REJ09B0152-0300

Everywhere you imagine.

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

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

16

H8/38602R Group

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer
H8 Family / H8/300H Super Low Power Series

H8/38602R HD64F38602R
HD64338602R
H8/38600R HD64338600R

Hardware Manual

Rev.3.00
Revision Date: May 15, 2007

Renesas Technology
www.renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

General Precautions on Handling of Product

1. Treatment of NC Pins

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

2. Treatment of Unused Input Pins

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

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

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

Configuration of This Manual

This manual comprises the following items:

1. General Precautions on Handling of Product
2. Configuration of This Manual
3. Preface
4. Contents
5. Overview
6. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

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

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

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

7. List of Registers
8. Electrical Characteristics
9. Appendix
10. Main Revisions and Additions in this Edition (only for revised versions)

The list of revisions is a summary of points that have been revised or added to earlier versions. This does not include all of the revised contents. For details, see the actual locations in this manual.

11. Index

Preface

The H8/38602R Group consists of single-chip microcomputers made up of the high-speed H8/300H CPU employing Renesas Technology original architecture as their cores, and the peripheral functions required to configure a system. The H8/300H CPU has an instruction set that is compatible with the H8/300 CPU.

Target Users: This manual was written for users who will be using the H8/38602R Group in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8/38602R Group to the target users.
Refer to the H8/300H Series Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

In order to understand the overall functions of the chip

Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions, and electrical characteristics.

In order to understand the details of the CPU's functions

Read the H8/300H Series Software Manual.

In order to understand the details of a register when its name is known

Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in section 20, List of Registers.

Example: Register name: The following notation is used for cases when the same or a similar function, e.g. serial communication interface, is implemented on more than one channel:
XXX_N (XXX is the register name and N is the channel number)

Bit order: The MSB is on the left and the LSB is on the right.

Notes:

When using an on-chip emulator (E7) for H8/38602R program development and debugging, the following restrictions must be noted.

1. The $\overline{\text{NMI}}$ pin is reserved for the E7, and cannot be used.
2. Area H'4000 to H'4FFF should not be accessed.
3. Area H'F780 to H'FB7F should not be accessed.
4. When the E7 is used, $\overline{\text{NMI}}$ is an input/output pin (open-drain in output mode).
5. When on-board programming/erasing is performed in boot mode, the SCI3 (P31/RXD3 and P32/TXD3) is used.
6. When the on-chip emulator is used, even though the on-chip oscillator is selected, connect a resonator to OSC1 and OSC2 or input an external clock to OSC1.
7. When using the E7, set the FROMCKSTP bit in clock halt register 1 to 1.

Related Manuals: The latest versions of all related manuals are available from our web site. Please ensure you have the latest versions of all documents you require. http://www.renesas.com/

H8/38602R Group manuals:

Document Title	Document No.
H8/38602R Group Hardware Manual	This manual
H8/300H Series Software Manual	REJ09B0213

User's manuals for development tools:

Document Title	Document No.
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	REJ10B0058
Microcomputer Development Environment System H8S, H8/300 Series Simulator/Debugger User's Manual	ADE-702-282
H8S, H8/300 Series High-performance Embedded Workshop 3 Tutorial	REJ10B0024
H8S, H8/300 Series High-performance Embedded Workshop 3 User's Manual	REJ10B0026

Application notes:

Document Title	Document No.
F-ZTAT Microcomputer On-Board Programming	REJ05B0523

Contents

Figures

Section 5 Power-Down Modes

Section 6 ROM

Section 8 I/O Ports

Section 9 Timer B1

Section 10 Timer W

Section 15 Synchronous Serial Communication Unit (SSU)

Section 21 Electrical Characteristics

Appendix

Tables

Appendix

Section 1 Overview

1.1 Features

- High-speed H8/300H central processing unit with an internal 16-bit architecture
  - — Upward-compatible with H8/300 CPU on an object level
  - — Sixteen 16-bit general registers
  - — 62 basic instructions
- Various peripheral functions
  - — RTC (can be used as a free-running counter)
  - — Asynchronous event counter (AEC)
  - — Timer B1
  - — Timer W
  - — Watchdog timer
  - — SCI (asynchronous or clock synchronous serial communication interface)
  - — SSU (synchronous serial communication unit)*
  - — I^2C bus interface (conforms to the I^2C bus interface format that is advocated by Philips Electronics)*
  - — 10-bit A/D converter
  - — Comparators

Note: * SSU and IIC2 are shared.

- On-chip memory

Product Classification		Model	ROM	RAM
Flash memory version (F-ZTAT™ version)	H8/38602RF	HD64F38602R	16 Kbytes	1 Kbyte
Masked ROM version	H8/38602R	HD64338602R	16 Kbytes	1 Kbyte
	H8/38600R	HD64338600R	8 Kbytes	512 bytes

Note: F-ZTAT™ is a trademark of Renesas Technology Corp.

- General I/O ports
  - — I/O pins: 13 I/O pins, including three large current ports (I_{OL} = 15 mA, @V_{OL} = 1.0 V)
  - — Input-only pins: 6 input pins (also used as analog input pins)
- Supports various power-down states

- Compact package

Package	Code	Body Size	Pin Pitch	Remarks
P-VQFN-32	TNP-32	5 × 6 mm	0.5 mm	
P-LQFP-32	32P6U-A	7 × 7 mm	0.8 mm	

1.2 Internal Block Diagram

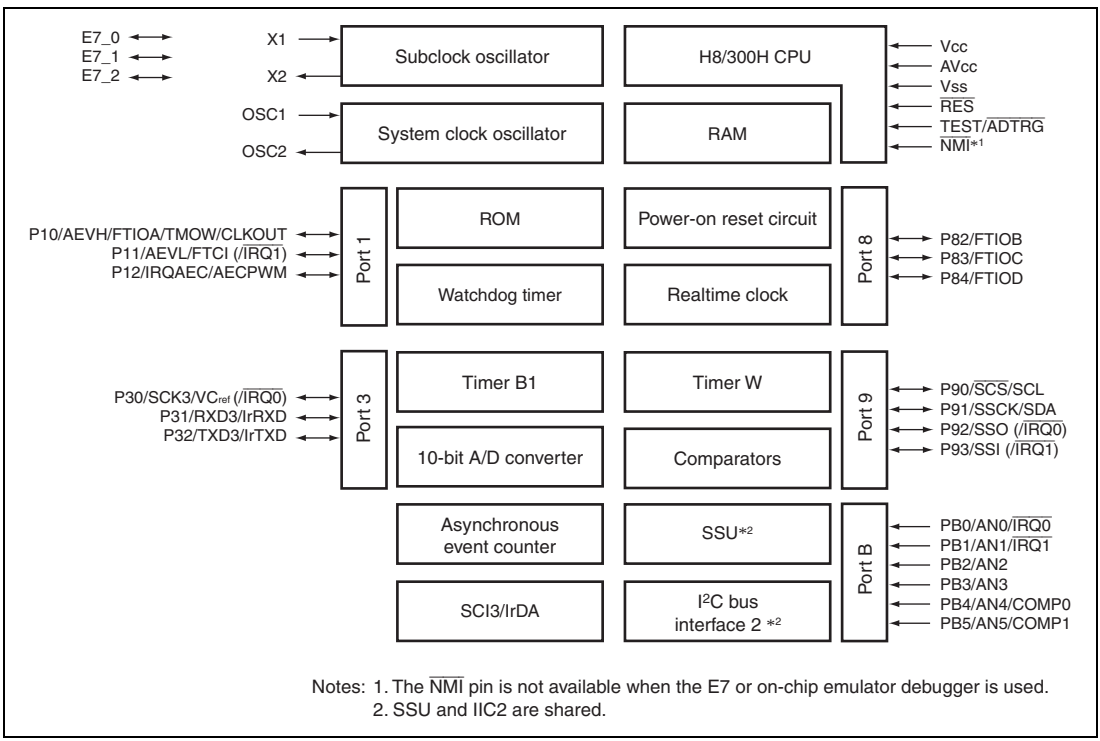


Figure 1.1 Internal Block Diagram of H8/38602R Group

1.3 Pin Assignment

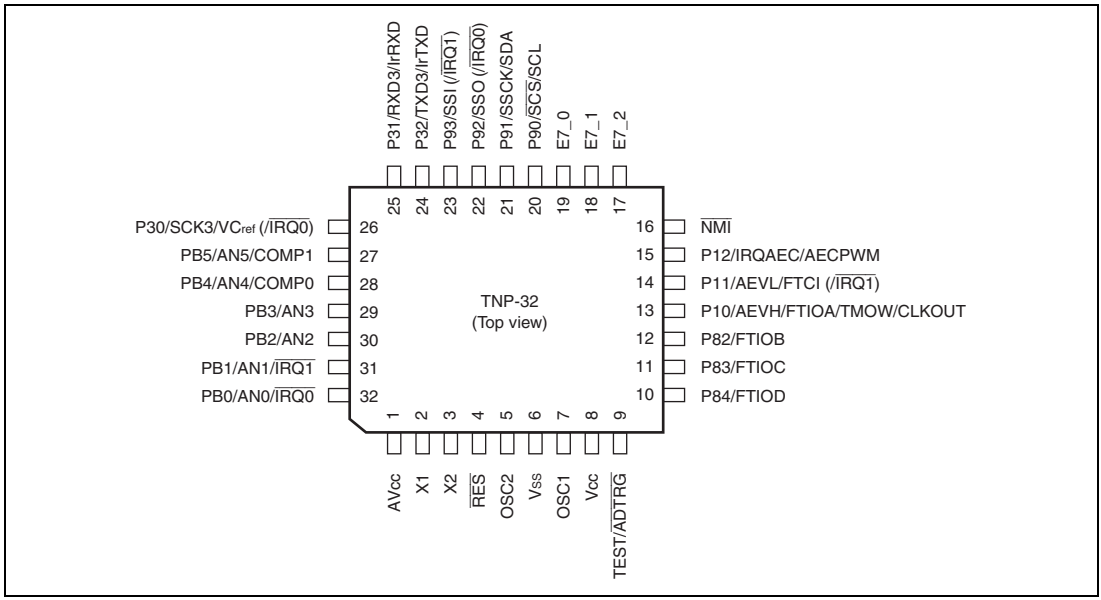


Figure 1.2 Pin Assignment of H8/38602R Group (TNP-32)

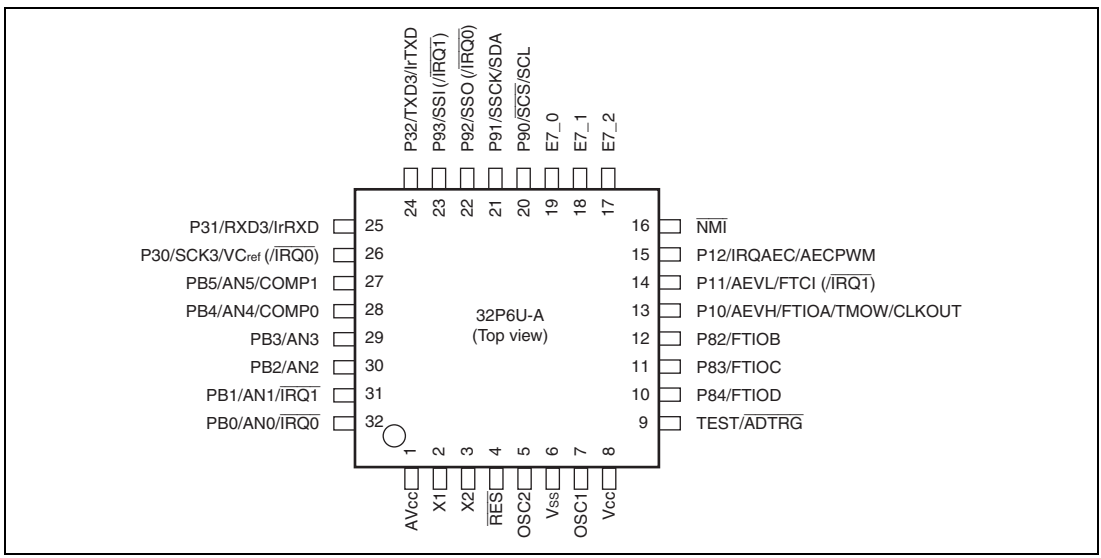


Figure 1.3 Pin Assignment of H8/38602R Group (32P6U-A)

1.4 Pin Functions

Table 1.1 Pin Functions

Type	Symbol	Pin No.	I/O	Functions
Power supply pins	Vcc	8	Input	Power supply pin. Connect this pin to the system power supply.
	Vss	6	Input	Ground pin. Connect this pin to the system power supply (0 V).
	AVcc	1	Input	Analog power supply pin for the A/D converter. When the A/D converter is not used, connect this pin to the system power supply.
Clock pins	OSC1	7	Input	These pins connect with crystal or ceramic resonator for the system clock, or can be used to input an external clock. See section 4, Clock Pulse Generators, for a typical connection.
	OSC2	5	Output	
	X1	2	Input	These pins connect with a 32.768- or 38.4-kHz crystal resonator for the subclock. See section 4, Clock Pulse Generators, for a typical connection.
	X2	3	Output	
	CLKOUT	13	Output	Clock output pin.
System control	$\overline{\text{RES}}$	4	Input	Reset pins. The power-on reset circuit is incorporated. When externally driven low, the chip is reset.
	TEST	9	Input	Test pins. Also used as the $\overline{\text{ADTRG}}$ pin. When this pin is not used as the $\overline{\text{ADTRG}}$ pin, users cannot use this pin. Connect this pin to Vss. When this pin is used as the $\overline{\text{ADTRG}}$ pin, see section 17.4.2, External Trigger Input Timing.

Type	Symbol	Pin No.	I/O	Functions
Interrupt pins	$\overline{\text{NMI}}$	16	Input	NMI interrupt request pin. Non-maskable interrupt request input pin. In the F-ZTAT version, the level on this pin determines whether to enter the user or boot mode when the reset state is released. To enter the user mode, pull up this pin to the Vcc level.
	$\overline{\text{IRQ0}}$	32 (22, 26)	Input	External interrupt request input pins. Can select the rising or falling edge.
	$\overline{\text{IRQ1}}$	31 (14, 23)	Input	
	IRQAEC	15	Input	Interrupt input pin for the asynchronous event counter. This pin enables the asynchronous event input.
Timer W	FTCI	14	Input	External event input pin.
	FTIOA to FTIOD	13 to 10	I/O	Output compare output/input capture input/PWM output pins.
Asynchronous event counter (AEC)	AEVL	14	Input	Event input pins for input to the asynchronous event counter.
	AEVH	13	Input	
	AECPWM	15	Output	PWM output pin for the AEC.
RTC	TMOW	13	Output	Divided clock output pin for the RTC.
Serial communication interface 3 (SCI3)	SCK3	26	I/O	SCI3 clock I/O pin.
	RXD3/ IrRXD	25	Input	SCI3 data input pins or data input pins for the IrDA format.
	TXD3/ IrTXD	24	Output	SCI3 data output pins or data output pins for the IrDA format.
Synchronous serial communication unit (SSU)	$\overline{\text{SCS}}$	20	I/O	SSU chip select I/O pin.
	SSCK	21	I/O	SSU clock I/O pin.
	SSI	23	I/O	SSU transmit/receive data I/O pins.
	SSO	22	I/O	
I^2C bus interface 2 (IIC2)	SDA	21	I/O	IIC data I/O pin.
	SCL	20	I/O	IIC clock I/O pin.

Type	Symbol	Pin No.	I/O	Functions
A/D converter	AN0 to AN5	32 to 27	Input	Analog data input pins for the A/D converter.
	$\overline{\text{ADTRG}}$	9	Input	External trigger input pin for the A/D converter.
Comparators	COMP0	28	Input	Analog data input pins for the comparator.
	COMP1	27		
	VCref	26	Input	Reference voltage pin for external input of threshold voltage of the comparator analog input pins.
I/O ports	P10 to P12	13 to 15	I/O	3-bit I/O pins. Input or output can be designated for each bit by means of the port control register 1 (PCR1).
	P30 to P32	26 to 24	I/O	3-bit I/O pins. Input or output can be designated for each bit by means of the port control register 3 (PCR3).
	P82 to P84	12 to 10	I/O	3-bit I/O pins. Input or output can be designated for each bit by means of the port control register 8 (PCR8).
	P90 to P93	20 to 23	I/O	4-bit I/O pins. Input or output can be designated for each bit by means of the port control register 9 (PCR9).
	PB0 to PB5	32 to 27	Input	6-bit input-only pins.
E7	E7_0 E7_1 E7_2	19 to 17	—	E7 emulator interface pins. E7_2 selects whether the on-chip oscillator is used. E7_2 is pulled up or down by a 100-kΩ resistance. For details, see section 4, Clock Pulse Generators.

Section 2 CPU

This LSI 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.

- Upward-compatible with H8/300 CPUs
  Can execute H8/300 CPUs 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-register architecture
  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, @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 one or two states

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

- Power-down state
  Transition to power-down state by SLEEP instruction

2.1 Address Space and Memory Map

The address space of this LSI is 64 Kbytes, which includes the program area and the data area. Figure 2.1 shows the memory map.

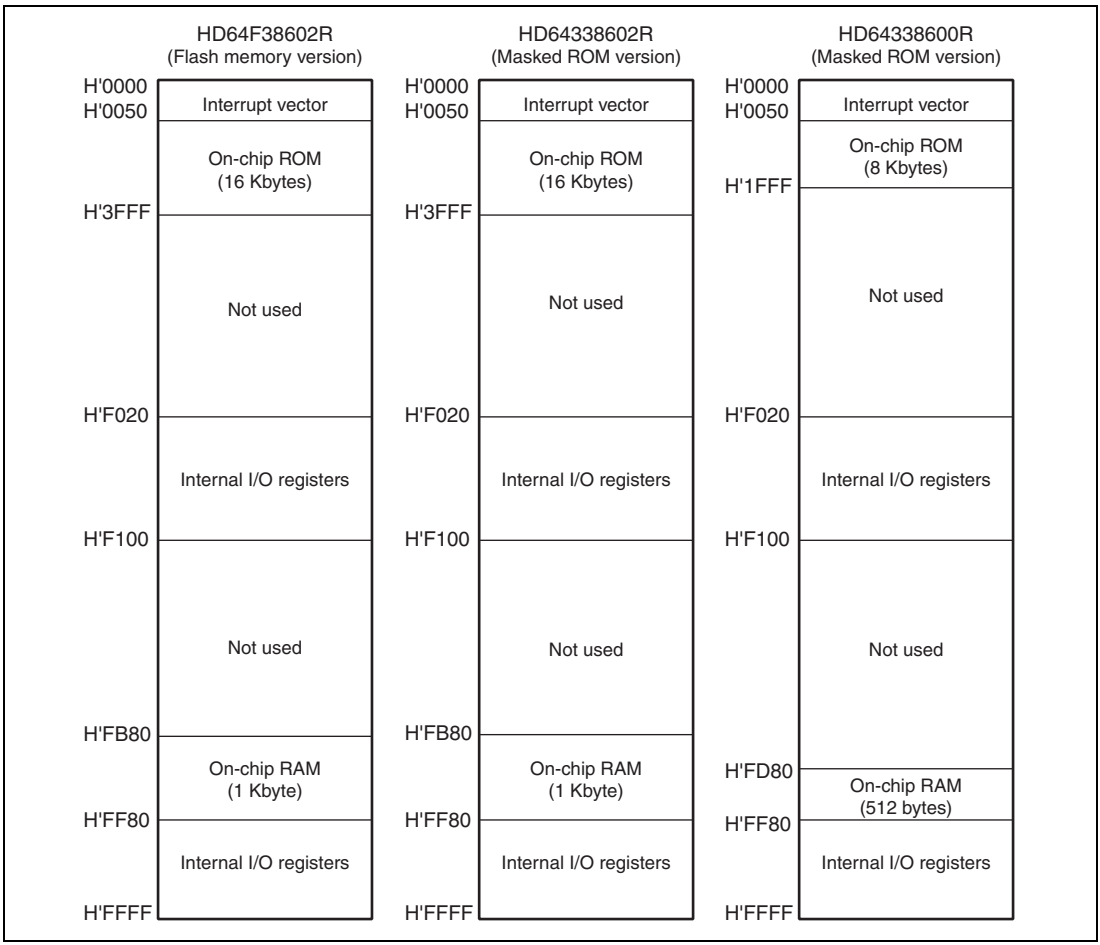


Figure 2.1 Memory Map

2.2 Register Configuration

The H8/300H CPU has the internal registers shown in figure 2.2. There are two types of registers; general registers and control registers. The control registers are a 24-bit program counter (PC), and an 8-bit condition-code register (CCR).

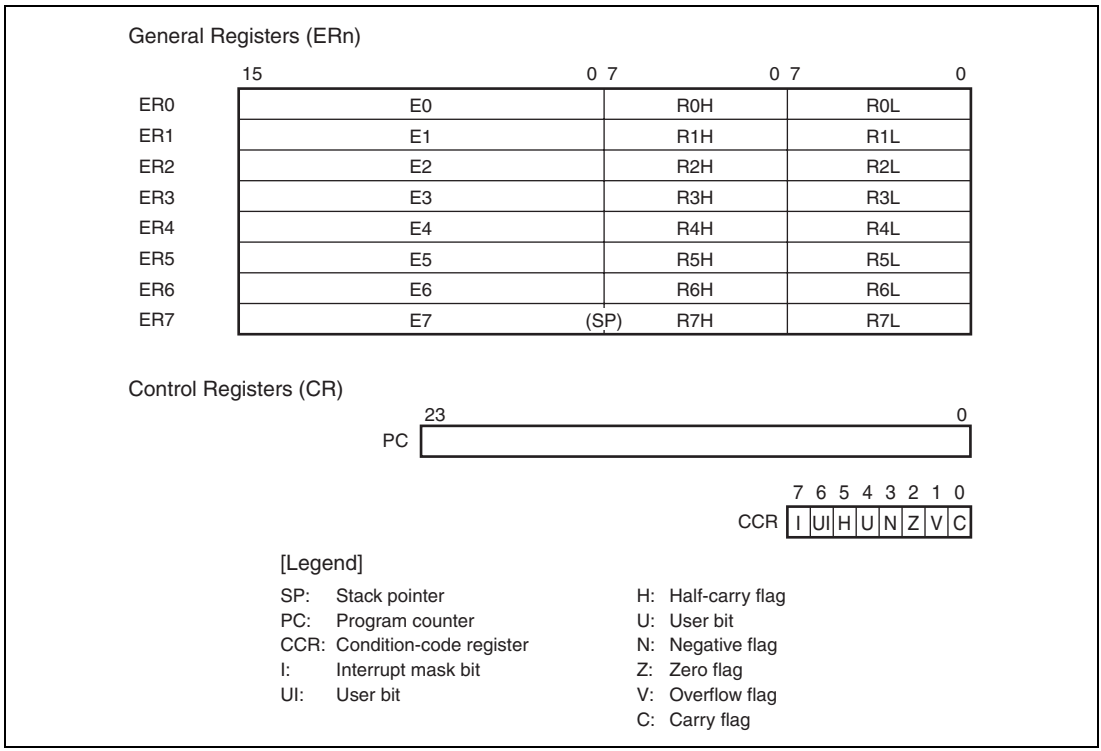


Figure 2.2 CPU Registers

2.2.1 General Registers

The H8/300H CPU has eight 32-bit general registers. These general registers are all functionally identical and can be used as both address registers and data 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. Figure 2.3 illustrates the usage of the general registers. When the general registers are used as 32-bit registers or 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 of sixteen 16-bit registers. The E registers (E0 to E7) are also referred to as extended registers.

The R registers divide into 8-bit 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.

The usage of each register can be selected independently.

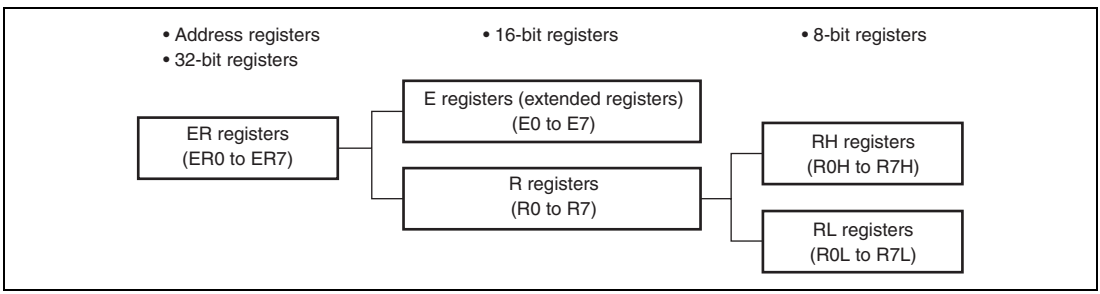


Figure 2.3 Usage of General Registers

General register ER7 has the function of the stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2.4 shows the relationship between the stack pointer and the stack area.

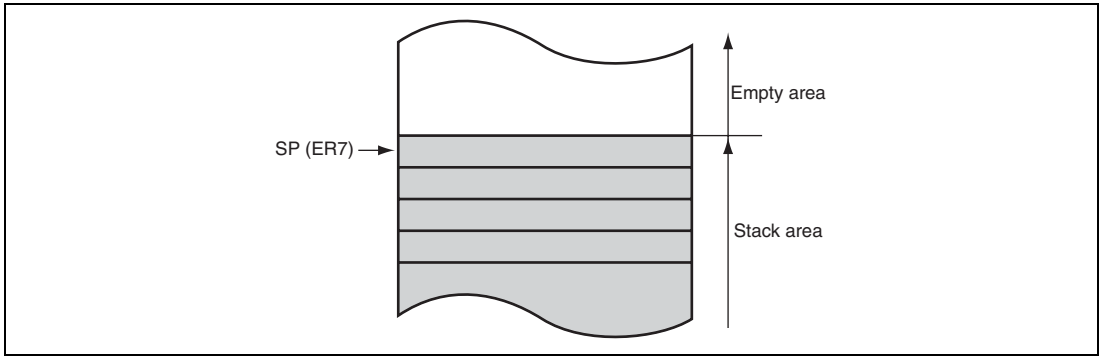


Figure 2.4 Relationship between Stack Pointer and Stack Area

2.2.2 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), 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.

2.2.3 Condition-Code Register (CCR)

This 8-bit register contains internal CPU status information, including an 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.

Some instructions leave flag bits unchanged. Operations can be performed on the CCR bits by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used as branching conditions for conditional branch (Bcc) instructions.

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

Bit	Bit Name	Initial Value	R/W	Description
7	I	1	R/W	Interrupt Mask Bit Masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting. The I bit is set to 1 at the start of an exception-handling sequence.
6	UI	Undefined	R/W	User Bit Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.
5	H	Undefined	R/W	Half-Carry Flag 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.
4	U	Undefined	R/W	User Bit Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit of data as a sign bit.
2	Z	Undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.
1	V	Undefined	R/W	Overflow Flag Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times.
0	C	Undefined	R/W	Carry Flag 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 indicate a carry The carry flag is also used as a bit accumulator by bit manipulation instructions.

2.3 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.

2.3.1 General Register Data Formats

Figure 2.5 shows the data formats in general registers.

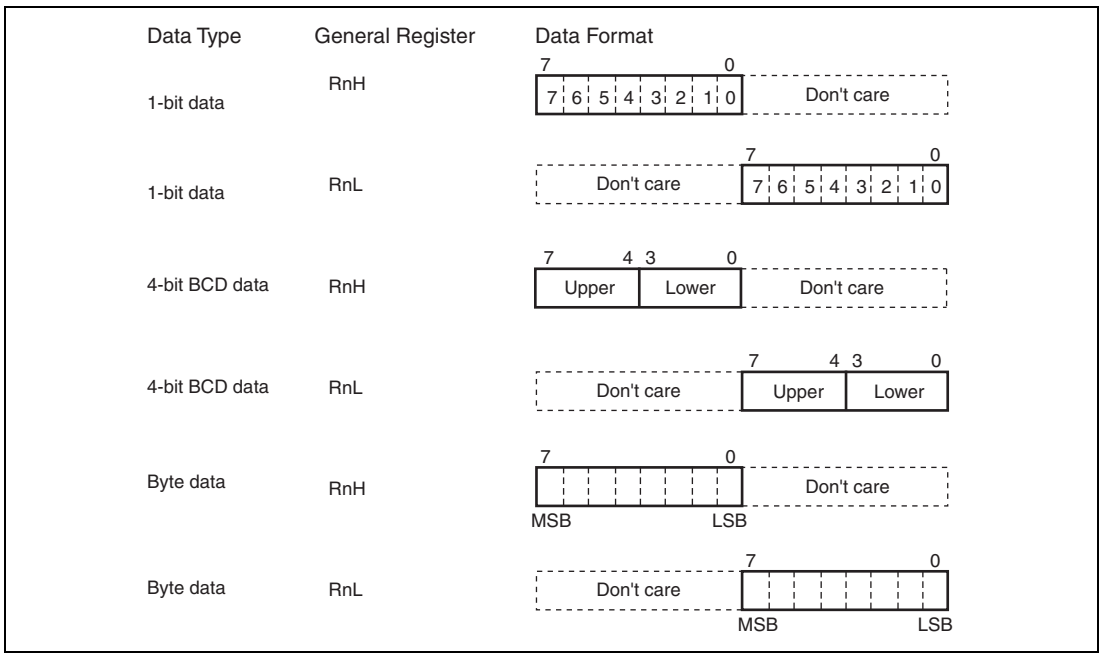


Figure 2.5 General Register Data Formats (1)

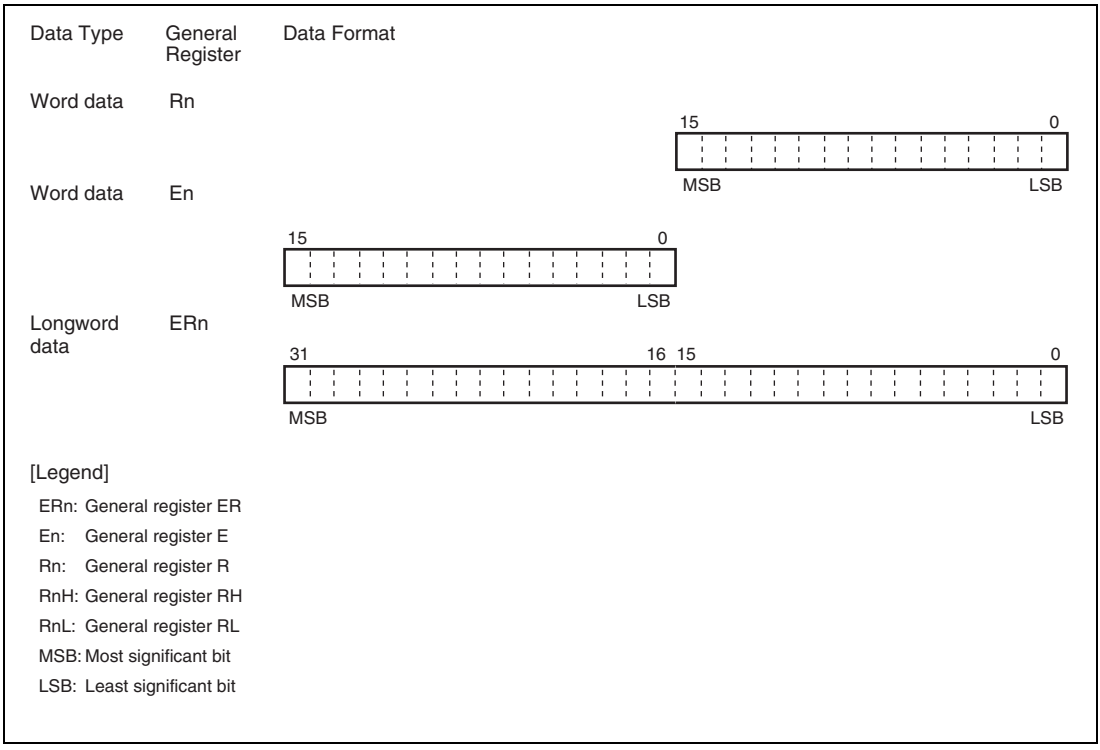


Figure 2.5 General Register Data Formats (2)

2.3.2 Memory Data Formats

Figure 2.6 shows the data formats in memory. The H8/300H CPU can access word data and longword data in memory, however 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, an address error does not occur, however the least significant bit of the address is regarded as 0, so access begins the preceding address. This also applies to instruction fetches.

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

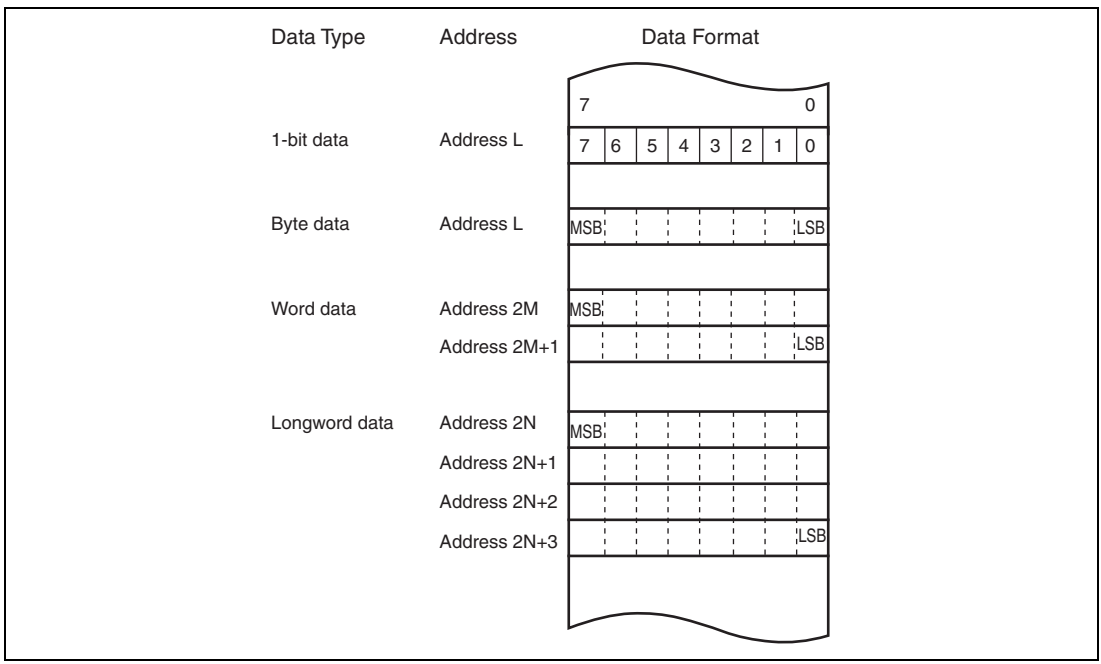


Figure 2.6 Memory Data Formats

2.4 Instruction Set

2.4.1 Table of Instructions Classified by Function

The H8/300H CPU has 62 instructions. Tables 2.2 to 2.9 summarize the instructions in each functional category. The notation used in tables 2.2 to 2.9 is defined below.

Table 2.1 Operation Notation

Symbol	Description
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 in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical XOR
→	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 registers/address register (ER0 to ER7).

Table 2.2 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 this LSI.
MOVTPE	B	Rs → (EAs) Cannot be used in this LSI.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. 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. PUSH.L ERn is identical to MOV.L ERn, @–SP.

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

Table 2.3 Arithmetic Operations Instructions (1)

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 byte 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 on byte 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 the 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.

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

Table 2.3 Arithmetic Operations Instructions (2)

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

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

Table 2.4 Logic Operations 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 (logical complement) of general register contents.

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

Table 2.5 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 flag.

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

Table 2.6 Bit Manipulation Instructions (1)

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 three 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 three bits of a general register.
BNOT	B	¬ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operand and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the 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.

Note: * Refers to the operand size.
B: Byte

Table 2.6 Bit Manipulation Instructions (2)

Instruction	Size*	Function
BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C XORs 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 XORs 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: * Refers to the operand size.
B: Byte

Table 2.7 Branch 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: * Bcc is the general name for conditional branch instructions.

Table 2.8 System Control Instructions

Instruction	Size*	Function
TRAPA	—	Starts trap-instruction exception handling
RTE	—	Returns from an exception-handling routine.
SLEEP	—	Causes a transition to a power-down state.
LDC	B/W	(EAs) → CCR Moves the source operand contents to the CCR. The CCR 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 CCR with immediate data.
ORC	B	CCR ∨ #IMM → CCR Logically ORs the CCR with immediate data.
XORC	B	CCR ⊕ #IMM → CCR Logically XORs the CCR with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

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

Table 2.9 Block Data Transfer Instructions

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. Starting from the address set in ER5, transfers data for the number of bytes set in R4L or R4 to the address location set in ER6. Execution of the next instruction begins as soon as the transfer is completed.

2.4.2 Basic Instruction Formats

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

Figure 2.7 shows examples of instruction formats.

- 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 four bits of the instruction. Some instructions have two operation fields.
- Register Field

  Specifies a general register. Address registers are specified by 3 bits, and 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. A24-bit address or displacement is treated as a 32-bit data in which the first 8 bits are 0 (H'00).
- Condition Field

  Specifies the branching condition of Bcc instructions.

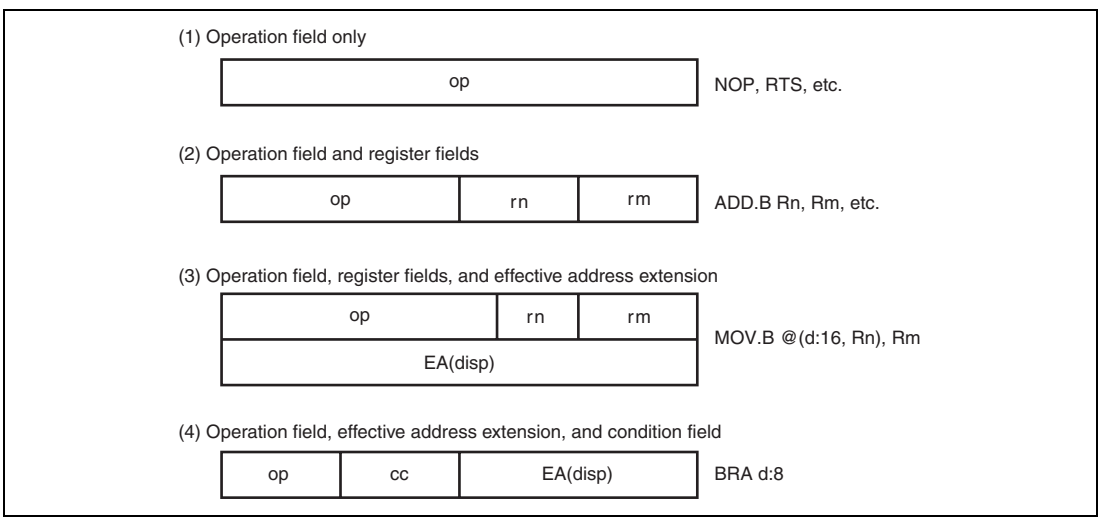


Figure 2.7 Instruction Formats

2.5 Addressing Modes and Effective Address Calculation

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

2.5.1 Addressing Modes

The H8/300H CPU supports the eight addressing modes listed in table 2.10. 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.4, Combinations of Instructions and Addressing Modes.

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 the absolute addressing mode (@aa:8) 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 2.10 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 Register indirect with pre-decrement	@ERn+ @–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

Register Direct—Rn

The register field of the instruction specifies an 8-, 16-, or 32-bit general 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.

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 on memory.

Register Indirect with Displacement**—@(d:16, ERn) or @(d:24, ERn)**

A 16-bit or 24-bit displacement contained in the instruction is added to an address register (ERn) specified by the register field of the instruction, and the lower 24 bits of the sum the address of a memory operand. A 16-bit displacement is sign-extended when added.

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 contains 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 the 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 is 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 the word or longword access, the register value should be even.

Absolute Address—@aa:8, @aa:16, @aa:24

The instruction code contains the absolute address of a memory operand. The absolute address may be 8 bits (@aa:8), 16 bits (@aa:16), or 24 bits (@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.

The access ranges of absolute addresses for this LSI are those shown in table 2.11, because the upper 8 bits are ignored.

Table 2.11 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

Immediate—#xx:8, #xx:16, or #xx:32

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

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

Program-Counter Relative—@(d:8, PC) or @(d:16, PC)

This mode is used in the BSR instruction. An 8-bit or 16-bit displacement contained in the instruction is sign-extended and added to the 24-bit PC contents to generate a 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.

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. Figure 2.8 shows how to specify branch address for in memory indirect mode. The upper bits of the absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF).

Note that the first part of the address range is also the exception vector area.

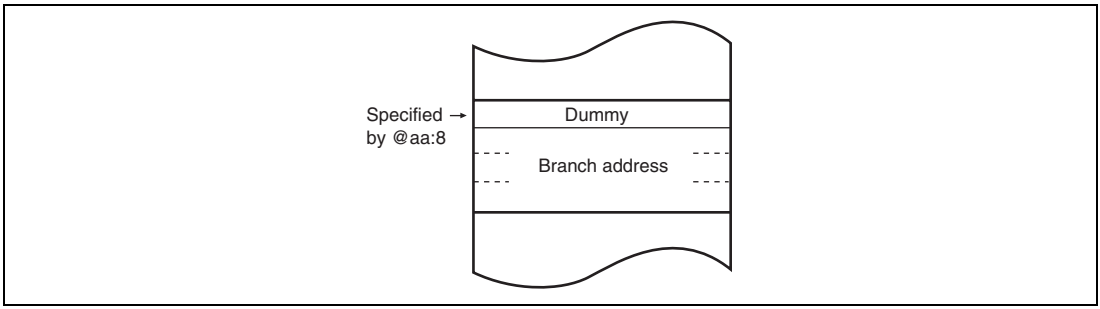


Figure 2.8 Branch Address Specification in Memory Indirect Mode

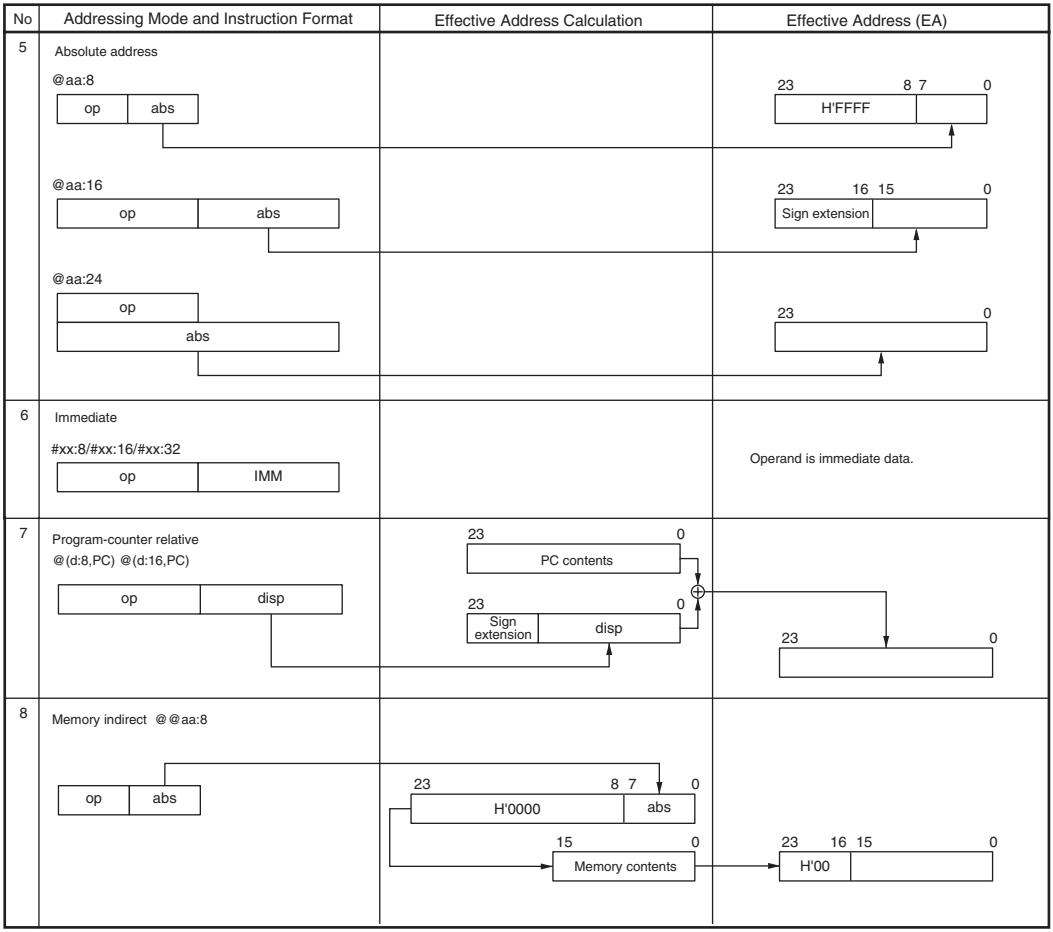
2.5.2 Effective Address Calculation

Table 2.12 indicates how effective addresses are calculated in each addressing mode. In this LSI the upper 8 bits of the effective address are ignored in order to generate a 16-bit effective address.

Table 2.12 Effective Address Calculation (1)

No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct(Rn) op rm rn		Operand is general register contents.
2	Register indirect(@ERn) op r	31 0 General register contents	23 0
3	Register indirect with displacement @(d:16,ERn) or @(d:24,ERn) op r disp	31 0 General register contents 31 0 Sign extension disp	23 0
4	Register indirect with post-increment or pre-decrement •Register indirect with post-increment @ERn+ op r •Register indirect with pre-decrement @-ERn op r	31 0 General register contents 1, 2, or 4 31 0 General register contents 1, 2, or 4 The value to be added or subtracted is 1 when the operand is byte size, 2 for word size, and 4 for longword size.	23 0 23 0

Table 2.12 Effective Address Calculation (2)



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

2.6 Basic Bus Cycle

CPU operation is synchronized by a system clock (ϕ) or a subclock (ϕ_{SUB}). The period from a rising edge of ϕ or ϕ_{SUB} to the next rising edge is called one state. A bus cycle consists of two states.

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

Access to on-chip memory takes place in two states. The data bus width is 16 bits, allowing access in byte or word size. Figure 2.9 shows the on-chip memory access cycle.

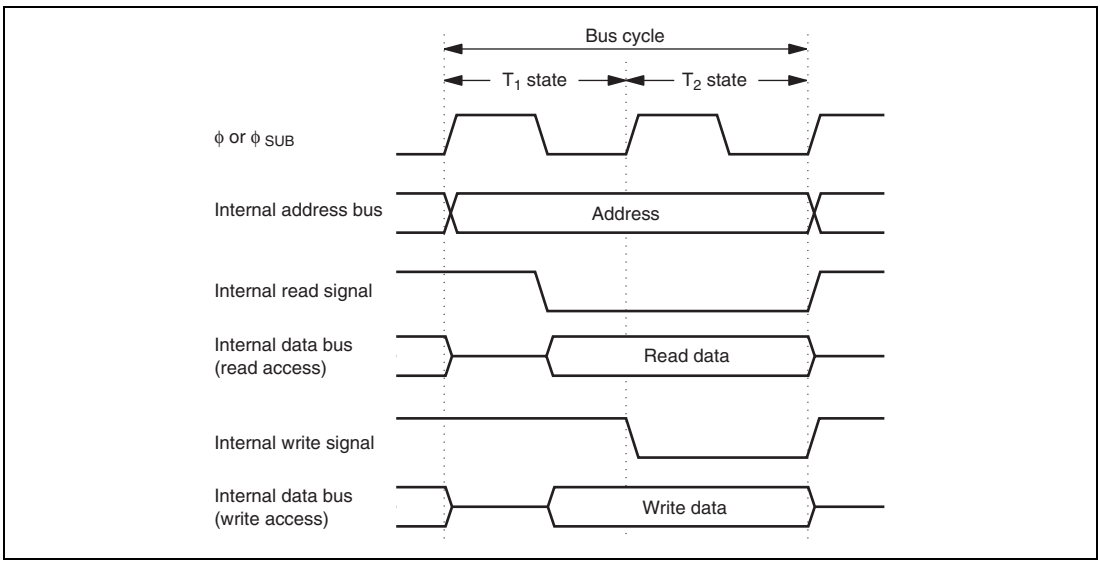


Figure 2.9 On-Chip Memory Access Cycle

2.6.2 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 and number of accessing states of each register, refer to section 20.1, Register Addresses (Address Order). 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, a bus cycle occurs twice. In two-state access, the operation timing is the same as that for on-chip memory.

Figure 2.10 shows the operation timing in the case of three-state access to an on-chip peripheral module.

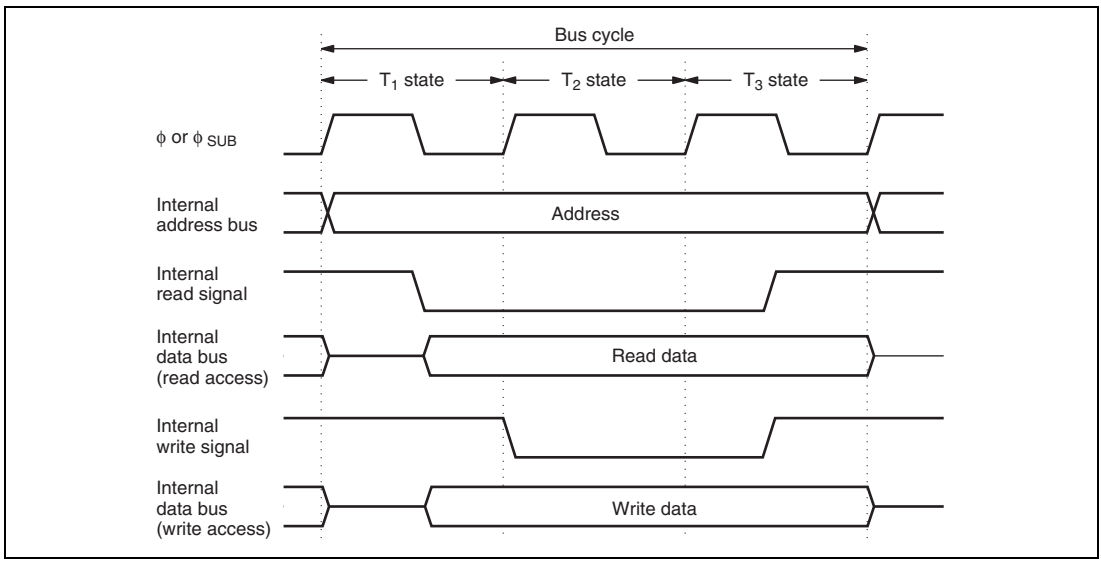


Figure 2.10 On-Chip Peripheral Module Access Cycle (3-State Access)

2.7 CPU States

There are four CPU states: the reset state, program execution state, program halt state, and exception-handling state. The program execution state includes active (high-speed or medium-speed) mode and subactive mode. For the program halt state, there are sleep (high-speed or medium-speed) mode, standby mode, watch mode, and subsleep mode. These states are shown in figure 2.11. Figure 2.12 shows the state transitions. For details on program execution state and program halt state, refer to section 5, Power-Down Modes. For details on exception handling, refer to section 3, Exception Handling.

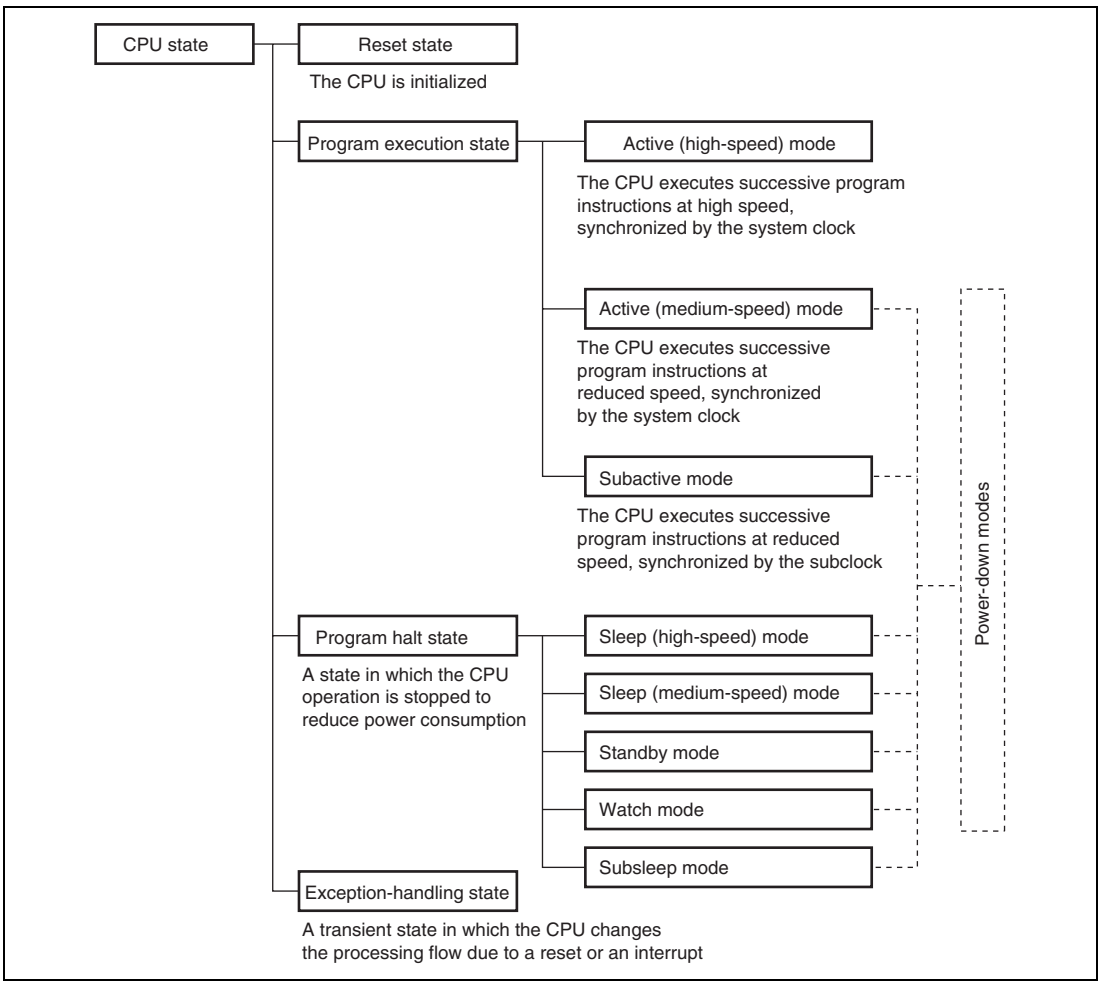


Figure 2.11 CPU Operating States

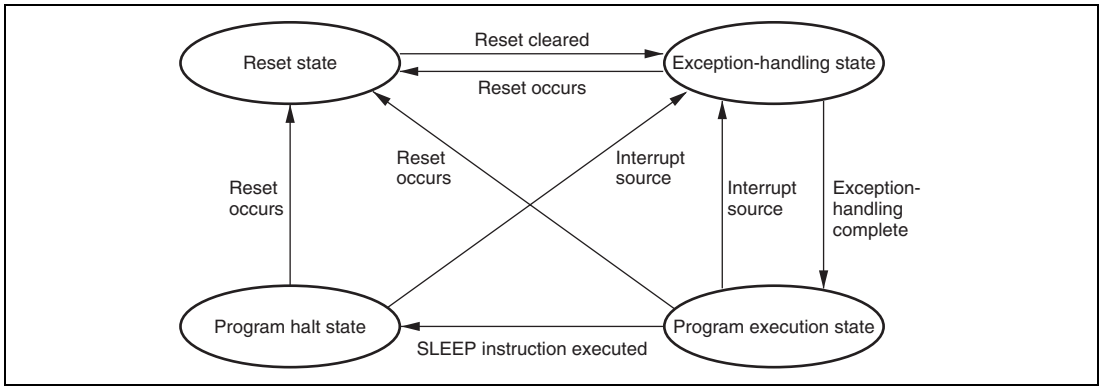


Figure 2.12 State Transitions

2.8 Usage Notes

2.8.1 Notes on Data Access to Empty Areas

The address space of this LSI includes empty areas in addition to the ROM, RAM, and on-chip I/O registers areas available to the user. When data is transferred from CPU to empty areas, the transferred data will be lost. This action may also cause the CPU to malfunction. When data is transferred from an empty area to CPU, the contents of the data cannot be guaranteed.

2.8.2 EEPMOV Instruction

EEPMOV is a block-transfer instruction and transfers the byte size of data indicated by R4 or R4L, which starts from the address indicated by R5, to the address indicated by R6. Set R4, R4L, and R6 so that the end address of the destination address (value of R6 + R4L or R6 + R4) does not exceed H'FFFF (the value of R6 must not change from H'FFFF to H'0000 during execution).

2.8.3 Bit-Manipulation Instruction

The BSET, BCLR, BNOT, BST, and BIST instructions read data from the specified address in byte units, manipulate the data of the target bit, and write data to the same address again in byte units. Special care is required when using these instructions in cases where two registers are assigned to the same address, or when a bit is directly manipulated for a port or a register containing a write-only bit, because this may rewrite data of a bit other than the bit to be manipulated.

Bit manipulation for two registers assigned to the same address

Example 1: Bit manipulation for the timer load register and timer counter

Figure 2.13 shows an example of a timer in which two timer registers are assigned to 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 takes place.

1. Data is read in byte units.
2. The CPU sets or resets the bit to be manipulated with the bit-manipulation instruction.
3. The written data is written again in byte units to the timer load register.

The timer 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 counter may be modified and the modified value may be written to the timer load register.

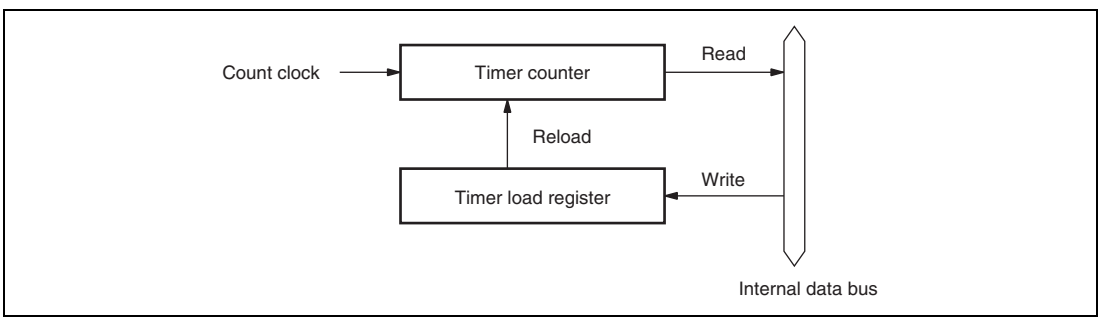


Figure 2.13 Example of Timer Configuration with Two Registers Allocated to Same Address

Example 2: When the BSET instruction is executed for port 5

P57 and P56 are input pins, with a low-level signal input at P57 and a high-level signal input at P56. P55 to P50 are output pins and output low-level signals. An example to output a high-level signal at P50 with a BSET instruction is shown below.

- Prior to executing BSET instruction

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
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

- BSET instruction executed instruction

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

The BSET instruction is executed for port 5.

- After executing BSET instruction

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
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

- Description on operation

1. When the BSET instruction is executed, first the CPU reads port 5.
   Since P57 and P56 are input pins, the CPU reads the pin states (low-level and high-level input).
   P55 to P50 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.
2. Next, the CPU sets bit 0 of the read data to 1, changing the PDR5 data to H'41.
3. Finally, the CPU writes H'41 to PDR5, completing execution of BSET instruction.

As a result of the BSET instruction, bit 0 in PDR5 becomes 1, and P50 outputs a high-level signal. However, bits 7 and 6 of PDR5 end up with different values. To prevent 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.

Prior to executing BSET instruction

```
MOV.B   #H'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.

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
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

- BSET instruction executed

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

The BSET instruction is executed designating the PDR5 work area (RAM0).

- After executing BSET instruction

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

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

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
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

P57 and P56 are input pins, with a low-level signal input at P57 and a high-level signal input at P56. P55 to P50 are output pins that output low-level signals. An example of setting the P50 pin as an input pin by the BCLR instruction is shown below. It is assumed that a high-level signal will be input to this input pin.

- Prior to executing BCLR instruction

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
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

- BCLR instruction executed

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

The BCLR instruction is executed for PCR5.

- After executing BCLR instruction

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
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

- Description on operation

1. 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.
2. Next, the CPU clears bit 0 in the read data to 0, changing the data to H'FE.
3. Finally, H'FE is written to PCR5 and BCLR instruction execution ends.

   As a result of this operation, bit 0 in PCR5 becomes 0, making P50 an input port. However, bits 7 and 6 in PCR5 change to 1, so that P57 and P56 change from input pins to output pins. To prevent this problem, store a copy of the PDR5 data in a work area in memory and manipulate data of the bit in the work area, then write this data to PDR5.

- Prior to executing BCLR instruction

```
MOV.B   #H'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.

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
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

- BCLR instruction executed

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

The BCLR instructions executed for the PCR5 work area (RAM0).

- After executing BCLR instruction

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

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

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
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

Section 3 Exception Handling

Exception handling is caused by a reset, a trap instruction (TRAPA), or interrupts.

- Reset

  A reset has the highest exception priority. Exception handling starts after the reset state is cleared by a negation of the $\overline{\text{RES}}$ signal. Exception handling is also started when the watchdog timer overflows. The exception handling executed at this time is the same as that for a reset by the $\overline{\text{RES}}$ pin.

- Trap Instruction

  Exception handling starts when a trap instruction (TRAPA) is executed. A vector address corresponding to a vector number from 0 to 3 which are specified in the instruction code is generated. Exception handling can be executed at all times in the program execution state, regardless of the setting of the I bit in CCR.

- Interrupts

  External interrupts other than the NMI and internal interrupts are masked by the I bit in CCR, and kept pending while the I bit is set to 1. Exception handling starts when the current instruction or exception handling ends, if an interrupt is requested.

3.1 Exception Sources and Vector Address

Table 3.1 shows the vector addresses and priority of each exception handling. When more than one interrupt is requested, handling is performed from the interrupt with the highest priority.

Table 3.1 Exception Sources and Vector Address

Source Origin	Exception Sources	Vector Number	Vector Address	Priority
Reset pin/ watchdog timer	Reset	0	H'0000 to H'0001	High
—	Reserved for system use	1 to 6	H'0002 to H'000D	
External interrupt	NMI	7	H'000E to H'000F	
Trap instruction TRAPA #0	Trap instruction #0	8	H'0010 to H'0011	
Trap instruction TRAPA #1	Trap instruction #1	9	H'0012 to H'0013	
Trap instruction TRAPA #2	Trap instruction #2	10	H'0014 to H'0015	
Trap instruction TRAPA #3	Trap instruction #3	11	H'0016 to H'0017	
—	Reserved for system use	12	H'0018 to H'0019	
CPU	Direct transition by executing the SLEEP instruction	13	H'001A to H'001B	
—	Reserved for system use	14, 15	H'001C to H'001F	
External interrupts	IRQ0	16	H'0020 to H'0021	
	IRQ1	17	H'0022 to H'0023	
	IRQAEC	18	H'0024 to H'0025	
—	Reserved for system use	19, 20	H'0026 to H'0029	
Comparators	COMP0	21	H'002A to H'002B	
	COMP1	22	H'002C to H'002D	
RTC	0.25-second overflow	23	H'002E to H'002F	
	0.5-second overflow	24	H'0030 to H'0031	
	Second periodic overflow	25	H'0032 to H'0033	
	Minute periodic overflow	26	H'0034 to H'0035	
	Hour periodic overflow	27	H'0036 to H'0037	
	Day-of-week periodic overflow	28	H'0038 to H'0039	
	Week periodic overflow	29	H'003A to H'003B	
	Free-running overflow	30	H'003C to H'003D	Low

Source Origin	Exception Sources	Vector Number	Vector Address	Priority
WDT	WDT overflow (interval timer)	31	H'003E to H'003F	High
Asynchronous event counter	Asynchronous event counter overflow	32	H'0040 to H'0041	
Timer B1	Overflow	33	H'0042 to H'0043	
Synchronous serial communication unit (SSU)/	Overrun error (SSU) Transmit data empty (SSU) Transmit end (SSU) Receive data full (SSU) Conflict error (SSU)/	34	H'0044 to H'0045	
IIC2*	Transmit data empty (IIC2) Transmit end (IIC2) Receive data full (IIC2) NACK detection (IIC2) Arbitration (IIC2) Overrun error (IIC2)			
Timer W	Input capture A/compare match A Input capture B/compare match B Input capture C/compare match C Input capture D/compare match D Overflow	35	H'0046 to H'0047	
—	Reserved for system use	36	H'0048 to H'0049	
SCI3	Transmit end Transmit data empty Receive data full Overrun error Framing error Parity error	37	H'004A to H'004B	
A/D converter	A/D conversion end	38	H'004C to H'004D	
—	Reserved for system use	39	H'004E to H'004F	Low

Note: * The SSU and IIC share the same vector address. When using the IIC, shift the SSU to standby mode using CKSTPR2.

3.2 Reset

A reset has the highest exception priority.

There are three sources to generate a reset. Table 3.2 lists the reset sources.

Table 3.2 Reset Sources

Reset Source	Description
$\overline{\text{RES}}$ pin	Low level input
Power-on reset circuit	When the power supply voltage (Vcc) rises For details, see section 19, Power-On Reset Circuit.
Watchdog timer	When the counter overflows For details, see section 12, Watchdog Timer.

3.2.1 Reset Exception Handling

When a reset source is generated, all the processing in execution is terminated and this LSI enters the reset state. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized by a reset.

To ensure that this LSI is reset, handle the $\overline{\text{RES}}$ pin as shown below.

- When power is supplied, or the system clock oscillator is stopped
  Hold the $\overline{\text{RES}}$ pin low until oscillation of the system clock oscillator has stabilized.
- When the system clock oscillator is operating
  Hold the $\overline{\text{RES}}$ pin low for the t_{REL} state, which is specified as the electrical characteristics.

After a reset source is generated, this LSI starts reset exception handling as follows.

1. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized, and the I bit in CCR is set to 1.
2. The reset exception handling vector address (H'0000 and H'0001) is read and transferred to the PC, and then program execution starts from the address indicated by the PC.

The reset exception handling sequence by the $\overline{\text{RES}}$ pin is shown in figure 3.1.

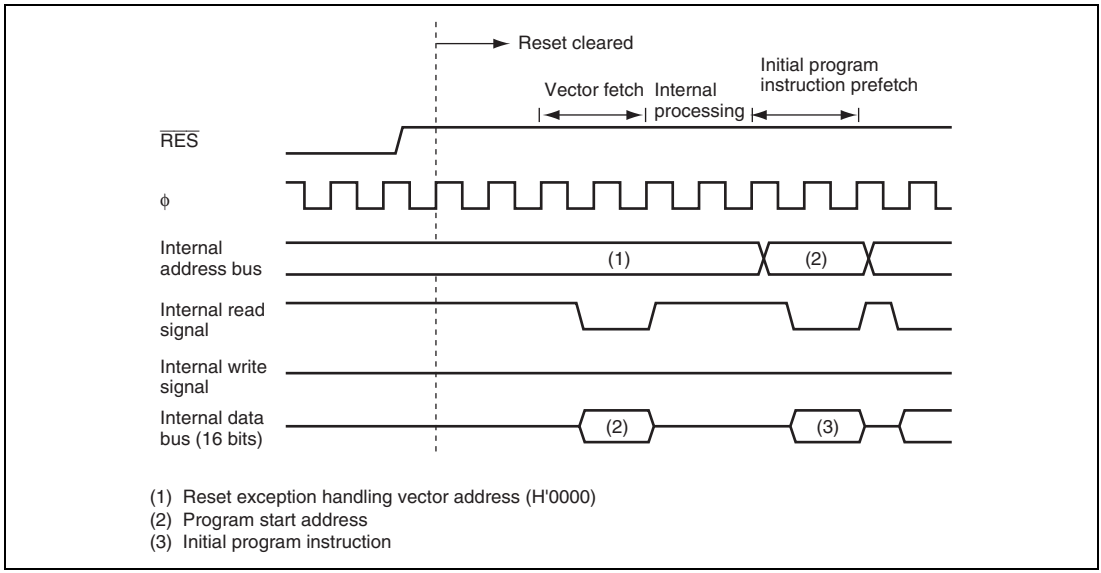


Figure 3.1 Reset Exception Handling Sequence

3.2.2 Interrupt Immediately after Reset

Immediately after a reset, if an interrupt is accepted before the stack pointer (SP) is initialized, PC and CCR will not be pushed onto the stack correctly, resulting in program runaway. To prevent this, immediately after reset exception handling all interrupts are masked. For this reason, the initial program instruction is always executed immediately after a reset. This instruction should initialize the stack pointer (e.g. MOV.L #xx: 32, SP).

3.3 Input/Output Pins

Table 3.3 shows the pin configuration of the interrupt controller.

Table 3.3 Pin Configuration

Name	I/O	Function
$\overline{\text{NMI}}$	Input	Nonmaskable external interrupt pin Rising or falling edge can be selected
IRQAEC	Input	Maskable external interrupt pin Rising, falling, or both edges can be selected
$\overline{\text{IRQ1}}$	Input	Maskable external interrupt pins Rising or falling edge can be selected
$\overline{\text{IRQ0}}$	Input	

3.4 Register Descriptions

The interrupt controller has the following registers.

- Interrupt edge select register (IEGR)
- Interrupt enable register 1 (IENR1)
- Interrupt enable register 2 (IENR2)
- Interrupt flag register 1 (IRR1)
- Interrupt flag register 2 (IRR2)

3.4.1 Interrupt Edge Select Register (IEGR)

IEGR selects whether interrupt requests of the $\overline{\text{NMI}}$, $\overline{\text{ADTRG}}$, $\overline{\text{IRQ1}}$, and $\overline{\text{IRQ0}}$ pins are generated at the rising edge or falling edge.

Bit	Bit Name	Initial Value	R/W	Descriptions
7	NMIEG	0	R/W	NMI Edge Select 0: Detects a falling edge of the $\overline{\text{NMI}}$ pin input 1: Detects a rising edge of the $\overline{\text{NMI}}$ pin input
6	—	0	—	Reserved This bit is always read as 0.
5	ADTRGNEG	0	R/W	ADTRG Edge Select 0: Detects a falling edge of the $\overline{\text{ADTRG}}$ pin input 1: Detects a rising edge of the $\overline{\text{ADTRG}}$ pin input
4 to 2	—	All 0	—	Reserved The write value should always be 0.
1	IEG1	0	R/W	IRQ1 Edge Select 0: Detects a falling edge of the $\overline{\text{IRQ1}}$ pin input 1: Detects a rising edge of the $\overline{\text{IRQ1}}$ pin input
0	IEG0	0	R/W	IRQ0 Edge Select 0: Detects a falling edge of the $\overline{\text{IRQ0}}$ pin input 1: Detects a rising edge of the $\overline{\text{IRQ0}}$ pin input

3.4.2 Interrupt Enable Register 1 (IENR1)

IENR1 enables the RTC, IRQAEC, IRQ1, and IRQ0 interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	IENRTC	0	R/W	RTC Interrupt Request Enable The RTC interrupt request is enabled when this bit is set to 1.
6 to 3	—	All 0	—	Reserved The write value should always be 0.
2	IENEC2	0	R/W	IRQAEC Interrupt Request Enable The IRQAEC interrupt request is enabled when this bit is set to 1.
1	IEN1	0	R/W	IRQ1 Interrupt Request Enable The IRQ1 interrupt request is enabled when this bit is set to 1.
0	IEN0	0	R/W	IRQ0 Interrupt Request Enable The IRQ0 interrupt request is enabled when this bit is set to 1.

3.4.3 Interrupt Enable Register 2 (IENR2)

IENR2 enables the A/D converter, timer B1, and asynchronous event counter interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved The write value should always be 0.
6	IENAD	0	R/W	A/D Converter Interrupt Request Enable The A/D converter interrupt request is enabled when this bit is set to 1.
5 to 3	—	All 0	—	Reserved The write value should always be 0.
2	IENTB1	0	R/W	Timer B1 Interrupt Request Enable The timer B1 interrupt request is enabled when this bit is set to 1.
1	—	0	—	Reserved The write value should always be 0.
0	IENEC	0	R/W	Asynchronous Event Counter Interrupt Request Enable The asynchronous event counter interrupt request is enabled when this bit is set to 1.

3.4.4 Interrupt Flag Register 1 (IRR1)

IRR1 indicates the IRQAEC, IRQ1, and IRQ0 interrupt request status.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	—	Reserved The write value should always be 0.
2	IRREC2	0	R/(W)*	IRQAEC Interrupt Request Flag [Setting condition] When the P12 pin is set to the IRQAEC/AECPWM pin and the specified edge is detected as the pin state [Clearing condition] When 0 is written to this bit
1	IRRI1	0	R/(W)*	IRQ1 Interrupt Request Flag [Setting condition] When the $\overline{\text{IRQ1}}$ pin is set as the interrupt input pin and the specified edge is detected [Clearing condition] When 0 is written to this bit
0	IRRI0	0	R/(W)*	IRQ0 Interrupt Request Flag [Setting condition] When the $\overline{\text{IRQ0}}$ pin is set as the interrupt input pin and the specified edge is detected [Clearing condition] When 0 is written to this bit

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

3.4.5 Interrupt Flag Register 2 (IRR2)

IRR2 indicates the interrupt request status of the A/D converter, timer B1, and asynchronous event counter.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved The write value should always be 0.
6	IRRAD	0	R/(W)*	A/D Converter Interrupt Request Flag [Setting condition] When A/D conversion ends [Clearing condition] When 0 is written to this bit
5 to 3	—	All 0	—	Reserved The write value should always be 0.
2	IRRTB1	0	R/(W)*	Timer B1 Interrupt Request Flag [Setting condition] When the timer B1 compare match or overflow occurs [Clearing condition] When 0 is written to this bit
1	—	0	—	Reserved The write value should always be 0.
0	IRREC	0	R/(W)*	Asynchronous Event Counter Interrupt Request Flag [Setting condition] When the asynchronous event counter overflows [Clearing condition] When 0 is written to this bit

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

3.5 Interrupt Sources

3.5.1 External Interrupts

There are four external interrupts: NMI, IRQAEC, IRQ1, and IRQ0.

(1) NMI Interrupt

NMI is the highest-priority interrupt, and is always accepted by the CPU regardless of the state of the I bit in CCR. The NMIEG bit in IEGR can be used to select whether an interrupt is requested at a rising edge or a falling edge on the $\overline{\text{NMI}}$ pin.

(2) IRQ1 and IRQ0 Interrupts

IRQ1 and IRQ0 interrupts are requested by input signals at $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ0}}$ pins.

Using the IEG1 and IEG0 bits in IEGR, it is possible to select whether an interrupt is generated by a rising or falling edge at $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ0}}$ pins.

When the specified edge is input while the $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ0}}$ pin functions are selected by PFCR and PMRB, the corresponding bit in IRR1 is set to 1 and an interrupt request is generated.

Clearing the IEN1 and IEN0 bits in IENR1 to 0 disables the interrupt request to be accepted. Setting the I bit in CCR to 1 masks all interrupts.

(3) IRQAEC Interrupts

An IRQAEC interrupt is requested by an input signal at the IRQAEC pin or IECPWM (PWM output for the AEC). When the IRQAEC pin is used as an external interrupt pin, clear the ECPWME bit in AEGSR to 0.

Using the AIEGS1 and AIEGS0 bits in AEGSR, it is possible to select whether an interrupt is generated by a rising edge, falling edge, or both edges.

When the IENEC2 bit in IENR1 is set to 1 and the specified edge is input, the corresponding bit in IRR1 is set to 1 and an interrupt request is generated. For details, see section 13, Asynchronous Event Counter (AEC).

3.5.2 Internal Interrupts

Internal interrupts generated from the on-chip peripheral modules have the following features:

- For each on-chip peripheral module, there are flags that indicate the interrupt request status, and enable bits that select enabling or disabling of these interrupts. Internal interrupts can be controlled independently. If an enable bit is set to 1, an interrupt request is sent to the interrupt controller.

3.6 Operation

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

Figure 3.2 shows a block diagram of the interrupt controller. Figure 3.3 shows the flow up to interrupt acceptance.

Interrupt operation is described as follows.

1. If an interrupt source whose interrupt enable register bit is set to 1 occurs, an interrupt request is sent to the interrupt controller.
2. When the interrupt controller receives an interrupt request, it sets the interrupt request flag.
3. 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 (see table 3.1).
4. The interrupt controller checks the I bit of CCR. If the I bit is 0, the selected interrupt request is accepted; if the I bit is 1, the interrupt request is held pending.
5. If the interrupt request is accepted, after processing of the current instruction is completed, both PC and CCR are pushed onto the stack. The state of the stack at this time is shown in figure 3.5. The PC value pushed onto the stack is the address of the first instruction to be executed upon return from interrupt handling.
6. The I bit of CCR is set to 1, masking further interrupts.
7. The vector address corresponding to the accepted interrupt is generated, and the interrupt handling routine located at the address indicated by the contents of the vector address is executed.

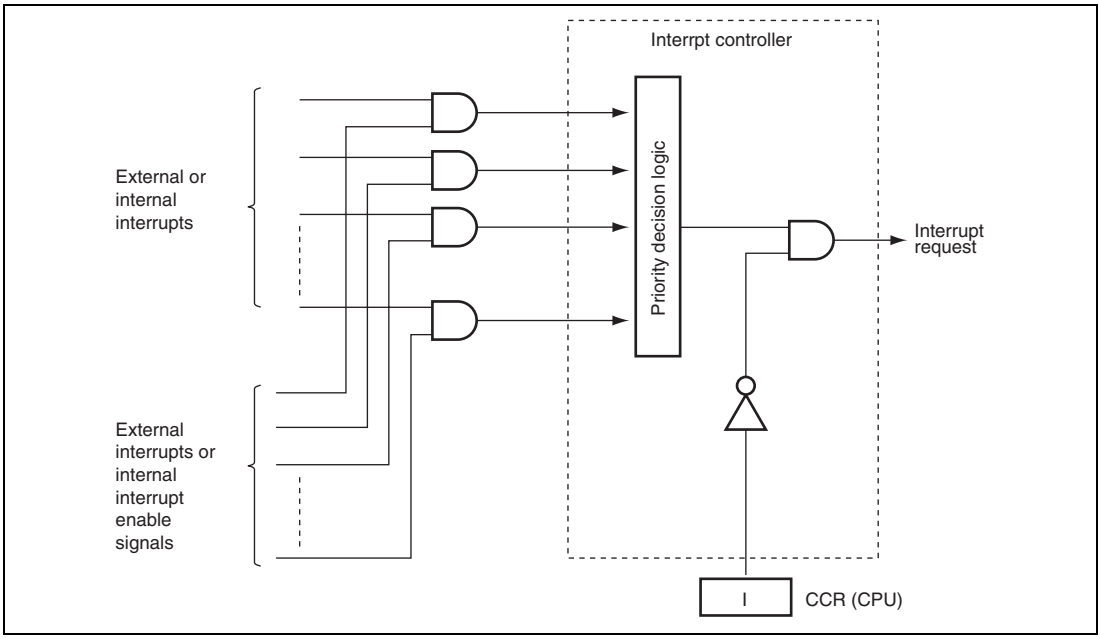


Figure 3.2 Block Diagram of Interrupt Controller

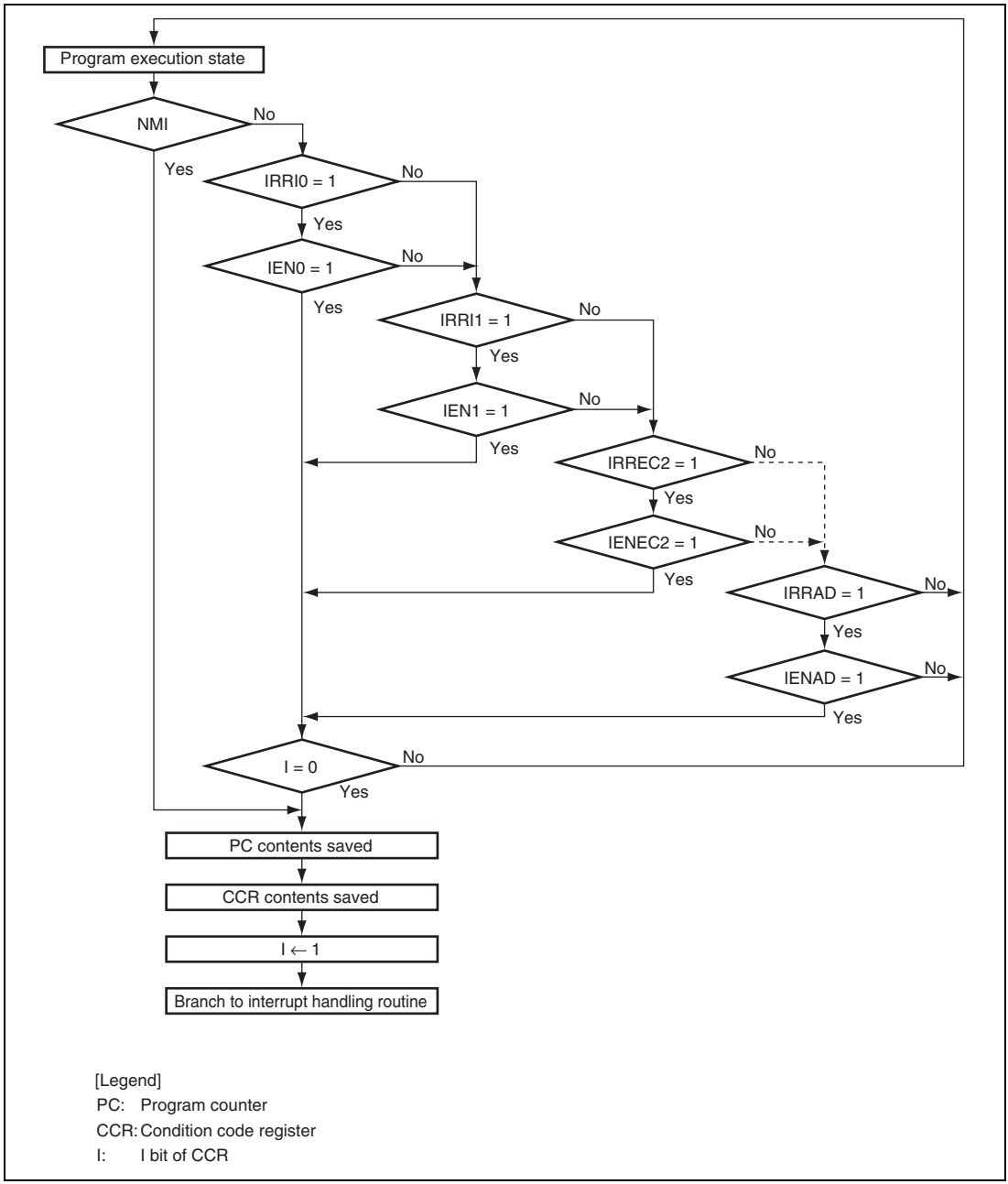


Figure 3.3 Flow up to Interrupt Acceptance

3.6.1 Interrupt Exception Handling Sequence

Figure 3.4 shows the interrupt exception handling sequence. The example shown is for the case where the program area and stack area are in a 16-bit and 2-state access space.

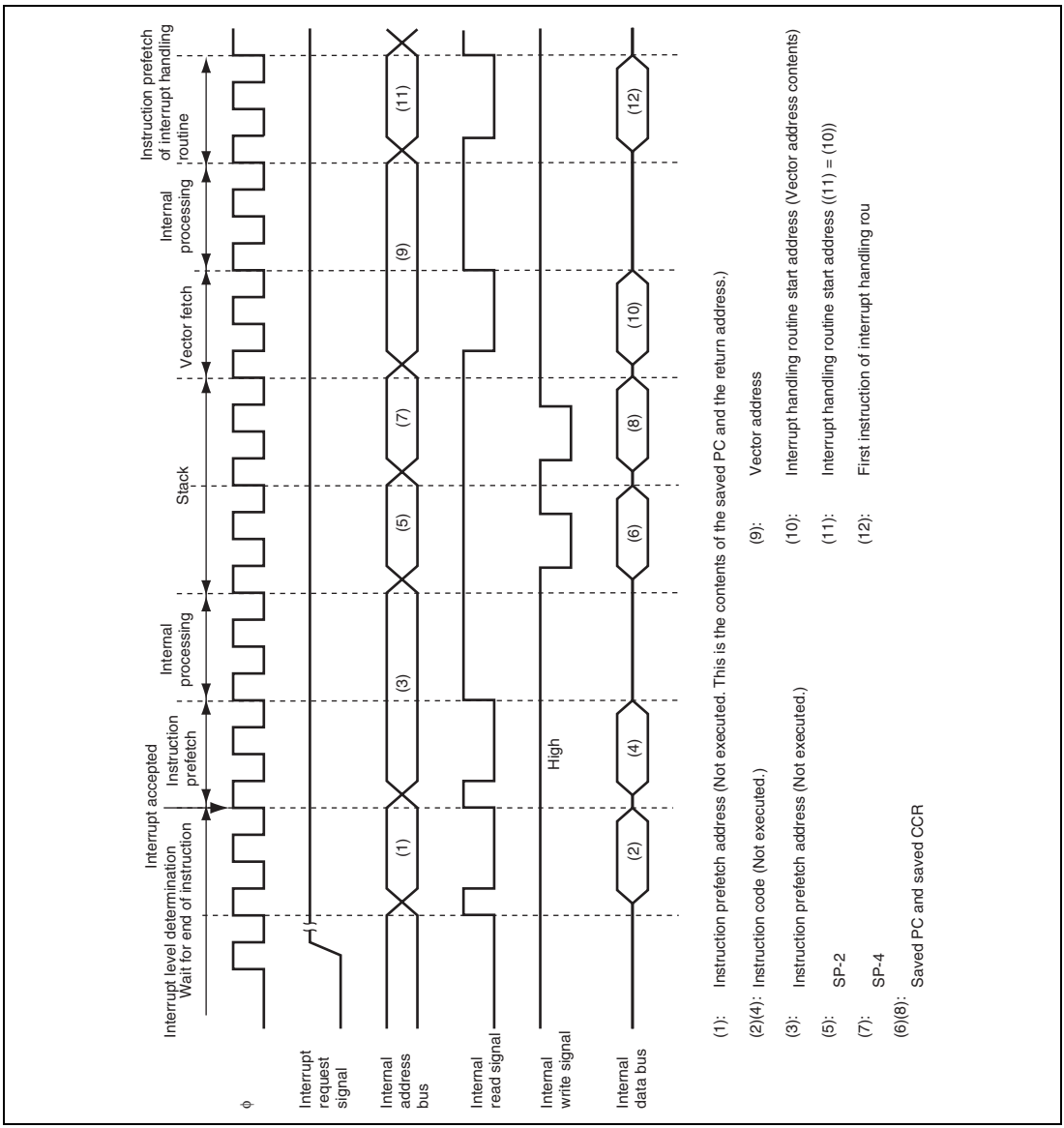
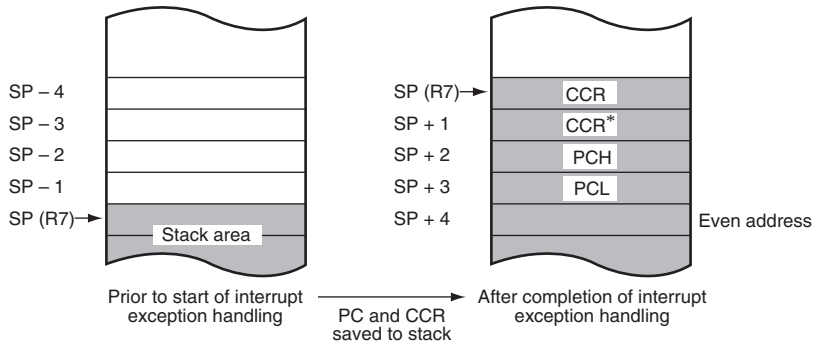


Figure 3.4 Interrupt Exception Handling Sequence

3.7 Stack Status after Exception Handling

Figures 3.5 shows the stack after completion of interrupt exception handling.



[Legend]
PCH: Upper 8 bits of program counter (PC)
PCL: 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.
* Ignored when returning from the interrupt handling routine.

Figure 3.5 Stack Status after Exception Handling

3.7.1 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	Total
Waiting time for completion of executing instruction*	1 to 23	15 to 37
Saving of PC and CCR to stack	4	
Vector fetch	2	
Instruction fetch	4	
Internal processing	4	

Note: * Excluding EEPMOV instruction.

3.8 Usage Notes

3.8.1 Notes on Stack Area Use

When word data is accessed in this LSI, 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. To save register values, use PUSH.W Rn (MOV.W Rn, @–SP) or PUSH.L ERn (MOV.L ERn, @–SP). To restore register values, use POP.W Rn (MOV.W @SP+, Rn) or POP.L ERn (MOV.L @SP+, ERn).

Setting an odd address in SP may cause a program to crash. An example is shown in figure 3.6.

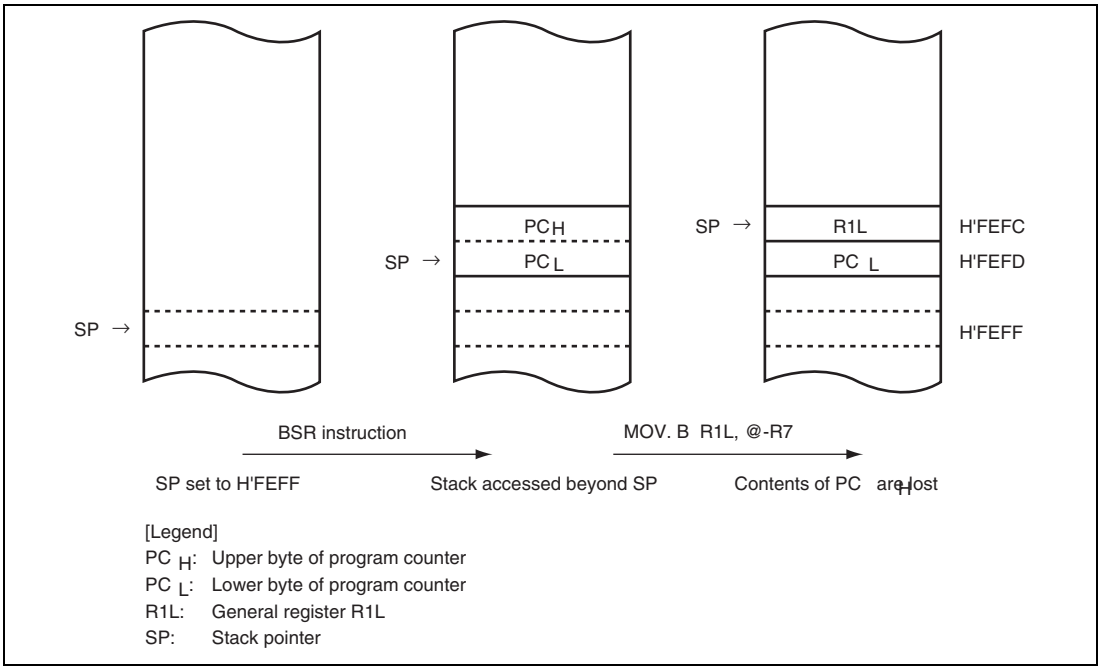


Figure 3.6 Operation when Odd Address is Set in SP

When CCR contents are saved to the stack during interrupt exception handling or restored when an RTE instruction 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.

3.8.2 Notes on Switching Functions of External Interrupt Pins

When PFCR and PMRB are rewritten to switch the functions of external interrupt pins and when the value of the ECPWME bit in AEGSR is rewritten to switch between selection and non-selection of IRQAEC, the following points should be observed.

When a pin function is switched by rewriting PFCR or PMRB that controls an external interrupt pin (IRQAEC, $\overline{\text{IRQ1}}$, or $\overline{\text{IRQ0}}$), the interrupt request flag is set to 1 at the time the pin function is switched, even if no valid interrupt is input at the pin. Be sure to clear the interrupt request flag to 0 after switching the pin function. When the value of the ECPWME bit in AEGSR that sets selection or non-selection of IRQAEC is rewritten, the interrupt request flag may be set to 1, even if a valid edge has not arrived on the selected IRQAEC or IECPWM (PWM output for the AEC). Therefore, be sure to clear the interrupt request flag to 0 after switching the pin function.

Figure 3.7 shows the procedure for setting a bit in PFCR and PMRB and clearing the interrupt request flag. This procedure also applies to AEGSR setting.

When switching a pin function, mask the interrupt before setting the bit in PFCR and PMRB (or AEGSR). After accessing PFCR and PMRB (or AEGSR), 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 to 0 is executed immediately after PFCR and PMRB (or AEGSR) access without executing an instruction, the flag will not be cleared.

An alternative method is to avoid the setting of interrupt request flags when pin functions are switched by keeping the pins at the high level. However, the procedure in figure 3.7 is recommended because IECPWM is an internal signal and determining its value is complicated.

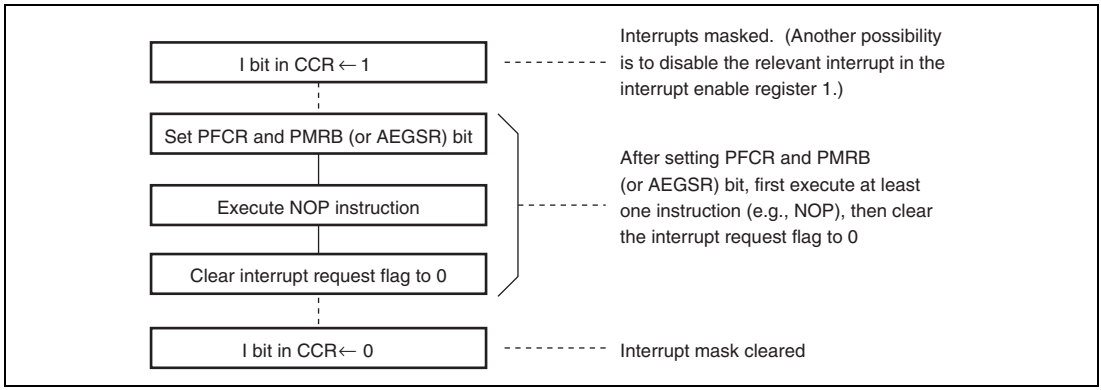


Figure 3.7 PFCR and PMRB (or AEGSR) Setting and Interrupt Request Flag Clearing Procedure

3.8.3 Method for Clearing Interrupt Request Flags

Use the recommended method given below when clearing the flags in interrupt request registers (IRR1 and IRR2).

- Recommended method

  Use a single instruction to clear flags. The bit manipulation instruction and byte-size data transfer instruction can be used. Two examples of program code for clearing IRRI1 (bit 1 in IRR1) are given below.

  Example 1:

  ```
  BCLR  #1,  @IRR1:8
  ```

  Example 2:

  ```
  MOV.B R1L, @IRR1:8 (set the value of R1L to B'11111101)
  ```

- Example of a malfunction

  When flags are cleared with multiple instructions, other flags might be cleared during execution of the instructions, even though they are currently set, and this will cause a malfunction.

  Here is an example in which IRRI0 is cleared and disabled in the process of clearing IRRI1 (bit 1 in IRR1).

  ```
  MOV.B @IRR1:8,R1L ......... IRRI0 = 0 at this time
  AND.B #B'11111101,R1L ..... Here, IRRI0 = 1
  MOV.B R1L,@IRR1:8 ......... IRRI0 is cleared to 0
  ```

  In the above example, it is assumed that an IRQ0 interrupt is generated while the AND.B instruction is executing.

  The IRQ0 interrupt is disabled because, although the original objective is clearing IRRI1, IRRI0 is also cleared.

3.8.4 Conflict between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to disable interrupts, the disabling becomes effective after execution of the instruction. When an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, and if an interrupt is generated during execution of the instruction, the interrupt concerned will still be enabled on completion of the instruction, and so interrupt exception handling for that interrupt will be executed on completion of the instruction. When an interrupt source flag is cleared to 0, the interrupt concerned will be ignored.

3.8.5 Instructions that Disable Interrupts

The instructions that disable interrupts are LDC, ANDC, ORC, and XORC.

When an interrupt request is generated, an interrupt is requested to the CPU. At that time, if the CPU is executing an instruction that disables interrupts, the CPU always executes the next instruction after the instruction execution is completed.

3.8.6 Interrupts during Execution of EEPMOV Instruction

Interrupt operation differs between the EEPMOV.B instruction and the EEPMOV.W instruction.

With the EEPMOV.B instruction, an interrupt request (including NMI) issued during transfer is not accepted until the transfer is completed.

With the EEPMOV.W instruction, even if an interrupt request other than the NMI is issued during transfer, the interrupt is not accepted until the transfer is completed. If the NMI interrupt request is issued, NMI exception handling starts at a break in the transfer cycle. The PC value saved on the stack in this case is the address of the next instruction.

Therefore, if an NMI interrupt is generated during execution of an EEPMOV.W instruction, the following coding should be used.

```
L1: EEPMOV.W
    MOV.W      R4,R4
    BNE        L1
```

3.8.7 IENR Clearing

When an interrupt request is disabled by clearing the interrupt enable register or when the interrupt flag register is cleared, the interrupt request should be masked (I bit = 1). If the above operation is executed while the I bit is 0 and conflict between the instruction execution and the interrupt request generation occurs, exception handling, which corresponds to the interrupt request generated after instruction execution of the above operation is completed, is executed.

Section 4 Clock Pulse Generators

The 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, system clock divider, and on-chip oscillator. The subclock pulse generator consists of a subclock oscillator, on-chip oscillator clock divider, and subclock divider. Figure 4.1 shows a block diagram of the clock pulse generators.

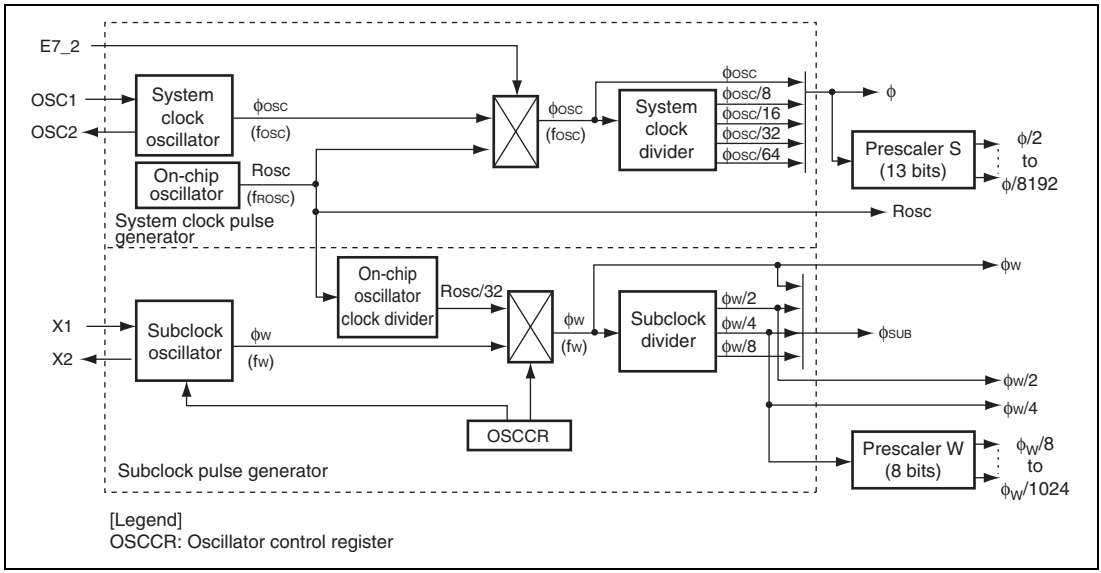


Figure 4.1 Block Diagram of Clock Pulse Generators

The reference clock signals that drive the CPU and on-chip peripheral modules are ϕ and ϕ_{SUB}. The system clock is divided by prescaler S to become a clock signal from $\phi/8192$ to $\phi/2$. $\phi_w/4$, which is 1/4th of the watch clock ϕ_w, is divided by prescaler W to become a clock signal from $\phi_w/1024$ to $\phi_w/8$. Both the system clock and subclock signals are provided to the on-chip peripheral modules.

