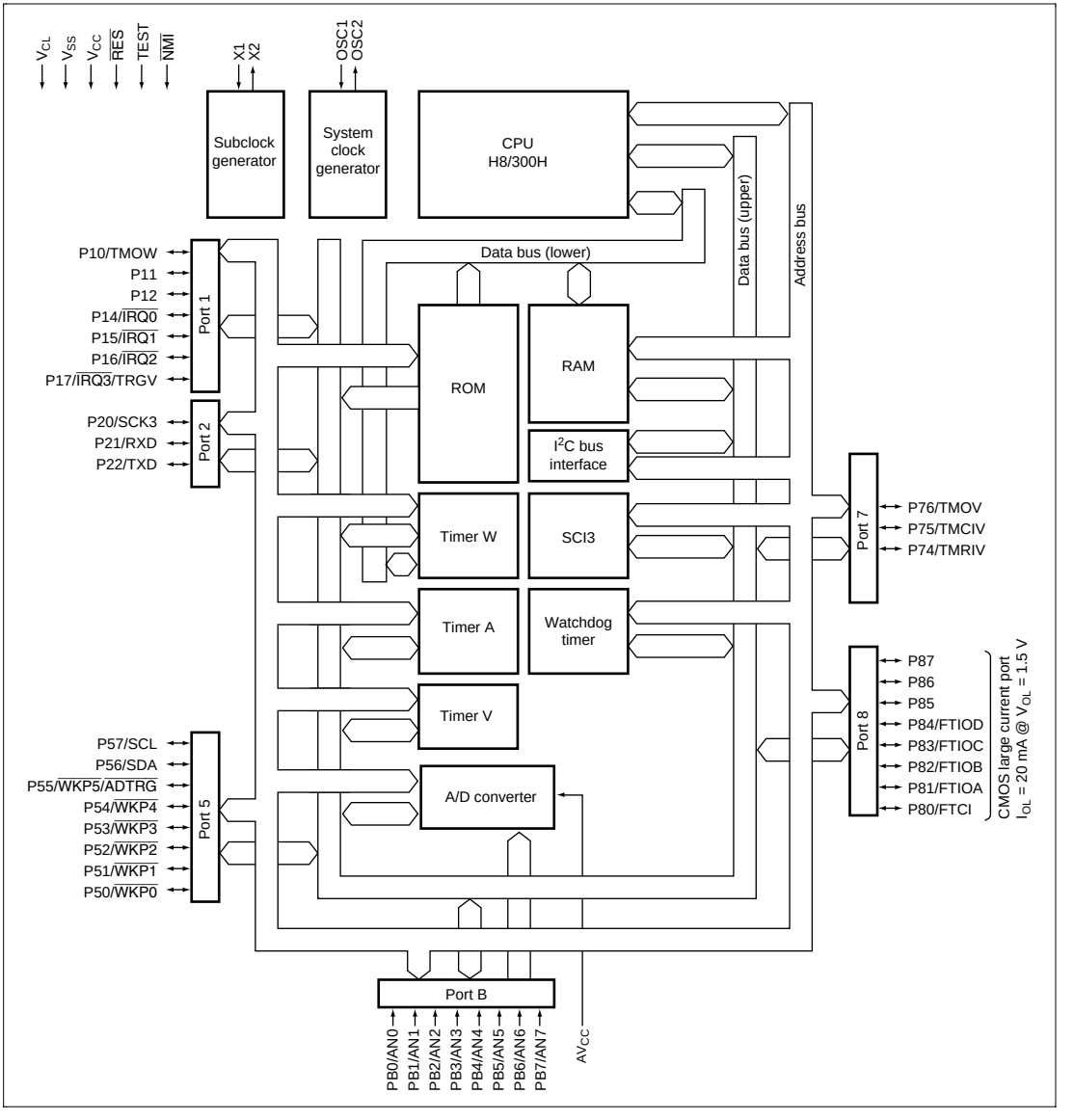


Figure 1.1 Block Diagram

1.3 Pin Arrangement

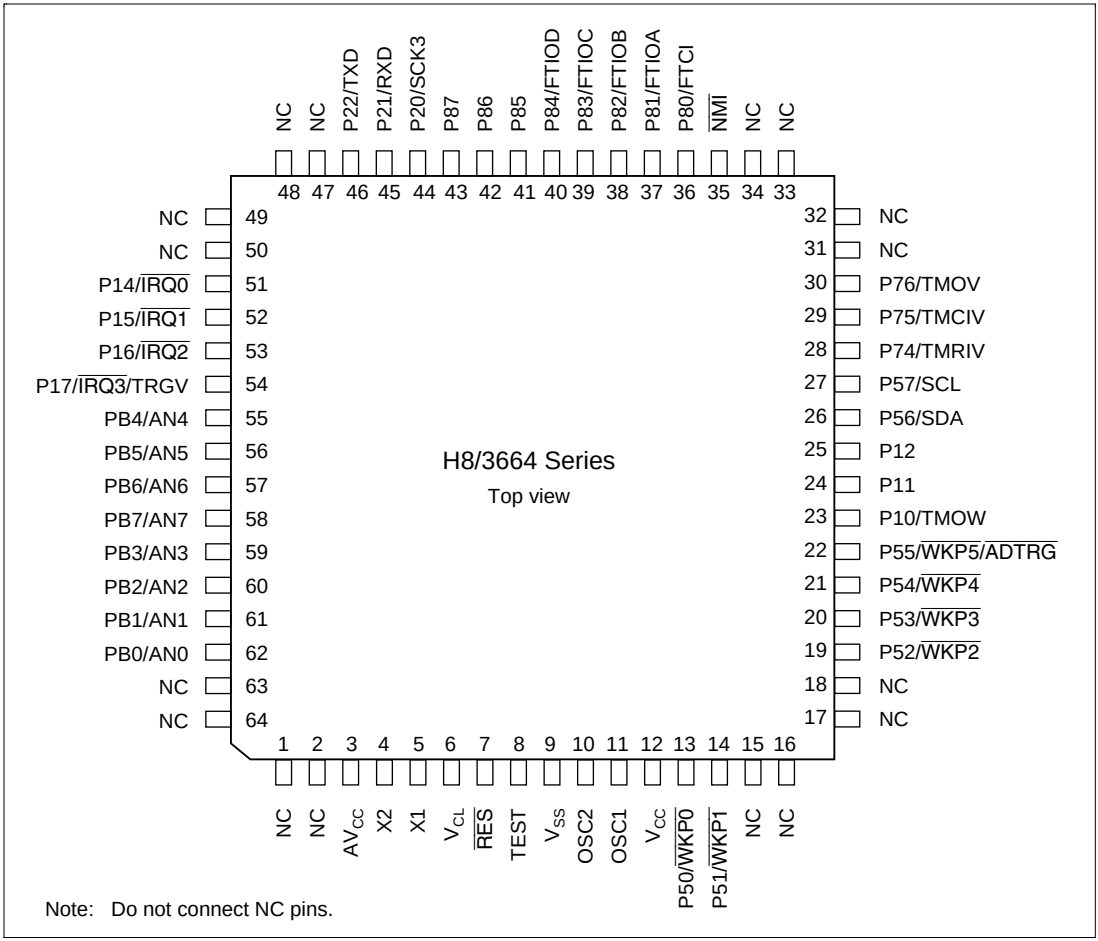
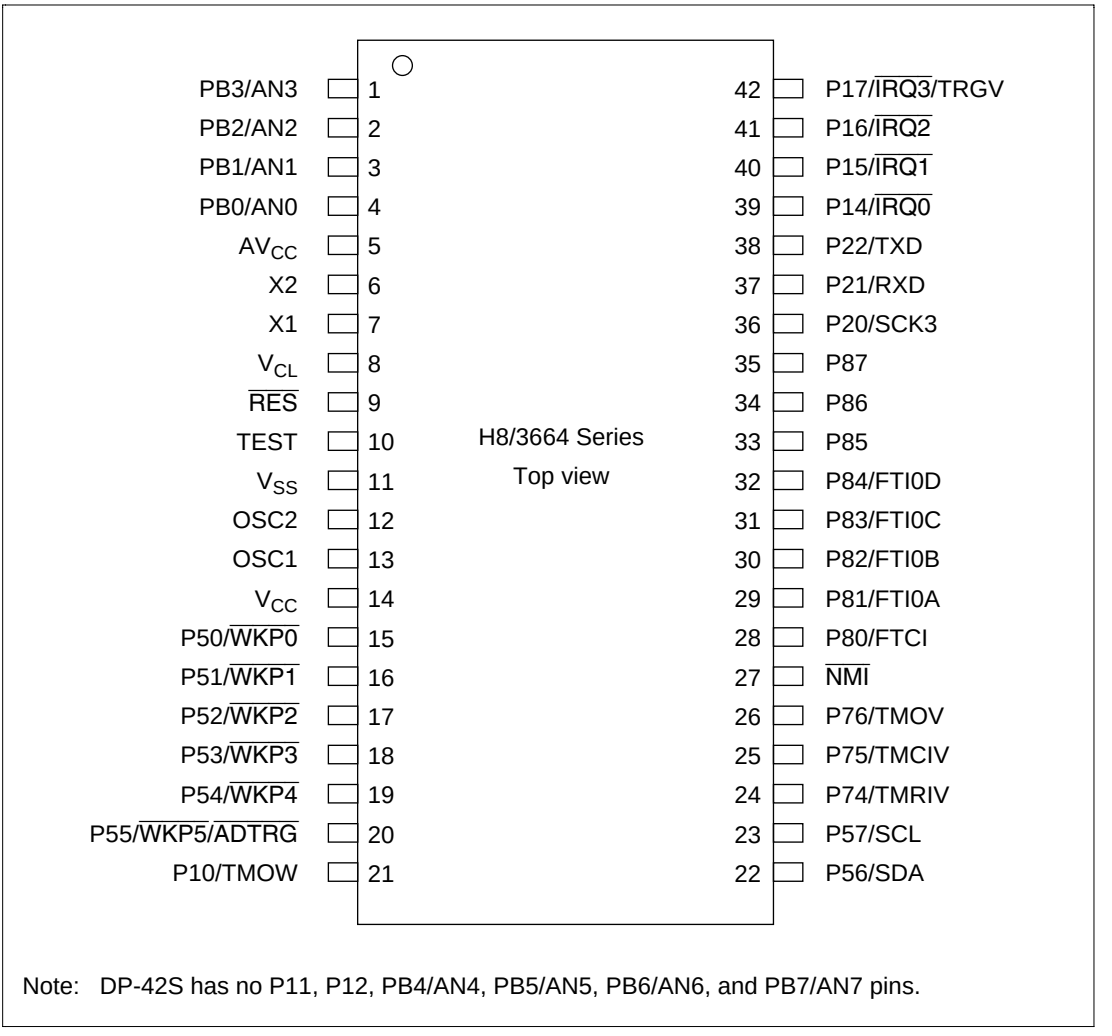


Figure 1.2 Pin Arrangement (FP-64E, FP-64A)



Note: DP-42S has no P11, P12, PB4/AN4, PB5/AN5, PB6/AN6, and PB7/AN7 pins.

Figure 1.3 Pin Arrangement (DP-42S)

1.4 Pin Functions

Table 1.2 Pin Functions

Type	Symbol	Pin No. FP-64E FP-64A	Pin No. DP-42S	I/O	Name and Functions
Power source pins	V_{CC}	12	14	Input	**Power supply:** All V_{CC} pins should be connected to the user system V_{CC}.
	V_{SS}	9	11	Input	**Ground:** All V_{SS} pins should be connected to the user system GND (0 V).
	AV_{CC}	3	5	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 user system V_{CC}.
	V_{CL}	6	8	Input	**Internal step-down power supply:** Connect a capacitor of around 0.1 µF between this pin and the V_{SS} pin for stabilization.
Clock pins	OSC1	11	13	Input	**System clock:** These pins connect to a crystal or ceramic oscillator, or can be used to input an external clock.
	OSC2	10	12	Output	See section 5, Clock Pulse Generators, for a typical connection diagram.
	X1	5	7	Input	**Subclock:** These pins connect to a 32.768-kHz crystal oscillator.
	X2	4	6	Output	See section 5, Clock Pulse Generators, for a typical connection diagram.
System control	$\overline{RES}$	7	9	Input	**Reset:** When this pin is driven low, the chip is reset.
	TEST	8	10	Input	**Test:** This is a test pin, not for use in application systems. It should be connected to V_{SS}.

Type	Symbol	Pin No. FP-64E FP-64A	Pin No. DP-42S	I/O	Name and Functions
Interrupt pins	$\overline{\text{NMI}}$	35	27	Input	**Non-maskable interrupt request input pin**
	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$	51 to 54	39 to 42	Input	**IRQ interrupt request 0 to 3:** These are input pins for edge-sensitive external interrupts, with a selection of rising or falling edge.
	$\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$	13, 14, 19 to 22	15 to 20	Input	**WKP interrupt request 0 to 5:** These are input pins for edge-sensitive external interrupts, with a selection of rising or falling edge.
Timer A	TMOW	23	21	Output	**Clock output:** This is an output pin for waveforms generated by the timer A output circuit.
Timer V	TMOV	30	26	Output	**Timer V output:** This is an output pin for waveforms generated by the timer V output compare function.
	TMCIV	29	25	Input	**Timer V event input:** This is an event input pin for input to the timer V counter.
	TMRIV	28	24	Input	**Timer V counter reset:** This is a counter reset input pin for timer V.
	TRGV	54	42	Input	**Timer V counter trigger input:** This is a trigger input pin for the timer V counter.
Timer W	FTCI	36	28	Input	**Timer W clock input:** This is an external clock input pin for input to the timer X counter.
	FTIOA to FTIOD	37 to 40	29 to 32	I/O	**Timer W output compare A/input capture/PWM output pin**
I^2C bus inerface	SDA	26	22	I/O	**I^2C data I/O:** Can directly drive a bus by NMOS open-drain output.
	SCL	27	23	I/O	**I^2C clock I/O:** Can directly drive a bus by NMOS open-drain output.

Type	Symbol	Pin No. FP-64E FP-64A	Pin No. DP-42S	I/O	Name and Functions
Serial communication interface (SCI)	TXD	46	38	Output	**SCI3 transmit data output:** This is the data output pin.
	RXD	45	37	Input	**SCI3 receive data input:** This is the data input pin.
	SCK3	44	36	Output	**SCI3 clock I/O:** This is the clock I/O pin.
A/D converter	AN7 to AN0	55 to 62	1 to 4	Input	**Analog input channels 7 to 0:** These are analog data input channels to the A/D converter.
	$\overline{\text{ADTRG}}$	22	20	Input	**A/D converter trigger input:** This is the external trigger input pin to the A/D converter.
I/O ports	PB7 to PB0	55 to 62	1 to 4	Input	**Port B:** This is an 8-bit input port.
	P17 to P14, P12 to P10	51 to 54, 23 to 25	39 to 42, 21	I/O	**Port 1:** This is a 7-bit I/O port.
	P22 to P20	44 to 46	36 to 38	I/O	**Port 2:** This is a 3-bit I/O port.
	P57 to P50	13, 14, 19 to 22, 26, 27	15 to 20, 22, 23	I/O	**Port 5:** This is an 8-bit I/O port.
	P76 to P74	28 to 30	24 to 26	I/O	**Port 7:** This is a 3-bit I/O port.
	P87 to P80	36 to 43	28 to 35	I/O	**Port 8:** This is an 8-bit I/O port.
Other	NC				**Non-connected pins:** These pins must be left unconnected.

Section 2 CPU

2.1 Features

The H8/3664 Series has an H8/300H CPU with an internal 32-bit architecture that is upward-compatible with the H8/300 CPU, and supports only normal mode, which has a 64-kbyte address space.

The H8/300H CPU has the following features.

- Upward compatibility with H8/300 CPU
  - — Can execute H8/300 Series object programs
  - — Additional eight 16-bit extended registers
  - — 32-bit transfer and arithmetic and logic instructions are added
  - — Signed multiply and divide instructions are added
- General registers
  - — Sixteen 16-bit general registers (also usable as sixteen 8-bit registers and eight 16-bit registers or eight 32-bit registers)
- Sixty-two basic instructions
  - — 8/16/32-bit data transfer and arithmetic and logic instructions
  - — Multiply and divide instructions
  - — Powerful bit-manipulation instructions
- Eight addressing modes
  - — Register direct [Rn]
  - — Register indirect [@ERn]
  - — Register indirect with displacement [@(d:16, ERn) or @(d:24, ERn)]
  - — Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - — Absolute address [@aa:8, @aa:16, or @aa:24]
  - — Immediate [#xx:8, #xx:16, or #xx:32]
  - — Program-counter relative [@(d:8, PC) or @(d:16, PC)]
  - — Memory indirect [@@aa:8]
- 64-kbyte address space
- High-speed operation
  - — All frequently-used instructions execute in two to four states
  - — Maximum clock frequency: 16 MHz
  - — 8/16/32-bit register-register add/subtract: 2 states
  - — 8 × 8-bit register-register multiply: 14 states
  - — 16 ÷ 8-bit register-register divide: 14 states

— 16 × 16-bit register-register multiply: 22 states
— 32 ÷ 16-bit register-register divide: 22 states

- Low-power mode
  Transition to low-power state by SLEEP instruction

2.2 Address Space and Memory Map

The address space of the H8/3664 Series CPU is 64 kbytes, which includes the program area and the data area.

Figures 2.1 and 2.2 show the memory map.

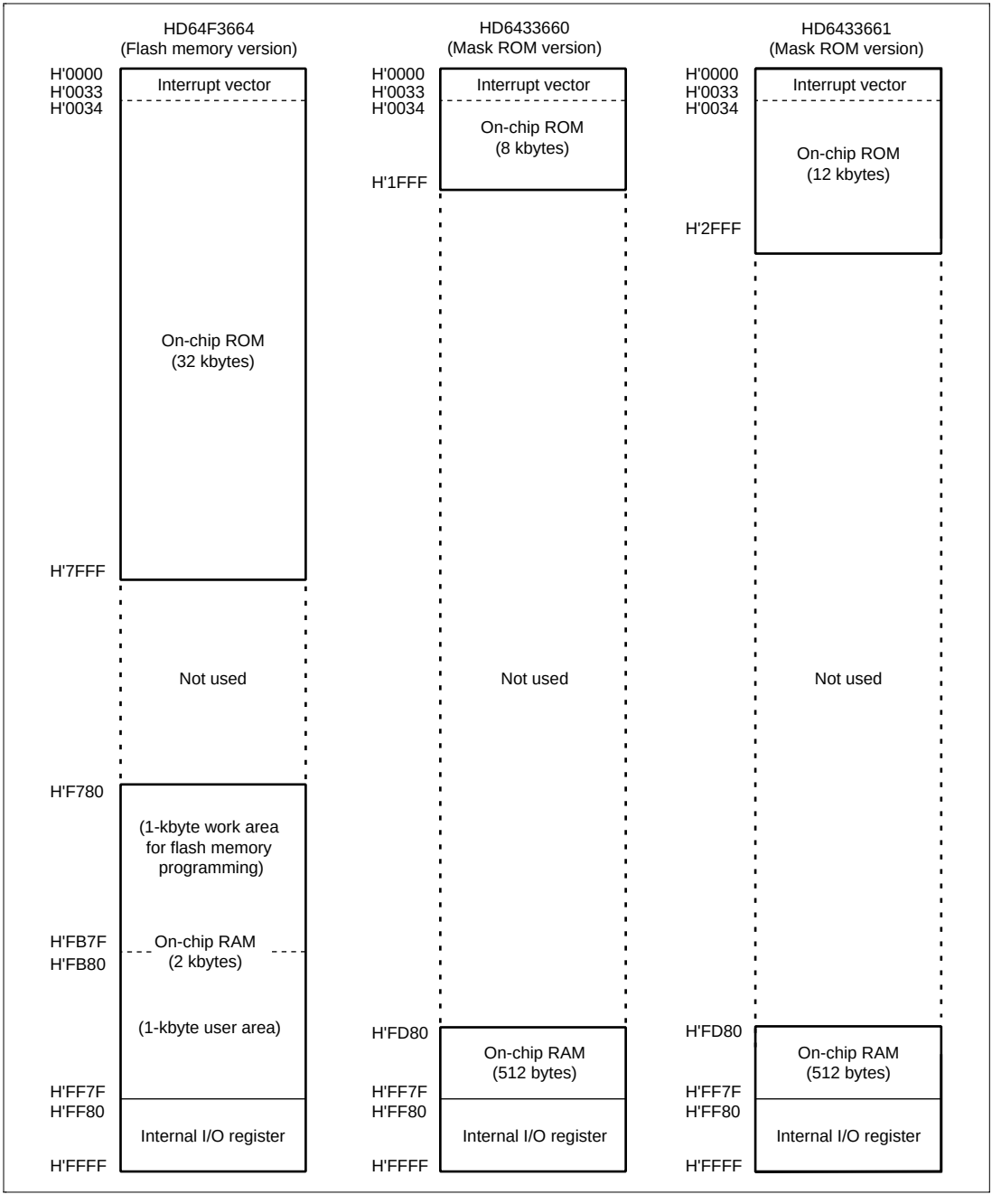


Figure 2.1 Memory Map (1)

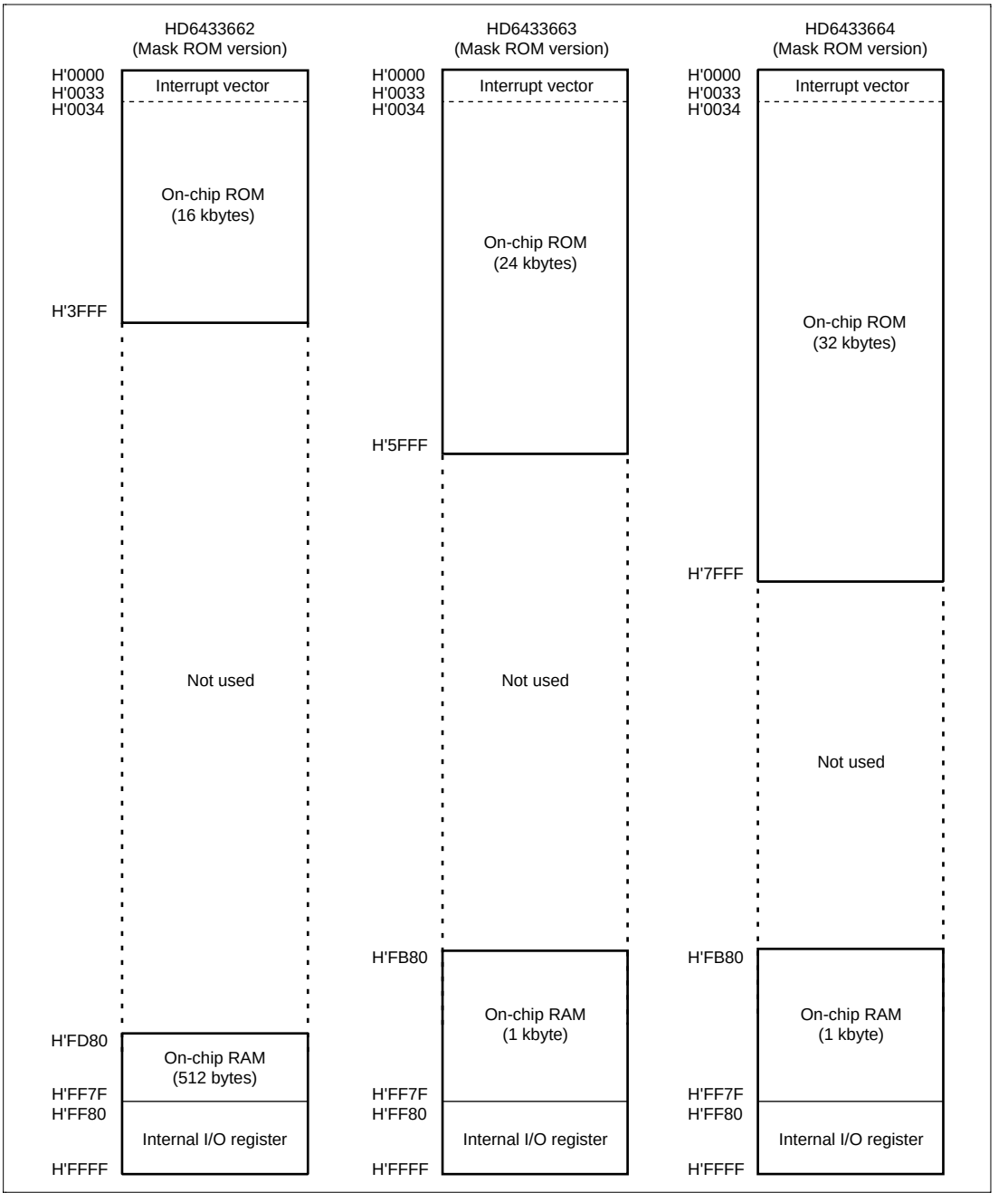


Figure 2.2 Memory Map (2)

2.3 Register Configuration

The H8/300H CPU has the internal registers shown in figure 2.3. There are two types of registers: general registers and control registers. Control registers are 24-bit program counter (PC) and 8-bit condition code register (CCR).

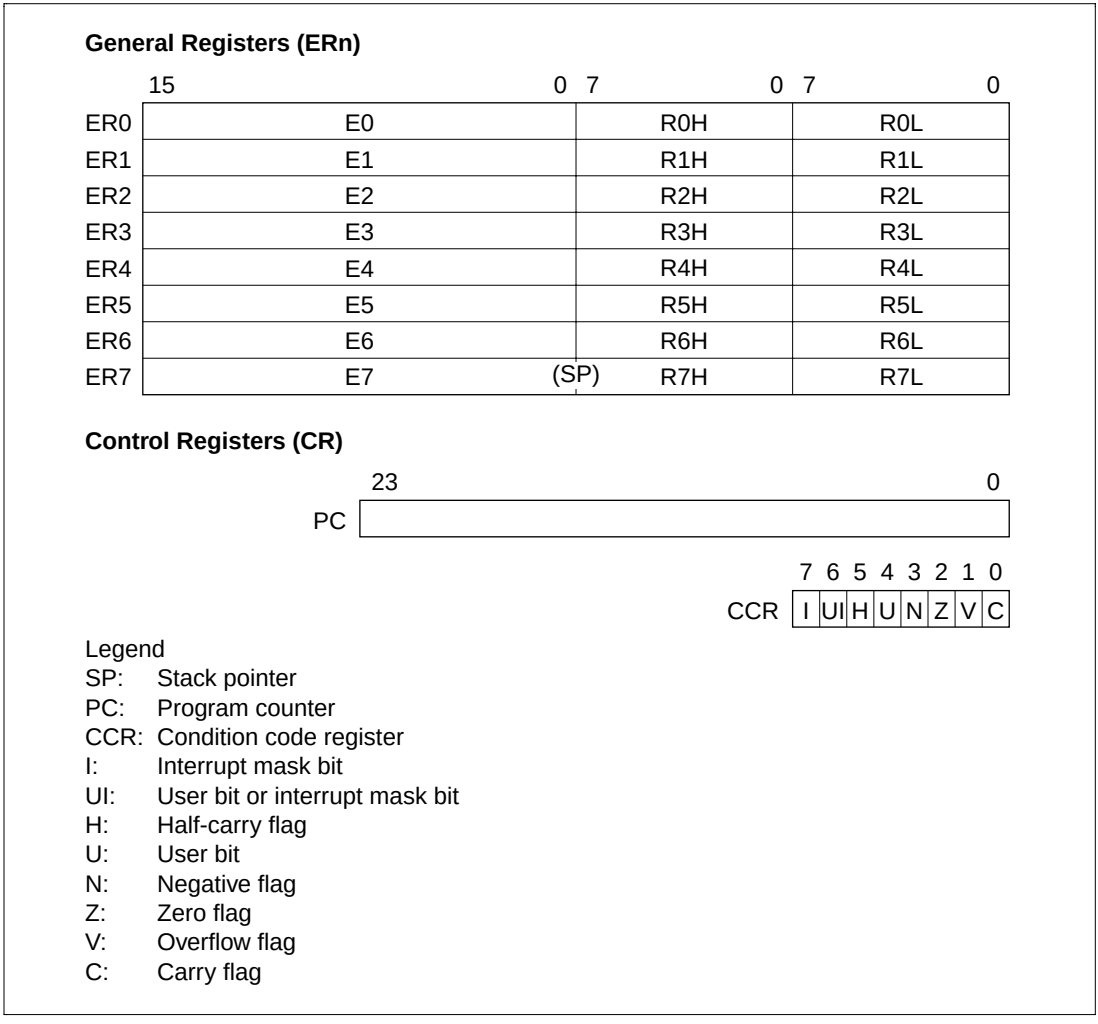


Figure 2.3 CPU Internal Registers

2.3.1 General Registers

The H8/300H CPU has eight 32-bit general registers. These general registers are all functionally alike and can be used without distinction between data registers and address registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. When the general registers are used as 32-bit registers or as address registers, they are designated by the letters ER (ER0 to ER7).

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

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

Figure 2.4 illustrates the usage of the general registers. The usage of each register can be selected independently.

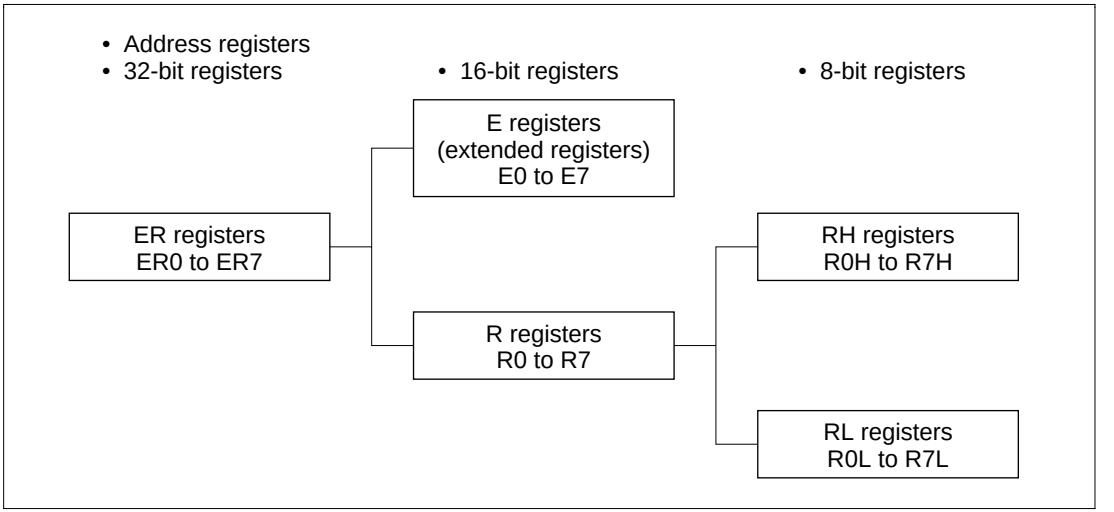


Figure 2.4 Usage of General Registers

General register ER7 has the function of stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2.5 shows the stack.

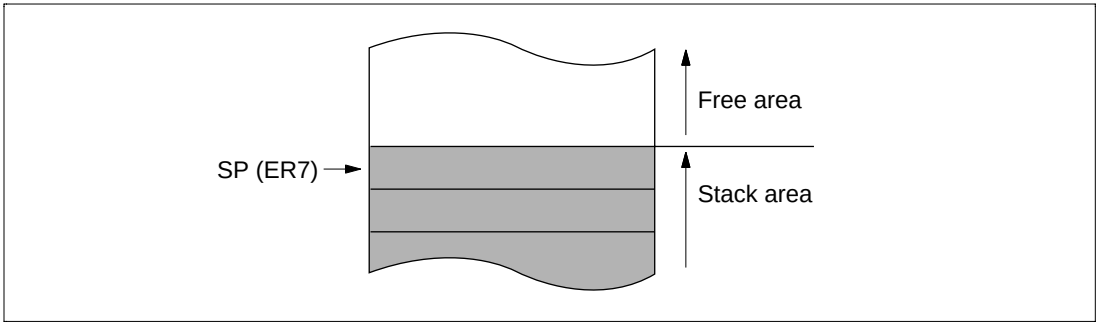


Figure 2.5 Relationship between Stack Pointer and Stack Area

2.3.2 Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 2 bytes (one word) or a multiple of 2 bytes, so the least significant PC bit is ignored. When an instruction is fetched, the least significant PC bit is regarded as 0. The PC is initialized when the start address is loaded by the vector address generated during reset exception-handling sequence.

2.3.3 Condition Code Register (CCR)

This 8-bit register contains internal CPU status information, including the interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags. The I bit is initialized to 1 by reset exception-handling sequence, but other bits are not initialized.

Bit 7—Interrupt Mask Bit (I): Masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting.

Bit 6—User Bit (UI): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.

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

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

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

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

Bit 1—Overflow Flag (V): Set to 1 when an arithmetic overflow occurs, and cleared to 0 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 flag bits unchanged. Operations can be performed on CCR by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used by conditional branch (Bcc) instructions.

For the action of each instruction on the flag bits, see appendix A.1, Instruction List.

2.4 Data Formats

The H8/300H CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit n (n = 0, 1, 2, …, 7) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

2.4.1 General Register Data Formats

Figures 2.6 and 2.7 show the data formats in general registers.

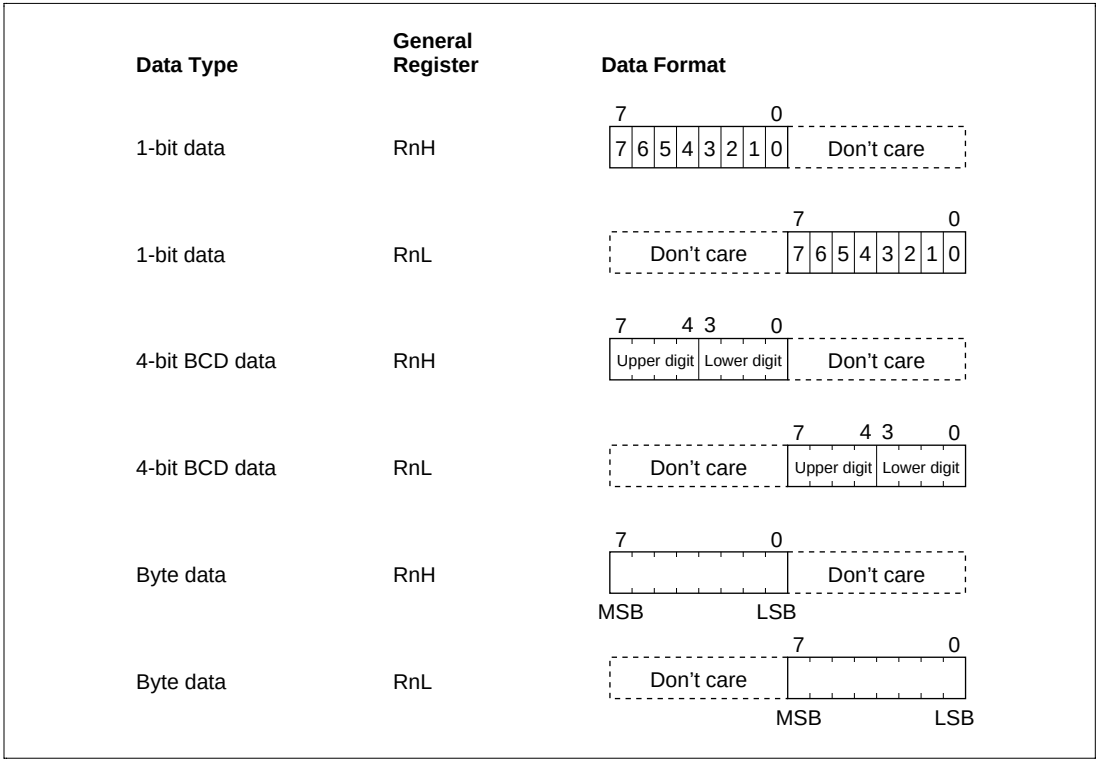
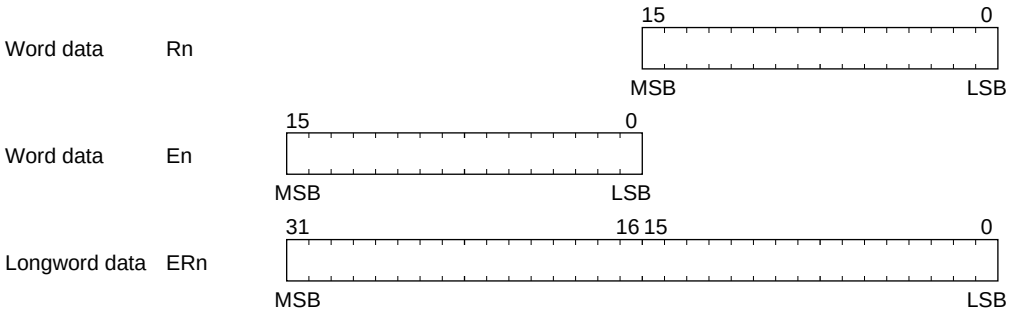


Figure 2.6 General Register Data Formats (1)



Notation
ERn: General register
En: General register E
Rn: General register R
RnH: General register RH
RnL: General register RL
MSB: Most significant bit
LSB: Least significant bit

Figure 2.7 General Register Data Formats (2)

2.4.2 Memory Data Formats

Figure 2.8 shows the data formats on memory. The H8/300H CPU can access word data and longword data on memory, but word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, no address error occurs but the least significant bit of the address is regarded as 0, so the access starts at the preceding address. This also applies to instruction fetches.

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

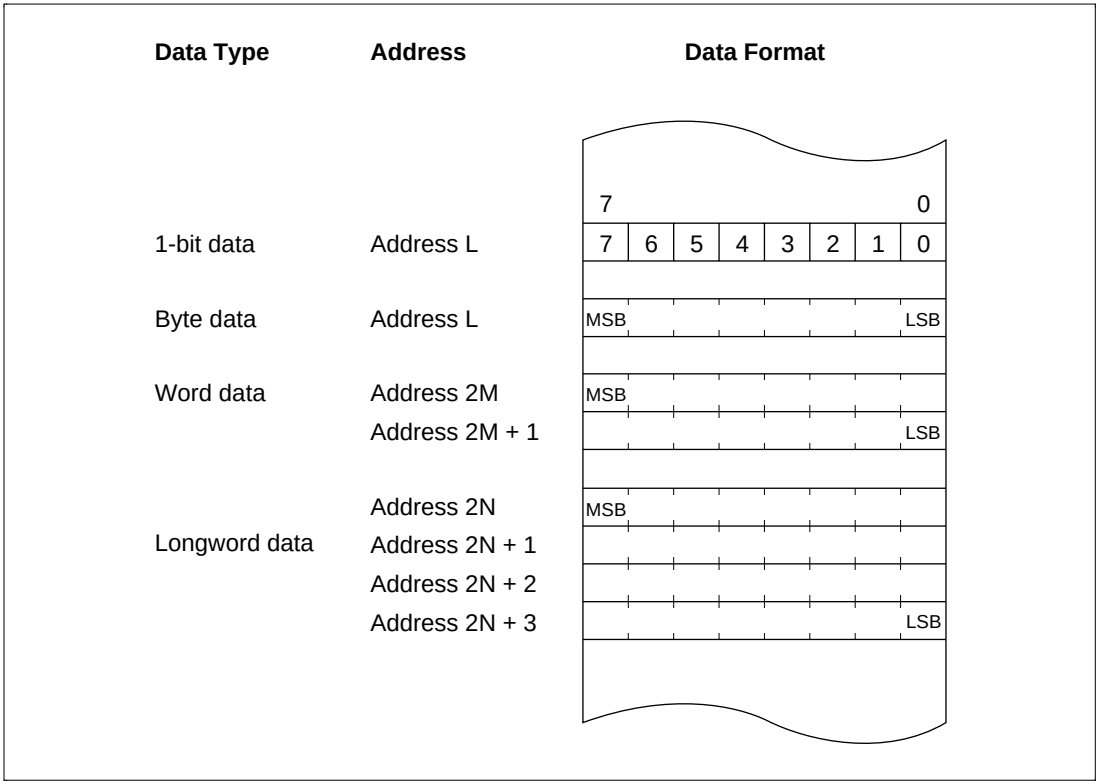


Figure 2.8 Memory Data Formats

2.5 Instruction Set

2.5.1 Instruction Set Overview

The H8/300H CPU has 62 types of instructions. Tables 2.1 to 2.8 summarize the instructions in each functional category. The operation notation used in these tables is defined next.

Operation Notation

Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register or address register)
(EAd)	Destination operand
(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
¬	NOT (logical complement)
:3/:8/:16/:24	3-, 8-, 16-, or 24-bit length

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

Table 2.1 Data Transfer Instructions

Instruction	Size*	Function
MOV	B/W/L	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
MOVFPE	B	(EAs) → Rd Cannot be used in the H8/3664 Series.
MOVTPE	B	Rs → (EAs) Cannot be used in the H8/3664 Series.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. Similarly, POP.L ERn is identical to MOV.L @SP+, ERn.
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @–SP. Similarly, PUSH.L ERn is identical to MOV.L ERn, @–SP.

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

Table 2.2 Arithmetic Operation Instructions

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

Instruction	Size*	Function
NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic complement) of data in a general register.
EXTS	W/L	Rd (sign extension) → Rd Extends byte data in the lower 8 bits of a 16-bit register to word data, or extends word data in the lower 16 bits of a 32-bit register to longword data, by extending the sign bit.
EXTU	W/L	Rd (zero extension) → Rd Extends byte data in the lower 8 bits of a 16-bit register to word data, or extends word data in the lower 16 bits of a 32-bit register to longword data, by padding with zeros.

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

Table 2.3 Logic Operation Instructions

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

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

Table 2.4 Shift Instructions

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

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

Table 2.5 Bit Manipulation Instructions

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

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

Note: * Size refers to the operand size.
B: Byte

Table 2.6 Branching Instructions

Instruction	Size	Function
Bcc*	—	Branches to a specified address if a specified condition is true. The branching conditions are listed below.

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

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

Note: * Conditional branch instructions are generally called the Bcc instructions.

Table 2.7 System Control Instructions

Instruction	Size*	Function
TRAPA	—	Starts trap-instruction exception handling
RTE	—	Returns from an exception-handling routine
SLEEP	—	Causes a transition to the power-down state
LDC	B/W	(EAs) → CCR Moves the source operand contents to the condition code register. The condition code register size is one byte, but in transfer from memory, data is read by word access.
STC	B/W	CCR → (EAd) Transfers the CCR contents to a destination location. The condition code register size is one byte, but in transfer to memory, data is written by word access.
ANDC	B	CCR ∧ #IMM → CCR Logically ANDs the condition code register with immediate data.
ORC	B	CCR ∨ #IMM → CCR Logically ORs the condition code register with immediate data.
XORC	B	CCR ⊕ #IMM → CCR Logically exclusive-ORs the condition code register with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

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

Table 2.8 Block Transfer Instruction

Instruction	Size	Function
EEPMOV.B	—	if R4L ≠ 0 then repeat @ER5+ → @ER6+, R4L – 1 → R4L until R4L = 0 else next;
EEPMOV.W	—	if R4 ≠ 0 then repeat @ER5+ → @ER6+, R4 – 1 → R4 until R4 = 0 else next; Transfers a data block according to parameters set in general registers R4L or R4, ER5, and ER6. R4L or R4: Size of block (bytes) ER5: Starting source address ER6: Starting destination address Execution of the next instruction begins as soon as the transfer is completed.

2.5.2 Basic Instruction Formats

The H8/300H instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (OP field), a register field (r field), an effective address extension (EA field), and a condition field (cc field).

Operation Field: Indicates the function of the instruction, the addressing mode, and the operation to be carried out on the operand. The operation field always includes the first 4 bits of the instruction. Some instructions have two operation fields.

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

Effective Address Extension: 8, 16, or 32 bits specifying immediate data, an absolute address, or a displacement. A 24-bit address or displacement is treated as 32-bit data in which the first 8 bits are 0 (H'00).

Condition Field: Specifies the branching condition of Bcc instructions.

Figure 2.9 shows examples of instruction formats.

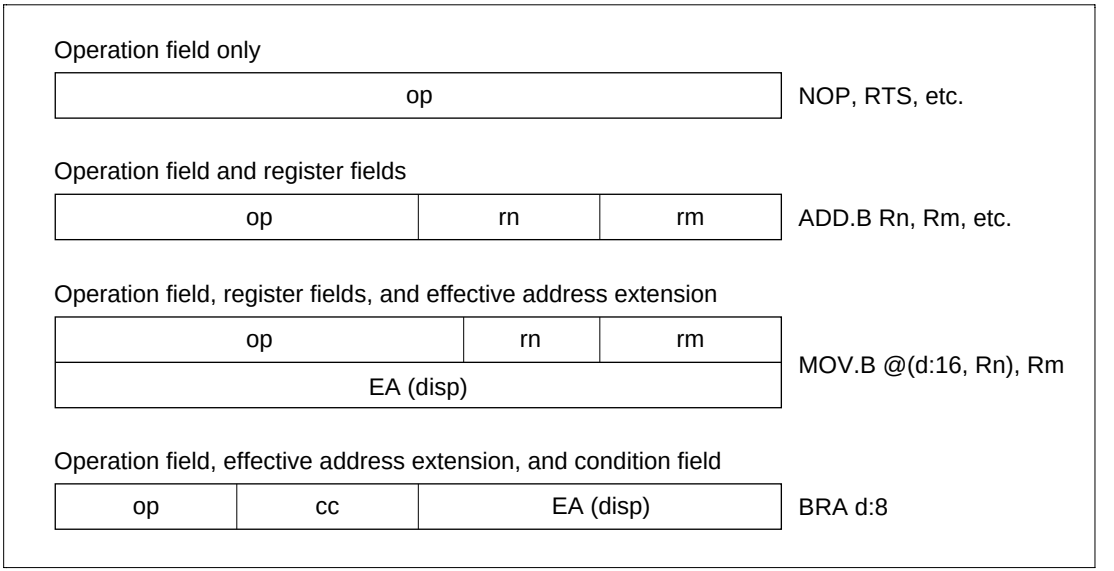


Figure 2.9 Instruction Formats

2.6 Addressing Modes and Effective Address Calculation

The following describes the H8/300H CPU. In the H8/3664 Series, the upper eight bits are ignored in the generated 24-bit address, so the effective address is 16 bits.

2.6.1 Addressing Modes

The H8/300H CPU supports the eight addressing modes listed in table 2.9. Each instruction uses a subset of these addressing modes. Addressing modes that can be used differ depending on the instruction. For details, refer to appendix A, CPU Instruction Set. Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions use register direct, register indirect, or absolute (@aa:8) addressing mode to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

Table 2.9 Addressing Modes

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

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

2 Register Indirect—@ERn: The register field of the instruction code specifies an address register (ERn), the lower 24 bits of which contain the address of the operand.

3 Register Indirect with Displacement—@(d:16, ERn) or @(d:24, ERn): A 16-bit or 24-bit displacement contained in the instruction code is added to the contents of an address register (ERn) specified by the register field of the instruction, and the lower 24 bits of the sum specify the address of a memory operand. A 16-bit displacement is sign-extended when added.

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

- Register indirect with post-increment—@ERn+

  The register field of the instruction code specifies an address register (ERn) the lower 24 bits of which contain the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents (32 bits) and the sum is stored in the address register. The value added is 1 for byte access, 2 for word access, or 4 for longword access. For word or longword access, the register value should be even.

- Register indirect with pre-decrement—@–ERn

  The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the register field in the instruction code, and the lower 24 bits of the result become the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word access, or 4 for longword access. For word or longword access, the resulting register value should be even.

5 Absolute Address—@aa:8, @aa:16, or @aa:24: The instruction code contains the absolute address of a memory operand. The absolute address may be 8 bits long (@aa:8), 16 bits long (@aa:16), or 24 bits long (@aa:24). For an 8-bit absolute address, the upper 16 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address the upper 8 bits are a sign extension. A 24-bit absolute address can access the entire address space. Table 2.10 indicates the accessible address ranges.

Table 2.10 Absolute Address Access Ranges

Absolute Address	Access Range
8 bits (@aa:8)	H'FF00 to H'FFFF
16 bits (@aa:16)	H'0000 to H'FFFF
24 bits (@aa:24)	H'0000 to H'FFFF

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

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

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

8 Memory Indirect—@@aa:8: This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand. This memory operand contains a branch address. The memory operand is accessed by longword access. The first byte of the memory operand is ignored, generating a 24-bit branch address. See figure 2.10. The upper bits of the 8-bit absolute address are assumed to be 0 (H'0000), so the address range is 0 to 255 (H'0000 to H'00FF). Note that the first part of this range is also the exception vector area.

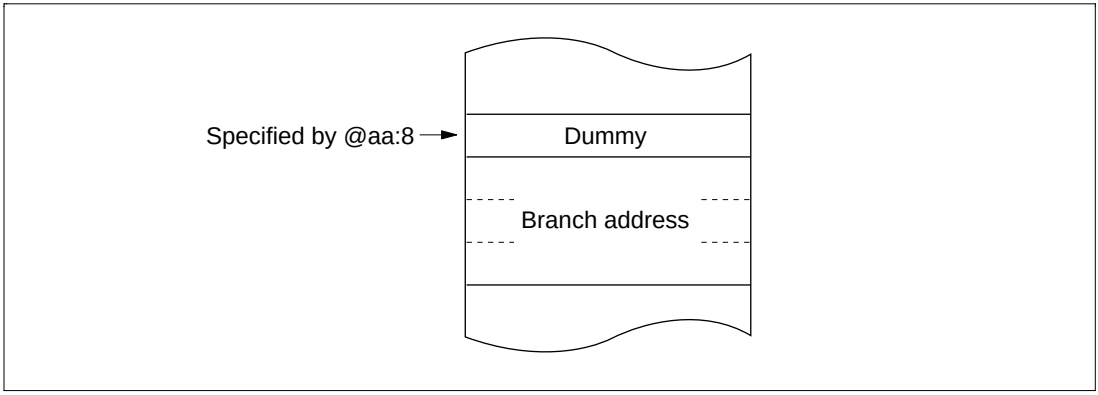
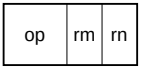
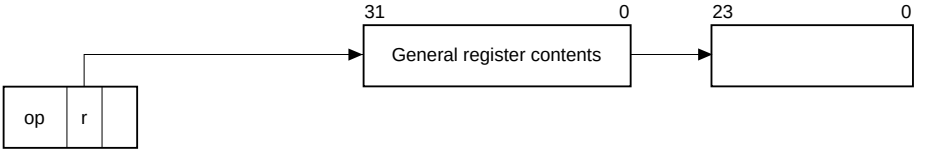
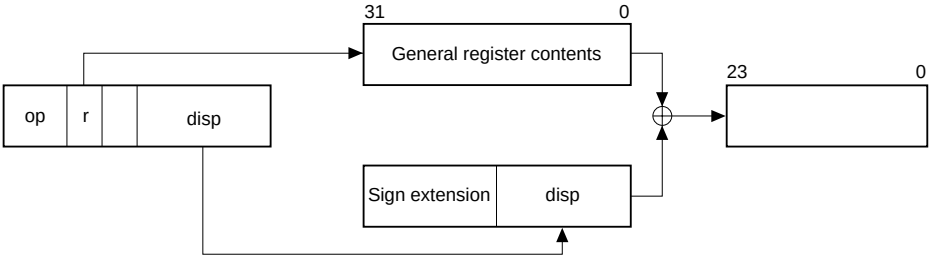
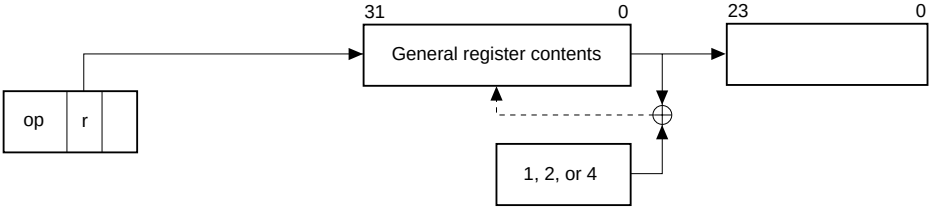
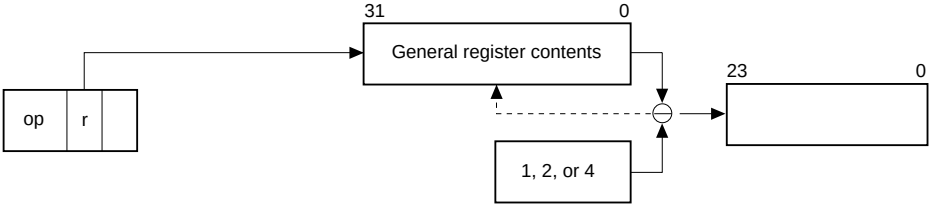


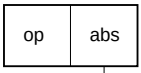
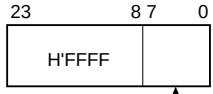
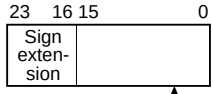
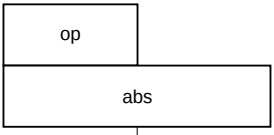
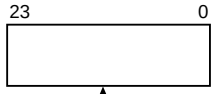
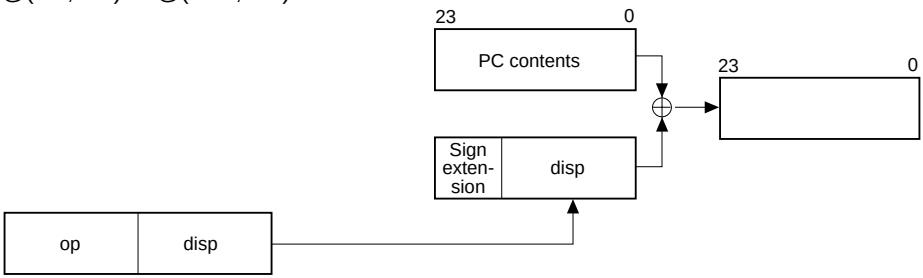
Figure 2.10 Memory-Indirect Branch Address Specification

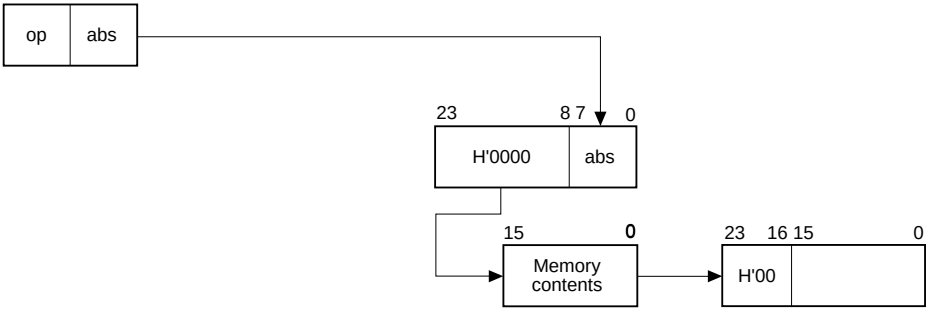
2.6.2 Effective Address Calculation

Table 2.11 explains how an effective address (EA) is calculated in each addressing mode. In the H8/3664 Series, the upper 8 bits of the calculated address are ignored in order to generate a 16-bit effective address.

Table 2.11 Effective Address Calculation

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address
1	Register direct (Rn) 		Operand is general register contents
2	Register indirect (@ERn) 		
3	Register indirect with displacement @(d:16, ERn)/@(d:24, ERn) 		
4	Register indirect with post-increment or pre-decrement Register indirect with post-increment @ERn+  Register indirect with pre-decrement @–ERn 		1 for a byte operand, 2 for a word operand, 4 for a longword operand

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

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

Legend:

r, rm, rn: Register field

op: Operation field

disp: Displacement

IMM: Immediate data

abs: Absolute address

Note: In the H8/3664 Series, the upper 8 bits of the calculation result are ignored.

2.7 Basic Bus Cycle

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.

2.7.1 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.

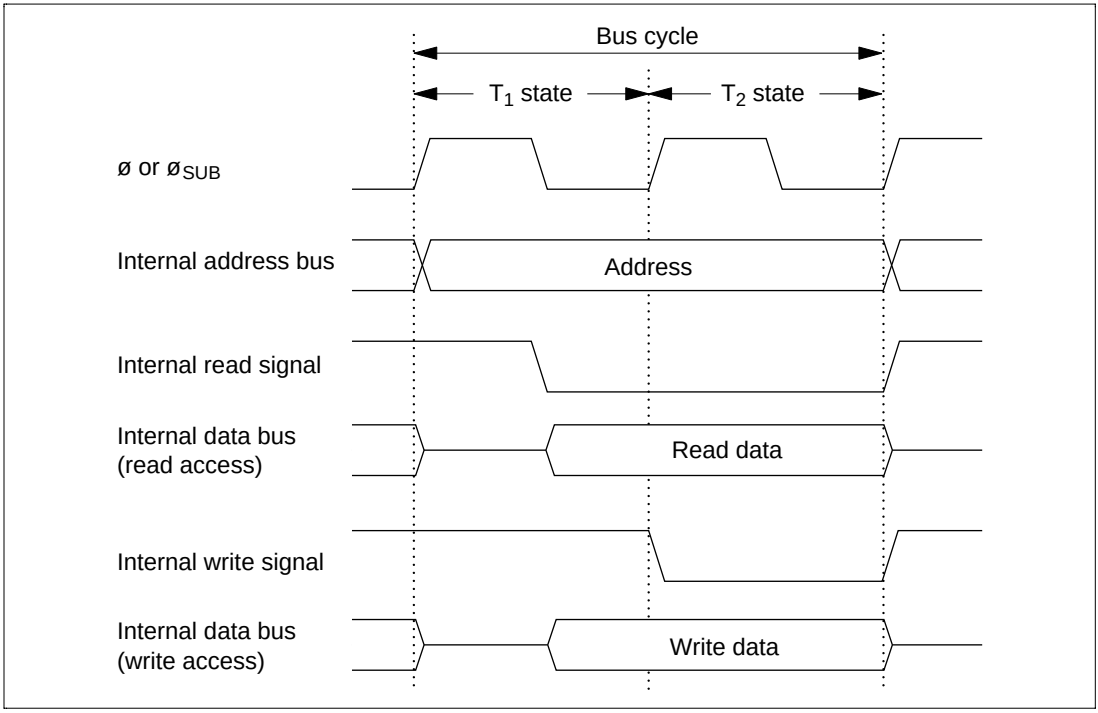


Figure 2.11 On-Chip Memory Access Cycle

2.7.2 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 or 16 bits depending on the register. For description on the data bus width of each register, refer to appendix B, Internal I/O Registers. Registers with 16-bit data bus width can be accessed by word size only. Registers with 8-bit data bus width can be accessed by byte or word size. When a register with 8-bit data bus width is accessed by word size, access is completed in two cycles. In two-state access, the operation timing is the same as that for on-chip memory.

Figure 2.12 shows the operation timing in the case of three-state access to an on-chip peripheral module.

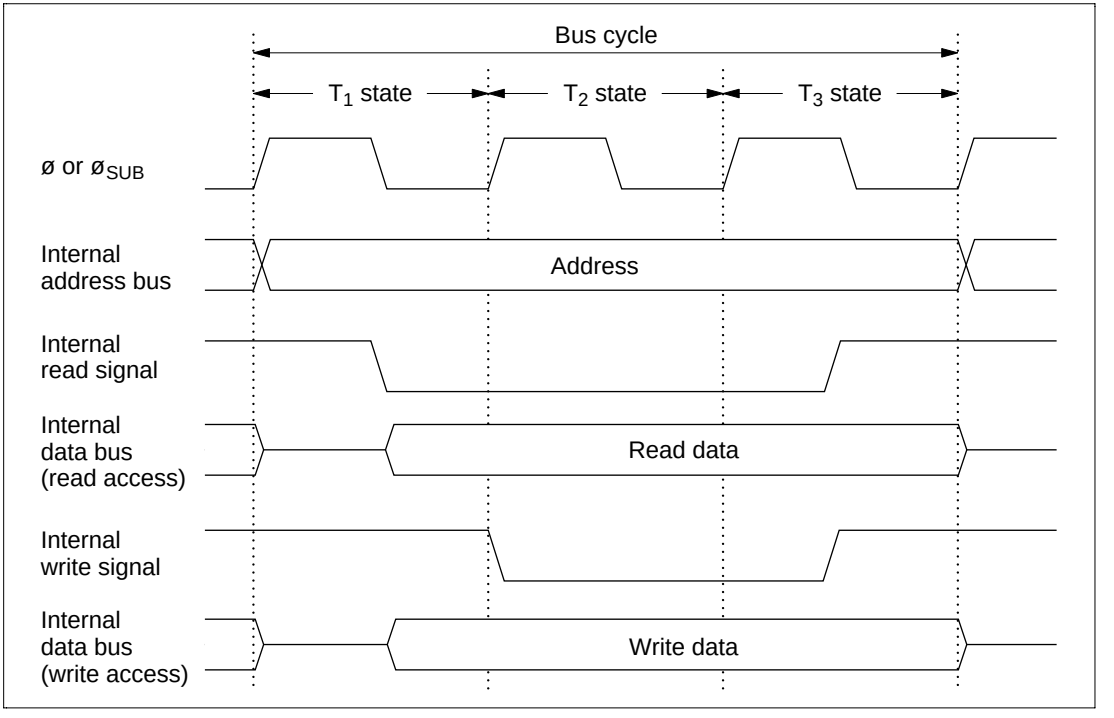


Figure 2.12 On-Chip Peripheral Module Access Cycle (3-State Access)

2.8 CPU States

2.8.1 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 mode and subactive mode. In the program halt state there are a sleep mode, standby mode, and sub-sleep mode. These states are shown in figure 2.13. Figure 2.14 shows the state transitions. For details on program execution state and program halt state, refer to section 6, Power-Down Modes. For details on exception processing, refer to section 3, Exception Handling.

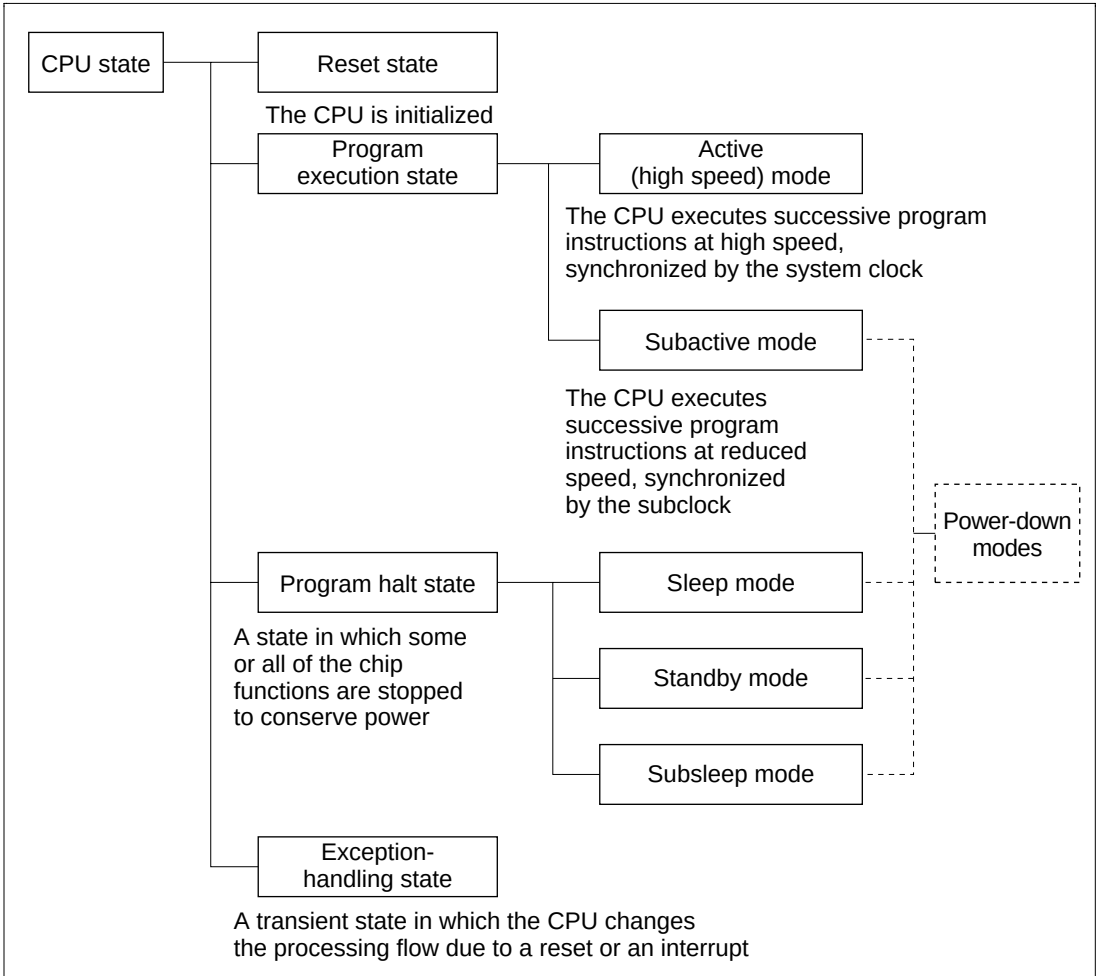


Figure 2.13 CPU Operation States

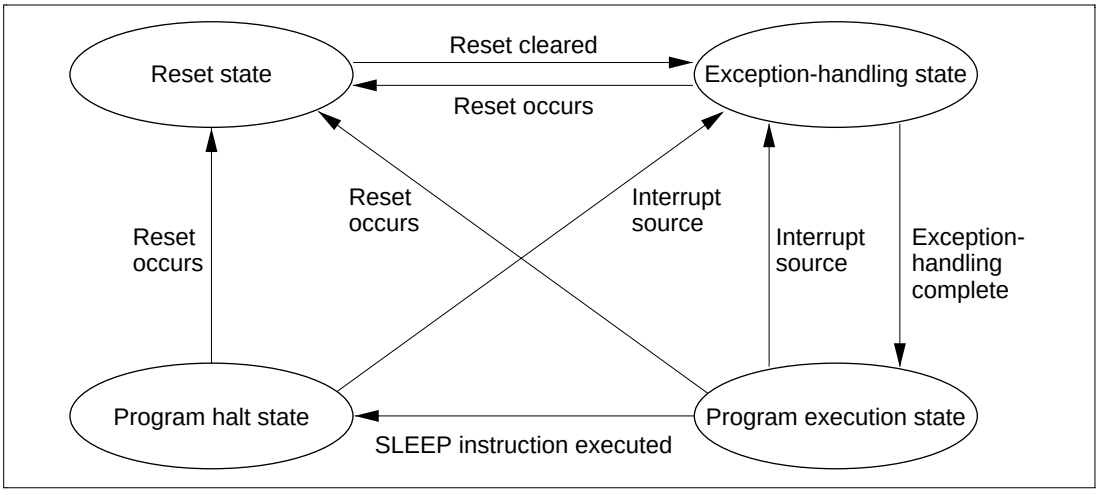


Figure 2.14 State Transitions

2.9 Application Notes

2.9.1 Notes on Data Access to Empty Areas

The address space of the H8/3664 Series 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.

2.9.2 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

Bit Manipulation in Two Registers Assigned to the Same Address

Example 1: Timer load register and timer counter (This applies to timers B and C. It does not apply to the H8/3664 Series.)

Figure 2.15 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 designated bit 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.

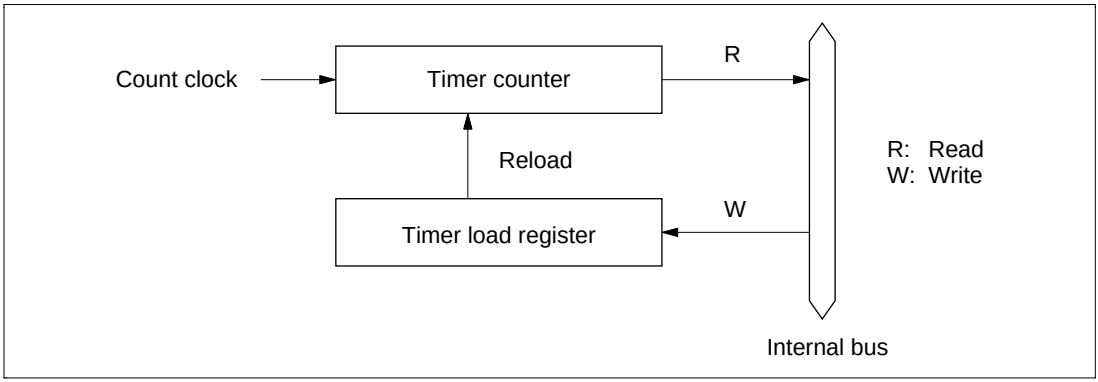


Figure 2.15 Timer Configuration Example

Example 2: BSET instruction executed designating port 5

$P5_7$ and $P5_6$ are designated as input pins, with a low-level signal input at $P5_7$ and a high-level signal input at $P5_6$. The remaining pins, $P5_5$ to $P5_0$, are output pins and output low-level signals. In this example, the BSET instruction is used to change pin $P5_0$ to high-level output.

[A: Prior to executing BSET]

	$P5_7$	**$P5_6$**	**$P5_5$**	**$P5_4$**	**$P5_3$**	**$P5_2$**	**$P5_1$**	**$P5_0$**
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
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	0

[B: BSET instruction executed]

```
BSET    #0,     @PDR5
```

The BSET instruction is executed designating port 5.

[C: After executing BSET]

	$P5_7$	**$P5_6$**	**$P5_5$**	**$P5_4$**	**$P5_3$**	**$P5_2$**	**$P5_1$**	**$P5_0$**
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
PCR5	0	0	1	1	1	1	1	1
PDR5	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 5.

Since $P5_7$ and $P5_6$ are input pins, the CPU reads the pin states (low-level and high-level input). $P5_5$ to $P5_0$ are output pins, so the CPU reads the value in PDR5. In this example PDR5 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 PDR5 data to H'41. Finally, the CPU writes this value (H'41) to PDR5, completing execution of BSET.

As a result of this operation, bit 0 in PDR5 becomes 1, and $P5_0$ outputs a high-level signal. However, bits 7 and 6 of PDR5 end up with different values.

To avoid this problem, store a copy of the PDR5 data in a work area in memory. Perform the bit manipulation on the data in the work area, then write this data to PDR5.

[A: Prior to executing BSET]

```
MOV.B   #80,  R0L
MOV.B   R0L,  @RAM0
MOV.B   R0L,  @PDR5
```

The PDR5 value (H'80) is written to a work area in memory (RAM0) as well as to PDR5.

	$P5_7$	**$P5_6$**	**$P5_5$**	**$P5_4$**	**$P5_3$**	**$P5_2$**	**$P5_1$**	**$P5_0$**
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
PCR5	0	0	1	1	1	1	1	1
PDR5	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 PDR5 work area (RAM0).

[C: After executing BSET]

```
MOV.B   @RAM0, R0L
MOV.B   R0L,  @PDR5
```

The work area (RAM0) value is written to PDR5.

	$P5_7$	**$P5_6$**	**$P5_5$**	**$P5_4$**	**$P5_3$**	**$P5_2$**	**$P5_1$**	**$P5_0$**
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
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	1
RAM0	1	0	0	0	0	0	0	1

Bit Manipulation in a Register Containing a Write-Only Bit

Example 3: BCLR instruction executed designating port 5 control register PCR5

As in the examples above, $P5_7$ and $P5_6$ are input pins, with a low-level signal input at $P5_7$ and a high-level signal input at $P5_6$. The remaining pins, $P5_5$ to $P5_0$, are output pins that output low-level signals. In this example, the BCLR instruction is used to change pin $P5_0$ to an input port. It is assumed that a high-level signal will be input to this input pin.

[A: Prior to executing BCLR]

	$P5_7$	**$P5_6$**	**$P5_5$**	**$P5_4$**	**$P5_3$**	**$P5_2$**	**$P5_1$**	**$P5_0$**
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
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	0

[B: BCLR instruction executed]

```
BCLR   #0,    @PCR5
```

The BCLR instruction is executed designating PCR5.

[C: After executing BCLR]

	$P5_7$	**$P5_6$**	**$P5_5$**	**$P5_4$**	**$P5_3$**	**$P5_2$**	**$P5_1$**	**$P5_0$**
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
PCR5	1	1	1	1	1	1	1	0
PDR5	1	0	0	0	0	0	0	0

[D: Explanation of how BCLR operates]

When the BCLR instruction is executed, first the CPU reads PCR5. Since PCR5 is a write-only register, the CPU reads a value of H'FF, even though the PCR5 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 PCR5 and BCLR instruction execution ends.

As a result of this operation, bit 0 in PCR5 becomes 0, making $P5_0$ an input port. However, bits 7 and 6 in PCR5 change to 1, so that $P5_7$ and $P5_6$ change from input pins to output pins.

To avoid this problem, store a copy of the PCR5 data in a work area in memory. Perform the bit manipulation on the data in the work area, then write this data to PCR5.

[A: Prior to executing BCLR]

```
MOV.B   #3F,  R0L
MOV.B   R0L,  @RAM0
MOV.B   R0L,  @PCR5
```

The PCR5 value (H'3F) is written to a work area in memory (RAM0) as well as to PCR5.

	$P5_7$	**$P5_6$**	**$P5_5$**	**$P5_4$**	**$P5_3$**	**$P5_2$**	**$P5_1$**	**$P5_0$**
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
PCR5	0	0	1	1	1	1	1	1
PDR5	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 PCR5 work area (RAM0).

[C: After executing BCLR]

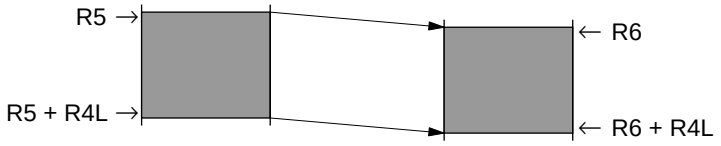
```
MOV.B   @RAM0, R0L
MOV.B   R0L,  @PCR5
```

The work area (RAM0) value is written to PCR5.

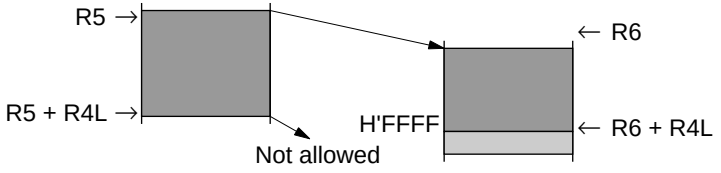
	$P5_7$	**$P5_6$**	**$P5_5$**	**$P5_4$**	**$P5_3$**	**$P5_2$**	**$P5_1$**	**$P5_0$**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR5	0	0	1	1	1	1	1	0
PDR5	1	0	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1	0

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

3.1.1 Exception Handling Types

Exception handling is performed in the H8/3664 Series when a reset, interrupt, or trap instruction occurs. Table 3.1 shows these three types of exception handling. A trap instruction can always be accepted when the program is being executed.

Table 3.1 Exception Handling Types

Exception Source	Time of Start of Exception Handling
Reset	Exception handling starts as soon as the reset state is cleared
Interrupt	When an interrupt is requested, exception handling starts after the present instruction or the exception handling in progress is completed
Trap instruction	Execution handling starts up when a TRAP instruction is executed

3.2 Reset

As soon as the $\overline{\text{RES}}$ pin is set to low, all processing is stopped and the chip enters the reset state. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized by the reset. To make sure the chip is reset properly, when turning the power on, the $\overline{\text{RES}}$ pin should be held at low until the clock pulse generator output stabilizes. When resetting during operation, the $\overline{\text{RES}}$ pin should be held at low for at least 10 system clock cycles. Reset exception handling begins when the $\overline{\text{RES}}$ pin is held at low for a given period, then returned to the high level.

3.2.1 Reset Sequence

A reset is the highest-priority exception handling. The sequence of the reset exception handling takes place as follows.

1. Set the I bit of the condition code register (CCR).
2. The CPU generates the reset exception handling vector address (H'0000 to H'0001), and transfers the address to PC as a start address. Then a program starts executing from the address indicated in PC. Figure 3.1 shows the reset sequence.

3.2.2 Reset by Watchdog Timer

When the watchdog timer overflows, the chip enters the reset state and reset exception handling begins. The same reset exception handling is carried out as for input at the $\overline{\text{RES}}$ pin. For details on the watchdog timer, see section 13, Watchdog Timer.

3.2.3 Interrupt Immediately after Reset

After a reset, if the CPU was to accept an interrupt before the stack pointer (SP) was initialized, PC and CCR would not be pushed onto the stack correctly, resulting in control over the program being lost. To prevent this, immediately after reset exception handling all interrupts including $\overline{\text{NMI}}$ 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).

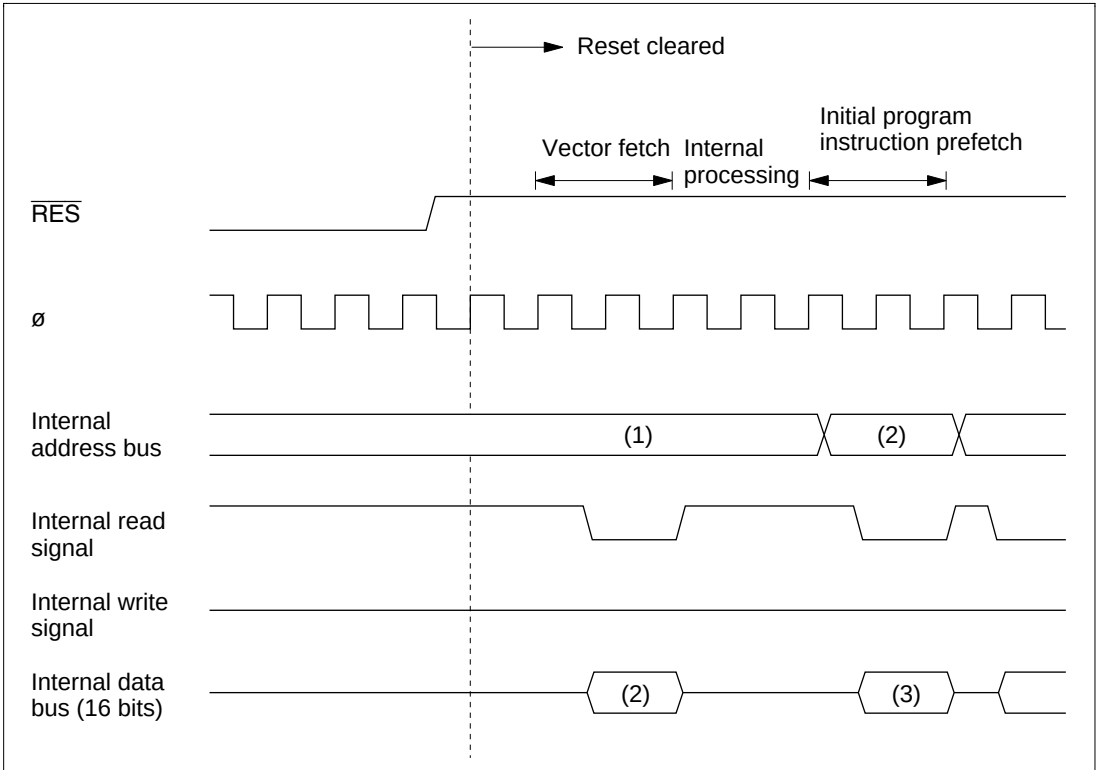


Figure 3.1 Reset Sequence

3.3 Interrupts

3.3.1 Interrupt and Vector Address

The interrupt sources that start the interrupt exception handling include 11 external interrupts and 20 internal interrupts. Table 3.2 shows the interrupts, their priorities, and their vector addresses. When more than one interrupt is requested, handling is performed from the interrupt with the highest priority.

$\overline{\text{NMI}}$ is the highest-priority interrupt, and cannot be masked by the I bit in CCR. All other external interrupts excluding $\overline{\text{NMI}}$ and internal interrupts excluding address break are masked by the I bit in CCR, and kept masked while the I bit is set to 1.

Table 3.2 Interrupt Priorities and Their Vector Addresses

Interrupt Source	Interrupt	Vector Number	Vector Address	Priority
$\overline{\text{RES}}$ Watchdog timer	Reset	0	H'0000 to H'0001	High
(Reserved by system)	(Reserved by system)	1	H'0002 to H'0003	
		2	H'0004 to H'0005	
		3	H'0006 to H'0007	
		4	H'0008 to H'0009	
		5	H'000A to H'000B	
		6	H'000C to H'000D	
External pin	NMI	7	H'000E to H'000F	
Trap instruction executed	Trap instruction #0	8	H'0010 to H'0011	
	Trap instruction #1	9	H'0012 to H'0013	
	Trap instruction #2	10	H'0014 to H'0015	
	Trap instruction #3	11	H'0016 to H'0017	
Address break	Break conditions satisfied	12	H'0018 to H'0019	
Sleep instruction executed	Transferred directly	13	H'001A to H'001B	
External pin	IRQ0	14	H'001C to H'001D	
	IRQ1	15	H'001E to H'001F	
	IRQ2	16	H'0020 to H'0021	
	IRQ3	17	H'0022 to H'0023	
	WKP5	18	H'0024 to H'0025	
Timer A	Timer A overflow	19	H'0026 to H'0027	Low

Interrupt Source	Interrupt	Vector Number	Vector Address	Priority
(Reserved by system)	(Reserved by system)	20	H'0028 to H'0029	High
Timer W	Input capture A / compare match A	21	H'002A to H'002B	
	Input capture B / compare match B			
	Input capture C / compare match C			
	Input capture D / compare match D			
	Timer W overflow			
Timer V	Timer V compare match A	22	H'002C to H'002D	
	Timer V compare match B			
	Timer V overflow			
SCI3	SCI3 transmit end	23	H'002E to H'002F	
	SCI3 transmit data empty			
	SCI3 receive data full			
	SCI3 overrun error			
	SCI3 framing error			
	SCI3 parity error			
IIC	Data transfer end	24	H'0030 to H'0031	
	Address inequality			
	Stop conditions detected			
A/D converter	A/D conversion end	25	H'0032 to H'0033	Low

3.4 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'C0	H'FFF3
Interrupt enable register 1	IENR1	R/W	H'10	H'FFF4
Interrupt flag register 1	IRR1	R/W*	H'30	H'FFF6
Wakeup interrupt flag register	IWPR	R/W*	H'C0	H'FFF8

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

3.4.1 Interrupt Edge Select Register 1 (IEGR1)

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

IEGR1 is an 8-bit read/write register used to designate whether pins $\overline{\text{NMI}}$ and $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$ are set to rising edge sensing or falling edge sensing. Upon reset, IEGR1 is initialized to H'70.

Bit 7—NMI Edge Select (NMIEG): Bit 7 selects the input sensing of pin $\overline{\text{NMI}}$.

Bit 7: NMIEG	Description	
0	Falling edge of $\overline{\text{NMI}}$ pin input is detected	(initial value)
1	Rising edge of $\overline{\text{NMI}}$ pin input is detected	

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

Bit 3—$\overline{\text{IRQ}}_3$ Edge Select (IEG3): Bit 3 selects the input sensing of pin $\overline{\text{IRQ3}}$.

Bit 3: IEG3	Description	
0	Falling edge of $\overline{\text{IRQ3}}$ pin input is detected	(initial value)
1	Rising edge of $\overline{\text{IRQ3}}$ pin input is detected	

Bit 2—IRQ_2 Edge Select (IEG2): Bit 2 selects the input sensing of pin $\overline{IRQ2}$.

Bit 2: IEG2	Description	
0	Falling edge of $\overline{IRQ2}$ pin input is detected	(initial value)
1	Rising edge of $\overline{IRQ2}$ pin input is detected	

Bit 1—IRQ_1 Edge Select (IEG1): Bit 1 selects the input sensing of pin $\overline{IRQ1}$.

Bit 1: IEG1	Description	
0	Falling edge of $\overline{IRQ1}$ pin input is detected	(initial value)
1	Rising edge of $\overline{IRQ1}$ pin input is detected	

Bit 0—IRQ_0 Edge Select (IEG0): Bit 0 selects the input sensing of pin $\overline{IRQ0}$.

Bit 0: IEG0	Description	
0	Falling edge of $\overline{IRQ0}$ pin input is detected	(initial value)
1	Rising edge of $\overline{IRQ0}$ pin input is detected	

3.4.2 Interrupt Edge Select Register 2 (IEGR2)

Bit	7	6	5	4	3	2	1	0
	—	—	WPEG5	WPEG4	WPEG3	WPEG2	WPEG1	WPEG0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	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 $\overline{WKP5}$ to $\overline{WKP0}$ are set to rising edge sensing or falling edge sensing. Upon reset, IEGR2 is initialized to H'C0.

Bits 7 and 6—Reserved Bits: Bits 7 and 6 are reserved; they are always read as 1, and cannot be modified.

Bit 5—WKP_5 Edge Select (WPEG5): Bit 5 selects the input sensing of pins $\overline{WKP5}$ and $\overline{ADTRG}$.

Bit 5: WPEG5	Description	
0	Falling edge of $\overline{WKP5}$ ($\overline{ADTRG}$) pin input is detected	(initial value)
1	Rising edge of $\overline{WKP5}$ ($\overline{ADTRG}$) pin input is detected	

Bits 4 to 0—$\overline{WKP_4}$ to $\overline{WKP_0}$ Edge Select (WPEG4 to WPEG0): Bits 4 to 0 select the input sensing of pins $\overline{WKP4}$ to $\overline{WKP0}$.

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

(n = 4 to 0)

3.4.3 Interrupt Enable Register 1 (IENR1)

Bit	7	6	5	4	3	2	1	0
	IENDT	IENTA	IENWP	—	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	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—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—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—Wakeup Interrupt Enable (IENWP): Bit 5 enables or disables $\overline{WKP5}$ to $\overline{WKP0}$ interrupt requests.

Bit 5: IENWP	Description	
0	Disables interrupt requests from pins $\overline{WKP5}$ to $\overline{WKP0}$	(initial value)
1	Enables interrupt requests from pins $\overline{WKP5}$ to $\overline{WKP0}$	

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

Bits 3 to 0—IRQ3 to IRQ0 Interrupt Enable (IEN3 to IEN0): Bits 3 to 0 enable or disable IRQ3 to IRQ0 interrupt requests.

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

(n = 3 to 0)

3.4.4 Interrupt Flag Register 1 (IRR1)

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

IRR1 is an 8-bit read/write register, in which a corresponding flag is set to 1 when a direct transfer, a timer A, or IRQ3 to IRQ0 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'30.

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—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	

Bits 5 and 4—Reserved Bits: Bits 5 and 4 are reserved: they are always read as 1 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—IRQ3 to IRQ0 Interrupt Request Flags (IRRI3 to IRRI0)

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

(n = 3 to 0)

3.4.5 Wakeup Interrupt Flag Register (IWPR)

Bit	7	6	5	4	3	2	1	0
	—	—	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

IWPR is an 8-bit read/write register, in which a corresponding flag is set to 1 when the designated signal edge is input at pin $\overline{WKP5}$ to $\overline{WKP0}$. The flags are not cleared automatically when an interrupt is accepted. It is necessary to write 0 to clear each flag. Upon reset, IWPR is initialized to H'C0.

Bits 7 and 6— Bits 7 and 6 are reserved; they are always read as 1, and cannot be modified.

Bits 5 to 0—WKP5 to WKP0 Interrupt Request Flags (IWPF5 to IWPF0)

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

(n = 5 to 0)

3.5 Interrupt Sources

3.5.1 External Interrupts

There are 11 external interrupts: NMI, IRQ3 to IRQ0, and WKP5 to WKP0.

- NMI Interrupt

  NMI interrupt is requested by input signal to pin $\overline{\text{NMI}}$. This interrupt is detected by either rising edge sensing or falling edge sensing, depending on the setting of bit NMIEG in IEGR1.

  NMI is the highest-priority interrupt, and can always be accepted without depending on the I bit value in CCR. When NMI interrupt exception handling is accepted, the I bit is set to 1 in CCR.

- IRQ3 to IRQ0 Interrupts

  IRQ3 to IRQ0 interrupts are requested by input signals to pins $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$. These four interrupts are given different vector addresses, and are detected individually by either rising edge sensing or falling edge sensing, depending on the settings of bits IEG3 to IEG0 in IEGR1.

  When pins $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$ are designated for interrupt input in PMR1 and the designated signal edge is input, the corresponding bit in IRR1 is set to 1, requesting the CPU of an interrupt. When IRQ3 to IRQ0 interrupt is accepted, the I bit is set to 1 in CCR. These interrupts can be masked by setting bits IEN3 to IEN0 in IENR1.

- WKP5 to WKP0 Interrupts

  WKP5 to WKP0 interrupts are requested by input signals to pins $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$. These six interrupts have the same vector addresses, and are detected individually by either rising edge sensing or falling edge sensing, depending on the settings of bits WPEG5 to WPEG0 in IEGR2.

  When pins $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$ are designated for interrupt input in PMR5 and the designated signal edge is input, the corresponding bit in IWPR is set to 1, requesting the CPU of an interrupt. When a WKP5 to WKP0 interrupt is accepted, the I bit is set to 1 in CCR. These interrupts can be masked by setting bit IENWP in IENR1.

3.5.2 Internal Interrupts

There are 20 internal interrupts that can be requested by the on-chip peripheral modules. Each on-chip peripheral module has a flag to show the interrupt request status and the enable bit to enable or disable the interrupt. For timer A interrupt requests and direct transfer interrupt requests generated by execution of a SLEEP instruction, this function is included in IRR1 and IENR1. Table 3.2 shows the order of priority of interrupts and their vector addresses.

When an on-chip peripheral module requests an interrupt, the corresponding interrupt request status flag is set to 1, requesting the CPU of an interrupt. When this interrupt is accepted, the I bit

is set to 1 in CCR. These interrupts can be masked by writing 0 to clear the corresponding enable bit.

3.5.3 Interrupt Operations

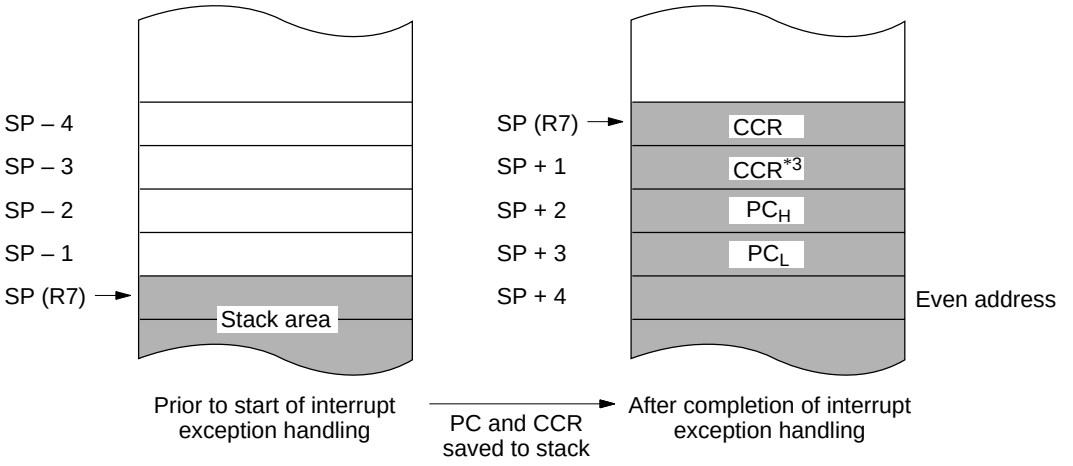
Interrupts are controlled by an interrupt controller.

Interrupt operation is described as follows.

1. If an interrupt occurs while the NMI or interrupt enable bit is set to 1, an interrupt request signal is sent to the interrupt controller.
2. 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.)
3. The CPU accepts the NMI and address break without depending on the I bit value. Other interrupt requests are accepted, if the I bit is cleared to 0 in CCR; if the I bit is set to 1, the interrupt request is held pending.
4. If the CPU accepts the interrupt after processing of the current instruction is completed, interrupt exception handling will begin. First, both PC and CCR are pushed onto the stack. The state of the stack at this time is shown in figure 3.2. The PC value pushed onto the stack is the address of the first instruction to be executed upon return from interrupt handling.
5. Then, the I bit of CCR is set to 1, masking further interrupts excluding the NMI and address break. Upon return from interrupt handling, the values of I bit and other bits in CCR will be restored and returned to the values prior to the start of interrupt exception handling.
6. Next, the CPU generates the vector address corresponding to the accepted interrupt, and transfers the address to PC as a start address of the interrupt handling-routine. Then a program starts executing from the address indicated in PC.

Figure 3.3 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.

Notes: 1. When disabling interrupts by clearing bits in an interrupt enable register, or when clearing bits in an interrupt flag 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 handling for the interrupt will be executed after the clear instruction has been executed.



Notation:
PC_H: Upper 8 bits of program counter (PC)
PC_L: Lower 8 bits of program counter (PC)
CCR: Condition code register
SP: Stack pointer

Notes: 1. PC shows the address of the first instruction to be executed upon return from the interrupt handling routine.
2. Register contents must always be saved and restored by word length, starting from an even-numbered address.
3. Ignored when returning from the interrupt handling routine.

Figure 3.2 Stack State after Completion of Interrupt Exception Handling

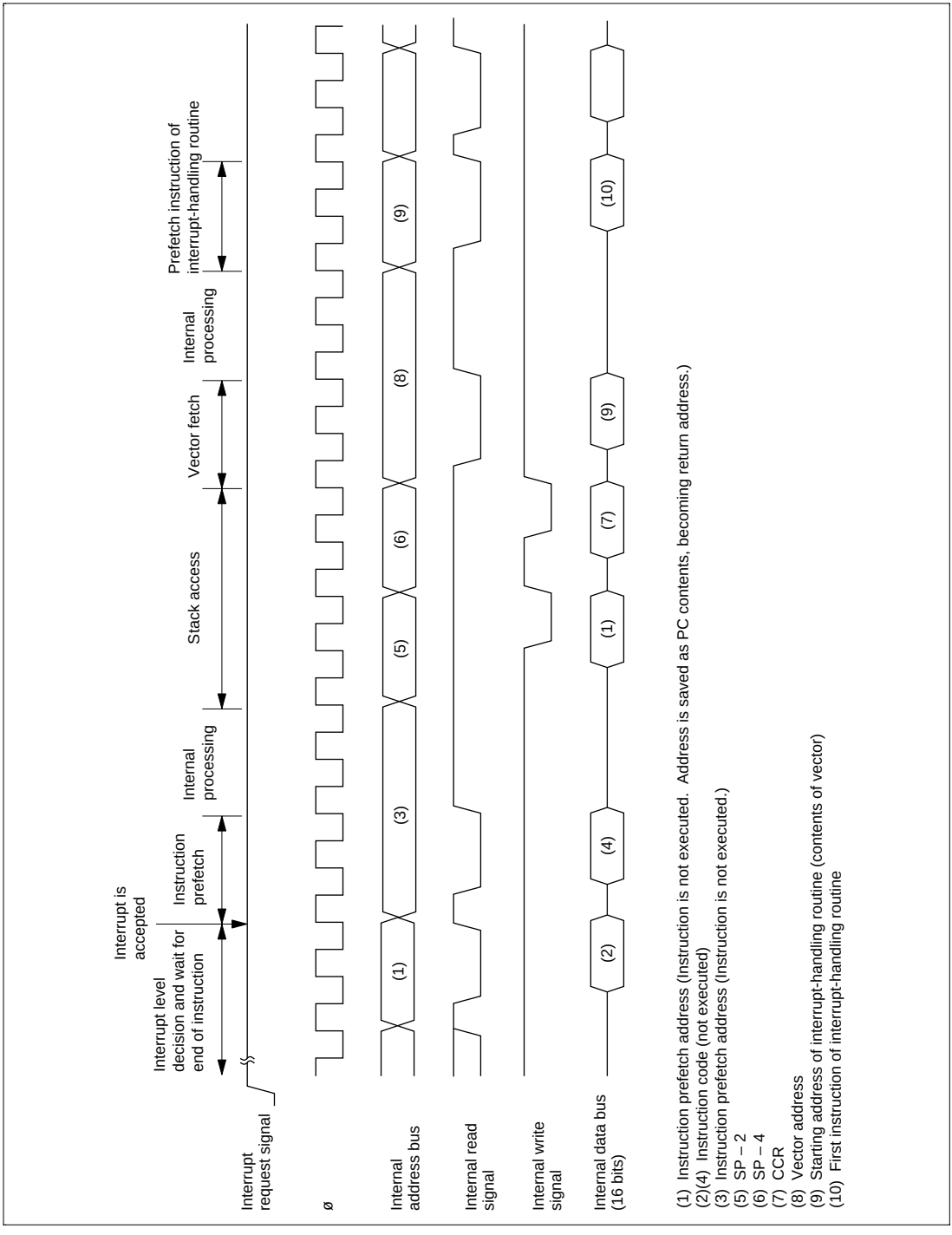


Figure 3.3 Interrupt Sequence

3.5.4 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 handling-routine 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.

3.6 Trap Instruction

When a TRAP instruction is executed, trap instruction exception handling starts up. A TRAP instruction generates vector addresses corresponding to the vector numbers 0 to 3 designated in the instruction code.

3.7 Application Notes

3.7.1 Notes on Stack Area Use

When word data is accessed in the H8/3664 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 3.4.

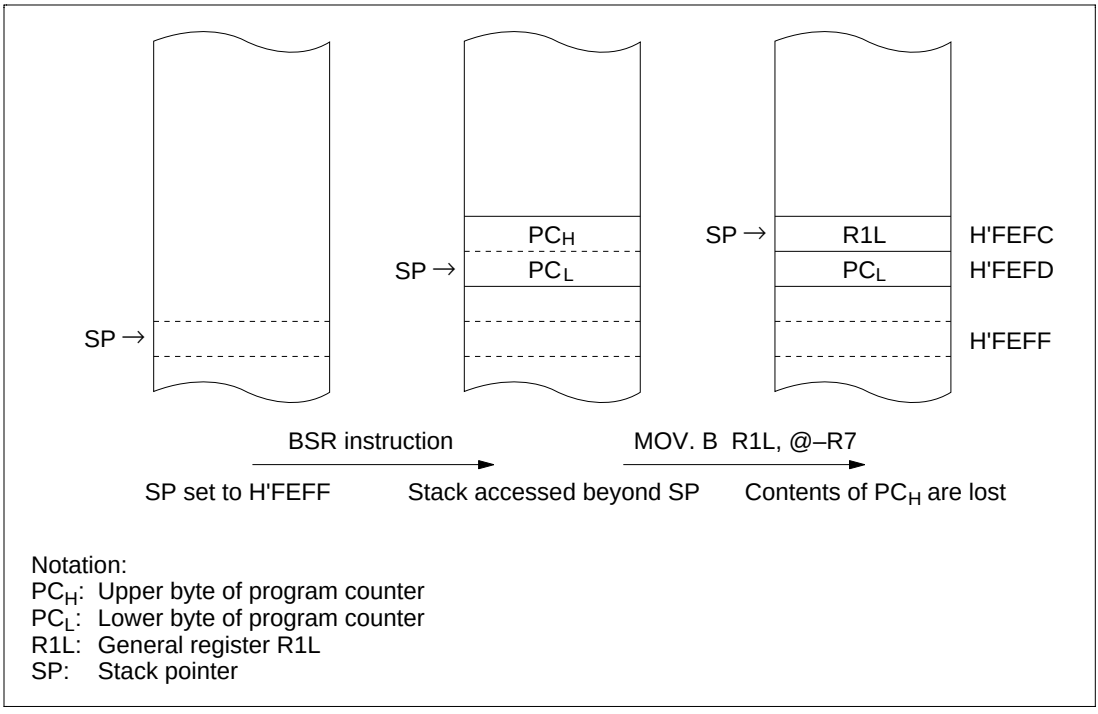


Figure 3.4 Operation when Odd Address is Set in SP

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.