4.1 Register Description

- Oscillator control register (OSCCR)

4.1.1 Oscillator Control Register (OSCCR)

OSCCR controls the subclock oscillator, on-chip feedback resistance, and on-chip oscillator.

Bit	Bit Name	Initial Value	R/W	Description
7	SUBSTP	0	R/W	Subclock Oscillator Control Controls start and stop of the subclock oscillator. When the subclock is not used, set this bit to 1. 0: Subclock oscillator operates 1: Subclock oscillator stops
6	RFCUT	0	R/W	On-chip Feedback Resistance Control Selects whether the on-chip feedback resistance in the system clock oscillator is used when an external clock is input or when the on-chip oscillator is used. After setting this bit in the state in which an external clock is input or the on-chip oscillator is used, temporarily transit to standby mode, watch mode, or subactive mode. The setting of whether the feedback resistance in the system clock oscillator is used or not takes effect when standby mode, watch mode, or subactive mode is entered. 0: On-chip feedback resistance in system clock oscillator is used 1: On-chip feedback resistance in system clock oscillator is not used
5	SUBSEL	0	R/W	Subclock Select Selects by which oscillator the subclock pulse generator operates. 0: Subclock oscillator operates 1: On-chip oscillator operates Note: The SUBSEL bit setting can be changed only when the subclock is not being used.

Bit	Bit Name	Initial Value	R/W	Description
4, 3	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
2	—	0	—	Reserved The write value should always be 0.
1	OSCF	—*	R	OSC Flag Indicates by which oscillator the system clock pulse generator operates. 0: System clock oscillator operates 1: On-chip oscillator operates (system clock oscillator is halted)
0	—	0	—	Reserved The write value should always be 0.

Note: * The value depends on the state of the E7_2 pin. Refer to table 4.1.

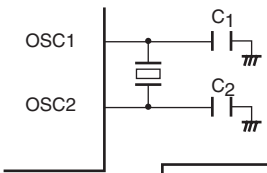
4.2 System Clock Oscillator

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

Either the system clock oscillator or on-chip oscillator can be selected, as shown in figure 4.1. For the selecting method, see section 4.2.4, On-Chip Oscillator Selection Method.

4.2.1 Connecting Crystal Resonator

Figure 4.2 shows a typical method of connecting a crystal resonator. An AT-cut parallel-resonance crystal resonator should be used. For notes on connecting, refer to section 4.5.2, Notes on Board Design.

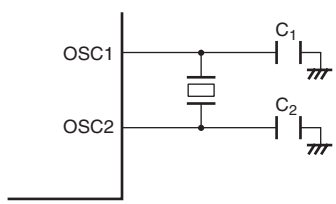


Frequency	Manufacturer	Product Type	C1, C2 Recommendation Value
4.194304 MHz	NIHON DEMPA KOGYO CO.,LTD.	NR-18	12 pF ±20%

Figure 4.2 Typical Connection to Crystal Resonator

4.2.2 Connecting Ceramic Resonator

Figure 4.3 shows a typical method of connecting a ceramic resonator. For notes on connecting, refer to section 4.5.2, Notes on Board Design.



Frequency	Manufacturer	C1, C2 Recommendation Value
4.19 MHz	Murata Manufacturing Co., Ltd.	15 pF (on-chip) 47 pF (on-chip)

Figure 4.3 Typical Connection to Ceramic Resonator

4.2.3 External Clock Input Method

Connect an external clock signal to pin OSC1, and leave pin OSC2 open. Figure 4.4 shows a typical connection. The duty cycle of the external clock signal must be 45 to 55%.

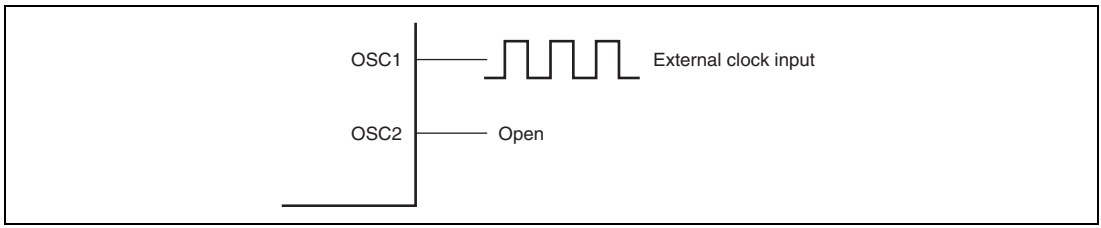


Figure 4.4 Example of External Clock Input

4.2.4 On-Chip Oscillator Selection Method

The on-chip oscillator is selected by the E7_2 pin input level during a reset. The methods for selecting the system clock oscillator and on-chip oscillator are shown in table 4.1. The input level on the E7_2 pin during a reset is pulled up or down using a resistor according to the selected oscillator, and fixed on exit from the reset state.

When the on-chip oscillator is selected, a resonator no longer needs to be connected to the OSC1 and OSC2 pins. In such a case, fix the OSC1 pin to GND or leave it open, and leave the OSC2 pin open.

Notes: 1. When programming or erasing the flash memory, such as performing on-board programming, the system clock oscillator must be selected. When the on-chip emulator is used, even though the on-chip oscillator is selected, connect a resonator or input an external clock.
2. When the on-chip debugger is connected, the value of the resistor should be high. When not connected, it is specified according to the selected oscillator.

Table 4.1 Methods for Selecting System Clock Oscillator and On-Chip Oscillator

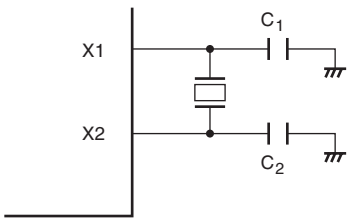
E7_2 Pin Input Level (during Reset)	Oscillator in System Clock Pulse Generator	OSCF
0	On-chip oscillator	1
1	System clock oscillator	0

4.3 Subclock Oscillator

A subclock can be provided by connecting a crystal resonator or inputting an external clock. Either the subclock oscillator or on-chip oscillator can be selected, as shown in figure 4.1. For the selecting method, see section 4.3.4, On-Chip Oscillator Selection Method.

4.3.1 Connecting 32.768-kHz/38.4-kHz Crystal Resonator

Figure 4.5 shows an example of connecting a 32.768-kHz or 38.4-kHz crystal resonator. Notes described in section 4.5.2, Notes on Board Design also apply to this connection.



Frequency	Manufacturer	Products Name	Equivalent Series Resistance
38.4 kHz	EPSON TOYOCOM CORPORATION	C-4-TYPE	30 kΩ (max.)
32.768 kHz	EPSON TOYOCOM CORPORATION	C-001R	35 kΩ (max.)

$C_1 = C_2 = 7$ pF (typ.)

Note: Consult with the crystal resonator manufacturer to determine the parameters.

Figure 4.5 Typical Connection to 32.768-kHz/38.4-kHz Crystal Resonator

1. When the resonator other than ones listed above is used, perform matching evaluation with the crystal resonator manufacture and connect it under the optimum condition. Even when the resonator listed above or the equivalent is used, as the oscillation characteristics depend on the board specification, perform matching evaluation on the mounting board.
2. Perform matching evaluation in the reset state (the $\overline{\text{RES}}$ pin is low) and on exit from the reset state (the $\overline{\text{RES}}$ pin is driven from low to high).

Figure 4.6 shows the equivalent circuit of the crystal resonator.

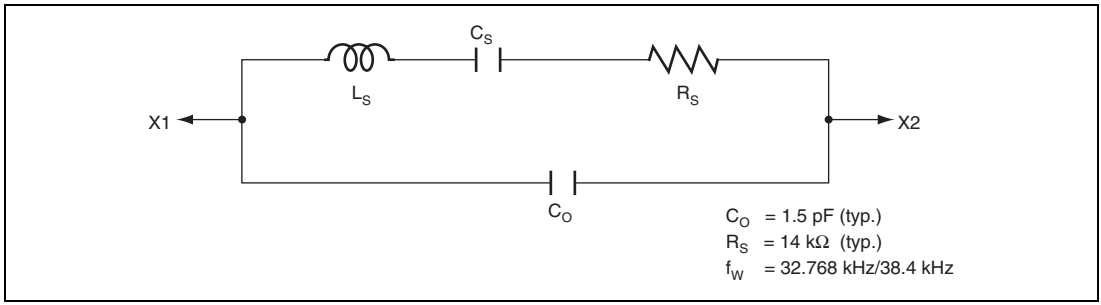


Figure 4.6 Equivalent Circuit of 32.768-kHz/38.4-kHz Crystal Resonator

4.3.2 Pin Connection when not Using Subclock

When the subclock is not used, connect the X1 pin to GND and leave the X2 pin open, as shown in figure 4.7.

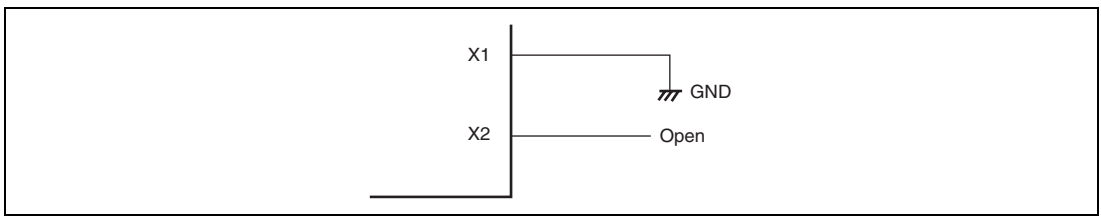


Figure 4.7 Pin Connection when not Using Subclock

4.3.3 External Clock Input Method

Connect the external clock to the X1 pin and leave the X2 pin open, as shown in figure 4.8.

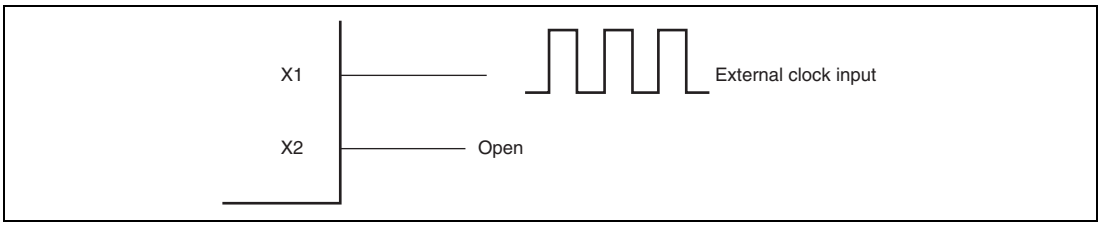


Figure 4.8 Pin Connection when Inputting External Clock

Frequency	Watch Clock (ϕ_w)
Duty	45% to 55%

4.3.4 On-Chip Oscillator Selection Method

The on-chip oscillator is selected by the SUBSEL bit in OSCCR. When the on-chip oscillator is selected, a resonator no longer needs to be connected to the X1 and X2 pins. In such a case, fix the X1 pin at GND.

4.4 Prescalers

This LSI is equipped with two on-chip prescalers (prescaler S and prescaler W), which have different input clocks.

Prescaler S is a 13-bit counter using the system clock (ϕ) as its input clock. Its prescaled outputs provide internal clock signals for on-chip peripheral modules. Prescaler W is an 8-bit counter using $\phi_w/4$, which is 1/4th of the watch clock ϕ_w, as its input clock. Its prescaled outputs provide internal clock signals for on-chip peripheral modules.

4.4.1 Prescaler S

Prescaler S is a 13-bit counter using the system clock (ϕ) as its input clock. A divided output is used as an internal clock of an on-chip peripheral module. Prescaler S is initialized to H'0000 at a reset, and starts counting up on exit from the reset state. In standby mode, watch mode, subactive mode, and subsleep mode, prescaler S stops and is initialized to H'0000. The CPU cannot read from or write to prescaler S.

The output from prescaler S is shared by the on-chip peripheral modules. In active (medium-speed) mode and sleep (medium-speed) mode, the clock input to prescaler S is determined by the division ratio designated by the MA1 and MA0 bits in SYSCR1.

4.4.2 Prescaler W

Prescaler W is an 8-bit counter using $\phi_w/4$, which is 1/4th of the watch clock ϕ_w, as its input clock. A divided output is used as an internal clock of an on-chip peripheral module. Prescaler W is initialized to H'00 at a reset, and starts counting up on exit from the reset state. In standby mode, prescaler W is halted. Even when transiting to watch mode, subactive mode, and subsleep mode, prescaler W continues operation.

4.5 Usage Notes

4.5.1 Note on Resonators and Resonator Circuits

Resonator characteristics are closely related to board design. Therefore, resonators should be assigned after being carefully evaluated by the user in the masked ROM version and flash memory version, with referring to the examples shown in this section. Resonator circuit constants will differ depending on a resonator, stray capacitance in its mounting circuit, and other factors. Suitable constants should be determined in consultation with the resonator manufacturer. Design the circuit so that the oscillator pin is never applied voltages exceeding its maximum rating.

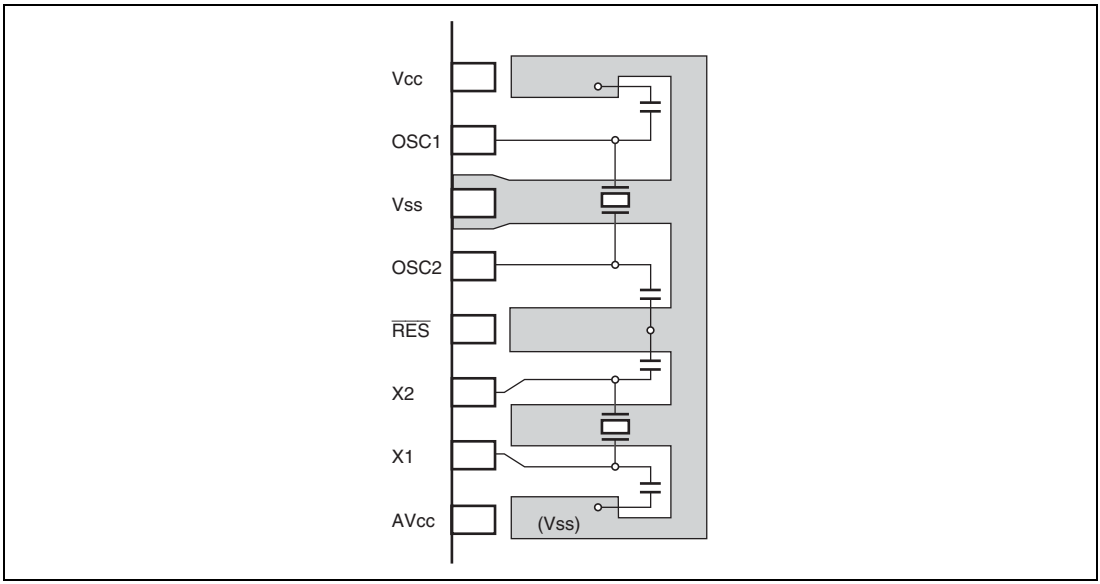


Figure 4.9 Example of Crystal and Ceramic Resonator Assignment

Figure 4.10 (1) shows an example measuring circuit with the negative resistance recommended by the resonator manufacturer. Note that if the negative resistance of the circuit is less than that recommended by the resonator manufacturer, it may be difficult to start the main oscillator.

If it is determined that oscillation does not occur because the negative resistance is lower than the level recommended by the resonator manufacturer, the circuit must be modified as shown in figure 4.10 (2) through (4). Which of the modification suggestions to use and the capacitor capacitance should be decided based upon evaluation results such as the negative resistance and the frequency deviation.

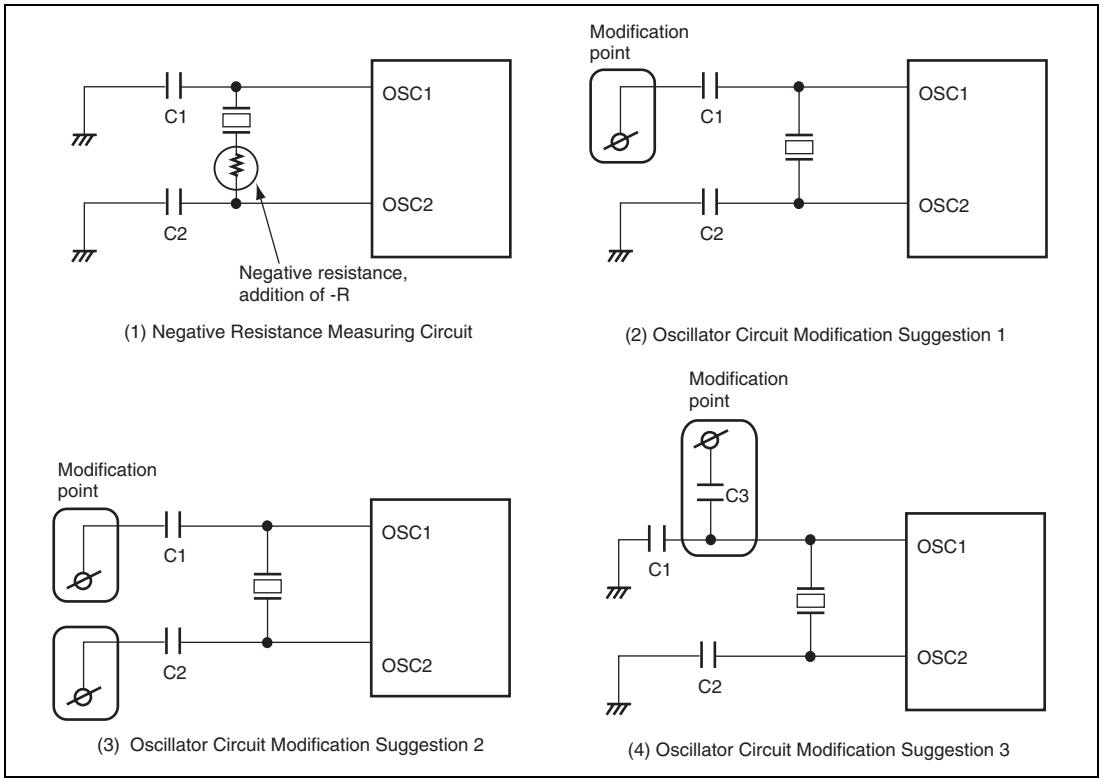


Figure 4.10 Negative Resistance Measurement and Circuit Modification Suggestions

4.5.2 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 resonator circuit to prevent induction from interfering with correct oscillation (see figure 4.11).

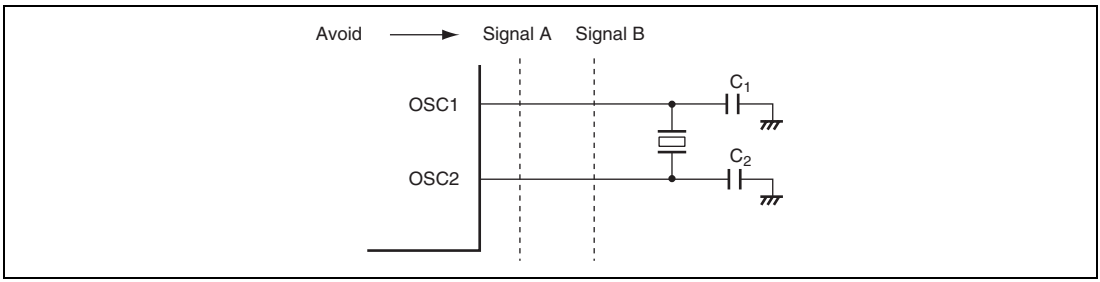


Figure 4.11 Example of Incorrect Board Design

Note: When a crystal resonator or ceramic resonator is connected, consult with the crystal resonator and ceramic resonator manufacturers to determine the circuit constants because the constants differ according to the resonator, stray capacitance of the mounting circuit, and so on.

4.5.3 Definition of Oscillation Stabilization Wait Time

Figure 4.12 shows the oscillation waveform (OSC2), system clock (ϕ), and microcomputer operating mode when a transition is made from standby mode, watch mode, or subactive mode, to active (high-speed/medium-speed) mode, with an resonator connected to the system clock oscillator.

As shown in figure 4.12, when a transition is made to active (high-speed/medium-speed) mode, from standby mode, watch mode, or subactive mode, in which the system clock oscillator is halted, the sum of the following two times (oscillation start time and wait time) is required.

(1) Oscillation Start Time

The time from the point at which the system clock oscillator oscillation waveform starts to change when an interrupt is generated, until the system clock starts to be generated.

(2) Wait Time

After the system clock is generated, the time required for the amplitude of the oscillation waveform to increase, the oscillation frequency to stabilize, and the CPU and peripheral functions to begin operating.

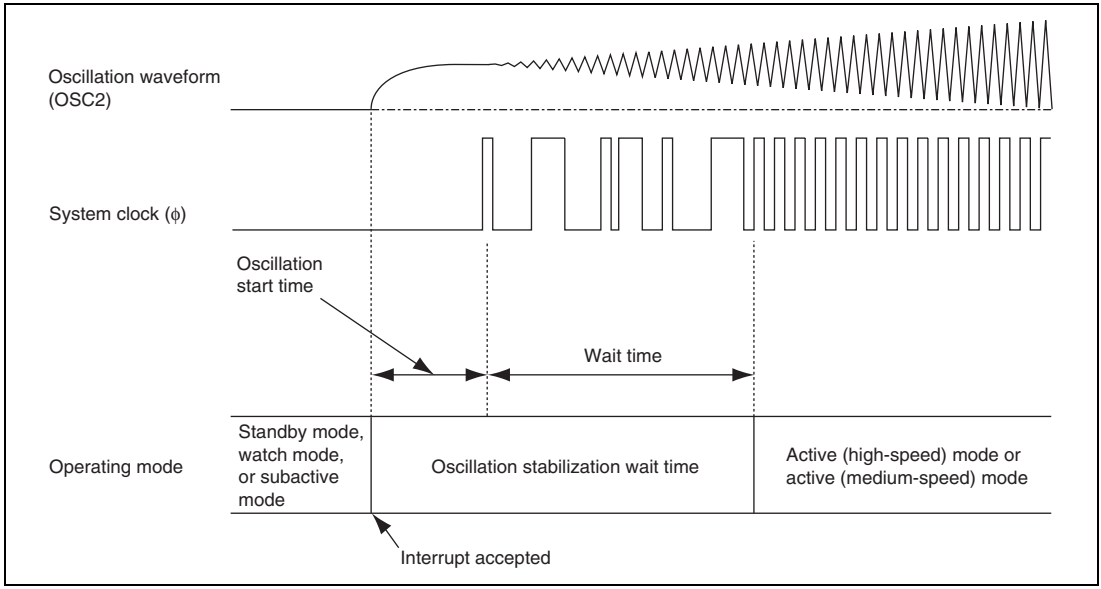


Figure 4.12 Oscillation Stabilization Wait Time

As the oscillation stabilization wait time required is the same as the oscillation stabilization time (t_{rc}) at power-on, specified in the AC characteristics, set the STS2 to STS0 bits in SYSCR1 to specify the time longer than the oscillation stabilization time (t_{rc}).

Therefore, when a transition is made from standby mode, watch mode, or subactive mode, to active (high-speed/medium-speed) mode, with an resonator connected to the system clock oscillator, careful evaluation must be carried out on the mounting circuit before deciding the oscillation stabilization wait time. For the wait time, secure the time required for the amplitude of the oscillation waveform to increase and the oscillation frequency to stabilize. In addition, since the oscillation start time differs according to mounting circuit constants, stray capacitance, and so forth, suitable constants should be determined in consultation with the resonator manufacturer.

4.5.4 Note on Subclock Stop State

To stop the subclock, a state transition should not be made except to mode in which the system clock operates. If the state transition is made to other mode, it may result in incorrect operation.

4.5.5 Note on the Oscillation Stabilization of Resonators

When a microcomputer operates, the internal power supply potential fluctuates slightly in synchronization with the system clock. Depending on the individual resonator characteristics, the oscillation waveform amplitude may not be sufficiently large immediately after the oscillation stabilization wait time, making the oscillation waveform susceptible to influence by fluctuations in the power supply potential. In this state, the oscillation waveform may be disrupted, leading to an unstable system clock and incorrect operation of the microcomputer.

If incorrect operation occurs, change the setting of the standby timer select bits 2 to 0 (STS2 to STS0) (bits 6 to 4 in the system control register 1 (SYSCR1)) to give a longer wait time.

For example, if incorrect operation occurs with a wait time setting of 512 states, check the operation with a wait time setting of 1,024 states or more.

If the same kind of incorrect operation occurs after a reset as after a state transition, hold the $\overline{\text{RES}}$ pin low for a longer period.

4.5.6 Note on Using Power-On Reset

The power-on reset circuit in this LSI adjusts the reset clear time by the capacitor capacitance, which is externally connected to the $\overline{\text{RES}}$ pin. The external capacitor capacitance should be adjusted to secure the oscillation stabilization time before reset clearing. For details, refer to section 19, Power-On Reset Circuit.

4.5.7 Note on Using On-Chip Emulator

When the on-chip emulator is used, system clock accuracy is necessary for flash memory programming/erasing. The frequency of the on-chip oscillator differs depending on the voltage and temperature conditions. Therefore, when using the on-chip emulator, the resonator must be connected to the OSC1 and OSC2 pins or an external clock must be supplied. In this case, the on-chip oscillator is used for user program execution, and the system clock is used for flash memory programming/erasing. This control is handled when the E7_2 pin is fixed to high level during a reset by the on-chip emulator.

Section 5 Power-Down Modes

This LSI has eight modes of operation after a reset. These include a normal active (high-speed) mode and seven power-down modes, in which power consumption is significantly reduced. The module standby function reduces power consumption by selectively halting on-chip module functions.

- Active (medium-speed) mode

  The CPU and all on-chip peripheral modules are operable on the system clock. The system clock frequency can be selected from $\phi_{OSC}/8$, $\phi_{OSC}/16$, $\phi_{OSC}/32$, and $\phi_{OSC}/64$.
- Subactive mode

  The CPU and all on-chip peripheral modules are operable on the subclock. The subclock frequency can be selected from ϕ_w, $\phi_w/2$, $\phi_w/4$, and $\phi_w/8$.
- Sleep (high-speed) mode

  The CPU halts. On-chip peripheral modules are operable on the system clock.
- Sleep (medium-speed) mode

  The CPU halts. On-chip peripheral modules are operable on the system clock. The system clock frequency can be selected from $\phi_{OSC}/8$, $\phi_{OSC}/16$, $\phi_{OSC}/32$, and $\phi_{OSC}/64$.
- Subsleep mode

  The CPU halts. The on-chip peripheral modules are operable on the subclock. The subclock frequency can be selected from ϕ_w, $\phi_w/2$, $\phi_w/4$, and $\phi_w/8$.
- Watch mode

  The CPU halts. The on-chip peripheral modules are operable on the subclock.
- Standby mode

  The CPU and all on-chip peripheral modules halt.
- Module standby function

  Independent of the above modes, power consumption can be reduced by halting on-chip peripheral modules that are not used in module units.

Note: In this manual, active (high-speed) mode and active (medium-speed) mode are collectively called active mode.

5.1 Register Descriptions

The registers related to power-down modes are as follows.

- System control register 1 (SYSCR1)
- System control register 2 (SYSCR2)
- Clock halt registers 1 and 2 (CKSTPR1 and CKSTPR2)

5.1.1 System Control Register 1 (SYSCR1)

SYSCR1 controls the power-down modes, as well as SYSCR2.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby Selects the mode to transit after the execution of the SLEEP instruction. 0: A transition is made to sleep mode or subsleep mode. 1: A transition is made to standby mode or watch mode. For details, see table 5.2.
6	STS2	0	R/W	Standby Timer Select 2 to 0 Designate the time the CPU and peripheral modules wait for stable clock operation after exiting from standby mode, subactive mode, or watch mode to active mode or sleep mode due to an interrupt. The designation should be made according to the operating frequency so that the waiting time is at least equal to the oscillation stabilization time. The relationship between the specified value and the number of wait states is shown in table 5.1. When an external clock is to be used, the minimum value (STS2 = 1, STS1 = 1, and STS0 = 1) is recommended. When the on-chip oscillator is to be used, the minimum value (STS2 = 1, STS1 = 1, and STS0 = 1) is recommended. If a setting other than the recommended value is made, operation may start before the end of the waiting time.
5	STS1	0	R/W	
4	STS0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
3	LSON	0	R/W	Selects the system clock (ϕ) or subclock (ϕ_{SUB}) as the CPU operating clock when watch mode is cleared. 0: The CPU operates on the system clock (ϕ) 1: The CPU operates on the subclock (ϕ_{SUB})
2	TMA3	0	R/W	Selects the mode to which the transition is made after the SLEEP instruction is executed with bits SSBY and LSON in SYSCR1 and bits DTON and MSON in SYSCR2. For details, see table 5.2.
1	MA1	1	R/W	Active Mode Clock Select 1 and 0
0	MA0	1	R/W	Select the operating clock frequency in active (medium-speed) mode and sleep (medium-speed) mode. The MA1 and MA0 bits should be written to in active (high-speed) mode or subactive mode. 00: ϕ_{OSC}/8 01: ϕ_{OSC}/16 10: ϕ_{OSC}/32 11: ϕ_{OSC}/64

Table 5.1 Operating Frequency and Waiting Time

Bit				Operating Frequency and Waiting Time						
STS2	STS1	STS0	Waiting States	10 MHz	8 MHz	6 MHz	5 MHz	4.194MHz	3 MHz	2 MHz
0	0	0	8,192 states	819.2	1,024.0*1	1,365.3*1	1,638.4	1953.3	2,730.7	4,096.0
		1	16,384 states	1,638.4	2,048.0	2,730.7	3,276.8	3906.5	5,461.3*1	8,192.0*1
	1	0	1,024 states	102.4	128.0	170.7	204.8	244.2	341.3	512.0
		1	2,048 states	204.8	256.0	341.3	409.6	488.3	682.7	1,024.0
1	0	0	4,096 states	409.6	512.0	682.7*1	819.2*1	976.6	1,365.3	2,048.0
		1	256 states	25.6	32.0	42.7*2	51.2*2	61.0	85.3*2	128.0*2
	1	0	512 states	51.2	64.0*2	85.3*2	102.4	122.1	170.7	256.0
		1	16 states	1.6	2.0	2.7	3.2	3.8	5.3	8.0

Notes: Time unit is μs.

: Recommended value when crystal resonator is used (Vcc = 2.7 V to 3.6 V)

: Recommended value when ceramic resonator is used (Vcc = 2.2 V to 3.6 V)

1. Reference value when crystal resonator is used
2. Reference value when ceramic resonator is used

5.1.2 System Control Register 2 (SYSCR2)

SYSCR2 controls the power-down modes, as well as SYSCR1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
4	NESEL	1	R/W	Noise Elimination Sampling Frequency Select The subclock pulse generator generates the watch clock signal (ϕ_w) and the system clock pulse generator generates the oscillator clock (ϕ_{osc}). This bit selects the sampling frequency of ϕ_{osc} when ϕ_w is sampled. When a system clock is used, clear this bit to 0.When the on-chip oscillator is selected, set this bit to 1. 0: Sampling rate is ϕ_{osc}/16. 1: Sampling rate is ϕ_{osc}/4.
3	DTON	0	R/W	Direct Transfer on Flag Selects the mode to which the transition is made after the SLEEP instruction is executed with bits SSBY, TMA3, and LSON in SYSCR1 and bit MSON in SYSCR2. For details, see table 5.2.
2	MSON	0	R/W	Medium Speed on Flag After standby, watch, or sleep mode is cleared, this bit selects active (high-speed) or active (medium-speed) mode. 0: Operation in active (high-speed) mode 1: Operation in active (medium-speed) mode
1	SA1	0	R/W	Subactive Mode Clock Select 1 and 0
0	SA0	0	R/W	Select the operating clock frequency in subactive and subsleep modes. The operating clock frequency changes to the set frequency after the SLEEP instruction is executed. 00: ϕ_w/8 01: ϕ_w/4 10: ϕ_w/2 11: ϕ_w

5.1.3 Clock Halt Registers 1 and 2 (CKSTPR1 and CKSTPR2)

CKSTPR1 and CKSTPR2 allow the on-chip peripheral modules to enter the standby state in module units.

- CKSTPR1

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
6	S3CKSTP	0	R/W	SCI3 Module Standby*[1] SCI3 enters standby mode when this bit is cleared to 0.
5	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
4	ADCKSTP	0	R/W	A/D Converter Module Standby A/D converter enters standby mode when this bit is cleared to 0.
3	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
2	TB1CKSTP	0	R/W	Timer B1 Module Standby Timer B1 enters standby mode when this bit is cleared to 0.
1	FROMCKSTP*[2]	1	R/W	Flash Memory Module Standby Flash memory enters standby mode when this bit is cleared to 0.
0	RTCCKSTP	1	R/W	RTC Module Standby RTC enters standby mode when this bit is cleared to 0.

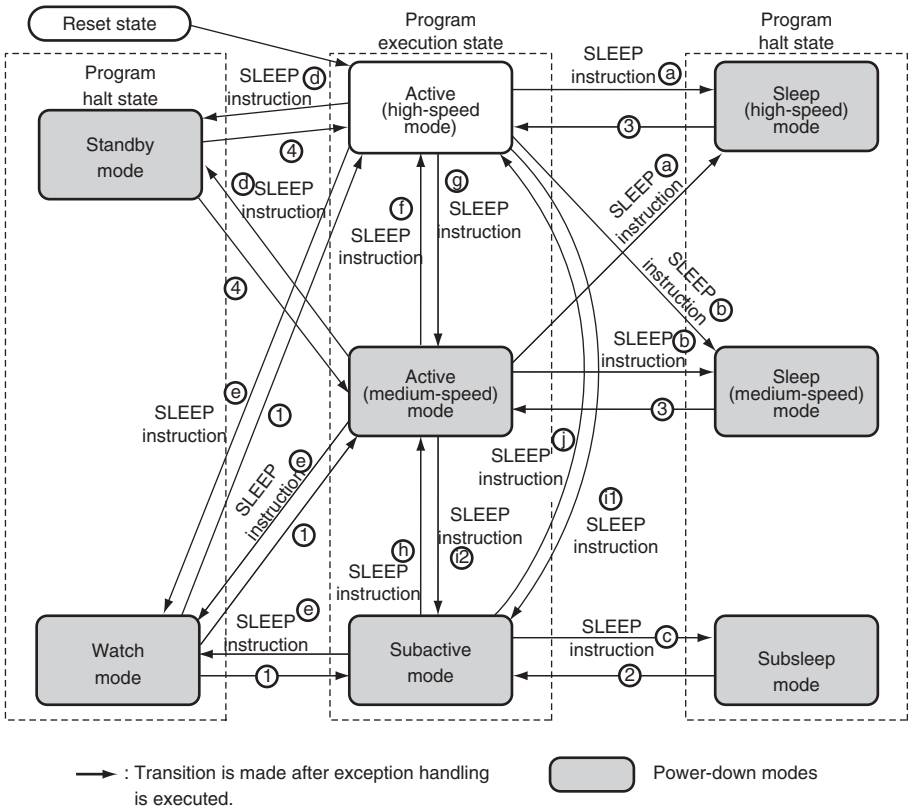
- CKSTPR2

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
6	TWCKSTP	0	R/W	Timer W Module Standby Timer W enters standby mode when this bit is cleared to 0.
5	IICCKSTP	0	R/W	IIC2 Module Standby The IIC2 enters standby mode when this bit is cleared to 0.
4	SSUCKSTP	0	R/W	SSU Module Standby The SSU enters standby mode when this bit is cleared to 0.
3	AECCKSTP	0	R/W	Asynchronous Event Counter Module Standby The asynchronous event counter enters standby mode when this bit is cleared to 0.
2	WDCKSTP	1	R/W*[3]	Watchdog Timer Module Standby The watchdog timer enters standby mode when this bit is cleared to 0.
1	COMPCKSTP	0	R/W	Comparator Module Standby The comparators enter standby mode when this bit is cleared to 0.
0	—	0	—	Reserved This bit is always read as 0 and cannot be modified.

Notes: 1. When the SCI3 module standby is set, all registers in the SCI3 enter the reset state.
2. When using the on-chip emulator, set this bit to 1.
3. This bit is valid when the WDON bit in TCSRW is 0. If this bit is cleared to 0 while the WDON bit is set to 1 (while the watchdog timer is operating), this bit is cleared to 0. However, the watchdog timer does not enter module standby mode and continues operating. When the WDON bit is cleared to 0 by software, this bit is valid and the watchdog timer enters module standby mode.

5.2 Mode Transitions and States of LSI

Figure 5.1 shows the possible transitions among these operating modes. A transition is made from the program execution state to the program halt state of the program by executing a SLEEP instruction. Interrupts allow for returning from the program halt state to the program execution state of the program. A direct transition between active mode and subactive mode, which are both program execution states, can be made without halting the program. The operating frequency can also be changed in the same modes by making a transition directly from active mode to active mode, and from subactive mode to subactive mode. $\overline{\text{RES}}$ input enables transitions from a mode to the reset state. Table 5.2 shows the transition conditions of each mode after the SLEEP instruction is executed and a mode to return by an interrupt. Table 5.3 shows the internal states of the LSI in each mode.



Mode Transition Conditions (1)

	LSON	MSON	SSBY	TMA3	DTON
ⓐ	0	0	0	x	0
ⓑ	0	1	0	x	0
ⓒ	1	x	0	1	0
ⓓ	0	x	1	0	0
ⓔ	x	x	1	1	0
ⓕ	0	0	0	x	1
ⓖ	0	1	0	x	1
ⓗ	0	1	1	1	1
ⓘ1	1	x	1	1	1
ⓘ2	1	1	1	1	1
ⓙ	0	0	1	1	1

Mode Transition Conditions (2)

	Interrupt Sources
①	NMI, IRQ0, IRQ1, IRQAEC, COMP, RTC, WDT, AEC, and timer B1
②	All interrupts except IIC2
③	All interrupts
④	NMI, IRQ0, IRQ1, IRQAEC, COMP, WDT, and AEC

x: Don't care

Note: A transition between different modes cannot be made to occur simply because an interrupt request is generated. Make sure that interrupt handling is accepted.

Figure 5.1 Mode Transition Diagram

Table 5.2 Transition Mode after SLEEP Instruction Execution and Interrupt Handling

	LSON	MSON	SSBY	TMA3	DTON	Transition Mode after SLEEP Instruction Execution	Transition Mode due to Interrupt
Active (high-speed) mode	0	0	0	x	0	Sleep (high-speed) mode	Active (high-speed) mode
	0	1	0	x	0	Sleep (medium-speed) mode	Active (medium-speed) mode
	0	0	1	0	0	Standby mode	Active (high-speed) mode
	0	1	1	0	0	Standby mode	Active (medium-speed) mode
	0	0	1	1	0	Watch mode	Active (high-speed) mode
	0	1	1	1	0	Watch mode	Active (medium-speed) mode
	1	x	1	1	0	Watch mode	Subactive mode
	0	0	0	x	1	Active (high-speed) mode (direct transition)	—
	0	1	0	x	1	Active (medium-speed) mode (direct transition)	—
	1	x	1	1	1	Subactive mode (direct transition)	—
Active (medium-speed) mode	0	0	0	x	0	Sleep (high-speed) mode	Active (high-speed) mode
	0	1	0	x	0	Sleep (medium-speed) mode	Active (medium-speed) mode
	0	0	1	0	0	Standby mode	Active (high-speed) mode
	0	1	1	0	0	Standby mode	Active (medium-speed) mode
	0	0	1	1	0	Watch mode	Active (high-speed) mode
	0	1	1	1	0	Watch mode	Active (medium-speed) mode
	1	1	1	1	0	Watch mode	Subactive mode
	0	0	0	x	1	Active (high-speed) mode (direct transition)	—
	0	1	0	x	1	Active (medium-speed) mode (direct transition)	—
	1	1	1	1	1	Subactive mode (direct transition)	—
Subactive mode	1	x	0	1	0	Subsleep mode	Subactive mode
	0	0	1	1	0	Watch mode	Active (high-speed) mode
	0	1	1	1	0	Watch mode	Active (medium-speed) mode
	1	x	1	1	0	Watch mode	Subactive mode
	0	0	1	1	1	Active (high-speed) mode (direct transition)	—
	0	1	1	1	1	Active (medium-speed) mode (direct transition)	—
	1	x	1	1	1	Subactive mode (direct transition)	—

[Legend] x: Don’t care.

Table 5.3 Internal State in Each Operating Mode

Function		Active Mode: High-speed	Active Mode: Medium-speed	Sleep Mode: High-speed	Sleep Mode: Medium-speed	Watch Mode	Subactive Mode	Subsleep Mode	Standby Mode
System clock oscillator		Functions	Functions	Functions	Functions	Halted	Halted	Halted	Halted
Subclock oscillator		Functions/ Halted	Functions/ Halted	Functions/ Halted	Functions/ Halted	Functions	Functions	Functions	Functions/ Halted
CPU	Instructions	Functions	Functions	Halted	Halted	Halted	Functions	Halted	Halted
	RAM			Retained	Retained	Retained		Retained	Retained
	Registers								
	I/O								Retained*[1]
External interrupts	NMI	Functions	Functions	Functions	Functions	Functions	Functions	Functions	Functions
	IRQ0								
	IRQ1								
	IRQAEC								
Peripheral modules	Timer B1	Functions	Functions	Functions	Functions	Functions/ Retained*[2]	Functions/ Retained*[2]	Functions/ Retained*[2]	Retained
	Timer W					Retained	Functions/ Retained*[3]	Functions/ Retained*[3]	Retained
	WDT					Functions/ Retained*[5]	Functions/ Retained*[5]	Functions/ Retained*[5]	Functions/ Retained*[4]
	RTC					Functions/ Retained*[6]	Functions/ Retained*[6]	Functions/ Retained*[6]	Retained
	Asynchro-nous event counter					Functions	Functions	Functions	Functions
	SCI3/ IrDA					Reset	Functions/ Retained*[7]	Functions/ Retained*[7]	Reset
	IIC2					Retained	Retained	Retained	Retained
	SSU					Retained	Functions/ Retained*[8]	Functions/ Retained*[8]	Retained
	A/D					Retained	Functions/ Retained*[9]	Functions/ Retained*[9]	Retained
	Comparator					Functions	Functions	Functions	Functions

Notes: 1. Register contents are retained. Output is the high-impedance state.

2. Functions if $\phi_W/256$ or $\phi_W/1024$ is selected as an internal clock. Halted and retained otherwise.

3. Functions if ϕ_w, $\phi_w/4$, or $\phi_w/16$ is selected as an internal clock. Halted and retained otherwise.
4. Functions if the on-chip oscillator is selected. Halted and retained otherwise.
5. Functions if the on-chip oscillator is selected or if $\phi_w/16$ or $\phi_w/256$ is selected as an internal clock. Halted and retained otherwise.
6. Functions if the 32.768-kHz RTC is selected as an internal clock. Halted and retained otherwise.
7. Functions if ϕ_w is selected as an internal clock. Halted and retained otherwise.
8. Functions if $\phi_{SUB}/2$ is selected as an internal clock. Halted and retained otherwise.
9. Functions if $\phi_w/2$ is selected as an internal clock. Halted and retained otherwise.

5.2.1 Sleep Mode

In sleep mode, CPU operation is halted but the system clock oscillator, subclock oscillator, and on-chip peripheral modules function. In sleep (medium-speed) mode, the on-chip peripheral modules function at the clock frequency set by the MA1 and MA0 bits in SYSCR1. CPU register contents are retained.

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

When the $\overline{\text{RES}}$ pin goes low, the CPU goes into the reset state and sleep mode is cleared. Since an interrupt request signal is synchronous with the system clock, the maximum time of $2/\phi$ (s) may be delayed from the point at which an interrupt request signal occurs until the interrupt exception handling is started.

5.2.2 Standby Mode

In standby mode, the system clock oscillator stops, and the CPU and on-chip peripheral modules stop functioning except for the WDT, asynchronous event counter, and comparators. However, as long as the rated voltage is supplied, the contents of CPU registers and some on-chip peripheral module registers are retained. On-chip RAM contents will be retained as long as the voltage set by the RAM data retention voltage is provided. The I/O ports go to the high-impedance state.

Standby mode is cleared by an interrupt. When an interrupt is requested, the system clock pulse generator starts. After the time set in bits STS2 to STS0 in SYSCR1 has elapsed, standby mode is cleared and interrupt exception handling starts. After standby mode is cleared, a transition is made to active (high-speed) or active (medium-speed) mode according to the MSON bit in SYSCR2. Standby mode is not cleared if the I bit in CCR is set to 1 or the requested interrupt is disabled by the interrupt enable bit.

When a reset source is generated in standby mode, the system clock oscillator starts. If a reset is generated by the $\overline{\text{RES}}$ pin, it must be kept low until the system clock oscillator output stabilizes and the t_{REL} period has elapsed. The CPU starts reset exception handling when the $\overline{\text{RES}}$ pin is driven high.

5.2.3 Watch Mode

In watch mode, the system clock oscillator and CPU operation stop, and on-chip peripheral modules stop functioning except for the WDT, RTC, timer B1, asynchronous event counter, and comparators. However, as long as the rated voltage is supplied, the contents of CPU registers, some on-chip peripheral module registers, and on-chip RAM are retained. The I/O ports retain their state before the transition.

Watch mode is cleared by an interrupt. When an interrupt is requested, watch mode is cleared and interrupt exception handling starts. When watch mode is cleared by an interrupt, a transition is made to active (high-speed) mode, active (medium-speed) mode, or subactive mode depending on the settings of the LSON bit in SYSCR1 and the MSON bit in SYSCR2. When the transition is made to active mode, after the time set in bits STS2 to STS0 in SYSCR1 has elapsed, interrupt exception handling starts. Watch mode is not cleared if the I bit in CCR is set to 1 or the requested interrupt is disabled by the interrupt enable register.

When a reset source is generated in watch mode, the system clock oscillator starts. If a reset is generated by the $\overline{\text{RES}}$ pin, it must be kept low until the system clock oscillator output stabilizes and the t_{REL} period has elapsed. The CPU starts reset exception handling when the $\overline{\text{RES}}$ pin is driven high.

5.2.4 Subsleep Mode

In subsleep mode, the CPU operation stops but on-chip peripheral modules function except for the IIC2. As long as a 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.

Subsleep mode is cleared by an interrupt. When an interrupt is requested, subsleep mode is cleared and interrupt exception handling starts. After subsleep mode is cleared, a transition is made to subactive mode. Subsleep mode is not cleared if the I bit in CCR is set to 1 or the requested interrupt is disabled by the interrupt enable register.

When a reset source is generated in subsleep mode, the system clock oscillator starts. If a reset is generated by the $\overline{\text{RES}}$ pin, it must be kept low until the system clock oscillator output stabilizes and the t_{REL} period has elapsed. The CPU starts reset exception handling when the $\overline{\text{RES}}$ pin is driven high.

5.2.5 Subactive Mode

In subactive mode, the system clock oscillator stops but on-chip peripheral modules function except for the IIC2. As long as a required voltage is applied, the contents of some registers of the on-chip peripheral modules are retained.

Subactive mode is cleared by the SLEEP instruction. When subactive mode is cleared, a transition to subsleep mode, active mode, or watch mode is made, depending on the combination of bits SSBY, LSON, and TMA3 in SYSCR1 and bits MSON and DTON in SYSCR2.

When a reset source is generated in subactive mode, the system clock oscillator starts. If a reset is generated by the $\overline{\text{RES}}$ pin, it must be kept low until the system clock oscillator output stabilizes and the t_{REL} period has elapsed. The CPU starts reset exception handling when the $\overline{\text{RES}}$ pin is driven high.

The operating frequency of subactive mode is selected from ϕ_w (watch clock), $\phi_w/2$, $\phi_w/4$, and $\phi_w/8$ by the SA1 and SA0 bits in SYSCR2. After the SLEEP instruction is executed, the operating frequency changes to the frequency which is set before the execution.

5.2.6 Active (Medium-Speed) Mode

In active (medium-speed) mode, the clock set by the MA1 and MA0 bits in SYSCR1 is used as the system clock, and the CPU and on-chip peripheral modules function.

Active (medium-speed) mode is cleared by the SLEEP instruction. When active (medium-speed) mode is cleared, a transition to standby mode is made depending on the combination of bits SSBY, LSON, and TMA3 in SYSCR1, a transition to watch mode is made depending on the combination of bits SSBY and TMA3 in SYSCR1, or a transition to sleep mode is made depending on the combination of bits SSBY and LSON in SYSCR1. Moreover, a transition to active (high-speed) mode or subactive mode is made by a direct transition. When the $\overline{\text{RES}}$ pin goes low, the CPU goes into the reset state and active (medium-sleep) mode is cleared.

5.3 Direct Transition

The CPU can execute programs in two modes: active and subactive modes. A direct transition is made between these two modes without stopping program execution. A direct transition can also be made when the operating clock is changed in active and subactive modes. The transition is made via the sleep or watch mode, by setting the DTON bit in SYSCR2 to 1 to execute a SLEEP instruction. After the mode transition, direct transition interrupt exception handling starts.

Note that if a direct transition is attempted while the I bit in CCR is 1, the transition is made to the sleep or watch mode, though not returning from the mode.

5.3.1 Direct Transition from Active (High-Speed) Mode to Active (Medium-Speed) Mode

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

Example: When ϕosc/8 is selected as the CPU operating clock after the transition

Direct transition time = (2 + 1) × 1tosc + 14 × 8tosc = 115tosc

For the legend of symbols used above, refer to section 21, Electrical Characteristics.

5.3.2 Direct Transition from Active (High-Speed) Mode to Subactive Mode

When a SLEEP instruction is executed in active (high-speed) mode while the SSBY, TMA3, and LSON bits in SYSCR1 are set to 1 and the DTON bit in SYSCR2 is set to 1, a transition is made to subactive mode via watch 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)} × (tcyc before transition) + (Number of interrupt exception handling execution states) × (tsubcyc after transition)…..…(2)

Example: When ϕw/8 is selected as the subactive operating clock after the transition

Direct transition time = (2 + 1) × 1tosc + 14 × 8tw = 3tosc + 112tw

For the legend of symbols used above, refer to section 21, Electrical Characteristics.

5.3.3 Direct Transition from Active (Medium-Speed) Mode to Active (High-Speed) Mode

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

The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (3).

Direct transition time = {(Number of SLEEP instruction execution states) + (Number of internal processing states)} × (tcyc before transition) + (Number of interrupt exception handling execution states) × (tcyc after transition)……..(3)

Example: When ϕosc/8 is selected as the CPU operating clock before the transition

Direct transition time = (2 + 1) × 8tosc + 14 × 1tosc = 38tosc

For the legend of symbols used above, refer to section 21, Electrical Characteristics.

5.3.4 Direct Transition from Active (Medium-Speed) Mode to Subactive Mode

When a SLEEP instruction is executed in active (medium-speed) mode while the SSBY, LSON, and TMA3 bits in SYSCR1 are set to 1 and the DTON bit in SYSCR2 is set to 1, a transition is made to subactive mode via watch 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 (4).

Direct transition time = {(Number of SLEEP instruction execution states) + (Number of internal processing states)} × (tcyc before transition) + (Number of interrupt exception handling execution states) × (tsubcyc after transition)……(4)

Example: When ϕosc/8 and ϕw/8 are selected as the CPU operating clock before and after the transition, respectively

Direct transition time = (2 + 1) × 8tosc + 14 × 8tw = 24tosc + 112tw

For the legend of symbols used above, refer to section 21, Electrical Characteristics.

5.3.5 Direct Transition from Subactive Mode to Active (High-Speed) Mode

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

The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (5).

Direct transition time = {(Number of SLEEP instruction execution states) + (Number of internal processing states)} × (tsubcyc before transition) + (Wait time set in bits STS2 to STS0) + (Number of interrupt exception handling execution states) × (tcyc after transition)……..(5)

Example: When ϕw/8 is selected as the CPU operating clock after the transition and wait time = 8192 states

Direct transition time = (2 + 1) × 8tw + (8192 + 14) × 1tosc = 24tw + 8206tosc

For the legend of symbols used above, refer to section 21, Electrical Characteristics.

5.3.6 Direct Transition from Subactive Mode to Active (Medium-Speed) Mode

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

The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (6).

Direct transition time = {(Number of SLEEP instruction execution states) + (Number of internal processing states)} × (tsubcyc before transition) + (Wait time set in bits STS2 to STS0) + (Number of interrupt exception handling execution states) × (tcyc after transition)……..……..(6)

Example: When ϕw/8 and ϕosc/8 are selected as the CPU operating clock before and after the transition, respectively, and wait time = 8192 states

Direct transition time = (2 + 1) × 8tw + 8192 × 1tosc + 14 × 8tosc = 24tw + 8304tosc

For the legend of symbols used above, refer to section 21, Electrical Characteristics.

5.3.7 Notes on External Input Signal Changes before/after Direct Transition

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

  Since the mode transition is performed via watch mode, see section 5.6.2, Notes on External Input Signal Changes before/after Standby Mode.
- Direct transition from active (medium-speed) mode to subactive mode

  Since the mode transition is performed via watch mode, see section 5.6.2, Notes on External Input Signal Changes before/after Standby Mode.
- Direct transition from subactive mode to active (high-speed) mode

  Since the mode transition is performed via watch mode, see section 5.6.2, Notes on External Input Signal Changes before/after Standby Mode.
- Direct transition from subactive mode to active (medium-speed) mode

  Since the mode transition is performed via watch mode, see section 5.6.2, Notes on External Input Signal Changes before/after Standby Mode.

5.4 Module Standby Function

The module-standby function can be set to any peripheral module. In module standby mode, the clock supply to modules stops to enter the power-down mode. Module standby mode enables each on-chip peripheral module to enter the standby state by clearing a bit that corresponds to each module in CKSTPR1 and CKSTPR2 to 0 and cancels the mode by setting the bit to 1. (See section 5.1.3, Clock Halt Registers 1 and 2 (CKSTPR1 and CKSTPR2).)

5.5 On-Chip Oscillator and Operation Mode

The on-chip oscillator can be used as the clock source for the watchdog timer (WDT), subclock generation circuit ($\phi_W = R_{OSC}/32$), and system clock generation circuit ($\phi_{OSC} = R_{OSC}$).

When the on-chip oscillator is used as the clock source for the watchdog timer (WDT), it operates in any modes, such as active, sleep, subactive, subsleep, watch, and standby modes.

When the on-chip oscillator is used as the clock source for the subclock generation circuit, it stops in standby mode and operates in other modes.

When the on-chip oscillator is used only as the clock source for the system clock generation circuit, it operates in active and sleep modes but halts the operation in subactive, subsleep, watch, and standby modes.

When the on-chip oscillator is not used as the clock source for the watchdog timer (WDT), subclock generation circuit, or system clock generation circuit, it halts the operation.

The on-chip oscillator operates at a reset and after a reset, because the watchdog timer (WDT) selects the on-chip oscillator as the clock source for the initial value.

5.6 Usage Notes

5.6.1 Standby Mode Transition and Pin States

When a SLEEP instruction is executed in active (high-speed) mode or active (medium-speed) mode while the SSBY and TMA3 bits in SYSCR1 and the LSON bit in SYSCR1 are cleared to 0, a transition is made to standby mode. At the same time, pins go to the high-impedance state (except pins for which the pull-up MOS is designated as on). Figure 5.2 shows the timing in this case.

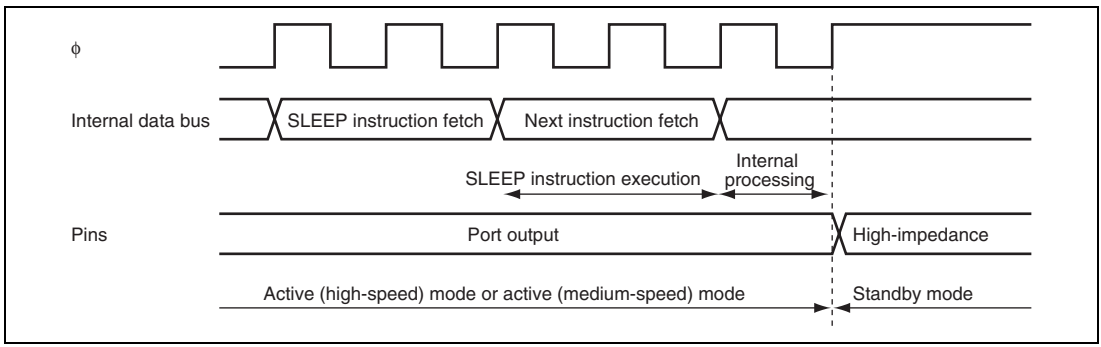


Figure 5.2 Standby Mode Transition and Pin States

5.6.2 Notes on External Input Signal Changes before/after Standby Mode

(1) When External Input Signal Changes before/after Standby Mode or Watch Mode

When an external input signal such as $\overline{\text{NMI}}$, $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, or IRQAEC is input, both the high- and low-level widths of the signal must be at least two cycles of system clock ϕ or subclock ϕ_{SUB} (referred to together in this section as the internal clock). As the internal clock stops in standby mode and watch mode, the width of external input signals requires careful attention when a transition is made via these operating modes. Ensure that external input signals conform to the conditions stated in (3), Recommended Timing of External Input Signals, below.

(2) When External Input Signals cannot be Captured because Internal Clock Stops

The case of falling edge capture is shown in figure 5.3.

As shown in the case marked "Capture not possible," when an external input signal falls immediately after a transition to active mode or subactive mode, after oscillation is started by an interrupt via a different signal, the external input signal cannot be captured if the high-level width at that point is less than 2 t_{cyc} or 2 t_{subcyc}.

(3) Recommended Timing of External Input Signals

To ensure dependable capture of an external input signal, high- and low-level signal widths of at least 2 t_{cyc} or 2 t_{subcyc} are necessary before a transition is made to standby mode or watch mode, as shown in "Capture possible: case 1."

External input signal capture is also possible with the timing shown in "Capture possible: case 2" and "Capture possible: case 3," in which a 2 t_{cyc} or 2 t_{subcyc} level width is secured.

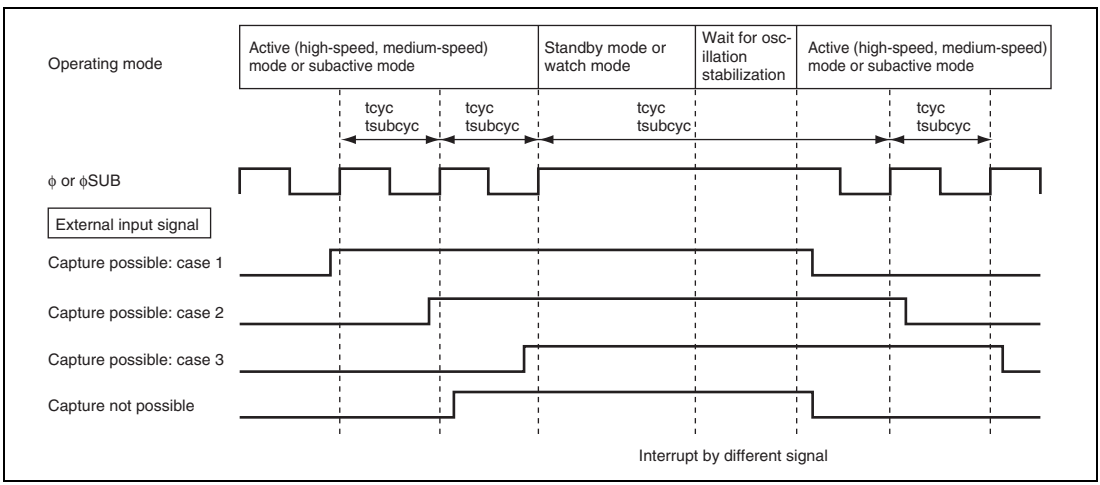


Figure 5.3 External Input Signal Capture when Signal Changes before/after Standby Mode or Watch Mode

(4) Input Pins to which these Notes Apply

$\overline{\text{NMI}}$, $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, IRQAEC, and $\overline{\text{ADTRG}}$

Section 6 ROM

The features of the 16-Kbyte flash memory built into the flash memory (F-ZTAT) version are summarized below.

- Programming/erasing methods

  The flash memory is programmed 128 bytes at a time. Erasure is performed in single-block units. The flash memory is configured as follows: 1 Kbyte × 4 blocks and 12 Kbytes × 1 block. To erase the entire flash memory, each block must be erased in turn.
- On-board programming

  On-board programming/erasure can be done in boot mode, in which the boot program built into this LSI is started to erase or program of the entire flash memory. In normal user program mode, individual blocks can be erased or programmed.
- Automatic bit rate adjustment

  For data transfer in boot mode, this LSI's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Programming/erasing protection

  Sets software protection against flash memory programming/erasure.
- Power-down mode

  Operation of the power supply circuit can be partly halted in subactive mode. As a result, flash memory can be read with low power consumption.
- Module standby mode

  Use of module standby mode enables this module to be placed in standby mode independently when not used. (For details, refer to section 5.4, Module Standby Function.) However, when using the on-chip emulator debugger, set the FROMCKSTP bit in clock halt register 1 to 1.

Note: The system clock oscillator must be used when programming or erasing the flash memory.

6.1 Block Configuration

Figure 6.1 shows the block configuration of flash memory. The thick lines indicate erasing a block, the narrow lines indicate programming units, and the values are addresses. The 16-Kbyte flash memory is divided into 1 Kbyte × 4 blocks and 12 Kbytes × 1 block. Erasure is performed in these units. Programming is performed in 128-byte units starting from an address with lower eight bits H'00 or H'80.

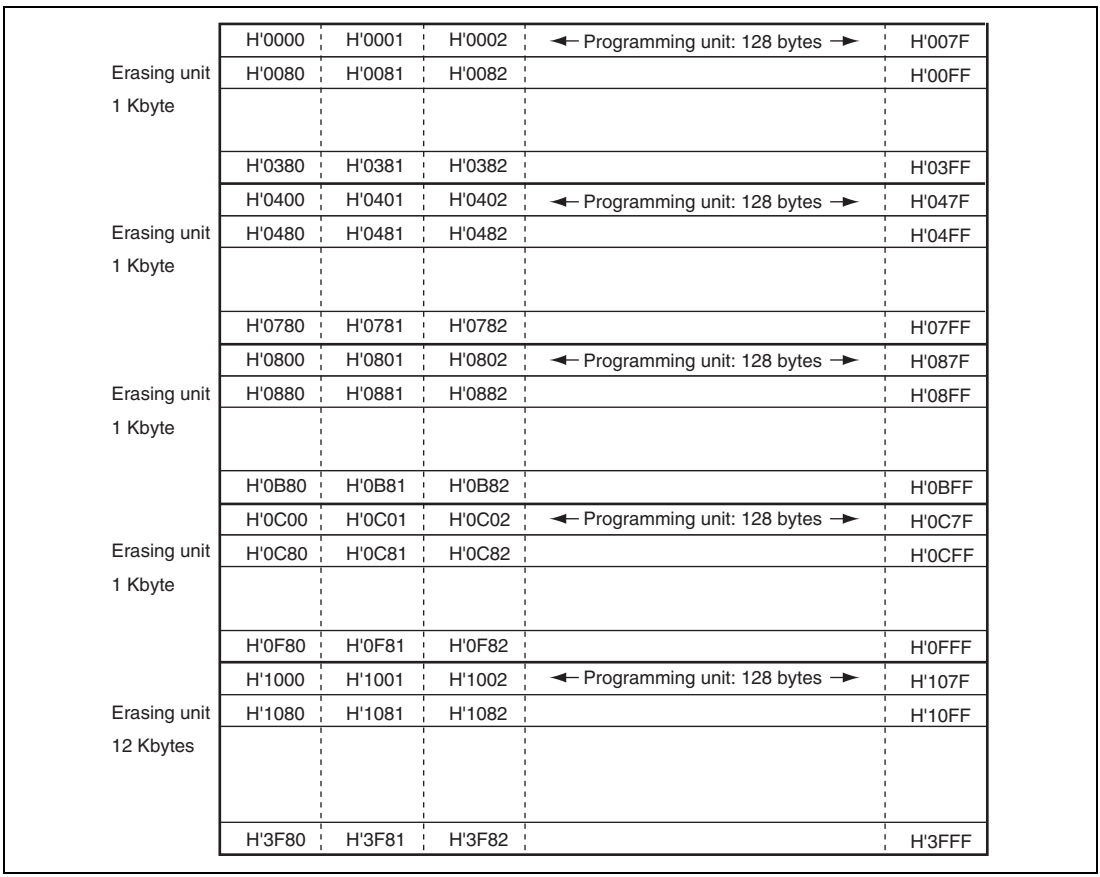


Figure 6.1 Flash Memory Block Configuration

6.2 Register Descriptions

The flash memory has the following registers.

- Flash memory control register 1 (FLMCR1)
- Flash memory control register 2 (FLMCR2)
- Erase block register 1 (EBR1)
- Flash memory power control register (FLPWCR)
- Flash memory enable register (FENR)

6.2.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is a register that makes the flash memory enter the programming mode, programming-verifying mode, erasing mode, or erasing-verifying mode. For details on register setting, refer to section 6.4, Flash Memory Programming/Erasure.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.
6	SWE	0	R/W	Software Write Enable When this bit is set to 1, flash memory programming/erasure is enabled. When this bit is cleared to 0, other FLMCR1 register bits and all EBR1 bits cannot be set.
5	ESU	0	R/W	Erase Setup When this bit is set to 1, the flash memory enters to the erasure setup state. When it is cleared to 0, the erasure setup state is released. Set this bit to 1 before setting the E bit in FLMCR1 to 1.
4	PSU	0	R/W	Program Setup When this bit is set to 1, the flash memory enters to the programming setup state. When it is cleared to 0, the programming setup state is released. Set this bit to 1 before setting the P bit in FLMCR1 to 1.

Bit	Bit Name	Initial Value	R/W	Description
3	EV	0	R/W	Erase-Verify When this bit is set to 1, the flash memory enters to erasing-verifying mode. When it is cleared to 0, erasing-verifying mode is released.
2	PV	0	R/W	Program-Verify When this bit is set to 1, the flash memory enters the programming-verifying mode. When it is cleared to 0, programming-verifying mode is released.
1	E	0	R/W	Erase When this bit is set to 1 while SWE=1 and ESU=1, the flash memory enters the erasing mode. When it is cleared to 0, the erasing mode is released.
0	P	0	R/W	Program When this bit is set to 1 while SWE=1 and PSU=1, the flash memory enters the programming mode. When it is cleared to 0, the programming mode is released.

6.2.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is a register that indicates the state of flash memory programming/erasure. FLMCR2 is a read-only register, and should not be written to.

Bit	Bit Name	Initial Value	R/W	Description
7	FLER	0	R	Flash Memory Error Indicates that an error has occurred during programming or erasing flash memory. When this bit is set to 1, flash memory enters the error-protection state. See section 6.5.3, Error Protection, for details.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

6.2.3 Erase Block Register 1 (EBR1)

EBR1 specifies the erase block of flash memory. EBR1 is initialized to H'00 when the SWE bit in FLMCR1 is 0. Do not set more than one bit at a time, as this will cause all the bits in EBR1 to be automatically cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	—	Reserved Although these bits are readable/writable, only 0 should be written to.
4	EB4	0	R/W	When this bit is set to 1, a12-Kbyte area of H'1000 to H'3FFF will be erased.
3	EB3	0	R/W	When this bit is set to 1, a 1-Kbyte area of H'0C00 to H'0FFF will be erased.
2	EB2	0	R/W	When this bit is set to 1, a 1-Kbyte area of H'0800 to H'0BFF will be erased.
1	EB1	0	R/W	When this bit is set to 1, a 1-Kbyte area of H'0400 to H'07FF will be erased.
0	EB0	0	R/W	When this bit is set to 1, a 1-Kbyte area of H'0000 to H'03FF will be erased.

6.2.4 Flash Memory Power Control Register (FLPWCR)

FLPWCR enables or disables a transition to the flash memory power-down mode when this LSI enters the subactive mode. There are two modes: mode in which operation of the power supply circuit of flash memory is partly halted in power-down mode and flash memory can be read, and mode in which even if a transition is made to subactive mode, operation of the power supply circuit of flash memory is retained and flash memory can be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PDWND	0	R/W	Power-Down Disable When this bit is 0 and a transition is made to subactive mode, the flash memory enters the power-down mode. When this bit is 1, the flash memory remains in the normal mode even after a transition is made to subactive mode.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

6.2.5 Flash Memory Enable Register (FENR)

Bit 7 (FLSHE) in FENR enables or disables the CPU access to the flash memory control registers, FLMCR1, FLMCR2, EBR1, and FLPWCR.

Bit	Bit Name	Initial Value	R/W	Description
7	FLSHE	0	R/W	Flash Memory Control Register Enable Flash memory control registers can be accessed when this bit is set to 1. Flash memory control registers cannot be accessed when this bit is set to 0.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

6.3 On-Board Programming Modes

The available mode for programming/erasing of the flash memory is boot mode, which enables on-board programming/erasure. On-board programming/erasure can also be performed in user program mode. When this LSI starts after releasing the reset state, it enters a mode depending on the signal levels on the TEST, $\overline{\text{NMI}}$, and E7_0 pins, as shown in table 6.1. The input level of each pin must be stable four states before the reset ends.

When entering the boot mode, the boot program built into this LSI is initiated. The boot program transfers the programming control program from the externally-connected host to on-chip RAM via SCI3. After erasing the entire flash memory, the programming control program is executed. This can be used for initializing flash memory mounted on the user board or for a forcible recovery if flash memory cannot be programmed or erased in user program mode. In user program mode, individual blocks can be erased and programmed by branching to the user programming/erasing control program prepared by the user.

Table 6.1 Setting Programming Modes

TEST	$\overline{\text{NMI}}$	E7_0	LSI State after Reset Released
0	1	x	User Mode
0	0	1	Boot Mode

[Legend]
x: Don’t care.

6.3.1 Boot Mode

Table 6.2 shows the boot mode operations between a reset released and a branch to the programming control program.

This LSI includes a system clock oscillator which is operated by a resonator or an external clock and on-chip oscillator.

In Boot Mode, since the system clock oscillator is selected, connect a resonator to OSC1 and OSC2, or an external clock signal to OSC1.

1. When the boot mode is used, the flash memory programming control program must be prepared in the host beforehand. Prepare a programming control program in accordance with the description in section 6.4, Flash Memory Programming/Erasure.
2. SCI3 is set to the asynchronous mode, and the transfer format as follows: 8-bit data, 1 stop bit, and no parity. The inversion function of the TXD and RXD pins by SPCR is set to “Not to be inverted,” so do not put inverters between the host and this LSI.
3. When the boot program is initiated, this LSI measures the low-level period of serial communication data (H'00) continuously transmitted in asynchronous mode from the host. This LSI then calculates the bit rate of the transfer from the host, and adjusts the SCI3 bit rate to match that of the host. The reset signal should be negated while the RXD pin is driven high. The RXD and TXD pins should be pulled up on the board if necessary. After the reset signal is negated, it takes approximately 100 states before this LSI is ready to measure the low-level period.
4. After matching the bit rates, SCI3 transmits one byte of H'00 to the host to indicate the completion of bit rate adjustment. The host should confirm that it has received this adjustment end code (H'00) normally and then transmit one byte of H'55 to this LSI. If reception could not be performed normally, initiate the boot mode again by a reset. Depending on the host's transfer bit rate and system clock frequency of this LSI, there will be a discrepancy between the bit rates of the host and this LSI. To operate the SCI properly, set the host's transfer bit rate and system clock frequency of this LSI within the ranges listed in table 6.3.
5. In boot mode, a part of the on-chip RAM area is used by the boot program. The programming control program transmitted from the host can be stored in the area from H'FB80 to H'FF7F. The boot program area cannot be used until control of the execution is switched from the boot program to the programming control program.

6. Before branching to the programming control program, this LSI terminates transfer operations by SCI3 (by clearing the RE and TE bits in SCR3 to 0), however the adjusted bit rate value remains set in BRR. Therefore, the programming control program can still use it for transferring program data or verify data to the host. The TXD pin is driven high (PCR32 = 1, P32 = 1). The contents of the CPU general registers are undefined immediately after branching to the programming control program. These registers must be initialized at the beginning of the programming control program because the stack pointer (SP), in particular, is used implicitly in subroutine calls, etc.
7. The boot mode can be released by a reset. Hold the reset signal low at least 20 states and then set the $\overline{\text{NMI}}$ pin before negating the reset signal. The boot mode is also released when a WDT overflow occurs.
8. Do not change the TEST pin and $\overline{\text{NMI}}$ pin input levels in boot mode.

Table 6.2 Boot Mode Operation

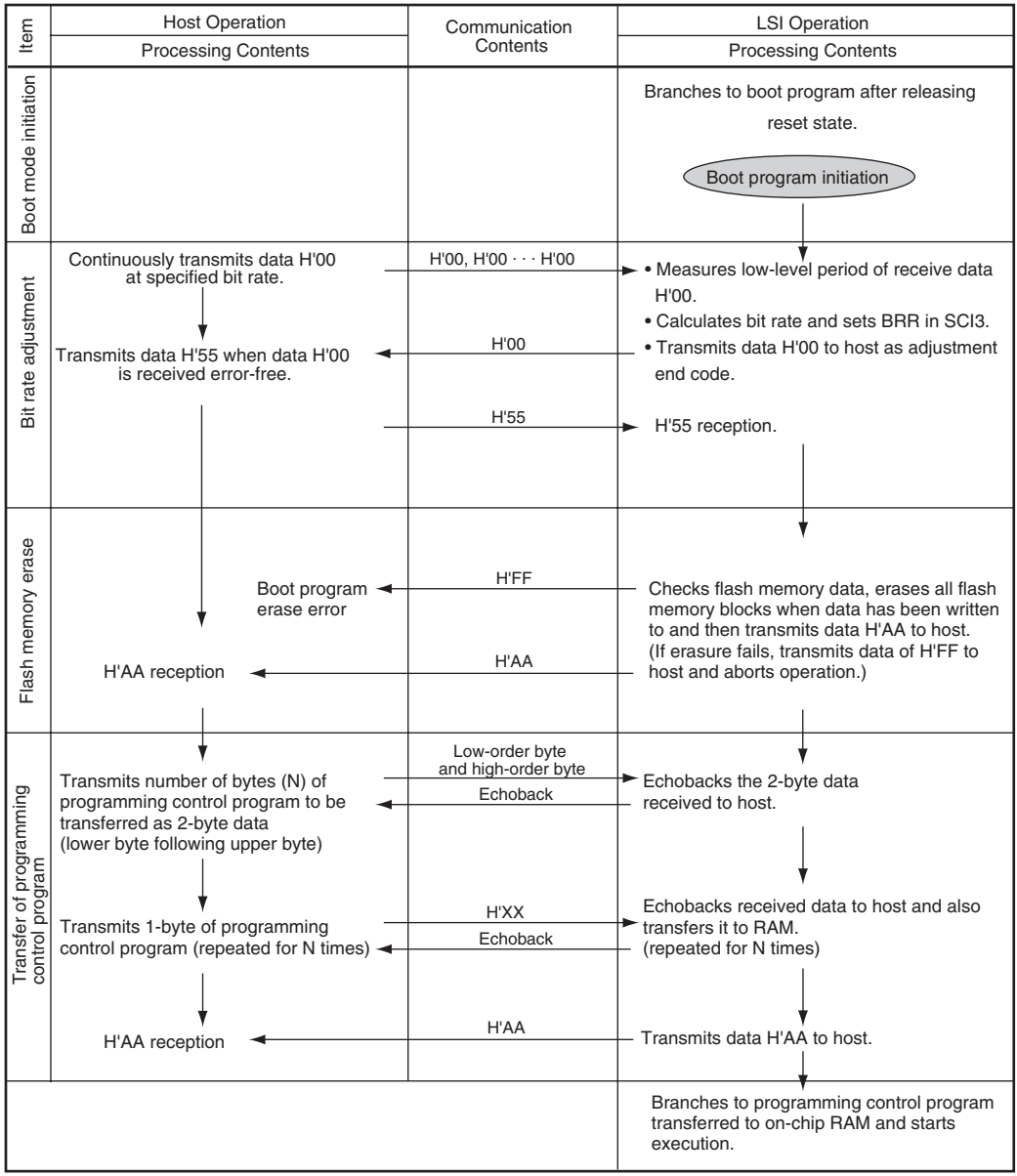


Table 6.3 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible

Host Bit Rate	System Clock Frequency Range of LSI
9,600 bps	8 to 10 MHz
4,800 bps	4 to 10 MHz
2,400 bps	2 to 10 MHz

6.3.2 Programming/Erasure in User Program Mode

On-board programming/erasing of an individual flash memory block can also be performed in user program mode by branching to a user programming/erasing control program. The user must prepare the settings for branching to the user programming/erasing control program and means to transfer programming data for on-board programming. The flash memory must contain the user programming/erasing control program or a program that transfer the user programming/erasing control program from external memory. Since the flash memory cannot be read during programming/erasure, transfer the user programming/erasing control program to on-chip RAM, as in boot mode. Figure 6.2 shows a sample procedure for programming/erasure in user program mode. Prepare a user programming/erasing control program in accordance with the description in section 6.4, Flash Memory Programming/Erasure.

The system clock oscillator must be used when programming or erasing the flash memory.

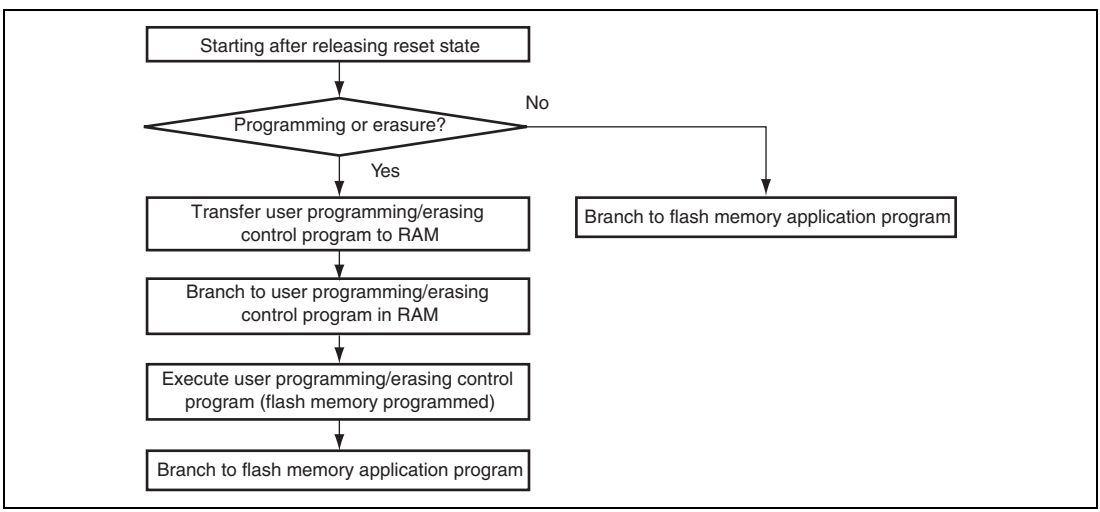


Figure 6.2 Programming/Erasing Flowchart Example in User Program Mode

6.4 Flash Memory Programming/Erasure

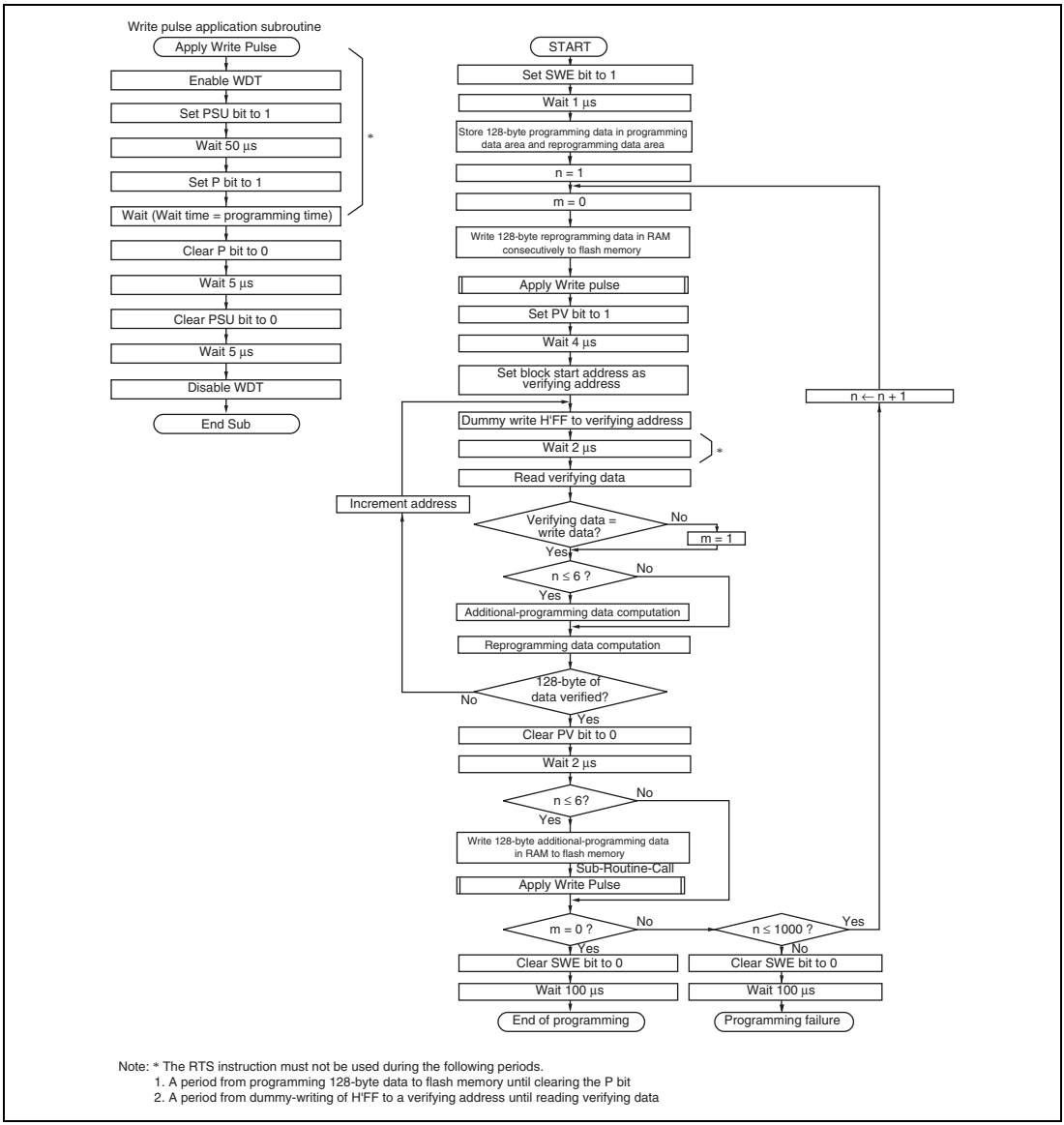
A software method using the CPU is employed to program and erase flash memory in the on-board programming modes. Depending on the FLMCR1 setting, the flash memory operates in one of the following four modes: Programming mode, programming-verifying mode, erasing mode, and erasing-verifying mode. The programming control program in boot mode and the user programming/erasing control program in user program mode use these operating modes in combination to perform programming/erasure. Flash memory programming and erasing should be performed in accordance with the descriptions in section 6.4.1, Programming/Programming-Verifying and section 6.4.2, Erasing/Erasing-Verifying, respectively.

6.4.1 Programming/Programming-Verifying

When writing data or programs to the flash memory, the programming/programming-verifying flowchart shown in figure 6.3 should be followed. Performing programming operations according to this flowchart will enable data or programs to be written to the flash memory without subjecting the chip to voltage stress or sacrificing program data reliability.

1. Programming must be performed on an erased area. Do not reprogram an address to which data has already been programmed.
2. Programming should be carried out 128 bytes at a time. A 128-byte data transfer must be performed even if programming fewer than 128 bytes. In this case, the remaining area must be filled with H'FF.
3. Prepare the following data storage areas in RAM: A 128-byte programming data area, a 128-byte reprogramming data area, and a 128-byte additional-programming data area. Perform reprogramming data computation according to table 6.4, and additional programming data computation according to table 6.5.
4. Consecutively transfer 128 bytes of data in bytes from the reprogramming data area or additional-programming data area to the flash memory. The programming address and 128-byte data are latched in the flash memory. The lower eight bits of the start address in the flash memory destination area must be H'00 or H'80.
5. The time during which the P bit is set to 1 is the programming time. Table 6.6 shows the allowable programming times.
6. The watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. An overflow cycle of approximately 6.6 ms is allowed.
7. For a dummy write to a verifying address, write 1-byte of data H'FF to an address whose lower two bits are B'00. Verifying data can be read in words or in longwords from the address to which a dummy write was performed.

8. The maximum number of repetitions of the programming/programming-verifying sequence of the same bit is 1,000.



Note: * The RTS instruction must not be used during the following periods.
1. A period from programming 128-byte data to flash memory until clearing the P bit
2. A period from dummy-writing of H'FF to a verifying address until reading verifying data

Figure 6.3 Program/Program-Verify Flowchart

Table 6.4 Reprogramming Data Computation Table

Programming Data	Verifying Data	Reprogramming Data	Comments
0	0	1	Programming completed
0	1	0	Needs to be programmed
1	0	1	—
1	1	1	Remains in erased state

Table 6.5 Additional-Program Data Computation Table

Reprogram Data	Verify Data	Additional-Program Data	Comments
0	0	0	Needs to be programmed additionally
0	1	1	No additional programming
1	0	1	No additional programming
1	1	1	No additional programming

Table 6.6 Programming Time

n (Programming Count)	Programming Time	Additional Programming Time	Comments
1 to 6 times	30 μs	10 μs	
7 to 1,000 times	200 μs	—	

Note: Time shown in μs.

6.4.2 Erasing/Erasing-Verifying

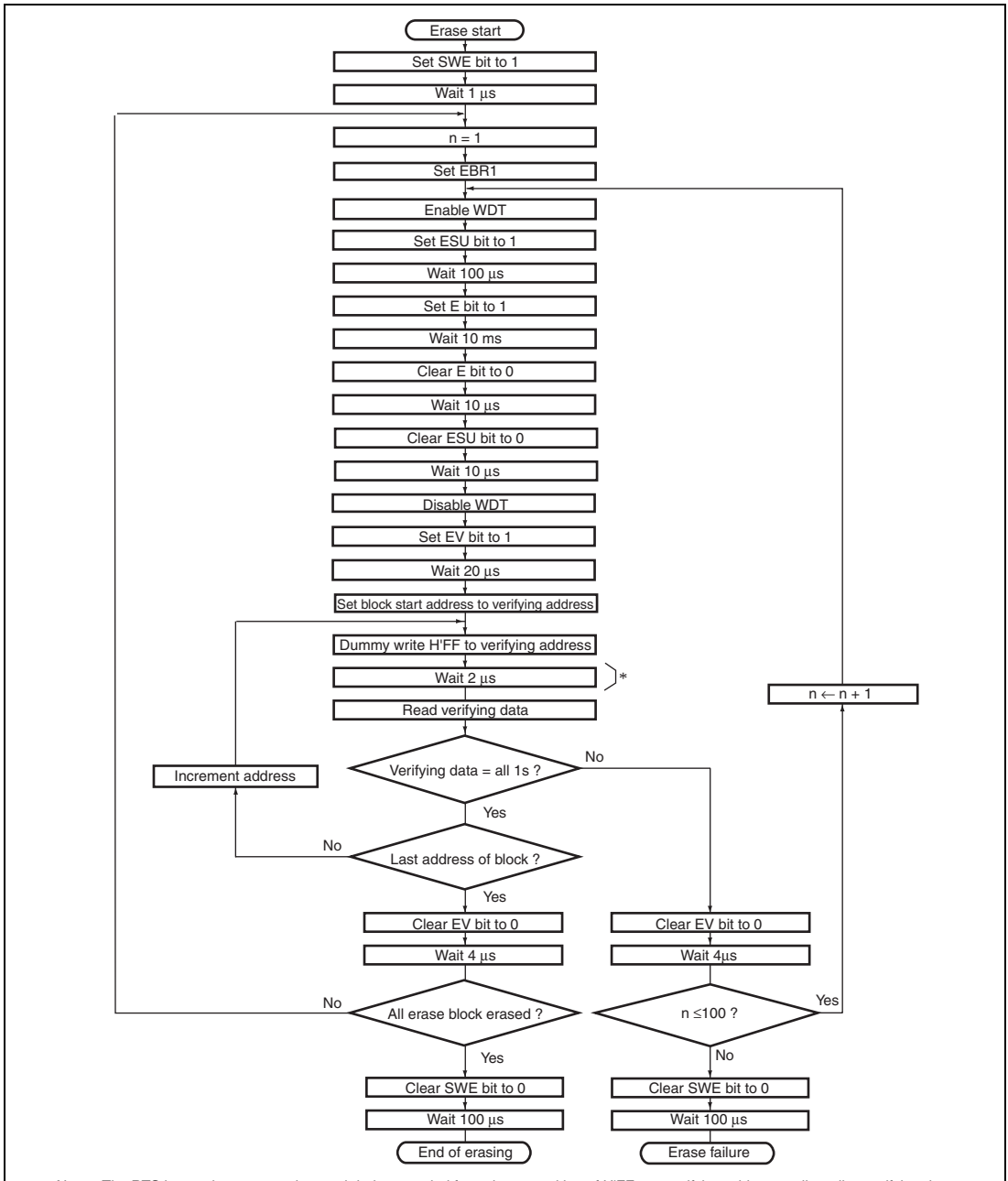
When erasing flash memory, the erasing/erasing-verifying flowchart shown in figure 6.4 should be followed.

1. Prewriting (setting erase block data to all 0s) is not necessary.
2. Erasure is performed in block units. Select a single block to be erased through erase block register 1 (EBR1). To erase multiple blocks, each block must be erased in turn.
3. The time during which the E bit is set to 1 is the flash memory erasing time.
4. The watchdog timer (WDT) is set to prevent the flash memory overerasing due to program crush, etc. An overflow cycle of approximately 19.8 ms is adequate.
5. For writing dummy data to a verifying address, write one byte of data H'FF to an address whose lower two bits are B'00. Verifying data can be read in longwords from the address to which a dummy data is written.
6. If the read data is not erased successfully, set erasing mode again, and repeat the erasing/erasing-verifying sequence as before. The maximum number of repetitions of the erase/erase-verify sequence is 100.

6.4.3 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts including the NMI interrupt are disabled while flash memory is being programmed or erased or while the boot program is executed for the following three reasons.

1. An interrupt during programming/erasure may cause a violation of the programming or erasing algorithm, with the result that normal operation cannot be assured.
2. If interrupt exception handling starts before programming the vector address or during programming/erasure, a correct vector cannot be fetched and the CPU malfunctions.
3. If an interrupt occurs during boot program execution, normal boot mode sequence cannot be carried out.



Note: *The RTS instruction must not be used during a period from dummy-writing of H'FF to a verifying address until reading verifying data

Figure 6.4 Erase/Erase-Verify Flowchart

6.5 Programming/Erasing Protection

There are three types of flash memory programming/erasing protection; hardware protection, software protection, and error protection.

6.5.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted because of a transition to the reset state, subactive mode, subsleep mode, watch mode, or standby mode. Flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), and erase block register 1 (EBR1) are initialized. For a reset by the $\overline{\text{RES}}$ pin, the reset state is entered when the $\overline{\text{RES}}$ signal is held low until oscillation stabilizes after switching on. For a reset during operation, hold the $\overline{\text{RES}}$ signal low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.

6.5.2 Software Protection

Software protection can protect programming/erasing of all flash memory blocks by clearing the SWE bit in FLMCR1. When software protection is enabled, setting the P or E bit in FLMCR1 does not cause a transition to programming mode or erasing mode. By setting the erase block register 1 (EBR1), erasing protection can be set for individual blocks. When EBR1 is set to H'00, erasing protection is set for all blocks.

6.5.3 Error Protection

Error protection is a state in which programming/erasure is forcibly aborted when an error is detected because CPU crush occurs during flash memory programming/erasure, or operation is not performed in accordance with the programming/erasing algorithm. Aborting programming/erasure prevents damage to the flash memory due to overprogramming or overerasing.

When the following errors are detected during programming/erasing of flash memory, the FLER bit in FLMCR2 is set to 1, and the error protection state is entered.

- When the flash memory address being programmed or erased is read (including vector read and instruction fetch)
- Exception handling excluding a reset is started during programming/erasure
- When the SLEEP instruction is executed during programming/erasure

The FLMCR1, FLMCR2, and EBR1 settings are retained, however programming mode or erasing mode is aborted when the error occurred. Programming mode or erasing mode cannot be re-entered by re-setting the P or E bit. However, settings of the PV and EV bits are retained, and a transition can be made to the verifying mode. The error protection state can be cleared only by a reset.

6.6 Power-Down States for Flash Memory

In user mode, the flash memory will operate in either of the following states:

- Normal operating mode
  The flash memory can be read at high speed.
- Power-down operating mode
  The power supply circuit of flash memory can be partly halted. As a result, flash memory can be read with low power consumption.
- Standby mode
  All flash memory circuits are halted.

Table 6.7 shows the correspondence between the operating modes of this LSI and the flash memory. In subactive mode, the flash memory can be set to operate in power-down mode with the PDWND bit in FLPWCR. When the flash memory returns to its normal operating state from the power-down mode or standby mode, a period to stabilize operation of the power supply circuits that were stopped is needed. When the flash memory returns to its normal operating state, bits STS2 to STS0 in SYSCR1 must be set to provide a wait time of at least 20 µs, even when the external clock is being used.

Table 6.7 Flash Memory Operating States

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

6.7 Notes on Setting Module Standby Mode

When the flash memory is set to enter the module standby mode, the system clock supply is stopped to the module, the function is stopped, and the state is the same as that in standby mode. Also programming is stopped in the flash memory. Therefore operation program should be transferred to the RAM and the program should run in the RAM. Then the flash memory should be set to enter the module standby mode.

If an interrupt is generated in module standby mode, the vector address cannot be fetched. As a result, the program may run away.

Before the flash memory is set to enter module standby mode, the corresponding bit in the interrupt enable register should be cleared to 0 and the I bit in CCR should be set to 1. Then after the flash memory enters the module standby mode, the NMI interrupt request should not be generated.

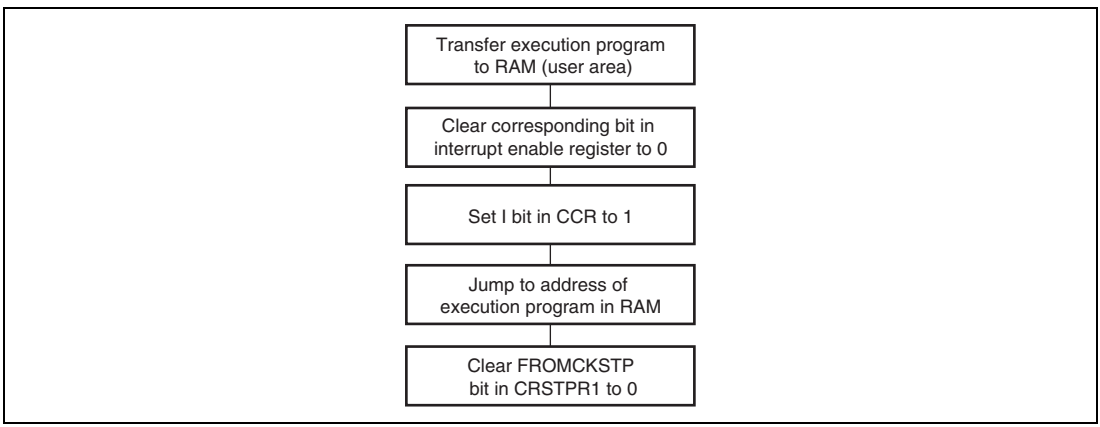


Figure 6.5 Module Standby Mode Setting

Section 7 RAM

This LSI has an on-chip high-speed static RAM. The RAM is connected to the CPU by a 16-bit data bus, enabling two-state access by the CPU to both byte data and word data.

Product Classification		RAM Size	RAM Address
Flash memory version	H8/38602RF	1 Kbyte	H'FB80 to H'FF7F
Masked ROM version	H8/38602R	1 Kbyte	H'FB80 to H'FF7F
	H8/38600R	512 bytes	H'FD80 to H'FF7F

Section 8 I/O Ports

The H8/38602R Group has 13 general I/O ports and six general input-only ports. Port 8 is a large current port, which can drive 15 mA (@V_{OL} = 1.0 V) when a low level signal is output. Any of these ports can become an input port immediately after a reset. They can also be used as I/O pins of the on-chip peripheral modules or external interrupt input pins, and these functions can be switched depending on the register settings. The registers for selecting these functions can be divided into two types: those included in I/O ports and those included in each on-chip peripheral module. General I/O ports are comprised of the port control register for controlling inputs/outputs and the port data register for storing output data and can select inputs/outputs in bit units.

For details on the execution of bit manipulation instructions to the port data register (PDR), see section 2.8.3, Bit-Manipulation Instruction.

For details on block diagrams for each port, see appendix B.1, I/O Port Block Diagrams.

8.1 Port 1

Port 1 is an I/O port also functioning as an asynchronous event counter input pin, timer W I/O pin, RTC output pin, CLKOUT output pin, and interrupt input pin. Figure 8.1 shows its pin configuration.

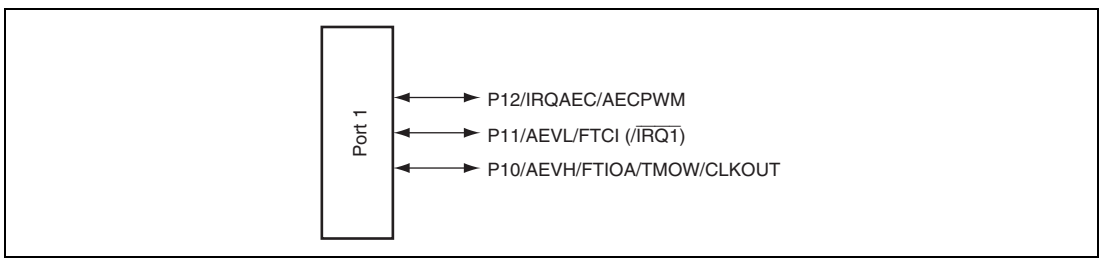


Figure 8.1 Port 1 Pin Configuration

Port 1 has the following registers.

- Port data register 1 (PDR1)
- Port control register 1 (PCR1)
- Port pull-up control register 1 (PUCR1)
- Port mode register 1 (PMR1)

8.1.1 Port Data Register 1 (PDR1)

PDR1 is a register that stores data of port 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
2	P12	0	R/W	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.
1	P11	0	R/W	
0	P10	0	R/W	

8.1.2 Port Control Register 1 (PCR1)

PCR1 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
2	PCR12	0	W	Setting a PCR1 bit to 1 makes the corresponding pin (P12 to P10) an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR1 and in PDR1 are valid when the corresponding pin is designated as a general I/O pin. PCR3 is a write-only register. The read value is undefined.
1	PCR11	0	W	
0	PCR10	0	W	

8.1.3 Port Pull-Up Control Register 1 (PUCR1)

PUCR1 controls the pull-up MOS of the port 1 pins in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
2	PUCR12	0	R/W	When a PCR1 bit is cleared to 0, setting the corresponding PUCR1 bit to 1 turns on the pull-up MOS for the corresponding pin, while clearing the bit to 0 turns off the pull-up MOS.
1	PUCR11	0	R/W	
0	PUCR10	0	R/W	

8.1.4 Port Mode Register 1 (PMR1)

PMR1 controls the selection of functions for port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
5	IRQAEC	0	R/W	P12/IRQAEC/AECPWM Pin Function Switch 0: P12 I/O pin 1: IRQAEC input pin or AECPWM output pin
4	FTCI*	0	R/W	P11/AEVL/FTCI/$\overline{\text{IRQ1}}$ Pin Function Switch
3	AEVL*	0	R/W	00: P11 I/O pin 01: AEVL input pin 1x: FTCI input pin

Bit	Bit Name	Initial Value	R/W	Description
2	CLKOUT	0	R/W	P10/AEVH/FTIOA/TMOW/CLKOUT Pin Function Switch
1	TMOW	0	R/W	
0	AEVH	0	R/W	000: P10 I/O pin and FTIOA I/O pin 001: AEVH input pin 01x: TMOW pin 100: CLKOUT output pin (ϕ_{osc}) 101: CLKOUT output pin (ϕ_{osc}/2) 110: CLKOUT output pin (ϕ_{osc}/4) 111: Setting prohibited

[Legend] x: Don't care.

Note: * When the IRQ1S1 and IRQ1S0 bits in PFCR are set to B'10, the pin function becomes the $\overline{\text{IRQ1}}$ input pin regardless of the setting of these bits.

8.1.5 Pin Functions

The relationship between the register settings and the port functions is shown below.

- P12/IRQAEC/AECPWM pin

Register Name	PMR1	AEGSR	PCR1	Pin Function
Bit Name	IRQAEC	ECPWME	PCR12	
Setting	0	x	0	P12 input pin
			1	P12 output pin
	1	1	x	AECPWM output pin
		0	x	IRQAEC input pin

[Legend] x: Don't care.

- P11/AEVL/FTCI (/$\overline{\text{IRQ1}}$) pin

Register Name	PFCR	PMR1		PCR1	Pin Function
Bit Name	IRQ1S1 and IRQ1S0	FTCI	AEVL	PCR11	
Setting	Other than B'10	0	0	0	P11 input pin
				1	P11 output pin
			1	x	AEVL input pin
		1	x	x	FTCI input pin
	B'10	x	x	x	$\overline{\text{IRQ1}}$ input pin

[Legend] x: Don't care.

- P10/AEVH/FTIOA/TMOW/CLKOUT pin

Register Name	PMR1			TIOR0			PCR1	Pin Function
Bit Name	CLKOUT	TMOW	AEVH	IOA2	IOA1	IOA0	PCR10	
Setting	0	0	0	0	0	0	0	P10 input pin
							1	P10 output pin
						1	x	FTIOA output pin
					1	0	x	FTIOA output pin
				1	x	x	0	P10 input/FTIOA input pin
				1	x	x	1	P10 output/FTIOA input pin
			1	x	x	x	x	AEVH input pin
		1	x	x	x	x	x	TMOW pin
	1	0	0	x	x	x	x	CLKOUT output pin (ϕ_{osc})*
			1	x	x	x	x	CLKOUT output pin (ϕ_{osc}/2)*
		1	0	x	x	x	x	CLKOUT output pin (ϕ_{osc}/4)*

[Legend] x: Don't care.

Note: * Switching the clock (ϕ_{osc}, ϕ_{osc}/2, or ϕ_{osc}/4) for CLKOUT output must be performed when CLKOUT output is halted (CLKOUT = 0).
When making a transition to a power-down mode wherein the system clock oscillator is halted, the output level is retained. (In standby mode, output is the high-impedance state.)
When making a transition from a power-down mode wherein the system clock oscillator is halted, to the active mode wherein the system clock oscillator operates, halt CLKOUT output (CLKOUT = 0) before the transition.

8.1.6 Input Pull-Up MOS

Port 1 has an on-chip input pull-up MOS 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 input pull-up MOS for that pin. The input pull-up MOS function is in the off state after a reset.

(n = 2 to 0)

PCR1n	0		1
PUCR1n	0	1	x
Input Pull-Up MOS	Off	On	Off

[Legend] x: Don't care.

8.2 Port 3

Port 3 is an I/O port also functioning as an SCI3/IrDA I/O pin, comparator reference voltage pin and interrupt pin. Figure 8.2 shows its pin configuration.

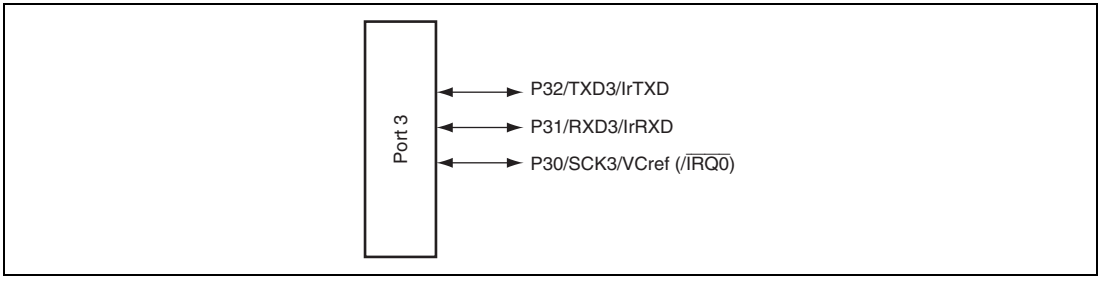


Figure 8.2 Port 3 Pin Configuration

Port 3 has the following registers.

- Port data register 3 (PDR3)
- Port control register 3 (PCR3)
- Port pull-up control register 3 (PUCR3)
- Port mode register 3 (PMR3)

8.2.1 Port Data Register 3 (PDR3)

PDR3 is a register that stores data of port 3.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
2	P32	0	R/W	If port 3 is read while PCR3 bits are set to 1, the values stored in PDR3 are read, regardless of the actual pin states. If port 3 is read while PCR3 bits are cleared to 0, the pin states are read.
1	P31	0	R/W	
0	P30	0	R/W	

8.2.2 Port Control Register 3 (PCR3)

PCR3 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 3.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
2	PCR32	0	W	Setting a PCR3 bit to 1 makes the corresponding pin (P32 to P30) an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR3 and in PDR3 are valid when the corresponding pin is designated as a general I/O pin. PCR3 is a write-only register. The read value is undefined.
1	PCR31	0	W	
0	PCR30	0	W	

8.2.3 Port Pull-Up Control Register 3 (PUCR3)

PUCR3 controls the pull-up MOS of the port 3 pins in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
2	PUCR32	0	R/W	When a PCR3 bit is cleared to 0, setting the corresponding PUCR3 bit to 1 turns on the pull-up MOS for the corresponding pin, while clearing the bit to 0 turns off the pull-up MOS.
1	PUCR31	0	R/W	
0	PUCR30	0	R/W	

8.2.4 Port Mode Register 3 (PMR3)

PMR3 controls the selection of functions for port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
0	VCref	0	R/W	P30/SCK3/VCref Pin Function Switch 0: P30 and SCK3 I/O pin 1: Comparator reference voltage (VCref) pin

8.2.5 Pin Functions

The relationship between the register settings and the port functions is shown below.

- P32/TXD3/IrTXD pin

Register Name	SPCR	IrCR	PCR3	Pin Function
Bit Name	SPC3	IrE	PCR32	
Setting	0	x	0	P32 input pin
			1	P32 output pin
	1	0	x	TXD3 output pin
		1	x	IrTXD output pin

[Legend] x: Don't care.

- P31/RXD3/IrRXD pin

Register Name	SCR3	IrCR	PCR3	Pin Function
Bit Name	RE	IrE	PCR31	
Setting	0	x	0	P31 input pin
			1	P31 output pin
	1	0	x	RXD3 input pin
		1	x	IrRXD input pin

[Legend] x: Don't care.

- P30/SCK3/VC_{ref} (/$\overline{IRQ0}$) pin

Register Name	PFCR	PMR3	SCR3		SMR3	PCR3	Pin Function
Bit Name	IRQ0S1 and IRQ0S0	VC_{ref}	CKE1	CKE0	COM	PCR30	
Setting	Other than B'10	0	0	0	0	0	P30 input pin
						1	P30 output pin
					1	x	SCK3 output pin
				1	x	x	SCK3 output pin
			1	0	x	x	SCK3 input pin
		1	x	x	x	x	VCref pin
	B'10	x	x	x	x	x	IRQ0 input pin

[Legend] x: Don't care.

8.2.6 Input Pull-Up MOS

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

(n = 2 to 0)

PCR3n	0		1
PUCR3n	0	1	x
Input Pull-Up MOS	Off	On	Off

[Legend] x: Don't care.

8.3 Port 8

Port 8 is an I/O port also functioning as a timer W I/O pin. Figure 8.3 shows its pin configuration.

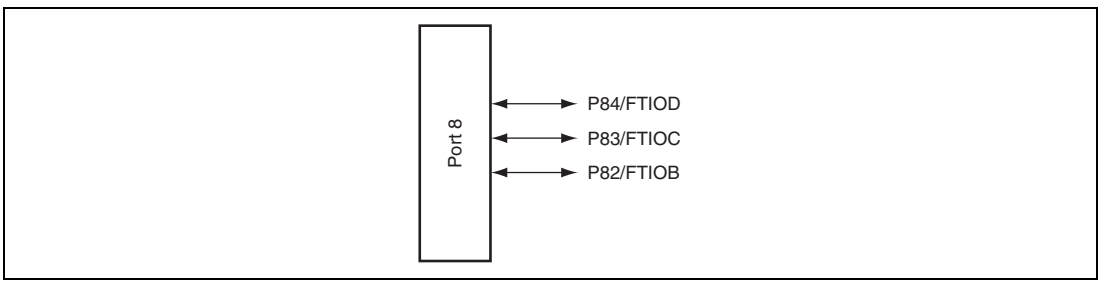


Figure 8.3 Port 8 Pin Configuration

Port 8 has the following registers.

- Port data register 8 (PDR8)
- Port control register 8 (PCR8)
- Port pull-up control register 8 (PUCR8)

8.3.1 Port Data Register 8 (PDR8)

PDR8 is a register that stores data of port 8.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
4	P84	0	R/W	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.
3	P83	0	R/W	
2	P82	0	R/W	
1, 0	—	—	—	Reserved The read value is undefined. These bits cannot be modified.

8.3.2 Port Control Register 8 (PCR8)

PCR8 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 8.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
4	PCR84	0	W	Setting a PCR8 bit to 1 makes the corresponding pin (P84 to P82) an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR8 and in PDR8 are valid when the corresponding pin is designated as a general I/O pin. PCR8 is a write-only register. The read value is undefined.
3	PCR83	0	W	
2	PCR82	0	W	
1, 0	—	—	—	Reserved The read value is undefined. These bits cannot be modified.

8.3.3 Port Pull-Up Control Register 8 (PUCR8)

PUCR8 controls the pull-up MOS of the port 8 pins in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
4	PUCR84	0	R/W	When a PCR8 bit is cleared to 0, setting the corresponding PUCR8 bit to 1 turns on the pull-up MOS for the corresponding pin, while clearing the bit to 0 turns off the pull-up MOS.
3	PUCR83	0	R/W	
2	PUCR82	0	R/W	
1, 0	—	—	—	Reserved The read value is undefined. These bits cannot be modified.

8.3.4 Pin Functions

The relationship between the register settings and the port functions is shown below.

- P84/FTIOD pin

Register Name	TMRW	TIOR1			PCR8	Pin Function
Bit Name	PWMD	IOD2	IOD1	IOD0	PCR84	
Setting	0	0	0	0	0	P84 input pin
					1	P84 output pin
				1	x	FTIOD output pin
			1	x	x	FTIOD output pin
		1	x	x	0	P84 input/FTIOD input pin
					1	P84 output/FTIOD input pin
	1	x	x	x	x	FTIOD output pin

[Legend] x: Don't care.

- P83/FTIOC pin

Register Name	TMRW	TIOR1			PCR8	Pin Function
Bit Name	PWMC	IOC2	IOC1	IOC0	PCR83	
Setting	0	0	0	0	0	P83 input pin
					1	P83 output pin
				1	x	FTIOC output pin
			1	x	x	FTIOC output pin
		1	x	x	0	P83 input/FTIOC input pin
					1	P83 output/FTIOC input pin
	1	x	x	x	x	FTIOC output pin

[Legend] x: Don't care.

- P82/FTIOB pin

Register Name	TMRW	TIOR0			PCR8	Pin Function
Bit Name	PWMB	IOB2	IOB1	IOB0	PCR82	
Setting	0	0	0	0	0	P82 input pin
					1	P82 output pin
				1	x	FTIOB output pin
			1	x	x	FTIOB output pin
		1	x	x	0	P82 input/FTIOB input pin
					1	P82 output/FTIOB input pin
	1	x	x	x	x	FTIOB output pin

[Legend] x: Don't care.

8.3.5 Input Pull-Up MOS

Port 8 has an on-chip input pull-up MOS function that can be controlled by software. When a PCR8 bit is cleared to 0, setting the corresponding PUCR8 bit to 1 turns on the input pull-up MOS for that pin. The input pull-up MOS function is in the off state after a reset.

(n = 4 to 2)

PCR8n	0		1
PUCR8n	0	1	x
Input Pull-Up MOS	Off	On	Off

[Legend] x: Don't care.

8.4 Port 9

Port 9 is an I/O port also functioning as an SSU I/O pin, IIC2 I/O pin and interrupt pin. Figure 8.4 shows its pin configuration.

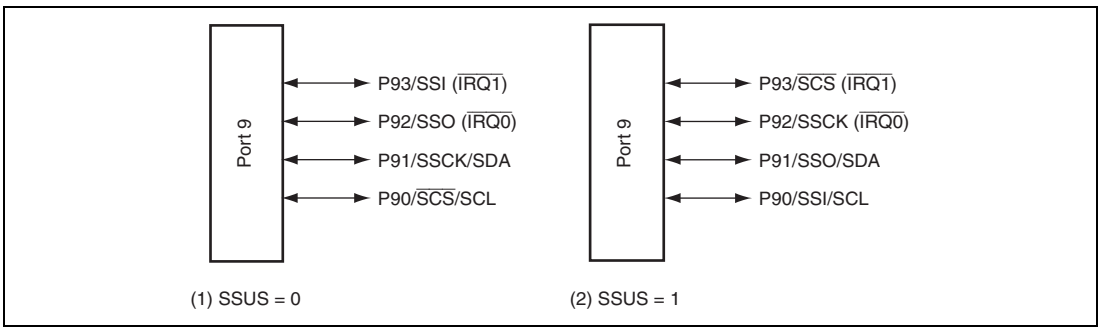


Figure 8.4 Port 9 Pin Configuration

Port 9 has the following registers.

- Port data register 9 (PDR9)
- Port control register 9 (PCR9)
- Port open-drain control register 9 (PODR9)
- Port pull-up control register 9 (PUCR9)

8.4.1 Port Data Register 9 (PDR9)

PDR9 is a register that stores data of port 9.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
3	P93	0	R/W	If port 9 is read while PCR9 bits are set to 1, the values stored in PDR9 are read, regardless of the actual pin states. If port 9 is read while PCR9 bits are cleared to 0, the pin states are read.
2	P92	0	R/W	
1	P91	0	R/W	
0	P90	0	R/W	

8.4.2 Port Control Register 9 (PCR9)

PCR9 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 9.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
3	PCR93	0	W	Setting a PCR9 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR9 and in PDR9 are valid when the corresponding pin is designated as a general I/O pin. PCR9 is a write-only register. The read value is undefined.
2	PCR92	0	W	
1	PCR91	0	W	
0	PCR90	0	W	

8.4.3 Port Open-Drain Control Register 9 (PODR9)

PODR9 selects the output format for port 9 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
3	P93ODR	0	R/W	When a bit among the P93ODR to P90ODR bits is set to 1, the corresponding pin among P93 to P90 functions as the NMOS open-drain output. When cleared to 0, the corresponding pin functions as the CMOS output.
2	P92ODR	0	R/W	
1	P91ODR	0	R/W	
0	P90ODR	0	R/W	

8.4.4 Port Pull-Up Control Register 9 (PUCR9)

PUCR9 controls the pull-up MOS of the port 9 pins in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
3	PUCR93	0	R/W	When a PCR9 bit is cleared to 0, setting the corresponding PUCR9 bit to 1 turns on the pull-up MOS for the corresponding pin, while clearing the bit to 0 turns off the pull-up MOS.
2	PUCR92	0	R/W	
1	PUCR91	0	R/W	
0	PUCR90	0	R/W	

8.4.5 Pin Functions

The relationship between the register settings and the port functions is shown below.

Note on the followings when port 9 is used.

1. When IIC is used, SSU should not be set.
2. When SSU is used, the ICE bit in IIC should be set to 0.
3. The port corresponding to the pins for SSU communication data (SSI and SSO) should not be set.
4. When the pins for communication data (SSI and SSO) are set to open-drain output with the SOOS bit in the SSCRH register of SSU, they are set to open-drain output regardless of the TE and RE bit settings in the SSER register.

- P93/SSI (/$\overline{\text{IRQ1}}$) pin

Register Name	PFCR		PCR9	Pin Function
Bit Name	IRQ1S1 and IRQ1S0	SSUS	PCR93	
Setting	Other than B'01	x	0	P93 input pin
			1	P93 output pin
		0	x	SSI I/O pin
		1	x	$\overline{\text{SCS}}$ I/O pin
	B'01	x	x	$\overline{\text{IRQ1}}$ input pin

[Legend] x: Don't care.

Note: When this pin is used as the SSI/$\overline{\text{SCS}}$ pin, register settings of the SSU are required. For details, see section 15.4.4, Communication Modes and Pin Functions, and appendix B.3, Port 9 Related Register Settings and Pin Functions.

- P92/SSO (/$\overline{\text{IRQ0}}$) pin

Register Name	PFCR		PCR9	Pin Function
Bit Name	IRQ0S1 and IRQ0S0	SSUS	PCR92	
Setting	Other than B'01	x	0	P92 input pin
			1	P92 output pin
		0	x	SSO I/O pin
		1	x	SSCK I/O pin
	B'01	x	x	$\overline{\text{IRQ0}}$ input pin

[Legend] x: Don't care.

Note: When this pin is used as the SSO/SSCK pin, register settings of the SSU are required. For details, see section 15.4.4, Communication Modes and Pin Functions, and appendix B.3, Port 9 Related Register Settings and Pin Functions.

- P91/SSCK/SDA pin

Register Name	PFCR	PCR9	Pin Function
Bit Name	SSUS	PCR91	
Setting	x	0	P91 input pin
		1	P91 output pin
	0	x	SSCK I/O pin
	1	x	SSO I/O pin
	x	x	SDA I/O pin

[Legend] x: Don't care.

Note: When this pin is used as the SSO/SSCK pin, register settings of the SSU are required. For details, see section 15.4.4, Communication Modes and Pin Functions, and appendix B.3, Port 9 Related Register Settings and Pin Functions. When this pin is used as the SDA pin, register settings of the IIC2 are required. For details, see section 16.3.1, I^2C Bus Control Register 1 (ICCR1).
Note that the priority when pin functions conflict is SSU pin > IIC2 pin > P91.

- P90/$\overline{\text{SCS}}$/SCL pin

Register Name	PFCR	PCR9	Pin Function
Bit Name	SSUS	PCR90	
Setting	x	0	P90 input pin
		1	P90 output pin
	0	x	$\overline{\text{SCS}}$ I/O pin
	1	x	SSI I/O pin
	x	x	SCL I/O pin

[Legend] x: Don't care.

Note: When this pin is used as the $\overline{\text{SCS}}$/SSI pin, register settings of the SSU are required. For details, see section 15.4.4, Communication Modes and Pin Functions, and appendix B.3, Port 9 Related Register Settings and Pin Functions. When this pin is used as the SCL pin, register settings of the IIC2 are required. For details, see section 16.3.1, I^2C Bus Control Register 1 (ICCR1).
Note that the priority when pin functions conflict is SSU pin > IIC2 pin > P90.

8.4.6 Input Pull-Up MOS

Port 9 has an on-chip input pull-up MOS function that can be controlled by software. When a PCR9 bit is cleared to 0, setting the corresponding PUCR9 bit to 1 turns on the input pull-up MOS for that pin. The input pull-up MOS function is in the off state after a reset.

(n = 3 to 0)

PCR9n	0		1
PUCR9n	0	1	x
Input Pull-Up MOS	Off	On	Off

[Legend] x: Don't care.

8.5 Port B

Port B is an input-only port also functioning as an interrupt input pin, analog input pin, and comparator pin. Figure 8.5 shows its pin configuration.

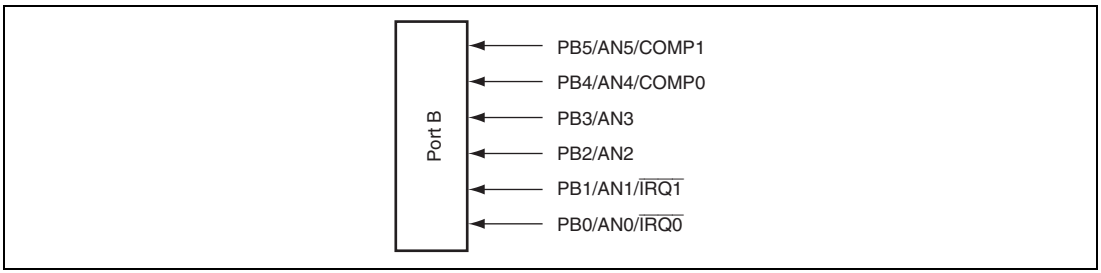


Figure 8.5 Port B Pin Configuration

Port B has the following registers.

- Port data register B (PDRB)
- Port mode register B (PMRB)

8.5.1 Port Data Register B (PDRB)

PDRB is a register that stores data of port B.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
5	PB5	Undefined	R	Reading PDRB always gives the pin states. However, if a port B pin is selected as an analog input channel by the CH3 to CH0 bits in AMR of the A/D converter, that pin is read as 0 regardless of the input voltage.
4	PB4	Undefined	R	
3	PB3	Undefined	R	
2	PB2	Undefined	R	
1	PB1	Undefined	R	
0	PB0	Undefined	R	

8.5.2 Port Mode Register B (PMRB)

PMRB controls the selection of the port B pin functions.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	—	—	Reserved The read value is undefined. These bits cannot be modified.
3	ADTSTCHG	0	R/W	TEST/$\overline{\text{ADTRG}}$ Pin Function Switch Selects whether pin TEST/$\overline{\text{ADTRG}}$ is used as TEST or as $\overline{\text{ADTRG}}$. 0: TEST pin 1: $\overline{\text{ADTRG}}$ input pin For details on the setting of the $\overline{\text{ADTRG}}$ input pin, refer to section 17.4.2, External Trigger Input Timing.
2	—	—	—	Reserved The read value is undefined. This bit cannot be modified.
1	IRQ1	0	R/W	PB1/AN1/$\overline{\text{IRQ1}}$ Pin Function Switch Selects whether pin PB1/AN1/$\overline{\text{IRQ1}}$ is used as PB1/AN1 or as $\overline{\text{IRQ1}}$. 0: PB1/AN1 input pin 1: $\overline{\text{IRQ1}}$ input pin*
0	IRQ0	0	R/W	PB0/AN0/$\overline{\text{IRQ0}}$ Pin Function Switch Selects whether pin PB0/AN0/$\overline{\text{IRQ0}}$ is used as PB0/AN0 or as $\overline{\text{IRQ0}}$. 0: PB0/AN0 input pin 1: $\overline{\text{IRQ0}}$ input pin*

Note: * When the IRQnS1 and IRQnS0 (n = 1 or 0) bits in PFCR are set to a value other than B'00, these bits should not be set since the $\overline{\text{IRQn}}$ pin is assigned to another port.

8.5.3 Pin Functions

The relationship between the register settings and the port functions is shown below.

- PB5/AN5/COMP1 pin

Register Name	AMR	Pin Function
Bit Name	CH3 to CH0	
Setting	Other than B'1001	PB5/COMP1 input pin
	B'1001	AN5 input pin

[Legend] x: Don't care.

- PB4/AN4/COMP0 pin

Register Name	AMR	Pin Function
Bit Name	CH3 to CH0	
Setting	Other than B'1000	PB4/COMP0 input pin
	B'1000	AN4 input pin

[Legend] x: Don't care.

- PB3/AN3 pin

Register Name	AMR	Pin Function
Bit Name	CH3 to CH0	
Setting	Other than B'0111	PB3 input pin
	B'0111	AN3 input pin

[Legend] x: Don't care.

- PB2/AN2 pin

Register Name	AMR	Pin Function
Bit Name	CH3 to CH0	
Setting	Other than B'0110	PB2 input pin
	B'0110	AN2 input pin

[Legend] x: Don't care.

- PB1/AN1/$\overline{\text{IRQ1}}$ pin

Register Name	PMRB	AMR	PFCR	Pin Function
Bit Name	IRQ1	CH3 to CH0	IRQ1S1 and IRQ1S0	
Setting	0	Other than B'0101	B'xx	PB1 input pin
		B'0101	B'xx	AN1 input pin
	1	B'xxxx	B'00	$\overline{\text{IRQ1}}$ input pin
			Other than B'00	Setting prohibited

[Legend] x: Don't care.

- PB0/AN0/$\overline{\text{IRQ0}}$ pin

Register Name	PMRB	AMR	PFCR	Pin Function
Bit Name	IRQ0	CH3 to CH0	IRQ0S1 and IRQ0S0	
Setting	0	Other than B'0100	B'xx	PB0 input pin
		B'0100	B'xx	AN0 input pin
	1	B'xxxx	B'00	$\overline{\text{IRQ0}}$ input pin
			Other than B'00	Setting prohibited

[Legend] x: Don't care.

8.6 Input/Output Data Inversion

8.6.1 Serial Port Control Register (SPCR)

SPCR switches input/output data inversion of the RXD3 (IrRXD) and TXD3 (IrTXD) pins.

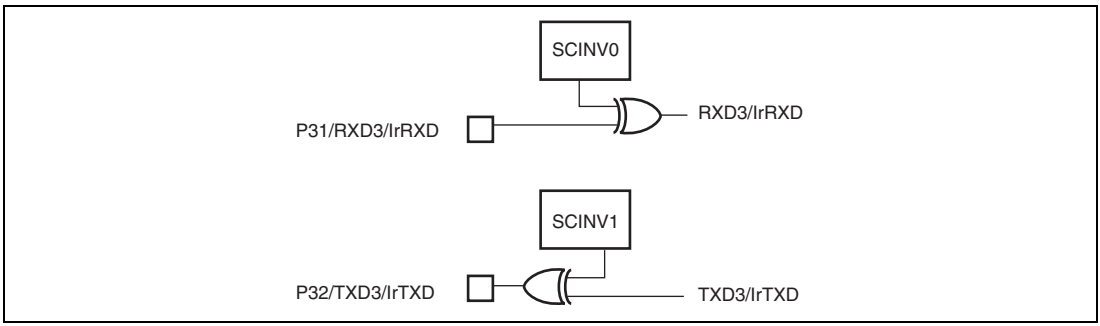


Figure 8.6 Input/Output Data Inversion Function

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
4	SPC3	0	R/W	P32/TXD3/IrTXD Pin Function Switch Selects whether pin P32/TXD3/IrTXD is used as P32 or as TXD3/IrTXD. 0: P32 I/O pin 1: TXD3/IrTXD output pin* Note: * Set the TE bit in SCR3 after setting this bit to 1.
3, 2	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
1	SCINV1	0	R/W	TXD3/IrTXD Pin Output Data Inversion Switch Specifies whether the logic level of output data of the TXD3/IrTXD pin is to be inverted or not. 0: TXD3/IrTXD output data is not inverted 1: TXD3/IrTXD output data is inverted

Bit	Bit Name	Initial Value	R/W	Description
0	SCINV0	0	R/W	RXD3/IrRXD Pin Input Data Inversion Switch Specifies whether the logic level of input data of the RXD3/IrRXD pin is to be inverted or not. 0: RXD3/IrRXD input data is not inverted 1: RXD3/IrRXD input data is inverted

Note: When the serial port control register is modified, the data being input or output up to that point is inverted immediately after the modification, and an invalid data change is input or output. When modifying the serial port control register, modification must be made in a state in which data changes are invalidated.

8.6.2 Port Function Control Register (PFCR)

PFCR changes the SSU pin assignments, and assigns the $\overline{\text{IRQ0}}$ and $\overline{\text{IRQ1}}$ input pins to other ports.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	—	Reserved These bits are always read as 0. These bits cannot be modified.
4	SSUS	0	R/W	SSU Pin Select Changes the SSU pin assignments. 0: SSI is assigned to P93 SSO is assigned to P92 $\overline{\text{SSCK}}$ is assigned to P91 $\overline{\text{SCS}}$ is assigned to P90 1: SSI is assigned to P90 SSO is assigned to P91 $\overline{\text{SSCK}}$ is assigned to P92 $\overline{\text{SCS}}$ is assigned to P93
3	IRQ1S1	0	R/W	IRQ1 Select 1, 0
2	IRQ1S0	0	R/W	00: $\overline{\text{IRQ1}}$ is input from PB1 01: $\overline{\text{IRQ1}}$ is input from P93 10: $\overline{\text{IRQ1}}$ is input from P11 11: Setting prohibited
1	IRQ0S1	0	R/W	IRQ0 Select 1, 0
0	IRQ0S0	0	R/W	00: $\overline{\text{IRQ0}}$ is input from PB0 01: $\overline{\text{IRQ0}}$ is input from P92 10: $\overline{\text{IRQ0}}$ is input from P30 11: Setting prohibited

8.7 Usage Notes

8.7.1 How to Handle Unused Pin

If an I/O pin not used by the user system is floating, pull it up or down.

- If an unused pin is an input pin, handle it in one of the following ways:
  - Pull it up to Vcc with an on-chip pull-up MOS.
  - Pull it up to Vcc with an external resistor of approximately 100 kΩ.
  - Pull it down to Vss with an external resistor of approximately 100 kΩ.
  - For a pin also used by the A/D converter, pull it up to AVcc with an external resistor of approximately 100 kΩ.
- If an unused pin is an output pin, handle it in one of the following ways:
  - Set the output of the unused pin to high and pull it up to Vcc with an external resistor of approximately 100 kΩ.
  - Set the output of the unused pin to low and pull it down to GND with an external resistor of approximately 100 kΩ.

8.7.2 Input Characteristics Difference due to Pin Function

When the functions of pins $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, IRQAEC, AEVL, AEVH, SCK3, FTIOA to FTIOD, FTCI, SSCK, $\overline{\text{SCS}}$, SDA, and SCL are selected, the corresponding pins have the schmitt-trigger input characteristics, which are different from the ones when they are used as the port input pins.

For example, the input high voltage and the input low voltage of the PB0/AN0/$\overline{\text{IRQ0}}$ pin differ when the pin is used as PB0 input or $\overline{\text{IRQ0}}$ input. For details, refer to table 21.2 which lists the DC characteristics for F-ZTAT version, and table 21.13 which lists the DC characteristics for masked ROM version.

Section 9 Timer B1

Timer B1 is an 8-bit timer that increments each time a clock pulse is input. This timer has two operating modes, interval and auto reload. Figure 9.1 shows a block diagram of timer B1.

9.1 Features

- Selection of eight internal clock sources (ϕ/8192, ϕ/2048, ϕ/256, ϕ/64, ϕ/16, ϕ/4, ϕ_w/1024, and ϕ_w/256).
- An interrupt is generated when the counter overflows.
- Use of module standby mode enables this module to be placed in standby mode independently when not used. (Timer B1 is halted as the initial value. For details, refer to section 5.4, Module Standby Function.)

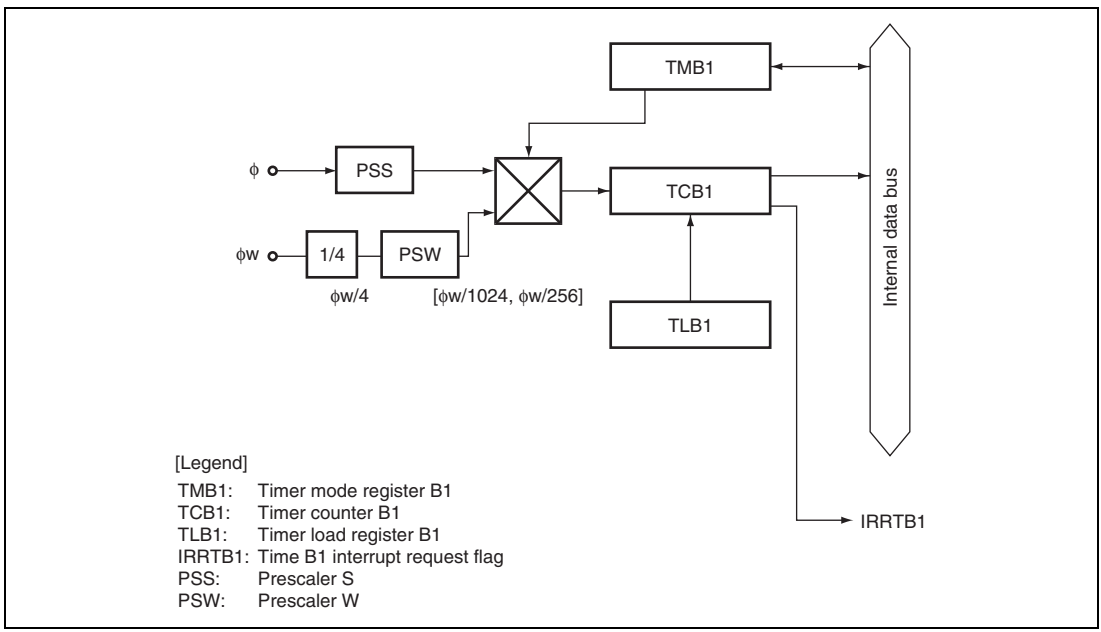


Figure 9.1 Block Diagram of Timer B1

9.2 Register Descriptions

Timer B1 has the following registers.

- Timer mode register B1 (TMB1)
- Timer counter B1 (TCB1)
- Timer load register B1 (TLB1)

9.2.1 Timer Mode Register B1 (TMB1)

TMB1 selects the auto-reload function and input clock.

Bit	Bit Name	Initial Value	R/W	Description
7	TMB17	0	R/W	Auto-Reload Function Select 0: Interval timer function selected 1: Auto-reload function selected
6	TMB16	0	R/W	Counter Operation/Stop Select 0: Counter stopped 1: Counter operates
5 to 3	—	All 1	—	Reserved These bits are always read as 1.
2	TMB12	0	R/W	Counter Clock Select
1	TMB11	0	R/W	000: Internal clock: ϕ/8192
0	TMB10	0	R/W	001: Internal clock: ϕ/2048 010: Internal clock: ϕ/256 011: Internal clock: ϕ/64 100: Internal clock: ϕ/16 101: Internal clock: ϕ/4 110: Internal clock: ϕ_w/1024 111: Internal clock: ϕ_w/256

9.2.2 Timer Counter B1 (TCB1)

TCB1 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 TMB12 to TMB10 in TMB1. TCB1 values can be read by the CPU. When TCB1 overflows from H'FF to H'00 or to the value set in TLB1, the IRRTB1 flag in IRR2 is set to 1. TCB1 is allocated to the same address as TLB1. TCB1 is initialized to H'00.

9.2.3 Timer Load Register B1 (TLB1)

TLB1 is an 8-bit write-only register for setting the reload value of TCB1. Setting the reload value to TLB1 must be done when bit TMB16 in TMB1 is cleared to 0. When a reload value is set in TLB1, the same value is loaded into TCB1 as well, and TCB1 starts counting up from that value. When TCB1 overflows during operation in auto-reload mode, the TLB1 value is loaded into TCB1. Accordingly, overflow periods can be set within the range of 1 to 256 input clocks. TLB1 is allocated to the same address as TCB1. TLB1 is initialized to H'00.

9.3 Usage Method

Figure 9.2 shows the initial setting flow of timer B1 after a reset, and figure 9.3 shows the processing flow for changing a setting during counter operation. Bit TMB16 in TMB1 must be cleared to 0 when setting timer B1, as shown in the figures. Operation is not guaranteed when a setting is made with bit TMB16 in TMB1 set to 1.

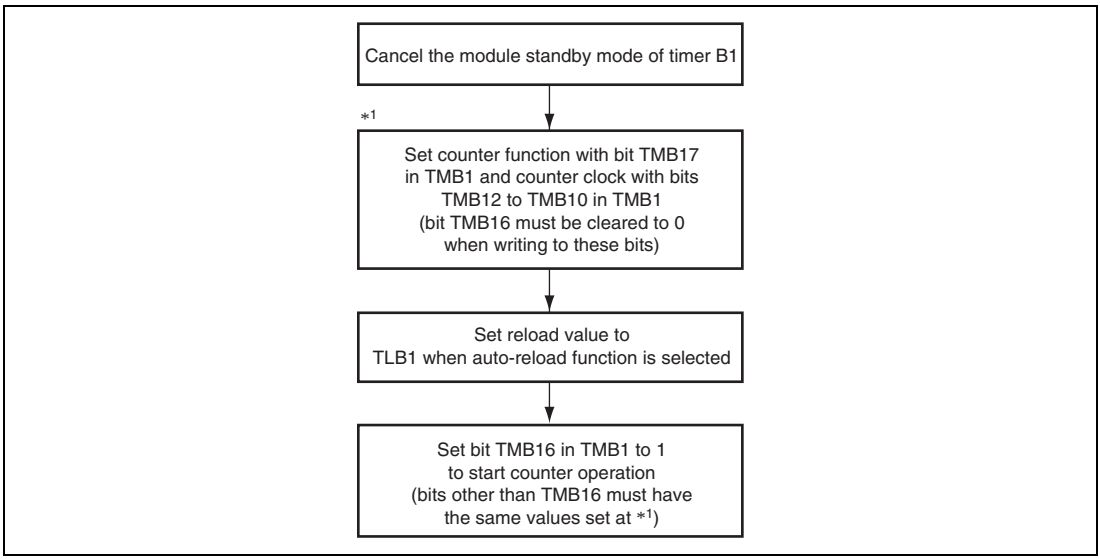


Figure 9.2 Timer B1 Initial Setting Flow

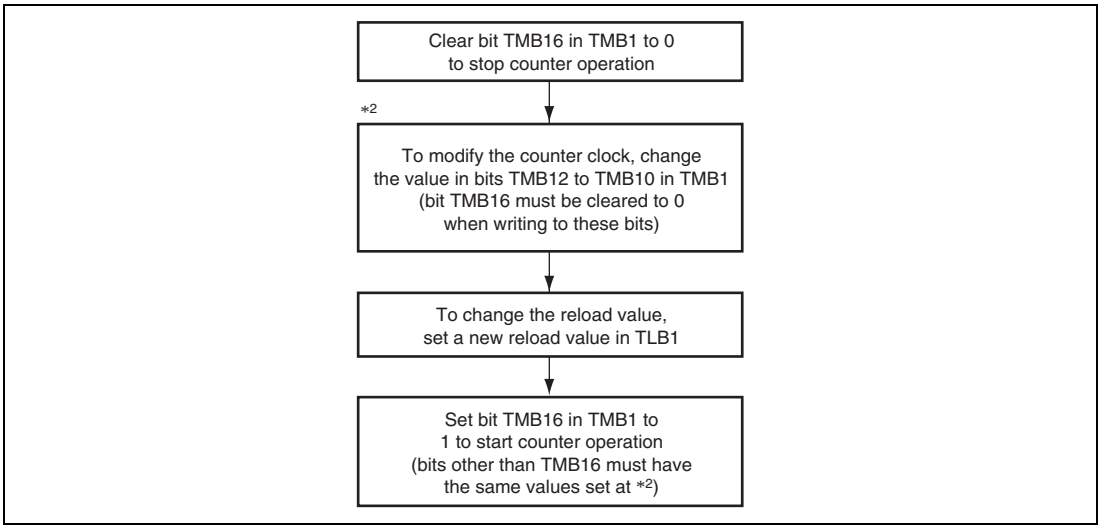


Figure 9.3 Processing Flow When Changing Setting during Counter Operation

9.4 Operation

9.4.1 Interval Timer Operation

When bit TMB17 in TMB1 is cleared to 0, timer B1 functions as an 8-bit interval timer. Upon reset, TCB1 is cleared to H'00 and bit TMB17 is cleared to 0, so the interval timer function is selected immediately after a reset. The operating clock of timer B1 is selected from eight internal clock signals output by prescaler S or prescaler W. The selection is made by bits TMB12 to TMB10 in TMB1.

After bit TMB16 in TMB1 is set to 1 to start the counter operation and the count value in TMB1 reaches H'FF, the next clock signal input causes timer B1 to overflow, setting flag IRRTB1 in IRR2 to 1. If IENTB1 in IENR2 is 1, an interrupt is requested to the CPU.

At overflow, TCB1 returns to H'00 and starts counting up again. Even though interval timer operation (TMB17 = 0) is selected, when a value is set in TLB1 with bit TMB16 in TMB1 cleared to 0, the same value is set in TCB1.

9.4.2 Auto-Reload Timer Operation

Setting bit TMB17 in TMB1 to 1 causes timer B1 to function as an 8-bit auto-reload timer. When a reload value is set in TLB1 with bit TMB16 in TMB1 cleared to 0, the same value is loaded into TCB1. After bit TMB16 in TMB1 is set to 1 to start the counter operation and the count value in TCB1 reaches H'FF, the next clock signal input causes timer B1 to overflow. The TLB1 value is then loaded into TCB1, and the count continues from that value. The overflow period can be set within a range from 1 to 256 input clocks, depending on the TLB1 value.

The clock sources and interrupts in auto-reload mode are the same as in interval mode. To set a new value in TLB1 in auto-reload mode (TMB17 = 1), clear bit TMB16 in TMB1 to 0 before making the new setting.

9.5 Timer B1 Operating Modes

Table 9.1 shows the timer B1 operating modes.

Table 9.1 Timer B1 Operating Modes

	Active		Sleep						Oscillation Stabilization Time		
Clock Source	High-speed	Medium-speed	High-speed	Medium-speed	Watch	Sub-active	Sub-sleep	Standby	Standby to Active	Subsleep to Active	Watch to Active
φw/256, φw/1024	o	o	o	o	o	o	o	×	×	o	o
φ/4, φ/16, φ/64, φ/256, φ/2048, φ/8192	o	o	o	o	×	×	×	×	×	×	×

[Legend] o: Counting enabled
×: Counting disabled (Counter value retained)

Section 10 Timer W

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

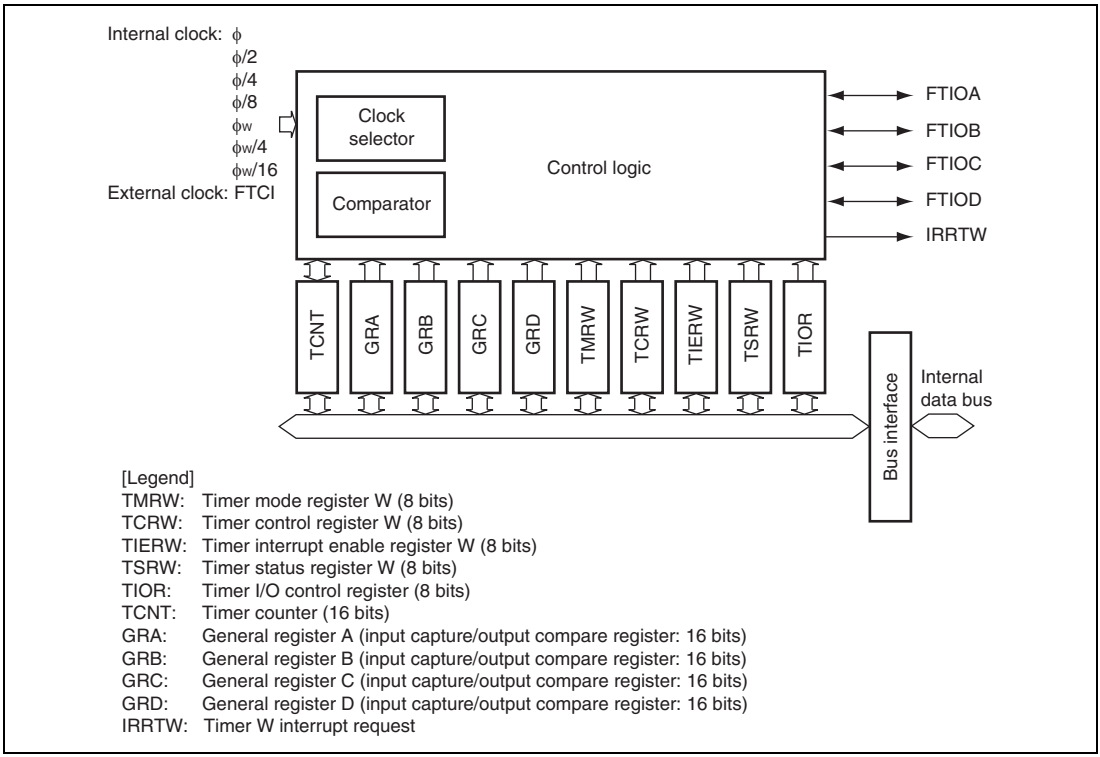
10.1 Features

- Selection of eight counter clock sources: seven internal clocks (ϕ, $\phi/2$, $\phi/4$, $\phi/8$, ϕ_w, $\phi_w/4$, and $\phi_w/16$) and an external clock (external events can be counted)
- Capability to process up to four pulse outputs or four pulse inputs
- Four general registers:
  - — Independently assignable output compare or input capture functions
  - — 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
  - — 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.
- Any initial timer output value can be set
- Five interrupt sources
  Four compare match/input capture interrupts and an overflow interrupt.
- Use of module standby mode enables this module to be placed in standby mode independently when not used. (The timer W is halted as the initial value. For details, refer to section 5.4, Module Standby Function.)

Table 10.1 summarizes the timer W functions, and figure 10.1 shows a block diagram of the timer W.

Table 10.1 Timer W Functions

Item		Counter	Input/Output Pins			
			FTIOA	FTIOB	FTIOC	FTIOD
Count clock		Internal clocks: ϕ, ϕ/2, ϕ/4, ϕ/8, ϕ_w, ϕ_w/4, and ϕ_w/16 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



[Legend]
TMRW: Timer mode register W (8 bits)
TCRW: Timer control register W (8 bits)
TIERW: Timer interrupt enable register W (8 bits)
TSRW: Timer status register W (8 bits)
TIOR: Timer I/O control register (8 bits)
TCNT: Timer counter (16 bits)
GRA: General register A (input capture/output compare register: 16 bits)
GRB: General register B (input capture/output compare register: 16 bits)
GRC: General register C (input capture/output compare register: 16 bits)
GRD: General register D (input capture/output compare register: 16 bits)
IRRTW: Timer W interrupt request

Figure 10.1 Timer W Block Diagram

10.2 Input/Output Pins

Table 10.2 shows the pin configuration of the timer W.

Table 10.2 Pin Configuration

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

10.3 Register Descriptions

The timer W has the following registers.

- Timer mode register W (TMRW)
- Timer control register W (TCRW)
- Timer interrupt enable register W (TIERW)
- Timer status register W (TSRW)
- Timer I/O control register 0 (TIOR0)
- Timer I/O control register 1 (TIOR1)
- Timer counter (TCNT)
- General register A (GRA)
- General register B (GRB)
- General register C (GRC)
- General register D (GRD)

10.3.1 Timer Mode Register W (TMRW)

TMRW selects the general register functions and the timer output mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CTS	0	R/W	Counter Start The counter operation is halted when this bit is 0, while it can be performed when this bit is 1.
6	—	1	—	Reserved This bit is always read as 1.
5	BUFEB	0	R/W	Buffer Operation B Selects the GRD function. 0: GRD operates as an input capture/output compare register 1: GRD operates as the buffer register for GRB
4	BUFEA	0	R/W	Buffer Operation A Selects the GRC function. 0: GRC operates as an input capture/output compare register 1: GRC operates as the buffer register for GRA
3	—	1	—	Reserved This bit is always read as 1.
2	PWMD	0	R/W	PWM Mode D Selects the output mode of the FTIOD pin. 0: FTIOD operates normally (output compare output) 1: PWM output
1	PWMC	0	R/W	PWM Mode C Selects the output mode of the FTIOC pin. 0: FTIOC operates normally (output compare output) 1: PWM output
0	PWMB	0	R/W	PWM Mode B Selects the output mode of the FTIOB pin. 0: FTIOB operates normally (output compare output) 1: PWM output

10.3.2 Timer Control Register W (TCRW)

TCRW selects the timer counter clock source, selects a clearing condition, and specifies the timer output levels.

Bit	Bit Name	Initial Value	R/W	Description
7	CCLR	0	R/W	Counter Clear The TCNT value is cleared by compare match A when this bit is 1. When it is 0, TCNT operates as a free-running counter.
6	CKS2	0	R/W	Clock Select 2 to 0 Select the TCNT clock source. 000: Internal clock: counts on ϕ 001: Internal clock: counts on $\phi/2$ 010: Internal clock: counts on $\phi/4$ 011: Internal clock: counts on $\phi/8$ 100: Internal clock: counts on ϕ_w 101: Internal clock: counts on $\phi_w/4$ 110: Internal clock: counts on $\phi_w/16$ 111: Counts on rising edges of the external event (FTCI) With a setting of 0xx, the timer W can be used only in active mode or sleep mode. Do not make this setting in subactive mode or subsleep mode. When 100 is set in subactive mode or subsleep mode, the timer W can be used only when ϕ_w is selected as the CPU operating clock. When 101 is set in subactive mode or subsleep mode, the timer W can be used only when ϕ_w or $\phi_w/2$ is selected as the CPU operating clock.
5	CKS1	0	R/W	
4	CKS0	0	R/W	
3	TOD	0	R/W	Timer Output Level Setting D Sets the output value of the FTIOD pin until the first compare match D is generated. 0: Output value is 0* 1: Output value is 1*
2	TOC	0	R/W	Timer Output Level Setting C Sets the output value of the FTIOC pin until the first compare match C is generated. 0: Output value is 0* 1: Output value is 1*

Bit	Bit Name	Initial Value	R/W	Description
1	TOB	0	R/W	Timer Output Level Setting B Sets the output value of the FTIOB pin until the first compare match B is generated. 0: Output value is 0* 1: Output value is 1*
0	TOA	0	R/W	Timer Output Level Setting A Sets the output value of the FTIOA pin until the first compare match A is generated. 0: Output value is 0* 1: Output value is 1*

[Legend] x: Don't care.

Note: * The change of the setting is immediately reflected in the output value.

10.3.3 Timer Interrupt Enable Register W (TIERW)

TIERW controls the timer W interrupt request.

Bit	Bit Name	Initial Value	R/W	Description
7	OVIE	0	R/W	Timer Overflow Interrupt Enable When this bit is set to 1, FOVI interrupt requested by OVF flag in TSRW is enabled.
6 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	IMIED	0	R/W	Input Capture/Compare Match Interrupt Enable D When this bit is set to 1, IMID interrupt requested by IMFD flag in TSRW is enabled.
2	IMIEC	0	R/W	Input Capture/Compare Match Interrupt Enable C When this bit is set to 1, IMIC interrupt requested by IMFC flag in TSRW is enabled.
1	IMIEB	0	R/W	Input Capture/Compare Match Interrupt Enable B When this bit is set to 1, IMIB interrupt requested by IMFB flag in TSRW is enabled.
0	IMIEA	0	R/W	Input Capture/Compare Match Interrupt Enable A When this bit is set to 1, IMIA interrupt requested by IMFA flag in TSRW is enabled.

10.3.4 Timer Status Register W (TSRW)

TSRW shows the status of interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT overflows from H'FFFF to H'0000 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
6 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	IMFD	0	R/(W)*	Input Capture/Compare Match Flag D [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 [Clearing condition] Read IMFD when IMFD = 1, then write 0 in IMFD
2	IMFC	0	R/(W)*	Input Capture/Compare Match Flag C [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 [Clearing condition] Read IMFC when IMFC = 1, then write 0 in IMFC

Bit	Bit Name	Initial Value	R/W	Description
1	IMFB	0	R/(W)*	Input Capture/Compare Match Flag B [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 [Clearing condition] Read IMFB when IMFB = 1, then write 0 in IMFB
0	IMFA	0	R/(W)*	Input Capture/Compare Match Flag A [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 [Clearing condition] Read IMFA when IMFA = 1, then write 0 in IMFA

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

10.3.5 Timer I/O Control Register 0 (TIOR0)

TIOR0 selects the functions of GRA and GRB, and specifies the functions of the FTIOA and FTIOB pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1.
6	IOB2	0	R/W	I/O Control B2 Selects the GRB function. 0: GRB functions as an output compare register 1: GRB functions as an input capture register

Bit	Bit Name	Initial Value	R/W	Description
5	IOB1	0	R/W	I/O Control B1 and B0
4	IOB0	0	R/W	When IOB2 = 0, 00: No output at compare match 01: 0 output to the FTIOB pin at GRB compare match 10: 1 output to the FTIOB pin at GRB compare match 11: Output toggles to the FTIOB pin at GRB compare match When IOB2 = 1, 00: Input capture at rising edge at the FTIOB pin 01: Input capture at falling edge at the FTIOB pin 1x: Input capture at rising and falling edges of the FTIOB pin
3	—	1	—	Reserved This bit is always read as 1.
2	IOA2	0	R/W	I/O Control A2 Selects the GRA function. 0: GRA functions as an output compare register 1: GRA functions as an input capture register
1	IOA1	0	R/W	I/O Control A1 and A0
0	IOA0	0	R/W	When IOA2 = 0, 00: No output at compare match 01: 0 output to the FTIOA pin at GRA compare match 10: 1 output to the FTIOA pin at GRA compare match 11: Output toggles to the FTIOA pin at GRA compare match When IOA2 = 1, 00: Input capture at rising edge of the FTIOA pin 01: Input capture at falling edge of the FTIOA pin 1x: Input capture at rising and falling edges of the FTIOA pin

[Legend] x: Don't care.

10.3.6 Timer I/O Control Register 1 (TIOR1)

TIOR1 selects the functions of GRC and GRD, and specifies the functions of the FTIOC and FTIOD pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1.
6	IOD2	0	R/W	I/O Control D2 Selects the GRD function. 0: GRD functions as an output compare register 1: GRD functions as an input capture register
5	IOD1	0	R/W	I/O Control D1 and D0
4	IOD0	0	R/W	When IOD2 = 0, 00: No output at compare match 01: 0 output to the FTIOD pin at GRD compare match 10: 1 output to the FTIOD pin at GRD compare match 11: Output toggles to the FTIOD pin at GRD compare match When IOD2 = 1, 00: Input capture at rising edge at the FTIOD pin 01: Input capture at falling edge at the FTIOD pin 1x: Input capture at rising and falling edges at the FTIOD pin
3	—	1	—	Reserved This bit is always read as 1.
2	IOC2	0	R/W	I/O Control C2 Selects the GRC function. 0: GRC functions as an output compare register 1: GRC functions as an input capture register

Bit	Bit Name	Initial Value	R/W	Description
1	IOC1	0	R/W	I/O Control C1 and C0
0	IOC0	0	R/W	When IOC2 = 0, 00: No output at compare match 01: 0 output to the FTIOC pin at GRC compare match 10: 1 output to the FTIOC pin at GRC compare match 11: Output toggles to the FTIOC pin at GRC compare match When IOC2 = 1, 00: Input capture to GRC at rising edge of the FTIOC pin 01: Input capture to GRC at falling edge of the FTIOC pin 1x: Input capture to GRC at rising and falling edges of the FTIOC pin

[Legend] x: Don't care.

10.3.7 Timer Counter (TCNT)

TCNT is a 16-bit readable/writable up-counter. 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 the CCLR in TCRW to 1. When TCNT overflows (changes from H'FFFF to H'0000), the OVF flag in TSRW is set to 1. If OVIE in TIERW is set to 1 at this time, an interrupt request is generated. TCNT must always be read or written in 16-bit units; 8-bit access is not allowed. TCNT is initialized to H'0000 by a reset.

10.3.8 General Registers A to D (GRA to GRD)

Each general register is a 16-bit readable/writable 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. An interrupt request is generated at this time, when IMIEA, IMIEB, IMIEC, or IMIED in TIERW is set to 1. Compare match output can be selected in TIOR.

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 generated. The edge of the input-capture signal is selected in TIOR.

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

For example, when GRA is set as an output-compare register and GRC is set as the buffer register for GRA, the value in the buffer register GRC is sent to GRA whenever compare match A is generated.

When GRA is set as an input-capture register and GRC is set as the buffer register for GRA, the value in TCNT is transferred to GRA and the value in GRA is transferred to GRC whenever an input capture is generated.

GRA to GRD must be written or read in 16-bit units; 8-bit access is not allowed. GRA to GRD are initialized to H'FFFF by a reset.

10.4 Operation

The timer W has the following operating modes.

- Normal Operation
- PWM Operation

10.4.1 Normal Operation

TCNT performs free-running or periodic counting operations. After a reset, TCNT is set as a free-running counter. When the CTS bit in TMRW is set to 1, TCNT starts incrementing the count. When the count overflows from H'FFFF to H'0000, the OVF flag in TSRW is set to 1. If the OVIE in TIERW is set to 1, an interrupt request is generated. Figure 10.2 shows free-running counting.

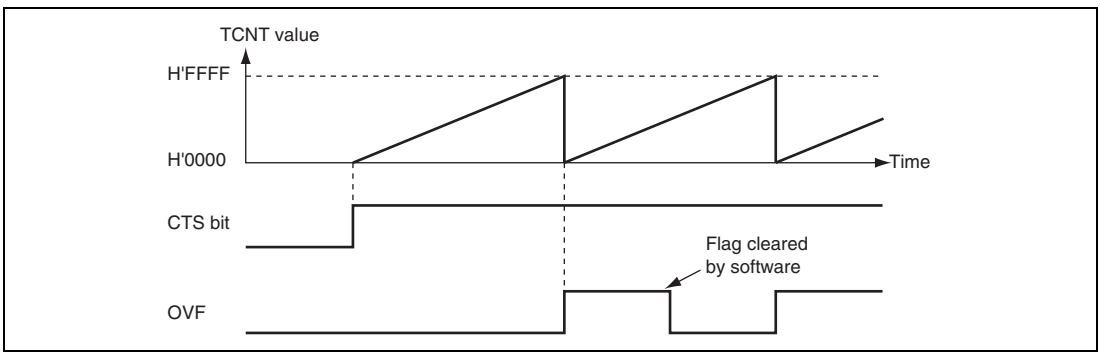


Figure 10.2 Free-Running Counter Operation

Periodic counting operation can be performed when GRA is set as an output compare register and CCLR bit in TCRW is set to 1. When the count matches GRA, TCNT is cleared to H'0000, the IMFA flag in TSRW is set to 1. If the corresponding IMIEA bit in TIERW is set to 1, an interrupt request is generated. TCNT continues counting from H'0000. Figure 10.3 shows periodic counting.

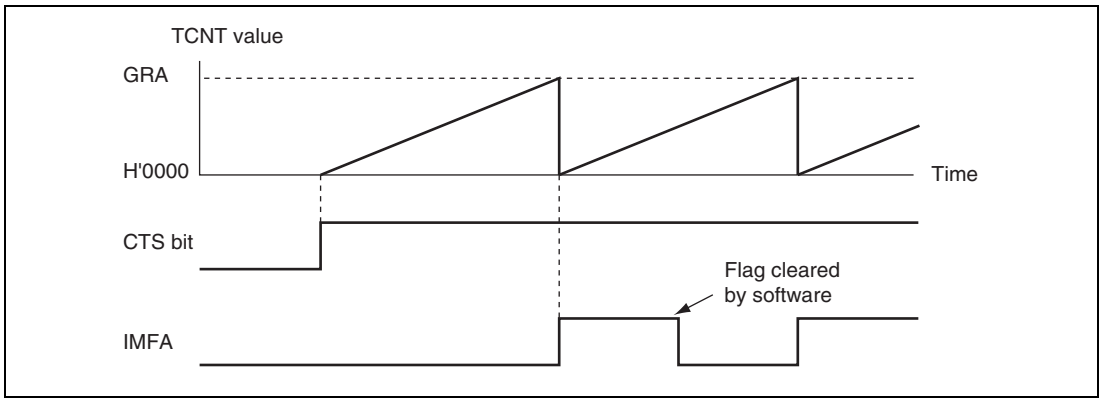


Figure 10.3 Periodic Counter Operation

By setting a general register as an output compare register, compare match A, B, C, or D can cause the output at the FTIOA, FTIOB, FTIOC, or FTIOD pin to output 0, output 1, or toggle. Figure 10.4 shows an example of 0 and 1 output when 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 signal is already at the selected output level, the signal level does not change at compare match.

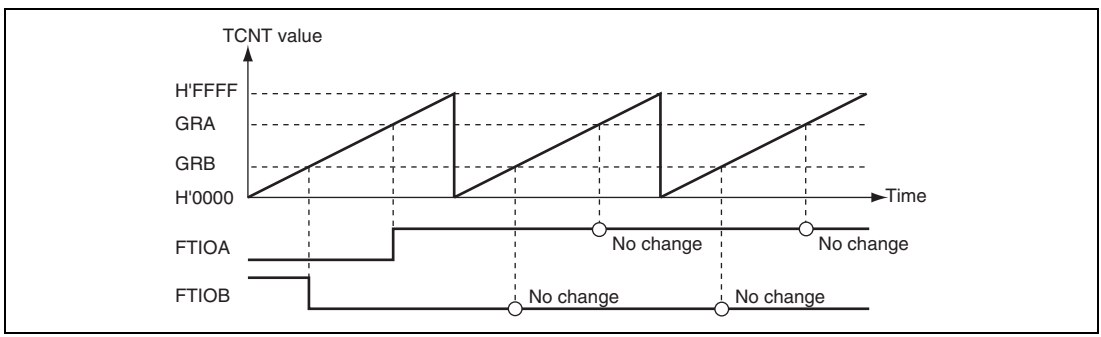


Figure 10.4 0 and 1 Output Example (TOA = 0, TOB = 1)

Figure 10.5 shows an example of toggle output when TCNT operates as a free-running counter, and toggle output is selected for both compare match A and B.

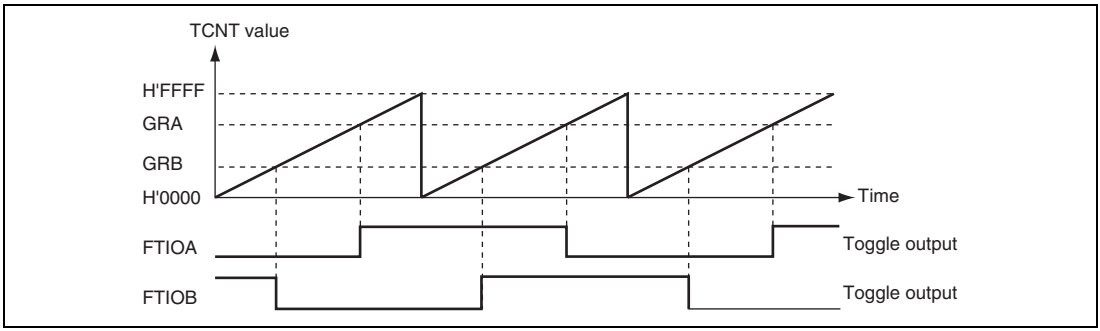


Figure 10.5 Toggle Output Example (TOA = 0, TOB = 1)

Figure 10.6 shows another example of toggle output when TCNT operates as a periodic counter, cleared by compare match A. Toggle output is selected for both compare match A and B.

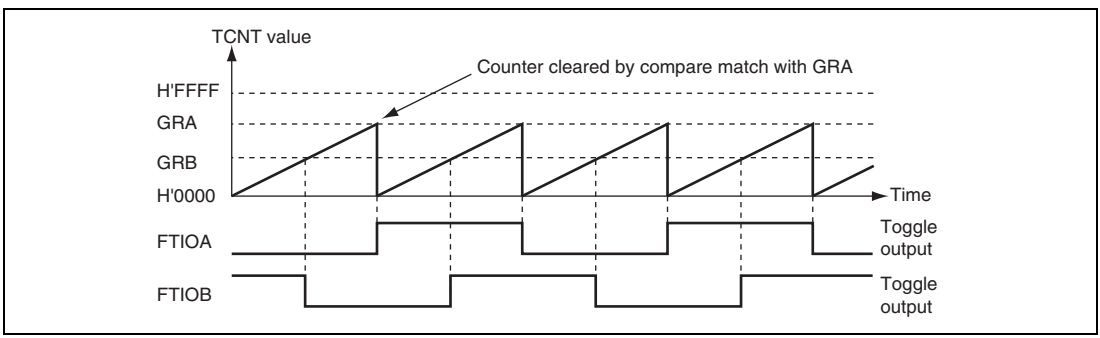


Figure 10.6 Toggle Output Example (TOA = 0, TOB = 1)

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. By using the input-capture function, the pulse width and periods can be measured. Figure 10.7 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.

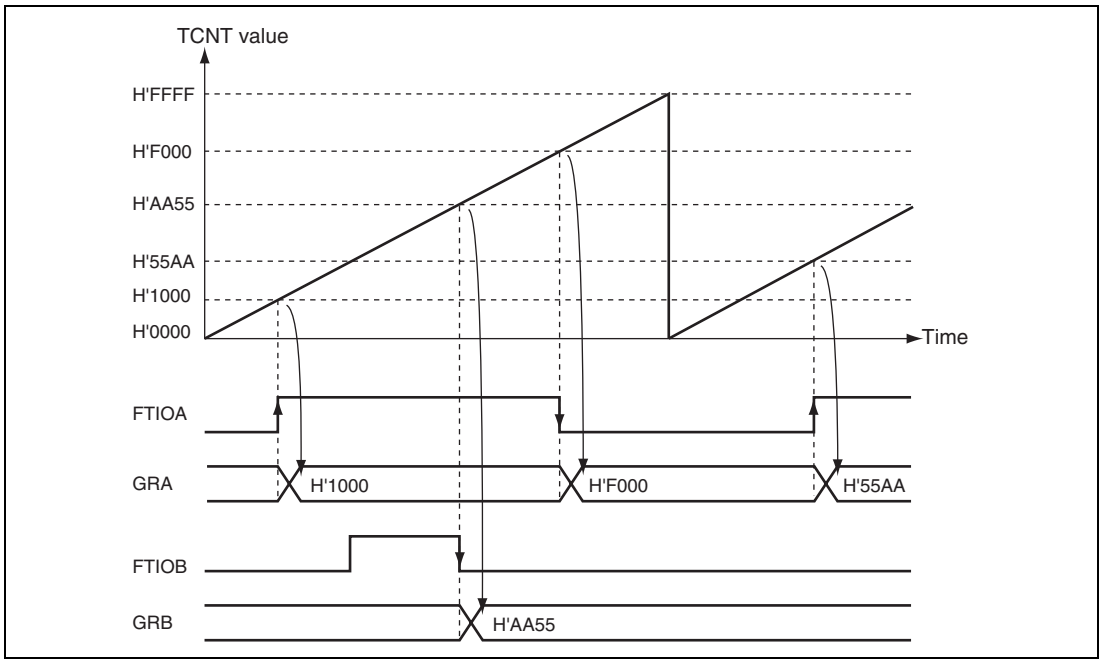


Figure 10.7 Input Capture Operating Example

Figure 10.8 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 edge of the input signal. Due to the buffer operation, the GRA value is transferred to GRC by input-capture A and the TCNT value is stored in GRA.

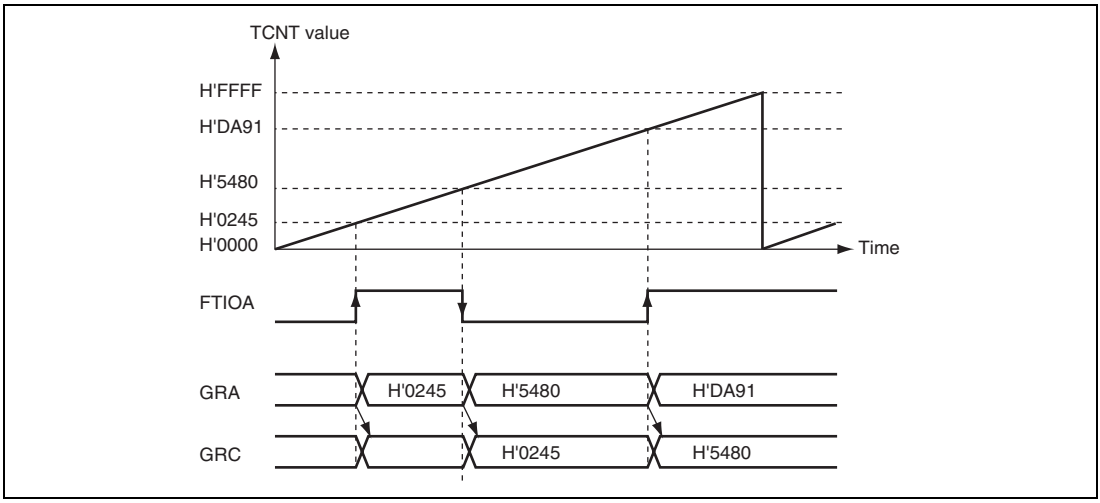


Figure 10.8 Buffer Operation Example (Input Capture)

10.4.2 PWM Operation

In PWM mode, PWM waveforms are generated by using GRA as the period register and GRB, GRC, and GRD as duty registers. PWM waveforms are output from the FTIOB, FTIOC, and FTIOD pins. Up to three-phase PWM waveforms can be output. In PWM mode, a general register functions as an output compare register automatically. The output level of each pin depends on the corresponding timer output level set bit (TOB, TOC, and TOD) in TCRW. When TOB is 1, the FTIOB output goes to 1 at compare match A and to 0 at compare match B. When TOB is 0, the FTIOB output goes to 0 at compare match A and to 1 at compare match B. Thus the compare match output level settings in TIOR0 and TIOR1 are ignored for the output pin set to PWM mode. If the same value is set in the cycle register and the duty register, the output does not change when a compare match occurs.

Figure 10.9 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).

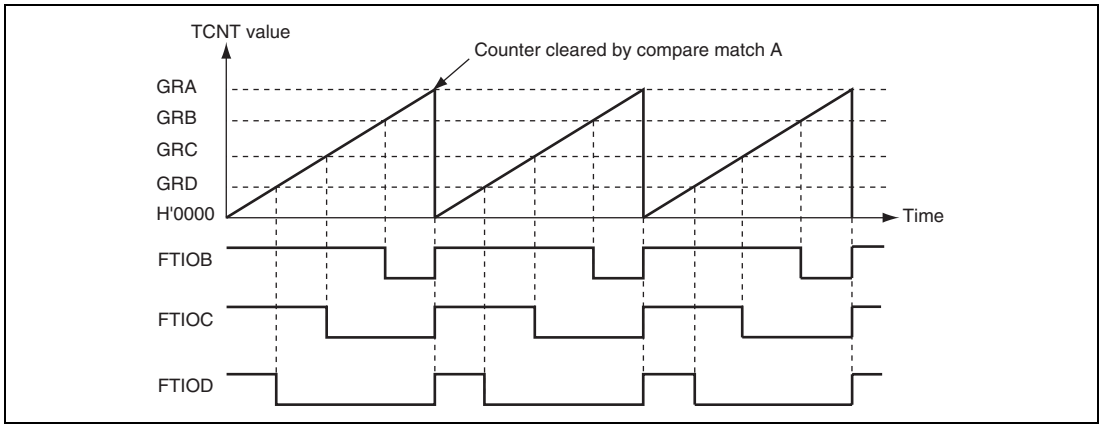


Figure 10.9 PWM Mode Example (1)

Figure 10.10 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 0).

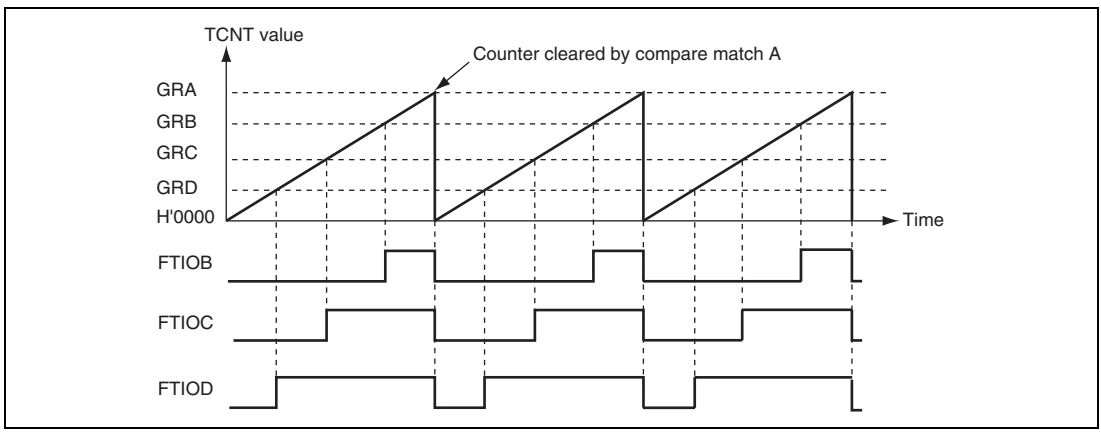


Figure 10.10 PWM Mode Example (2)

Figure 10.11 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.

Due to the buffer operation, the FTIOB output level changes and the value of buffer register GRD is transferred to GRB whenever compare match B occurs. This procedure is repeated every time compare match B occurs.

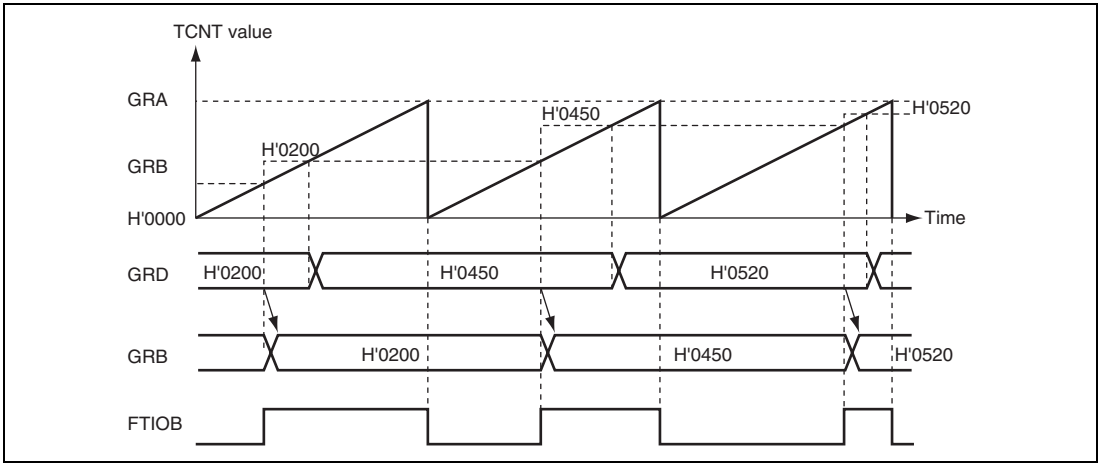


Figure 10.11 Buffer Operation Example (Output Compare)

Figures 10.12 and 10.13 show examples of the output of PWM waveforms with duty cycles of 0% and 100%.

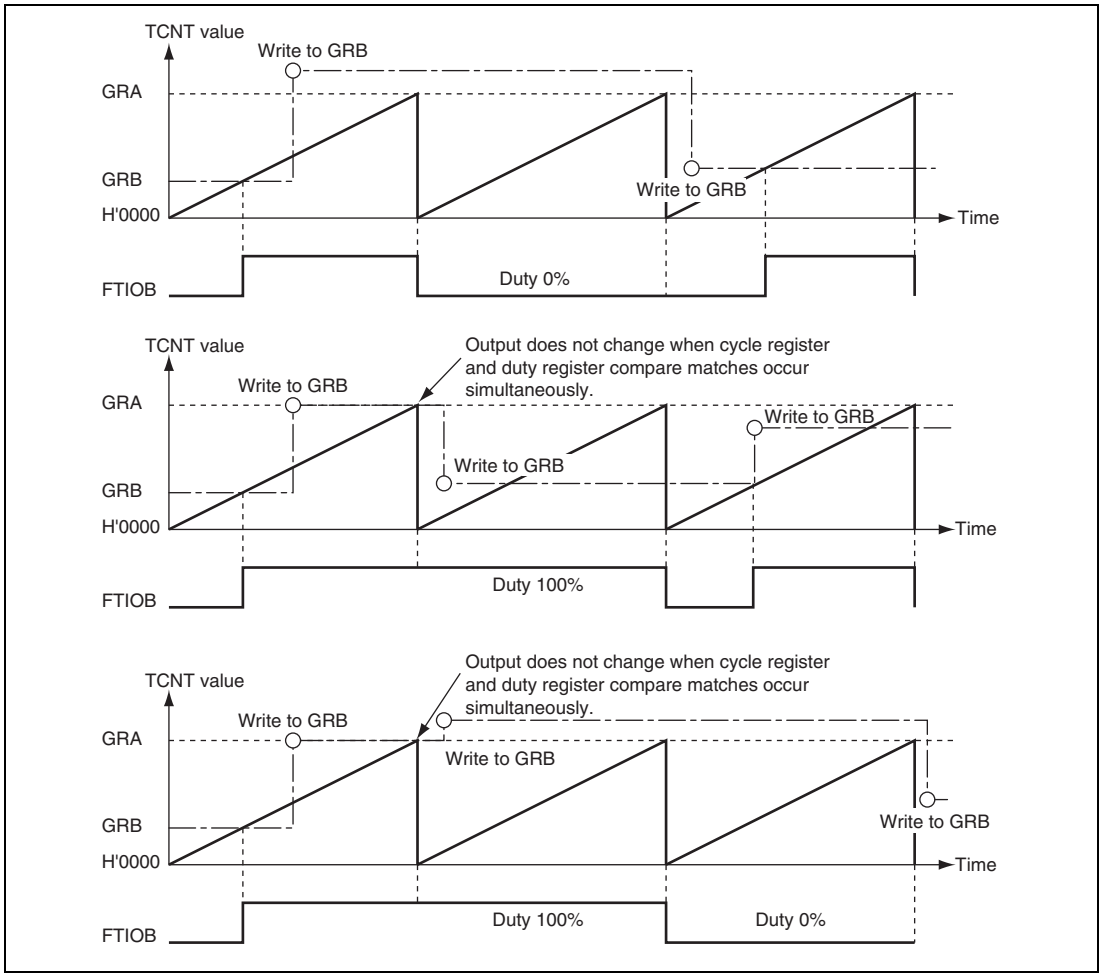


Figure 10.12 PWM Mode Example
(TOB, TOC, and TOD = 0: initial output values are set to 0)

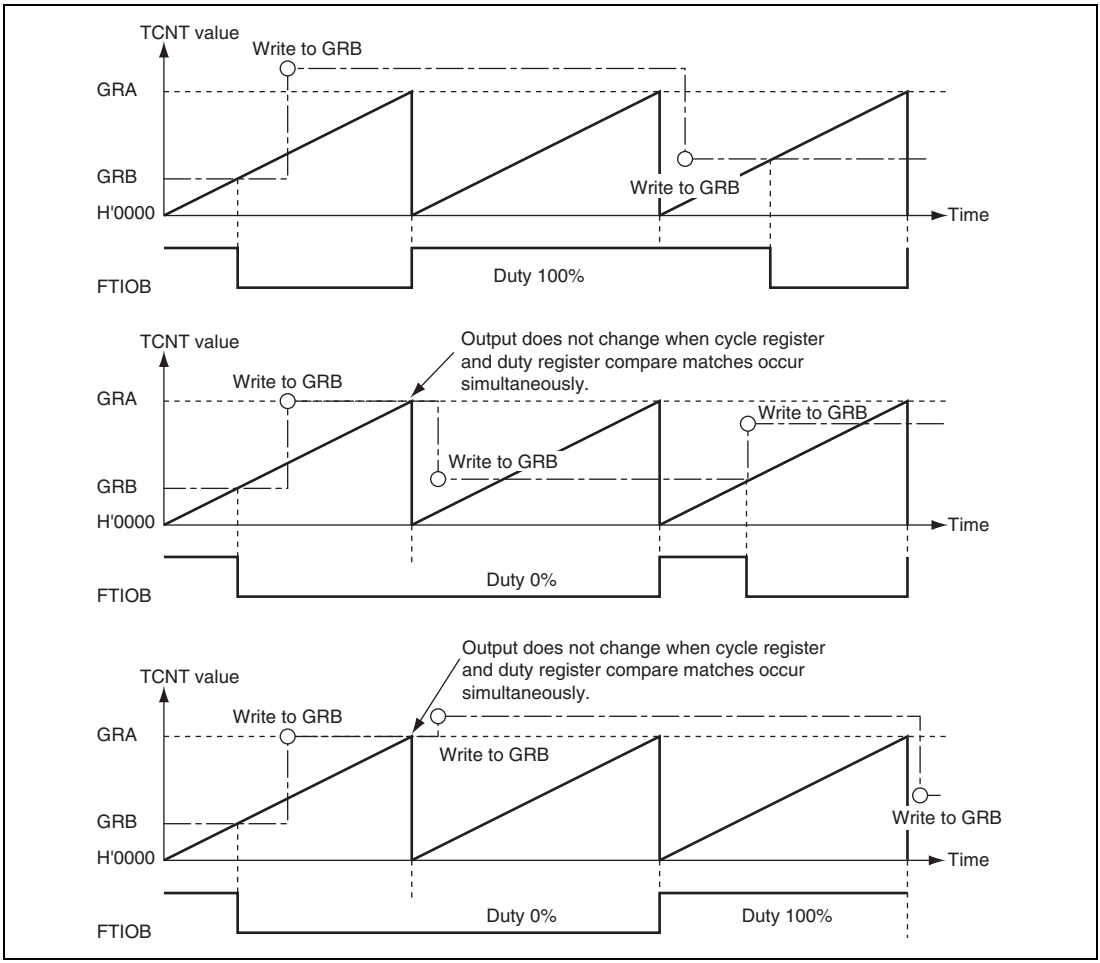


Figure 10.13 PWM Mode Example
(TOB, TOC, and TOD = 1: initial output values are set to 1)

10.5 Operation Timing

10.5.1 TCNT Count Timing

Figure 10.14 shows the TCNT count timing when the internal clock source is selected. Figure 10.15 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.

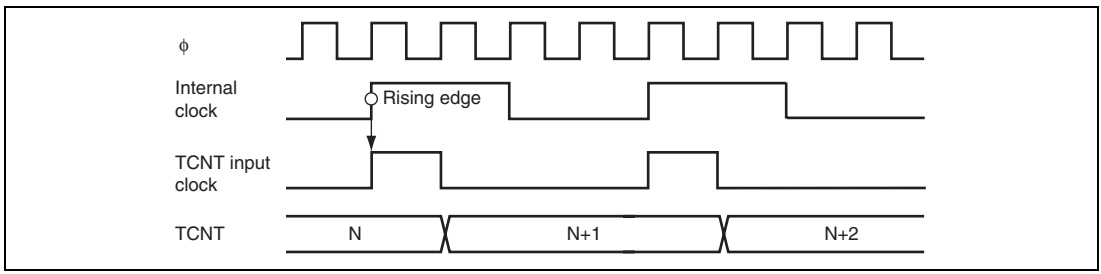


Figure 10.14 Count Timing for Internal Clock Source

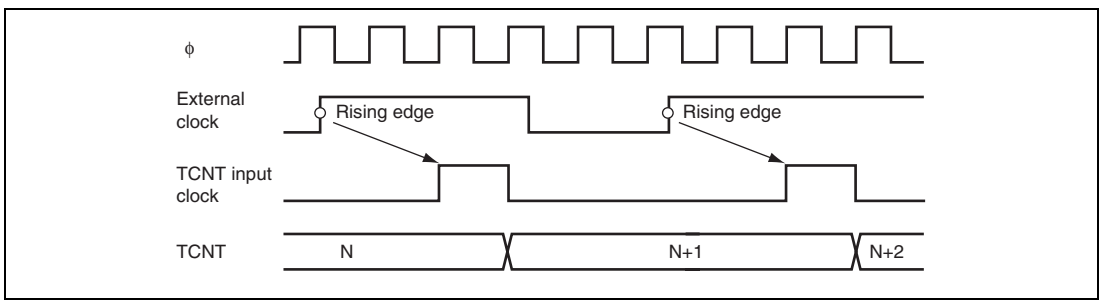


Figure 10.15 Count Timing for External Clock Source

10.5.2 Output Compare Output Timing

The compare match signal is generated in the last state in which TCNT and GR 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 GR, the compare match signal is generated only after the next counter clock pulse is input.

Figure 10.16 shows the output compare timing.

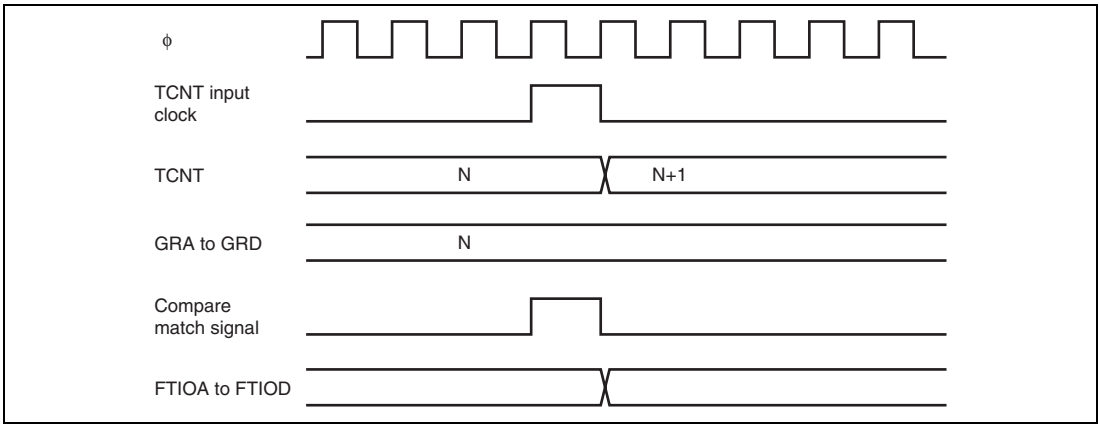


Figure 10.16 Output Compare Output Timing

10.5.3 Input Capture Timing

Input capture on the rising edge, falling edge, or both edges can be selected through settings in TIOR0 and TIOR1. Figure 10.17 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.

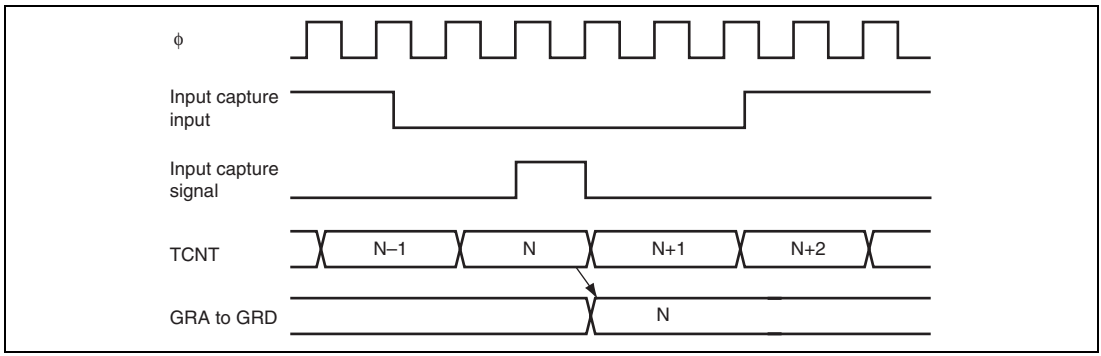


Figure 10.17 Input Capture Input Signal Timing

10.5.4 Timing of Counter Clearing by Compare Match

Figure 10.18 shows the timing when the counter is cleared by compare match A. When the GRA value is N, the counter counts from 0 to N, and its cycle is N + 1.

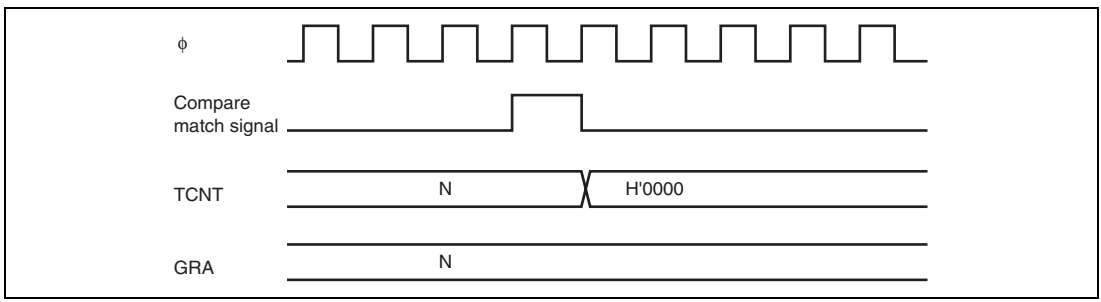


Figure 10.18 Timing of Counter Clearing by Compare Match

10.5.5 Buffer Operation Timing

Figures 10.19 and 10.20 show the buffer operation timing.

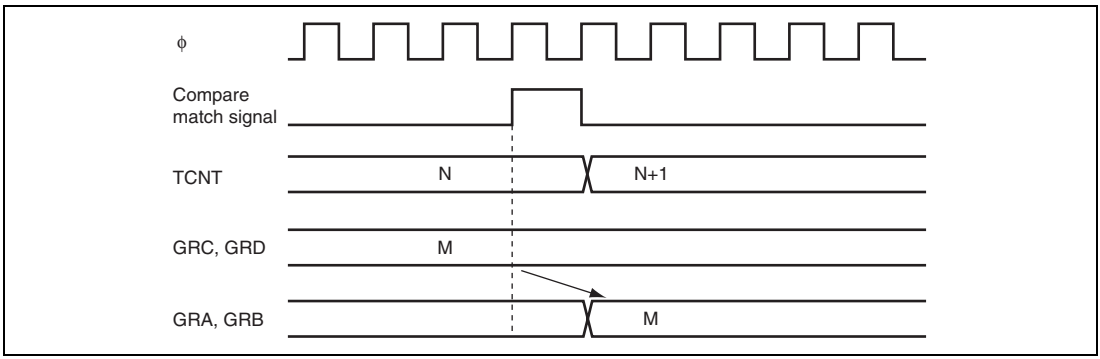


Figure 10.19 Buffer Operation Timing (Compare Match)

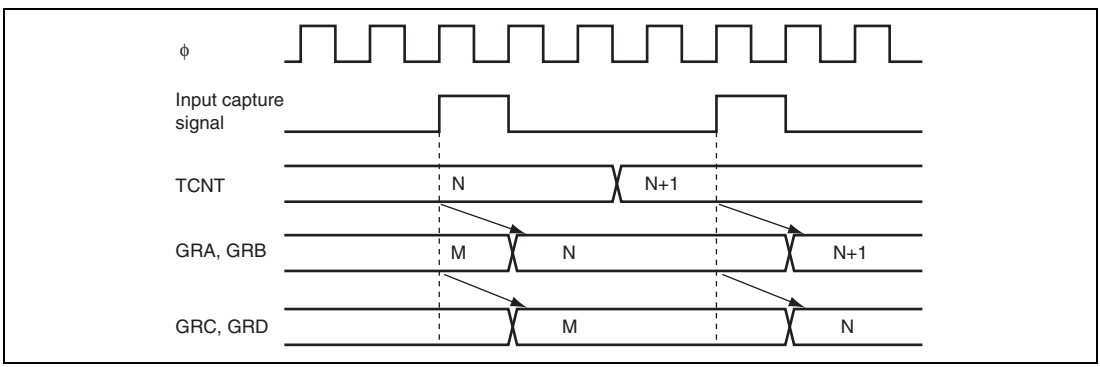


Figure 10.20 Buffer Operation Timing (Input Capture)

10.5.6 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 10.21 shows the timing of the IMFA to IMFD flag setting at compare match.

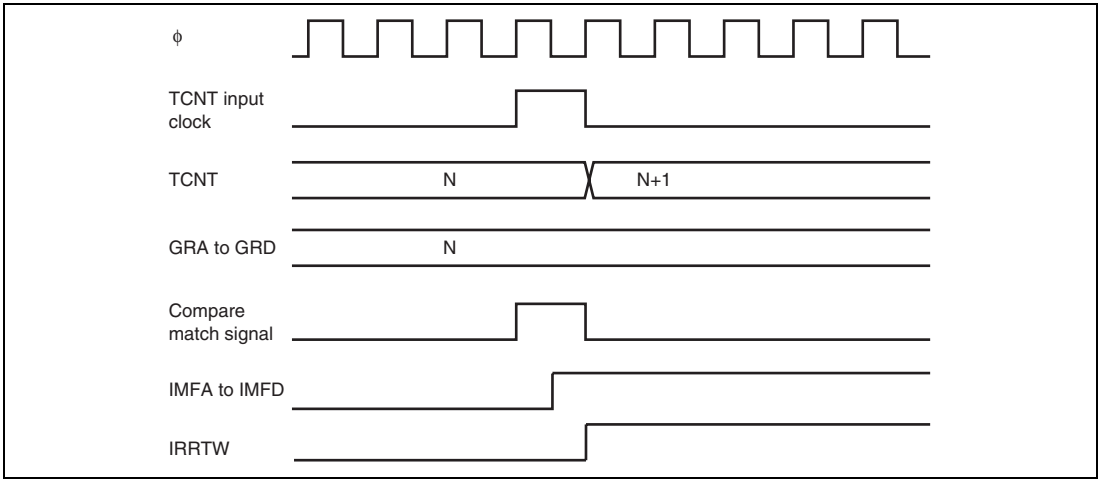


Figure 10.21 Timing of IMFA to IMFD Flag Setting at Compare Match

10.5.7 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 10.22 shows the timing of the IMFA to IMFD flag setting at input capture.

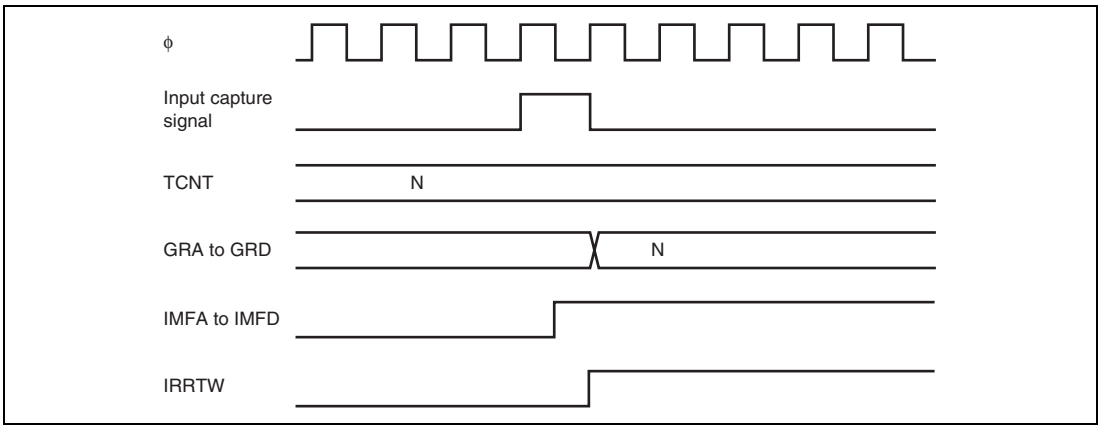


Figure 10.22 Timing of IMFA to IMFD Flag Setting at Input Capture

10.5.8 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 10.23 shows the status flag clearing timing.

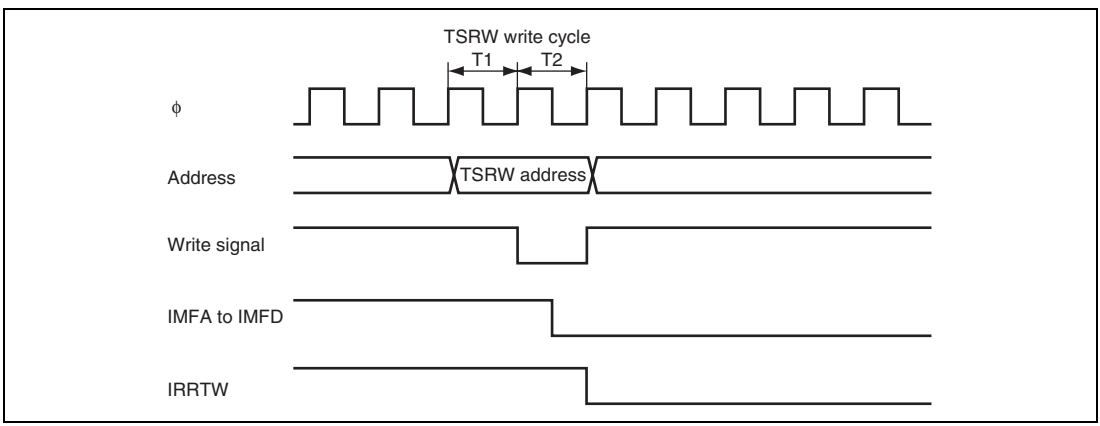


Figure 10.23 Timing of Status Flag Clearing by CPU

10.6 Timer W Operating Modes

Table 10.3 shows the timer W operating modes.

Table 10.3 Timer W Operating Modes

	Active		Sleep						Oscillation Stabilization Time		
Clock Source	High-speed	Medium-speed	High-speed	Medium-speed	Watch	Sub-active	Sub-sleep	Standby	Standby to Active	Subsleep to Active	Watch to Active
FTCI	o	o	o	o	×	o	o	×	×	×	×
ϕw, ϕw/4, ϕw/16	o	o	o	o	×	o	o	×	×	×	×
ϕ, ϕ/2, ϕ/4, ϕ/8	o	o	o	o	×	×	×	×	×	×	×

[Legend] o: Counting enabled
×: Counting disabled (Counter value retained)

10.7 Usage Notes

The following types of contention or operation can occur in timer W operation.