3.7.2 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 $\overline{IRQ}_1$, and $\overline{WKP}_5$ to $\overline{WKP}_0$, 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 bit IRQ3 in PMR1 is changed from 0 to 1 while pin $\overline{IRQ_3}$ is low and bit IEG3 in IEGR1 = 0.
		When bit IRQ3 in PMR1 is changed from 1 to 0 while pin $\overline{IRQ_3}$ is low and bit IEG3 in IEGR1 = 1.
	IRRI2	When bit IRQ2 in PMR1 is changed from 0 to 1 while pin $\overline{IRQ_2}$ is low and bit IEG2 in IEGR1 = 0.
		When bit IRQ2 in PMR1 is changed from 1 to 0 while pin $\overline{IRQ_2}$ is low and bit IEG2 in IEGR1 = 1.
	IRRI1	When bit IRQ1 in PMR1 is changed from 0 to 1 while pin $\overline{IRQ_1}$ is low and bit IEG1 in IEGR1 = 0.
		When bit IRQ1 in PMR1 is changed from 1 to 0 while pin $\overline{IRQ_1}$ is low and bit IEG1 in IEGR1 = 1.
	IRRI0	When bit IRQ0 in PMR1 is changed from 0 to 1 while pin $\overline{IRQ_0}$ is low and bit IEG0 in IEGR1 = 0.
		When bit IRQ0 in PMR1 is changed from 1 to 0 while pin $\overline{IRQ_0}$ is low and bit IEG0 in IEGR1 = 1.
IWPR	IWPF5	When bit WKP5 in PMR5 is changed from 0 to 1 while pin $\overline{WKP_5}$ is low and bit WPEG5 in IEGR2 = 0.
		When bit WKP5 in PMR5 is changed from 1 to 0 while pin $\overline{WKP_5}$ is low and bit WPEG5 in IEGR2 = 1.
	IWPF4	When bit WKP4 in PMR5 is changed from 0 to 1 while pin $\overline{WKP_4}$ is low and bit WPEG4 in IEGR2 = 0.
		When bit WKP4 in PMR5 is changed from 1 to 0 while pin $\overline{WKP_4}$ is low and bit WPEG4 in IEGR2 = 1.
	IWPF3	When bit WKP3 in PMR5 is changed from 0 to 1 while pin $\overline{WKP_3}$ is low and bit WPEG3 in IEGR2 = 0.
		When bit WKP3 in PMR5 is changed from 1 to 0 while pin $\overline{WKP_3}$ is low and bit WPEG3 in IEGR2 = 1.
	IWPF2	When bit WKP2 in PMR5 is changed from 0 to 1 while pin $\overline{WKP_2}$ is low and bit WPEG2 in IEGR2 = 0.
		When bit WKP2 in PMR5 is changed from 1 to 0 while pin $\overline{WKP_2}$ is low and bit WPEG2 in IEGR2 = 1.
	IWPF1	When bit WKP1 in PMR5 is changed from 0 to 1 while pin $\overline{WKP_1}$ is low and bit WPEG1 in IEGR2 = 0.
		When bit WKP1 in PMR5 is changed from 1 to 0 while pin $\overline{WKP_1}$ is low and bit WPEG1 in IEGR2 = 1.
	IWPF0	When bit WKP0 in PMR5 is changed from 0 to 1 while pin $\overline{WKP_0}$ is low and bit WPEG0 in IEGR2 = 0.
		When bit WKP0 in PMR5 is changed from 1 to 0 while pin $\overline{WKP_0}$ is low and bit WPEG0 in IEGR2 = 1.

Figure 3.5 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. If the instruction 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 to avoid the setting of interrupt request flags when pin functions are switched is to keep the pins at the high level so that the conditions in table 3.5 do not occur.

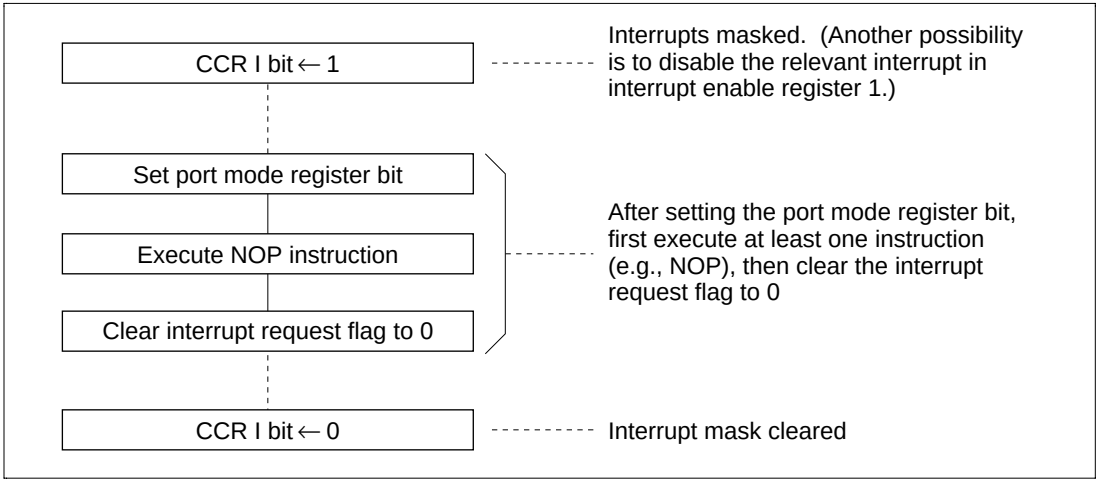


Figure 3.5 Port Mode Register Setting and Interrupt Request Flag Clearing Procedure

Section 4 Address Break

4.1 Overview

The address break simplifies on-board program debugging. It requests an address break interrupt when the set break condition is satisfied. The interrupt request is not affected by the I bit of CCR. Break conditions that can be set include instruction execution at a specific address and a combination of access and data at a specific address. With the address break function, the execution start point of a program containing a bug is detected and execution is branched to the correcting program.

4.1.1 Block Diagram

Figure 4.1 shows a block diagram of the address break.

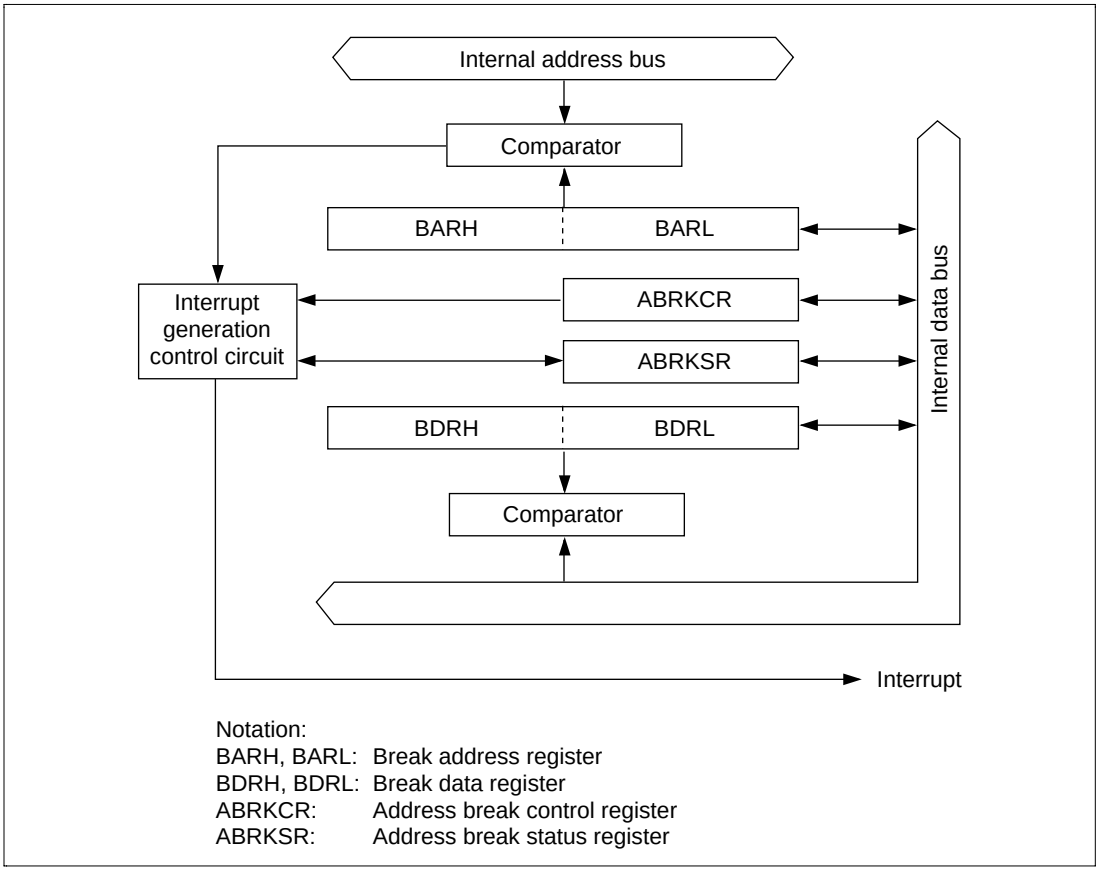


Figure 4.1 Block Diagram of an Address Break

4.1.2 Register Configuration

Table 4.1 shows the address break register configuration.

Table 4.1 Address Break Registers

Name	Abbrev.	R/W	Initial Value	Address
Address break control register	ABRKCR	R/W	H'80	H'FFC8
Address break status register	ABRKSR	R/W	H'3F	H'FFC9
Break address register (H)	BARH	R/W	H'FF	H'FFCA
Break address register (L)	BARL	R/W	H'FF	H'FFCB
Break data register (H)	BDRH	R/W	Undefined	H'FFCC
Break data register (L)	BDRL	R/W	Undefined	H'FFCD

4.2 Register Descriptions

4.2.1 Address Break Control Register (ABRKCR)

Bit	7	6	5	4	3	2	1	0
	RTINTE	CSEL1	CSEL0	ACMP2	ACMP1	ACMP0	DCMP1	DCMP0
Initial value	1	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ABRKCR is an 8-bit read/write register that sets address break conditions.

Bit 7—RTE Interrupt Enable (RTINTE): Bit 7 enables or disables an interrupt after RTE instruction execution.

Bit 7: RTINTE	Description
0	Disables an interrupt after RTE instruction execution (one instruction is executed)
1	Enables an interrupt after RTE instruction execution (Initial value)

Bits 6 and 5—Condition Select (CSEL1, CSEL0): Bits 6 and 5 set address break conditions. When CSEL1=0 and CSEL0=0, data is not compared regardless of the values of DCMP1 and DCMP0.

Bit 6: CSEL1	Bit 5: CSEL0	Description	
0	0	Instruction execution cycle	(Initial value)
	1	CPU data read cycle	
1	0	CPU data write cycle	
	1	CPU data read/write cycle	

Bits 4 to 2—Address Compare Condition Select (ACMP2 to ACMP0): Bits 4 to 2 set comparison condition between the address set in BAR and the internal address bus.

Bit 4: ACMP2	Bit 3: ACMP1	Bit 2: ACMP0	Description	
0	0	0	Compare 16-bit addresses	(Initial value)
		1	Compares upper 12-bit addresses	
	1	0	Compares upper 8-bit addresses	
		1	Compares upper 4-bit addresses	
1	*	*	Reserved	

Note: * Don't care.

Bits 1 and 0—Data Compare Condition Select (DCMP1, DCMP0): Bits 1 and 0 set the comparison condition between the data set in BDR and the internal data bus.

Bit 1: DCMP1	Bit 0: DCMP0	Description	
0	0	No data comparison	(Initial value)
	1	Compares lower 8-bit data between BDRL and data bus	
1	0	Compares upper 8-bit data between BDRH and data bus	
	1	Compares 16-bit data between BDR and data bus	

When an address break is set in the data read cycle or data write cycle, the data bus used will depend on the combination of the byte/word access and address. Table 4.2 shows the access and data bus used.

Table 4.2 Access and Data Bus Used

	Word Access		Byte Access	
	Even Address	**Odd Address**	**Even Address**	**Odd Address**
ROM space	Upper 8 bits	Lower 8 bits	Upper 8 bits	Upper 8 bits
RAM space	Upper 8 bits	Lower 8 bits	Upper 8 bits	Upper 8 bits
I/O register space (except H'FF86 to H'FF8F)	Upper 8 bits*	Upper 8 bits*	Upper 8 bits	Upper 8 bits
I/O register space (H'FF86 to H'FF8F)	Upper 8 bits	Lower 8 bits	—	—

Note: * When the I/O register space, except H'FF86 to H'FF8F, is accessed by word, byte access occurs twice.

4.2.2 Address Break Status Register (ABRKSR)

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

ABRKSR is an 8-bit read/write register that consists of the address break interrupt flag and the address break interrupt enable bit.

Bit 7—Address Break Interrupt Flag (ABIF): Bit 7 indicates an address break interrupt.

Bit 7: ABIF	**Description**
0	An address break interrupt request is not generated [Clearing condition] When 0 is written after ABIF=1 is read (Initial value)
1	An address break interrupt request is generated [Set condition] When the condition set in ABRKCR is satisfied

Bit 6—Address Break Interrupt Enable (ABIE): Bit 6 enables or disables an address break interrupt.

Bit 6: ABIE	Description	
0	Disables an address break interrupt request	(Initial value)
1	Enables an address break interrupt request	

Bits 5 to 0—Reserved Bits: Bits 5 to 0 are reserved; they are always read as 1 and cannot be modified.

4.2.3 Break Address Registers (BARH, BARL)

Bit	7	6	5	4	3	2	1	0
	BARH7	BARH6	BARH5	BARH4	BARH3	BARH2	BARH1	BARH0
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
	BARL7	BARL6	BARL5	BARL4	BARL3	BARL2	BARL1	BARL0
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

BAR (BARH, BARL) is a 16-bit read/write register that sets the address for generating an address break interrupt. When setting the address break condition to the instruction execution cycle, set the first byte address of the instruction.

4.2.4 Break Data Registers (BDRH, BDRL)

Bit	7	6	5	4	3	2	1	0
	BDRH7	BDRH6	BDRH5	BDRH4	BDRH3	BDRH2	BDRH1	BDRH0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
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
	BDRL7	BDRL6	BDRL5	BDRL4	BDRL3	BDRL2	BDRL1	BDRL0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BDR (BDRH, BDRL) is a 16-bit read/write register that sets the data for generating an address break interrupt. BDRH is compared with the upper 8-bit data bus. BDRL is compared with the lower 8-bit data bus. When memory or registers are accessed by byte, the upper 8-bit data bus is used for even and odd addresses in the data transmission. Therefore, comparison data must be set in BDRH for byte access. For word access, the data bus used depends on the address. See section 4.2.1, Address Break Control Register, for details.

4.3 Operation

When the ABIE bit in ABRKSR is set to 1, if the ABIF bit in ABRKSR is set to 1 by the combination of the address set in BAR, the data set in BDR, and the conditions set in ABRKCR, the address break function generates an interrupt request to the CPU. When the interrupt request is accepted, interrupt exception handling starts after the instruction being executed ends. The address break interrupt is not masked because of the I bit in CCR of the CPU.

Figures 4.2 to 4.4 show the operation examples of the address break interrupt setting.

When the address break is specified in instruction execution cycle

Register setting
- ABRKCR = H'80
- BAR = H'025A

Program

```
  0258   NOP
* 025A   NOP
  025C   MOV.W @H'025A,R0
  0260   NOP
  0262   NOP
   :      :
```

Underline indicates the address to be stacked.

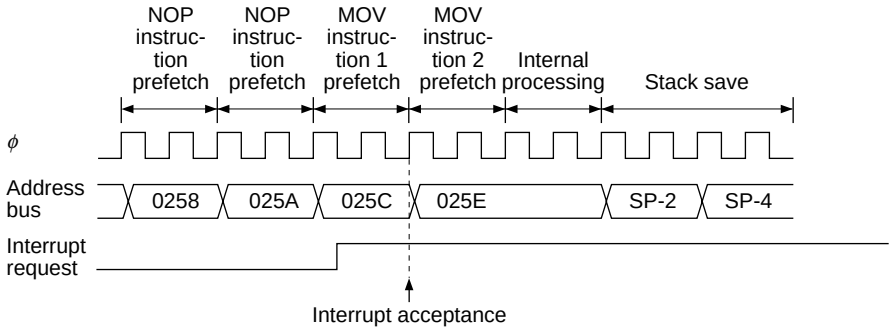


Figure 4.2 Address Break Interrupt Operation Example (1)

When the address break is specified in the data read cycle

Register setting
- ABRKCR = H'A0
- BAR = H'025A

Program

```
  0258   NOP
  025A   NOP
* 025C   MOV.W @H'025A,R0
  0260   NOP
  0262   NOP
   :      :
```

Underline indicates the address to be stacked.

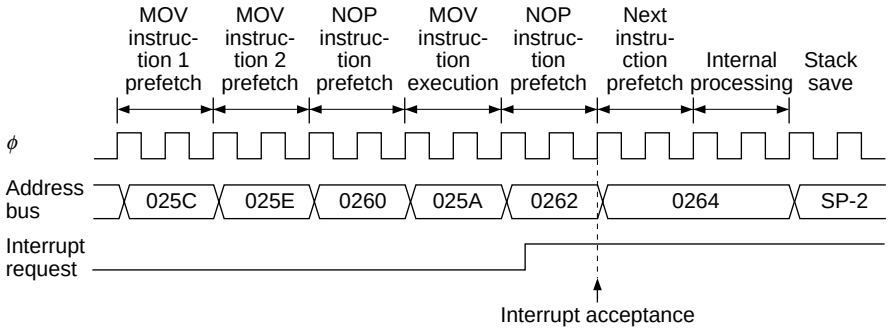


Figure 4.3 Address Break Interrupt Operation Example (2)

When the interrupt acceptance is prohibited after the RTE (RTB) instruction

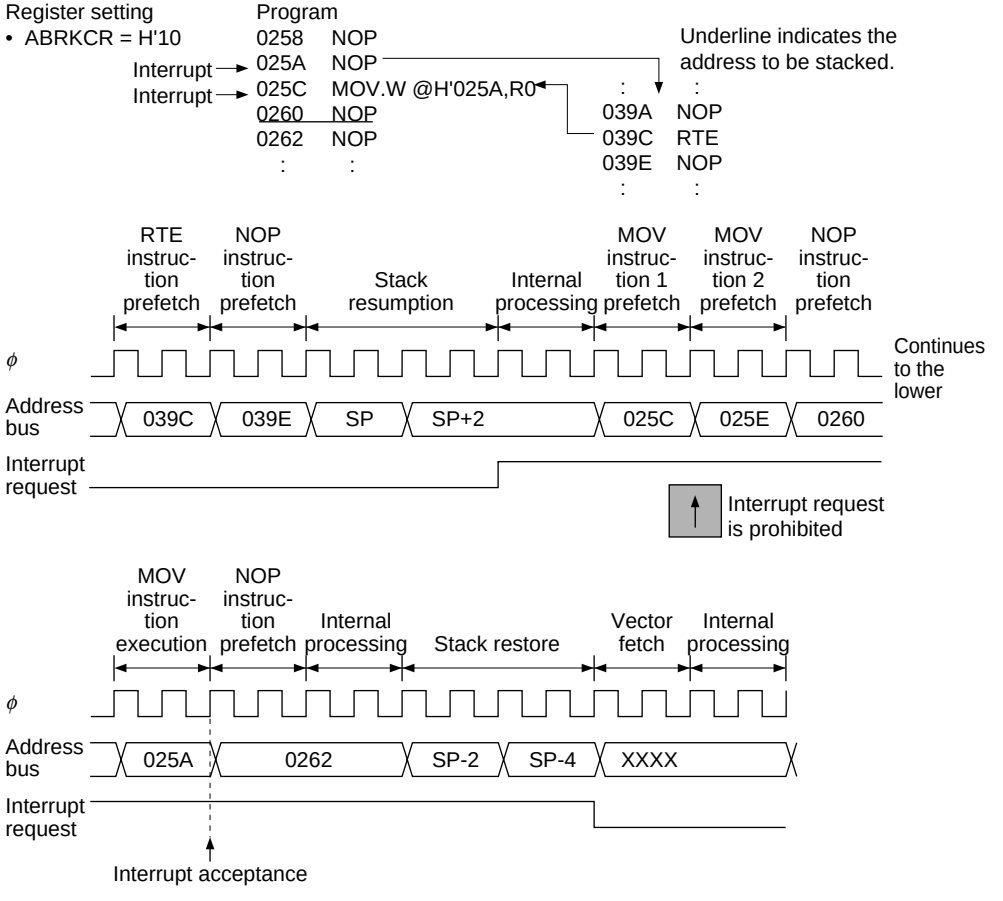


Figure 4.4 Address Break Interrupt Operation Example (3)

Section 5 Clock Pulse Generators

5.1 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.

5.1.1 Block Diagram

Figure 5.1 shows a block diagram of the clock pulse generators.

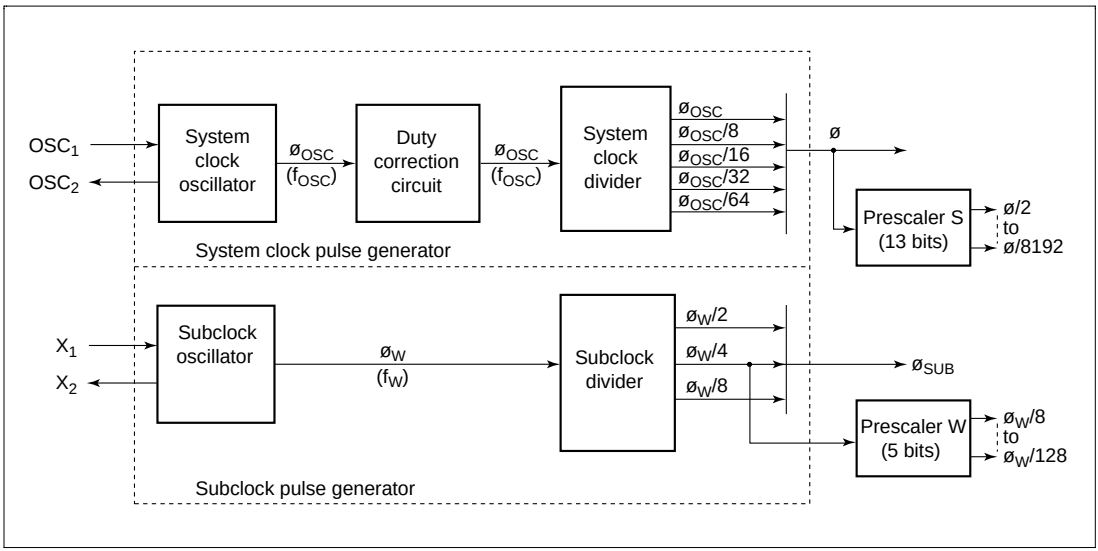


Figure 5.1 Block Diagram of Clock Pulse Generators

5.1.2 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, $ø_W$/8, $ø_W$/16, $ø_W$/32, $ø_W$/64, and $ø_W$/128. The clock requirements differ from one module to another.

5.2 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.

Connecting a Crystal Oscillator: Figure 5.2 shows a typical method of connecting a crystal oscillator. An AT-cut parallel-resonance crystal resonator should be used.

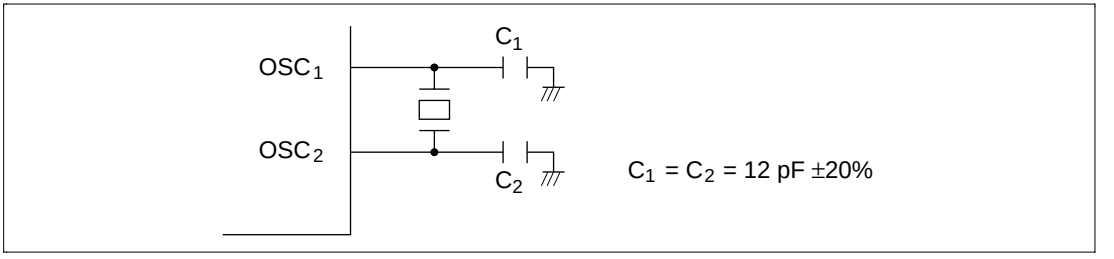


Figure 5.2 Typical Connection to Crystal Oscillator

Figure 5.3 shows the equivalent circuit of a crystal oscillator. An oscillator having the characteristics given in table 5.1 should be used.

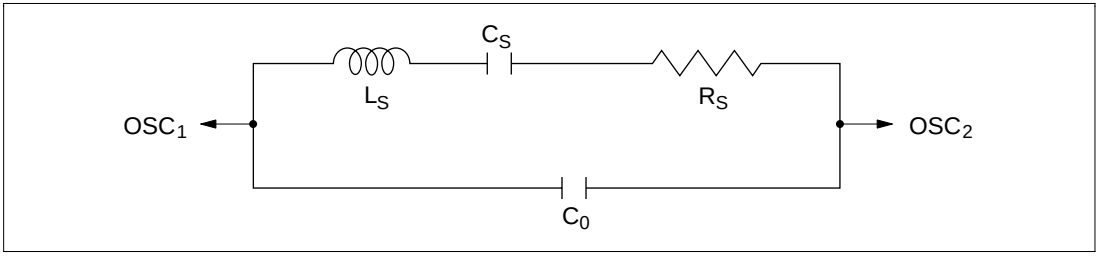


Figure 5.3 Equivalent Circuit of Crystal Oscillator

Table 5.1 Crystal Oscillator Parameters

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

Connecting a Ceramic Oscillator: Figure 5.4 shows a typical method of connecting a ceramic oscillator.

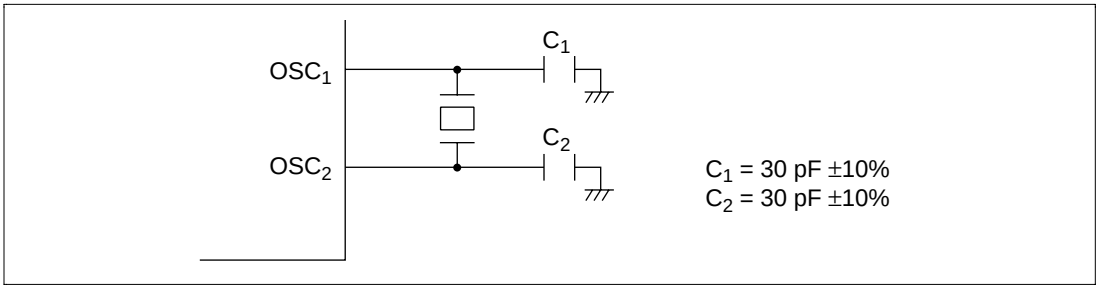


Figure 5.4 Typical Connection to Ceramic Oscillator

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 5.5.)

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.

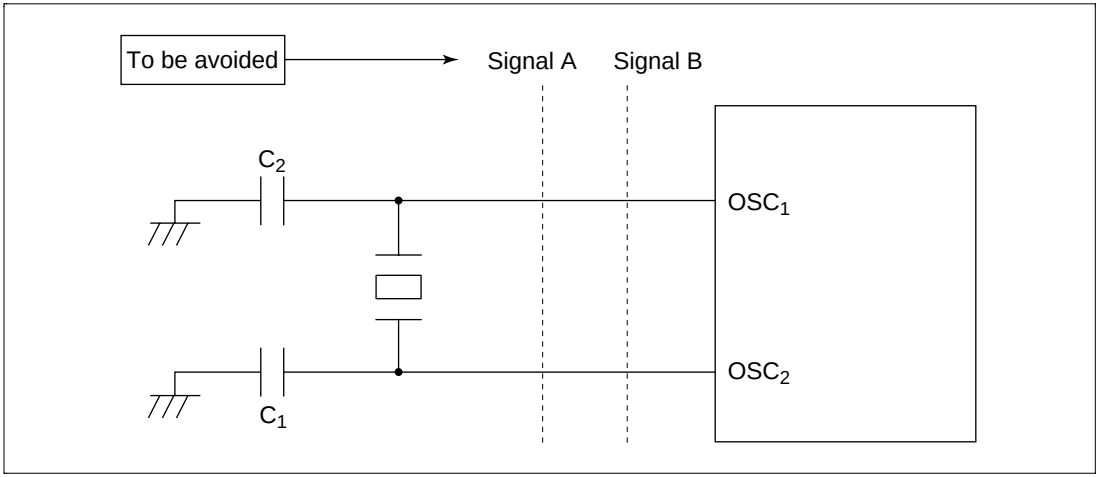


Figure 5.5 Board Design of Oscillator Circuit

External Clock Input Method: Connect an external clock signal to pin OSC_1, and leave pin OSC_2 open. Figure 5.6 shows a typical connection. The duty cycle of the external clock signal must be 45 to 55%.

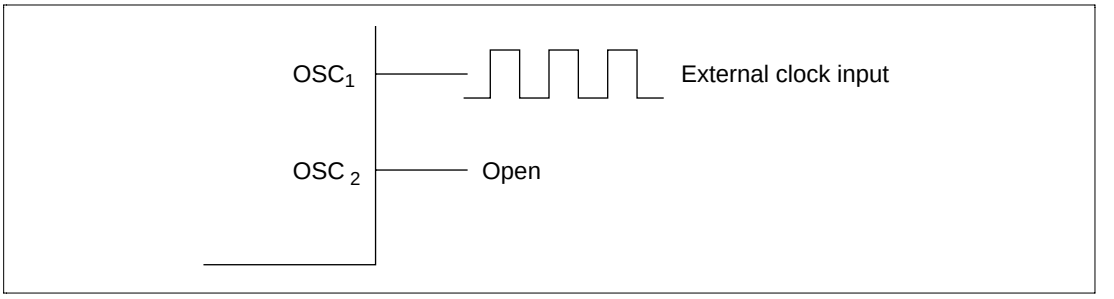


Figure 5.6 Example of External Clock Input

5.3 Subclock Generator

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 5.7. Follow the same precautions as noted under 5.2 Notes on Board Design.

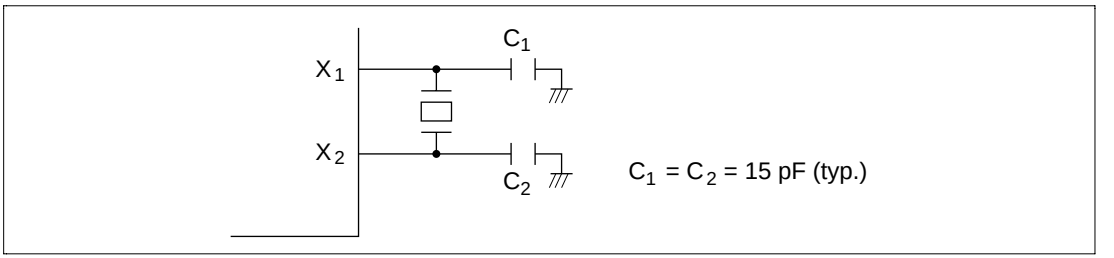


Figure 5.7 Typical Connection to 32.768-kHz Crystal Oscillator

Figure 5.8 shows the equivalent circuit of the 32.768-kHz crystal oscillator.

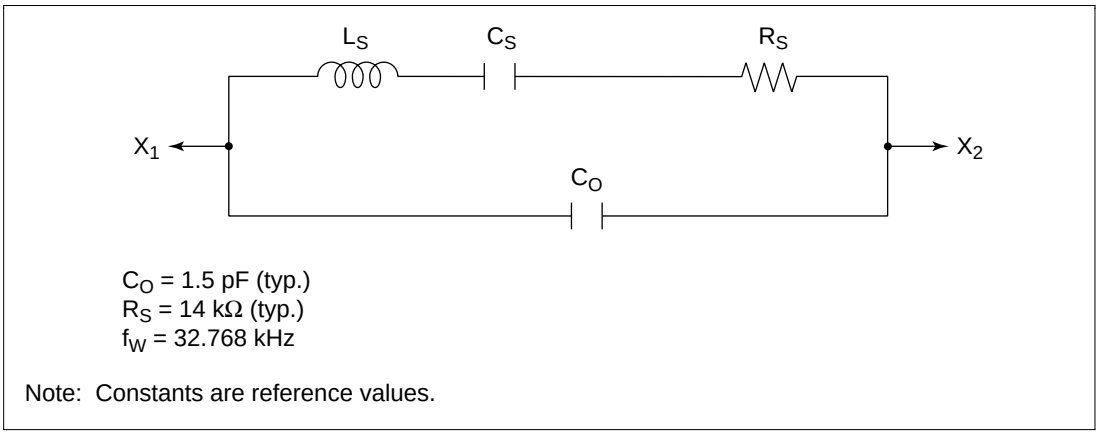


Figure 5.8 Equivalent Circuit of 32.768-kHz Crystal Oscillator

Pin Connection when Not Using Subclock: When the subclock is not used, connect pin X_1 to V_{CL} or V_{SS} and leave pin X_2 open, as shown in figure 5.9.

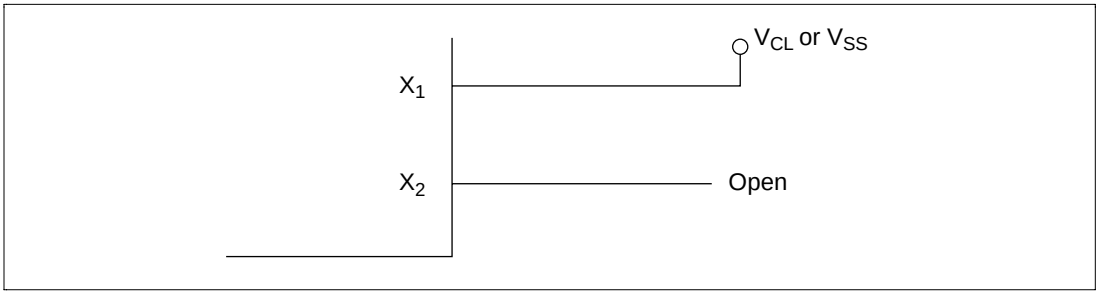


Figure 5.9 Pin Connection when not Using Subclock

5.4 Prescalers

This LSI 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.

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, 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 in SYSCR1.

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, 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.

5.5 Usage Notes

5.5.1 Note on Oscillators

Oscillator characteristics are closely related to board design and should be carefully evaluated by the user, referring to the 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.

5.5.2 Notes on Board Design

When using a crystal resonator (ceramic resonator), place the resonator and its load capacitors as close as possible to the OSC1 and OSC2 pins. Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation (see figure 5.10).

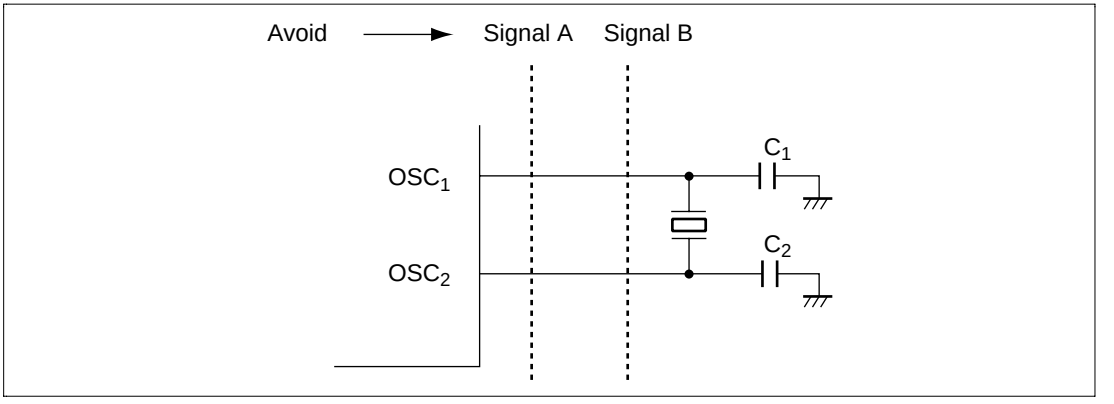


Figure 5.10 Example of Incorrect Board Design

Section 6 Power-down Modes

6.1 Overview

This LSI has six modes of operation after a reset. These include a normal active mode and four power-down modes, in which power dissipation is significantly reduced. The module standby mode reduces power dissipation by selectively halting on-chip module functions. Table 6.1 summarizes the six operating modes.

Table 6.1 Operating Modes

Operating Mode	Description
Active mode	The CPU and all on-chip peripheral modules are operable on the system clock. The system clock frequency can be selected from øosc, øosc/8, øosc/16, øosc/32, and øosc/64. For details, see 6.2.2, System Control Register 2.
Subactive mode	The CPU and all on-chip peripheral modules are operable on the subclock. The subclock frequency can be selected from øw/2, øw/4, and øw/8.
Sleep mode	The CPU halts. On-chip peripheral functions are operable on the system clock.
Subsleep mode	The CPU halts. On-chip peripheral functions are operable on the subclock.
Standby mode	The CPU and all on-chip peripheral modules halt. When the clock time-base function is selcted, timer A is operable.
Module standby mode	The on-chip peripheral modules specified by software stop operating. For details, see 6.2.3, Module Standby Control Register 1.

6.1.1 Register Configuration

Table 6.2 shows the power-down mode register configuration.

Table 6.2 Power-down Mode Registers

Name	Abbreviation	R/W	Initial Value	Address
System control register 1	SYSCR1	R/W	H'00	H'FFF0
System control register 2	SYSCR2	R/W	H'00	H'FFF1
Module standby control register 1	MSTCR1	R/W	H'00	H'FFF9

6.2 Register Descriptions

6.2.1 System Control Register 1 (SYSCR1)

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	NESEL	—	—	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	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'00.

Bit 7—Software Standby (SSBY): This bit designates the transition to the sleep mode, subsleep mode, or standby mode.

Bit 7: SSBY	Description
0	When a SLEEP instruction is executed in the active mode, a transition is made to the sleep mode or subsleep mode. (Initial value)
1	When a SLEEP instruction is executed in the active mode, a transition is made to the standby 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 the standby mode, subactive mode, or subsleep mode to the active mode or sleep 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)
		1	Wait time = 16,384 states
	1	0	Wait time = 32,768 states
		1	Wait time = 65,536 states
1	0	0	Wait time = 131,072 states
		1	Wait time = 1,024 states
	1	0	Wait time = 128 states
		1	Wait time = 16 states

Bit 3—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 3: NESEL	Description	
0	Sampling rate is $ø_{OSC}$/16	(Initial value)
1	Sampling rate is $ø_{OSC}$/4	

Bits 2 to 0—Reserved Bits: Bits 2 to 0 are reserved: they are always read as 0 and cannot be modified.

6.2.2 System Control Register 2 (SYSCR2)

Bit	7	6	5	4	3	2	1	0
	SMSEL	LSON	DTON	MA2	MA1	MA0	SA1	SA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	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'00.

Bit 7—Sleep Mode Selection (SMSEL): This bit chooses the transition to the sleep mode or subsleep mode when the SLEEP instruction is executed. The transition after the SLEEP instruction is executed depends on a combination of this and other control bits.

Bit 7: SMSEL	Description	
0	A transition is made to sleep mode.	(Initial value)
1	A transition is made to subsleep mode.	

Bit 6—Low Speed on Flag (LSON): This bit chooses the system clock (ø) or subclock ($ø_{SUB}$) as the CPU operating clock. The resulting operation mode after the SLEEP instruction is executed depends on the combination of other control bits.

Bit 6: LSON	Description	
0	The CPU operates on the system clock (ø)	(Initial value)
1	The CPU operates on the subclock ($ø_{SUB}$)	

Bit 5—Direct Transfer on Flag (DTON): This bit designates whether to make direct transitions between the active and subactive modes 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 5: DTON	Description
0	When a SLEEP instruction is executed, a transition is made to standby mode, sleep mode, or subsleep mode. (Initial value)
1	When a SLEEP instruction is executed, a direct transition is made to active mode if LSON = 0, or to subactive mode if LSON = 1.

Bits 4 to 2—Active Mode Clock Select (MA2 to MA0): These bits select the operating clock frequency in the active and sleep modes. The operating clock frequency changes to the set frequency after the SLEEP instruction is executed.

Bit 4: MA2	Bit 3: MA1	Bit 2: MA0	Description
0	*	*	$ø_{osc}$ (Initial value)
1	0	0	$ø_{osc}/8$
		1	$ø_{osc}/16$
	1	0	$ø_{osc}/32$
		1	$ø_{osc}/64$

Note: * Don’t care

Bits 1 and 0— Subactive Mode Clock Select (SA1, SA0): These bits select the operating clock frequency in the subactive and subsleep modes. The operating clock frequency changes to the set frequency after the SLEEP instruction is executed.

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

Note: * Don’t care

6.2.3 Module Standby Control Register 1 (MSTCR1)

Bit	7	6	5	4	3	2	1	0
	—	MSTIIC	MSTS3	MSTAD	MSTWD	MSTTW	MSTTV	MSTTA
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTCR1 is an 8-bit read/write register that enables each on-chip peripheral module to enter the standby state.

Upon reset, MSTCR1 is initialized to H'00.

Bit 7—Reserved Bit: This bit is reserved: it is always read as 0 and cannot be modified.

Bit 6—IIC Module Standby (MSTIIC): This bit enables IIC to enter the standby state.

Bit 6: MSTIIC	Description	
0	The IIC operates normally.	(Initial value)
1	The IIC enters the standby state.	

Bit 5—SCI3 Module Standby (MSTS3): This bit enables SCI3 to enter the standby state.

Bit 5: MSTS3	Description	
0	The SCI3 operates normally.	(Initial value)
1	The SCI3 enters the standby state.	

Bit 4—A/D Converter Module Standby (MSTAD): This bit enables the A/D converter to enter the standby state.

Bit 4: MSTAD	Description	
0	The A/D converter operates normally.	(Initial value)
1	The A/D converter enters the standby state.	

Bit 3—Watchdog Timer Module Standby (MSTWD): This bit enables the watchdog timer to enter the standby state. When the internal oscillator is selected for the watchdog timer clock, the watchdog timer operates regardless of the setting of this bit.

Bit 3: MSTWD	Description	
0	The watchdog timer operates normally.	(Initial value)
1	The watchdog timer enters the standby state.	

Bit 2—Timer W Module Standby (MSTTW): This bit enables timer W to enter the standby state.

Bit 2: MSTTW	Description	
0	Timer W operates normally.	(Initial value)
1	Timer W enters the standby state.	

Bit 1—Timer V Module Standby (MSTTV): This bit enables timer V to enter the standby state.

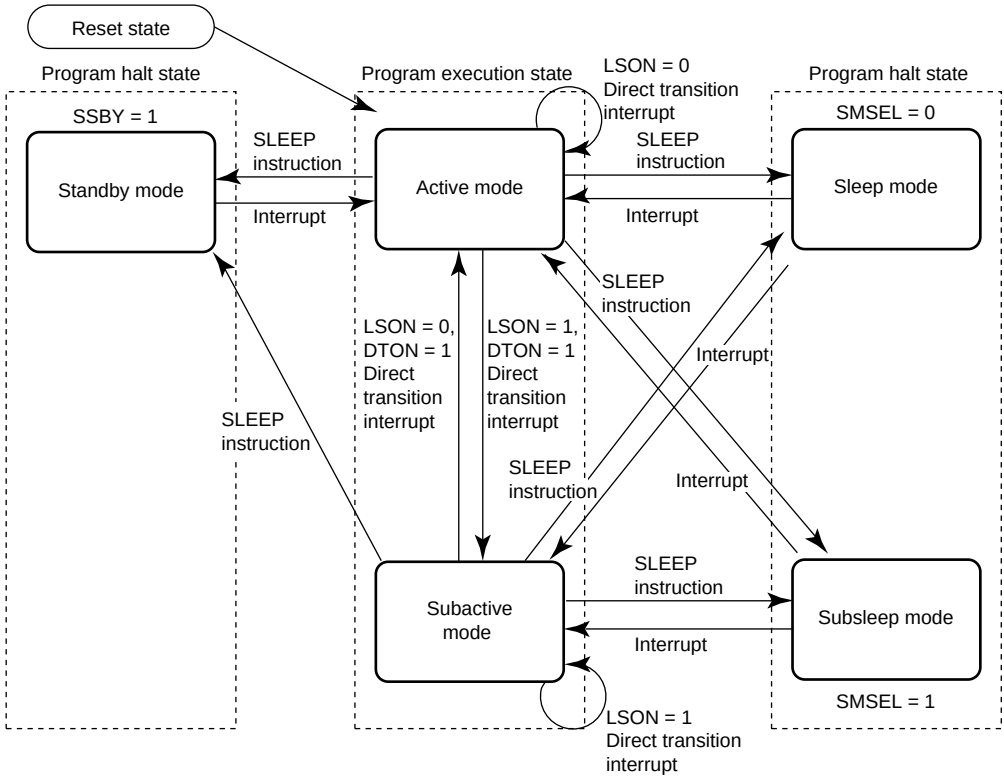
Bit 1: MSTTV	Description	
0	Timer V operates normally.	(Initial value)
1	Timer V enters the standby state.	

Bit 0—Timer A Module Standby (MSTTA): This bit enables timer A to enter the standby state.

Bit 0: MSTTA	Description	
0	Timer A operates normally.	(Initial value)
1	Timer A enters the standby state.	

6.3 Mode Transition Conditions

Figure 6.1 shows the transitions among these operation modes. Table 6.3 shows the transition mode after the SLEEP instruction is executed and when an interrupt occurs. Table 6.4 indicates the internal states in each mode.



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 explanation of each mode in sections 6.4 to 6.9.

Figure 6.1 Mode Transition Diagram

Table 6.3 Transition Mode after the SLEEP Instruction Execution and Interrupt Handling

DTON	SSBY	SMSEL	LSON	Transition Mode after SLEEP Instruction Execution	Transition Mode due to Interrupt
0	0	0	0	Sleep mode	Active mode
			1		Subactive mode
		1	0	Subsleep mode	Active mode
			1		Subactive mode
	1	*	*	Standby mode	Active mode
1	*	0*[1]	0	Active mode	—
	*	*	1	Subactive mode	—

Notes: * Don't care

1. When state transition is performed while SMSEL is 1, timer V, SCI3, and the A/D converter are reset, and all registers are set to their initial values. To use these after entering active mode, reset the registers.

Table 6.4 Internal State in Each Operating Mode

Function		Active Mode	Sleep Mode	Subactive Mode	Subsleep Mode	Standby Mode
System clock oscillator		Functions	Functions	Halted	Halted	Halted
Subclock oscillator		Functions	Functions	Functions	Functions	Functions
CPU operations	Instructions	Functions	Halted	Functions	Halted	Halted
	Registers		Retained		Retained	Retained
RAM						
IO ports						Retained*1
External interrupts	IRQ_0	Functions	Functions	Functions	Functions	Functions
	IRQ_1					
	IRQ_2					
	IRQ_3					
	WKP_0	Functions	Functions	Functions	Functions	Functions
	WKP_1					
	WKP_2					
	WKP_3					
	WKP_4					
	WKP_5					
Peripheral functions*3	Timer A	Functions	Functions	Functions*2	Functions*2	Functions*2
	Timer V			Reset	Reset	Reset
	Timer W			Retained*4*6	Retained*4	Retained
	Watchdog timer			Retained*5*6	Retained*5	Retained*5
	SCI3			Reset	Reset	Reset
	IIC			Retained*6	Retained	Retained
	A/D converter			Reset	Reset	Reset

Notes:
1. Register contents are retained, but output is the high-impedance state.
2. Functions if the timekeeping time-base function is selected, and retained if not selected. Retained in the module standby mode regardless of time-base function selection.
3. Peripheral functions can be in the standby state in each mode by setting module standby control register 1 (MSTCR1).
4. If internal clock ø is selected as a count clock, the counter is incremented by a subclock.
5. Functions if the internal oscillator is selected as a count clock.
6. Registers can be read or written.

6.4 Sleep Mode

6.4.1 Transition to the Sleep Mode

The system goes from the active mode to the sleep mode when a SLEEP instruction is executed while the SSBY bit in SYSCR1 and the DTON and SMSEL bits in SYSCR2 are all cleared to 0. In the sleep mode, CPU operation is halted but the on-chip peripheral modules function at the clock frequency set by the MA2, MA1, and MA0 bits in SYSCR2. CPU register contents are retained.

6.4.2 Clearing the Sleep Mode

The sleep mode is cleared by any interrupt or by input at the $\overline{\text{RES}}$ pin.

- Clearing by interrupt

  When an interrupt is requested, the sleep mode is cleared and interrupt exception handling starts. The sleep mode is not cleared if the I bit of the condition code register (CCR) is set to 1 or the requested interrupt is disabled in the interrupt enable register. After the sleep mode is cleared, a transition is made to active mode when the LSON bit in SYSCR2 is 0, and a transition is made to subactive mode when the bit is 1.

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

6.5 Standby Mode

6.5.1 Transition to the Standby Mode

The system goes from the active mode to the standby mode when a SLEEP instruction is executed while the SSBY bit in SYSCR1 is set to 1 and the DTON bit in SYSCR1 is cleared to 0. In the standby mode, the clock pulse generator stops, so the CPU and on-chip peripheral modules stop functioning. However, 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.

6.5.2 Clearing the Standby Mode

The standby mode is cleared by an interrupt or by input at the $\overline{\text{RES}}$ 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, the standby mode is cleared, and interrupt exception handling starts. The standby mode is not cleared if the I bit of CCR is set to 1 or the requested interrupt is disabled in the interrupt enable register.

- Clearing by $\overline{\text{RES}}$ input

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

6.5.3 Oscillator Settling Time after the Standby Mode is Cleared

- When a crystal oscillator is used

  Table 6.5 lists the settings for various operating frequencies. Bits STS2 to STS0 in SYSCR1 must be set so that the waiting time is longer than the oscillator settling time.

Table 6.5 Operating Frequency and Waiting Time (ø = øosc)

STS2	STS1	STS0	Waiting Time	10 MHz	8 MHz	4 MHz	2 MHz	1 MHz	0.5 MHz
0	0	0	8,192 states	0.8	1.6	2.0	4.1	8.1	16.4
		1	16,384 states	1.6	2.0	4.1	8.2	16.4	32.8
	1	0	32,768 states	3.3	4.1	8.2	16.4	32.8	65.5
		1	65,536 states	6.6	8.2	16.4	32.8	65.5	131.1
1	0	0	131,072 states	13.1	16.4	32.8	65.5	131.1	262.1
		1	1,024 states	0.10	0.13	0.26	0.51	1.02	2.05
	1	0	128 states	0.01	0.02	0.03	0.06	0.13	0.26
		1	16 states	0.00	0.00	0.00	0.01	0.02	0.03

(Unit: ms)

- When an external clock is used

  Any values can be set. Normally the minimum time (STS2 = STS1 = STS0 = 1) should be set.

6.6 Subsleep Mode

6.6.1 Transition to the Subsleep Mode

The system goes from the active or subactive mode to the subsleep mode when a SLEEP instruction is executed while the SSBY bit in SYSCR1 is cleared to 0, the DTON bit in SYSCR2 is cleared to 0, and the SMSEL bit in SYSCR2 is set to 1. In the subsleep mode, operation of the CPU and 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.

6.6.2 Clearing the Subsleep Mode

The subsleep mode is cleared by an interrupt or by input at the $\overline{\text{RES}}$ pin.

- Clearing by interrupt

  When an interrupt is requested, the subsleep mode is cleared and interrupt exception handling starts. The subsleep mode is not cleared if the I bit of CCR is set to 1 or the requested interrupt is disabled in the interrupt enable register. After the subsleep mode is cleared, a transition is made to the active mode when the LSON bit in SYSCR2 is 0, and a transition is made to the subactive mode when the bit is 1.

- Clearing by $\overline{\text{RES}}$ input

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

6.7 Subactive Mode

6.7.1 Transition to the Subactive Mode

The subactive mode is entered from the sleep or subsleep mode if an interrupt is requested while the LSON bit in SYSCR2 is set to 1. The operating frequency of the subactive mode is selected from $\phi_W/2$, $\phi_W/4$, and $\phi_W/8$ by the SA1 and SA0 bits in SYSCR2. The operating frequency changes to the set frequency after SLEEP instruction execution. A transition to the subactive mode does not take place if the I bit of CCR is set to 1 or the requested interrupt is disabled in the interrupt enable register.

The subactive mode can be directly entered from the active mode. See section 6.9, Direct Transition, for details.

6.7.2 Clearing the Subactive Mode

The subactive mode is cleared by a SLEEP instruction or by input at the $\overline{\text{RES}}$ pin.

- Clearing by SLEEP instruction

  If a SLEEP instruction is executed in the subactive mode, the subactive mode is cleared and another mode is entered. If the SSBY bit in SYSCR1 and the DTON and SMSEL bits in SYSCR2 are cleared to 0, the sleep mode is entered; if the SSBY bit in SYSCR1, the DTON bit in SYSCR2 are cleared to 0, and the SMSEL bit is set to 1, the subsleep mode is entered; if the SSBY bit in SYSCR1 is set to 1 and the DTON bit in SYSCR2 is cleared to 0, the standby mode is entered; if the DTON bit in SYSCR2 is set to 1 and the LSON bit is cleared to 0, the active mode is directly entered. For direct transition to the active mode, see section 6.9, Direct Transition.

- Clearing by $\overline{\text{RES}}$ pin

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

6.8 Active Mode

6.8.1 Transition to the Active Mode

The active mode is entered from the standby, sleep, or subsleep mode if an interrupt is requested while the LSON bit in SYSCR2 is cleared to 0. The system directly goes to the active mode when a SLEEP instruction is executed while the DTON bit in SYSCR2 is set to 1 and the LSON bit in SYSCR2 is cleared to 0. The operating frequency changes to the set frequency after SLEEP instruction execution. A transition to the active mode does not take place if the I bit of CCR is set to 1 or the requested interrupt is disabled in the interrupt enable register. In the active mode, the CPU and on-chip peripheral modules function at the clock frequency set by the MA2, MA1, and MA0 bits in SYSCR2.

6.8.2 Transition from the Active Mode to Other Modes

A transition from the active mode to the standby mode takes place if the SLEEP instruction is executed while the SSBY bit in SYSCR1 is set to 1 and the DTON bit in SYSCR1 is cleared to 0. If the SLEEP instruction is executed while SSBY is cleared to 0 and the DTON and SMSEL bits in SYSCR2 are cleared to 0, the sleep mode is entered. If the SLEEP instruction is executed while SSBY is cleared to 0, the DTON bit is cleared to 0, and the SMSEL bit is set to 1, the subsleep mode is entered. Direct transition from the active mode to the subactive mode is also possible. See section 6.9, Direct Transition, for details.

6.8.3 Operating Frequency in the Active Mode

Operation in the active mode is clocked at the frequency designated by the MA1 and MA0 bits in SYSCR1. The operating frequency changes to the set frequency after SLEEP instruction execution.

6.9 Direct Transition

The CPU can execute programs in two modes: active and subactive mode. A direct transition is a transition between these two modes without stopping program execution. A direct transition can be made by executing a SLEEP instruction while the DTON bit in SYSCR2 is set to 1. The direct transition also enables operating frequency modification in the active or subactive mode. After the mode transition, direct transition interrupt exception handling starts.

If the direct transition interrupt is disabled in interrupt enable register 1, a transition is made instead to the sleep or subsleep mode. Note that if a direct transition is attempted while the I bit in CCR is set to 1, the sleep or subsleep mode will be entered, and the resulting mode cannot be cleared by means of an interrupt.

- Direct transition from the active mode to the subactive mode

  When a SLEEP instruction is executed in the active mode while the DTON and LSON bits in SYSCR2 are set to 1, a transition is made to the subactive mode.
- Direct transition from the subactive mode to the active mode

  When a SLEEP instruction is executed in the subactive mode while the DTON bit in SYSCR2 is set to 1 and the LSON bit in SYSCR2 is cleared to 0, a direct transition is made to the active mode after the waiting time set in SYSCR1 bits STS2 to STS0 has elapsed.

6.9.1 Direct Transition Time

- Direct transition from the active mode to the subactive mode

  When a SLEEP instruction is executed in the active mode while the DTON and LSON bits in SYSCR2 are set to 1, a transition is made to the subactive mode. The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (1).

  Direct transition time = {(number of SLEEP instruction execution states) + (number of internal processing states)}× (tcyc before transition) + (number of interrupt exception handling states) × (tsubcyc after transition) ..... (1)

  [Example]
  Direct transition time = (2 + 1) × tosc + 14 × 8tw = 3tosc + 112tw
  (when the CPU operating clock of øosc → øw/8 is selected)

  <Symbols>
  tosc: OSC clock cycle time
  tw: watch clock cycle time
  tcyc: system clock (ø) cycle time
  tsubcyc: subclock (øSUB) cycle time

- Direct transition from the subactive mode to the active mode

  When a SLEEP instruction is executed in the subactive mode while the DTON bit in SYSCR2 is set to 1 and the LSON bit in SYSCR2 is cleared to 0, a direct transition is made from the subactive mode to the active mode. The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (2).

  Direct transition time = {(number of SLEEP instruction execution states) + (number of internal processing states)} × (tsubcyc before transition) + {(waiting time set in bits STS2 to STS0) + (number of interrupt exception handling states)} × (tcyc after transition)..... (2)

  [Example]
  Direct transition time = (2 + 1) × 8tw + (8192 + 14) × tosc = 24tw + 8206tosc
  (when the CPU operating clock of øw/8 → øosc and a waiting time of 8192 states are selected)

  <Symbols>
  tosc: OSC clock cycle time
  tw: watch clock cycle time
  tcyc: system clock (ø) cycle time
  tsubcyc: subclock (øSUB) cycle time

6.10 Module Standby Mode

The module standby mode can be set to any peripheral module. In the module standby mode, the clock supply to modules and the module functions stop much like the states in the standby mode. Therefore, power consumption can be reduced.

The module standby mode can be specified by setting each bit of module standby control register 1 (MSTCR1) to 1. The module standby mode can be canceled by clearing each bit of module standby control register 1 (MSTCR1) to 0.

Section 7 ROM

7.1 Features

The HD64F3664 has 32 kbytes of on-chip flash memory. The features of the flash memory are summarized below.

- Four flash memory operating modes
  - Program mode
  - Erase mode
  - Program-verify mode
  - Erase-verify mode
- Programming/erase methods

  The flash memory is programmed 128 bytes at a time. The flash memory includes four 1-kB blocks and one 28-kB block. To erase the entire flash memory, each block must be erased in turn.
- Reprogramming capability

  The flash memory can be reprogrammed up to 100 times.
- On-board programming modes

  There are two modes in which flash memory can be programmed/erased/verified on-board:
  - Boot mode
  - User program mode
- Automatic bit rate adjustment

  With data transfer in boot mode, the LSI's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Protect modes

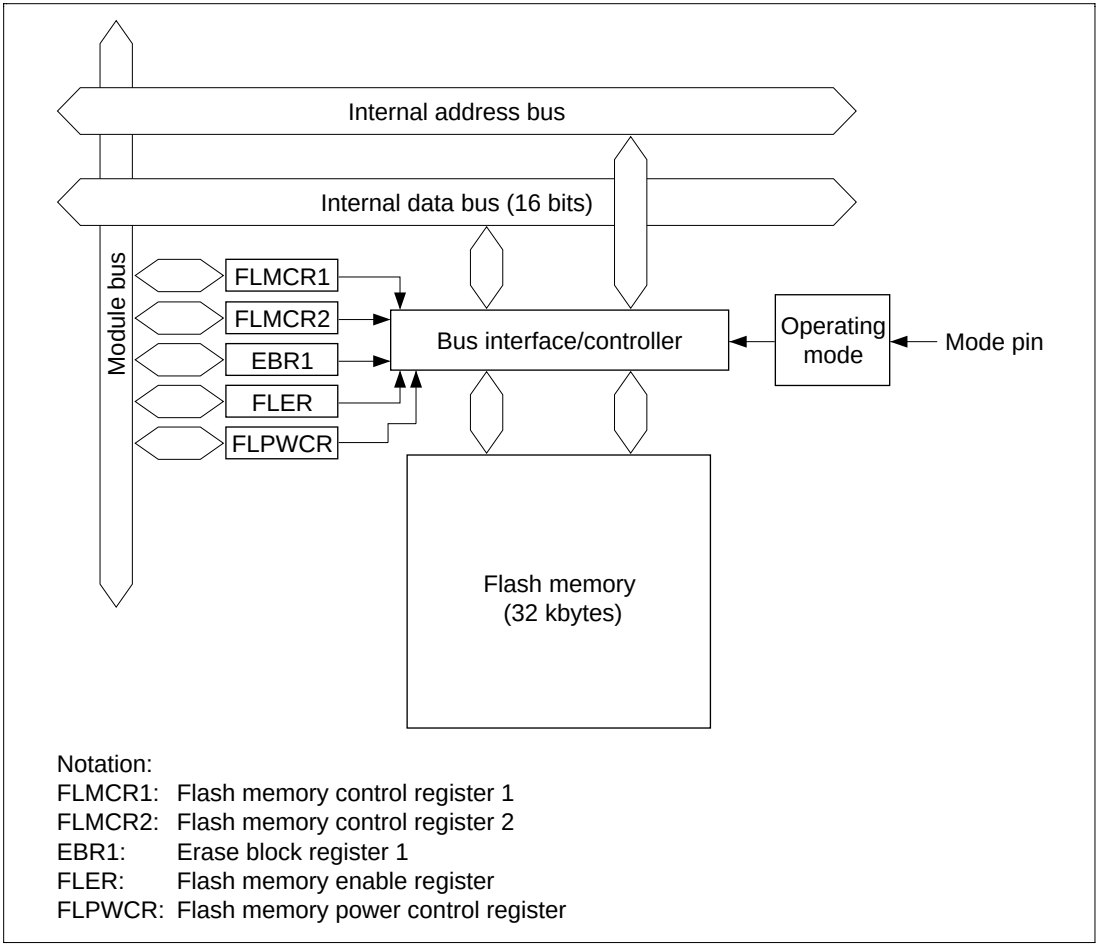
  The software protect mode is available in which protected status is designated for flash memory program/erase/verify operations.
- Programmer mode

  Flash memory can be programmed/erased in programmer mode, using a PROM programmer, as well as in on-board programming mode.
- Power-down mode

  In the sub-active mode, data is read in the power-down mode by stopping the operation of part of the power supply circuit.

7.2 Overview

7.2.1 Block Diagram



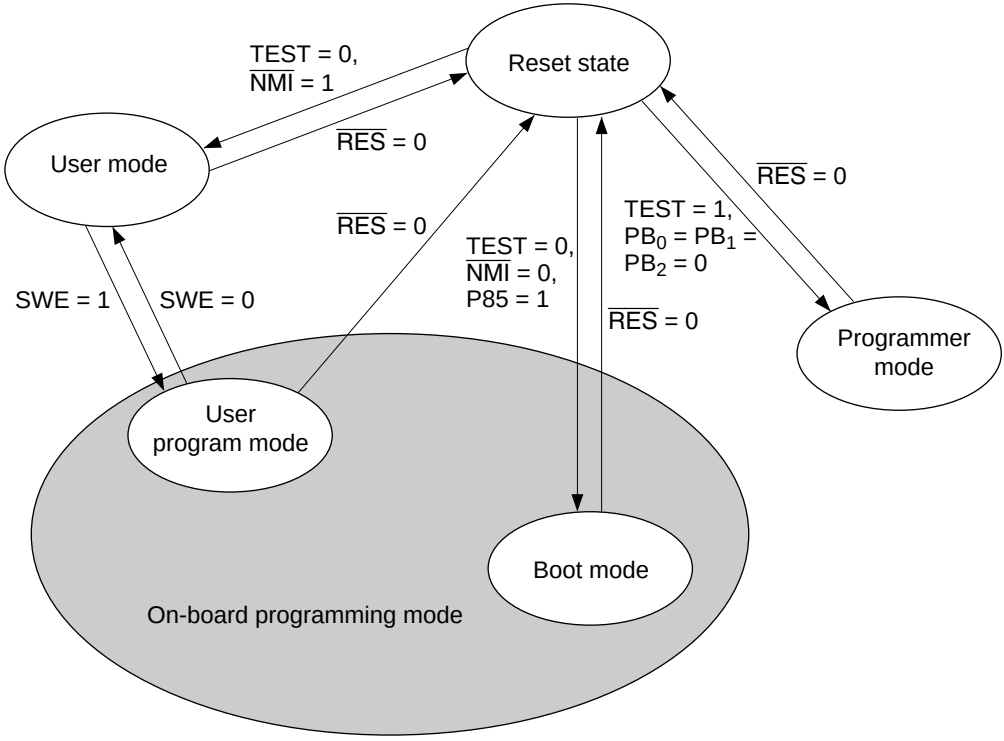
Notation:
FLMCR1: Flash memory control register 1
FLMCR2: Flash memory control register 2
EBR1: Erase block register 1
FLER: Flash memory enable register
FLPWCR: Flash memory power control register

Figure 7.1 Block Diagram of Flash Memory

7.2.2 On-board Programming Mode

For flash memory program/erase mode, boot mode and user program mode, in which on-board program/erase is performed, and programmer mode, in which program/erase is performed using a PROM programmer, are available. When the HD64F3664 is activated from the reset state, the HD64F3664 enters different operating modes, depending on the state of the TEST pin, the $\overline{\text{NMI}}$ pin, and the port input level as shown in figure 7.2.

The input level of the TEST pin and $\overline{\text{NMI}}$ pin must be stabilized at least four states before reset cancellation.



Note: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.

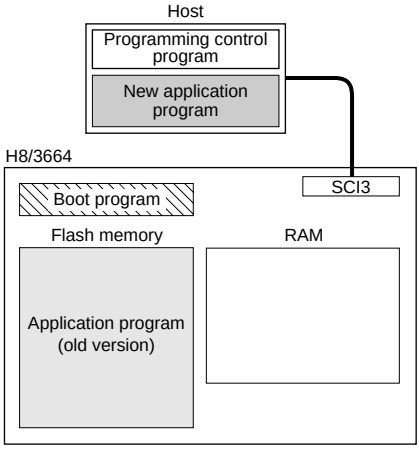
Figure 7.2 Flash Memory State Transitions

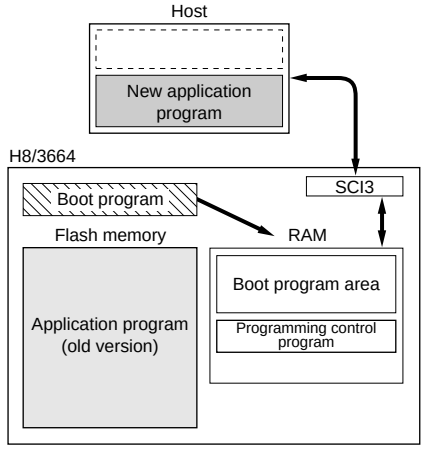
When the boot mode is entered, the boot program built in the LSI is activated. The boot mode is shown in figure 7.3. In this mode, the programming control program prepared in the host is received via SCI3, and is serially transmitted to the programming control program area of the on-chip RAM. The old data in the flash memory is entirely erased, the program branches to the start address of the programming control program area, and the execution of the programming control program is started. Prepare the programming control program according to section 7.6, Programming/erasing Flash Memory Program.

In user program mode, control branches from the user mode to the programming control program prepared by the user; desired blocks can be erased and programmed in this mode. The user must set the branch condition and prepare the means for providing on-board programming data. Since the flash memory cannot be read during program/erase, execute the programming control program after transmitting the program to the on-chip RAM, as in boot mode. Figure 7.4 shows an example of user program mode.

1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.

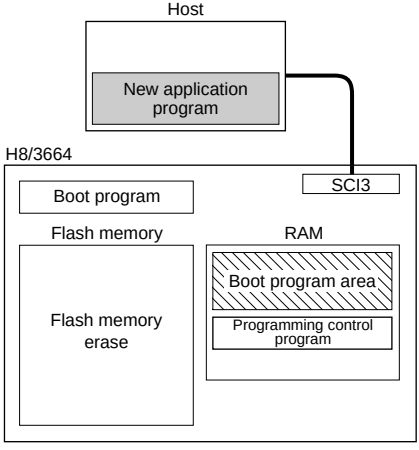
2. Programming control program transfer
   When boot mode is entered, the boot program in the H8/3664 (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot program area.

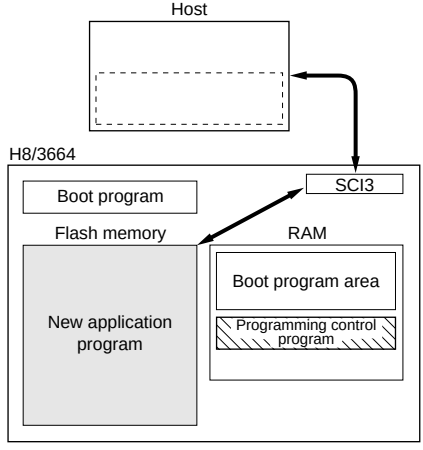




3. Flash memory initialization
   The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, the entire flash memory area is erased, without regard to blocks.

4. Writing new application program
   The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.



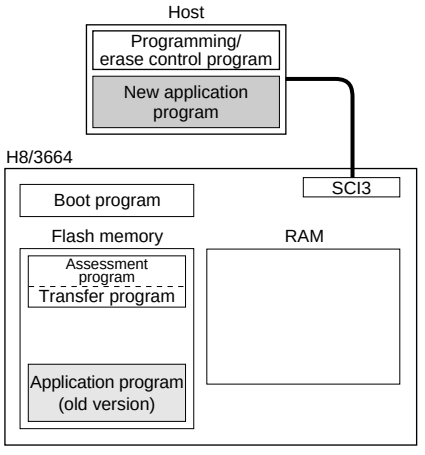


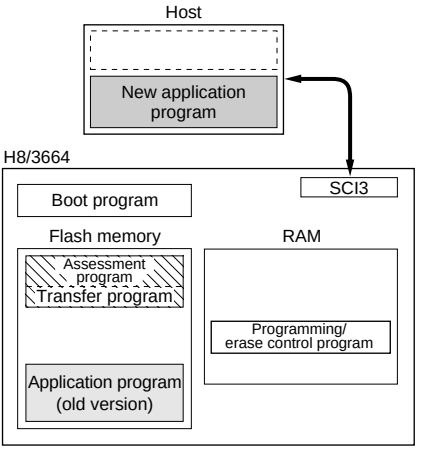
Program execution state

Figure 7.3 Boot Mode

1. Initial state
   The means for entering the user program mode has been prepared, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control program should be prepared in the host or in the flash memory.

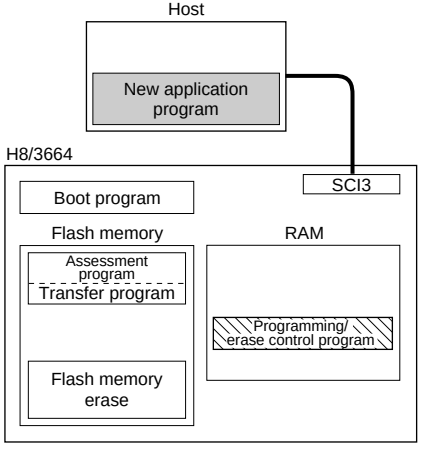
2. Programming/erase control program transfer
   When user program mode is entered, user software confirms this fact, executes transfer program in the flash memory, and transfers the programming/erase control program to RAM.





3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.

4. Writing new application program
   Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.



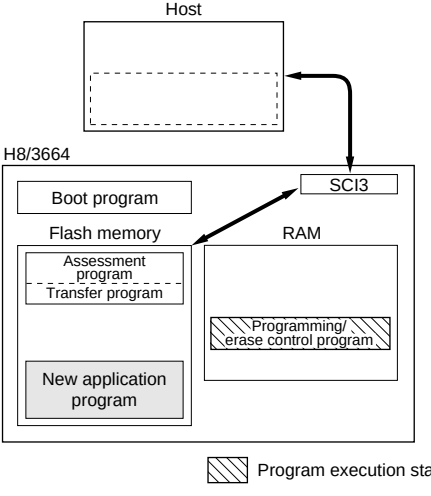
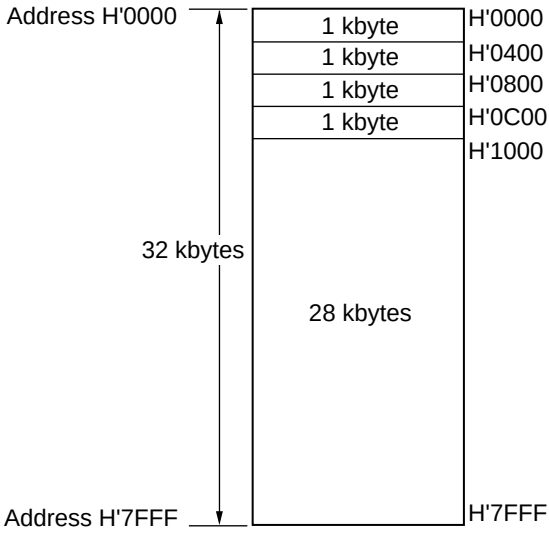


Figure 7.4 User Program Mode

7.2.3 Block Configuration

The flash memory is divided into four 1-kbyte blocks and one 28-kbyte block. Erasure is performed in this unit.



7.2.4 Pin Configuration

The flash memory is controlled by means of the pins shown in table 7.1.

Table 7.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Reset	$\overline{\text{RES}}$	Input	Reset
Mode	TEST	Input	Sets LSI operating mode
Port PB0	PB0	Input	Sets LSI operating mode when TEST = 1
Port PB1	PB1	Input	Sets LSI operating mode when TEST = 1
Port PB2	PB2	Input	Sets LSI operating mode when TEST = 1
Transmit data	TxD	Output	Serial transmit data output
Receive data	RxD	Input	Serial receive data input

7.2.5 Register Configuration

The registers used to control the on-chip flash memory when enabled are shown in table 7.2.

Table 7.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address
Flash memory control register 1	FLMCR1	R/W*1	H'00	H'FF90
Flash memory control register 2	FLMCR2	R*1	H'00	H'FF91
Erase block register 1	EBR1	R/W*1	H'00*2	H'FF93
Flash memory power control register	FLPWCR	R/W*1	H'00*2	H'FF92
Flash memory enable register	FENR	R/W	H'00	H'FF9B

Notes: 1. To access these registers, set the FLSHE bit to 1 in the flash memory enable register.
2. When the SWE bit of FLMCR1 is not set, these registers are initialized to H'00.

7.3 Register Descriptions

7.3.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is an 8-bit register used for flash memory operating mode control. Program-verify mode or erase-verify mode is entered by setting the SWE bit to 1, then setting the PV or EV bit. Program mode is entered by setting the SWE bit to 1, then setting the PSU bit, and finally setting the P bit. Erase mode is entered by setting the SWE bit to 1, then setting the ESU bit, and finally setting the E bit. FLMCR1 is initialized to H'00 by a power-on reset, or is initialized in standby mode.

Bit	7	6	5	4	3	2	1	0
	—	SWE	ESU	PSU	EV	PV	E	P
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Reserved: This bit always reads 0.

Bit 6—Software Write Enable Bit (SWE): Enables or disables flash memory programming and erasing. Set this bit before setting bits 5 to 0 of this register and bits 4 to 0 of EBR1.

Bit 6: SWE	Description	
0	Write disabled	(Initial value)
1	Write enabled	

Bit 5—Erase Setup Bit (ESU): Prepares for a transition to erase mode. Set this bit to 1 before setting the E bit in FLMCR1 to 1. Do not set the SWE, PSU, EV, PV, E, or P bit at the same time.

Bit 5: ESU	Description	
0	Erase setup cleared	(Initial value)
1	Erase setup [Setting condition] When SWE = 1	

Bit 4—Program Setup Bit (PSU): Prepares for a transition to program mode. Set this bit to 1 before setting the P bit in FLMCR1 to 1. Do not set the SWE, ESU, EV, PV, E, or P bit at the same time.

Bit 4: PSU	Description	
0	Program setup cleared	(Initial value)
1	Program setup [Setting condition] When SWE = 1	

Bit 3—Erase-Verify (EV): Selects erase-verify mode transition or clearing. Do not set the SWE, ESU, PSU, PV, E, or P bit at the same time.

Bit 3: EV	Description	
0	Erase-verify mode cleared	(Initial value)
1	Transition to erase-verify mode [Setting condition] When SWE = 1	

Bit 2—Program-Verify (PV): Selects program-verify mode transition or clearing. Do not set the SWE, ESU, PSU, EV, E, or P bit at the same time.

Bit 2: PV	Description	
0	Program-verify mode cleared	(Initial value)
1	Transition to program-verify mode [Setting condition] When SWE = 1	

Bit 1—Erase (E): Selects erase mode transition or clearing. Do not set the SWE, ESU, PSU, EV, PV, or P bit at the same time.

Bit 1: E	Description	
0	Erase mode cleared	(Initial value)
1	Transition to erase mode [Setting condition] When SWE = 1 and ESU = 1	

Bit 0—Program (P): Selects program mode transition or clearing. Do not set the SWE, PSU, ESU, EV, PV, or E bit at the same time.

Bit 0: P	Description	
0	Program mode cleared	(Initial value)
1	Transition to program mode [Setting condition] When SWE = 1 and PSU = 1	

7.3.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is an 8-bit register used for indicating flash memory program/erase status. FLMCR2 is initialized to H'00 by a power-on reset or is initialized in standby mode.

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

Note: FLMCR2 is a read-only register, and must not be written to.

Bit 7—Flash Memory Error (FLER): Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state.

Bit 7: FLER	Description
0	Flash memory is operating normally (Initial value) Flash memory program/erase protection (error protection) is disabled [Clearing condition] Power-on reset or standby mode
1	An error has occurred during flash memory programming/erasing Flash memory program/erase protection (error protection) is enabled [Setting condition] See section 7.7.3, Error Protection

Bits 6 to 0—Reserved: These bits always read 0.

7.3.3 Erase Block Register 1 (EBR1)

EBR1 is an 8-bit register that specifies the flash memory erase area block by block. EBR1 is initialized to H'00 by a power-on reset, or in standby mode, or when the SWE bit in FLMCR1 is 0. When a bit in EBR1 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Do not set more than one bit, as this will cause all the bits in EBR1 to be automatically cleared to 0.

The flash memory erase blocks are shown in table 7.3.

Bit	7	6	5	4	3	2	1	0
	—	—	—	EB4	EB3	EB2	EB1	EB0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.3 Flash Memory Erase Blocks

Block (Size)	Addresses
EB0 (1 kbyte)	H'0000–H'03FF
EB1 (1 kbyte)	H'0400–H'07FF
EB2 (1 kbyte)	H'0800–H'0BFF
EB3 (1 kbyte)	H'0C00–H'0FFF
EB4 (28 kbytes)	H'1000–H'7FFF

7.3.4 Flash Memory Power Control Register (FLPWCR)

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

FLPWCR enables or disables a transition to the flash memory power-down mode when the LSI switches to subactive mode.

Bit 7—Power-Down Disable (PDWND): Enables or disables a transition to the flash memory power-down mode when the LSI switches to subactive mode.

Bit 7: PDWND	Description	
0	Transition to flash memory power-down mode enabled	(Initial value)
1	Transition to flash memory power-down mode disabled	

Bits 6 to 0—Reserved: These bits always read 0.

7.3.5 Flash Memory Enable Register (FENR)

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

FENR is an 8-bit readable/writable register that controls on-chip flash memory.

FENR is initialized to H'00 by a reset or in standby mode.

Bit 7—Flash Memory Control Register Enable (FLSHE): Controls CPU access to the flash memory control registers (FLMCR1, FLMCR2, EBR1, and FLPWCR). Setting the FLSHE bit to 1 enables read/write access to the flash memory control registers. If FLSHE is cleared to 0, the flash memory control registers cannot be accessed. In this case, the flash memory control register contents are retained.

Bit 7: FLSHE	Description
0	Flash memory control registers in area H'FF90 to H'FF93 cannot be accessed (Initial value)
1	Flash memory control registers in area H'FF90 to H'FF93 can be accessed

Bits 6 to 0—Reserved: Write 0 when writing.

7.4 Boot Mode

When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. The SCI3 to be used is set to asynchronous mode. The transmission/receive format is 8-bit data, 1 stop-bit, and no parity.

When the boot program built into the LSI is activated, the bit rate of SCI3 is set according to the host bit rate. Next, the programming control program prepared in the host is received via SCI3, and is serially transmitted to the programming control program area of the on-chip RAM. The old data in the flash memory is entirely erased, the program branches to the start address of the programming control program area, and the execution of the programming control program is started.

When the control branches to the programming control program, SCI3 terminates the transmission/receive operation (RE and TE of SCR are 0). However, since the bit rate of SCI3 that has been set is retained, SCI3 can be used continuously for transmission/receive of programming data or verify data. The TxD pin is in the high-level output state (PCR22 = PDR22 = 1). The values of the general-purpose registers of the CPU immediately after the branch to the programming control program are undefined. In particular, since the stack pointer (SP) is used for subroutine calls, specify the stack area at the beginning of the programming control program.

The system configuration in boot mode is shown in figure 7.5, and the boot mode execution procedure is shown in figure 7.6.

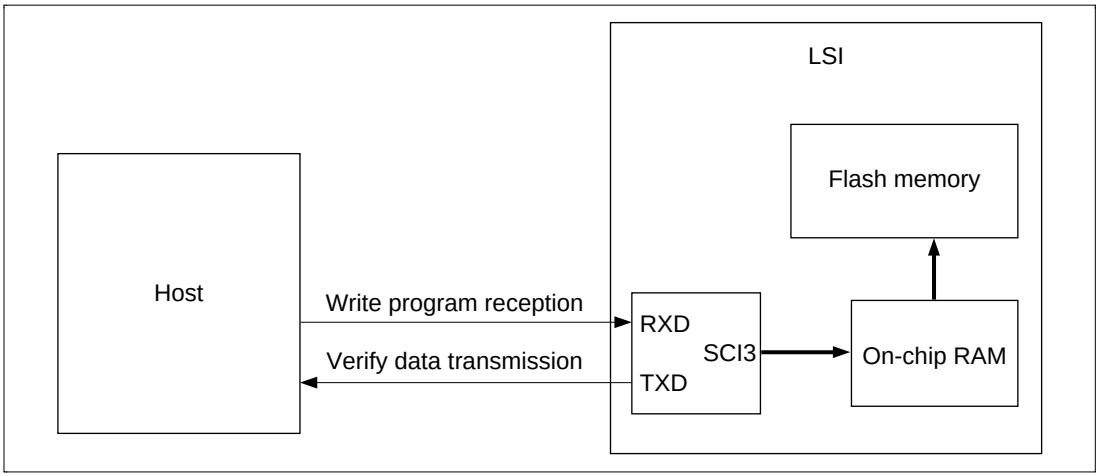
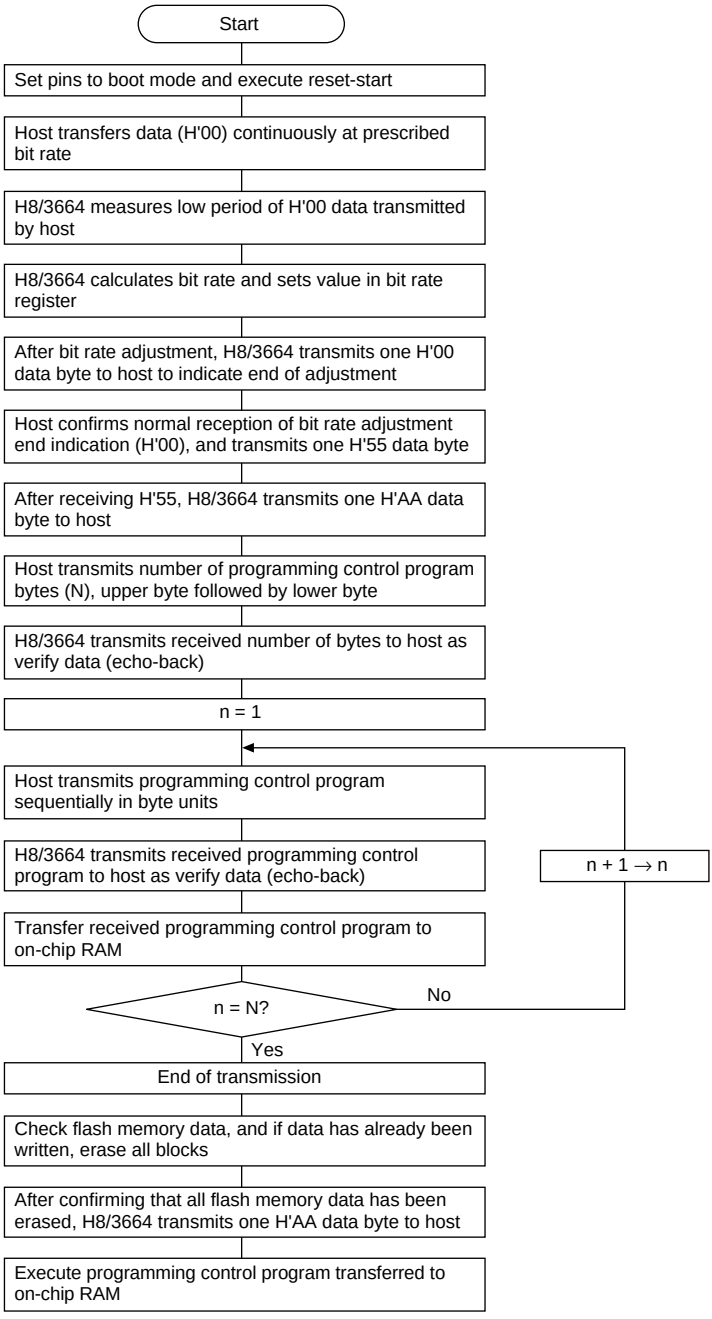


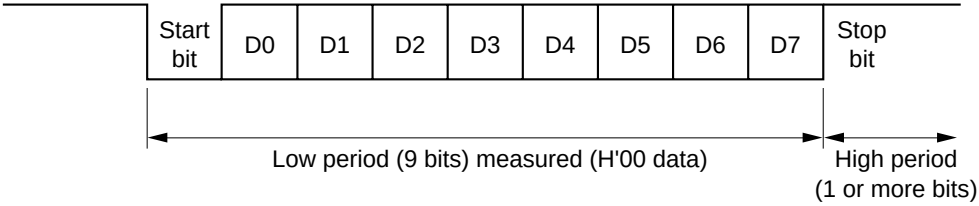
Figure 7.5 System Configuration in Boot Mode



Note: If a memory cell does not operate normally and cannot be erased, one H'FF byte is transmitted as an erase error, and the erase operation and subsequent operations are halted.

Figure 7.6 Boot Mode Execution Procedure

7.4.1 Automatic SCI Bit Rate Adjustment



When the boot program is initiated, the boot program measures the low period of the asynchronous SCI communication data (H'00) transmitted continuously from the host. After the reset ends, it takes approximately 100 states before the chip is ready to measure the low-level period. The boot program adjusts the bit rate of the SCI3 to the bit rate of the transmission from the host and transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the LSI. If reception cannot be performed normally, initiate boot mode again (reset), and repeat the above operations. Depending on the host's transmission bit rate and the LSI's system clock frequency, there will be a discrepancy between the bit rates of the host and the LSI. Set the host transfer bit rate and the system clock frequencies within the range shown in Table 7.4.

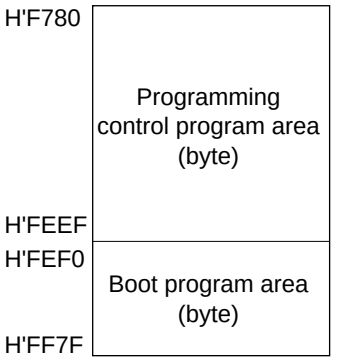
Table 7.4 shows host transfer bit rates and system clock frequencies for which automatic adjustment of the LSI bit rate is possible.

Table 7.4 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible

Host Bit Rate	System Clock Frequency for Which Automatic Adjustment of LSI Bit Rate is Possible
19200 bps	16 MHz
9600 bps	8 to 16 MHz
4800 bps	4 to 16 MHz
2400 bps	2 to 16 MHz

7.4.2 Programming Control Program Area

In boot mode, a part of the RAM area is used by the boot program. An area to which the programming control program is transferred is shown in figure 7.7. The boot program area cannot be used until the execution state in boot mode switches to the programming control program transferred from the host.



Note: The boot program area cannot be used until a transition is made to the execution state for the programming control program transferred to RAM. Note also that the boot program remains in this area of the on-chip RAM even after control branches to the programming control program.

Figure 7.7 RAM Areas in Boot Mode

7.4.3 Notes on Use of Boot Mode

1. When the boot program is initiated, it measures the low-level period of the input at the SCI3's RxD pin. The reset should end with RxD high. The RxD and TxD pins should be pulled up on the board if necessary.
2. Do not change input levels of the TEST pin and the $\overline{\text{NMI}}$ pin in boot mode.
3. Boot mode can be cleared by driving the reset pin low, waiting at least 20 states, then setting the TEST pin and $\overline{\text{NMI}}$ pin, and executing reset release. Boot mode can also be cleared by a WDT overflow reset.
4. If the TEST pin input levels are changed (for example, from low to high) during a reset, the state of ports will change according to the change in the microcomputer's operating mode*. Therefore, care must be taken to make pin settings to prevent these pins from becoming output signal pins during a reset, or to prevent collision with signals outside the microcomputer.

Note: * See appendix D, Pin States in the Different Processing States.

7.5 User Program Mode

The on-board reprogramming of the on-chip flash memory in user program mode can be carried out by providing on-board means for supplying programming data, and by storing a program/erase control program or a program to supply it from outside in part of the flash memory as necessary.

In this mode, peripheral functions other than flash memory operate as they normally would in user mode.

The flash memory itself cannot be read while the SWE bit is set to 1 to perform programming or erasing, so the control program that performs programming and erasing should be run in on-chip RAM.

Figure 7.8 shows the procedure for executing the program/erase control program when transferred to on-chip RAM.

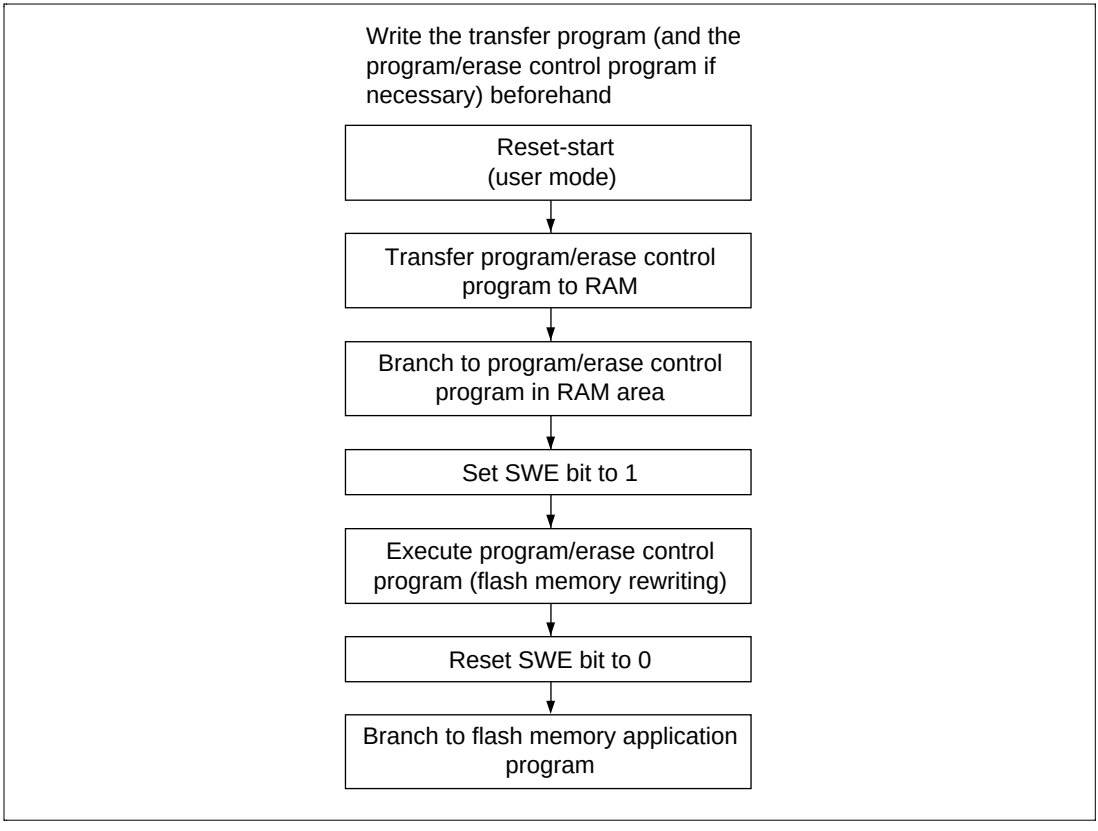


Figure 7.8 User Program Mode Execution Procedure

7.6 Programming/Erasing Flash Memory

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. There are flash memory operating modes: program mode, program-verify mode, erase mode, and erase-verify mode. Transitions to these modes are made by setting FLMCR1. The programming control program and the program/erase control program in on-board programming mode are used by combining these operating modes. The programming into the flash memory should be followed by “7.6.1 Program/Program-Verify Mode” and the erasing of the flash memory should be followed by “7.6.2 Erase/Erase-Verify Mode”.

Notes: 1. Operation is not guaranteed if bits SWE, ESU, PSU, EV, PV, E, and P of FLMCR1 are set/reset by a program in flash memory in the corresponding address areas.

7.6.1 Program/Program-Verify

To write to flash memory, follow the program/program-verify flowchart shown in figure 7.9. Performing programming operations according to this flowchart will enable flash memory to be programmed without subjecting the device to voltage stress or sacrificing data reliability.

1. Perform programming in the erased state. Do not reprogram addresses that have already been programmed.
2. The unit for one programming operation is 128 bytes. A 128-byte data transfer must be performed even if writing fewer than 128 bytes of data; in this case, write H'FF data to addresses that do not need to be programmed.
3. Provide a 128-byte program data area, 128-byte reprogram data area, and 128-byte additional-program data area in RAM. Refer to tables 7.5 and 7.6 for reprogram data and additional-program data computation.
4. Write 128 bytes successively in byte units from the reprogram data area or additional-program data area to flash memory. The program address and 128-byte data are latched in the flash memory. The lower 8 bits of the flash memory start address must be H'00 or H'80.
5. The time during which the P bit is set is the programming time. Set the programming time according to table 7.7.
6. A watchdog timer setting is made to prevent overprogramming due to program runaway, etc. Set an overflow period of around 6.6 ms.
7. For the dummy write to verify addresses, write one H'FF byte to addresses with b'00 in the lower 2 bits. Verify data can be read in longword format from the address on which the dummy write was executed.
8. Do not repeat the program/program-verify sequence more than 1,000 times on the same bit.

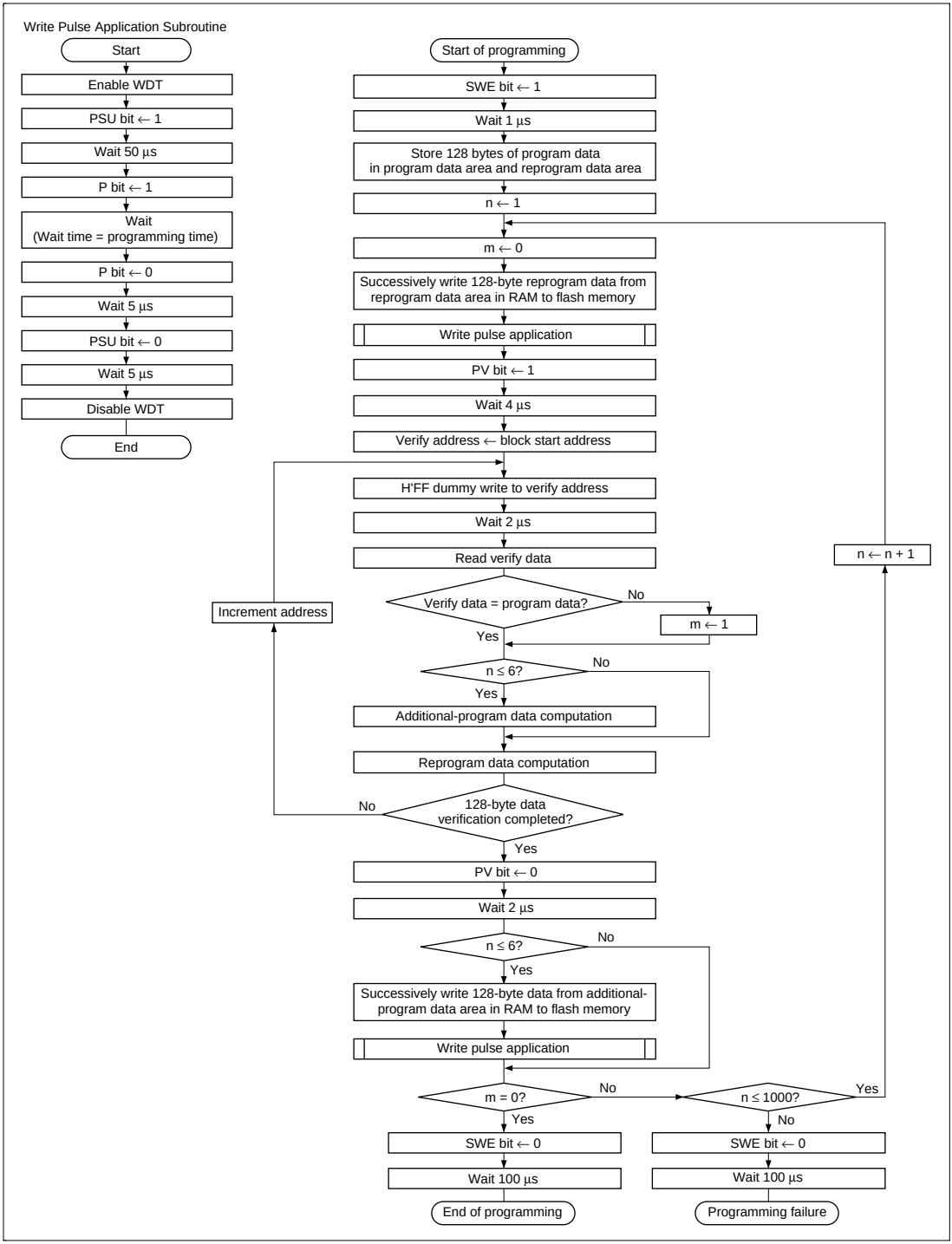


Figure 7.9 Program/Program-Verify Flowchart

Table 7.5 Reprogram Data Computation Table

Program Data	Verify Data	Reprogram Data	Comments
0	0	1	Programmed bit
0	1	0	Reprogram bit
1	0	1	—
1	1	1	Remains in erased state

Table 7.6 Additional-Program Data Computation Table

Reprogram Data	Verify Data	Additional-Program Data	Comments
0	0	0	Additional-program bit
0	1	1	No additional programming
1	0	1	No additional programming
1	1	1	No additional programming

Table 7.7 Programming Time

n (Number of Writes)	Programming Time	In Additional Programming	Comments
1 to 6	30	10	
7 to 1,000	200	—	

Note: Time shown in µs.

7.6.2 Erase/Erase-Verify

To erase flash memory, follow the erase/erase-verify flowchart shown in figure 7.10.

1. Prewriting (setting all data in the memory to be erased to all 0s) is not necessary before erasing.
2. Erasing is performed on one block at a time. Select the block to be erased in erase block register 1 (EBR1). When erasing a number of blocks, each block must be erased in turn.
3. The time during which the E bit is set is the erase time.
4. A watchdog timer setting is made to prevent overerasing due to program runaway, etc. Set an overflow period of around 19.8 ms.
5. For the dummy write to verify addresses, write one H'FF byte to addresses with b'00 in the lower 2 bits. Verify data can be read in longword format from the address on which the dummy write was executed.
6. If the read data is unerased, set erase mode again and repeat the erase/erase-verify sequence as before. However, ensure that this sequence is not repeated more than 100 times.

7.6.3 Interrupts during Flash Memory Programming/Erasing

For the following reasons, all interrupts, including $\overline{\text{NMI}}$, should be disabled when flash memory is being programmed or erased.

1. If an interrupt occurred during programming or erasing, it would not be possible to guarantee normal operation in accordance with the programming or erasing algorithm.
2. If interrupt exception handling were initiated during programming or erasing before the vector start address had been written, a normal vector fetch would not be possible and CPU runaway would result.
3. If an interrupt occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.

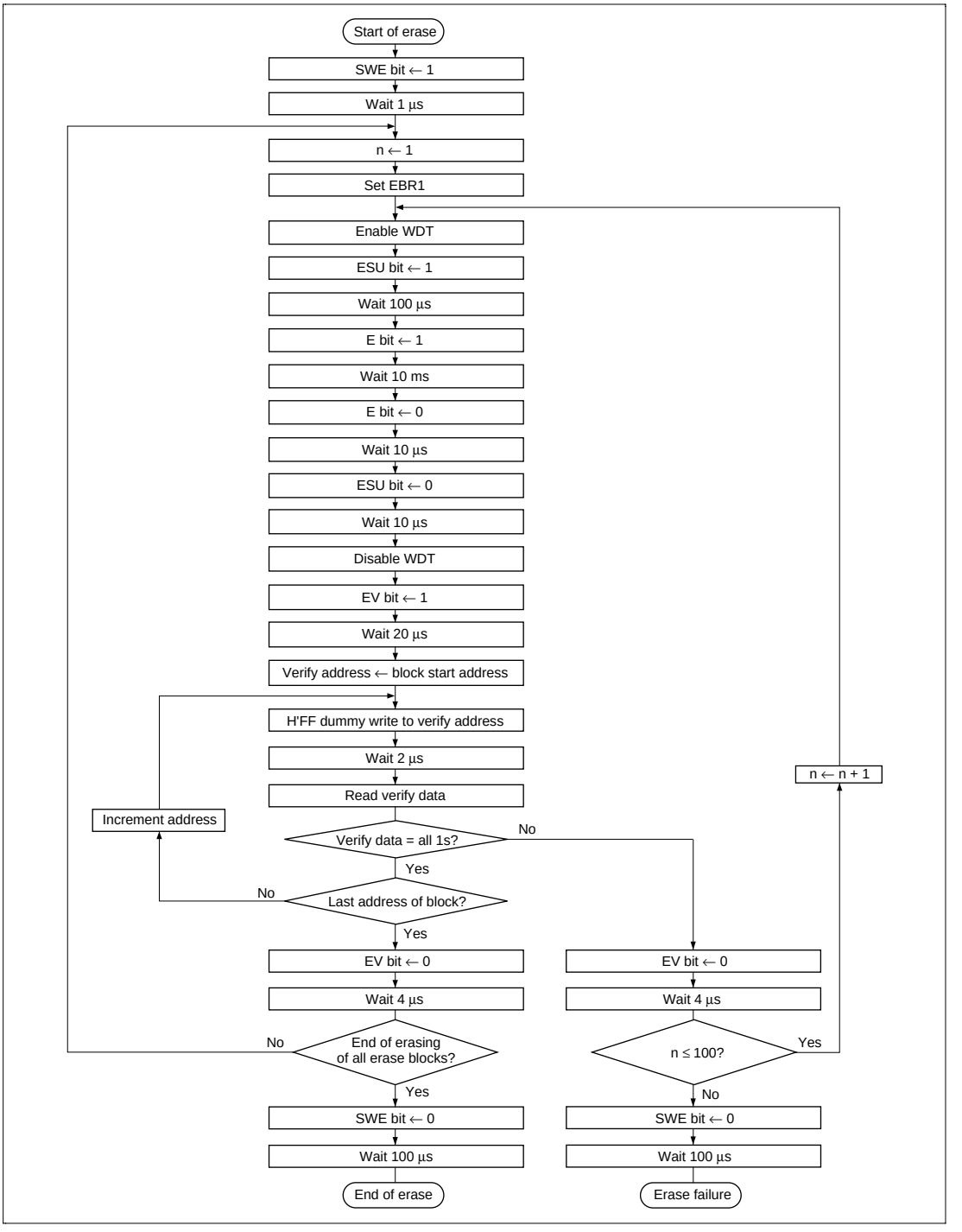


Figure 7.10 Erase/Erase-Verify Flowchart

7.7 Protection

There are three kinds of flash memory program/erase protection: hardware protection, software protection, and error protection.

7.7.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. Hardware protection is reset by settings in flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), and erase block register 1 (EBR1).

Table 7.8 Hardware Protection

		Functions	
Item	**Description**	**Program**	**Erase**
Reset/standby protection	• In a power-on reset (including a WDT power-on reset) and in standby mode, FLMCR1, FLMCR2, and EBR1 are initialized, and the program/erase-protected state is entered. • In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.	Yes	Yes

7.7.2 Software Protection

Software protection can be implemented by clearing the SWE bit in FLMCR1, and by clearing each bit in erase block register 1 (EBR1). When software protection is in effect, setting the P or E bit in flash memory control register 1 (FLMCR1), does not cause a transition to program mode or erase mode. (See table 7.9.)

Table 7.9 Software Protection

Item	Description	Functions	
		Program	Erase
SWE bit protection	• Setting bit SWE in FLMCR1 to 0 will place all blocks in the program/erase-protected state. (Execute the program in the on-chip RAM/external memory)	Yes	Yes
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block register 1 (EBR1) • Setting EBR1 to H'00 places all blocks in the erase-protected state.	—	Yes

7.7.3 Error Protection

In error protection, an error is detected when CPU runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is forcibly aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

If the CPU malfunctions during flash memory programming/erasing, the FLER bit is set to 1 in FLMCR2 and the error protection state is entered.

(1) The flash memory is read during programming/erasing (including vector read and instruction fetch)

(2) Exception handling (excluding a reset) during programming/erasing

(3) SLEEP instruction (including software standby) is executed during programming/erasing

The FLMCR1, FLMCR2, and EBR1 settings are retained, but program mode or erase mode is aborted at the point at which the error is detected. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit setting is enabled, and a transition can be made to verify mode. Error protection is released only by a power-on reset and in hardware standby mode.

Figure 7.11 shows the flash memory state transition diagram.

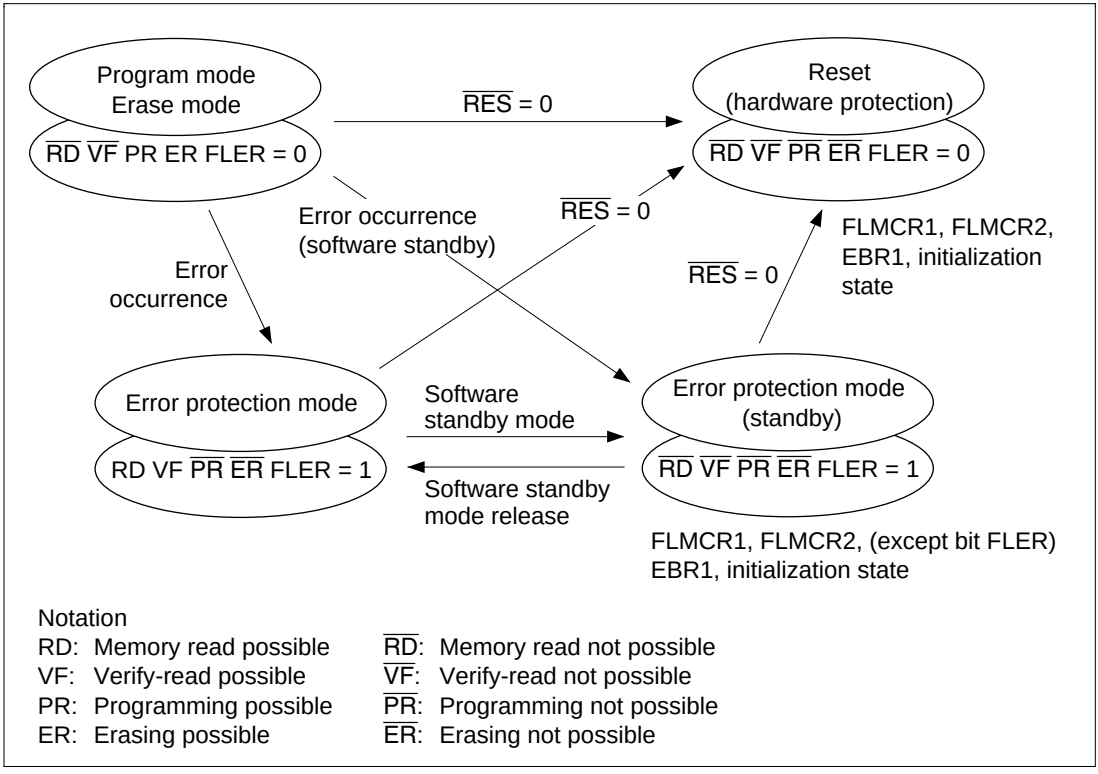


Figure 7.11 Flash Memory State Transitions

7.8 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including NMI interrupt is disabled when flash memory is being programmed or erased (when the P or E bit is set in FLMCR1), and while the boot program is executing. There are three reasons for this:

(1) Interrupt during programming or erasing might cause a violation of the programming or erasing algorithm, with the result that normal operation could not be assured.

(2) In the interrupt exception handling sequence during programming or erasing, the vector would not be fetched correctly, possibly resulting in MCU runaway.

(3) If interrupt occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.

Notes: 1. Interrupt requests must be disabled inside and outside the MCU until the programming control program has completed programming.

7.9 Flash Memory and Power-Down States

There are three flash memory operating states in user mode:

(1) Normal operating mode: The flash memory can be read and written to at high speed.
(2) Power-down mode: Part of the power supply circuitry is halted, and the flash memory can be read when the LSI is operating in the subactive mode.
(3) Standby mode: When the LSI transits to standby mode, all flash memory circuits are halted, and the flash memory cannot be read or written to.

Table 7.10 shows the correspondence between the operating states of the LSI and the flash memory.

The flash memory is transited to power-down mode if the LSI is transited to the subactive mode while the PDWND bit in FLPWCR is cleared.

When the flash memory is in a power-down state or in standby mode, part or all of the internal power supply circuitry is halted. Therefore, a power supply circuit stabilization period must be provided when returning to normal operation. When the flash memory returns to its normal operating state from a power-down state, bits STS2 to STS0 in SBYCR must be set to provide a wait time of at least 20 µs including when using an external clock.

Table 7.10 Flash Memory Operating States

	Flash Memory Operating State	
LSI Operating State	**When PDWND = 0 (Initial value)**	**When PDWND = 1**
Active mode	Normal mode	Normal mode
Subactive mode	Power-down mode	Normal mode
Sleep mode	Normal mode	Normal mode
Subsleep mode	Standby mode	Standby mode
Standby mode	Standby mode	Standby mode

7.10 Flash Memory Programmer Mode

Programs and data can be written and erased in programmer mode as well as in the on-board programming modes. In programmer mode, flash memory read mode, auto-program mode, auto-erase mode, and status read mode are supported. In auto-program mode, auto-erase mode, and status read mode, a status polling procedure is used, and in status read mode, detailed internal signals are output after execution of an auto-program or auto-erase operation.

In programmer mode, set the mode pins to programmer mode (see table 7.11) and input a 10 MHz input clock. Use a PROM programmer that supports the Hitachi 64-kbyte on-chip flash memory microcomputer device type (FZTAT64V5).

Table 7.11 shows the pin settings for programmer mode. For the pin names in programmer mode, see section 1.4, Pin Functions.

Table 7.11 PROM Mode Pin Settings

Pin Names	Settings
Mode pins: TEST	High level input to TEST
Mode setting pins: PB2, PB1, PB0	Low level input to PB2, PB1, and PB0
$\overline{\text{RES}}$ pin	Power-on reset circuit
OSC2, OSC1	Oscillator circuit

7.10.1 Socket Adapter Pin Correspondence Diagram

Connect the socket adapter to the chip as shown in figure 7.12. This will enable conversion to a 32-pin arrangement. The on-chip ROM memory map is shown in figure 7.12, and the socket adapter pin correspondence diagram in figure 7.13.

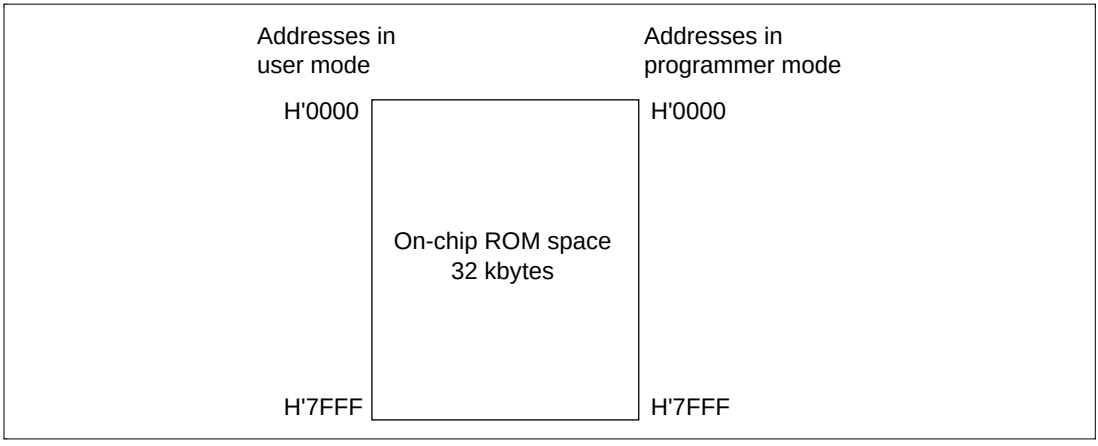


Figure 7.12 On-Chip ROM Memory Map

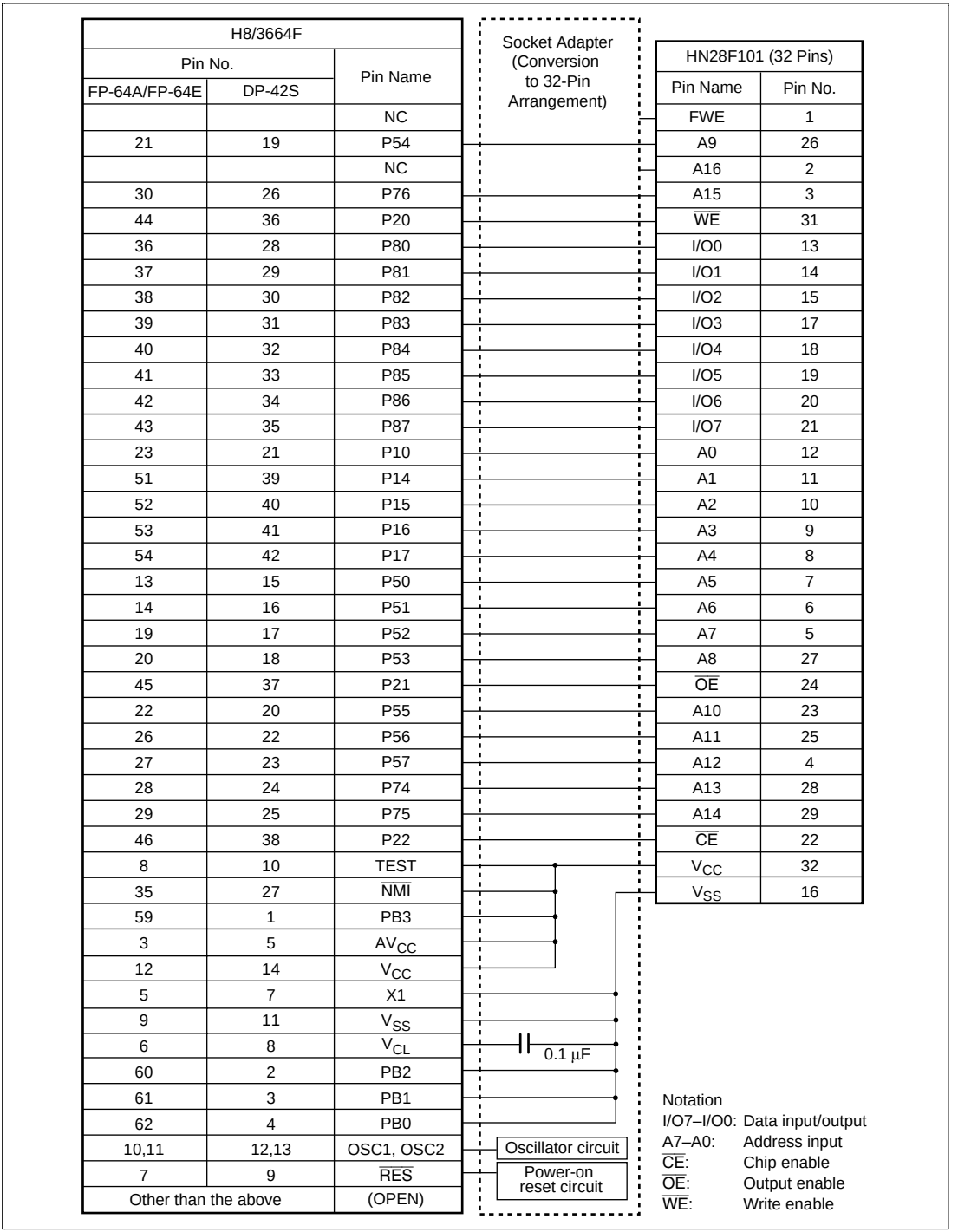


Figure 7.13 Socket Adapter Pin Correspondence Diagram

7.10.2 Programmer Mode Operation

Table 7.12 shows how the different operating modes are set when using programmer mode, and table 7.13 lists the commands used in programmer mode. Details of each mode are given below.

- Memory Read Mode

  Memory read mode supports byte reads.
- Auto-Program Mode

  Auto-program mode supports programming of 128 bytes at a time. Status polling is used to confirm the end of auto-programming.
- Auto-Erase Mode

  Auto-erase mode supports automatic erasing of the entire flash memory. Status polling is used to confirm the end of auto-programming.
- Status Read Mode

  Status polling is used for auto-programming and auto-erasing, and normal termination can be confirmed by reading the I/O6 signal. In status read mode, error information is output if an error occurs.

Table 7.12 Settings for Various Operating Modes In Programmer Mode

	Pin Names				
Mode	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{WE}$**	**I/O7– I/O0**	**A18–A0**
Read	L	L	H	Data output	Ain
Output disable	L	H	H	Hi-z	X
Command write	L	H	L	Data input	*Ain
Chip disable	H	X	X	Hi-z	X

Notes: Chip disable is not a standby state; internally, it is an operation state.
*Ain indicates that there is also address input in auto-program mode.

Table 7.13 Programmer Mode Commands

Command Name	Number of Cycles	1st Cycle			2nd Cycle		
		Mode	Address	Data	Mode	Address	Data
Memory read mode	1 + n	Write	X	H'00	Read	RA	Dout
Auto-program mode	129	Write	X	H'40	Write	WA	Din
Auto-erase mode	2	Write	X	H'20	Write	X	H'20
Status read mode	2	Write	X	H'71	Write	X	H'71

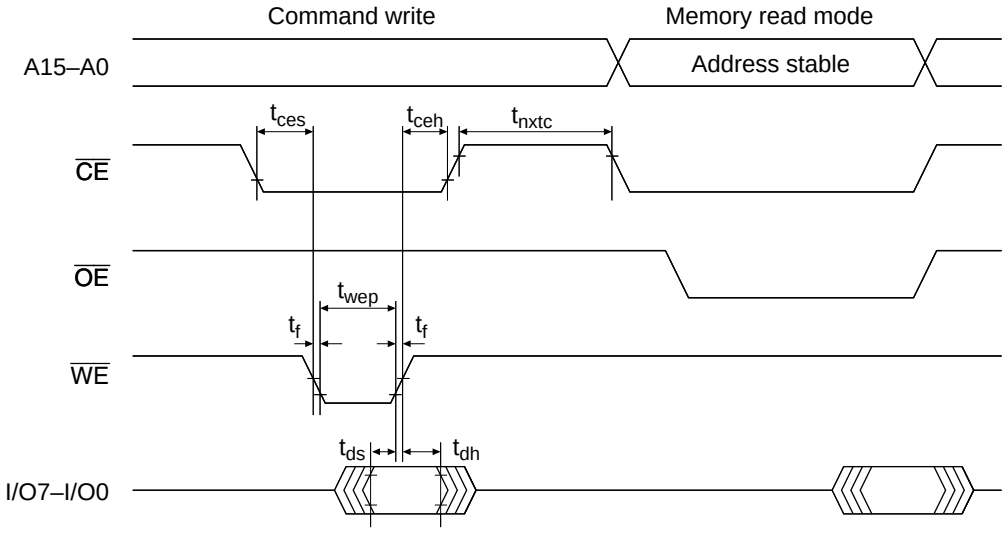
Notes: 1. In auto-program mode, 129 cycles are required for command writing by a simultaneous 128-byte write.
2. In memory read mode, the number of cycles depends on the number of address write cycles (n).

7.10.3 Memory Read Mode

1. After completion of auto-program/auto-erase/status read operations, a transition is made to the command wait state. When reading memory contents, a transition to memory read mode must first be made with a command write, after which the memory contents are read.
2. In memory read mode, command writes can be performed in the same way as in the command wait state.
3. Once memory read mode has been entered, consecutive reads can be performed.
4. After powering on, memory read mode is entered.

Table 7.14 AC Characteristics in Transition to Memory Read Mode (Conditions: V_{CC} = 5.0 V ±0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	Figure 7.14
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	



Note: Data is latched on the rising edge of $\overline{WE}$.

Figure 7.14 Timing Waveforms for Memory Read after Memory Write

Table 7.15 AC Characteristics in Transition from Memory Read Mode to Another Mode (Conditions: V_{CC} = 5.0 V ±0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20	—	µs	Figure 7.15
$\overline{CE}$ hold time	t_{ceh}	0	—	ns	
$\overline{CE}$ setup time	t_{ces}	0	—	ns	
Data hold time	t_{dh}	50	—	ns	
Data setup time	t_{ds}	50	—	ns	
Write pulse width	t_{wep}	70	—	ns	
$\overline{WE}$ rise time	t_r	—	30	ns	
$\overline{WE}$ fall time	t_f	—	30	ns	

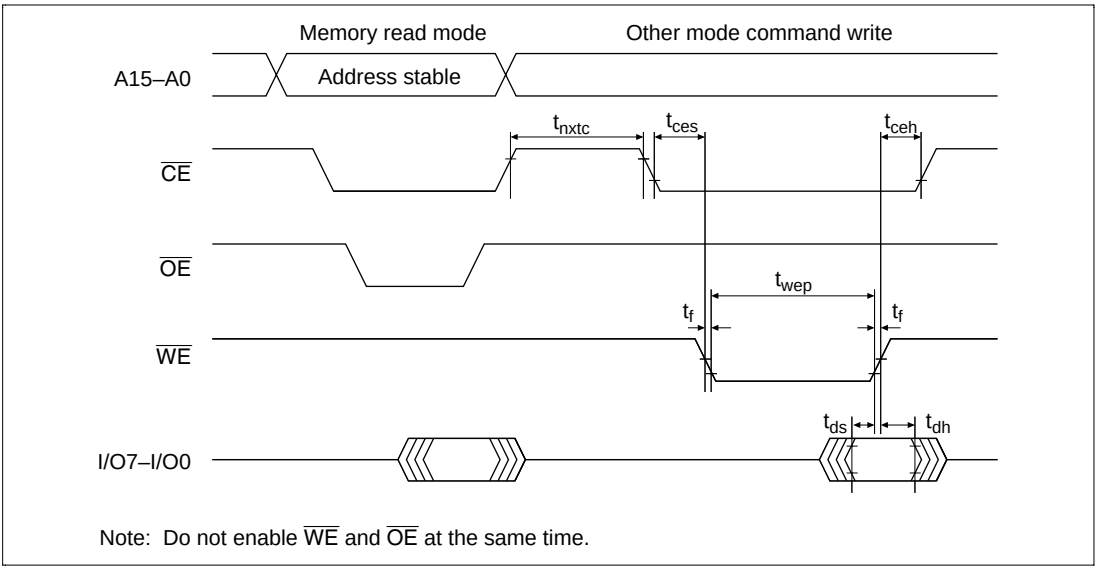


Figure 7.15 Timing Waveforms in Transition from Memory Read Mode to Another Mode

Table 7.16 AC Characteristics in Memory Read Mode (Conditions: V_{CC} = 5.0 V ±0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Access time	t_{acc}	—	20	µs	Figure 7.16
$\overline{CE}$ output delay time	t_{ce}	—	150	ns	Figure 7.17
$\overline{OE}$ output delay time	t_{oe}	—	150	ns	
Output disable delay time	t_{df}	—	100	ns	
Data output hold time	t_{oh}	5	—	ns	

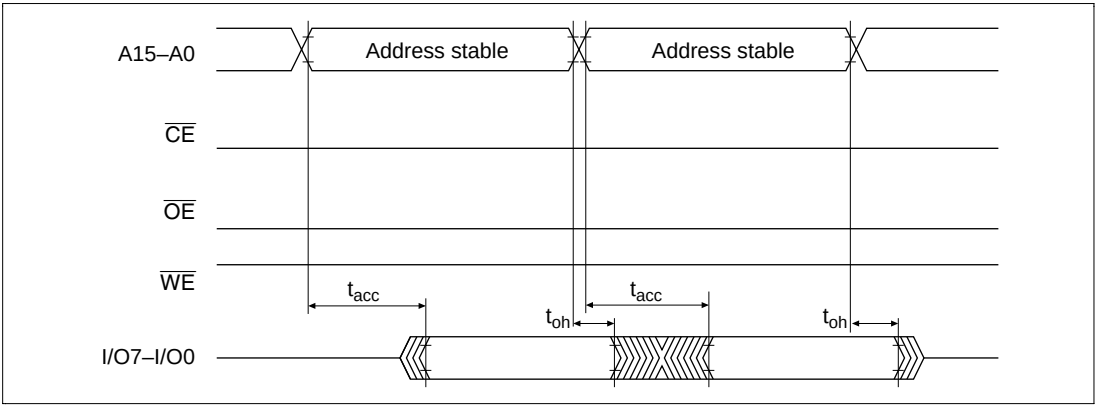


Figure 7.16 $\overline{CE}$ and $\overline{OE}$ Enable State Read Timing Waveforms

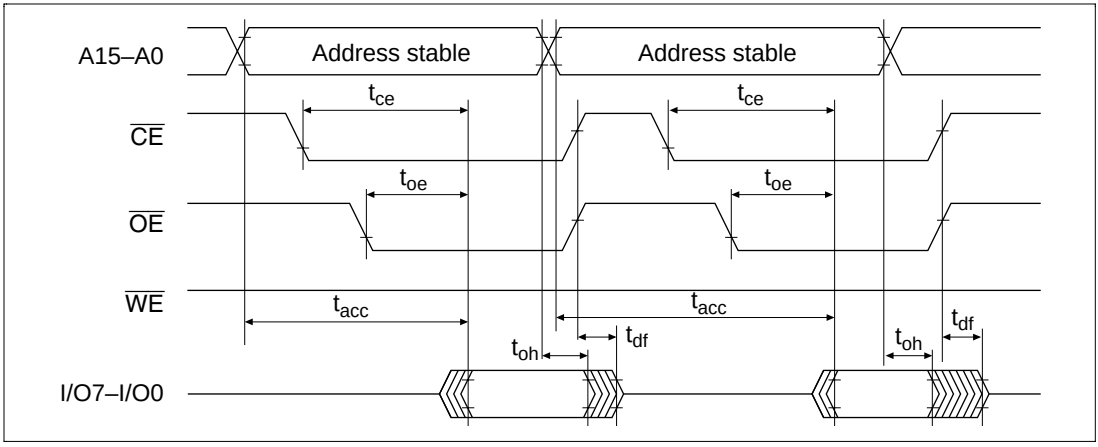


Figure 7.17 $\overline{CE}$ and $\overline{OE}$ Clock System Read Timing Waveforms

7.10.4 Auto-Program Mode

1. When reprogramming previously programmed addresses, perform auto-erasing before auto-programming.
2. Perform auto-programming once only on the same address block. It is not possible to program an address block that has already been programmed.
3. In auto-program mode, 128 bytes are programmed simultaneously. This should be carried out by executing 128 consecutive byte transfers.
4. A 128-byte data transfer is necessary even when programming fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
5. The lower 7 bits of the transfer address must be low. If a value other than an effective address is input, processing will switch to a memory write operation but a write error will be flagged.
6. Memory address transfer is performed in the second cycle (figure 7.18). Do not perform transfer after the third cycle.
7. Do not perform a command write during a programming operation.
8. Perform one auto-program operation for a 128-byte block for each address. Two or more additional programming operations cannot be performed on a previously programmed address block.
9. Confirm normal end of auto-programming by checking I/O6. Alternatively, status read mode can also be used for this purpose (I/O7 status polling uses the auto-program operation end decision pin).
10. Status polling I/O6 and I/O7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 7.17 AC Characteristics in Auto-Program Mode (Conditions: V_{CC} = 5.0 V ±0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20	—	µs	Figure 7.18
$\overline{CE}$ hold time	t_{ceh}	0	—	ns	
$\overline{CE}$ setup time	t_{ces}	0	—	ns	
Data hold time	t_{dh}	50	—	ns	
Data setup time	t_{ds}	50	—	ns	
Write pulse width	t_{wep}	70	—	ns	
Status polling start time	t_{wsts}	1	—	ms	
Status polling access time	t_{spa}	—	150	ns	
Address setup time	t_{as}	0	—	ns	
Address hold time	t_{ah}	60	—	ns	
Memory write time	t_{write}	1	3000	ms	
$\overline{WE}$ rise time	t_r	—	30	ns	
$\overline{WE}$ fall time	t_f	—	30	ns	

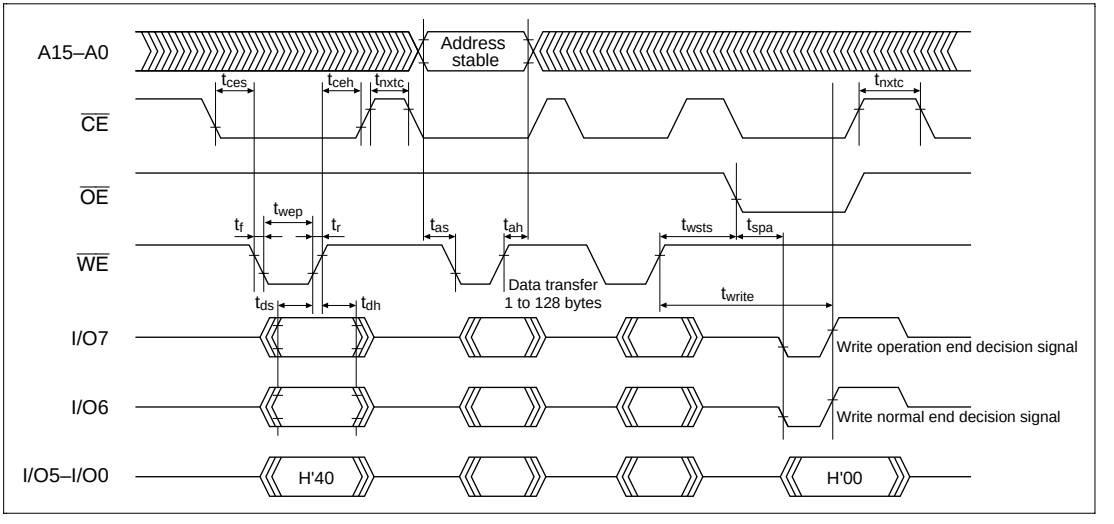


Figure 7.18 Auto-Program Mode Timing Waveforms

7.10.5 Auto-Erase Mode

1. Auto-erase mode supports only entire memory erasing.
2. Do not perform a command write during auto-erasing.
3. Confirm normal end of auto-erasing by checking I/O6. Alternatively, status read mode can also be used for this purpose (I/O7 status polling uses the auto-erase operation end decision pin).
4. Status polling I/O6 and I/O7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 7.18 AC Characteristics in Auto-Erase Mode (Conditions: V_{CC} = 5.0 V ±0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20	—	µs	Figure 7.19
$\overline{CE}$ hold time	t_{ceh}	0	—	ns	
$\overline{CE}$ setup time	t_{ces}	0	—	ns	
Data hold time	t_{dh}	50	—	ns	
Data setup time	t_{ds}	50	—	ns	
Write pulse width	t_{wep}	70	—	ns	
Status polling start time	t_{ests}	1	—	ms	
Status polling access time	t_{spa}	—	150	ns	
Memory erase time	t_{erase}	100	40000	ms	
$\overline{WE}$ rise time	t_r	—	30	ns	
$\overline{WE}$ fall time	t_f	—	30	ns	

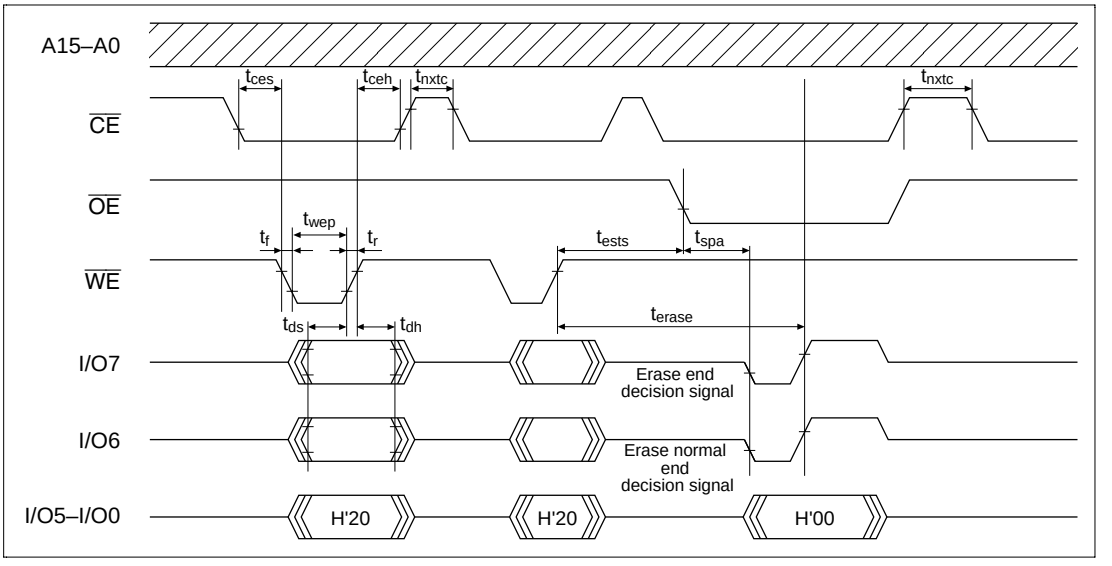


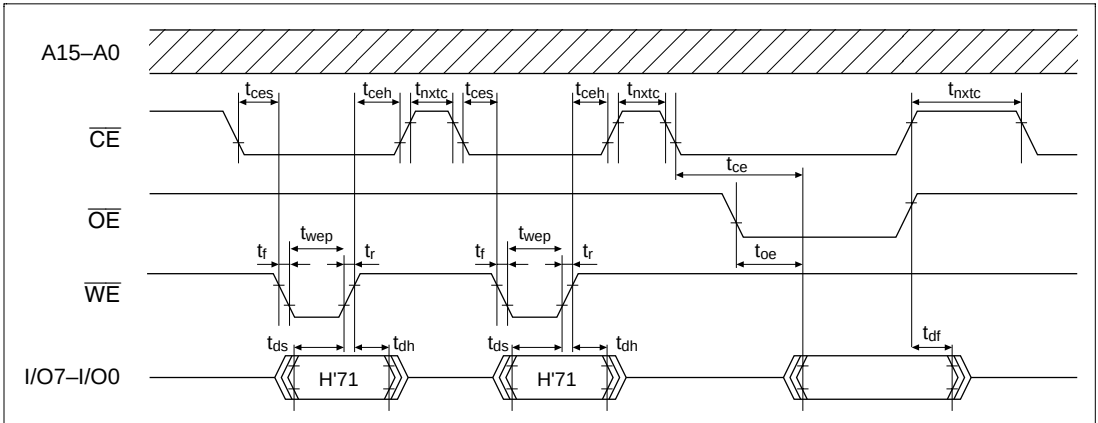
Figure 7.19 Auto-Erase Mode Timing Waveforms

7.10.6 Status Read Mode

1. Status read mode is provided to identify the kind of abnormal end. Use this mode when an abnormal end occurs in auto-program mode or auto-erase mode.
2. The return code is retained until a command write other than a status read mode command write is executed.

Table 7.19 AC Characteristics in Status Read Mode (Conditions: V_{CC} = 5.0 V ±0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Read time after command write	t_{nxtc}	20	—	µs	Figure 7.20
$\overline{CE}$ hold time	t_{ceh}	0	—	ns	
$\overline{CE}$ setup time	t_{ces}	0	—	ns	
Data hold time	t_{dh}	50	—	ns	
Data setup time	t_{ds}	50	—	ns	
Write pulse width	t_{wep}	70	—	ns	
$\overline{OE}$ output delay time	t_{oe}	—	150	ns	
Disable delay time	t_{df}	—	100	ns	
$\overline{CE}$ output delay time	t_{ce}	—	150	ns	
$\overline{WE}$ rise time	t_r	—	30	ns	
$\overline{WE}$ fall time	t_f	—	30	ns	



Note: I/O2 and I/O3 are undefined.

Figure 7.20 Status Read Mode Timing Waveforms

Table 7.20 Status Read Mode Return Commands

Pin Name	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
Attribute	Normal end decision	Command error	Program-ming error	Erase error	—	—	Program-ming or erase count exceeded	Effective address error
Initial value	0	0	0	0	0	0	0	0
Indications	Normal end: 0 Abnormal end: 1	Command error: 1 Otherwise: 0	Program-ming error: 1 Otherwise: 0	Erasing error: 1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0	Effective address error: 1 Otherwise: 0

Note: I/O2 and I/O3 are undefined.

7.10.7 Status Polling

1. The I/O7 status polling flag indicates the operating status in auto-program/auto-erase mode.
2. The I/O6 status polling flag indicates a normal or abnormal end in auto-program/auto-erase mode.

Table 7.21 Status Polling Output Truth Table

Pin Name	During Internal Operation	Abnormal End	—	Normal End
I/O7	0	1	0	1
I/O6	0	0	1	1
I/O0–I/O5	0	0	0	0

7.10.8 Programmer Mode Transition Time

Commands cannot be accepted during the oscillation stabilization period or the programmer mode setup period. After the programmer mode setup time, a transition is made to memory read mode.

Table 7.22 Stipulated Transition Times to Command Wait State

Item	Symbol	Min	Max	Unit	Notes
Standby release (oscillation stabilization time)	t_{osc1}	10/5	—	ms	10 ms: Crystal oscillator 5 ms: Ceramic oscillator
Programmer mode setup time	t_{bmv}	10	—	ms	
V_{CC} hold time	t_{dwn}	0	—	ms	

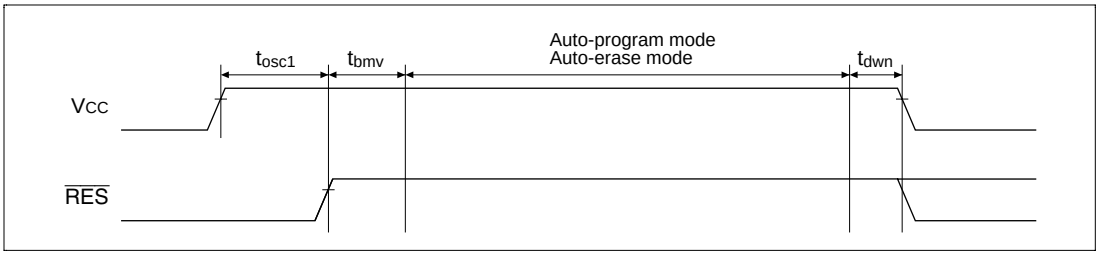


Figure 7.21 Oscillation Stabilization Time, Boot Program Transfer Time, and Power-Down Sequence

7.10.9 Notes on Memory Programming

1. When programming addresses which have previously been programmed, carry out auto-erasing before auto-programming.
2. When performing programming using programmer mode on a chip that has been programmed/erased in an on-board programming mode, auto-erasing is recommended before carrying out auto-programming.

Notes: 1. The flash memory is initially in the erased state when the device is shipped by Hitachi. For other chips for which the erasure history is unknown, it is recommended that auto-erasing be executed to check and supplement the initialization (erase) level.

2. Auto-programming should be performed once only on the same address block. Additional programming cannot be performed on previously programmed address blocks.

Section 8 RAM

8.1 Overview

H8/3664 Series has 1 kbyte and 512 bytes 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.

8.1.1 Block Diagram

Figure 8.1 shows a block diagram of the on-chip RAM.

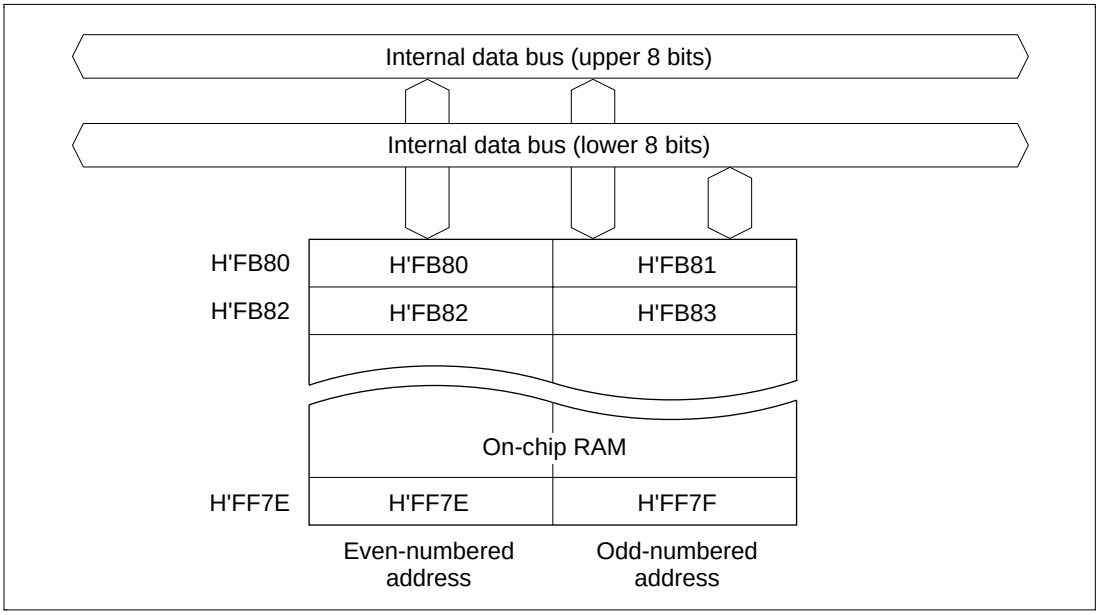


Figure 8.1 RAM Block Diagram

Section 9 I/O Ports

9.1 Overview

The H8/3664 Series is provided with two 8-bit I/O ports, one 7-bit I/O port, two 3-bit I/O ports, and one 8-bit input-only 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 section 2.9.2, Notes on Bit Manipulation, for information on executing bit-manipulation instructions to write data in PCR or PDR.

Pin states in each operation mode are given in Appendix D, Port States in the Different Processing State.

9.2 Port 1

9.2.1 Overview

Port 1 is a 7-bit I/O port. Figure 9.1 shows its pin configuration.

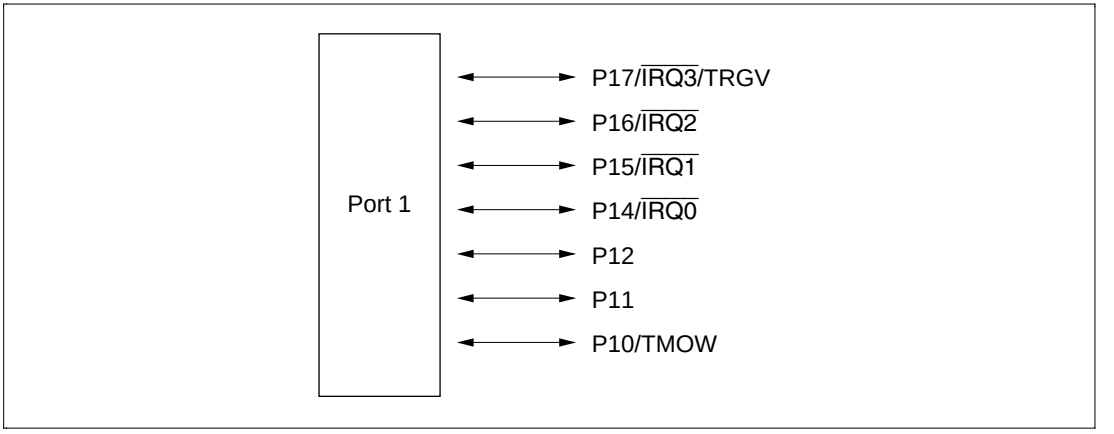


Figure 9.1 Port 1 Pin Configuration

9.2.2 Register Configuration and Description

Table 9.1 shows the port 1 register configuration.

Table 9.1 Port 1 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 1	PDR1	R/W	H'08	H'FFD4
Port control register 1	PCR1	W	H'00	H'FFE4
Port pull-up control register 1	PUCR1	R/W	H'08	H'FFD0
Port mode register 1	PMR1	R/W	H'0C	H'FFE0

9.2.3 Port Data Register 1 (PDR1)

Bit	7	6	5	4	3	2	1	0
	P17	P16	P15	P14	—	P12	P11	P10
Initial value	0	0	0	0	1*	0	0	0
Read/Write	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W

Note: * Bit 3 is reserved; it is always read as 1 and cannot be modified.

PDR1 is an 8-bit register that stores data for port 1 pins P17 to P14 and P12 to P10. 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'08.

9.2.4 Port Control Register 1 (PCR1)

Bit	7	6	5	4	3	2	1	0
	PCR17	PCR16	PCR15	PCR14	—	PCR12	PCR11	PCR10
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	—	W	W	W

PCR1 is an 8-bit register for controlling whether each of the port 1 pins P17 to P14 and P12 to P10 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.

9.2.5 Port Pull-Up Control Register 1 (PUCR1)

Bit	7	6	5	4	3	2	1	0
	PUCR17	PUCR16	PUCR15	PUCR14	—	PUCR12	PUCR11	PUCR10
Initial value	0	0	0	0	1*	0	0	0
Read/Write	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W

Note: * Bit 3 is reserved; it is always read as 1 and cannot be modified.

PUCR1 controls whether the MOS pull-up of each of the port 1 pins P17 to P14 and P12 to P10 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'08.

9.2.6 Port Mode Register 1 (PMR1)

Bit	7	6	5	4	3	2	1	0
	IRQ3	IRQ2	IRQ1	IRQ0	—	—	TXD	TMOW
Initial value	0	0	0	0	1	1	0	0
Read/Write	R/W	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 ports 1 and 2 pins.

Upon reset, PMR1 is initialized to H'0C.

Bit 7—P17/$\overline{\text{IRQ3}}$/TRGV Pin Function Switch (IRQ3): This bit selects whether pin P17/$\overline{\text{IRQ3}}$/TRGV is used as P17 or as $\overline{\text{IRQ3}}$/TRGV.

Bit 7: IRQ3	Description	
0	Functions as P17 I/O pin	(Initial value)
1	Functions as $\overline{\text{IRQ3}}$/TRGV input pin	

Note: Rising or falling edge sensing can be designated for $\overline{\text{IRQ3}}$. Rising, falling, or both edge sensing can be designated for TRGV. For details on TRGV settings, see section 11.2.5, Timer Control Register V1 (TCRV1).

Bit 6—P16/$\overline{\text{IRQ2}}$ Pin Function Switch (IRQ2): This bit selects whether pin P16/$\overline{\text{IRQ2}}$ is used as P16 or as $\overline{\text{IRQ2}}$.

Bit 6: IRQ2	Description	
0	Functions as P16 I/O pin	(Initial value)
1	Functions as $\overline{\text{IRQ2}}$ input pin	

Note: Rising or falling edge sensing can be designated for $\overline{\text{IRQ2}}$.

Bit 5—P15/$\overline{\text{IRQ1}}$ Pin Function Switch (IRQ1): This bit selects whether pin P15/$\overline{\text{IRQ1}}$ is used as P15 or as $\overline{\text{IRQ1}}$.

Bit 5: IRQ1	Description	
0	Functions as P15 I/O pin	(Initial value)
1	Functions as $\overline{\text{IRQ1}}$ input pin	

Note: Rising or falling edge sensing can be designated for $\overline{\text{IRQ1}}$.

Bit 4—P14/$\overline{\text{IRQ0}}$ Pin Function Switch (IRQ0): This bit selects whether pin P14/$\overline{\text{IRQ0}}$ is used as P14 or as $\overline{\text{IRQ0}}$.

Bit 4: IRQ0	Description	
0	Functions as P14 I/O pin	(Initial value)
1	Functions as $\overline{\text{IRQ0}}$ output pin	

Bit 3—Reserved Bit: Bit 3 is reserved: it is always read as 1 and cannot be modified.

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

Bit 1—P22/TXD Pin Function Switch (TXD): This bit selects whether pin P22/TXD is used as P22 or as TXD.

Bit 1: TXD	Description	
0	Functions as P22 I/O pin	(Initial value)
1	Functions as TXD output pin	

Bit 0—P10/TMOW Pin Function Switch (TMOW): This bit selects whether pin P10/TMOW is used as P10 or as TMOW.

Bit 0: TMOW	Description	
0	Functions as P10 I/O pin	(Initial value)
1	Functions as TMOW output pin	

9.2.7 Pin Functions

Table 9.2 shows the port 1 pin functions.

Table 9.2 Port 1 Pin Functions

Pin	Pin Functions and Selection Method
P17/$\overline{\text{IRQ3}}$/TRGV	The pin function depends on bit IRQ3 in PMR1 and bit PCR17 in PCR1.

IRQ3	0		1
PCR17	0	1	*
Pin function	P17 input pin	P17 output pin	$\overline{\text{IRQ3}}$/TRGV input pin

P16/$\overline{\text{IRQ2}}$
P15/$\overline{\text{IRQ1}}$
P14/$\overline{\text{IRQ0}}$

The pin function depends on bits IRQ2 and IRQ1 in PMR1 and bit PCR1n in IRQ0 and PCR1.

(m = n – 4, n = 6, 5, 4)

IRQm	0		1
PCR1n	0	1	*
Pin function	P1n input pin	P1n output pin	$\overline{\text{IRQm}}$ input pin

P12
P11

The pin function depends on bit PCR1n in PCR1.

PCR1n	0	1
Pin function	P1n input pin	P1n output pin

P10/TMOW

The pin function depends on bit TMOW in PMR1 and bit PCR10 in PCR1.

TMOW	0		1
PCR10	0	1	*
Pin function	P10 input pin	P10 output pin	TMOW output pin

Note: * Don't care

9.2.8 MOS Input Pull-Up

Port 1 has an on-chip 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.

PCR1n	0		1
PUCR1n	0	1	*
MOS input pull-up	Off	On	Off

(n = 7 to 4, 2 to 0)

Note: * Don't care

9.3 Port 2

9.3.1 Overview

Port 2 is a 3-bit I/O port, configured as shown in figure 9.2.

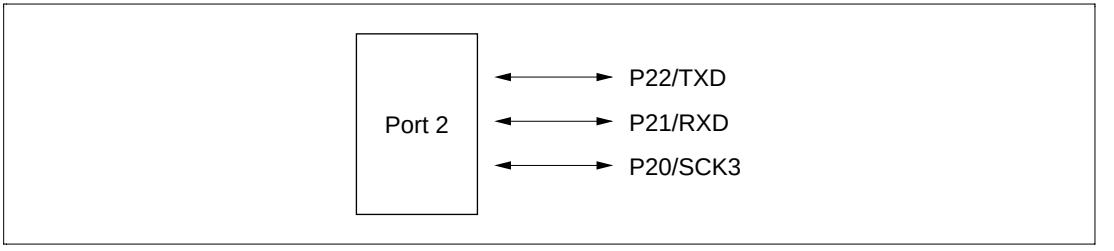


Figure 9.2 Port 2 Pin Configuration

9.3.2 Register Configuration and Description

Table 9.3 shows the port 2 register configuration.

Table 9.3 Port 2 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 2	PDR2	R/W	H'F8	H'FFD5
Port control register 2	PCR2	W	H'00	H'FFE5

9.3.3 Port Data Register 2 (PDR2)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	P22	P21	P20
Initial value	1*	1*	1*	1*	1*	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

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

PDR2 is an 8-bit register that stores data for port 2 pins P22 to P20. 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'F8.

9.3.4 Port Control Register 2 (PCR2)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	PCR22	PCR21	PCR20
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	—	—	W	W	W

PCR2 is an 8-bit register for controlling whether each of the port 1 pins P22 to P20 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.

9.3.5 Pin Functions

Table 9.4 shows the port 2 pin functions.

Table 9.4 Port 2 Pin Functions

P22/TXD — The pin function depends on bit TXD in PMR1 and bit PCR22 in PCR2.

TXD	0		1
PCR22	0	1	*
Pin function	P22 input pin	P22 output pin	TXD output pin

P21/RXD — The pin function depends on bit RE in SCR3 and bit PCR21 in PCR2.

RE	0		1
PCR21	0	1	*
Pin function	P21 input pin	P21 output pin	RXD input pin

P20/SCK3 — The pin function depends on bits CKE1 and CKE0 in SCR3, bit COM in SMR, and bit PCR20 in PCR2.

CKE1	0			1
CKE0	0		1	*
COM	0		1	*	*
PCR20	0	1	*	*
Pin function	P20 input pin	P20 output pin	SCK3 output pin	SCK3 input pin

Note: * Don't care

9.4 Port 5

9.4.1 Overview

Port 5 is an 8-bit I/O port, configured as shown in figure 9.3.

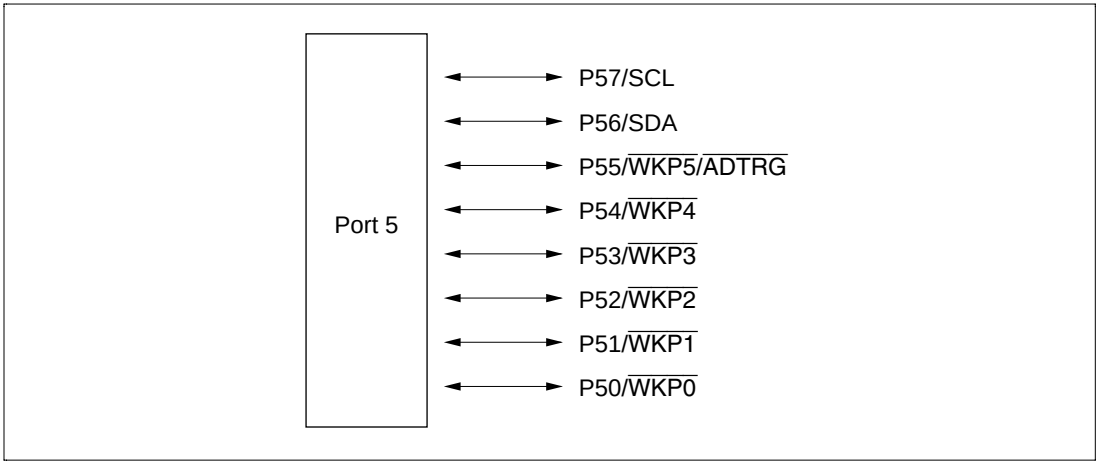


Figure 9.3 Port 5 Pin Configuration

9.4.2 Register Configuration and Description

Table 9.5 shows the port 5 register configuration.

Table 9.5 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'FFD1
Port mode register 5	PMR5	R/W	H'00	H'FFE1

9.4.3 Port Data Register 5 (PDR5)

Bit	7	6	5	4	3	2	1	0
	P57	P56	P55	P54	P53	P52	P51	P50
Initial value	0	0	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 P57 to P50. 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.

9.4.4 Port Control Register 5 (PCR5)

Bit	7	6	5	4	3	2	1	0
	PCR57	PCR56	PCR55	PCR54	PCR53	PCR52	PCR51	PCR50
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 P57 to P50 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.

9.4.5 Port Pull-Up Control Register 5 (PUCR5)

Bit	7	6	5	4	3	2	1	0
	—	—	PUCR55	PUCR54	PUCR53	PUCR52	PUCR51	PUCR50
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 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.

9.4.6 Port Mode Register 5 (PMR5)

Bit	7	6	5	4	3	2	1	0
	—	—	WKP5	WKP4	WKP3	WKP2	WKP1	WKP0
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

PMR5 is an 8-bit read/write register, controlling the selection of pin functions for port 5 pins.

Upon reset, PMR5 is initialized to H'00.

Bit n—P5n/$\overline{\text{WKPn}}$ Pin Function Switch (WKPn): This bit selects whether pin P5n/$\overline{\text{WKPn}}$ is used as P5n or as $\overline{\text{WKPn}}$.

Bit n: WKPn	Description	
0	Functions as P5n I/O pin	(Initial value)
1	Functions as $\overline{\text{WKPn}}$ input pin	

(n = 5 to 0)

9.4.7 Pin Functions

Table 9.6 shows the port 5 pin functions.

The output type of SCL and SDA is open-drain, providing direct bus driving capability.

Table 9.6 Port 5 Pin Functions

Pin	Pin Functions and Selection Method			
P57/SCL	The pin function depends on bit PCR57 in PCR5 and bit ICE in I2C.			
	ICE	0		1
	PCR57	0	1	*
	Pin function	P57 input pin	P57 output pin	SCL I/O pin
	The SCL output format is NMOS open drain and direct bus driving is enabled.			
P56/SDA	The pin function depends on bit PCR56 in PCR5 and bit ICE in I2C.			
	ICE	0		1
	PCR56	0	1	*
	Pin function	P56 input pin	P56 output pin	SDA I/O pin
	The SDA output format is NMOS open drain and direct bus driving is enabled.			
P55/$\overline{\text{WKP5}}$/ $\overline{\text{ADTRG}}$	The pin function depends on bit PCR55 in PCR5 and bit WKP5 in PMR5.			
	WKP5	0		1
	PCR55	0	1	*
	Pin function	P55 input pin	P55 output pin	$\overline{\text{WKP5}}$ input pin $\overline{\text{ADTRG}}$ input pin
P54/$\overline{\text{WKP4}}$ to P50/$\overline{\text{WKP0}}$	The pin function depends on bit PCR5n in PCR5 and bit WKPn in PMR5.			
	WKPn	0		1
	PCR5n	0	1	*
	Pin function	P5n input pin	P5n output pin	$\overline{\text{WKPn}}$ input pin
	(n = 4 to 0)			

Note: * Don't care.

9.4.8 MOS Input Pull-Up

Port 5 has an on-chip 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.

PCR5n	0		1
PUCR5n	0	1	*
MOS input pull-up	Off	On	Off

(n = 5 to 0)

Note: * Don't care

9.5 Port 7

9.5.1 Overview

Port 7 is a 3-bit I/O port, configured as shown in figure 9.4.

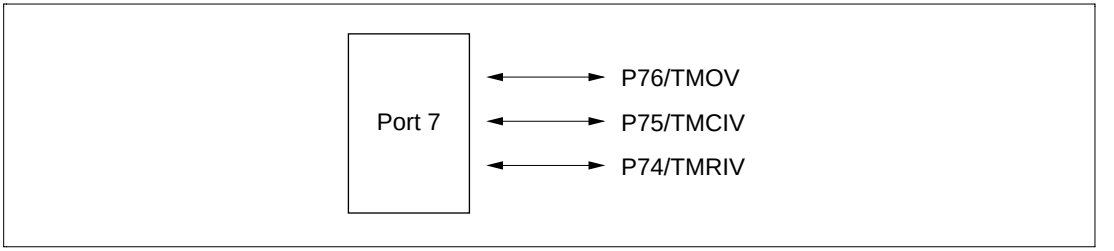


Figure 9.4 Port 7 Pin Configuration

9.5.2 Register Configuration and Description

Table 9.7 shows the port 7 register configuration.

Table 9.7 Port 7 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 7	PDR7	R/W	H'8F	H'FFDA
Port control register 7	PCR7	W	H'00	H'FFEA

9.5.3 Port Data Register 7 (PDR7)

Bit	7	6	5	4	3	2	1	0
	—	P76	P75	P74	—	—	—	—
Initial value	1*	0	0	0	1*	1*	1*	1*
Read/Write	—	R/W	R/W	R/W	—	—	—	—

Note: * Bits 7 and 3 to 0 are reserved; they are always read as 1 and cannot be modified.

PDR7 is an 8-bit register that stores data for port 7 pins P76 to P74. 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'8F.

9.5.4 Port Control Register 7 (PCR7)

Bit	7	6	5	4	3	2	1	0
	—	PCR76	PCR75	PCR74	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	W	W	W	—	—	—	—

PCR7 is an 8-bit register for controlling whether each of the port 7 pins P76 to P74 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.

9.5.5 Pin Functions

Table 9.8 shows the port 7 pin functions.

Table 9.8 Port 7 Pin Functions

Pin	Pin Functions and Selection Method
P76/TMOV	The pin function depends on bit PCR76 in PCR7 and bits OS3 to OS0 in TCSRV.

OS3 to OS0	0000		Not 0000
PCR76	0	1	*
Pin function	P76 input pin	P76 output pin	TMOV output pin

P75/TMCIV	The pin function depends on bit PCR75 in PCR7.

PCR75	0	1
Pin function	P75 input pin	P75 output pin
	TMCIV input pin	

P74/TMRIV	The pin function depends on bit PCR74 in PCR7.

PCR74	0	1
Pin function	P74 input pin	P74 output pin
	TMRIV input pin	

Note: * Don't care

9.6 Port 8

9.6.1 Overview

Port 8 is an 8-bit I/O port configured as shown in figure 9.5.

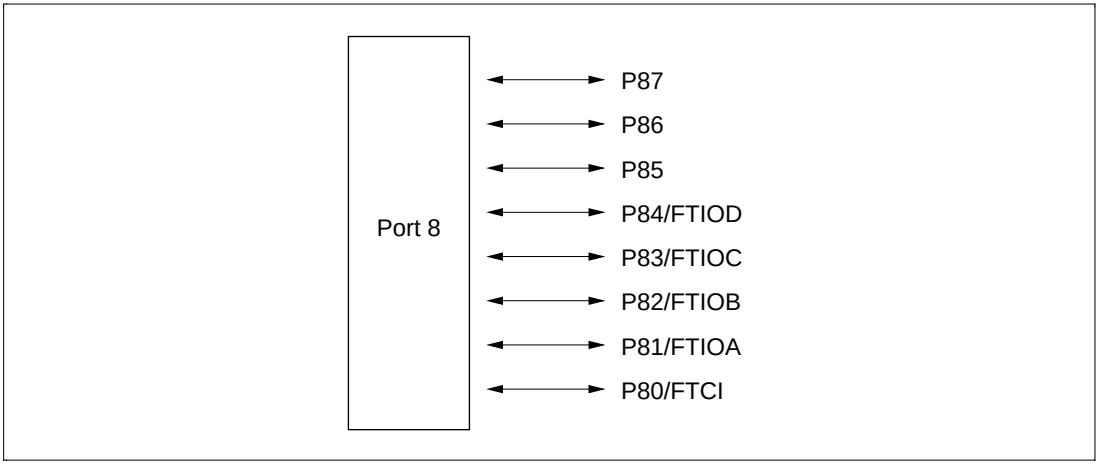


Figure 9.5 Port 8 Pin Configuration

9.6.2 Register Configuration and Description

Table 9.9 shows the port 8 register configuration.

Table 9.9 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

9.6.3 Port Data Register 8 (PDR8)

Bit	7	6	5	4	3	2	1	0
	P87	P86	P85	P84	P83	P82	P81	P80
Initial value	0	0	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 P87 to P80. 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.

9.6.4 Port Control Register 8 (PCR8)

Bit	7	6	5	4	3	2	1	0
	PCR87	PCR86	PCR85	PCR84	PCR83	PCR82	PCR81	PCR80
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 P87 to P80 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.

9.6.5 Pin Functions

Table 9.10 shows the port 8 pin functions.

Table 9.10 Port 8 Pin Functions

<table>
<tr><th>Pin</th><th>Pin Functions and Selection Method</th></tr>
<tr><td>P87</td><td>The pin function depends on bit PCR87 in PCR8.
<table>
<tr><td>PCR87</td><td>0</td><td>1</td></tr>
<tr><td>Pin function</td><td>P87 input pin</td><td>P87 output pin</td></tr>
</table></td></tr>
<tr><td>P86</td><td>The pin function depends on bit PCR86 in PCR8.
<table>
<tr><td>PCR86</td><td>0</td><td>1</td></tr>
<tr><td>Pin function</td><td>P86 input pin</td><td>P86 output pin</td></tr>
</table></td></tr>
<tr><td>P85</td><td>The pin function depends on bit PCR85 in PCR8.
<table>
<tr><td>PCR85</td><td>0</td><td>1</td></tr>
<tr><td>Pin function</td><td>P85 input pin</td><td>P85 output pin</td></tr>
</table></td></tr>
<tr><td>P84/FTIOD</td><td>The pin function depends on bit PCR84 in PCR8 and bit TIOR1 in timer W.
<table>
<tr><td>Timer W Setting</td><td>(2)</td><td colspan="2">(1)</td><td>(2)</td></tr>
<tr><td>IOD2</td><td colspan="3">0</td><td>1</td></tr>
<tr><td>IOD1</td><td>0</td><td>0</td><td>1</td><td>—</td></tr>
<tr><td>IOD0</td><td>0</td><td>1</td><td>—</td><td>—</td></tr>
</table>
<table>
<tr><td>Timer W Setting</td><td>(1) above</td><td colspan="2">(2) above</td></tr>
<tr><td>PCR84</td><td>—</td><td>0</td><td>1</td></tr>
<tr><td>Pin function</td><td rowspan="2">FTIOD output pin</td><td>P84 input pin</td><td>P84 output pin</td></tr>
<tr><td colspan="2">FTIOD input pin</td></tr>
</table></td></tr>
</table>

Pin	Pin Functions and Selection Method
P83/FTIOC	The pin function depends on bit PCR83 in PCR8 and bit TIOR1 in timer W.

Timer W Setting	(2)	(1)		(2)
IOC2	0			1
IOC1	0	0	1	—
IOC0	0	1	—	—

Timer W Setting	(1) above	(2) above	
PCR83	—	0	1
Pin function	FTIOC output pin	P83 input pin	P83 output pin
		FTIOC input pin	

P82/FTIOB	The pin function depends on bit PCR82 in PCR8 and bit TIOR0 in timer W.

Timer W Setting	(2)	(1)		(2)
IOB2	0			1
IOB1	0	0	1	—
IOB0	0	1	—	—

Timer W Setting	(1) above	(2) above	
PCR82	—	0	1
Pin function	FTIOB output pin	P82 input pin	P82 output pin
		FTIOB input pin	

Pin	Pin Functions and Selection Method
P82/FTIOA	The pin function depends on bit PCR81 in PCR8 and bit TIOR0 in timer W.

Timer W Setting	(2)	(1)		(2)
IOA2	0			1
IOA1	0	0	1	—
IOA0	0	1	—	—

Timer W Setting	(1) above	(2) above	
PCR81	—	0	1
Pin function	FTIOA output pin	P81 input pin	P81 output pin
		FTIOA input pin	

P80/FTCI	The pin function depends on bit PCR80 in PCR8.

PCR80	0	1
Pin function	P80 input pin	P80 output pin
	FTCI input pin	

9.7 Port B

9.7.1 Overview

Port B is an 8-bit input-only port, configured as shown in figure 9.6.

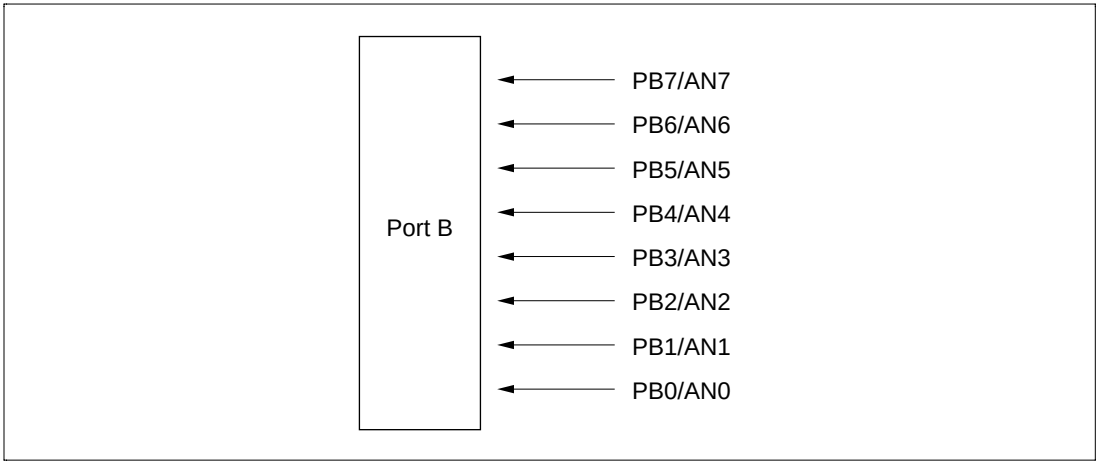


Figure 9.6 Port B Pin Configuration

9.7.2 Register Configuration and Description

Table 9.11 shows the port B register configuration.

Table 9.11 Port B Register

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

9.7.3 Port Data Register B (PDRB)

Bit	7	6	5	4	3	2	1	0
	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
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 by the A/D converter's ADCSR register, that pin reads 0.

9.7.4 Pin Functions

Table 9.12 shows the port B pin functions.

Table 9.12 Port B Pin Functions

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

Section 10 Timer A

10.1 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.

10.1.1 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.

10.1.2 Block Diagram

Figure 10.1 shows a block diagram of timer A.

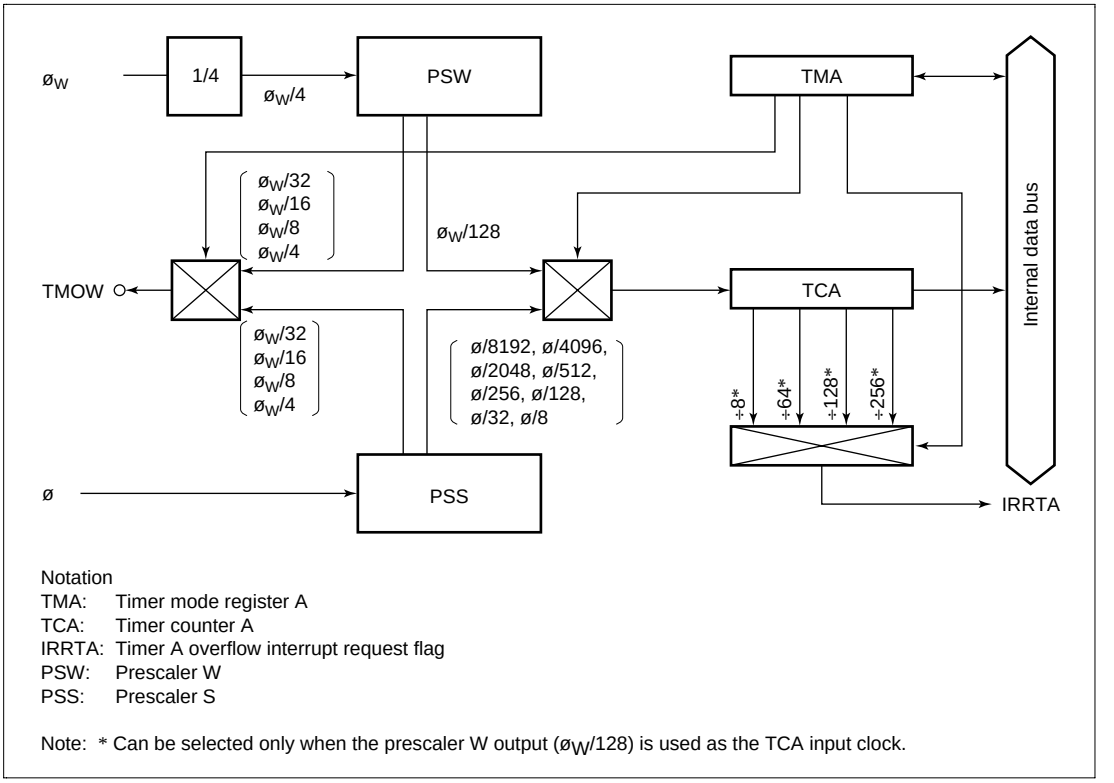


Figure 10.1 Block Diagram of Timer A

10.1.3 Pin Configuration

Table 10.1 shows the timer A pin configuration.

Table 10.1 Pin Configuration

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

10.1.4 Register Configuration

Table 10.2 shows the register configuration of timer A.

Table 10.2 Timer A Registers

Name	Abbrev.	R/W	Initial Value	Address
Timer mode register A	TMA	R/W	H'10	H'FFA6
Timer counter A	TCA	R	H'00	H'FFA7

10.2 Register Descriptions

10.2.1 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			

10.2.2 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.

10.3 Timer Operation

10.3.1 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.

10.3.2 Clock Time Base Operation

When bit TMA3 in TMA is set to 1, timer A functions as a 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.

10.3.3 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, and subactive mode.

10.4 Timer A Operation States

Table 10.3 summarizes the timer A operation states.

Table 10.3 Timer A Operation States

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

Note: When the 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.

Section 11 Timer V

11.1 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.

11.1.1 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 count stop 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.

11.1.2 Block Diagram

Figure 11.1 shows a block diagram of timer V.

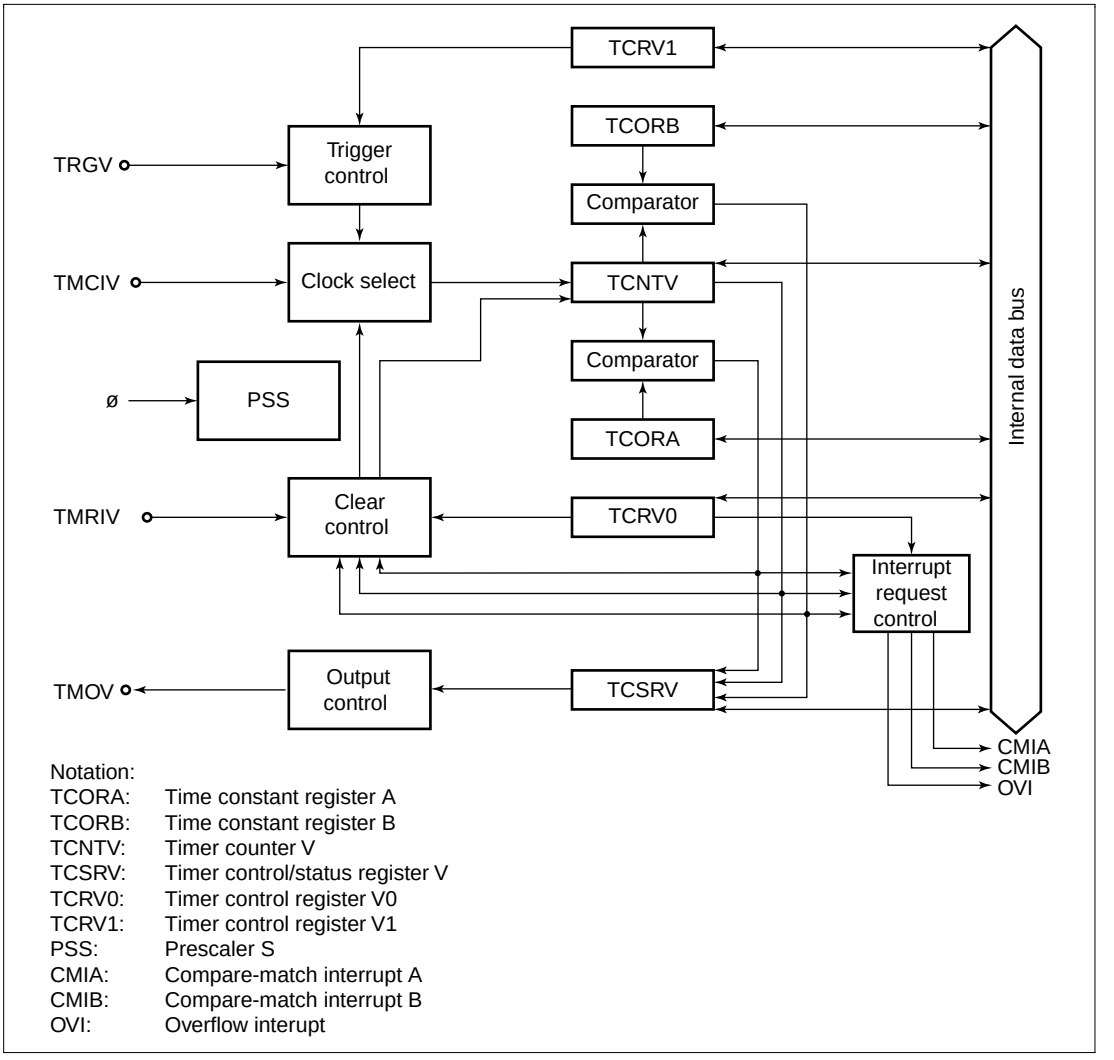


Figure 11.1 Block Diagram of Timer V

11.1.3 Pin Configuration

Table 11.1 shows the timer V pin configuration.

Table 11.1 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

11.1.4 Register Configuration

Table 11.2 shows the register configuration of timer V.

Table 11.2 Timer V Registers

Name	Abbrev.	R/W	Initial Value	Address
Timer control register V0	TCRV0	R/W	H'00	H'FFA0
Timer control/status register V	TCSRV	R/(W)*	H'10	H'FFA1
Time constant register A	TCORA	R/W	H'FF	H'FFA2
Time constant register B	TCORB	R/W	H'FF	H'FFA3
Timer counter V	TCNTV	R/W	H'00	H'FFA4
Timer control register V1	TCRV1	R/W	H'E2	H'FFA5

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

11.2 Register Descriptions

11.2.1 Timer Counter V (TCNTV)

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

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, subsleep mode, and subactive mode.

11.2.2 Time Constant Registers A and B (TCORA, TCORB)

Bit	7	6	5	4	3	2	1	0
	TCORn7	TCORn6	TCORn5	TCORn4	TCORn3	TCORn2	TCORn1	TCORn0
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, subsleep mode, and subactive mode.

TCORB is similar to TCORA.

11.2.3 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, 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 (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)
	1	Cleared by compare match A	
1	0	Cleared by compare match B	
	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)
		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	

11.2.4 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, 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)
	1	0 output at compare match B	
1	0	1 output at compare match B	
	1	Output toggles at compare match B	

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

11.2.5 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 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)
	1	Rising edge is selected	
1	0	Falling edge is selected	
	1	Rising and falling edges are both selected	

Bit 2—TRGV Input Enable (TRGE): Bit 2 enables TCNTV counting to be triggered by input at the TRGV pin, and enables 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 11.2.3 Timer Control Register V0.

11.3 Timer Operation

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.

When the counter clear event selected by bits CCLR1 and CCLR0 in TCRV0 occurs, TCNTV is cleared and the count-up is halted. TCNTV starts counting when the signal edge selected by bits TVEG1 and TVEG0 in TCRV1 is input at the TRGV pin.

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 11.2 shows the timing.

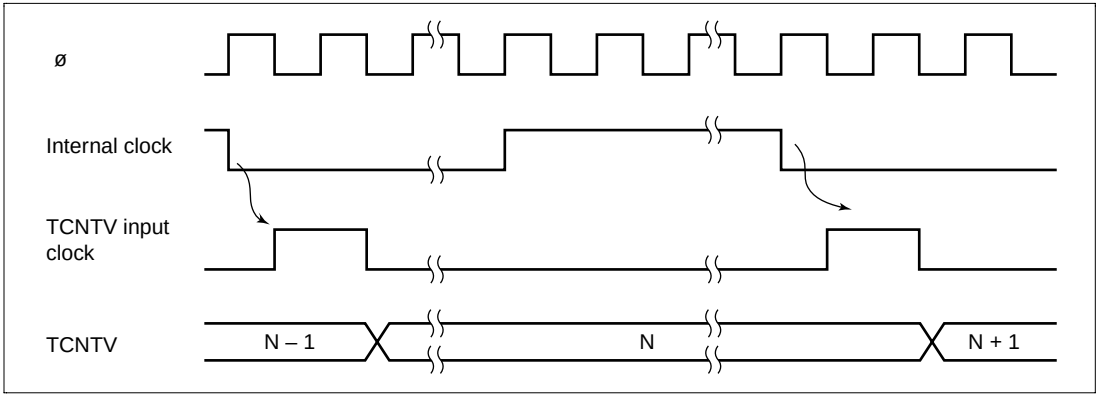


Figure 11.2 Increment Timing with Internal Clock

- 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 11.3 shows the timing when both the rising and falling edges of the external clock are selected.

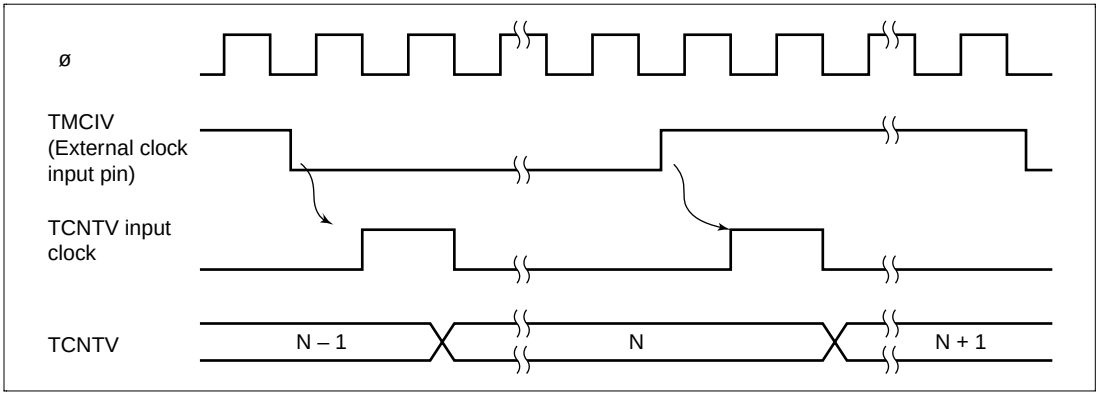


Figure 11.3 Increment Timing with External Clock

Overflow flag Set Timing: The overflow flag (OVF) is set to 1 when TCNTV overflows from H'FF to H'00. Figure 11.4 shows the timing.

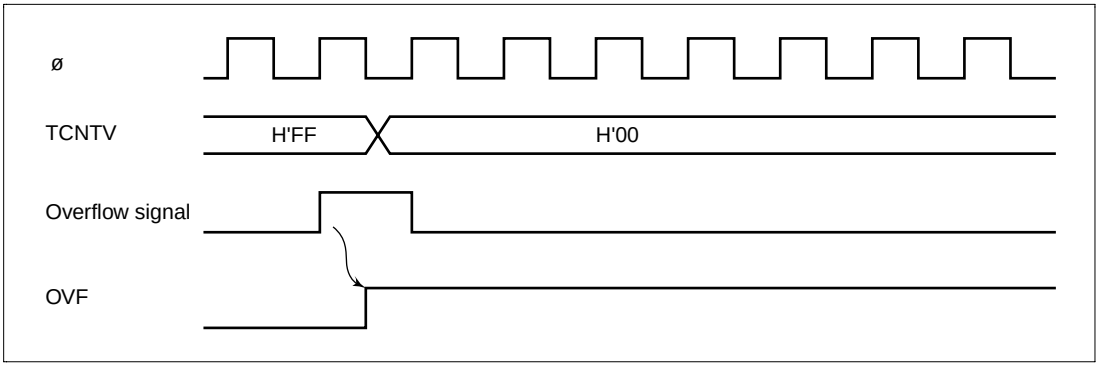


Figure 11.4 OVF Set Timing

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 11.5 shows the timing.

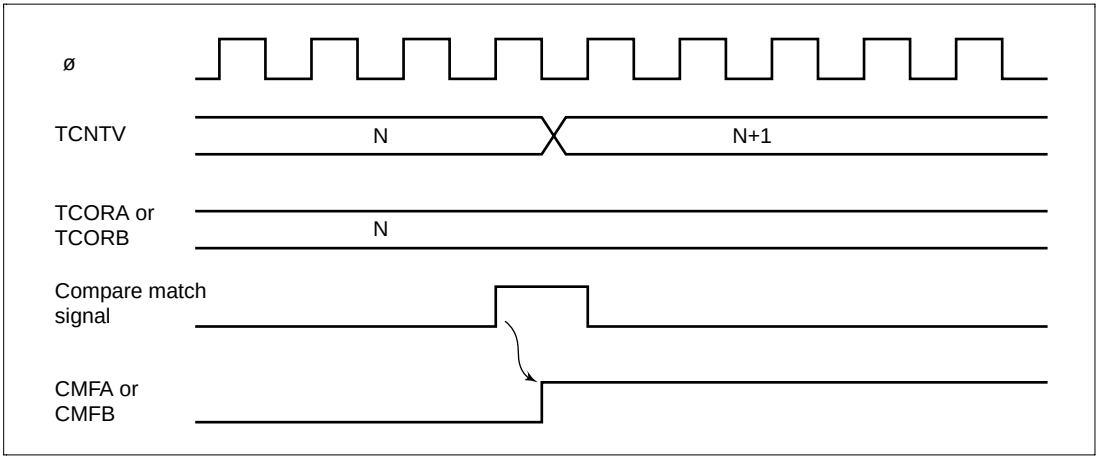


Figure 11.5 CMFA and CMFB Set Timing

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 11.6 shows the timing when the output is toggled by compare match A.

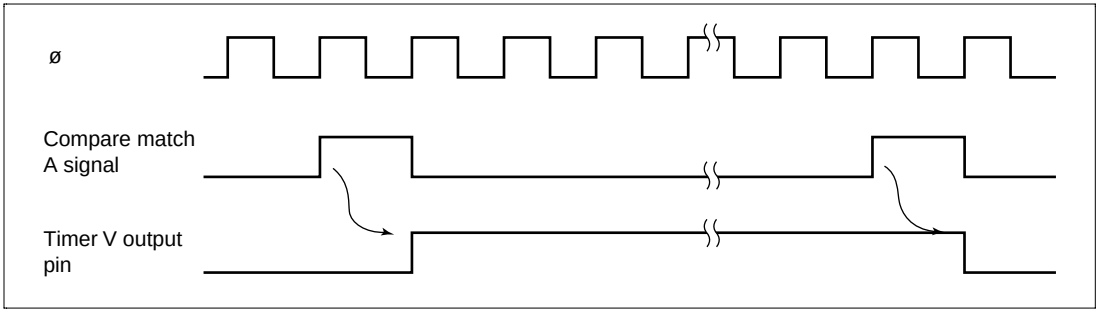


Figure 11.6 TMOV Output Timing

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 117 shows the timing.

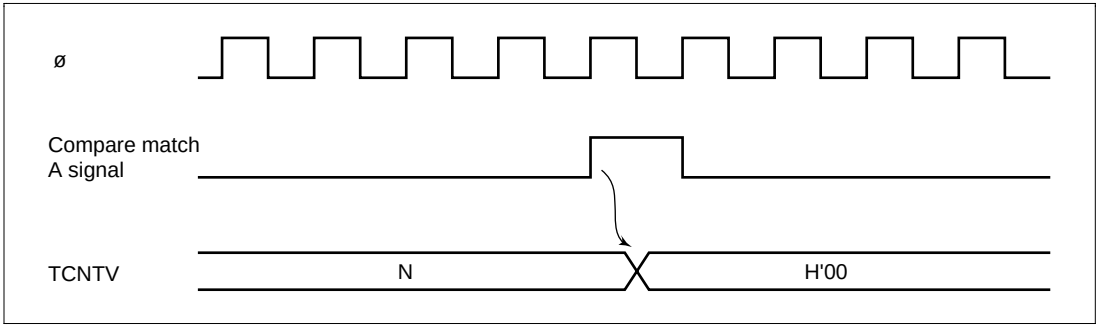


Figure 11.7 Clear Timing by Compare Match

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 11.8 shows the timing.

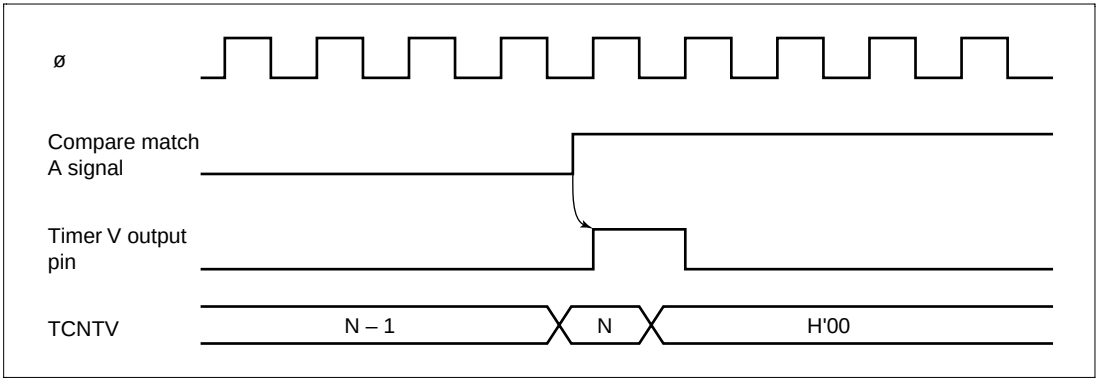


Figure 11.8 Clear Timing by TMRIV Input

11.3.1 Timer V Operation Modes

Table 11.3 summarizes the timer V operation states.

Table 11.3 Timer V Operation States

Operation Mode	Reset	Active	Sleep	Sub-active	Sub-sleep	Standby
TCNTV	Reset	Functions	Functions	Reset	Reset	Reset
TCRV0, TCRV1	Reset	Functions	Functions	Reset	Reset	Reset
TCORA, TCORB	Reset	Functions	Functions	Reset	Reset	Reset
TCSRV	Reset	Functions	Functions	Reset	Reset	Reset

11.3.2 Interrupt Sources

Timer V has three interrupt sources: CMIA, CMIB, and OVI. Table 11.4 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 11.4 Timer V Interrupt Sources

Interrupt	Description	Vector Address
CMIA	Generated from CMFA	H'0022
CMIB	Generated from CMFB	
OVI	Generated from OVF	

11.3.3 Application Examples

Pulse Output with Arbitrary Duty Cycle: Figure 11.9 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.

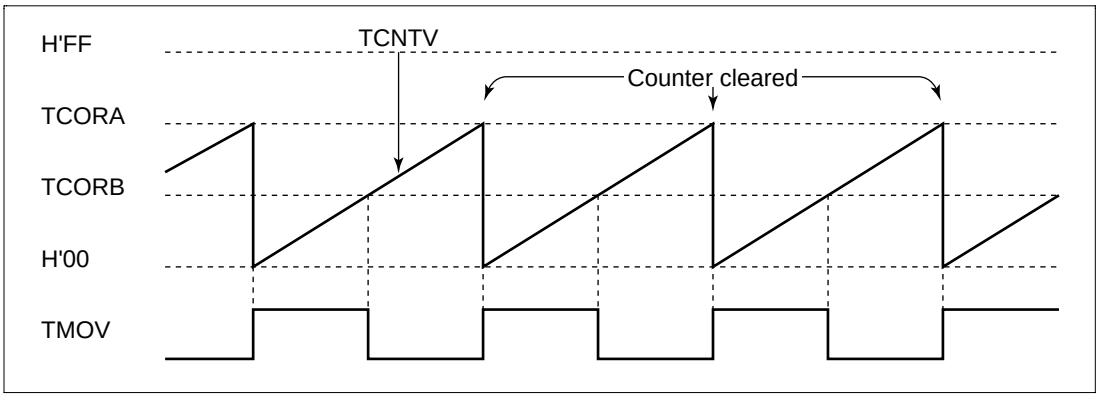


Figure 11.9 Pulse Output Example

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 11.10. 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).

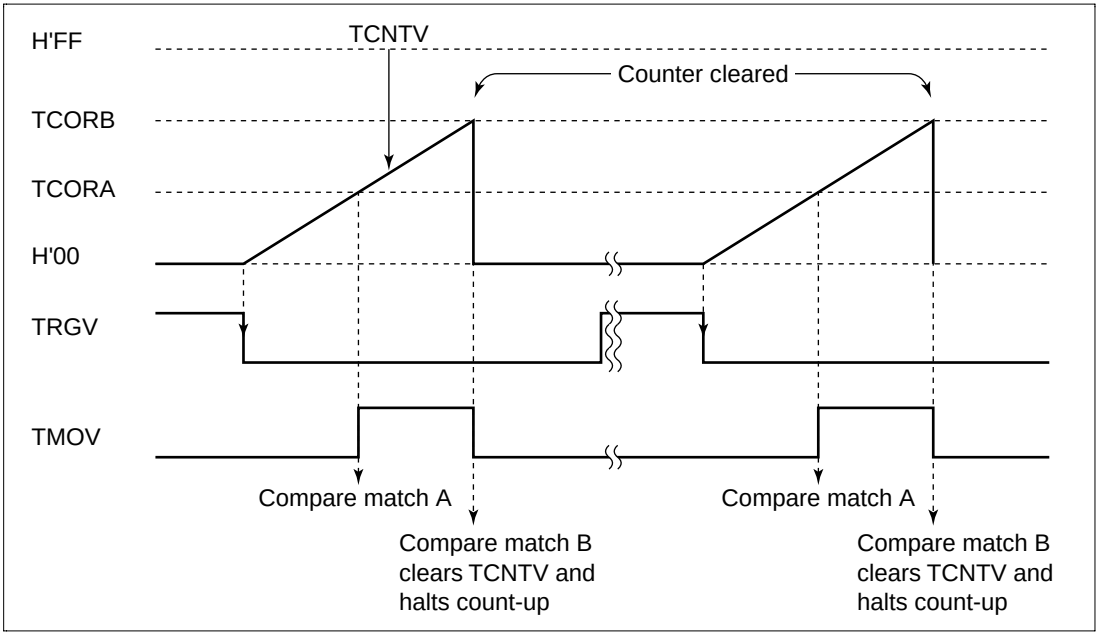


Figure 11.10 Pulse Output Synchronized to TRGV Input

11.3.4 Application Notes

The following types of contention can occur in timer V operation.

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 11.11 shows the timing.

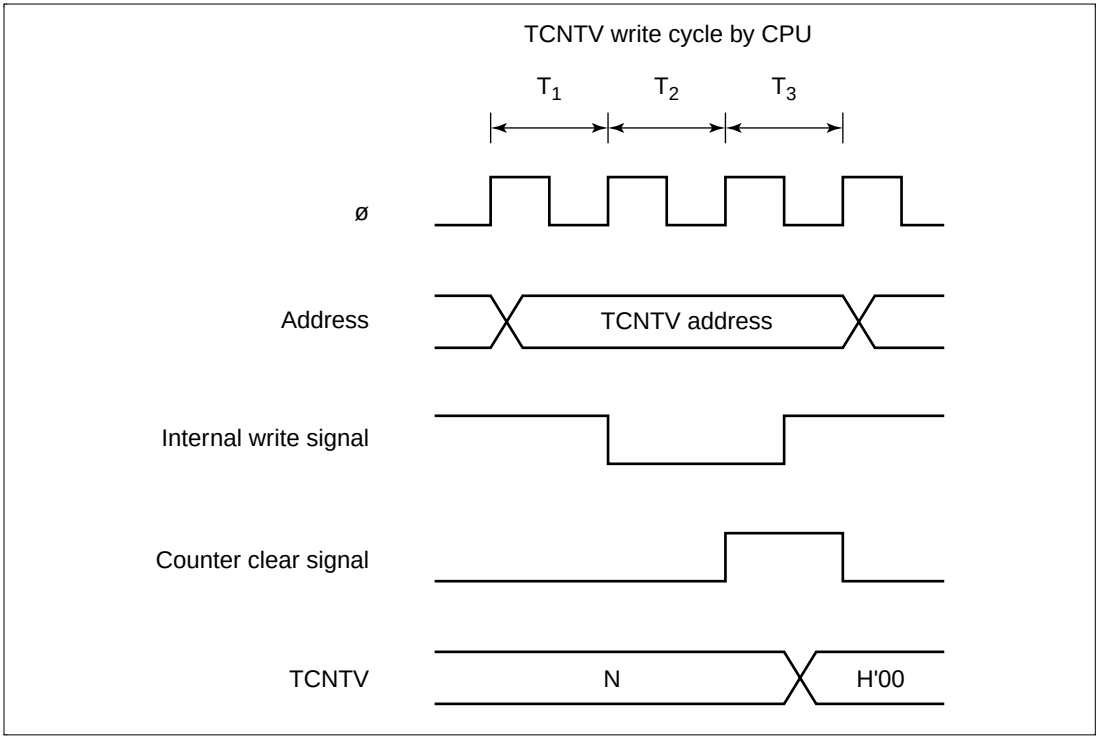


Figure 11.11 Contention between TCNTV Write and Clear

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 11.12 shows the timing.