1. The pulse width of the input clock signal and the input capture signal must be at least two system clock cycles; shorter pulses will not be detected correctly. The system clock described here indicates the clock set for the CPU operation. For example, in the ϕw/8 operation, at least ϕw x 16 clock cycles are required as the pulse width.
2. Writing to registers is performed in the T2 state of a TCNT write cycle.

   If 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, as shown in figure 10.24. If counting-up is generated in the TCNT write cycle to contend with the TCNT counting-up, writing takes precedence.
3. Depending on the timing, TCNT may be incremented by a switch between different internal clock sources. When TCNT is internally clocked, an increment pulse is generated from the rising edge of an internal clock signal, that is, the divided system clock (ϕ). Therefore, as shown in figure 10.25 the switch is from a low clock signal to a high clock signal, the switchover is seen as a rising edge, causing TCNT to increment.
4. If timer W enters module standby mode while an interrupt request is generated, the interrupt request cannot be cleared. Before entering module standby mode, disable interrupt requests.

5. When an input capture function is specified, inputting a valid edge to the FTIOA to FTIOD pins sets the status bit of the corresponding TSRW, even if the CTS bit in TMRW is 0 (counting disabled state). When the relevant interrupt is enabled, this inputting generates an interrupt.
6. When the input capture timing conflicts with the corresponding GRA to GRD write timing,
   a. the written values are reflected in GRA to GRD.
   b. the status flag of the corresponding TSRW is set.
7. When the input capture timing conflicts with the GRA to GRD read timing, the read values are ones before capturing. The captured values can be read one clock after the capturing.
8. When the input capture A or B conflicts with the GRC or GRD write timing as the input capture operation in buffer mode,
   a. the captured values are reflected in GRA or GRB.
   b. the written values are reflected in GRC or GRD. (The values in GRC or GRD are not ones in GRA or GRB before capturing.)
9. When the compare match timing conflicts with the GRA to GRD write timing as the compare match operation,
   a. the written values are reflected in GRA to GRD.
   b. the FTIOA to FTIOD output changes by the compare match.
10. When the compare match A or B conflicts with the GRA or GRB write timing as the compare match operation in buffer mode,
    a. the written values are reflected in GRA or GRB. (The values in GRA or GRB are not ones in GRC or GRD of the buffer register.)
    b. the FTIOA or FTIOB output changes by the compare match.
11. When the compare match A or B conflicts with the GRC or GRD write timing as the compare match operation in buffer mode,
    a. the values in GRA or GRB are ones in GRC or GRD before writing.
    b. the FTIOA or FTIOB output changes by the compare match.
12. When GRC or GRD is specified to the compare match output as the compare match operation in buffer mode, FTIOC or FTIOD output changes by the GRC or GRD compare match.
13. When ϕw, $\phi w/4$, $\phi w/16$, or FTCI input is selected as the count clock, counting is enabled even in subactive and subsleep modes. Counting is disabled during the oscillation stabilization time in transition to the active mode.
14. When ϕw, $\phi w/4$, $\phi w/16$, or FTCI input is selected as the count clock, counting is enabled in active and sleep modes although counting may be misaligned by one in transition from the active to subactive mode.

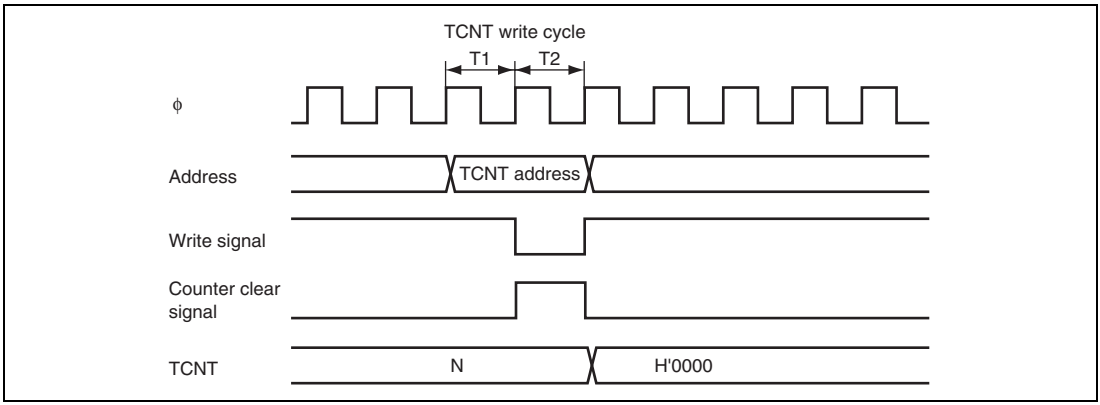


Figure 10.24 Contention between TCNT Write and Clear

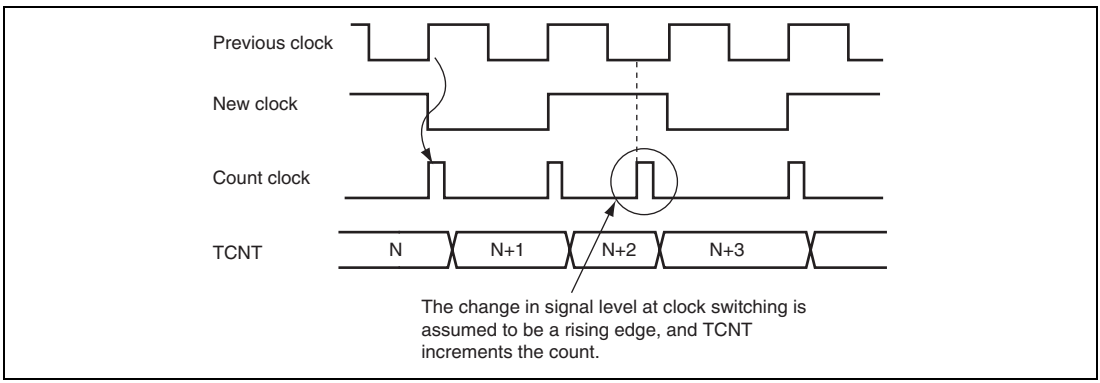


Figure 10.25 Internal Clock Switching and TCNT Operation

Section 11 Realtime Clock (RTC)

The realtime clock (RTC) is a timer used to count time ranging from a second to a week. Interrupts can be generated ranging from 0.25 seconds to a week. Figure 11.1 shows the block diagram of the RTC.

11.1 Features

- Counts seconds, minutes, hours, and day-of-week
- Start/stop function
- Reset function
- Readable/writable counter of seconds, minutes, hours, and day-of-week with BCD codes
- Periodic (0.25 seconds, 0.5 seconds, one second, minute, hour, day, and week) interrupts
- 8-bit free running counter
- Selection of clock source
- Use of module standby mode enables this module to be placed in standby mode independently when not used. (The RTC is operating as the initial value. For details, refer to section 5.4, Module Standby Function.)

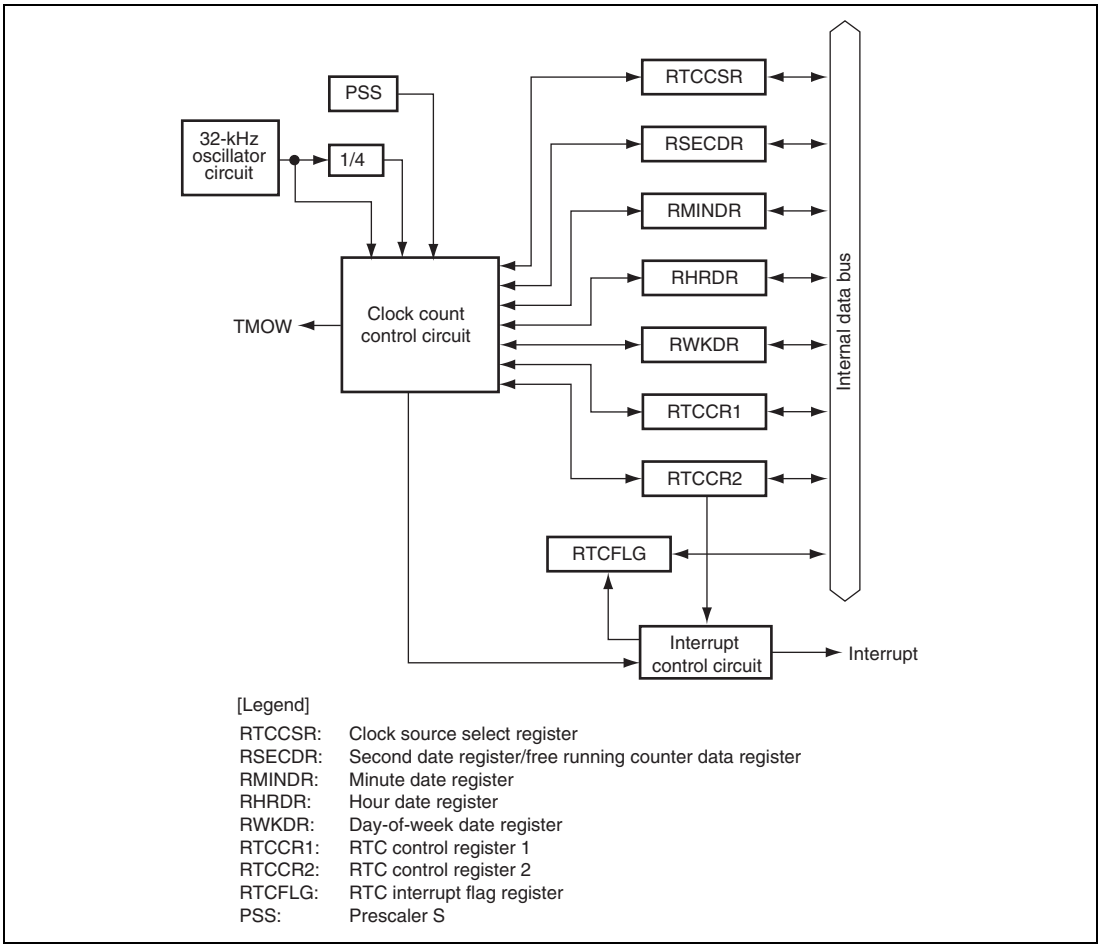


Figure 11.1 Block Diagram of RTC

11.2 Input/Output Pin

Table 11.1 shows the pin configuration of the RTC.

Table 11.1 Pin Configuration

Name	Abbreviation	I/O	Function
Clock output	TMOW	Output	RTC divided clock output

11.3 Register Descriptions

The RTC has the following registers.

- Second data register/free running counter data register (RSECDR)
- Minute data register (RMINDR)
- Hour data register (RHRDR)
- Day-of-week data register (RWKDR)
- RTC control register 1 (RTCCR1)
- RTC control register 2 (RTCCR2)
- Clock source select register (RTCCSR)
- RTC interrupt flag register (RTCFLG)

11.3.1 Second Data Register/Free Running Counter Data Register (RSECDR)

RSECDR counts the BCD-coded second value. The setting range is decimal 00 to 59. It is an 8-bit read register used as a counter, when it operates as a free running counter. For more information on reading seconds, minutes, hours, and day-of-week, see section 11.4.3, Data Reading Procedure.

Bit	Bit Name	Initial Value	R/W	Description
7	BSY	—/(0)*	R	RTC Busy This bit is set to 1 when the RTC is updating (operating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted.
6	SC12	—/(0)*	R/W	Counting Ten's Position of Seconds
5	SC11	—/(0)*	R/W	Counts on 0 to 5 for 60-second counting.
4	SC10	—/(0)*	R/W	
3	SC03	—/(0)*	R/W	Counting One's Position of Seconds
2	SC02	—/(0)*	R/W	Counts on 0 to 9 once per second. When a carry is generated, 1 is added to the ten's position.
1	SC01	—/(0)*	R/W	
0	SC00	—/(0)*	R/W	

Note: * Initial value after a reset caused by the RST bit in RTCCR1.

11.3.2 Minute Data Register (RMINDR)

RMINDR counts the BCD-coded minute value on the carry generated once per minute by the RSECDR counting. The setting range is decimal 00 to 59.

Bit	Bit Name	Initial Value	R/W	Description
7	BSY	—/(0)*	R	RTC Busy This bit is set to 1 when the RTC is updating (operating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted.
6	MN12	—/(0)*	R/W	Counting Ten's Position of Minutes
5	MN11	—/(0)*	R/W	Counts on 0 to 5 for 60-minute counting.
4	MN10	—/(0)*	R/W	
3	MN03	—/(0)*	R/W	Counting One's Position of Minutes
2	MN02	—/(0)*	R/W	Counts on 0 to 9 once per minute. When a carry is generated, 1 is added to the ten's position.
1	MN01	—/(0)*	R/W	
0	MN00	—/(0)*	R/W	

Note: * Initial value after a reset caused by the RST bit in RTCCR1.

11.3.3 Hour Data Register (RHRDR)

RHRDR counts the BCD-coded hour value on the carry generated once per hour by RMINDR. The setting range is either decimal 00 to 11 or 00 to 23 by the selection of the 12/24 bit in RTCCR1.

Bit	Bit Name	Initial Value	R/W	Description
7	BSY	—/(0)*	R	RTC Busy This bit is set to 1 when the RTC is updating (operating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted.
6	—	0	—	Reserved This bit is always read as 0.
5	HR11	—/(0)*	R/W	Counting Ten's Position of Hours
4	HR10	—/(0)*	R/W	Counts on 0 to 2 for ten's position of hours.
3	HR03	—/(0)*	R/W	Counting One's Position of Hours
2	HR02	—/(0)*	R/W	Counts on 0 to 9 once per hour. When a carry is generated, 1 is added to the ten's position.
1	HR01	—/(0)*	R/W	
0	HR00	—/(0)*	R/W	

Note: * Initial value after a reset caused by the RST bit in RTCCR1.

11.3.4 Day-of-Week Data Register (RWKDR)

RWKDR counts the BCD-coded day-of-week value on the carry generated once per day by RHRDR. The setting range is decimal 0 to 6 using bits WK2 to WK0.

Bit	Bit Name	Initial Value	R/W	Description
7	BSY	—/(0)*	R	RTC Busy This bit is set to 1 when the RTC is updating (operating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted.
6 to 3	—	All 0	—	Reserved These bits are always read as 0.
2	WK2	—/(0)*	R/W	Day-of-Week Counting
1	WK1	—/(0)*	R/W	Day-of-week is indicated with a binary code
0	WK0	—/(0)*	R/W	000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: Reserved (setting prohibited)

Note: * Initial value after a reset caused by the RST bit in RTCCR1.

11.3.5 RTC Control Register 1 (RTCCR1)

RTCCR1 controls start/stop and reset of the clock timer. For the definition of time expression, see figure 11.2.

Bit	Bit Name	Initial Value	R/W	Description
7	RUN	—/(0)*	R/W	RTC Operation Start 0: Stops RTC operation 1: Starts RTC operation
6	12/24	—/(0)*	R/W	Operating Mode 0: RTC operates in 12-hour mode. RHRDR counts on 0 to 11. 1: RTC operates in 24-hour mode. RHRDR counts on 0 to 23.
5	PM	—/(0)*	R/W	A.M./P.M. 0: Indicates a.m. when RTC is in the 12-hour mode. 1: Indicates p.m. when RTC is in the 12-hour mode.
4	RST	0	R/W	Reset 0: Normal operation 1: Resets registers and control circuits except RTCCSR and this bit. Clear this bit to 0 after having been set to 1.
3	INT	—/(0)*	R/W	Interrupt Occurrence Timing 0: Periodic interrupts of second, minute, hour, and day-of-week occur during the RTC busy period. 1: Periodic interrupts of second, minute, hour, and day-of-week occur immediately after the RTC busy period finishes.
2 to 0	—	All 0	—	Reserved These bits are always read as 0.

Note: * Initial value after a reset caused by the RST bit in RTCCR1.

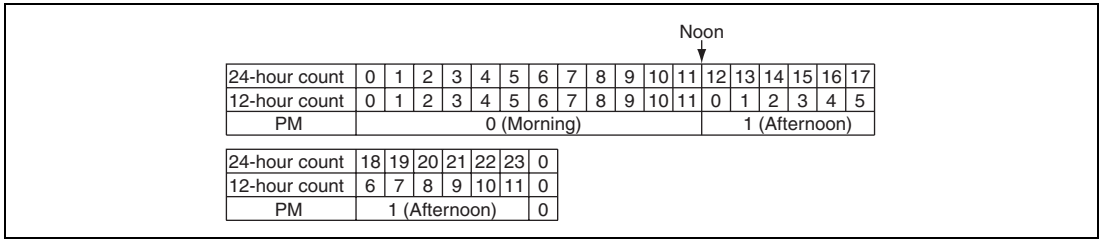


Figure 11.2 Definition of Time Expression

11.3.6 RTC Control Register 2 (RTCCR2)

RTCCR2 controls RTC periodic interrupts of week, day, hour, minute, one second, 0.5 seconds, and 0.25 seconds. Enabling interrupts of week, day, hour, minute, one second, 0.5 seconds, and 0.25 seconds sets the corresponding flag to 1 in the RTC interrupt flag register (RTCFLG) when an interrupt occurs. It also controls an overflow interrupt of a free running counter when RTC operates as a free running counter.

Bit	Bit Name	Initial Value	R/W	Description
7	FOIE	—/(0)*	R/W	Free Running Counter Overflow Interrupt Enable 0: Disables an overflow interrupt 1: Enables an overflow interrupt
6	WKIE	—/(0)*	R/W	Week Periodic Interrupt Enable 0: Disables a week periodic interrupt 1: Enables a week periodic interrupt
5	DYIE	—/(0)*	R/W	Day Periodic Interrupt Enable 0: Disables a day periodic interrupt 1: Enables a day periodic interrupt
4	HRIE	—/(0)*	R/W	Hour Periodic Interrupt Enable 0: Disables an hour periodic interrupt 1: Enables an hour periodic interrupt
3	MNIE	—/(0)*	R/W	Minute Periodic Interrupt Enable 0: Disables a minute periodic interrupt 1: Enables a minute periodic interrupt
2	1SEIE	—/(0)*	R/W	One-Second Periodic Interrupt Enable 0: Disables a one-second periodic interrupt 1: Enables a one-second periodic interrupt
1	05SEIE	—/(0)*	R/W	0.5-Second Periodic Interrupt Enable 0: Disables a 0.5-second periodic interrupt 1: Enables a 0.5-second periodic interrupt
0	025SEIE	—/(0)*	R/W	0.25-Second Periodic Interrupt Enable 0: Disables a 0.25-second periodic interrupt 1: Enables a 0.25-second periodic interrupt

Note: * Initial value after a reset caused by the RST bit in RTCCR1.

11.3.7 Clock Source Select Register (RTCCSR)

RTCCSR selects clock source. A free running counter controls start/stop of counter operation by the RUN bit in RTCCR1. When a clock other than $\phi_w/4$ is selected, the RTC is disabled and operates as an 8-bit free running counter. When the RTC operates as an 8-bit free running counter, RSECDR enables counter values to be read. An interrupt can be generated by setting 1 to the FOIE bit in RTCCR2 and enabling an overflow interrupt of the free running counter. A clock generated by dividing the system clock by 32, 16, 8, or 4 is output in active or sleep mode. ϕ_w is output in active, sleep, subactive, subsleep, or watch mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.
6	RCS6	0	R/W	Clock Output Selection
5	RCS5	0	R/W	Select a clock output from the TMOW pin when enabling TMOW output in PMR1.
4	SUB32K	0	R/W	000: $\phi/4$ 010: $\phi/8$ 100: $\phi/16$ 110: $\phi/32$ xx1: ϕ_w
3	RCS3	1	R/W	Clock Source Selection
2	RCS2	0	R/W	0000: $\phi/8$ ····· Free running counter operation
1	RCS1	0	R/W	0001: $\phi/32$ ····· Free running counter operation
0	RCS0	0	R/W	0010: $\phi/128$ ····· Free running counter operation 0011: $\phi/256$ ····· Free running counter operation 0100: $\phi/512$ ····· Free running counter operation 0101: $\phi/2048$ ····· Free running counter operation 0110: $\phi/4096$ ····· Free running counter operation 0111: $\phi/8192$ ····· Free running counter operation 1000: $\phi_w/4$ ····· RTC operation 1001 to 1111: Setting prohibited

11.3.8 RTC Interrupt Flag Register (RTCFLG)

RTCFLG sets the corresponding flag when an interrupt occurs. Each flag is not cleared automatically even if the interrupt is accepted. To clear the flag, 0 should be written to the flag.

Bit	Bit Name	Initial Value	R/W	Description
7	FOIFG	—/(0)*[1]	R/(W)*[2]	[Setting condition] When a free running counter overflows [Clearing condition] 0 is written to FOIFG when FOIFG = 1
6	WKIFG	—/(0)*[1]	R/(W)*[2]	[Setting condition] When a week periodic interrupt occurs [Clearing condition] 0 is written to WKIFG when WKIFG = 1
5	DYIFG	—/(0)*[1]	R/(W)*[2]	[Setting condition] When a day periodic interrupt occurs [Clearing condition] 0 is written to DYIFG when DYIFG = 1
4	HRIFG	—/(0)*[1]	R/(W)*[2]	[Setting condition] When an hour periodic interrupt occurs [Clearing condition] 0 is written to HRIFG when HRIFG = 1
3	MNIFG	—/(0)*[1]	R/(W)*[2]	[Setting condition] When a minute periodic interrupt occurs [Clearing condition] 0 is written to MNIFG when MNIFG = 1
2	1SEIFG	—/(0)*[1]	R/(W)*[2]	[Setting condition] When a one-second periodic interrupt occurs [Clearing condition] 0 is written to 1SEIFG when 1SEIFG = 1

Bit	Bit Name	Initial Value	R/W	Description
1	05SEIFG	—/(0)*[1]	R/(W)*[2]	[Setting condition] When a 0.5-second periodic interrupt occurs [Clearing condition] 0 is written to 05SEIFG when 05SEIFG = 1
0	025SEIFG	—/(0)*[1]	R/(W)*[2]	[Setting condition] When a 0.25-second periodic interrupt occurs [Clearing condition] 0 is written to 025SEIFG when 025SEIFG = 1

Notes: 1. Initial value after a reset caused by the RST bit in RTCCR1.
2. Only 0 can be written to clear the flag.

11.4 Operation

11.4.1 Initial Settings of Registers after Power-On

The RTC registers that store second, minute, hour, and day-of-week data, control registers, and interrupt registers are not reset by a $\overline{\text{RES}}$ input, or by a reset source caused by a watchdog timer. Therefore, all registers must be set to their initial values after power-on. Once the register setting are made, the RTC provides an accurate time as long as power is supplied regardless of a $\overline{\text{RES}}$ input.

11.4.2 Initial Setting Procedure

Figure 11.3 shows the procedure for the initial setting of the RTC. To set the RTC again, also follow this procedure.

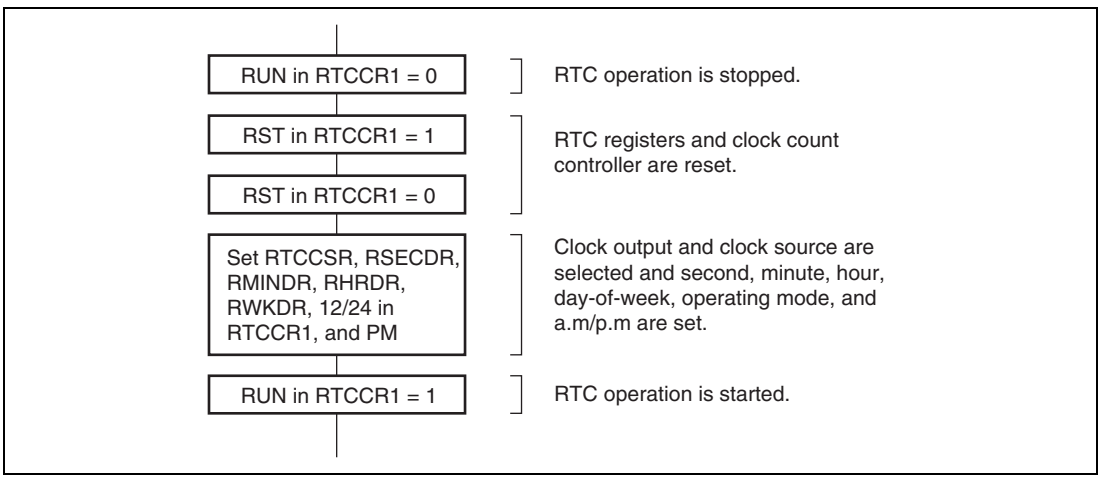


Figure 11.3 Initial Setting Procedure

11.4.3 Data Reading Procedure

When the seconds, minutes, hours, or day-of-week datum is updated while time data is being read, the data obtained may not be correct, and so the time data must be read again. Figure 11.4 shows an example in which correct data is not obtained. In this example, since only RSECDR is read after data update, about 1-minute inconsistency occurs.

To avoid reading in this timing, the following processing must be performed.

1. Check the setting of the BSY bit, and when the BSY bit changes from 1 to 0, read from the second, minute, hour, and day-of-week registers. When about 62.5 ms is passed after the BSY bit is set to 1, the registers are updated, and the BSY bit is cleared to 0.
2. When INT in RTCCR1 is cleared to 0 and an interrupt is used, read from the second, minute, hour, and day-of-week registers after the relevant flag in RTCFLG is set to 1 and the BSY bit is confirmed to be 0.

   When INT in RTCCR1 is set to 1 and an interrupt is used, read from the second, minute, hour, and day-of-week registers after the relevant flag in RTCFLG is set to 1.
3. Read from the second, minute, hour, and day-of-week registers twice in a row, and if there is no change in the read data, the read data is used.

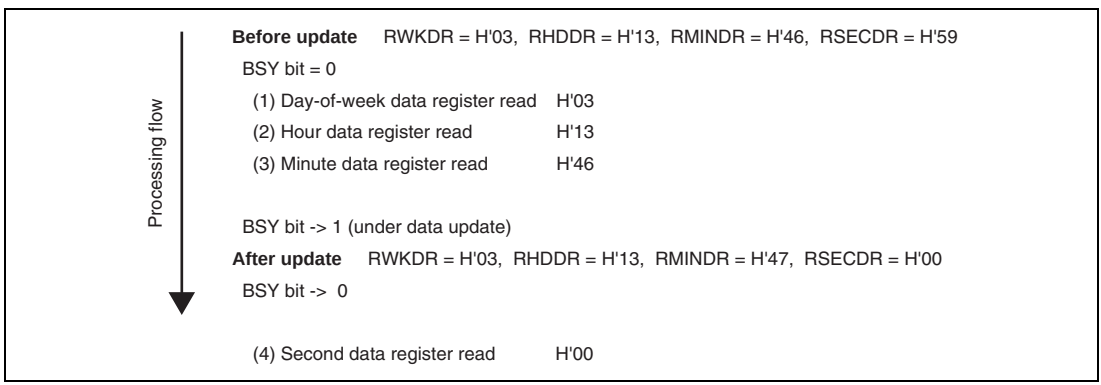


Figure 11.4 Example: Reading of Inaccurate Time Data

11.5 Interrupt Sources

There are eight kinds of RTC interrupts: a free-running counter overflow, week interrupt, day interrupt, hour interrupt, minute interrupt, one-second interrupt, 0.5-second interrupt, and 0.25-second interrupt.

When using an interrupt, set the IENRTC (RTC interrupt request enable) bit in IENR1 to 1 last after other registers are set.

When an interrupt request of the RTC occurs, the corresponding flag in RTCFLG is set to 1. When clearing the flag, write 0.

Table 11.2 Interrupt Sources

Interrupt Name	Interrupt Source	Interrupt Enable Bit
Overflow interrupt	Occurs when the free running counter is overflowed.	FOIE
Week periodic interrupt	Occurs every week when the day-of-week date register value becomes 0.	WKIE
Day periodic interrupt	Occurs every day when the day-of-week date register is counted.	DYIE
Hour periodic interrupt	Occurs every hour when the hour date register is counted.	HRIE
Minute periodic interrupt	Occurs every minute when the minute date register is counted.	MNIE
One-second periodic interrupt	Occurs every second when the one-second date register is counted.	1SEIE
0.5-second periodic interrupt	Occurs every 0.5 seconds.	05SEIE
0.25-second periodic interrupt	Occurs every 0.25 seconds.	025SEIE

11.6 Usage Notes

11.6.1 Note on Clock Count

The subclock must be connected to the 32.768-kHz resonator. When the 38.4-kHz resonator etc. is connected, the correct time count is not possible.

11.6.2 Note when Using RTC Interrupts

The RTC registers are not reset by a $\overline{\text{RES}}$ input, power-on, or overflow of the watchdog timer, and their values are undefined after power-on.

When using RTC interrupts, make sure to initialize the values before setting the IENRTC bit in IENR1 to 1.

Section 12 Watchdog Timer

This LSI incorporates the watchdog timer (WDT). The WDT is an 8-bit timer that can generate an internal reset signal if a system crash prevents the CPU from writing to the timer counter, thus allowing it to overflow.

When this watchdog timer function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is generated each time the counter overflows.

12.1 Features

The WDT features are described below.

- Selectable from eleven counter input clocks
  Ten internal clock sources (ϕ/64, ϕ/128, ϕ/256, ϕ/512, ϕ/1024, ϕ/2048, ϕ/4096, ϕ/8192, ϕ_w/16, and ϕ_w/256) or the on-chip oscillator (R_{OSC}/2048) can be selected as the timer-counter clock.
- Watchdog timer mode
  If the counter overflows, this LSI is internally reset.
- Interval timer mode
  If the counter overflows, an interval timer interrupt is generated.
- Use of module standby mode enables this module to be placed in standby mode independently when not used. (The WDT is operating as the initial value. For details, refer to section 5.4, Module Standby Function.)

Figure 12.1 shows a block diagram of the WDT.

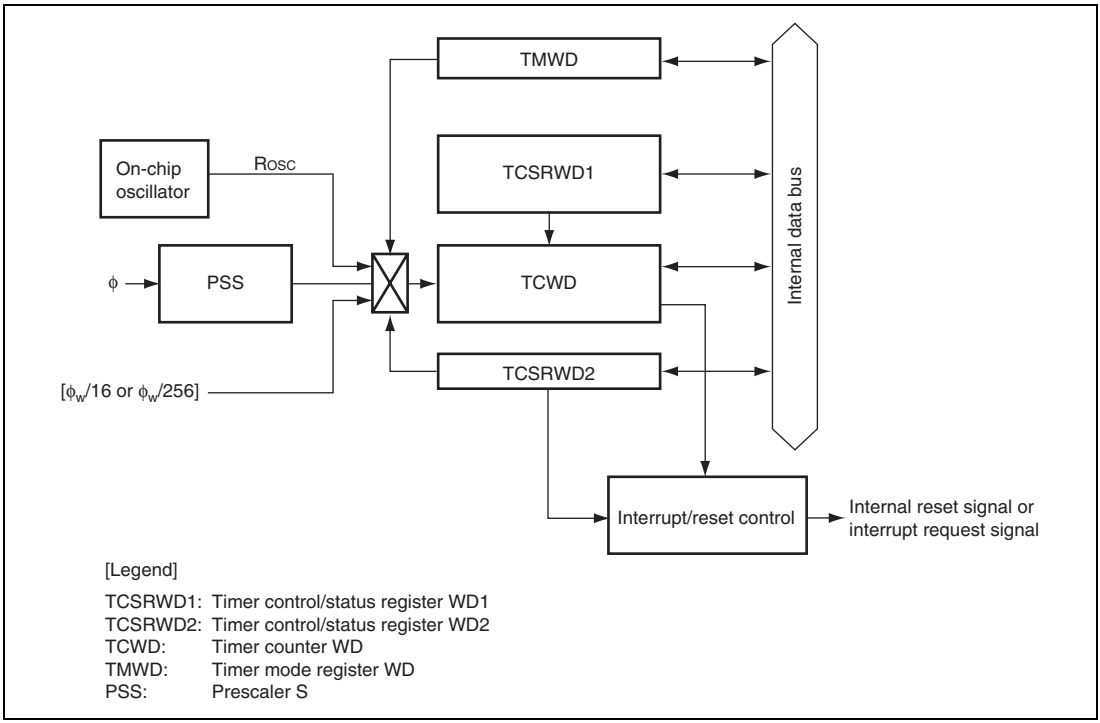


Figure 12.1 Block Diagram of Watchdog Timer

12.2 Register Descriptions

The watchdog timer has the following registers.

- Timer control/status register WD1 (TCSRWD1)
- Timer control/status register WD2 (TCSRWD2)
- Timer counter WD (TCWD)
- Timer mode register WD (TMWD)

12.2.1 Timer Control/Status Register WD1 (TCSRWD1)

TCSRWD1 performs the TCSRWD1 and TCWD write control. TCSRWD1 also controls the watchdog timer operation and indicates the operating state. TCSRWD1 must be rewritten by using the MOV instruction. The bit manipulation instruction cannot be used to change the setting value.

Bit	Bit Name	Initial Value	R/W	Description
7	B6WI	1	R/W	Bit 6 Write Inhibit The TCWE bit can be written only when the write value of the B6WI bit is 0. This bit is always read as 1.
6	TCWE	0	R/W	Timer Counter WD Write Enable TCWD can be written when the TCWE bit is set to 1. When writing data to this bit, the write value for bit 7 must be 0.
5	B4WI	1	R/W	Bit 4 Write Inhibit The TCSRWE bit can be written only when the write value of the B4WI bit is 0. This bit is always read as 1.
4	TCSRWE	0	R/W	Timer Control/Status Register WD Write Enable The WDON and WRST bits can be written when the TCSRWE bit is set to 1. When writing data to this bit, the write value for bit 5 must be 0.
3	B2WI	1	R/W	Bit 2 Write Inhibit The WDON bit can be written only when the write value of the B2WI bit is 0. This bit is always read as 1.

Bit	Bit Name	Initial Value	R/W	Description
2	WDON	1	R/W	Watchdog Timer On TCWD starts counting up when the WDON bit is set to 1 and halts when the WDON bit is cleared to 0. [Setting condition] • When 1 is written to the WDON bit and 0 to the B2WI bit while the TCSRWE bit is 1 • Reset by $\overline{\text{RES}}$ pin [Clearing conditions] • When 0 is written to the WDON bit and 0 to the B2WI bit while the TCSRWE bit is 1
1	B0WI	1	R/W	Bit 0 Write Inhibit The WRST bit can be written only when the write value of the B0WI bit is 0. This bit is always read as 1.
0	WRST	0	R/W	Watchdog Timer Reset Indicates whether a reset caused by the watchdog timer is generated. This bit is not cleared by a reset caused by the watchdog timer. [Setting condition] When TCWD overflows and an internal reset signal is generated [Clearing conditions] • Reset by $\overline{\text{RES}}$ pin • When 0 is written to the WRST bit and 0 to the B0WI bit while the TCSRWE bit is 1

12.2.2 Timer Control/Status Register WD2 (TCSRWD2)

TCSRWD2 performs the TCSRWD2 write control, mode switching, and interrupt control. TCSRWD2 must be rewritten by using the MOV instruction. The bit manipulation instruction cannot be used to change the setting value.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*[1]	Overflow Flag Indicates that TCWD has overflowed (changes from H'FF to H'00). [Setting condition] When TCWD overflows (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, this bit is cleared automatically by the internal reset after it has been set. [Clearing condition] • When TCSRWD2 is read when OVF = 1, then 0 is written to OVF
6	B5WI	1	R/(W)*[2]	Bit 5 Write Inhibit The WT/$\overline{IT}$ bit can be written only when the write value of the B5WI bit is 0. This bit is always read as 1.
5	WT/$\overline{IT}$	0	R/(W)*[3]	Timer Mode Select Selects whether the WDT is used as a watchdog timer or interval timer. 0: Watchdog timer mode 1: Interval timer mode
4	B3WI	1	R/(W)*[2]	Bit 3 Write Inhibit The IEOVF bit can be written only when the write value of the B3WI bit is 0. This bit is always read as 1.
3	IEOVF	0	R/(W)*[3]	Overflow Interrupt Enable Enables or disables an overflow interrupt request in interval timer mode. 0: Disables an overflow interrupt 1: Enables an overflow interrupt

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	—	All 1	—	Reserved These bits are always read as 1.

Notes: 1. Only 0 can be written to clear the flag.
2. Write operation is necessary because this bit controls data writing to other bit. This bit is always read as 1.
3. Writing is possible only when the write conditions are satisfied.

12.2.3 Timer Counter WD (TCWD)

TCWD is an 8-bit readable/writable up-counter. When TCWD overflows from H'FF to H'00, the internal reset signal is generated and the WRST bit in TCSRWD1 is set to 1. TCWD is initialized to H'00.

12.2.4 Timer Mode Register WD (TMWD)

TMWD selects the input clock.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	CKS3	0	R/W	Clock Select 3 to 0
2	CKS2	0	R/W	Select the clock to be input to TCWD.
1	CKS1	0	R/W	00xx: On-chip oscillator: counts on R_{osc}/2048
0	CKS0	0	R/W	0100: Internal clock: counts on ϕ_w/16 0101: Internal clock: counts on ϕ_w/256 011x: Reserved 1000: Internal clock: counts on ϕ/64 1001: Internal clock: counts on ϕ/128 1010: Internal clock: counts on ϕ/256 1011: Internal clock: counts on ϕ/512 1100: Internal clock: counts on ϕ/1024 1101: Internal clock: counts on ϕ/2048 1110: Internal clock: counts on ϕ/4096 1111: Internal clock: counts on ϕ8192 For the on-chip oscillator overflow periods, see section 21, Electrical Characteristics. In active (medium-speed), sleep (medium-speed), subactive, and subsleep modes, the 00xx value and the interval timer mode cannot be set simultaneously. In subactive and subsleep modes, when the subclock frequency is ϕ_w/8, the 010x value and the interval timer mode cannot be set simultaneously.

[Legend] x: Don't care.

12.3 Operation

12.3.1 Watchdog Timer Mode

The watchdog timer is provided with an 8-bit up-counter. To use it as the watchdog timer, clear the WT/$\overline{IT}$ bit in TCSRWD2 to 0. (To write the WT/$\overline{IT}$ bit, two write accesses are required.) If 1 is written to the WDON bit and 0 to the B2WI bit simultaneously when the TCSRWE bit in TCSRWD1 is set to 1, TCWD begins counting up. (To operate the watchdog timer, two write accesses to TCSRWD1 are 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. The internal reset signal is output for a period of 512 clock cycles by the on-chip oscillator (R_{OSC}). 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 set value.

Figure 12.2 shows an example of watchdog timer operation.

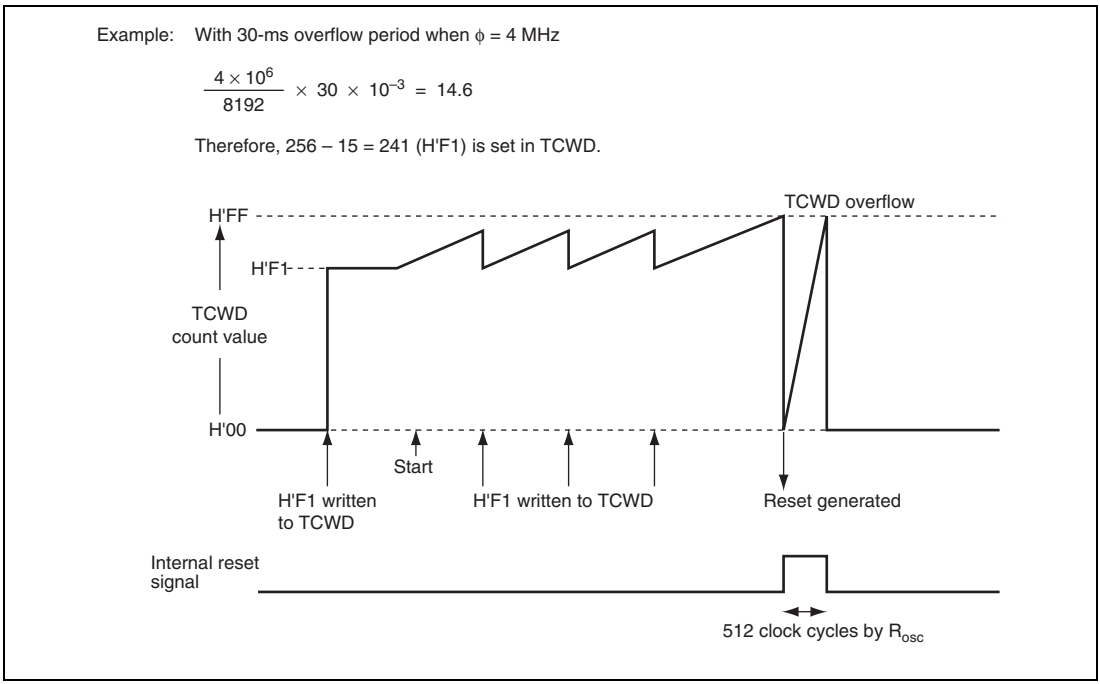


Figure 12.2 Example of Watchdog Timer Operation

12.3.2 Interval Timer Mode

Figure 12.3 shows the operation in interval timer mode. To use the WDT as an interval timer, set the WT/$\overline{IT}$ bit in TCSRWD2 to 1.

When the WDT is used as an interval timer, an interval timer interrupt request is generated each time the TCWD overflows. Therefore, an interval timer interrupt can be generated at intervals.

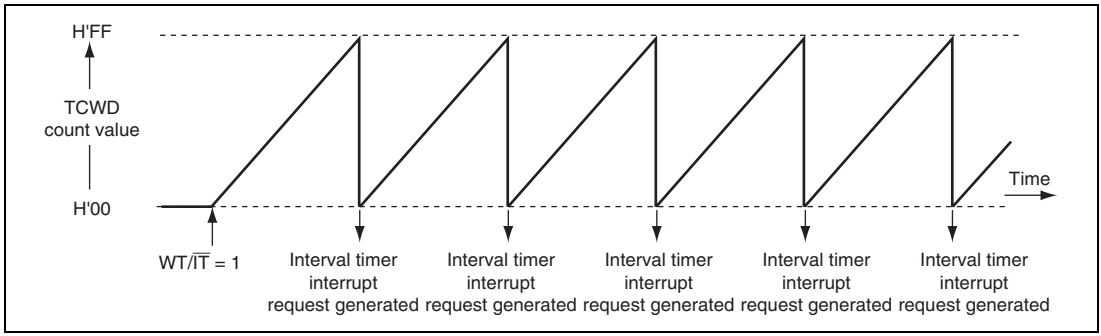


Figure 12.3 Interval Timer Mode Operation

12.3.3 Timing of Overflow Flag (OVF) Setting

Figure 12.4 shows the timing of the OVF flag setting. The OVF flag in TCSRWD2 is set to 1 if TCWD overflows. At the same time, a reset signal is output in watchdog timer mode and an interval timer interrupt is generated in interval timer mode.

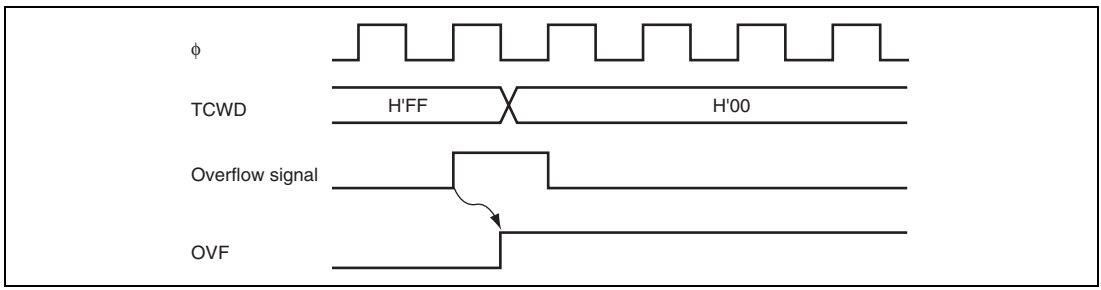


Figure 12.4 Timing of OVF Flag Setting

12.4 Interrupt

During interval timer mode operation, an overflow generates an interval timer interrupt. The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSRWD2. The OVF flag must be cleared to 0 in the interrupt handling routine.

12.5 Usage Notes

12.5.1 Switching between Watchdog Timer Mode and Interval Timer Mode

If modes are switched between watchdog timer and interval timer, while the WDT is operating, an error may occur in the count value. Software must stop the watchdog timer (by clearing the WDON bit to 0) before switching modes.

12.5.2 Module Standby Mode Control

The WDCKSTP bit in CKSTPR2 is valid when the WDON bit in the timer control/status register WD1 (TCSRWD1) is cleared to 0. The WDCKSTP bit can be cleared to 0 while the WDON bit is set to 1 (while the watchdog timer is operating). However, the watchdog timer does not enter module standby mode but continues operating. When the WDON bit is cleared to 0 by software after the watchdog timer stops operating, the WDCKSTP bit is valid at the same time and the watchdog timer enters module standby mode.

12.5.3 Clearing the WT/$\overline{\text{IT}}$ or IEOVF Bit in TCSRWD2 to 0

When clearing the WT/$\overline{\text{IT}}$ or IEOVF bit in the timer control/status register WD2 (TCSRWD2) to 0, the corresponding bit may not be cleared to 0 depending on the program address. In particular, if lower two bits in the address of the MOV.B instruction to transfer a value to TCSRWD2 are B'10, the WT/$\overline{\text{IT}}$ or IEOVF bit is successfully cleared to 0, whereas if lower two bits in the address are B'00, the WT/$\overline{\text{IT}}$ or IEOVF bit may not be cleared to 0. To avoid this failure, make sure to use the assembly program shown in table 12.1, when clearing the WT/$\overline{\text{IT}}$ or IEOVF bit to 0. Specify TCSRWD2 by the 8-bit absolute address, and LABEL by the 16-bit absolute address. Don't change nor add instructions. The value of "xx" in line 1 and line 4 must be set according to table 12.2. Use an arbitrary 8-bit general register for Rn and Rm. In addition, Address1 in table 12.1 shows an example when the WT/$\overline{\text{IT}}$ or IEOVF bit is cleared to 0 successfully by the MOV.B instruction in line 2. Address2 in table 12.1 shows an example when the WT/$\overline{\text{IT}}$ or IEOVF bit fails to be cleared to 0 by the MOV.B instruction in line 2, but cleared to 0 by the MOV.B instruction in line 6.

Table 12.1 Assembly Program for Clearing WT/$\overline{IT}$ or IEOVF Bit to 0

Address1	Address2		Assembly Program		
H'00A0	H'0232		MOV.B	#H'xx,Rn	
H'00A2	H'0234		MOV.B	Rn,@TCSRWD2:8	; Clear success in case of Address1 and failure in case of Address2
H'00A4	H'0236		MOV.B	@TCSRWD2:8,Rm	; TCSRWD2 read
H'00A6	H'0238		AND.B	#H'xx,Rm	; Judgment of clear
H'00A8	H'023A		BEQ	LABEL:16	; Jumps to LABEL if it is a clear success
H'00AC	H'023E		MOV.B	Rn,@TCSRWD2:8	; Clear success in case of Address2
H'00AE	H'0240	LABEL	NOP		

Table 12.2 The Value of "xx"

Bit(s) Cleared to 0	The Value of "xx" in Line 1	The Value of "xx" in Line 4
Both WT/$\overline{IT}$ and IEOVF	07	28
Only WT/$\overline{IT}$	17	20
Only IEOVF	47	08

Section 13 Asynchronous Event Counter (AEC)

The asynchronous event counter (AEC) is an event counter that is incremented by external event clock or internal clock input. Figure 13.1 shows a block diagram of the asynchronous event counter.

13.1 Features

- Can count asynchronous events

  Can count external events input asynchronously without regard to the operation of system clocks (ϕ) or subclocks (ϕ_{SUB}).
- Can be used as two-channel independent 8-bit event counter or single-channel independent 16-bit event counter.
- Event/clock input is enabled when IRQAEC goes high or event counter PWM output (IECPWM) goes high.
- Both rising and falling edge sensing can be used for IRQAEC or event counter PWM output (IECPWM) interrupts. When the asynchronous counter is not used, they can be used as independent interrupts.
- When an event counter PWM is used, event clock input enabling/disabling can be controlled at a constant cycle.

  An event counter PWM can be output to the AECPWM pin.
- Selection of four clock sources

  Three internal clocks ($\phi/2$, $\phi/4$, or $\phi/8$) or external event can be selected.
- Both rising and falling edge counting is possible for the AEVL and AEVH pins.
- Counter resetting and halting of the count-up function can be controlled by software.
- Automatic interrupt generation on detection of an event counter overflow
- Use of module standby mode enables this module to be placed in standby mode independently when not used. (The asynchronous event counter is halted as the initial value. For details, refer to section 5.4, Module Standby Function.)

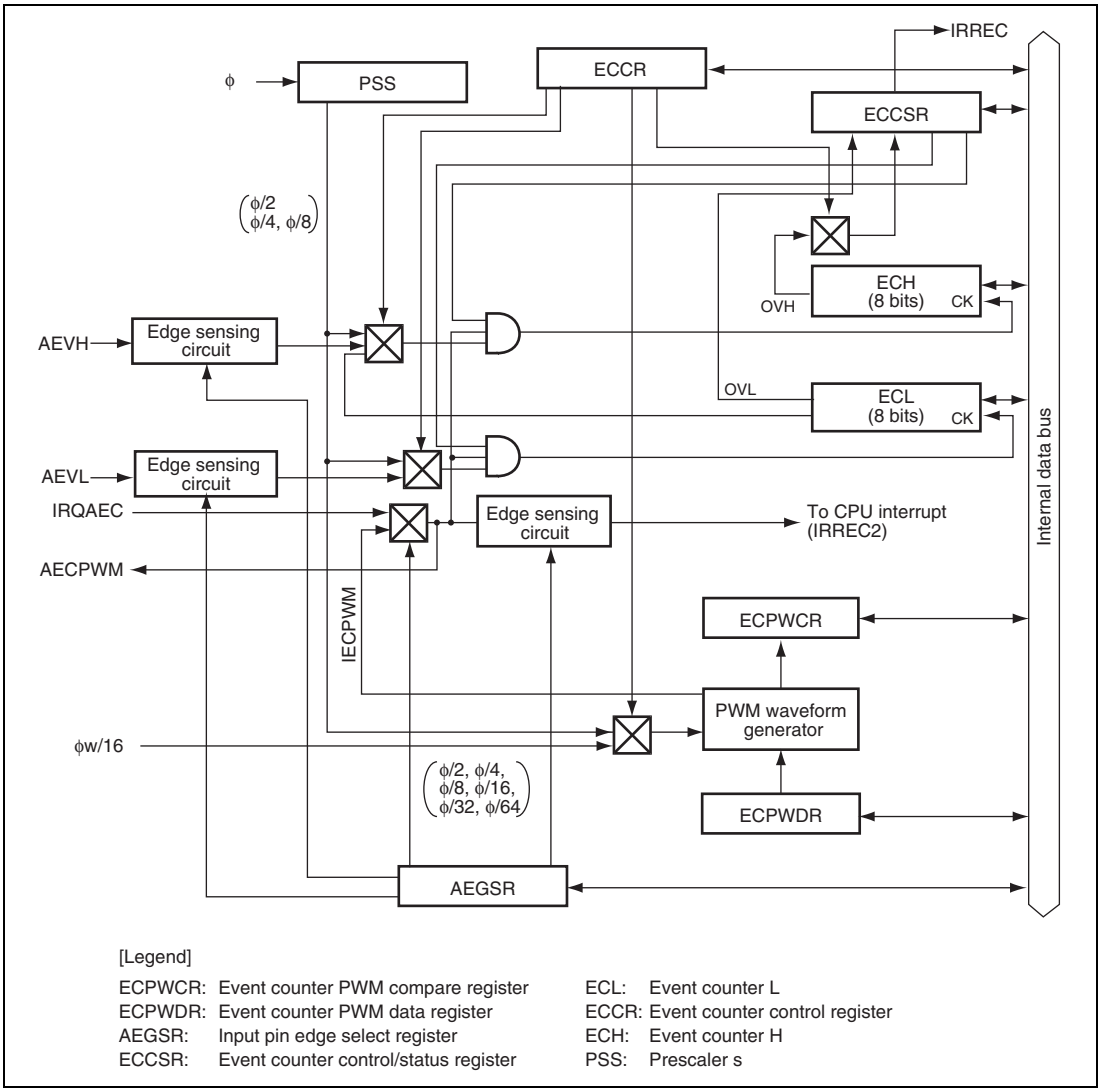


Figure 13.1 Block Diagram of Asynchronous Event Counter

13.2 Input/Output Pins

Table 13.1 shows the pin configuration of the asynchronous event counter.

Table 13.1 Pin Configuration

Name	Abbreviation	I/O	Function
Asynchronous event input H	AEVH	Input	Event input pin for input to event counter H
Asynchronous event input L	AEVL	Input	Event input pin for input to event counter L
Event input enable interrupt input	IRQAEC	Input	Input pin for interrupt enabling event input
Event counter PWM output	AECPWM	Output	Event counter PWM output pin

13.3 Register Descriptions

The asynchronous event counter has the following registers.

- Event counter PWM compare register (ECPWCR)
- Event counter PWM data register (ECPWDR)
- Input pin edge select register (AEGSR)
- Event counter control register (ECCR)
- Event counter control/status register (ECCSR)
- Event counter H (ECH)
- Event counter L (ECL)

13.3.1 Event Counter PWM Compare Register (ECPWCR)

ECPWCR sets the one conversion period of the event counter PWM waveform.

Bit	Bit Name	Initial Value	R/W	Description
15	ECPWCR15	1	R/W	One Conversion Period of Event Counter PWM Waveform When the ECPWME bit in AEGSR is 1, the event counter PWM is operating and therefore ECPWCR should not be modified. When changing the conversion period, the event counter PWM must be halted by clearing the ECPWME bit in AEGSR to 0 before modifying ECPWCR.
14	ECPWCR14	1	R/W	
13	ECPWCR13	1	R/W	
12	ECPWCR12	1	R/W	
11	ECPWCR11	1	R/W	
10	ECPWCR10	1	R/W	
9	ECPWCR9	1	R/W	
8	ECPWCR8	1	R/W	
7	ECPWCR7	1	R/W	
6	ECPWCR6	1	R/W	
5	ECPWCR5	1	R/W	
4	ECPWCR4	1	R/W	
3	ECPWCR3	1	R/W	
2	ECPWCR2	1	R/W	
1	ECPWCR1	1	R/W	
0	ECPWCR0	1	R/W	

13.3.2 Event Counter PWM Data Register (ECPWDR)

ECPWDR controls data of the event counter PWM waveform generator.

Bit	Bit Name	Initial Value	R/W	Description
15	ECPWDR15	0	W	Data Control of Event Counter PWM Waveform Generator When the ECPWME bit in AEGSR is 1, the event counter PWM is operating and therefore ECPWDR should not be modified. When changing the conversion cycle, the event counter PWM must be halted by clearing the ECPWME bit in AEGSR to 0 before modifying ECPWDR. The read value is undefined.
14	ECPWDR14	0	W	
13	ECPWDR13	0	W	
12	ECPWDR12	0	W	
11	ECPWDR11	0	W	
10	ECPWDR10	0	W	
9	ECPWDR9	0	W	
8	ECPWDR8	0	W	
7	ECPWDR7	0	W	
6	ECPWDR6	0	W	
5	ECPWDR5	0	W	
4	ECPWDR4	0	W	
3	ECPWDR3	0	W	
2	ECPWDR2	0	W	
1	ECPWDR1	0	W	
0	ECPWDR0	0	W	

13.3.3 Input Pin Edge Select Register (AEGSR)

AEGSR selects rising, falling, or both edge sensing for the AEVH, AEVL, and IRQAEC pins, and controls IRQAEC/IECPWM.

Bit	Bit Name	Initial Value	R/W	Description
7	AHEGS1	0	R/W	AEC Edge Select H
6	AHEGS0	0	R/W	Select rising, falling, or both edge sensing for the AEVH pin. 00: Falling edge on AEVH pin is sensed 01: Rising edge on AEVH pin is sensed 10: Both edges on AEVH pin are sensed 11: Setting prohibited
5	ALEGS1	0	R/W	AEC Edge Select L
4	ALEGS0	0	R/W	Select rising, falling, or both edge sensing for the AEVL pin. 00: Falling edge on AEVL pin is sensed 01: Rising edge on AEVL pin is sensed 10: Both edges on AEVL pin are sensed 11: Setting prohibited
3	AIEGS1	0	R/W	IRQAEC Edge Select
2	AIEGS0	0	R/W	Select rising, falling, or both edge sensing for the IRQAEC pin. 00: Falling edge on IRQAEC pin is sensed 01: Rising edge on IRQAEC pin is sensed 10: Both edges on IRQAEC pin are sensed 11: Setting prohibited
1	ECPWME	0	R/W	Event Counter PWM Enable Controls operation of event counter PWM and selection of IRQAEC. 0: AEC PWM halted, IRQAEC selected 1: AEC PWM enabled, IRQAEC not selected
0	—	0	R/W	Reserved Although this bit is readable/writable, only 0 should be written to.

13.3.4 Event Counter Control Register (ECCR)

ECCR controls the counter input clock and PWM clock.

Bit	Bit Name	Initial Value	R/W	Description
7	ACKH1	0	R/W	AEC Clock Select H
6	ACKH0	0	R/W	Select the clock used by ECH. 00: AEVH pin input 01: $\phi/2$ 10: $\phi/4$ 11: $\phi/8$
5	ACKL1	0	R/W	AEC Clock Select L
4	ACKL0	0	R/W	Select the clock used by ECL. 00: AEVL pin input 01: $\phi/2$ 10: $\phi/4$ 11: $\phi/8$
3	PWCK2	0	R/W	Event Counter PWM Clock Select
2	PWCK1	0	R/W	Select the event counter PWM clock.
1	PWCK0	0	R/W	000: $\phi/2$ 001: $\phi/4$ 010: $\phi/8$ 011: $\phi/16$ 100: $\phi/32$ 101: $\phi/64$ 110: $\phi_w/16$ 111: Setting prohibited When changing the event counter PWM clock, the ECPWME bit in AEGSR must be cleared to 0 to stop the PWM before rewriting this setting.
0	—	0	R/W	Reserved Although this bit is readable/writable, only 0 should be written to.

13.3.5 Event Counter Control/Status Register (ECCSR)

ECCSR controls counter overflow detection, counter resetting, and count-up function.

Bit	Bit Name	Initial Value	R/W	Description
7	OVH	0	R/W*	Counter Overflow H This is a status flag indicating that ECH has overflowed. [Setting condition] When ECH overflows from H'FF to H'00 [Clearing condition] When this bit is written to 0 after reading OVH = 1
6	OVL	0	R/W*	Counter Overflow L This is a status flag indicating that ECL has overflowed. [Setting condition] When ECL overflows from H'FF to H'00 while CH2 is set to 1 [Clearing condition] When this bit is written to 0 after reading OVL = 1
5	—	0	R/W	Reserved Although this bit is readable/writable, only 0 should be written to.
4	CH2	0	R/W	Channel Select Selects how ECH and ECL event counters are used 0: ECH and ECL are used together as a single-channel 16-bit event counter 1: ECH and ECL are used as two-channel 8-bit event counter
3	CUEH	0	R/W	Count-Up Enable H Enables event clock input to ECH. 0: ECH event clock input is disabled (ECH value is retained) 1: ECH event clock input is enabled

Bit	Bit Name	Initial Value	R/W	Description
2	CUEL	0	R/W	Count-Up Enable L Enables event clock input to ECL. 0: ECL event clock input is disabled (ECL value is retained) 1: ECL event clock input is enabled
1	CRCH	0	R/W	Counter Reset Control H Controls resetting of ECH. 0: ECH is reset 1: ECH reset is cleared and count-up function is enabled
0	CRCL	0	R/W	Counter Reset Control L Controls resetting of ECL. 0: ECL is reset 1: ECL reset is cleared and count-up function is enabled

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

13.3.6 Event Counter H (ECH)

ECH is an 8-bit read-only up-counter that operates as an independent 8-bit event counter. ECH also operates as the upper 8-bit up-counter of a 16-bit event counter configured in combination with ECL.

Bit	Bit Name	Initial Value	R/W	Description
7	ECH7	0	R	Either the external asynchronous event AEVH pin, $\phi/2$, $\phi/4$, or $\phi/8$, or the overflow signal from lower 8-bit counter ECL can be selected as the input clock source. ECH can be cleared to H'00 when the CRCH bit in ECCSR is cleared to 0.
6	ECH6	0	R	
5	ECH5	0	R	
4	ECH4	0	R	
3	ECH3	0	R	
2	ECH2	0	R	
1	ECH1	0	R	
0	ECH0	0	R	

13.3.7 Event Counter L (ECL)

ECL is an 8-bit read-only up-counter that operates as an independent 8-bit event counter. ECL also operates as the lower 8-bit up-counter of a 16-bit event counter configured in combination with ECH.

Bit	Bit Name	Initial Value	R/W	Description
7	ECL7	0	R	Either the external asynchronous event AEVL pin, $\phi/2$, $\phi/4$, or $\phi/8$ can be selected as the input clock source. ECL can be cleared to H'00 when the CRCL bit in ECCSR is cleared to 0.
6	ECL6	0	R	
5	ECL5	0	R	
4	ECL4	0	R	
3	ECL3	0	R	
2	ECL2	0	R	
1	ECL1	0	R	
0	ECL0	0	R	

13.4 Operation

13.4.1 16-Bit Counter Operation

When bit CH2 is cleared to 0 in ECCSR, ECH and ECL operate as a 16-bit event counter.

Any of four input clock sources—$\phi/2$, $\phi/4$, $\phi/8$, or AEVL pin input—can be selected by means of bits ACKL1 and ACKL0 in ECCR. When AEVL pin input is selected, input sensing is selected with bits ALEGS1 and ALEGS0.

Note that the input clock is enabled when IRQAEC is high or IECPWM is high. When IRQAEC is low or IECPWM is low, the input clock is not input to the counter, which therefore does not operate. Figure 13.2 shows the software procedure when ECH and ECL are used as a 16-bit event counter.

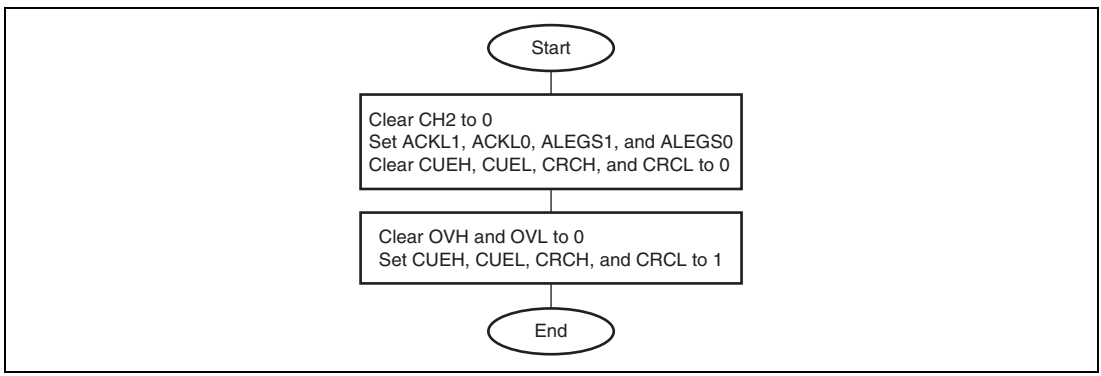


Figure 13.2 Software Procedure when Using ECH and ECL as 16-Bit Event Counter

As CH2 is cleared to 0 by a reset, ECH and ECL operate as a 16-bit event counter after a reset, and as ACKL1 and ACKL0 are cleared to B′00, the operating clock is asynchronous event input from the AEVL pin (using falling edge sensing).

When the next clock is input after the count value reaches H'FF in both ECH and ECL, ECH and ECL overflow from H'FFFF to H'0000, the OVH flag is set to 1 in ECCSR, the ECH and ECL count values each return to H'00, and counting up is restarted. When an overflow occurs, the IRREC bit is set to 1 in IRR2. If the IENEC bit in IENR2 is 1 at this time, an interrupt request is sent to the CPU.

13.4.2 8-Bit Counter Operation

When bit CH2 is set to 1 in ECCSR, ECH and ECL operate as independent 8-bit event counters.

$\phi/2$, $\phi/4$, $\phi/8$, or AEVH pin input can be selected as the input clock source for ECH by means of bits ACKH1 and ACKH0 in ECCR, and $\phi/2$, $\phi/4$, $\phi/8$, or AEVL pin input can be selected as the input clock source for ECL by means of bits ACKL1 and ACKL0 in ECCR. Input sensing is selected with bits AHEGS1 and AHEGS0 when AEVH pin input is selected, and with bits ALEGS1 and ALEGS0 when AEVL pin input is selected.

Note that the input clock is enabled when IRQAEC is high or IECPWM is high. When IRQAEC is low or IECPWM is low, the input clock is not input to the counter, which therefore does not operate. Figure 13.3 shows the software procedure when ECH and ECL are used as 8-bit event counters.

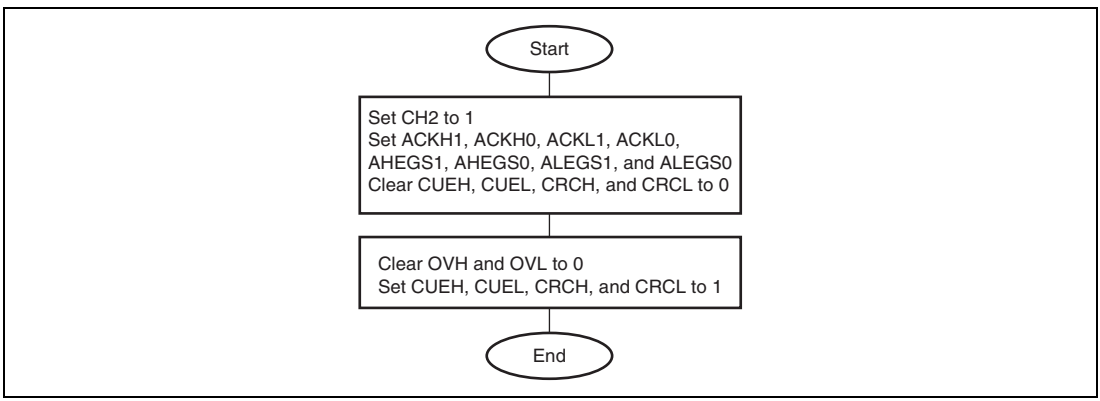


Figure 13.3 Software Procedure when Using ECH and ECL as 8-Bit Event Counters

When the next clock is input after the ECH count value reaches H'FF, ECH overflows, the OVH flag is set to 1 in ECCSR, the ECH count value returns to H'00, and counting up is restarted. Similarly, when the next clock is input after the ECL count value reaches H'FF, ECL overflows, the OVL flag is set to 1 in ECCSR, the ECL count value returns to H'00, and counting up is restarted. When an overflow occurs, the IRREC bit is set to 1 in IRR2. If the IENEC bit in IENR2 is 1 at this time, an interrupt request is sent to the CPU.

13.4.3 IRQAEC Operation

When the ECPWME bit in AEGSR is 0, the ECH and ECL input clocks are enabled when IRQAEC goes high. When IRQAEC goes low, the input clocks are not input to the counters, and so ECH and ECL do not count. ECH and ECL count operations can therefore be controlled from outside by controlling IRQAEC. In this case, ECH and ECL cannot be controlled individually.

IRQAEC can also operate as an interrupt source.

Interrupt enabling is controlled by IENEC2 in IENR1. When an IRQAEC interrupt is generated, interrupt request flag IRREC2 in IRR1 is set to 1. If IENEC2 in IENR1 is set to 1 at this time, an interrupt request is sent to the CPU.

Rising, falling, or both edge sensing can be selected for the IRQAEC input pin with bits AIEGS1 and AIEGS0 in AEGSR.

13.4.4 Event Counter PWM Operation

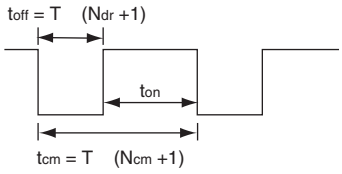
When the ECPWME bit in AEGSR is 1, the ECH and ECL input clocks are enabled when event counter PWM output (IECPWM) is high. When IECPWM is low, the input clocks are not input to the counters, and so ECH and ECL do not count. ECH and ECL count operations can therefore be controlled cyclically by controlling event counter PWM. In this case, ECH and ECL cannot be controlled individually.

IECPWM can also operate as an interrupt source.

Interrupt enabling is controlled by IENEC2 in IENR1. When an IECPWM interrupt is generated, interrupt request flag IRREC2 in IRR1 is set to 1. If IENEC2 in IENR1 is set to 1 at this time, an interrupt request is sent to the CPU.

Rising, falling, or both edge detection can be selected for IECPWM interrupt sensing with bits AIEGS1 and AIEGS0 in AEGSR.

Figure 13.4 and table 13.2 show examples of event counter PWM operation.



ton: Clock input enable time
toff: Clock input disable time
tcm: One conversion period
T: ECPWM input clock cycle
Ndr: Value of ECPWDR
Ncm: Value of ECPWCR

Figure 13.4 Event Counter Operation Waveform

Note: Ndr and Ncm above must be set so that Ndr < Ncm. If the settings do not satisfy this condition, event counter PWM output (IECPWM) is fixed low.

Table 13.2 Examples of Event Counter PWM Operation

Conditions: f_{OSC} = 4 MHz, fϕ = 4 MHz, f_W = 32.768 kHz, fϕ_W = 32.768 kHz, high-speed active mode, ECPWCR value (Ncm) = H'7A11, ECPWDR value (Ndr) = H'16E3

Clock Source Selection	Clock Source Cycle (T)*	ECPWCR Value (Ncm)	ECPWDR Value (Ndr)	toff = T × (Ndr + 1)	tcm = T × (Ncm + 1)	ton = tcm – toff
ϕ/2	0.5 µs	H'7A11	H'16E3	2.93 ms	15.625 ms	12.695 ms
ϕ/4	1 µs	D'31249	D'5859	5.86 ms	31.25 ms	25.39 ms
ϕ/8	2 µs			11.72 ms	62.5 ms	50.78 ms
ϕ/16	4 µs			23.44 ms	125.0 ms	101.56 ms
ϕ/32	8 µs			46.88 ms	250.0 ms	203.12 ms
ϕ/64	16 µs			93.76 ms	500.0 ms	406.24 ms
ϕ_W/16	488 µs			2861.59 ms	15260.19 ms	12398.60 ms

Note: * toff minimum width

13.4.5 Operation of Clock Input Enable/Disable Function

The clock input to the event counter can be controlled by the IRQAEC pin when ECPWME in AEGSR is 0, and by the event counter PWM output, IECPWM when ECPWME in AEGSR is 1. As this function forcibly terminates the clock input by each signal, a maximum error of one count will occur depending on the IRQAEC or IECPWM timing. Figure 13.5 shows an example of the operation.

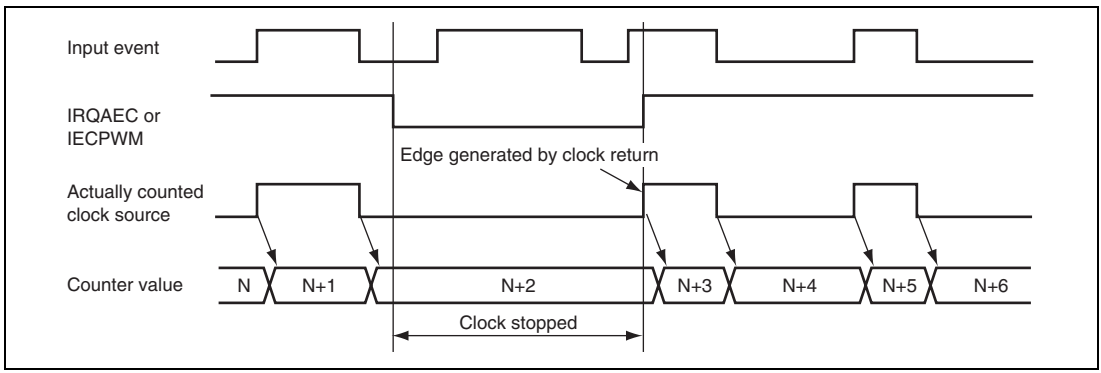


Figure 13.5 Example of Clock Control Operation

13.5 Operating States of Asynchronous Event Counter

The operating states of the asynchronous event counter are shown in table 13.3.

Table 13.3 Operating States of Asynchronous Event Counter

		Active		Sleep						Oscillation Stabilization Time			
Counter	Clock Source	High-speed	Medium-speed	High-speed	Medium-speed	Watch	Sub-active	Sub-sleep	Standby	Standby to Active	Subsleep to Active	Watch to Active	Remarks
ECH, ECL	AEVH, AEVL	o	o	o	o	o	o	o	o	o	o	o	*1
	ϕ/2, ϕ/4, ϕ/8	o	o	o	o	×	×	×	×	×	×	×	
PWM	ϕw/16	o	o	o	o	o	o	o	×	×	o	o	*2
	ϕ/2, ϕ/4, ϕ/8, ϕ/16, ϕ/32, ϕ/64	o	o	o	o	×	×	×`	×	×	×	×	

[Legend] o: Counting enabled

×: Counting disabled (Counter value retained)

Notes: 1. The count-up function is enabled only when IRQAEC/IECPWM = 1.

2. Output is in the high-impedance state during standby mode or the oscillation stabilization time from standby mode.

13.6 Usage Notes

1. When reading the values in ECH and ECL, first clear bits CUEH and CUEL to 0 in ECCSR in 8-bit mode and clear bit CUEL to 0 in 16-bit mode to prevent asynchronous event input to the counter. The correct value will not be returned if the event counter increments while being read.
2. For input to the AEVH and AEVL pins, use a clock with a frequency of up to 4.2 MHz within the range from 1.8 to 3.6 V and up to 10 MHz within the range from 2.7 to 3.6 V. For the high and low widths of the clock, see section 21, Electrical Characteristics. The duty cycle is arbitrary.

Table 13.4 Maximum Clock Frequency

Mode		Maximum Clock Frequency Input to AEVH/AEVL Pin
Active (high-speed), sleep (high-speed)		4 to 10 MHz (2.7 to 3.6 V)
		2 to 4.2 MHz (1.8 to 3.6 V)
Active (medium-speed), sleep (medium-speed)	(ϕ_{osc}/8)	$2 \cdot f_{osc}$
	(ϕ_{osc}/16)	f_{osc}
f_{osc} = 4 MHz to 10 MHz (2.7 to 3.6 V)	(ϕ_{osc}/32)	$1/2 \cdot f_{osc}$
f_{osc} = 2 MHz to 4.2 MHz (1.8 to 3.6 V)	(ϕ_{osc}/64)	$1/4 \cdot f_{osc}$
Watch, subactive, subsleep, standby	(ϕ_w)	2000 kHz
	(ϕ_w/2)	1000 kHz
	(ϕ_w/4)	500 kHz
ϕ_w = 32.768 kHz or 38.4 kHz	(ϕ_w/8)	250 kHz

3. When AEC uses with 16-bit mode, set CUEH in ECCSR to 1 first, set CRCH in ECCSR to 1 second, or set both CUEH and CRCH to 1 at same time before clock input. When AEC is operating on 16-bit mode, do not change CUEH. Otherwise, ECH will be miscounted up.
4. When ECPWME in AEGSR is 1, the event counter PWM is operating and therefore ECPWCR and ECPWDR should not be modified.

   When changing the data, clear the ECPWME bit in AEGSR to 0 (halt the event counter PWM) before modifying these registers.
5. The event counter PWM data register and event counter PWM compare register must be set so that event counter PWM data register < event counter PWM compare register. If the settings do not satisfy this condition, do not set ECPWME to 1 in AEGSR.

6. As synchronization is established internally when an IRQAEC interrupt is generated, a maximum error of 1 tcyc or 1tsubcyc will occur between clock halting and interrupt acceptance.
7. When pins in port 1 are used for AEC input/output, the PFCR and PMR1 registers should be set in the following order.
   a. Set the PFCR register.
   b. Set bits 4 to 0 after bit 5 (IRQAEC bit) in the PMR1 register has been cleared to 0.
   c. Set bit 5 (IRQAEC bit) in the PMR1 register to 1. At this time, bits 4 to 0 should not be changed and remain the same.

Section 14 Serial Communication Interface 3 (SCI3, IrDA)

The serial communication interface 3 (SCI3) can handle both asynchronous and clock synchronous serial communication. In the asynchronous method, serial data communication can be carried out using standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or an Asynchronous Communication Interface Adapter (ACIA).

The SCI3 can transmit and receive IrDA communication waveforms based on the Infrared Data Association (IrDA) standard version 1.0.

14.1 Features

- Choice of asynchronous or clock synchronous serial communication mode
- Full-duplex communication capability

  The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously.

  Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected
- On-chip baud rate generator, internal clock, or external clock can be selected as a transfer clock source.
- Six interrupt sources

  Transmit-end, transmit-data-empty, receive-data-full, overrun error, framing error, and parity error.
- Use of module standby mode enables this module to be placed in standby mode independently when not used. (The SCI3 is halted as the initial value. For details, refer to section 5.4, Module Standby Function.)

Asynchronous mode

- Data length: 7, 8, or 5 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Receive error detection: Parity, overrun, and framing errors
- Break detection: Break can be detected by reading the RXD3 pin level directly in the case of a framing error

Clock synchronous mode

- Data length: 8 bits
- Receive error detection: Overrun errors detected

Note: When using serial communication interface 3, the system clock oscillator or subclock oscillator must be used.

Figure 14.1 shows a block diagram of the SCI3.

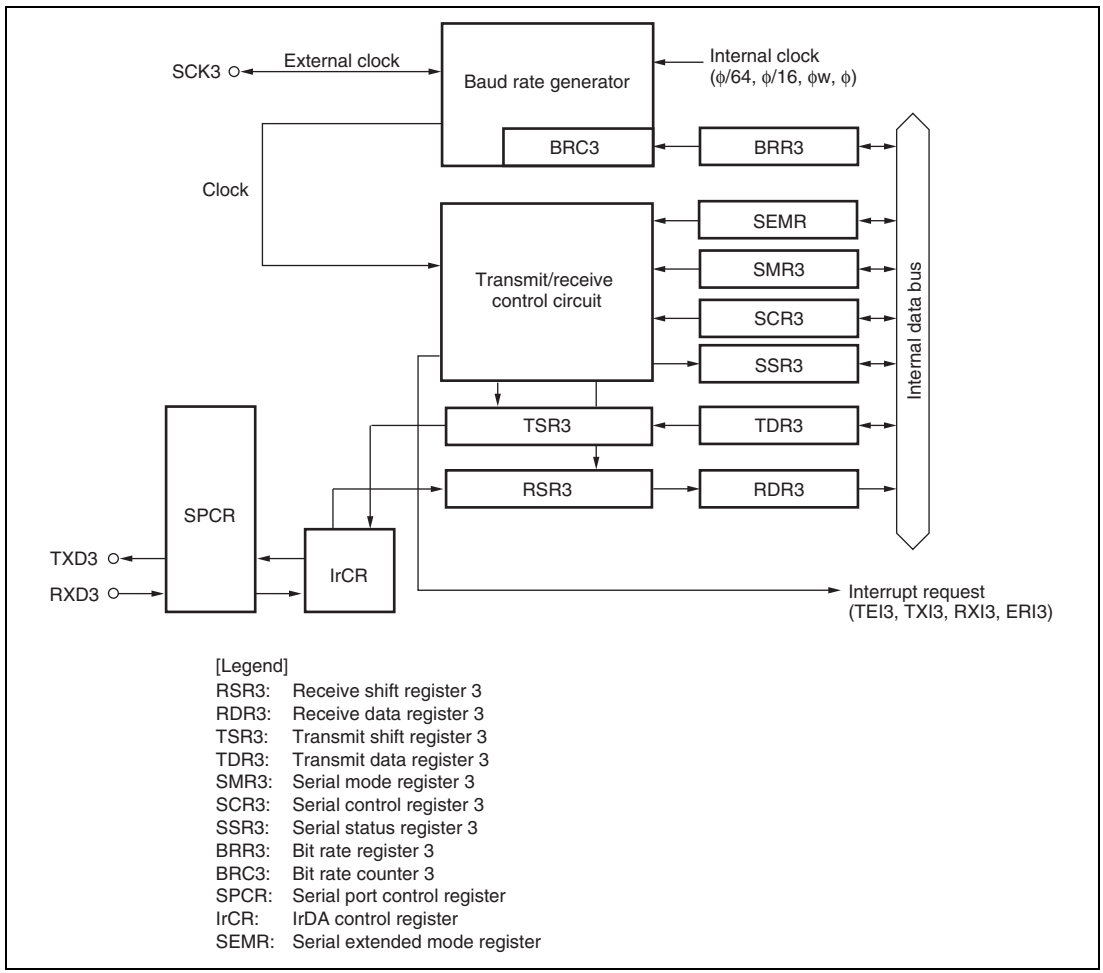


Figure 14.1 Block Diagram of SCI3

14.2 Input/Output Pins

Table 14.1 shows the pin configuration of the SCI3.

Table 14.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
SCI3 clock	SCK3	I/O	SCI3 clock input/output
SCI3 receive data input	RXD3	Input	SCI3 receive data input
SCI3 transmit data output	TXD3	Output	SCI3 transmit data output

14.3 Register Descriptions

The SCI3 has the following registers for each channel.

- Receive shift register 3 (RSR3)*
- Receive data register 3 (RDR3)*
- Transmit shift register 3 (TSR3)*
- Transmit data register 3 (TDR3)*
- Serial mode register 3 (SMR3)*
- Serial control register 3 (SCR3)*
- Serial status register 3 (SSR3)*
- Bit rate register 3 (BRR3)*
- Serial port control register (SPCR)
- IrDA control register (IrCR)
- Serial extended mode register (SEMR)

Note: * These register names are abbreviated to RSR, RDR, TSR, TDR, SMR, SCR, SSR, and BRR in the text.

14.3.1 Receive Shift Register (RSR)

RSR is a shift register that receives serial data input from the RXD3 pin and converts it into parallel data. When one byte of data has been received, it is transferred to RDR automatically. RSR cannot be directly accessed by the CPU.

14.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores receive data. When the SCI3 has received one byte of serial data, it transfers the received serial data from RSR to RDR, where it is stored. After this, RSR is receive-enabled. As RSR and RDR function as a double buffer in this way, continuous receive operations are possible. After confirming that the RDRF bit in SSR is set to 1, read RDR only once. RDR cannot be written to by the CPU. RDR is initialized to H'00.

RDR is initialized to H'00 by a reset or in standby mode, watch mode, or module standby mode.

14.3.3 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI3 first transfers transmit data from TDR to TSR automatically, then sends the data that starts from the LSB to the TXD3 pin. Data transfer from TDR to TSR is not performed if no data has been written to TDR (if the TDRE bit in SSR is set to 1). TSR cannot be directly accessed by the CPU.

14.3.4 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for transmission. When the SCI3 detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structure of TDR and TSR enables continuous serial transmission. If the next transmit data has already been written to TDR during transmission of one-frame data, the SCI3 transfers the written data to TSR to continue transmission. To achieve reliable serial transmission, write transmit data to TDR only once after confirming that the TDRE bit in SSR is set to 1. TDR is initialized to H'FF.

TDR is initialized to H'FF by a reset or in standby mode, watch mode, or module standby mode.

14.3.5 Serial Mode Register (SMR)

SMR sets the SCI3's serial communication format and selects the clock source for the on-chip baud rate generator.

SMR is initialized to H'00 by a reset or in standby mode, watch mode, or module standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	COM	0	R/W	Communication Mode 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous mode) 0: Selects 8 or 5 bits as the data length. 1: Selects 7 or 5 bits as the data length. When 7-bit data is selected. the MSB (bit 7) in TDR is not transmitted. To select 5 bits as the data length, set 1 to both the PE and MP bits. The three most significant bits (bits 7, 6, and 5) in TDR are not transmitted. In clock synchronous mode, the data length is fixed to 8 bits regardless of the CHR bit setting.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. In clock synchronous mode, parity bit addition and checking is not performed regardless of the PE bit setting.

Bit	Bit Name	Initial Value	R/W	Description
4	PM	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity. When even parity is selected, a parity bit is added in transmission so that the total number of 1 bits in the transmit data plus the parity bit is an even number, in reception, a check is carried out to confirm that the number of 1 bits in the receive data plus the parity bit is an even number. When odd parity is selected, a parity bit is added in transmission so that the total number of 1 bits in the transmit data plus the parity bit is an odd number, in reception, a check is carried out to confirm that the number of 1bits in the receive data plus the parity bit is an odd number. If parity bit addition and checking is disabled in clock synchronous mode and asynchronous mode, the PM bit setting is invalid.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous mode) Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits For reception, only the first stop bit is checked, regardless of the value in the bit. If the second stop bit is 0, it is treated as the start bit of the next transmit character.
2	MP	0	R/W	5-Bit Communication When this bit is set to 1, the 5-bit communication format is enabled. Make sure to set bit 5 (PF) to 1 when setting this bit (MP) to 1.

Bit	Bit Name	Initial Value	R/W	Description
1	CKS1	0	R/W	Clock Select 0 and 1
0	CKS0	0	R/W	These bits select the clock source for the on-chip baud rate generator. 00: ϕ clock (n = 0) 01: ϕ_w clock (n = 0) 10: ϕ/16 clock (n = 2) 11: ϕ/64 clock (n = 3) When the setting value is 01 in subactive mode or subsleep mode, the SCI3 can be used only when ϕ_w is selected for the CPU operating clock. For the relationship between the bit rate register setting and the baud rate, see section 14.3.8, Bit Rate Register (BRR). n is the decimal representation of the value of n in BRR (see section 14.3.8, Bit Rate Register (BRR)).

14.3.6 Serial Control Register (SCR)

SCR enables or disables SCI3 transfer operations and interrupt requests, and selects the transfer clock source. For details on interrupt requests, refer to section 14.7, Interrupt Requests.

SCR is initialized to H'00 by a reset or in standby mode, watch mode, or module standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, the TXI3 interrupt request is enabled. TXI3 can be released by clearing the TDRE it or TI bit to 0.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, the RXI3 and ERI3 interrupt requests are enabled. RXI3 and ERI3 can be released by clearing the RDRF bit or the FER, PER, or OER error flag to 0, or by clearing the RIE bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled. When this bit is 0, the TDRE bit in SSR is fixed at 1. When transmit data is written to TDR while this bit is 1, Bit TDRE in SSR is cleared to 0 and serial data transmission is started. Be sure to carry out SMR settings, and setting of bit SPC3 in SPCR, to decide the transmission format before setting bit TE to 1.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled. In this state, serial data reception is started when a start bit is detected in asynchronous mode or serial clock input is detected in clock synchronous mode. Be sure to carry out the SMR settings to decide the reception format before setting bit RE to 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
3	MPIE	0	R/W	Reserved
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, the TEI3 interrupt request is enabled. TEI3 can be released by clearing bit TDRE to 0 and clearing bit TEND to 0 in SSR, or by clearing bit TEIE to 0.

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	Clock Enable 0 and 1
0	CKE0	0	R/W	Select the clock source. Asynchronous mode: 00: Internal baud rate generator (SCK3 pin functions as an I/O port) 01: Internal baud rate generator (Outputs a clock of the same frequency as the bit rate from the SCK3 pin) 10: External clock (Inputs a clock with a frequency 16 times the bit rate from the SCK3 pin) 11: Reserved Clock synchronous mode: 00: Internal clock (SCK3 pin functions as clock output) 01: Reserved 10: External clock (SCK3 pin functions as clock input) 11: Reserved

14.3.7 Serial Status Register (SSR)

SSR consists of status flags of the SCI3. 1 cannot be written to flags TDRE, RDRF, OER, PER, and FER; they can only be cleared.

SSR is initialized to H'84 by a reset or in standby mode, watch mode, or module standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Indicates that transmit data is stored in TDR. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the transmit data is written to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that the received data is stored in RDR. [Setting condition] • When serial reception ends normally and receive data is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 When data is read from RDR If an error is detected in reception, or if the RE bit in SCR 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 occur and the receive data will be lost.

Bit	Bit Name	Initial Value	R/W	Description
5	OER	0	R/(W)*	Overrun Error [Setting condition] • When an overrun error occurs in reception [Clearing condition] • When 0 is written to OER after reading OER = 1 When bit RE in SCR is cleared to 0, bit OER is not affected and retains its previous state. When an overrun error occurs, 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 clock synchronous mode, transmission cannot be continued either.
4	FER	0	R/(W)*	Framing Error [Setting condition] • When a framing error occurs in reception [Clearing condition] • When 0 is written to FER after reading FER = 1 When bit RE in SCR is cleared to 0, bit FER is not affected and retains its previous state. 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 clock synchronous mode, neither transmission nor reception is possible when bit FER is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/(W)*	Parity Error [Setting condition] • When a parity error is generated during reception [Clearing condition] • When 0 is written to PER after reading PER = 1 When bit RE in SCR is cleared to 0, bit PER is not affected and retains its previous state. • Receive data in which 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 clock synchronous mode, neither transmission nor reception is possible when bit PER is set to 1.
2	TEND	1	R	Transmit End [Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the transmit data is written to TDR
1	MPBR	0	R	Reserved This bit is always read as 0 and cannot be modified.
0	MPBT	0	R/W	Reserved The write value should always be 0.

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

14.3.8 Bit Rate Register (BRR)

BRR is an 8-bit readable/writable register that adjusts the bit rate. BRR is initialized to H'FF. Tables 14.2 and 14.3 show the relationship between the N setting in BRR and the n setting in bits CKS1 and CKS0 in SMR in asynchronous mode. Table 14.5 shows the maximum bit rate for each frequency in asynchronous mode. The values shown in these tables are values in active (high-speed) mode. When the ABCS bit in SEMR is set to 1 in asynchronous mode, the maximum bit rate in table 14.5 is doubled. Table 14.6 shows the relationship between the N setting in BRR and the n setting in bits CKS1 and CKS0 in SMR in clock synchronous mode. The values shown in table 14.6 are values in active (high-speed) mode. The N setting in BRR and error for other operating frequencies and bit rates can be obtained by the following formulas:

[Asynchronous Mode and ABCS Bit is 0]

$$N = \frac{\phi}{32 \times 2^{2n} \times B} - 1$$

$$\text{Error (\%)} = \frac{\text{B (bit rate obtained from n, N, } \phi\text{)} - \text{R (bit rate in left-hand column in table 14.2)}}{\text{R (bit rate in left-hand column in table 14.2)}} \times 100$$

[Asynchronous Mode and ABCS Bit is 1]

$$N = \frac{\phi}{16 \times 2^{2n} \times B} - 1$$

$$\text{Error (\%)} = \frac{\text{B (bit rate obtained from n, N, } \phi\text{)} - \text{R (bit rate in left-hand column in table 14.3)}}{\text{R (bit rate in left-hand column in table 14.3)}} \times 100$$

[Legend]

B: Bit rate (bit/s)

N: BRR setting for baud rate generator ($0 \leq N \leq 255$)

ϕ: Operating frequency (Hz)

n: Baud rate generator input clock number (n = 0, 2, or 3)
(The relation between n and the clock is shown in table 14.4)

Table 14.2 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and ABCS Bit is 0) (1)

Bit Rate (bit/s)	32.8 kHz			38.4 kHz			2 MHz			2.097152 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	—	—	—	0	10	–0.83	2	35	–1.36	2	36	0.64
150	0	6	–2.38	0	7	0.00	2	25	0.16	2	26	1.14
200	0	4	2.50	0	5	0.00	2	19	–2.34	2	19	2.40
250	0	3	2.50	—	—	—	0	249	0.00	2	15	2.40
300	—	—	—	0	3	0.00	0	207	0.16	0	217	0.21
600	—	—	—	0	1	0.00	0	103	0.16	0	108	0.21
1200	—	—	—	0	0	0.00	0	51	0.16	0	54	–0.70
2400	—	—	—	—	—	—	0	25	0.16	0	26	1.14
4800	—	—	—	—	—	—	0	12	0.16	0	13	–2.48
9600	—	—	—	—	—	—	—	—	—	0	6	–2.48
19200	—	—	—	—	—	—	—	—	—	—	—	—
31250	—	—	—	—	—	—	0	1	0.00	—	—	—
38400	—	—	—	—	—	—	—	—	—	—	—	—

Table 14.2 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and ABCS Bit is 0) (2)

Bit Rate (bit/s)	2.4576 MHz			3 MHz			3.6864 MHz			4 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	43	–0.83	2	52	0.50	2	64	0.70	2	70	0.03
150	2	31	0.00	2	38	0.16	2	47	0.00	2	51	0.16
200	2	23	0.00	2	28	1.02	2	35	0.00	2	38	0.16
250	2	18	1.05	2	22	1.90	2	28	–0.69	2	30	0.81
300	0	255	0.00	2	19	–2.34	2	23	0.00	2	25	0.16
600	0	127	0.00	0	155	0.16	0	191	0.00	0	207	0.16
1200	0	63	0.00	0	77	0.16	0	95	0.00	0	103	0.16
2400	0	31	0.00	0	38	0.16	0	47	0.00	0	51	0.16
4800	0	15	0.00	0	19	–2.34	0	23	0.00	0	25	0.16
9600	0	7	0.00	0	9	–2.34	0	11	0.00	0	12	0.16
19200	0	3	0.00	0	4	–2.34	0	5	0.00	—	—	—
31250	—	—	—	0	2	0.00	—	—	—	0	3	0.00
38400	0	1	0.00	—	—	—	0	2	0.00	—	—	—

Table 14.2 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and ABCS Bit is 0) (3)

	4.194304 MHz			4.9152 MHz			5 MHz			6 MHz		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	73	0.64	2	86	0.31	2	88	–0.25	2	106	–0.44
150	2	54	–0.70	2	63	0.00	2	64	0.16	2	77	0.16
200	2	40	–0.10	2	47	0.00	2	48	–0.35	2	58	–0.69
250	2	32	–0.70	2	37	1.05	2	38	0.16	2	46	-0.27
300	2	26	1.14	2	31	0.00	2	32	–1.36	2	38	0.16
600	0	217	0.21	0	255	0.00	2	15	1.73	2	19	–2.34
1200	0	108	0.21	0	127	0.00	0	129	0.16	0	155	0.16
2400	0	54	–0.70	0	63	0.00	0	64	0.16	0	77	0.16
4800	0	26	1.14	0	31	0.00	0	32	–1.36	0	38	0.16
9600	0	13	–2.48	0	15	0.00	0	15	1.73	0	19	–2.34
19200	0	6	–2.48	0	7	0.00	0	7	1.73	0	9	–2.34
31250	—	—	—	0	4	–1.70	0	4	0.00	0	5	0.00
38400	—	—	—	0	3	0.00	0	3	1.73	0	4	–2.34

Table 14.2 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and ABCS Bit is 0) (4)

	6.144 MHz			7.3728 MHz			8 MHz			9.8304 MHz			10 MHz		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	108	0.08	2	130	–0.07	2	141	0.03	2	174	–0.26	2	177	–0.25
150	2	79	0.00	2	95	0.00	2	103	0.16	2	127	0.00	2	129	0.16
200	2	59	0.00	2	71	0.00	2	77	0.16	2	95	0.00	2	97	–0.35
250	2	47	0.00	2	57	–0.69	2	62	-0.79	2	76	-0.26	2	77	0.16
300	2	39	0.00	2	47	0.00	2	51	0.16	2	63	0.00	2	64	0.16
600	2	19	0.00	2	23	0.00	2	25	0.16	2	31	0.00	2	32	–1.36
1200	0	159	0.00	0	191	0.00	0	207	0.16	0	255	0.00	2	15	1.73
2400	0	79	0.00	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
4800	0	39	0.00	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
9600	0	19	0.00	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
19200	0	9	0.00	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
31250	0	5	2.40	—	—	—	0	7	0.00	0	9	–1.70	0	9	0.00
38400	0	4	0.00	0	5	0.00	—	—	—	0	7	0.00	0	7	1.73

Table 14.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and ABCS Bit is 1) (1)

	32.8 kHz			38.4 kHz			2 MHz			2.097152 MHz		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	0	18	–1.91	0	21	-0.83	2	70	0.03	2	73	0.64
150	0	13	–2.38	0	15	0.00	2	51	0.16	2	54	–0.70
200	0	9	2.50	0	11	0.00	2	38	0.16	2	40	–0.10
250	0	7	2.50	—	—	—	2	30	0.81	2	32	–0.70
300	0	6	–2.38	0	7	0.00	2	25	0.16	2	26	1.14
600	—	—	—	0	3	0.00	0	207	0.16	0	217	0.21
1200	—	—	—	0	1	0.00	0	103	0.16	0	108	0.21
2400	—	—	—	0	0	0.00	0	51	0.16	0	54	-0.70
4800	—	—	—	—	—	—	0	25	0.16	0	26	1.14
9600	—	—	—	—	—	—	0	12	0.16	0	13	–2.48
19200	—	—	—	—	—	—	—	—	—	0	6	–2.48
31250	—	—	—	—	—	—	0	3	0.00	—	—	—
38400	—	—	—	—	—	—	—	—	—	—	—	—

Table 14.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and ABCS Bit is 1) (2)

	2.4576 MHz			3 MHz			3.6864 MHz			4 MHz		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	86	0.31	2	106	–0.44	2	130	-0.07	2	141	0.03
150	2	63	0.00	2	77	0.16	2	95	0.00	2	103	0.16
200	2	47	0.00	2	58	–0.69	2	71	0.00	2	77	0.16
250	2	37	1.05	2	46	–0.27	2	57	-0.69	2	62	–0.79
300	2	31	0.00	2	38	0.16	2	47	0.00	2	51	0.16
600	0	255	0.00	2	19	–2.34	2	23	0.00	2	25	0.16
1200	0	127	0.00	0	155	0.16	0	191	0.00	0	207	0.16
2400	0	63	0.00	0	77	0.16	0	95	0.00	0	103	0.16
4800	0	31	0.00	0	38	0.16	0	47	0.00	0	51	0.16
9600	0	15	0.00	0	19	–2.34	0	23	0.00	0	25	0.16
19200	0	7	0.00	0	9	–2.34	0	11	0.00	0	12	0.16
31250	0	4	–1.70	0	5	0.00	—	—	—	0	7	0.00
38400	0	3	0.00	0	4	–2.34	0	5	0.00	—	—	—

Table 14.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and ABCS Bit is 1) (3)

	4.194304 MHz			4.9152 MHz			5 MHz			6 MHz		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	148	–0.04	2	174	–0.26	2	177	–0.25	2	212	0.03
150	2	108	0.21	2	127	0.00	2	129	0.16	2	155	0.16
200	2	81	–0.10	2	95	0.00	2	97	–0.35	2	116	0.16
250	2	65	–0.70	2	76	–0.26	2	77	0.16	2	93	–0.27
300	2	54	–0.70	2	63	0.00	2	64	0.16	2	77	0.16
600	2	26	1.14	2	31	0.00	2	32	–1.36	2	38	0.16
1200	0	217	0.21	0	255	0.00	2	15	1.73	2	19	–2.34
2400	0	108	0.21	0	127	0.00	0	129	0.16	0	155	0.16
4800	0	54	–0.70	0	63	0.00	0	64	0.16	0	77	0.16
9600	0	26	1.14	0	31	0.00	0	32	–1.36	0	38	0.16
19200	0	13	–2.48	0	15	0.00	0	15	1.73	0	19	–2.34
31250	—	—	—	0	9	–1.70	0	9	0.00	0	11	0.00
38400	0	6	-2.48	0	7	0.00	0	7	1.73	0	9	–2.34

Table 14.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and ABCS Bit is 1) (4)

	6.144 MHz			7.3728 MHz			8 MHz			9.8304 MHz			10 MHz		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	3	64	0.70	3	70	0.03	3	86	0.31	3	88	–0.25
150	2	159	0.00	2	191	0.00	2	207	0.16	2	255	0.00	3	64	0.16
200	2	119	0.00	2	143	0.00	2	155	0.16	2	191	0.00	2	194	0.16
250	2	95	0.00	2	114	0.17	2	124	0.00	2	153	-0.26	2	155	0.16
300	2	79	0.00	2	95	0.00	2	103	0.16	2	127	0.00	2	129	0.16
600	2	39	0.00	2	47	0.00	2	51	0.16	2	63	0.00	2	64	0.16
1200	2	19	0.00	2	23	0.00	2	25	0.16	2	31	0.00	2	32	–1.36
2400	0	159	0.00	0	191	0.00	0	207	0.16	0	255	0.00	2	15	1.73
4800	0	79	0.00	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
9600	0	39	0.00	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
19200	0	19	0.00	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
31250	0	11	2.40	0	14	–1.70	0	15	0.00	0	19	–1.70	0	19	0.00
38400	0	9	0.00	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73

Table 14.4 Relation between n and Clock

		SMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	ϕ	0	0
0	ϕw*	0	1
2	ϕ/16	1	0
3	ϕ/64	1	1

Note: * In subactive or subsleep mode, the SCI3 can be operated only when the CPU operating clock is ϕ_w.

Table 14.5 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

	Maximum Bit Rate (bit/s)		Setting	
ϕ (MHz)	**ABCS = 0**	**ABCS = 1**	**n**	**N**
0.0328*	1025	2050	0	0
0.0384*	1200	2400	0	0
2	62500	125000	0	0
2.097152	65536	131072	0	0
2.4576	76800	153600	0	0
3	93750	187500	0	0
3.6864	115200	230400	0	0
4	125000	250000	0	0
4.194304	131072	262144	0	0
4.9152	153600	307200	0	0
5	156250	312500	0	0
6	187500	375000	0	0
6.144	192000	384000	0	0
7.3728	230400	460800	0	0
8	250000	500000	0	0
9.8304	307200	614400	0	0
10	312500	625000	0	0

Note: * When CKS1 = 0 and CKS0 = 1 in SMR

Table 14.6 BRR Settings for Various Bit Rates (Clock Synchronous Mode) (1)

ϕ	32.8 kHz			38.4 kHz			2 MHz		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
200	0	40	0.00	0	47	0.00	2	155	0.16
250	0	32	–0.61	0	37	1.05	2	124	0.00
300	0	26	1.23	0	31	0.00	2	103	0.16
500	0	15	2.50	0	18	1.05	2	62	–0.79
1k	0	7	2.50	—	—	—	2	30	0.81
2.5k	—	—	—	—	—	—	0	199	0.00
5k	—	—	—	—	—	—	0	99	0.00
10k	—	—	—	—	—	—	0	49	0.00
25k	—	—	—	—	—	—	0	19	0.00
50k	—	—	—	—	—	—	0	9	0.00
100k	—	—	—	—	—	—	0	4	0.00
250k	—	—	—	—	—	—	0	1	0.00
500k	—	—	—	—	—	—	0*	0*	0.00*
1M	—	—	—	—	—	—	—	—	—

Note: * Continuous transmission/reception is not possible.

Table 14.6 BRR Settings for Various Bit Rates (Clock Synchronous Mode) (2)

ϕ	4 MHz			8 MHz			10 MHz		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
200	3	77	0.16	3	155	0.16	3	194	0.16
250	2	249	0.00	3	124	0.00	3	155	0.16
300	2	207	0.16	3	103	0.16	3	129	0.16
500	2	124	0.00	2	249	0.00	3	77	0.16
1k	2	62	–0.79	2	124	0.00	2	155	0.16
2.5k	2	24	0.00	2	49	0.00	2	62	–0.79
5k	0	199	0.00	2	24	0.00	2	30	0.81
10k	0	99	0.00	0	199	0.00	0	249	0.00
25k	0	39	0.00	0	79	0.00	0	99	0.00
50k	0	19	0.00	0	39	0.00	0	49	0.00
100k	0	9	0.00	0	19	0.00	0	24	0.00
250k	0	3	0.00	0	7	0.00	0	9	0.00
500k	0	1	0.00	0	3	0.00	0	4	0.00
1M	0*	0*	0.00*	0	1	0.00	—	—	—

Note: * Continuous transmission/reception is not possible.

The value set in BRR is given by the following formula:

$$N = \frac{\phi}{4 \times 2^{2n} \times B} - 1$$

B: Bit rate (bit/s)

N: BRR setting for baud rate generator ($0 \leq N \leq 255$)

ϕ: Operating frequency (Hz)

n: Baud rate generator input clock number (n = 0, 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	ϕ	0	0
0	ϕ_w*	0	1
2	ϕ/16	1	0
3	ϕ/64	1	1

Note: * In subactive or subsleep mode, the SCI3 can be operated only when the CPU operating clock is ϕ_w.

14.3.9 Serial Port Control Register (SPCR)

SPCR selects the function of the TXD3 (IrTXD) pin and whether to invert the input/output data of the RXD3 (IrRXD) and TXD3 (IrTXD) pins.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
5	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
4	SPC3	0	R/W	P32/TXD3/IrTXD Pin Function Switch Selects whether pin P32/TXD3/IrTXD is used as P32 or as TXD3/IrTXD. 0: P32 I/O pin 1: TXD3/IrTXD output pin Set the TE bit in SCR after setting this bit to 1.
3, 2	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
1	SCINV1	0	R/W	TXD3/IrTXD Pin Output Data Inversion Switch Selects whether output data of the TXD3/IrTXD pin is inverted or not. 0: Output data of TXD3/IrTXD pin is not inverted. 1: Output data of TXD3/IrTXD pin is inverted.
0	SCINV0	0	R/W	RXD3/IrRXD Pin Input Data Inversion Switch Selects whether input data of the RXD3/IrRXD pin is inverted or not. 0: Input data of RXD3/IrRXD pin is not inverted. 1: Input data of RXD3/IrRXD pin is inverted.

Note: When the serial port control register is modified, the data being input or output up to that point is inverted immediately after the modification, and an invalid data change is input or output. When modifying the serial port control register, modification must be made in a state in which data changes are invalidated.

14.3.10 IrDA Control Register (IrCR)

IrCR controls the IrDA operation of the SCI3.

Bit	Bit Name	Initial Value	R/W	Description
7	IrE	0	R/W	IrDA Enable Selects whether the SCI3 I/O pins function as the SCI3 or IrDA. 0: TXD3/IrTXD and RXD3/IrRXD pins function as TXD3 and RXD3 1: TXD3/IrTXD and RXD3/IrRXD pins function as IrTXD and IrRXD
6	IrCKS2	0	R/W	IrDA Clock Select If the IrDA function is enabled, these bits set the high-pulse width when encoding the IrTXD output pulse. 000: Bit rate × 3/16 001: ϕ/2 010: ϕ/4 011: ϕ/8 100: ϕ/16 101: Setting prohibited 11x: Setting prohibited
5	IrCKS1	0	R/W	
4	IrCKS0	0	R/W	
3 to 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

[Legend] x: Don't care.

14.3.11 Serial Extended Mode Register (SEMR)

SEMR sets the basic clock used in asynchronous mode.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	ABCS	0	R/W	Asynchronous Mode Basic Clock Select Selects the basic clock for the bit period in asynchronous mode. This setting is enabled only in asynchronous mode (COM bit in SMR3 is 0). 0: Operates on a basic clock with a frequency of 16 times the transfer rate 1: Operates on a basic clock with a frequency of eight times the transfer rate Clear the ABCS bit to 0, when the IrDA function is enabled.
2 to 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

14.4 Operation in Asynchronous Mode

Figure 14.2 shows the general format for asynchronous serial communication. One frame consists of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low level), and finally stop bits (high level). In asynchronous mode, synchronization is performed at 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. When the ABCS bit in SEMR is 1, the data is sampled on the 4th pulse of a clock with a frequency eight times the bit period. Inside the SCI3, the transmitter and receiver are independent units, enabling full duplex. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer. Table 14.8 shows the 16 data transfer formats that can be set in asynchronous mode. The format is selected by the settings in SMR as shown in table 14.9.

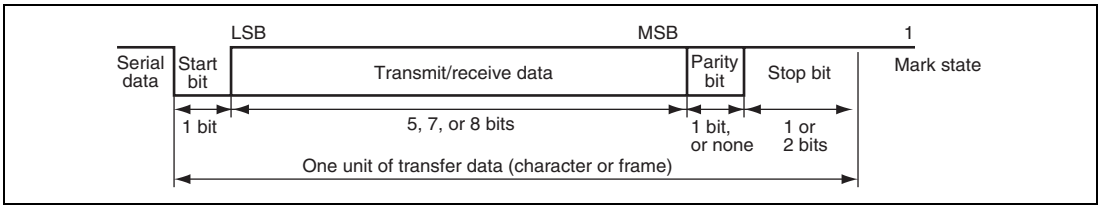


Figure 14.2 Data Format in Asynchronous Communication

14.4.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK3 pin can be selected as the SCI3's serial clock source, according to the setting of the COM bit in SMR and the CKE0 and CKE1 bits in SCR. When an external clock is input at the SCK3 pin, the clock frequency should be 16 times the bit rate used (when the ABCS bit in SEMR is 1, the clock frequency should be eight times the bit rate used). For details on selection of the clock source, see table 14.10. When the SCI3 is operated on an internal clock, the clock can be output from the SCK3 pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transfer data, as shown in figure 14.3.

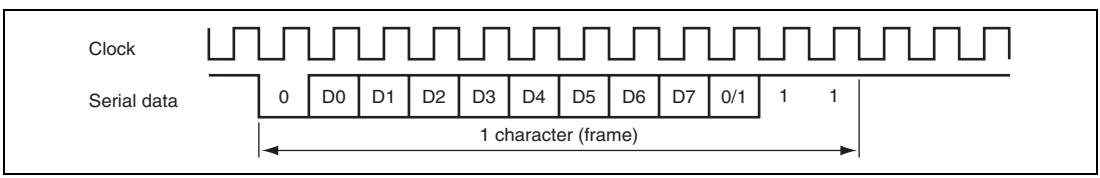


Figure 14.3 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode) (Example with 8-Bit Data, Parity, Two Stop Bits)

Table 14.8 Data Transfer Formats (Asynchronous Mode)

SMR				Serial Data Transfer Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	START 8-bit data STOP
0	0	0	1	START 8-bit data STOP STOP
0	0	1	0	Setting prohibited
0	0	1	1	Setting prohibited
0	1	0	0	START 8-bit data P STOP
0	1	0	1	START 8-bit data P STOP STOP
0	1	1	0	START 5-bit data STOP
0	1	1	1	START 5-bit data STOP STOP
1	0	0	0	START 7-bit data STOP
1	0	0	1	START 7-bit data STOP STOP
1	0	1	0	Setting prohibited
1	0	1	1	Setting prohibited
1	1	0	0	START 7-bit data P STOP
1	1	0	1	START 7-bit data P STOP STOP
1	1	1	0	START 5-bit data P STOP
1	1	1	1	START 5-bit data P STOP STOP

[Legend]

START: Start bit
STOP: Stop bit
P: Parity bit

Table 14.9 SMR Settings and Corresponding Data Transfer Formats

SMR						Data Transfer Format		
Bit 7	Bit 6	Bit 2	Bit 5	Bit 3				
COM	CHR	MP	PE	STOP	Mode	Data Length	Parity Bit	Stop Bit Length
0	0	0	0	0	Asynchronous mode	8-bit data	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	0		Setting prohibited		
				1				
			1	0		5-bit data	No	1 bit
				1				2 bits
	1		0	0		Setting prohibited		
				1				
			1	0		5-bit data	Yes	1 bit
				1				2 bits
1	x	0	x	x	Clock synchronous mode	8-bit data	No	No

[Legend] x: Don’t care.

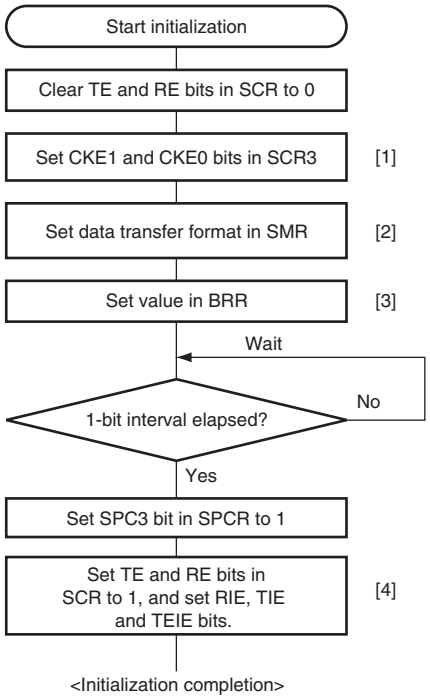
Table 14.10 SMR and SCR Settings and Clock Source Selection

SMR	SCR				
Bit 7	Bit 1	Bit 0		Transmit/Receive Clock	
COM	CKE1	CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Asynchronous mode	Internal	I/O port (SCK3 pin not used)
		1			Outputs a clock with the same frequency as the bit rate
	1	0		External	Inputs a clock with a frequency 16 times the bit rate*
1	0	0	Clock synchronous mode	Internal	Outputs the serial clock
	1	0		External	Inputs the serial clock
0	1	1	Reserved (Do not specify these combinations)		
1	0	1			
1	1	1			

Note: * When the ABCS bit in SEMR is 1, inputs a clock with a frequency eight times the bit rate.

14.4.2 SCI3 Initialization

Follow the flowchart as shown in figure 14.4 to initialize the SCI3. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not initialize the contents of the RDRF, PER, FER, and OER flags, or the contents of RDR. When the external clock is used in asynchronous mode, the clock must be supplied even during initialization. When the external clock is used in clock synchronous mode, the clock must not be supplied during initialization.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock output is selected in asynchronous mode, clock is output immediately after CKE1 and CKE0 settings are made. When the clock output is selected at reception in clock synchronous mode, clock is output immediately after CKE1, CKE0, and RE are set to 1.

[2] Set the data transfer format in SMR.

[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Setting bits TE and RE enables the TXD3 and RXD3 pins to be used. Also set the RIE, TIE, and TEIE bits, depending on whether interrupts are required. In asynchronous mode, the bits are marked at transmission and idled at reception to wait for the start bit.

Figure 14.4 Sample SCI3 Initialization Flowchart

14.4.3 Data Transmission

Figure 14.5 shows an example of operation for transmission in asynchronous mode. In transmission, the SCI3 operates as described below.

1. The SCI3 monitors the TDRE flag in SSR. If the flag is cleared to 0, the SCI3 recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI3 sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a TXI3 interrupt request is generated. Continuous transmission is possible because the TXI3 interrupt routine writes next transmit data to TDR before transmission of the current transmit data has been completed.
3. The SCI3 checks the TDRE flag at the timing for sending the stop bit.
4. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.
5. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the “mark state” is entered, in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.
6. Figure 14.6 shows a sample flowchart for transmission in asynchronous mode.

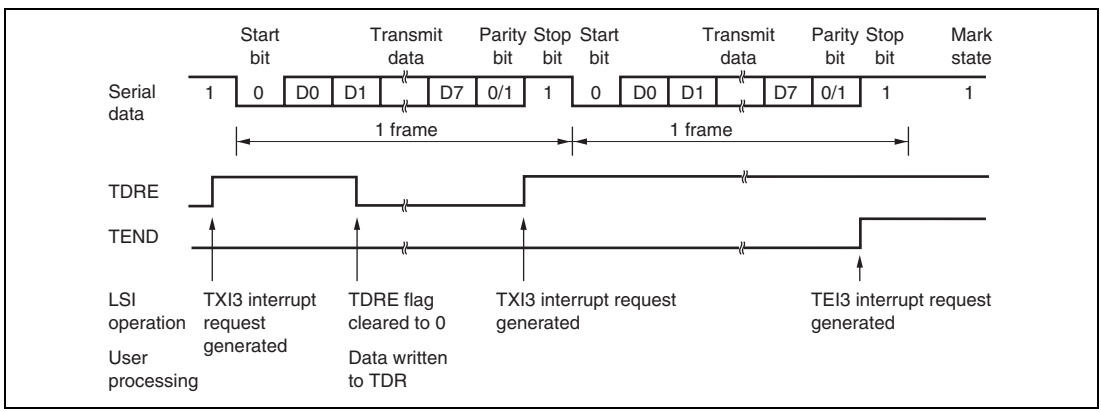
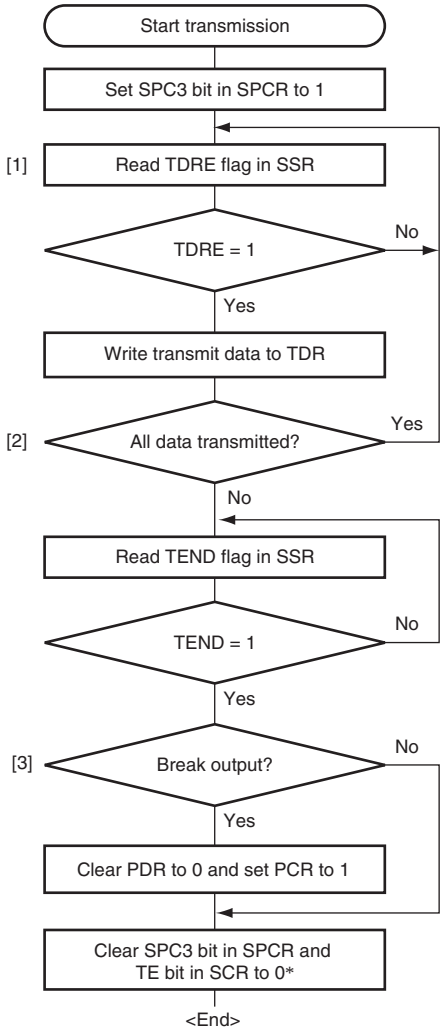


Figure 14.5 Example SCI3 Operation in Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)



[1] Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. When data is written to TDR, the TDRE flag is automatically cleared to 0. (After the TE bit is set to 1, one frame of 1 is output, then transmission is possible.)

[2] To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR. When data is written to TDR, the TDRE flag is automatically cleared to 0.

[3] To output a break in serial transmission, after setting PCR to 1 and PDR to 0, clear the SPC3 bit in SPCR and the TE bit in SCR to 0.

Note: * When the SPC3 bit in SPCR is cleared to 0, the pin functions as an I/O port.

Figure 14.6 Sample Serial Transmission Flowchart (Asynchronous Mode)

14.4.4 Serial Data Reception

Figure 14.7 shows an example of operation for reception in asynchronous mode. In serial reception, the SCI3 operates as described below.

1. The SCI3 monitors the communication line. If a start bit is detected, the SCI3 performs internal synchronization, receives data in RSR, and checks the parity bit and stop bit.

- Parity check
  The 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
  The SCI3 checks that the stop bit is 1. If two stop bits are used, only the first is checked.
- Status check
  The SCI3 checks that bit RDRF is set to 0, indicating that the receive data can be transferred from RSR to RDR.

2. If an overrun error occurs (when reception of the next data is completed while the RDRF flag is still set to 1), the OER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI3 interrupt request is generated. Receive data is not transferred to RDR.
3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI3 interrupt request is generated.
4. If a framing error is detected (when the stop bit is 0), the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI3 interrupt request is generated.
5. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI3 interrupt request is generated. Continuous reception is possible because the RXI3 interrupt routine reads the receive data transferred to RDR before reception of the next receive data has been completed.

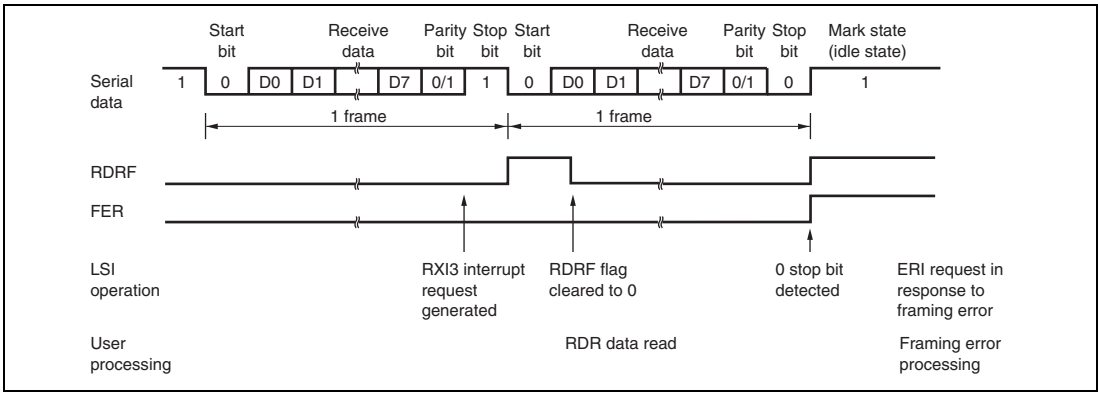


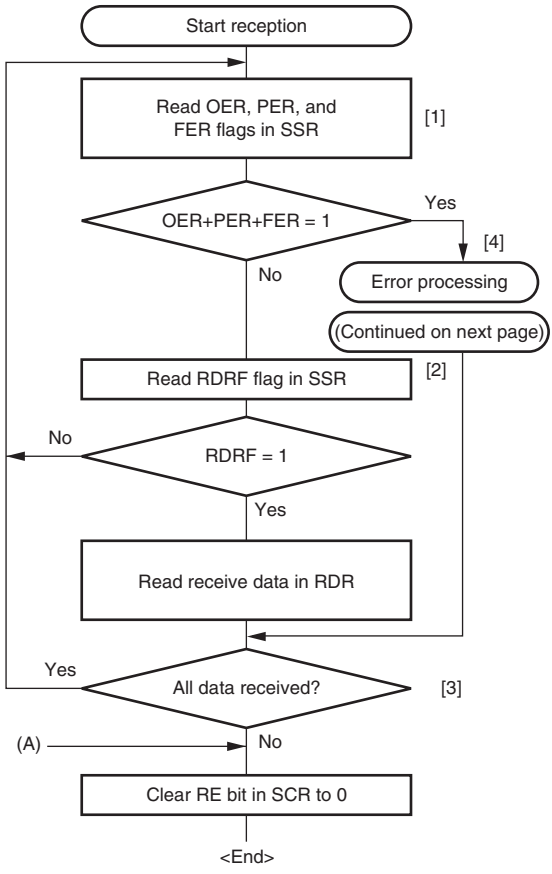
Figure 14.7 Example SCI3 Operation in Reception in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)

Table 14.11 shows the states of the SSR status flags and receive data handling when a receive error is detected. If a receive error is detected, the RDRF flag retains its state before receiving data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the OER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 14.8 shows a sample flowchart for serial data reception.

Table 14.11 SSR Status Flags and Receive Data Handling

SSR Status Flag					
RDRF*	**OER**	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	1	0	0	Lost	Overrun error
0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + framing error
1	1	0	1	Lost	Overrun error + parity error
0	0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	1	Lost	Overrun error + framing error + parity error

Note: * The RDRF flag retains the state it had before data reception. However, note that if RDR is read after an overrun error has occurred in a frame because reading of the receive data in the previous frame was delayed, the RDRF flag will be cleared to 0.



[1] Read the OER, PER, and FER flags in SSR to identify the error. If a receive error occurs, performs the appropriate error processing.

[2] Read SSR and check that RDRF = 1, then read the receive data in RDR. The RDRF flag is cleared automatically.

[3] To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag and read RDR.
The RDRF flag is cleared automatically.

[4] If a receive error occurs, read the OER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the OER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RXD3 pin.

Figure 14.8 Sample Serial Data Reception Flowchart (Asynchronous Mode) (1)

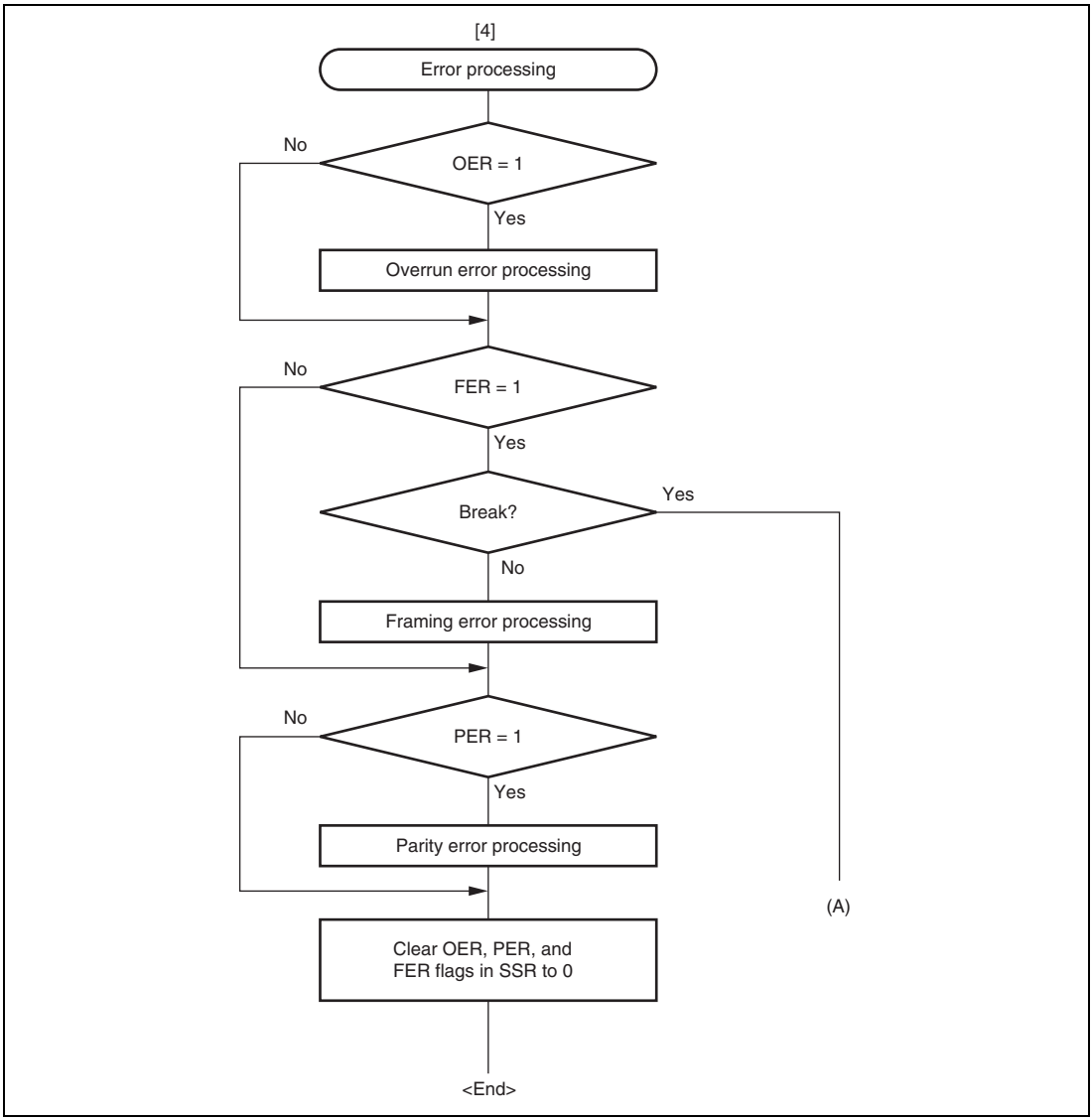


Figure 14.8 Sample Serial Data Reception Flowchart (Asynchronous Mode) (2)

14.5 Operation in Clock Synchronous Mode

Figure 14.9 shows the general format for clock synchronous communication. In clock synchronous mode, data is transmitted or received synchronous with clock pulses. A single character in the transmit data consists of the 8-bit data starting from the LSB. In clock synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. In clock synchronous mode, the SCI3 receives data in synchronous with the rising edge of the serial clock. After 8-bit data is output, the transmission line holds the MSB state. In clock synchronous mode, no parity bit is added. Inside the SCI3, the transmitter and receiver are independent units, enabling full-duplex communication through the use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer.

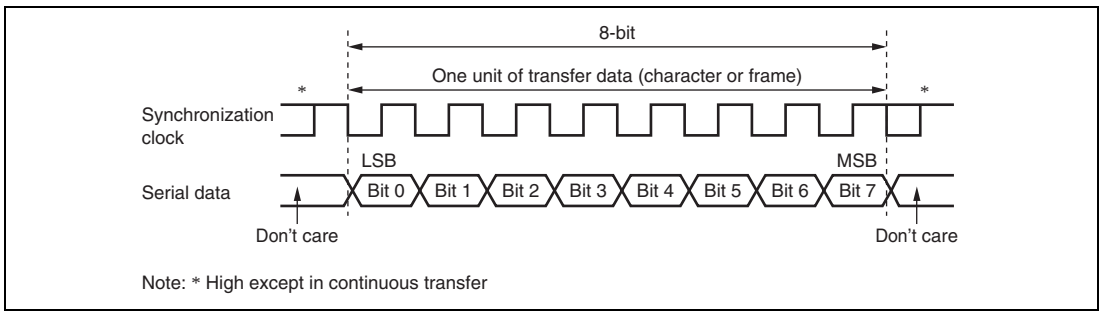


Figure 14.9 Data Format in Clock Synchronous Communication

14.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK3 pin can be selected, according to the setting of the COM bit in SMR and CKE0 and CKE1 bits in SCR. When the SCI3 is operated on an internal clock, the serial clock is output from the SCK3 pin. Eight serial clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high.

14.5.2 SCI3 Initialization

Before transmitting and receiving data, the SCI3 should be initialized as described in a sample flowchart in figure 14.4.

14.5.3 Serial Data Transmission

Figure 14.10 shows an example of SCI3 operation for transmission in clock synchronous mode. In serial transmission, the SCI3 operates as described below.

1. The SCI3 monitors the TDRE flag in SSR, and if the flag is 0, the SCI3 recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. The SCI3 sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a TXI3 interrupt request is generated.
3. 8-bit data is sent from the TXD3 pin synchronized with the output clock when output clock mode has been specified, and synchronized with the input clock when use of an external clock has been specified. Serial data is transmitted sequentially from the LSB (bit 0), from the TXD3 pin.
4. The SCI3 checks the TDRE flag at the timing for sending the MSB (bit 7).
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TDRE flag maintains the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI3 interrupt request is generated.
7. The SCK3 pin is fixed high.

Figure 14.11 shows a sample flowchart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (OER, FER, or PER) is set to 1. Make sure that the receive error flags are cleared to 0 before starting transmission.

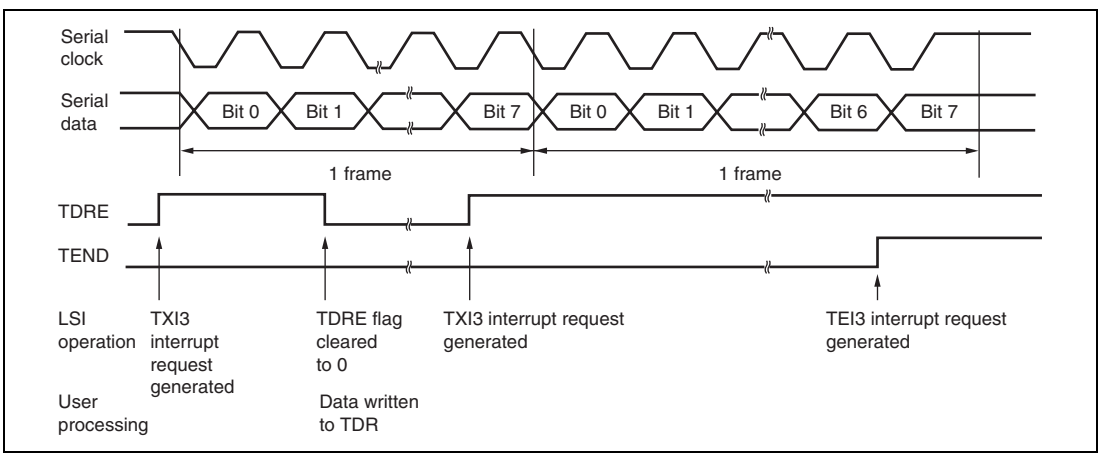


Figure 14.10 Example of SCI3 Operation in Transmission in Clock Synchronous Mode

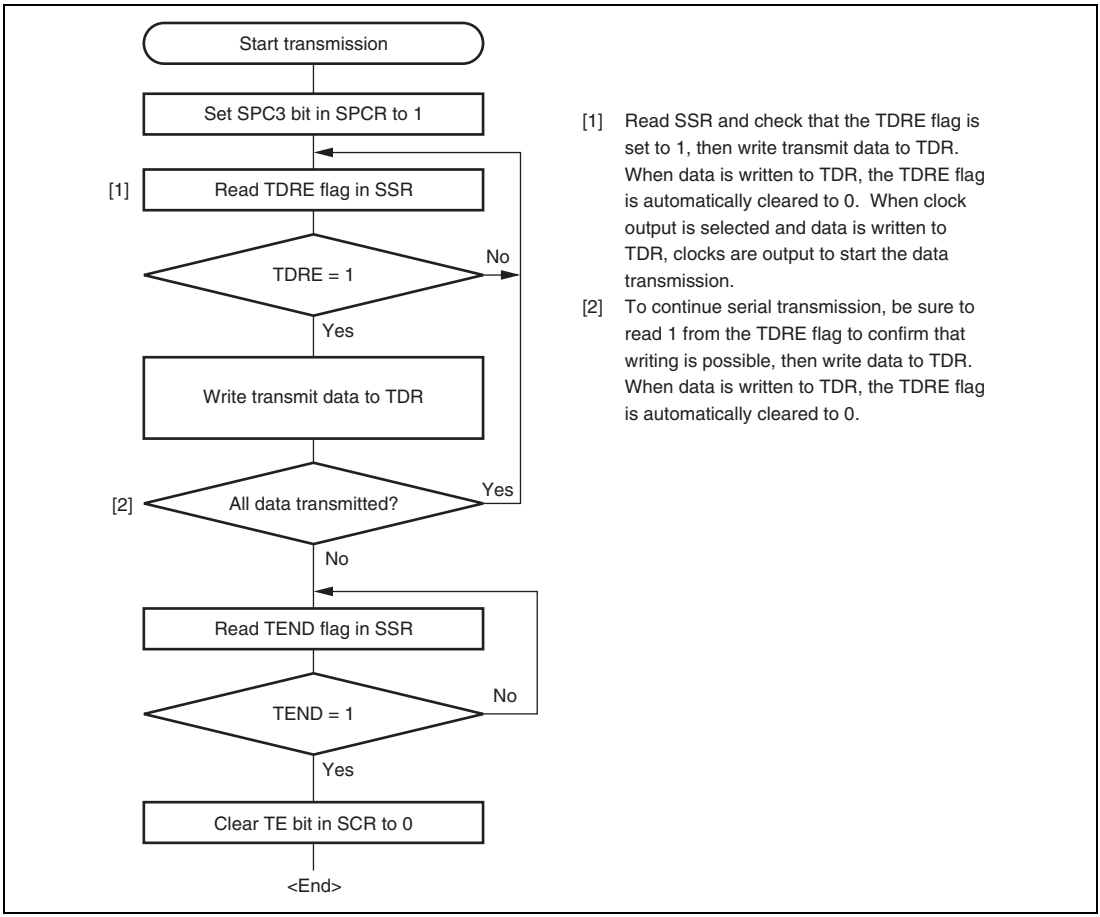


Figure 14.11 Sample Serial Transmission Flowchart (Clock Synchronous Mode)

14.5.4 Serial Data Reception (Clock Synchronous Mode)

Figure 14.12 shows an example of SCI3 operation for reception in clock synchronous mode. In serial reception, the SCI3 operates as described below.

1. The SCI3 performs internal initialization synchronous with a synchronous clock input or output, starts receiving data.
2. The SCI3 stores the received data in RSR.
3. If an overrun error occurs (when reception of the next data is completed while the RDRF flag in SSR is still set to 1), the OER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI3 interrupt request is generated, receive data is not transferred to RDR, and the RDRF flag remains to be set to 1.
4. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI3 interrupt request is generated.

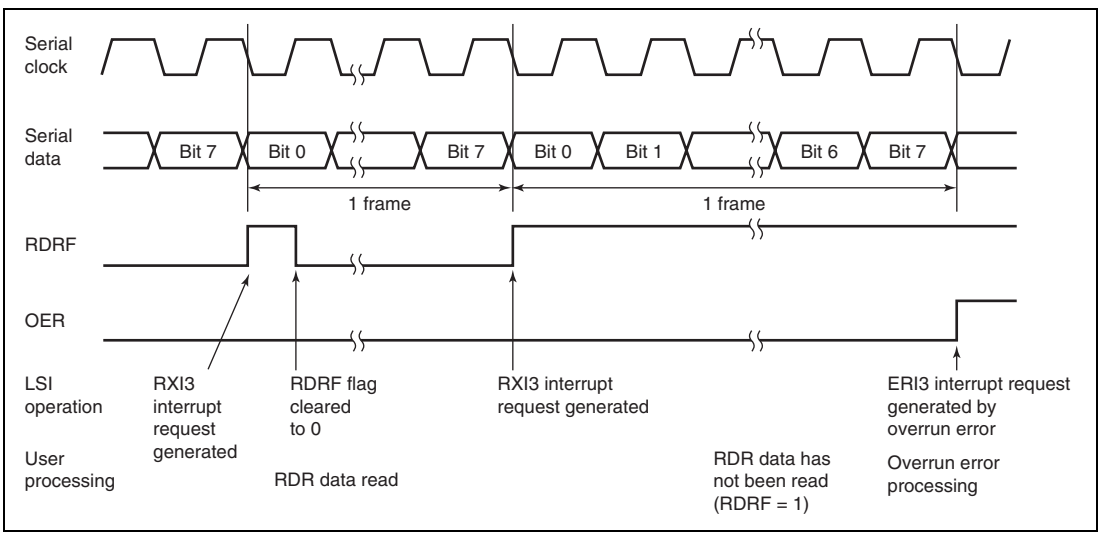
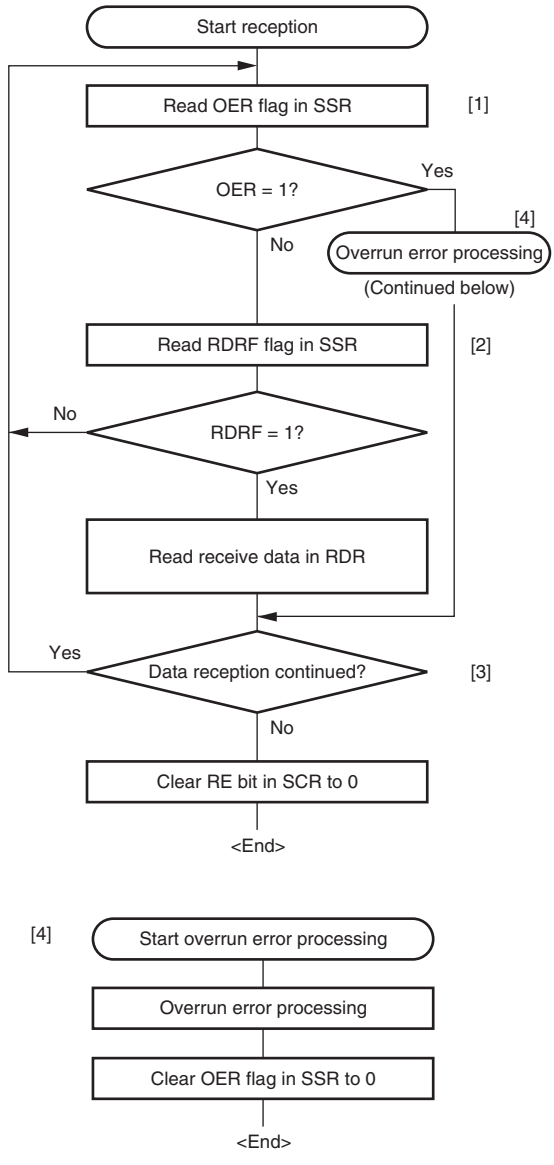


Figure 14.12 Example of SCI3 Reception Operation in Clock Synchronous Mode

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the OER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 14.13 shows a sample flowchart for serial data reception.



[1] Read the OER flag in SSR to determine if there is an error. If an overrun error has occurred, execute overrun error processing.

[2] Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR. When data is read from RDR, the RDRF flag is automatically cleared to 0.

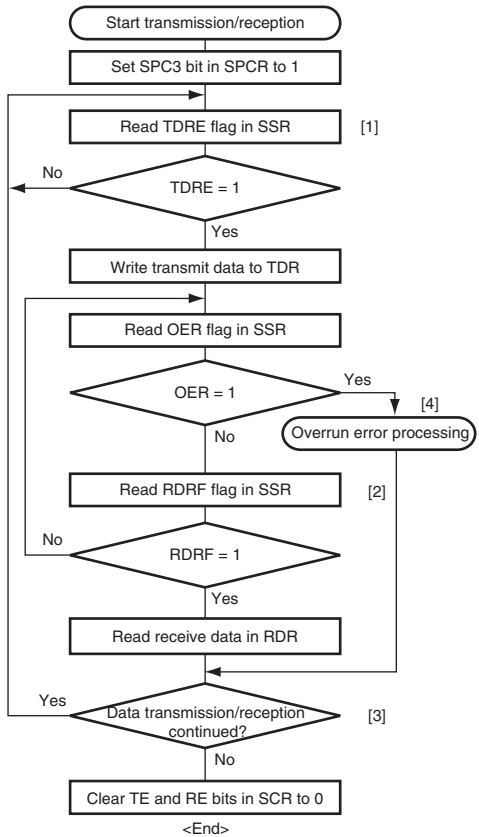
[3] To continue serial reception, before the MSB (bit 7) of the current frame is received, reading the RDRF flag and reading RDR should be finished. When data is read from RDR, the RDRF flag is automatically cleared to 0.

[4] If an overrun error occurs, read the OER flag in SSR, and after performing the appropriate error processing, clear the OER flag to 0. Reception cannot be resumed if the OER flag is set to 1.

Figure 14.13 Sample Serial Reception Flowchart (Clock Synchronous Mode)

14.5.5 Simultaneous Serial Data Transmission and Reception

Figure 14.14 shows a sample flowchart for simultaneous serial transmit and receive operations. The following procedure should be used for simultaneous serial data transmit and receive operations. To switch from transmit mode to simultaneous transmit and receive mode, after checking that the SCI3 has finished transmission and the TDRE and TEND flags are set to 1, clear TE to 0. Then simultaneously set TE and RE to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI3 has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (OER, FER, and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.



[1] Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR.
When data is written to TDR, the TDRE flag is automatically cleared to 0.

[2] Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR.
When data is read from RDR, the RDRF flag is automatically cleared to 0.

[3] To continue serial transmission/ reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR.
When data is written to TDR, the TDRE flag is automatically cleared to 0. When data is read from RDR, the RDRF flag is automatically cleared to 0.

[4] If an overrun error occurs, read the OER flag in SSR, and after performing the appropriate error processing, clear the OER flag to 0.
Transmission/reception cannot be resumed if the OER flag is set to 1.
For overrun error processing, see figure 14.13.

Figure 14.14 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations (Clock Synchronous Mode)

14.6 IrDA Operation

IrDA operation can be used with the SCI3. Figure 14.15 shows an IrDA block diagram.

If the IrDA function is enabled using the IrE bit in IrCR, the TXD3 and RXD3 pins in the SCI3 are allowed to encode and decode the waveform based on the IrDA standard version 1.0 (function as the IrTXD and IrRXD pins). Connecting these pins to the infrared data transceiver/receiver achieves infrared data communication based on the system defined by the IrDA standard version 1.0.

In the system defined by the IrDA standard version 1.0, communication is started at a transfer rate of 9600 bps, which can be modified as required. The IrDA interface provided by this LSI does not incorporate the capability of automatic modification of the transfer rate; the transfer rate must be modified through programming.

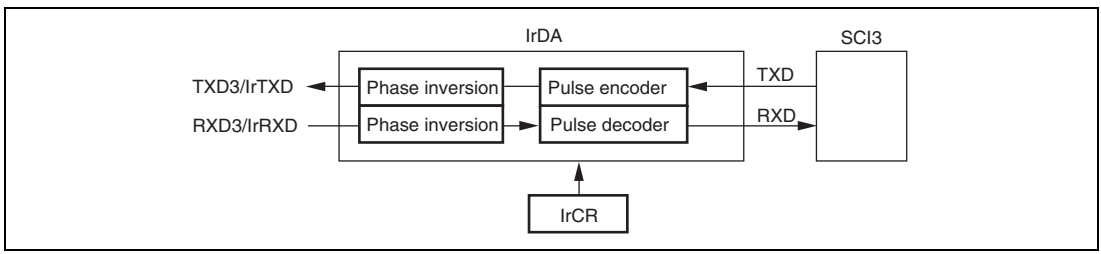


Figure 14.15 IrDA Block Diagram

14.6.1 Transmission

During transmission, the output signals from the SCI3 (UART frames) are converted to IR frames using the IrDA interface (see figure 14.16).

For serial data of level 0, a high-level pulse having a width of 3/16 of the bit rate (1-bit interval) is output (initial setting). The high-level pulse can be selected using the IrCKS2 to IrCKS0 bits in IrCR.

According to the standard, the high-level pulse width is defined to be 1.41 μs at minimum and (3/16 + 2.5%) × bit rate or (3/16 × bit rate) +1.08 μs at maximum. For example, when the frequency of system clock ϕ is 10 MHz, being equal to or greater than 1.41 μs, the high-level pulse width at minimum can be specified as 1.6 μs.

For serial data of level 1, no pulses are output.

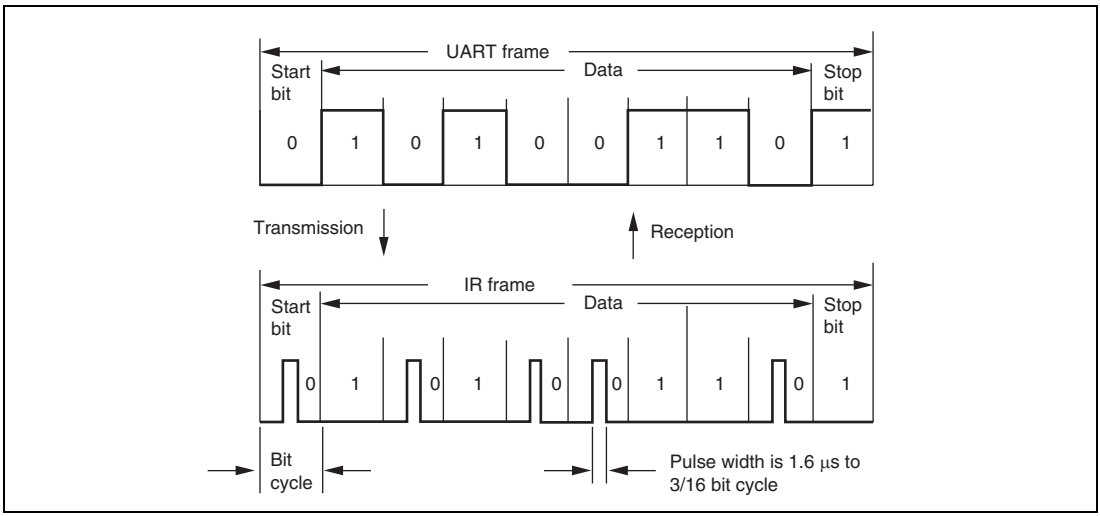


Figure 14.16 IrDA Transmission and Reception

14.6.2 Reception

During reception, IR frames are converted to UART frames using the IrDA interface before inputting to the SCI3.

Data of level 0 is output each time a high-level pulse is detected and data of level 1 is output when no pulse is detected in a bit cycle. If a pulse has a high-level width of less than 1.41 μs, the minimum width allowed, the pulse is recognized as level 0.

14.6.3 High-Level Pulse Width Selection

Table 14.12 shows possible settings for bits IrCKS2 to IrCKS0 (minimum pulse width), and this LSI's operating frequencies and bit rates, for making the pulse width shorter than 3/16 times the bit rate in transmission.

Table 14.12 IrCKS2 to IrCKS0 Bit Settings

Operating Frequency	Bit Rate (bps) (Upper Row) / Bit Interval × 3/16 (μs) (Lower Row)			
	2400	9600	19200	38400
ϕ (MHz)	78.13	19.53	9.77	4.88
2	010	010	010	010
2.097152	010	010	010	010
2.4576	010	010	010	010
3	011	011	011	011
3.6864	011	011	011	011
4.9152	011	011	011	011
5	011	011	011	011
6	100	100	100	100
6.144	100	100	100	100
7.3728	100	100	100	100
8	100	100	100	100
9.8304	100	100	100	100
10	100	100	100	100

14.7 Interrupt Requests

The SCI3 creates the following six interrupt requests: transmit end, transmit data empty, receive data full, and receive errors (overrun error, framing error, and parity error). Table 14.13 shows the interrupt sources.

Table 14.13 SCI3 Interrupt Requests

Interrupt Requests	Abbreviation	Interrupt Sources
Receive Data Full	RXI	Setting RDRF in SSR
Transmit Data Empty	TXI	Setting TDRE in SSR
Transmission End	TEI	Setting TEND in SSR
Receive Error	ERI	Setting OER, FER, and PER in SSR

Each interrupt request can be enabled or disabled by means of bits TIE and RIE in SCR.

When the TDRE bit in SSR is set to 1, a TXI3 interrupt is requested. When the TEND bit in SSR is set to 1, a TEI3 interrupt is requested. These two interrupts are generated during transmission.

The initial value of the TDRE flag in SSR is 1. Thus, when the TIE bit in SCR is set to 1 before transferring the transmit data to TDR, a TXI3 interrupt request is generated even if the transmit data is not ready. The initial value of the TEND flag in SSR is 1. Thus, when the TEIE bit in SCR is set to 1 before transferring the transmit data to TDR, a TEI3 interrupt request is generated even if the transmit data has not been sent. It is possible to make use of the most of these interrupt requests efficiently by transferring the transmit data to TDR in the interrupt routine. To prevent the generation of these interrupt requests (TXI3 and TEI3), clear the enable bits (TIE and TEIE) that correspond to these interrupt requests to 0, after transferring the transmit data to TDR.

When the RDRF bit in SSR is set to 1, an RXI3 interrupt is requested, and if any of bits OER, PER, and FER is set to 1, an ERI3 interrupt is requested. These two interrupt requests are generated during reception.

The SCI3 can carry out continuous reception using an RXI3 and continuous transmission using a TXI3.

These interrupts are shown in table 14.14.

Table 14.14 Transmit/Receive Interrupts

Interrupt	Flags	Interrupt Request Conditions	Notes
RXI3	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, an RXI3 is enabled and an interrupt is requested. (See figure 14.17 (a).)	The RXI3 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.
TXI3	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, a TXI3 is enabled and an interrupt is requested. (See figure 14.17 (b).)	The TXI3 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.
TEI31	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, a TEI3 is enabled and an interrupt is requested. (See figure 14.17 (c).)	A TEI3 indicates that the next transmit data has not been written to TDR when the last bit of the transmit character in TSR is transmitted.

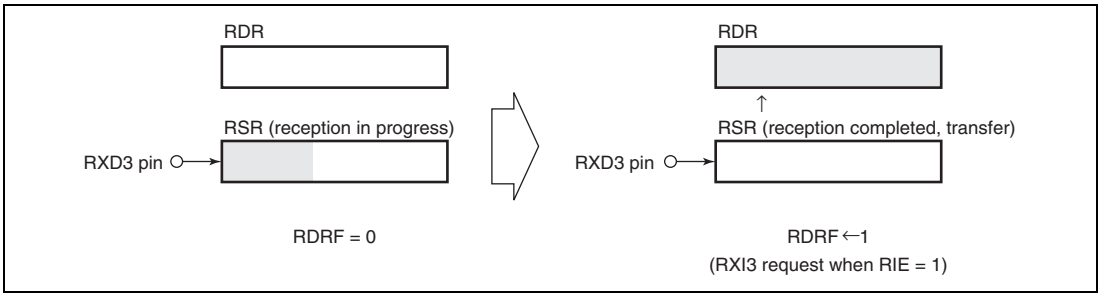


Figure 14.17 (a) RDRF Setting and RXI Interrupt

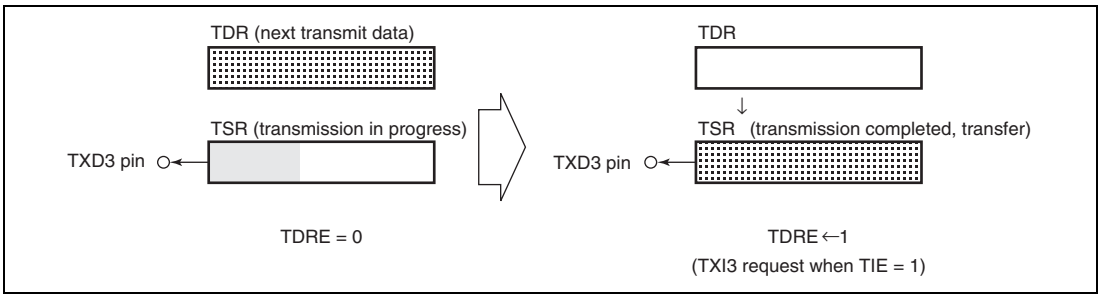


Figure 14.17 (b) TDRE Setting and TXI Interrupt

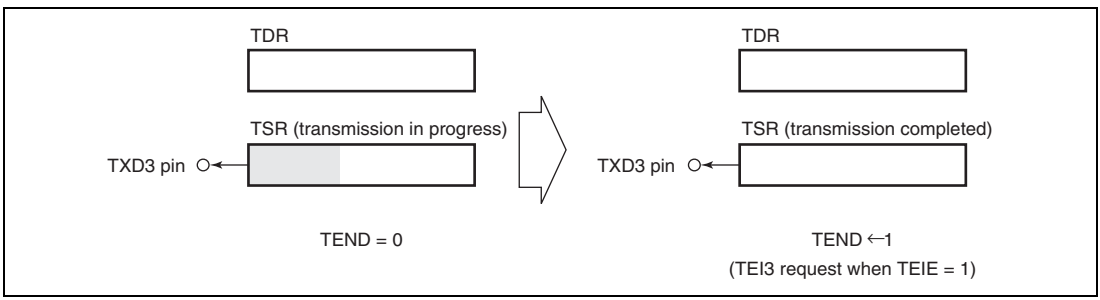


Figure 14.17 (c) TEND Setting and TEI Interrupt

14.8 Usage Notes

14.8.1 Break Detection and Processing

When framing error detection is performed, a break can be detected by reading the RXD3 pin value directly. In a break, the input from the RXD3 pin becomes all 0, setting the FER flag, and possibly the PER flag. Note that as the SCI3 continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

14.8.2 Mark State and Break Sending

When the SPC3 bit in SPCR is 0, the TXD3 pin functions as an I/O port whose direction (input or output) and level are determined by PCR and PDR, regardless of the TE setting. This can be used to set the TXD3 pin to the mark state (high level) or send a break during data transmission. To maintain the communication line at the mark state until the SPC3 bit in SPCR is set to 1, set both PCR and PDR to 1. As the SPC3 bit in SPCR is cleared to 0 at this point, the TXD3 pin functions as an I/O port, and 1 is output from the TXD3 pin. To send a break during data transmission, first set PCR to 1 and PDR to 0, and then clear the SPC3 and TE bits to 0. When the TE bit is cleared to 0 directly after the SPC3 bit is cleared to 0, the transmitter is initialized regardless of the current transmission state after the TE bit is cleared, the TXD3 pin functions as an I/O port after the SPC3 bit is cleared, and 0 is output from the TXD3 pin.

14.8.3 Receive Error Flags and Transmit Operations (Clock Synchronous Mode Only)

Transmission cannot be started when a receive error flag (OER, PER, or FER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to 0.

14.8.4 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI3 operates on a basic clock with a frequency of 16 times the transfer rate. In reception, the SCI3 samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 8th pulse of the basic clock as shown in figure 14.18.

Thus, the reception margin in asynchronous mode is given by formula (1) below.

$$M = \left\{ (0.5 - \frac{1}{2N}) - \frac{D - 0.5}{N} - (L - 0.5)\,F \right\} \times 100(\%) \quad \text{... Formula (1)}$$

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 rate deviation

Assuming values of F (absolute value of clock rate deviation) = 0 and D (clock duty) = 0.5 in formula (1), the reception margin can be given by the formula.

$$M = \{0.5 - 1/(2 \times 16)\} \times 100\ [\%] = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed for in system design.

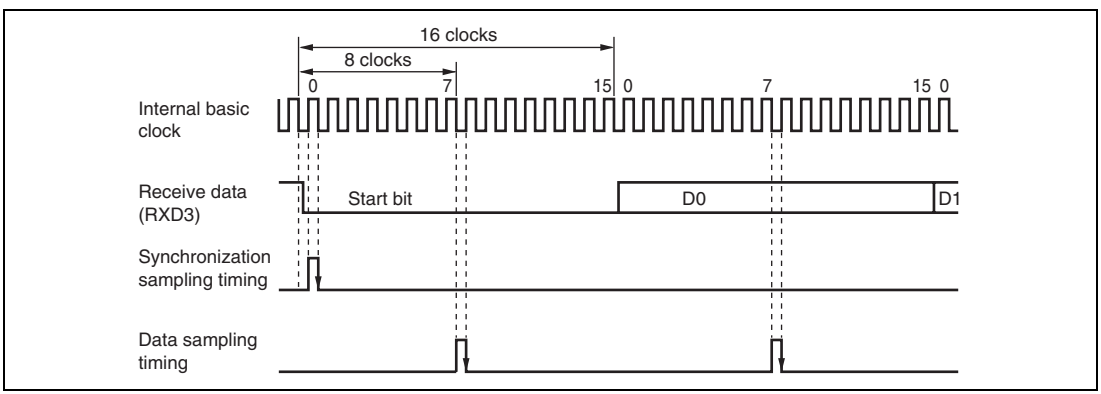


Figure 14.18 Receive Data Sampling Timing in Asynchronous Mode

14.8.5 Note on Switching SCK3 Pin Function

If the SCK3 pin is used as a clock output pin by the SCI3 in clock synchronous mode and is then switched to a general input/output pin (a pin with a different function), the SCK3 pin outputs a low level signal for half a system clock (ϕ) cycle immediately after it is switched.

This can be prevented by either of the following methods according to the situation.

(1) When SCK3 Pin Function is Switched from Clock Output to Non Clock-Output

When stopping data transfer, issue one instruction to clear bits TE and RE to 0 and to set bits CKE1 and CKE0 in SCR to 1 and 0, respectively.

In this case, bit COM in SMR should be left 1. The above prevents the SCK3 pin from being used as a general input/output pin. To avoid an intermediate level of voltage from being applied to the SCK3 pin, the line connected to the SCK3 pin should be pulled up to the V_{CC} level via a resistor, or supplied with output from an external device.

(2) When SCK3 Pin Function is Switched from Clock Output to General Input/Output

When stopping data transfer,

1. Issue one instruction to clear bits TE and RE to 0 and to set bits CKE1 and CKE0 in SCR to 1 and 0, respectively.
2. Clear bit COM in SMR to 0
3. Clear bits CKE1 and CKE0 in SCR to 0. Note that special care is also needed here to avoid an intermediate level of voltage from being applied to the SCK3 pin.

14.8.6 Relation between Writing 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 the 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 if 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 only once (not two or more times).

14.8.7 Relation between RDR Reading and bit RDRF

In a receive operation, the 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 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 shown in figure 14.19.

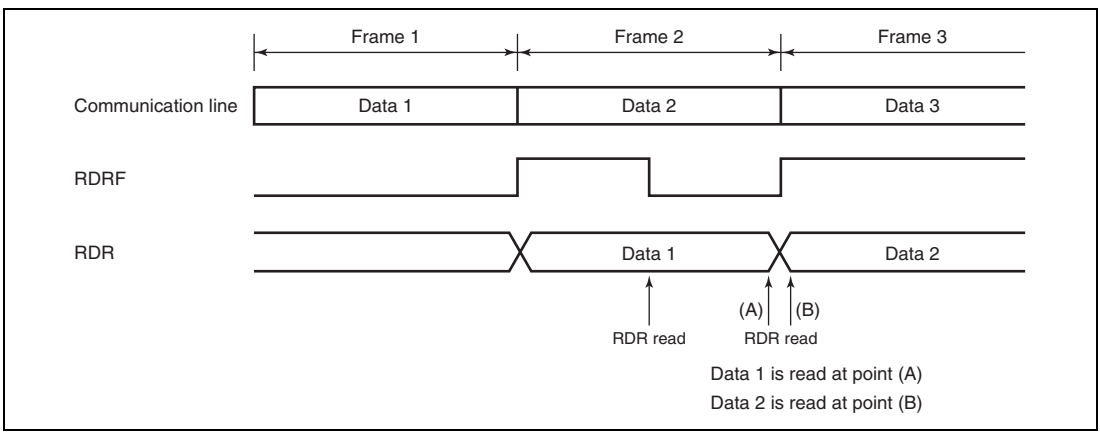


Figure 14.19 Relation between RDR Read Timing and Data

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 clock synchronous mode, or before the STOP bit is transferred in asynchronous mode.

14.8.8 Transmit and Receive Operations when Making State Transition

Make sure that transmit and receive operations have completely finished before carrying out state transition processing.

14.8.9 Setting in Subactive or Subsleep Mode

In subactive or subsleep mode, the SCI3 can operate only when the CPU clock is ϕ_w. The SA1 and SA0 bits in SYSCR2 should be set to 1.

14.8.10 Oscillator when Serial Communication Interface 3 is Used

When serial communication interface 3 is used, the system clock oscillator or subclock oscillator must be used. Do not use the on-chip oscillator. For details on selecting the system clock oscillator or on-chip oscillator, see section 4.2.4, On-Chip Oscillator Selection Method. For details on selecting the subclock oscillator or on-chip oscillator, see section 4.1.1, Oscillator Control Register (OSCCR).

Section 15 Synchronous Serial Communication Unit (SSU)

The synchronous serial communication unit (SSU) can handle clocked synchronous serial data communication.

Figure 15.1 shows a block diagram of the SSU.

15.1 Features

- Can be operated in clocked synchronous communication mode or four-line bus communication mode (including bidirectional communication mode)
- Can be operated as a master or a slave device
- Choice of eight internal clocks (ϕ/256, ϕ/128, ϕ/64, ϕ/32, ϕ/16, ϕ/8, ϕ/4, and ϕ_{SUB}/2) and an external clock as a clock source
- Clock polarity and phase of SSCK can be selected
- Choice of data transfer direction (MSB-first or LSB-first)
- Receive error detection: overrun error
- Multimaster error detection: conflict error
- Five interrupt sources: transmit-end, transmit-data-empty, receive-data-full, overrun error, and conflict error
- Continuous transmission and reception of serial data are enabled since both transmitter and receiver have buffer structure
- Use of module standby mode enables this module to be placed in standby mode independently when not used. (The SSU is halted as the initial value. For details, refer to section 5.4, Module Standby Function.)

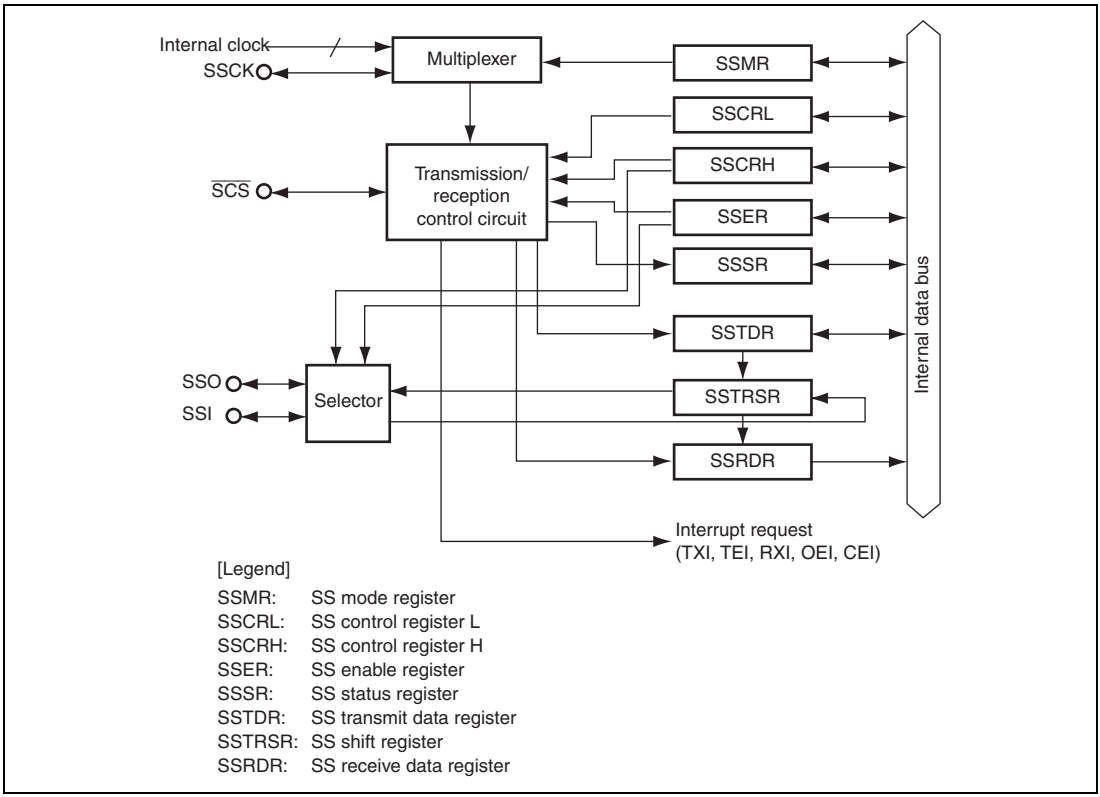


Figure 15.1 Block Diagram of SSU

15.2 Input/Output Pins

Table 15.1 shows the pin configuration of the SSU.

Table 15.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
SSU clock	SSCK	I/O	SSU clock input/output
SSU data input/output	SSI	I/O	SSU data input/output
SSU data input/output	SSO	I/O	SSU data input/output
SSU chip select input/output	$\overline{\text{SCS}}$	I/O	SSU chip select input/output

15.3 Register Descriptions

The SSU has the following registers.

- SS control register H (SSCRH)
- SS control register L (SSCRL)
- SS mode register (SSMR)
- SS enable register (SSER)
- SS status register (SSSR)
- SS receive data register (SSRDR)
- SS transmit data register (SSTDR)
- SS shift register (SSTRSR)

15.3.1 SS Control Register H (SSCRH)

SSCRH is a register that selects a master or a slave device, enables bidirectional mode, selects open-drain output of the serial data output pin, selects an output value of the serial data output pin, selects the SSCK pin, and selects the $\overline{\text{SCS}}$ pin.

Bit	Bit Name	Initial Value	R/W	Description
7	MSS	0	R/W	Master/Slave Device Select Selects whether this module is used as a master device or a slave device. When this module is used as a master device, transfer clock is output from the SSCK pin. When the CE bit in SSSR is set, this bit is automatically cleared. 0: Operates as a slave device 1: Operates as a master device
6	BIDE	0	R/W	Bidirectional Mode Enable Selects whether the serial data input pin and the output pin are both used or only one pin is used. For details, refer to section 15.4.3, Relationship between Data Input/Output and Shift Register. When the SSUMS bit in SSCRL is 0, this setting is invalid. 0: Normal mode. Communication is performed by using two pins. 1: Bidirectional mode. Communication is performed by using only one pin.

Bit	Bit Name	Initial Value	R/W	Description
5	SOOS	0	R/W	Serial Data Open-Drain Output Select Selects whether the serial data output pin is CMOS output or NMOS open-drain output. The serial data output pin is changed according to the register setting value. For details, refer to section 15.4.3, Relationship between Data Input/Output and Shift Register. 0: CMOS output 1: NMOS open-drain output
4	SOL	0	R/W	Serial Data Output Level Setting Although the value in the last bit of transmit data is retained in the serial data output after the end of transmission, the output level of serial data can be changed by manipulating this bit before or after transmission. When the output level is changed, the SOLP bit should be cleared to 0 and the MOV instruction should be used. If this bit is written during data transfer, erroneous operation may occur. Therefore this bit must not be manipulated during transmission. 0: Shows serial data output level to low in reading. Changes serial data output level to low in writing 1: Shows serial data output level to high in reading. Changes serial data output level to high in writing
3	SOLP	1	R/W	SOL Write Protect When output level of serial data is changed, the MOV instruction is used to set the SOL bit to 1 and clear this bit to 0 or to clear the SOL bit and this bit to 0. 0: In writing, output level can be changed according to the value of the SOL bit. 1: In reading, this bit is always read as 1. In writing, output level cannot be changed. (See section 15.5, Usage Note.)

Bit	Bit Name	Initial Value	R/W	Description
2	SCKS	0	R/W	SSCK Pin Select Selects whether the SSCK pin functions as a port or a serial clock pin. 0: Functions as a port 1: Functions as a serial clock pin
1	CSS1	0	R/W	$\overline{\text{SCS}}$ Pin Select
0	CSS0	0	R/W	Selects whether the $\overline{\text{SCS}}$ pin functions as a port, an $\overline{\text{SCS}}$ input, or $\overline{\text{SCS}}$ output. When the SSUMS bit in SSCRL is 0, the $\overline{\text{SCS}}$ pin functions as a port regardless of the setting of this bit. 00: Functions as a port 01: Functions as an $\overline{\text{SCS}}$ input 1x: Functions as an $\overline{\text{SCS}}$ output (however, functions as an $\overline{\text{SCS}}$ input before starting transfer)

[Legend] x: Don't care.

15.3.2 SS Control Register L (SSCRL)

SSCRL is a register that controls mode, software reset, and open-drain output of the SSCK and SCS pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
6	SSUMS	0	R/W	SSU Mode Select Selects which combination of the serial data input pin and serial data output pin is used. For details, refer to section 15.4.3, Relationship between Data Input/Output and Shift Register. 0: Clocked synchronous communication mode Data input: SSI pin, Data output: SSO pin 1: Four-line bus communication mode When MSS = 1 and BIDE = 0 in SSCRH: Data input: SSI pin, Data output: SSO pin When MSS = 0 and BIDE = 0 in SSCRH: Data input: SSO pin, Data output: SSI pin When BIDE = 1 in SSCRH: Data input and output: SSO pin
5	SRES	0	R/W	Software Reset When this bit is set to 1, the SSU internal sequencer is forcibly reset. Then this bit is automatically cleared. The register value in the SSU is retained.
4	SCKOS	0	R/W	SSCK Pin Open-Drain Output Select Selects whether the SSCK pin functions as CMOS output or NMOS open-drain output. 0: CMOS output 1: NMOS open-drain output
3	CSOS	0	R/W	SCS Pin Open-Drain Output Select Selects whether the SCS pin functions as CMOS output or NMOS open-drain output. 0: CMOS output 1: NMOS open-drain output
2 to 0	—	All 0	—	Reserved These bits are always read as 0.

15.3.3 SS Mode Register (SSMR)

SSMR is a register that selects MSB-first or LSB-first, clock polarity, clock phase, and transfer clock rate.

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select Selects whether data transfer is performed in MSB-first or LSB-first. 0: LSB-first 1: MSB-first
6	CPOS	0	R/W	Clock Polarity Select Selects the clock polarity of SSCK. 0: Idle state = high 1: Idle state = low
5	CPHS	0	R/W	Clock Phase Select Selects the clock phase of SSCK. 0: Data change at first edge 1: Data latch at first edge
4, 3	—	All 0	—	Reserved These bits are always read as 0.
2	CKS2	0	R/W	Transfer clock rate select
1	CKS1	0	R/W	Sets transfer clock rate (prescaler division ratio) when the internal clock is selected.
0	CKS0	0	R/W	The system clock (ϕ) is halted in subactive mode or subsleep mode. Select $\phi_{SUB}/2$ in these modes. 000: $\phi/256$ 001: $\phi/128$ 010: $\phi/64$ 011: $\phi/32$ 100: $\phi/16$ 101: $\phi/8$ 110: $\phi/4$ 111: $\phi_{SUB}/2$

15.3.4 SS Enable Register (SSER)

SSER is a register that sets transmit enable, receive enable, and interrupt enable.

Bit	Bit Name	Initial Value	R/W	Description
7	TE	0	R/W	Transmit enable When this bit is 1, transmit operation is enabled.
6	RE	0	R/W	Receive enable When this bit is 1, receive operation is enabled.
5	RSSTP	0	R/W	Receive single stop When this bit is 1, receive operation is completed after receiving one byte.
4	—	0	—	Reserved This bit is always read as 0.
3	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, a TEI interrupt request is enabled.
2	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, a TXI interrupt request is enabled.
1	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, an RXI and an OEI interrupt requests are enabled.
0	CEIE	0	R/W	Conflict Error Interrupt Enable When this bit is set to 1, a CEI interrupt request is enabled.

15.3.5 SS Status Register (SSSR)

SSSR is a register that sets interrupt flags.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.
6	ORER	0	R/(W)*	Overrun Error Flag Indicates that the RDRF bit is abnormally terminated in reception because an overrun error has occurred. SSRDR retains received data before the overrun error occurs and the received data after the overrun error occurs is lost. When this bit is set to 1, subsequent serial reception cannot be continued. When the MSS bit in SSCRH is 1, this is also applied to serial transmission. [Setting condition] • When the next serial reception is completed while RDRF = 1 [Clearing condition] • When 0 is written to this bit after reading 1
5, 4	—	All 0	—	Reserved These bits are always read as 0.
3	TEND	0	R/(W)*	Transmit End [Setting condition] • When the last bit of data is transmitted, the TDRE bit is 1 [Clearing conditions] • When 0 is written to this bit after reading 1 • When data is written in SSTDR

Bit	Bit Name	Initial Value	R/W	Description
2	TDRE	1	R/(W)*	Transmit Data Empty [Setting conditions] • When the TE bit in SSER is 0 • When data transfer is performed from SSTDR to SSTRSR and data can be written in SSTDR [Clearing conditions] • When 0 is written to this bit after reading 1 • When data is written in SSTDR
1	RDRF	0	R/(W)*	Receive Data Register Full [Setting condition] • When serial reception is completed normally and receive data is transferred from SSTRSR to SSRDR [Clearing conditions] • When 0 is written to this bit after reading 1 • When data is read from SSRDR
0	CE	0	R/(W)*	Conflict Error Flag [Setting conditions] • When serial communication is started while SSUMS = 1 and MSS =1, the $\overline{SCS}$ pin input is low • When the $\overline{SCS}$ pin level changes from low to high during transfer while SSUMS = 1 and MSS = 0 [Clearing condition] • When 0 is written to this bit after reading 1

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

15.3.6 SS Receive Data Register (SSRDR)

SSRDR is an 8-bit register that stores received serial data. When the SSU has received one byte of serial data, it transfers the received serial data from SSTRSR and the data is stored. After this, SSTRSR is receive-enabled. As SSTRSR and SSRDR function as a double buffer in this way, continuous receive operations are possible. SSRDR is a read-only register and cannot be written to by the CPU. SSRDR is initialized to H'00.

15.3.7 SS Transmit Data Register (SSTDR)

SSTDR is an 8-bit register that stores serial data for transmission. SSTDR can be read or written to by the CPU at all times. When the SSU detects that SSTRSR is empty, it transfers the transmit data written in SSTDR to SSTRSR and starts serial transmission. If the next transmit data has already been written to SSTDR during serial transmission, continuous serial transmission is possible. SSTDR is initialized to H'00.

15.3.8 SS Shift Register (SSTRSR)

SSTRSR is a shift register that transmits and receives serial data. When transmit data is transferred from SSTDR to SSTRSR, bit 0 in SSTDR is transferred to bit 0 in SSTRSR while the MLS bit in SSMR is 0 (LSB-first transfer) and bit 7 in SSTDR is transferred to bit 0 in SSTRSR while the MLS bit in SSMR is 1 (MSB-first transfer). SSTRSR cannot be directly accessed by the CPU.

15.4 Operation

15.4.1 Transfer Clock

Transfer clock can be selected from eight internal clocks and an external clock. When this module is used, the SSCK pin must be selected as a serial clock by setting the SCKS bit in SSCRH to 1. When the MSS bit in SSCRH is 1, an internal clock is selected and the SSCK pin is in the output state. If transfer is started, the SSCK pin outputs clocks of the transfer rate set in the CKS2 to CKS0 bits in SSMR. When the MSS bit is 0, an external clock is selected and the SSCK pin is in the input state.

15.4.2 Relationship between Clock Polarity and Phase, and Data

Relationship between clock polarity and phase, and transfer data changes according to a combination of the SSUMS bit in SSCRL and the CPOS and CPHS bits in SSMR. Figure 15.2 shows the relationship.

MSB-first transfer or LSB first transfer can be selected by the setting of the MLS bit in SSMR. When the MLS bit is 0, transfer is started from LSB to MSB. When the MLS bit is 1, transfer is started from MSB to LSB.

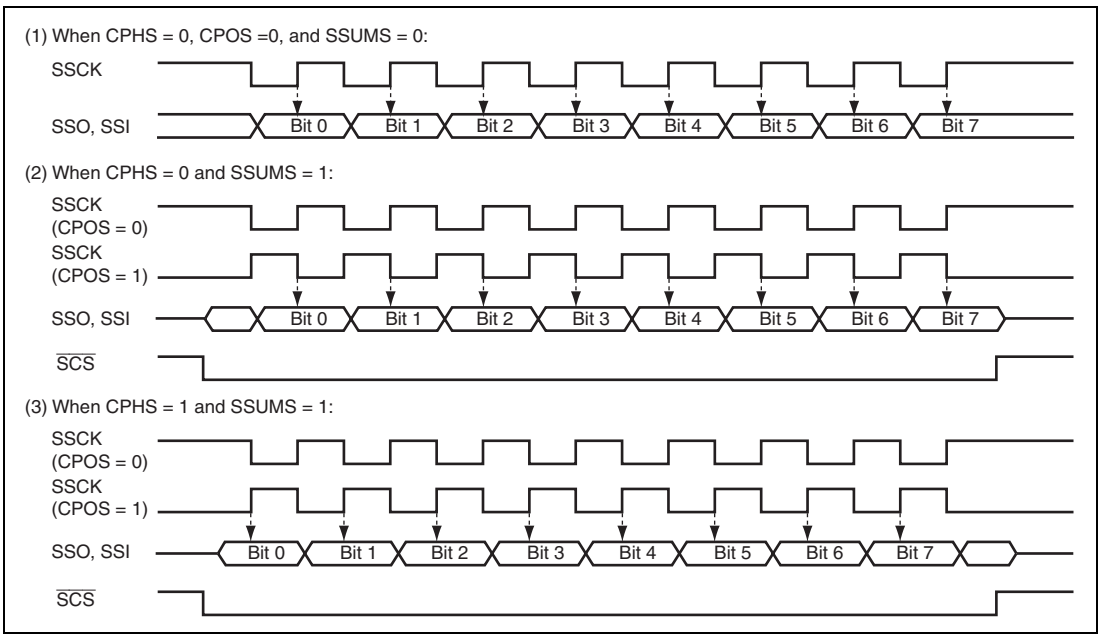


Figure 15.2 Relationship between Clock Polarity and Phase, and Data

15.4.3 Relationship between Data Input/Output and Shift Register

Relationship of connection between the data input/output pin and SSTRSR changes according to a combination of the MSS bit in SSCRH and the SSUMS bit in SSCRL. It also changes by the BIDE bit in SSCRH. Figure 15.3 shows the relationship.

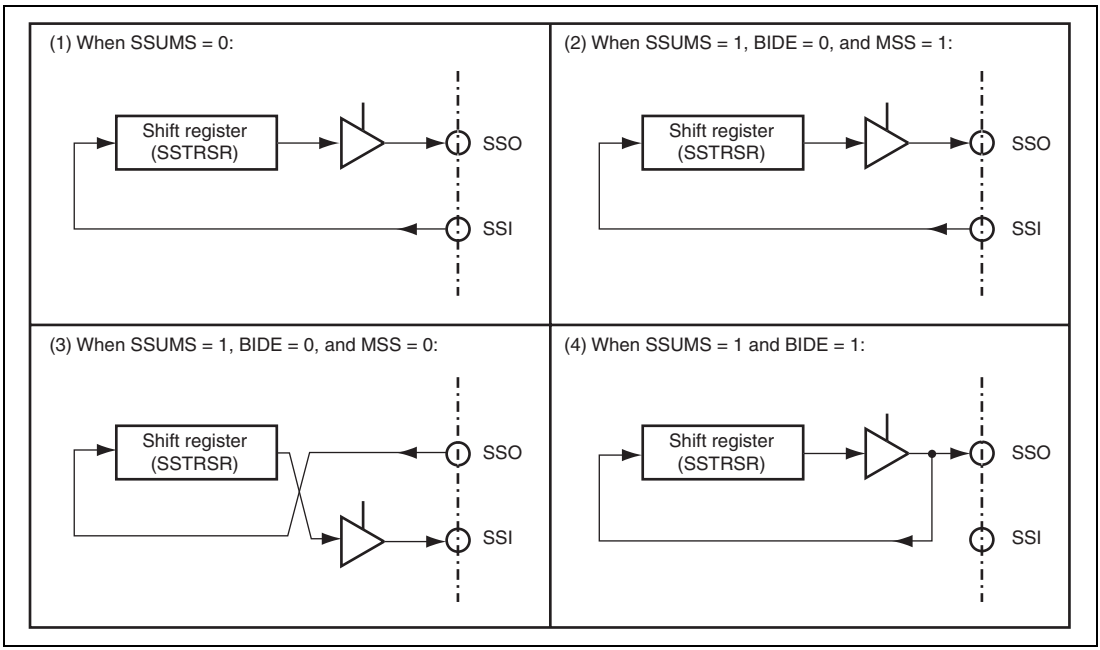


Figure 15.3 Relationship between Data Input/Output Pin and Shift Register

15.4.4 Communication Modes and Pin Functions

The SSU switches functions of the input/output pin in each communication mode according to the settings of the MSS bit in SSCRH and the RE and TE bits in SSER. Figure 15.2 shows the relationship between communication modes and the input/output pins. In bidirectional communication mode, neither TE nor RE should be set to 1.

Table 15.2 Relationship between Communication Modes and Input/Output Pins

Communication Mode	Register State					Pin State		
	SSUMS	BIDE	MSS	TE	RE	SSI	SSO	SSCK
Clocked Synchronous Communication Mode	0	x	0	0	1	In	—	In
				1	0	—	Out	In
					1	In	Out	In
			1	0	1	In	—	Out
				1	0	—	Out	Out
					1	In	Out	Out
Four-Line Bus Communication Mode	1	0	0	0	1	—	In	In
				1	0	Out	—	In
					1	Out	In	In
			1	0	1	In	—	Out
				1	0	—	Out	Out
					1	In	Out	Out
Four-Line Bus (Bidirectional) Communication Mode	1	1	0	0	1	—	In	In
				1	0	—	Out	In
			1	0	1	—	In	Out
				1	0	—	Out	Out

[Legend] x: Don't care.
—: Can be used as a general I/O port.

15.4.5 Operation in Clocked Synchronous Communication Mode

Initialization in Clocked Synchronous Communication Mode: Figure 15.4 shows the initialization in clocked synchronous communication mode. Before transmitting and receiving data, the TE and RE bits in SSER should be cleared to 0, then the SSU should be initialized.

Note: When the operating mode, or transfer format, is changed for example, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not change the contents of the RDRF and ORER flags, or the contents of SSRDR.

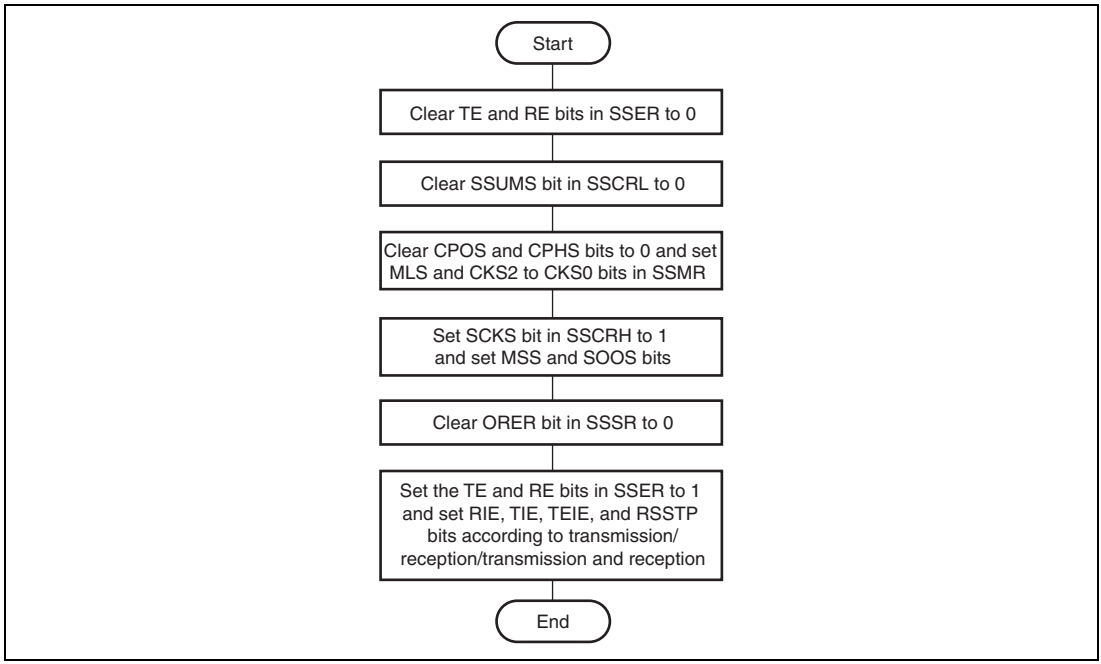


Figure 15.4 Initialization in Clocked Synchronous Communication Mode

Serial Data Transmission: Figure 15.5 shows an example of the SSU operation for transmission. In serial transmission, the SSU operates as described below.

When the SSU is set as a master device, it outputs a synchronous clock and data. When the SSU is set as a slave device, it outputs data in synchronized with the input clock.

When the SSU writes transmit data in SSTDR after setting the TE bit to 1, the TDRE flag is automatically cleared to 0 and data is transferred from SSTDR to SSTRSR. Then the SSU sets the TDRE flag to 1 and starts transmission. If the TIE bit in SSER is set to 1 at this time, a TXI is generated.

When the TDRE flag is 0 and one frame of data has transferred, data is transferred from SSTDR to SSTRSR and serial transmission of the next frame is started. If the eighth bit is transmitted while the TDRE flag is 1, the TEND bit in SSSR is set to 1 and the state is retained. If the TEIE bit in SSER is set to 1 at this time, a TEI is generated. After transmission is ended, the SSCK pin is fixed high.

While the ORER bit in SSSR is set to 1, transmission cannot be performed. Therefore confirm that the ORER bit is cleared to 0 before transmission.

Figure 15.6 shows a sample flowchart for serial data transmission.

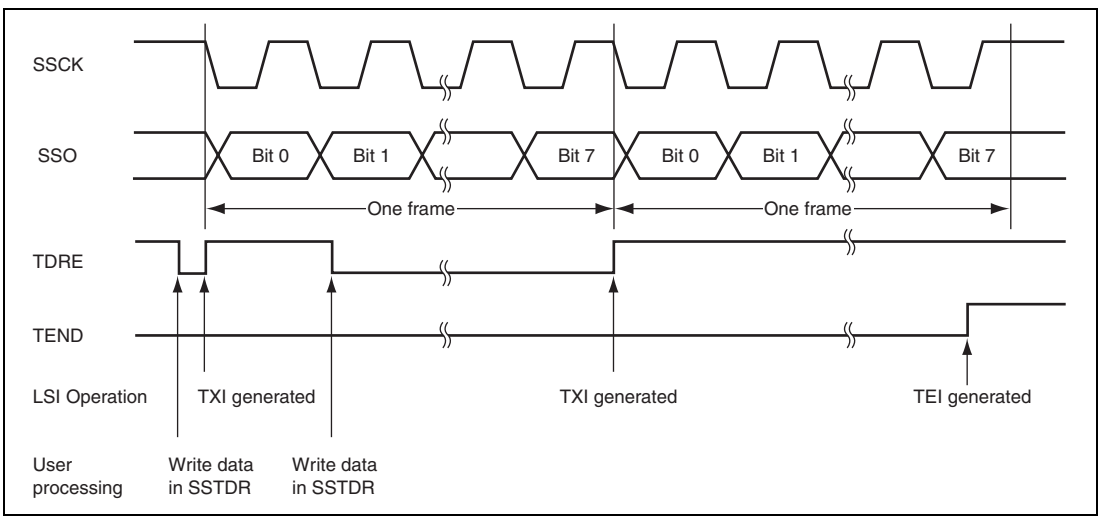


Figure 15.5 Example of Operation in Data Transmission

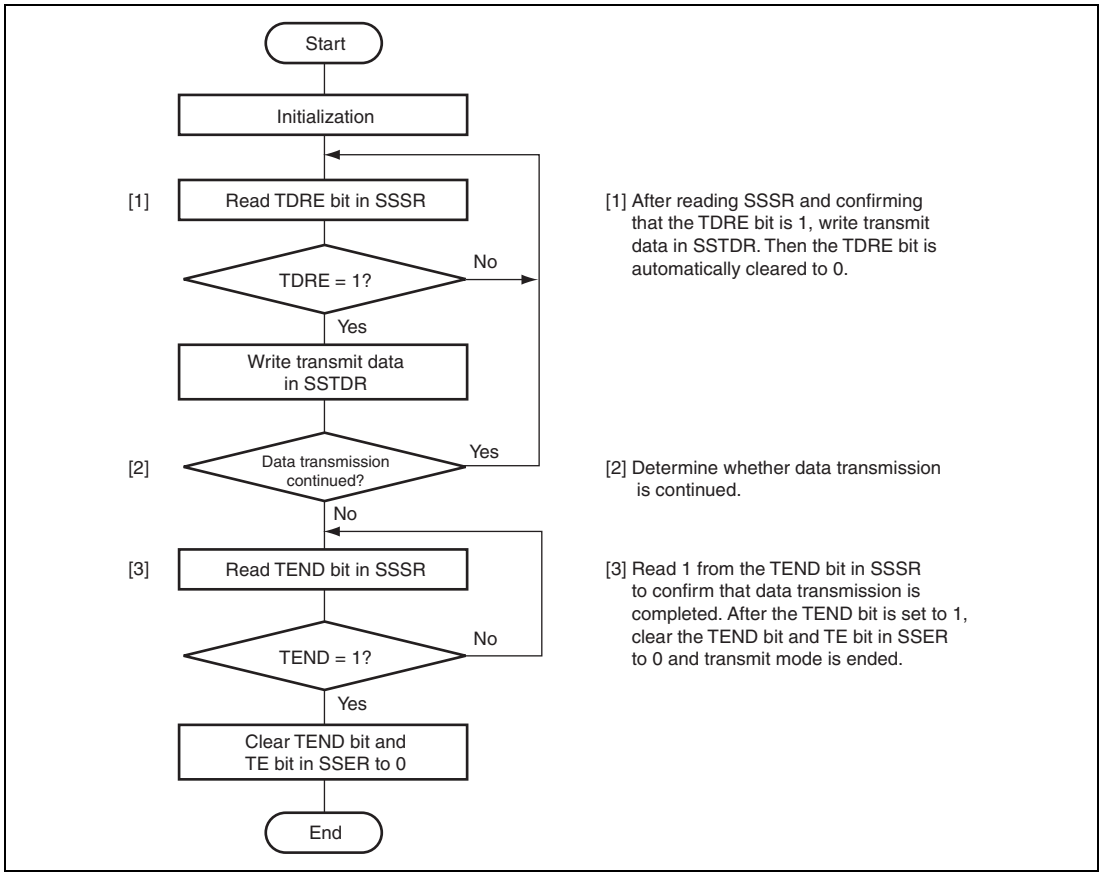


Figure 15.6 Sample Serial Transmission Flowchart

Serial Data Reception: Figure 15.7 shows an example of the SSU operation for reception. In serial reception, the SSU operates as described below.

When the SSU is set as a master device, it outputs a synchronous clock and inputs data. When the SSU is set as a slave device, it inputs data in synchronized with the input clock. When the SSU is set as a master device, it outputs a receive clock and starts reception by performing dummy read on SSRDR.

After eight bits of data is received, the RDRF bit in SSSR is set to 1 and received data is stored in SSRDR. If the RIE bit in SSER is set to 1 at this time, a RXI is generated. If SSRDR is read, the RDRF bit is automatically cleared to 0.

When the SSU is set as a master device and reception is ended, received data is read after setting the RSSTP bit in SSER to 1. Then the SSU outputs eight bits of clocks and operation is stopped. After that, the RE and RSSTP bits are cleared to 0 and the last received data is read. Note that if SSRDR is read while the RE bit is set to 1, received clock is output again.

When the eighth clock rises while the RDRF bit is 1, the ORER bit in SSSR is set. Then an overrun error (OEI) is generated and operation is stopped. When the ORER bit in SSSR is set to 1, reception cannot be performed. Therefore confirm that the ORER bit is cleared to 0 before reception.

Figure 15.8 shows a sample flowchart for serial data reception.

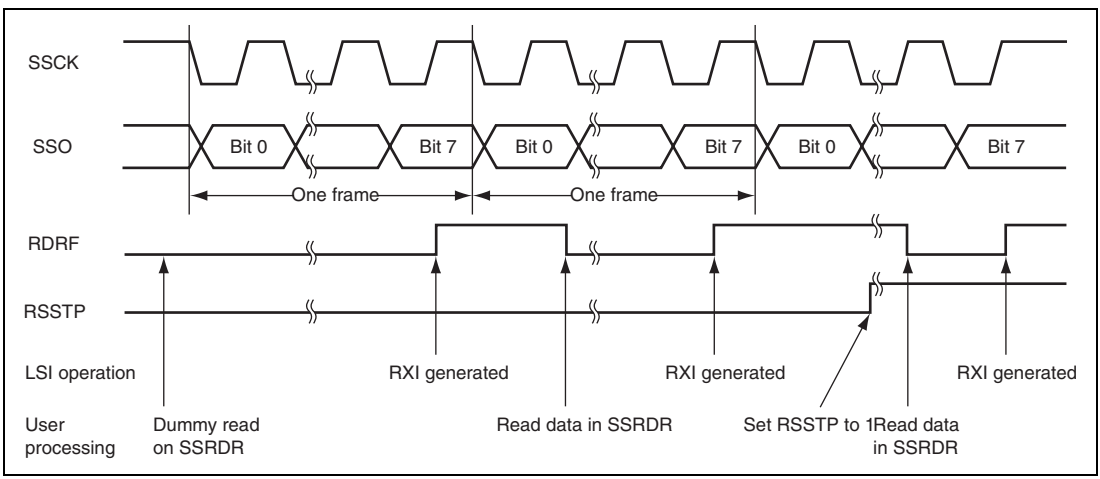


Figure 15.7 Example of Operation in Data Reception (MSS = 1)

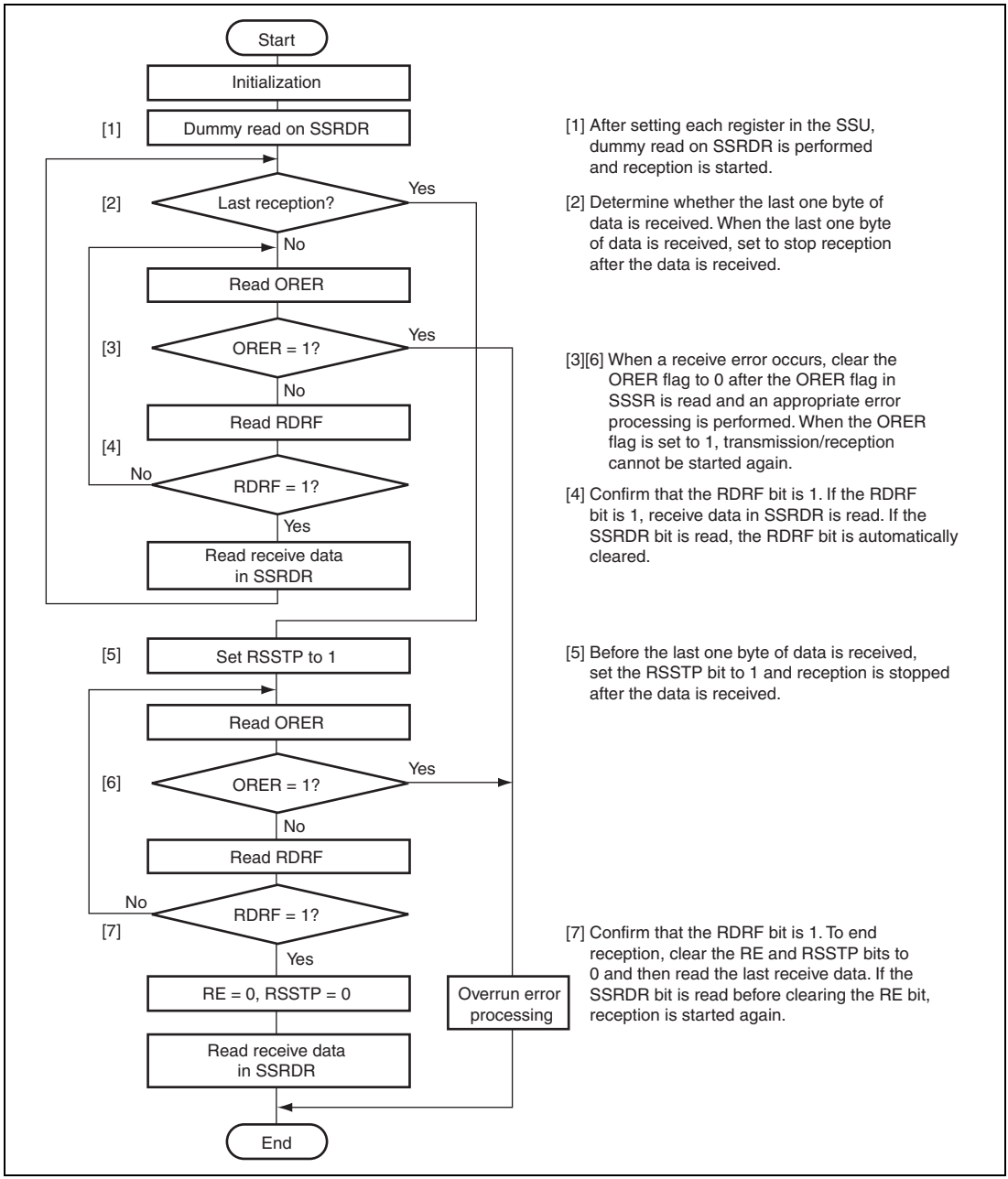


Figure 15.8 Sample Serial Reception Flowchart (MSS = 1)

Serial Data Transmission and Reception: Data transmission and reception is a combined operation of data transmission and reception which are described before. Transmission and reception is started by writing data in SSTDR. When the eighth clock rises or the ORER bit is set to 1 while the TDRE bit is set to 1, transmission and reception is stopped.