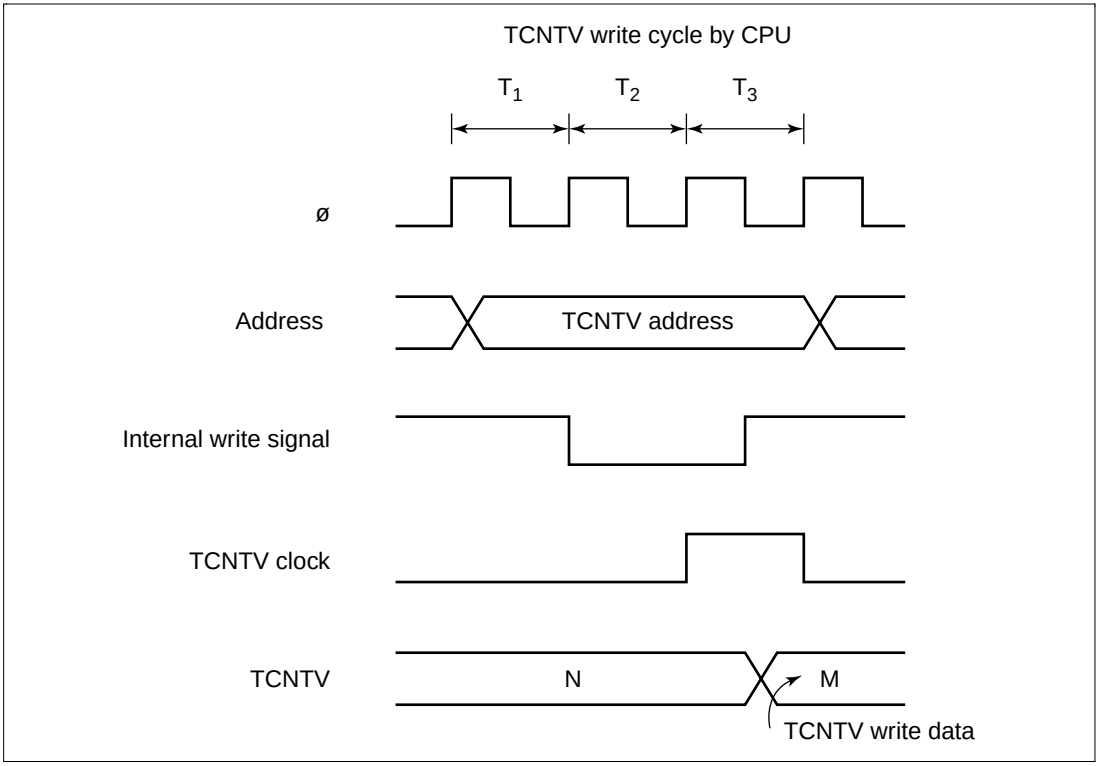


Figure 11.12 Contention between TCNTV Write and Increment

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 11.13 shows the timing.

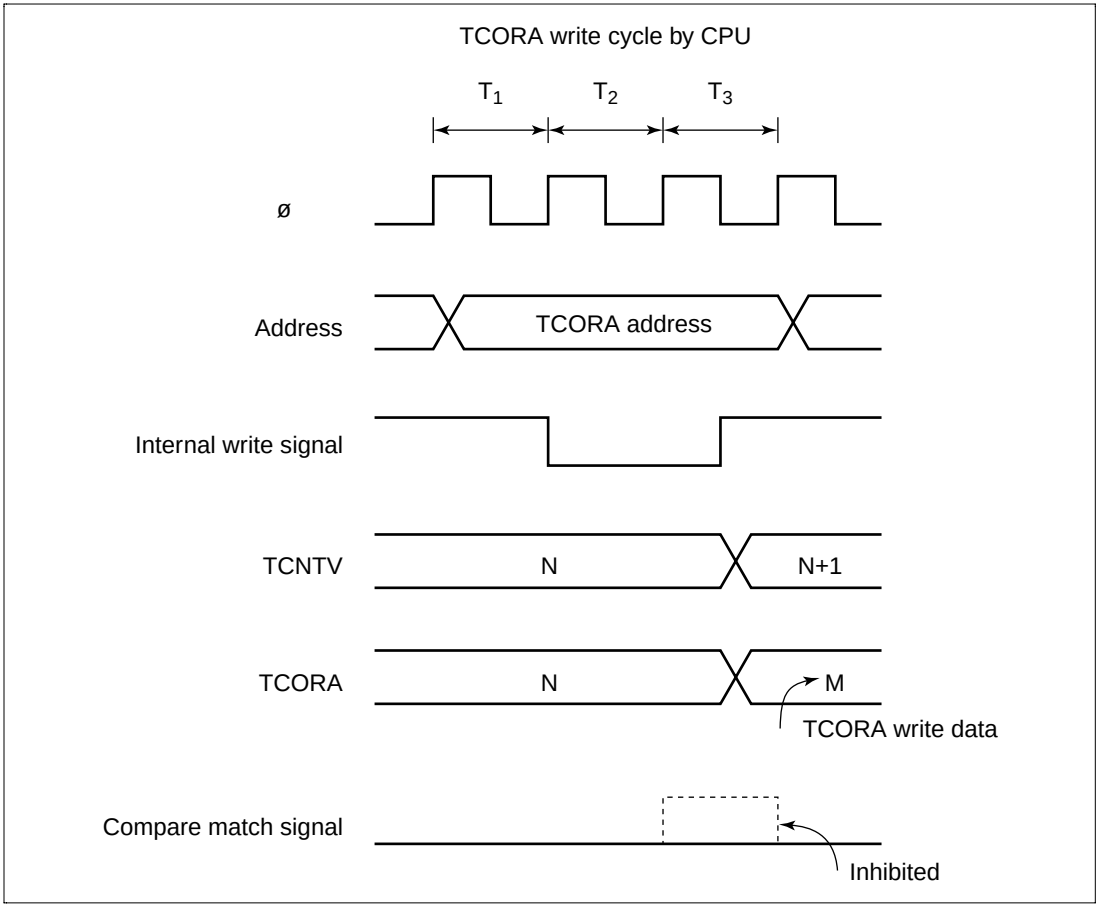


Figure 11.13 Contention between TCORA Write and Compare Match

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 11.5.

Table 11.5 Timer Output Priority Order

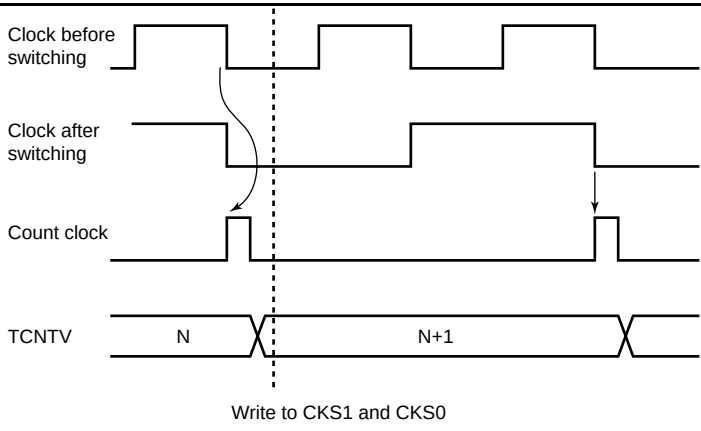
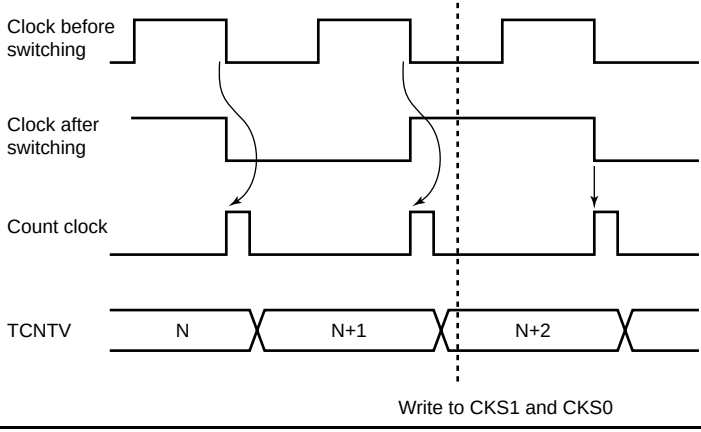
Output Setting	Priority
Toggle output	High
1 output	↑
0 output	
No change	Low

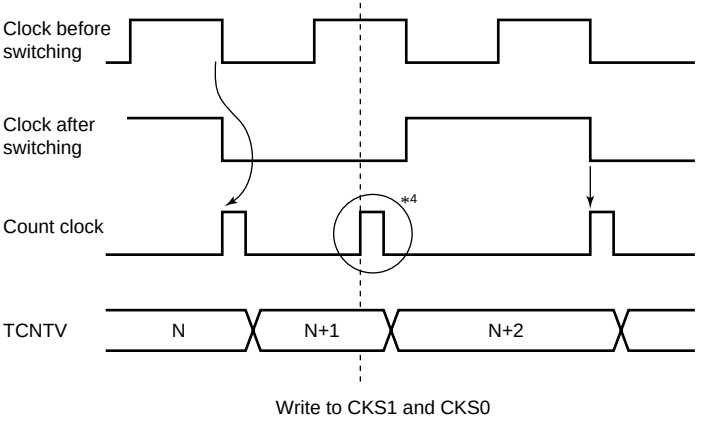
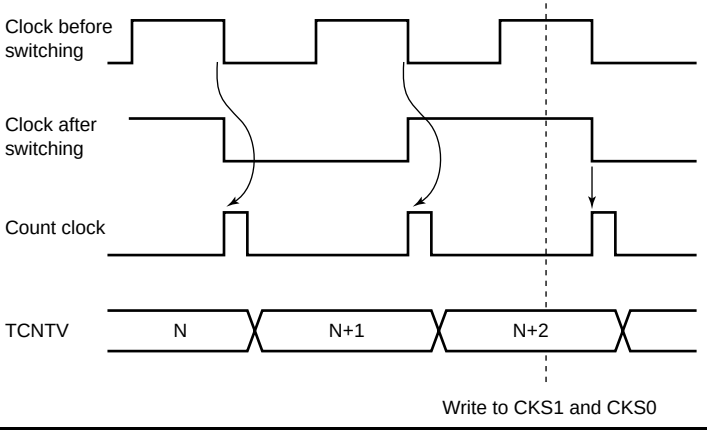
Internal Clock Switching and Counter Operation: Depending on the timing, TCNTV may be incremented by a switch between different internal clock sources. Table 11.5 shows the relation between internal clock switchover timing (by writing to bits CKS1 and CKS0) and TCNTV operation.

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 11.6 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 11.6 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]	

No.	Clock Levels Before and After Modifying Bits CKS1 and CKS0	TCNTV Operation
3	Goes from high level to low level*3	
4	Goes from high to high	

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.
3. Including a transition from the high level to the stopped state.
4. The switchover is seen as a falling edge, and TCNTV is incremented.

Section 12 Timer W

12.1 Overview

The H8/3664 has timer W, a 16-bit timer having output compare and input capture functions. Timer W can count external events and output pulses with a desired duty ratio by compare match between the timer counter and four general registers. Thus, it can be applied to various systems.

12.1.1 Features

Timer W features are listed below.

- Selection of five counter clock sources: four internal clocks (φ, φ/2, φ/4, φ/8) and an external clock (external events can be counted)
- Capability to process up to four pulse outputs or four pulse inputs
- Four general registers with independently assignable output compare or input capture functions
- Four general registers usable as two pairs of registers; one register of each pair operates as a buffer for the output compare or input capture register
- Four selectable operating modes :
  - — Waveform output by compare match
    Selection of 0 output, 1 output, or toggle output
  - — Input capture function
    Rising edge, falling edge, or both edges (selectable)
  - — Counter clearing function
    Counters can be cleared by compare match
  - — PWM mode
    Up to three-phase PWM output can be provided with desired duty ratio.
- High-speed access via an on-chip 16-bit bus
  The 16-bit timer counter and 16-bit general registers can be accessed at a high speed via a 16-bit bus interface.
- Any initial timer output value can be set
- Five interrupt sources
  Four compare match/input capture interrupts and an overflow interrupt.

Table 12.1 summarizes the timer W functions.

Table 12.1 Timer W Functions

Item		Counter	Input/Output Pins			
			FTIOA	FTIOB	FTIOC	FTIOD
Clock sources		Internal clocks: ϕ, ϕ/2, ϕ/4, ϕ/8 External clock: FTCI				
General registers (output compare/input capture registers)		Period specified in GRA	GRA	GRB	GRC (buffer register for GRA in buffer mode)	GRD (buffer register for GRB in buffer mode)
Counter clearing function		GRA compare match	GRA compare match	—	—	—
Initial output value setting function		—	Yes	Yes	Yes	Yes
Buffer function		—	Yes	Yes	—	—
Compare match output	0	—	Yes	Yes	Yes	Yes
	1	—	Yes	Yes	Yes	Yes
	Toggle	—	Yes	Yes	Yes	Yes
Input capture function		—	Yes	Yes	Yes	Yes
PWM mode		—	—	Yes	Yes	Yes
Interrupt sources		Overflow	Compare match/input capture	Compare match/input capture	Compare match/input capture	Compare match/input capture

12.1.2 Block Diagrams

Figure 12.1 is a block diagram of timer W.

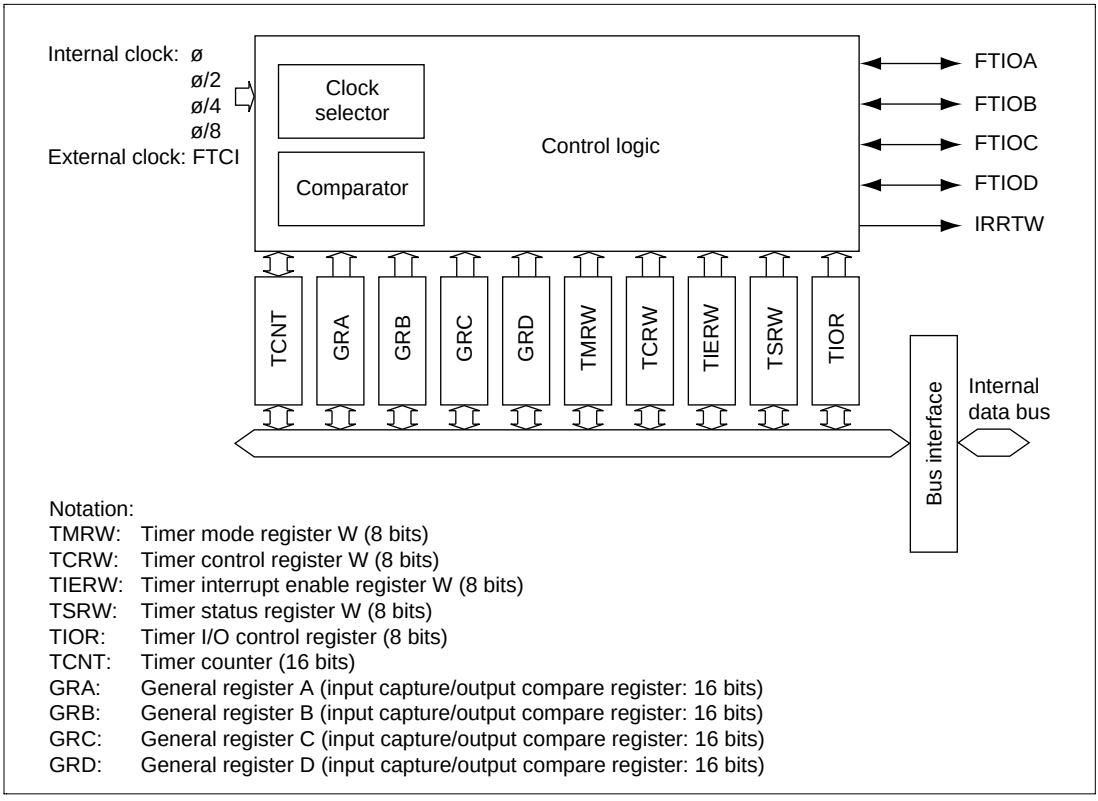


Figure 12.1 Timer W Block Diagram

12.1.3 Input/Output Pins

Table 12.2 summarizes the timer W pins.

Table 12.2 Timer W Pins

Name	Abbreviation	Input/Output	Function
External clock input	FTCI	Input	External clock input pin
Input capture/output compare A	FTIOA	Input/output	Output pin for GRA output compare or input pin for GRA input capture
Input capture/output compare B	FTIOB	Input/output	Output pin for GRB output compare, input pin for GRB input capture, or PWM output pin in PWM mode
Input capture/output compare C	FTIOC	Input/output	Output pin for GRC output compare, input pin for GRC input capture, or PWM output pin in PWM mode
Input capture/output compare D	FTIOD	Input/output	Output pin for GRD output compare, input pin for GRD input capture, or PWM output pin in PWM mode

12.1.4 Register Configuration

Table 12.3 summarizes the timer W registers.

Table 12.3 Timer W Registers

Name	Abbreviation	R/W	Initial Value	Address
Timer mode register W	TMRW	R/W	H'48	H'FF80
Timer control register W	TCRW	R/W	H'00	H'FF81
Timer interrupt enable register W	TIERW	R/W	H'70	H'FF82
Timer status register W	TSRW	R/(W)*1	H'70	H'FF83
Timer I/O control register 0	TIOR0	R/W	H'88	H'FF84
Timer I/O control register 1	TIOR1	R/W	H'88	H'FF85
Timer counter	TCNT	R/W	H'0000	H'FF86*2
General register A	GRA	R/W	H'FFFF	H'FF88*2
General register B	GRB	R/W	H'FFFF	H'FF8A*2
General register C	GRC	R/W	H'FFFF	H'FF8C*2
General register D	GRD	R/W	H'FFFF	H'FF8E*2

Notes: 1. Only 0 can be written to clear the flags.
2. Must be always read or written in 16-bit units; 8-bit access is not allowed.

12.2 Register Description

12.2.1 Timer Mode Register W (TMRW)

Bit	7	6	5	4	3	2	1	0
	CTS	—	BUFEB	BUFEA	—	PWMD	PWMC	PWMB
Initial value	0	1	0	0	1	0	0	0
Read/Write	R/W	—	R/W	R/W	—	R/W	R/W	R/W

TMRW is an 8-bit read/write register that selects PWM mode and buffer operation. TMRW is initialized to H'48 by a reset.

Bit 7—Counter Start (CTS): Starts and stops TCNT.

Bit 7: CTS	Description	
0	Stops TCNT	(Initial value)
1	Starts TCNT	

Note: Writing 0 to the CST bit while a compare match output pin is outputting a signal stops the counter, but the output level at the pin is retained. When the TCRW is modified to change the initial output level while the CST bit is 0, the output level at the pin is updated to the modified initial level.

Bit 6—Reserved: This bit cannot be modified and is always read as 1.

Bit 5—Buffer Operation B (BUFEB): Selects whether GRD operates as an input capture/output compare register or the buffer register for GRB. When GRD is used as the buffer register, no input capture or compare match occurs for GRD.

Bit 5: BUFEB	Description	
0	GRD operates as an input capture/output compare register	(Initial value)
1	GRD operates as the buffer register for GRB	

Bit 4—Buffer Operation A (BUFEA): Selects whether GRC operates as an input capture/output compare register or the buffer register for GRA. When GRC is used as the buffer register, no input capture or compare match occurs for GRC.

Bit 4: BUFEA	Description	
0	GRC operates as an input capture/output compare register	(Initial value)
1	GRC operates as the buffer register for GRA	

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bit 2—PWM Mode D (PWMD): Selects whether the compare match output pin (FTIOD) operates normally or in PWM mode.

Bit 2: PWMD	Description	
0	FTIOD operates normally (output compare output)	(Initial value)
1	FTIOD operates in PWM mode*	

Note: * The period is specified in GRA.

Bit 1—PWM Mode C (PWMC): Selects whether the compare match output pin (FTIOC) operates normally or in PWM mode.

Bit 1: PWMC	Description	
0	FTIOC operates normally (output compare output)	(Initial value)
1	FTIOC operates in PWM mode*	

Note: * The period is specified in GRA.

Bit 0—PWM Mode B (PWMB): Selects whether the compare match output pin (FTIOB) operates normally or in PWM mode.

Bit 0: PWMB	Description	
0	FTIOB operates normally (output compare output)	(Initial value)
1	FTIOB operates in PWM mode*	

Note: * The period is specified in GRA.

12.2.2 Timer Control Register W (TCRW)

Bit	7	6	5	4	3	2	1	0
	CCLR	CKS2	CKS1	CKS0	TOD	TOC	TOB	TOA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCRW is an 8-bit read/write register that specifies the timer output levels, selects the timer counter clock source, and selects how the counter is cleared.

TCRW is initialized to H'00 by a reset.

Bit 7—Counter Clear (CCLR): Selects how TCNT is cleared.

Bit 7: CCLR	Description	
0	TCNT is not cleared by GRA compare match	(Initial value)
1	TCNT is cleared by GRA compare match	

Bits 6 to 4—Clock Select (CKS2 to CKS0): These bits select the TCNT clock source from four internal clock sources and one external event.

Bit 6: CKS2	Bit 5: CKS1	Bit 4: CKS0	Function	
0	0	0	Internal clock: ϕ*	(Initial value)
		1	Internal clock: ϕ/2	
	1	0	Internal clock: ϕ/4	
		1	Internal clock: ϕ/8	
1	0 or 1	0 or 1	Rising edges of the external event input (FTCI)	

Note: * When internal clock ϕ is selected, the counter operates by the subclock in subactive or subsleep mode.

Bit 3—Timer Output Level Setting D (TOD): Sets the value output from the FTIOD pin after reset until the first compare match D (TCNT and GRD matching signal) is generated. After a compare match is generated, FTIOD outputs the value specified in timer I/O control register 1 (IOD2 to IOD0).

Bit 3: TOD	Description	
0	FTIOD is 0	(Initial value)
1	FTIOD is 1	

Bit 2—Timer Output Level Setting C (TOC): Sets the value output from the FTIOC pin after reset until the first compare match C (TCNT and GRC matching signal) is generated. After a compare match is generated, FTIOC outputs the value specified in timer I/O control register 1 (IOC2 to IOC0).

Bit 2: TOC	Description	
0	FTIOC is 0	(Initial value)
1	FTIOC is 1	

Bit 1—Timer Output Level Setting B (TOB): Sets the value output from the FTIOB pin after reset until the first compare match B (TCNT and GRB matching signal) is generated. After a compare match is generated, FTIOB outputs the value specified in timer I/O control register 0 (IOB2 to IOB0).

Bit 1: TOB	Description	
0	FTIOB is 0	(Initial value)
1	FTIOB is 1	

Bit 0—Timer Output Level Setting A (TOA): Sets the value output from the FTIOA pin after reset until the first compare match A (TCNT and GRA matching signal) is generated. After a compare match is generated, FTIOA outputs the value specified in timer I/O control register 0 (IOA2 to IOA0).

Bit 0: TOA	Description	
0	FTIOA is 0	(Initial value)
1	FTIOA is 1	

12.2.3 Timer Interrupt Enable Register W (TIERW)

Bit	7	6	5	4	3	2	1	0
	OVIE	—	—	—	IMIED	IMIEC	IMIEB	IMIEA
Initial value	0	1	1	1	0	0	0	0
Read/Write	R/W	—	—	—	R/W	R/W	R/W	R/W

TIERW is an 8-bit read/write register that enables or disables the TCNT overflow interrupt request and general register (GRA, GRB, GRC, and GRD) compare match or input capture interrupt requests.

TIERW is initialized to H'70 by a reset.

Bit 7—Timer Overflow Interrupt Enable (OVIE): Enables or disables the FOVI interrupt requested by the OVF flag of TSRW when OVF is set to 1.

Bit 7: OVIE	Description	
0	FOVI interrupt requested by OVF flag is disabled	(Initial value)
1	FOVI interrupt requested by OVF flag is enabled	

Bits 6 to 4—Reserved: These bits cannot be modified and are always read as 1.

Bit 3—Input Capture/Compare Match Interrupt Enable D (IMIED): Enables or disables the IMID interrupt requested by the IMFD flag of TSRW when IMFD is set to 1.

Bit 3: IMIED	Description	
0	IMID interrupt requested by IMFD flag is disabled	(Initial value)
1	IMID interrupt requested by IMFD flag is enabled	

Bit 2—Input Capture/Compare Match Interrupt Enable C (IMIEC): Enables or disables the IMIC interrupt requested by the IMFC flag of TSRW when IMFC is set to 1.

Bit 2: IMIEC	Description	
0	IMIC interrupt requested by IMFC flag is disabled	(Initial value)
1	IMIC interrupt requested by IMFC flag is enabled	

Bit 1—Input Capture/Compare Match Interrupt Enable B (IMIEB): Enables or disables the IMIB interrupt requested by the IMFB flag of TSRW when IMFB is set to 1.

Bit 1: IMIEB	Description	
0	IMIB interrupt requested by IMFB flag is disabled	(Initial value)
1	IMIB interrupt requested by IMFB flag is enabled	

Bit 0—Input Capture/Compare Match Interrupt Enable A (IMIEA): Enables or disables the IMIA interrupt requested by the IMFA flag of TSRW when IMFA is set to 1.

Bit 0: IMIEA	Description	
0	IMIA interrupt requested by IMFA flag is disabled	(Initial value)
1	IMIA interrupt requested by IMFA flag is enabled	

12.2.4 Timer Status Register W (TSRW)

Bit	7	6	5	4	3	2	1	0
	OVF	—	—	—	IMFD	IMFC	IMFB	IMFA
Initial value	0	1	1	1	0	0	0	0
Read/Write	R/(W)*	—	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*

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

TSRW is an 8-bit read/write register that shows the TCNT overflow interrupt request and general register (GRA, GRB, GRC, and GRD) compare match or input capture interrupt requests.

TSRW is initialized to H'70 by a reset.

Bit 7—Timer Overflow Flag (OVF): This status flag indicates the TCNT has overflowed (from H'FFFF to H'0000). This flag is cleared by software and set by hardware; it cannot be set by software.

Bit 7: OVF	Description
0	[Clearing condition] (Initial value) Read OVF when OVF =1, then write 0 in OVF
1	[Setting condition] TCNT overflowed from H'FFFF to H'0000

Bits 6 to 4—Reserved: These bits cannot be modified and are always read as 1.

Bit 3—Input Capture/Compare Match Flag D (IMFD): This status flag indicates a GRD compare match or input capture event has occurred. This flag is cleared by software and set by hardware; it cannot be set by software.

Bit 3: IMFD	Description
0	[Clearing condition] (Initial value) Read IMFD when IMFD =1, then write 0 in IMFD
1	[Setting conditions] • TCNT = GRD when GRD functions as an output compare register • The TCNT value is transferred to GRD by an input capture signal when GRD functions as an input capture register

Bit 2—Input Capture/Compare Match Flag C (IMFC): This status flag indicates a GRC compare match or input capture event has occurred. This flag is cleared by software and set by hardware; it cannot be set by software.

Bit 2: IMFC	Description
0	[Clearing condition] (Initial value) Read IMFC when IMFC =1, then write 0 in IMFC
1	[Setting conditions] • TCNT = GRC when GRC functions as an output compare register • The TCNT value is transferred to GRC by an input capture signal when GRC functions as an input capture register

Bit 1—Input Capture/Compare Match Flag B (IMFB): This status flag indicates a GRB compare match or input capture event has occurred. This flag is cleared by software and set by hardware; it cannot be set by software.

Bit 1: IMFB	Description
0	[Clearing condition] (Initial value) Read IMFB when IMFB =1, then write 0 in IMFB
1	[Setting conditions] • TCNT = GRB when GRB functions as an output compare register • The TCNT value is transferred to GRB by an input capture signal when GRB functions as an input capture register

Bit 0—Input Capture/Compare Match Flag A (IMFA): This status flag indicates a GRA compare match or input capture event has occurred. This flag is cleared by software and set by hardware; it cannot be set by software.

Bit 0: IMFA	Description
0	[Clearing condition] (Initial value) Read IMFA when IMFA =1, then write 0 in IMFA
1	[Setting conditions] • TCNT = GRA when GRA functions as an output compare register • The TCNT value is transferred to GRA by an input capture signal when GRA functions as an input capture register

12.2.5 Timer I/O Control Register 0 (TIOR0)

Bit	7	6	5	4	3	2	1	0
	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
Initial value	1	0	0	0	1	0	0	0
Read/Write	—	R/W	R/W	R/W	—	R/W	R/W	R/W

TIOR0 is an 8-bit read/write register that selects the output compare or input capture function for GRA and GRB, and specifies the functions of the FTIOA and FTIOB pins. If the output compare function is selected, TIOR0 also selects the type of output. If input capture is selected, TIOR0 also selects the edges of the input capture signal.

TIOR0 is initialized to H'88 by a reset.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bits 6 to 4—I/O Control B2 to B0 (IOB2 to IOB0): These bits select the GRB function.

Bit 6: IOB2	Bit 5: IOB1	Bit 4: IOB0	Function	
0	0	0	GRB is an output compare register	No output at compare match (Initial value)
		1		0 output at GRB compare match*[1]
	1	0		1 output at GRB compare match*[1]
		1		Output toggles at GRB compare match*[1]
1	0	0	GRB is an input capture register	Input capture at rising edge
		1		Input capture at falling edge
	1	*[2]		Input capture at both edges
		*[2]		

Notes: 1. After a reset, the output conforms to the TOB setting in TCRW until the first compare match.
2. Don't care

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bits 2 to 0—I/O Control A2 to A0 (IOA2 to IOA0): These bits select the GRA function.

Bit 2: IOA2	Bit 1: IOA1	Bit 0: IOA0	Function	
0	0	0	GRA is an output compare register	No output at compare match (Initial value)
		1		0 output at GRA compare match*[1]
	1	0		1 output at GRA compare match*[1]
		1		Output toggles at GRA compare match*[1]
1	0	0	GRA is an input capture register	Input capture at rising edge
		1		Input capture at falling edge
	1	*[2]		Input capture at both edges
		*[2]		

Notes: 1. After a reset, the output conforms to the TOA setting in TCRW until the first compare match.
2. Don't care

12.2.6 Timer I/O Control Register 1 (TIOR1)

Bit	7	6	5	4	3	2	1	0
	—	IOD2	IOD1	IOD0	—	IOC2	IOC1	IOC0
Initial value	1	0	0	0	1	0	0	0
Read/Write	—	R/W	R/W	R/W	—	R/W	R/W	R/W

TIOR1 is an 8-bit read/write register that selects the output compare or input capture function for GRC and GRD, and specifies the functions of the FTIOC and FTIOD pins. If the output compare function is selected, TIOR0 also selects the type of output. If input capture is selected, TIOR0 also selects the edges of the input capture signal.

TIOR1 is initialized to H'88 by a reset.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bits 6 to 4—I/O Control D2 to D0 (IOD2 to IOD0): These bits select the GRD function.

Bit 6: IOD2	Bit 5: IOD1	Bit 4: IOD0	Function	
0	0	0	GRD is an output compare register	No output at compare match (Initial value)
		1		0 output at GRD compare match*[1]
	1	0		1 output at GRD compare match*[1]
		1		Output toggles at GRD compare match*[1]
1	0	0	GRD is an input capture register	Input capture at rising edge
		1		Input capture at falling edge
	1	*[2]		Input capture at both edges
		*[2]		

Notes: 1. After a reset, the output conforms to the TOD setting in TCRW until the first compare match.
2. Don't care

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bits 2 to 0—I/O Control C2 to C0 (IOC2 to IOC0): These bits select the GRC function.

Bit 2: IOC2	Bit 1: IOC1	Bit 0: IOC0	Function	
0	0	0	GRC is an output compare register	No output at compare match (Initial value)
		1		0 output at GRC compare match*[1]
	1	0		1 output at GRC compare match*[1]
		1		Output toggles at GRC compare match*[1]
1	0	0	GRC is an input capture register	Input capture at rising edge
		1		Input capture at falling edge
	1	*[2]		Input capture at both edges
		*[2]		

Notes: 1. After a reset, the output conforms to the TOC setting in TCRW until the first compare match.
2. Don't care

12.2.7 Timer Counter (TCNT)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNT is a 16-bit read/write up-counter that counts the pulses input from the internal or external clock source. The clock source is selected by bits CKS2 to CKS0 in TCRW.

TCNT can be cleared to H'0000 through a compare match with GRA by setting CCLR of TCRW to 1.

When TCNT overflows (changes from H'FFFF to H'0000), the OVF flag in TCRW is set to 1, and if OVIE in TIERW is set to 1 at this time, an interrupt request is sent to the CPU.

TCNT must be always read or written in 16-bit units; 8-bit access is not allowed.

TCNT is initialized to H'0000 by a reset.

12.2.8 General Registers A to D (GRA to GRD)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Each general register is a 16-bit read/write register that can function as either an output compare register or an input capture register. The function is selected by settings in TIOR0 and TIOR1.

When a general register is used as an output compare register, its value is constantly compared with the TCNT value. When the two values match (a compare match), the corresponding flag (IMFA, IMFB, IMFC, or IMFD) in TSRW is set to 1. If the corresponding interrupt enable bit (IMIEA, IMIEB, IMIEC, or IMIED) in TIERW is set to 1 at this time, an interrupt request is sent to the CPU. Compare match output can be selected in TIOR0 and TIOR1.

When a general register is used as an input capture register, an external input capture signal is detected and the current TCNT value is stored in the general register. The corresponding flag (IMFA, IMFB, IMFC, or IMFD) in TSRW is set to 1. If the corresponding interrupt enable bit (IMIEA, IMIEB, IMIEC, or IMIED) in TIERW is set to 1 at this time, an interrupt request is sent to the CPU. The edges of the input capture signal are selected in TIOR0 and TIOR1.

GRC and GRD can be used as the buffer registers of GRA and GRB, respectively, by setting BUFEA and BUFEB in TMRW.

The general registers are initialized to H'FFFF by a reset.

12.3 CPU Interface

12.3.1 16-Bit Registers

TCNT, GRA, GRB, GRC, and GRD are 16-bit registers connected to the CPU by an on-chip 16-bit data bus. These registers must be written or read in 16-bit units; 8-bit access is not allowed.

Figure 12.2 shows the interface between the CPU and a 16-bit register.

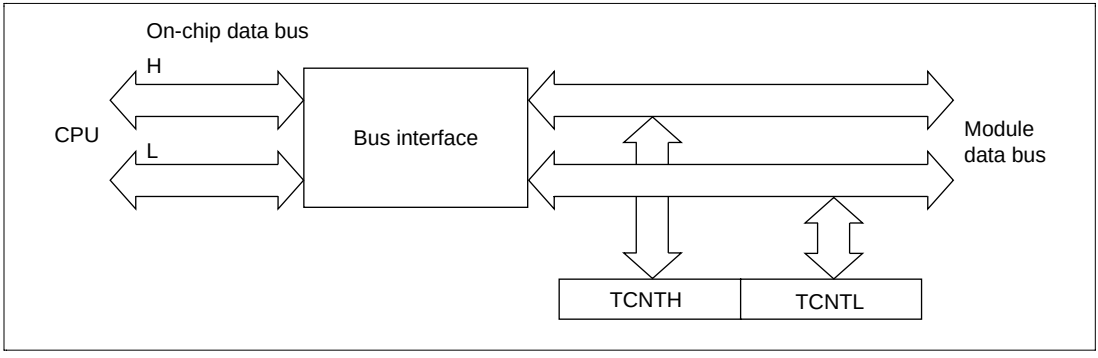


Figure 12.2 16-Bit Register Interface (CPU and TCNT (16 bits))

12.3.2 8-Bit Registers

The registers other than TCNT, GRA, GRB, GRC, and GRD are 8-bit registers, and are connected to the CPU by an on-chip 8-bit data bus.

Figure 12.3 shows the interface between the CPU and an 8-bit register.

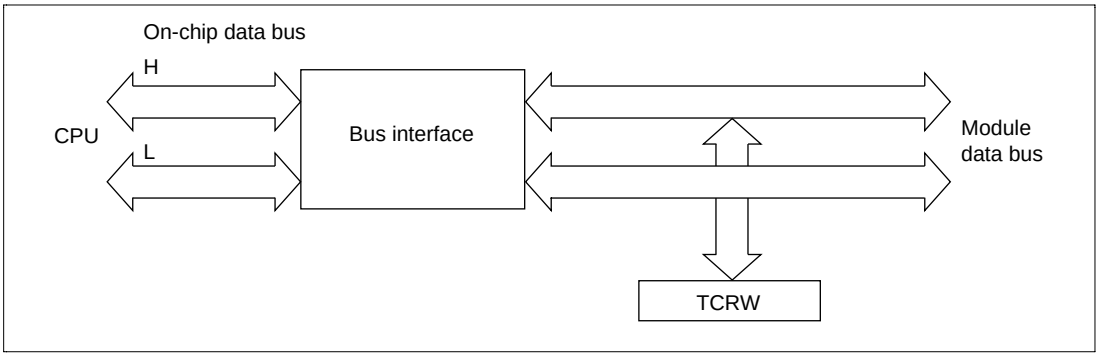


Figure 12.3 8-Bit Register Interface (CPU and TCRW (8 bits))

12.4 Operation

12.4.1 Overview

A summary of operations in the various modes is given below.

- Normal Operation

  Timer W has a timer counter (TCNT) and general registers (GRA to GRD). TCNT is a 16-bit counter that increments the count each time a clock pulse is input, and that can operate as a free-running counter or an external event counter. GRA to GRD can be used for input capture or output compare.

- Buffer Operation

  If a compare match is generated when a GR is used as an output compare register, , the corresponding buffer register value is transferred to the GR. If input capture is generated when a GR is used as an input capture register, the GR value is transferred to the buffer register and the TCNT value is transferred to the GR.

- PWM Mode

  PWM waveforms are output from the FTIOB, FTIOC, and FTIOD pins. The period can be specified in the GRA, and the duty ratio can be varied from 0% to 100% depending on the settings of GRB to GRD. When an output pin is set to PWM mode, the corresponding GR automatically becomes an output compare register.

Counting Operation: TCNT performs free-running or periodic counting operations. Figure 12.4 shows an example of the setup procedure for counting.

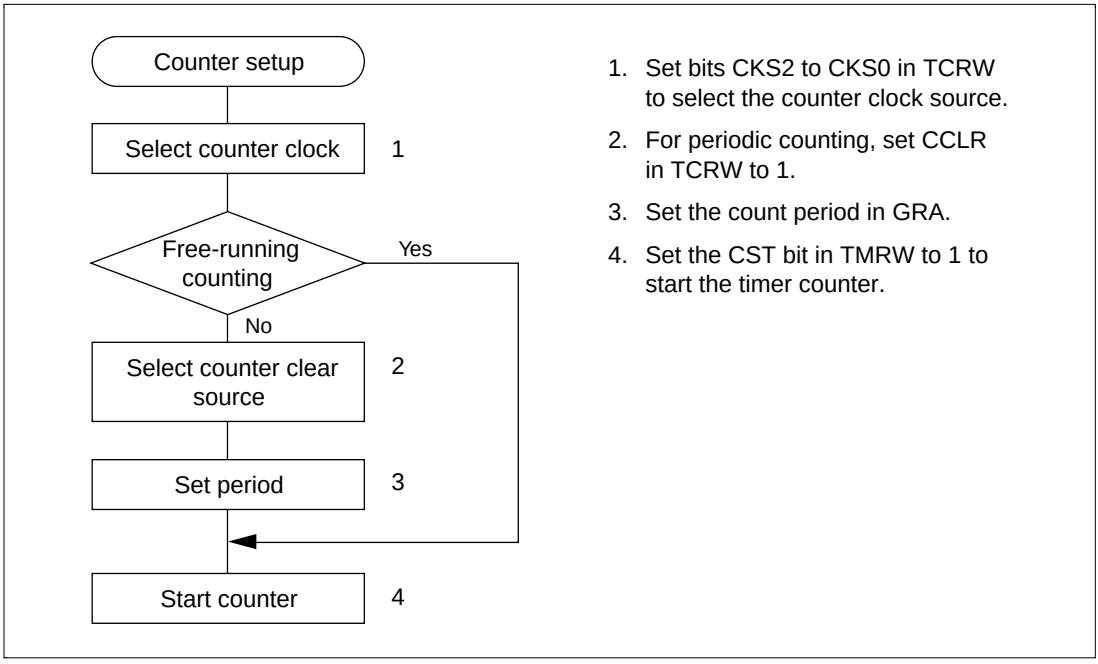


Figure 12.4 Counter Setup Procedure (Example)

After a reset, TCNT is initialized to H'0000 and set as a free-running counter. A free-running counter starts incrementing the count when the CST bit in TMRW is set to 1. The input clock source can be selected from four internal clocks and an external clock by the CKS2 to CKS0 bits in TCRW. When the count overflows from H'FFFF to H'0000, the OVF flag in TSRW is set to 1. If the OVIE bit in TIERW is set to 1 at this time, an interrupt request is sent to the CPU. After the overflow, the counter continues counting from H'0000. Figure 12.5 illustrates free-running counting.

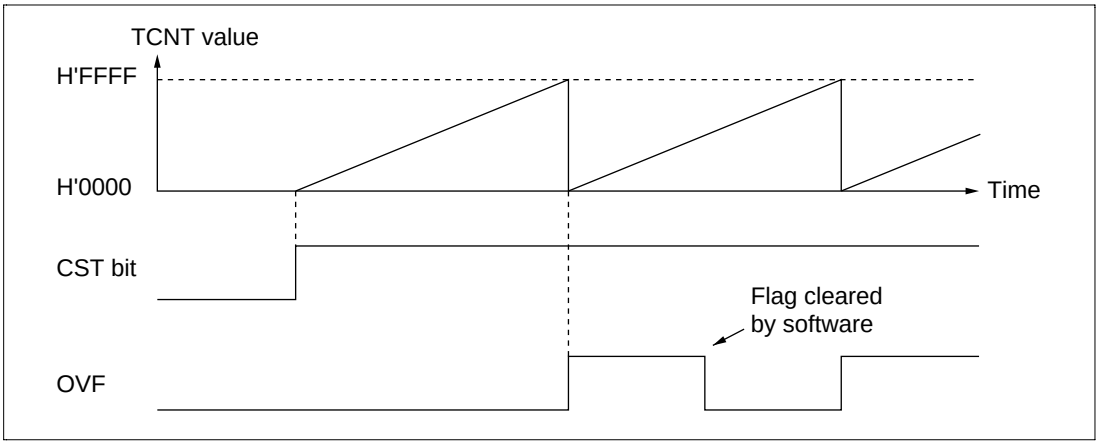


Figure 12.5 Free-Running Counter Operation

When compare match is selected as the TCNT clear source, TCNT operates as a periodic counter. Select the output compare function for GRA, set bit CCLR in TCRW to 1, and set the count period in GRA. When the count matches GRA, the IMFA flag in TSRW is set to 1 and TCNT is cleared to H'0000. If the IMIEA bit in TSRW is set to 1 at this time, an interrupt request is sent to the CPU. After the compare match, TCNT continues counting from H'0000. Figure 12.6 illustrates periodic counting.

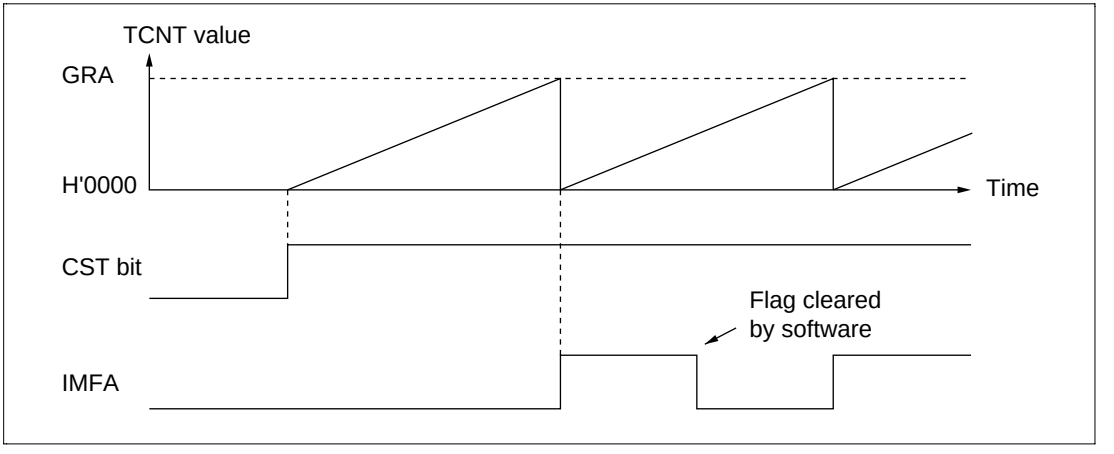


Figure 12.6 Periodic Counter Operation

Signal Output by Compare Match: Compare match A, B, C, or D can cause the output at the FTIOA, FTIOB, FTIOC, or FTIOD pin to go to 0, go to 1, or toggle. Figure 12.7 shows an example of the setup procedure for outputting signals by compare match.

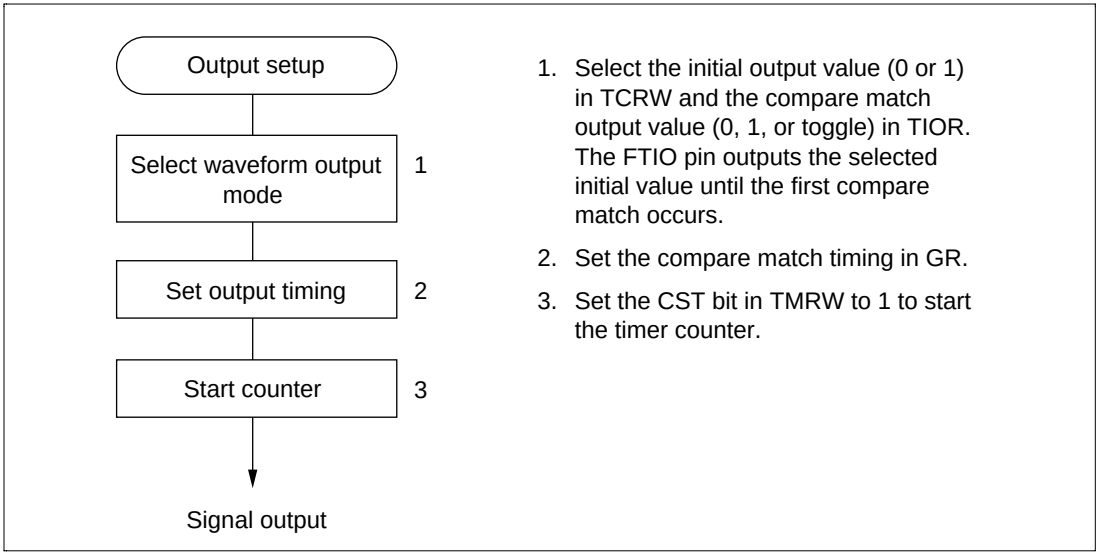


Figure 12.7 Setup Procedure for Outputting Signals by Compare Match (Example)

Figure 12.8 shows an example of 0 and 1 output. TCNT operates as a free-running counter, 1 output is selected for compare match A, and 0 output is selected for compare match B. When the signal is already at the selected output level, the signal level does not change at compare match.

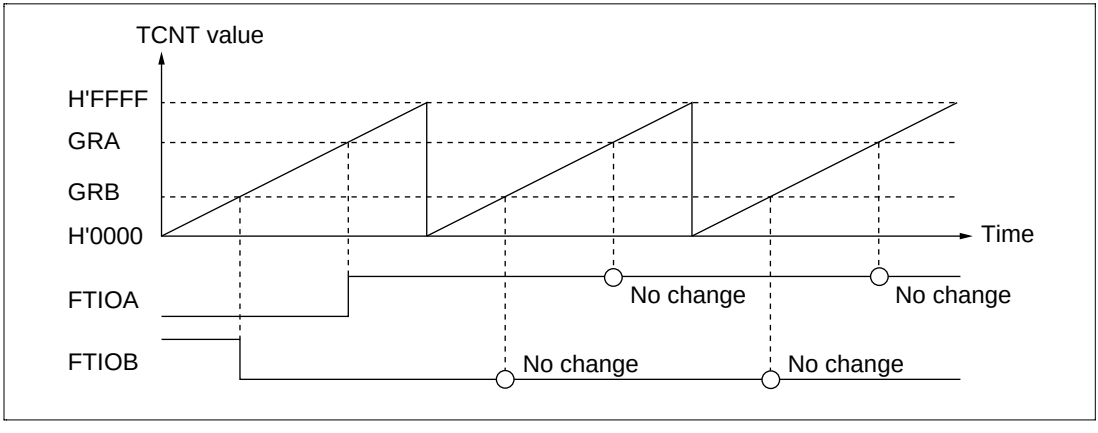


Figure 12.8 0 and 1 Output (TOA = 0, TOB = 1)

Figure 12.9 shows an example of toggle output. TCNT operates as a free-running counter, and toggle output is selected for both compare match A and B.

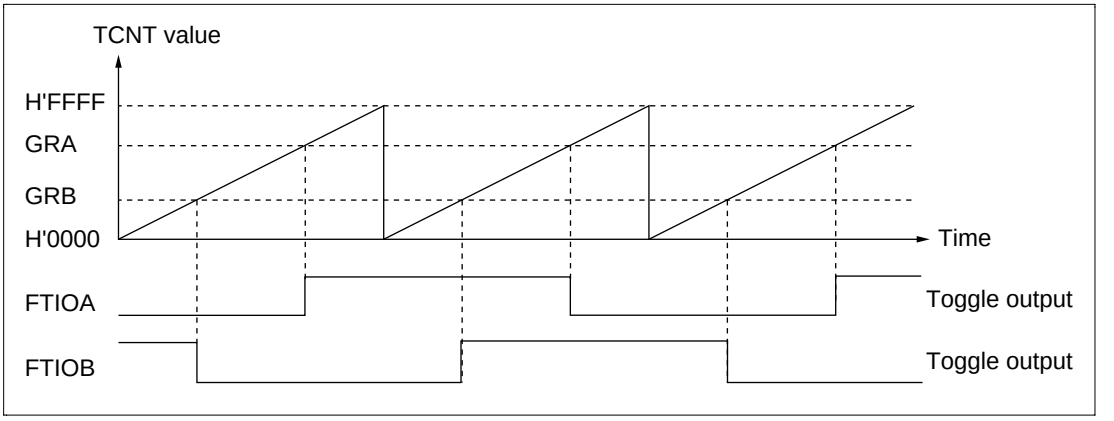


Figure 12.9 Toggle Output (1) (TOA = 0, TOB = 1)

Figure 12.10 shows another example of toggle output. TCNT operates as a periodic counter, cleared by compare match A. Toggle output is selected for both compare match A and B.

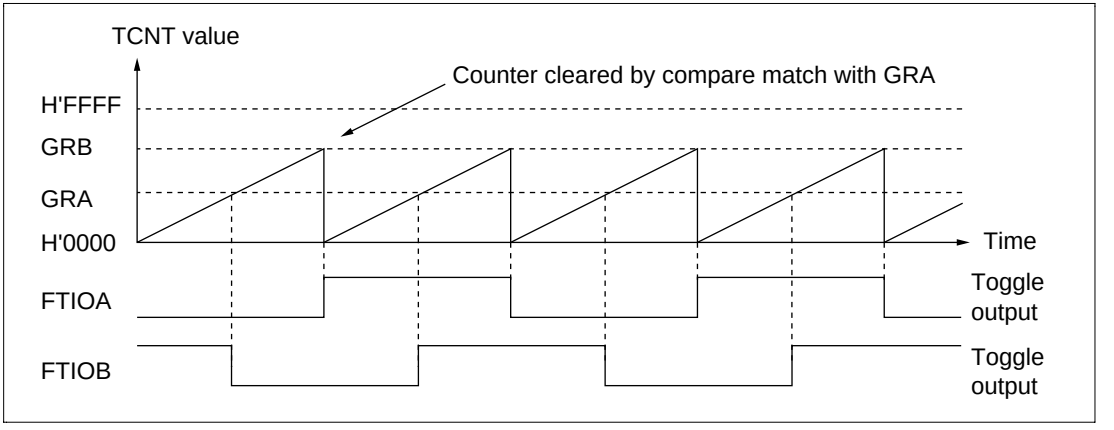
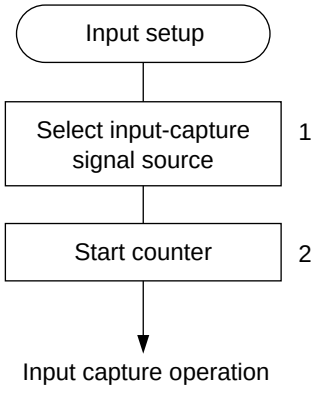


Figure 12.10 Toggle Output (2) (TOA = 0, TOB = 1)

Input Capture Function: The TCNT value can be captured into a general register (GRA, GRB, GRC, or GRD) when a signal level changes at an input capture pin (FTIOA, FTIOB, FTIOC, or FTIOD). Capture can take place on the rising edge, falling edge, or both edges. Figure 12.11 shows an example of the procedure for setting up input capture.



1. Set TIOR to select the input capture function for a general register, and select the input-capture signal source and the capture edges (rising edge, falling edge, or both edges) of the input capture signal.
2. Set the CST bit in TMRW to 1 to start the timer counter.

Figure 12.11 Setup Procedure for Input Capture (Example)

Figure 12.22 shows an example of input capture when both edges of FTIOA and the falling edge of FTIOB are selected as capture edges. TCNT operates as a free-running counter.

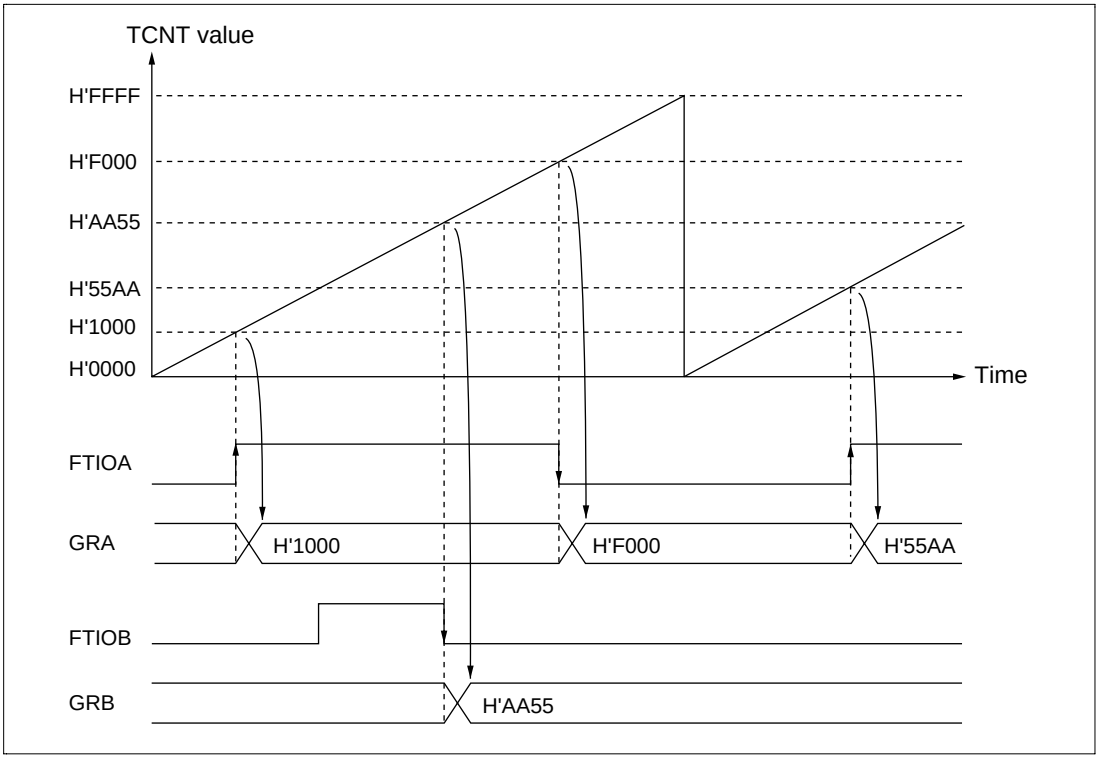


Figure 12.12 Input Capture (Example)

Buffer Operation: GRC and GRD can be used as buffer registers for GRA and GRB, respectively. The buffer operation differs depending on whether the GR is set as an input capture register or as an output compare register.

Table 12.4 shows the register combination for buffer operation.

Table 12.4 Register Combination for Buffer Operation

General Register	Buffer Register
GRA	GRC
GRB	GRD

- When GR Operates as an Output Compare Register

  When a compare match occurs, the buffer register value is transferred to the corresponding general register. Figure 12.13 shows the compare buffer operation.

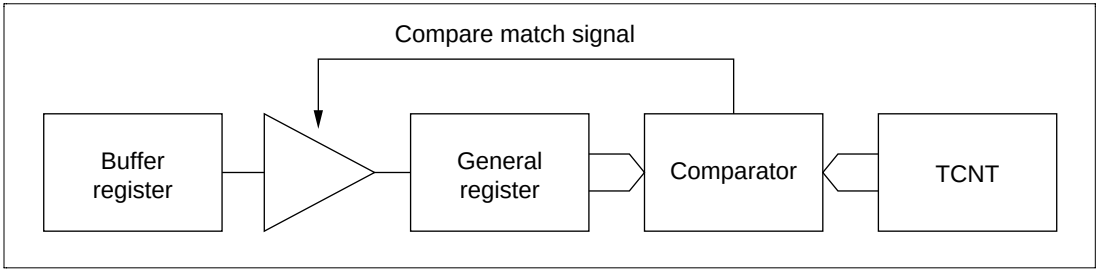


Figure 12.13 Compare Buffer Operation

- When GR Operates as an Input Capture Register

  When an input capture occurs, the general register value is transferred to the corresponding buffer register, and the TCNT value is transferred to the general register. Figure 12.14 shows the input capture buffer operation.

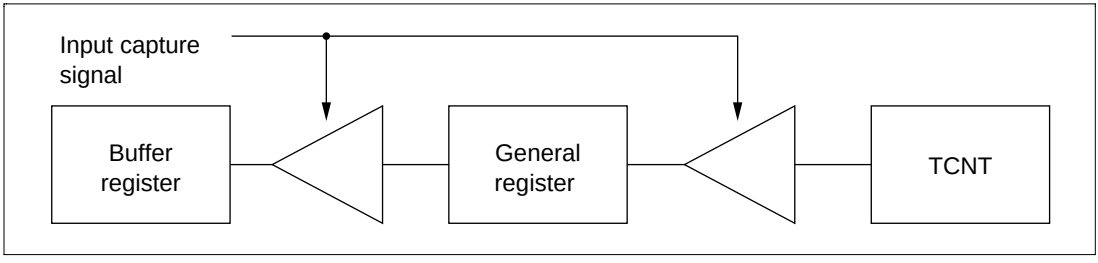
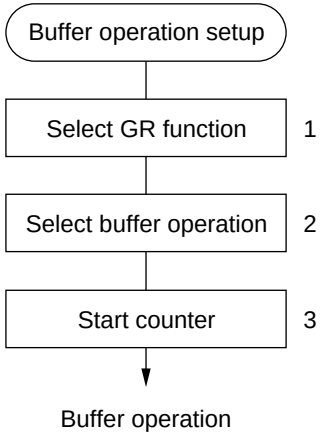


Figure 12.14 Input Capture Buffer Operation

Figure 12.15 shows an example of the procedure for setting up buffer operation.



1. Set TIOR to select the input capture or output compare function for a general register.
2. Set the BUFEA or BUFEB bit in TMRW to select the buffer operation for a general register.
3. Set the CST bit in TMRW to 1 to start the timer counter.

Figure 12.15 Setup Procedure for Buffer Operation (Example)

Figure 12.16 shows an example of buffer operation when the FTIOB pin is set to PWM mode and GRD is set as the buffer register for GRB. TCNT is cleared by compare match A, and FTIOB outputs 1 at compare match B and 0 at compare match A.

Every time compare match B occurs, the FTIOB output level changes and the value of buffer register GRD is transferred to GRB.

For details on PWM mode, refer to the description of PWM Operation in this section.

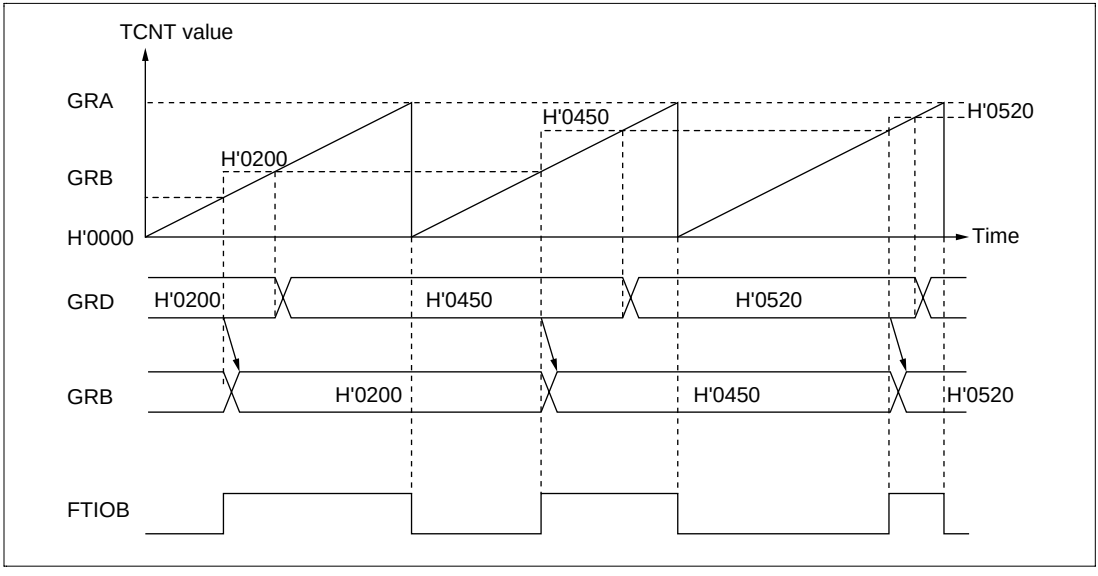


Figure 12.16 Buffer Operation Example (Output Compare)

Figure 12.17 shows an example of buffer operation when the GRA is set as an input capture register and GRC is set as the buffer register for GRA. TCNT operates as a free-running counter, and FTIOA captures both rising and falling edges of the input signal.

Every time compare match A occurs, the GRA value is transferred to GRC and the TCNT value is stored in GRA.

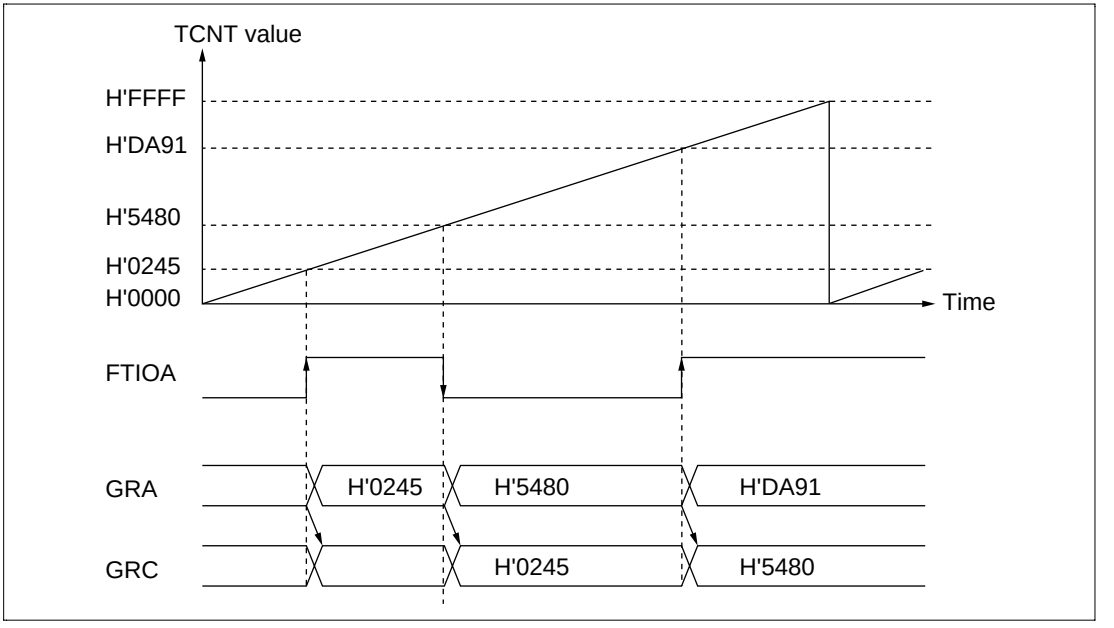


Figure 12.17 Buffer Operation Example (Input Capture)

PWM Operation: In PWM mode, PWM waveforms are output from the FTIOB, FTIOC, and FTIOD pins. PWM waveforms are generated by using GRA as the period register and GRB, GRC, and GRD as duty registers. The output level of each pin depends on the corresponding timer output level set bit (TOB, TOC, or TOD) in TCRW. For example, when TOB is 1, the FTIOB output goes to 0 at compare match B and to 1 at compare match A. When TOB is 0, the FTIOB output goes to 1 at compare match B and to 0 at compare match A. Thus the compare match output level settings in the timer I/O control register are ignored for the output pin set to PWM mode.

If the same value is set in the period register and the duty register, the output does not change when a compare match occurs.

Up to three-phase PWM waveforms can be output.

Figure 12.18 shows an example of a procedure for setting up PWM mode.

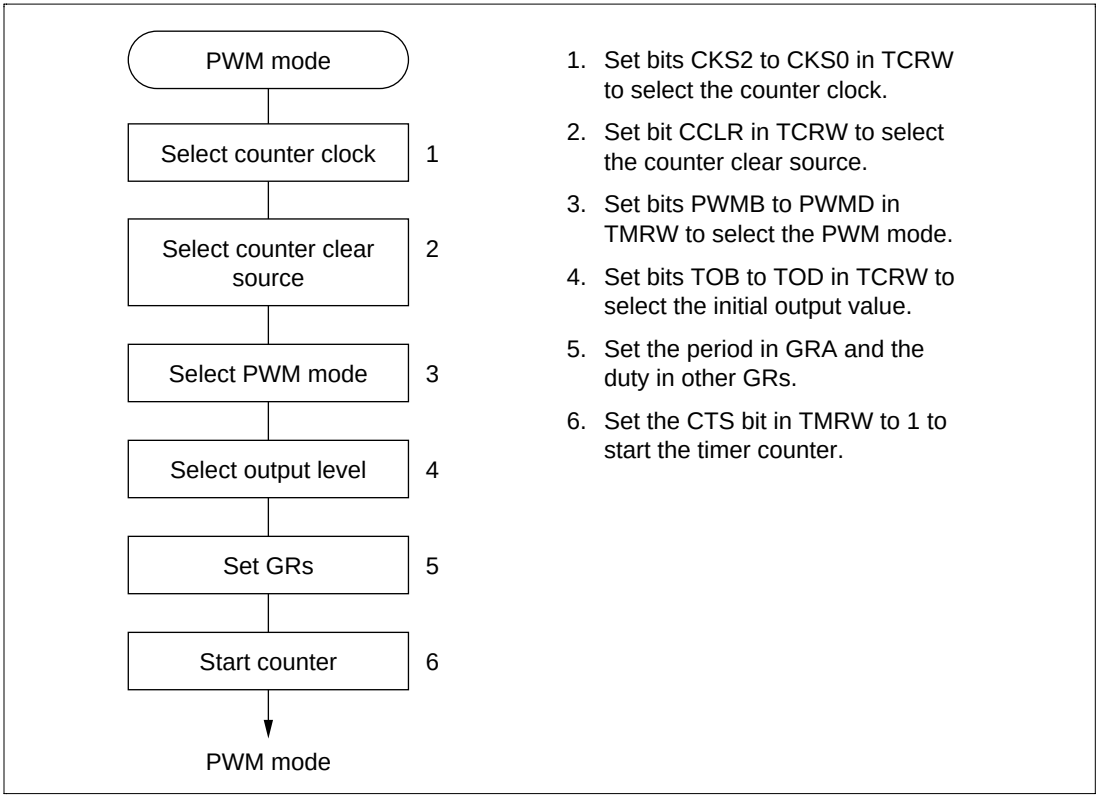


Figure 12.18 Setup Procedure for PWM Mode (Example)

Figure 12.19 shows an example of operation in PWM mode. The output signals go to 1 and TCNT is cleared at compare match A, and the output signals go to 0 at compare match B, C, and D (TOB, TOC, and TOD = 1: initial output values are set to 1).

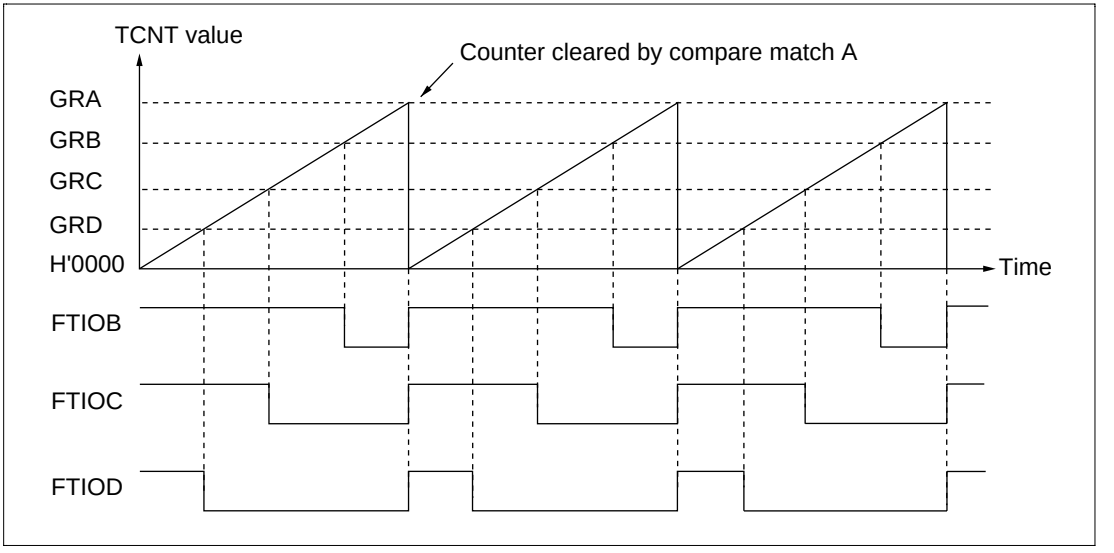


Figure 12.19 PWM Mode (Example 1)

Figure 12.20 shows another example of operation in PWM mode. The output signals go to 0 and TCNT is cleared at compare match A, and the output signals go to 1 at compare match B, C, and D (TOB, TOC, and TOD = 0: initial output values are set to 1).

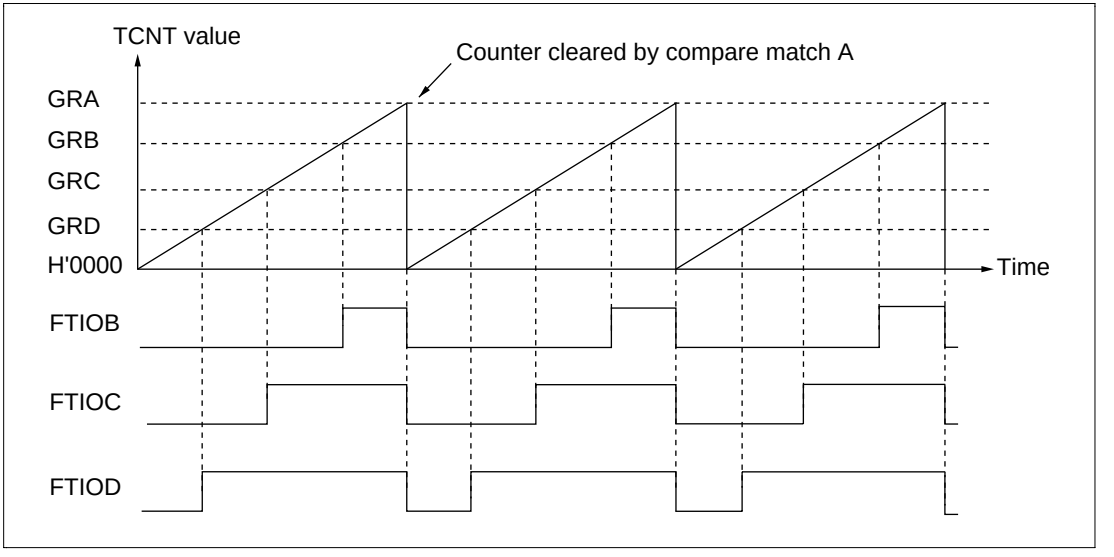


Figure 12.20 PWM Mode (Example 2)

Figures 12.21 and 12.22 show examples of the output of PWM waveforms with duty cycles of 0% and 100%.

- TOB, TOC, and TOD = 0: initial output values are set to 0

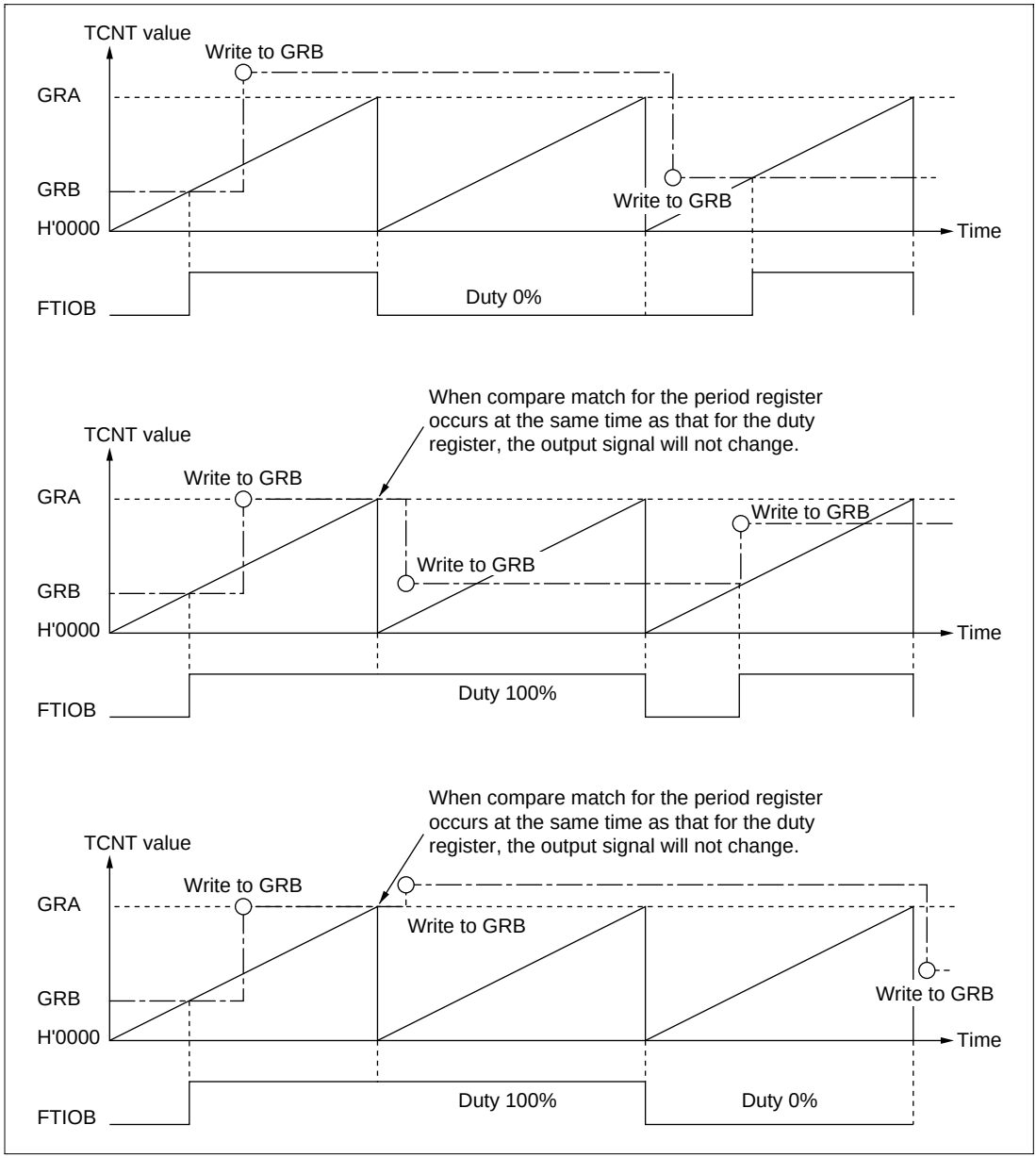


Figure 12.21 PWM Mode (Example 3)

- TOB, TOC, and TOD = 1: initial output values are set to 1

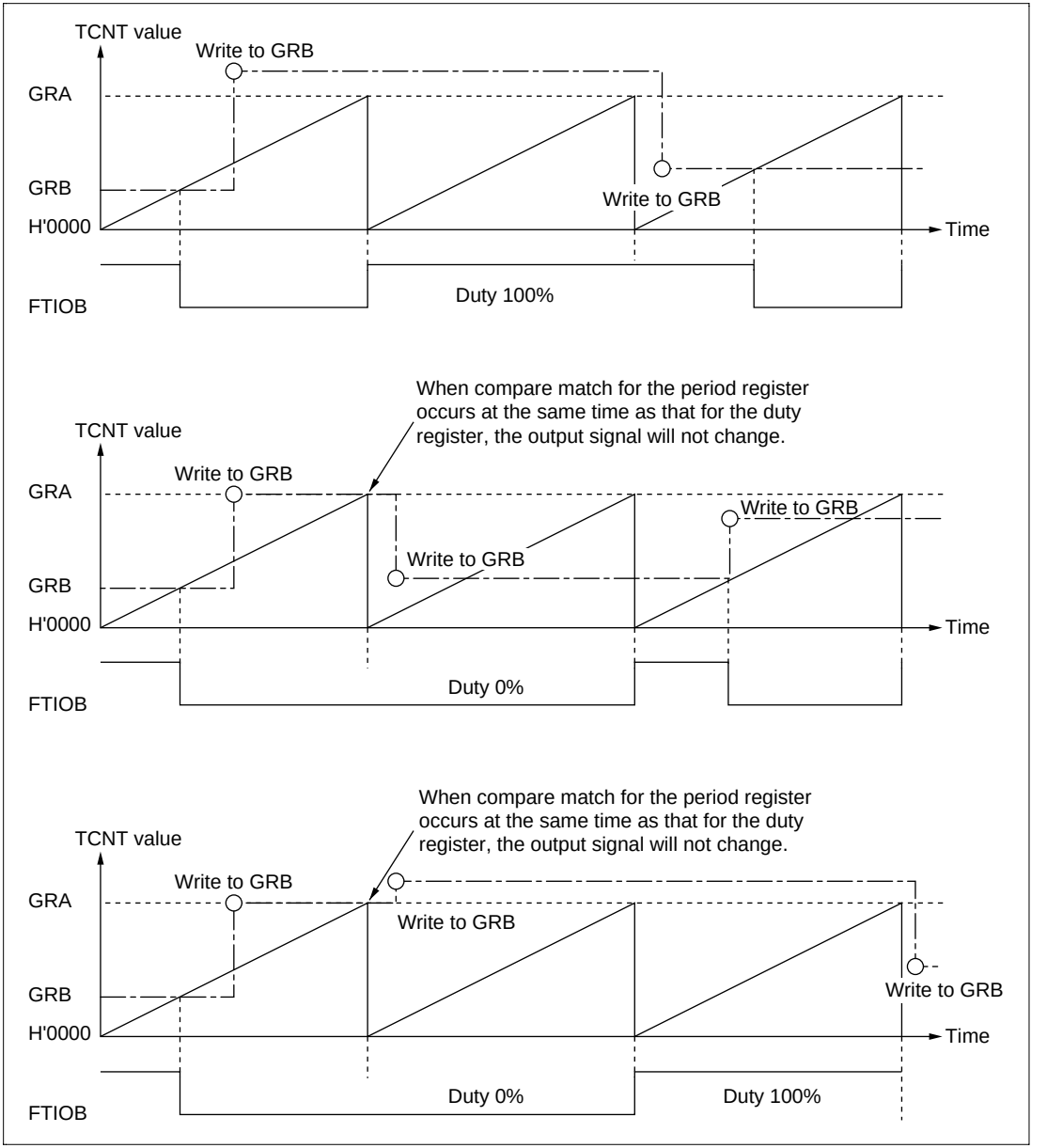


Figure 12.22 PWM Mode (Example 4)

12.4.2 Operation Timing

TCNT Count Timing: Figure 12.23 shows the TCNT count timing when the internal clock source is selected. Figure 12.24 shows the timing when the external clock source is selected. The pulse width of the external clock signal must be at least two system clock (ϕ) cycles; shorter pulses will not be counted correctly.

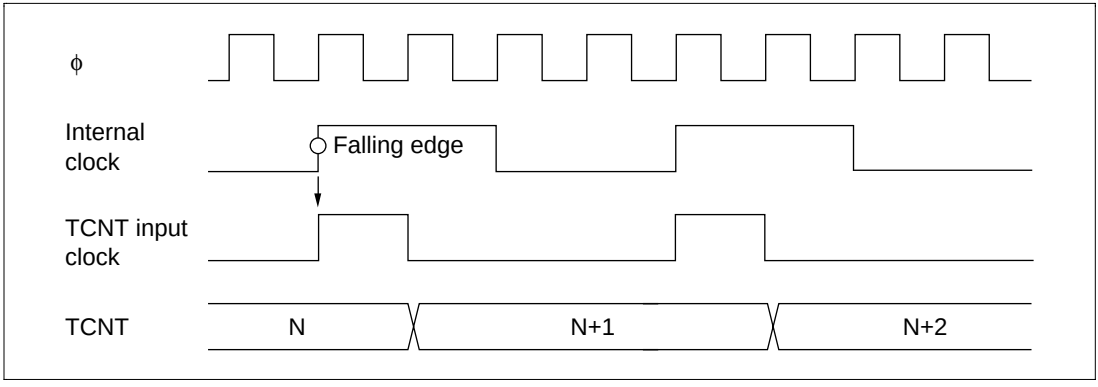


Figure 12.23 Count Timing for Internal Clock Source

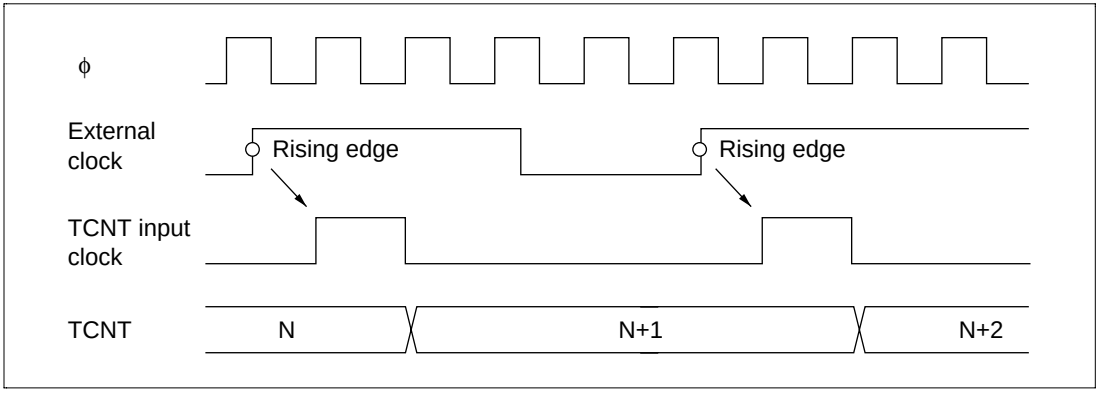


Figure 12.24 Count Timing for External Clock Source

Output Compare Timing: The compare match signal is generated in the last state in which TCNT and the general register match (when TCNT changes from the matching value to the next value). When the compare match signal is generated, the output value selected in TIOR is output at the compare match output pin (FTIOA, FTIOB, FTIOC, or FTIOD). When TCNT matches a general register, the compare match signal is generated only after the next counter clock pulse is input.

Figure 12.25 shows the output compare timing.

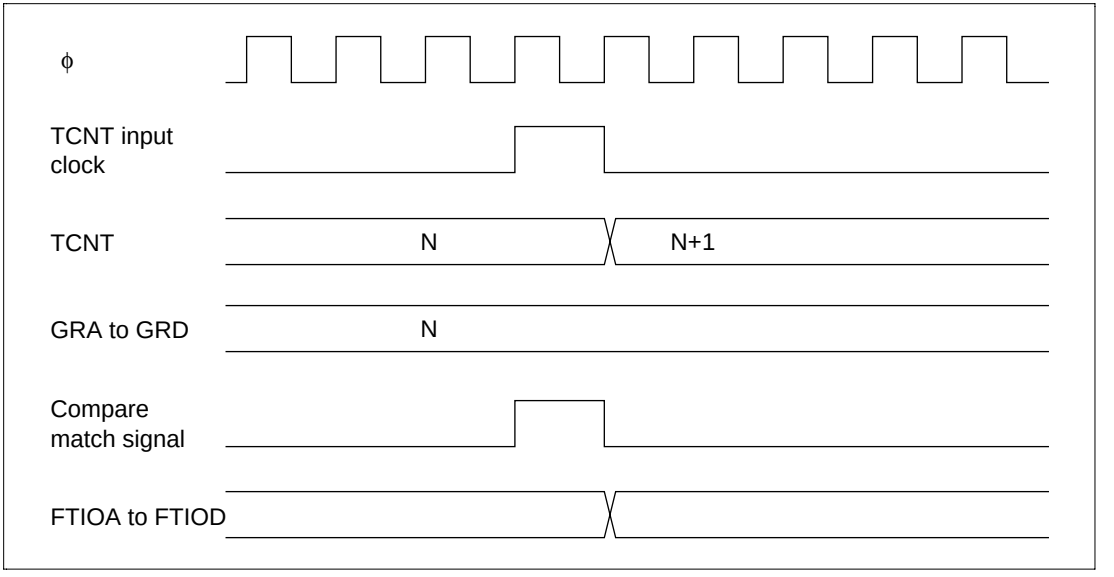


Figure 12.25 Output Compare Timing

Input Capture Timing: Input capture on the rising edge, falling edge, or both edges can be selected through settings in TIOR0 and TIOR1. Figure 12.26 shows the timing when the falling edge is selected. The pulse width of the input capture signal must be at least two system clock (ϕ) cycles; shorter pulses will not be detected correctly.

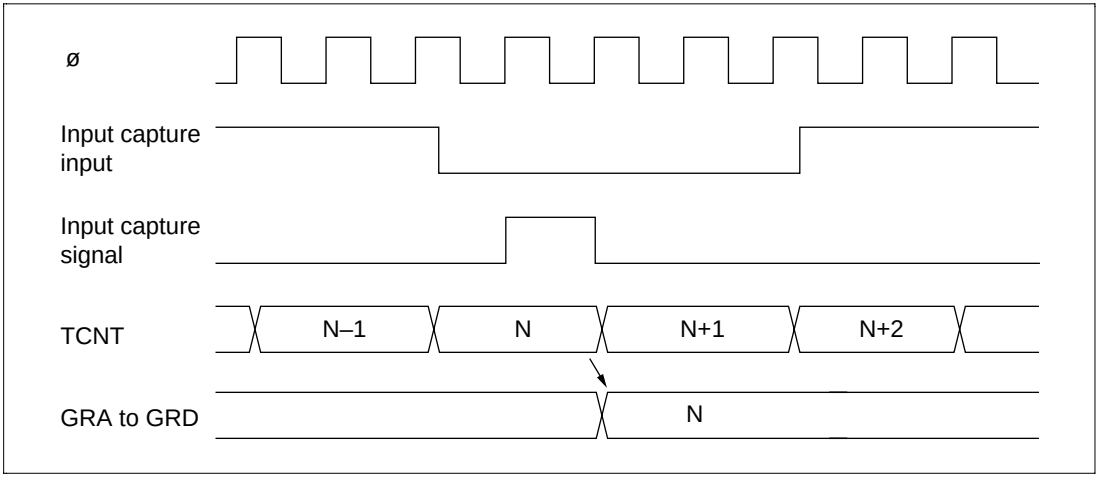


Figure 12.26 Input Capture Timing

Timing of Counter Clearing by Compare Match: Figure 12.27 shows the timing when the counter is cleared by compare match A.

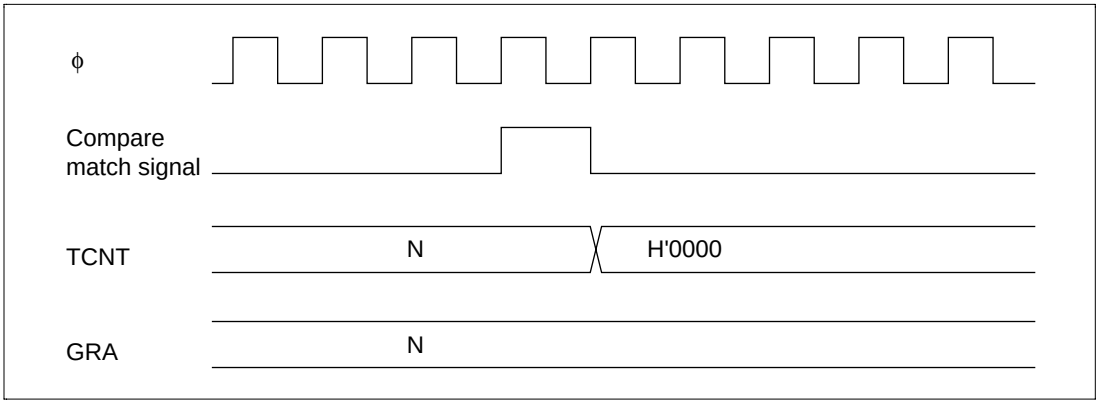


Figure 12.27 Timing of Counter Clearing by Compare Match

Buffer Operation Timing: Figures 12.28 and 12.29 show the buffer operation timing.

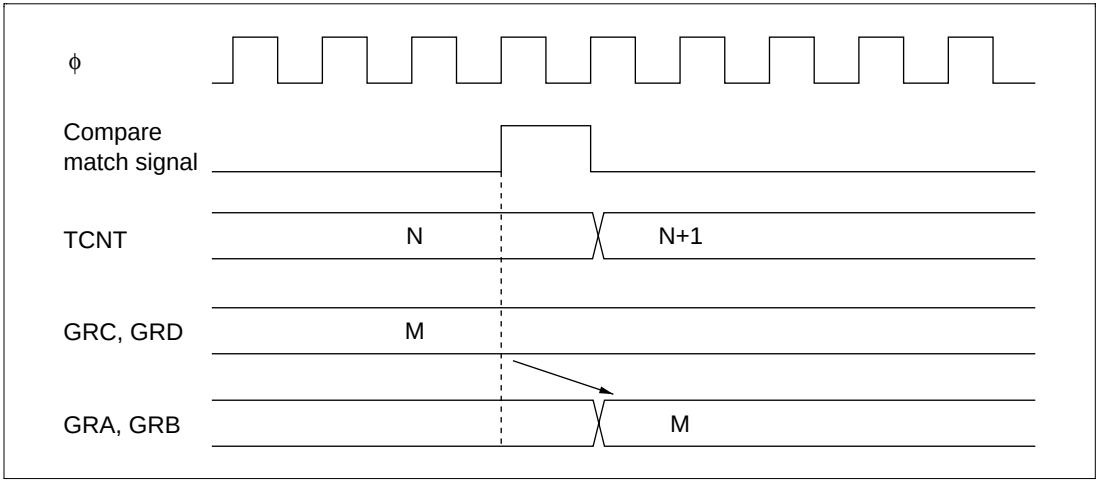


Figure 12.28 Buffer Operation Timing (Compare Match)

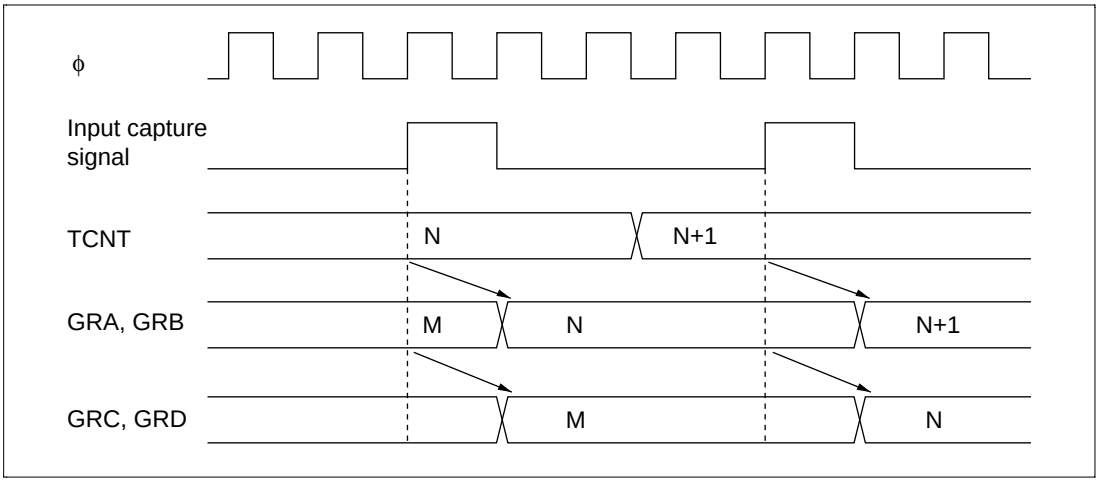


Figure 12.29 Buffer Operation Timing (Input Capture)

Timing of IMFA to IMFD Flag Setting at Compare Match: If a general register (GRA, GRB, GRC, or GRD) is used as an output compare register, the corresponding IMFA, IMFB, IMFC, or IMFD flag is set to 1 when TCNT matches the general register. The compare match signal is generated in the last state in which the values match (when TCNT is updated from the matching count to the next count). Therefore, when TCNT matches a general register, the compare match signal is generated only after the next TCNT clock pulse is input. Figure 12.30 shows the timing of the IMFA to IMFD flag setting at compare match.

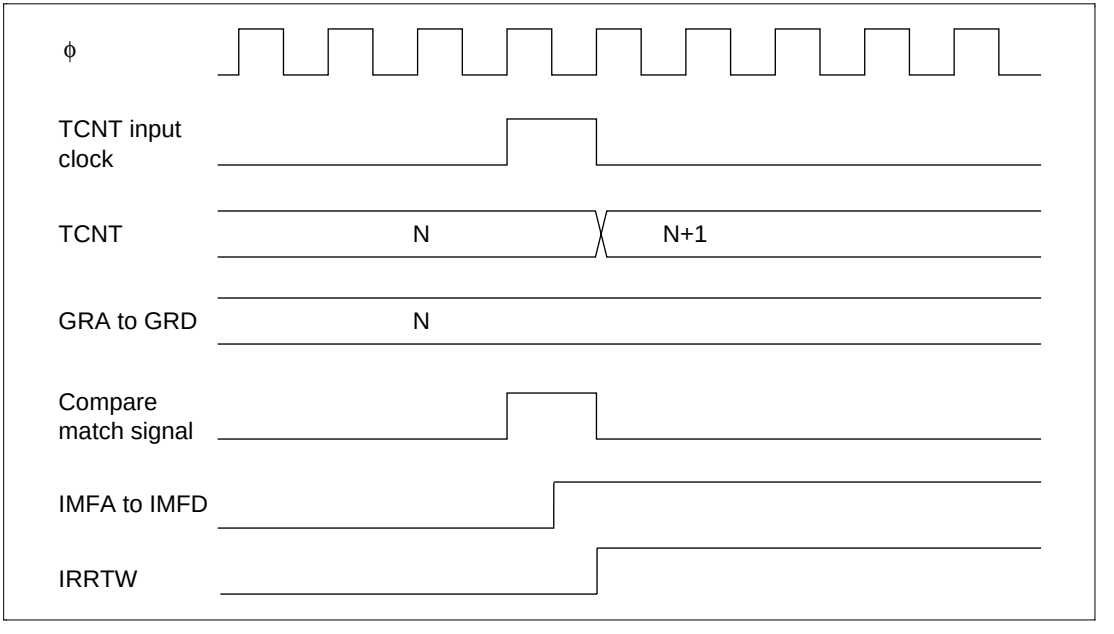


Figure 12.30 Timing of IMFA to IMFD Flag Setting at Compare Match

Timing of IMFA to IMFD Setting at Input Capture: If a general register (GRA, GRB, GRC, or GRD) is used as an input capture register, the corresponding IMFA, IMFB, IMFC, or IMFD flag is set to 1 when an input capture occurs. Figure 12.31 shows the timing of the IMFA to IMFD flag setting at input capture.

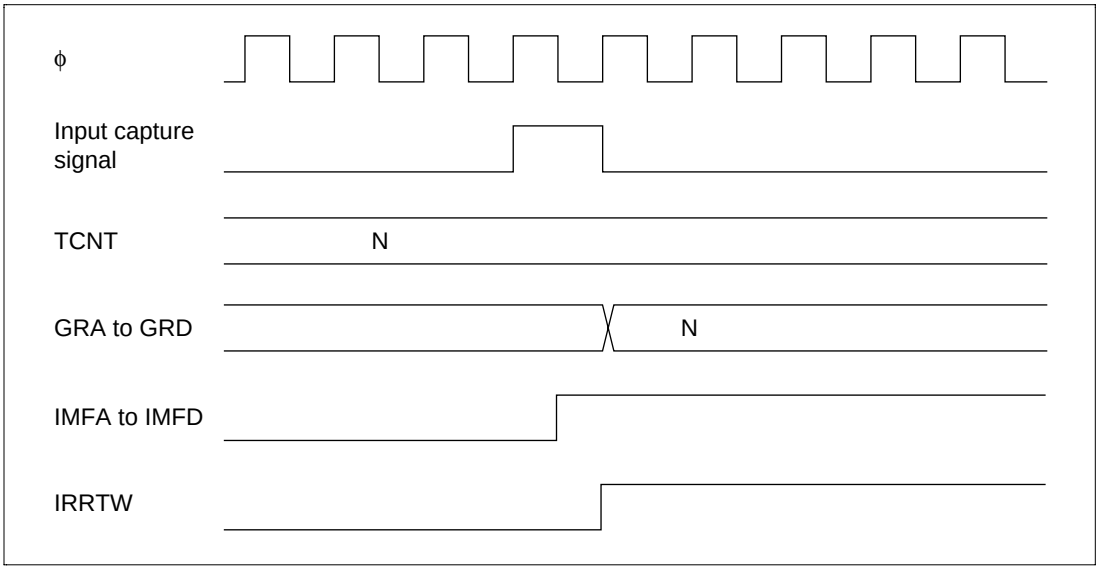


Figure 12.31 Timing of IMFA to IMFD Flag Setting at Input Capture

Timing of Status Flag Clearing: When the CPU reads a status flag while it is set to 1, then writes 0 in the status flag, the status flag is cleared. Figure 12.32 shows the status flag clearing timing.

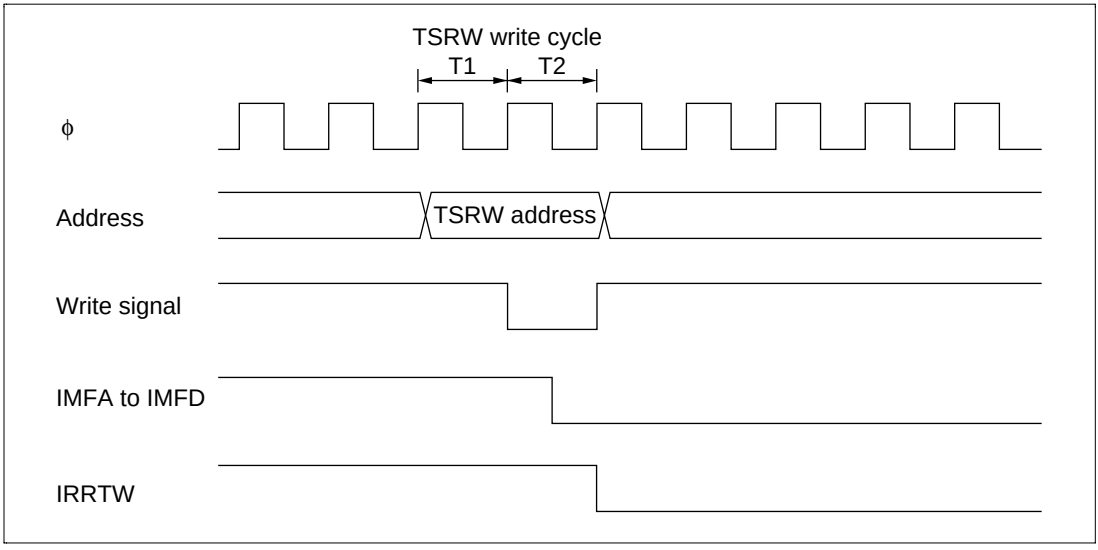


Figure 12.32 Timing of Status Flag Clearing

12.5 Usage Notes

Input Pulse Width: The pulse width of the input clock signal and the input capture signal must be at least two system clock (ϕ) cycles; shorter pulses will not be detected correctly.

Note on Waveform Period Setting: When compare match is selected as the TCNT clearing source, TCNT is cleared in the last state in which the TCNT value matches GRA (when TCNT is updated from the matching count to the next count). The actual counter frequency is therefore given by the following formula:

$$f = \frac{\phi}{(N + 1)}$$

(f: counter frequency; ϕ: system clock frequency; N: value set in GRA)

Contention between TCNT Write and Clear: If a counter clear signal occurs in the T2 state of a TCNT write cycle, clearing of the counter takes priority and the write is not performed. Figure 12.33 shows this timing.

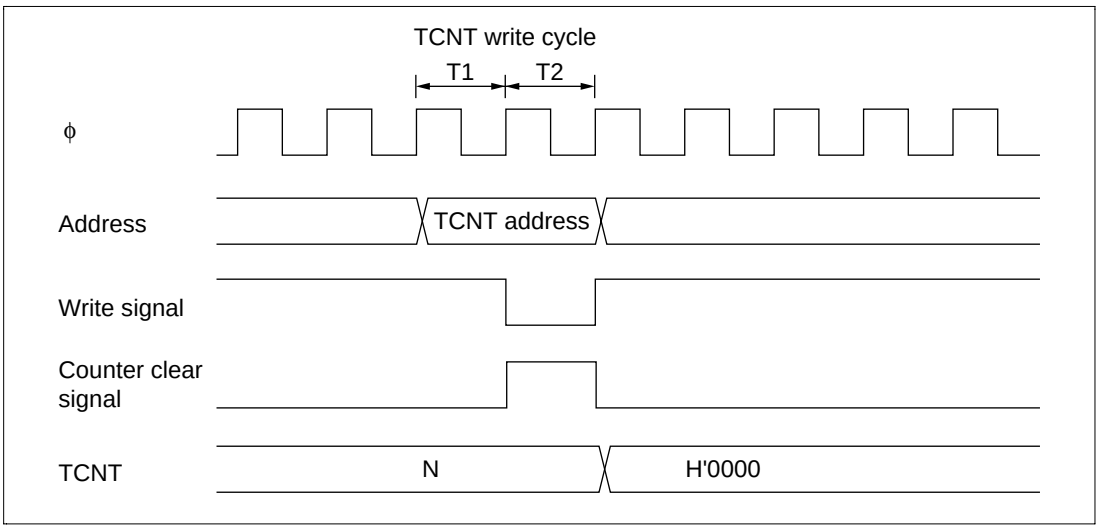


Figure 12.33 Contention between TCNT Write and Clear

Contention between TCNT Write and Increment: If an increment pulse occurs in the T2 state of a TCNT write cycle, writing takes priority and TCNT is not incremented. Figure 12.34 shows this timing.

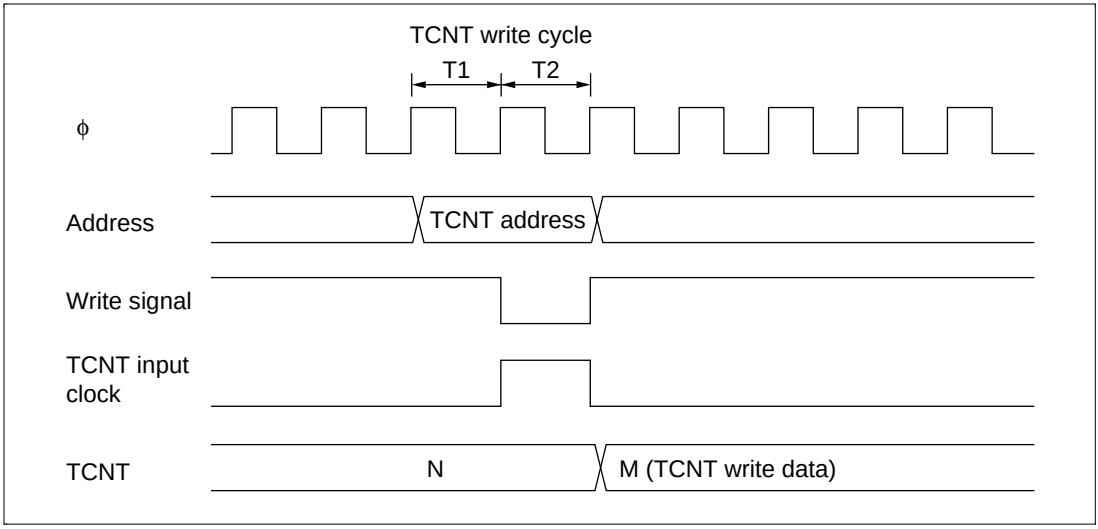


Figure 12.34 Contention between TCNT Write and Increment

Contention between General Register Write and Compare Match in Buffer Operation: If a compare match occurs in the T2 state of a general register write cycle, writing takes priority and the buffer operation (data transfer from the buffer register to the general register) is not performed. Figure 12.35 shows this timing.

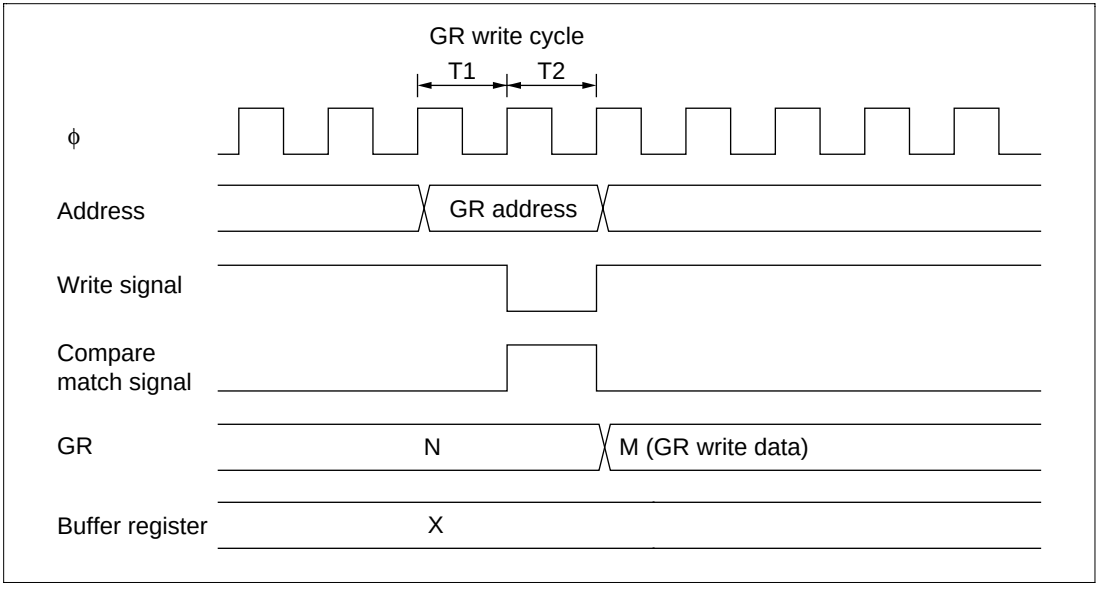


Figure 12.35 Contention between General Register Write and Compare Match in Buffer Operation

Contention between Buffer Register Write and Compare Match in Buffer Operation: If a compare match occurs in the T2 state of a buffer register write cycle, the old data (before being updated) in the buffer register is transferred to the general data in a buffer operation. Figure 12.36 shows this timing.

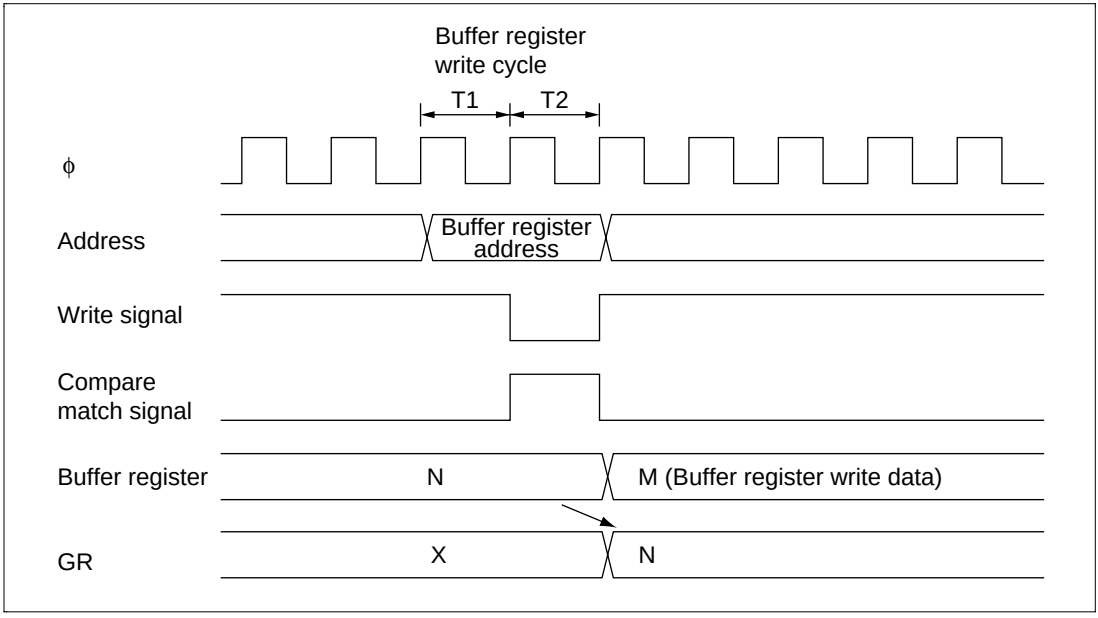


Figure 12.36 Contention between Buffer Register Write and Compare Match

Contention between General Register Write and Compare Match: If a compare match (TCNT = GR data before writing) occurs in the T2 state of a general register write cycle, the compare match signal is generated. Figure 12.37 shows this timing.

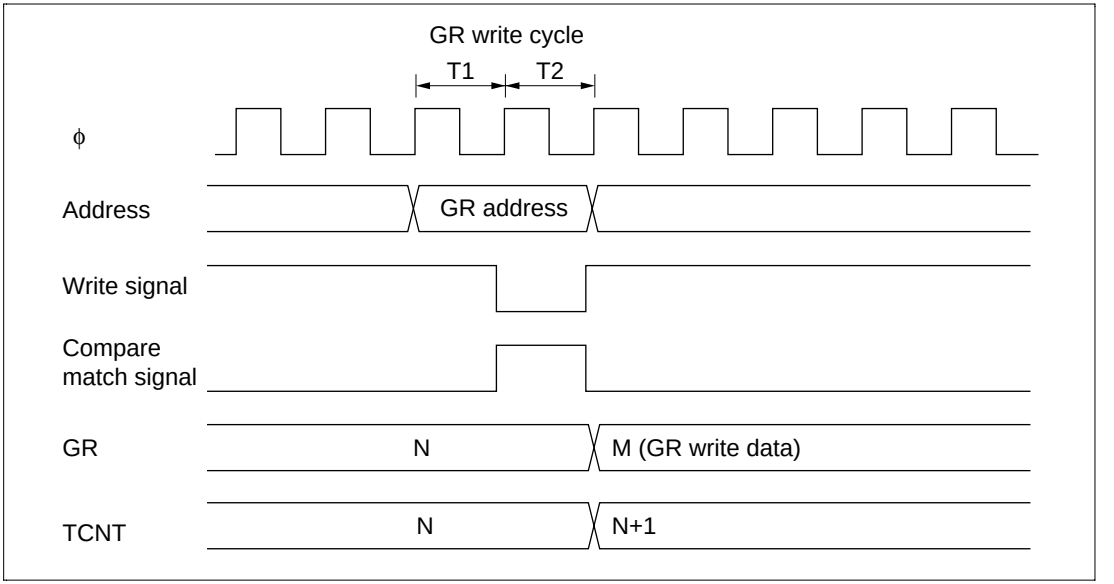


Figure 12.37 Contention between General Register Write and Compare Match

Contention between General Register Write and Input Capture: If a capturing signal is generated in the T2 state of a general register write cycle, writing to GR takes priority and input capture (data transfer from TCNT to GR) is not performed. Figure 12.38 shows this timing.

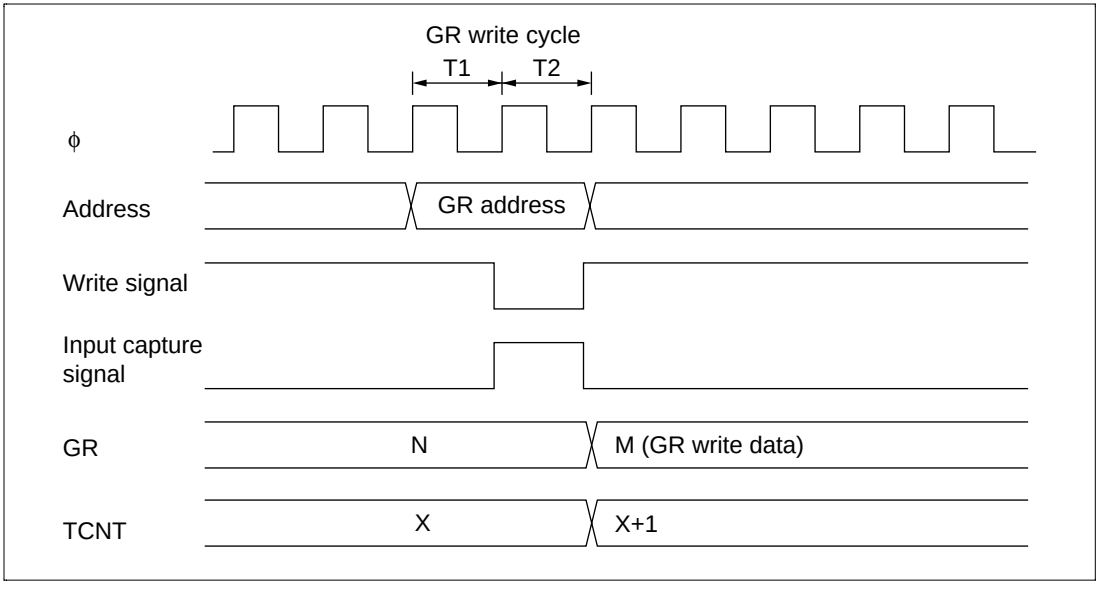


Figure 12.38 Contention between General Register Write and Input Capture

Contention between Buffer Register Write and Input Capture in Buffer Operation: If a capturing signal is generated in the T2 state of a buffer register write cycle, writing to the buffer register takes priority and input capture (data transfer from GR to the buffer register) is not performed. Figure 12.39 shows this timing.

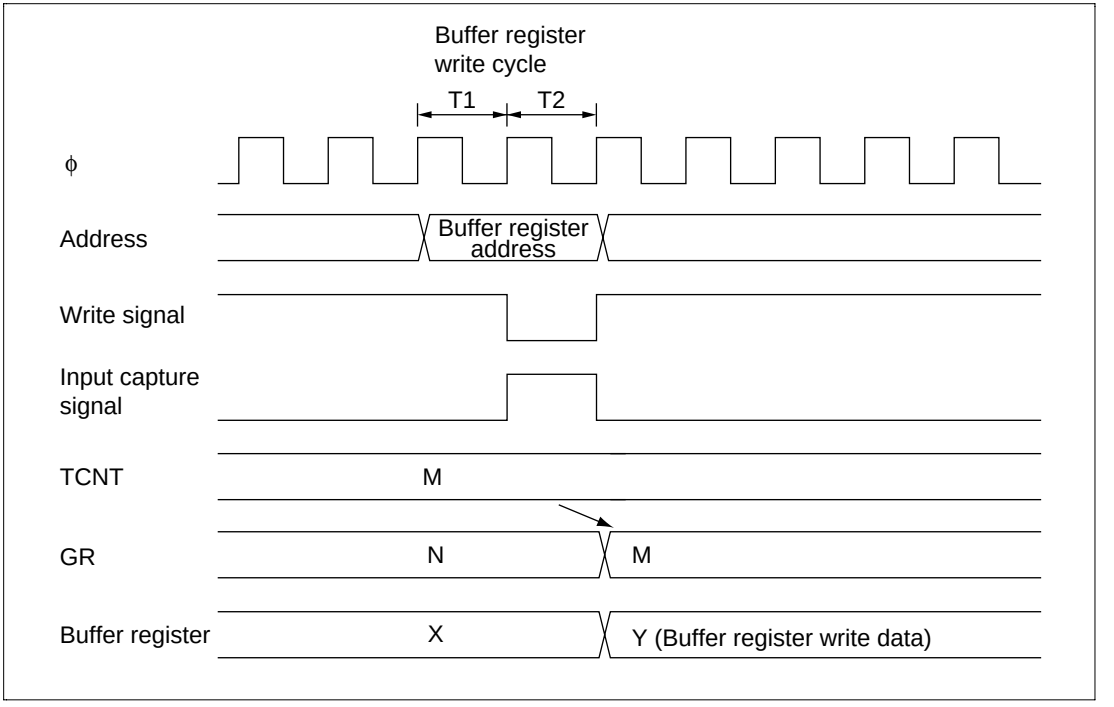


Figure 12.39 Contention between Buffer Register Write and Input Capture

Contention between General Register Read and Input Capture: If a capturing signal is generated in the T1 state of a general register read cycle, the value before input capture is read. Figure 12.40 shows this timing.

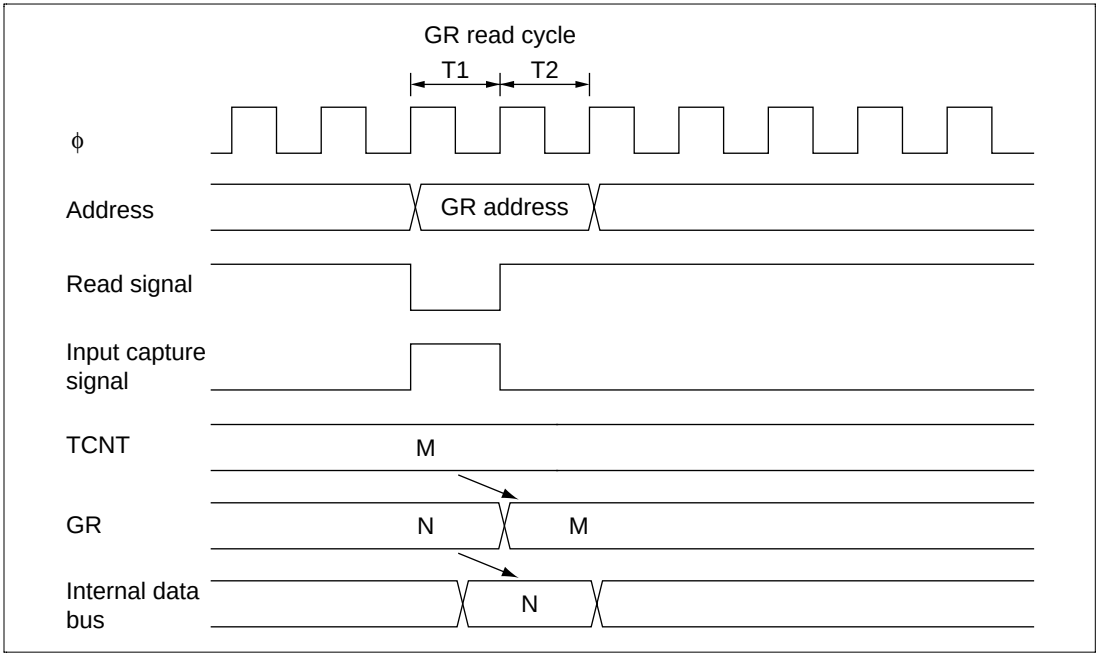


Figure 12.40 Contention between General Register Read and Input Capture

TCNT Read Timing Error: This LSI takes two states (two system clock (ϕ) cycles) to access timer W. Therefore, when ϕ is selected as the TCNT input clock and TCNT is read, the read timing will include 1/ϕ-second error.

Internal Clock Switching and TCNT Operation: When the input clock is switched between internal clocks, the TCNT may erroneously increase the count .

When an internal clock is selected, the count clock is generated by detecting the rising edges of the internal clock obtained by dividing the system clock (ϕ). If the input clock is switched when the previous clock is low and the new clock is high, the change in clock signal level (low to high) is assumed to be a clock rising edge; a count clock pulse is generated and TCNT erroneously increments the count. Figure 12.41 shows this timing.

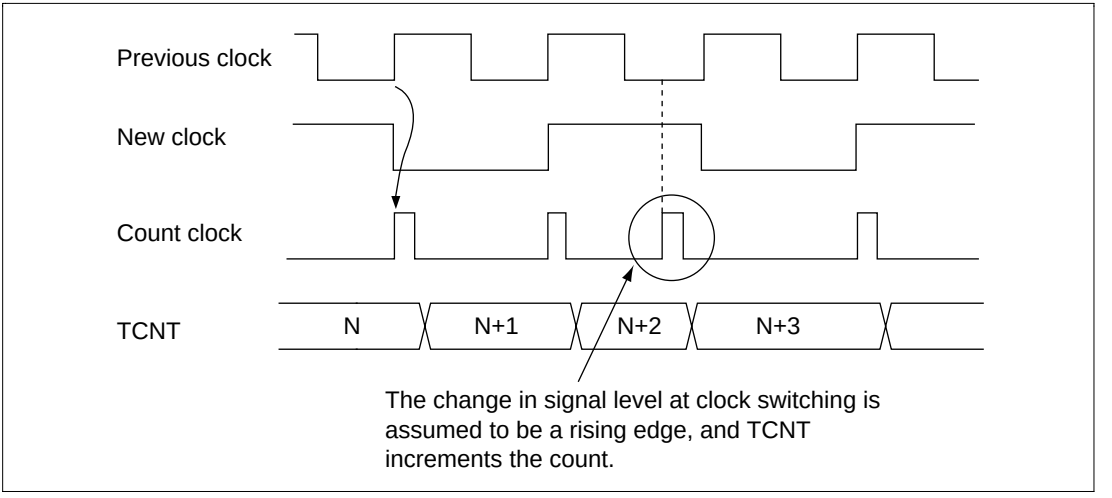


Figure 12.41 Internal Clock Switching and TCNT Operation

Interrupt in Module Stop Mode: If timer W enters module stop mode while an interrupt request is sent to the CPU, the interrupt request cannot be cleared. Before entering module stop mode, disable interrupt requests.

Section 13 Watchdog Timer

13.1 Overview

The watchdog timer (WDT) is equipped with an 8-bit counter that is incremented by an input clock. An internal chip reset can be executed if the counter overflows because it is not updated normally due to a system crash, etc.

13.1.1 Features

Features of the watchdog timer are given below.

- Nine internal clocks selectable
- Choice of eight internal clock sources (ø/64, ø/128, ø/256, ø/512, ø/1024, ø/2048, ø/4096, ø/8192) and the internal oscillator
- Reset signal generated on counter overflow
  An overflow period of 1 to 256 times the selected clock can be set.

13.1.2 Block Diagram

Figure 13.1 shows a block diagram of the watchdog timer.

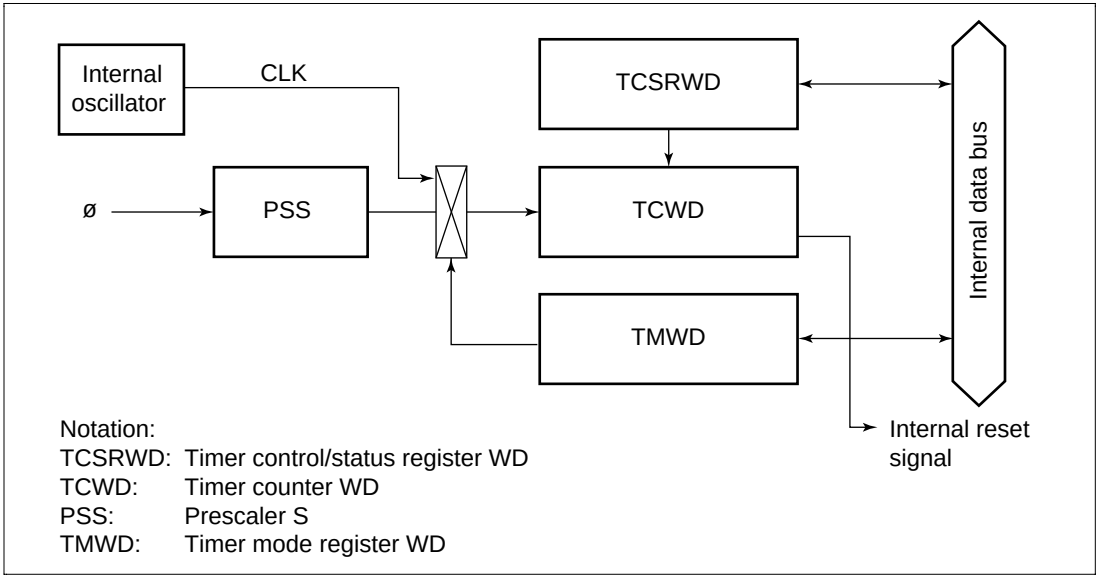


Figure 13.1 Block Diagram of Watchdog Timer

13.1.3 Register Configuration

Table 13.1 shows the watchdog timer register configuration.

Table 13.1 Watchdog Timer Registers

Name	Abbrev.	R/W	Initial Value	Address
Timer control/status register WD	TCSRWD	R/W	H'AA	H'FFC0
Timer counter WD	TCWD	R/W	H'00	H'FFC1
Timer mode register WD	TMWD	R/W	H'FF	H'FFC2

13.2 Register Descriptions

13.2.1 Timer Control/Status Register WD (TCSRWD)

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: * Can be written to only when the write condition is satisfied. For the write conditions, see the individual bit descriptions.

TCSRWD is an 8-bit read/write register that performs TCSRWD and TCWD write control and watchdog timer operation control, and indicates the operation status.

Bit 7—Bit 6 Write Inhibit (B6WI): Bit 7 controls writing of data to bit 6 of TCSRWD.

Bit 7: B6WI	Description	
0	Writing to bit 6 is enabled	
1	Writing to bit 6 is disabled	(Initial value)

This bit is always read as 1. Data is not stored if written to this bit.

Bit 6—Timer Counter WD Write Enable (TCWE): Bit 6 controls writing of 8-bit data to TCWD.

Bit 6: TCWE	Description	
0	Writing of 8-bit data to TCWD is disabled	(Initial value)
1	Writing of 8-bit data to TCWD is enabled	

Bit 5—Bit 4 Write Inhibit (B4WI): Bit 5 controls writing of data to bit 4 of TCSRWD.

Bit 5: B4WI	Description	
0	Writing to bit 4 is enabled	
1	Writing to bit 4 is disabled	(Initial value)

This bit is always read as 1. Data is not stored if written to this bit.

Bit 4—Timer Control/Status Register W Write Enable (TCSRWE): Bit 4 controls writing of data to bits 2 and 0 of TCSRWD.

Bit 4: TCSRWE	Description	
0	Writing to bits 2 and 0 is disabled	(Initial value)
1	Writing to bits 2 and 0 is enabled	

Bit 3—Bit 2 Write Inhibit (B2WI): Bit 3 controls writing of data to bit 2 of TCSRWD.

Bit 3: B2WI	Description	
0	Writing to bit 2 is enabled	
1	Writing to bit 2 is disabled	(Initial value)

This bit is always read as 1. Data is not stored if written to this bit.

Bit 2—Watchdog Timer On (WDON): Bit 2 controls watchdog timer operation.

Bit 2: WDON	Description	
0	Watchdog timer operation is disabled	(Initial value)
	[Clearing condition]	
	In a reset, or when 0 is written to WDON while writing 0 to B2WI when TCSRWE = 1	
1	Watchdog timer operation is enabled	
	[Setting condition]	
	When 1 is written to WDON while writing 0 to B2WI when TCSRWE = 1	

The count-up starts when this bit is set to 1, and stops when it is cleared to 0.

Bit 1—Bit 0 Write Inhibit (B0WI): Bit 1 controls writing of data to bit 0 of timer control/status register W.

Bit 1: B0WI	Description	
0	Writing to bit 0 is enabled	
1	Writing to bit 0 is disabled	(Initial value)

This bit is always read as 1. Data is not stored if written to this bit.

Bit 0—Watchdog Timer Reset (WRST): Bit 0 indicates that TCW has overflowed and an internal reset signal has been generated. The internal reset signal generated by the overflow resets the entire chip.

WRST is cleared by a reset via the $\overline{\text{RES}}$ pin or by a 0 write by software.

Bit 0: WRST	Description	
0	[Clearing conditions]	(Initial value)
	• Reset by $\overline{\text{RES}}$ pin	
	• When 0 is written to WRST while writing 0 to B0WI when TCSRWE = 1	
1	[Setting condition]	
	When TCW overflows and an internal reset signal is generated	

13.2.2 Timer Counter WD (TCWD)

Bit	7	6	5	4	3	2	1	0
	TCWD7	TCWD6	TCWD5	TCWD4	TCWD3	TCWD2	TCWD1	TCWD0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCWD is an 8-bit read/write up-counter that is incremented by an input internal clock. The TCWD value can be read or written by the CPU at any time.

When TCWD overflows (from H'FF to H'00), an internal reset signal is generated and WRST in TCSRWD is set to 1. Upon reset, TCWD is initialized to H'00.

13.2.3 Timer Mode Register WD (TMWD)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	CKS3	CKS2	CKS1	CKS0
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

TMWD is an 8-bit read/write register that selects the input clock.

Upon reset, TMWD is initialized to H'FF.

Bits 7 to 4—Reserved Bits: Bits 7 to 4 are reserved; they are always read as 1 and cannot be modified.

Bits 3 to 0—Clock Select 3 to 0 (CKS3 to CKS0): Bits 3 to 0 select the clock to be input to TCWD.

Bit 3: CKS3	Bit 2: CKS2	Bit 1: CKS1	Bit 0: CKS0	Description	
1	0	0	0	Internal clock: ø/64	
			1	Internal clock: ø/128	
		1	0	Internal clock: ø/256	
			1	Internal clock: ø/512	
	1	0	0	Internal clock: ø/1024	
			1	Internal clock: ø/2048	
		1	0	Internal clock: ø/4096	
			1	Internal clock: ø/8192	(Initial value)
0	*	*	*	Internal oscillator	

Note: * Don’t care

13.3 Operation

The watchdog timer is provided with an 8-bit counter that increments with the input clock. If 1 is written to WDON while writing 0 to B2WI when TCSRWE in TCSRWD is set to 1, TCWD begins counting up. (To operate the watchdog timer, two write accesses to TCSRWD is required.) When a clock pulse is input after the TCWD count value has reached H'FF, the watchdog timer overflows and an internal reset signal is generated one base clock (ø) cycle later. The internal reset signal is output for a period of 512 $ø_{osc}$ clock cycles. TCWD is a writable counter, and when a value is set in TCWD, the count-up starts from that value. An overflow period in the range of 1 to 256 input clock cycles can therefore be set, according to the TCWD value.

Figure 13.2 shows an example of watchdog timer operation.

Example: With 30-ms overflow period when ø = 4 MHz (ø/8192 selected)

$$\frac{4 \times 10^6}{8192} \times 30 \times 10^{-3} = 14.6$$

Therefore, 256 – 15 = 241 (H'F1) is set in TCW.

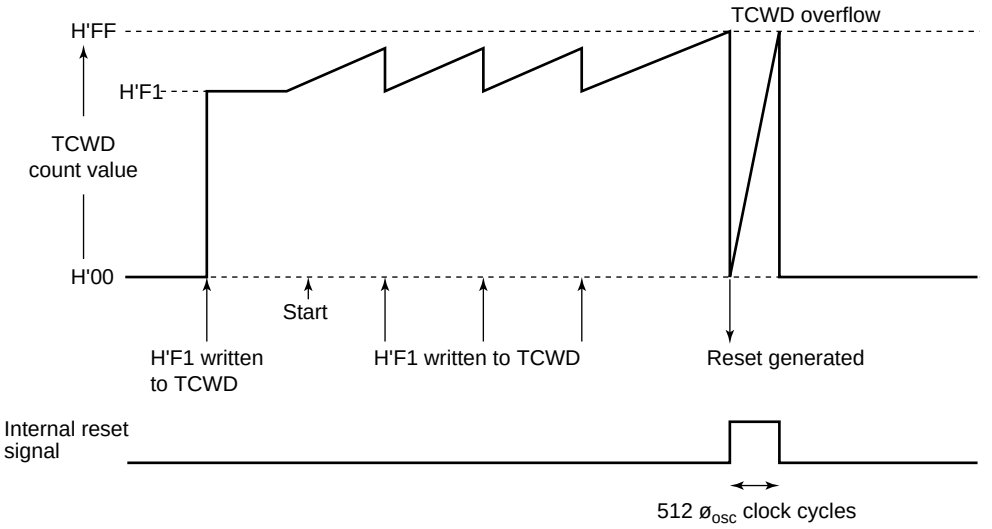


Figure 13.2 Example of Watchdog Timer Operation

13.3.1 Watchdog Timer Operating Modes

Watchdog timer operating modes are shown in table 13.2.

Table 13.2 Watchdog Timer Operating Modes

Operating mode	Reset	Active	Sleep	Subactive	Subsleep	Standby
TCWD	Reset	Functions	Functions	Halted*	Halted*	Halted*
TCSRWD	Reset	Functions	Functions	Functions	Retained	Retained
TMWD	Reset	Functions	Retained	Functions	Retained	Retained

Note: * The watchdog timer functions if the internal oscillator is selected as the clock source.

Section 14 Serial Communication Interface 3

14.1 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.

14.1.1 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

14.1.2 Block Diagram

Figure 14.1 shows a block diagram of SCI3.

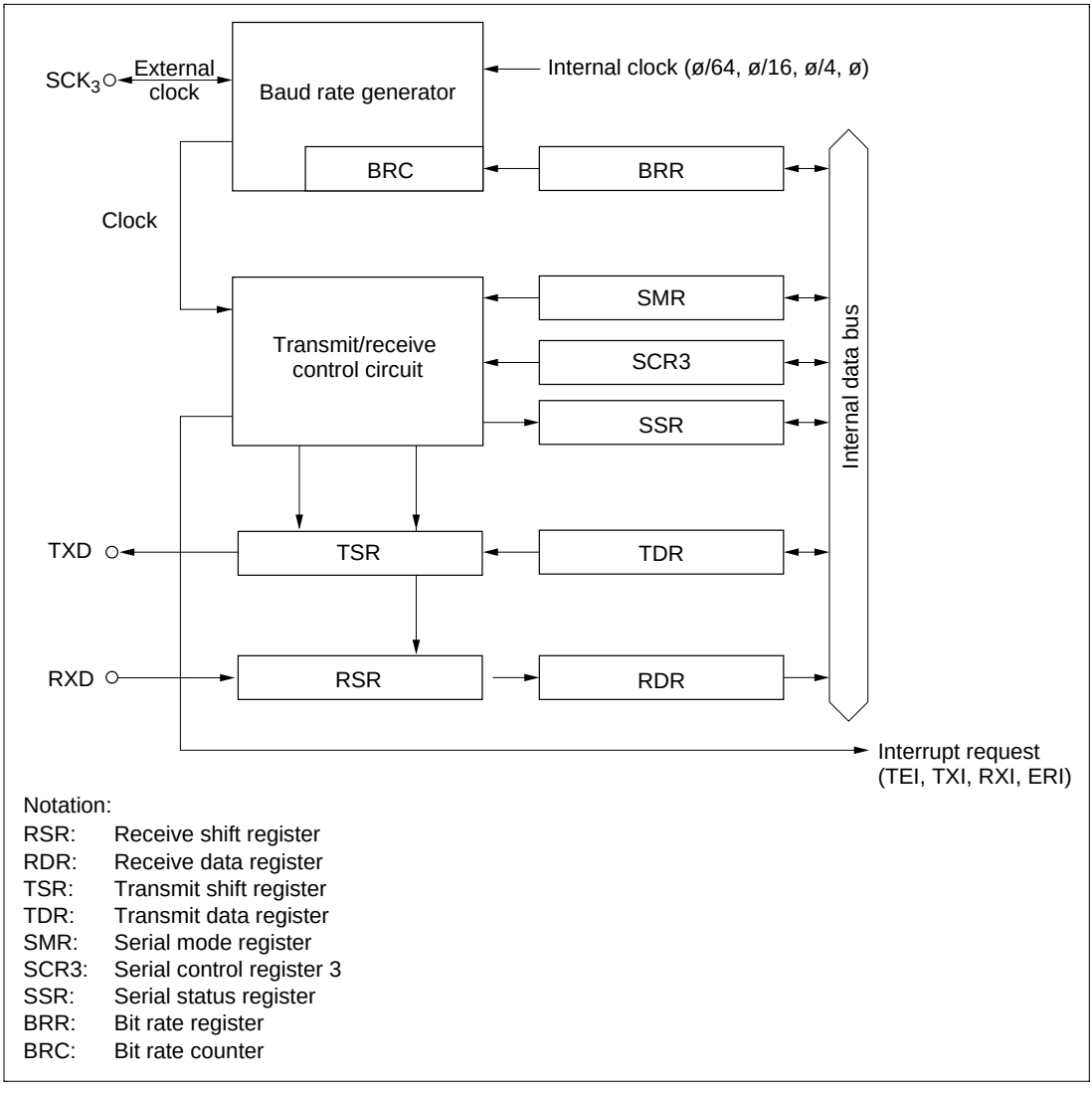


Figure 14.1 SCI3 Block Diagram

14.1.3 Pin Configuration

Table 14.1 shows the SCI3 pin configuration.

Table 14.1 Pin Configuration

Name	Abbrev.	I/O	Function
SCI3 clock	SCK3	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

14.1.4 Register Configuration

Table 14.2 shows the SCI3 register configuration.

Table 14.2 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 status 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	—	—

14.2 Register Descriptions

14.2.1 Receive Shift Register (RSR)

Bit	7	6	5	4	3	2	1	0
Read/Write	—	—	—	—	—	—	—	—

RSR is a 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.

14.2.2 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, subactive, or subsleep mode.

14.2.3 Transmit Shift Register (TSR)

Bit	7	6	5	4	3	2	1	0
Read/Write	—	—	—	—	—	—	—	—

TSR is a register used to transmit serial data. 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 from TDR to TSR, 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).

TSR cannot be read or written directly by the CPU.

14.2.4 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, subactive, or subsleep mode.

14.2.5 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, 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 1s 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 1s 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 1s 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 1s 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 in 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 enabled, 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 14.6.

Bit 2: MP	Description	
0	Multiprocessor communication function disabled	(Initial value)
1	Multiprocessor communication function enabled	

Bits 1 and 0—Clock Select 1, 0 (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 section 14.2.8, Bit Rate Register (BRR). The set value of n is in decimal notation, and represents a value of n in section 14.2.8, bit rate register (BRR).

Bit 1: CKS1	Bit 0: CKS0	Description	
0	0	ø clock (n = 0)	(Initial value)
	1	ø/4 clock (n = 1)	
1	0	ø/16 clock (n = 2)	
	1	ø/64 clock (n = 3)	

14.2.6 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, 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)*3 (Initial value)
1	Transmit operation enabled*2 (TXD pin is transmit data pin) *3

Notes: 1. Bit TDRE in SSR is fixed at 1.
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.
3. When bit TXD in PMR7 is set to 1. When bit TXD is cleared to 0, the TXD pin functions as an I/O port regardless of the TE bit setting.

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) (Initial value) [Clearing condition] When data is received in which the multiprocessor bit is set to 1
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 (SCR) 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 14.7 in 14.2.8, Operation.

		Description		
Bit 1: CKE1	**Bit 0: 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
	1	Asynchronous	Internal clock	Clock output*2
		Synchronous	Reserved	
1	0	Asynchronous	External clock	Clock input*3
		Synchronous	External clock	Serial clock input
	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.

14.2.7 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 only a write of 1 is possible 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, 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 (Initial value) [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 condition] 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	(Initial value)
	[Clearing condition]	
	After reading OER = 1, cleared by writing 0 to OER	
1	An overrun error has occurred during reception*2	
	[Setting condition]	
	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	(Initial value)
	[Clearing condition]	
	After reading FER = 1, cleared by writing 0 to FER	
1	A framing error has occurred during reception*2	
	[Setting condition]	
	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] (Initial value) [Clearing condition] After reading PER = 1, cleared by writing 0 to PER
1	A parity error has occurred during reception*[2] [Setting condition] 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 PER 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 (Initial value) [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

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	Multiprocessor bit value in transmit data is 0	(Initial value)
1	Multiprocessor bit value in transmit data is 1	

14.2.8 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, subactive, or subsleep mode.

Table 14.3 shows examples of BRR settings in asynchronous mode. The values shown are for active (high-speed) mode.

Table 14.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)

	ø (MHz)											
	2			2.097152			2.4576			3		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	141	0.03	1	148	–0.04	1	174	–0.26	1	212	0.03
150	1	103	0.16	1	108	0.21	1	127	0.00	1	155	0.16
300	0	207	0.16	0	217	0.21	0	255	0.00	1	77	0.16
600	0	103	0.16	0	108	0.21	0	127	0.00	0	155	0.16
1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	77	0.16
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38	0.16
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	19	–2.34
9600	0	6	–6.99	0	6	–2.48	0	7	0.00	0	9	–2.34
19200	0	2	8.51	0	2	13.78	0	3	0.00	0	4	–2.34
31250	0	1	0.00	0	1	4.86	0	1	22.88	0	2	0.00
38400	0	1	–18.62	0	1	–14.67	0	1	0.00	—	—	—

	ø (MHz)											
	3.6864			4			4.9152			5		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88	–0.25
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64	0.16
300	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
19200	0	5	0.00	0	6	–6.99	0	7	0.00	0	7	1.73
31250	—	—	—	0	3	0.00	0	4	–1.70	0	4	0.00
38400	0	2	0.00	0	2	8.51	0	3	0.00	0	3	1.73

	ø (MHz)											
	6			6.144			7.3728			8		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	106	–0.44	2	108	0.08	2	130	–0.07	2	141	0.03
150	2	77	0.16	2	79	0.00	2	95	0.00	2	103	0.16
300	1	155	0.16	1	159	0.00	1	191	0.00	1	207	0.16
600	1	77	0.16	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	0.00	0	5	2.40	0	6	5.33	0	7	0.00
38400	0	4	–2.34	0	4	0.00	0	5	0.00	0	6	–6.99

	ø (MHz)											
	9.8304			10			12			12.888		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

	ø (MHz)								
	14			14.7456			16		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	64	0.70	3	70	0.03
150	2	181	0.16	2	191	0.00	2	207	0.16
300	2	90	0.16	2	95	0.00	2	103	0.16
600	1	181	0.16	1	191	0.00	1	207	0.16
1200	1	90	0.16	1	95	0.00	1	103	0.16
2400	0	181	0.16	0	191	0.00	0	207	0.16
4800	0	90	0.16	0	95	0.00	0	103	0.16
9600	0	45	–0.93	0	47	0.00	0	51	0.16
19200	0	22	–0.93	0	23	0.00	0	25	0.16
31250	0	13	0.00	0	14	–1.70	0	15	0.00
38400	—	—	—	0	11	0.00	0	12	0.16

—: A setting can be made, but an error will result.

Notes: 1. The value set in BRR is given by the following equation:

$$N = \frac{ø}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

where

B: Bit rate (bit/s)
N: Baud rate generator BRR setting ($0 \leq N \leq 255$)
ø: Operating frequency (MHz)
n: Value set in bits CKS1 and CKS0 in SMR
(The relation between n and the clock is shown in table 14.4.)

Table 14.4 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

2. The bit rate error in asynchronous mode is given by the following equation:

$$\text{Error (\%)} = \left\{ \frac{\text{ø} \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 14.5 shows the maximum bit rate for each frequency. The values shown are for active (high-speed) mode.

Table 14.5 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

		Setting	
ø (MHz)	**Maximum Bit Rate (bits/s)**	**n**	**N**
2	62500	0	0
2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
5	156250	0	0
6	187500	0	0
6.144	192000	0	0
7.3728	230400	0	0
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0

Table 14.6 shows examples of BRR settings in synchronous mode. The values shown are for active (high-speed) mode.

Table 14.6 Examples of BRR Settings for Various Bit Rates (Synchronous Mode)

Bit Rate (bits/s)	OSC (MHz)									
	2		4		8		10		16	
	n	N	n	N	n	N	n	N	n	N
110	3	70	—	—	—	—	—	—	3	249
250	2	124	2	249	3	124	—	—	3	124
500	1	249	2	124	2	249	—	—	2	249
1k	1	124	1	249	2	124	—	—	2	99
2.5k	0	199	1	99	1	199	1	249	1	199
5k	0	99	0	199	1	99	1	124	1	99
10k	0	49	0	99	0	199	0	249	0	159
25k	0	19	0	39	0	79	0	99	0	79
50k	0	9	0	19	0	39	0	49	0	39
100k	0	4	0	9	0	19	0	24	0	15
250k	0	1	0	3	0	7	0	9	0	7
500k	0	0*	0	1	0	3	0	4	0	3
1M			0	0*	0	1	—	—	0	1
2M					0	0*	—	—	—	—
2.5M					—	—	0	0*	0	0*

Blank: Cannot be set.

— : A setting can be made, but an error will result.

* : Continuous transmission/reception is not possible.

Note: The value set in BRR is given by the following equation:

$$N = \frac{ø}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

where

B: Bit rate (bit/s)

N: Baud rate generator BRR setting ($0 \leq N \leq 255$)

ø: Operating frequency (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 14.7.)

Table 14.7 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

14.3 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 14.8.

The clock source for SCI3 is determined by bit COM in SMR and bits CKE1 and CKE0 in SCR3, as shown in table 14.9.

14.3.1 Asynchronous Mode

- Data length is selectable: 7 or 8 bits.
- Parity and multiprocessor bits are selectable. So is the stop bit length (1 or 2 bits). These selections determine the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, overrun errors, and the break state.
- An internal or external clock can be selected as the SCI clock source.
  - — When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and can output a serial clock signal with a frequency matching the bit rate.
  - — When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

14.3.2 Synchronous Mode

- The communication format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors.
- An internal or external clock can be selected as the SCI clock source.
  - — When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and outputs a serial clock signal to external devices.
  - — When an external clock is selected, the SCI operates on the input serial clock. The on-chip baud rate generator is not used.

Table 14.8 SMR Settings and Corresponding Data Transfer Formats

SMR Setting						Communication Format			
Bit 7: COM	Bit 6: CHR	Bit 2: MP	Bit 5: PE	Bit 3: STOP	Mode	Data Length	Multipro-cessor Bit	Parity Bit	Stop Bit Length
0	0	0	0	0	Asynchronous mode	8-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

Note: * Don't care

Table 14.9 SMR and SCR3 Settings and Clock Source Selection

SMR	SCR3			Transmit/Receive Clock	
Bit 7: COM	Bit 1: CKE1	Bit 0: CKE0	Mode	Clock Source	SCK3 Pin Function
0	0	0	Asynchronous mode	Internal	I/O port (SCK3 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			

14.3.3 Interrupts and Continuous Transmission/Reception

SCI3 can carry out continuous reception using RXI and continuous transmission using TXI. These interrupts are shown in table 14.10.

Table 14.10 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 14.2 (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 14.2 (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 14.2 (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.

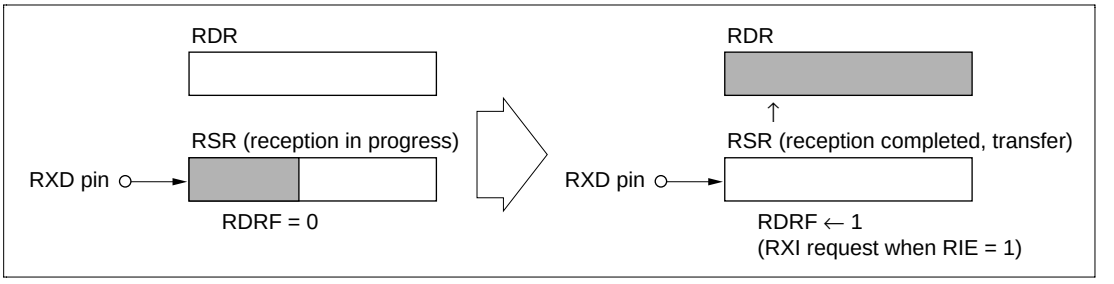


Figure 14.2 (a) RDRF Setting and RXI Interrupt

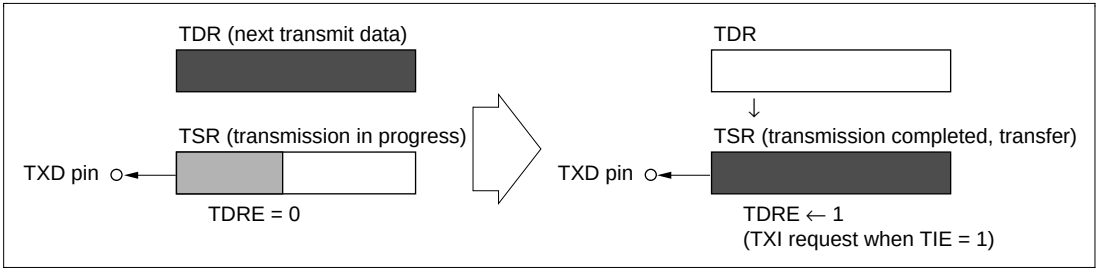


Figure 14.2 (b) TDRE Setting and TXI Interrupt

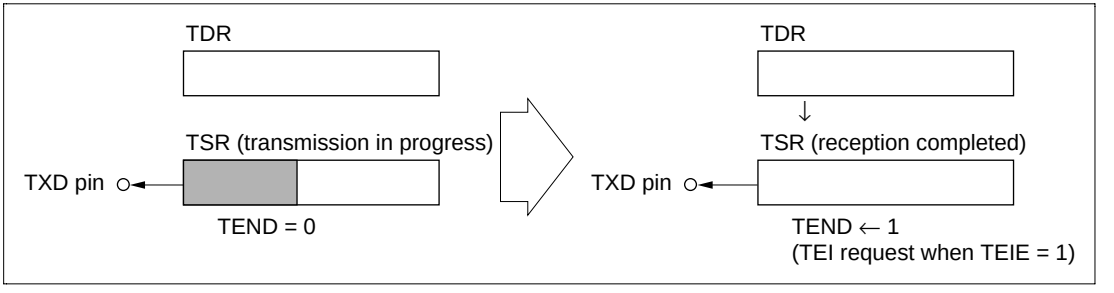


Figure 14.2 (c) TEND Setting and TEI Interrupt

14.4 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.

14.4.1 Data Transfer Format

The general data transfer format in asynchronous communication is shown in figure 14.3.

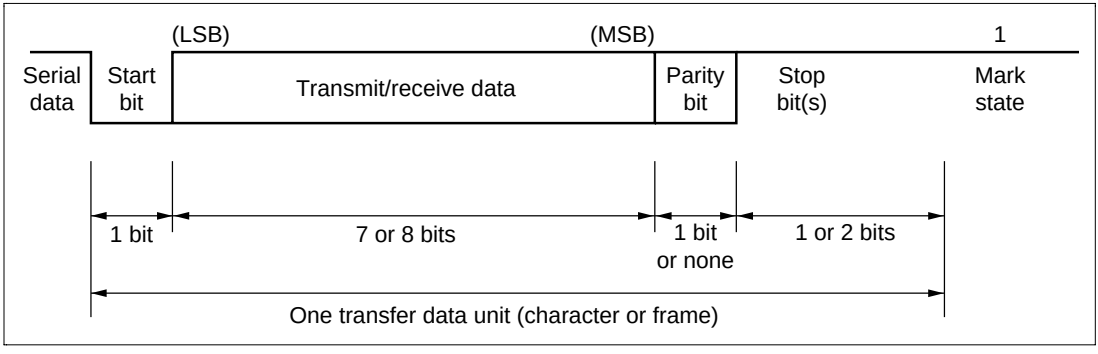


Figure 14.3 Data Format in Asynchronous Communication

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 14.11 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 14.11 Data Transfer Formats (Asynchronous Mode)

SMR Settings				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	

Note: * Don't care

Notation:

S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

14.4.2 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 CKE1 and CKE0 in SCR3. See table 14.9 for details on clock source selection.

When an external clock is input at the SCK_3 pin, a clock with a frequency of 16 times the bit rate used should be input.

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 14.4.

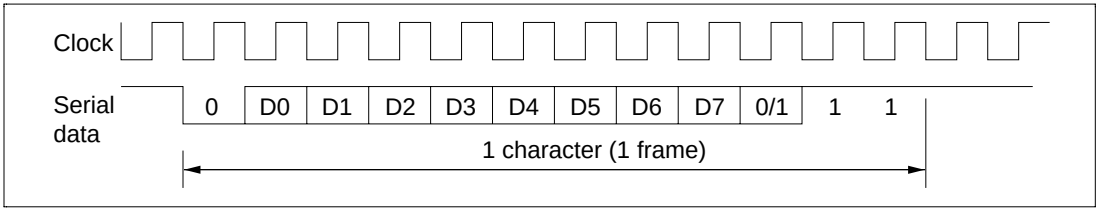


Figure 14.4 Phase Relationship between Output Clock and Transfer Data (Asynchronous Mode) (8-Bit Data, Parity, 2 Stop Bits)

14.4.3 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.

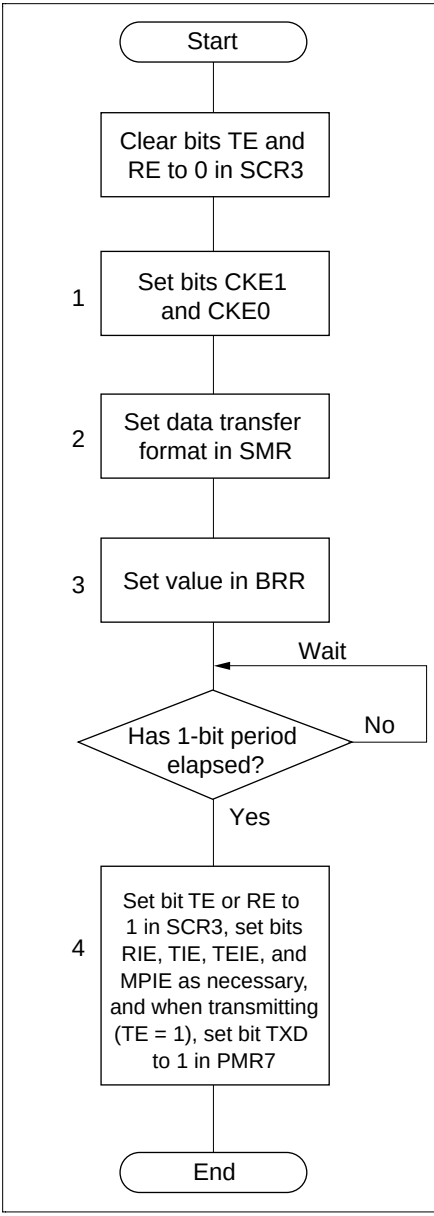
Note: If the operation mode or data transfer format is changed, bits TE and RE must first be cleared to 0.

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 14.5 shows an example of a flowchart for initializing SCI3.



1. Set clock selection in SCR3. Be sure to clear the other bits to 0. If clock output is selected in asynchronous mode, the clock is output immediately after setting bits CKE1 and CKE0. If clock output is selected for reception in synchronous mode, the clock is output immediately after bits CKE1, CKE0, and RE are set to 1.

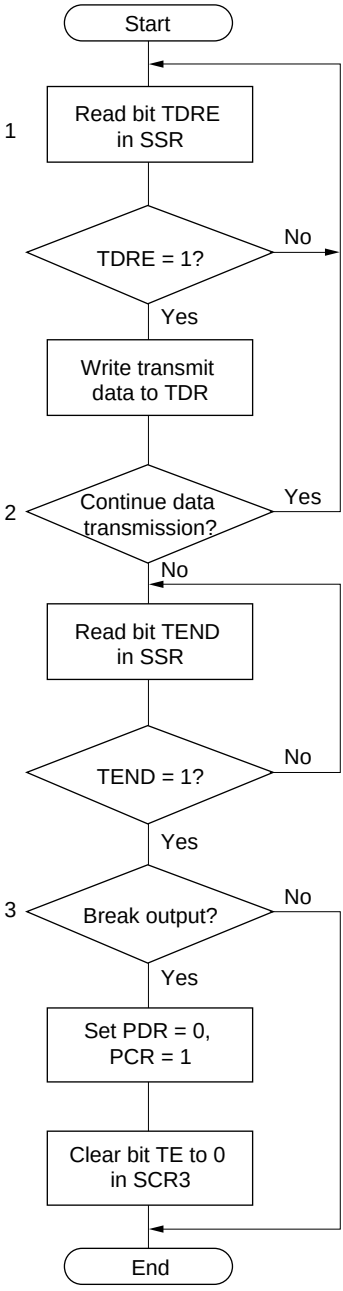
2. Set the data transfer format in the serial mode register (SMR).

3. Write the value corresponding to the transfer rate in BRR. This operation is not necessary when an external clock is selected.

4. Wait for at least the interval required to transmit or receive one bit, then set TE or RE in the serial control register (SCR3). Setting RE enables the RxD pin to be used, and when transmitting, setting bit TXD in PMR7 enables the TXD output pin to be used.
   Also set the RIE, TIE, TEIE, and MPIE bits as necessary to enable interrupts. The initial states are the mark transmit state and the idle receive state (waiting for a start bit).

Figure 14.5 Example of SCI3 Initialization Flowchart

Transmitting: Figure 14.6 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.

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 TXD in PMR7 and bit TE in SCR3 to 0.

Figure 14.6 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 14.11. 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, bit TEND in SSR is set to 1, 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 14.7 shows an example of the operation when transmitting in asynchronous mode.

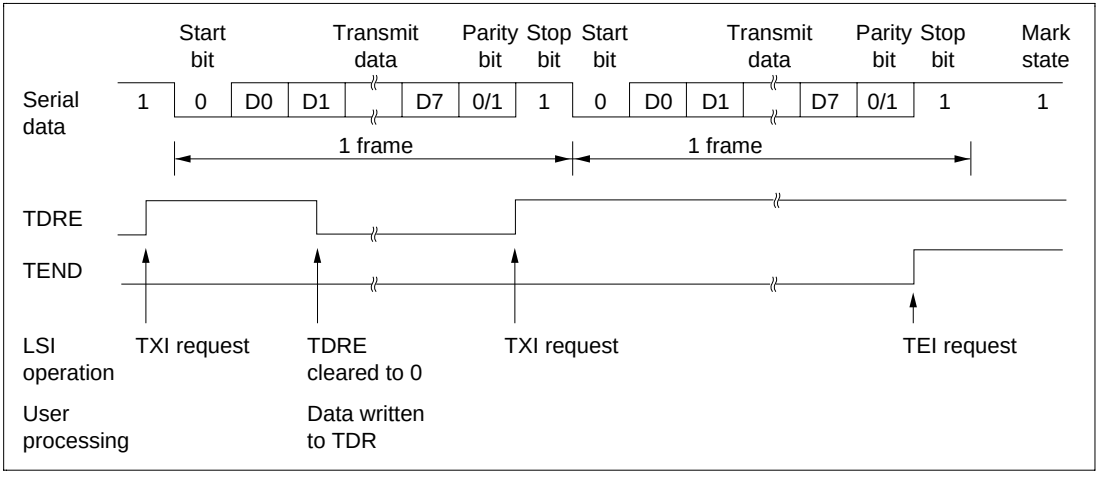


Figure 14.7 Example of Operation when Transmitting in Asynchronous Mode (8-Bit Data, Parity, 1 Stop Bit)

Receiving: Figure 14.8 shows an example of a flowchart for data reception. This procedure should be followed for data reception after initializing SCI3.

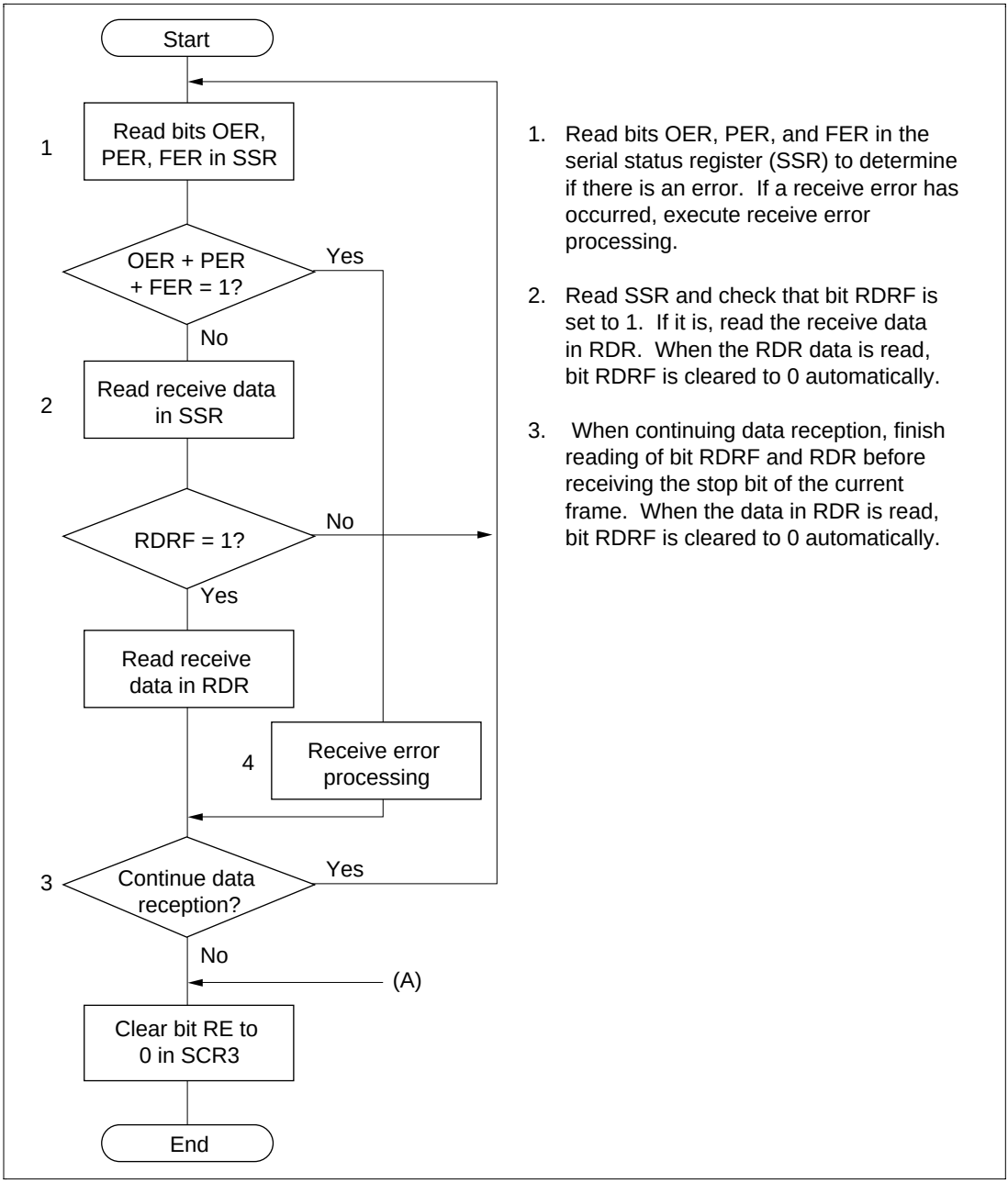
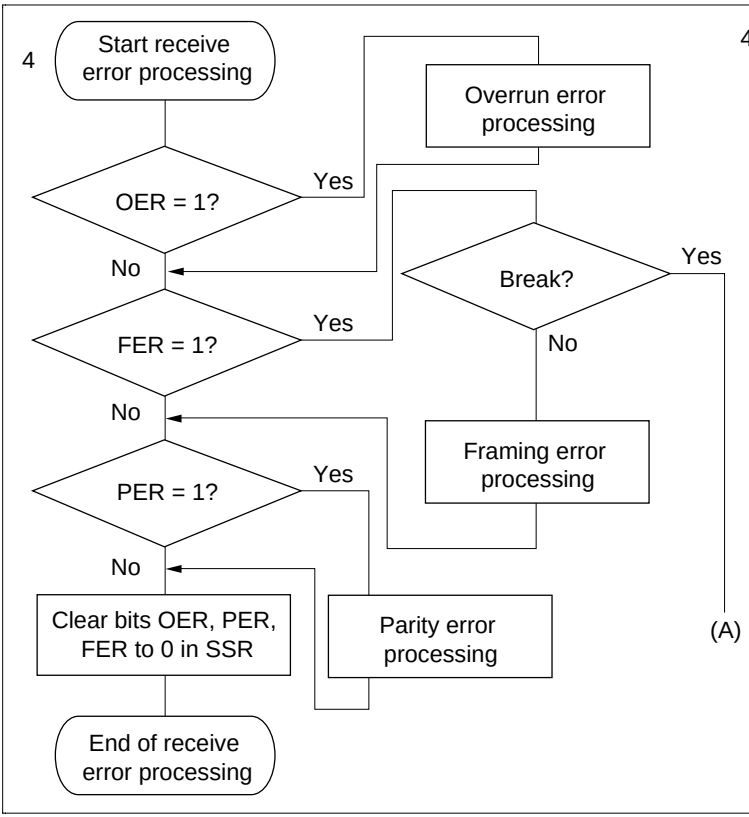


Figure 14.8 Example of Data Reception Flowchart (Asynchronous Mode)



4. If a receive error has occurred, read bits OER, PER, and FER in SSR to identify the error, and after carrying out the necessary error processing, ensure that bits OER, PER, and FER are all cleared to 0. Reception cannot be resumed if any 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 14.8 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 14.11. 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 1, 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 14.12 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 14.12 Receive Error Detection Conditions and Receive Data Processing

Receive Error	Abbreviation	Detection Conditions	Received 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 14.9 shows an example of the operation when receiving in asynchronous mode.

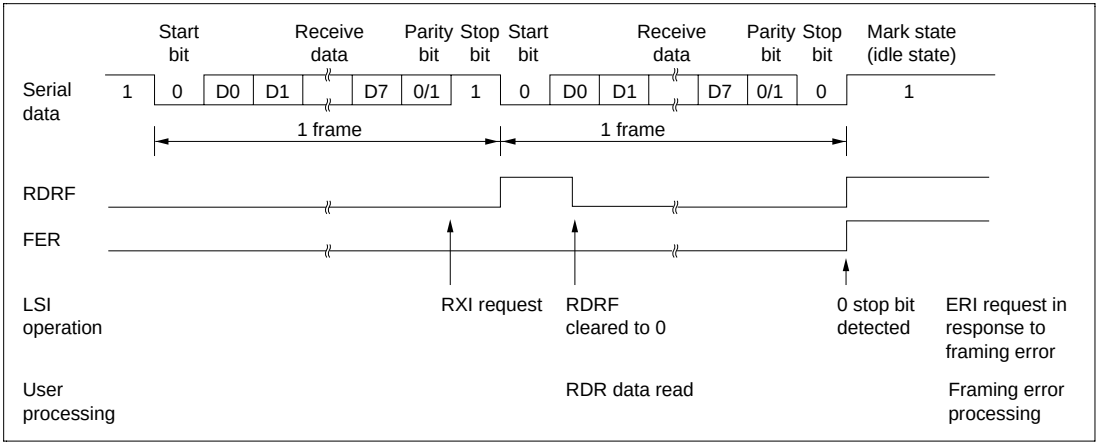


Figure 14.9 Example of Operation when Receiving in Asynchronous Mode (8-Bit Data, Parity, 1 Stop Bit)

14.5 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.

14.5.1 Data Transfer Format

The general data transfer format in synchronous communication is shown in figure 14.10.

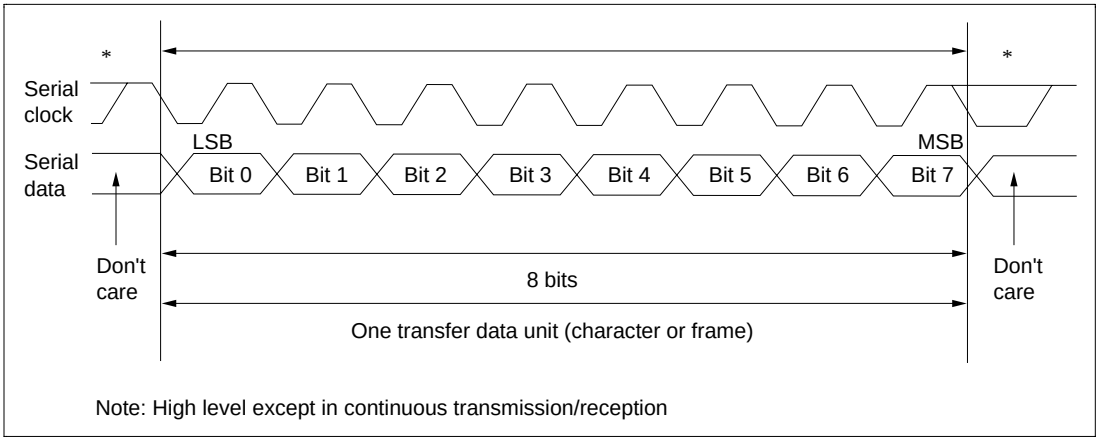


Figure 14.10 Data Format in Synchronous Communication

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.

14.5.2 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 CKE1 and CKE0 in SCR3. See table 14.12 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.

14.5.3 Data Transfer Operations

SCI3 Initialization: Data transfer on SCI3 first of all requires that SCI3 be initialized as described in 14.4.3, SCI3 Initialization, and shown in figure 14.5.

Transmitting: Figure 14.11 shows an example of a flowchart for data transmission. This procedure should be followed for data transmission after initializing SCI3.

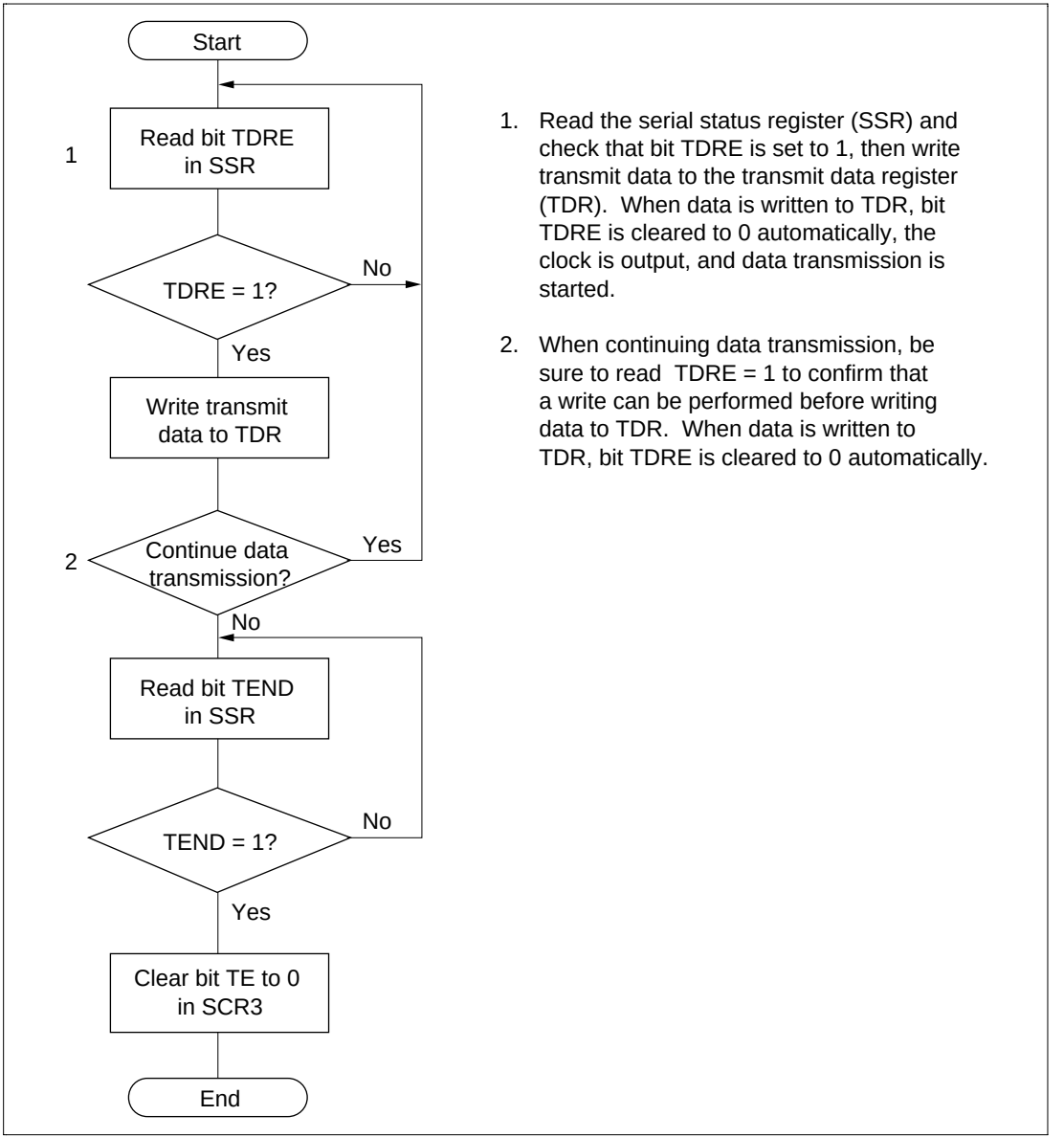


Figure 14.11 Example of Data Transmission Flowchart (Synchronous Mode)

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 (OER, FER, and PER) are all cleared to 0 before a transmit operation.

Figure 14.12 shows an example of the operation when transmitting in synchronous mode.

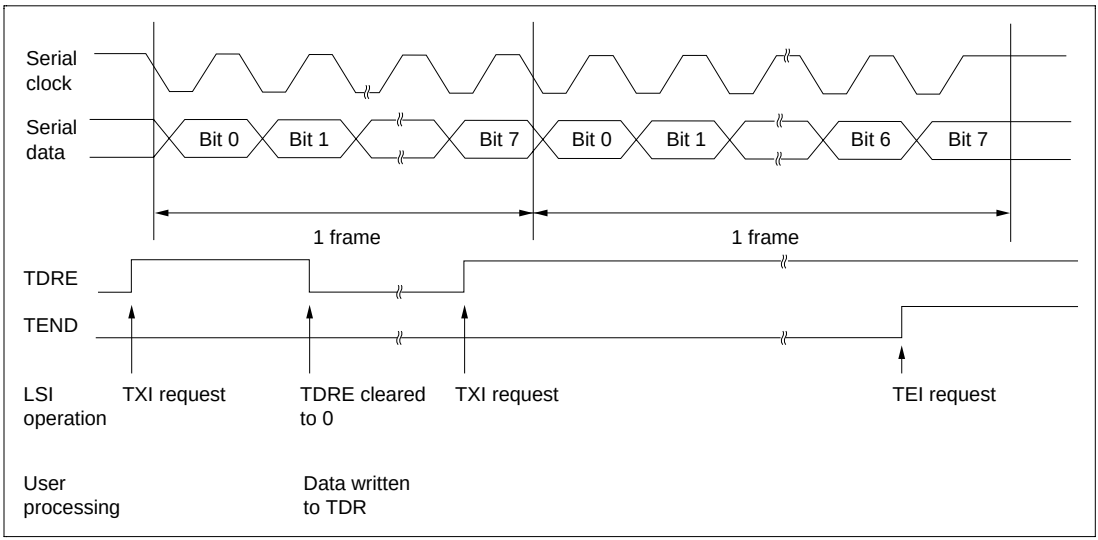


Figure 14.12 Example of Operation when Transmitting in Synchronous Mode

Receiving: Figure 14.13 shows an example of a flowchart for data reception. This procedure should be followed for data reception after initializing SCI3.

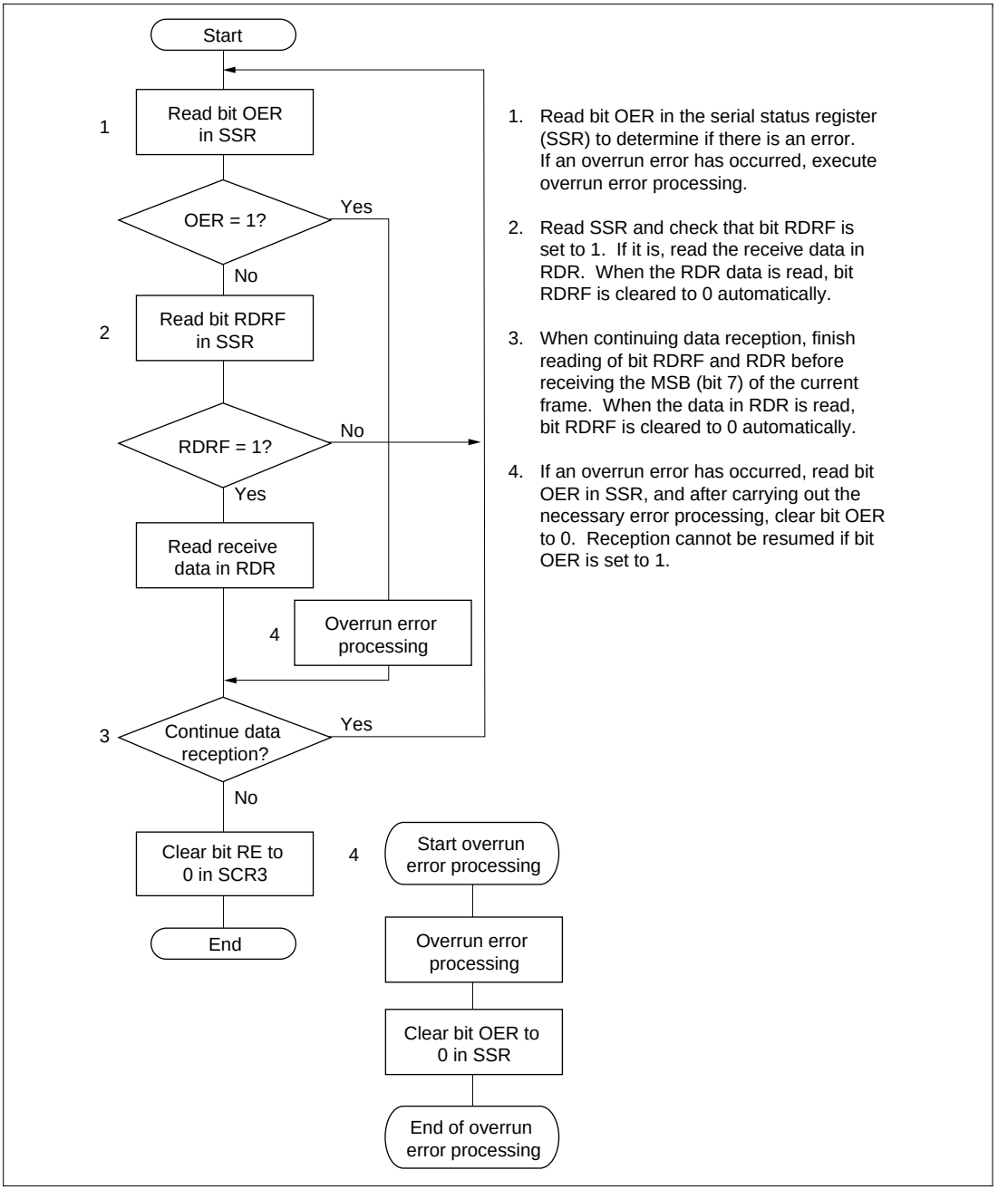


Figure 14.13 Example of Data Reception Flowchart (Synchronous Mode)

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 14.12 for the conditions for detecting an overrun 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 14.14 shows an example of the operation when receiving in synchronous mode.

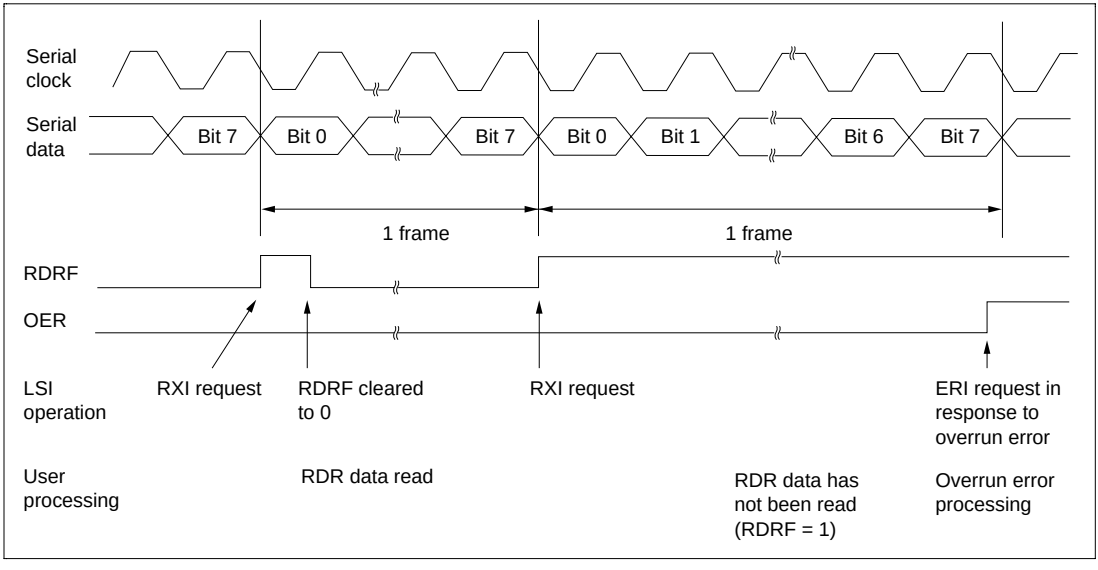
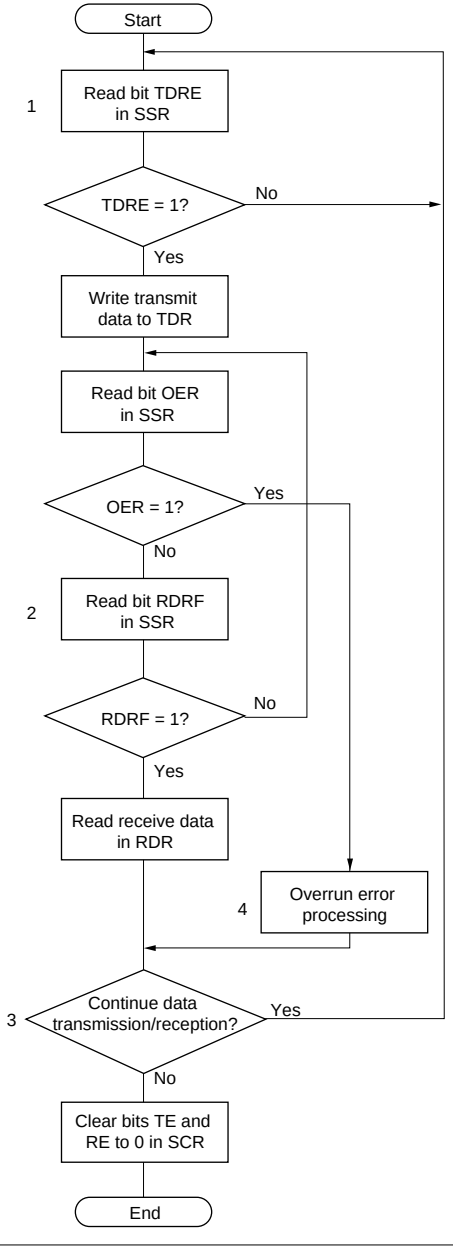


Figure 14.14 Example of Operation when Receiving in Synchronous Mode

Simultaneous Transmit/Receive: Figure 14.15 shows an example of a flowchart for a simultaneous transmit/receive operation. This procedure should be followed for simultaneous transmission/reception 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.
2. Read SSR and check that bit RDRF is set to 1. If it is, read the receive data in RDR. When the RDR data is read, bit RDRF is cleared to 0 automatically.
3. When continuing data transmission/reception, finish reading of bit RDRF and RDR before receiving the MSB (bit 7) of the current frame. Before transmitting the MSB (bit 7) of the current frame, also read TDRE = 1 to confirm that a write can be performed, then write data to TDR. When data is written to TDR, bit TDRE is cleared to 0 automatically, and when the data in RDR is read, bit RDRF is cleared to 0 automatically.
4. If an overrun error has occurred, read bit OER in SSR, and after carrying out the necessary error processing, clear bit OER to 0. Transmission and reception cannot be resumed if bit OER is set to 1. See figure 14.13 for details on overrun error processing.

Figure 14.15 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 simultaneously with a single instruction.

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 simultaneously with a single instruction.

14.6 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 14.16 shows an example of communication between processors using the multiprocessor format.

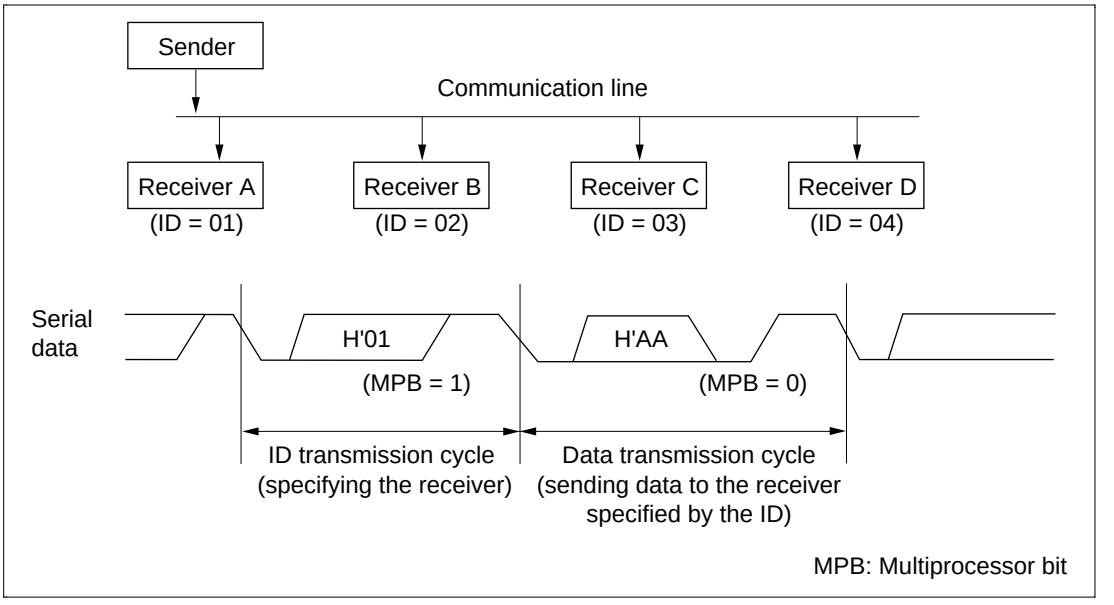
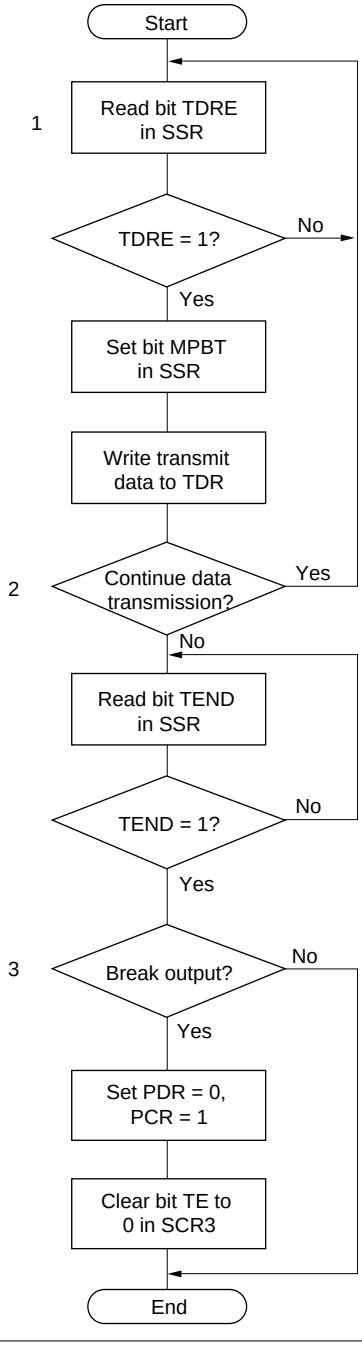


Figure 14.16 Example of Inter-Processor Communication Using Multiprocessor Format (Sending Data H'AA to Receiver A)

There is a choice of four data transfer formats. If a multiprocessor format is specified, the parity bit specification is invalid. See table 14.11 for details.

For details on the clock used in multiprocessor communication, see section 14.4, Operation in Asynchronous Mode.

Multiprocessor Transmitting: Figure 14.17 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 14.17 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 14.11. 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, bit TEND in SSR is set to 1, 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 14.18 shows an example of the operation when transmitting using the multiprocessor format.

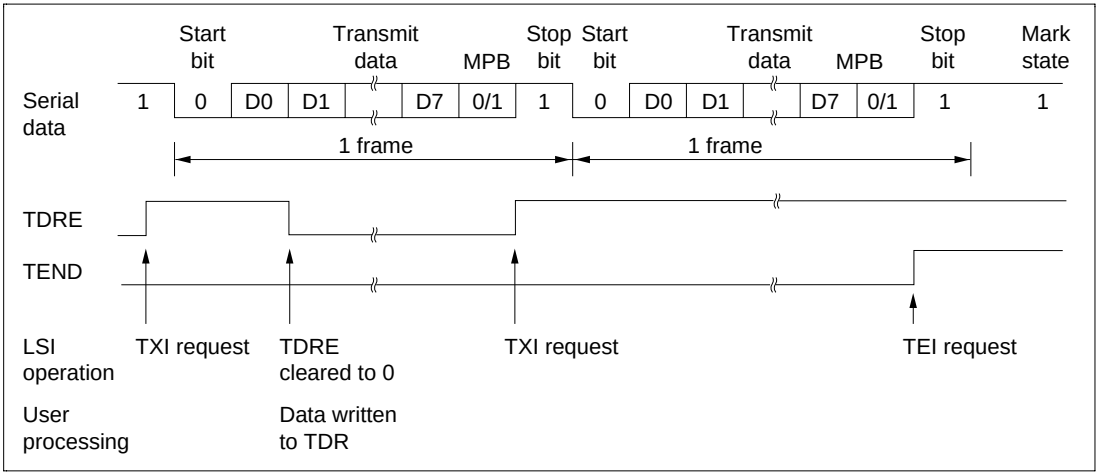
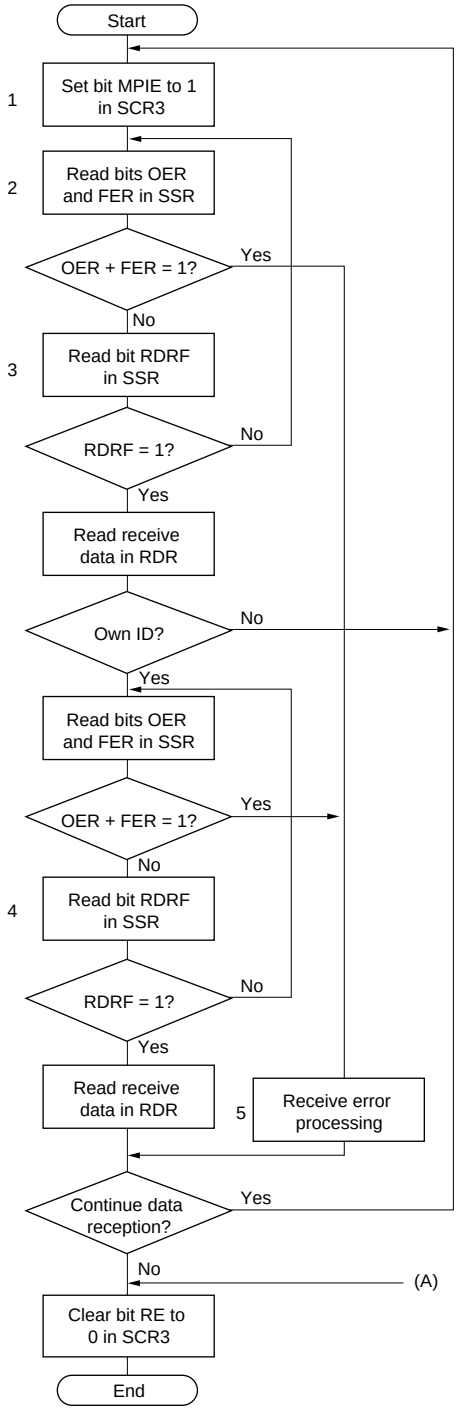


Figure 14.18 Example of Operation when Transmitting using Multiprocessor Format (8-Bit Data, Multiprocessor Bit, 1 Stop Bit)

Multiprocessor Receiving: Figure 14.19 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 14.19 Example of Multiprocessor Data Reception Flowchart

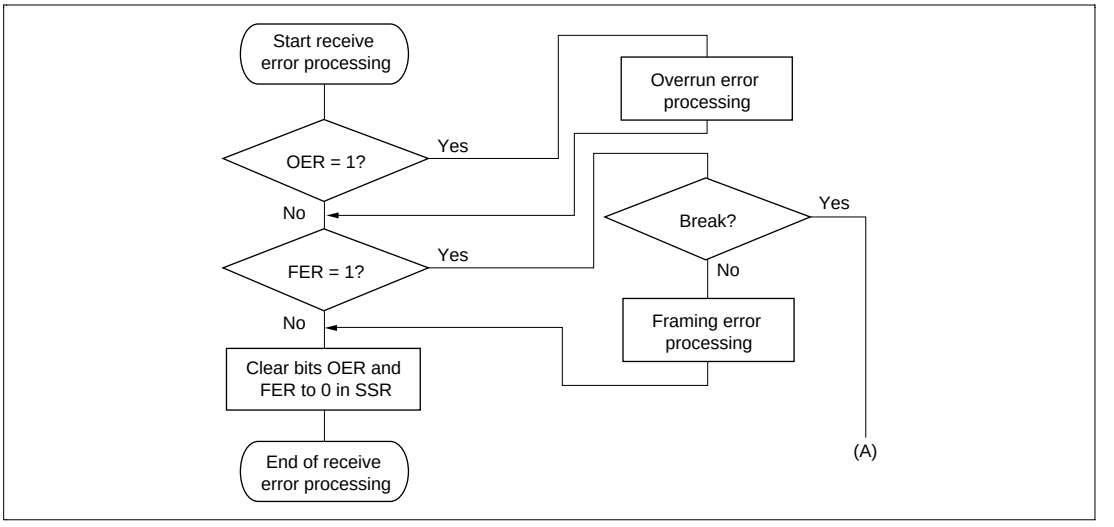
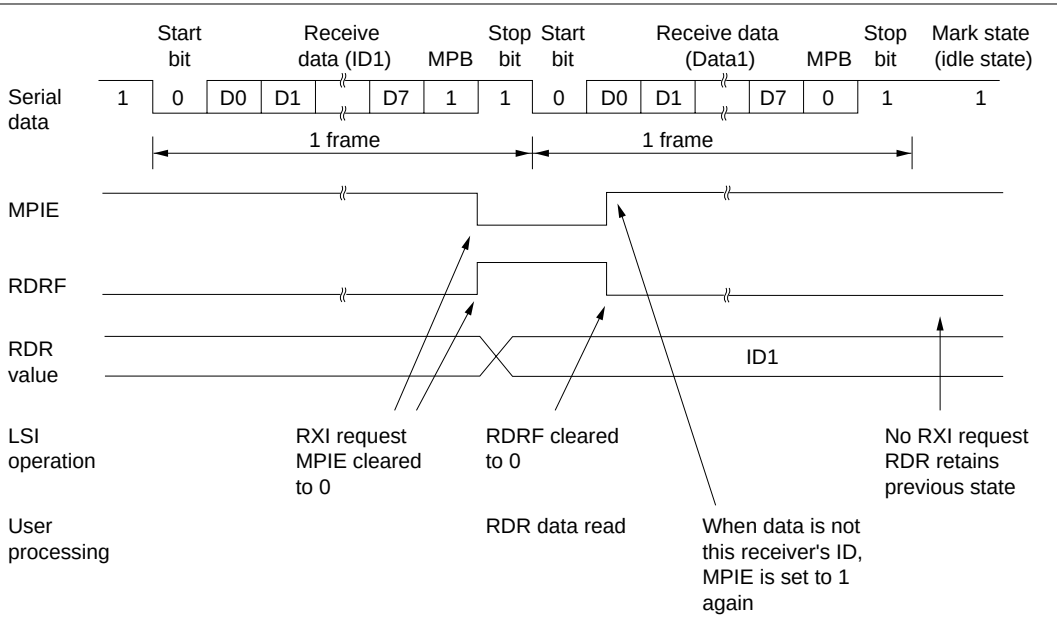
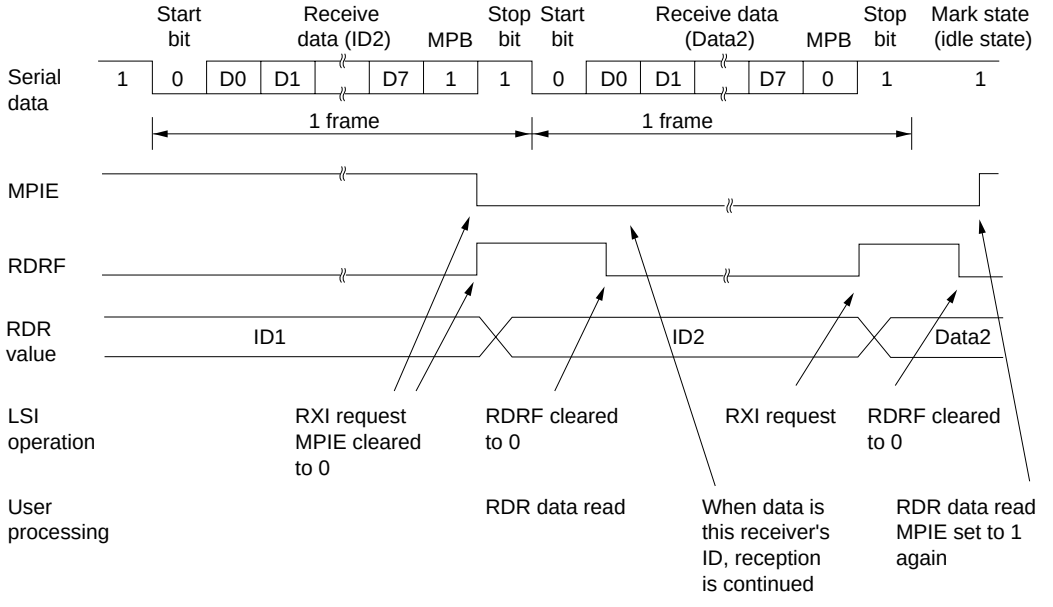


Figure 14.19 Example of Multiprocessor Data Reception Flowchart (cont)

Figure 14.20 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 14.20 Example of Operation when Receiving using Multiprocessor Format (8-Bit Data, Multiprocessor Bit, 1 Stop Bit)

14.7 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 14.13.