To switch from transmit mode (TE = 1) or receive mode (RE = 1) to transmit and receive mode (TE = RE = 1), the TE and RE bits should be cleared to 0. After confirming that the TEND, RDRF, and ORER bits are cleared to 0, set the TE and RE bits to 1.

Figure 15.9 shows a sample flowchart for serial transmit and receive operations.

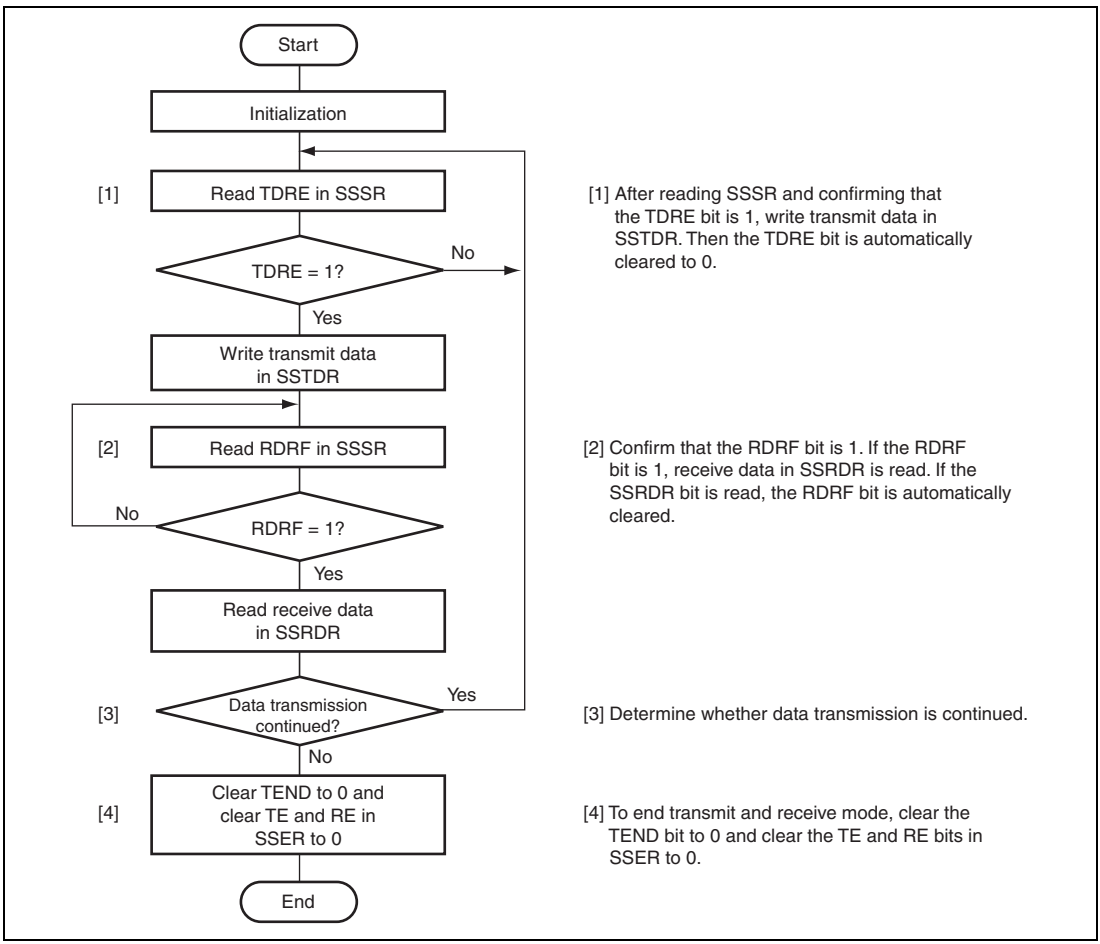


Figure 15.9 Sample Flowchart for Serial Transmit and Receive Operations

15.4.6 Operation in Four-Line Bus Communication Mode

Four-line bus communication mode is a mode which communicates with the four-line bus; a clock line, a data input line, a data output line, and a chip select line. This mode includes bidirectional mode in which the data input line and the data output line function as a single pin. The data input line and the data output line are changed according to the settings of the MSS and BIDE bits in SSCRH. For details, refer to section 15.4.3, Relationship between Data Input/Output and Shift Register. In this mode, relationship between clock polarity and phase, and data can be set by the CPOS and CPHS bits in SSMR. For details, refer to section 15.4.2, Relationship between Clock Polarity and Phase, and Data.

When the SSU is set as a master device, the chip select line controls output. When the SSU is set as a slave device, the chip select line controls input. When the SSU is set as a master device, the chip select line controls output of the $\overline{\text{SCS}}$ pin or controls output of a general port by setting the CSS1 bit in SSCRH to 1. When the SSU is set as a slave device, the chip select line sets the $\overline{\text{SCS}}$ pin as an input pin by setting the CSS1 and CSS0 bits in SSCRH to 01.

In four-line bus communication mode, the MLS bit in SSMR is set to 1 and transfer is performed in MSB-first order.

15.4.7 Initialization in Four-Line Bus Communication Mode

Figure 15.10 shows the initialization in four-line bus communication mode. Before transmitting and receiving data, the TE and RE bits in SSER should be cleared to 0, then the SSU should be initialized.

Note: When the operating mode, or transfer format, is changed for example, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not change the contents of the RDRF and ORER flags, or the contents of SSRDR.

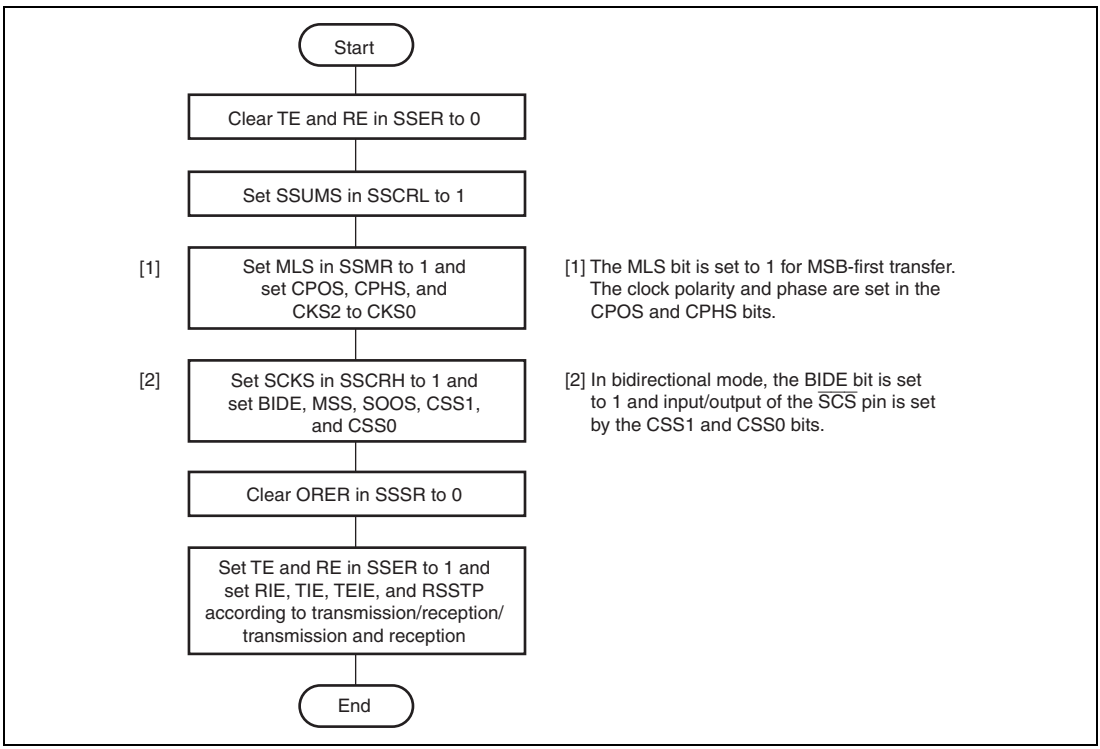


Figure 15.10 Initialization in Four-Line Bus Communication Mode

15.4.8 Serial Data Transmission

Figure 15.11 shows an example of the SSU operation for transmission. In serial transmission, the SSU operates as described below.

When the SSU is set as a master device, it outputs a synchronous clock and data. When the SSU is set as a slave device, the $\overline{\text{SCS}}$ pin is in the low-input state and the SSU outputs data in synchronized with the input clock.

When the SSU writes transmit data in SSTDR after setting the TE bit to 1, the TDRE flag is automatically cleared to 0 and data is transferred from SSTDR to SSTRSR. Then the SSU sets the TDRE flag to 1 and starts transmission. If the TIE bit in SSER is set to 1 at this time, a TXI is generated.

When the TDRE flag is 0 and one frame of data has transferred, data is transferred from SSTDR to SSTRSR and serial transmission of the next frame is started. If the eighth bit is transmitted while the TDRE flag is 1, the TEND bit in SSSR is set to 1 and the state is retained. If the TEIE bit in SSER is set to 1 at this time, a TEI is generated. After transmission is ended, the SSCK pin is fixed high and the $\overline{\text{SCS}}$ pin goes high. When continuous transmission is performed with the $\overline{\text{SCS}}$ pin low, the next data should be written to SSTDR before transmitting the eighth bit of the frame.

While the ORER bit in SSSR is set to 1, transmission cannot be performed. Therefore confirm that the ORER bit is cleared to 0 before transmission.

The difference between this mode and clocked synchronous communication mode is as follows: when the SSU is set as a master device, the SSO pin is in the Hi-Z state if the $\overline{\text{SCS}}$ pin is in the Hi-Z state and when the SSU is set as a slave device, the SSI pin is in the Hi-Z state if the $\overline{\text{SCS}}$ pin is in the high-input state. The sample flowchart for serial data transmission is the same as that in clocked synchronous communication mode.

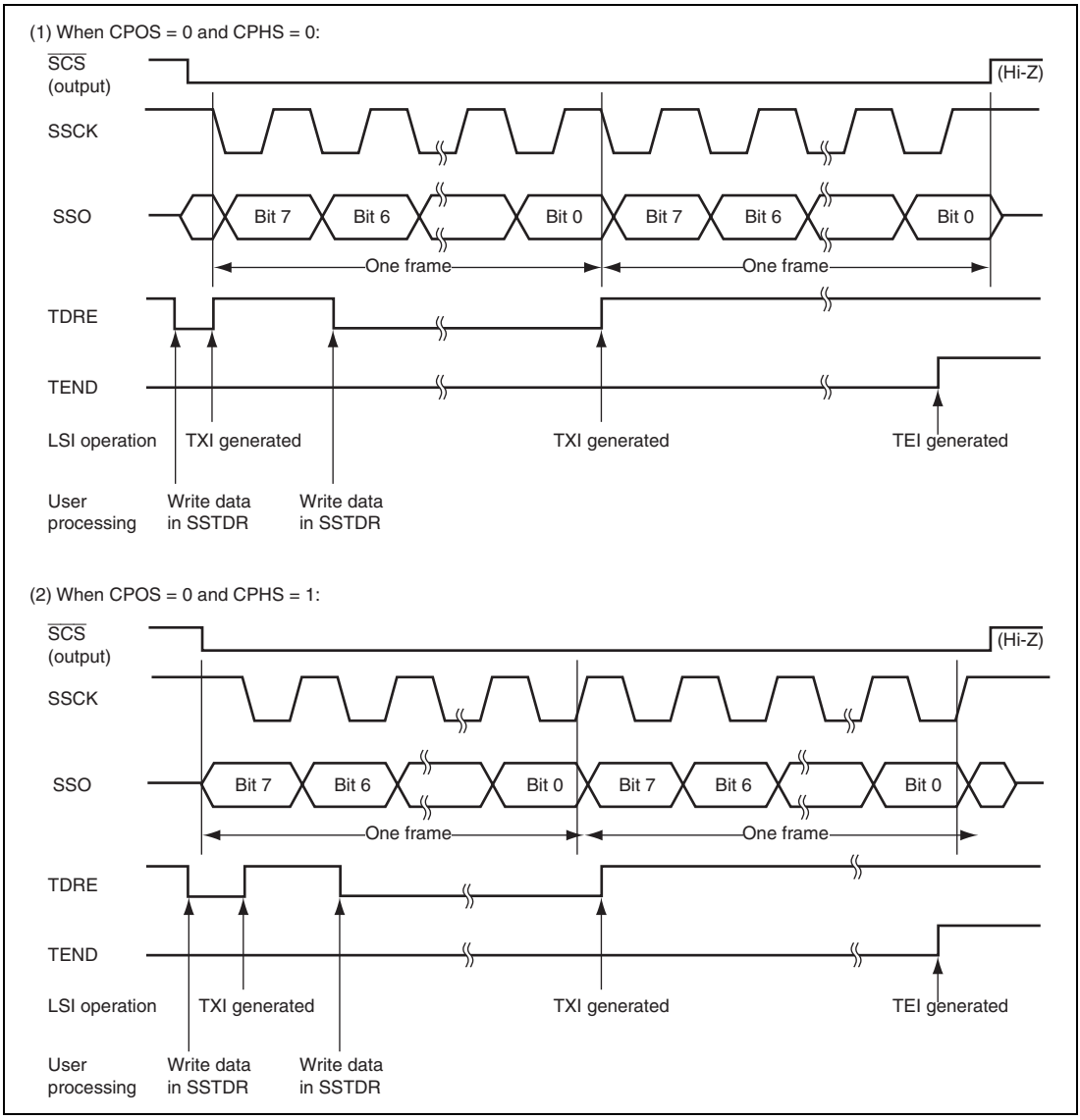


Figure 15.11 Example of Operation in Data Transmission (MSS = 1)

15.4.9 Serial Data Reception

Figure 15.12 shows an example of the SSU operation for reception. In serial reception, the SSU operates as described below.

When the SSU is set as a master device, it outputs a synchronous clock and inputs data. When the SSU is set as a slave device, the $\overline{SCS}$ pin is in the low-input state and inputs data in synchronized with the input clock. When the SSU is set as a master device, it outputs a receive clock and starts reception by performing dummy read on SSRDR.

After eight bits of data is received, the RDRF bit in SSSR is set to 1 and received data is stored in SSRDR. If the RIE bit in SSER is set to 1 at this time, an RXI is generated. If SSRDR is read, the RDRF bit is automatically cleared to 0.

When the SSU is set as a master device and reception is ended, received data is read after setting the RSSTP bit in SSER to 1. Then the SSU outputs eight bits of clocks and operation is stopped. After that, the RE and RSSTP bits are cleared to 0 and the last received data is read. Note that if SSRDR is read while the RE bit is set to 1, received clock is output again.

When the eighth clock rises while the RDRF bit is 1, the ORER bit in SSSR is set. Then an overrun error (OEI) is generated and operation is stopped. When the ORER bit in SSSR is set to 1, reception cannot be performed. Therefore confirm that the ORER bit is cleared to 0 before reception.

The set timings of the RDRF and ORER flags differ according to the CPHS setting. These timings are shown in figure 15.12. When the CPHS bit is set to 1, the flag is set during the frame. Therefore care should be taken at the end of reception.

The sample flowchart for serial data reception is the same as that in clocked synchronous communication mode.

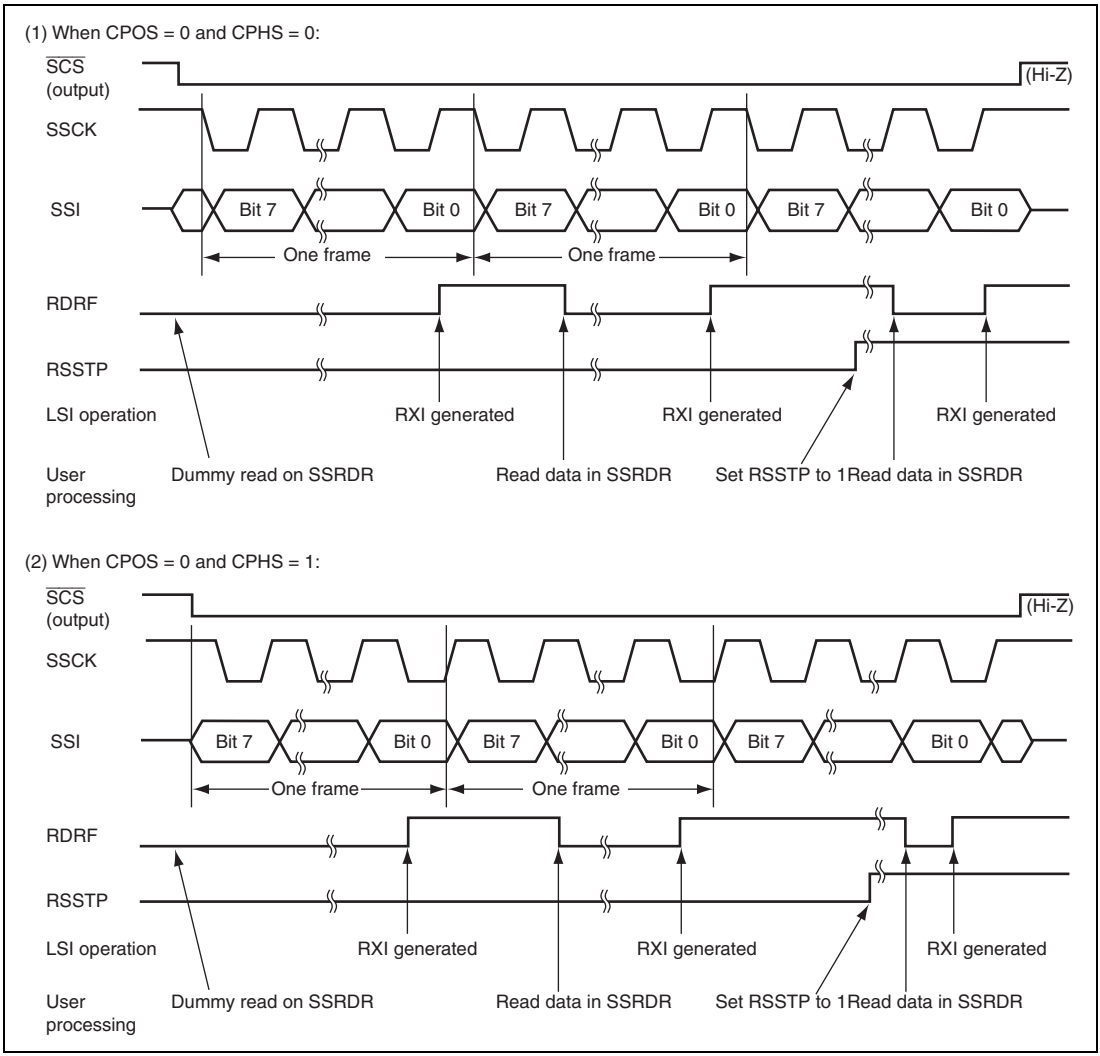


Figure 15.12 Example of Operation in Data Reception (MSS = 1)

15.4.10 $\overline{\text{SCS}}$ Pin Control and Arbitration

When the SSUMS bit in SSCRL is set to 1 and the CSS1 bit in SSCRH is set to 1, the MSS bit in SSCRH is set to 1 and then the arbitration of the $\overline{\text{SCS}}$ pin is checked before starting serial transfer. If the SSU detects that the synchronized internal $\overline{\text{SCS}}$ pin goes low in this period, the CE bit in SSSR is set and the MSS bit is cleared.

Note: When a conflict error is set, subsequent transmit operation is not possible. Therefore the CE bit must be cleared to 0 before starting transmission.
When the multimaster error is used, the CSOS bit in SSCRL should be set to 1.

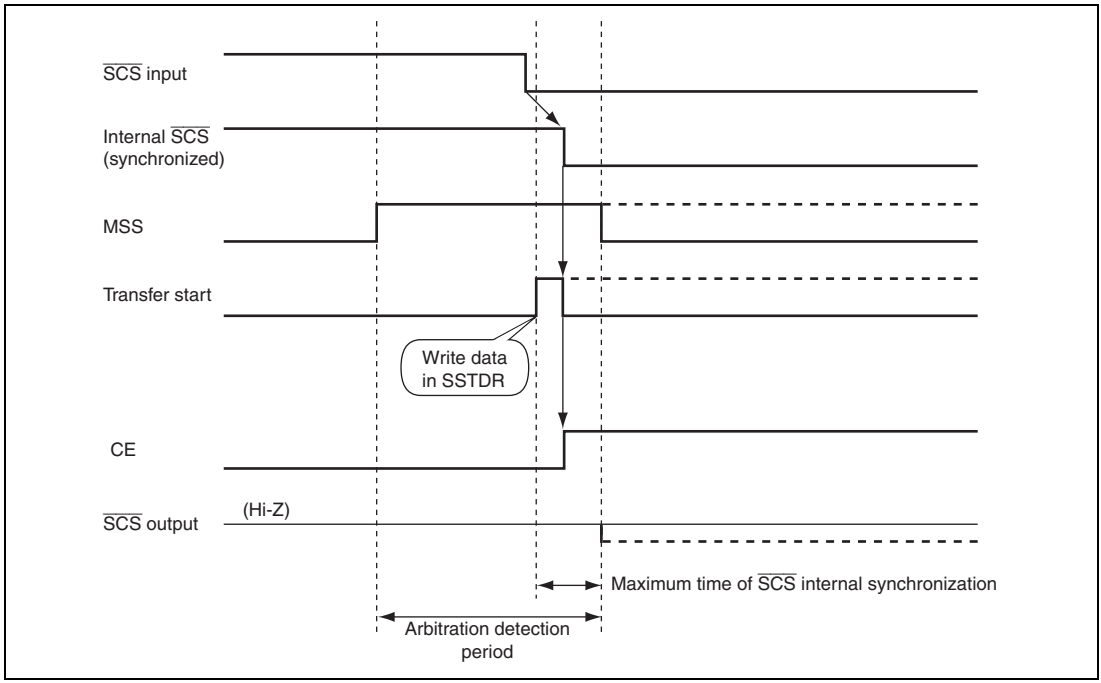


Figure 15.13 Arbitration Check Timing

15.4.11 Interrupt Requests

The SSU has five interrupt requests: transmit data empty, transmit end, receive data full, overrun error, and conflict error. Since these interrupt requests are assigned to the common vector address, interrupt sources must be determined by flags. Table 15.3 lists the interrupt requests.

Table 15.3 Interrupt Requests

Interrupt Request	Abbreviation	Interrupt Condition
Transmit data empty	TXI	(TIE = 1), (TDRE = 1)
Transmit end	TEI	(TEIE = 1), (TEND = 1)
Receive data full	RXI	(RIE = 1), (RDRF = 1)
Overrun error	OEI	(RIE = 1), (ORER = 1)
Conflict error	CEI	(CEIE = 1), (CE = 1)

When an interrupt condition shown in table 15.3 is 1 and the I bit in CCR is 0, the CPU executes the interrupt exception handling. Each interrupt source must be cleared during the exception handling. Note that the TDRE and TEND bits are automatically cleared by writing transmit data in SSTDR and the RDRF bit is automatically cleared by reading SSRDR. When transmit data is written in SSTDR, the TDRE bit is set again at the same time. Then if the TDRE bit is cleared, additional one byte of data may be transmitted.

15.5 Usage Note

When writing 1 to the SOLP bit in SSCRH (to enable write protect) after writing 0 to it (to disable write protect), the SOL bit may be changed without being protected.

To avoid this, before writing 1 to the SOLP bit (to enable write protect), write the current value of the SOL bit to itself. With this procedure, the write protect can be performed on the SOL bit.

Section 16 I²C Bus Interface 2 (IIC2)

The I²C bus interface 2 conforms to and provides a subset of the Philips I²C bus (inter-IC bus) interface functions. The configuration of the registers that control the I²C bus differs partly from the Philips configuration, however. Figure 16.1 shows a block diagram of the I²C bus interface 2. Figure 16.2 shows an example of I/O pin connections to external circuits.

16.1 Features

- Selection of I²C format or clock synchronous serial format
- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independent from each other, the continuous transmission/reception can be performed.
- Use of module standby mode enables this module to be placed in standby mode independently when not used. (The IIC2 is halted as the initial value. For details, refer to section 5.4, Module Standby Function.)

I²C bus format

- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization/wait function

  In master mode, the state of SCL is monitored per bit, and the timing is synchronized automatically.

  If transmission/reception is not yet possible, set the SCL to low until preparations are completed.
- Six interrupt sources

  Transmit data empty (including slave-address match), transmit end, receive data full (including slave-address match), arbitration lost, NACK detection, and stop condition detection
- Direct bus drive

  Two pins, SCL and SDA pins, function as CMOS outputs in normal operation (when the port/serial function is selected) and NMOS outputs when the bus drive function is selected.

Clock synchronous format

- Four interrupt sources

  Transmit-data-empty, transmit-end, receive-data-full, and overrun error

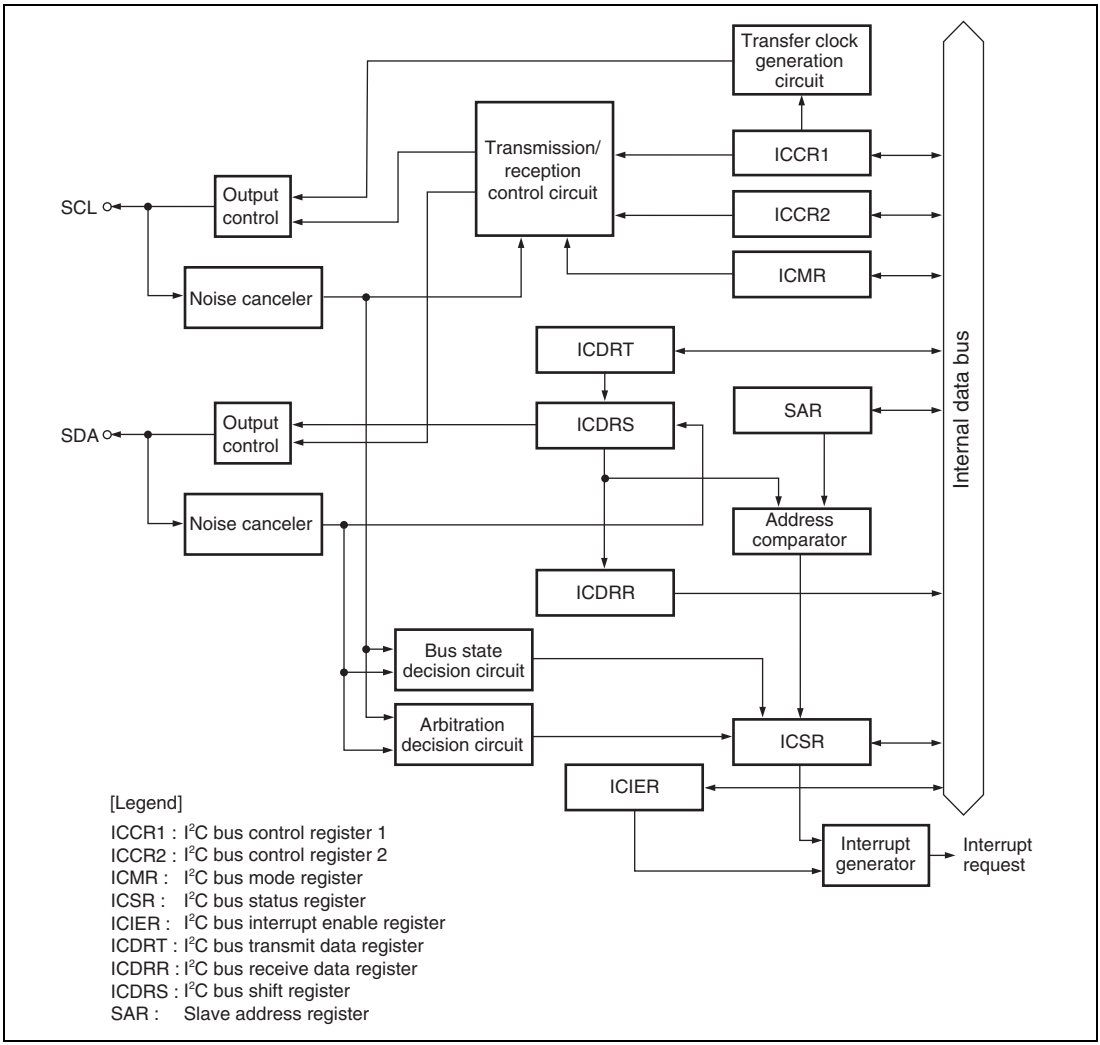


Figure 16.1 Block Diagram of I²C Bus Interface 2

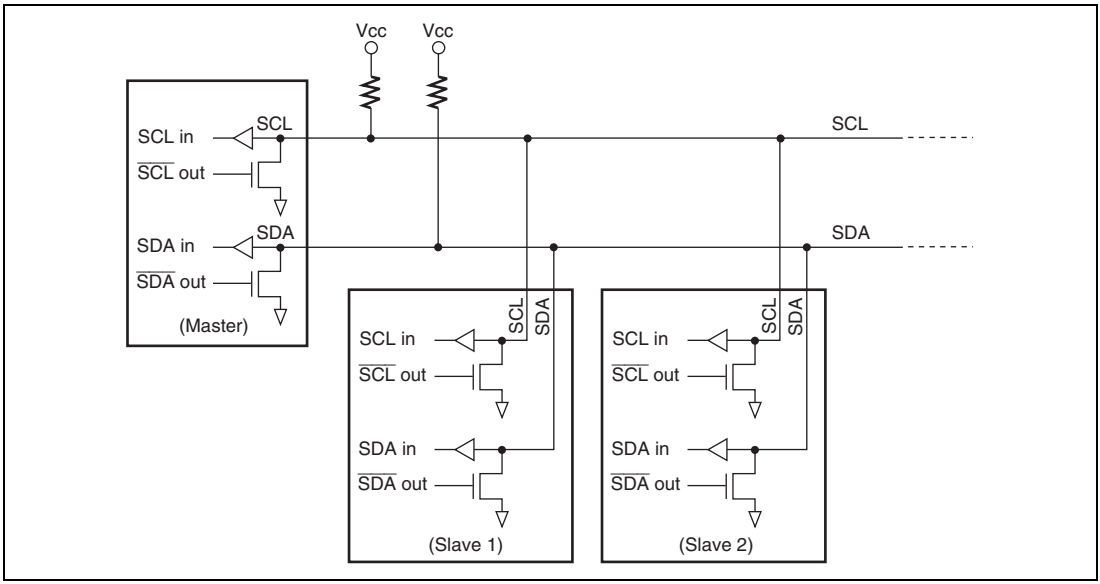


Figure 16.2 External Circuit Connections of I/O Pins

16.2 Input/Output Pins

Table 16.1 shows the pin configuration of the I²C bus interface 2.

Table 16.1 Pin Configuration

Name	Abbreviation	I/O	Function
Serial clock pin	SCL	I/O	IIC serial clock input/output
Serial data pin	SDA	I/O	IIC serial data input/output

16.3 Register Descriptions

The I²C bus interface 2 has the following registers.

- I²C bus control register 1 (ICCR1)
- I²C bus control register 2 (ICCR2)
- I²C bus mode register (ICMR)
- I²C bus interrupt enable register (ICIER)
- I²C bus status register (ICSR)
- Slave address register (SAR)
- I²C bus transmit data register (ICDRT)
- I²C bus receive data register (ICDRR)
- I²C bus shift register (ICDRS)

16.3.1 I²C Bus Control Register 1 (ICCR1)

ICCR1 enables or disables the I²C bus interface 2, controls transmission or reception, and selects master or slave mode, transmission or reception, and transfer clock frequency in master mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface Enable 0: This module is halted. (SCL and SDA pins are set to the port/serial function.) 1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable This bit enables or disables the next operation when TRS is 0 and ICDRR is read. 0: Enables next reception 1: Disables next reception

Bit	Bit Name	Initial Value	R/W	Description
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select 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. Modification of the TRS bit should be made between transfer frames. After data receive has been started in slave receive mode, when the first seven bits of the receive data agree with the slave address that is set to SAR and the eighth bit is 1, TRS is automatically set to 1. If an overrun error occurs in master mode with the clock synchronous serial format, MST is cleared to 0 and slave receive mode is entered. Operating modes are described below according to MST and TRS combination. When clock synchronous serial format is selected and MST is 1, clock is output. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode
3	CKS3	0	R/W	Transfer Clock Select 3 to 0 These bits are valid only in master mode and should be set according to the necessary transfer rate (refer to table 16.2). These bit are used to specify the data setup time in slave transmit mode. The data setup time is secured for 10tcyc when CKS3 = 0 and for 20tcyc when CKS3 = 1.
2	CKS2	0	R/W	
1	CKS1	0	R/W	
0	CKS0	0	R/W	

Table 16.2 Transfer Rate

Bit 3	Bit 2	Bit 1	Bit 0		Transfer Rate		
CKS3	CKS2	CKS1	CKS0	Clock	φ = 2 MHz	φ = 5 MHz	φ = 10 MHz
0	0	0	0	φ/28	71.4 kHz	179 kHz	357 kHz
			1	φ/40	50.0 kHz	125 kHz	250 kHz
		1	0	φ/48	41.7 kHz	104 kHz	208 kHz
			1	φ/64	31.3 kHz	78.1 kHz	156 kHz
	1	0	0	φ/80	25.0 kHz	62.5 kHz	125 kHz
			1	φ/100	20.0 kHz	50.0 kHz	100 kHz
		1	0	φ/112	17.9 kHz	44.6 kHz	89.3 kHz
			1	φ/128	15.6 kHz	39.1 kHz	78.1 kHz
1	0	0	0	φ/56	35.7 kHz	89.3 kHz	179 kHz
			1	φ/80	25.0 kHz	62.5 kHz	125 kHz
		1	0	φ/96	20.8 kHz	52.1 kHz	104 kHz
			1	φ/128	15.6 kHz	39.1 kHz	78.1 kHz
	1	0	0	φ/160	12.5 kHz	31.3 kHz	62.5 kHz
			1	φ/200	10.0 kHz	25.0 kHz	50.0 kHz
		1	0	φ/224	8.9 kHz	22.3 kHz	44.6 kHz
			1	φ/256	7.8 kHz	19.5 kHz	39.1 kHz

16.3.2 I²C Bus Control Register 2 (ICCR2)

ICCR2 issues start/stop conditions, manipulates the SDA pin, monitors the SCL pin, and controls reset in the control part of the I²C bus interface 2.

Bit	Bit Name	Initial Value	R/W	Description
7	BBSY	0	R/W	Bus Busy This bit enables to confirm whether the I²C bus is occupied or released and to issue start/stop conditions in master mode. With the clock synchronous serial format, this bit has no meaning. With the I²C bus format, this bit is set to 1 when the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. Follow this procedure when also re-transmitting a start condition. Write 0 in BBSY and 0 in SCP to issue a stop condition. To issue start/stop conditions, use the MOV instruction.
6	SCP	1	R/W	Start/Stop Issue Condition Disable The SCP bit controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 in BBSY and 0 in SCP. A repeated 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.
5	SDAO	1	R/W	SDA Output Value Control This bit is used with SDAOP when modifying output level of SDA. This bit should not be manipulated during transfer. 0: When reading, SDA pin outputs low. When writing, SDA pin is changed to output low. 1: When reading, SDA pin outputs high. When writing, SDA pin is changed to output Hi-Z (outputs high by external pull-up resistance).

Bit	Bit Name	Initial Value	R/W	Description
4	SDAOP	1	R/W	SDAO Write Protect This bit controls change of output level of the SDA pin by modifying the SDAO bit. To change the output level, clear SDAO and SDAOP to 0 or set SDAO to 1 and clear SDAOP to 0 by the MOV instruction. This bit is always read as 1.
3	SCLO	1	R	This bit monitors SCL output level. When SCLO is 1, SCL pin outputs high. When SCLO is 0, SCL pin outputs low.
2	—	1	—	Reserved This bit is always read as 1, and cannot be modified.
1	IICRST	0	R/W	IIC Control Part Reset This bit resets the control part except for I²C registers. If this bit is set to 1 when hang-up occurs because of communication failure during I²C operation, I²C control part can be reset without setting ports and initializing registers.
0	—	1	—	Reserved This bit is always read as 1, and cannot be modified.

16.3.3 I²C Bus Mode Register (ICMR)

ICMR selects whether the MSB or LSB is transferred first, performs master mode wait control, and selects the transfer bit count.

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I²C bus format is used.
6	WAIT	0	R/W	Wait Insertion Bit In master mode with the I²C bus format, this bit selects whether to insert a wait after data transfer except the acknowledge bit. When WAIT is set to 1, after the fall of the clock for the final data bit, low period is extended for two transfer clocks. If WAIT is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted. The setting of this bit is invalid in slave mode with the I²C bus format or with the clock synchronous serial format.
5, 4	—	All 1	—	Reserved These bits are always read as 1, and cannot be modified.
3	BCWP	1	R/W	BC Write Protect This bit controls the BC2 to BC0 modifications. When modifying BC2 to BC0, this bit should be cleared to 0 and use the MOV instruction. In clock synchronous serial mode, BC should not be modified. 0: When writing, values of BC2 to BC0 are set. 1: When reading, 1 is always read. When writing, settings of BC2 to BC0 are invalid.

Bit	Bit Name	Initial Value	R/W	Description
2	BC2	0	R/W	Bit Counter 2 to 0
1	BC1	0	R/W	These bits specify the number of bits to be transferred next. When read, the remaining number of transfer bits is indicated. With the I²C bus format, 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 pin is low. The value returns to 000 at the end of a data transfer, including the acknowledge bit. With the clock synchronous serial format, these bits should not be modified.
0	BC0	0	R/W	

I²C Bus Format	Clock Synchronous Serial Format
000: 9 bits	000: 8 bits
001: 2 bits	001: 1 bits
010: 3 bits	010: 2 bits
011: 4 bits	011: 3 bits
100: 5 bits	100: 4 bits
101: 6 bits	101: 5 bits
110: 7 bits	110: 6 bits
111: 8 bits	111: 7 bits

16.3.4 I²C Bus Interrupt Enable Register (ICIER)

ICIER enables or disables interrupt sources and acknowledge bits, sets acknowledge bits to be transferred, and confirms acknowledge bits to be received.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, this bit enables or disables the transmit data empty interrupt (TXI). 0: Transmit data empty interrupt request (TXI) is disabled. 1: Transmit data empty interrupt request (TXI) is enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable This bit enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by clearing the TEND bit or the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabled. 1: Transmit end interrupt request (TEI) is enabled.
5	RIE	0	R/W	Receive Interrupt Enable This bit enables or disables the receive data full interrupt request (RXI) and the overrun error interrupt request (ERI) with the clock synchronous format, when a receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) and overrun error interrupt request (ERI) with the clock synchronous format are disabled. 1: Receive data full interrupt request (RXI) and overrun error interrupt request (ERI) with the clock synchronous format are enabled.

Bit	Bit Name	Initial Value	R/W	Description
4	NAKIE	0	R/W	NACK Receive Interrupt Enable This bit enables or disables the NACK receive interrupt request (NAKI) and the overrun error (setting of the OVE bit in ICSR) interrupt request (ERI) with the clock synchronous format, when the NACKF and AL bits in ICSR are set to 1. NAKI can be canceled by clearing the NACKF, OVE, or NAKIE bit to 0. 0: NACK receive interrupt request (NAKI) is disabled. 1: NACK receive interrupt request (NAKI) is enabled.
3	STIE	0	R/W	Stop Condition Detection Interrupt Enable 0: Stop condition detection interrupt request (STPI) is disabled. 1: Stop condition detection interrupt request (STPI) is enabled.
2	ACKE	0	R/W	Acknowledge Bit Judgment Select 0: The value of the receive acknowledge bit is ignored, and continuous transfer is performed. 1: If the receive acknowledge bit is 1, continuous transfer is halted.
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowledge data that are returned by the receive device. This bit cannot be modified. 0: Receive acknowledge = 0 1: Receive acknowledge = 1
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, this bit specifies the bit to be sent at the acknowledge timing. 0: 0 is sent at the acknowledge timing. 1: 1 is sent at the acknowledge timing.

16.3.5 I²C Bus Status Register (ICSR)

ICSR performs confirmation of interrupt request flags and status.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	0	R/(W)*	Transmit Data Register Empty [Setting conditions] • When data is transferred from ICDRT to ICDRS and ICDRT becomes empty • When TRS is set • When a start condition (including re-transfer) has been issued • When transmit mode is entered from receive mode in slave mode [Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1 • When data is written to ICDRT with an instruction
6	TEND	0	R/(W)*	Transmit End [Setting conditions] • When the ninth clock of SCL rises with the I²C bus format while the TDRE flag is 1 • When the final bit of transmit frame is sent with the clock synchronous serial format [Clearing conditions] • When 0 is written in TEND after reading TEND = 1 • When data is written to ICDRT with an instruction
5	RDRF	0	R/(W)*	Receive Data Register Full [Setting condition] • When a receive data is transferred from ICDRS to ICDRR [Clearing conditions] • When 0 is written in RDRF after reading RDRF = 1 • When ICDRR is read with an instruction

Bit	Bit Name	Initial Value	R/W	Description
4	NACKF	0	R/(W)*	No Acknowledge Detection Flag [Setting condition] • When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1 [Clearing condition] • When 0 is written in NACKF after reading NACKF = 1
3	STOP	0	R/(W)*	Stop Condition Detection Flag [Setting conditions] • In master mode, when a stop condition is detected after the completion of frame transfer • In slave mode, when a stop condition is detected, after the slave address of the first byte, following the general call and the detection of the start condition, matches the address set in SAR [Clearing condition] • When 0 is written in STOP after reading STOP = 1

Bit	Bit Name	Initial Value	R/W	Description
2	AL/OVE	0	R/(W)*	Arbitration Lost Flag/Overrun Error Flag This flag indicates that arbitration was lost in master mode with the I²C bus format and that the final bit has been received while RDRF = 1 with the clock synchronous format. When two or more master devices attempt to seize the bus at nearly the same time, if the I²C bus interface detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been taken by another master. [Setting conditions] • If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode • When the SDA pin outputs high in master mode while a start condition is detected • When the final bit is received with the clock synchronous format while RDRF = 1 [Clearing condition] • When 0 is written in AL/OVE after reading AL/OVE = 1
1	AAS	0	R/(W)*	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR. [Setting conditions] • When the slave address is detected in slave receive mode • When the general call address is detected in slave receive mode. [Clearing condition] • When 0 is written in AAS after reading AAS = 1

Bit	Bit Name	Initial Value	R/W	Description
0	ADZ	0	R/(W)*	General Call Address Recognition Flag This bit is valid in I²C bus format slave receive mode. [Setting condition] • When the general call address is detected in slave receive mode [Clearing condition] • When 0 is written in ADZ after reading ADZ = 1

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

16.3.6 Slave Address Register (SAR)

SAR selects the communication format and sets the slave address. When the chip is in slave mode with the I²C bus format, 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.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA6 to SVA0	All 0	R/W	Slave Address 6 to 0 These bits set a unique address in bits SVA6 to SVA0, differing form the addresses of other slave devices connected to the I²C bus.
0	FS	0	R/W	Format Select 0: I²C bus format is selected. 1: Clock synchronous serial format is selected.

16.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. If the next transfer data is written to ICDRT during transferring data of ICDRS, continuous transfer is possible. If the MLS bit of ICMR is set to 1 and when the data is written to ICDRT, the MSB/LSB inverted data is read. The initial value of ICDRT is H'FF. The initial value of ICDRT is H'FF.

16.3.8 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit register that stores the receive data. When data of one byte is received, ICDRR transfers the receive data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register, therefore the CPU cannot write to this register. The initial value of ICDRR is H'FF.

16.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred from ICDRS to ICDRR after data of one byte is received. This register cannot be read directly from the CPU.

16.4 Operation

The I²C bus interface 2 can communicate either in I²C bus mode or clock synchronous serial mode by setting FS in SAR.

16.4.1 I²C Bus Format

Figure 16.3 shows the I²C bus formats. Figure 16.4 shows the I²C bus timing. The first frame following a start condition always consists of 8 bits.

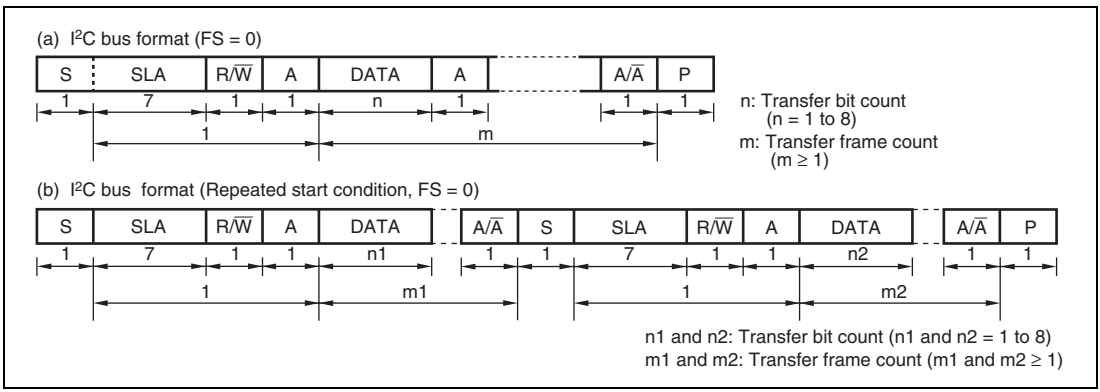


Figure 16.3 I²C Bus Formats

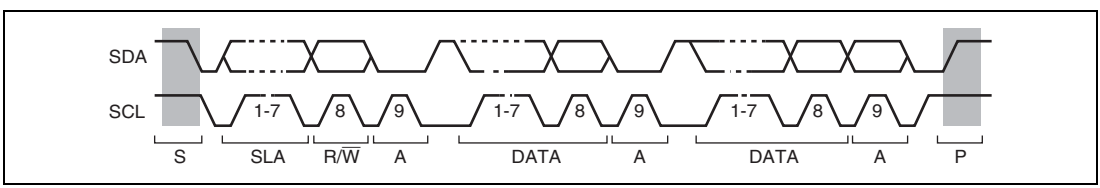


Figure 16.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high.
SLA: Slave address
R/W: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.
A: Acknowledge. The receive device drives SDA to low.
DATA: Transfer data
P: Stop condition. The master device drives SDA from low to high while SCL is high.

16.4.2 Master Transmit Operation

In master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For master transmit mode operation timing, refer to figures 16.5 and 16.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is free. Set the MST and TRS bits in ICCR1 to select master transmit mode. Then, write 1 to BBSY and 0 to SCP using MOV instruction. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first byte data show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cleared to 0, and data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR is set to 1 at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confirm that the slave device has been selected. Then, write second byte data to ICDRT. When ACKBR is 1, the slave device has not been acknowledged, so issue the stop condition. To issue the stop condition, write 0 to BBSY and SCP using MOV instruction. SCL is fixed low until the transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the end of last byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) from the receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TEND or NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mode.

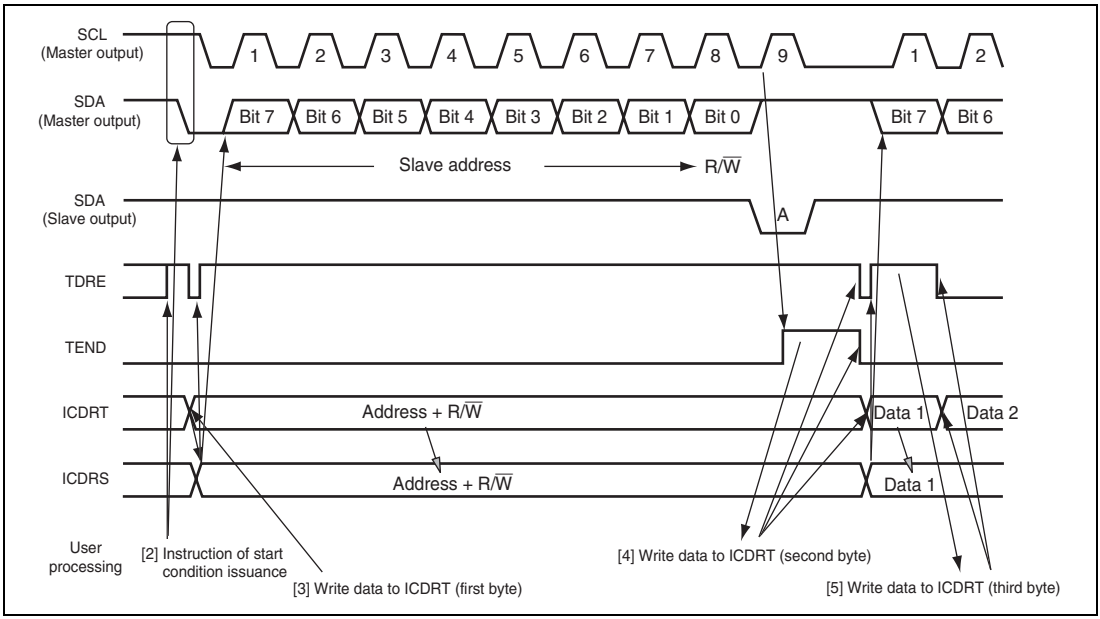


Figure 16.5 Master Transmit Mode Operation Timing (1)

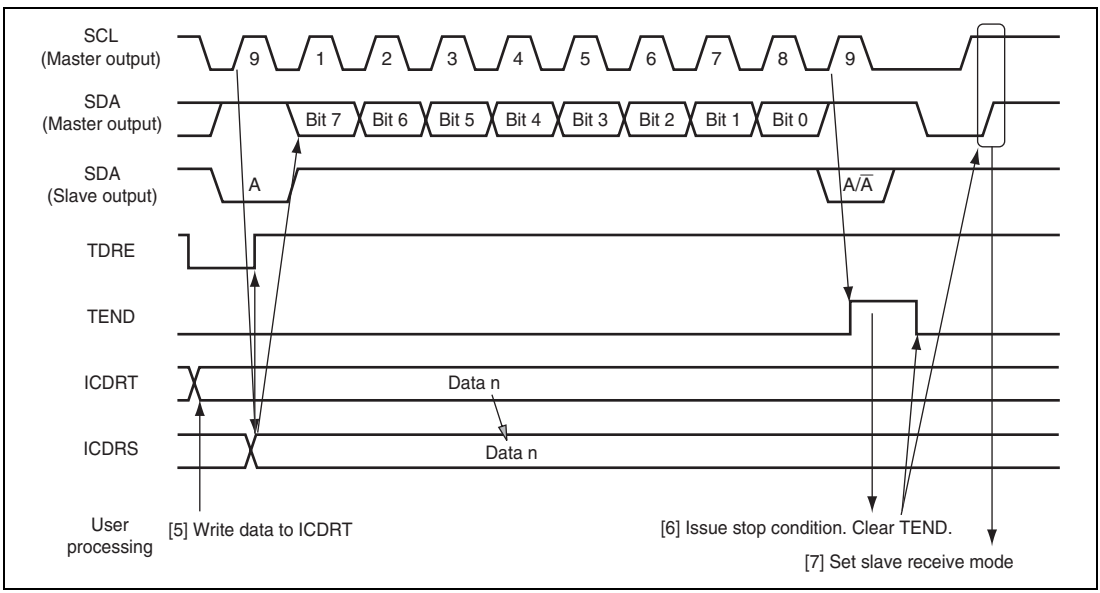


Figure 16.6 Master Transmit Mode Operation Timing (2)

16.4.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data from the slave device, and returns an acknowledge signal. For master receive mode operation timing, refer to figures 16.7 and 16.8. The reception procedure and operations in master receive mode are shown below.

1. Clear the TEND bit in ICSR to 0, then clear the TRS bit in ICCR1 to 0 to switch from master transmit mode to master receive mode. Then, clear the TDRE bit to 0.
2. When ICDRR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. The master device outputs the level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.
3. After the reception of first frame data is completed, the RDRF bit in ICST is set to 1 at the rise of 9th receive clock pulse. At this time, the receive data is read by reading ICDRR, and RDRF is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. If 8th receive clock pulse falls after reading ICDRR by the other processing while RDRF is 1, SCL is fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1 before reading ICDRR. This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, issue the stage condition.
7. When the STOP bit in ICSR is set to 1, read ICDRR. Then clear the RCVD bit to 0.
8. The operation returns to the slave receive mode.

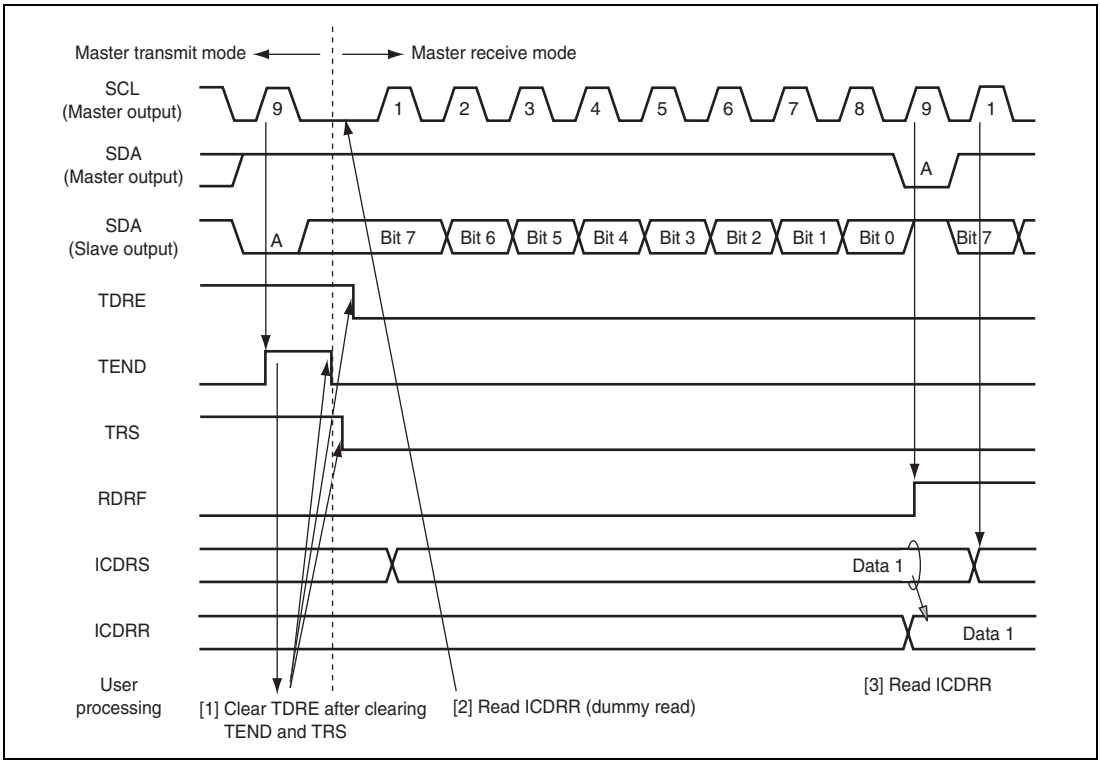


Figure 16.7 Master Receive Mode Operation Timing (1)

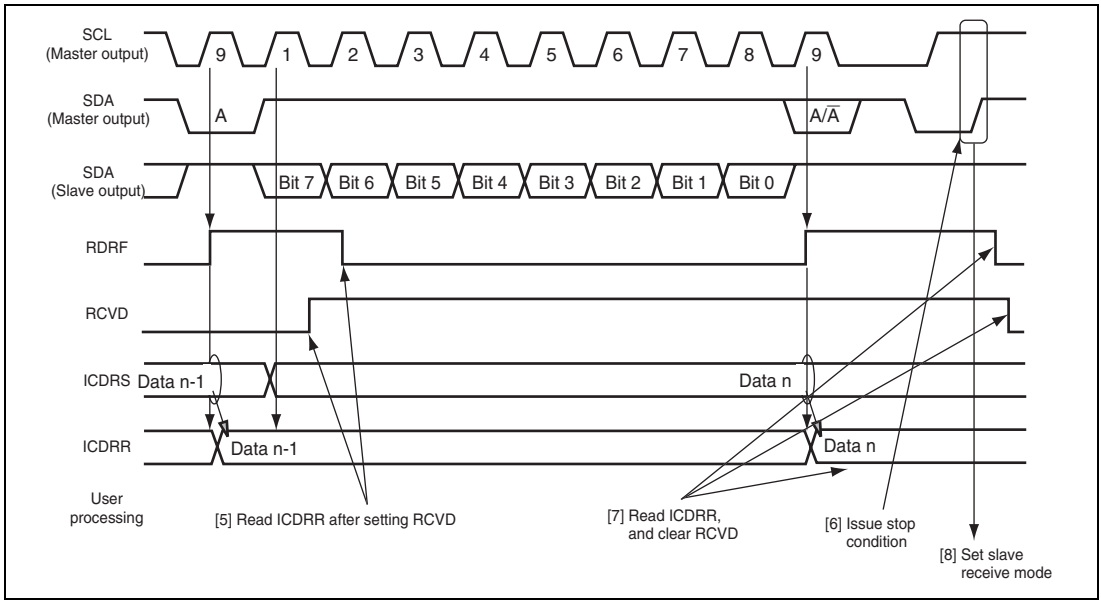


Figure 16.8 Master Receive Mode Operation Timing (2)

16.4.4 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. For slave transmit mode operation timing, refer to figures 16.9 and 16.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At this time, if the 8th bit data (R/$\overline{W}$) is 1, the TRS and ICSR bits in ICCR1 are set to 1, and the mode changes to slave transmit mode automatically. The continuous transmission is performed by writing transmit data to ICDRT every time TDRE is set.
3. If TDRE is set after writing last transmit data to ICDRT, wait until TEND in ICSR is set to 1, with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for the end processing, and read ICDRR (dummy read). SCL is free.
5. Clear TDRE.

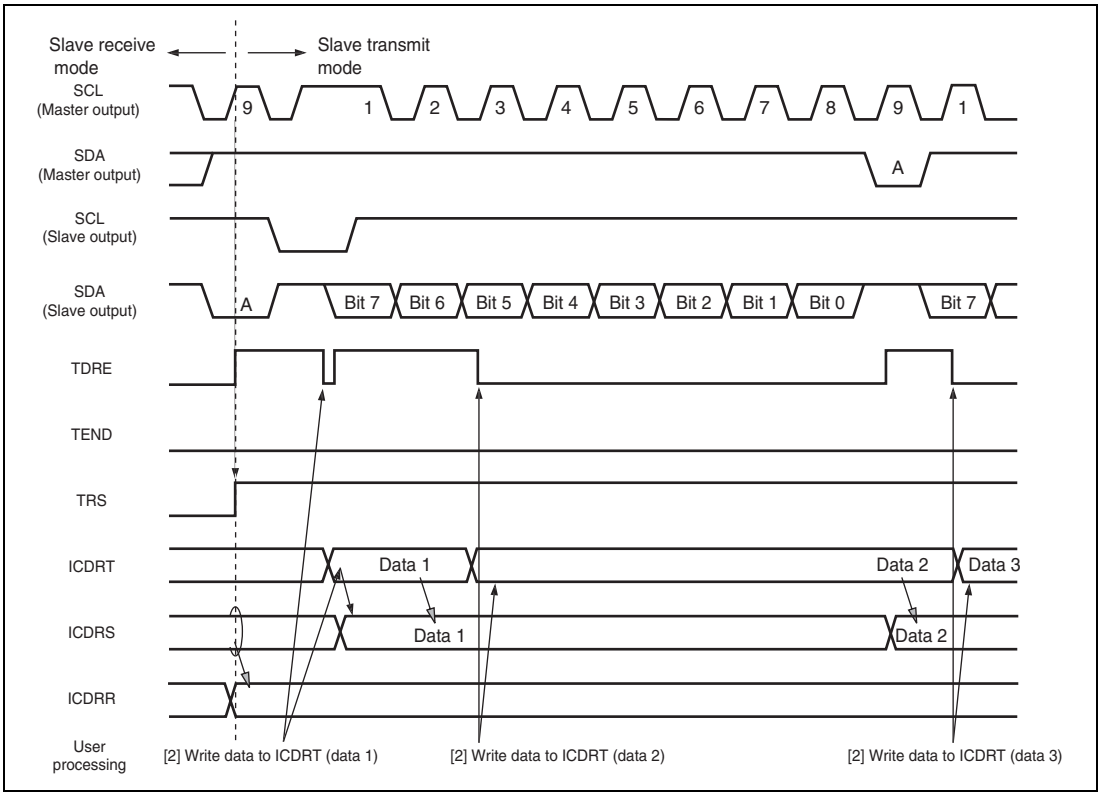


Figure 16.9 Slave Transmit Mode Operation Timing (1)

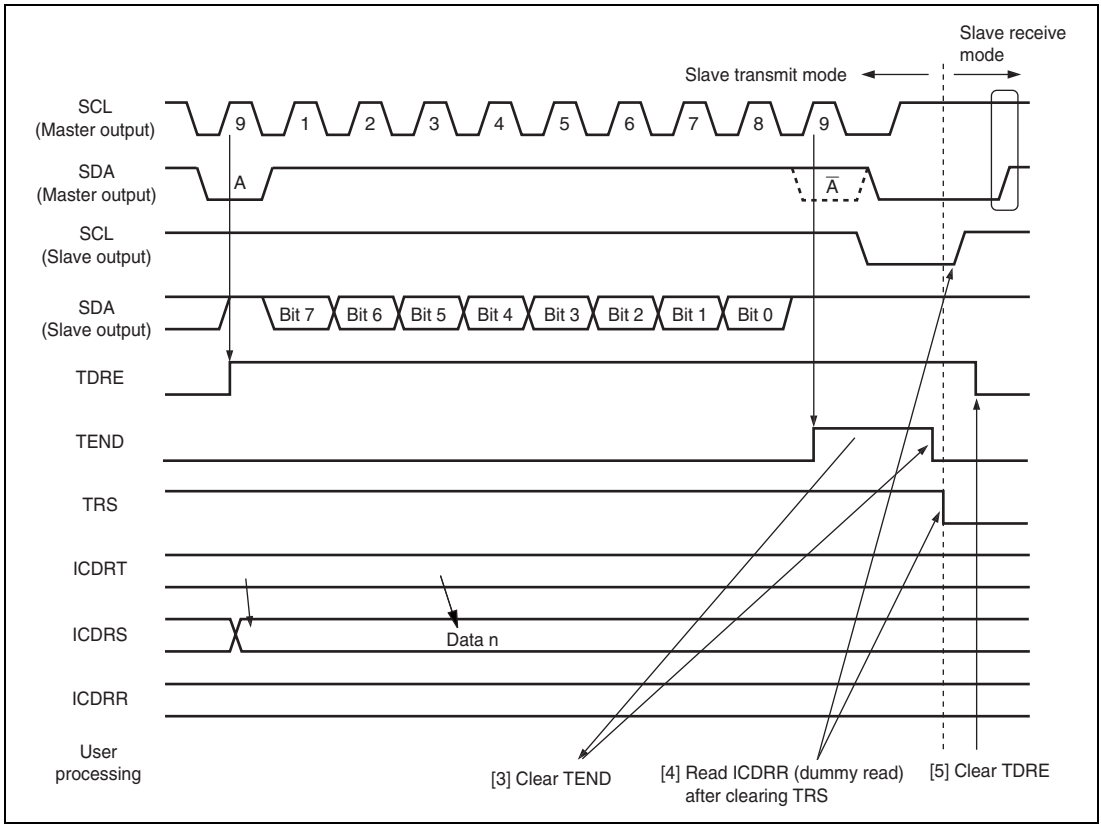


Figure 16.10 Slave Transmit Mode Operation Timing (2)

16.4.5 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. For slave receive mode operation timing, refer to figures 16.11 and 16.12. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (Since the read data show the slave address and R/$\overline{W}$, it is not used.)
3. Read ICDRR every time RDRF is set. If 8th receive clock pulse falls while RDRF is 1, SCL is fixed low until ICDRR is read. The change of the acknowledge before reading ICDRR, to be returned to the master device, is reflected to the next transmit frame.
4. The last byte data is read by reading ICDRR.

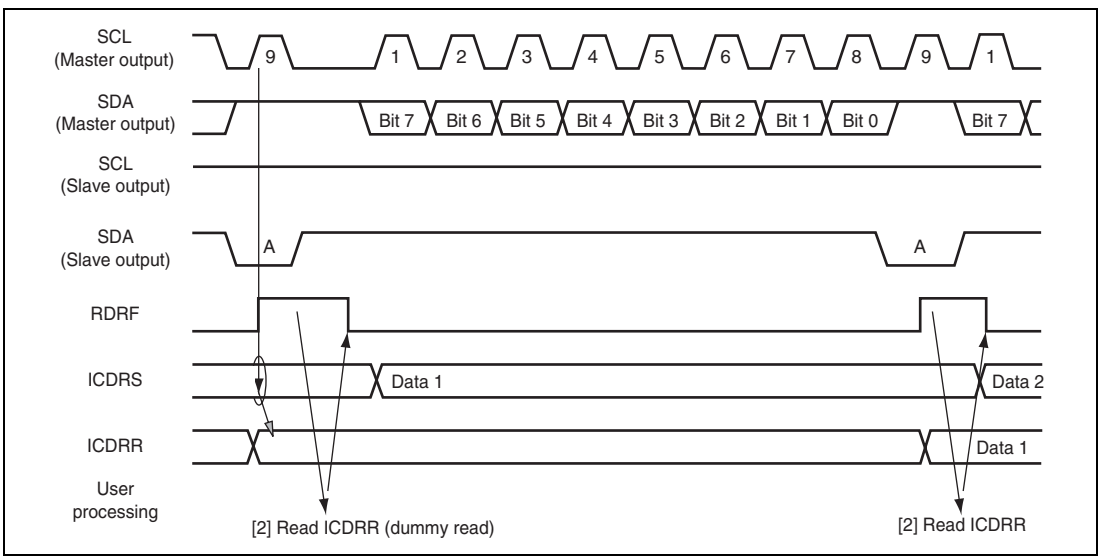


Figure 16.11 Slave Receive Mode Operation Timing (1)

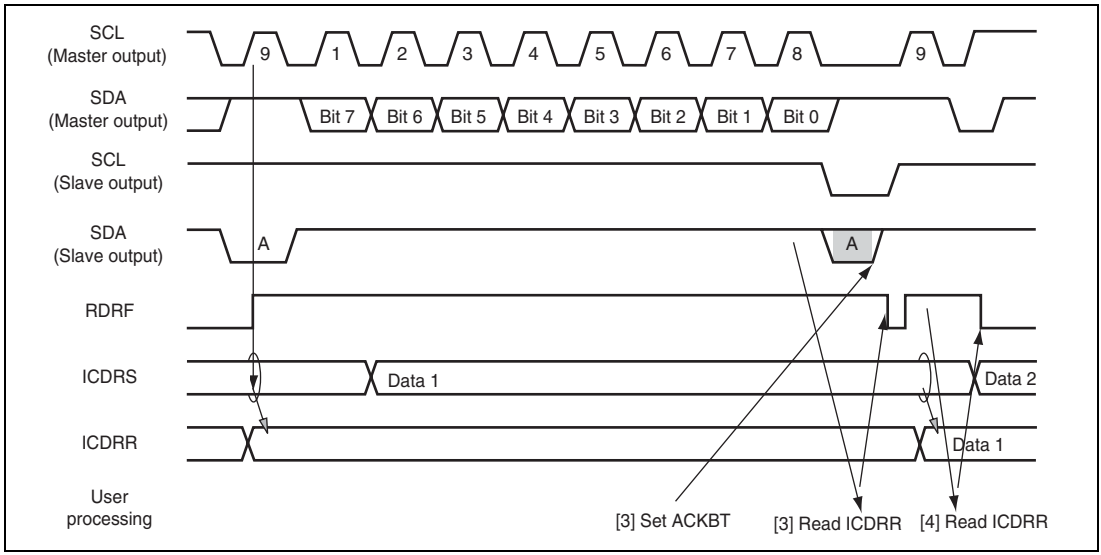


Figure 16.12 Slave Receive Mode Operation Timing (2)

16.4.6 Clock Synchronous Serial Format

This module can be operated with the clock synchronous serial format, by setting the FS bit in SAR to 1. When the MST bit in ICCR1 is 1, the transfer clock output from SCL is selected. When MST is 0, the external clock input is selected.

(1) Data Transfer Format

Figure 16.13 shows the clock synchronous serial transfer format.

The transfer data is output from the rise to the fall of the SCL clock, and the data at the rising edge of the SCL clock is guaranteed. The MLS bit in ICMR sets the order of data transfer, in either the MSB first or LSB first. The output level of SDA can be changed during the transfer wait, by the SDAO bit in ICCR2.

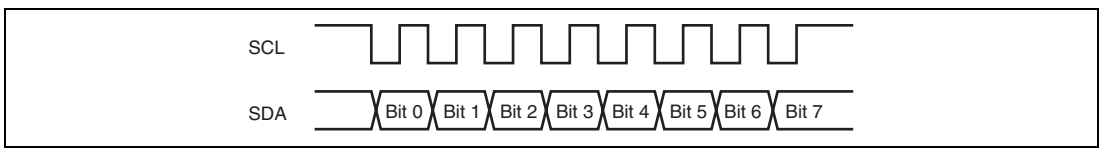


Figure 16.13 Clock Synchronous Serial Transfer Format

(2) Transmit Operation

In transmit mode, transmit data is output from SDA, in synchronization with the fall of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For transmit mode operation timing, refer to figure 16.14. The transmission procedure and operations in transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. Set the TRS bit in ICCR1 to select the transmit mode. Then, TDRE in ICSR is set.
3. Confirm that TDRE has been set. Then, write the transmit data to ICDRT. The data is transferred from ICDRT to ICDRS, and TDRE is set automatically. The continuous transmission is performed by writing data to ICDRT every time TDRE is set. When changing from transmit mode to receive mode, clear TRS while TDRE is 1.

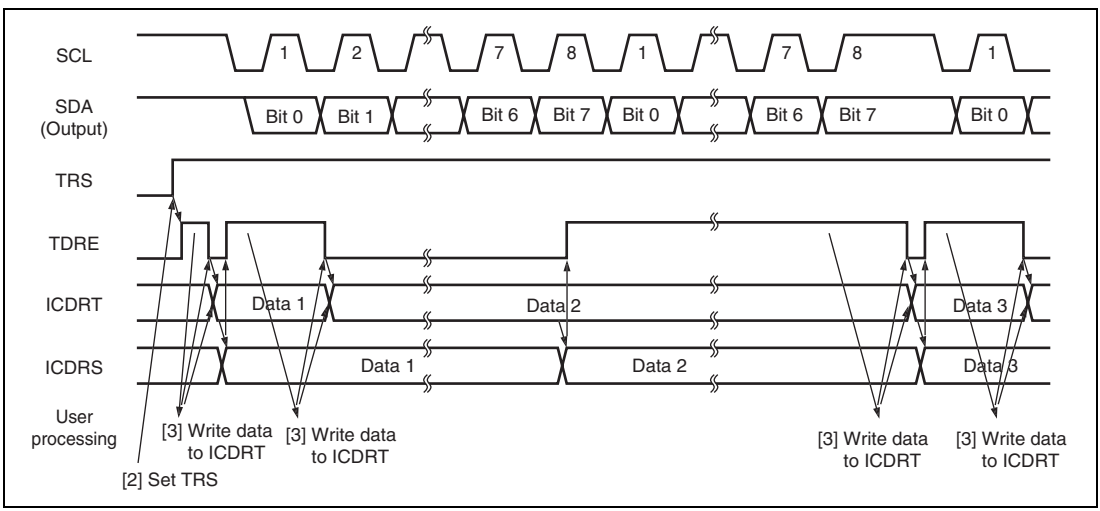


Figure 16.14 Transmit Mode Operation Timing

(3) Receive Operation

In receive mode, data is latched at the rise of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For receive mode operation timing, refer to figure 16.15. The reception procedure and operations in receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. When the transfer clock is output, set MST to 1 to start outputting the receive clock.
3. When the receive operation is completed, data is transferred from ICDRS to ICDRR and RDRF in ICSR is set. When MST = 1, the next byte can be received, so the clock is continually output. The continuous reception is performed by reading ICDRR every time RDRF is set. When the 8th clock is risen while RDRF is 1, the overrun is detected and AL/OVE in ICSR is set. At this time, the previous reception data is retained in ICDRR.
4. To stop receiving when MST = 1, set RCVD in ICCR1 to 1, then read ICDRR. Then, SCL is fixed high after receiving the next byte data.

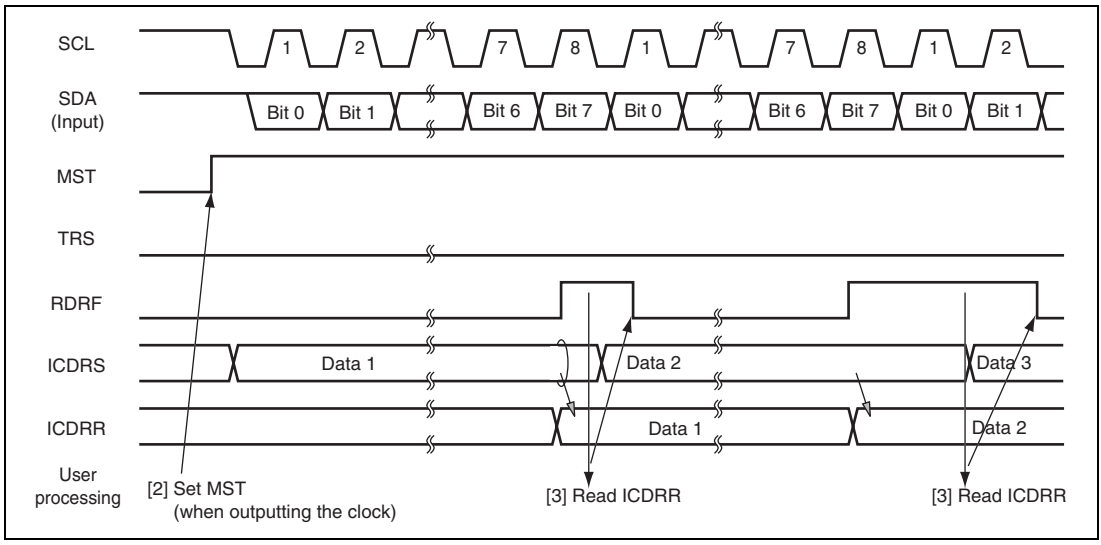


Figure 16.15 Receive Mode Operation Timing

16.4.7 Noise Canceler

The logic levels on the SCL and SDA pins are internally latched via noise cancelers. Figure 16.16 shows a block diagram of the noise canceler circuit.

The noise canceler consists of two cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the system clock, but is not passed forward to the next circuit unless the outputs of both latches agree. If they do not agree, the previous value is held.

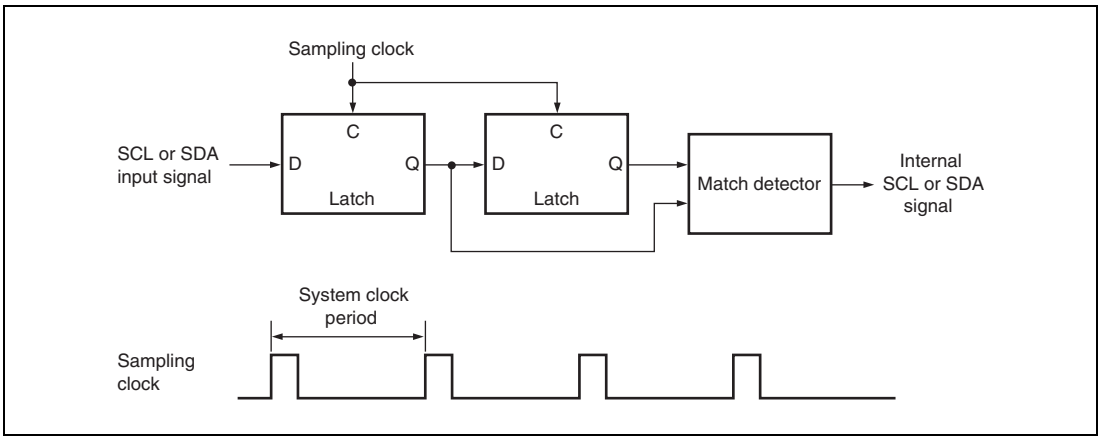


Figure 16.16 Block Diagram of Noise Conceler

16.4.8 Example of Use

Flowcharts in respective modes that use the I²C bus interface 2 are shown in figures 16.17 to 16.20.

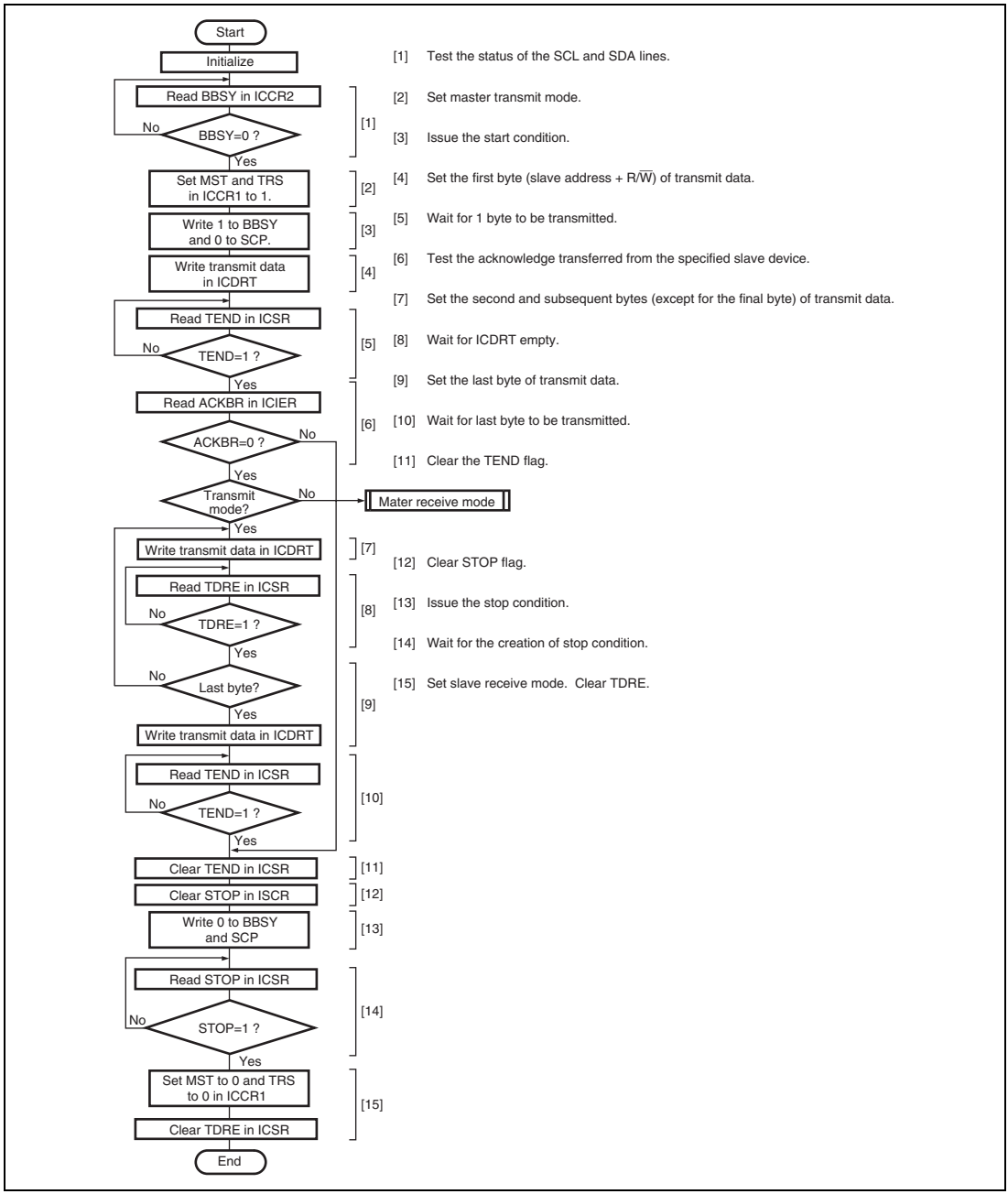
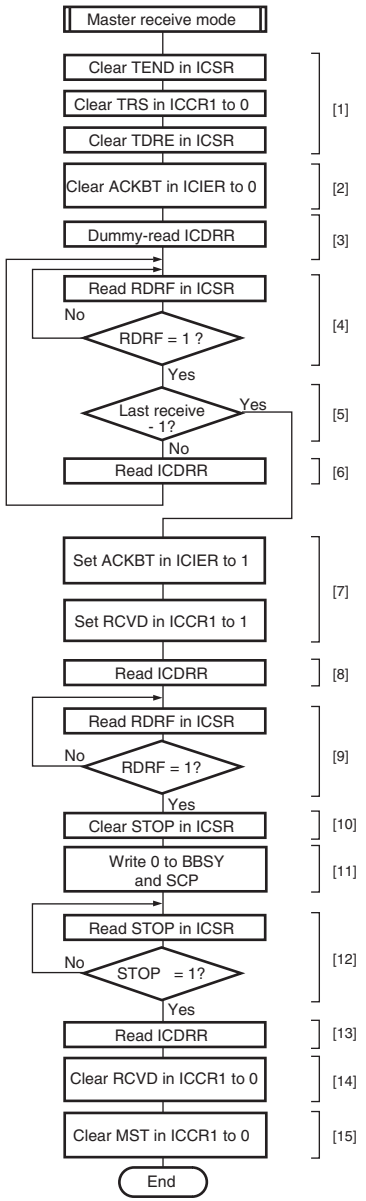


Figure 16.17 Sample Flowchart for Master Transmit Mode



[1] Clear TEND, select master receive mode, and then clear TDRE.*

[2] Set acknowledge to the transmit device.*

[3] Dummy-read ICDDR.*

[4] Wait for 1 byte to be received

[5] Check whether it is the (last receive - 1).

[6] Read the receive data last.

[7] Set acknowledge of the final byte. Disable continuous reception (RCVD = 1).

[8] Read the (final byte - 1) of receive data.

[9] Wait for the last byte to be receive.

[10] Clear STOP flag.

[11] Issue the stop condition.

[12] Wait for the creation of stop condition.

[13] Read the last byte of receive data.

[14] Clear RCVD.

[15] Set slave receive mode.

Note: * Do not activate an interrupt during the execution of steps [1] to [3].

For single-byte reception, skip steps [2] to [6] after step [1]. Then, jump to step [7]. Dummy-read in step [8].

Figure 16.18 Sample Flowchart for Master Receive Mode

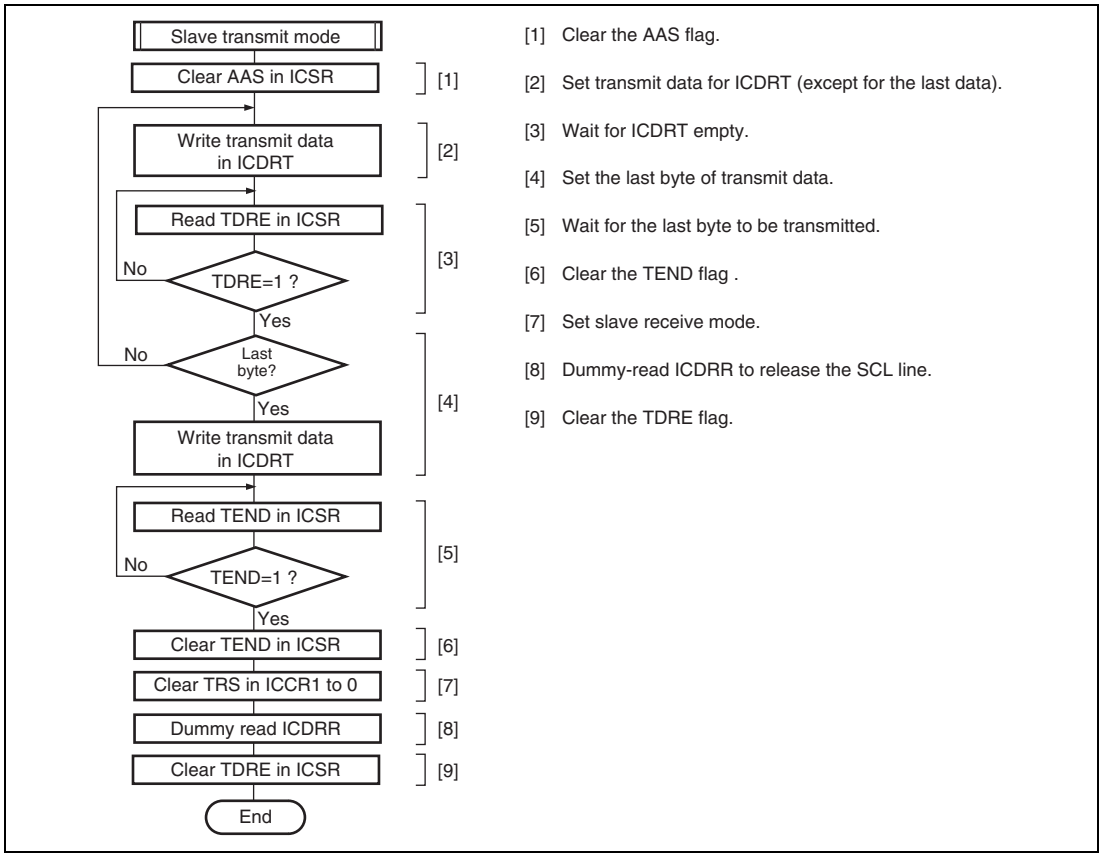


Figure 16.19 Sample Flowchart for Slave Transmit Mode

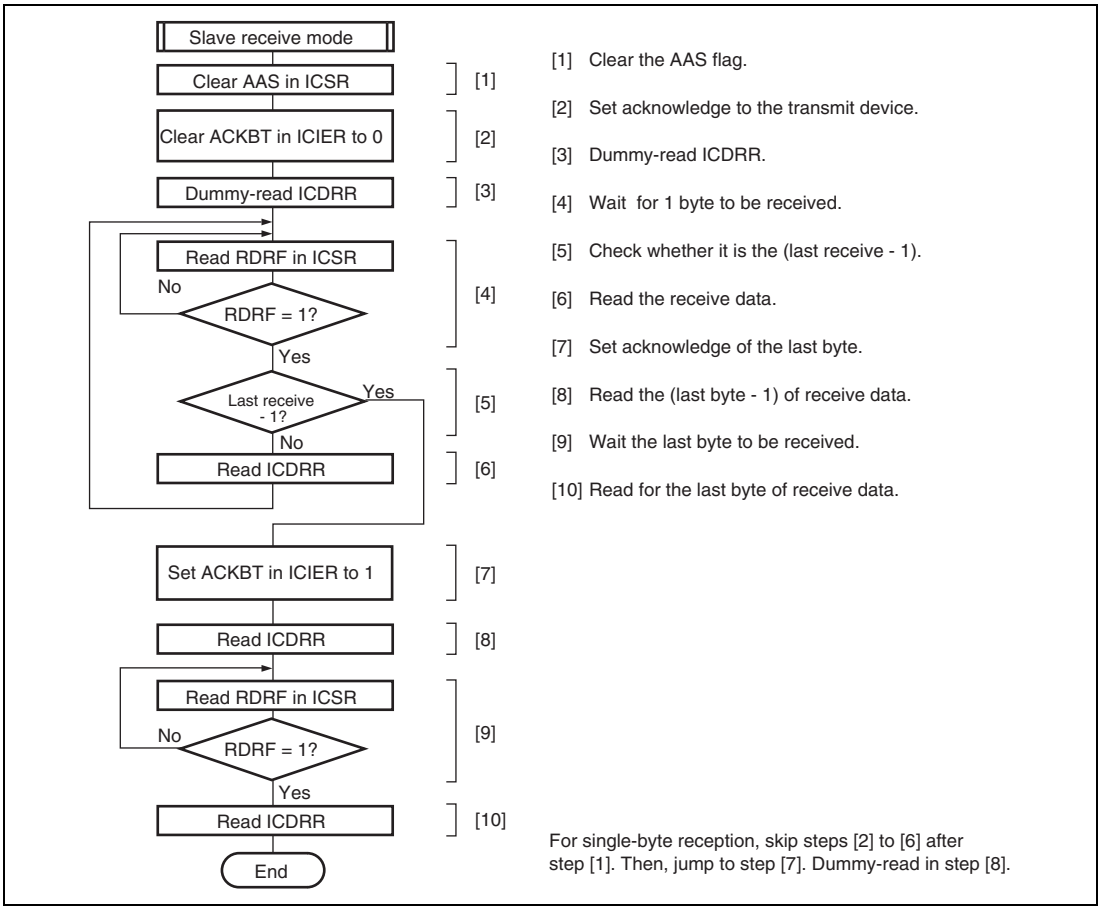


Figure 16.20 Sample Flowchart for Slave Receive Mode

16.5 Interrupt Request

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK receive, STOP recognition, and arbitration lost/overrun. Table 16.3 shows the contents of each interrupt request.

Table 16.3 Interrupt Requests

Interrupt Request	Abbreviation	Interrupt Condition	I²C Mode	Clock Synchronous Mode
Transmit Data Empty	TXI	(TDRE = 1) • (TIE = 1)	○	○
Transmit End	TEI	(TEND = 1) • (TEIE = 1)	○	○
Receive Data Full	RXI	(RDRF = 1) • (RIE = 1)	○	○
STOP Recognition	STPI	(STOP = 1) • (STIE = 1)	○	×
NACK Receive	NAKI	{(NACKF = 1) + (AL = 1)} • (NAKIE = 1)	○	×
Arbitration Lost/Overrun			○	○

When interrupt conditions described in table 16.3 are 1 and the I bit in CCR is 0, the CPU executes interrupt exception processing. Interrupt sources should be cleared in the exception processing. TDRE and TEND are automatically cleared to 0 by writing the transmit data to ICDRT. RDRF are automatically cleared to 0 by reading ICDRR. TDRE is set to 1 again at the same time when transmit data is written to ICDRT. When TDRE is cleared to 0, then an excessive data of one byte may be transmitted.

16.6 Bit Synchronous Circuit

In master mode, this module has a possibility that high level period may be short in the two states described below.

- When SCL is driven to low by the slave device
- When the rising speed of SCL is lowered by the load of the SCL line (load capacitance or pull-up resistance)

Therefore, it monitors SCL and communicates by bit with synchronization.

Figure 16.21 shows the timing of the bit synchronous circuit and table 16.4 shows the time when SCL output changes from low to Hi-Z then SCL is monitored.

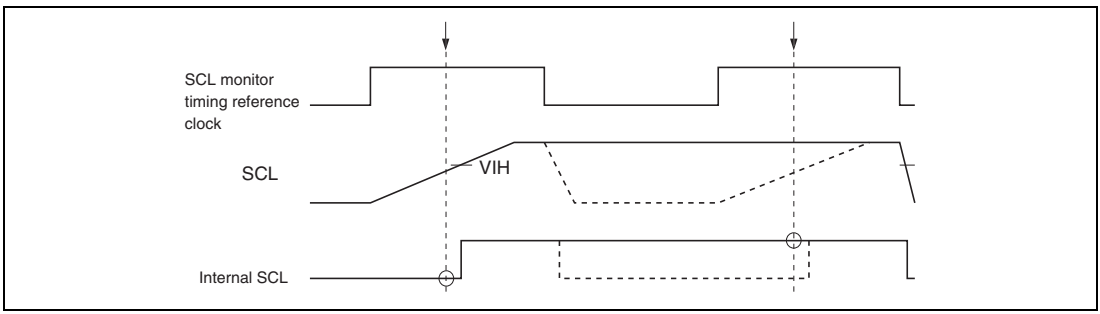


Figure 16.21 Timing of Bit Synchronous Circuit

Table 16.4 Time for Monitoring SCL

CKS3	CKS2	Time for Monitoring SCL
0	0	7.5 tcyc
	1	19.5 tcyc
1	0	17.5 tcyc
	1	41.5 tcyc

16.7 Usage Notes

16.7.1 Note on Issuing Stop Condition and Start (Re-Transmit) Condition

The stop condition or start (re-transmit) condition should be issued after recognizing the falling edge of the ninth clock. The falling edge of the ninth clock can be recognized by checking the SCLO bit in the I2C control register 2 (ICCR2). Note that if the stop condition or start (re-transmit) condition is issued in a particular timing and the situations shown below, these conditions may not correctly output.

1. The rising edge of the SCL becomes less sharp and longer due to the SCL bus load (load capacitor and pull-up resistor) than the period defined in section 16.6, Bit Synchronous Circuit.
2. When the slave device elongates the low level period between the eighth and ninth clocks and activates the bit synchronous circuit.

16.7.2 Note on Setting WAIT Bit in I2C Bus Mode Register (ICMR)

The WAIT bit in the I²C bus mode register (ICMR) should be set to 0. Note that if the WAIT bit is set to 1, when a slave device holds the SCL signal low more than one transfer clock cycle during the eighth clock, the high level period of the ninth clock may be shorter than a given period.

16.7.3 Restriction on Transfer Rate Setting in Multimaster Operation

In multimaster operation, if the IIC transfer rate setting in this LSI is slower than those of the other masters, SCL may be output with an unexpected width. To avoid this phenomenon, set the transfer rate by 1/1.8 or faster than the fastest rate of the other masters. For example, if the fastest transfer rate of the other masters is set to 400 kbps, the IIC transfer rate in this LSI should be set to 223 kbps (= 400/1.18) or more.

16.7.4 Restriction on the Use of Bit Manipulation Instructions for MST and TRS Setting in Multimaster Operation

In multimaster operation, if the master transmit is set with bit manipulation instructions in the order from the MST bit to the TRS bit, the AL bit in the ICSR register will be set to 1 but the master transmit mode (MST = 1, TRS = 1) may be set, depending on the arbitration lost timing. To avoid this phenomenon, the following actions should be performed:

- In multimaster operation, use the MOV instruction to set bits MST and TRS.
- When arbitration is lost, confirm the contents of bits MST and TRS. If the contents are other than MST = 0 and TRS = 0, set MST = 0 and TRS = 0 again.

16.7.5 Usage Note on Master Receive Mode

In master receive mode, SCL is fixed low on the falling edge of the 8th clock while the RDRF bit is set to 1. When ICDRR is read around the falling edge of the 8th clock, the clock is only fixed low in the 8th clock of the next round of data reception. The SCL is then released from its fixed state without reading ICDRR and the 9th clock is output. As a result, some receive data is lost. To avoid this phenomenon, the following actions should be performed:

- Read ICDRR in master receive mode before the rising edge of the 8th clock.
- Set RCVD to 1 in master receive mode and perform communication in units of one byte.

Section 17 A/D Converter

This LSI includes a successive approximation type 10-bit A/D converter that allows up to six analog input channels to be selected. The block diagram of the A/D converter is shown in figure 17.1.

17.1 Features

- 10-bit resolution
- Six input channels
- High-speed conversion: 12.4 μs per channel (at 10-MHz operation)
- Sample and hold function
- Conversion start method

  A/D conversion can be started by software and external trigger.
- Interrupt source

  An A/D conversion end interrupt request can be generated.
- Use of module standby mode enables this module to be placed in standby mode independently when not used. (The A/D converter is halted as the initial value. For details, refer to section 5.4, Module Standby Function.)

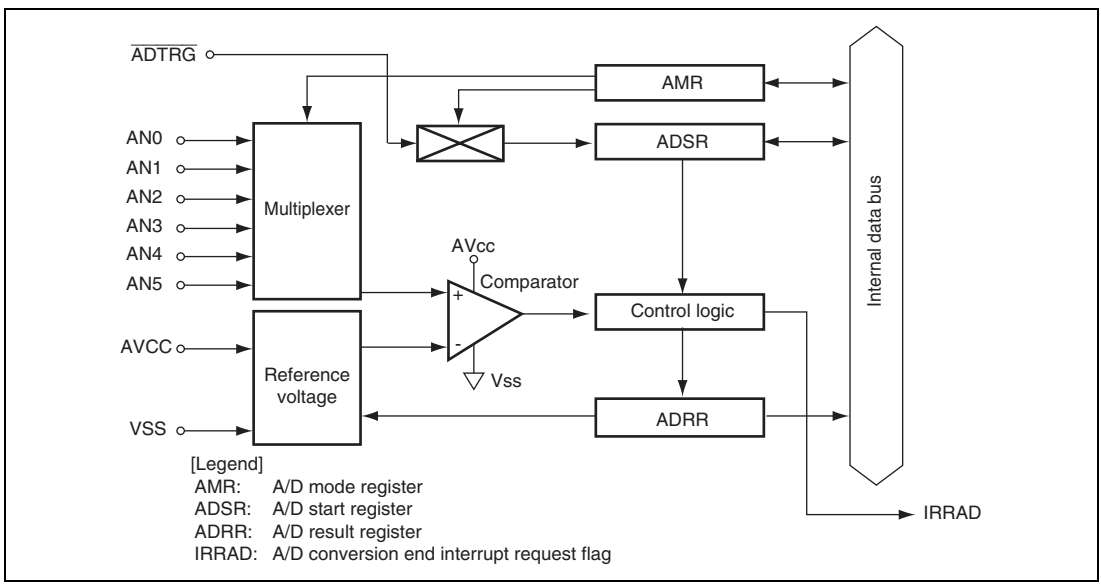


Figure 17.1 Block Diagram of A/D Converter

17.2 Input/Output Pins

Table 17.1 shows the pin configuration of the A/D converter.

Table 17.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AVcc	Input	Power supply and reference voltage of analog part
Ground pin	Vss	Input	Ground and reference voltage
Analog input pin 0	AN0	Input	Analog input pins
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
External trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input that controls the A/D conversion start.

17.3 Register Descriptions

The A/D converter has the following registers.

- A/D result register (ADRR)
- A/D mode register (AMR)
- A/D start register (ADSR)

17.3.1 A/D Result Register (ADRR)

ADRR is a 16-bit read-only register that stores the results of A/D conversion. The data is stored in the upper 10 bits of ADRR. ADRR can be read by the CPU at any time, but the ADRR value during A/D conversion is undefined. After A/D conversion is completed, the conversion result is stored as 10-bit data, and this data is retained until the next conversion operation starts. The initial value of ADRR is undefined. This register must be read in words.

17.3.2 A/D Mode Register (AMR)

AMR sets the A/D conversion time, and selects the external trigger and analog input pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
6	TRGE	0	R/W	External Trigger Select Enables or disables the A/D conversion start by the external trigger input. 0: Disables the A/D conversion start by the external trigger input. 1: Starts A/D conversion at the rising or falling edge of the $\overline{\text{ADTRG}}$ pin The edge of the $\overline{\text{ADTRG}}$ pin is selected by the ADTRGNEG bit in IEGR.
5	CKS1	0	R/W	Clock Select
4	CKS0	0	R/W	Select the A/D conversion clock source. 00: ϕ/8 (conversion time = 124 states (max.) (reference clock = ϕ) 01: ϕ/4 (conversion time = 62 states (max.) (reference clock = ϕ) 10: ϕ/2 (conversion time = 31 states (max.) (reference clock = ϕ) 11: ϕw/2 (conversion time = 31 states (max.) (reference clock = ϕ_{SUB}) While CKS1 and CKS0 are all 1 in subactive or subsleep mode, the A/D converter can be used only when the CPU operating clock is ϕw.