Table 14.13 SCI3 Interrupt Requests

Interrupt Abbreviation	Interrupt Request	Vector Address
RXI	Interrupt request initiated by receive data full flag (RDRF)	H'002A
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 section 3.3, Interrupts.

14.8 Usage Notes

The following points should be noted when using SCI3.

14.8.1 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).

14.8.2 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 14.14. If an overrun error is detected, data transfer from RSR to RDR will not be performed, and the receive data will be lost.

Table 14.14 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	O	Framing error
0	0	0	1	O	Parity error
1	1	1	0	×	Overrun error + framing error
1	1	0	1	×	Overrun error + parity error
0	0	1	1	O	Framing error + parity error
1	1	1	1	×	Overrun error + framing error + parity error

O: 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.

14.8.3 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.

14.8.4 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 during transmission, 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.

14.8.5 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.

14.8.6 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 14.21.

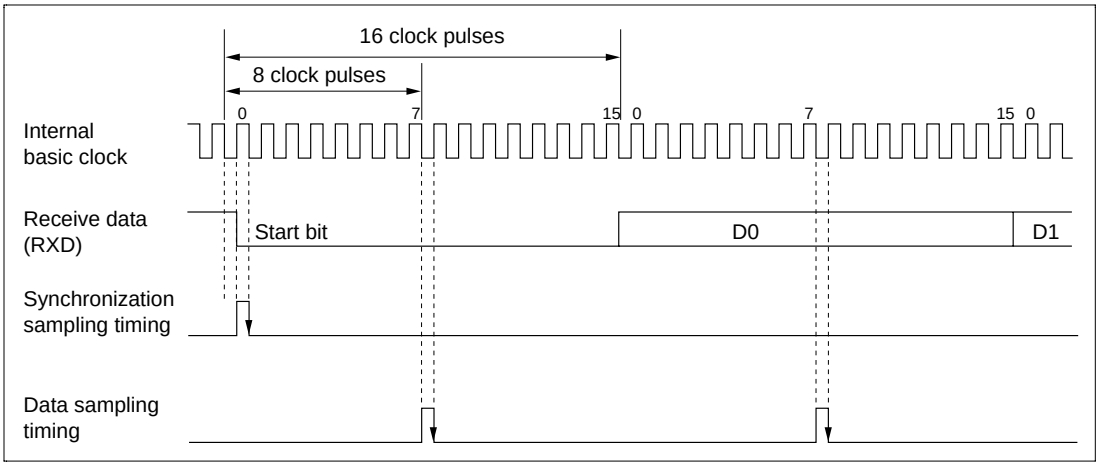


Figure 14.21 Receive Data Sampling Timing in Asynchronous Mode

Consequently, the receive margin in asynchronous mode can be expressed as shown in equation (1).

$$M = \left\{(0.5 - \frac{1}{2N}) - \frac{D - 0.5}{N} - (L - 0.5)\,F\right\} \times 100(\%) \quad \ldots\ldots\ldots\ldots \text{Equation (1)}$$

where 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,

$$M = \{0.5 - 1/(2 \times 16)\} \times 100\ [\%] = 46.875\% \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{Equation (2)}$$

However, this is only a computed value, and a margin of 20% to 30% should be allowed when carrying out system design.

14.8.7 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 14.22.

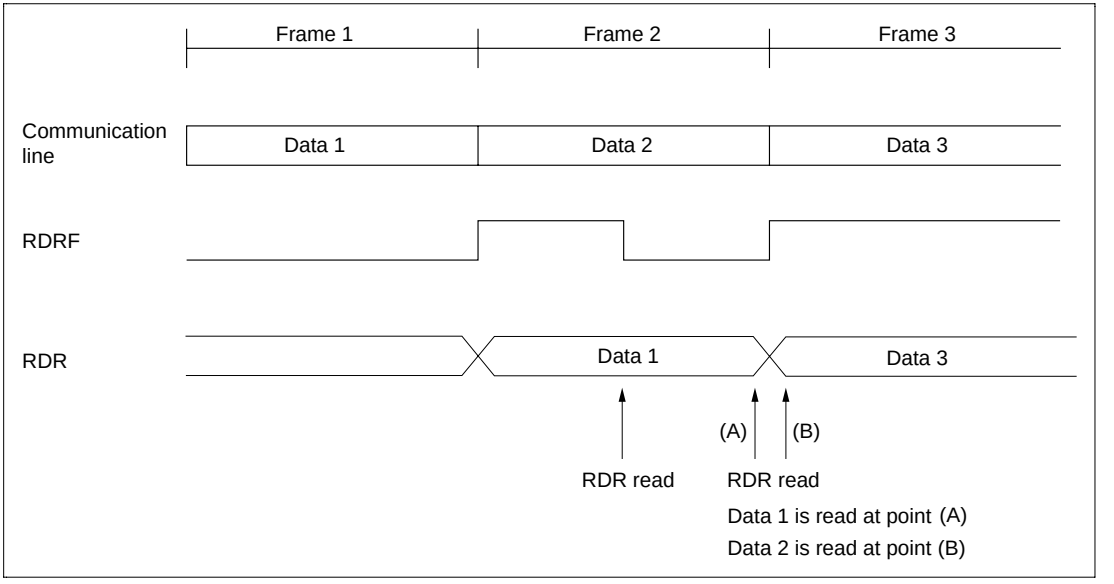


Figure 14.22 Relation between RDR Read Timing and Data

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 15 I^2C Bus Interface (IIC)

15.1 Overview

A two-channel I^2C bus interface is available for the H8/3664 Series. The I^2C bus interface conforms to and provides a subset of the Philips I^2C bus (inter-IC bus) interface functions. The register configuration that controls the I^2C bus differs partly from the Philips configuration, however.

15.1.1 Features

- Selection of I^2C format or synchronous serial format
  - — I^2C bus format: addressing format with acknowledge bit, for master/slave operation
  - — Synchronous serial format: non-addressing format without acknowledge bit, for master operation only
- Conforms to Philips I^2C bus interface (I^2C bus format)
- Two ways of setting slave address (I^2C bus format)
- Start and stop conditions generated automatically in master mode (I^2C bus format)
- Selection of acknowledge output levels when receiving (I^2C bus format)
- Automatic loading of acknowledge bit when transmitting (I^2C bus format)
- Wait function in master mode (I^2C bus format)

  A wait can be inserted by driving the SCL pin low after data transfer, excluding acknowledgement. The wait can be cleared by clearing the interrupt flag.
- Wait function in slave mode (I^2C bus format)

  A wait request can be generated by driving the SCL pin low after data transfer, excluding acknowledgement. The wait request is cleared when the next transfer becomes possible.
- Three interrupt sources
  - — Data transfer end (including transmission mode transition with I^2C bus format and address reception after loss of master arbitration)
  - — Address match: when any slave address matches or the general call address is received in slave receive mode (I^2C bus format)
  - — Stop condition detection
- Selection of 16 internal clocks (in master mode)
- Direct bus drive (with SCL and SDA pins)
  - — Two pins—(normally NMOS push-pull outputs) function as NMOS open-drain outputs when the bus drive function is selected.

15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the I²C bus interface.

Figure 15.2 shows an example of I/O pin connections to external circuits. The I/O pins are NMOS open drains. Set the upper limit of voltage applied to the power supply (V_{CC}) voltage range + 0.3 V, i.e. 5.8 V.

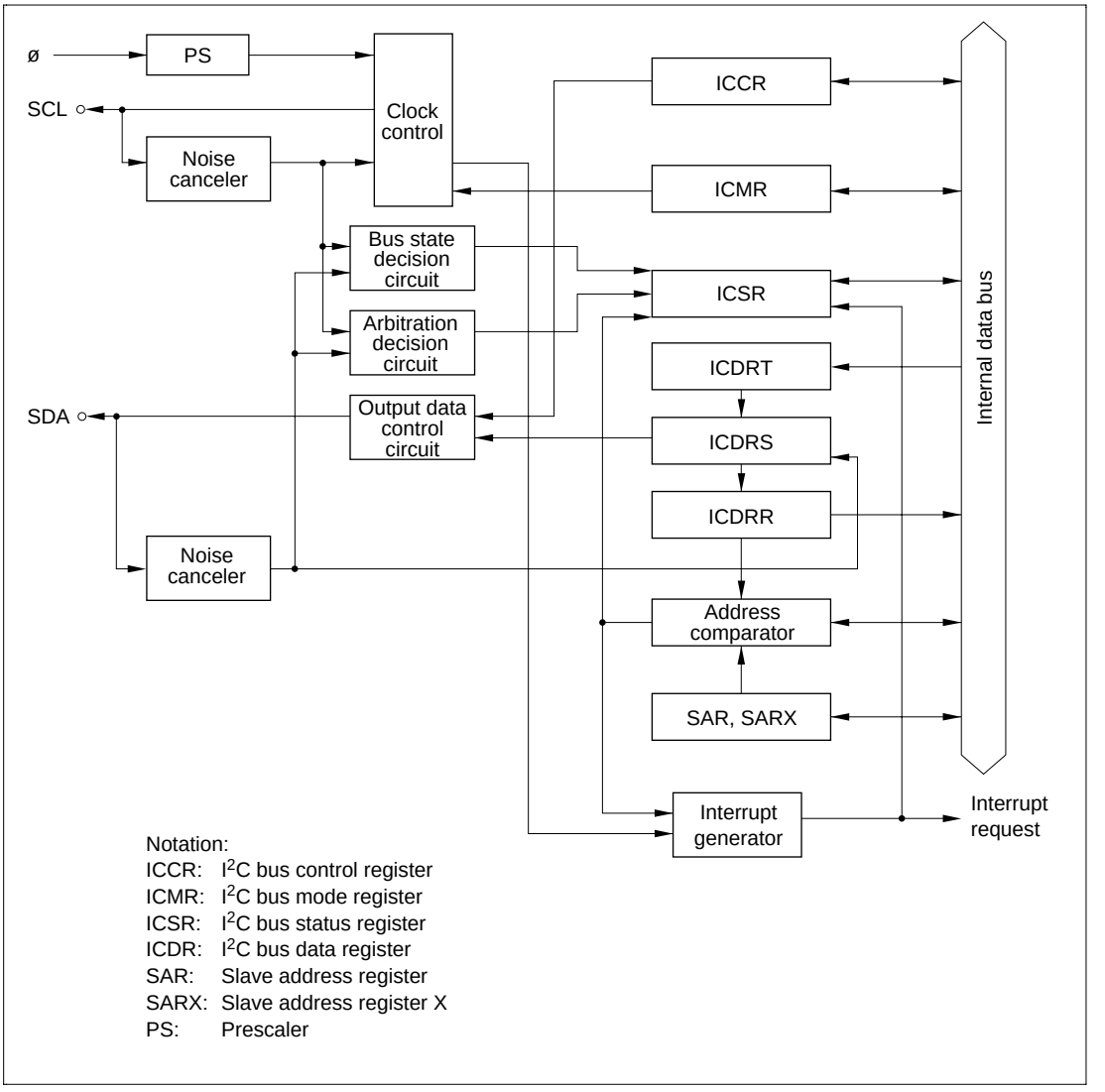


Figure 15.1 Block Diagram of I²C Bus Interface

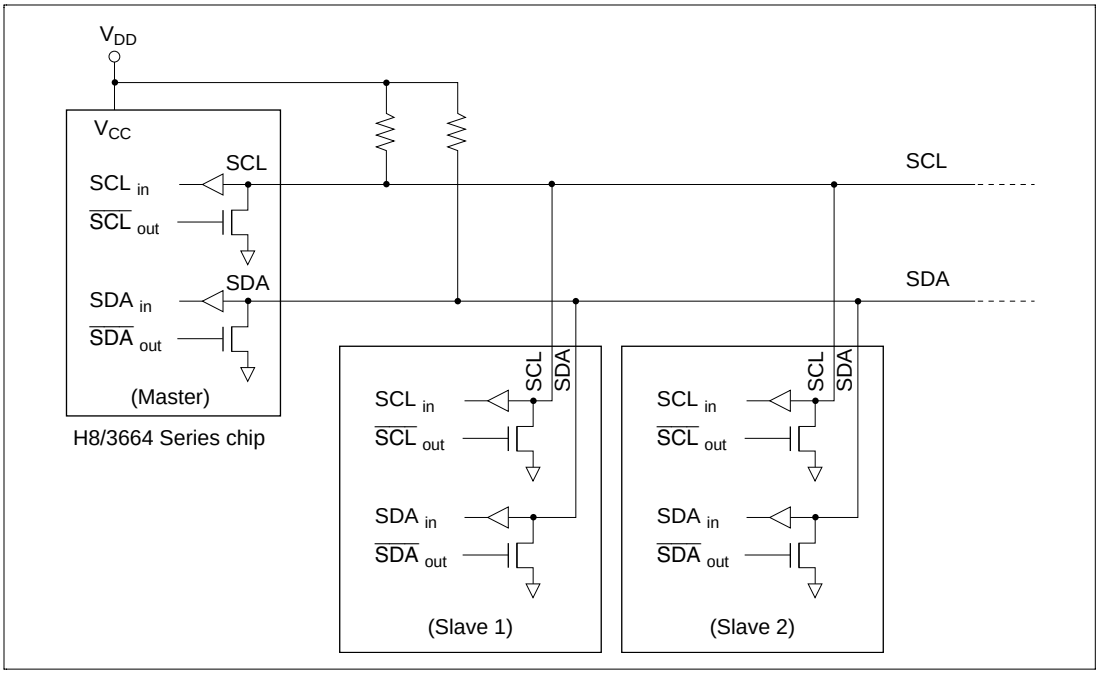


Figure 15.2 I^2C Bus Interface Connections (Example: H8/3664 Series Chip as Master)

15.1.3 Pin Configuration

Table 15.1 summarizes the input/output pins used by the I^2C bus interface.

Table 15.1 I^2C Bus Interface Pins

Name	Abbreviation	I/O	Function
Serial clock	SCL	I/O	IIC serial clock input/output
Serial data	SDA	I/O	IIC serial data input/output

15.1.4 Register Configuration

Table 15.2 summarizes the registers of the I^2C bus interface.

Table 15.2 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address
I^2C bus control register	ICCR	R/W	H'01	H'FFC4
I^2C bus status register	ICSR	R/W	H'00	H'FFC5
I^2C bus data register	ICDR	R/W	Undefined	H'FFC6*
I^2C bus mode register	ICMR	R/W	H'00	H'FFC7*
Slave address register	SAR	R/W	H'00	H'FFC7*
Second slave address register	SARX	R/W	H'01	H'FFC6*
Timer serial control register	TSCR	R/W	H'00	H'FFFC

Note: * The register that can be written or read depends on the ICE bit in the I^2C bus control register. The slave address register can be accessed when ICE = 0, and the I^2C bus mode register can be accessed when ICE = 1.

15.2 Register Descriptions

15.2.1 I²C Bus Data Register (ICDR)

Bit	7	6	5	4	3	2	1	0
	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0
Initial value	—	—	—	—	—	—	—	—
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- ICDRR

Bit	7	6	5	4	3	2	1	0
	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0
Initial value	—	—	—	—	—	—	—	—
Read/Write	R	R	R	R	R	R	R	R

- ICDRS

Bit	7	6	5	4	3	2	1	0
	ICDRS7	ICDRS6	ICDRR5	ICDRS4	ICDRS3	ICDRS2	ICDRS1	ICDRS0
Initial value	—	—	—	—	—	—	—	—
Read/Write	—	—	—	—	—	—	—	—

- ICDRT

Bit	7	6	5	4	3	2	1	0
	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0
Initial value	—	—	—	—	—	—	—	—
Read/Write	W	W	W	W	W	W	W	W

- TDRE, RDRF (internal flags)

Bit							—	—
							TDRE	RDRF
Initial value							0	0
Read/Write							—	—

ICDR is an 8-bit readable/writable register that is used as a transmit data register when transmitting and a receive data register when receiving. ICDR is divided internally into a shift register (ICDRS), receive buffer (ICDRR), and transmit buffer (ICDRT). ICDRS cannot be read or written by the CPU, ICDRR is read-only, and ICDRT is write-only. Data transfers among the three registers are performed automatically in coordination with changes in the bus state, and affect the status of internal flags such as TDRE and RDRF.

If I^2C is in transmit mode and the next data is in ICDRT (the TDRE flag is 0) following transmission/reception of one frame of data using ICDRS, data is transferred automatically from ICDRT to ICDRS. If IIC is in receive mode and no previous data remains in ICDRR (the RDRF flag is 0) following transmission/reception of one frame of data using ICDRS, data is transferred automatically from ICDRS to ICDRR.

If the number of bits in a frame, excluding the acknowledge bit, is less than 8, transmit data and receive data are stored differently. Transmit data should be written justified toward the MSB side when MLS = 0, and toward the LSB side when MLS = 1. Receive data bits read from the LSB side should be treated as valid when MLS = 0, and bits read from the MSB side when MLS = 1.

ICDR is assigned to the same address as SARX, and can be written and read only when the ICE bit is set to 1 in ICCR.

The value of ICDR is undefined after a reset.

When TDRE is 1 and the transmit buffer is empty, TDRE shows that the next transmit data can be written from the CPU. When RDRF is 1, it shows that the valid receive data is stored in the receive buffer.

The TDRE and RDRF flags are set and cleared under the conditions shown below. Setting the TDRE and RDRF flags affects the status of the interrupt flags.

TDRE	Description
0	[Clearing conditions] • When transmit data is written in ICDR (ICDRT) in transmit mode (TRS = 1) • When a stop condition is detected in the bus line state after a stop condition is issued with the I^2C bus format or serial format selected • When a stop condition is detected with the I^2C bus format selected • In receive mode (TRS = 0) (A 0 write to TRS during transfer is valid after reception of a frame containing an acknowledge bit)
1	[Setting conditions] • In transmit mode (TRS = 1), when a start condition is detected in the bus line state after a start condition is issued in master mode with the I^2C bus format or serial format selected • When data is transferred from ICDRT to ICDRS (Data transfer from ICDRT to ICDRS when TRS = 1 and TDRE = 0, and ICDRS is empty) • When a switch is made from receive mode (TRS = 0) to transmit mode (TRS = 1) after detection of a start condition

RDRF	Description
0	[Clearing condition] When ICDR (ICDRR) receive data is read in receive mode
1	[Setting condition] When data is transferred from ICDRS to ICDRR (Data transfer from ICDRS to ICDRR in case of normal termination with TRS = 0 and RDRF = 0)

15.2.2 Slave Address Register (SAR)

Bit	7	6	5	4	3	2	1	0
	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SAR is an 8-bit readable/writable register that stores the slave address and selects the communication format. When the chip is in slave mode (and the addressing format is selected), if the upper 7 bits of SAR match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device specified by the master device. SAR is assigned to the same address as ICMR, and can be written and read only when the ICE bit is cleared to 0 in ICCR.

SAR is initialized to H'00 by a reset.

Bits 7 to 1—Slave Address (SVA6 to SVA0): Set a unique address in bits SVA6 to SVA0, differing from the addresses of other slave devices connected to the I^2C bus.

Bit 0—Format Select (FS): Used together with the FSX bit in SARX to select the communication format.

The FS bit also specifies whether or not SAR slave address recognition is performed in slave mode. For details on formats, refer to section 15.3.1, I^2C Bus Data Format.

SAR	SARX	
Bit 0: FS	**Bit 0: FSX**	**Operating Mode**
0	0	I^2C bus format • SAR and SARX slave addresses recognized
	1	I^2C bus format (Initial value) • SAR slave address recognized • SARX slave address ignored
1	0	I^2C bus format • SAR slave address ignored • SARX slave address recognized
	1	Synchronous serial format • SAR and SARX slave addresses ignored

15.2.3 Second Slave Address Register (SARX)

Bit	7	6	5	4	3	2	1	0
	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SARX is an 8-bit readable/writable register that stores the second slave address and selects the communication format. When the chip is in slave mode (and the addressing format is selected), if the upper 7 bits of SARX match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device specified by the master device. SARX is assigned to the same address as ICDR, and can be written and read only when the ICE bit is cleared to 0 in ICCR.

SARX is initialized to H'01 by a reset.

Bits 7 to 1—Second Slave Address (SVAX6 to SVAX0): Set a unique address in bits SVAX6 to SVAX0, differing from the addresses of other slave devices connected to the I^2C bus.

Bit 0—Format Select X (FSX): Used together with the FS bit in SAR to select the communication format.

The FSX bit also specifies whether or not SARX slave address recognition is performed in slave mode. For details, see the description of the FS bit in SAR.

15.2.4 I^2C Bus Mode Register (ICMR)

Bit	7	6	5	4	3	2	1	0
	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ICMR is an 8-bit readable/writable register that selects whether the MSB or LSB is transferred first, performs master mode wait control, and selects the master mode transfer clock frequency and the transfer bit count. ICMR is assigned to the same address as SAR. ICMR can be written and read only when the ICE bit is set to 1 in ICCR.

ICMR is initialized to H'00 by a reset.

Bit 7—MSB-First/LSB-First Select (MLS): Selects whether data is transferred MSB-first or LSB-first.

If the number of bits in a frame, excluding the acknowledge bit, is less than 8, transmit data and receive data are stored differently. Transmit data should be written justified toward the MSB side when MLS = 0, and toward the LSB side when MLS = 1. Receive data bits read from the LSB side should be treated as valid when MLS = 0, and bits read from the MSB side when MLS = 1.

Do not set this bit to 1 when the I^2C bus format is used.

Bit 7: MLS	Description	
0	MSB-first	(Initial value)
1	LSB-first	

Bit 6—Wait Insertion Bit (WAIT): Selects whether to insert a wait between the transfer of data and the acknowledge bit, in master mode with the I^2C bus format. When WAIT is set to 1, after the fall of the clock for the final data bit, the IRIC flag is set to 1 in ICCR, and a wait state begins (with SCL at the low level). When the IRIC flag is cleared to 0 in ICCR, the wait ends and the acknowledge bit is transferred. If WAIT is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted.

The IRIC flag in ICCR is set to 1 on completion of the acknowledge bit transfer, regardless of the WAIT setting.

The setting of this bit is invalid in slave mode.

Bit 6: WAIT	Description	
0	Data and acknowledge bits transferred consecutively	(Initial value)
1	Wait inserted between data and acknowledge bits	

Bits 5 to 3—Serial Clock Select (CKS2 to CKS0): These bits, together with the IICX bit in the TSCR register, select the serial clock frequency in master mode. They should be set according to the required transfer rate.

TSCR					Transfer Rate			
Bit 0: IICX	**Bit 5: CKS2**	**Bit 4: CKS1**	**Bit 3: CKS0**	**Clock**	**ø = 5 MHz**	**ø = 8 MHz**	**ø = 10 MHz**	**ø = 16 MHz**
0	0	0	0	ø/28	179 kHz	286 kHz	357 kHz	517 kHz
			1	ø/40	125 kHz	200 kHz	250 kHz	400 kHz
		1	0	ø/48	104 kHz	167 kHz	208 kHz	333 kHz
			1	ø/64	78.1 kHz	125 kHz	156 kHz	250 kHz
	1	0	0	ø/80	62.5 kHz	100 kHz	125 kHz	200 kHz
			1	ø/100	50.0 kHz	80.0 kHz	100 kHz	160 kHz
		1	0	ø/112	44.6 kHz	71.4 kHz	89.3 kHz	143 kHz
			1	ø/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz
1	0	0	0	ø/56	89.3 kHz	143 kHz	179 kHz	286 kHz
			1	ø/80	62.5 kHz	100 kHz	125 kHz	200 kHz
		1	0	ø/96	52.1 kHz	83.3 kHz	104 kHz	167 kHz
			1	ø/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz
	1	0	0	ø/160	31.3 kHz	50.0 kHz	62.5 kHz	100 kHz
			1	ø/200	25.0 kHz	40.0 kHz	50.0 kHz	80 kHz
		1	0	ø/224	22.3 kHz	35.7 kHz	44.6 kHz	71.4 kHz
			1	ø/256	19.5 kHz	31.3 kHz	39.1 kHz	62.5 kHz

Bits 2 to 0—Bit Counter (BC2 to BC0): Bits BC2 to BC0 specify the number of bits to be transferred next. With the I²C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the data is transferred with one addition acknowledge bit. Bit BC2 to BC0 settings should be made during an interval between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL line is low.

The bit counter is initialized to 000 by a reset and when a start condition is detected. The value returns to 000 at the end of a data transfer, including the acknowledge bit.

Bit 2: BC2	Bit 1: BC1	Bit 0: BC0	Bits/Frame: Synchronous Serial Format	Bits/Frame: I²C Bus Format	
0	0	0	8	9	(Initial value)
		1	1	2	
	1	0	2	3	
		1	3	4	
1	0	0	4	5	
		1	5	6	
	1	0	6	7	
		1	7	8	

15.2.5 I²C Bus Control Register (ICCR)

Bit	7	6	5	4	3	2	1	0
	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	W

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

ICCR is an 8-bit readable/writable register that enables or disables the I²C bus interface, enables or disables interrupts, selects master or slave mode and transmission or reception, enables or disables acknowledgement, confirms the I²C bus interface bus status, issues start/stop conditions, and performs interrupt flag confirmation.

ICCR is initialized to H'01 by a reset.

Bit 7—I²C Bus Interface Enable (ICE): Selects whether or not the I²C bus interface is to be used. When ICE is set to 1, port pins function as SCL and SDA input/output pins and transfer operations are enabled. When ICE is cleared to 0, the I²C bus interface module is disabled.

The SAR and SARX registers can be accessed when ICE is 0. The ICMR and ICDR registers can be accessed when ICE is 1.

Bit 7: ICE	Description
0	I²C bus interface module disabled, with SCL and SDA signal pins set to port function (Initial value) SAR and SARX can be accessed
1	I²C bus interface module enabled for transfer operations (pins SCL and SCA are driving the bus) ICMR and ICDR can be accessed

Bit 6—I²C Bus Interface Interrupt Enable (IEIC): Enables or disables interrupts from the I²C bus interface to the CPU.

Bit 6: IEIC	Description
0	Interrupts disabled (Initial value)
1	Interrupts enabled

Bit 5—Master/Slave Select (MST)
Bit 4—Transmit/Receive Select (TRS)
MST selects whether the I²C bus interface operates in master mode or slave mode.

TRS selects whether the I²C bus interface operates in transmit mode or receive mode.

In master mode with the I²C bus format, when arbitration is lost, MST and TRS are both reset by hardware, causing a transition to slave receive mode. In slave receive mode with the addressing format (FS = 0 or FSX = 0), hardware automatically selects transmit or receive mode according to the R/W bit in the first frame after a start condition.

Modification of the TRS bit during transfer is deferred until transfer of the frame containing the acknowledge bit is completed, and the changeover is made after completion of the transfer.

MST and TRS select the operating mode as follows.

Bit 5: MST	Bit 4: TRS	Operating Mode	
0	0	Slave receive mode	(Initial value)
	1	Slave transmit mode	
1	0	Master receive mode	
	1	Master transmit mode	

Bit 5: MST	Description
0	Slave mode (Initial value) [Clearing conditions] 1. When 0 is written by software 2. When bus arbitration is lost after transmission is started in I^2C bus format master mode
1	Master mode [Setting conditions] 1. When 1 is written by software (in cases other than clearing condition 2) 2. When 1 is written in MST after reading MST = 0 (in case of clearing condition 2)

Bit 4: TRS	Description
0	Receive mode (Initial value) [Clearing conditions] 1. When 0 is written by software (in cases other than setting condition 3) 2. When 0 is written in TRS after reading TRS = 1 (in case of clearing condition 3) 3. When bus arbitration is lost after transmission is started in I^2C bus format master mode
1	Transmit mode [Setting conditions] 1. When 1 is written by software (in cases other than clearing conditions 3 and 4) 2. When 1 is written in TRS after reading TRS = 0 (in case of clearing conditions 3 and 4) 3. When a 1 is received as the R/W bit of the first frame in I^2C bus format slave mode

Bit 3—Acknowledge Bit Judgement Selection (ACKE): Specifies whether the value of the acknowledge bit returned from the receiving device when using the I^2C bus format is to be ignored and continuous transfer is performed, or transfer is to be aborted and error handling, etc., performed if the acknowledge bit is 1. When the ACKE bit is 0, the value of the received acknowledge bit is not indicated by the ACKB bit, which is always 0.

Bit 3: ACKE	Description
0	The value of the acknowledge bit is ignored, and continuous transfer is performed (Initial value)
1	If the acknowledge bit is 1, continuous transfer is interrupted

Bit 2—Bus Busy (BBSY): The BBSY flag can be read to check whether the I^2C bus (SCL, SDA) is busy or free. In master mode, this bit is also used to issue start and stop conditions.

A high-to-low transition of SDA while SCL is high is recognized as a start condition, setting BBSY to 1. A low-to-high transition of SDA while SCL is high is recognized as a stop condition, clearing BBSY to 0.

To issue a start condition, use a MOV instruction to write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, use a MOV instruction to write 0 in BBSY and 0 in SCP. It is not possible to write to BBSY in slave mode; the I^2C bus interface must be set to master transmit mode before issuing a start condition. MST and TRS should both be set to 1 before writing 1 in BBSY and 0 in SCP.

Bit 2: BBSY	Description
0	Bus is free (Initial value) [Clearing condition] When a stop condition is detected
1	Bus is busy [Setting condition] When a start condition is detected

Bit 1—I^2C Bus Interface Interrupt Request Flag (IRIC): Indicates that the I^2C bus interface has issued an interrupt request to the CPU. IRIC is set to 1 at the end of a data transfer, when a slave address or general call address is detected in slave receive mode, when bus arbitration is lost in master transmit mode, and when a stop condition is detected. IRIC is set at different times depending on the FS bit in SAR and the WAIT bit in ICMR. See section 15.3.6, IRIC Setting Timing and SCL Control. The conditions under which IRIC is set also differ depending on the setting of the ACKE bit in ICCR.

IRIC is cleared by reading IRIC after it has been set to 1, then writing 0 in IRIC.

When the DTC is used, IRIC is cleared automatically and transfer can be performed continuously without CPU intervention.

Bit 1: IRIC	Description
0	[Clearing conditions] (Initial value) 1. When 0 is written in IRIC after reading IRIC = 1 2. When ICDR is written or read by the DTC (When the TDRE or RDRF flag is cleared to 0) (This is not always a clearing condition; see the description of DTC operation for details)
1	[Setting conditions] • I^2C bus format master mode 1. When a start condition is detected in the bus line state after a start condition is issued (when the TDRE flag is set to 1 because of first frame transmission) 2. When a wait is inserted between the data and acknowledge bit when WAIT = 1 3. At the end of data transfer (when the TDRE or RDRF flag is set to 1) 4. When a slave address is received after bus arbitration is lost (when the AL flag is set to 1) 5. When 1 is received as the acknowledge bit when the ACKE bit is 1 (when the ACKB bit is set to 1) • I^2C bus format slave mode 1. When the slave address (SVA, SVAX) matches (when the AAS and AASX flags are set to 1) and at the end of data transfer up to the subsequent retransmission start condition or stop condition detection (when the TDRE or RDRF flag is set to 1) 2. When the general call address is detected (when the ADZ flag is set to 1) and at the end of data transfer up to the subsequent retransmission start condition or stop condition detection (when the TDRE or RDRF flag is set to 1) 3. When 1 is received as the acknowledge bit when the ACKE bit is 1 (when the ACKB bit is set to 1) 4. When a stop condition is detected (when the STOP or ESTP flag is set to 1) • Synchronous serial format 1. At the end of data transfer (when the TDRE or RDRF flag is set to 1) 2. When a start condition is detected with serial format selected

When, with the I^2C bus format selected, IRIC is set to 1 and an interrupt is generated, other flags must be checked in order to identify the source that set IRIC to 1. Although each source has a corresponding flag, caution is needed at the end of a transfer.

When the TDRE or RDRF internal flag is set, the readable IRTR flag may or may not be set.

Even when the IRIC flag and IRTR flag are set, the TDRE or RDRF internal flag may not be set.

Table 15.3 shows the relationship between the flags and the transfer states.

Table 15.3 Flags and Transfer States

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	State
1/0	1/0	0	0	0	0	0	0	0	0	0	Idle state (flag clearing required)
1	1	0	0	0	0	0	0	0	0	0	Start condition issuance
1	1	1	0	0	1	0	0	0	0	0	Start condition established
1	1/0	1	0	0	0	0	0	0	0	0/1	Master mode wait
1	1/0	1	0	0	1	0	0	0	0	0/1	Master mode transmit/receive end
0	0	1	0	0	0	1/0	1	1/0	1/0	0	Arbitration lost
0	0	1	0	0	0	0	0	1	0	0	SAR match by first frame in slave mode
0	0	1	0	0	0	0	0	1	1	0	General call address match
0	0	1	0	0	0	1	0	0	0	0	SARX match
0	1/0	1	0	0	0	0	0	0	0	0/1	Slave mode transmit/receive end (except after SARX match)
0	1/0	1	0	0	1	1	0	0	0	0	Slave mode transmit/receive end (after SARX match)
0	1	1	0	0	0	1	0	0	0	1	
0	1/0	0	1/0	1/0	0	0	0	0	0	0/1	Stop condition detected

Bit 0—Start Condition/Stop Condition Prohibit (SCP): Controls the issuing of start and stop conditions in master mode. To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. If 1 is written, the data is not stored.

Bit 0: SCP	Description
0	Writing 0 issues a start or stop condition, in combination with the BBSY flag
1	Reading always returns a value of 1 (Initial value) Writing is ignored

15.2.6 I^2C Bus Status Register (ICSR)

Bit	7	6	5	4	3	2	1	0
	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W

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

ICSR is an 8-bit readable/writable register that performs flag confirmation and acknowledge confirmation and control.

ICSR is initialized to H'00 by a reset.

Bit 7—Error Stop Condition Detection Flag (ESTP): Indicates that a stop condition has been detected during frame transfer in I^2C bus format slave mode.

Bit 7: ESTP	Description
0	[Clearing conditions] 1. When 0 is written in ESTP after reading ESTP = 1 2. When the IRIC flag is cleared to 0
1	[Setting conditions] • In I^2C bus format slave mode When a stop condition is detected during frame transfer • In other modes No meaning

Bit 6—Normal Stop Condition Detection Flag (STOP): Indicates that a stop condition has been detected after completion of frame transfer in I^2C bus format slave mode.

Bit 6: STOP	Description
0	[Clearing conditions] 1. When 0 is written in STOP after reading STOP = 1 2. When the IRIC flag is cleared to 0
1	[Setting conditions] • In I^2C bus format slave mode When a stop condition is detected after completion of frame transfer • In other modes No meaning

Bit 5—I^2C Bus Interface Continuous Transmission/Reception Interrupt Request Flag (IRTR): Indicates that the I^2C bus interface has issued an interrupt request to the CPU, and the source is completion of reception/transmission of one frame in continuous transmission/reception. When the IRTR flag is set to 1, the IRIC flag is also set to 1 at the same time.

IRTR flag setting is performed when the TDRE or RDRF flag is set to 1. IRTR is cleared by reading IRTR after it has been set to 1, then writing 0 in IRTR. IRTR is also cleared automatically when the IRIC flag is cleared to 0.

Bit 5: IRTR	Description
0	Waiting for transfer, or transfer in progress [Clearing conditions] 1. When 0 is written in IRTR after reading IRTR = 1 2. When the IRIC flag is cleared to 0
1	Continuous transfer state [Setting conditions] • In I^2C bus interface slave mode When the TDRE or RDRF flag is set to 1 when AASX = 1 • In other modes When the TDRE or RDRF flag is set to 1

Bit 4—Second Slave Address Recognition Flag (AASX): In I^2C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVAX6 to SVAX0 in SARX.

AASX is cleared by reading AASX after it has been set to 1, then writing 0 in AASX. AASX is also cleared automatically when a start condition is detected.

Bit 4: AASX	Description
0	[Clearing conditions] (Initial value) 1. When 0 is written in AASX after reading AASX = 1 2. When a start condition is detected 3. In master mode
1	[Setting condition] When the second slave address is detected in slave receive mode and FSX = 0

Bit 3—Arbitration Lost (AL): This flag indicates that arbitration was lost in master mode. The I^2C bus interface monitors the bus. When two or more master devices attempt to seize the bus at nearly the same time, if the I^2C bus interface detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been taken by another master.

AL is cleared by reading AL after it has been set to 1, then writing 0 in AL. In addition, AL is reset automatically by write access to ICDR in transmit mode, or read access to ICDR in receive mode.

Bit 3: AL	Description
0	[Clearing conditions] 1. When ICDR data is written (transmit mode) or read (receive mode) 2. When 0 is written in AL after reading AL = 1
1	[Setting conditions] 1. If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode 2. If the internal SCL line is high at the fall of SCL in master transmit mode

Bit 2—Slave Address Recognition Flag (AAS): In I^2C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR, or if the general call address (H'00) is detected.

AAS is cleared by reading AAS after it has been set to 1, then writing 0 in AAS. In addition, AAS is reset automatically by write access to ICDR in transmit mode, or read access to ICDR in receive mode.

Bit 2: AAS	Description
0	[Clearing conditions] 1. When ICDR data is written (transmit mode) or read (receive mode) 2. When 0 is written in AAS after reading AAS = 1 3. In master mode
1	[Setting condition] When the slave address or general call address is detected in slave receive mode and FS = 0

Bit 1—General Call Address Recognition Flag (ADZ): In I^2C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition is the general call address (H'00).

ADZ is cleared by reading ADZ after it has been set to 1, then writing 0 in ADZ. In addition, ADZ is reset automatically by write access to ICDR in transmit mode, or read access to ICDR in receive mode.

Bit 1: ADZ	Description
0	[Clearing conditions] 1. When ICDR data is written (transmit mode) or read (receive mode) 2. When 0 is written in ADZ after reading ADZ = 1 3. In master mode
1	[Setting condition] When the general call address is detected in slave receive mode and (FSX = 0 or FS = 0)

Bit 0—Acknowledge Bit (ACKB): Stores acknowledge data. In transmit mode, after the receiving device receives data, it returns acknowledge data, and this data is loaded into ACKB. In receive mode, after data has been received, the acknowledge data set in this bit is sent to the transmitting device.

When this bit is read, in transmission (when TRS = 1), the value loaded from the bus line (returned by the receiving device) is read. In reception (when TRS = 0), the value set by internal software is read.

Bit 0: ACKB	Description
0	Receive mode: 0 is output at acknowledge output timing (Initial value) Transmit mode: Indicates that the receiving device has acknowledged the data (signal is 0)
1	Receive mode: 1 is output at acknowledge output timing Transmit mode: Indicates that the receiving device has not acknowledged the data (signal is 1)

15.2.7 Timer Serial Control Register (TSCR)

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

TSCR is an 8-bit readable/writable register that controls, the I^2C interface operating mode.

TSCR is initialized to H'FC by a reset.

Bits 7 to 2—Reserved: Reserved bits.

Bit 1—I^2C Control Unit Reset (IICRST): Resets the control unit except for the I^2C registers. When a hang up occurs due to illegal communication during I^2C operation, setting IICRST to 1 can set a port or reset the I^2C control unit without initializing rregisters.

Bit 0—I^2C Transfer Rate Select (IICX): This bit, together with bits CKS2 to CKS0 in ICMR, selects the transfer rate in master mode. For details, see section 15.2.4, I^2C Bus Mode Register (ICMR).

15.3 Operation

15.3.1 I²C Bus Data Format

The I²C bus interface has serial and I²C bus formats.

The I²C bus formats are addressing formats and an acknowledge bit is inserted. These are shown in figures 15.3 (a) and (b). The first frame following a start condition always consists of 8 bits.

The serial format is a non-addressing format with no acknowledge bit. This is shown in figure 15.4.

Figure 15.5 shows the I²C bus timing.

The symbols used in figures 15.3 to 15.5 are explained in table 15.4.

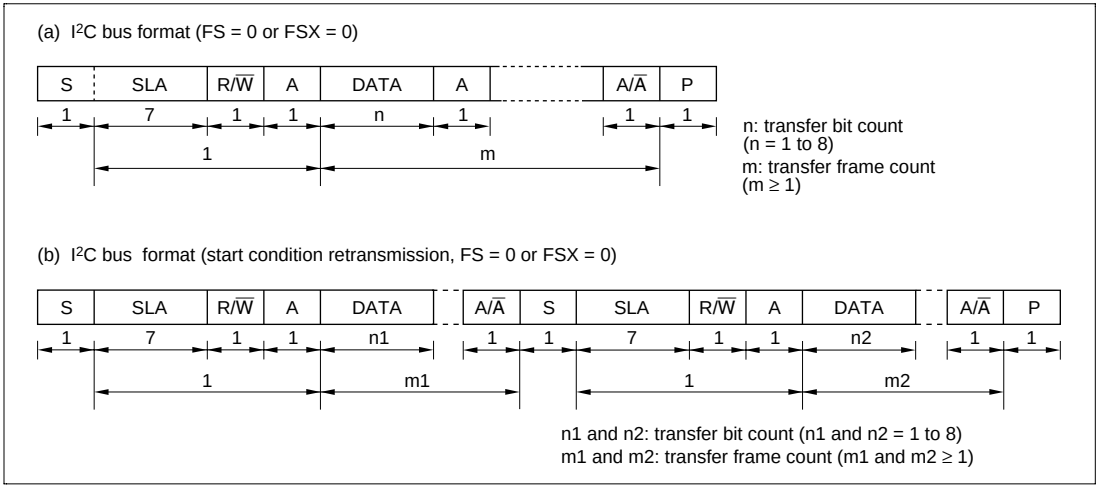


Figure 15.3 I²C Bus Data Formats (I²C Bus Formats)

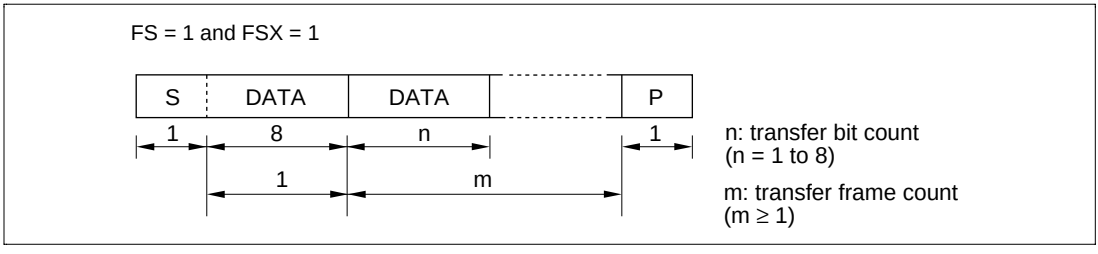


Figure 15.4 I²C Bus Data Format (Serial Format)

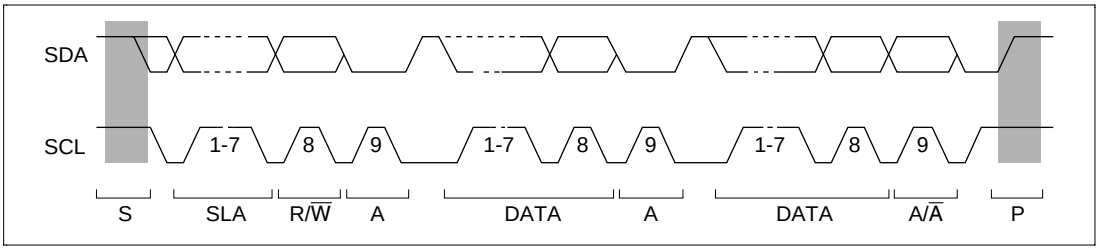


Figure 15.5 I²C Bus Timing

Table 15.4 I²C Bus Data Format Symbols

S	Start condition. The master device drives SDA from high to low while SCL is high
SLA	Slave address, by which the master device selects a slave device
R/$\overline{W}$	Indicates the direction of data transfer: from the slave device to the master device when R/$\overline{W}$ is 1, or from the master device to the slave device when R/$\overline{W}$ is 0
A	Acknowledge. The receiving device (the slave in master transmit mode, or the master in master receive mode) drives SDA low to acknowledge a transfer
DATA	Transferred data. The bit length is set by bits BC2 to BC0 in ICMR. The MSB-first or LSB-first format is selected by bit MLS in ICMR
P	Stop condition. The master device drives SDA from low to high while SCL is high

15.3.2 Master Transmit Operation

In I²C bus format master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. The transmission procedure and operations synchronize with the ICDR writing are described below.

[1] Set the ICE bit in ICCR to 1. Set bits MLS, WAIT, and CKS2 to CKS0 in ICMR, and bit IICX in TSCR, according to the operating mode.

[2] Read the BBSY flag in ICCR to confirm that the bus is free.

[3] Set bits MST and TRS to 1 in ICCR to select master transmit mode.

[4] Write 1 to BBSY and 0 to SCP. This changes SDA from high to low when SCL is high, and generates the start condition.

[5] Then IRIC and IRTR flags are set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU.

[6] Write the data (slave address + R/W) to ICDR. With the I²C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the first frame data following the start condition indicates the 7-bit slave address and transmit/receive direction. As indicating the end of the transfer, and so the IRIC flag is cleared to 0. After writing ICDR, clear IRIC immediately not to execute other interrupt handling routine. If one frame of data has been transmitted before the IRIC clearing, it can not be determine the end of transmission. The master device

sequentially sends the transmission clock and the data written to ICDR using the timing shown in figure 3.1. The selected slave device (i.e. the slave device with the matching slave address) drives SDA low at the 9th transmit clock pulse and returns an acknowledge signal.

[7] When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of the 9th transmit clock pulse. After one frame has been transmitted SCL is automatically fixed low in synchronization with the internal clock until the next transmit data is written.

[8] Read the ACKB bit in ICSR to confirm that ACKB is cleared to 0. When the slave device has not acknowledged (ACKB bit is 1), operate the step [12] to end transmission, and retry the transmit operation.

[9] Write the transmit data to ICDR. As indicating the end of the transfer, and so the IRIC flag is cleared to 0. After writing ICDR, clear IRIC immediately not to execute other interrupt handling routine. The master device sequentially sends the transmission clock and the data written to ICDR. Transmission of the next frame is performed in synchronization with the internal clock.

[10] When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of the 9th transmit clock pulse. After one frame has been transmitted SCL is automatically fixed low in synchronization with the internal clock until the next transmit data is written.

[11] Read the ACKB bit in ICSR and confirm ACKB is cleared to 0. When there is data to be transmitted, go to the step [9] to continue next transmission. When the slave device has not acknowledged (ACKB bit is set to 1), operate the step [12] to end transmission.

[12] Clear the IRIC flag to 0. And write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.

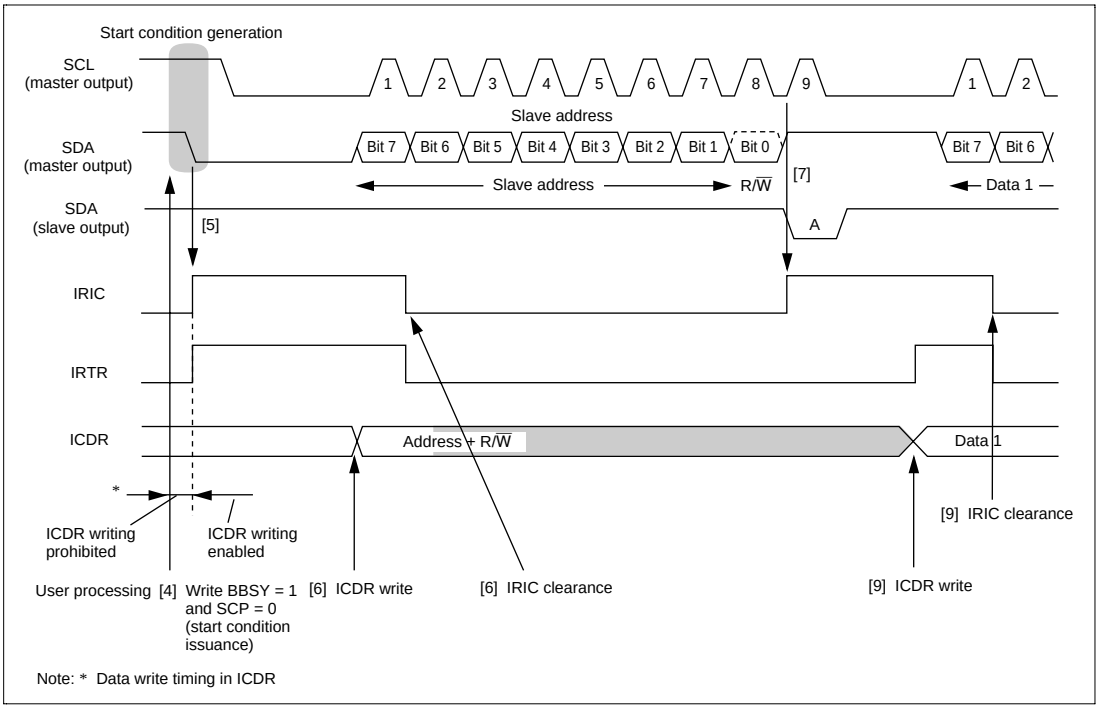


Figure 15.6 Example of Master Transmit Mode Operation Timing (MLS = WAIT = 0)

15.3.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data, and returns an acknowledge signal. The slave device transmits data. The reception procedure and operations with the wait function in master receive mode are described below.

[1] Clear the TRS bit in ICCR to 0 to switch from transmit mode to receive mode, and set the WAIT bit in ICMR to 1. Also clear the bit in ICSR to ACKB 0 (acknowledge data setting).

[2] When ICDR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. In order to detect wait operation, set the IRIC flag in ICCR must be cleared to 0. After reading ICDR, clear IRIC immediately not to execute other interrupt handling routine. If one frame of data has been received before the IRIC clearing, it can not be determine the end of reception.

[3] The IRIC flag is set to 1 at the fall of the 8th receive clock pulse. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. SCL is automatically fixed low in synchronization with the internal clock until the IRIC flag clearing. If the first frame is the last receive data, execute step [10] to halt reception.

[4] Clear the IRIC flag to release from the Wait State. The master device outputs the 9th clock and drives SDA at the 9th receive clock pulse to return an acknowledge signal.

[5] When one frame of data has been received, the IRIC flag in ICCR and the IRTR flag in ICSR are set to 1 at the rise of the 9th receive clock pulse. The master device outputs SCL clock to receive next data.

[6] Read ICDR.

[7] Clear the IRIC flag to detect next wait operation. Data reception process from [5] to [7] should be executed during one byte reception period after IRIC flag clearing in [4] or [9] to release wait status.

[8] The IRIC flags set to 1 at the fall of 8th receive clock pulse. SCL is automatically fixed low in synchronization with the internal clock until the IRIC flag clearing. If this frame is the last receive data, execute step [10] to halt reception.

[9] Clear the IRIC flag in ICCR to cancel wait operation. The master device outputs the 9th clock and drives SDA at the 9th receive clock pulse to return an ackowledge signal. Data can be received continuously by repeating step [5] to [9].

[10] Set the ACKB bit in ICSR to 1 so as to return “No acknowledge” data. Also set the TRS bit to 1 to switch from receive mode to transmit mode.

[11] Clear IRIC flag to 0 to release from the Wait State.

[12] When one frame of data has been received, the IRIC flag is set to 1 at the rise of the 9th receive clock pulse.

[13] Clear the WAIT bit to 0 to switch from wait mode to no wait mode. Read ICDR and the IRIC flag to 0. Clearing of the IRIC flag should be after the WAIT = 0.

[14] Clear the BBSY bit and SCP bit to 0. This changes SDA from low to high when SCL is high, and generates the stop condition.

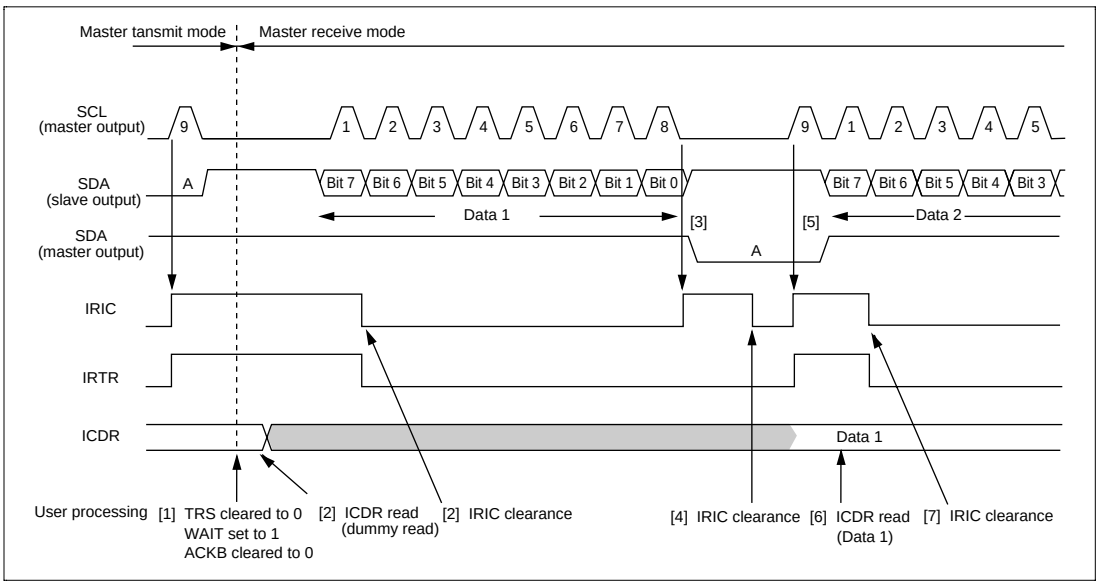


Figure 15.7 Example of Master Receive Mode Operation Timing (1)
(NLS = ACKB = 0, WAIT = 1)

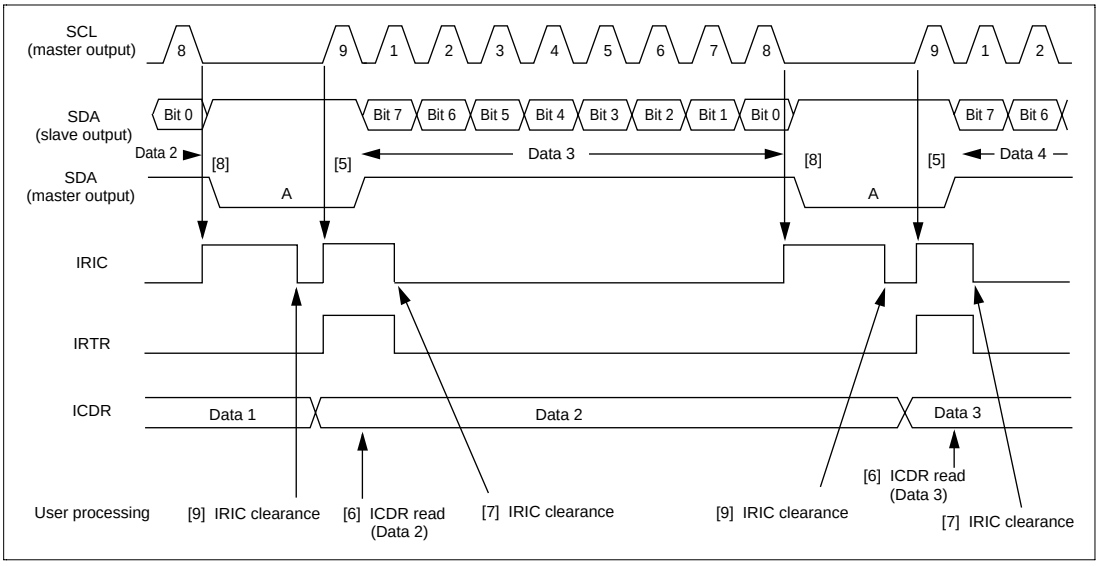


Figure 15.7 Example of Master Receive Mode Operation Timing (2)
(MLS = ACKB = 0, WAIT = 1)

15.3.4 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. The reception procedure and operations in slave receive mode are described below.

[1] Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.

[2] When the start condition output by the master device is detected, the BBSY flag in ICCR is set to 1.

[3] When the slave address matches in the first frame following the start condition, the device operates as the slave device specified by the master device. If the 8th data bit (R/$\overline{W}$) is 0, the TRS bit in ICCR remains cleared to 0, and slave receive operation is performed.

[4] At the 9th clock pulse of the receive frame, the slave device drives SDA low and returns an acknowledge signal. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. If the RDRF internal flag has been cleared to 0, it is set to 1, and the receive operation continues. If the RDRF internal flag has been set to 1, the slave device drives SCL low from the fall of the receive clock until data is read into ICDR.

[5] Read ICDR and clear the IRIC flag in ICCR to 0. The RDRF flag is cleared to 0.

Receive operations can be performed continuously by repeating steps [4] and [5]. When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

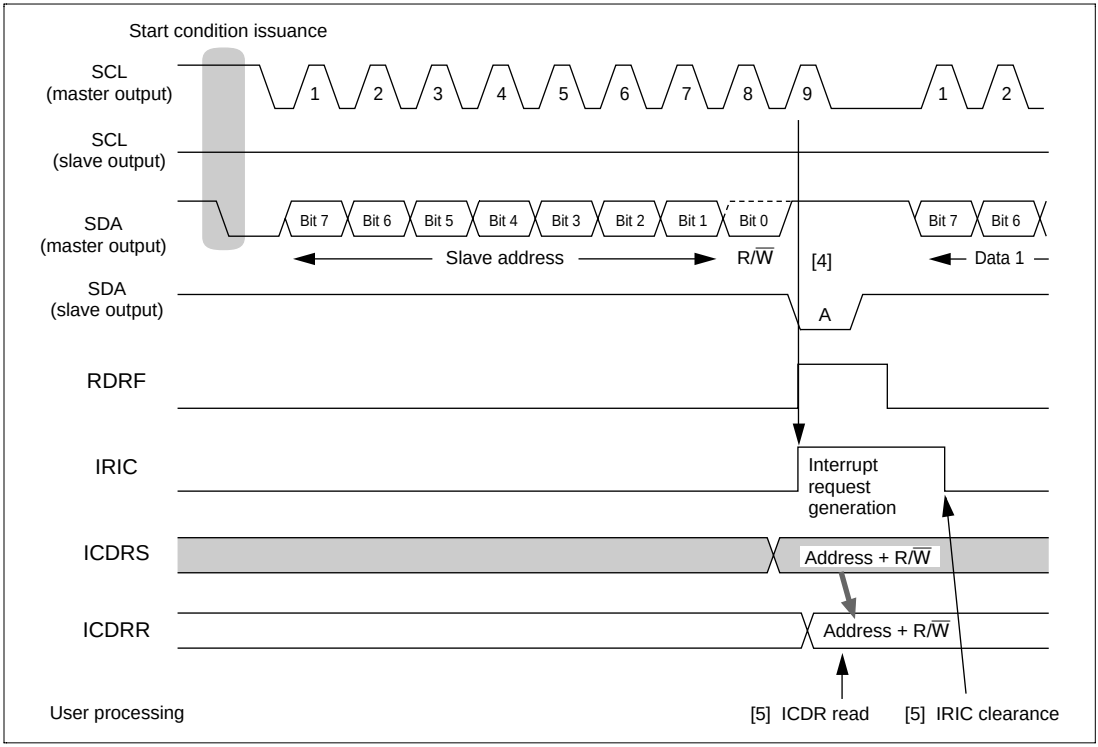


Figure 15.8 Example of Slave Receive Mode Operation Timing (1) (MLS = ACKB = 0)

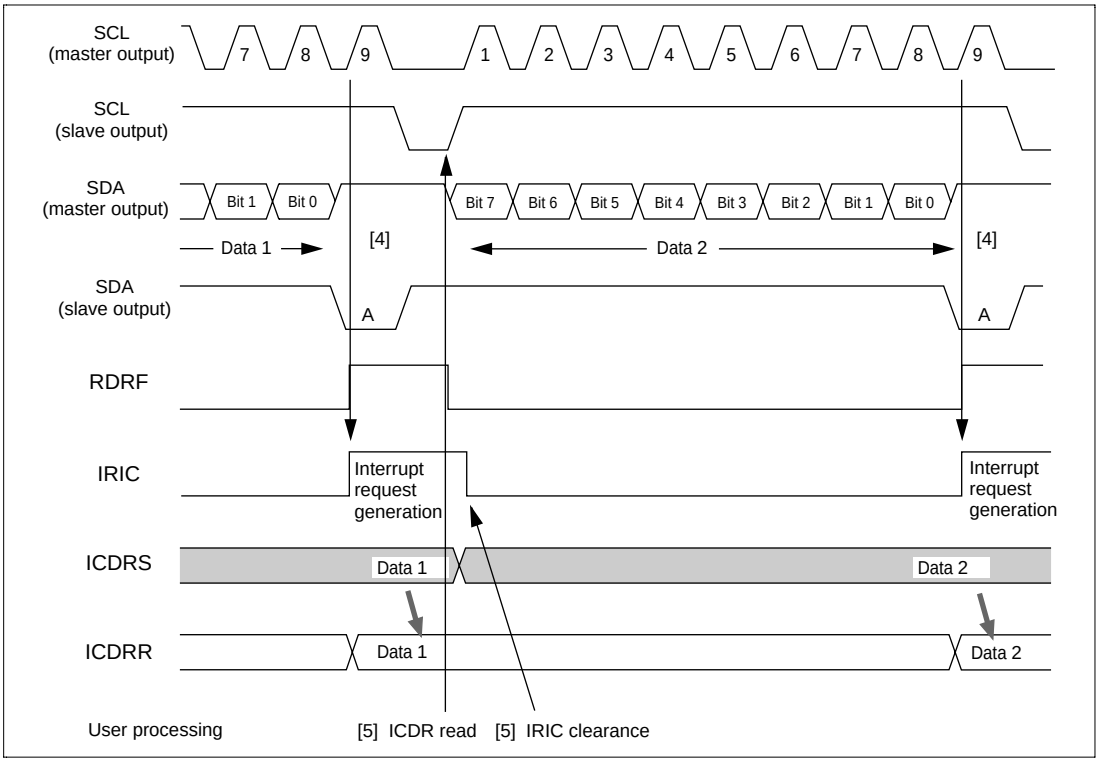


Figure 15.9 Example of Slave Receive Mode Operation Timing (2)
(MLS = ACKB = 0)

15.3.5 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. The transmission procedure and operations in slave transmit mode are described below.

[1] Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.

[2] When the slave address matches in the first frame following detection of the start condition, the slave device drives SDA low at the 9th clock pulse and returns an acknowledge signal. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. If the 8th data bit (R/$\overline{W}$) is 1, the TRS bit in ICCR is set to 1, and the mode changes to slave transmit mode automatically. The TDRE internal flag is set to 1. The slave device drives SCL low from the fall of the transmit clock until ICDR data is written.

[3] After clearing the IRIC flag to 0, write data to ICDR. The TDRE internal flag is cleared to 0. The written data is transferred to ICDRS, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again. After clearing the IRIC flag to 0, write the next data to ICDR. The

slave device sequentially sends the data written into ICDR in accordance with the clock output by the master device at the timing shown in figure 15.10.

[4] When one frame of data has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit clock pulse. If the TDRE internal flag has been set to 1, this slave device drives SCL low from the fall of the transmit clock until data is written to ICDR. The master device drives SDA low at the 9th clock pulse, and returns an acknowledge signal. As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used to determine whether the transfer operation was performed normally. When the TDRE internal flag is 0, the data written into ICDR is transferred to ICDRS, transmission is started, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again.

[5] To continue transmission, clear the IRIC flag to 0, then write the next data to be transmitted into ICDR. The TDRE flag is cleared to 0.

Transmit operations can be performed continuously by repeating steps [4] and [5]. To end transmission, write H'FF to ICDR. When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

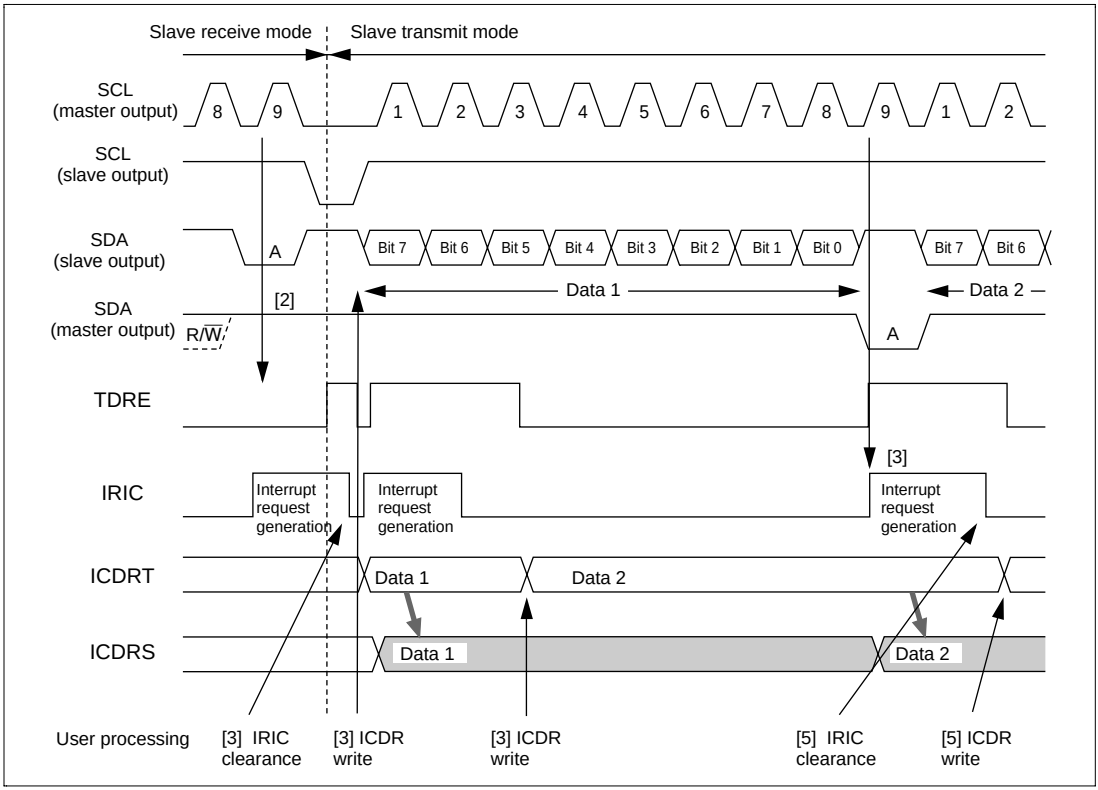


Figure 15.10 Example of Slave Transmit Mode Operation Timing (MLS = 0)

15.3.6 IRIC Setting Timing and SCL Control

The interrupt request flag (IRIC) is set at different times depending on the WAIT bit in ICMR, the FS bit in SAR, and the FSX bit in SARX. If the TDRE or RDRF internal flag is set to 1, SCL is automatically held low after one frame has been transferred; this timing is synchronized with the internal clock. Figure 15.11 shows the IRIC set timing and SCL control.

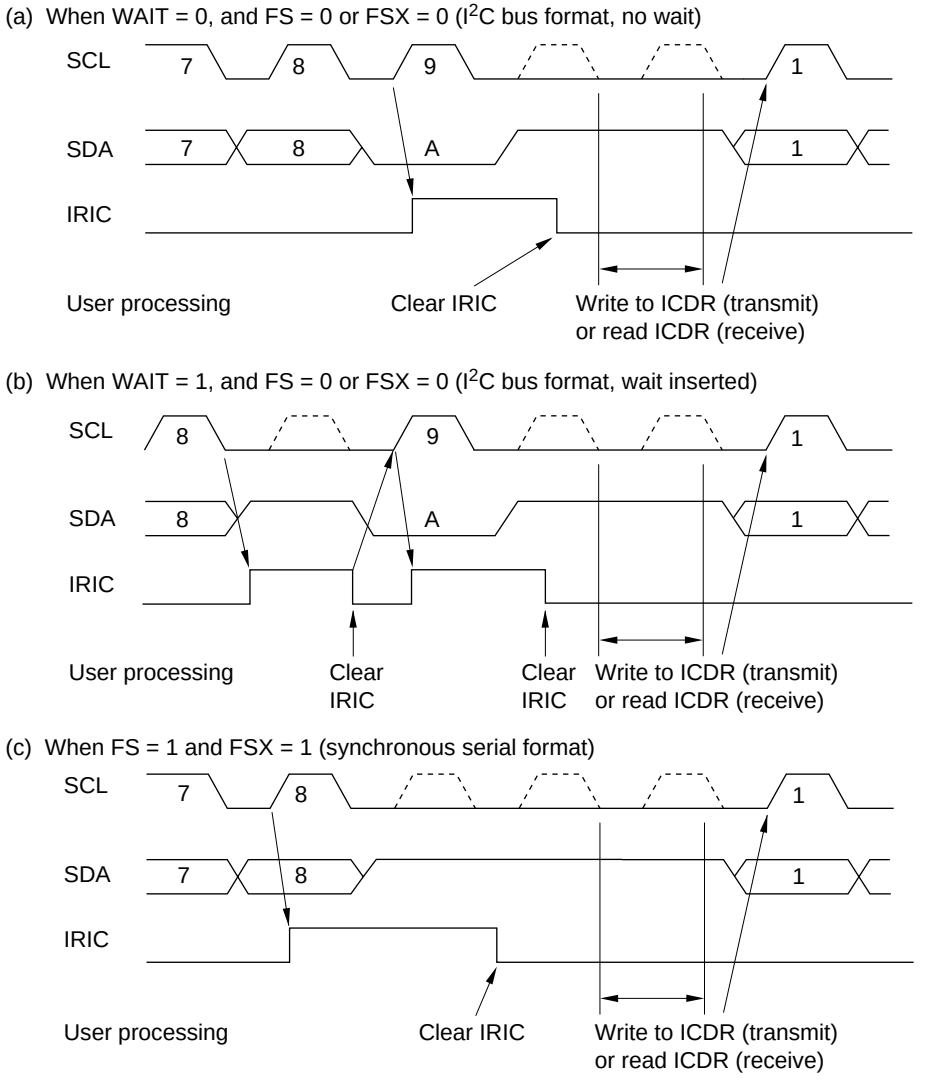


Figure 15.11 IRIC Setting Timing and SCL Control

15.3.7 Noise Canceler

The logic levels at the SCL and SDA pins are routed through noise cancelers before being latched internally. Figure 15.12 shows a block diagram of the noise canceler circuit.

The noise canceler consists of two cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the system clock, but is not passed forward to the next circuit unless the outputs of both latches agree. If they do not agree, the previous value is held.

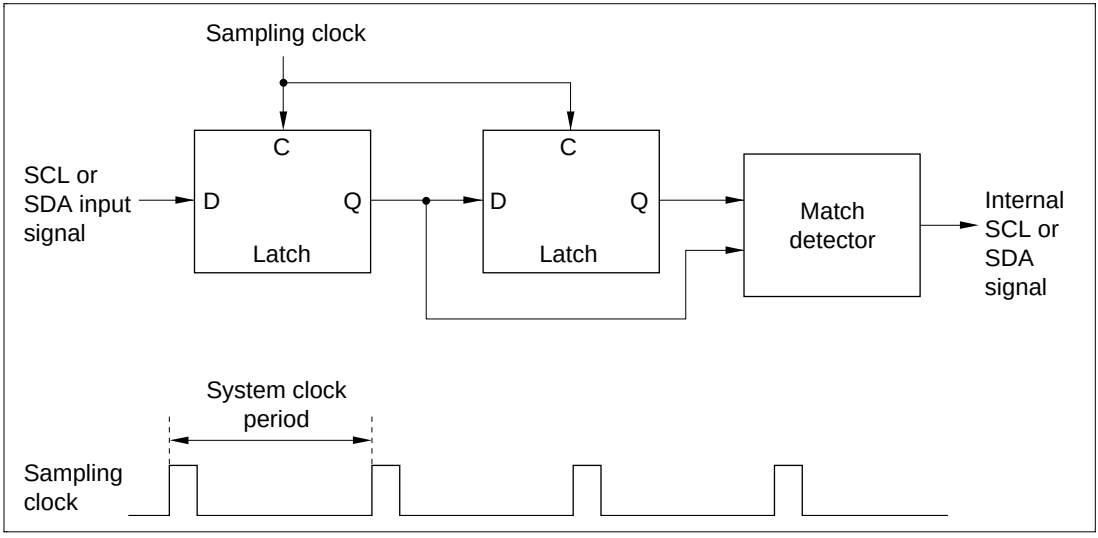


Figure 15.12 Block Diagram of Noise Canceler

15.3.8 Sample Flowcharts

Figures 15.13 to 15.16 show sample flowcharts for using the I^2C bus interface in each mode.

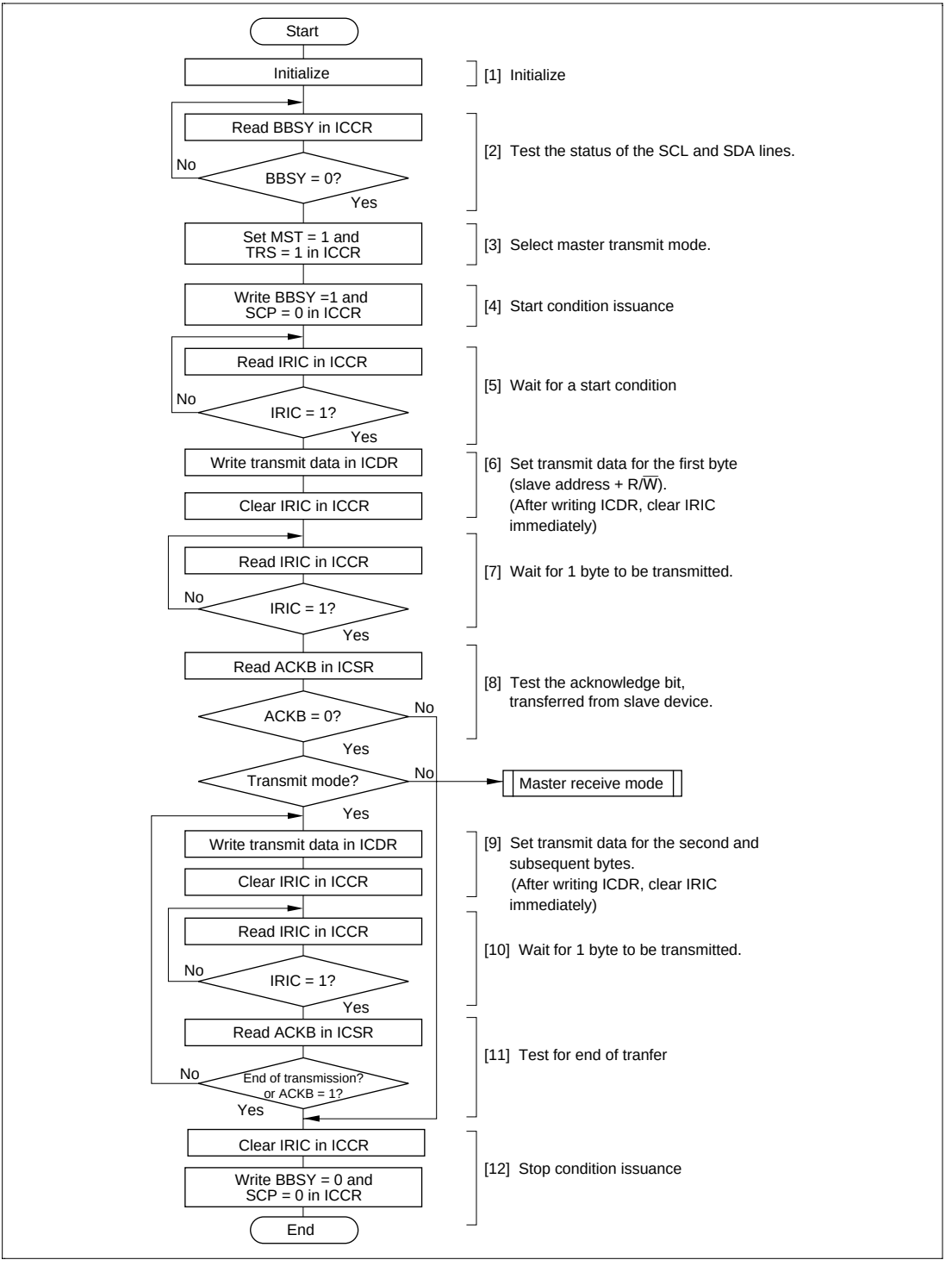


Figure 15.13 Flowchart for Master Transmit Mode (Example)

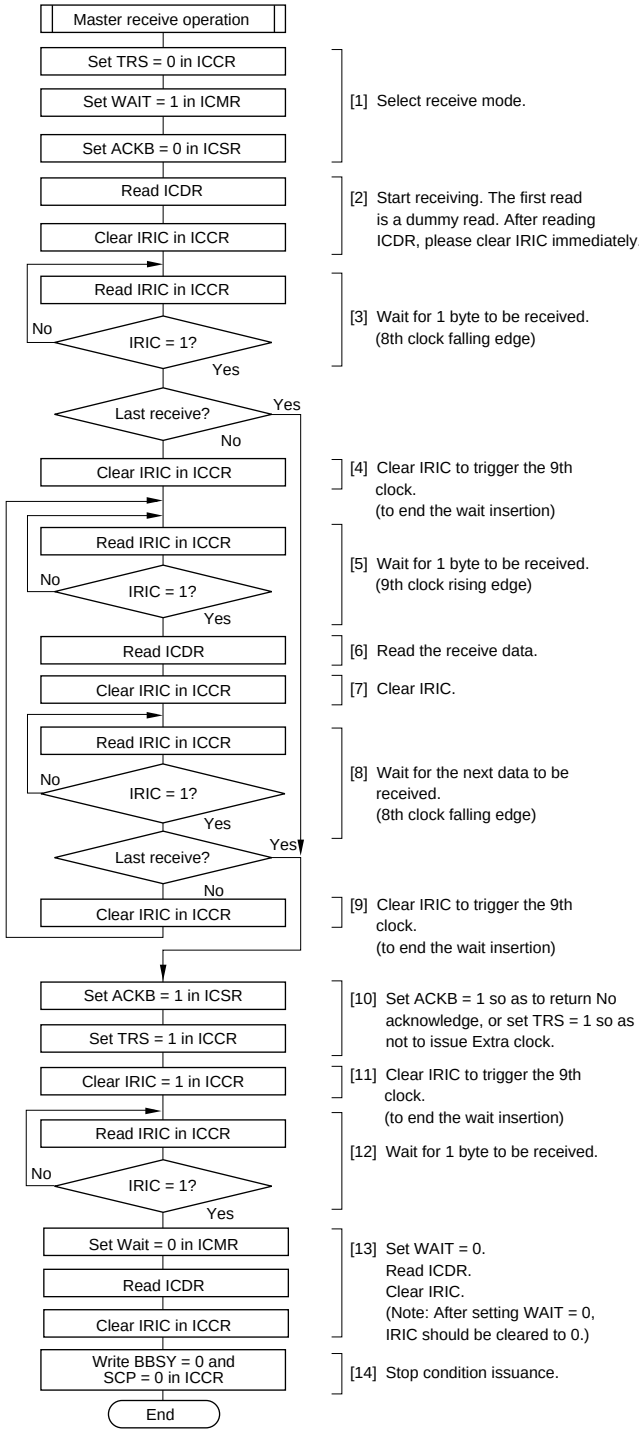


Figure 15.14 Flowchart for Master Receive Mode (Example)

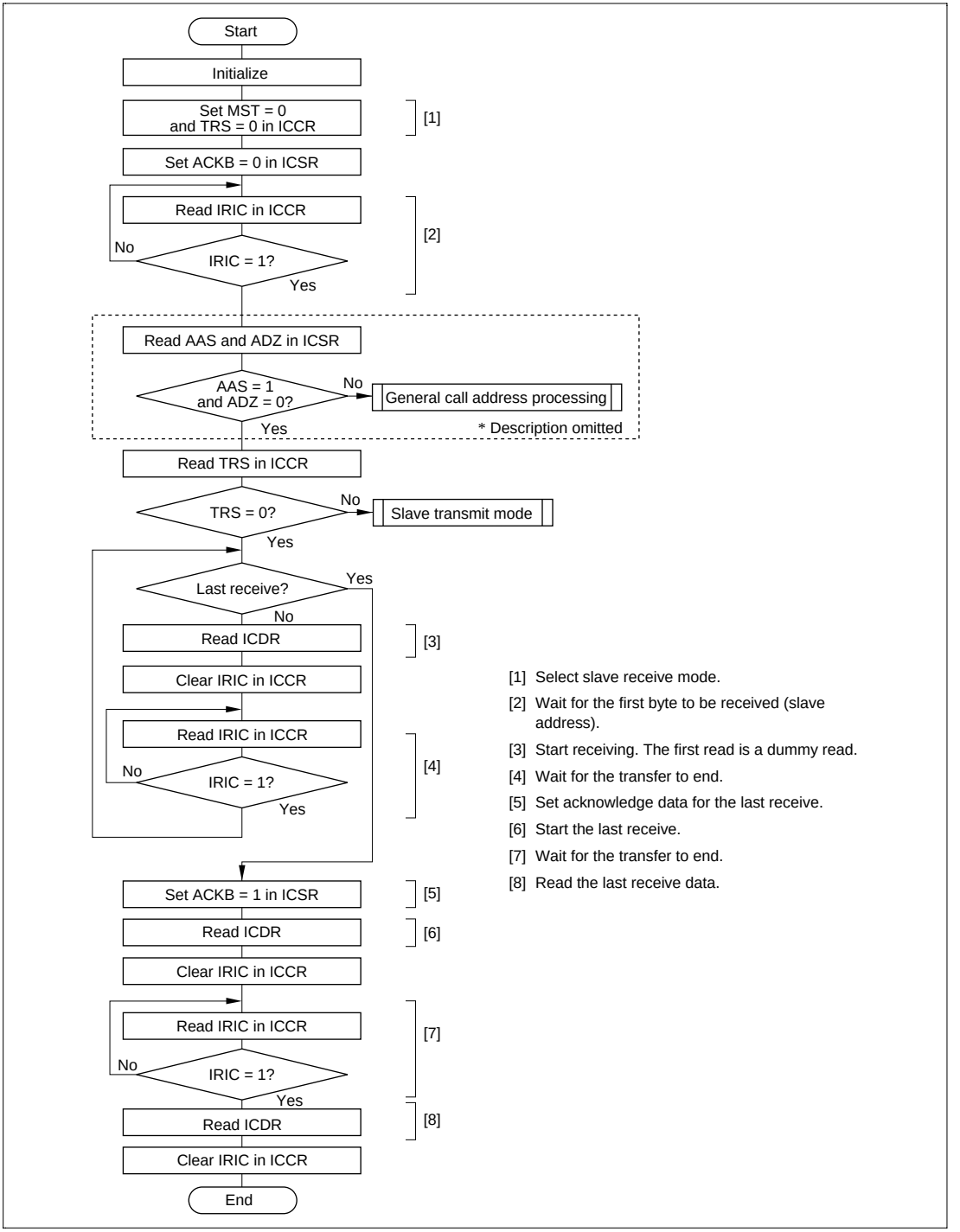


Figure 15.15 Flowchart for Slave Receive Mode (Example)

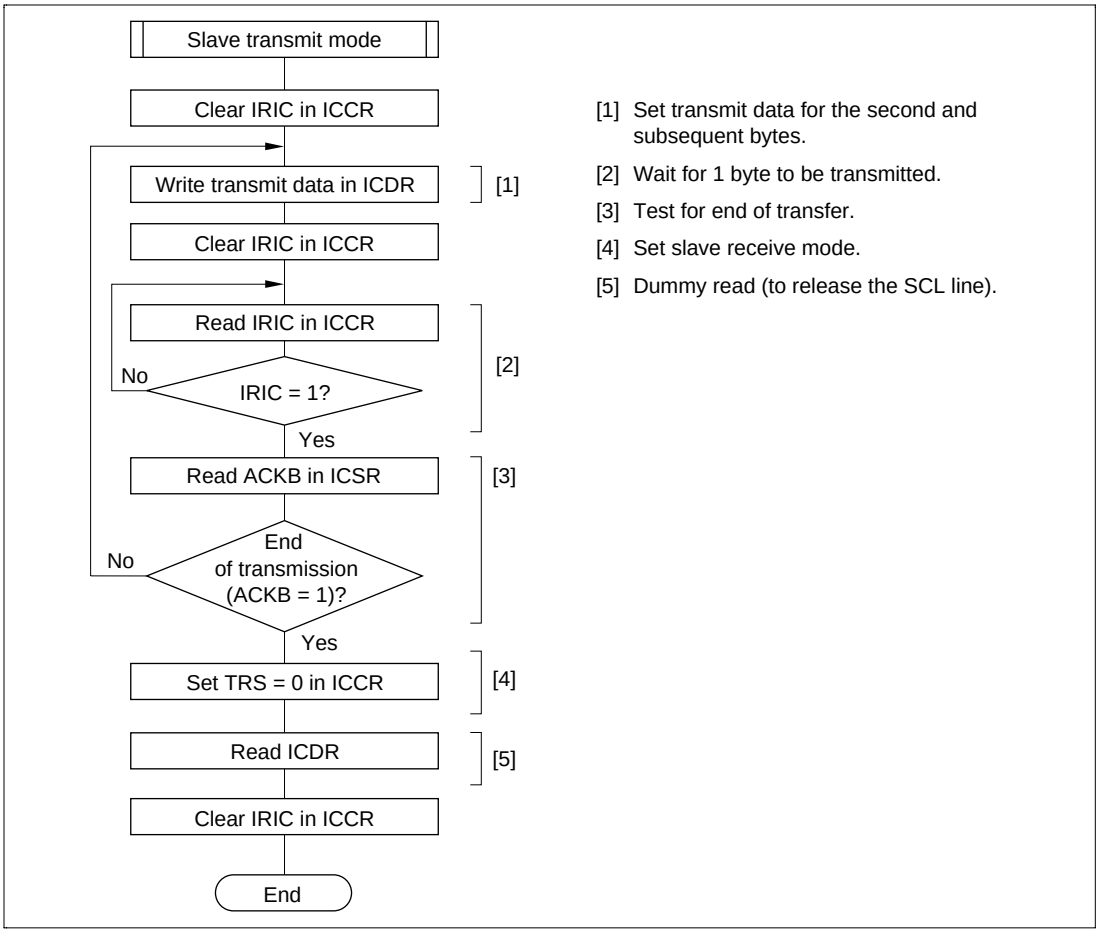


Figure 15.16 Flowchart for Slave Transmit Mode (Example)

15.4 Usage Notes

- In master mode, if an instruction to generate a start condition is immediately followed by an instruction to generate a stop condition, neither condition will be output correctly. To output consecutive start and stop conditions, after issuing the instruction that generates the start condition, read the relevant ports, check that SCL and SDA are both low, then issue the instruction that generates the stop condition. Note that SCL may not yet have gone low when BBSY is cleared to 0.

- Either of the following two conditions will start the next transfer. Pay attention to these conditions when reading or writing to ICDR.
  - — Write access to ICDR when ICE = 1 and TRS = 1 (including automatic transfer from ICDRT to ICDRS)
  - — Read access to ICDR when ICE = 1 and TRS = 0 (including automatic transfer from ICDRS to ICDRR)

- Table 15.5 shows the timing of SCL and SDA output in synchronization with the internal clock. Timings on the bus are determined by the rise and fall times of signals affected by the bus load capacitance, series resistance, and parallel resistance.

Table 15.5 I^2C Bus Timing (SCL and SDA Output)

Item	Symbol	Output Timing	Unit	Notes
SCL output cycle time	t_{SCLO}	$28t_{cyc}$ to $256t_{cyc}$	ns	
SCL output high pulse width	t_{SCLHO}	$0.5t_{SCLO}$	ns	
SCL output low pulse width	t_{SCLLO}	$0.5t_{SCLO}$	ns	
SDA output bus free time	t_{BUFO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Start condition output hold time	t_{STAHO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Retransmission start condition output setup time	t_{STASO}	$1t_{SCLO}$	ns	
Stop condition output setup time	t_{STOSO}	$0.5t_{SCLO} + 2t_{cyc}$	ns	
Data output setup time (master)	t_{SDASO}	$1t_{SCLLO} – 3t_{cyc}$	ns	
Data output setup time (slave)		$1t_{SCLL} – 3t_{cyc}$	ns	
Data output hold time	t_{SDAHO}	$3t_{cyc}$	ns	

- SCL and SDA inputs are sampled in synchronization with the internal clock. The AC timing therefore depends on the system clock cycle t_{cyc}, as shown in table 17.4 in section 17, Electrical Characteristics. Note that the I^2C bus interface AC timing specifications will not be met with a system clock frequency of less than 5 MHz.

- The I^2C bus interface specification for the SCL rise time t_{sr} is under 1000 ns (300 ns for high-speed mode). In master mode, the I^2C bus interface monitors the SCL line and synchronizes one bit at a time during communication. If t_{sr} (the time for SCL to go from low to V_{IH}) exceeds the time determined by the input clock of the I^2C bus interface, the high period of SCL is extended. The SCL rise time is determined by the pull-up resistance and load capacitance of the SCL line. To insure proper operation at the set transfer rate, adjust the pull-up resistance and load capacitance so that the SCL rise time does not exceed the values given in the table below.

Table 15.6 Permissible SCL Rise Time (t_{sr}) Values

			Time Indication				
IICX	t_{cyc} Indication		I^2C Bus Specification (Max.)	ø = 5 MHz	ø = 8 MHz	ø = 10 MHz	ø = 16 MHz
0	$7.5t_{cyc}$	Normal mode	1000 ns	1000 ns	937 ns	750 ns	468 ns
		High-speed mode	300 ns	300 ns	300 ns	300 ns	300 ns
1	$17.5t_{cyc}$	Normal mode	1000 ns	1000 ns	1000 ns	1000 ns	1000 ns
		High-speed mode	300 ns	300 ns	300 ns	300 ns	300 ns

- The I^2C bus interface specifications for the SCL and SDA rise and fall times are under 1000 ns and 300 ns. The I^2C bus interface SCL and SDA output timing is prescribed by t_{Scyc} and t_{cyc}, as shown in table 15.6. However, because of the rise and fall times, the I^2C bus interface specifications may not be satisfied at the maximum transfer rate. Table 15.8 shows output timing calculations for different operating frequencies, including the worst-case influence of rise and fall times.

  t_{BUFO} fails to meet the I^2C bus interface specifications at any frequency. The solution is either (a) to provide coding to secure the necessary interval (approximately 1 µs) between issuance of a stop condition and issuance of a start condition, or (b) to select devices whose input timing permits this output timing for use as slave devices connected to the I^2C bus.

  t_{SCLLO} in high-speed mode and t_{STASO} in standard mode fail to satisfy the I^2C bus interface specifications for worst-case calculations of t_{Sr}/t_{Sf}. Possible solutions that should be investigated include (a) adjusting the rise and fall times by means of a pull-up resistor and capacitive load, (b) reducing the transfer rate to meet the specifications, or (c) selecting devices whose input timing permits this output timing for use as slave devices connected to the I^2C bus.

Table 15.7 I^2C Bus Timing (with Maximum Influence of t_{Sr}/t_{Sf})

			Time Indication (at Maximum Transfer Rate) [ns]					
Item	**t_{cyc} Indication**		**t_{Sr}/t_{Sf} Influence (Max.)**	**I^2C Bus Specification (Min.)**	**ø = 5 MHz**	**ø = 8 MHz**	**ø = 10 MHz**	**ø = 16 MHz**
t_{SCLHO}	$0.5t_{SCLO}$ $(-t_{Sr})$	Standard mode	–1000	4000	4000	4000	4000	4000
		High-speed mode	–300	600	950	950	950	950
t_{SCLLO}	$0.5t_{SCLO}$ $(-t_{Sf})$	Standard mode	–250	4700	4750	4750	4750	4750
		High-speed mode	–250	1300	1000*[1]	1000*[1]	1000*[1]	1000*[1]
t_{BUFO}	$0.5t_{SCLO}-1t_{cyc}$ $(-t_{Sr})$	Standard mode	–1000	4700	3800*[1]	3875*[1]	3900*[1]	3938*[1]
		High-speed mode	–300	1300	750*[1]	825*[1]	850*[1]	888*[1]
t_{STAHO}	$0.5t_{SCLO}-1t_{cyc}$ $(-t_{Sf})$	Standard mode	–250	4000	4550	4625	4650	4688
		High-speed mode	–250	600	800	875	900	938
t_{STASO}	$1t_{SCLO}$ $(-t_{Sr})$	Standard mode	–1000	4700	9000	9000	9000	9000
		High-speed mode	–300	600	2200	2200	2200	2200
t_{STOSO}	$0.5t_{SCLO}+2t_{cyc}$ $(-t_{Sr})$	Standard mode	–1000	4000	4400	4250	4200	4125
		High-speed mode	–300	600	1350	1200	1150	1075
t_{SDASO} (master)	$1t_{SCLLO}$*[2] $-3t_{cyc}$ $(-t_{Sr})$	Standard mode	–1000	250	3100	3325	3400	3513
		High-speed mode	–300	100	400	625	700	813
t_{SDASO} (slave)	$1t_{SCLL}$*[2] $-3t_{cyc}$ $(-t_{Sr})$	Standard mode	–1000	250	3100	3325	2400	3513
		High-speed mode	–300	100	400	625	700	813
t_{SDAHO}	$3t_{cyc}$	Standard mode	0	0	600	375	300	188
		High-speed mode	0	0	600	375	300	188

Notes: 1. Does not meet the I^2C bus interface specification. Remedial action such as the following is necessary: (a) secure a start/stop condition issuance interval; (b) adjust the rise and fall times by means of a pull-up resistor and capacitive load; (c) reduce the transfer rate; (d) select slave devices whose input timing permits this output timing.

The values in the above table will vary depending on the settings of the IICX bit and bits CKS0 to CKS2. Depending on the frequency it may not be possible to achieve the maximum transfer rate; therefore, whether or not the I^2C bus interface specifications are met must be determined in accordance with the actual setting conditions.

2. Calculated using the I^2C bus specification values (standard mode: 4700 ns min.; high-speed mode: 1300 ns min.).

- Note on ICDR Read at end of Master Reception

  To halt reception after completion of a receive operation in master receive mode, set the TRS bit to 1 and write 0 to BBSY and SCP in ICCR. This changes the SDA pin from low to high when the SCL pin is high, and generates the stop condition. After this, receive data can be read by means of an ICDR read, but if data remains in the buffer the ICDRS receive data will not be transferred to ICDR, and so it will not be possible to read the second byte of data. If it is necessary to read the second byte of data, issue the stop condition in master receive mode (i.e. with the TRS bit cleared to 0). When reading the receive data, first confirm that the BBSY bit in ICCR is cleared to 0, the stop condition has been generated, and the bus has been released, then read ICDR with TRS cleared to 0. Note that if the receive data (ICDR data) is read in the interval between execution of the instruction for issuance of the stop condition (writing of 0 to BBSY and SCP in ICCR) and the actual generation of the stop condition, the clock may not be output correctly in subsequent master transmission.

- Notes on Start Condition Issuance for Retransmission

  Depending on the timing combination with the start condition issuance and the subsequently writing data to ICDR, it may not be possible to issue the retransmission and the data transmission after retransmission condition issuance.

  After start condition issuance is done and determined the start condition, write the transmit data to ICDR, as shown below. Figure 15.17 shows the timing of start condition issuance for retransmission, and the timing for subsequently writing data to ICDR, together with the corresponding flowchart.

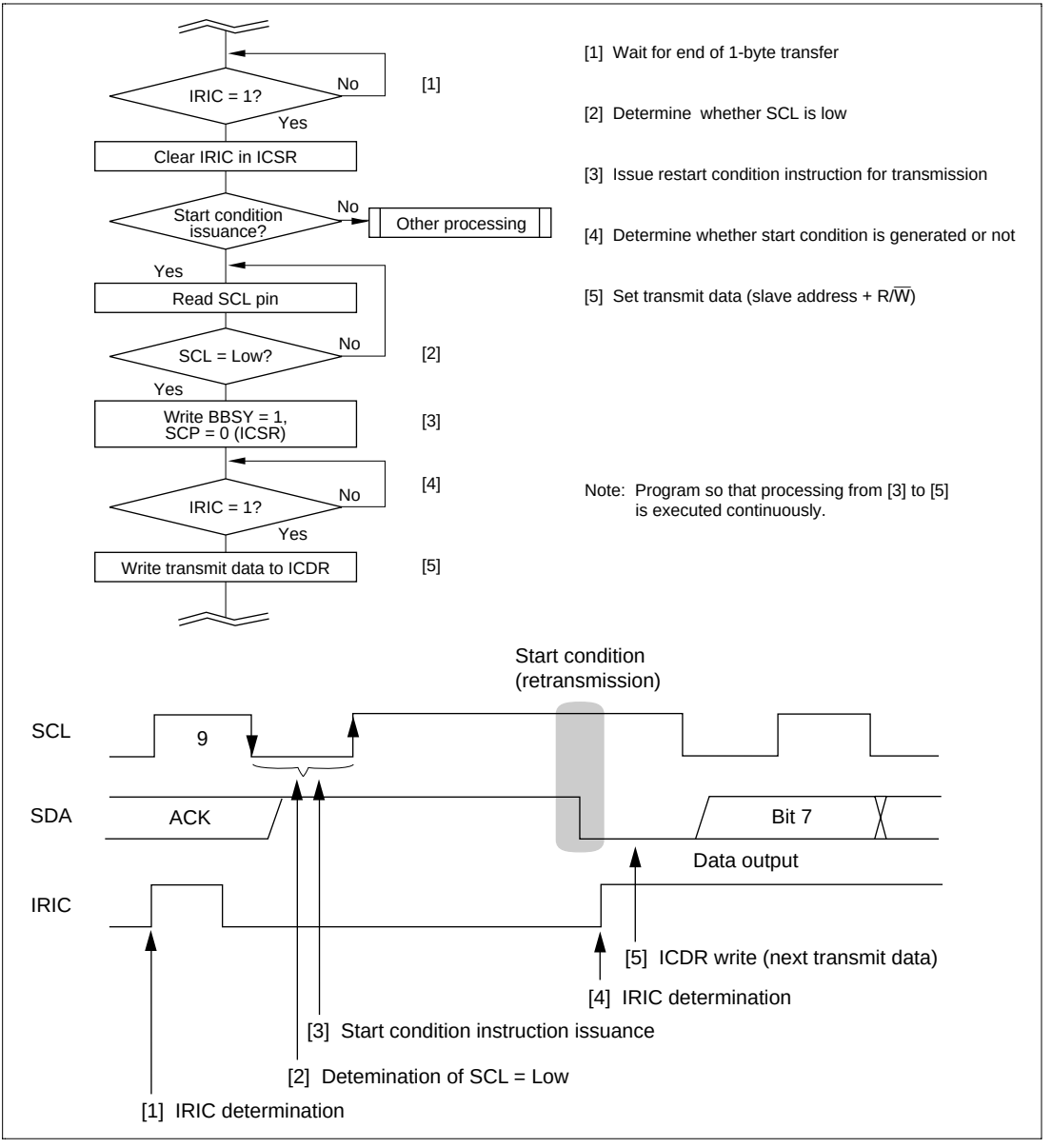


Figure 15.17 Flowchart and Timing of Start Condition Instruction Issuance for Retransmission

Section 16 A/D Converter

16.1 Overview

The H8/3664 Series includes a 10-bit successive-approximations A/D converter with a selection of up to eight analog input channels.