Bit	Bit Name	Initial Value	R/W	Description
3	CH3	0	R/W	Channel Select 3 to 0
2	CH2	0	R/W	Select the analog input channel.
1	CH1	0	R/W	00xx: No channel selected
0	CH0	0	R/W	0100: AN0
				0101: AN1
				0110: AN2
				0111: AN3
				1000: AN4
				1001: AN5
				101x: No channel selected
				11xx: No channel selected
				The channel selection should be made while the ADSF bit is cleared to 0.

[Legend] x: Don't care.

17.3.3 A/D Start Register (ADSR)

ADSR starts and stops the A/D conversion.

Bit	Bit Name	Initial Value	R/W	Description
7	ADSF	0	R/W	When this bit is set to 1, A/D conversion is started. When conversion is completed, the converted data is set in ADRR and at the same time this bit is cleared to 0. If this bit is written to 0, A/D conversion can be forcibly terminated.
6	LADS	0	R/W	Ladder Resistance Select
				0: Ladder resistance operates while the A/D converter is idle.
				1: Ladder resistance is halted while the A/D converter is idle.
				The ladder resistance is always halted in standby mode, watch mode, or module standby mode, and at a reset.
5 to 0	—	All 1	—	Reserved
				These bits are always read as 1 and cannot be modified.

17.4 Operation

The A/D converter operates by successive approximation with 10-bit resolution. When changing the conversion time or analog input channel, in order to prevent incorrect operation, first clear the bit ADSF to 0 in ADSR.

17.4.1 A/D Conversion

1. A/D conversion is started from the selected channel when the ADSF bit in ADSR is set to 1, according to software.
2. When A/D conversion is completed, the result is transferred to the A/D result register.
3. On completion of conversion, the IRRAD flag in IRR2 is set to 1. If the IENAD bit in IENR2 is set to 1 at this time, an A/D conversion end interrupt request is generated.
4. The ADSF bit remains set to 1 during A/D conversion. When A/D conversion ends, the ADSF bit is automatically cleared to 0 and the A/D converter enters the wait state.

17.4.2 External Trigger Input Timing

The A/D converter can also start A/D conversion by input of an external trigger signal. External trigger input is enabled at the $\overline{\text{ADTRG}}$ pin when the ADTSTCHG bit in PMRB is set to 1* and the TRGE bit in AMR is set to 1. Then when the input signal edge designated in the ADTRGNEG bit in IEGR is detected at the $\overline{\text{ADTRG}}$ pin, the ADSF bit in ADSR will be set to 1, starting A/D conversion.

Figure 17.2 shows the timing.

Note: * The $\overline{\text{ADTRG}}$ input pin is shared with the TEST pin. Therefore when the pin is used as the $\overline{\text{ADTRG}}$ pin, reset should be cleared while the 0-fixed signal is input to the TEST pin. Then the ADTSTCHG bit should be set to 1 after the TEST signal is fixed.

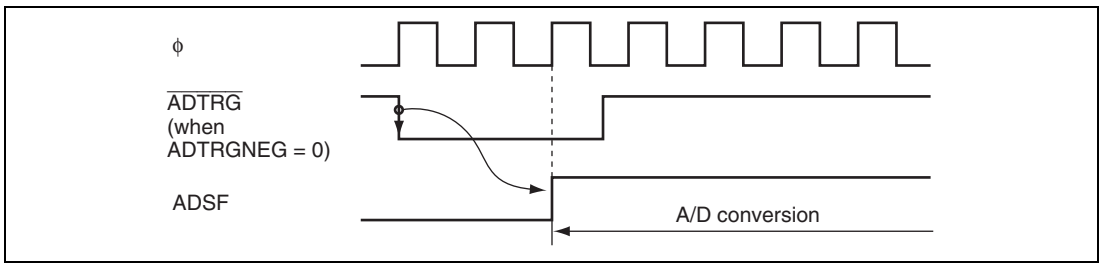


Figure 17.2 External Trigger Input Timing

17.4.3 Operating States of A/D Converter

Table 17.2 shows the operating states of the A/D converter.

Table 17.2 Operating States of A/D Converter

Operating Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby	Module Standby
AMR	Reset	Functions	Retained	Retained	Functions/ Retained*2	Retained	Retained	Retained
ADSR	Reset	Functions	Functions	Retained	Functions/ Retained*2	Functions/ Retained*2	Retained	Retained
ADRR	Retained*1	Functions	Functions	Retained	Functions/ Retained*2	Functions/ Retained*2	Retained	Retained

Notes: 1. Undefined at a power-on reset.
2. Function if ϕw/2 is selected as the internal clock. Halted and retained otherwise.

17.5 Example of Use

An example of how the A/D converter can be used is given below, using channel 1 (pin AN1) as the analog input channel. Figure 17.3 shows the operation timing.

1. Bits CH3 to CH0 in the A/D mode register (AMR) are set to 0101, making pin AN1 the analog input channel. A/D interrupts are enabled by setting bit IENAD to 1, and A/D conversion is started by setting bit ADSF to 1.
2. When A/D conversion is completed, bit IRRAD is set to 1, and the A/D conversion result is stored in ADRR. At the same time bit ADSF is cleared to 0, and the A/D converter goes to the idle state.
3. Bit IENAD = 1, so an A/D conversion end interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The A/D conversion result is read and processed.
6. The A/D interrupt handling routine ends.

If bit ADSF is set to 1 again afterward, A/D conversion starts and steps 2 through 6 take place.

Figures 17.4 and 17.5 show flowcharts of procedures for using the A/D converter.

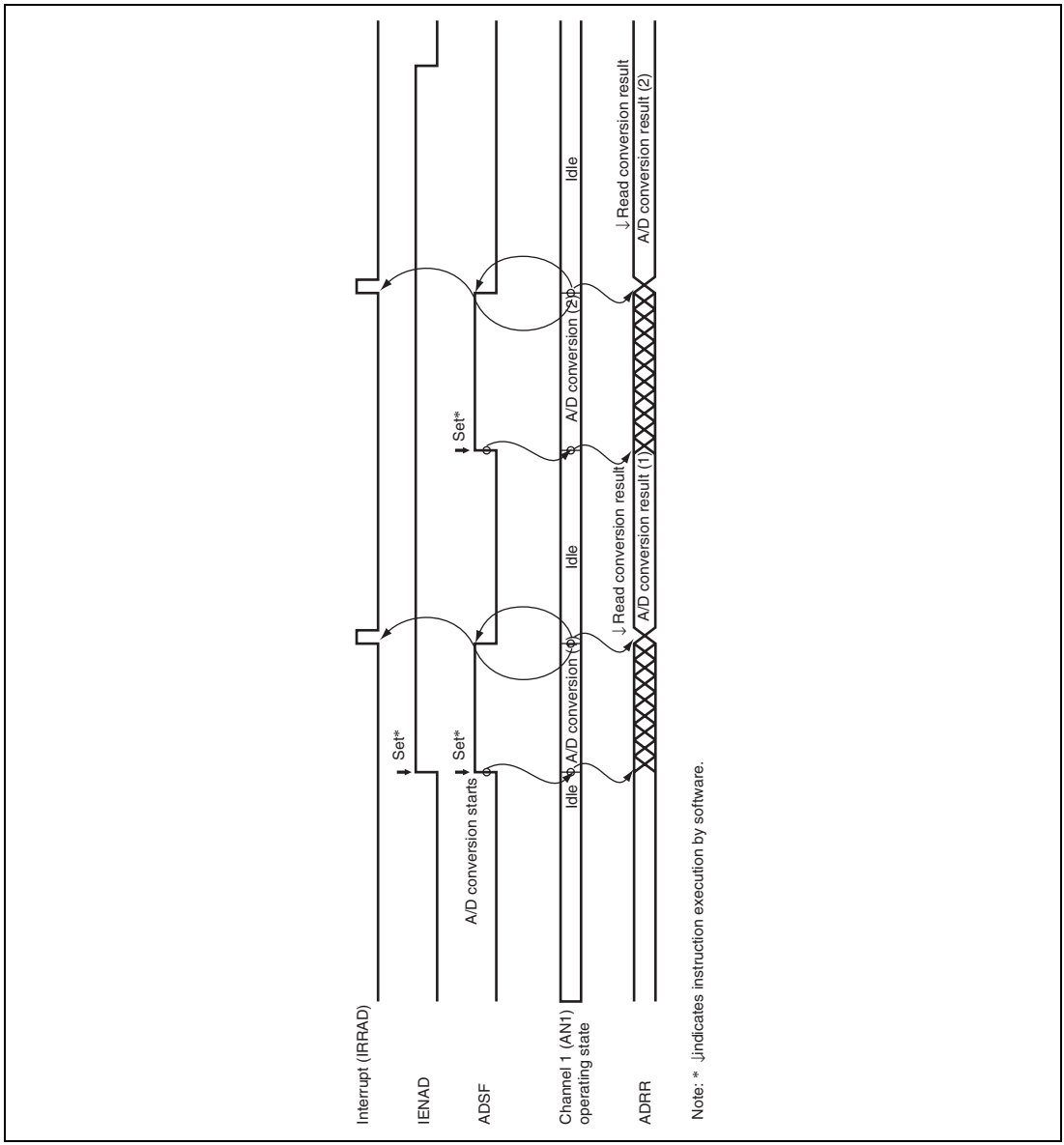


Figure 17.3 Example of A/D Conversion Operation

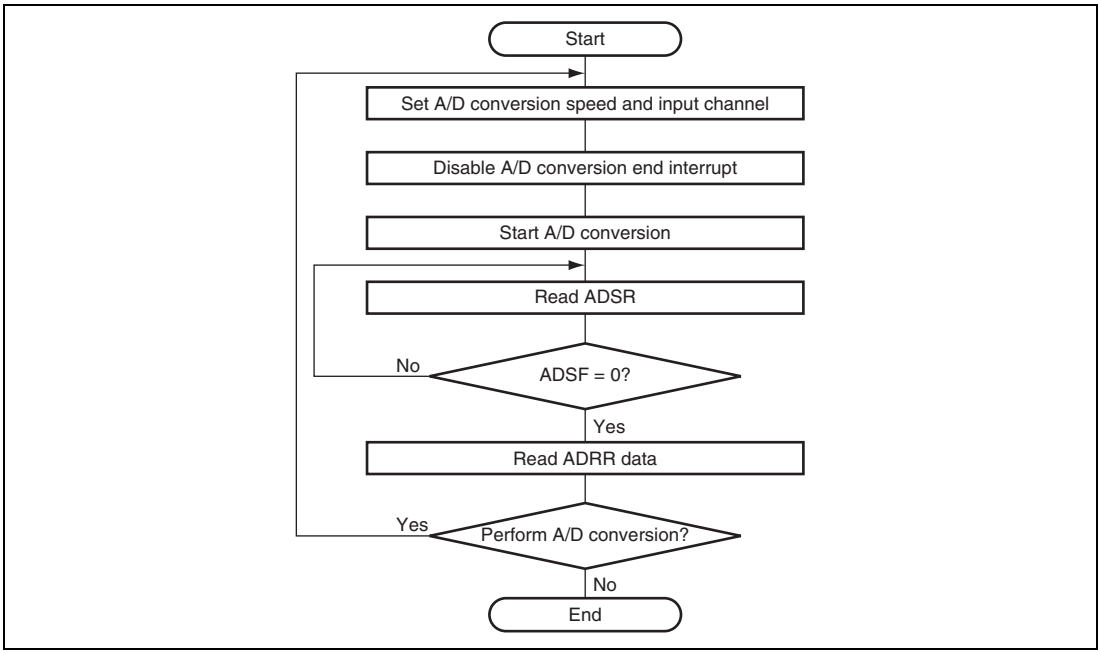


Figure 17.4 Flowchart of Procedure for Using A/D Converter (Polling by Software)

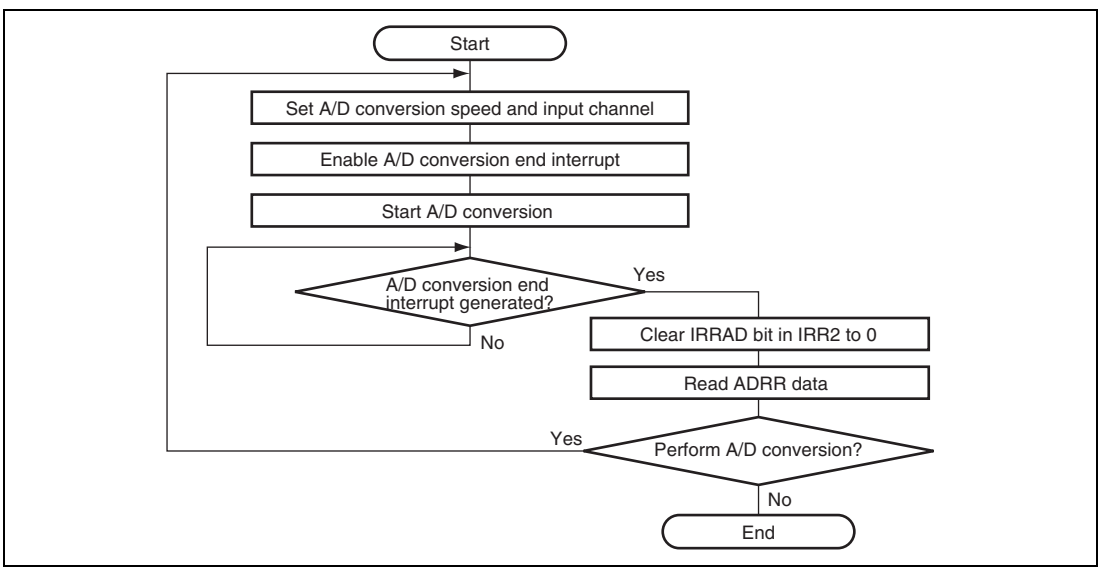


Figure 17.5 Flowchart of Procedure for Using A/D Converter (Interrupts Used)

17.6 A/D Conversion Accuracy Definitions

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital output codes
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 17.6).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value 0000000000 to 0000000001 (see figure 17.7).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from 1111111110 to 1111111111 (see figure 17.7).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristics between zero voltage and full-scale voltage. Does not include offset error, full-scale error, or quantization error.
- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes offset error, full-scale error, quantization error, and nonlinearity error.

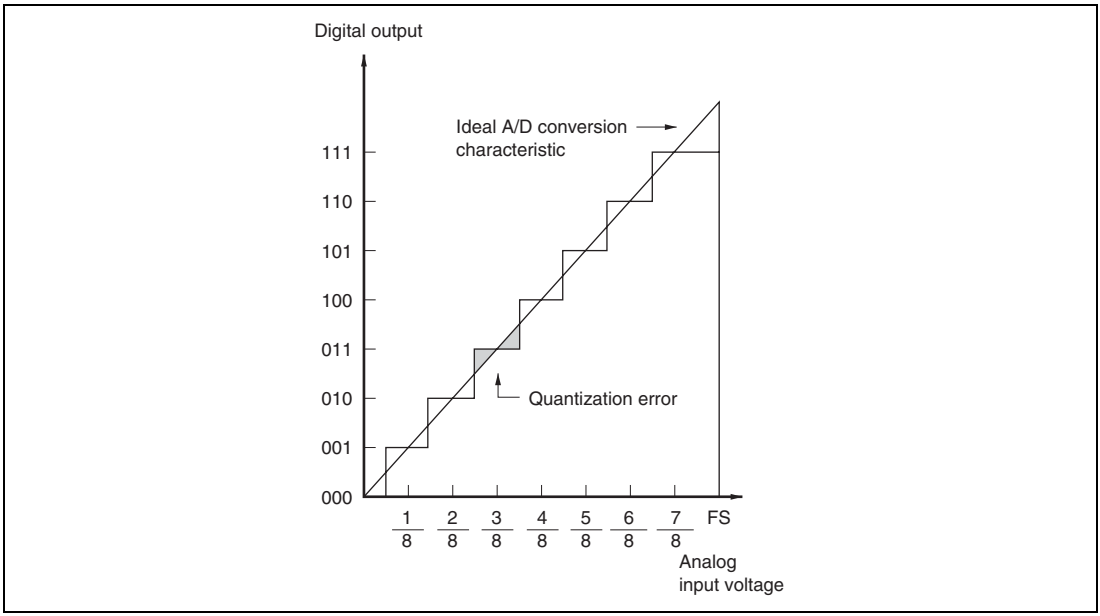


Figure 17.6 A/D Conversion Accuracy Definitions (1)

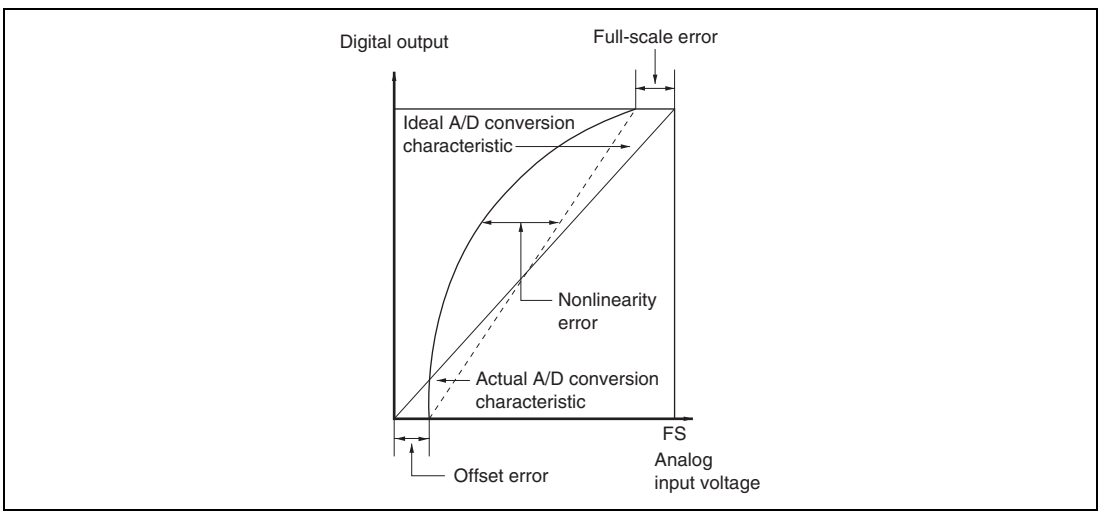


Figure 17.7 A/D Conversion Accuracy Definitions (2)

17.7 Usage Notes

17.7.1 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion accuracy is guaranteed for an input signal for which the signal source impedance is 10 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 10 kΩ, charging may be insufficient and it may not be possible to guarantee A/D conversion accuracy. However, with a large capacitance provided externally, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored. However, as a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/μs or greater) (see figure 17.8). When converting a high-speed analog signal, a low-impedance buffer should be inserted.

17.7.2 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute accuracy. Be sure to make the connection to an electrically stable GND.

Care is also required to ensure that filter circuits do not interfere with digital signals or act as antennas on the mounting board.

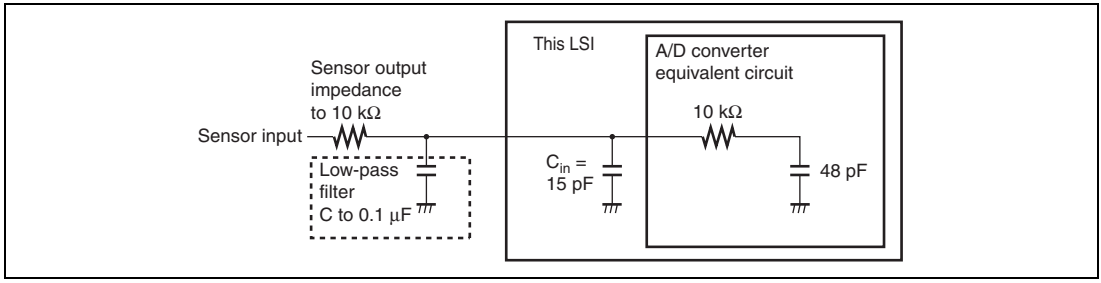


Figure 17.8 Example of Analog Input Circuit

17.7.3 Usage Notes

1. ADRR should be read only when the ADSF bit in ADSR is cleared to 0.
2. Changing the digital input signal at an adjacent pin during A/D conversion may adversely affect conversion accuracy.
3. When A/D conversion is started after clearing module standby mode, wait for 10ϕ clock cycles before starting A/D conversion.

Section 18 Comparators

This LSI includes comparators to compare the input voltage and reference voltage.

The block diagram of the comparators is shown in figure 18.1.

18.1 Features

- Reference voltage can be specified as internal power supply or external input (VCref).
- When the internal power supply is selected as the reference voltage, programmable selection of sixteen types of voltages is possible.
- When the internal power supply is selected, the hysteresis characteristics of the comparison result can be selected.
- Two analog input channels

  Each channel includes its own comparator.
- Use of module standby mode enables this module to be placed in standby mode independently when not used. (A comparator is halted as the initial value. For details, refer to section 5.4, Module Standby Function.)

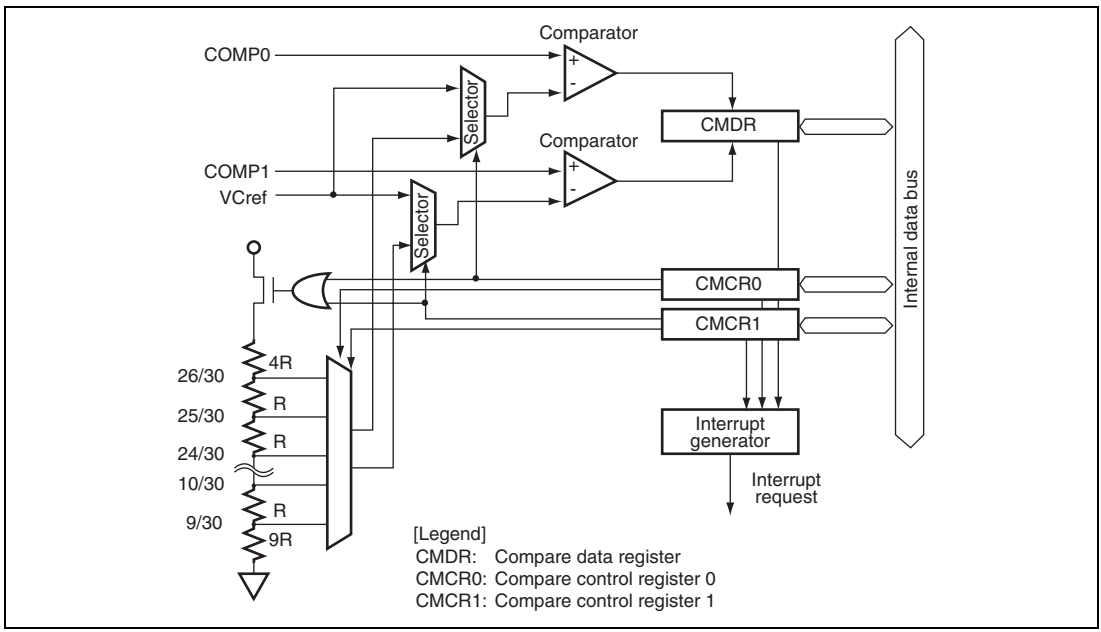


Figure 18.1 Block Diagram of Comparators

18.2 Input/Output Pins

Table 18.1 shows the pin configuration of the comparators.

Table 18.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Comparator reference voltage	VCref	Input	Comparator reference voltage pin (external input)
Analog input channel 0	COMP0	Input	Comparator analog input pin 0
Analog input channel 1	COMP1	Input	Comparator analog input pin 1

18.3 Register Descriptions

The comparators have the following registers. For details on register addresses and register states during each processing, refer to section 20, List of Registers.

- Compare control registers 0, 1 (CMCR0, CMCR1)
- Compare data register (CMDR)

18.3.1 Compare Control Registers 0, 1 (CMCR0, CMCR1)

CMCR0 and CMCR1 control the comparators.

Bit	Bit Name	Initial Value	R/W	Description
7	CME	0	R/W	Comparator Enable 0: Comparator halted 1: Comparator operates
6	CMIE	0	R/W	Comparator Interrupt Enable 0: Disables a comparator interrupt 1: Enables a comparator interrupt
5	CMR	0	R/W	Comparator Reference Voltage Select 0: Selects internal power supply as reference voltage 1: Reference voltage is input from VCref pin For the combination of the CMR and CMLS bits.

Bit	Bit Name	Initial Value	R/W	Description
4	CMLS	0	R/W	Comparator Hysteresis Select 0: Selects non-hysteresis 1: Selects hysteresis When CMR = 1, clear this bit to 0. For the combination of the CMR and CMLS bits.
3	CRS3	0	R/W	Internal Reference Voltage Select When CMR = 0 and CMLS = 0, the electric potential of V_{IL} is selected as the internal power supply. When CMR = 0 and CMLS = 1, V_{IL} will be as follows. When CMR = 1, CRS3 to CRS0 settings are disabled.
2	CRS2	0	R/W	
1	CRS1	0	R/W	
0	CRS0	0	R/W	

VIH	VIL
0000: 11/30Vcc	9/30Vcc
0001: 12/30Vcc	10/30Vcc
0010: 13/30Vcc	11/30Vcc
0011: 14/30Vcc	12/30Vcc
0100: 15/30Vcc	13/30Vcc
0101: 16/30Vcc	14/30Vcc
0110: 17/30Vcc	15/30Vcc
0111: 18/30Vcc	16/30Vcc
1000: 19/30Vcc	17/30Vcc
1001: 20/30Vcc	18/30Vcc
1010: 21/30Vcc	19/30Vcc
1011: 22/30Vcc	20/30Vcc
1100: 23/30Vcc	21/30Vcc
1101: 24/30Vcc	22/30Vcc
1110: 25/30Vcc	23/30Vcc
1111: 26/30Vcc	24/30Vcc

For the selectable range by the CRS bits, see section 21, Electrical Characteristics.

Table 18.2 Combination of CMR and CMLS Bits

CMR	CMLS	Function
0	0	Compares the internal power supply (voltage set for V_{IH} by the CRS3 to CRS0 bits) and electric potential of the COMP pin. No hysteresis.
	1	Compares the internal power supply and electric potential of the COMP pin. With hysteresis. V_{IH} and V_{IL} are set by the CRS3 to CRS0 bits.
1	0	Compares the electric potential of the VCref and COMP pins. No hysteresis.
	1	Setting prohibited

18.3.2 Compare Data Register (CMDR)

CMDR stores the result of comparing the analog input pin and reference voltage.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0.
5	CMF1	0	R/(W)*[1]	COMP1 Interrupt Flag [Setting condition] When COMP1 interrupt occurs [Clearing condition] 0 is written to CMF1 after reading CMF1 = 1
4	CMF0	0	R/(W)*[1]	COMP0 Interrupt Flag [Setting condition] When COMP0 interrupt occurs [Clearing condition] 0 is written to CMF0 after reading CMF0 = 1
3, 2	—	All 0	—	Reserved These bits are always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
1	CDR1	—*[2]	R	[Setting condition] COMP1 pin > Reference voltage [Clearing condition] COMP1 pin ≤ Reference voltage
0	CDR0	—*[2]	R	[Setting condition] COMP0 pin > Reference voltage [Clearing condition] COMP0 pin ≤ Reference voltage

Notes: 1. Only 0 can be written to clear the flag.
2. Depends on the pin state and reference voltage.

18.4 Operation

18.4.1 Operation Sequence

The operation sequence of a comparator is as follows:

1. When using VCref, the pins to be used are enabled by the corresponding port mode registers. For details, see section 8, I/O Ports.
2. Select the reference voltage (CMR setting: internal power supply or VCref).
   When the internal power supply is selected as the reference voltage, select the hysteresis characteristics (CMLS setting) and reference voltage (CRS3 to CRS0 setting).
3. Set the comparator enable bit (CME).
4. After setting CME, wait for the conversion time (see section 21, Electrical Characteristics) so that the comparator becomes stabilized.
5. Read from CDR.
6. After reading the CMF flag, write 0 to it (reading the CMF flag can be performed simultaneously with step 5).
7. If an interrupt is to be generated, set the comparator interrupt enable bit (CMIE).

Note: Steps 2 and 3 can be done simultaneously by writing to the entire register.

18.4.2 Hysteresis Characteristics of Comparator

Figure 18.2 shows CDR when hysteresis is or is not selected by the CMLS bit CMCR and the input voltage to the COMP pins. The hysteresis characteristics for the comparison result (CDR) by the comparator can be selected by the CMLS bit, as shown in figure 18.2.

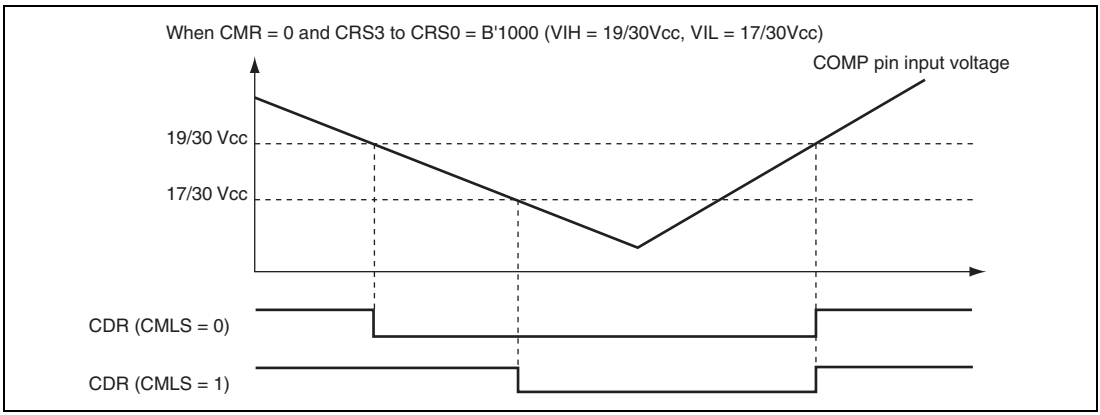


Figure 18.2 Hysteresis/Non-Hysteresis Selection by CDR

18.4.3 Interrupt Setting

When the CDR bit is read while the comparator interrupt is enabled and both the CME and CMIE bits are set to 1, it is latched in the internal latch. When a difference occurs between the output of the latch and the CDR bit, the interrupt is generated. While the CDR bit is being read, the interrupt is masked.

To set the interrupt, follow the procedure shown in figure 18.3 or 18.4.

[1] Set the CME bit. Wait a conversion time for the comparator stabilized.

[2] Read the CDR bit.

[3] Set the CMIE bit.

[4] Read the CDR bit. At this time, the CDR bit is latched in the internal latch for the comparator and the internal interrupt enable signal is asserted.

[5] As the relationship between the voltage on the COMP pin and reference voltage is changed, a difference occurs between the output level of the internal latch and the CDR bit. Then an interrupt is generated.

[6] Clear the CMF bit in the interrupt handler. When reading the CMF bit for clearing it, the CDR bit is also read since those bits are in the same register. Therefore, the output of the internal latch is updated. Go to step [5] to continue use of the interrupt.

[7] Clear the CMIE bit to clear the interrupt setting and clear the CME bit to stop the comparator. Clearing the CMIE bit negates the internal interrupt enable signal.

The interrupt flag may be set depending on the internal states of the comparator, pin states, the timing of setting the internal interrupt enable signal shown in step [4], and the timing of the CDR bit latched. To avoid this, execute steps [2] to [4] continuously or ensure that the CMF bit is cleared using the I bit in CCR as shown in figure 18.4.

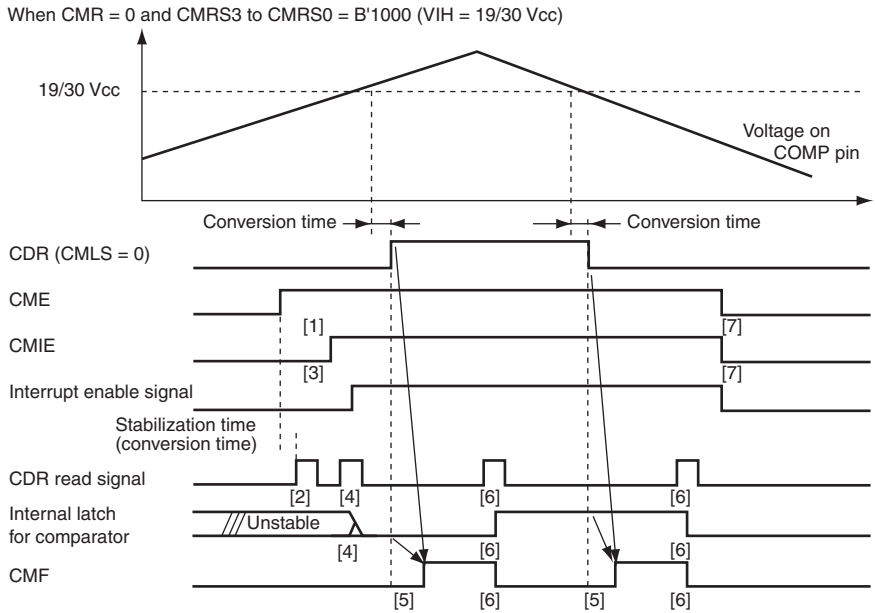


Figure 18.3 Procedure for Setting Interrupt (1)

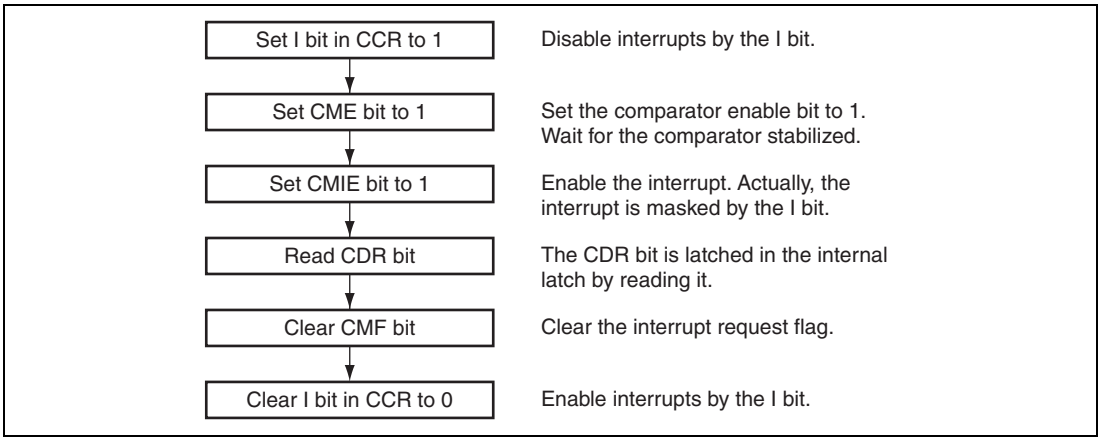


Figure 18.4 Procedure for Setting Interrupt (2)

18.5 Usage Notes

1. The COMP pin whose channel is operating as a comparator becomes a comparator analog input pin. It cannot be used for any other function.
2. When external input is used as the reference voltage (CMR0 = 1 or CMR1 = 1), the VCref pin cannot be used for any other function.
3. To stop the operation of a comparator, clear the CME0 and CME1 bits in CMCR0 and CMCR1 to 0, before clearing the COMPCKSTP bit in CKSTPR2 to 0.
4. If the LSI enters the standby mode or watch mode when a comparator is operating, the internal operation of the comparator is maintained. Since the comparator operates even in standby mode or watch mode, it returns to the same mode after the specified interrupt is canceled, though the current for the comparator is consumed.
   If a comparator is not required to return to the standby mode or watch mode when an interrupt is canceled and the current consumption needs to be reduced, stop the comparator by clearing the CME0 and CME1 bits in CMCR0 and CMCR1 to 0 before shifting the mode.

Section 19 Power-On Reset Circuit

This LSI has an on-chip power-on reset circuit. A block diagram of the power-on reset circuit is shown in figure 19.1.

19.1 Feature

- Power-on reset circuit

  An internal reset signal is generated at turning the power on by externally connecting a capacitor.

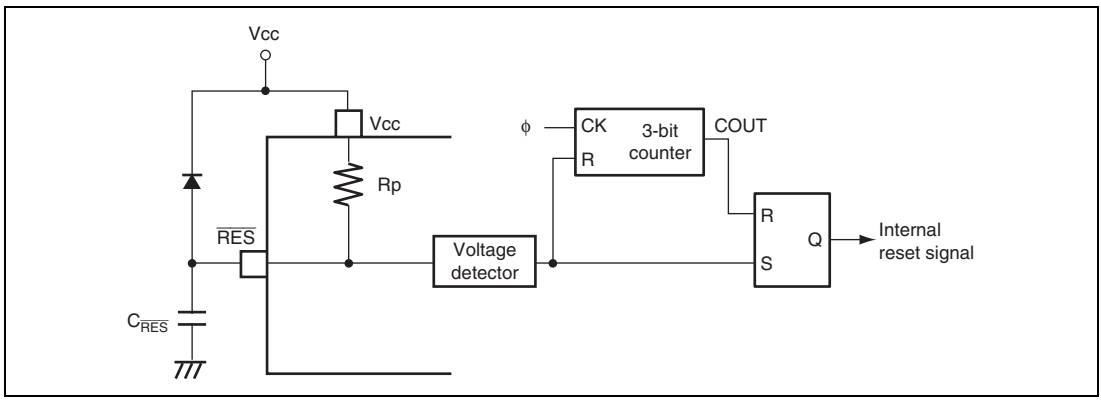


Figure 19.1 Power-On Reset Circuit

19.2 Operation

19.2.1 Power-On Reset Circuit

The operation timing of the power-on reset circuit is shown in figure 19.2. As the power supply voltage rises, the capacitor, which is externally connected to the $\overline{RES}$ pin, is gradually charged through the on-chip pull-up resistor (Rp). The low level of the $\overline{RES}$ pin is sent to the LSI and the whole LSI is reset. When the level of the $\overline{RES}$ pin reaches to the predetermined level, a voltage detection circuit detects it. Then a 3-bit counter starts counting up. When the 3-bit counter counts ϕ for 8 times, an overflow signal is generated and an internal reset signal is negated.

The capacitance ($C_{\overline{RES}}$) which is connected to the $\overline{RES}$ pin can be computed using the following formula; where the $\overline{RES}$ rising time is t. For the on-chip resistor (Rp), see section 21, Electrical Characteristics. The power supply rising time (t_vtr) should be shorter than half the $\overline{RES}$ rising time (t). The $\overline{RES}$ rising time (t) is also should be longer than the oscillation stabilization time (trc).

$$C_{\overline{RES}} = \frac{t}{Rp} \quad (t > trc,\ t > t\_vtr \times 2)$$

Note that the power supply voltage (Vcc) must fall below Vpor = 100 mV and rise after charge on the $\overline{RES}$ pin is removed. To remove charge on the $\overline{RES}$ pin, it is recommended that the diode should be placed near Vcc. If the power supply voltage (Vcc) rises from the point above Vpor, a power-on reset may not occur.

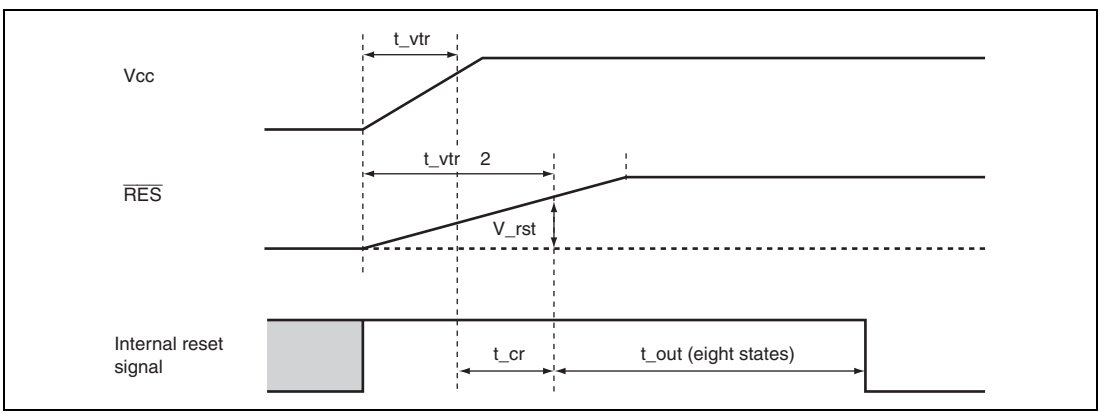


Figure 19.2 Power-On Reset Circuit Operation Timing

Section 20 List of Registers

The register list gives information on the on-chip I/O register addresses, how the register bits are configured, and the register states in each operating mode. The information is given as shown below.

1. Register addresses (address order)

- Registers are listed from the lower allocation addresses.
- Registers are classified by functional modules.
- The data bus width is indicated.
- The number of access states is indicated.

2. Register bits

- Bit configurations of the registers are described in the same order as the register addresses.
- Reserved bits are indicated by — in the bit name column.
- When the bit number is in the bit name column, it indicates that the entire register is allocated to a counter or data.
- When registers consist of 16 bits, bits are described from the MSB side.

3. Register states in each operating mode

- Register states are described in the same order as the register addresses.
- The register states described here are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, refer to the section on that on-chip peripheral module.

20.1 Register Addresses (Address Order)

The data bus width indicates the number of bits by which the register is accessed.

The number of access states indicates the number of states based on the specified reference clock.

Register Name	Abbre-viation	Address	Module Name	Data Bus Width	Access State
Flash memory control register 1	FLMCR1	H'F020	ROM	8	2
Flash memory control register 2	FLMCR2	H'F021	ROM	8	2
Flash memory power control register	FLPWCR	H'F022	ROM	8	2
Erase block register 1	EBR1	H'F023	ROM	8	2
Flash memory enable register	FENR	H'F02B	ROM	8	2
RTC interrupt flag register	RTCFLG	H'F067	RTC	8	2
Second data register/free running counter data register	RSECDR	H'F068	RTC	8	2
Minute data register	RMINDR	H'F069	RTC	8	2
Hour data register	RHRDR	H'F06A	RTC	8	2
Day-of-week data register	RWKDR	H'F06B	RTC	8	2
RTC control register 1	RTCCR1	H'F06C	RTC	8	2
RTC control register 2	RTCCR2	H'F06D	RTC	8	2
Clock source select register	RTCCSR	H'F06F	RTC	8	2
I^2C bus control register 1	ICCR1	H'F078	IIC2	8	2
I^2C bus control register 2	ICCR2	H'F079	IIC2	8	2
I^2C bus mode register	ICMR	H'F07A	IIC2	8	2
I^2C bus interrupt enable register	ICIER	H'F07B	IIC2	8	2
I^2C bus status register	ICSR	H'F07C	IIC2	8	2
Slave address register	SAR	H'F07D	IIC2	8	2
I^2C bus transmit data register	ICDRT	H'F07E	IIC2	8	2
I^2C bus receive data register	ICDRR	H'F07F	IIC2	8	2

Register Name	Abbre-viation	Address	Module Name	Data Bus Width	Access State
Port function control register	PFCR	H'F085	System	8	2
Port pull-up control register 8	PUCR8	H'F086	I/O ports	8	2
Port pull-up control register 9	PUCR9	H'F087	I/O ports	8	2
Port open-drain control register 9	PODR9	H'F08C	I/O ports	8	2
Timer mode register B1	TMB1	H'F0D0	Timer B1	8	2
Timer counter B1/ Timer load register B1	TCB1 (R)/ TLB1 (W)	H'F0D1	Timer B1	8	2
Compare control register 0	CMCR0	H'F0DC	Comparator	8	2
Compare control register 1	CMCR1	H'F0DD	Comparator	8	2
Compare data register	CMDR	H'F0DE	Comparator	8	2
SS control register H	SSCRH	H'F0E0	SSU*[1]	8	3
SS control register L	SSCRL	H'F0E1	SSU*[1]	8	3
SS mode register	SSMR	H'F0E2	SSU*[1]	8	3
SS enable register	SSER	H'F0E3	SSU*[1]	8	3
SS status register	SSSR	H'F0E4	SSU*[1]	8	3
SS receive data register	SSRDR	H'F0E9	SSU*[1]	8	3
SS transmit data register	SSTDR	H'F0EB	SSU*[1]	8	3
Timer mode register W	TMRW	H'F0F0	Timer W	8	2
Timer control register W	TCRW	H'F0F1	Timer W	8	2
Timer interrupt enable register W	TIERW	H'F0F2	Timer W	8	2
Timer status register W	TSRW	H'F0F3	Timer W	8	2
Timer I/O control register 0	TIOR0	H'F0F4	Timer W	8	2
Timer I/O control register 1	TIOR1	H'F0F5	Timer W	8	2
Timer counter	TCNT	H'F0F6	Timer W	16	2
General register A	GRA	H'F0F8	Timer W	16	2
General register B	GRB	H'F0FA	Timer W	16	2
General register C	GRC	H'F0FC	Timer W	16	2
General register D	GRD	H'F0FE	Timer W	16	2

Register Name	Abbre-viation	Address	Module Name	Data Bus Width	Access State
Event counter PWM compare register	ECPWCR	H'FF8C	AEC*[2]	16	2
Event counter PWM data register	ECPWDR	H'FF8E	AEC*[2]	16	2
Serial port control register	SPCR	H'FF91	SCI3	8	2
Input pin edge select register	AEGSR	H'FF92	AEC*[2]	8	2
Event counter control register	ECCR	H'FF94	AEC*[2]	8	2
Event counter control/status register	ECCSR	H'FF95	AEC*[2]	8	2
Event counter H	ECH	H'FF96	AEC*[2]	8	2
Event counter L	ECL	H'FF97	AEC*[2]	8	2
Serial mode register 3	SMR3	H'FF98	SCI3	8	3
Bit rate register 3	BRR3	H'FF99	SCI3	8	3
Serial control register 3	SCR3	H'FF9A	SCI3	8	3
Transmit data register 3	TDR3	H'FF9B	SCI3	8	3
Serial status register 3	SSR3	H'FF9C	SCI3	8	3
Receive data register 3	RDR3	H'FF9D	SCI3	8	3
Serial extended mode register	SEMR	H'FFA6	SCI3	8	3
IrDA control register	IrCR	H'FFA7	IrDA	8	2
Timer mode register WD	TMWD	H'FFB0	WDT*[3]	8	2
Timer control/status register WD1	TCSRWD1	H'FFB1	WDT*[3]	8	2
Timer control/status register WD2	TCSRWD2	H'FFB2	WDT*[3]	8	2
Timer counter WD	TCWD	H'FFB3	WDT*[3]	8	2
A/D result register	ADRR	H'FFBC	A/D converter	16	2
A/D mode register	AMR	H'FFBE	A/D converter	8	2
A/D start register	ADSR	H'FFBF	A/D converter	8	2

Register Name	Abbre-viation	Address	Module Name	Data Bus Width	Access State
Port mode register 1	PMR1	H'FFC0	I/O ports	8	2
Port mode register 3	PMR3	H'FFC2	I/O ports	8	2
Port mode register B	PMRB	H'FFCA	I/O ports	8	2
Port data register 1	PDR1	H'FFD4	I/O ports	8	2
Port data register 3	PDR3	H'FFD6	I/O ports	8	2
Port data register 8	PDR8	H'FFDB	I/O ports	8	2
Port data register 9	PDR9	H'FFDC	I/O ports	8	2
Port data register B	PDRB	H'FFDE	I/O ports	8	2
Port pull-up control register 1	PUCR1	H'FFE0	I/O ports	8	2
Port pull-up control register 3	PUCR3	H'FFE1	I/O ports	8	2
Port control register 1	PCR1	H'FFE4	I/O ports	8	2
Port control register 3	PCR3	H'FFE6	I/O ports	8	2
Port control register 8	PCR8	H'FFEB	I/O ports	8	2
Port control register 9	PCR9	H'FFEC	I/O ports	8	2
System control register 1	SYSCR1	H'FFF0	System	8	2
System control register 2	SYSCR2	H'FFF1	System	8	2
Interrupt edge select register	IEGR	H'FFF2	Interrupts	8	2
Interrupt enable register 1	IENR1	H'FFF3	Interrupts	8	2
Interrupt enable register 2	IENR2	H'FFF4	Interrupts	8	2
Oscillator control register	OSCCR	H'FFF5	System	8	2
Interrupt flag register 1	IRR1	H'FFF6	Interrupts	8	2
Interrupt flag register 2	IRR2	H'FFF7	Interrupts	8	2
Clock stop register 1	CKSTPR1	H'FFFA	System	8	2
Clock stop register 2	CKSTPR2	H'FFFB	System	8	2

Notes: 1. SSU: Synchronous serial communication unit
2. AEC: Asynchronous event counter
3. WDT: Watchdog timer

20.2 Register Bits

Register bit names of the on-chip peripheral modules are described below.

The 16-bit register is indicated in two rows, 8 bits for each row.

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
FLMCR1	—	SWE	ESU	PSU	EV	PV	E	P	ROM
FLMCR2	FLER	—	—	—	—	—	—	—	
FLPWCR	PDWND	—	—	—	—	—	—	—	
EBR1	—	—	—	EB4	EB3	EB2	EB1	EB0	
FENR	FLSHE	—	—	—	—	—	—	—	
RTCFLG	FOIFG	WKIFG	DYIFG	HRIFG	MNIFG	1SEIFG	05SEIFG	025SEIFG	RTC
RSECDR	BSY	SC12	SC11	SC10	SC03	SC02	SC01	SC00	
RMINDR	BSY	MN12	MN11	MN10	MN03	MN02	MN01	MN00	
RHRDR	BSY	—	HR11	HR10	HR03	HR02	HR01	HR00	
RWKDR	BSY	—	—	—	—	WK2	WK1	WK0	
RTCCR1	RUN	12/24	PM	RST	INT	—	—	—	
RTCCR2	FOIE	WKIE	DYIE	HRIE	MNIE	1SEIE	05SEIE	025SEIE	
RTCCSR	—	RCS6	RCS5	SUB32K	RCS3	RCS2	RCS1	RCS0	
ICCR1	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC2
ICCR2	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—	
ICMR	MLS	WAIT	—	—	BCWP	BC2	BC1	BC0	
ICIER	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ	
SAR	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ICDRT	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	
PFCR	—	—	—	SSUS	IRQ1S1	IRQ1S0	IRQ0S1	IRQ0S0	System
PUCR8	—	—	—	PUCR84	PUCR83	PUCR82	—	—	I/O ports
PUCR9	—	—	—	—	PUCR93	PUCR92	PUCR91	PUCR90	
PODR9	—	—	—	—	P93ODR	P92ODR	P91ODR	P90ODR	
TMB1	TMB17	TMB16	—	—	—	TMB12	TMB11	TMB10	Timer B1
TCB1 (R)/ TLB1 (W)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
CMCR0	CME0	CMIE0	CMR0	CMLS0	CRS03	CRS02	CRS01	CRS00	Comparator
CMCR1	CME1	CMIE1	CMR1	CMLS1	CRS13	CRS12	CRS11	CRS10	
CMDR	—	—	CMF1	CMF0	—	—	CDR1	CDR0	
SSCRH	MSS	BIDE	SOOS	SOL	SOLP	SCKS	CSS1	CSS0	SSU*[1]
SSCRL	—	SSUMS	SRES	SCKOS	CSOS	—	—	—	
SSMR	MLS	CPOS	CPHS	—	—	CKS2	CKS1	CKS0	
SSER	TE	RE	RSSTP	—	TEIE	TIE	RIE	CEIE	
SSSR	—	ORER	—	—	TEND	TDRE	RDRF	CE	
SSRDR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSTDR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
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	
ECPWCR	ECPWCR15	ECPWCR14	ECPWCR13	ECPWCR12	ECPWCR11	ECPWCR10	ECPWCR9	ECPWCR8	AEC*[2]
	ECPWCR7	ECPWCR6	ECPWCR5	ECPWCR4	ECPWCR3	ECPWCR2	ECPWCR1	ECPWCR0	
ECPWDR	ECPWDR15	ECPWDR14	ECPWDR13	ECPWDR12	ECPWDR11	ECPWDR10	ECPWDR9	ECPWDR8	
	ECPWDR7	ECPWDR6	ECPWDR5	ECPWDR4	ECPWDR3	ECPWDR2	ECPWDR1	ECPWDR0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
SPCR	—	—	—	SPC3	—	—	SCINV1	SCINV0	SCI3
AEGSR	AHEGS1	AHEGS0	ALEGS1	ALEGS0	AIEGS1	AIEGS0	ECPWME	—	AEC*2
ECCR	ACKH1	ACKH0	ACKL1	ACKL0	PWCK2	PWCK1	PWCK0	—	
ECCSR	OVH	OVL	—	CH2	CUEH	CUEL	CRCH	CRCL	
ECH	ECH7	ECH6	ECH5	ECH4	ECH3	ECH2	ECH1	ECH0	
ECL	ECL7	ECL6	ECL5	ECL4	ECL3	ECL2	ECL1	ECL0	
SMR3	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0	SCI3
BRR3	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0	
SCR3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR3	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	
SSR3	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT	
RDR3	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0	
SEMR	—	—	—	—	ABCS	—	—	—	
IrCR	IrE	IrCKS2	IrCKS1	IrCKS0	—	—	—	—	IrDA
TMWD	—	—	—	—	CKS3	CKS2	CKS1	CKS0	WDT*3
TCSRWD1	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI	WRST	
TCSRWD2	OVF	B5WI	WT/$\overline{IT}$	B3WI	IEOVF	—	—	—	
TCWD	TCW7	TCW6	TCW5	TCW4	TCW3	TCW2	TCW1	TCW0	
ADRR	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	A/D converter
	ADR1	ADR0	—	—	—	—	—	—	
AMR	—	TRGE	CKS1	CKS0	CH3	CH2	CH1	CH0	
ADSR	ADSF	LADS	—	—	—	—	—	—	
PMR1	—	—	IRQAEC	FTCI	AEVL	CLKOUT	TMOW	AEVH	I/O ports
PMR3	—	—	—	—	—	—	—	VCref	
PMRB	—	—	—	—	ADTSTCHG	—	IRQ1	IRQ0	
PDR1	—	—	—	—	—	P12	P11	P10	
PDR3	—	—	—	—	—	P32	P31	P30	
PDR8	—	—	—	P84	P83	P82	—	—	
PDR9	—	—	—	—	P93	P92	P91	P90	
PDRB	—	—	PB5	PB4	PB3	PB2	PB1	PB0	
PUCR1	—	—	—	—	—	PUCR12	PUCR11	PUCR10	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
PUCR3	—	—	—	—	—	PUCR32	PUCR31	PUCR30	I/O ports
PCR1	—	—	—	—	—	PCR12	PCR11	PCR10	
PCR3	—	—	—	—	—	PCR32	PCR31	PCR30	
PCR8	—	—	—	PCR84	PCR83	PCR82	—	—	
PCR9	—	—	—	—	PCR93	PCR92	PCR91	PCR90	
SYSCR1	SSBY	STS2	STS1	STS0	LSON	TMA3	MA1	MA0	System
SYSCR2	—	—	—	NESEL	DTON	MSON	SA1	SA0	
IEGR	NMIEG	—	ADTRGNEG	—	—	—	IEG1	IEG0	Interrupts
IENR1	IENRTC	—	—	—	—	IENEC2	IEN1	IEN0	
IENR2	—	IENAD	—	—	—	IENTB1	—	IENEC	
OSCCR	SUBSTP	RFCUT	SUBSEL	—	—	—	OSCF	—	System
IRR1	—	—	—	—	—	IRREC2	IRRI1	IRRI0	Interrupts
IRR2	—	IRRAD	—	—	—	IRRTB1	—	IRREC	
CKSTPR1	—	S3CKSTP	—	ADCKSTP	—	TB1CKSTP	FROMCKSTP	RTCCKSTP	System
CKSTPR2	—	TWCKSTP	IICCKSTP	SSUCKSTP	AECCKSTP	WDCKSTP	COMPCKSTP	—	

Notes: 1. SSU: Synchronous serial communication unit
2. AEC: Asynchronous event counter
3. WDT: Watchdog timer

20.3 Register States in Each Operating Mode

Register Abbreviation	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module
FLMCR1	Initialized	—	—	—	—	—	Initialized	ROM
FLMCR2	Initialized	—	—	—	—	—	—	
FLPWCR	Initialized	—	—	—	—	—	—	
EBR1	Initialized	—	—	—	—	—	Initialized	
FENR	Initialized	—	—	—	—	—	—	
RTCFLG	—	—	—	—	—	—	—	RTC
RSECDR	—	—	—	—	—	—	—	
RMINDR	—	—	—	—	—	—	—	
RHRDR	—	—	—	—	—	—	—	
RWKDR	—	—	—	—	—	—	—	
RTCCR1	—	—	—	—	—	—	—	
RTCCR2	—	—	—	—	—	—	—	
RTCCSR	Initialized	—	—	—	—	—	—	
ICCR1	Initialized	—	—	—	—	—	—	IIC2
ICCR2	Initialized	—	—	—	—	—	—	
ICMR	Initialized	—	—	—	—	—	—	
ICIER	Initialized	—	—	—	—	—	—	
ICSR	Initialized	—	—	—	—	—	—	
SAR	Initialized	—	—	—	—	—	—	
ICDRT	Initialized	—	—	—	—	—	—	
ICDRR	Initialized	—	—	—	—	—	—	
PFCR	Initialized	—	—	—	—	—	—	System
PUCR8	Initialized	—	—	—	—	—	—	I/O ports
PUCR9	Initialized	—	—	—	—	—	—	
PODR9	Initialized	—	—	—	—	—	—	
TMB1	Initialized	—	—	—	—	—	—	Timer B1
TCB1/TLB1	Initialized	—	—	—	—	—	—	

Register Abbreviation	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module
CMCR0	Initialized	—	—	—	—	—	—	Comparator
CMCR1	Initialized	—	—	—	—	—	—	
CMDR	Initialized	—	—	—	—	—	—	
SSCRH	Initialized	—	—	—	—	—	—	SSU*[1]
SSCRL	Initialized	—	—	—	—	—	—	
SSMR	Initialized	—	—	—	—	—	—	
SSER	Initialized	—	—	—	—	—	—	
SSSR	Initialized	—	—	—	—	—	—	
SSRDR	Initialized	—	—	—	—	—	—	
SSTDR	Initialized	—	—	—	—	—	—	
TMRW	Initialized	—	—	—	—	—	—	Timer W
TCRW	Initialized	—	—	—	—	—	—	
TIERW	Initialized	—	—	—	—	—	—	
TSRW	Initialized	—	—	—	—	—	—	
TIOR0	Initialized	—	—	—	—	—	—	
TIOR1	Initialized	—	—	—	—	—	—	
TCNT	Initialized	—	—	—	—	—	—	
GRA	Initialized	—	—	—	—	—	—	
GRB	Initialized	—	—	—	—	—	—	
GRC	Initialized	—	—	—	—	—	—	
GRD	Initialized	—	—	—	—	—	—	
ECPWCR	Initialized	—	—	—	—	—	—	AEC*[2]
ECPWDR	Initialized	—	—	—	—	—	—	
SPCR	Initialized	—	—	—	—	—	—	SCI3
AEGSR	Initialized	—	—	—	—	—	—	AEC*[2]
ECCR	Initialized	—	—	—	—	—	—	
ECCSR	Initialized	—	—	—	—	—	—	
ECH	Initialized	—	—	—	—	—	—	
ECL	Initialized	—	—	—	—	—	—	

Register Abbreviation	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module
SMR3	Initialized	—	—	Initialized	—	—	Initialized	SCI3
BRR3	Initialized	—	—	Initialized	—	—	Initialized	
SCR3	Initialized	—	—	Initialized	—	—	Initialized	
TDR3	Initialized	—	—	Initialized	—	—	Initialized	
SSR3	Initialized	—	—	Initialized	—	—	Initialized	
RDR3	Initialized	—	—	Initialized	—	—	Initialized	
SEMR	Initialized	—	—	Initialized	—	—	Initialized	
IrCR	Initialized	—	—	Initialized	—	—	Initialized	IrDA
TMWD	Initialized	—	—	—	—	—	—	WDT*[3]
TCSRWD1	Initialized	—	—	—	—	—	—	
TCSRWD2	Initialized	—	—	—	—	—	—	
TCWD	Initialized	—	—	—	—	—	—	
ADRR	—	—	—	—	—	—	—	A/D converter
AMR	Initialized	—	—	—	—	—	—	
ADSR	Initialized	—	—	—	—	—	—	
PMR1	Initialized	—	—	—	—	—	—	I/O ports
PMR3	Initialized	—	—	—	—	—	—	
PMRB	Initialized	—	—	—	—	—	—	
PDR1	Initialized	—	—	—	—	—	—	
PDR3	Initialized	—	—	—	—	—	—	
PDR8	Initialized	—	—	—	—	—	—	
PDR9	Initialized	—	—	—	—	—	—	
PDRB	Initialized	—	—	—	—	—	—	
PUCR1	Initialized	—	—	—	—	—	—	
PUCR3	Initialized	—	—	—	—	—	—	
PCR1	Initialized	—	—	—	—	—	—	
PCR3	Initialized	—	—	—	—	—	—	
PCR8	Initialized	—	—	—	—	—	—	
PCR9	Initialized	—	—	—	—	—	—	

Register Abbreviation	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module
SYSCR1	Initialized	—	—	—	—	—	—	System
SYSCR2	Initialized	—	—	—	—	—	—	
IEGR	Initialized	—	—	—	—	—	—	Interrupts
IENR1	Initialized	—	—	—	—	—	—	
IENR2	Initialized	—	—	—	—	—	—	
OSCCR	Initialized	—	—	—	—	—	—	System
IRR1	Initialized	—	—	—	—	—	—	Interrupts
IRR2	Initialized	—	—	—	—	—	—	
CKSTPR1	Initialized	—	—	—	—	—	—	System
CKSTPR2	Initialized	—	—	—	—	—	—	

Notes: — is not initialized.

1. SSU: Synchronous serial communication unit
2. AEC: Asynchronous event counter
3. WDT: Watchdog timer

Section 21 Electrical Characteristics

21.1 Absolute Maximum Ratings for F-ZTAT Version

Table 21.1 lists the absolute maximum ratings.

Table 21.1 Absolute Maximum Ratings

Item		Symbol	Value	Unit	Note
Power supply voltage		V_{CC}	–0.3 to +4.3	V	*1
Analog power supply voltage		AV_{CC}	–0.3 to +4.3	V	
Input voltage	Other than port B	V_{in}	–0.3 to V_{CC} +0.3	V	
	Port B	AV_{in}	–0.3 to AV_{CC} +0.3	V	
Operating temperature		T_{opr}	–20 to +75 (general specifications)*2	°C	
			–40 to +85 (wide temperature range specifications)*2		
Storage temperature		T_{stg}	–55 to +125	°C	

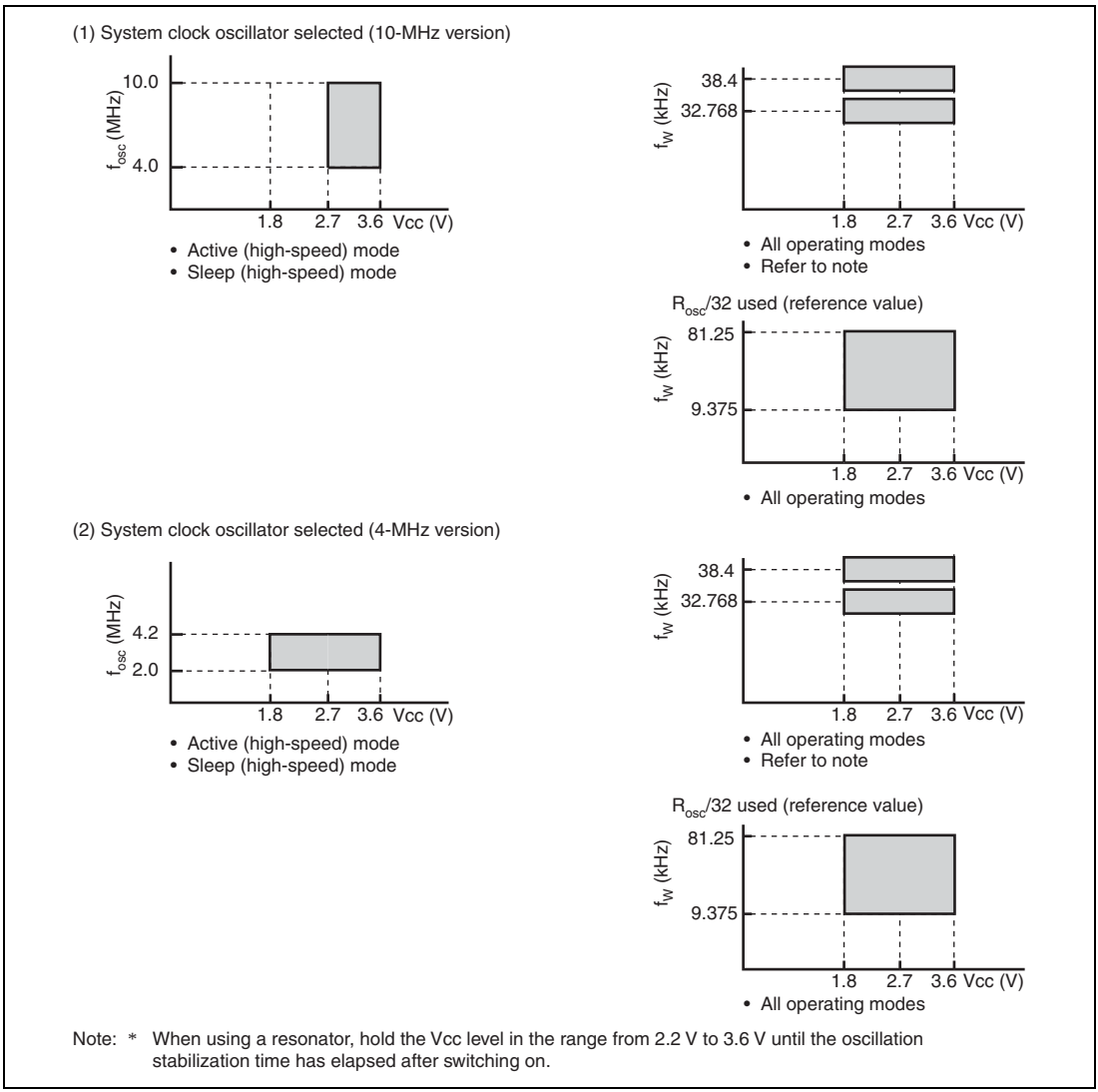
Notes: 1. Permanent damage may occur to the LSI if absolute maximum ratings are exceeded. Normal operation should be under the conditions specified in Electrical Characteristics. Exceeding these values can result in incorrect operation and reduced reliability.

2. The operating temperature range for flash memory programming/erasing is T_a = 0 to +75°C.

21.2 Electrical Characteristics for F-ZTAT Version

21.2.1 Power Supply Voltage and Operating Range

The power supply voltage and operating range are indicated by the shaded region in the figures.



Note: * When using a resonator, hold the Vcc level in the range from 2.2 V to 3.6 V until the oscillation stabilization time has elapsed after switching on.

Figure 21.1 Power Supply Voltage and Oscillation Frequency Range (1)

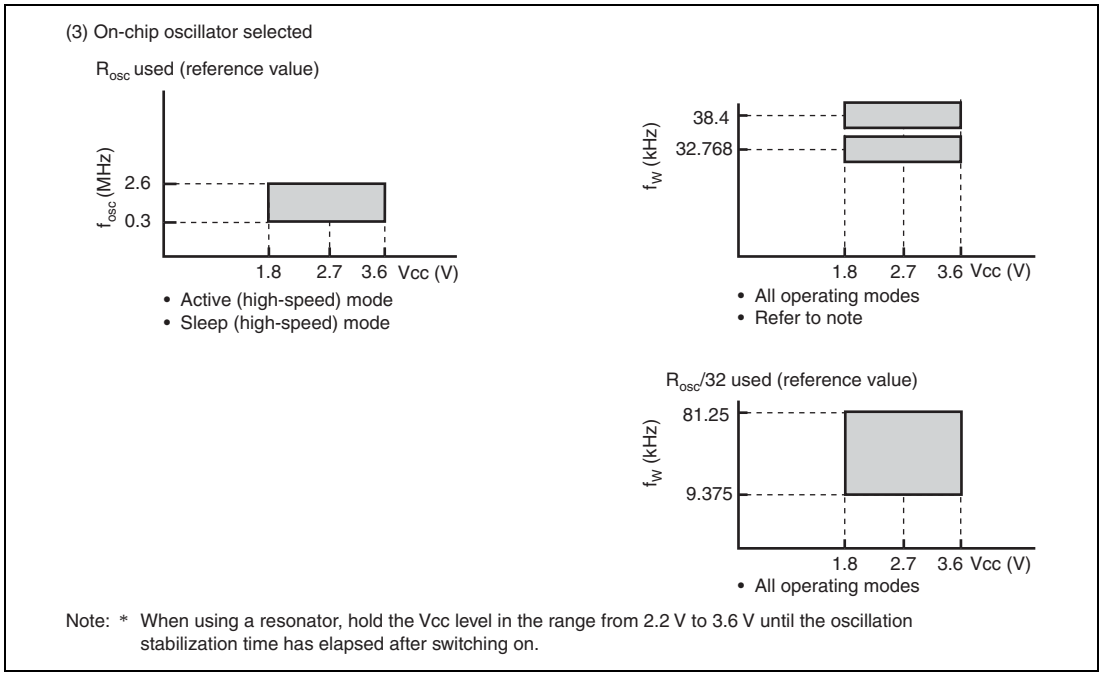


Figure 21.2 Power Supply Voltage and Oscillation Frequency Range (2)

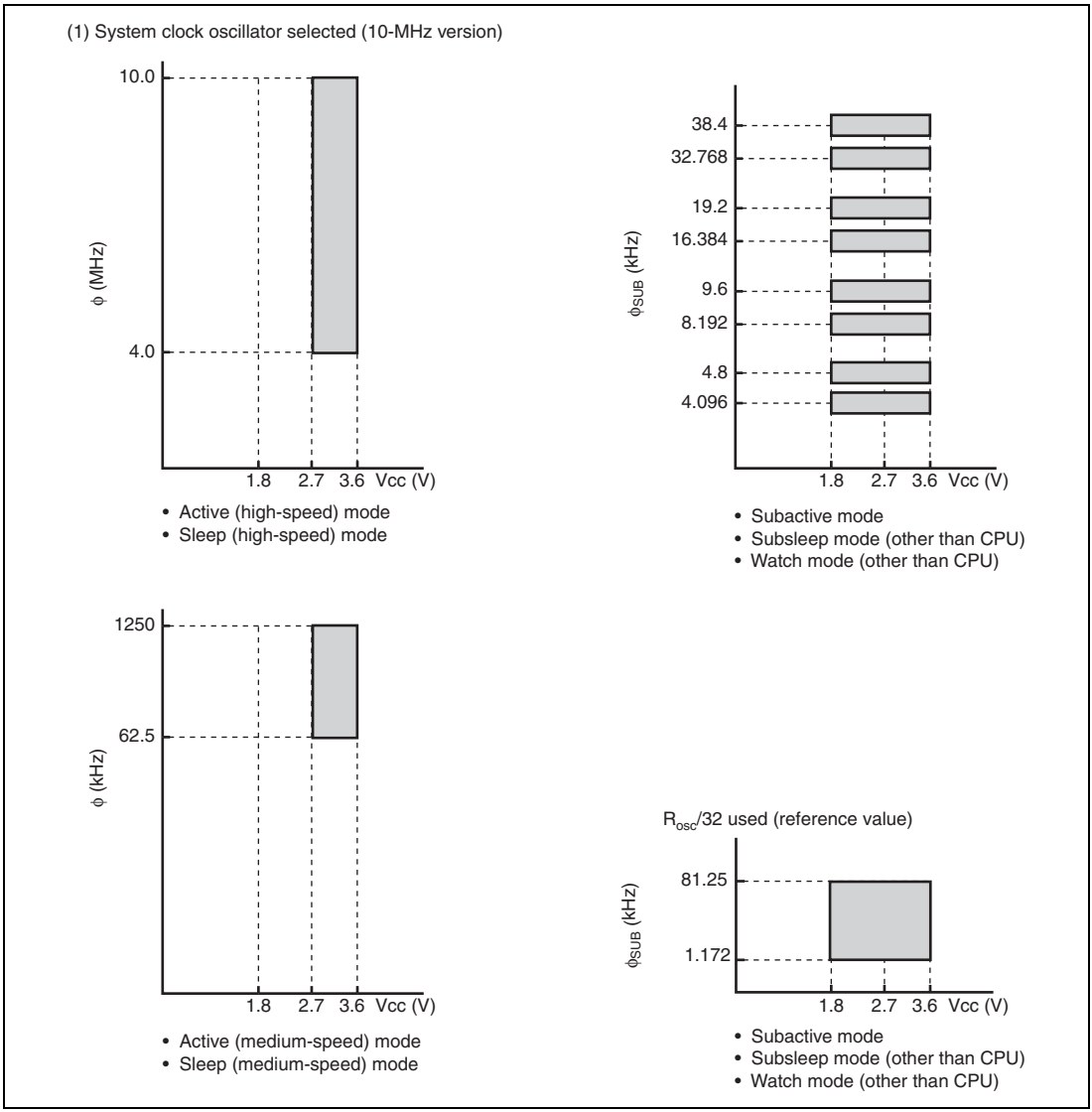


Figure 21.3 Power Supply Voltage and Operating Frequency Range (1)

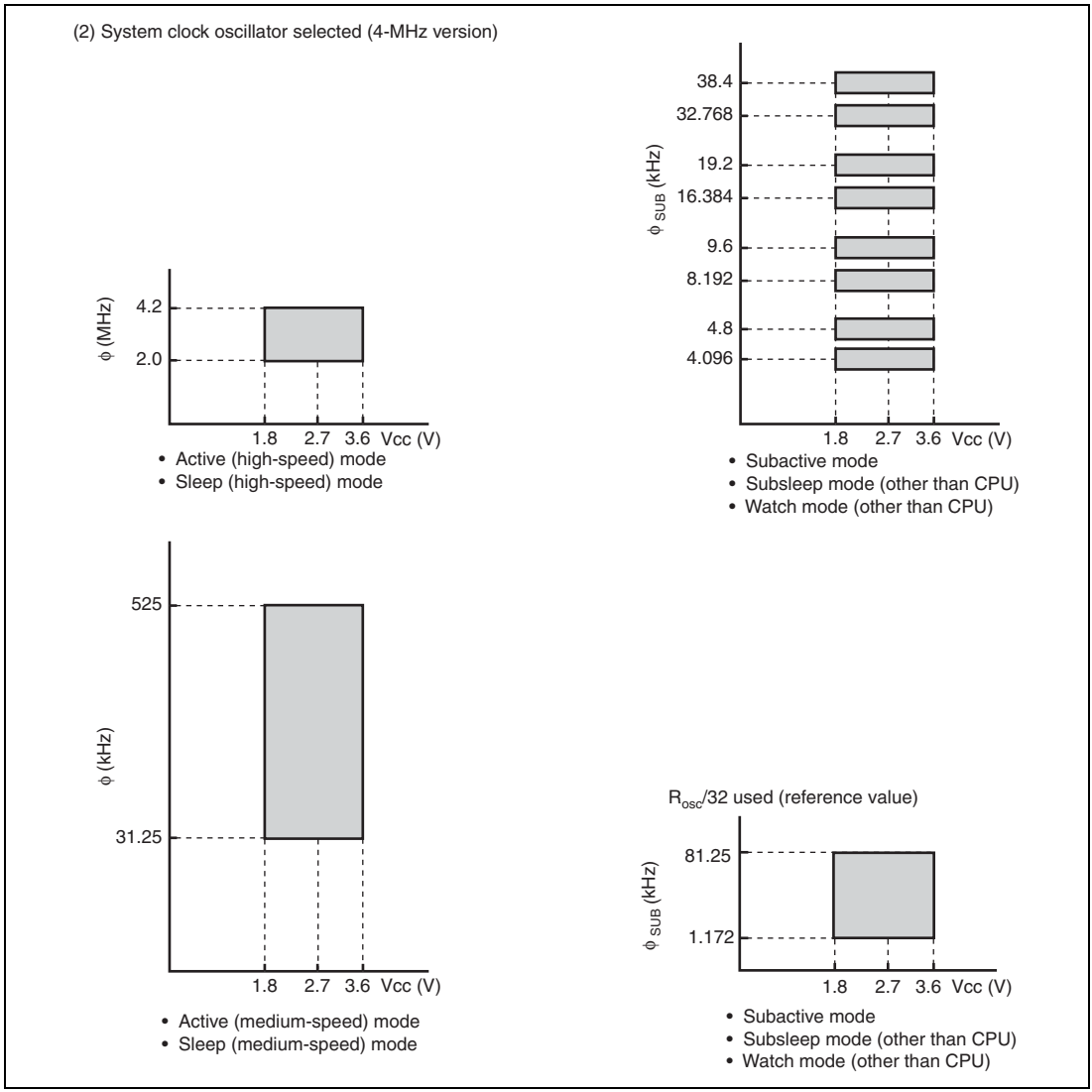


Figure 21.4 Power Supply Voltage and Operating Frequency Range (2)

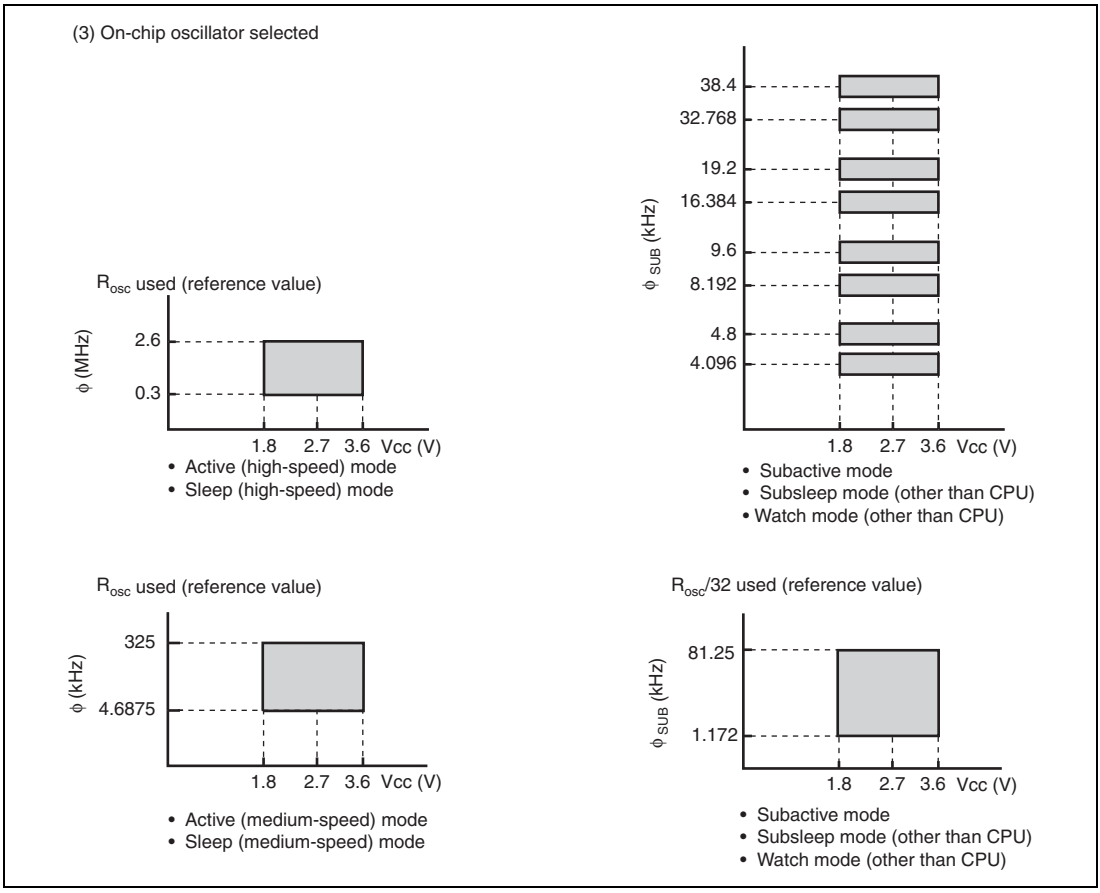


Figure 21.5 Power Supply Voltage and Operating Frequency Range (3)

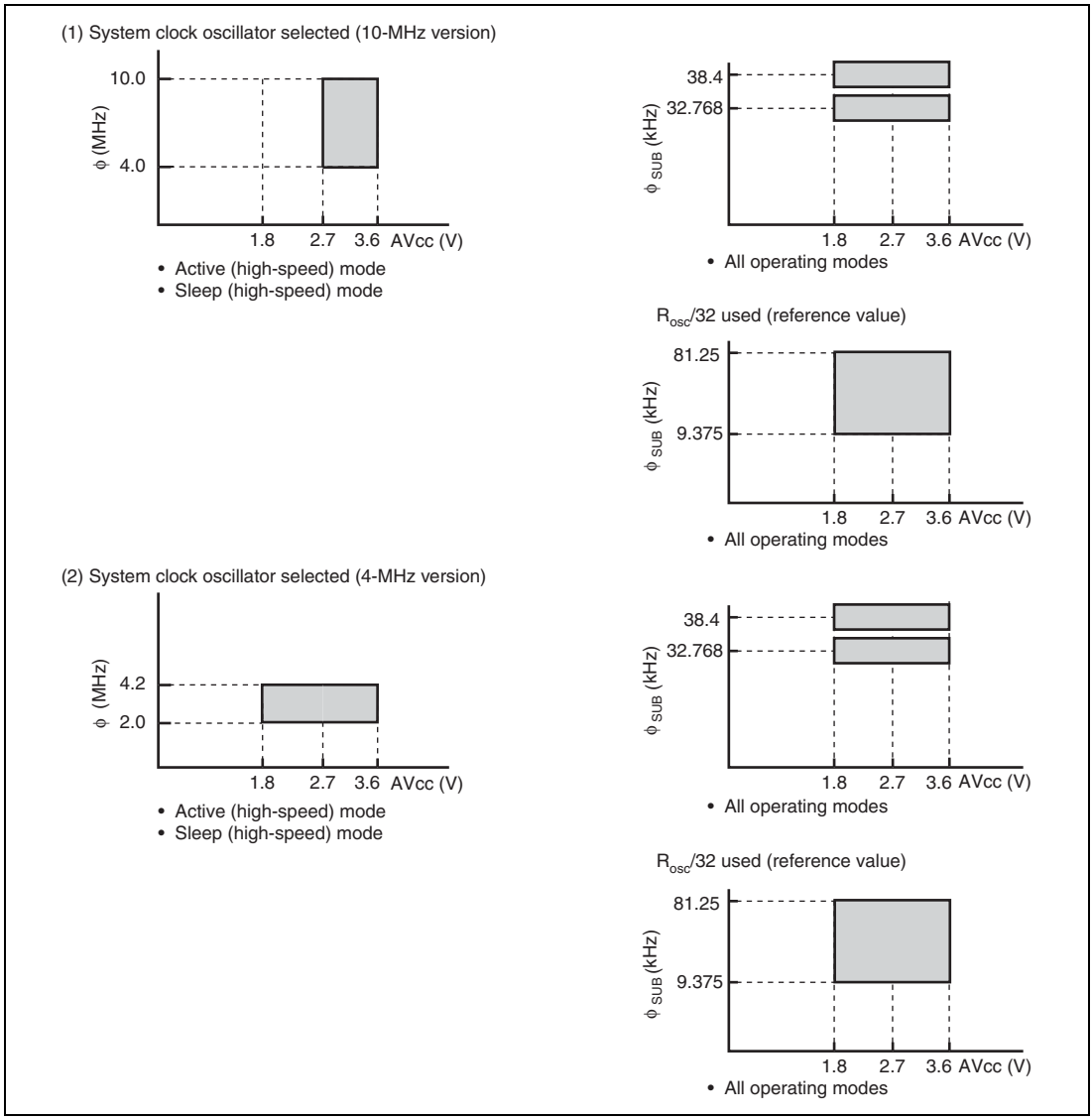


Figure 21.6 Analog Power Supply Voltage and Operating Frequency Range of A/D Converter (1)

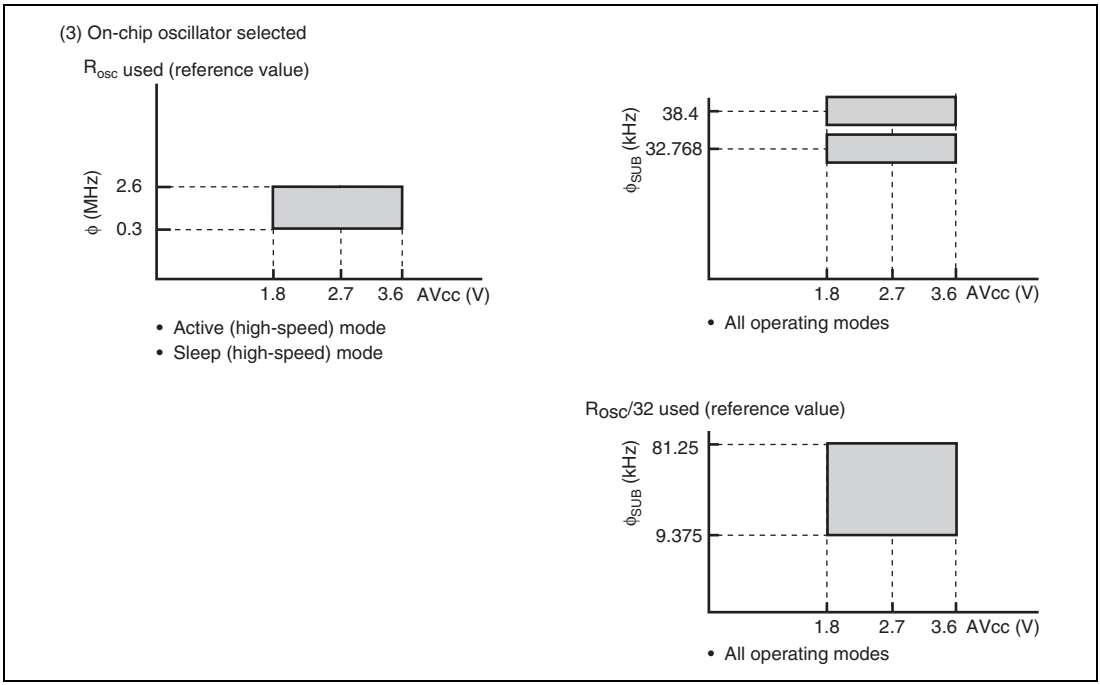


Figure 21.7 Analog Power Supply Voltage and Operating Frequency Range of A/D Converter (2)

21.2.2 DC Characteristics

Table 21.2 lists the DC characteristics.

Table 21.2 DC Characteristics

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, unless otherwise specified.

				Values				
Item	**Symbol**	**Applicable Pins**	**Test Condition**	**Min.**	**Typ.**	**Max.**	**Unit**	**Notes**
Input high voltage	V_{IH}	$\overline{RES}$, TEST, $\overline{NMI}$*[3], AEVL, AEVH, $\overline{ADTRG}$, SCK3, IRQAEC		0.9V_{CC}	—	V_{CC} + 0.3	V	
		$\overline{IRQ0}$*[4], $\overline{IRQ1}$*[4]		0.9V_{CC}	—	AV_{CC} + 0.3		
		RXD3, IrRXD		0.8V_{CC}	—	V_{CC} + 0.3		
		OSC1		0.9V_{CC}	—	V_{CC} + 0.3		
		X1		0.9V_{CC}	—	V_{CC} + 0.3		
		P10 to P12, P30 to P32, P82 to P84, P90 to P93, SSI, SSO, SSCK, $\overline{SCS}$, FTCI, FTIOA, FTIOB, FTIOC, FTIOD, E7_0 to E7_2, SCL, SDA		0.8V_{CC}	—	V_{CC} + 0.3		
		PB0 to PB5		0.8V_{CC}	—	AV_{CC} + 0.3		

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Notes
Input low voltage	V_{IL}	$\overline{RES}$, TEST, $\overline{NMI}$*[3], $\overline{IRQ0}$, $\overline{IRQ1}$, IRQAEC, AEVL, AEVH, $\overline{ADTRG}$, SCK3		–0.3	—	$0.1V_{CC}$	V	
		RXD3, IrRXD		–0.3	—	$0.2V_{CC}$		
		OSC1		–0.3	—	$0.1V_{CC}$		
		X1		–0.3	—	$0.1V_{CC}$		
		P10 to P12, P30 to P32, P82 to P84, P90 to P93, SCL, SDA, PB0 to PB5, SSI, SSO, SSCK, $\overline{SCS}$, FTCI, FTIOA, FTIOB, FTIOC, FTIOD, E7_0 to E7_2, SCL, SDA		–0.3	—	$0.2V_{CC}$		
Output high voltage	V_{OH}	P10 to P12, P30 to P32, P90 to P93	$-I_{OH}$ = 1.0 mA Vcc = 2.7 V to 3.6 V	V_{CC} – 1.0	—	—	V	
			$-I_{OH}$ = 0.1 mA	V_{CC} – 0.3	—	—		
		P82 to P84	$-I_{OH}$ = 1.0 mA V_{CC} = 2.7 V to 3.6 V	V_{CC} – 1.0	—	—		
			$-I_{OH}$ = 0.1 mA	V_{CC} – 0.3	—	—		
Output low voltage	V_{OL}	P10 to P12, P30 to P32, P90 to P93	I_{OL} = 0.4 mA	—	—	0.5	V	
		P82 to P84	I_{OL} = 15 mA, Vcc = 2.7 V to 3.6 V	—	—	1.0		
			I_{OL} = 10 mA, Vcc = 2.2 V to 3.6 V	—	—	0.5		
			I_{OL} = 8 mA	—	—	0.5		
		SCL, SDA	I_{OL} = 3.0 mA	—	—	0.4		

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Notes
Input/output leakage current	$\lvert I_{IL} \rvert$	TEST, $\overline{NMI}$*[3], OSC1, X1, P10 to P12, P30 to P32, P82 to P84, P90 to P93, E7_0 to E7_2	V_{IN} = 0.5 V to V_{CC} – 0.5 V	—	—	1.0	µA	
		PB0 to PB5	V_{IN} = 0.5 V to AV_{CC} – 0.5 V	—	—	1.0		
Pull-up MOS current	$-I_p$	P10 to P12, P30 to P32, P82 to P84, P90 to P93	V_{CC} = 3 V, V_{IN} = 0 V	30	—	180	µA	
Input capacitance	C_{IN}	All input pins except power supply pin	f = 1 MHz, V_{IN} =0 V, Ta = 25°C	—	—	15.0	pF	
Active mode supply current	I_{OPE1}	V_{CC}	Active (high-speed) mode, V_{CC} = 1.8 V, f_{OSC} = 2 MHz	—	1.1	—	mA	Max. guideline = 1.1 × typ.*[1]*[2]
			Active (high-speed) mode, V_{CC} = 3 V, f_{OSC} = R_{OSC}	—	1.2	—		Max. guideline = 1.1 × typ.*[1]*[2] Reference value
			Active (high-speed) mode, V_{CC} = 3 V, f_{OSC} = 4.2 MHz	—	2.6	4.0		*[1]*[2] 4-MHz version
			Active (high-speed) mode, V_{CC} = 3 V, f_{OSC} = 10 MHz	—	6.0	10.0		*[1]*[2] 10-MHz version

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Notes
Active mode supply current	I_{OPE2}	V_{CC}	Active (medium-speed) mode, V_{CC} = 1.8 V, f_{OSC} = 2 MHz, ϕ_{osc}/64	—	0.4	—	mA	Max. guideline = 1.1 × typ.*1*2
			Active (medium-speed) mode, V_{CC} = 3 V, f_{OSC} = 4.2 MHz, ϕ_{osc}/64	—	0.7	1.1		*1*2 4-MHz version
			Active (medium-speed) mode, V_{CC} = 3 V, f_{OSC} = 10 MHz, ϕ_{osc}/64	—	0.8	1.3		*1*2 10-MHz version
Sleep mode supply current	I_{SLEEP}	V_{CC}	V_{CC} = 1.8 V, f_{OSC} = 2 MHz	—	0.9	—	mA	Max. guideline = 1.1 × typ.*1*2
			V_{CC} = 3 V, f_{OSC} = 4.2 MHz	—	2.0	3.2		*1*2 4-MHz version
			V_{CC} = 3 V, f_{OSC} = 10 MHz	—	4.2	6.4		*1*2 10-MHz version
Subactive mode supply current	I_{SUB}	V_{CC}	V_{CC} = 2.7 V, 32-kHz crystal resonator (ϕ_{SUB} = ϕ_W/8)	—	7.0	—	µA	*1*2 Reference value
			V_{CC} = 2.7 V, 32-kHz crystal resonator (ϕ_{SUB} = ϕ_W/2)	—	25	—		*1*2 Reference value
			V_{CC} = 2.7 V, on-chip oscillator/32 (ϕ_{SUB} = ϕ_W = R_{OSC}/32)	—	80	—		*1*2 Reference value
			V_{CC} = 2.7 V, 32-kHz crystal resonator (ϕ_{SUB} = ϕ_W)	—	45	75		*1*2

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Notes
Subsleep mode supply current	I_{SUBSP}	V_{CC}	V_{CC} = 2.7 V, 32-kHz crystal resonator (ϕ_{SUB} = ϕ_W/2)	—	3.5	—	µA	*1*2 Reference value
			V_{CC} = 2.7 V, on-chip oscillator/32 (ϕ_{SUB} = ϕ_W = R_{OSC}/32)	—	34	—		*1*2 Reference value
			V_{CC} = 2.7 V, 32-kHz crystal resonator (ϕ_{SUB} = ϕ_W)	—	5.1	16.0		*1*2
Watch mode supply current	I_{WATCH}	V_{CC}	V_{CC} = 1.8 V, Ta = 25°C, 32-kHz crystal resonator	—	0.5	—	µA	*1*2 Reference value
			V_{CC} = 2.7 V, 32-kHz crystal resonator	—	1.5	5.0		*1*2
Standby mode supply current	I_{STBY}	V_{CC}	V_{CC} = 3.0 V, Ta = 25°C, 32-kHz crystal resonator not used	—	0.1	—	µA	*1*2 Reference value
			32-kHz crystal resonator not used	—	1.0	5.0		*1*2
RAM data retaining voltage	V_{RAM}	V_{CC}		1.5	—	—	V	
Permissible output low current (per pin)	I_{OL}	Output pins except port 8		—	—	0.5	mA	
		Port 8		—	—	15.0		
Permissible output low current (total)	ΣI_{OL}	Output pins except port 8		—	—	20.0	mA	
		Port 8		—	—	45.0		

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Notes
Permissible output high current (per pin)	$-I_{OH}$	All output pins	V_{CC} = 2.7 V to 3.6 V	—	—	2.0	mA	
			Other than above	—	—	0.2		
Permissible output high current (total)	$\Sigma - I_{OH}$	All output pins		—	—	10.0	mA	

Notes: 1. Pin states during current measurement.

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pins
Active (high-speed) mode (I_{OPE1})	V_{CC}	Only CPU operates	V_{CC}	System clock oscillator: Crystal resonator Subclock oscillator: Pin X1 = GND
Active (medium-speed) mode (I_{OPE2})				
Sleep mode	V_{CC}	Only on-chip timers operate	V_{CC}	
Subactive mode	V_{CC}	Only CPU operates	V_{CC}	System clock oscillator: Crystal resonator Subclock oscillator: Crystal resonator
Subsleep mode	V_{CC}	Only on-chip timers operate, CPU stops	V_{CC}	
Watch mode	V_{CC}	Only timer base operates, CPU stops	V_{CC}	
Standby mode	V_{CC}	CPU and timers both stop, SUBSTP = 1	V_{CC}	System clock oscillator: Crystal resonator Subclock oscillator: Pin X1 = Crystal resonator

2. Excludes current in pull-up MOS transistors and output buffers.
3. Used for the determination of user mode or boot mode when the reset is released.
4. When bits IRQ0S1 and IRQ0S0 are set to B'01 or B'10, and bits IRQ1S1 and IRQ1S0 are set to B'01 or B'10, the maximum value is given V_{CC} + 0.3 (V).

21.2.3 AC Characteristics

Table 21.3 lists the control signal timing, table 21.4 lists the serial interface timing, table 21.5 lists the synchronous serial communication unit timing, and table 21.6 lists the I^2C bus interface timing.

Table 21.3 Control Signal Timing

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
System clock oscillation frequency	f_{OSC}	OSC1, OSC2	V_{CC} = 2.7 V to 3.6 V (10-MHZ version)	4.0	—	10.0	MHz	
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	2.0	—	4.2		
OSC clock (ϕ_{OSC}) cycle time	t_{OSC}	OSC1, OSC2	V_{CC} = 2.7 V to 3.6 V (10-MHz version)	100	—	250	ns	Figure 21.15
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	238	—	500		
System clock (ϕ) cycle time	t_{cyc}			1	—	64	t_{OSC}	
			V_{CC} = 2.7 V to 3.6 V (10-MHz version)	—	—	16	µs	
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	—	—	32		
On-chip oscillator oscillation frequency	t_{ROSC}			0.3	—	2.6	MHz	Reference value
On-chip oscillator clock cycle time	t_{ROSC}			0.38	—	3.3	µs	Reference value
Subclock oscillator oscillation frequency	f_W	X1, X2		—	32.768 or 38.4	—	kHz	
Watch clock (ϕ_W) cycle time	t_W	X1, X2		—	30.5 or 26.0	—	µs	Figure 21.15
Subclock (ϕ_{SUB}) cycle time	t_{subcyc}			1	—	8	t_W	*[1]
Instruction cycle time				2	—	—	t_{cyc} t_{subcyc}	

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Reference Figure
Oscillation stabilization time	t_{rc}	OSC1, OSC2	Ceramic resonator (V_{cc} = 2.2 V to 3.6 V)	—	20	45	µs	Figure 21.28
			Ceramic resonator (Other than above)	—	80	—		
			Crystal resonator (V_{cc} = 2.7 V to 3.6 V)	—	300	800		
			Crystal resonator (V_{cc} = 2.2 V to 3.6 V)	—	600	1000		
			Other than above	—	—	50	ms	
		On-chip oscillator	At switching on	—	15	25	µs	
		X1, X2	V_{cc} = 2.2 V to 3.6 V	—	—	2	s	Figures 4.6 and 4.7
			Other than above	—	4	—		
External clock high width	t_{CPH}	OSC1	V_{cc} = 2.7 V to 3.6 V (10-MHz version)	40	—	—	ns	Figure 21.15
			V_{cc} = 1.8 V to 3.6 V (4-MHz version)	95	—	—		
		X1		—	15.26 or 13.02	—	µs	
External clock low width	t_{CPL}	OSC1	V_{cc} = 2.7 V to 3.6 V (10-MHz version)	40	—	—	ns	Figure 21.15
			V_{cc} = 1.8 V to 3.6 V (4-MHz version)	95	—	—		
		X1		—	15.26 or 13.02	—	µs	
External clock rising time	t_{CPr}	OSC1	V_{cc} = 2.7 V to 3.6 V (10-MHz version)	—	—	10	ns	Figure 21.15
			V_{cc} = 1.8 V to 3.6 V (4-MHz version)	—	—	24		
		X1		—	—	55.0	ns	

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Reference Figure
External clock falling time	t_{CPf}	OSC1	V_{CC} = 2.7 V to 3.6 V (10-MHz version)	—	—	10	ns	Figure 21.15
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	—	—	24		
		X1		—	—	55.0	ns	
$\overline{RES}$ pin low width	t_{REL}	$\overline{RES}$	At switching on or other than below	t_{rc} + 20 × t_{cyc}	—	—	µs	Figure 21.16*[2]
			Active mode or sleep mode	20	—	—	t_{cyc}	
Input pin high width	t_{IH}	$\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{NMI}$, IRQAEC, $\overline{ADTRG}$, FTCI, FTIOA, FTIOB, FTIOC, FTIOD		2	—	—	t_{cyc} t_{subcyc}	Figure 21.17
		AEVL, AEVH	V_{CC} = 2.7 V to 3.6 V (10-MHz version)	50	—	—	ns	
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	110	—	—		
Input pin low width	t_{IL}	$\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{NMI}$, IRQAEC, $\overline{ADTRG}$, FTCI, FTIOA, FTIOB, FTIOC, FTIOD		2	—	—	t_{cyc} t_{subcyc}	Figure 21.17
		AEVL, AEVH	V_{CC} = 2.7 V to 3.6 V (10-MHz version)	50	—	—	ns	
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	110	—	—		

Notes: 1. Selected with the SA1 and SA0 bits in the system control register 2 (SYSCR2).
2. For details on the power-on reset characteristics, refer to table 21.10 and figure 21.26.