When the A/D converter is not used, it can be halted independently to conserve power. For details see section 6.10, Module Standby Mode.

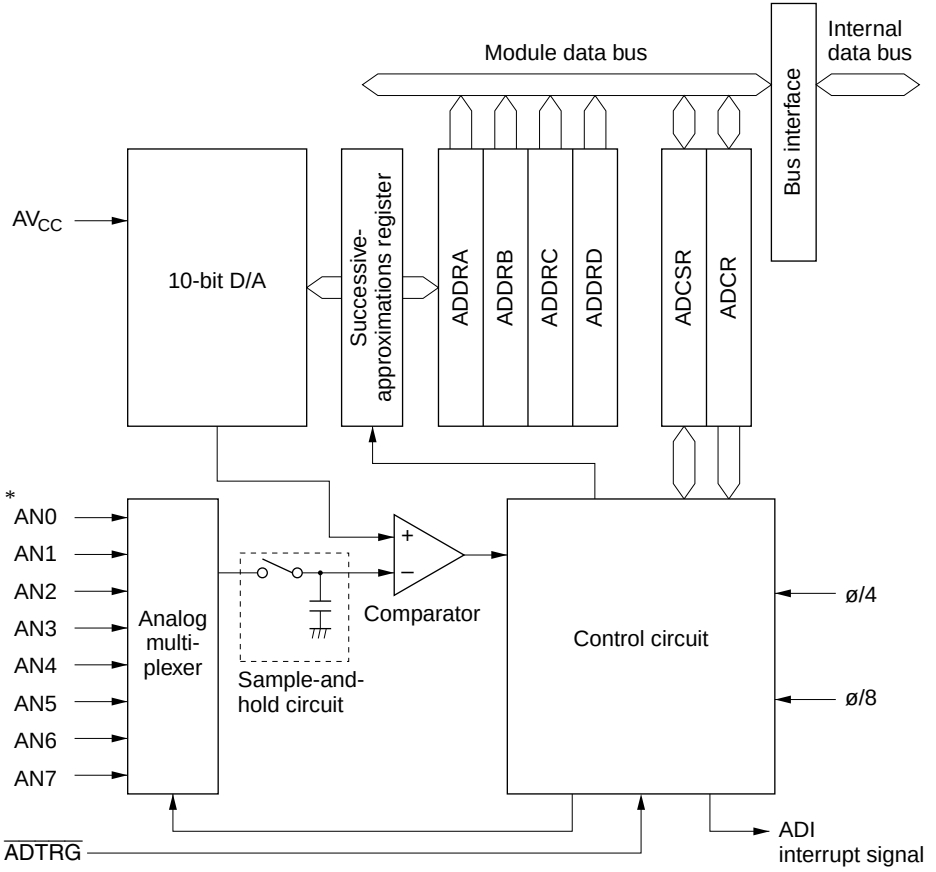
16.1.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Eight input channels (for 42-pin type: four input channels)
- High-speed conversion
  Conversion time: minimum 4.4 µs per channel (with 16 MHz system clock)
- Two operating modes
  Single mode: A/D conversion of one channel
  Scan mode: continuous A/D conversion on one to four channels
- Four 16-bit data registers
  A/D conversion results are transferred for storage into data registers corresponding to the channels.
- Sample-and-hold function
- Two conversion start sources
  The A/D converter can be activated by software or an external trigger.
- A/D interrupt requested at end of conversion
  At the end of A/D conversion, an A/D end interrupt (ADI) can be requested.

16.1.2 Block Diagram

Figure 16.1 shows a block diagram of the A/D converter.



Legend:

ADCR: A/D control register
ADCSR: A/D control/status register
ADDRA: A/D data register A
ADDRB: A/D data register B
ADDRC: A/D data register C
ADDRD: A/D data register D

Note: * The 42-pin type does not have pins AN4, AN5, AN6, and AN7.

Figure 16.1 A/D Converter Block Diagram

16.1.3 Input Pins

Table 16.1 summarizes the A/D converter's input pins. The eight analog input pins are divided into two groups: group 0 (AN0 to AN3), and group 1 (AN4 to AN7). AV_{CC} is the power supply for the analog circuits in the A/D converter.

Table 16.1 A/D Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog power supply
Analog input pin 0	AN0	Input	Group 0 analog inputs
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	Group 1 analog inputs
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

16.1.4 Register Configuration

Table 16.2 summarizes the A/D converter's registers.

Table 16.2 A/D Converter Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FFFB0	A/D data register A H	ADDRAH	R	H'00
H'FFFB1	A/D data register A L	ADDRAL	R	H'00
H'FFFB2	A/D data register B H	ADDRBH	R	H'00
H'FFFB3	A/D data register B L	ADDRBL	R	H'00
H'FFFB4	A/D data register C H	ADDRCH	R	H'00
H'FFFB5	A/D data register C L	ADDRCL	R	H'00
H'FFFB6	A/D data register D H	ADDRDH	R	H'00
H'FFFB7	A/D data register D L	ADDRDL	R	H'00
H'FFFB8	A/D control/status register	ADCSR	R/(W)*	H'00
H'FFFB9	A/D control register	ADCR	R/W	H'7E

Note: * Only 0 can be written in bit 7, to clear the flag.

16.2 Register Descriptions

16.2.1 A/D Data Registers A to D (ADDRA to ADDRD)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRn	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write (n = A to D)	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

A/D conversion data
10-bit data giving an A/D conversion result

Reserved bits

The four A/D data registers (ADDRA to ADDRD) are 16-bit read-only registers that store the results of A/D conversion.

An A/D conversion produces 10-bit data, which is transferred for storage into the A/D data register corresponding to the selected channel. The upper 8 bits of the result are stored in the upper byte of the A/D data register. The lower 2 bits are stored in the lower byte. Bits 5 to 0 of an A/D data register are reserved bits that are always read as 0. Table 16.3 indicates the pairings of analog input channels and A/D data registers.

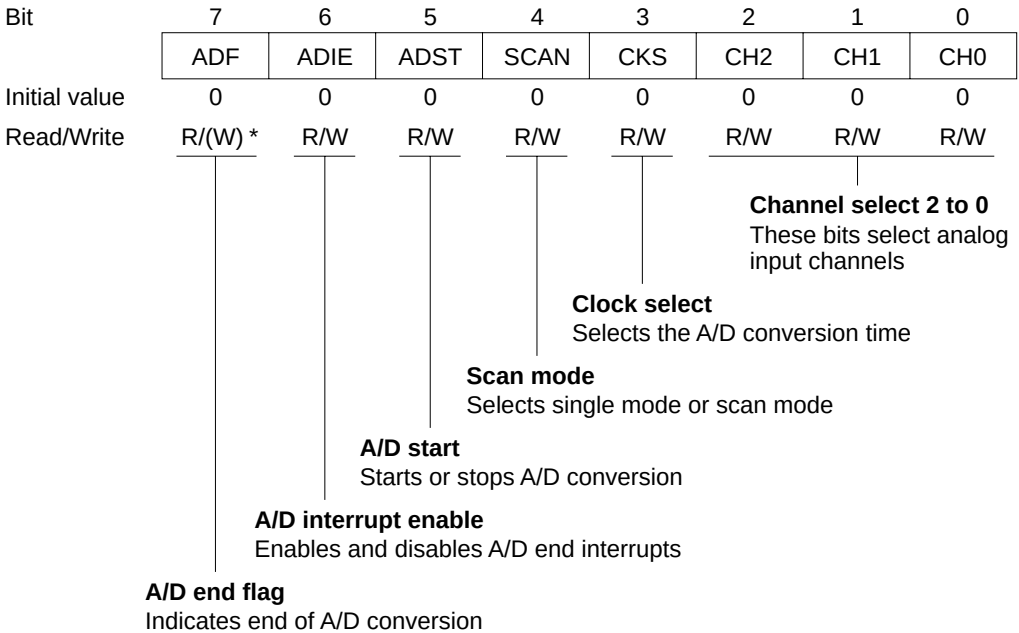
The CPU can always read the A/D data registers. The upper byte can be read directly, but the lower byte is read through a temporary register (TEMP). For details see section 16.3, CPU Interface.

The A/D data registers are initialized to H'0000 by a reset and in standby mode.

Table 16.3 Analog Input Channels and A/D Data Registers (ADDRA to ADDRD)

Analog Input Channel		
Group 0	**Group 1**	**A/D Data Register**
AN0	AN4	ADDRA
AN1	AN5	ADDRB
AN2	AN6	ADDRC
AN3	AN7	ADDRD

16.2.2 A/D Control/Status Register (ADCSR)



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

ADCSR is an 8-bit readable/writable register that selects the mode and controls the A/D converter. ADCSR is initialized to H'00 by a reset and in standby mode.

Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion.

Bit 7: ADF	Description	
0	[Clearing condition] Read ADF when ADF =1, then write 0 in ADF.	(Initial value)
1	[Setting conditions] • Single mode: A/D conversion ends • Scan mode: A/D conversion ends in all selected channels	

Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the interrupt (ADI) requested at the end of A/D conversion.

Bit 6: ADIE	Description	
0	A/D end interrupt request (ADI) is disabled	(Initial value)
1	A/D end interrupt request (ADI) is enabled	

Bit 5—A/D Start (ADST): Starts or stops A/D conversion. The ADST bit remains set to 1 during A/D conversion. It can also be set to 1 by external trigger input at the $\overline{\text{ADTRG}}$ pin.

Bit 5: ADST	Description	
0	A/D conversion is stopped	(Initial value)
1	Single mode: A/D conversion starts; ADST is automatically cleared to 0 when conversion ends. Scan mode: A/D conversion starts and continues, cycling among the selected channels, until ADST is cleared to 0 by software, by a reset, or by a transition to standby mode.	

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode. For further information on operation in these modes, see section 16.4, Operation. Clear the ADST bit to 0 before switching the conversion mode.

Bit 4: SCAN	Description	
0	Single mode	(Initial value)
1	Scan mode	

Bit 3—Clock Select (CKS): Selects the A/D conversion time. Clear the ADST bit to 0 before switching the conversion time.

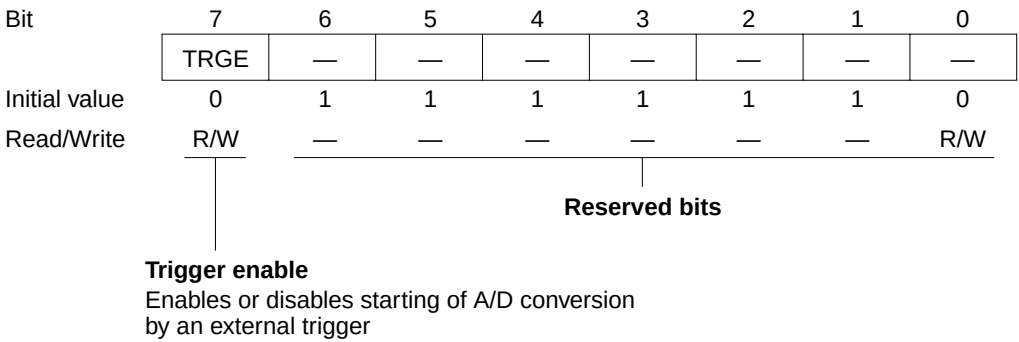
Bit 3: CKS	Description	
0	Conversion time = 134 states (maximum)	(Initial value)
1	Conversion time = 70 states (maximum)	

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): These bits and the SCAN bit select the analog input channels. Clear the ADST bit to 0 before changing the channel selection.

Group Selection	Channel Selection		Description	
CH2	CH1	CH0	Single Mode	Scan Mode
0	0	0	AN0 (Initial value)	AN0
		1	AN1	AN0, AN1
	1	0	AN2	AN0 to AN2
		1	AN3	AN0 to AN3
1*	0	0	AN4	AN4
		1	AN5	AN4, AN5
	1	0	AN6	AN4 to AN6
		1	AN7	AN4 to AN7

Note: 1 can not be set on the 42-pin type.

16.2.3 A/D Control Register (ADCR)



ADCR is an 8-bit readable/writable register that enables or disables starting of A/D conversion by external trigger input. ADCR is initialized to H'7E by a reset and in standby mode.

Bit 7—Trigger Enable (TRGE): Enables or disables starting of A/D conversion by an external trigger.

Bit 7: TRGE	Description	
0	Starting of A/D conversion by an external trigger is disabled	(Initial value)
1	A/D conversion is started at the falling edge and the rising edge of the external trigger signal ($\overline{\text{ADTRG}}$)	

Note: The selection between the falling edge and rising edge of the external trigger pin ($\overline{\text{ADTRG}}$) conforms to the setting of the interrupt edge select register 2 (IEGR2).

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

Bit 0—Reserved: This bit can be read or written, but must not be set to 1.

16.3 CPU Interface

ADDRA to ADDRD are 16-bit registers, but they are connected to the CPU by an 8-bit data bus. Therefore, although the upper byte can be be accessed directly by the CPU, the lower byte is read through an 8-bit temporary register (TEMP).

An A/D data register is read as follows. When the upper byte is read, the upper-byte value is transferred directly to the CPU and the lower-byte value is transferred into TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the CPU.

When reading an A/D data register, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, incorrect data may be obtained.

Figure 16.2 shows the data flow for access to an A/D data register.

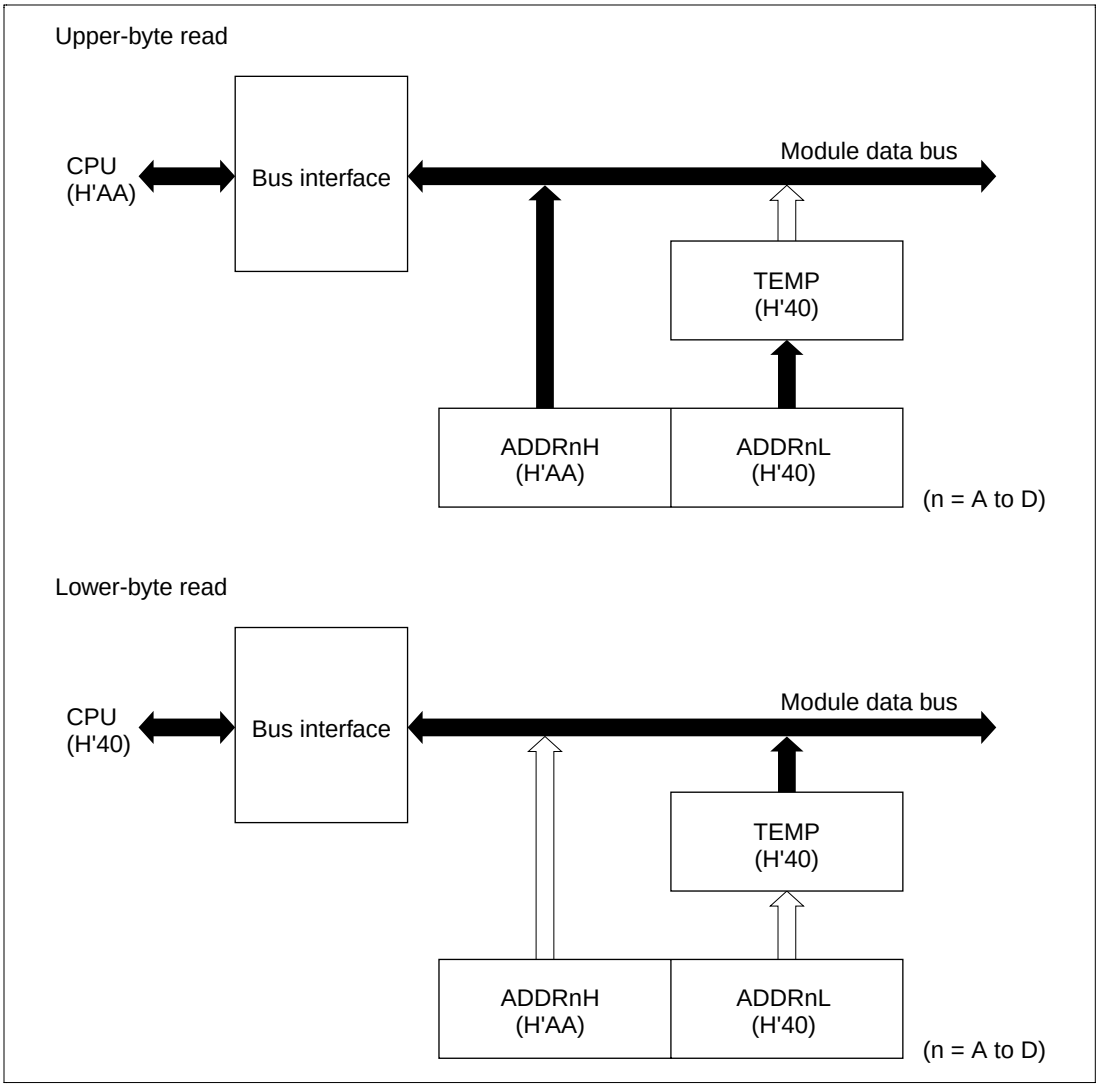


Figure 16.2 A/D Data Register Access Operation (Reading H'AA40)

16.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode.

16.4.1 Single Mode (SCAN = 0)

Single mode should be selected when only one A/D conversion on one channel is required. A/D conversion starts when the ADST bit is set to 1 by software, or by external trigger input. The ADST bit remains set to 1 during A/D conversion and is automatically cleared to 0 when conversion ends.

When conversion ends the ADF flag is set to 1. If the ADIE bit is also set to 1, an ADI interrupt is requested at this time. To clear the ADF flag to 0, first read ADCSR, then write 0 in ADF.

When the mode or analog input channel must be switched during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the mode or channel is changed.

Typical operations when channel 1 (AN1) is selected in single mode are described next.

Figure 16.3 shows a timing diagram for this example.

1. Single mode is selected (SCAN = 0), input channel AN1 is selected (CH2 = CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the result is transferred into ADDRB. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADCSR, then writes 0 in the ADF flag.
6. The routine reads and processes the conversion result (ADDRB).
7. Execution of the A/D interrupt handling routine ends. After that, if the ADST bit is set to 1, A/D conversion starts again and steps 2 to 7 are repeated.

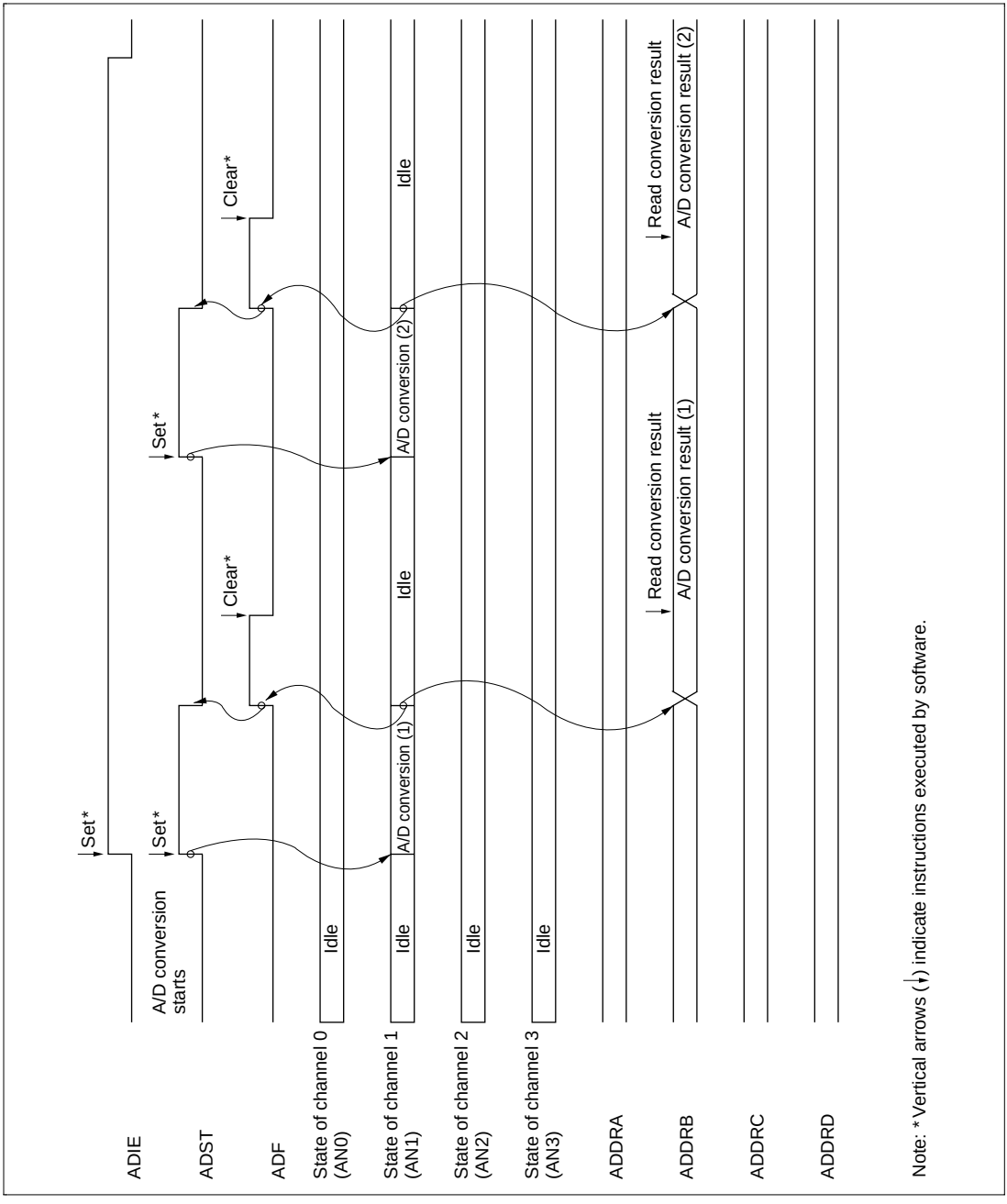


Figure 16.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

16.4.2 Scan Mode (SCAN = 1)

Scan mode is useful for monitoring analog inputs in a group of one or more channels. When the ADST bit is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN0 when CH2 = 0, AN4 when CH2 = 1). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN1 or AN5) starts immediately. A/D conversion continues cyclically on the selected channels until the ADST bit is cleared to 0. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the mode or analog input channel selection must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels in group 0 (AN0 to AN2) are selected in scan mode are described next. Figure 16.4 shows a timing diagram for this example.

1. Scan mode is selected (SCAN = 1), scan group 0 is selected (CH2 = 0), analog input channels AN_0 to AN_2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
2. When A/D conversion of the first channel (AN0) is completed, the result is transferred into ADDRA. Next, conversion of the second channel (AN1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN2).
4. When conversion of all selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set to 1, an ADI interrupt is requested when A/D conversion ends.
5. Steps 2 to 4 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN0).

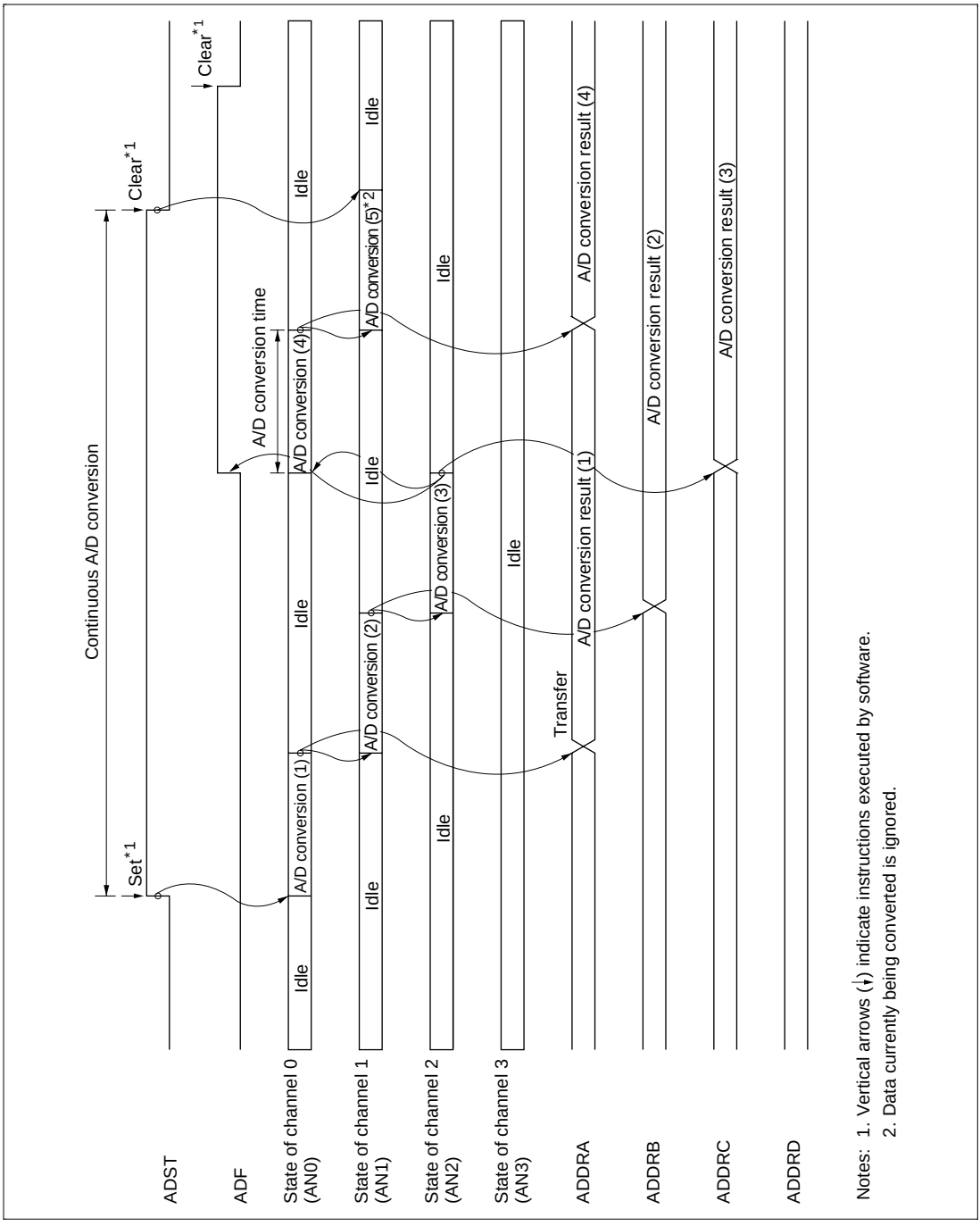


Figure 16.4 Example of A/D Converter Operation (Scan Mode, Channels AN0 to AN2 Selected)

16.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the input at a time t_D after the ADST bit is set to 1, then starts conversion. Figure 16.5 shows the A/D conversion timing. Table 16.4 indicates the A/D conversion time.

As indicated in figure 16.5, the A/D conversion time includes t_D and the input sampling time. The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 16.4.

In scan mode, the values given in table 16.4 apply to the first conversion. In the second and subsequent conversions the conversion time is fixed at 128 states when CKS = 0 or 66 states when CKS = 1.

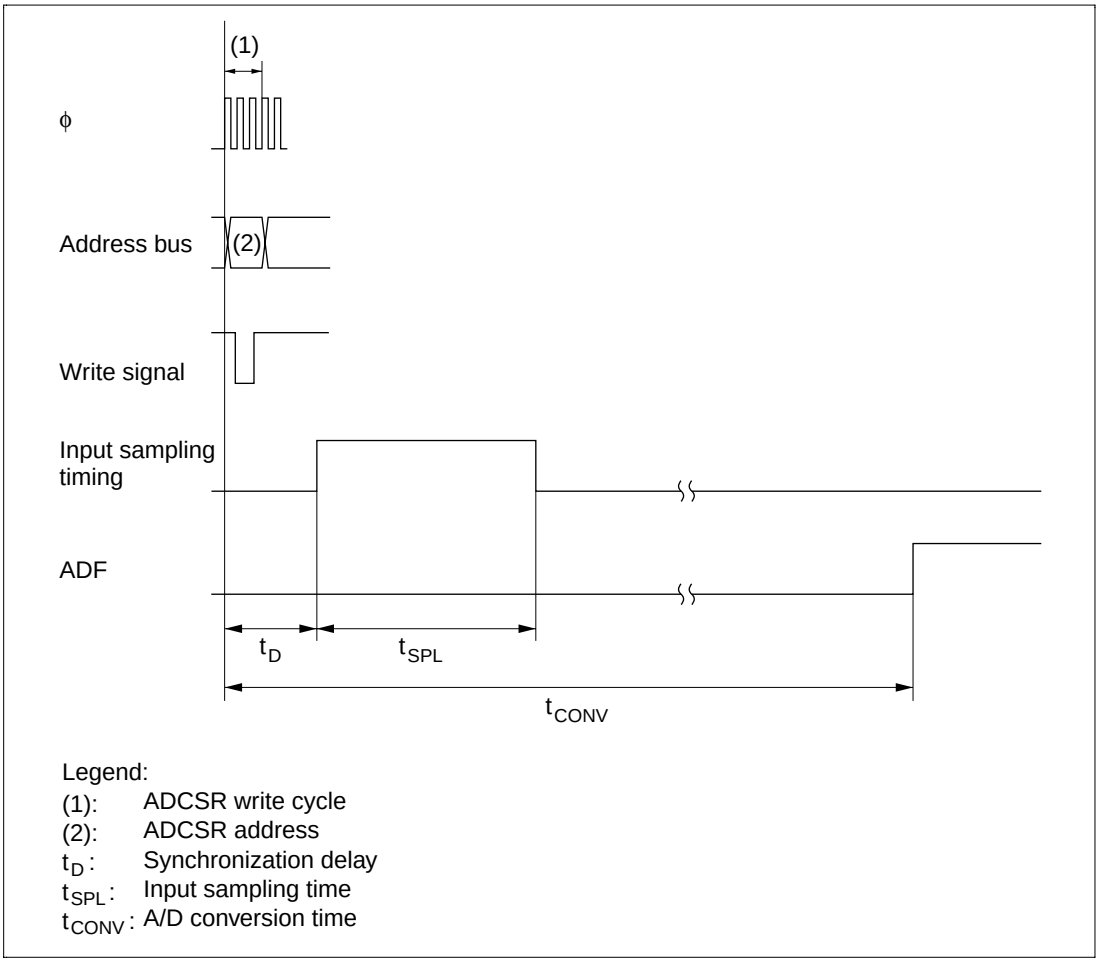


Figure 16.5 A/D Conversion Timing

Table 16.4 A/D Conversion Time (Single Mode)

		CKS = 0			CKS = 1		
	Symbol	Min	Typ	Max	Min	Typ	Max
Synchronization delay	t_D	6	—	9	4	—	5
Input sampling time	t_{SPL}	—	31	—	—	15	—
A/D conversion time	t_{CONV}	131	—	134	69	—	70

Note: Values in the table are numbers of states.

16.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGE bit is set to 1 in ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A rising or falling edge on the $\overline{\text{ADTRG}}$ input pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single and scan modes, are the same as if the ADST bit had been set to 1 by software. Figure 16.6 shows the timing.

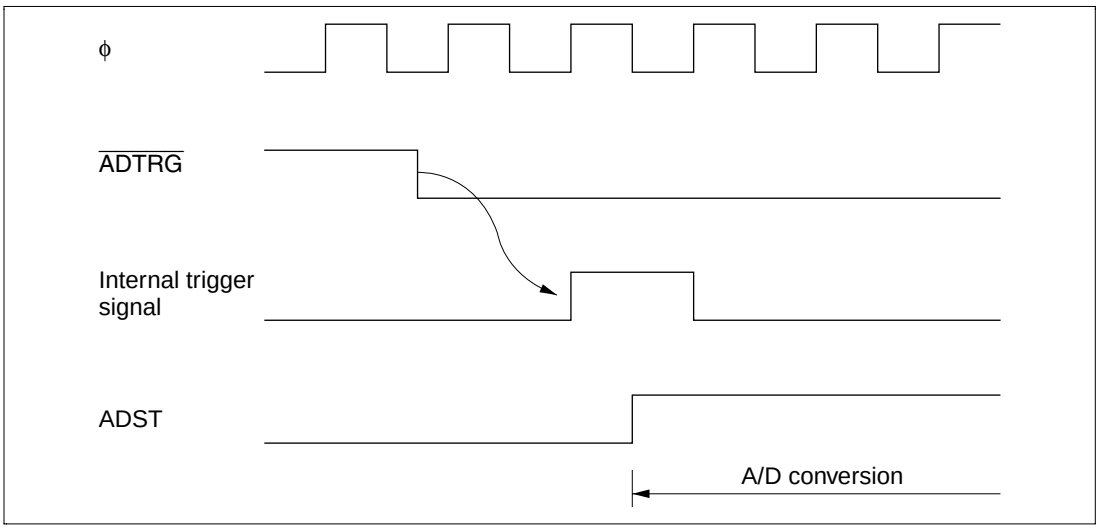


Figure 16.6 External Trigger Input Timing

16.5 Interrupts

The A/D converter generates an interrupt (ADI) at the end of A/D conversion. The ADI interrupt request can be enabled or disabled by the ADIE bit in ADCSR.

16.6 Usage Notes

When using the A/D converter, note the following points:

1. A/D Conversion Accuracy Definitions: A/D conversion accuracy in the H8/3664 Series is defined as follows:

   - Resolution
     Digital output code length of A/D converter
   - Offset error
     Deviation from ideal A/D conversion characteristic of analog input voltage required to raise digital output from minimum voltage value 0000000000 to 0000000001 (figure 16.8)
   - Full-scale error
     Deviation from ideal A/D conversion characteristic of analog input voltage required to raise digital output from 1111111110 to 1111111111 (figure 16.8)
   - Quantization error
     Intrinsic error of the A/D converter; 1/2 LSB (figure 16.7)
   - Nonlinearity error
     Deviation from ideal A/D conversion characteristic in range from zero volts to full scale, exclusive of offset error, full-scale error, and quantization error.
   - Absolute accuracy
     Deviation of digital value from analog input value, including offset error, full-scale error, quantization error, and nonlinearity error.

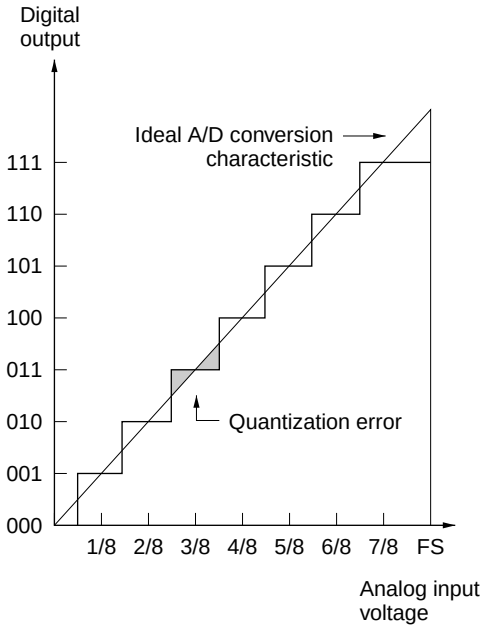


Figure 16.7 A/D Converter Accuracy Definitions (1)

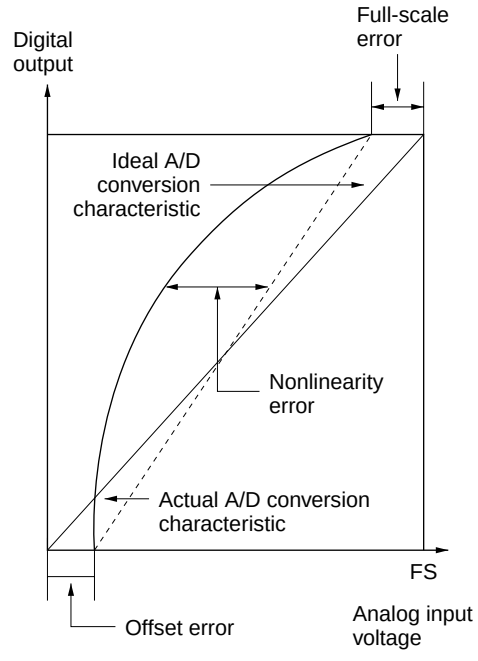


Figure 16.8 A/D Converter Accuracy Definitions (2)

2. Allowable Signal-Source Impedance: The analog inputs of the H8/3664 Series are designed to assure accurate conversion of input signals with a signal-source impedance not exceeding 10 kΩ. The reason for this rating is that it enables the input capacitor in the sample-and-hold circuit in the A/D converter to charge within the sampling time. If the sensor output impedance exceeds 10 kΩ, charging may be inadequate and the accuracy of A/D conversion cannot be guaranteed.

   If a large external capacitor is provided in single mode, then the internal 10-kΩ input resistance becomes the only significant load on the input. In this case the impedance of the signal source is not a problem.

   A large external capacitor, however, acts as a low-pass filter. This may make it impossible to track analog signals with high dv/dt (e.g. a variation of 5 mV/µs) (figure 16.9). To convert high-speed analog signals or to use scan mode, insert a low-impedance buffer.

3. Effect on Absolute Accuracy: Attaching an external capacitor creates a coupling with ground, so if there is noise on the ground line, it may degrade absolute accuracy. The capacitor must be connected to an electrically stable ground.

   If a filter circuit is used, be careful of interference with digital signals on the same board, and make sure the circuit does not act as an antenna.

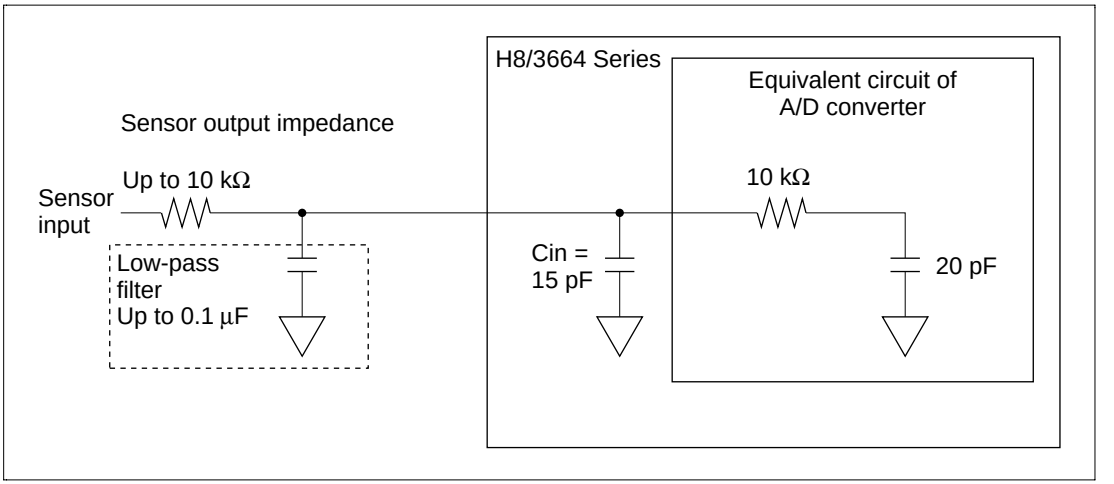


Figure 16.9 Analog Input Circuit (Example)

Section 17 Power Supply Circuit

17.1 Overview

The H8/3664 Series incorporates an internal power supply step-down circuit. Use of this circuit enables the internal power supply to be fixed at a constant level of approximately 3.0 V, independently of the voltage of the power supply connected to the external V_{CC} pin. As a result, the current consumed when an external power supply is used at 3.0 V or above can be held down to virtually the same low level as when used at approximately 3.0 V. If the external power supply is 3.0 V or below, the internal voltage will be practically the same as the external voltage. It is, of course, also possible to use the same level of external power supply voltage and internal power supply voltage without using the internal power supply step-down circuit.

17.2 When Using the Internal Power Supply Step-Down Circuit

Connect the external power supply to the V_{CC} pin, and connect a capacitance of approximately 0.1 µF between CV_{CC} and V_{SS}, as shown in figure 17.1. The internal step-down circuit is made effective simply by adding this external circuit.

Notes: 1. In the external circuit interface, the external power supply voltage connected to V_{CC} and the GND potential connected to V_{SS} are the reference levels. For example, for port input/output levels, the V_{CC} level is the reference for the high level, and the V_{SS} level is that for the low level.

2. The A/D converter analog power supply is not affected by internal step-down processing.

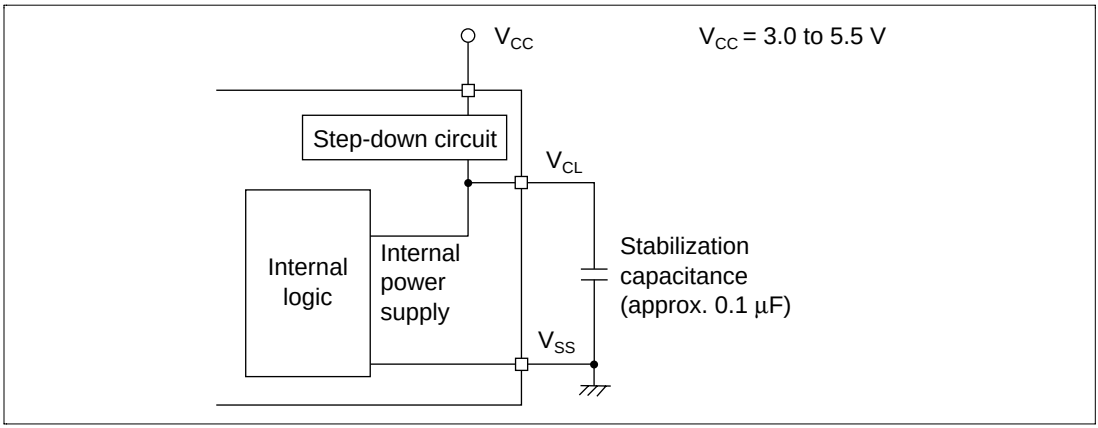


Figure 17.1 Power Supply Connection when Internal Step-Down Circuit Is Used

17.3 When Not Using the Internal Power Supply Step-Down Circuit

When the internal power supply step-down circuit is not used, connect the external power supply to the V_{CC} pin and CV_{CC} pin, as shown in figure 17.2. The external power supply is then input directly to the internal power supply.

Note: The permissible range for the power supply voltage is 3.0 V to 3.6 V. Operation cannot be guaranteed if a voltage outside this range (less than 3.0 V or more than 3.6 V) is input.

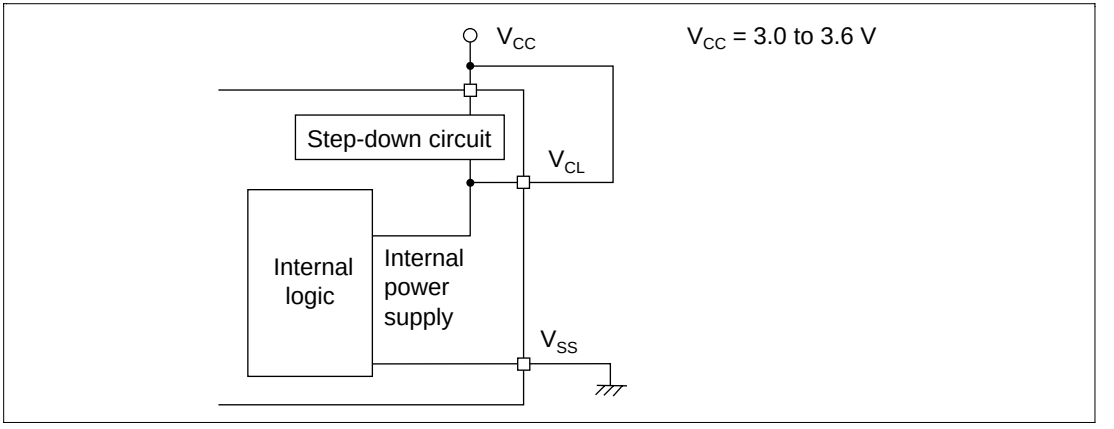


Figure 17.2 Power Supply Connection when Internal Step-Down Circuit Is Not Used

Section 18 Electrical Characteristics

18.1 Absolute Maximum Ratings

Table 18.1 lists the absolute maximum ratings.

Table 18.1 Absolute Maximum Ratings

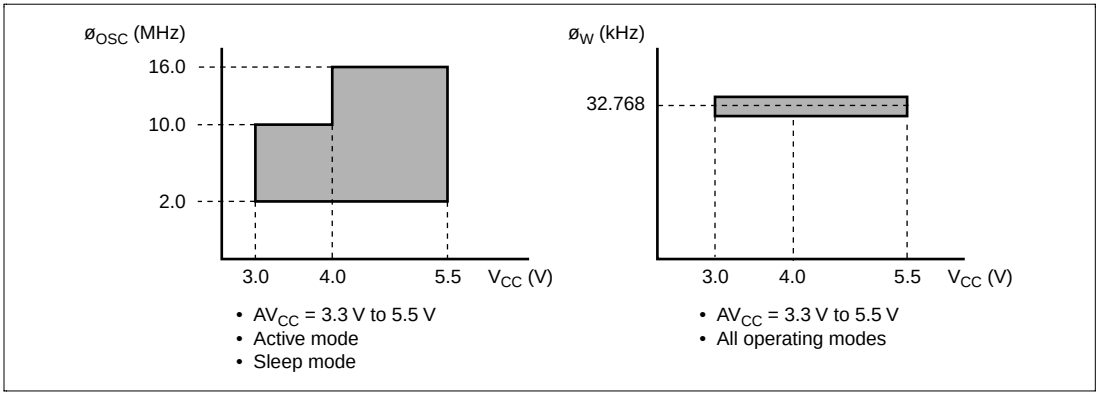
Item		Symbol	Value	Unit	Note
Power supply voltage		V_{CC}	–0.3 to +7.0	V	*
Analog power supply voltage		AV_{CC}	–0.3 to +7.0	V	*
Input voltage	Ports other than Port B and X1	V_{IN}	–0.3 to V_{CC} +0.3	V	*
	Port B		–0.3 to AV_{CC} +0.3	V	*
	X1		–0.3 to 4.3	V	*
Operating temperature		T_{opr}	–20 to +75	°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.

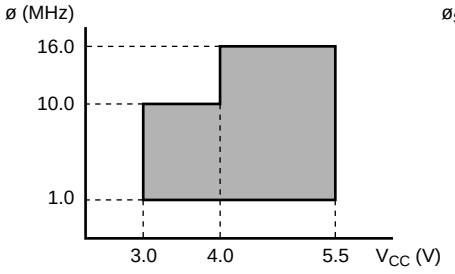
18.2 Electrical Characteristics (F-ZTAT™ Version)

18.2.1 Power Supply Voltage and Operating Ranges

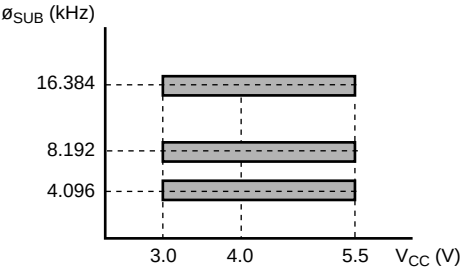
Power Supply Voltage and Oscillation Frequency Range



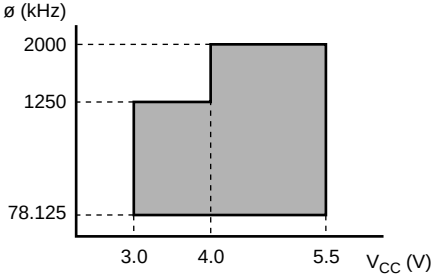
Power Supply Voltage and Operating Frequency Range



- AV_{CC} = 3.3 V to 5.5 V
- Active mode
- Sleep mode
  (When MA2 = 0 in SYSCR2)

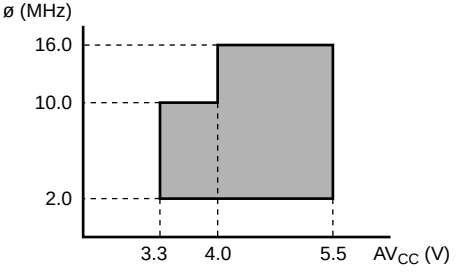


- AV_{CC} = 3.3 V to 5.5 V
- Subactive mode
- Subsleep mode



- AV_{CC} = 3.3 V to 5.5 V
- Active mode
- Sleep mode
  (When MA2 = 1 in SYSCR2)

Analog Power Supply Voltage and A/D Converter Accuracy Guarantee Range



- V_{CC} = 3.0 V to 5.5 V
- Active mode
- Sleep mode

18.2.2 DC Characteristics

Table 18.2 lists the DC characteristics.

Table 18.2 DC Characteristics (1)

V_{CC} = 3.0 V to 5.5 V, V_{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 high voltage	V_{IH}	$\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$,TMRIV,	0.8 V_{CC}	—	V_{CC} + 0.3	V	V_{CC} = 4.0 V to 5.5 V	
		TMCIV, FTCI, FTIOA to FTIOD, SCK3, TRGV	0.9 V_{CC}	—	V_{CC} + 0.3			
		RXD, SCL, SDA, P10 to P12, P14 to P17, P20 to P22,	0.7 V_{CC}	—	V_{CC} + 0.3	V	V_{CC} = 4.0 V to 5.5 V	
		P50 to P57, P74 to P76, P80 to P87	0.8 V_{CC}	—	V_{CC} + 0.3			
		PB0 to PB7	0.7 V_{CC}	—	AV_{CC} + 0.3	V	V_{CC} = 4.0 V to 5.5 V	
			0.8 V_{CC}	—	AV_{CC} + 0.3			
		OSC1	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			
Input low voltage	V_{IL}	$\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$,TMRIV,	–0.3	—	0.2 V_{CC}	V	V_{CC} = 4.0 V to 5.5 V	
		TMCIV, FTCI, FTIOA to FTIOD, SCK3, TRGV	–0.3	—	0.1 V_{CC}			
		RXD, SCL, SDA, P10 to P12, P14 to P17, P20 to P22,	–0.3	—	0.3 V_{CC}	V	V_{CC} = 4.0 V to 5.5 V	
		P50 to P57, P74 to P76, P80 to P87, PB0 to PB7	–0.3	—	0.2 V_{CC}			
		OSC1	–0.3	—	0.5	V	V_{CC} = 4.0 V to 5.5 V	
			–0.3	—	0.3			

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Output high voltage	V_{OH}	P10 to P12, P14 to P17, P20 to P22,	V_{CC} – 1.0	—	—	V	V_{CC} = 4.0 V to 5.5 V $-I_{OH}$ = 1.5 mA	
		P50 to P55, P74 to P76, P80 to P87,	V_{CC} – 0.5	—	—		$-I_{OH}$ = 0.1 mA	
		P56, P57	V_{CC} – 2.5	—	—	V	V_{CC} = 4.0 V to 5.5 V $-I_{OH}$ = 0.1 mA	
			V_{CC} – 2.0	—	—		V_{CC} = 3.0 V to 4.0 V $-I_{OH}$ = 0.1 mA	
Output low voltage	V_{OL}	P10 to P12, P14 to P17, P20 to P22,	—	—	0.6	V	V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	
		P50 to P57, P74 to P76,	—	—	0.4		I_{OL} = 0.4 mA	
		P80 to P87	—	—	1.5	V	V_{CC} = 4.0 V to 5.5 V I_{OL} = 20.0 mA	
			—	—	1.0		V_{CC} = 4.0 V to 5.5 V I_{OL} = 10.0 mA	
			—	—	0.4		V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	
			—	—	0.4		I_{OL} = 0.4 mA	
		SCL, SDA	—	—	0.6	V	V_{CC} = 4.0 V to 5.5 V I_{OL} = 6.0 mA	
			—	—	0.4		I_{OL} = 3.0 mA	
Input/ output leakage current	$\lvert I_{IL} \rvert$	OSC1, $\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$, TRGV, TMRIV, TMCIV, FTCI, FTIOA to FTIOD, RXD, SCK3, SCL, SDA	—	—	1.0	µA	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	
		P10 to P12, P14 to P17, P20 to P22, P50 to P57, P74 to P76, P80 to P87,	—	—	1.0	µA	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	
		PB0 to PB7	—	—	1.0	µA	V_{IN} = 0.5 V to (AV_{CC} – 0.5 V)	

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Pull-up MOS current	$-I_p$	P10 to P12, P14 to P17, P50 to P55	50.0	—	300.0	µA	V_{CC} = 5.0 V, V_{IN} = 0.0 V	
			—	60.0	—		V_{CC} = 3.0 V, V_{IN} = 0.0 V	Reference value
Input capacitance	C_{in}	All input pins except power supply pins	—	—	15.0	pF	f = 1 MHz, V_{IN} = 0.0 V, T_a = 25°C	
Active mode current dissipation	I_{OPE1}	V_{CC}	—	15.0	22.5	mA	Active mode 1 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	*
			—	8.0	—		Active mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	* Reference value
	I_{OPE2}	V_{CC}	—	1.8	2.7	mA	Active mode 2 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	*
			—	1.2	—		Active mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	* Reference value
Sleep mode current dissipation	I_{SLEEP1}	V_{CC}	—	11.5	17.0	mA	Sleep mode 1 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	*
			—	6.5	—		Sleep mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	* Reference value
	I_{SLEEP2}	V_{CC}	—	1.7	2.5	mA	Sleep mode 2 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	*
			—	1.1	—		Sleep mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	* Reference value
Subactive mode current dissipation	I_{SUB}	V_{CC}	—	35.0	70.0	µA	V_{CC} = 3.0 V 32-kHz crystal oscillator ($ø_{SUB} = ø_W/2$)	*
			—	25.0	—		V_{CC} = 3.0 V 32-kHz crystal oscillator ($ø_{SUB} = ø_W/8$)	* Reference value
Subsleep mode current dissipation	I_{SUBSP}	V_{CC}	—	25.0	50.0	µA	V_{CC} = 3.0 V 32-kHz crystal oscillator ($ø_{SUB} = ø_W/2$)	*

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Standby mode current dissipation	I_{STBY}	V_{CC}	—	—	5.0	µA	32-kHz crystal oscillator not used	*
RAM data retaining voltage	V_{RAM}	V_{CC}	2.0	—	—	V		

Note: * Pin states during current dissipation measurement are given below (excluding current in the pull-up MOS transistors and output buffers).

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pins
Active mode 1	V_{CC}	Operates	V_{CC}	System clock oscillator: ceramic or crystal
Active mode 2		Operates (ø/64)		Subclock oscillator: Pin X1 = V_{SS}
Sleep mode 1	V_{CC}	Only timers operate	V_{CC}	
Sleep mode 2		Only timers operate (ø/64)		
Subactive mode	V_{CC}	Operates	V_{CC}	System clock oscillator: ceramic or crystal
Subsleep mode	V_{CC}	Only time base operates	V_{CC}	Subclock oscillator: crystal
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	System clock oscillator: ceramic or crystal
				Subclock oscillator: Pin X1 = V_{SS}

Table 18.2 DC Characteristics (2)

V_{CC} = 3.0 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise indicated.

Item		Symbol	Values Min	Typ	Max	Unit	Test Condition
Allowable output low current (per pin)	Output pins except port 8, SCL and SDA	I_{OL}	—	—	2.0	mA	V_{CC} = 4.0 V to 5.5 V
	Port 8		—	—	20.0		
	SCL and SDA		—	—	6.0		
	Port 8		—	—	10.0		
	Output pins except port 8, SCL and SDA		—	—	0.5		
Allowable output low current (total)	Output pins except port 8, SCL and SDA	ΣI_{OL}	—	—	40.0	mA	V_{CC} = 4.0 V to 5.5 V
	Port 8, SCL and SDA		—	—	80.0		
	Output pins except port 8, SCL and SDA		—	—	20.0		
	Port 8, SCL and SDA		—	—	40.0		
Allowable output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0	mA	V_{CC} = 4.0 V to 5.5 V
			—	—	0.2		
Allowable output high current (total)	All output pins	$\Sigma(-I_{OH})$	—	—	30.0	mA	V_{CC} = 4.0 V to 5.5 V
			—	—	8.0		

18.2.3 AC Characteristics

Table 18.3 lists the AC characteristics. Tables 18.4 and 18.5 list the PC bus interface timing and the serial interface timing, respectively.

Table 18.3 AC Characteristics

V_{CC} = 3.0 V to 5.5 V, V_{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}	OSC1, OSC2	2.0	—	16.0	MHz	V_{CC} = 4.0 V to 5.5 V	*1
			2.0		10.0			
System clock (ø) cycle time	t_{cyc}		1	—	64	t_{OSC}		*2
			—	—	12.8	µs		
Subclock oscillation frequency	f_W	X1, X2	—	32.768	—	kHz		
Watch clock ($ø_W$) cycle time	t_W	X1, X2	—	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}	OSC1, OSC2	—	—	10.0	ms		
Oscillation stabilization time (ceramic oscillator)	t_{rc}	OSC1, OSC2	—	—	5.0	ms		
Oscillation stabilization time	t_{rcx}	X1, X2	—	—	2.0	s		
External clock high width	t_{CPH}	OSC1	25.0	—	—	ns	V_{CC} = 4.0 V to 5.5 V	Figure 18.1
			40.0	—	—			
External clock low width	t_{CPL}	OSC1	25.0	—	—	ns	V_{CC} = 4.0 V to 5.5 V	
			40.0	—	—			
External clock rise time	t_{CPr}	OSC1	—	—	10.0	ns	V_{CC} = 4.0 V to 5.5 V	
			—	—	15.0			
External clock fall time	t_{CPf}	OSC1	—	—	10.0	ns	V_{CC} = 4.0 V to 5.5 V	
			—	—	15.0			

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Reference Figure
$\overline{\text{RES}}$ pin low width	t_{REL}	$\overline{\text{RES}}$	t_{rc}	—	—	ms	At power-on and in modes other than those below	Figure 18.2
			10	—	—	t_{cyc}	In active mode and sleep mode operation	
Input pin high width	t_{IH}	$\overline{\text{NMI}}$, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, FTCI, FTIOA to FTIOD	2	—	—	t_{cyc} t_{subcyc}		Figure 18.3
Input pin low width	t_{IL}	$\overline{\text{NMI}}$, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, FTIOA to FTIOD	2	—	—	t_{cyc} t_{subcyc}		

Notes: 1. When an external clock is input, the minimum system clock oscillator frequency is 1.0 MHz.

2. Determined by MA2, MA1, MA0, SA1, and SA0 of system control register 2 (SYSCR2).

Table 18.4 I^2C Bus Interface Timing

V_{CC} = 3.0 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20 to +75°C, unless otherwise specified.

Item	Symbol	Values			Unit	Test Condition	Reference Figure
		Min	Typ	Max			
SCL input cycle time	t_{SCL}	$12t_{cyc}$ + 600	—	—	ns		Figure 18.4
SCL input high width	t_{SCLH}	$3t_{cyc}$ + 300	—	—	ns		
SCL input low width	t_{SCLL}	$5t_{cyc}$ + 300	—	—	ns		
Input fall time of SCL and SDA	t_{Sf}	—	—	300	ns		
SCL and SDA input spike pulse removal time	t_{SP}	—	—	$1t_{cyc}$	ns		
SDA input bus-free time	t_{BUF}	$5t_{cyc}$	—	—	ns		
Start condition input hold time	t_{STAH}	$3t_{cyc}$	—	—	ns		
Retransmission start condition input setup time	t_{STAS}	$3t_{cyc}$	—	—	ns		
Setup time for stop condition input	t_{STOS}	$3t_{cyc}$	—	—	ns		
Data-input setup time	t_{SDAS}	$0.5t_{cyc}$	—	—	ns		
Data-input hold time	t_{SDAH}	0	—	—	ns		
Capacitive load of SCL and SDA	c_b	0	—	400	pF		
SCL and SDA output fall time	t_{Sf}	—	—	250	ns	V_{CC} = 4.0 V to 5.5 V	
		—	—	300			

Note: * The value can be changed to $17.5t_{cyc}$, depending on the clock used in the I^2C module.

Table 18.5 Serial Interface (SCI3) Timing

V_{CC} = 3.0 V to 5.5 V, V_{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 clock cycle	Asynchronous	t_{Scyc}	SCK3	4	—	—	t_{cyc}		Figure 18.5
	Synchronous			6	—	—			
Input clock pulse width		t_{SCKW}	SCK3	0.4	—	0.6	t_{Scyc}		Figure 18.5
Transmit data delay time (synchronous)		t_{TXD}	TXD	—	—	1	t_{cyc}	V_{CC} = 4.0 V to 5.5 V	Figure 18.6
				—	—	1			
Receive data setup time (synchronous)		t_{RXS}	RXD	62.5	—	—	ns	V_{CC} = 4.0 V to 5.5 V	Figure 18.6
				100.0	—	—			
Receive data hold time (synchronous)		t_{RXH}	RXD	62.5	—	—	ns	V_{CC} = 4.0 V to 5.5 V	Figure 18.6
				100.0	—	—			

18.2.4 A/D Converter Characteristics

Table 18.6 A/D Converter Characteristics

V_{CC} = 3.0 V to 5.5 V, V_{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
Analog power supply voltage	AV_{CC}	AV_{CC}	3.3	V_{CC}	5.5	V		*1
Analog input voltage	AV_{IN}	AN0 to AN7	V_{SS} – 0.3	—	AV_{CC} + 0.3	V		
Analog power supply current	AI_{OPE}	AV_{CC}	—	—	2.0	mA	AV_{CC} = 5.0 V f_{OSC} = 16 MHz	
	AI_{STOP1}	AV_{CC}	—	50	—	µA		*2 Reference value
	AI_{STOP2}	AV_{CC}	—	—	5.0	µA		*3
Analog input capacitance	C_{AIN}	AN0 to AN7	—	—	30.0	pF		
Allowable signal source impedance	R_{AIN}	AN0 to AN7	—	—	5.0	kΩ		
Resolution			10	10	10	bit		
Conversion time (single mode)			134	—	—	t_{cyc}	AV_{CC} = 3.3 V to 5.5 V	
Nonlinearity error			—	—	±7.5	LSB		
Offset error			—	—	±7.5	LSB		
Full-scale error			—	—	±7.5	LSB		
Quantization error			—	—	±0.5	LSB		
Absolute accuracy			—	—	±8.0	LSB		
Conversion time (single mode)			70	—	—	t_{cyc}	AV_{CC} = 4.0 V to 5.5 V	
Nonlinearity error			—	—	±7.5	LSB		
Offset error			—	—	±7.5	LSB		
Full-scale error			—	—	±7.5	LSB		
Quantization error			—	—	±0.5	LSB		
Absolute accuracy			—	—	±8.0	LSB		

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Reference Figure
Conversion time (single mode)			134	—	—	t_{cyc}	AV_{CC} = 4.0 V to 5.5 V	
Nonlinearity error			—	—	±3.5	LSB		
Offset error			—	—	±3.5	LSB		
Full-scale error			—	—	±3.5	LSB		
Quantization error			—	—	±0.5	LSB		
Absolute accuracy			—	—	±4.0	LSB		

Notes: 1. Set $AV_{CC} = V_{CC}$ when the A/D converter is not used.
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, subactive, and subsleep modes while the A/D converter is idle.

18.2.5 Watchdog Timer

Table 18.7 Watchdog Timer Characteristics

V_{CC} = 3.0 V to 5.5 V, V_{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
On-chip oscillator overflow time	t_{OVF}		0.2	0.4	—	s		*

Note: * Shows the time to count from 0 to 255, at which point an internal reset is generated, when the internal oscillator is selected.

18.2.6 Flash Memory Characteristics (Preliminary)

Table 18.8 Flash Memory Characteristics (Preliminary)

V_{CC} = 3.0 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item		Symbol	Test Condition	Values Min	Typ	Max	Unit	Notes
Programming time (per 128 bytes)		t_P		—	7	—	ms	1, 2, 4
Erase time (per block)		t_E		—	100	—	ms	1, 3, 6
Reprogramming count		N_{WEC}		—	—	100	Times	
Programming	Wait time after SWE bit setting	x		1	—	—	µs	1
	Wait time after PSU bit setting	y		50	—	—	µs	1
	Wait time after P bit setting	z1	1 ≤ n ≤ 6	28	30	32	µs	1, 4
		z2	7 ≤ n ≤ 1000	198	200	202	µs	
		z3	Additional-programming	8	10	12	µs	
	Wait time after P bit clear	α		5	—	—	µs	1
	Wait time after PSU bit clear	β		5	—	—	µs	1
	Wait time after PV bit setting	γ		4	—	—	µs	1
	Wait time after dummy write	ε		2	—	—	µs	1
	Wait time after PV bit clear	η		2	—	—	µs	1
	Wait time after SWE bit clear	θ		100	—	—	µs	1
	Maximum programming count	N		—	—	1000	Times	1, 4, 5

Item		Symbol	Test Condition	Values Min	Typ	Max	Unit	Notes
Erase	Wait time after SWE bit setting	x		1	—	—	µs	1
	Wait time after ESU bit setting	y		100	—	—	µs	1
	Wait time after E bit setting	z		10	—	100	ms	1, 6
	Wait time after E bit clear	α		10	—	—	µs	1
	Wait time after ESU bit clear	β		10	—	—	µs	1
	Wait time after EV bit setting	γ		20	—	—	µs	1
	Wait time after dummy write	ε		2	—	—	µs	1
	Wait time after EV bit clear	η		4	—	—	µs	1
	Wait time after SWE bit clear	θ		100	—	—	µs	1
	Maximum erase count	N		—	—	120	Times	1, 6, 7

Notes:
1. Make the time settings in accordance with the program/erase algorithms.
2. The programming time for 128 bytes. (Indicates the total time for which the P bit in flash memory control register 1 (FLMCR1) is set. The program-verify time is not included.)
3. The time required to erase one block. (Indicates the time for which the E bit in flash memory control register 1 (FLMCR1) is set. The erase-verify time is not included.)
4. Programming time maximum value (t_P(MAX)) = wait time after P bit setting (z) × maximum number of writes (N)
5. Set the maximum number of writes (N) according to the actual set values of z1, z2, and z3, so that it does not exceed the programming time maximum value (t_P(MAX)). The wait time after P bit setting (z1, z2) should be changed as follows according to the value of the number of writes (n).

   Number of writes (n)

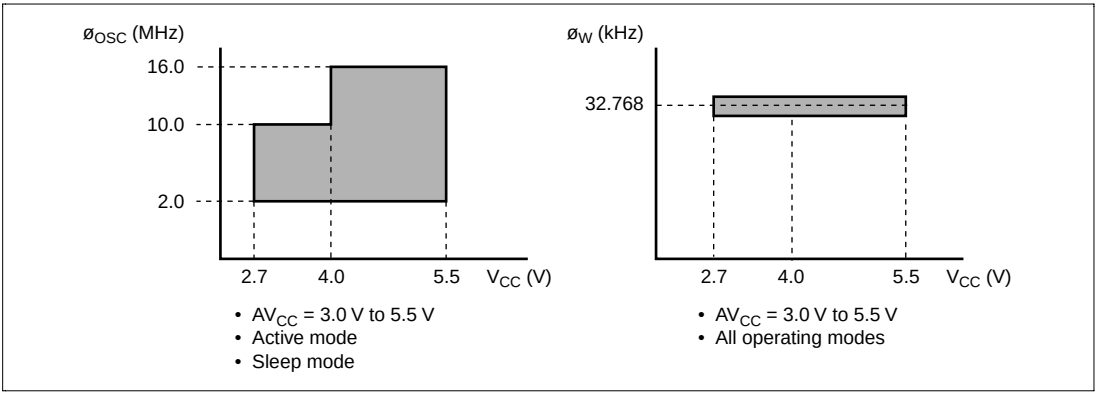
   $1 \leq n \leq 6$ z1 = 30 µs

   $7 \leq n \leq 1000$ z2 = 200 µs
6. Erase time maximum value (t_E(max)) = wait time after E bit setting (z) × maximum number of erases (N)
7. Set the maximum number of erases (N) according to the actual set value of (z), so that it does not exceed the erase time maximum value (t_E(max)).

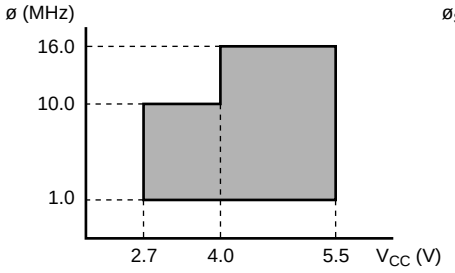
18.3 Electrical Characteristics (Mask ROM Version)

18.3.1 Power Supply Voltage and Operating Ranges

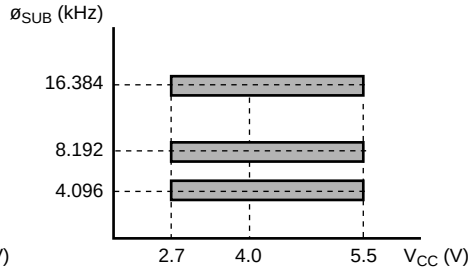
Power Supply Voltage and Oscillation Frequency Range



Power Supply Voltage and Operating Frequency Range



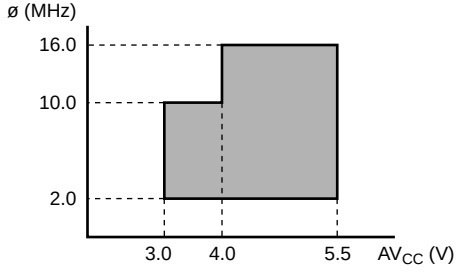
- AV_{CC} = 3.0 V to 5.5 V
- Active mode
- Sleep mode
  (When MA2 = 0 in SYSCR2)



- AV_{CC} = 3.0 V to 5.5 V
- Subactive mode
- Subsleep mode

- AV_{CC} = 3.0 V to 5.5 V
- Active mode
- Sleep mode
  (When MA2 = 1 in SYSCR2)

Analog Power Supply Voltage and A/D Converter Accuracy Guarantee Range



- AV_{CC} = 2.7 V to 5.5 V
- V_{CC} = 3.0 V to 5.5 V
- Active mode
- Sleep mode

18.3.2 DC Characteristics

Table 18.9 lists the DC characteristics.

Table 18.9 DC Characteristics (1)

V_{CC} = 2.7 V to 5.5 V, V_{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}	$\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$,TMRIV,	0.8 V_{CC}	—	V_{CC} + 0.3	V	V_{CC} = 4.0 V to 5.5 V	
		TMCIV, FTCI, FTIOA to FTIOD, SCK3, TRGV	0.9 V_{CC}	—	V_{CC} + 0.3			
		RXD, SCL, SDA, P10 to P12, P14 to P17, P20 to P22,	0.7 V_{CC}	—	V_{CC} + 0.3	V	V_{CC} = 4.0 V to 5.5 V	
		P50 to P57, P74 to P76, P80 to P87	0.8 V_{CC}	—	V_{CC} + 0.3			
		PB0 to PB7	0.7 V_{CC}	—	AV_{CC} + 0.3	V	V_{CC} = 4.0 V to 5.5 V	
			0.8 V_{CC}	—	AV_{CC} + 0.3			
		OSC1	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			
Input low voltage	V_{IL}	$\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$,TMRIV,	–0.3	—	0.2 V_{CC}	V	V_{CC} = 4.0 V to 5.5 V	
		TMCIV, FTCI, FTIOA to FTIOD, SCK3, TRGV	–0.3	—	0.1 V_{CC}			
		RXD, SCL, SDA, P10 to P12, P14 to P17, P20 to P22,	–0.3	—	0.3 V_{CC}	V	V_{CC} = 4.0 V to 5.5 V	
		P50 to P57, P74 to P76, P80 to P87, PB0 to PB7	–0.3	—	0.2 V_{CC}			
		OSC1	–0.3	—	0.5	V	V_{CC} = 4.0 V to 5.5 V	
			–0.3	—	0.3			

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Output high voltage	V_{OH}	P10 to P12, P14 to P17, P20 to P22, P50 to P55, P74 to P76, P80 to P87,	V_{CC} – 1.0	—	—	V	V_{CC} = 4.0 V to 5.5 V $-I_{OH}$ = 1.5 mA	
			V_{CC} – 0.5	—	—		$-I_{OH}$ = 0.1 mA	
		P56, P57	V_{CC} – 2.5	—	—	V	V_{CC} = 4.0 V to 5.5 V $-I_{OH}$ = 0.1 mA	
			V_{CC} – 2.0	—	—		V_{CC} = 3.0 V to 4.0 V $-I_{OH}$ = 0.1 mA	
Output low voltage	V_{OL}	P10 to P12, P14 to P17, P20 to P22, P50 to P57, P74 to P76,	—	—	0.6	V	V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	
			—	—	0.4		I_{OL} = 0.4 mA	
		P80 to P87	—	—	1.5	V	V_{CC} = 4.0 V to 5.5 V I_{OL} = 20.0 mA	
			—	—	1.0		V_{CC} = 4.0 V to 5.5 V I_{OL} = 10.0 mA	
			—	—	0.4		V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	
			—	—	0.4		I_{OL} = 0.4 mA	
		SCL, SDA	—	—	0.6	V	V_{CC} = 4.0 V to 5.5 V I_{OL} = 6.0 mA	
			—	—	0.4		I_{OL} = 3.0 mA	
Input/ output leakage current	$\lvert I_{IL} \rvert$	OSC1, $\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$, TRGV, TMRIV, TMCIV, FTCI, FTIOA to FTIOD, RXD, SCK3, SCL, SDA	—	—	1.0	µA	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	
		P10 to P12, P14 to P17, P20 to P22, P50 to P57, P74 to P76, P80 to P87,	—	—	1.0	µA	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	
		PB0 to PB7	—	—	1.0	µA	V_{IN} = 0.5 V to (AV_{CC} – 0.5 V)	

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Pull-up MOS current	$-I_p$	P10 to P12, P14 to P17, P50 to P55	50.0	—	300.0	µA	V_{CC} = 5.0 V, V_{IN} = 0.0 V	
			—	60.0	—		V_{CC} = 3.0 V, V_{IN} = 0.0 V	Reference value
Input capacitance	C_{in}	All input pins except power supply pins	—	—	15.0	pF	f = 1 MHz, V_{IN} = 0.0 V, T_a = 25°C	
Active mode current dissipation	I_{OPE1}	V_{CC}	—	15.0	22.5	mA	Active mode 1 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	*
			—	8.0	—		Active mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	* Reference value
	I_{OPE2}	V_{CC}	—	1.8	2.7	mA	Active mode 2 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	*
			—	1.2	—		Active mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	* Reference value
Sleep mode current dissipation	I_{SLEEP1}	V_{CC}	—	7.1	13.0	mA	Sleep mode 1 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	*
			—	4.0	—		Sleep mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	* Reference value
	I_{SLEEP2}	V_{CC}	—	1.1	2.0	mA	Sleep mode 2 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	*
			—	0.5	—		Sleep mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	* Reference value
Subactive mode current dissipation	I_{SUB}	V_{CC}	—	35.0	70.0	µA	V_{CC} = 3.0 V 32-kHz crystal oscillator ($ø_{SUB} = ø_W/2$)	*
			—	25.0	—		V_{CC} = 3.0 V 32-kHz crystal oscillator ($ø_{SUB} = ø_W/8$)	* Reference value
Subsleep mode current dissipation	I_{SUBSP}	V_{CC}	—	25.0	50.0	µA	V_{CC} = 3.0 V 32-kHz crystal oscillator ($ø_{SUB} = ø_W/2$)	*

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Standby mode current dissipation	I_{STBY}	V_{CC}	—	—	5.0	µA	32-kHz crystal oscillator not used	*
RAM data retaining voltage	V_{RAM}	V_{CC}	2.0	—	—	V		

Note: * Pin states during current dissipation measurement are given below (excluding current in the pull-up MOS transistors and output buffers).

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pins
Active mode 1	V_{CC}	Operates	V_{CC}	System clock oscillator: ceramic or crystal
Active mode 2		Operates (ø/64)		Subclock oscillator: Pin X1 = V_{SS}
Sleep mode 1	V_{CC}	Only timers operate	V_{CC}	
Sleep mode 2		Only timers operate (ø/64)		
Subactive mode	V_{CC}	Operates	V_{CC}	System clock oscillator: ceramic or crystal
Subsleep mode	V_{CC}	Only time base operates	V_{CC}	Subclock oscillator: crystal
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	System clock oscillator: ceramic or crystal Subclock oscillator: Pin X1 = V_{SS}

Table 18.9 DC Characteristics (2)

V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise indicated.

Item		Symbol	Values Min	Typ	Max	Unit	Test Condition
Allowable output low current (per pin)	Output pins except port 8, SCL and SDA	I_{OL}	—	—	2.0	mA	V_{CC} = 4.0 V to 5.5 V
	Port 8		—	—	20.0		
	SCL and SDA		—	—	6.0		
	Port 8		—	—	10.0		
	Output pins except port 8, SCL and SDA		—	—	0.5		
Allowable output low current (total)	Output pins except port 8, SCL and SDA	ΣI_{OL}	—	—	40.0	mA	V_{CC} = 4.0 V to 5.5 V
	Port 8, SCL and SDA		—	—	80.0		
	Output pins except port 8, SCL and SDA		—	—	20.0		
	Port 8, SCL and SDA		—	—	40.0		
Allowable output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0	mA	V_{CC} = 4.0 V to 5.5 V
			—	—	0.2		
Allowable output high current (total)	All output pins	$\Sigma(-I_{OH})$	—	—	30.0	mA	V_{CC} = 4.0 V to 5.5 V
			—	—	8.0		

18.3.3 AC Characteristics

Table 18.10 lists the AC characteristics. Tables 18.11 and 18.12 list the PC bus interface timing and the serial interface timing, respectively.

Table 18.10 AC Characteristics

V_{CC} = 2.7 V to 5.5 V, V_{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}	OSC1, OSC2	2.0	—	16.0	MHz	V_{CC} = 4.0 V to 5.5 V	*1
			2.0		10.0			
System clock (ø) cycle time	t_{cyc}		1	—	64	t_{OSC}		*2
			—	—	12.8	µs		
Subclock oscillation frequency	f_W	X1, X2	—	32.768	—	kHz		
Watch clock ($ø_W$) cycle time	t_W	X1, X2	—	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}	OSC1, OSC2	—	—	10.0	ms		
Oscillation stabilization time (ceramic oscillator)	t_{rc}	OSC1, OSC2	—	—	5.0	ms		
Oscillation stabilization time	t_{rcx}	X1, X2	—	—	2.0	s		
External clock high width	t_{CPH}	OSC1	25.0	—	—	ns	V_{CC} = 4.0 V to 5.5 V	Figure 18.1
			40.0	—	—			
External clock low width	t_{CPL}	OSC1	25.0	—	—	ns	V_{CC} = 4.0 V to 5.5 V	
			40.0	—	—			
External clock rise time	t_{CPr}	OSC1	—	—	10.0	ns	V_{CC} = 4.0 V to 5.5 V	
			—	—	15.0			
External clock fall time	t_{CPf}	OSC1	—	—	10.0	ns	V_{CC} = 4.0 V to 5.5 V	
			—	—	15.0			

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Reference Figure
$\overline{\text{RES}}$ pin low width	t_{REL}	$\overline{\text{RES}}$	t_{rc}	—	—	ms	At power-on and in modes other than those below	Figure 18.2
			10	—	—	t_{cyc}	In active mode and sleep mode operation	
Input pin high width	t_{IH}	$\overline{\text{NMI}}$, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, FTCI, FTIOA to FTIOD	2	—	—	t_{cyc} t_{subcyc}		Figure 18.3
Input pin low width	t_{IL}	$\overline{\text{NMI}}$, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, FTIOA to FTIOD	2	—	—	t_{cyc} t_{subcyc}		

Notes: 1. When an external clock is input, the minimum system clock oscillator frequency is 1.0 MHz.

2. Determined by MA2, MA1, MA0, SA1, and SA0 of system control register 2 (SYSCR2).

Table 18.11 I^2C Bus Interface Timing

V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20 to +75°C, unless otherwise specified.

Item	Symbol	Values			Unit	Test Condition	Reference Figure
		Min	Typ	Max			
SCL input cycle time	t_{SCL}	$12t_{cyc}$ + 600	—	—	ns		Figure 18.4
SCL input high width	t_{SCLH}	$3t_{cyc}$ + 300	—	—	ns		
SCL input low width	t_{SCLL}	$5t_{cyc}$ + 300	—	—	ns		
Input fall time of SCL and SDA	t_{Sf}	—	—	300	ns		
SCL and SDA input spike pulse removal time	t_{SP}	—	—	$1t_{cyc}$	ns		
SDA input bus-free time	t_{BUF}	$5t_{cyc}$	—	—	ns		
Start condition input hold time	t_{STAH}	$3t_{cyc}$	—	—	ns		
Retransmission start condition input setup time	t_{STAS}	$3t_{cyc}$	—	—	ns		
Setup time for stop condition input	t_{STOS}	$3t_{cyc}$	—	—	ns		
Data-input setup time	t_{SDAS}	$0.5t_{cyc}$	—	—	ns		
Data-input hold time	t_{SDAH}	0	—	—	ns		
Capacitive load of SCL and SDA	c_b	0	—	400	pF		
SCL and SDA output fall time	t_{Sf}	—	—	250	ns	V_{CC} = 4.0 V to 5.5 V	
		—	—	300			

Note: * The value can be changed to $17.5t_{cyc}$, depending on the clock used in the I^2C module.

Table 18.12 Serial Interface (SCI3) Timing

V_{CC} = 2.7 V to 5.5 V, V_{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 clock cycle	Asynchronous	t_{Scyc}	SCK3	4	—	—	t_{cyc}		Figure 18.5
	Synchronous			6	—	—			
Input clock pulse width		t_{SCKW}	SCK3	0.4	—	0.6	t_{Scyc}		Figure 18.5
Transmit data delay time (synchronous)		t_{TXD}	TXD	—	—	1	t_{cyc}	V_{CC} = 4.0 V to 5.5 V	Figure 18.6
				—	—	1			
Receive data setup time (synchronous)		t_{RXS}	RXD	62.5	—	—	ns	V_{CC} = 4.0 V to 5.5 V	Figure 18.6
				100.0	—	—			
Receive data hold time (synchronous)		t_{RXH}	RXD	62.5	—	—	ns	V_{CC} = 4.0 V to 5.5 V	Figure 18.6
				100.0	—	—			

18.3.4 A/D Converter Characteristics

Table 18.13 A/D Converter Characteristics

V_{CC} = 2.7 V to 5.5 V, V_{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
Analog power supply voltage	AV_{CC}	AV_{CC}	3.0	V_{CC}	5.5	V		*1
Analog input voltage	AV_{IN}	AN0 to AN7	V_{SS} – 0.3	—	AV_{CC} + 0.3	V		
Analog power supply current	AI_{OPE}	AV_{CC}	—	—	2.0	mA	AV_{CC} = 5.0 V f_{OSC} = 16 MHz	
	AI_{STOP1}	AV_{CC}	—	50	—	µA		*2 Reference value
	AI_{STOP2}	AV_{CC}	—	—	5.0	µA		*3
Analog input capacitance	C_{AIN}	AN0 to AN7	—	—	30.0	pF		
Allowable signal source impedance	R_{AIN}	AN0 to AN7	—	—	5.0	kΩ		
Resolution			10	10	10	bit		
Conversion time (single mode)			134	—	—	t_{cyc}	AV_{CC} = 3.0 V to 5.5 V	
Nonlinearity error			—	—	±7.5	LSB		
Offset error			—	—	±7.5	LSB		
Full-scale error			—	—	±7.5	LSB		
Quantization error			—	—	±0.5	LSB		
Absolute accuracy			—	—	±8.0	LSB		
Conversion time (single mode)			70	—	—	t_{cyc}	AV_{CC} = 4.0 V to 5.5 V	
Nonlinearity error			—	—	±7.5	LSB		
Offset error			—	—	±7.5	LSB		
Full-scale error			—	—	±7.5	LSB		
Quantization error			—	—	±0.5	LSB		
Absolute accuracy			—	—	±8.0	LSB		

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Reference Figure
Conversion time (single mode)			134	—	—	t_{cyc}	AV_{CC} = 4.0 V to 5.5 V	
Nonlinearity error			—	—	±3.5	LSB		
Offset error			—	—	±3.5	LSB		
Full-scale error			—	—	±3.5	LSB		
Quantization error			—	—	±0.5	LSB		
Absolute accuracy			—	—	±4.0	LSB		

Notes: 1. Set $AV_{CC} = V_{CC}$ when the A/D converter is not used.
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, subactive, and subsleep modes while the A/D converter is idle.

18.3.5 Watchdog Timer

Table 18.14 Watchdog Timer Characteristics

V_{CC} = 2.7 V to 5.5 V, V_{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
On-chip oscillator overflow time	t_{OVF}		0.2	0.4	—	s		*

Note: * Shows the time to count from 0 to 255, at which point an internal reset is generated, when the internal oscillator is selected.

18.4 Operation Timing

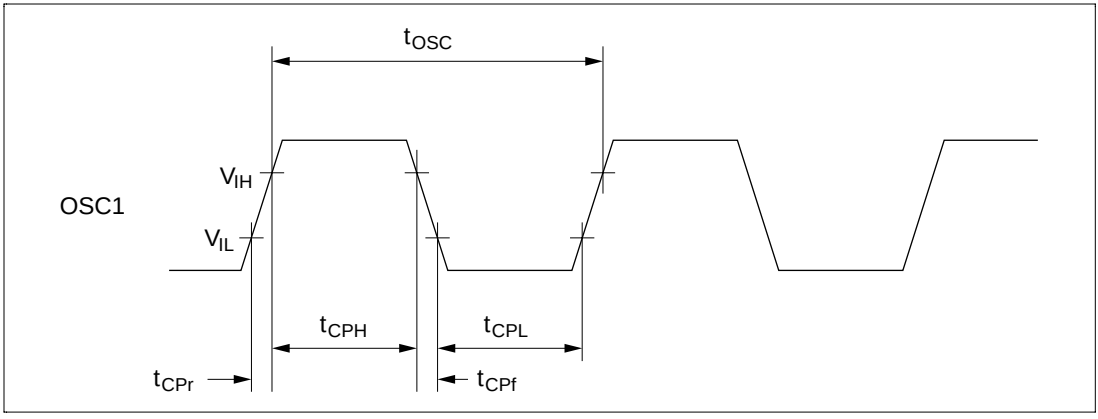


Figure 18.1 System Clock Input Timing

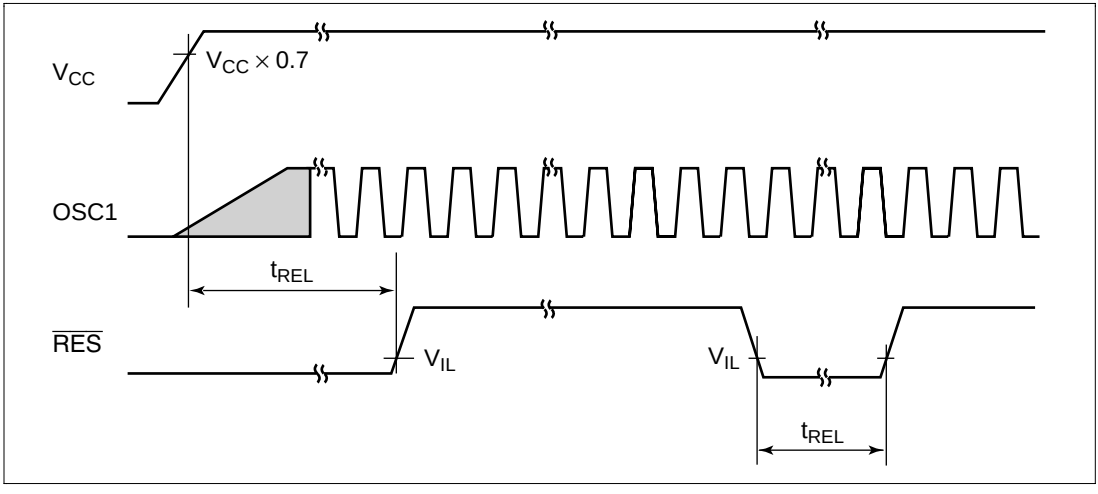


Figure 18.2 $\overline{RES}$ Low Width Timing

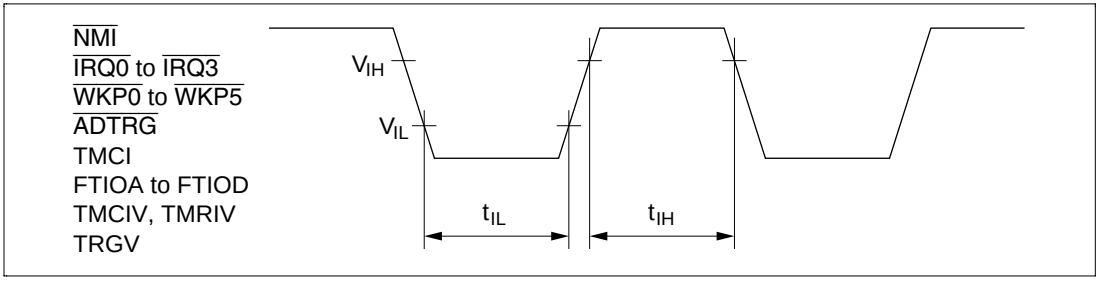


Figure 18.3 Input Timing

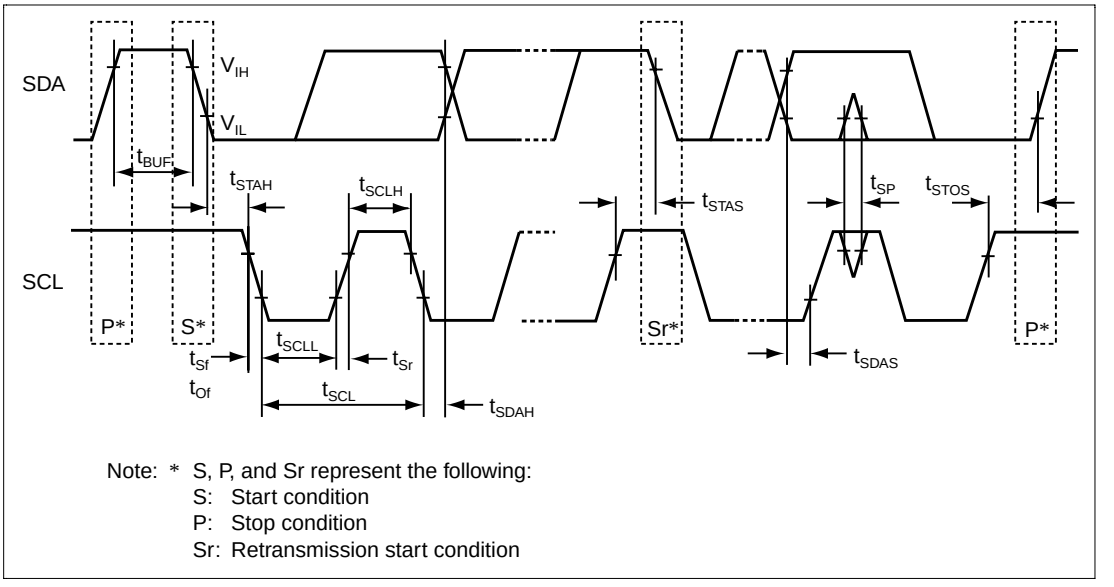


Figure 18.4 I²C Bus Interface Input/Output Timing

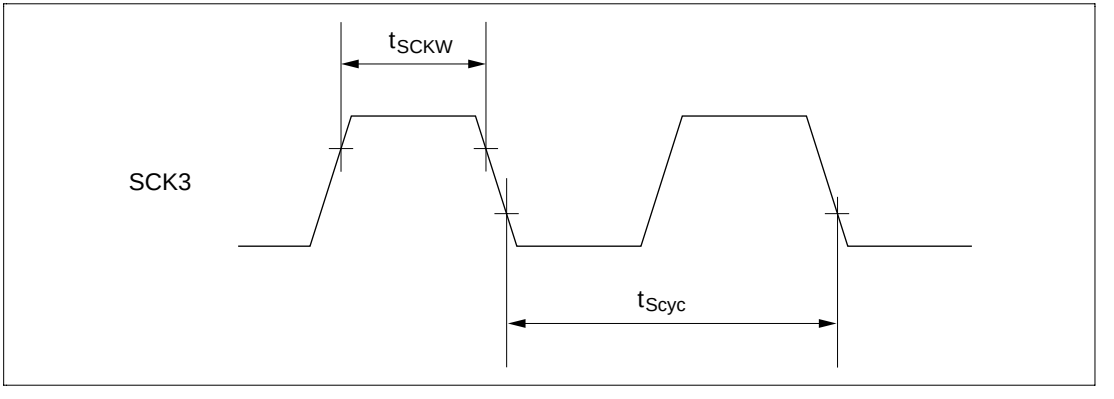
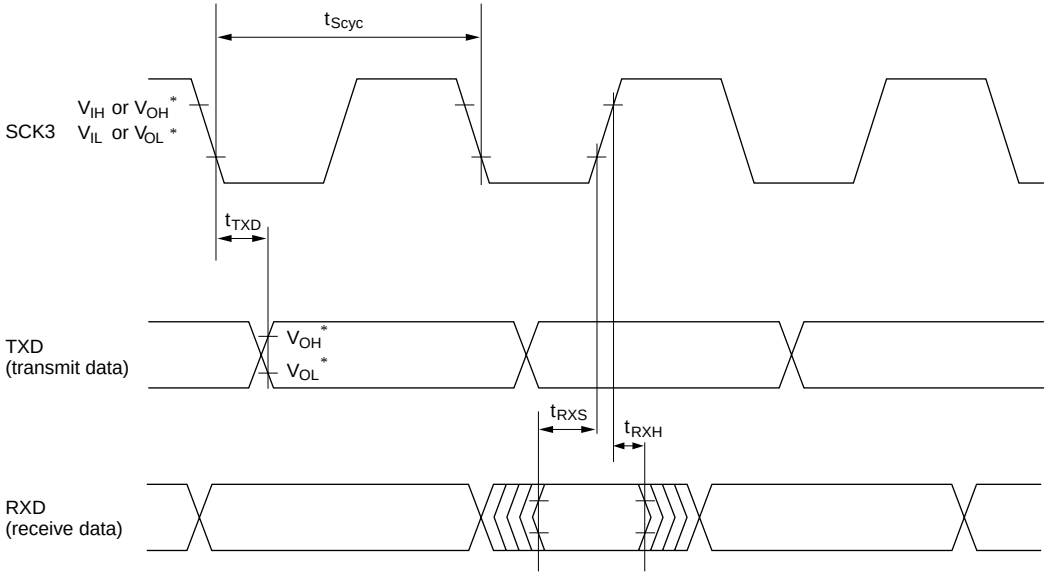


Figure 18.5 SCK3 Input Clock Timing



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 18.7.

Figure 18.6 Serial Interface 3 Synchronous Mode Input/Output Timing

18.5 Output Load Circuit

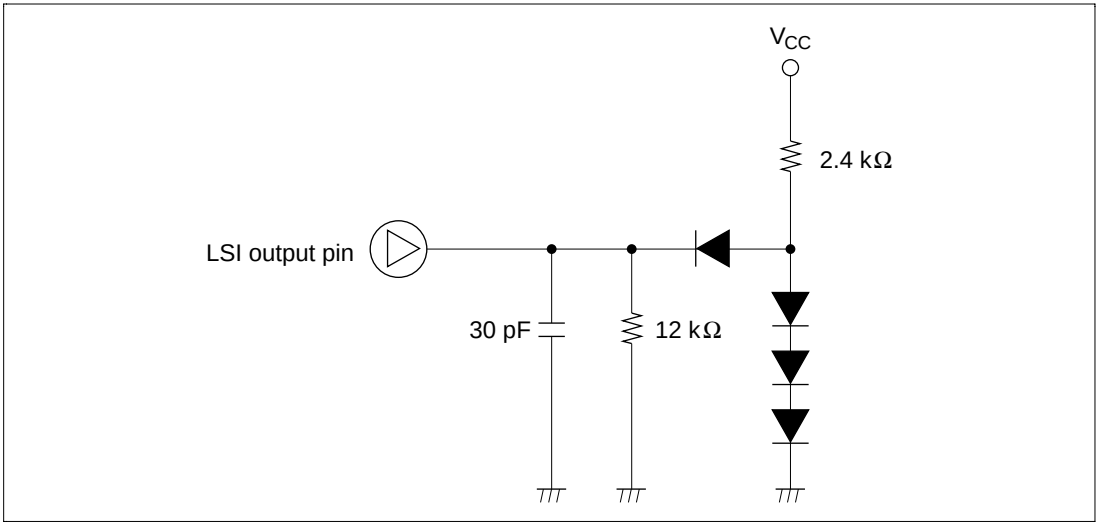


Figure 18.7 Output Load Condition

Appendix A Instruction Set

A.1 Instruction List

Operand Notation

Symbol	Description
Rd	General destination register
Rs	General source register
Rn	General register
ERd	General destination register (address register or 32-bit register)
ERs	General source register (address register or 32-bit register)
ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
PC	Program counter
SP	Stack pointer
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
disp	Displacement
→	Transfer from the operand on the left to the operand on the right, or transition from the state on the left to the state on the right
+	Addition of the operands on both sides
–	Subtraction of the operand on the right from the operand on the left
×	Multiplication of the operands on both sides
÷	Division of the operand on the left by the operand on the right
∧	Logical AND of the operands on both sides
∨	Logical OR of the operands on both sides
⊕	Exclusive logical OR of the operands on both sides
¬	NOT (logical complement)
( ), < >	Contents of operand

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

Condition Code Notation

Symbol	Description
↕	Changed according to execution result
*	Undetermined (no guaranteed value)
0	Cleared to 0
1	Set to 1
—	Not affected by execution of the instruction
Δ	Varies depending on conditions, described in notes

Table A.1 Instruction Set

1. Data transfer instructions

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—	I	H	N	Z	V	C	Normal	Advanced
MOV.B #xx:8, Rd	B	#xx:8 → Rd8	2									—	—	↕	↕	0	—	2	
MOV.B Rs, Rd	B	Rs8 → Rd8		2								—	—	↕	↕	0	—	2	
MOV.B @ERs, Rd	B	@ERs → Rd8			2							—	—	↕	↕	0	—	4	
MOV.B @(d:16, ERs), Rd	B	@(d:16, ERs) → Rd8				4						—	—	↕	↕	0	—	6	
MOV.B @(d:24, ERs), Rd	B	@(d:24, ERs) → Rd8				8						—	—	↕	↕	0	—	10	
MOV.B @ERs+, Rd	B	@ERs → Rd8 ERs32+1 → ERs32					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 @aa:24, Rd	B	@aa:24 → Rd8						6				—	—	↕	↕	0	—	8	
MOV.B Rs, @ERd	B	Rs8 → @ERd			2							—	—	↕	↕	0	—	4	
MOV.B Rs, @(d:16, ERd)	B	Rs8 → @(d:16, ERd)				4						—	—	↕	↕	0	—	6	
MOV.B Rs, @(d:24, ERd)	B	Rs8 → @(d:24, ERd)				8						—	—	↕	↕	0	—	10	
MOV.B Rs, @–ERd	B	ERd32–1 → ERd32 Rs8 → @ERd					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.B Rs, @aa:24	B	Rs8 → @aa:24						6				—	—	↕	↕	0	—	8	
MOV.W #xx:16, Rd	W	#xx:16 → Rd16	4									—	—	↕	↕	0	—	4	
MOV.W Rs, Rd	W	Rs16 → Rd16		2								—	—	↕	↕	0	—	2	
MOV.W @ERs, Rd	W	@ERs → Rd16			2							—	—	↕	↕	0	—	4	
MOV.W @(d:16, ERs), Rd	W	@(d:16, ERs) → Rd16				4						—	—	↕	↕	0	—	6	
MOV.W @(d:24, ERs), Rd	W	@(d:24, ERs) → Rd16				8						—	—	↕	↕	0	—	10	
MOV.W @ERs+, Rd	W	@ERs → Rd16 ERs32+2 → @ERd32					2					—	—	↕	↕	0	—	6	
MOV.W @aa:16, Rd	W	@aa:16 → Rd16						4				—	—	↕	↕	0	—	6	
MOV.W @aa:24, Rd	W	@aa:24 → Rd16						6				—	—	↕	↕	0	—	8	
MOV.W Rs, @ERd	W	Rs16 → @ERd			2							—	—	↕	↕	0	—	4	
MOV.W Rs, @(d:16, ERd)	W	Rs16 → @(d:16, ERd)				4						—	—	↕	↕	0	—	6	
MOV.W Rs, @(d:24, ERd)	W	Rs16 → @(d:24, ERd)				8						—	—	↕	↕	0	—	10	

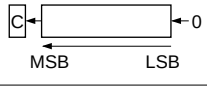
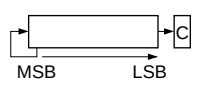
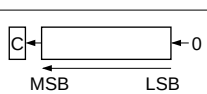
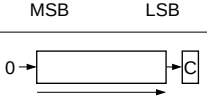
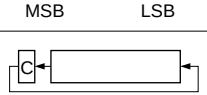
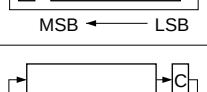
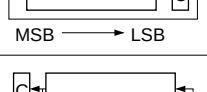
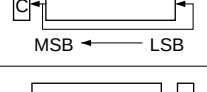
Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|	I	H	N	Z	V	C	Normal	Advanced
MOV.W Rs, @–ERd	W	ERd32–2 → ERd32 Rs16 → @ERd					2					—	—	↕	↕	0	—	6	
MOV.W Rs, @aa:16	W	Rs16 → @aa:16						4				—	—	↕	↕	0	—	6	
MOV.W Rs, @aa:24	W	Rs16 → @aa:24						6				—	—	↕	↕	0	—	8	
MOV.L #xx:32, Rd	L	#xx:32 → Rd32	6									—	—	↕	↕	0	—	6	
MOV.L ERs, ERd	L	ERs32 → ERd32		2								—	—	↕	↕	0	—	2	
MOV.L @ERs, ERd	L	@ERs → ERd32			4							—	—	↕	↕	0	—	8	
MOV.L @(d:16, ERs), ERd	L	@(d:16, ERs) → ERd32				6						—	—	↕	↕	0	—	10	
MOV.L @(d:24, ERs), ERd	L	@(d:24, ERs) → ERd32				10						—	—	↕	↕	0	—	14	
MOV.L @ERs+, ERd	L	@ERs → ERd32 ERs32+4 → ERs32					4					—	—	↕	↕	0	—	10	
MOV.L @aa:16, ERd	L	@aa:16 → ERd32										—	—	↕	↕	0	—	10	
MOV.L @aa:24, ERd	L	@aa:24 → ERd32						6				—	—	↕	↕	0	—	12	
MOV.L ERs, @ERd	L	ERs32 → @ERd			4			8				—	—	↕	↕	0	—	8	
MOV.L ERs, @(d:16, ERd)	L	ERs32 → @(d:16, ERd)				6						—	—	↕	↕	0	—	10	
MOV.L ERs, @(d:24, ERd)	L	ERs32 → @(d:24, ERd)				10						—	—	↕	↕	0	—	14	
MOV.L ERs, @–ERd	L	ERd32–4 → ERd32 ERs32 → @ERd					4					—	—	↕	↕	0	—	10	
MOV.L ERs, @aa:16	L	ERs32 → @aa:16										—	—	↕	↕	0	—	10	
MOV.L ERs, @aa:24	L	ERs32 → @aa:24						6				—	—	↕	↕	0	—	12	
POP.W Rn	W	@SP → Rn16 SP+2 → SP						8			2	—	—	↕	↕	0	—	6	
POP.L ERn	L	@SP → ERn32 SP+4 → SP									4	—	—	↕	↕	0	—	10	
PUSH.W Rn	W	SP–2 → SP Rn16 → @SP									2	—	—	↕	↕	0	—	6	
PUSH.L ERn	L	SP–4 → SP ERn32 → @SP									4	—	—	↕	↕	0	—	10	
MOVFPE @aa:16, Rd	B	Cannot be used in the H8/3664 Series						4				Cannot be used in the H8/3664 Series							
MOVTPE Rs, @aa:16	B	Cannot be used in the H8/3664 Series						4				Cannot be used in the H8/3664 Series							

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—	I	H	N	Z	V	C	Normal	Advanced
ADD.B #xx:8, Rd	B	Rd8+#xx:8 → Rd8	2									—	↕	↕	↕	↕	↕	2	
ADD.B Rs, Rd	B	Rd8+Rs8 → Rd8		2								—	↕	↕	↕	↕	↕	2	
ADD.W #xx:16, Rd	W	Rd16+#xx:16 → Rd16	4									—	(1)	↕	↕	↕	↕	4	
ADD.W Rs, Rd	W	Rd16+Rs16 → Rd16		2								—	(1)	↕	↕	↕	↕	2	
ADD.L #xx:32, ERd	L	ERd32+#xx:32 → ERd32	6									—	(2)	↕	↕	↕	↕	6	
ADD.L ERs, ERd	L	ERd32+ERs32 → ERd32		2								—	(2)	↕	↕	↕	↕	2	
ADDX.B #xx:8, Rd	B	Rd8+#xx:8 +C → Rd8	2									—	↕	↕	(3)	↕	↕	2	
ADDX.B Rs, Rd	B	Rd8+Rs8 +C → Rd8		2								—	↕	↕	(3)	↕	↕	2	
ADDS.L #1, ERd	L	ERd32+1 → ERd32		2								—	—	—	—	—	—	2	
ADDS.L #2, ERd	L	ERd32+2 → ERd32		2								—	—	—	—	—	—	2	
ADDS.L #4, ERd	L	ERd32+4 → ERd32		2								—	—	—	—	—	—	2	
INC.B Rd	B	Rd8+1 → Rd8		2								—	—	↕	↕	↕	—	2	
INC.W #1, Rd	W	Rd16+1 → Rd16		2								—	—	↕	↕	↕	—	2	
INC.W #2, Rd	W	Rd16+2 → Rd16		2								—	—	↕	↕	↕	—	2	
INC.L #1, ERd	L	ERd32+1 → ERd32		2								—	—	↕	↕	↕	—	2	
INC.L #2, ERd	L	ERd32+2 → ERd32		2								—	—	↕	↕	↕	—	2	
DAA Rd	B	Rd8 decimal adjust → Rd8		2								—	*	↕	↕	*	—	2	
SUB.B Rs, Rd	B	Rd8–Rs8 → Rd8		2								—	↕	↕	↕	↕	↕	2	
SUB.W #xx:16, Rd	W	Rd16–#xx:16 → Rd16	4									—	(1)	↕	↕	↕	↕	4	
SUB.W Rs, Rd	W	Rd16–Rs16 → Rd16		2								—	(1)	↕	↕	↕	↕	2	
SUB.L #xx:32, ERd	L	ERd32–#xx:32 → ERd32	6									—	(2)	↕	↕	↕	↕	6	
SUB.L ERs, ERd	L	ERd32–ERs32 → ERd32		2								—	(2)	↕	↕	↕	↕	2	
SUBX.B #xx:8, Rd	B	Rd8–#xx:8–C → Rd8	2									—	↕	↕	(3)	↕	↕	2	
SUBX.B Rs, Rd	B	Rd8–Rs8–C → Rd8		2								—	↕	↕	(3)	↕	↕	2	
SUBS.L #1, ERd	L	ERd32–1 → ERd32		2								—	—	—	—	—	—	2	
SUBS.L #2, ERd	L	ERd32–2 → ERd32		2								—	—	—	—	—	—	2	
SUBS.L #4, ERd	L	ERd32–4 → ERd32		2								—	—	—	—	—	—	2	
DEC.B Rd	B	Rd8–1 → Rd8		2								—	—	↕	↕	↕	—	2	
DEC.W #1, Rd	W	Rd16–1 → Rd16		2								—	—	↕	↕	↕	—	2	
DEC.W #2, Rd	W	Rd16–2 → Rd16		2								—	—	↕	↕	↕	—	2	

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|	I	H	N	Z	V	C	Normal	Advanced
DEC.L #1, ERd	L	ERd32–1 → ERd32		2								—	—	↕	↕	↕	—	2	
DEC.L #2, ERd	L	ERd32–2 → ERd32		2								—	—	↕	↕	↕	—	2	
DAS.Rd	B	Rd8 decimal adjust → Rd8		2								—	*	↕	↕	*	—	2	
MULXU. B Rs, Rd	B	Rd8 × Rs8 → Rd16 (unsigned multiplication)		2								—	—	—	—	—	—	14	
MULXU. W Rs, ERd	W	Rd16 × Rs16 → ERd32 (unsigned multiplication)		2								—	—	—	—	—	—	22	
MULXS. B Rs, Rd	B	Rd8 × Rs8 → Rd16 (signed multiplication)		4								—	—	↕	↕	—	—	16	
MULXS. W Rs, ERd	W	Rd16 × Rs16 → ERd32 (signed multiplication)		4								—	—	↕	↕	—	—	24	
DIVXU. B Rs, Rd	B	Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division)		2								—	—	(6)	(7)	—	—	14	
DIVXU. W Rs, ERd	W	ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (unsigned division)		2								—	—	(6)	(7)	—	—	22	
DIVXS. B Rs, Rd	B	Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division)		4								—	—	(8)	(7)	—	—	16	
DIVXS. W Rs, ERd	W	ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division)		4								—	—	(8)	(7)	—	—	24	
CMP.B #xx:8, Rd	B	Rd8–#xx:8	2									—	↕	↕	↕	↕	↕	2	
CMP.B Rs, Rd	B	Rd8–Rs8		2								—	↕	↕	↕	↕	↕	2	
CMP.W #xx:16, Rd	W	Rd16–#xx:16	4									—	(1)	↕	↕	↕	↕	4	
CMP.W Rs, Rd	W	Rd16–Rs16		2								—	(1)	↕	↕	↕	↕	2	
CMP.L #xx:32, ERd	L	ERd32–#xx:32	6									—	(2)	↕	↕	↕	↕	4	
CMP.L ERs, ERd	L	ERd32–ERs32		2								—	(2)	↕	↕	↕	↕	2	

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States[*1]	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|	I	H	N	Z	V	C	Normal	Advanced
NEG.B Rd	B	0–Rd8 → Rd8		2								—	↕	↕	↕	↕	↕	2	
NEG.W Rd	W	0–Rd16 → Rd16		2								—	↕	↕	↕	↕	↕	2	
NEG.L ERd	L	0–ERd32 → ERd32		2								—	↕	↕	↕	↕	↕	2	
EXTU.W Rd	W	0 → (<bits 15 to 8> of Rd16)		2								—	—	0	↕	0	—	2	
EXTU.L ERd	L	0 → (<bits 31 to 16> of ERd32)		2								—	—	0	↕	0	—	2	
EXTS.W Rd	W	(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)		2								—	—	↕	↕	0	—	2	
EXTS.L ERd	L	(<bit 15> of ERd32) → (<bits 31 to 16> of ERd32)		2								—	—	↕	↕	0	—	2	

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|	I	H	N	Z	V	C	Normal	Advanced
AND.B #xx:8, Rd	B	Rd8∧#xx:8 → Rd8	2									—	—	↕	↕	0	—	2	
AND.B Rs, Rd	B	Rd8∧Rs8 → Rd8		2								—	—	↕	↕	0	—	2	
AND.W #xx:16, Rd	W	Rd16∧#xx:16 → Rd16	4									—	—	↕	↕	0	—	4	
AND.W Rs, Rd	W	Rd16∧Rs16 → Rd16		2								—	—	↕	↕	0	—	2	
AND.L #xx:32, ERd	L	ERd32∧#xx:32 → ERd32	6									—	—	↕	↕	0	—	6	
AND.L ERs, ERd	L	ERd32∧ERs32 → ERd32		4								—	—	↕	↕	0	—	4	
OR.B #xx:8, Rd	B	Rd8∨#xx:8 → Rd8	2									—	—	↕	↕	0	—	2	
OR.B Rs, Rd	B	Rd8∨Rs8 → Rd8		2								—	—	↕	↕	0	—	2	
OR.W #xx:16, Rd	W	Rd16∨#xx:16 → Rd16	4									—	—	↕	↕	0	—	4	
OR.W Rs, Rd	W	Rd16∨Rs16 → Rd16		2								—	—	↕	↕	0	—	2	
OR.L #xx:32, ERd	L	ERd32∨#xx:32 → ERd32	6									—	—	↕	↕	0	—	6	
OR.L ERs, ERd	L	ERd32∨ERs32 → ERd32		4								—	—	↕	↕	0	—	4	
XOR.B #xx:8, Rd	B	Rd8⊕#xx:8 → Rd8	2									—	—	↕	↕	0	—	2	
XOR.B Rs, Rd	B	Rd8⊕Rs8 → Rd8		2								—	—	↕	↕	0	—	2	
XOR.W #xx:16, Rd	W	Rd16⊕#xx:16 → Rd16	4									—	—	↕	↕	0	—	4	
XOR.W Rs, Rd	W	Rd16⊕Rs16 → Rd16		2								—	—	↕	↕	0	—	2	
XOR.L #xx:32, ERd	L	ERd32⊕#xx:32 → ERd32	6									—	—	↕	↕	0	—	6	
XOR.L ERs, ERd	L	ERd32⊕ERs32 → ERd32		4								—	—	↕	↕	0	—	4	
NOT.B Rd	B	¬ Rd8 → Rd8		2								—	—	↕	↕	0	—	2	
NOT.W Rd	W	¬ Rd16 → Rd16		2								—	—	↕	↕	0	—	2	
NOT.L ERd	L	¬ Rd32 → Rd32		2								—	—	↕	↕	0	—	2	

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—	I	H	N	Z	V	C	Normal	Advanced
SHAL.B Rd	B			2								—	—	↕	↕	↕	↕	2	
SHAL.W Rd	W			2								—	—	↕	↕	↕	↕	2	
SHAL.L ERd	L			2								—	—	↕	↕	↕	↕	2	
SHAR.B Rd	B			2								—	—	↕	↕	0	↕	2	
SHAR.W Rd	W			2								—	—	↕	↕	0	↕	2	
SHAR.L ERd	L			2								—	—	↕	↕	0	↕	2	
SHLL.B Rd	B			2								—	—	↕	↕	0	↕	2	
SHLL.W Rd	W			2								—	—	↕	↕	0	↕	2	
SHLL.L ERd	L			2								—	—	↕	↕	0	↕	2	
SHLR.B Rd	B			2								—	—	↕	↕	0	↕	2	
SHLR.W Rd	W			2								—	—	↕	↕	0	↕	2	
SHLR.L ERd	L			2								—	—	↕	↕	0	↕	2	
ROTXL.B Rd	B			2								—	—	↕	↕	0	↕	2	
ROTXL.W Rd	W			2								—	—	↕	↕	0	↕	2	
ROTXL.L ERd	L			2								—	—	↕	↕	0	↕	2	
ROTXR.B Rd	B			2								—	—	↕	↕	0	↕	2	
ROTXR.W Rd	W			2								—	—	↕	↕	0	↕	2	
ROTXR.L ERd	L			2								—	—	↕	↕	0	↕	2	
ROTL.B Rd	B			2								—	—	↕	↕	0	↕	2	
ROTL.W Rd	W			2								—	—	↕	↕	0	↕	2	
ROTL.L ERd	L			2								—	—	↕	↕	0	↕	2	
ROTR.B Rd	B			2								—	—	↕	↕	0	↕	2	
ROTR.W Rd	W			2								—	—	↕	↕	0	↕	2	
ROTR.L ERd	L			2								—	—	↕	↕	0	↕	2	

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)								Condition Code						No. of States[*1]	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|	I	H	N	Z	V	C	Normal	Advanced
BSET #xx:3, Rd	B	(#xx:3 of Rd8) ← 1		2								—	—	—	—	—	—	2	
BSET #xx:3, @ERd	B	(#xx:3 of @ERd) ← 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, @ERd	B	(Rn8 of @ERd) ← 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, @ERd	B	(#xx:3 of @ERd) ← 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, @ERd	B	(Rn8 of @ERd) ← 0			4							—	—	—	—	—	—	8	
BCLR Rn, @aa:8	B	(Rn8 of @aa:8) ← 0						4				—	—	—	—	—	—	8	
BNOT #xx:3, Rd	B	(#xx:3 of Rd8) ← ¬ (#xx:3 of Rd8)		2								—	—	—	—	—	—	2	
BNOT #xx:3, @ERd	B	(#xx:3 of @ERd) ← ¬ (#xx:3 of @ERd)			4							—	—	—	—	—	—	8	
BNOT #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← ¬ (#xx:3 of @aa:8)						4				—	—	—	—	—	—	8	
BNOT Rn, Rd	B	(Rn8 of Rd8) ← ¬ (Rn8 of Rd8)		2								—	—	—	—	—	—	2	
BNOT Rn, @ERd	B	(Rn8 of @ERd) ← ¬ (Rn8 of @ERd)			4							—	—	—	—	—	—	8	
BNOT Rn, @aa:8	B	(Rn8 of @aa:8) ← ¬ (Rn8 of @aa:8)						4				—	—	—	—	—	—	8	
BTST #xx:3, Rd	B	¬ (#xx:3 of Rd8) → Z		2								—	—	—	↕	—	—	2	
BTST #xx:3, @ERd	B	¬ (#xx:3 of @ERd) → 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, @ERd	B	¬ (Rn8 of @ERd) → 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	

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—	I	H	N	Z	V	C	Normal	Advanced
BLD #xx:3, @ERd	B	(#xx:3 of @ERd) → 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, @ERd	B	¬ (#xx:3 of @ERd) → 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, @ERd	B	C → (#xx:3 of @ERd24)			4							—	—	—	—	—	—	8	
BST #xx:3, @aa:8	B	C → (#xx:3 of @aa:8)						4				—	—	—	—	—	—	8	
BIST #xx:3, Rd	B	¬ C → (#xx:3 of Rd8)		2								—	—	—	—	—	—	2	
BIST #xx:3, @ERd	B	¬ C → (#xx:3 of @ERd24)			4							—	—	—	—	—	—	8	
BIST #xx:3, @aa:8	B	¬ C → (#xx:3 of @aa:8)						4				—	—	—	—	—	—	8	
BAND #xx:3, Rd	B	C∧(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2	
BAND #xx:3, @ERd	B	C∧(#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕	6	
BAND #xx:3, @aa:8	B	C∧(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6	
BIAND #xx:3, Rd	B	C∧ ¬ (#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2	
BIAND #xx:3, @ERd	B	C∧ ¬ (#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕	6	
BIAND #xx:3, @aa:8	B	C∧ ¬ (#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6	
BOR #xx:3, Rd	B	C∨(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2	
BOR #xx:3, @ERd	B	C∨(#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕	6	
BOR #xx:3, @aa:8	B	C∨(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6	
BIOR #xx:3, Rd	B	C∨ ¬ (#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2	
BIOR #xx:3, @ERd	B	C∨ ¬ (#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕	6	
BIOR #xx:3, @aa:8	B	C∨ ¬ (#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6	
BXOR #xx:3, Rd	B	C⊕(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2	
BXOR #xx:3, @ERd	B	C⊕(#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕	6	
BXOR #xx:3, @aa:8	B	C⊕(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6	
BIXOR #xx:3, Rd	B	C⊕ ¬ (#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2	
BIXOR #xx:3, @ERd	B	C⊕ ¬ (#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕	6	
BIXOR #xx:3, @aa:8	B	C⊕ ¬ (#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6	

Mnemonic	Operand Size	Operation	Branch Condition	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
				#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|	I	H	N	Z	V	C	Normal	Advanced
BRA d:8 (BT d:8)	—	If condition is true then PC ← PC+d else next;	Always							2			—	—	—	—	—	—	4	
BRA d:16 (BT d:16)	—									4			—	—	—	—	—	—	6	
BRN d:8 (BF d:8)	—		Never							2			—	—	—	—	—	—	4	
BRN d:16 (BF d:16)	—									4			—	—	—	—	—	—	6	
BHI d:8	—		C ∨ Z = 0							2			—	—	—	—	—	—	4	
BHI d:16	—									4			—	—	—	—	—	—	6	
BLS d:8	—		C ∨ Z = 1							2			—	—	—	—	—	—	4	
BLS d:16	—									4			—	—	—	—	—	—	6	
BCC d:8 (BHS d:8)	—		C = 0							2			—	—	—	—	—	—	4	
BCC d:16 (BHS d:16)	—									4			—	—	—	—	—	—	6	
BCS d:8 (BLO d:8)	—		C = 1							2			—	—	—	—	—	—	4	
BCS d:16 (BLO d:16)	—									4			—	—	—	—	—	—	6	
BNE d:8	—		Z = 0							2			—	—	—	—	—	—	4	
BNE d:16	—									4			—	—	—	—	—	—	6	
BEQ d:8	—		Z = 1							2			—	—	—	—	—	—	4	
BEQ d:16	—									4			—	—	—	—	—	—	6	
BVC d:8	—		V = 0							2			—	—	—	—	—	—	4	
BVC d:16	—									4			—	—	—	—	—	—	6	
BVS d:8	—		V = 1							2			—	—	—	—	—	—	4	
BVS d:16	—									4			—	—	—	—	—	—	6	
BPL d:8	—		N = 0							2			—	—	—	—	—	—	4	
BPL d:16	—									4			—	—	—	—	—	—	6	
BMI d:8	—		N = 1							2			—	—	—	—	—	—	4	
BMI d:16	—									4			—	—	—	—	—	—	6	
BGE d:8	—		N⊕V = 0							2			—	—	—	—	—	—	4	
BGE d:16	—									4			—	—	—	—	—	—	6	
BLT d:8	—		N⊕V = 1							2			—	—	—	—	—	—	4	
BLT d:16	—									4			—	—	—	—	—	—	6	
BGT d:8	—		Z ∨ (N⊕V) = 0							2			—	—	—	—	—	—	4	
BGT d:16	—									4			—	—	—	—	—	—	6	
BLE d:8	—		Z ∨ (N⊕V) = 1							2			—	—	—	—	—	—	4	
BLE d:16	—									4			—	—	—	—	—	—	6	

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—	I	H	N	Z	V	C	Normal	Advanced
JMP @ERn	—	PC ← ERn			2							—	—	—	—	—	—	4	
JMP @aa:24	—	PC ← aa:24						4				—	—	—	—	—	—	6	
JMP @@aa:8	—	PC ← @aa:8								2		—	—	—	—	—	—	8	10
BSR d:8	—	PC → @–SP PC ← PC+d:8							2			—	—	—	—	—	—	6	8
BSR d:16	—	PC → @–SP PC ← PC+d:16							4			—	—	—	—	—	—	8	10
JSR @ERn	—	PC → @–SP PC ← ERn			2							—	—	—	—	—	—	6	8
JSR @aa:24	—	PC → @–SP PC ← aa:24						4				—	—	—	—	—	—	8	10
JSR @@aa:8	—	PC → @–SP PC ← @aa:8								2		—	—	—	—	—	—	8	12
RTS	—	PC ← @SP+									2	—	—	—	—	—	—	8	10

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)								Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|	I	H	N	Z	V	C	Normal	Advanced
TRAPA #x:2	—	PC → @–SP CCR → @–SP <vector> → PC									2	1	—	—	—	—	—	14	16
RTE	—	CCR ← @SP+ PC ← @SP+										↕	↕	↕	↕	↕	↕	10	
SLEEP	—	Transition to power-down state										—	—	—	—	—	—	2	
LDC #xx:8, CCR	B	#xx:8 → CCR	2									↕	↕	↕	↕	↕	↕	2	
LDC Rs, CCR	B	Rs8 → CCR		2								↕	↕	↕	↕	↕	↕	2	
LDC @ERs, CCR	W	@ERs → CCR			4							↕	↕	↕	↕	↕	↕	6	
LDC @(d:16, ERs), CCR	W	@(d:16, ERs) → CCR				6						↕	↕	↕	↕	↕	↕	8	
LDC @(d:24, ERs), CCR	W	@(d:24, ERs) → CCR				10						↕	↕	↕	↕	↕	↕	12	
LDC @ERs+, CCR	W	@ERs → CCR ERs32+2 → ERs32					4					↕ ↕	↕ ↕	↕ ↕	↕ ↕	↕ ↕	↕ ↕	8	
LDC @aa:16, CCR	W	@aa:16 → CCR						6				↕	↕	↕	↕	↕	↕	8	
LDC @aa:24, CCR	W	@aa:24 → CCR						8				↕	↕	↕	↕	↕	↕	10	
STC CCR, Rd	B	CCR → Rd8		2								—	—	—	—	—	—	2	
STC CCR, @ERd	W	CCR → @ERd			4							—	—	—	—	—	—	6	
STC CCR, @(d:16, ERd)	W	CCR → @(d:16, ERd)				6						—	—	—	—	—	—	8	
STC CCR, @(d:24, ERd)	W	CCR → @(d:24, ERd)				10						—	—	—	—	—	—	12	
STC CCR, @–ERd	W	ERd32–2 → ERd32 CCR → @ERd					4					—	—	—	—	—	—	8	
STC CCR, @aa:16	W	CCR → @aa:16						6				—	—	—	—	—	—	8	
STC CCR, @aa:24	W	CCR → @aa:24						8				—	—	—	—	—	—	10	
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	

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States[*1]	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|	I	H	N	Z	V	C	Normal	Advanced
EEPMOV. B	—	if R4L ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4L–1 → R4L until R4L=0 else next									4	—	—	—	—	—	—	8+ 4n[*2]	
EEPMOV. W	—	if R4 ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4–1 → R4 until R4=0 else next									4	—	—	—	—	—	—	8+ 4n[*2]	

Notes: 1. The number of states is the number of states required for execution when the instruction and its operands are located in on-chip memory. For other cases see section A.3, Combinations of Instructions and Addressing Modes.

2. n is the value set in register R4L or R4.

(1) Set to 1 when a carry or borrow occurs at bit 11; otherwise cleared to 0.

(2) Set to 1 when a carry or borrow occurs at bit 27; otherwise cleared to 0.

(3) Retains its previous value when the result is zero; otherwise cleared to 0.

(4) Set to 1 when the adjustment produces a carry; otherwise retains its previous value.

(5) The number of states required for execution of an instruction that transfers data in synchronization with the E clock is variable.

(6) Set to 1 when the divisor is negative; otherwise cleared to 0.

(7) Set to 1 when the divisor is zero; otherwise cleared to 0.

(8) Set to 1 when the quotient is negative; otherwise cleared to 0.

Table A.2 Operation Code Map (1)

Instruction code:

1st byte		2nd byte	
AH	AL	BH	BL

Instruction when most significant bit of BH is 0.

Instruction when most significant bit of BH is 1.

AH \ AL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	Table A-2 (2)	STC	LDC	ORC	XORC	ANDC	LDC	ADD		Table A-2 (2)	Table A-2 (2)	MOV		ADDX	Table A-2 (2)
1	Table A-2 (2)	Table A-2 (2)	Table A-2 (2)	Table A-2 (2)	OR.B	XOR.B	AND.B	Table A-2 (2)	SUB		Table A-2 (2)	Table A-2 (2)	CMP		SUBX	Table A-2 (2)
2	MOV.B															
3																
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5	MULXU	DIVXU	MULXU	DIVXU	RTS	BSR	RTE	TRAPA	Table A-2 (2)	JMP			BSR	JSR		
6	BSET	BNOT	BCLR	BTST	OR	XOR	AND	BST / BIST	MOV							
7					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD	MOV	Table A-2 (2)	Table A-2 (2)	EEPMOV	Table A-2 (3)			
8	ADD															
9	ADDX															
A	CMP															
B	SUBX															
C	OR															
D	XOR															
E	AND															
F	MOV															

Instruction code:

1st byte		2nd byte	
AH	AL	BH	BL

AH AL \ BH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
01	MOV				LDC/STC				SLEEP				Table A-2 (3)	Table A-2 (3)		Table A-2 (3)
0A	INC								ADD							
0B	ADDS					INC		INC	ADDS					INC		INC
0F	DAA								MOV							
10	SHLL			SHLL					SHAL			SHAL				
11	SHLR			SHLR					SHAR			SHAR				
12	ROTXL			ROTXL					ROTL			ROTL				
13	ROTXR			ROTXR					ROTR			ROTR				
17	NOT			NOT		EXTU		EXTU	NEG			NEG		EXTS		EXTS
1A	DEC								SUB							
1B	SUBS					DEC		DEC	SUB					DEC		DEC
1F	DAS								CMP							
58	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
79	MOV	ADD	CMP	SUB	OR	XOR	AND									
7A	MOV	ADD	CMP	SUB	OR	XOR	AND									

Instruction code:

1st byte		2nd byte		3rd byte		4th byte	
AH	AL	BH	BL	CH	CL	DH	DL

Instruction when most significant bit of DH is 0.

Instruction when most significant bit of DH is 1.

CL / AH / ALBH / BLCH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
01406										LDC / STC		LDC / STC		LDC / STC		LDC / STC
01C05	MULXS		MULXS													
01D05		DIVXS		DIVXS												
01F06					OR	XOR	AND									
7Cr06 *1				BTST												
7Cr07 *1				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
7Dr06 *1	BSET	BNOT	BCLR					BST / BIST								
7Dr07 *1	BSET	BNOT	BCLR													
7Eaa6 *2				BTST												
7Eaa7 *2				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
7Faa6 *2	BSET	BNOT	BCLR					BST / BIST								
7Faa7 *2	BSET	BNOT	BCLR													

Notes: 1. r is the register designation field.
2. aa is the absolute address field.

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:

$$\text{Execution states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

Examples: 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 section B.1, Register Addresses.

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					

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		

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					

Note: n: Initial value in R4L. The source and destination operands are accessed n + 1 times each.

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MOV	MOV.B @Rs, Rd	1		1			
	MOV.B @(d:16, Rs), Rd	2		1			
	MOV.B @Rs+, Rd	1		1			2
	MOV.B @aa:8, Rd	1		1			
	MOV.B @aa:16, Rd	2		1			
	MOV.B Rs, @Rd	1			1		
	MOV.B Rs, @(d:16, Rd)	2			1		
	MOV.B Rs, @–Rd	1			1		2
	MOV.B Rs, @aa:8	1			1		
	MOV.B Rs, @aa:16	2			1		
	MOV.W #xx:16, Rd	2					
	MOV.W Rs, Rd	1					
	MOV.W @Rs, Rd	1				1	
	MOV.W @(d:16, Rs), Rd	2				1	
	MOV.W @Rs+, Rd	1				1	2
	MOV.W @aa:16, Rd	2				1	
	MOV.W Rs, @Rd	1				1	
	MOV.W Rs, @(d:16d)	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					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
RTE	RTE	2		2			2
RTS	RTS	2		1			2
SHAL	SHAL.B Rd	1					
SHAR	SHAR.B Rd	1					
SHLL	SHLL.B Rd	1					
SHLR	SHLR.B Rd	1					
SLEEP	SLEEP	1					
STC	STC CCR, Rd	1					
SUB	SUB.B Rs, Rd	1					
	SUB.W Rs, Rd	1					
SUBS	SUBS.W #1, 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					

A.4 Combinations of Instructions and Addressing Modes

Table A.4 Combinations of Instructions and Addressing Modes

Functions	Instructions	Addressing Mode												
		#xx	Rn	@ERn	@(d:16.ERn)	@(d:24.ERn)	@ERn+/@ERn	@aa:8	@aa:16	@aa:24	@(d:8.PC)	@(d:16.PC)	@@aa:8	|
Data transfer instructions	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL	—	—	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—	—	WL
	MOVFPE, MOVTPE	—	—	—	—	—	—	—	—	—	—	—	—	—
Arithmetic operations	ADD, CMP	BWL	BWL	—	—	—	—	—	—	—	—	—	—	—
	SUB	WL	BWL	—	—	—	—	—	—	—	—	—	—	—
	ADDX, SUBX	B	B	—	—	—	—	—	—	—	—	—	—	—
	ADDS, SUBS	—	L	—	—	—	—	—	—	—	—	—	—	—
	INC, DEC	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	DAA, DAS	—	B	—	—	—	—	—	—	—	—	—	—	—
	MULXU, MULXS, DIVXU, DIVXS	—	BW	—	—	—	—	—	—	—	—	—	—	—
	NEG	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	EXTU, EXTS	—	WL	—	—	—	—	—	—	—	—	—	—	—
Logical operations	AND, OR, XOR	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	NOT	—	BWL	—	—	—	—	—	—	—	—	—	—	—
Shift operations		—	BWL	—	—	—	—	—	—	—	—	—	—	—
Bit manipulations		—	B	B	—	—	—	B	—	—	—	—	—	—
Branching instructions	BCC, BSR	—	—	—	—	—	—	—	—	—	—	—	—	—
	JMP, JSR	—	—	◯	—	—	—	—	—	—	◯	◯	—	—
	RTS	—	—	—	—	—	—	—	—	◯	—	—	◯	—
System control instructions	TRAPA	—	—	—	—	—	—	—	—	—	—	—	—	◯
	RTE	—	—	—	—	—	—	—	—	—	—	—	—	◯
	SLEEP	—	—	—	—	—	—	—	—	—	—	—	—	◯
	LDC	B	B	W	W	W	W	—	W	W	—	—	—	◯
	STC	—	B	W	W	W	W	—	W	W	—	—	—	—
	ANDC, ORC, XORC	B	—	—	—	—	—	—	—	—	—	—	—	—
	NOP	—	—	—	—	—	—	—	—	—	—	—	—	◯
Block data transfer instructions		—	—	—	—	—	—	—	—	—	—	—	—	BW

Appendix B Internal I/O Registers

B.1 Register Addresses

Register Name	Abbre-viation	Bits	Address	Module Name	Data Bus Width	Access State
Timer mode register W	TMRW	8	H'FF80	Timer W	8	2
Timer control register W	TCRW	8	H'FF81	Timer W	8	2
Timer interrupt enable register W	TIERW	8	H'FF82	Timer W	8	2
Timer status register W	TSRW	8	H'FF83	Timer W	8	2
Timer I/O control register 0	TIOR0	8	H'FF84	Timer W	8	2
Timer I/O control register 1	TIOR1	8	H'FF85	Timer W	8	2
Timer counter	TCNT	16	H'FF86	Timer W	16*[1]	2
General register A	GRA	16	H'FF88	Timer W	16*[1]	2
General register B	GRB	16	H'FF8A	Timer W	16*[1]	2
General register C	GRC	16	H'FF8C	Timer W	16*[1]	2
General register D	GRD	16	H'FF8E	Timer W	16*[1]	2
Flash memory control register 1	FLMCR1	8	H'FF90	ROM	8	2
Flash memory control register 2	FLMCR2	8	H'FF91	ROM	8	2
Flash memory power control register	FLPWCR	8	H'FF92	ROM	8	2
Erase block register 1	EBR1	8	H'FF93	ROM	8	2
Flash memory enable register	FENR	8	H'FF9B	ROM	8	2
Timer control register V0	TCRV0	8	H'FFA0	Timer V	8	3
Timer control/status register V	TCSRV	8	H'FFA1	Timer V	8	3
Timer constant register A	TCORA	8	H'FFA2	Timer V	8	3
Timer constant register B	TCORB	8	H'FFA3	Timer V	8	3
Timer counter V	TCNTV	8	H'FFA4	Timer V	8	3
Timer control register V1	TCRV1	8	H'FFA5	Timer V	8	3
Timer mode register A	TMA	8	H'FFA6	Timer A	8	2
Timer counter A	TCA	8	H'FFA7	Timer A	8	2
Serial mode register	SMR	8	H'FFA8	SCI3	8	3
Bit rate register	BRR	8	H'FFA9	SCI3	8	3
Serial control register 3	SCR3	8	H'FFAA	SCI3	8	3

Register Name	Abbre-viation	Bits	Address	Module Name	Data Bus Width	Access State
Transmit data register	TDR	8	H'FFAB	SCI3	8	3
Serial status register	SSR	8	H'FFAC	SCI3	8	3
Receive data register	RDR	8	H'FFAD	SCI3	8	3
A/D data register A	ADDRA	16	H'FFB0	A/D converter	8	3
A/D data register B	ADDRB	16	H'FFB2	A/D converter	8	3
A/D data register C	ADDRC	16	H'FFB4	A/D converter	8	3
A/D data register D	ADDRD	16	H'FFB6	A/D converter	8	3
A/D control/status register	ADCSR	8	H'FFB8	A/D converter	8	3
A/D control register	ADCR	8	H'FFB9	A/D converter	8	3
Timer control/status register WD	TCSRWD	8	H'FFC0	WDT*[2]	8	2
Timer counter WD	TCWD	8	H'FFC1	WDT*[2]	8	2
Timer mode register WD	TMWD	8	H'FFC2	WDT*[2]	8	2
I^2C bus control register	ICCR	8	H'FFC4	IIC	8	2
I^2C bus status register	ICSR	8	H'FFC5	IIC	8	2
I^2C bus data register	ICDR	8	H'FFC6	IIC	8	2
Second slave address register	SARX	8	H'FFC6	IIC	8	2
I^2C bus mode register	ICMR	8	H'FFC7	IIC	8	2
Slave address register	SAR	8	H'FFC7	IIC	8	2
Address break control register	ABRKCR	8	H'FFC8	Address break	8	2
Address break status register	ABRKSR	8	H'FFC9	Address break	8	2
Break address register H	BARH	8	H'FFCA	Address break	8	2
Break address register L	BARL	8	H'FFCB	Address break	8	2
Break data register H	BDRH	8	H'FFCC	Address break	8	2
Break data register L	BDRL	8	H'FFCD	Address break	8	2
Port pull-up control register 1	PUCR1	8	H'FFD0	I/O port	8	2
Port pull-up control register 5	PUCR5	8	H'FFD1	I/O port	8	2
Port data register 1	PDR1	8	H'FFD4	I/O port	8	2
Port data register 2	PDR2	8	H'FFD5	I/O port	8	2
Port data register 5	PDR5	8	H'FFD8	I/O port	8	2
Port data register 7	PDR7	8	H'FFDA	I/O port	8	2
Port data register 8	PDR8	8	H'FFDB	I/O port	8	2

Register Name	Abbre-viation	Bits	Address	Module Name	Data Bus Width	Access State
Port data register B	PDRB	8	H'FFDD	I/O port	8	2
Port mode register 1	PMR1	8	H'FFE0	I/O port	8	2
Port mode register 5	PMR5	8	H'FFE1	I/O port	8	2
Port control register 1	PCR1	8	H'FFE4	I/O port	8	2
Port control register 2	PCR2	8	H'FFE5	I/O port	8	2
Port control register 5	PCR5	8	H'FFE8	I/O port	8	2
Port control register 7	PCR7	8	H'FFEA	I/O port	8	2
Port control register 8	PCR8	8	H'FFEB	I/O port	8	2
System control register 1	SYSCR1	8	H'FFF0	Power-down	8	2
System control register 2	SYSCR2	8	H'FFF1	Power-down	8	2
Interrupt edge select register 1	IEGR1	8	H'FFF2	Interrupts	8	2
Interrupt edge select register 2	IEGR2	8	H'FFF3	Interrupts	8	2
Interrupt enable register 1	IENR1	8	H'FFF4	Interrupts	8	2
Interrupt flag register 1	IRR1	8	H'FFF6	Interrupts	8	2
Wake-up interrupt flag register	IWPR	8	H'FFF8	Interrupts	8	2
Module standby control register 1	MSTCR1	8	H'FFF9	Power-down	8	2
Timer serial control register	TSCR	8	H'FFFC	IIC	8	2

Notes: 1. Only word access can be used.
2. WDT: Watchdog timer

B.2 Register Bits

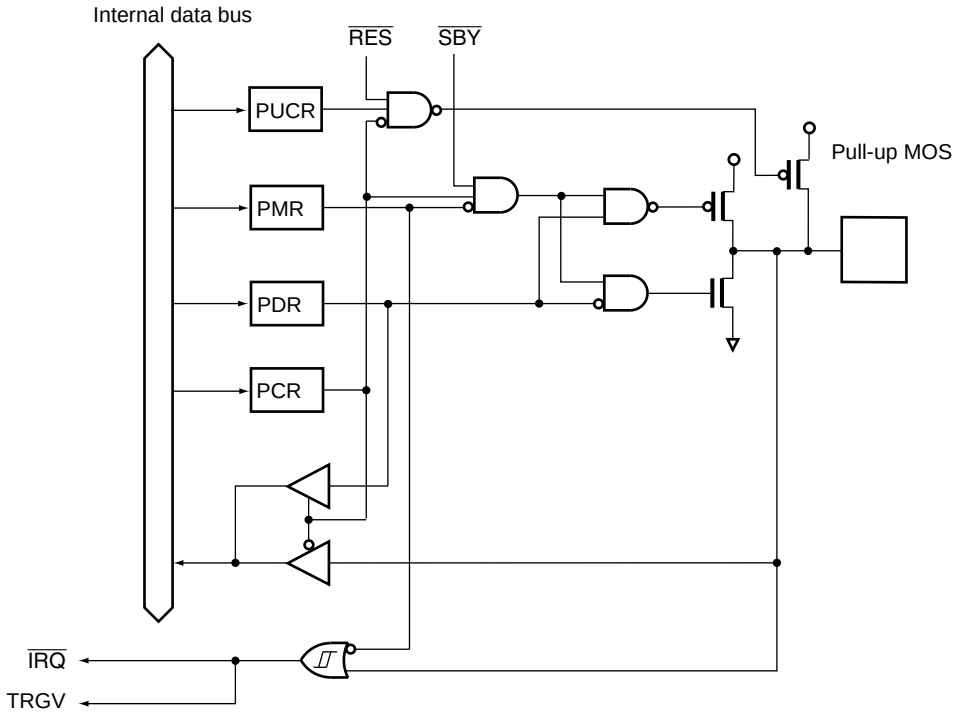
Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
TMRW	CTS	—	BUFEB	BUFEA	—	PWMD	PWMC	PWMB	Timer W
TCRW	CCLR	CKS2	CKS1	CKS0	TOD	TOC	TOB	TOA	
TIERW	OVIE	—	—	—	IMIED	IMIEC	IMIEB	IMIEA	
TSRW	OVF	—	—	—	IMFD	IMFC	IMFB	IMFA	
TIOR0	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0	
TIOR1	—	IOD2	IOD1	IOD0	—	IOC2	IOC1	IOC0	
TCNT	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8	
	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0	
GRA	GRA15	GRA14	GRA13	GRA12	GRA11	GRA10	GRA9	GRA8	
	GRA7	GRA6	GRA5	GRA4	GRA3	GRA2	GRA1	GRA0	
GRB	GRB15	GRB14	GRB13	GRB12	GRB11	GRB10	GRB9	GRB8	
	GRB7	GRB6	GRB5	GRB4	GRB3	GRB2	GRB1	GRB0	
GRC	GRC15	GRC14	GRC13	GRC12	GRC11	GRC10	GRC9	GRC8	
	GRC7	GRC6	GRC5	GRC4	GRC3	GRC2	GRC1	GRC0	
GRD	GRD15	GRD14	GRD13	GRD12	GRD11	GRD10	GRD9	GRD8	
	GRD7	GRD6	GRD5	GRD4	GRD3	GRD2	GRD1	GRD0	
FLMCR1	—	SWE	ESU	PSU	EV	PV	E	P	ROM
FLMCR2	FLER	—	—	—	—	—	—	—	
FLPWCR	PDWND	—	—	—	—	—	—	—	
EBR1	—	—	—	EB4	EB3	EB2	EB1	EB0	
FENR	FLSHE	—	—	—	—	—	—	—	
TCRV0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	Timer V
TCSRV	CMFB	CFMA	OVF	—	OS3	OS2	OS1	OS0	
TCORA	TCORA7	TCORA6	TCORA5	TCORA4	TCORA3	TCORA2	TCORA1	TCORA0	
TCORB	TCORB7	TCORB6	TCORB5	TCORB4	TCORB3	TCORB2	TCORB1	TCORB0	
TCNTV	TCNTV7	TCNTV6	TCNTV5	TCNTV4	TCNTV3	TCNTV2	TCNTV1	TCNTV0	
TCRV1	—	—	—	TVEG1	TVEG0	TRGE	—	ICKS0	
TMA	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0	Timer A
TCA	TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0	
SMR	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0	SCI3
BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0	
SCR3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
SSR	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT	SCI3
RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0	
ADDRA	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D converter
	AD1	AD0	—	—	—	—	—	—	
ADDRB	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRC	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRD	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0	
ADCR	TRGE	—	—	—	—	—	—	—	
TCSRWD	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI	WRST	WDT*
TCWD	TCWD7	TCWD6	TCWD5	TCWD4	TCWD3	TCWD2	TCWD1	TCWD0	
TMWD	—	—	—	—	CKS3	CKS2	CKS1	CKS0	
ICCR	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	IIC
ICSR	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	
ICDR	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0	
SARX	SVAX7	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	
ICMR	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0	
SAR	SVA7	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	
ABRKCR	RTINTE	CSEL1	CSEL0	ACMP2	ACMP1	ACMP0	DCMP1	DCMP0	Address break
ABRKSR	ABIF	ABIE	—	—	—	—	—	—	
BARH	BARH7	BARH6	BARH5	BARH4	BARH3	BARH2	BARH1	BARH0	
BARL	BARL7	BARL6	BARL5	BARL4	BARL3	BARL2	BARL1	BARL0	
BDRH	BDRH7	BDRH6	BDRH5	BDRH4	BDRH3	BDRH2	BDRH1	BDRH0	
BDRL	BDRL7	BDRL6	BDRL5	BDRL4	BDRL3	BDRL2	BDRL1	BDRL0	
PUCR1	PUCR17	PUCR16	PUCR15	PUCR14	—	PUCR12	PUCR11	PUCR10	I/O port
PUCR5	—	—	PUCR55	PUCR54	PUCR53	PUCR52	PUCR51	PUCR50	
PDR1	P17	P16	P15	P14	—	P12	P11	P10	
PDR2	—	—	—	—	—	P22	P21	P20	
PDR5	P57	P56	P55	P54	P53	P52	P51	P50	
PDR7	—	P76	P75	P74	—	—	—	—	
PDR8	P87	P86	P85	P84	P83	P82	P81	P80	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
PDRB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	I/O port
PMR1	IRQ3	IRQ2	IRQ1	IRQ0	—	—	TXD	TMOW	
PMR5	—	—	WKP5	WKP4	WKP3	WKP2	WKP1	WKP0	
PCR1	PCR17	PCR16	PCR15	PCR14	—	PCR12	PCR11	PCR10	
PCR2	—	—	—	—	—	PCR22	PCR21	PCR20	
PCR5	PCR57	PCR56	PCR55	PCR54	PCR53	PCR52	PCR51	PCR50	
PCR7	—	PCR6	PCR5	PCR4	—	—	—	—	
PCR8	PCR87	PCR86	PCR85	PCR84	PCR83	PCR82	PCR81	PCR80	
SYSCR1	SSBY	STS2	STS1	STS0	NESEL	—	—	—	Power-down
SYSCR2	SMSEL	LSON	DTON	MA2	MA1	MA0	SA1	SA0	
IEGR1	NMIEG	—	—	—	IEG3	IEG2	IEG1	IEG0	Interrupts
IEGR2	—	—	WPEG5	WPEG4	WPEG3	WPEG2	WPEG1	WPEG0	
IENR1	IENDT	IENTA	IENWP	—	IEN3	IEN2	IEN1	IEN0	
IRR1	IRRDT	IRRTA	—	—	IRRI3	IRRI2	IRRI1	IRRI0	
IWPR	—	—	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0	
MSTCR1	—	MSTIIC	MSTS3	MSTAD	MSTWD	MSTTW	MSTTV	MSTTA	Power-down
TSCR	—	—	—	—	—	—	IICRST	IICX	IIC

Note: * WDT: Watchdog timer

$\overline{RES}$ goes low in a reset, and $\overline{SBY}$ goes low in a reset and in standby mode.



PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure C.1 Port 1 Block Diagram (P17)

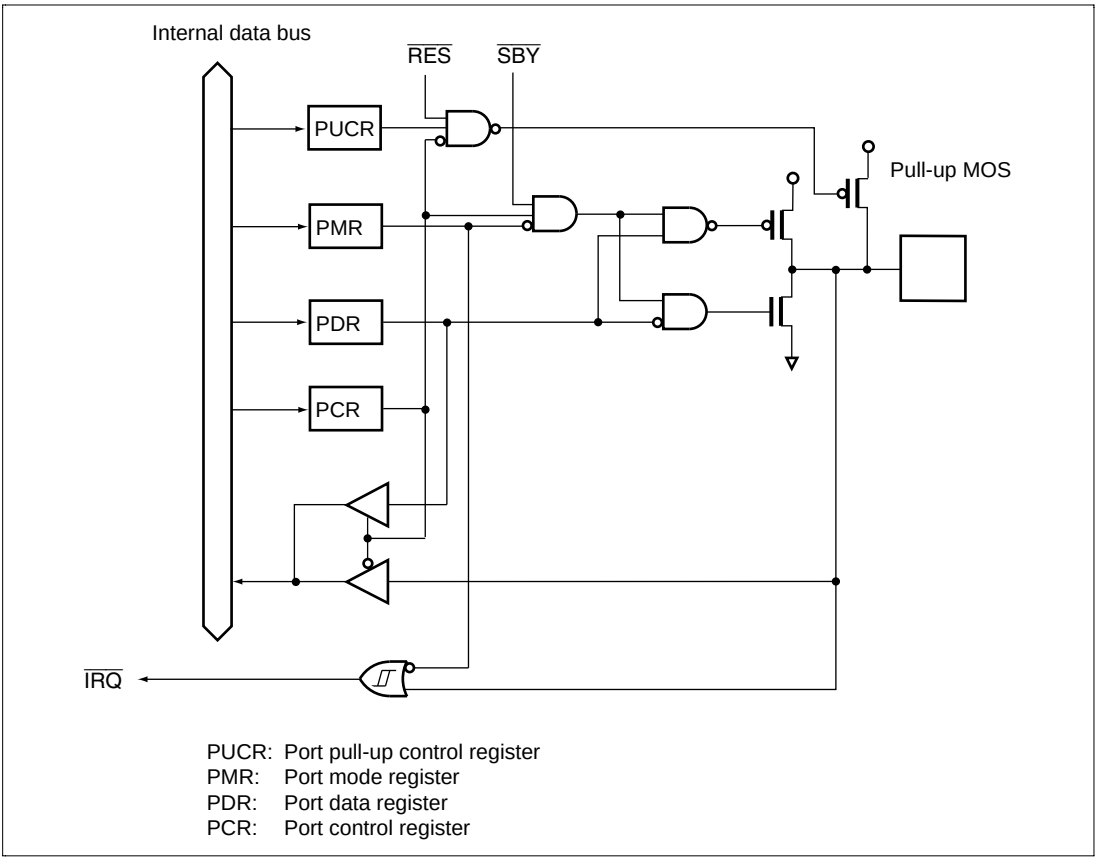
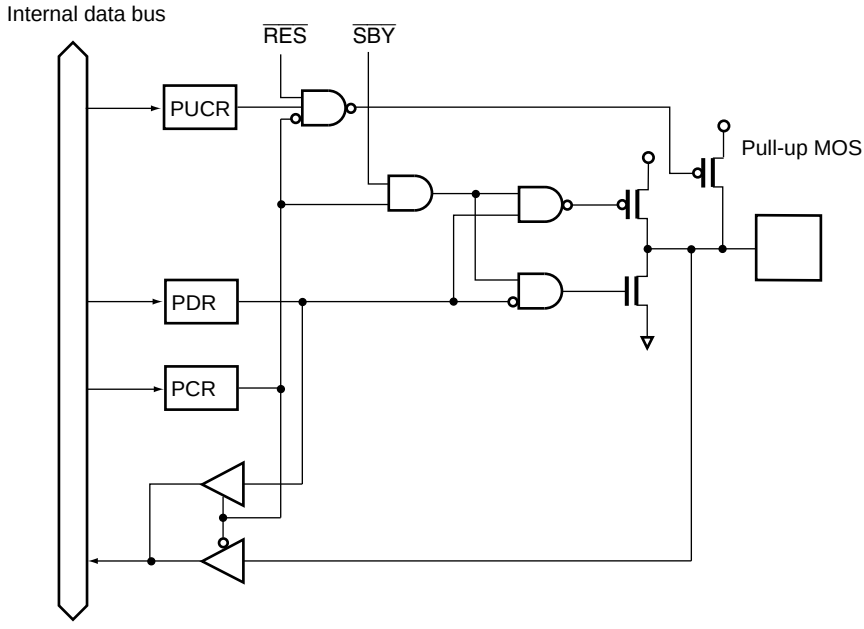


Figure C.2 Port 1 Block Diagram (P16 to P14)



PUCR: Port pull-up control register
PDR: Port data register
PCR: Port control register

Figure C.3 Port 1 Block Diagram (P12, P11)

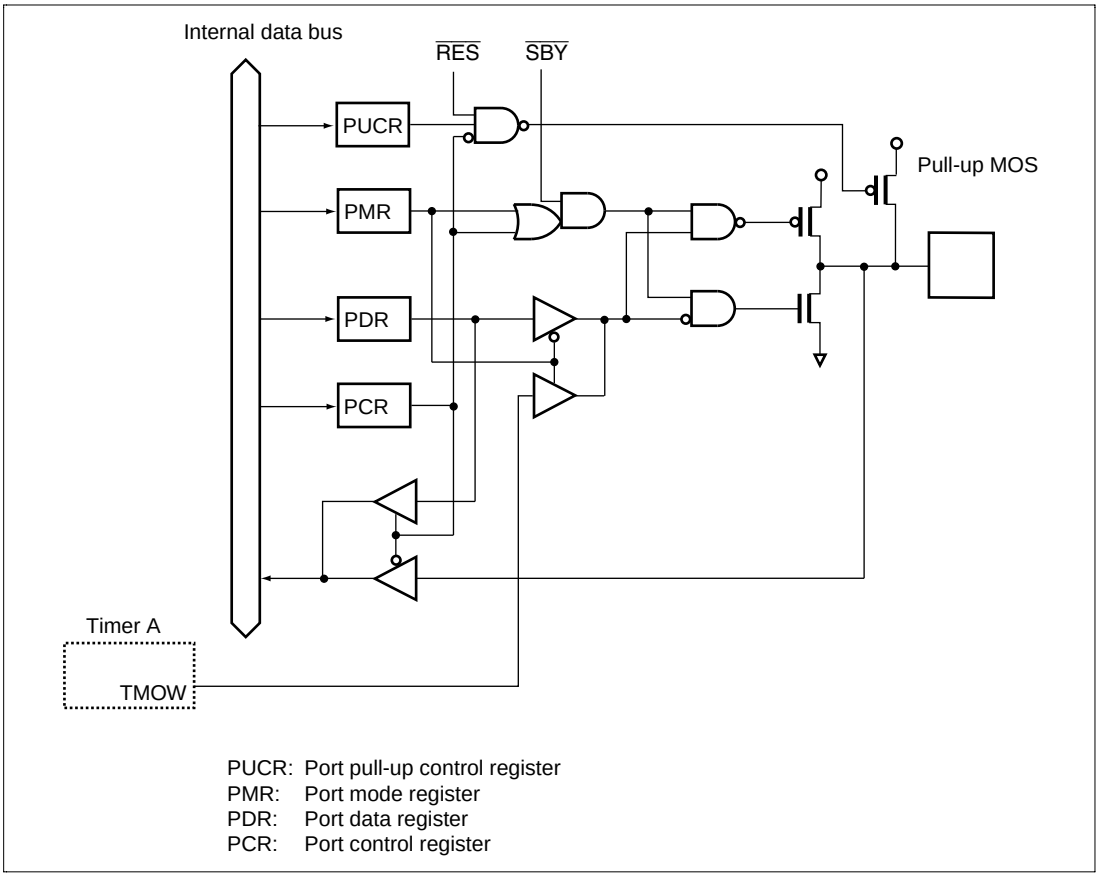


Figure C.4 Port 1 Block Diagram (P10)

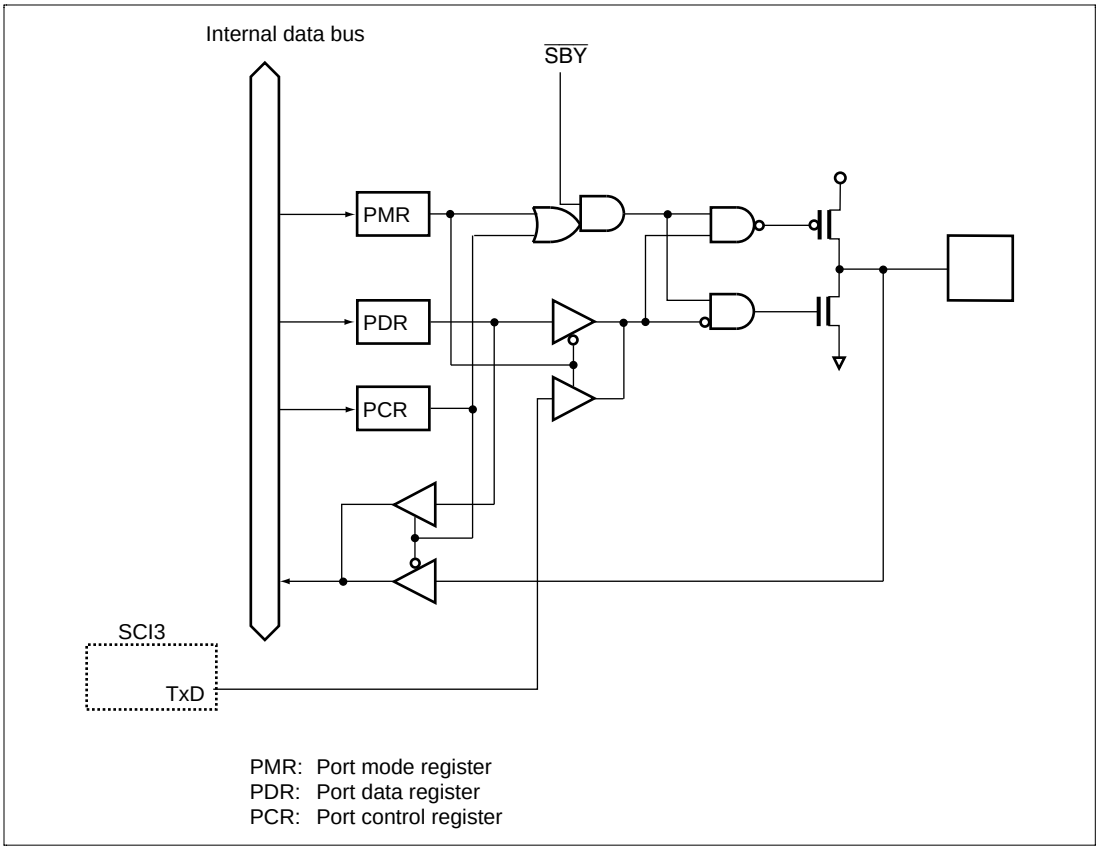


Figure C.5 Port 2 Block Diagram (P22)

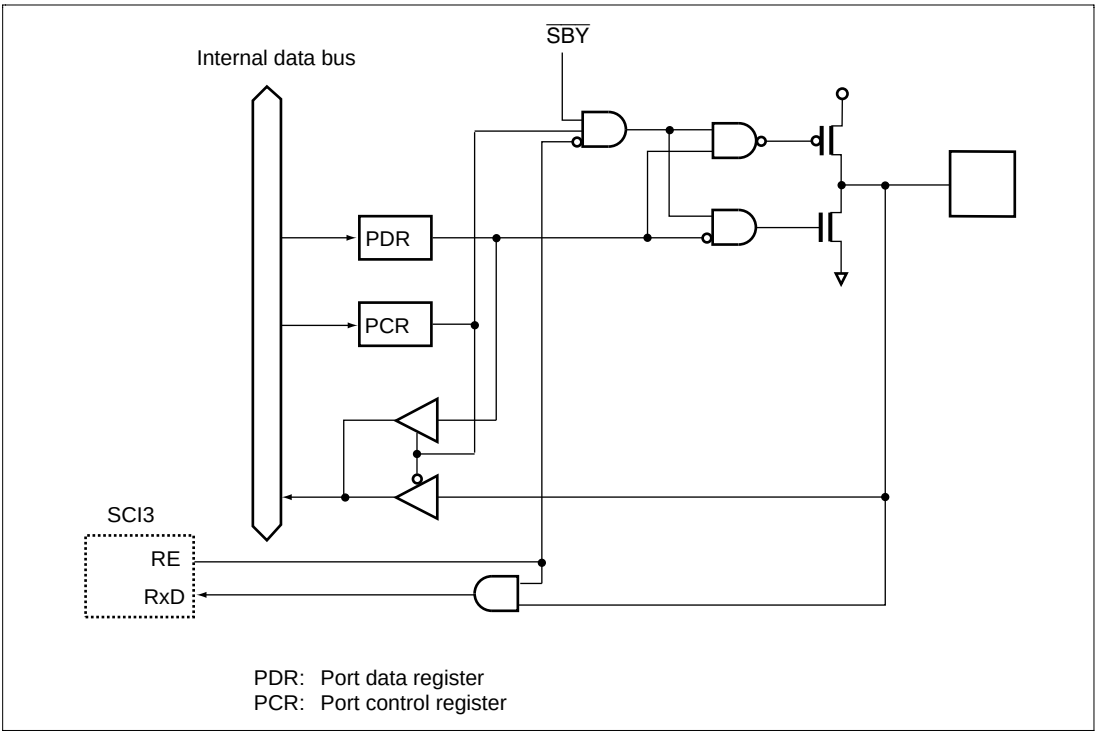


Figure C.6 Port 2 Block Diagram (P21)

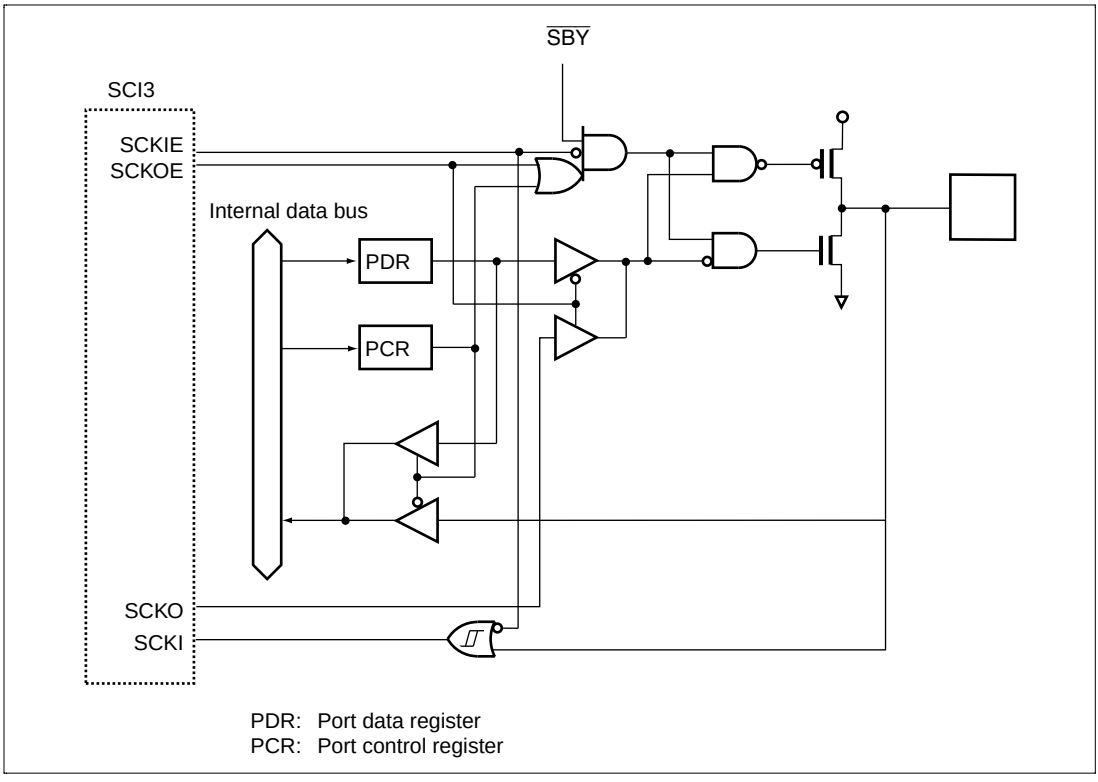


Figure C.7 Port 2 Block Diagram (P20)

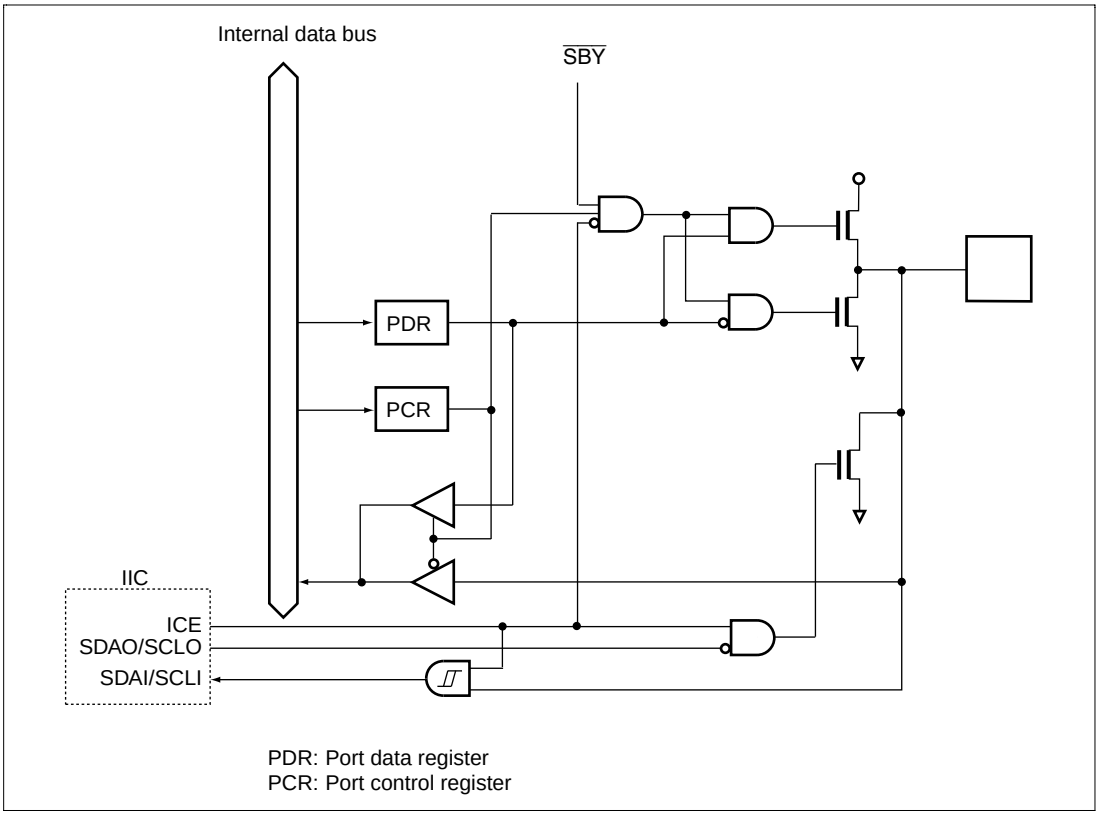
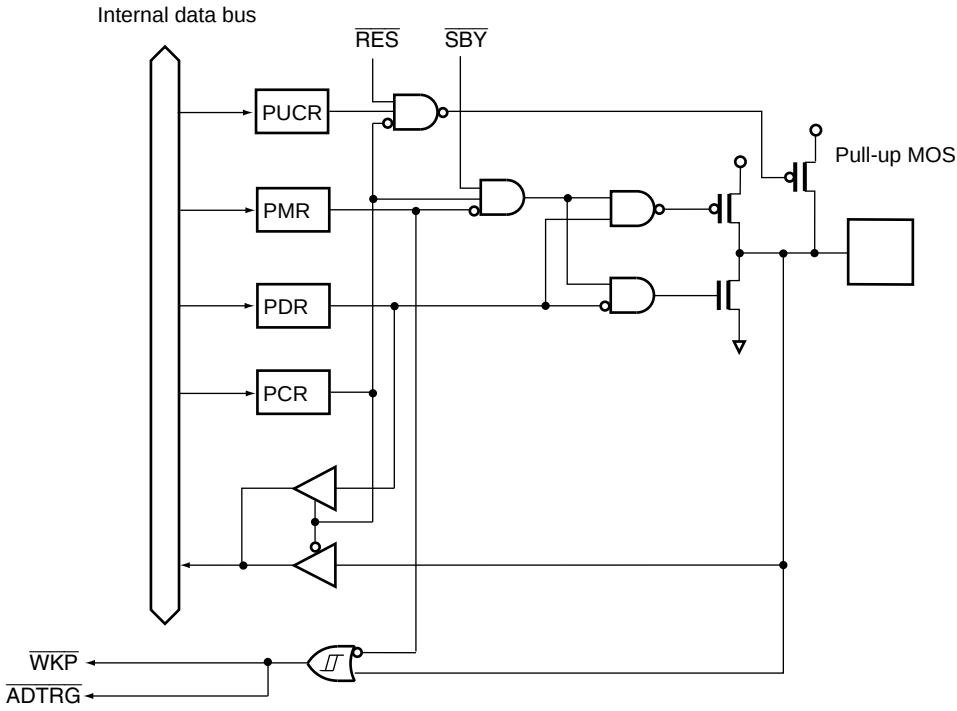
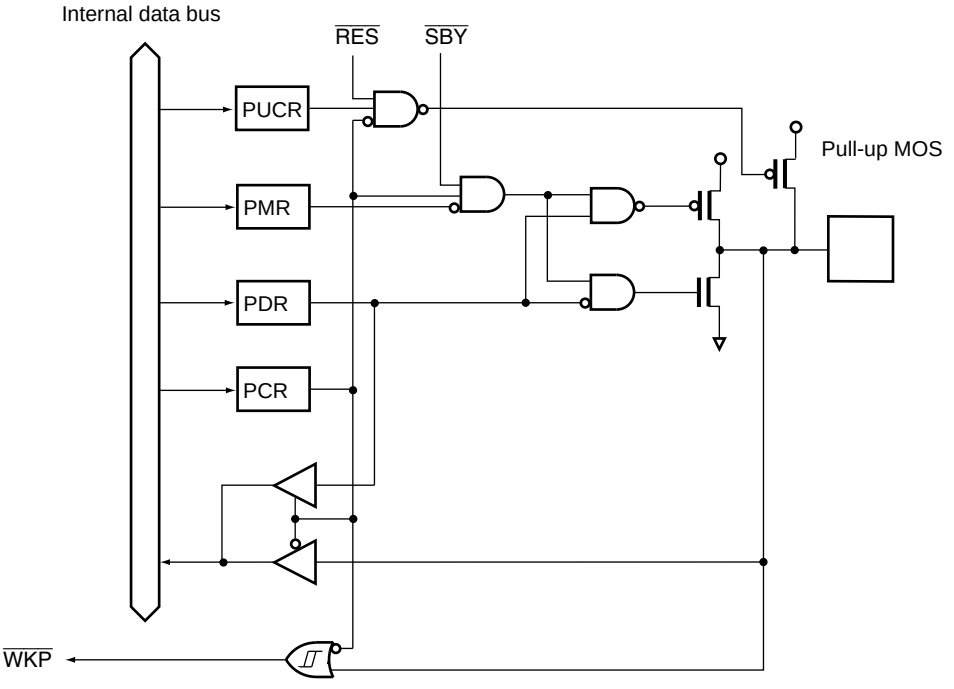


Figure C.8 Port 5 Block Diagram (P57, P56)



PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure C.9 Port 5 Block Diagram (P55)



PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure C.10 Port 5 Block Diagram (P54 to P50)

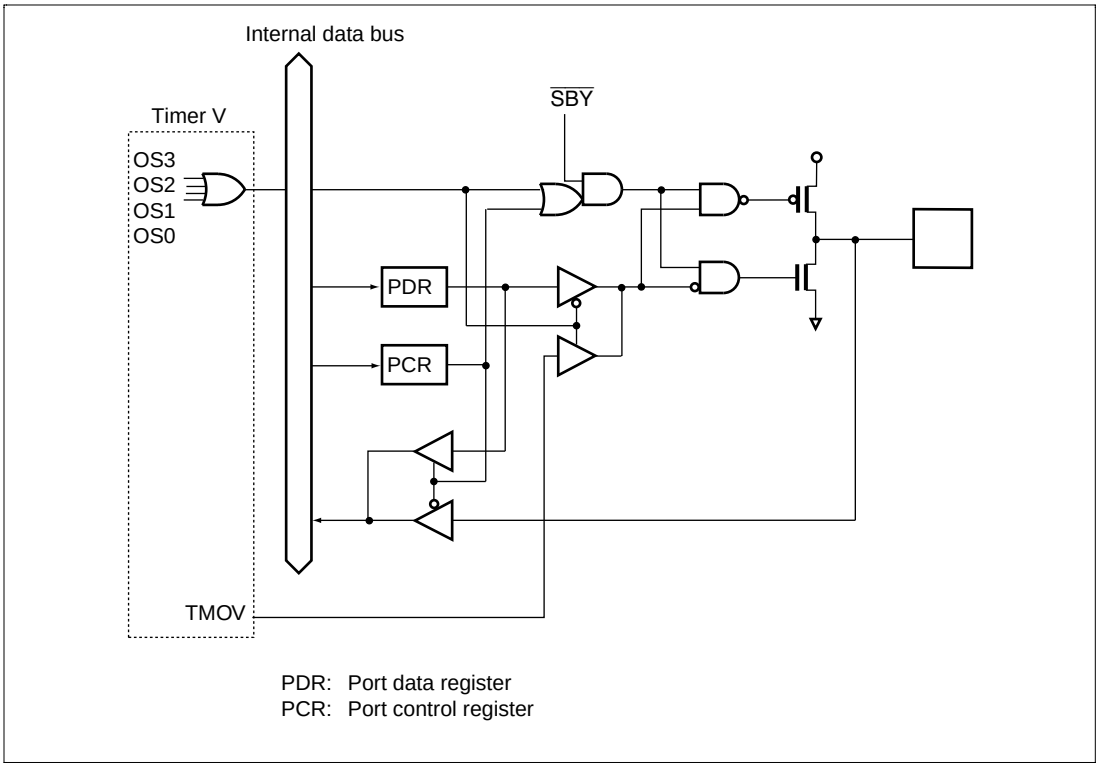


Figure C.11 Port 7 Block Diagram (P76)

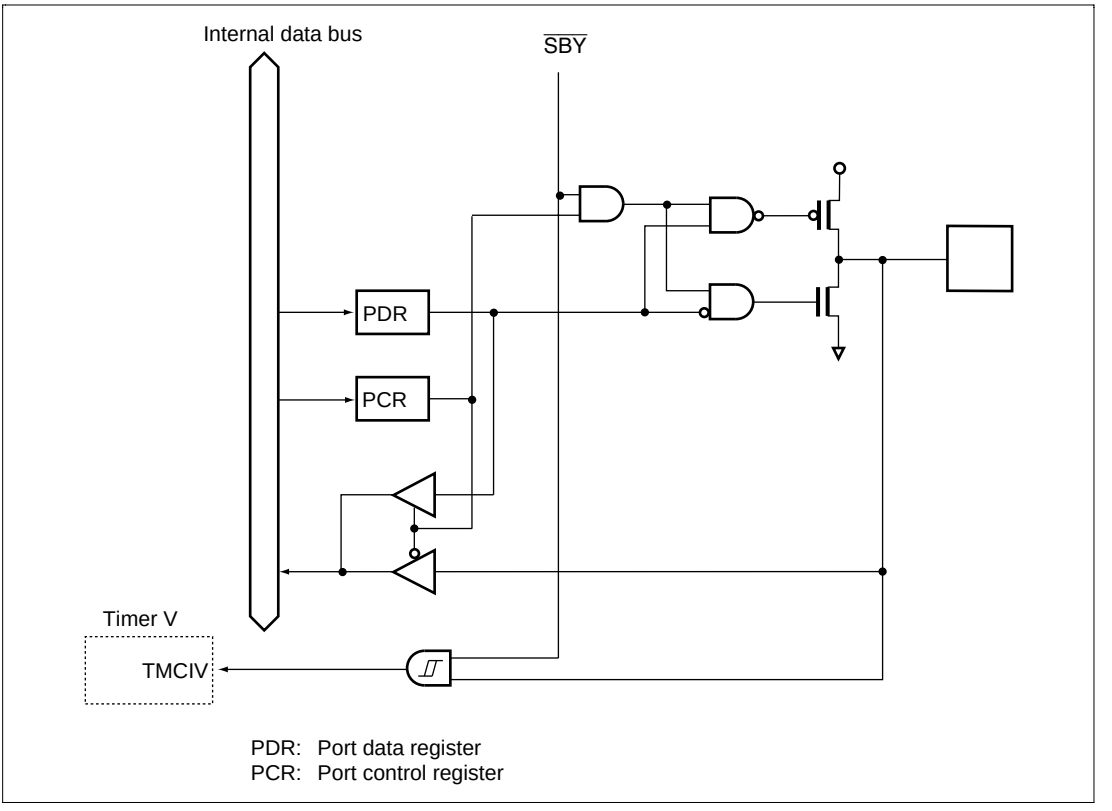


Figure C.12 Port 7 Block Diagram (P75)

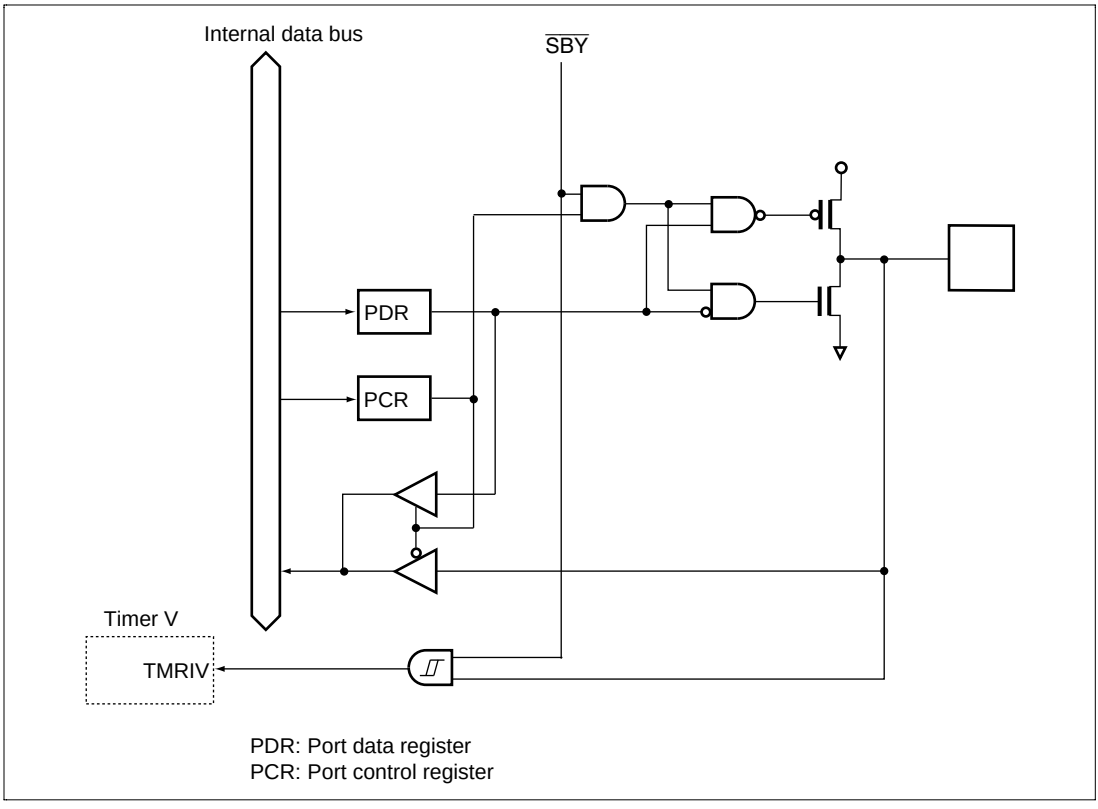


Figure C.13 Port 7 Block Diagram (P74)

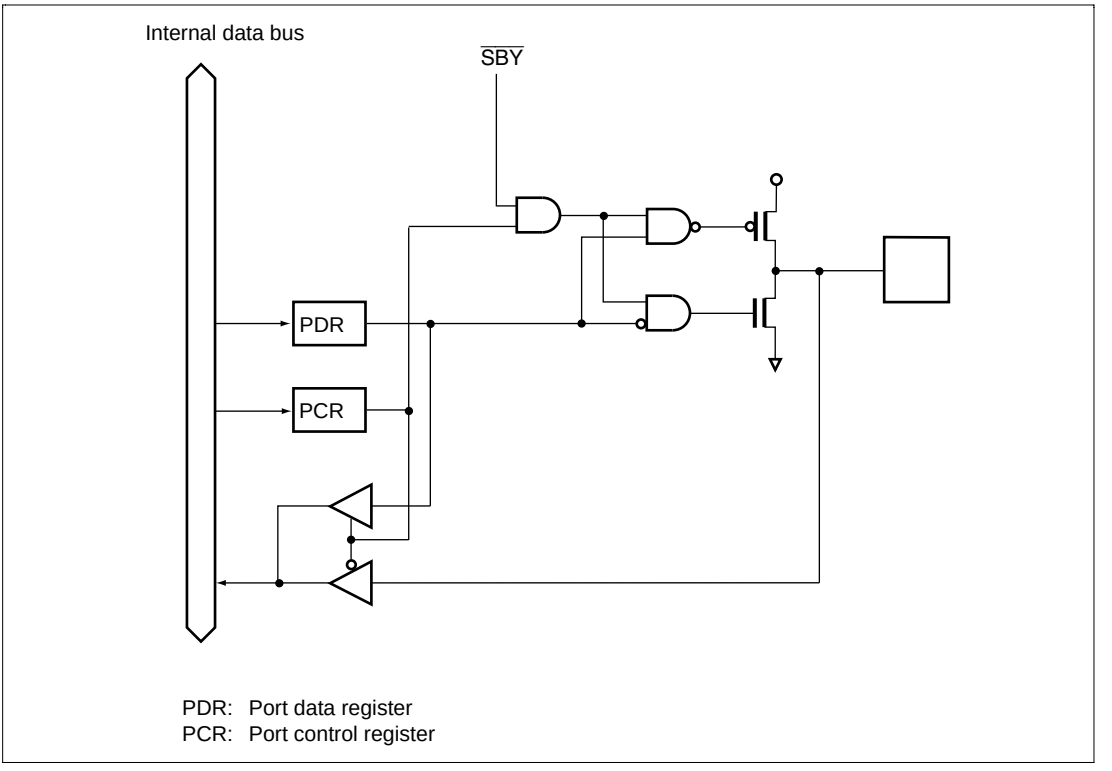


Figure C.14 Port 8 Block Diagram (P87 to P85)

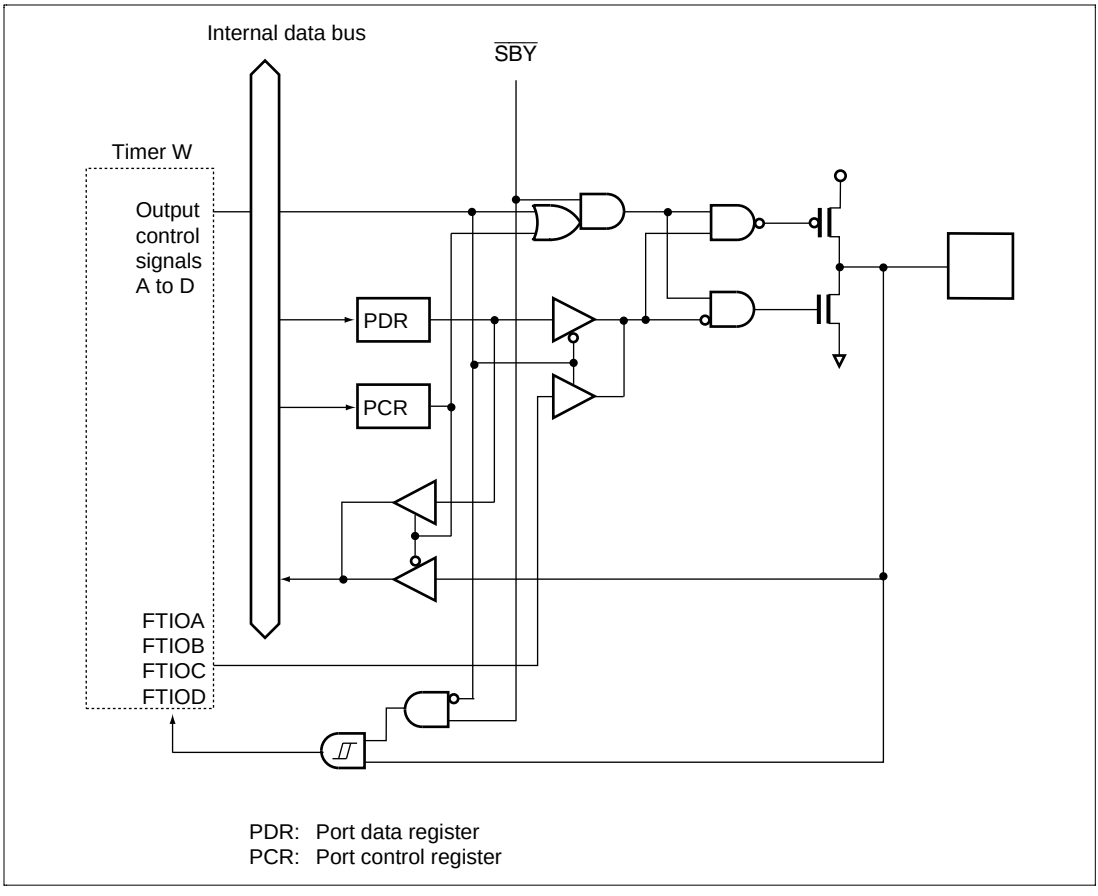


Figure C.15 Port 8 Block Diagram (P84 to P81)

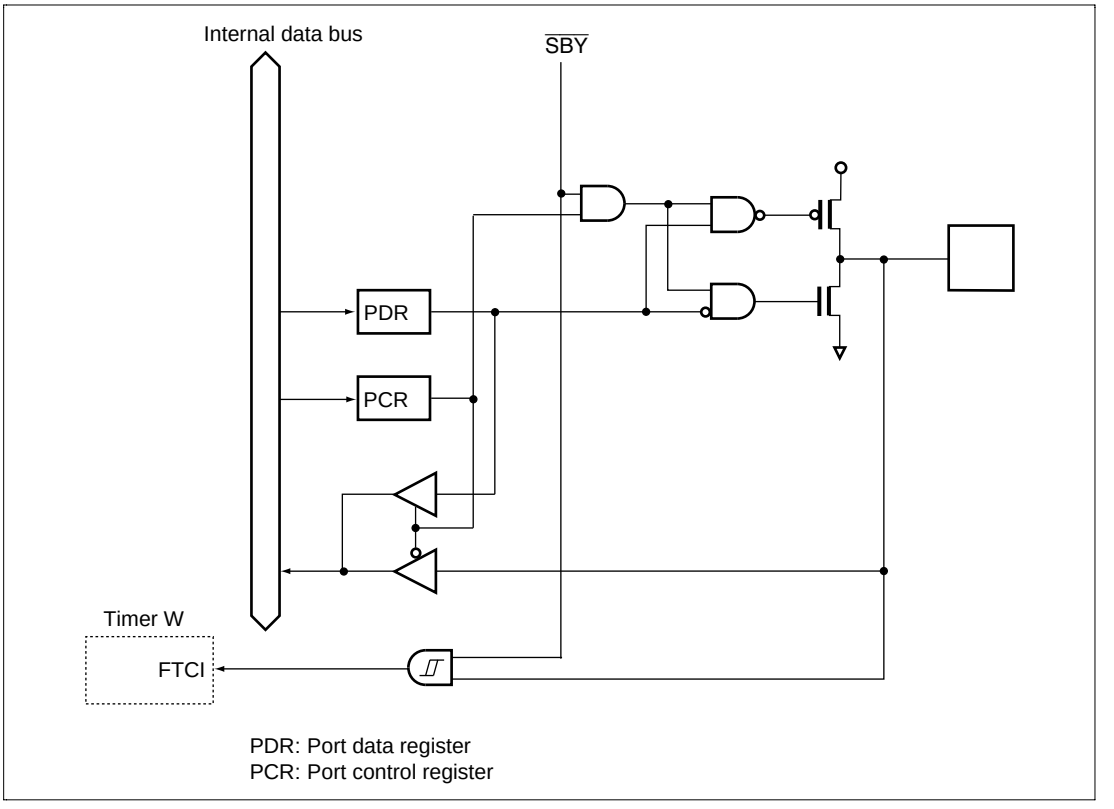


Figure C.16 Port 8 Block Diagram (P80)

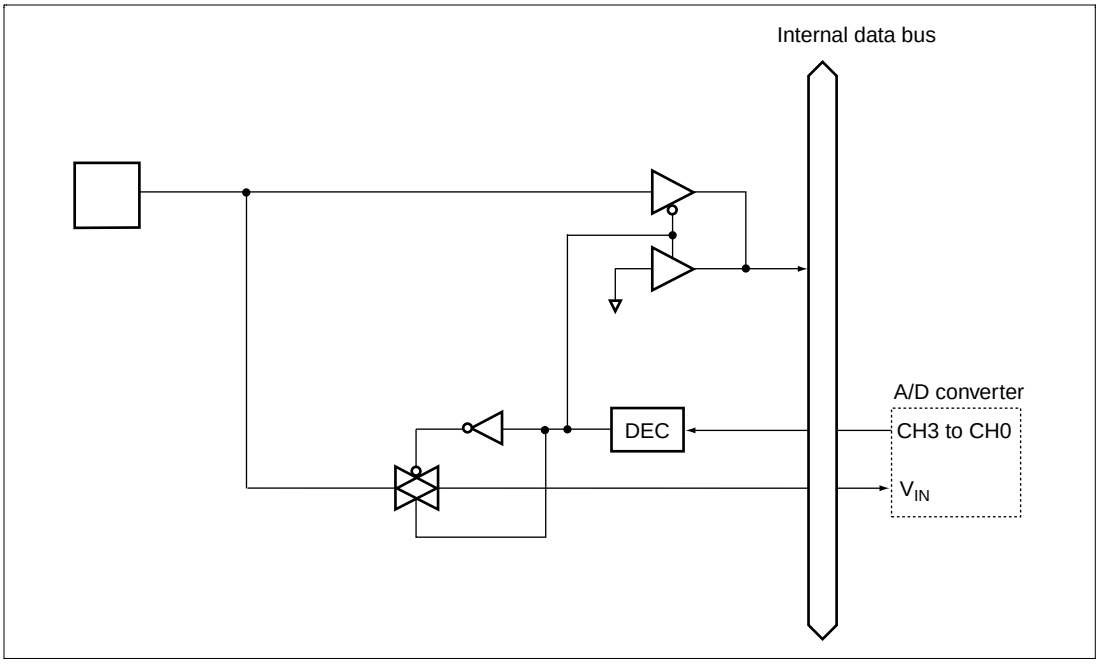


Figure C.17 Port B Block Diagram (PB7 to PB0)

Appendix D Port States in the Different Processing States

Table D.1 Port States Overview

Port	Reset	Sleep	Subsleep	Standby	Subactive	Active
$P1_7$ to $P1_4$, $P1_2$ to $P1_0$	High impedance	Retained	Retained	High impedance*	Functions	Functions
$P2_2$ to $P2_0$	High impedance	Retained	Retained	High impedance*	Functions	Functions
$P5_7$ to $P5_0$	High impedance	Retained	Retained	High impedance	Functions	Functions
$P7_6$ to $P7_4$	High impedance	Retained	Retained	High impedance	Functions	Functions
$P8_7$ to $P8_0$	High impedance	Retained	Retained	High impedance	Functions	Functions
PB_7 to PB_0	High impedance	High impedance	High impedance	High impedance	High impedance	High impedance

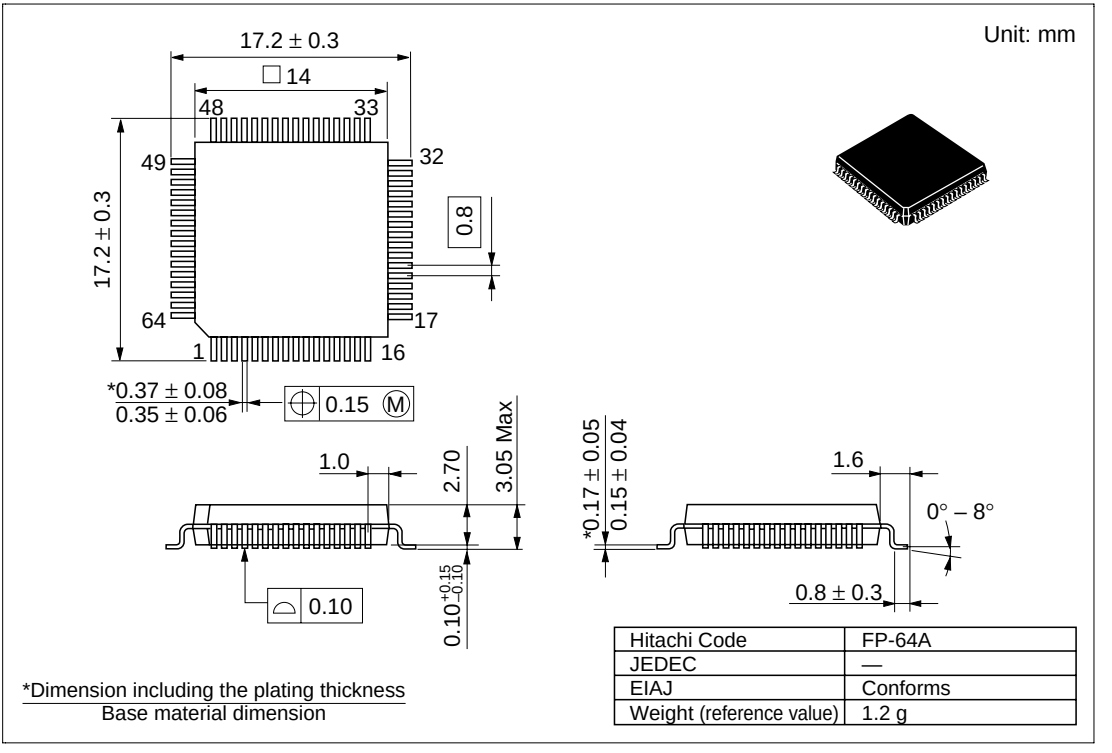
Note: * High level output when MOS pull-up is in on state.

Appendix E Model Names

Product Type			Package (Package Code)		
			QFP-64 (FP-64A)	QFP-64 (FP-64E)	SDIP-42 (DP-42S)
H8/3664	Flash memory version	Standard product	HD64F3664H	HD64F3664FP	HD64F3664BP
	Mask ROM version	Standard product	HD6433664H	HD6433664FP	HD6433664BP
H8/3663	Mask ROM version	Standard product	HD6433663H	HD6433663FP	HD6433663BP
H8/3662	Mask ROM version	Standard product	HD6433662H	HD6433662FP	HD6433662BP
H8/3661	Mask ROM version	Standard product	HD6433661H	HD6433661FP	HD6433661BP
H8/3660	Mask ROM version	Standard product	HD6433660H	HD6433660FP	HD6433660BP

Appendix F Package Dimensions

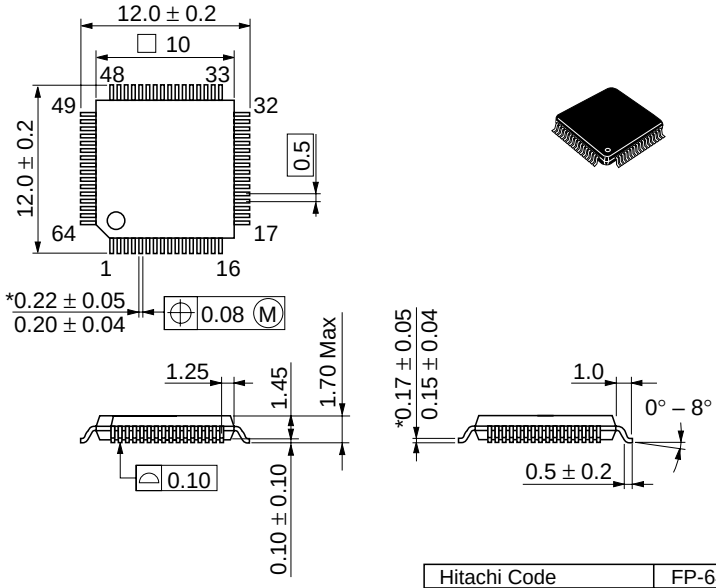
In case of discrepancy, the package dimensions given in the publication Hitachi Semiconductor Packages apply.



Hitachi Code	FP-64A
JEDEC	—
EIAJ	Conforms
Weight (reference value)	1.2 g

Figure F.1 FP-64A Package Dimensions

Unit: mm

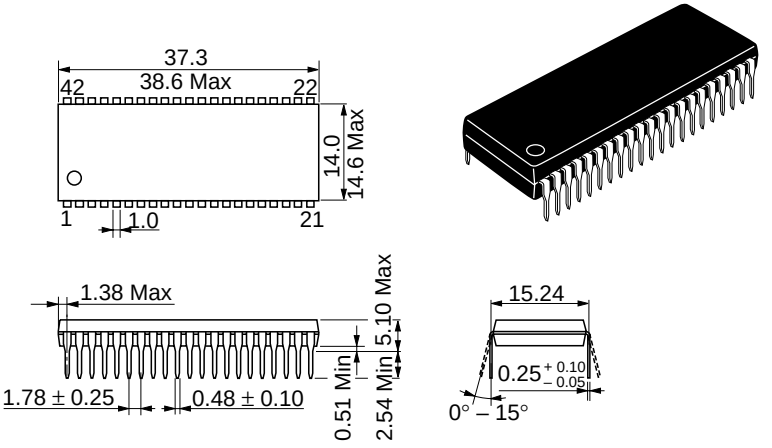


Hitachi Code	FP-64E
JEDEC	—
EIAJ	Conforms
Weight (reference value)	0.4 g

*Dimension including the plating thickness
Base material dimension

Figure F.2 FP-64E Package Dimensions

Unit: mm



Hitachi Code	DP-42S
JEDEC	—
EIAJ	Conforms
Weight (reference value)	4.8 g

Figure F.3 DP-42S Package Dimensions

H8/3664 Series, H8/3664F-ZTAT™ Hardware Manual

Publication Date: 1st Edition, March 2000
2nd Edition, September 2000
Published by: Electronic Devices Sales & Marketing Group
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
Hitachi Kodaira Semiconductor Co., Ltd.