Table 21.4 Serial Interface Timing

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, unless otherwise specified.

Item		Symbol	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
Input clock cycle	Asynchronous	t_{scyc}		4	—	—	t_{cyc} or t_{subcyc}	Figure 21.18
	Clock synchronous			6	—	—		
Input clock pulse width		t_{SCKW}		0.4	—	0.6	t_{scyc}	Figure 21.18
Transmit data delay time (clock synchronous)		t_{TXD}		—	—	1	t_{cyc} or t_{subcyc}	Figure 21.19
Receive data setup time (clock synchronous)		t_{RXS}		400.0	—	—	ns	Figure 21.19
Receive data hold time (clock synchronous)		t_{RXH}		400.0	—	—	ns	Figure 21.19

Table 21.5 Synchronous Serial Communication Unit (SSU) Timing

V_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, output load = 100 pF, unless otherwise specified.

Item		Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Reference Figure
Clock cycle		t_{sucyc}	SSCK		4	—	—	t_{cyc}	Figures 21.20 to 21.24
Clock high pulse width		t_{HI}	SSCK		0.4	—	0.6	t_{sucyc}	
Clock low pulse width		t_{LO}	SSCK		0.4	—	0.6	t_{sucyc}	
Clock rising time	Master	t_{RISE}	SSCK		—	—	1	t_{cyc}	
	Slave				—	—	1.0	µs	
Clock falling time	Master	t_{FALL}	SSCK		—	—	1	t_{cyc}	
	Slave				—	—	1.0	µs	
Data input setup time		t_{SU}	SSO SSI		1	—	—	t_{cyc}	
Data input hold time		t_{H}	SSO SSI		1	—	—	t_{cyc}	
$\overline{SCS}$ setup time	Slave	t_{LEAD}	$\overline{SCS}$		$1t_{cyc}$ + 100	—	—	ns	
$\overline{SCS}$ hold time	Slave	t_{LAG}	$\overline{SCS}$		$1t_{cyc}$ + 100	—	—	ns	
Data output delay time		t_{OD}	SSO SSI		—	—	1	t_{cyc}	
Slave access time		t_{SA}	SSI		—	—	$1t_{cyc}$ + 100	ns	
Slave out release time		t_{OR}	SSI		—	—	$1t_{cyc}$ + 100	ns	

Table 21.6 I²C Bus Interface Timing

V_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, Ta = –20 to +75°C, unless otherwise specified.

Item	Symbol	Test Condition	Values			Unit	Reference Figure
			Min.	Typ.	Max.		
SCL input cycle time	t_{SCL}		$12t_{cyc}$ + 600	—	—	ns	Figure 21.25
SCL input high width	t_{SCLH}		$3t_{cyc}$ + 300	—	—	ns	
SCL input low width	t_{SCLL}		$5t_{cyc}$ + 300	—	—	ns	
Falling time for SCL and SDA inputs	t_{Sf}		—	—	300	ns	
Pulse width of spike on SCL and SDA to be suppressed	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	
Repeated start condition input setup time	t_{STAS}		$3t_{cyc}$	—	—	ns	
Stop condition input setup time	t_{STOS}		$3t_{cyc}$	—	—	ns	
Data-input setup time	t_{SDAS}		$1t_{cyc}$ + 20	—	—	ns	
Data-input hold time	t_{SDAH}		0	—	—	ns	
Capacitive load of SCL and SDA	Cb		0	—	400	pF	
Falling time of SCL and SDA output	t_{Sf}		—	—	300	ns	

21.2.4 A/D Converter Characteristics

Table 21.7 lists the A/D converter characteristics.

Table 21.7 A/D Converter Characteristics

V_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Notes
Analog power supply voltage	AV_{CC}	AV_{CC}		1.8	—	3.6	V	*1
Analog input voltage	AV_{IN}	AN0 to AN5		–0.3	—	AV_{CC} + 0.3	V	
Analog power supply current	AI_{OPE}	AV_{CC}	AV_{CC} = 3.0 V	—	—	1.0	mA	
	AI_{STOP1}	AV_{CC}		—	600	—	µA	*2 Reference value
	AI_{STOP2}	AV_{CC}		—	—	5	µA	*3
Analog input capacitance	C_{AIN}	AN0 to AN5		—	—	15.0	pF	
Permissible signal source impedance	R_{AIN}			—	—	10.0	kΩ	
Resolution (data length)				—	—	10	Bits	
Nonlinearity error			AV_{CC} = 2.7 V to 3.6 V V_{CC} = 2.7 V to 3.6 V	—	—	±3.5	LSB	Other than subclock operation
			AV_{CC} = 2.0 V to 3.6 V V_{CC} = 2.0 V to 3.6 V	—	—	±5.5		
			Subclock operating	—	—	±5.5		Subactive or subsleep mode, conversion time = 31/ϕ_w
			Other than above	—	—	±7.5		*4
Quantization error				—	—	±0.5	LSB	

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Notes
Absolute accuracy			AV_{CC} = 2.7 V to 3.6 V V_{CC} = 2.7 V to 3.6 V	—	—	±4.0	LSB	
			AV_{CC} = 2.0 V to 3.6 V V_{CC} = 2.0 V to 3.6 V	—	—	±6.0		
			Subclock operating	—	—	±6.0		Subactive or subsleep mode, conversion time = $31/\phi_W$
			Other than above	—	—	±8.0		*4
Conversion time			AV_{CC} = 2.7 V to 3.6 V V_{CC} = 2.7 V to 3.6 V	12.4	—	124	µs	System clock oscillator is selected
				31	62	124		On-chip oscillator is selected Reference value (f_{ROSC} = 1 MHz)
				—	807	—		ϕ_{SUB} = 38.4 kHz
				—	945	—		ϕ_{SUB} = 32.8 kHz
				—	992	—		$\phi_{SUB} = R_{OSC}/32$ Reference value (f_{ROSC} = 1 MHz)
			Other than AV_{CC} = 2.7 V to 3.6 V V_{CC} = 2.7 V to 3.6 V	29.5	—	124		System clock oscillator is selected
				31	62	124		On-chip oscillator is selected Reference value (f_{ROSC} = 1 MHz)
				—	807	—		ϕ_{SUB} = 38.4 kHz
				—	945	—		ϕ_{SUB} = 32.8 kHz
				—	992	—		$\phi_{SUB} = R_{OSC}/32$ Reference value (f_{ROSC} = 1 MHz)

Notes: 1. Connect AV_{CC} to V_{CC} when the A/D converter is not used.

2. AI_{STOP1} is the current flowing through the ladder resistor while the A/D converter is idle.

3. AI_{STOP2} is the current flowing at a reset, in standby mode or watch mode, through the ladder resistor while the A/D converter is idle.

4. Conversion time is 29.5 µs.

21.2.5 Comparator Characteristics

Table 21.8 shows the comparator characteristics.

Table 21.8 Comparator Characteristics

V_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, unless otherwise specified.

		Values				
Item	**Test Condition**	**Min.**	**Typ.**	**Max.**	**Unit**	**Notes**
Accuracy	1LSB = V_{CC}/30	—	1/2	—	LSB	Comparing with internal resistor network
Conversion time		—	—	15	µs	
External input reference voltage	VCref pin	0.9	—	0.9 × V_{CC}	V	
Internal resistance compare voltage		0.9	—	26/30 × V_{CC}	V	
Comparator input voltage	COMP0 and COMP1 pins	–0.3	—	AV_{CC} + 0.3	V	
Ladder resistance		—	3	—	MΩ	Reference value

21.2.6 Watchdog Timer Characteristics

Table 21.9 shows the watchdog timer characteristics.

Table 21.9 Watchdog Timer Characteristics

V_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, unless otherwise specified.

		Applicable		Values				
Item	**Symbol**	**Pins**	**Test Condition**	**Min.**	**Typ.**	**Max.**	**Unit**	**Notes**
On-chip oscillator overflow time	t_{OVF}			0.2	0.4	—	s	*

Note: * Indicates that the period from when the counter starts with 0 to when the counter reaches 255 and an internal reset occurs while the on-chip oscillator is selected.

21.2.7 Power-On Reset Circuit Characteristics

Table 21.10 lists the power-on reset circuit characteristics.

Table 21.10 Power-On Reset Circuit Characteristics

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V,
Ta = –20 to +75°C (general specifications), Ta = –40 to +85°C (wide temperature range specifications), unless otherwise specified.

			Values				
Item	**Symbol**	**Test Condition**	**Min.**	**Typ.**	**Max.**	**Unit**	**Notes**
Reset voltage	V_rst		0.7Vcc	0.8Vcc	0.9Vcc	V	
Power supply rising time	t_vtr		The Vcc rising time should be shorter than half the $\overline{RES}$ rising time.				
Reset count time	t_out		0.8	—	4.0	μs	
			3.2	—	26.7		On-chip oscillator is selected (reference value)
Count start time	t_cr		Adjustable by the value of the external capacitor connected to the $\overline{RES}$ pin.				
Pull-up resistance	R_P		60	100	—	kΩ	

21.2.8 Flash Memory Characteristics

Table 21.11 lists the flash memory characteristics.

Table 21.11 Flash Memory Characteristics

AV_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, V_{CC} = 1.8 V to 3.6 V (operating voltage range in reading),
V_{CC} = 3.0 V to 3.6 V (operating voltage range in programming/erasing),
Ta = 0 to +75°C (operating temperature range in programming/erasing)

Item		Symbol	Test Condition	Values Min.	Typ.	Max.	Unit
Programming time (per 128 bytes)*1*2*4		t_P		—	7	200	ms
Erasing time (per block)*1*3*6		t_E		—	100	1200	ms
Maximum programming count		N_{WEC}		1000 *8*11	10000 *9	—	Times
				100 *8*12	10000 *9	—	
Data retention time		t_{DRP}		10*10	—	—	Years
Programming	Wait time after setting SWE bit*1	x		1	—	—	µs
	Wait time after setting PSU bit*1	y		50	—	—	µs
	Wait time after setting P bit*1*4	z1	1 ≤ n ≤ 6	28	30	32	µs
		z2	7 ≤ n ≤ 1000	198	200	202	µs
		z3	Additional-programming	8	10	12	µs
	Wait time after clearing P bit*1	α		5	—	—	µs
	Wait time after clearing PSU bit*1	β		5	—	—	µs
	Wait time after setting PV bit*1	γ		4	—	—	µs
	Wait time after dummy write*1	ε		2	—	—	µs
	Wait time after clearing PV bit*1	η		2	—	—	µs
	Wait time after clearing SWE bit*1	θ		100	—	—	µs
	Maximum programming count*1*4*5	N		—	—	1000	Times

Item		Symbol	Test Condition	Values Min.	Typ.	Max.	Unit
Erase	Wait time after setting SWE bit*[1]	x		1	—	—	µs
	Wait time after setting ESU bit*[1]	y		100	—	—	µs
	Wait time after setting E bit*[1]*[6]	z		10	—	100	ms
	Wait time after clearing E bit*[1]	α		10	—	—	µs
	Wait time after clearing ESU bit*[1]	β		10	—	—	µs
	Wait time after setting EV bit*[1]	γ		20	—	—	µs
	Wait time after dummy write*[1]	ε		2	—	—	µs
	Wait time after clearing EV bit*[1]	η		4	—	—	µs
	Wait time after clearing SWE bit*[1]	θ		100	—	—	µs
	Maximum erasing count*[1]*[6]*[7]	N		—	—	120	Times

Notes:
1. Make the time settings in accordance with the programming/erasing algorithms.
2. The programming time for 128 bytes. (Indicates the total time for which the P bit in the flash memory control register 1 (FLMCR1) is set. The programming-verifying time is not included.)
3. The time required to erase one block. (Indicates the total time for which the E bit in the flash memory control register 1 (FLMCR1) is set. The erasing-verifying time is not included.)
4. Programming time maximum value (t_P (max.)) = wait time after setting P bit (z) × maximum programming count (N)
5. Set the maximum programming count (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 setting P bit (z1, z2) should be changed as follows according to the value of the programming count (n).

   Programming count (n)

   $1 \leq n \leq 6$ z1 = 30 µs

   $7 \leq n \leq 1000$ z2 = 200 µs
6. Erasing time maximum value (t_E (max.)) = wait time after setting E bit (z) × maximum erasing count (N)
7. Set the maximum erasing count (N) according to the actual set value of (z), so that it does not exceed the erasing time maximum value (t_E (max.)).
8. The minimum number of times in which all characteristics are guaranteed following reprogramming. (The guarantee covers the range from 1 to the minimum value.)
9. Reference value at 25°C. (Guideline showing programming count over which functioning will be retained under normal circumstances.)
10. Data retention characteristics within the range indicated in the specifications, including the minimum programming count.
11. Applies to an operating voltage range when reading data of 2.7 to 3.6 V.
12. Applies to an operating voltage range when reading data of 1.8 to 3.6 V.

21.3 Absolute Maximum Ratings for Masked ROM Version

Table 21.12 lists the absolute maximum ratings.

Table 21.12 Absolute Maximum Ratings

Item		Symbol	Value	Unit	Note
Power supply voltage		V_{CC}	–0.3 to +4.3	V	*
Analog power supply voltage		AV_{CC}	–0.3 to +4.3	V	
Input voltage	Other than port B	V_{in}	–0.3 to V_{CC} +0.3	V	
	Port B	AV_{in}	–0.3 to AV_{CC} +0.3	V	
Operating temperature		T_{opr}	–20 to +75 (general specifications)	°C	
			–40 to +85 (wide temperature range specifications)		
Storage temperature		T_{stg}	–55 to +125	°C	

Note: * Permanent damage may occur to the chip if absolute 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.

21.4 Electrical Characteristics for Masked ROM Version

21.4.1 Power Supply Voltage and Operating Range

The power supply voltage and operating range are indicated by the shaded region in the figures.

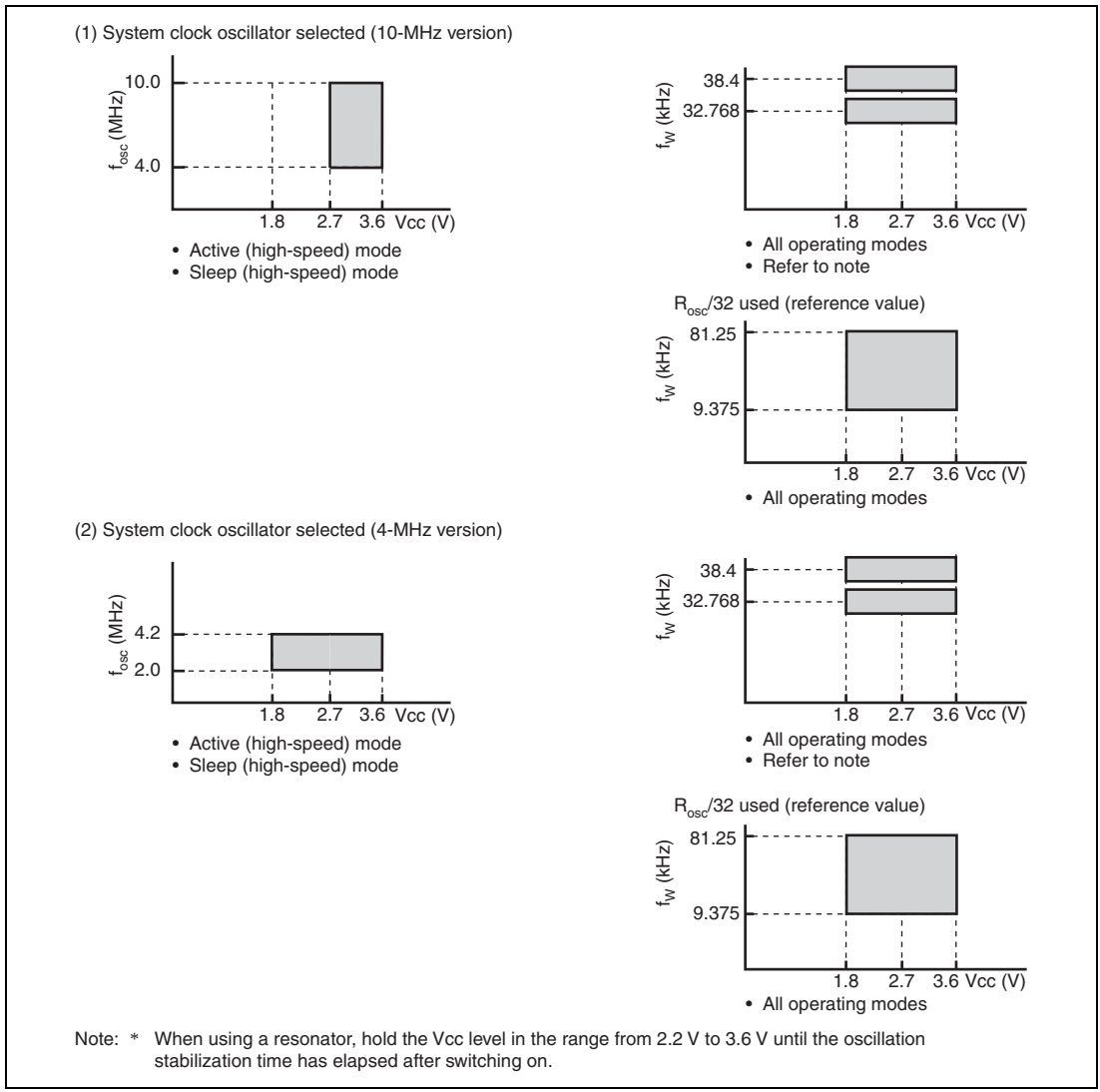
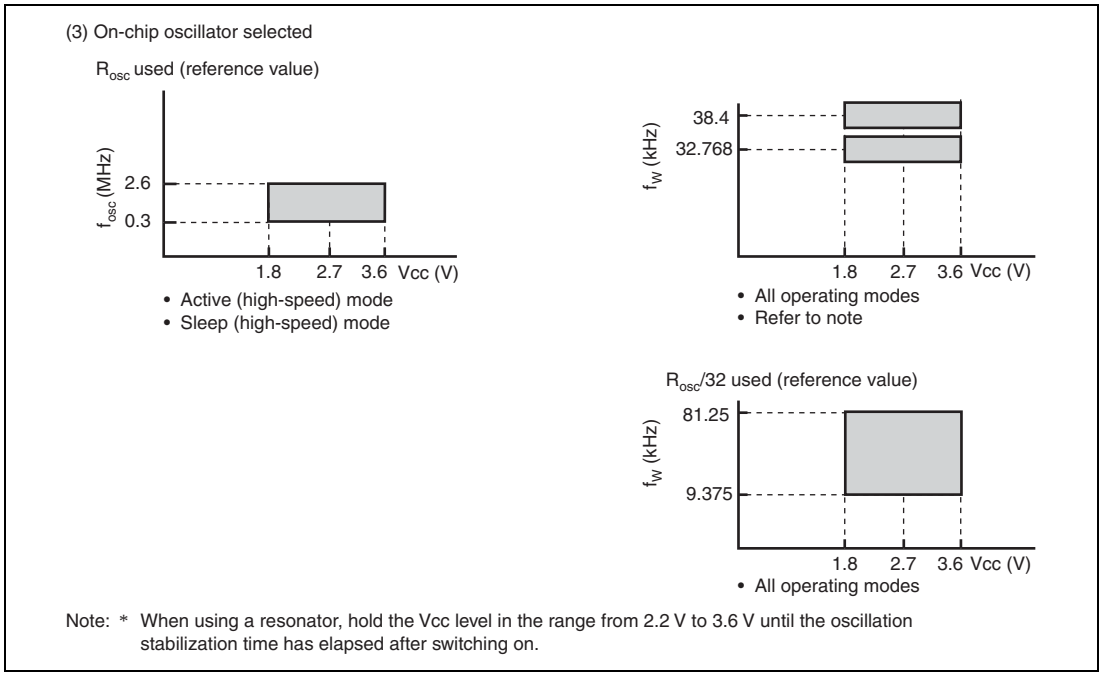


Figure 21.8 Power Supply Voltage and Oscillation Frequency Range (1)



Note: * When using a resonator, hold the Vcc level in the range from 2.2 V to 3.6 V until the oscillation stabilization time has elapsed after switching on.

Figure 21.9 Power Supply Voltage and Oscillation Frequency Range (2)

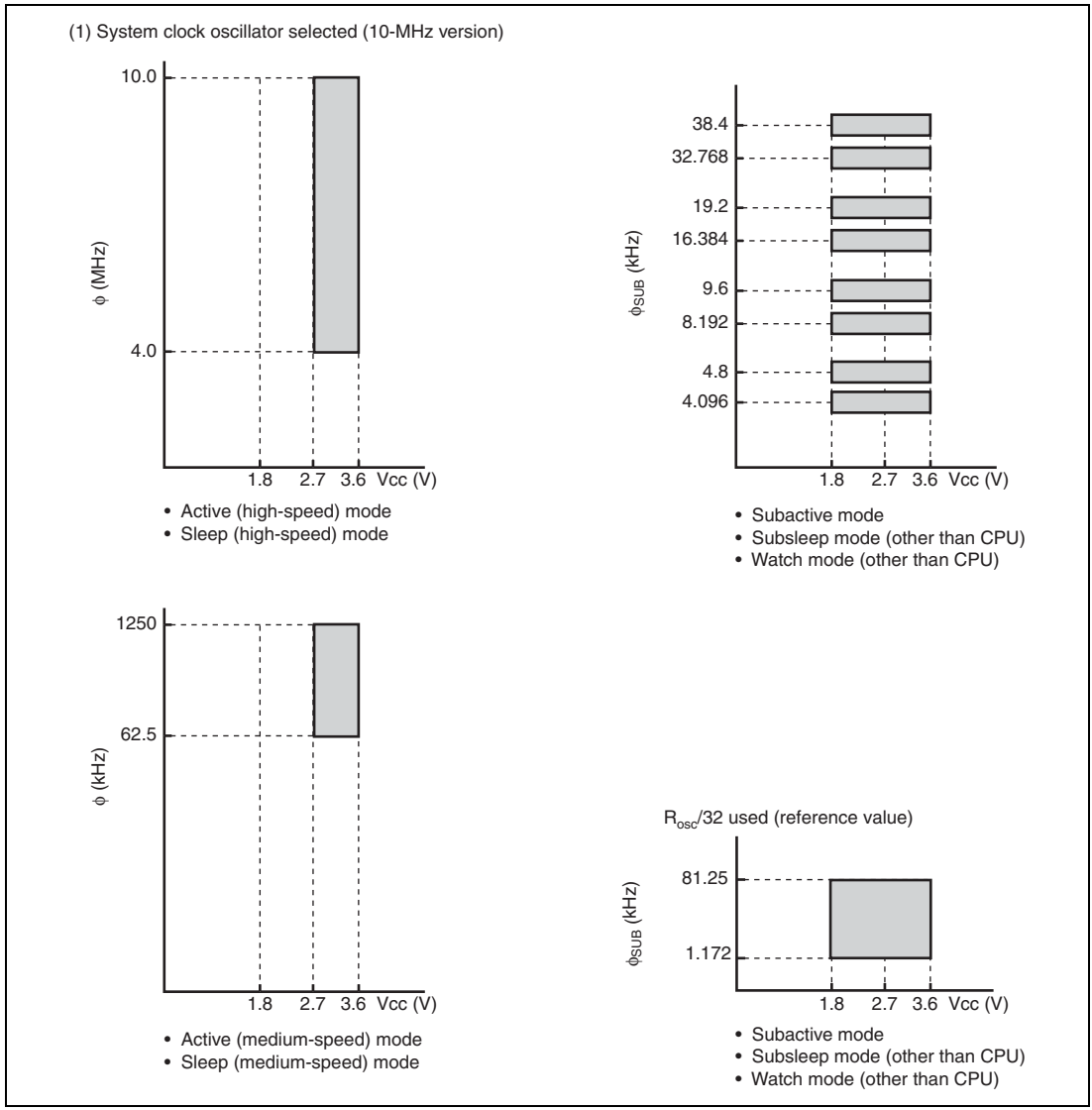


Figure 21.10 Power Supply Voltage and Operating Frequency Range (1)

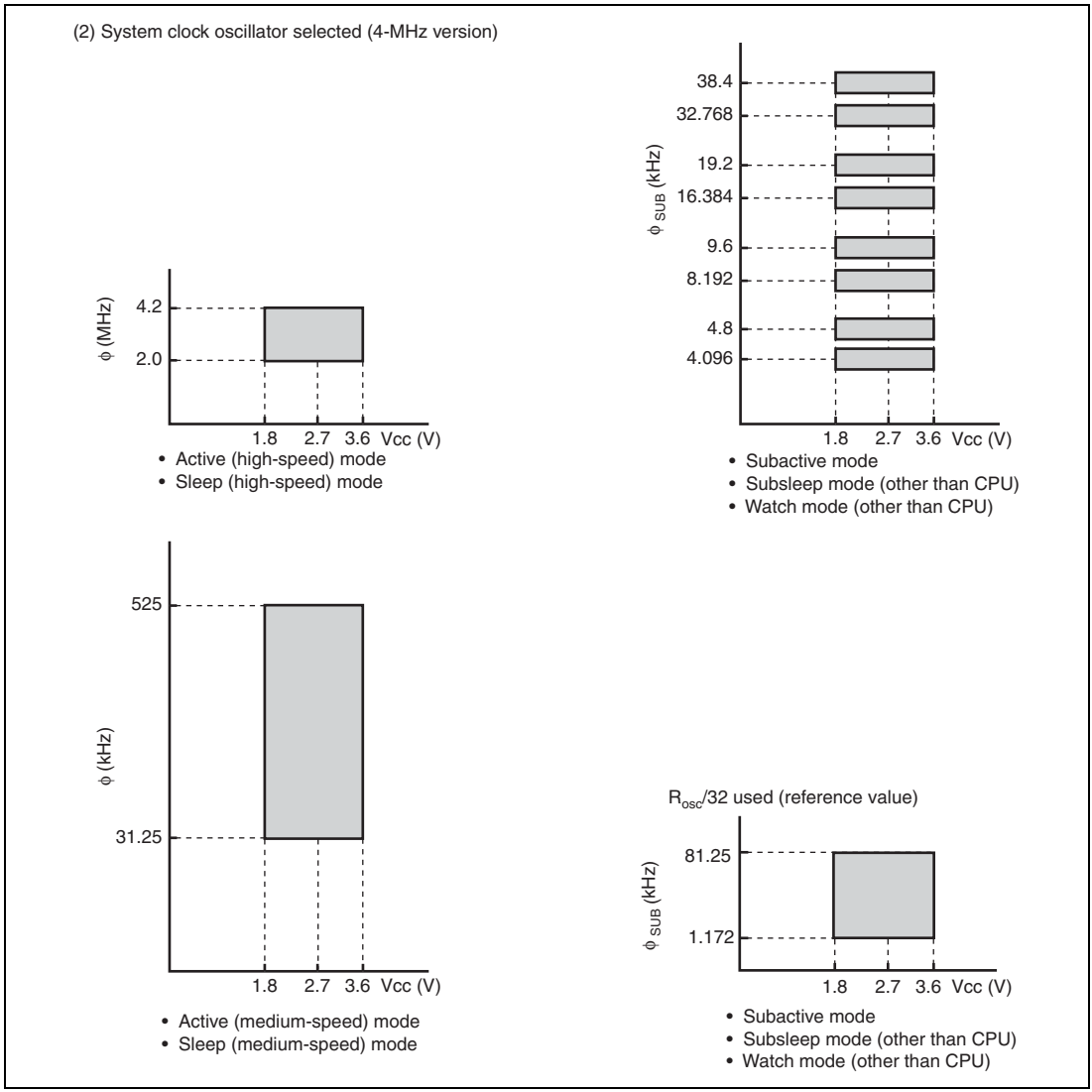


Figure 21.11 Power Supply Voltage and Operating Frequency Range (2)

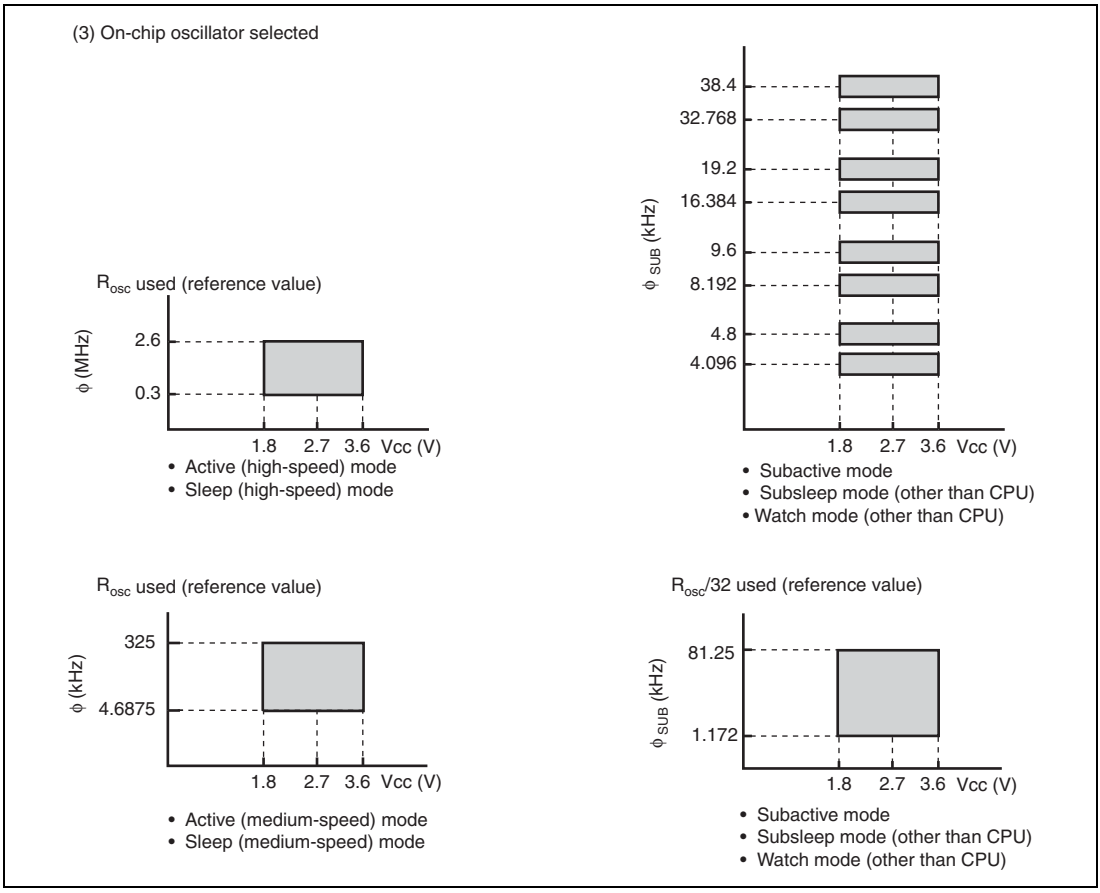


Figure 21.12 Power Supply Voltage and Operating Frequency Range (3)

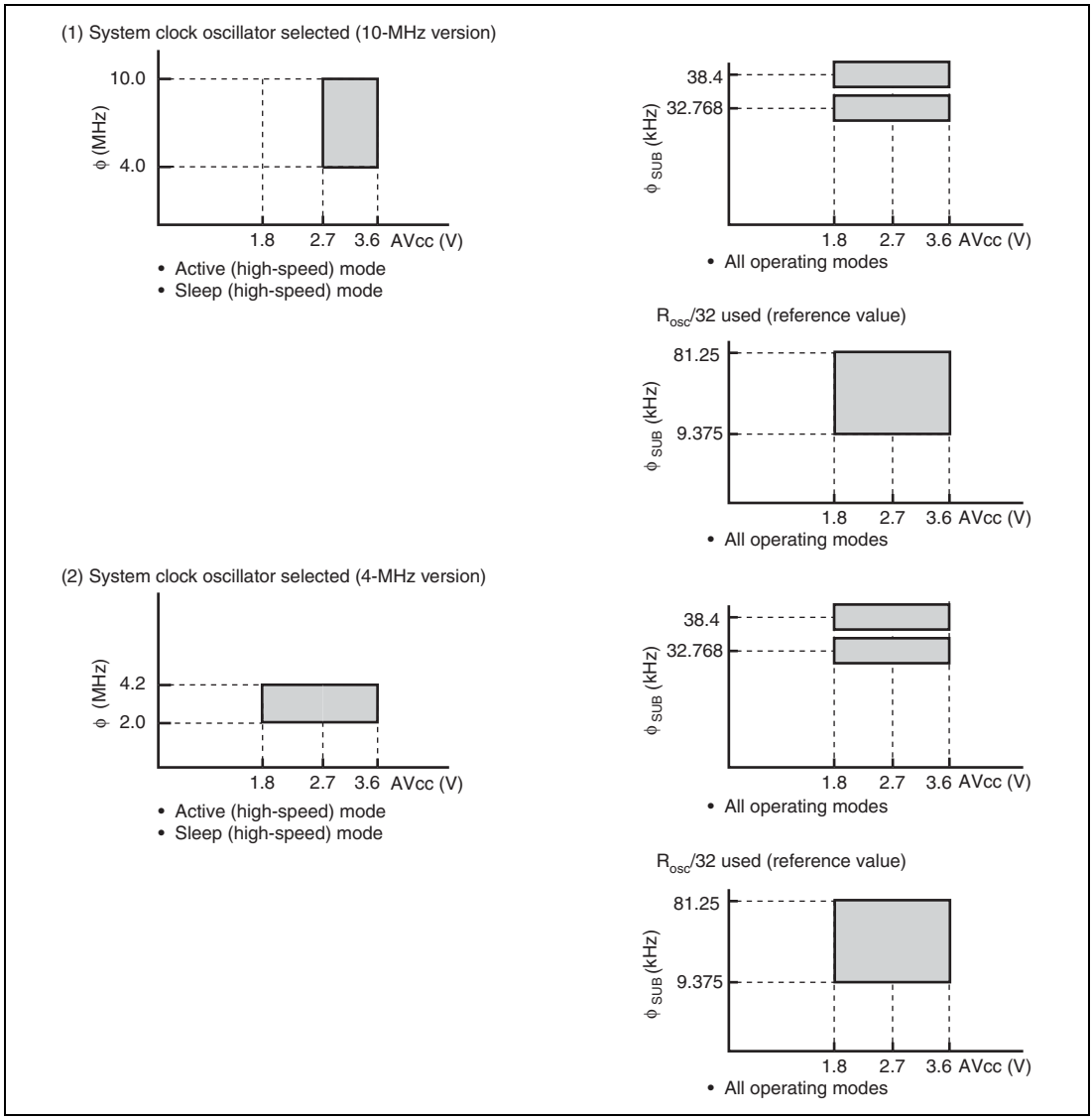


Figure 21.13 Analog Power Supply Voltage and Operating Frequency Range of A/D Converter (1)

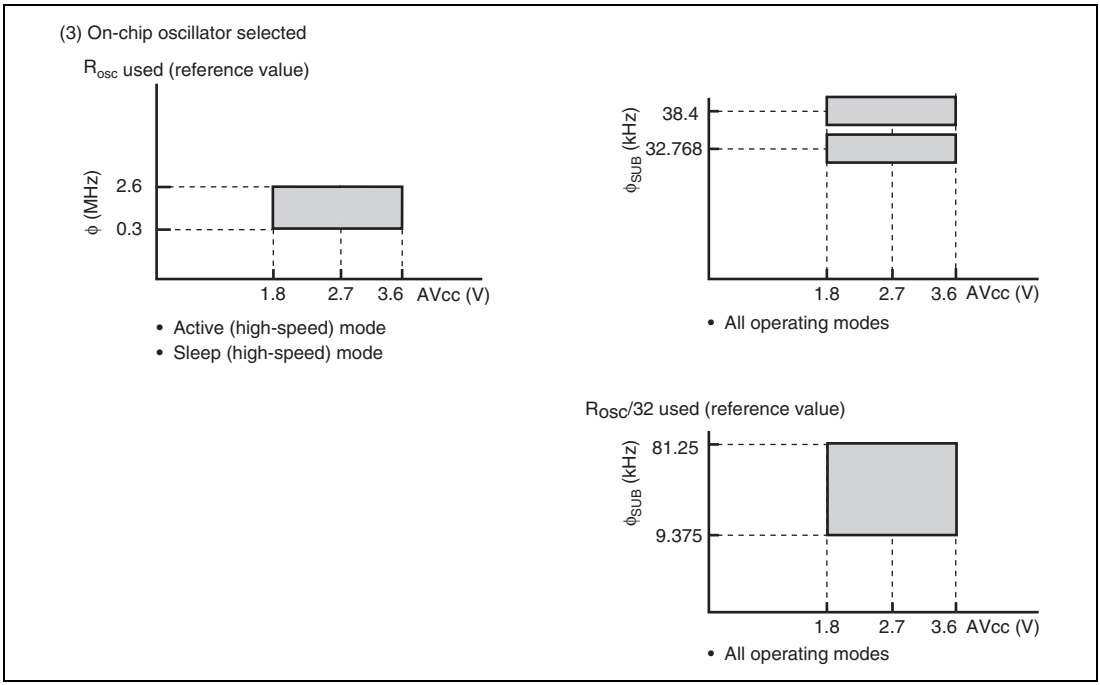


Figure 21.14 Analog Power Supply Voltage and Operating Frequency Range of A/D Converter (2)

21.4.2 DC Characteristics

Table 21.13 lists the DC characteristics.

Table 21.13 DC Characteristics

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, unless otherwise specified.

				Values				
Item	**Symbol**	**Applicable Pins**	**Test Condition**	**Min.**	**Typ.**	**Max.**	**Unit**	**Notes**
Input high voltage	V_{IH}	$\overline{RES}$, TEST, $\overline{NMI}$, AEVL, AEVH, $\overline{ADTRG}$, SCK3, IRQAEC		$0.9V_{CC}$	—	V_{CC} + 0.3	V	
		$\overline{IRQ0}$*[3], $\overline{IRQ1}$*[3]		$0.9V_{CC}$	—	AV_{CC} + 0.3		
		RXD3, IrRXD		$0.8V_{CC}$	—	V_{CC} + 0.3		
		OSC1		$0.9V_{CC}$	—	V_{CC} + 0.3		
		X1		$0.9V_{CC}$	—	V_{CC} + 0.3		
		P10 to P12, P30 to P32, P82 to P84, P90 to P93, SSI, SSO, SSCK, $\overline{SCS}$, FTCI, FTIOA, FTIOB, FTIOC, FTIOD, E7_0 to E7_2, SCL, SDA		$0.8V_{CC}$	—	V_{CC} + 0.3		
		PB0 to PB5		$0.8V_{CC}$	—	AV_{CC} + 0.3		

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Notes
Input low voltage	V_{IL}	TEST, $\overline{NMI}$, $\overline{IRQ0}$, $\overline{IRQ1}$, IRQAEC, AEVL, AEVH, $\overline{ADTRG}$, SCK3		–0.3	—	$0.1V_{CC}$	V	
		RXD3, IrRXD		–0.3	—	$0.2V_{CC}$		
		OSC1		–0.3	—	$0.1V_{CC}$		
		X1		–0.3	—	$0.1V_{CC}$		
		P10 to P12, P30 to P32, P82 to P84, P90 to P93, SCL, SDA, PB0 to PB5, SSI, SSO, SSCK, $\overline{SCS}$, FTCI, FTIOA, FTIOB, FTIOC, FTIOD, E7_0 to E7_2		–0.3	—	$0.2V_{CC}$		
Output high voltage	V_{OH}	P10 to P12, P30 to P32, P90 to P93	$-I_{OH}$ = 1.0 mA Vcc = 2.7 V to 3.6 V	V_{CC} – 1.0	—	—	V	
			$-I_{OH}$ = 0.1 mA	V_{CC} – 0.3	—	—		
		P82 to P84	$-I_{OH}$ = 1.0 mA V_{CC} = 2.7 V to 3.6 V	V_{CC} – 1.0	—	—		
			$-I_{OH}$ = 0.1 mA	V_{CC} – 0.3	—	—		
Output low voltage	V_{OL}	P10 to P12, P30 to P32, P90 to P93	I_{OL} = 0.4 mA	—	—	0.5	V	
		P82 to P84	I_{OL} = 15 mA, Vcc = 2.7 V to 3.6 V	—	—	1.0		
			I_{OL} = 10 mA, Vcc = 2.2 V to 3.6 V	—	—	0.5		
			I_{OL} = 8 mA	—	—	0.5		
		SCL, SDA	I_{OL} = 3.0 mA	—	—	0.4		

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Notes
Input/output leakage current	$\lvert I_{IL} \rvert$	TEST, $\overline{NMI}$, OSC1, X1, P10 to P12, P30 to P32, P82 to P84, P90 to P93, E7_0 to E7_2	V_{IN} = 0.5 V to V_{CC} – 0.5 V	—	—	1.0	µA	
		PB0 to PB5	V_{IN} = 0.5 V to AV_{CC} – 0.5 V	—	—	1.0		
Pull-up MOS current	$-I_p$	P10 to P12, P30 to P32, P82 to P84, P90 to P93	V_{CC} = 3 V, V_{IN} = 0 V	30	—	180	µA	
Input capacitance	C_{IN}	All input pins except power supply pin	f = 1 MHz, V_{IN} =0 V, Ta = 25°C	—	—	15.0	pF	
Active mode supply current	I_{OPE1}	V_{CC}	Active (high-speed) mode, V_{CC} = 1.8 V, f_{OSC} = 2 MHz	—	0.5	—	mA	Max. guideline = 1.1 × typ.*1*2
			Active (high-speed) mode, V_{CC} = 3 V, f_{OSC} = R_{OSC}	—	0.6	—		Max. guideline = 1.1 × typ.*1*2 Reference value
			Active (high-speed) mode, V_{CC} = 3 V, f_{OSC} = 4.2 MHz	—	2.0	3.0		*1*2 4-MHz version
			Active (high-speed) mode, V_{CC} = 3 V, f_{OSC} = 10 MHz	—	4.5	6.8		*1*2 10-MHz version

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Notes
Active mode supply current	I_{OPE2}	V_{CC}	Active (medium-speed) mode, V_{CC} = 1.8 V, f_{OSC} = 2 MHz, ϕ_{osc}/64	—	0.1	—	mA	Max. guideline = 1.1 × typ. *1*2
			Active (medium-speed) mode, V_{CC} = 3 V, f_{OSC} = 4.2 MHz, ϕ_{osc}/64	—	0.3	0.5		*1*2 4-MHz version
			Active (medium-speed) mode, V_{CC} = 3 V, f_{OSC} = 10 MHz, ϕ_{osc}/64	—	0.5	0.7		*1*2 10-MHz version
Sleep mode supply current	I_{SLEEP}	V_{CC}	V_{CC} = 1.8 V, f_{OSC} = 2 MHz	—	0.3	—	mA	Max. guideline = 1.1 × typ. *1*2
			V_{CC} = 3 V, f_{OSC} = 4.2 MHz	—	1.0	1.5		*1*2 4-MHz version
			V_{CC} = 3 V, f_{OSC} = 10 MHz	—	1.8	2.7		*1*2 10-MHz version
Subactive mode supply current	I_{SUB}	V_{CC}	V_{CC} = 1.8 V, 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	4.0	—	µA	*1*2 Reference value
			V_{CC} = 2.7 V, 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/8$)	—	3.6	—		*1*2 Reference value
			V_{CC} = 2.7 V, 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	7.4	—		*1*2 Reference value
			V_{CC} = 2.7 V, on-chip oscillator/32 ($\phi_{SUB} = \phi_W = R_{OSC}/32$)	—	40	—		*1*2 Reference value
			V_{CC} = 2.7 V, 32-kHz crystal resonator ($\phi_{SUB} = \phi_W$)	—	13	25		*1*2

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Notes
Subsleep mode supply current	I_{SUBSP}	V_{CC}	V_{CC} = 2.7 V, 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	3.1	—	µA	*1*2 Reference value
			V_{CC} = 2.7 V, on-chip oscillator/32 ($\phi_{SUB} = \phi_W = R_{OSC}/32$)	—	30	—		*1*2 Reference value
			V_{CC} = 2.7 V, 32-kHz crystal resonator ($\phi_{SUB} = \phi_W$)	—	5.0	10.0		*1*2
Watch mode supply current	I_{WATCH}	V_{CC}	V_{CC} = 1.8 V, Ta = 25°C, 32-kHz crystal resonator	—	0.4	—	µA	*1*2 Reference value
			V_{CC} = 2.7 V, 32-kHz crystal resonator	—	1.5	5.0		*1*2
Standby mode supply current	I_{STBY}	V_{CC}	V_{CC} = 3.0 V, Ta = 25°C, 32-kHz crystal resonator not used	—	0.1	—	µA	*1*2 Reference value
			32-kHz crystal resonator not used	—	1.0	5.0		*1*2
RAM data retaining voltage	V_{RAM}	V_{CC}		1.5	—	—	V	
Permissible output low current (per pin)	I_{OL}	Output pins except port 8		—	—	0.5	mA	
		Port 8		—	—	15.0		
Permissible output low current (total)	$\sum I_{OL}$	Output pins except port 8		—	—	20.0	mA	
		Port 8		—	—	45.0		

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Notes
Permissible output high current (per pin)	$-I_{OH}$	All output pins	V_{CC} = 2.7 V to 3.6 V	—	—	2.0	mA	
			Other than above	—	—	0.2		
Permissible output high current (total)	$\sum -I_{OH}$	All output pins		—	—	10.0	mA	

Notes: 1. Pin states during current measurement.

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pins
Active (high-speed) mode (I_{OPE1})	V_{CC}	Only CPU operates	V_{CC}	System clock oscillator: Crystal resonator Subclock oscillator: Pin X1 = GND
Active (medium-speed) mode (I_{OPE2})				
Sleep mode	V_{CC}	Only on-chip timers operate	V_{CC}	
Subactive mode	V_{CC}	Only CPU operates	V_{CC}	System clock oscillator: Crystal resonator Subclock oscillator: Crystal resonator
Subsleep mode	V_{CC}	Only on-chip timers operate, CPU stops	V_{CC}	
Watch mode	V_{CC}	Only timer base operates, CPU stops	V_{CC}	
Standby mode	V_{CC}	CPU and timers both stop, SUBSTP = 1	V_{CC}	System clock oscillator: Crystal resonator Subclock oscillator: Pin X1 = Crystal resonator

2. Excludes current in pull-up MOS transistors and output buffers.
3. When bits IRQ0S1 and IRQ0S0 are set to B'01 or B'10, and bits IRQ1S1 and IRQ1S0 are set to B'01 or B'10, the maximum value is given V_{CC} + 0.3 (V).

21.4.3 AC Characteristics

Table 21.14 lists the control signal timing, table 21.15 lists the serial interface timing, table 21.16 lists the synchronous serial communication unit timing, and table 21.17 lists the I^2C bus interface timing.

Table 21.14 Control Signal Timing

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Values Typ.	Values Max.	Unit	Reference Figure
System clock oscillation frequency	f_{OSC}	OSC1, OSC2	V_{CC} = 2.7 V to 3.6 V (10-MHz version)	4.0	—	10.0	MHz	
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	2.0	—	4.2		
OSC clock (ϕ_{OSC}) cycle time	t_{OSC}	OSC1, OSC2	V_{CC} = 2.7 V to 3.6 V (10-MHz version)	100	—	250	ns	Figure 21.15
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	238	—	500		
System clock (ϕ) cycle time	t_{cyc}			1	—	64	t_{OSC}	
			V_{CC} = 2.7 V to 3.6 V (10-MHz version)	—	—	16	µs	
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	—	—	32		
On-chip oscillator oscillation frequency	t_{ROSC}			0.3	—	2.6	MHz	Reference value
On-chip oscillator clock cycle time	t_{ROSC}			0.38	—	3.3	µs	Reference value
Subclock oscillator oscillation frequency	f_W	X1, X2		—	32.768 or 38.4	—	kHz	
Watch clock (ϕ_W) cycle time	t_W	X1, X2		—	30.5 or 26.0	—	µs	Figure 21.15

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Reference Figure
Subclock (ϕ_{SUB}) cycle time	t_{subcyc}			1	—	8	t_W	*[1]
Instruction cycle time				2	—	—	t_{cyc} t_{subcyc}	
Oscillation stabilization time	t_{rc}	OSC1, OSC2	Ceramic resonator (V_{CC} = 2.2 V to 3.6 V)	—	20	45	µs	Figure 21.28
			Ceramic resonator (Other than above)	—	80	—		
			Crystal resonator (V_{CC} = 2.7 V to 3.6 V)	—	300	800		
			Crystal resonator (V_{CC} = 2.2 V to 3.6 V)	—	600	1000		
			Other than above	—	—	50	ms	
		On-chip oscillator	At switching on	—	15	25	µs	
		X1, X2	V_{CC} = 2.2 V to 3.6 V	—	—	2	s	Figures 4.6 and 4.7
			Other than above	—	4	—		
External clock high width	t_{CPH}	OSC1	V_{CC} = 2.7 V to 3.6 V (10-MHz version)	40	—	—	ns	Figure 21.15
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	95	—	—		
		X1		—	15.26 or 13.02	—	µs	
External clock low width	t_{CPL}	OSC1	V_{CC} = 2.7 V to 3.6 V (10-MHz version)	40	—	—	ns	Figure 21.15
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	95	—	—		
		X1		—	15.26 or 13.02	—	µs	

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Reference Figure
External clock rising time	t_{CPr}	OSC1	V_{CC} = 2.7 V to 3.6 V (10-MHz version)	—	—	10	ns	Figure 21.15
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	—	—	24		
		X1		—	—	55.0	ns	
External clock falling time	t_{CPf}	OSC1	V_{CC} = 2.7 V to 3.6 V (10-MHz version)	—	—	10	ns	Figure 21.15
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	—	—	24		
		X1		—	—	55.0	ns	
$\overline{RES}$ pin low width	t_{REL}	$\overline{RES}$	At switching on or other than below	t_{rc} + 20 × t_{cyc}	—	—	µs	Figure 21.16*[2]
			Active mode or sleep mode	20			t_{cyc}	
Input pin high width	t_{IH}	$\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{NMI}$, IRQAEC, $\overline{ADTRG}$, FTCI, FTIOA, FTIOB, FTIOC, FTIOD		2	—	—	t_{cyc} t_{subcyc}	Figure 21.17
		AEVL, AEVH	V_{CC} = 2.7 V to 3.6 V (10-MHz version)	50	—	—	ns	
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	110	—	—		
Input pin low width	t_{IL}	$\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{NMI}$, IRQAEC, $\overline{ADTRG}$, FTCI, FTIOA, FTIOB, FTIOC, FTIOD		2	—	—	t_{cyc} t_{subcyc}	Figure 21.17
		AEVL, AEVH	V_{CC} = 2.7 V to 3.6 V (10-MHz version)	50	—	—	ns	
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	110	—	—		

Notes: 1. Selected with the SA1 and SA0 bits in the system control register 2 (SYSCR2).
2. For details on the power-on reset characteristics, refer to table 21.21 and figure 21.26.

Table 21.15 Serial Interface Timing

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, unless otherwise specified.

Item		Symbol	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
Input clock cycle	Asynchronous	t_{scyc}		4	—	—	t_{cyc} or t_{subcyc}	Figure 21.18
	Clock synchronous			6	—	—		
Input clock pulse width		t_{SCKW}		0.4	—	0.6	t_{scyc}	Figure 21.18
Transmit data delay time (clock synchronous)		t_{TXD}		—	—	1	t_{cyc} or t_{subcyc}	Figure 21.19
Receive data setup time (clock synchronous)		t_{RXS}		400.0	—	—	ns	Figure 21.19
Receive data hold time (clock synchronous)		t_{RXH}		400.0	—	—	ns	Figure 21.19

Table 21.16 Synchronous Serial Communication Unit (SSU) Timing

V_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, output load = 100 pF, unless otherwise specified.

Item		Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
					Min.	Typ.	Max.		
Clock cycle		t_{sucyc}	SSCK		4	—	—	t_{cyc}	Figures 21.20 to 21.24
Clock high pulse width		t_{HI}	SSCK		0.4	—	0.6	t_{sucyc}	
Clock low pulse width		t_{LO}	SSCK		0.4	—	0.6	t_{sucyc}	
Clock rising time	Master	t_{RISE}	SSCK		—	—	1	t_{cyc}	
	Slave				—	—	1.0	µs	
Clock falling time	Master	t_{FALL}	SSCK		—	—	1	t_{cyc}	
	Slave				—	—	1.0	µs	
Data input setup time		t_{SU}	SSO SSI		1	—	—	t_{cyc}	
Data input hold time		t_{H}	SSO SSI		1	—	—	t_{cyc}	
$\overline{SCS}$ setup time	Slave	t_{LEAD}	$\overline{SCS}$		$1t_{cyc}$ + 100	—	—	ns	
$\overline{SCS}$ hold time	Slave	t_{LAG}	$\overline{SCS}$		$1t_{cyc}$ + 100	—	—	ns	
Data output delay time		t_{OD}	SSO SSI		—	—	1	t_{cyc}	
Slave access time		t_{SA}	SSI		—	—	$1t_{cyc}$ + 100	ns	
Slave out release time		t_{OR}	SSI		—	—	$1t_{cyc}$ + 100	ns	

Table 21.17 I²C Bus Interface Timing

V_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, Ta = –20 to +75°C, unless otherwise specified.

Item	Symbol	Test Condition	Values			Unit	Reference Figure
			Min.	Typ.	Max.		
SCL input cycle time	t_{SCL}		$12t_{cyc}$ + 600	—	—	ns	Figure 21.25
SCL input high width	t_{SCLH}		$3t_{cyc}$ + 300	—	—	ns	
SCL input low width	t_{SCLL}		$5t_{cyc}$ + 300	—	—	ns	
Falling time for SCL and SDA inputs	t_{Sf}		—	—	300	ns	
Pulse width of spike on SCL and SDA to be suppressed	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	
Repeated start condition input setup time	t_{STAS}		$3t_{cyc}$	—	—	ns	
Stop condition input setup time	t_{STOS}		$3t_{cyc}$	—	—	ns	
Data-input setup time	t_{SDAS}		$1t_{cyc}$ + 20	—	—	ns	
Data-input hold time	t_{SDAH}		0	—	—	ns	
Capacitive load of SCL and SDA	Cb		0	—	400	pF	
Falling time of SCL and SDA output	t_{Sf}		—	—	300	ns	

21.4.4 A/D Converter Characteristics

Table 21.18 lists the A/D converter characteristics.

Table 21.18 A/D Converter Characteristics

V_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Notes
				Min.	Typ.	Max.		
Analog power supply voltage	AV_{CC}	AV_{CC}		1.8	—	3.6	V	*1
Analog input voltage	AV_{IN}	AN0 to AN5		–0.3	—	AV_{CC} + 0.3	V	
Analog power supply current	AI_{OPE}	AV_{CC}	AV_{CC} = 3.0 V	—	—	1.0	mA	
	AI_{STOP1}	AV_{CC}		—	600	—	µA	*2 Reference value
	AI_{STOP2}	AV_{CC}		—	—	5	µA	*3
Analog input capacitance	C_{AIN}	AN0 to AN5		—	—	15.0	pF	
Permissible signal source impedance	R_{AIN}			—	—	10.0	kΩ	
Resolution (data length)				—	—	10	Bits	
Nonlinearity error			AV_{CC} = 2.7 V to 3.6 V V_{CC} = 2.7 V to 3.6 V	—	—	±3.5	LSB	Other than subclock operation
			AV_{CC} = 2.0 V to 3.6 V V_{CC} = 2.0 V to 3.6 V	—	—	±5.5		
			Subclock operating	—	—	±5.5		Subactive or subsleep mode, conversion time = 31/ϕ_W
			Other than above	—	—	±7.5		*4
Quantization error				—	—	±0.5	LSB	

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Notes
Absolute accuracy			AV_{CC} = 2.7 V to 3.6 V V_{CC} = 2.7 V to 3.6 V	—	—	±4.0	LSB	
			AV_{CC} = 2.0 V to 3.6 V V_{CC} = 2.0 V to 3.6 V	—	—	±6.0		
			Subclock operating	—	—	±6.0		Subactive or subsleep mode, conversion time = $31/\phi_W$
			Other than above	—	—	±8.0		*4
Conversion time			AV_{CC} = 2.7 V to 3.6 V V_{CC} = 2.7 V to 3.6 V	12.4	—	124	µs	System clock oscillator is selected
				31	62	124		On-chip oscillator is selected Reference value (f_{ROSC} = 1 MHz)
				—	807	—		ϕ_{SUB} = 38.4 kHz
				—	945	—		ϕ_{SUB} = 32.8 kHz
				—	992	—		$\phi_{SUB} = R_{OSC}/32$ Reference value (f_{ROSC} = 1 MHz)
			Other than AV_{CC} = 2.7 V to 3.6 V V_{CC} = 2.7 V to 3.6 V	29.5	—	124		System clock oscillator is selected
				31	62	124		On-chip oscillator is selected Reference value (f_{ROSC} = 1 MHz)
				—	807	—		ϕ_{SUB} = 38.4 kHz
				—	945	—		ϕ_{SUB} = 32.8 kHz
				—	992	—		$\phi_{SUB} = R_{OSC}/32$ Reference value (f_{ROSC} = 1 MHz)

Notes: 1. Connect AV_{CC} to V_{CC} when the A/D converter is not used.
2. AI_{STOP1} is the current flowing through the ladder resistor while the A/D converter is idle.
3. AI_{STOP2} is the current flowing at a reset, in standby mode or watch mode, through the ladder resistor while the A/D converter is idle.
4. Conversion time is 29.5 µs.

21.4.5 Comparator Characteristics

Table 21.19 shows the comparator characteristics.

Table 21.19 Comparator Characteristics

V_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, unless otherwise specified.

		Values				
Item	**Test Condition**	**Min.**	**Typ.**	**Max.**	**Unit**	**Notes**
Accuracy	1LSB = V_{CC}/30	—	1/2	—	LSB	Comparing with internal resistor network
Conversion time		—	—	15	µs	
External input reference voltage	VCref pin	0.9	—	0.9 × V_{CC}	V	
Internal resistance compare voltage		0.9	—	26/30 × V_{CC}	V	
Comparator input voltage	COMP0 and COMP1 pins	–0.3	—	AV_{CC} + 0.3	V	
Ladder resistance		—	3	—	MΩ	Reference value

21.4.6 Watchdog Timer Characteristics

Table 21.20 shows the watchdog timer characteristics.

Table 21.20 Watchdog Timer Characteristics

V_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, unless otherwise specified.

		Applicable		Values				
Item	**Symbol**	**Pins**	**Test Condition**	**Min.**	**Typ.**	**Max.**	**Unit**	**Notes**
On-chip oscillator overflow time	t_{OVF}			0.2	0.4	—	s	*

Note: * Indicates that the period from when the counter starts with 0 to when the counter reaches 255 and an internal reset occurs while the on-chip oscillator is selected.

21.4.7 Power-On Reset Circuit Characteristics

Table 21.21 lists the power-on reset circuit characteristics.

Table 21.21 Power-On Reset Circuit Characteristics

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V,
Ta = –20 to +75°C (general specifications), Ta = –40 to +85°C (wide temperature range specifications), unless otherwise specified.

			Values				
Item	**Symbol**	**Test Condition**	**Min.**	**Typ.**	**Max.**	**Unit**	**Notes**
Reset voltage	V_rst		0.7Vcc	0.8Vcc	0.9Vcc	V	
Power supply rising time	t_vtr		The Vcc rising time should be shorter than half the $\overline{RES}$ rising time.				
Reset count time	t_out		0.8	—	4.0	μs	
			3.2	—	26.7		On-chip oscillator is selected (reference value)
Count start time	t_cr		Adjustable by the value of the external capacitor connected to the $\overline{RES}$ pin.				
Pull-up resistance	R_P		60	100	—	kΩ	

21.5 Operation Timing

Figures 21.15 to 21.26 show operation timings.

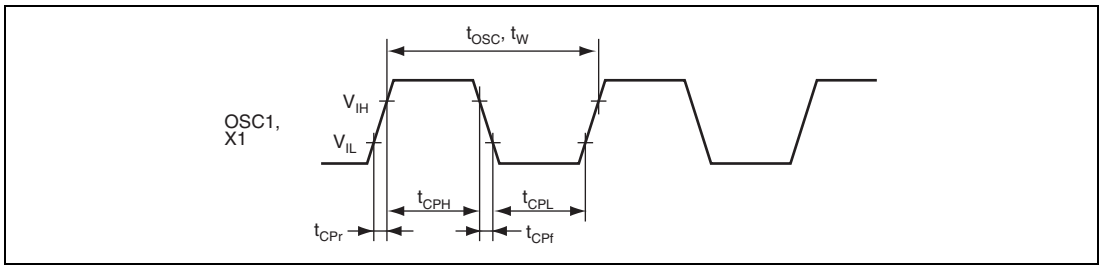


Figure 21.15 Clock Input Timing

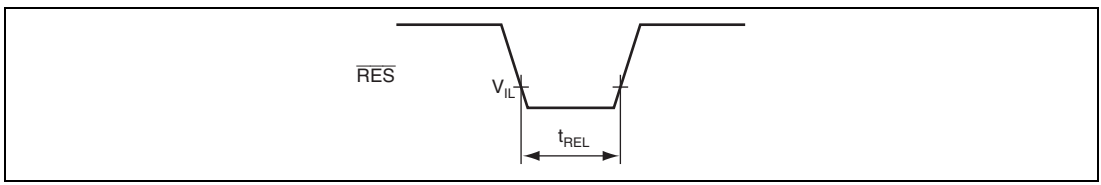


Figure 21.16 $\overline{RES}$ Low Width Timing

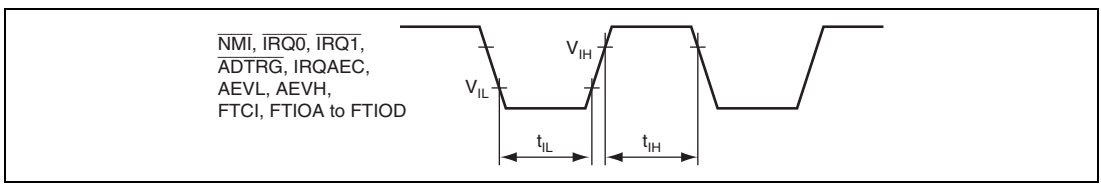


Figure 21.17 Input Timing

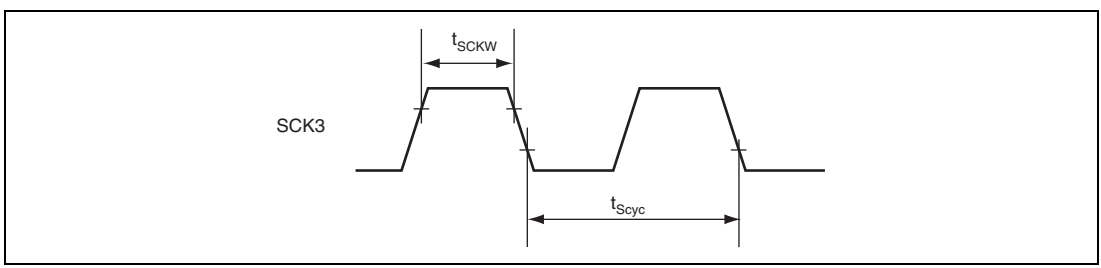
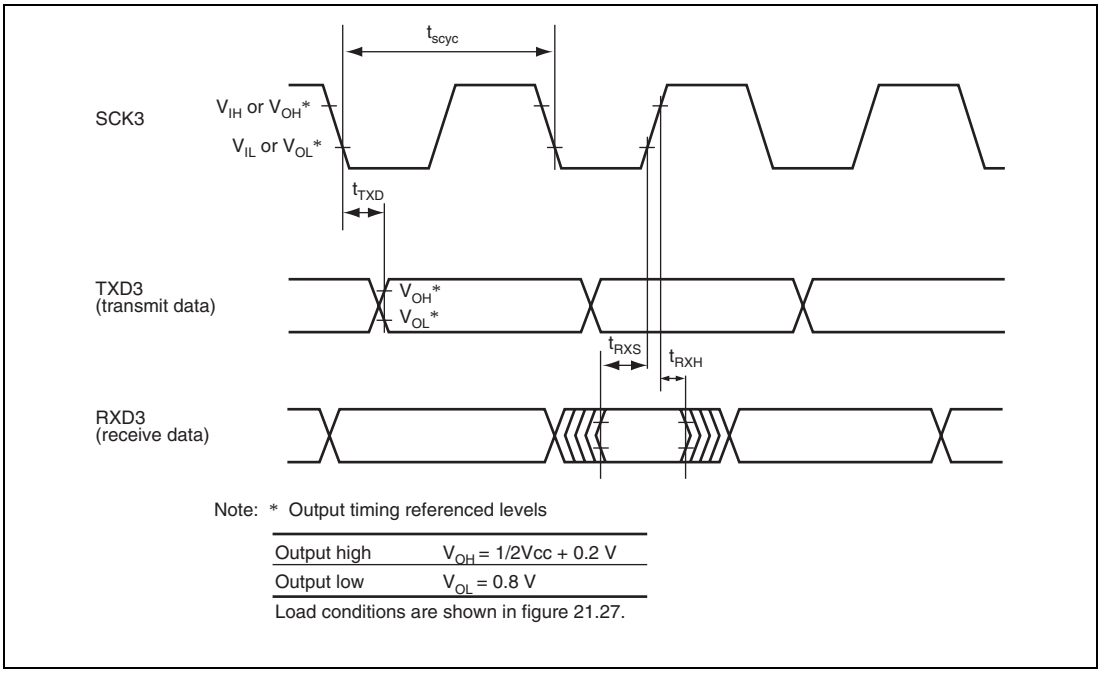


Figure 21.18 SCK3 Input Clock Timing



Note: * Output timing referenced levels

Output high	V_{OH} = 1/2Vcc + 0.2 V
Output low	V_{OL} = 0.8 V

Load conditions are shown in figure 21.27.

Figure 21.19 SCI3 Input/Output Timing in Clock Synchronous Mode

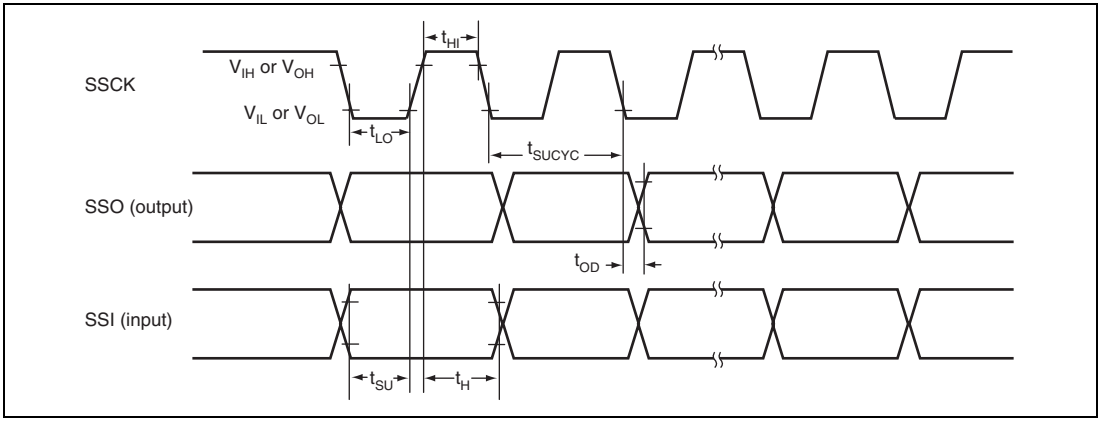


Figure 21.20 SSU Input/Output Timing in Clock Synchronous Mode

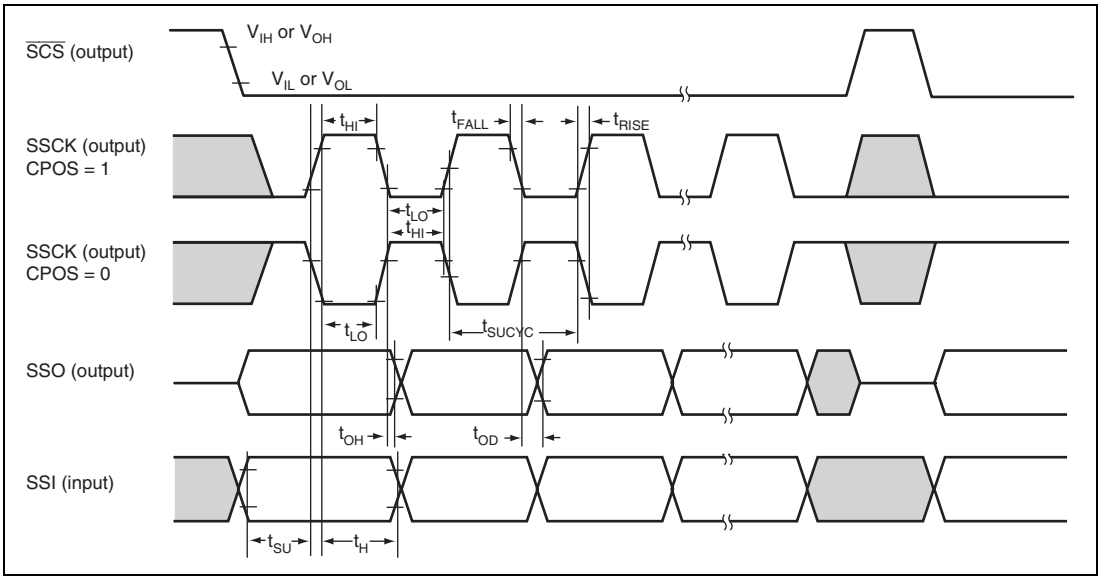


Figure 21.21 SSU Input/Output Timing
(Four-Line Bus Communication Mode, Master, CPHS = 1)

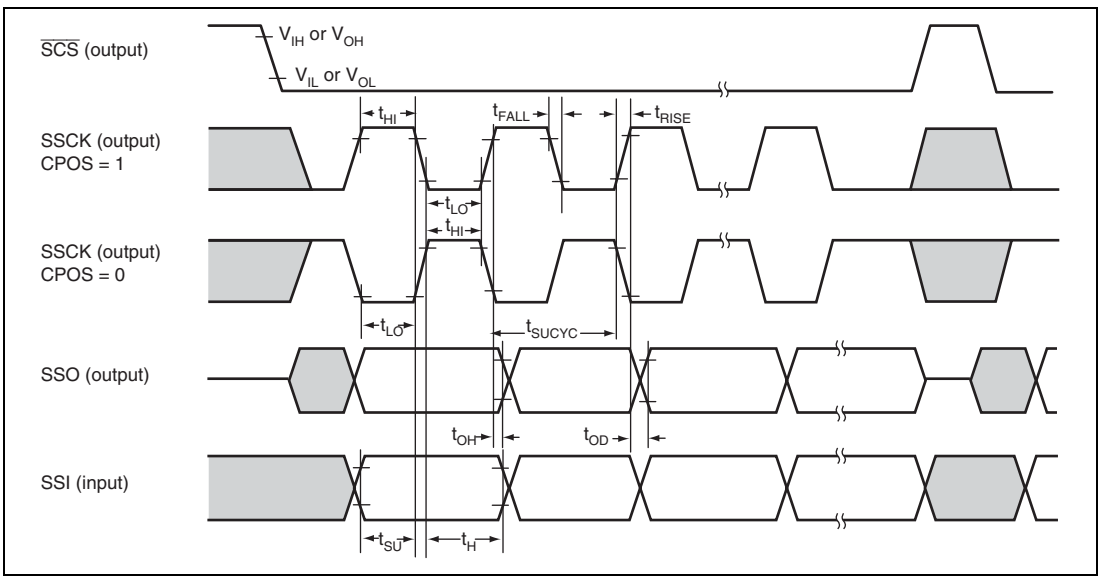


Figure 21.22 SSU Input/Output Timing
(Four-Line Bus Communication Mode, Master, CPHS = 0)

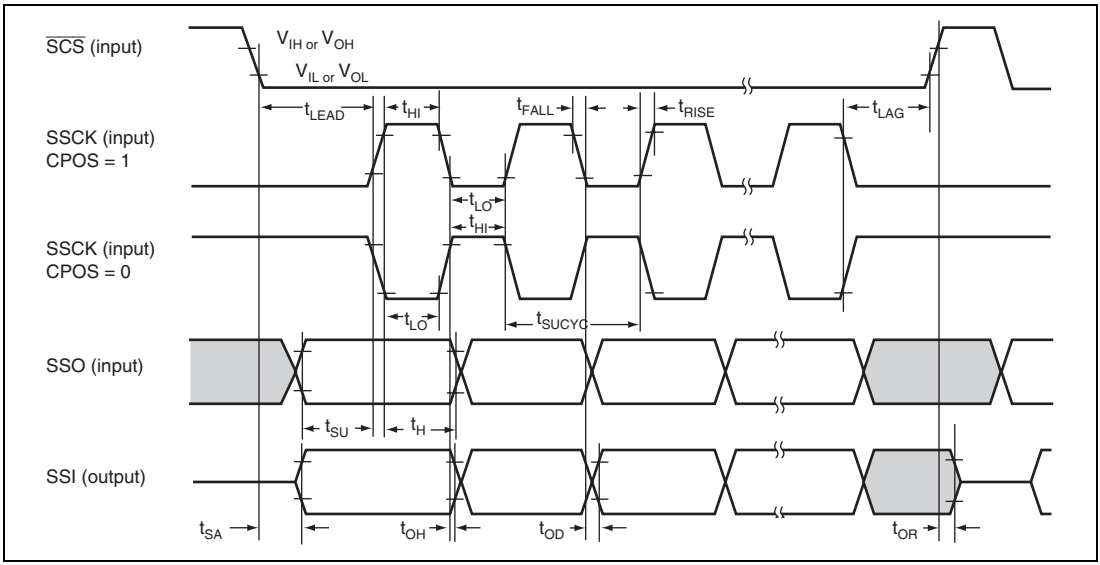


Figure 21.23 SSU Input/Output Timing
(Four-Line Bus Communication Mode, Slave, CPHS = 1)

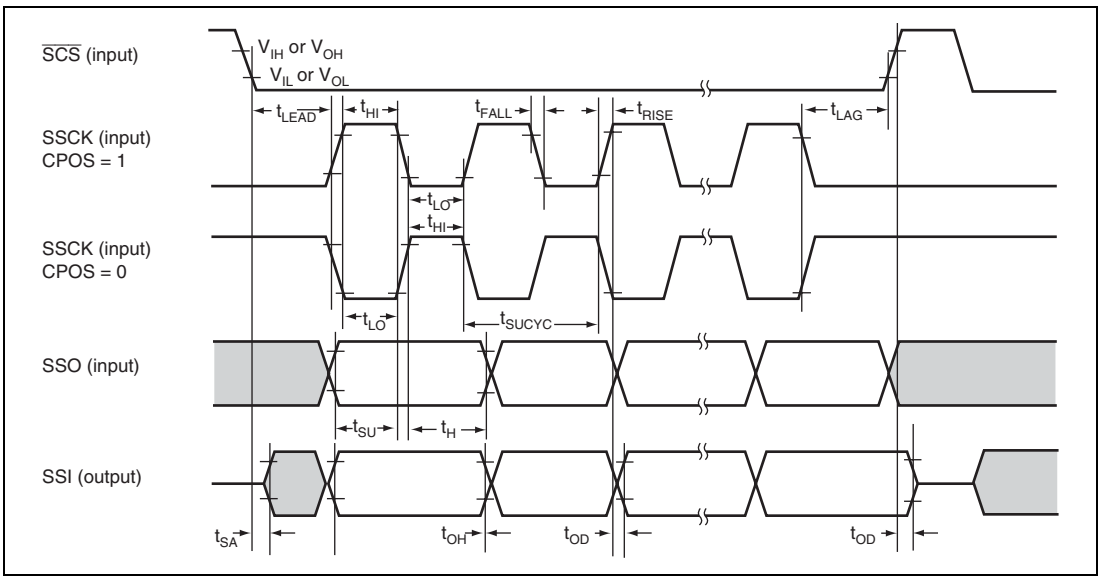


Figure 21.24 SSU Input/Output Timing
(Four-Line Bus Communication Mode, Slave, CPHS = 0)

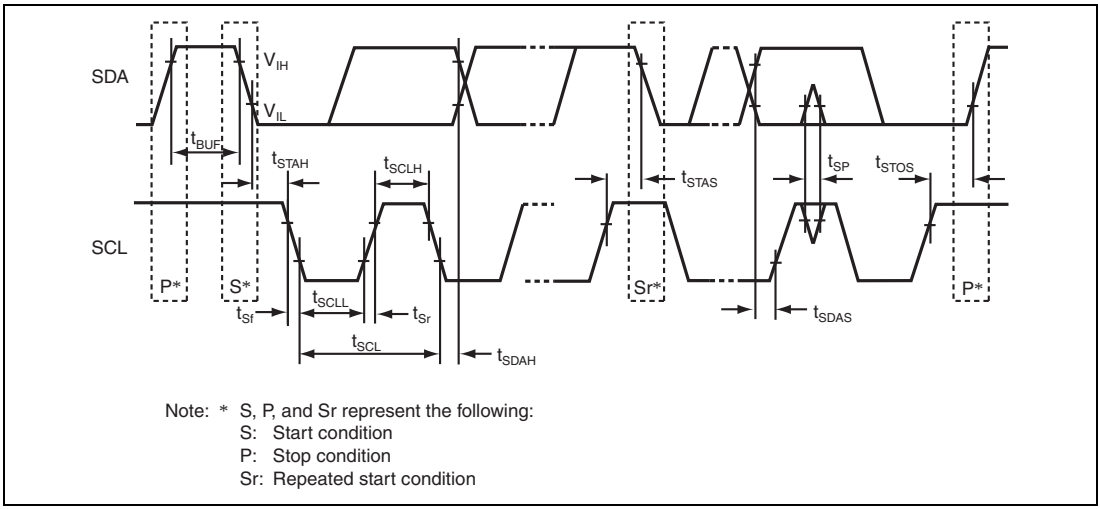


Figure 21.25 I²C Bus Interface Input/Output Timing

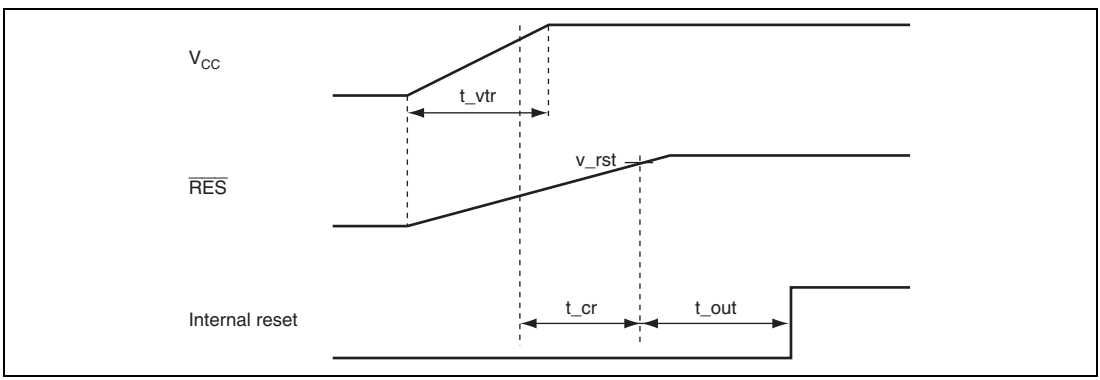


Figure 21.26 Power-On Reset Circuit Reset Timing

21.6 Output Load Circuit

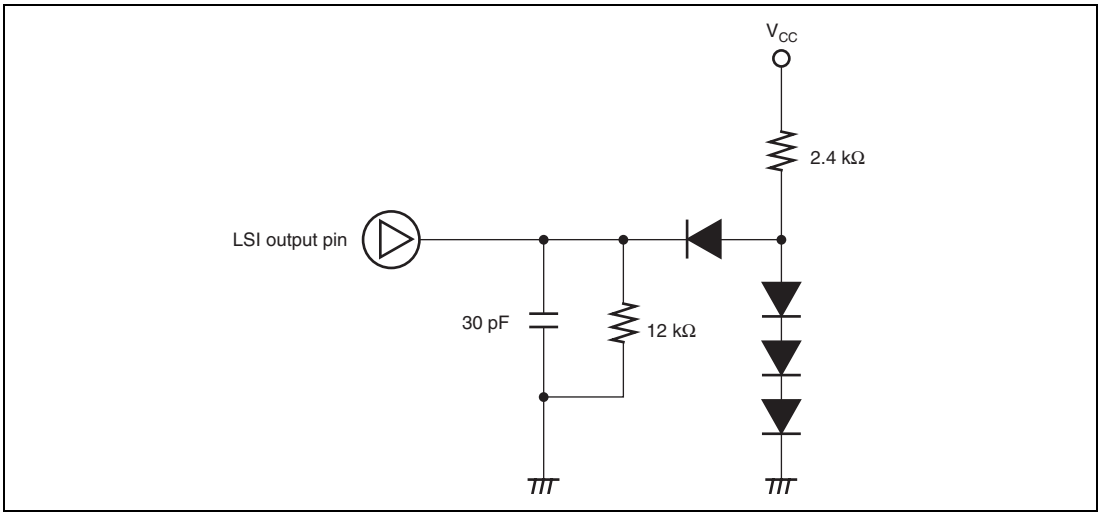


Figure 21.27 Output Load Condition

21.7 Recommended Resonators

(1) Recommended Crystal Resonators

Frequency (MHz)	Manufacturer	Part No.
4.194304	NIHON DEMPA KOGYO CO., LTD.	NR-18
10	NIHON DEMPA KOGYO CO., LTD.	NR-18

(2) Recommended Ceramic Resonators

Frequency (MHz)	Manufacturer	Part No.
2	Murata Manufacturing Co., Ltd.	CSTCC2M00G53-B0
		CSTCC2M00G56-B0
4.19	Murata Manufacturing Co., Ltd.	CSTLS4M19G53-B0
		CSTLS4M19G56-B0
10	Murata Manufacturing Co., Ltd.	CSTLS10M0G53-B0
		CSTLS10M0G56-B0

Figure 21.28 Recommended Resonators

21.8 Usage Note

The F-ZTAT and masked ROM versions satisfy the electrical characteristics shown in this manual, however actual electrical characteristic values, operating margins, noise margins, and other properties may vary due to differences in manufacturing process, on-chip ROM, layout patterns, and so on.

When system evaluation testing is carried out using the F-ZTAT version, the same evaluation testing should also be conducted for the masked ROM version when changing over to that version.

Appendix

A. Instruction Set

A.1 Instruction List

Condition Code

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
⊕	Logical exclusive 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 (cont)

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	Addressing Mode and Instruction Length (bytes)									Operation	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	MOV.B #xx:8, Rd	B	2									#xx:8 → Rd8	—	—	↕	↕	0	—	2	
	MOV.B Rs, Rd	B		2								Rs8 → Rd8	—	—	↕	↕	0	—	2	
	MOV.B @ERs, Rd	B			2							@ERs → Rd8	—	—	↕	↕	0	—	4	
	MOV.B @(d:16, ERs), Rd	B				4						@(d:16, ERs) → Rd8	—	—	↕	↕	0	—	6	
	MOV.B @(d:24, ERs), Rd	B				8						@(d:24, ERs) → Rd8	—	—	↕	↕	0	—	10	
	MOV.B @ERs+, Rd	B					2					@ERs → Rd8 ERs32+1 → ERs32	—	—	↕	↕	0	—	6	
	MOV.B @aa:8, Rd	B						2				@aa:8 → Rd8	—	—	↕	↕	0	—	4	
	MOV.B @aa:16, Rd	B						4				@aa:16 → Rd8	—	—	↕	↕	0	—	6	
	MOV.B @aa:24, Rd	B						6				@aa:24 → Rd8	—	—	↕	↕	0	—	8	
	MOV.B Rs, @ERd	B			2							Rs8 → @ERd	—	—	↕	↕	0	—	4	
	MOV.B Rs, @(d:16, ERd)	B				4						Rs8 → @(d:16, ERd)	—	—	↕	↕	0	—	6	
	MOV.B Rs, @(d:24, ERd)	B				8						Rs8 → @(d:24, ERd)	—	—	↕	↕	0	—	10	
	MOV.B Rs, @–ERd	B					2					ERd32–1 → ERd32 Rs8 → @ERd	—	—	↕	↕	0	—	6	
	MOV.B Rs, @aa:8	B						2				Rs8 → @aa:8	—	—	↕	↕	0	—	4	
	MOV.B Rs, @aa:16	B						4				Rs8 → @aa:16	—	—	↕	↕	0	—	6	
	MOV.B Rs, @aa:24	B						6				Rs8 → @aa:24	—	—	↕	↕	0	—	8	
	MOV.W #xx:16, Rd	W	4									#xx:16 → Rd16	—	—	↕	↕	0	—	4	
	MOV.W Rs, Rd	W		2								Rs16 → Rd16	—	—	↕	↕	0	—	2	
	MOV.W @ERs, Rd	W			2							@ERs → Rd16	—	—	↕	↕	0	—	4	
	MOV.W @(d:16, ERs), Rd	W				4						@(d:16, ERs) → Rd16	—	—	↕	↕	0	—	6	
	MOV.W @(d:24, ERs), Rd	W				8						@(d:24, ERs) → Rd16	—	—	↕	↕	0	—	10	
	MOV.W @ERs+, Rd	W					2					@ERs → Rd16 ERs32+2 → @ERd32	—	—	↕	↕	0	—	6	
	MOV.W @aa:16, Rd	W						4				@aa:16 → Rd16	—	—	↕	↕	0	—	6	
	MOV.W @aa:24, Rd	W						6				@aa:24 → Rd16	—	—	↕	↕	0	—	8	
	MOV.W Rs, @ERd	W			2							Rs16 → @ERd	—	—	↕	↕	0	—	4	
	MOV.W Rs, @(d:16, ERd)	W				4						Rs16 → @(d:16, ERd)	—	—	↕	↕	0	—	6	
	MOV.W Rs, @(d:24, ERd)	W				8						Rs16 → @(d:24, ERd)	—	—	↕	↕	0	—	10	

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	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	MOV.W Rs, @–ERd	W					2					ERd32–2 → ERd32 Rs16 → @ERd	—	—	↕	↕	0	—	6	
	MOV.W Rs, @aa:16	W						4				Rs16 → @aa:16	—	—	↕	↕	0	—	6	
	MOV.W Rs, @aa:24	W						6				Rs16 → @aa:24	—	—	↕	↕	0	—	8	
	MOV.L #xx:32, ERd	L	6									#xx:32 → ERd32	—	—	↕	↕	0	—	6	
	MOV.L ERs, ERd	L		2								ERs32 → ERd32	—	—	↕	↕	0	—	2	
	MOV.L @ERs, ERd	L			4							@ERs → ERd32	—	—	↕	↕	0	—	8	
	MOV.L @(d:16, ERs), ERd	L				6						@(d:16, ERs) → ERd32	—	—	↕	↕	0	—	10	
	MOV.L @(d:24, ERs), ERd	L				10						@(d:24, ERs) → ERd32	—	—	↕	↕	0	—	14	
	MOV.L @ERs+, ERd	L					4					@ERs → ERd32 ERs32+4 → ERs32	—	—	↕	↕	0	—	10	
	MOV.L @aa:16, ERd	L						6				@aa:16 → ERd32	—	—	↕	↕	0	—	10	
	MOV.L @aa:24, ERd	L						8				@aa:24 → ERd32	—	—	↕	↕	0	—	12	
	MOV.L ERs, @ERd	L			4							ERs32 → @ERd	—	—	↕	↕	0	—	8	
	MOV.L ERs, @(d:16, ERd)	L				6						ERs32 → @(d:16, ERd)	—	—	↕	↕	0	—	10	
	MOV.L ERs, @(d:24, ERd)	L				10						ERs32 → @(d:24, ERd)	—	—	↕	↕	0	—	14	
	MOV.L ERs, @–ERd	L					4					ERd32–4 → ERd32 ERs32 → @ERd	—	—	↕	↕	0	—	10	
	MOV.L ERs, @aa:16	L						6				ERs32 → @aa:16	—	—	↕	↕	0	—	10	
	MOV.L ERs, @aa:24	L						8				ERs32 → @aa:24	—	—	↕	↕	0	—	12	
POP	POP.W Rn	W									2	@SP → Rn16 SP+2 → SP	—	—	↕	↕	0	—	6	
	POP.L ERn	L									4	@SP → ERn32 SP+4 → SP	—	—	↕	↕	0	—	10	
PUSH	PUSH.W Rn	W									2	SP–2 → SP Rn16 → @SP	—	—	↕	↕	0	—	6	
	PUSH.L ERn	L									4	SP–4 → SP ERn32 → @SP	—	—	↕	↕	0	—	10	
MOVFPE	MOVFPE @aa:16, Rd	B						4				Cannot be used in this LSI	Cannot be used in this LSI							
MOVTPE	MOVTPE Rs, @aa:16	B						4				Cannot be used in this LSI	Cannot be used in this LSI							

2. Arithmetic Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	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	ADD.B #xx:8, Rd	B	2									Rd8+#xx:8 → Rd8	—	↕	↕	↕	↕	↕	2	
	ADD.B Rs, Rd	B		2								Rd8+Rs8 → Rd8	—	↕	↕	↕	↕	↕	2	
	ADD.W #xx:16, Rd	W	4									Rd16+#xx:16 → Rd16	—	(1)	↕	↕	↕	↕	4	
	ADD.W Rs, Rd	W		2								Rd16+Rs16 → Rd16	—	(1)	↕	↕	↕	↕	2	
	ADD.L #xx:32, ERd	L	6									ERd32+#xx:32 → ERd32	—	(2)	↕	↕	↕	↕	6	
	ADD.L ERs, ERd	L		2								ERd32+ERs32 → ERd32	—	(2)	↕	↕	↕	↕	2	
ADDX	ADDX.B #xx:8, Rd	B	2									Rd8+#xx:8 +C → Rd8	—	↕	↕	(3)	↕	↕	2	
	ADDX.B Rs, Rd	B		2								Rd8+Rs8 +C → Rd8	—	↕	↕	(3)	↕	↕	2	
ADDS	ADDS.L #1, ERd	L		2								ERd32+1 → ERd32	—	—	—	—	—	—	2	
	ADDS.L #2, ERd	L		2								ERd32+2 → ERd32	—	—	—	—	—	—	2	
	ADDS.L #4, ERd	L		2								ERd32+4 → ERd32	—	—	—	—	—	—	2	
INC	INC.B Rd	B		2								Rd8+1 → Rd8	—	—	↕	↕	↕	—	2	
	INC.W #1, Rd	W		2								Rd16+1 → Rd16	—	—	↕	↕	↕	—	2	
	INC.W #2, Rd	W		2								Rd16+2 → Rd16	—	—	↕	↕	↕	—	2	
	INC.L #1, ERd	L		2								ERd32+1 → ERd32	—	—	↕	↕	↕	—	2	
	INC.L #2, ERd	L		2								ERd32+2 → ERd32	—	—	↕	↕	↕	—	2	
DAA	DAA Rd	B		2								Rd8 decimal adjust → Rd8	—	*	↕	↕	*	↕	2	
SUB	SUB.B Rs, Rd	B		2								Rd8–Rs8 → Rd8	—	↕	↕	↕	↕	↕	2	
	SUB.W #xx:16, Rd	W	4									Rd16–#xx:16 → Rd16	—	(1)	↕	↕	↕	↕	4	
	SUB.W Rs, Rd	W		2								Rd16–Rs16 → Rd16	—	(1)	↕	↕	↕	↕	2	
	SUB.L #xx:32, ERd	L	6									ERd32–#xx:32 → ERd32	—	(2)	↕	↕	↕	↕	6	
	SUB.L ERs, ERd	L		2								ERd32–ERs32 → ERd32	—	(2)	↕	↕	↕	↕	2	
SUBX	SUBX.B #xx:8, Rd	B	2									Rd8–#xx:8–C → Rd8	—	↕	↕	(3)	↕	↕	2	
	SUBX.B Rs, Rd	B		2								Rd8–Rs8–C → Rd8	—	↕	↕	(3)	↕	↕	2	
SUBS	SUBS.L #1, ERd	L		2								ERd32–1 → ERd32	—	—	—	—	—	—	2	
	SUBS.L #2, ERd	L		2								ERd32–2 → ERd32	—	—	—	—	—	—	2	
	SUBS.L #4, ERd	L		2								ERd32–4 → ERd32	—	—	—	—	—	—	2	
DEC	DEC.B Rd	B		2								Rd8–1 → Rd8	—	—	↕	↕	↕	—	2	
	DEC.W #1, Rd	W		2								Rd16–1 → Rd16	—	—	↕	↕	↕	—	2	
	DEC.W #2, Rd	W		2								Rd16–2 → Rd16	—	—	↕	↕	↕	—	2	

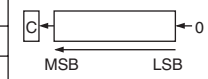
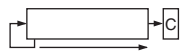
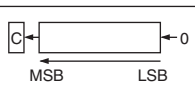
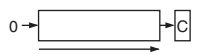
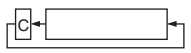
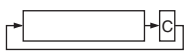
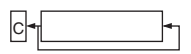
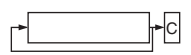
Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
DEC	DEC.L #1, ERd	L		2								ERd32–1 → ERd32	—	—	↕	↕	↕	—	2	
	DEC.L #2, ERd	L		2								ERd32–2 → ERd32	—	—	↕	↕	↕	—	2	
DAS	DAS.Rd	B		2								Rd8 decimal adjust → Rd8	—	*	↕	↕	*	—	2	
MULXU	MULXU. B Rs, Rd	B		2								Rd8 × Rs8 → Rd16 (unsigned multiplication)	—	—	—	—	—	—	14	
	MULXU. W Rs, ERd	W		2								Rd16 × Rs16 → ERd32 (unsigned multiplication)	—	—	—	—	—	—	22	
MULXS	MULXS. B Rs, Rd	B		4								Rd8 × Rs8 → Rd16 (signed multiplication)	—	—	↕	↕	—	—	16	
	MULXS. W Rs, ERd	W		4								Rd16 × Rs16 → ERd32 (signed multiplication)	—	—	↕	↕	—	—	24	
DIVXU	DIVXU. B Rs, Rd	B		2								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	(6)	(7)	—	—	14	
	DIVXU. W Rs, ERd	W		2								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	(6)	(7)	—	—	22	
DIVXS	DIVXS. B Rs, Rd	B		4								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	(8)	(7)	—	—	16	
	DIVXS. W Rs, ERd	W		4								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	(8)	(7)	—	—	24	
CMP	CMP.B #xx:8, Rd	B	2									Rd8–#xx:8	—	↕	↕	↕	↕	↕	2	
	CMP.B Rs, Rd	B		2								Rd8–Rs8	—	↕	↕	↕	↕	↕	2	
	CMP.W #xx:16, Rd	W	4									Rd16–#xx:16	—	(1)	↕	↕	↕	↕	4	
	CMP.W Rs, Rd	W		2								Rd16–Rs16	—	(1)	↕	↕	↕	↕	2	
	CMP.L #xx:32, ERd	L	6									ERd32–#xx:32	—	(2)	↕	↕	↕	↕	4	
	CMP.L ERs, ERd	L		2								ERd32–ERs32	—	(2)	↕	↕	↕	↕	2	

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	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	NEG.B Rd	B		2								0–Rd8 → Rd8	—	↕	↕	↕	↕	↕	2	
	NEG.W Rd	W		2								0–Rd16 → Rd16	—	↕	↕	↕	↕	↕	2	
	NEG.L ERd	L		2								0–ERd32 → ERd32	—	↕	↕	↕	↕	↕	2	
EXTU	EXTU.W Rd	W		2								0 → (<bits 15 to 8> of Rd16)	—	—	0	↕	0	—	2	
	EXTU.L ERd	L		2								0 → (<bits 31 to 16> of ERd32)	—	—	0	↕	0	—	2	
EXTS	EXTS.W Rd	W		2								(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)	—	—	↕	↕	0	—	2	
	EXTS.L ERd	L		2								(<bit 15> of ERd32) → (<bits 31 to 16> of ERd32)	—	—	↕	↕	0	—	2	

3. Logic Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	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	AND.B #xx:8, Rd	B	2									Rd8∧#xx:8 → Rd8	—	—	↕	↕	0	—	2	
	AND.B Rs, Rd	B		2								Rd8∧Rs8 → Rd8	—	—	↕	↕	0	—	2	
	AND.W #xx:16, Rd	W	4									Rd16∧#xx:16 → Rd16	—	—	↕	↕	0	—	4	
	AND.W Rs, Rd	W		2								Rd16∧Rs16 → Rd16	—	—	↕	↕	0	—	2	
	AND.L #xx:32, ERd	L	6									ERd32∧#xx:32 → ERd32	—	—	↕	↕	0	—	6	
	AND.L ERs, ERd	L		4								ERd32∧ERs32 → ERd32	—	—	↕	↕	0	—	4	
OR	OR.B #xx:8, Rd	B	2									Rd8∨#xx:8 → Rd8	—	—	↕	↕	0	—	2	
	OR.B Rs, Rd	B		2								Rd8∨Rs8 → Rd8	—	—	↕	↕	0	—	2	
	OR.W #xx:16, Rd	W	4									Rd16∨#xx:16 → Rd16	—	—	↕	↕	0	—	4	
	OR.W Rs, Rd	W		2								Rd16∨Rs16 → Rd16	—	—	↕	↕	0	—	2	
	OR.L #xx:32, ERd	L	6									ERd32∨#xx:32 → ERd32	—	—	↕	↕	0	—	6	
	OR.L ERs, ERd	L		4								ERd32∨ERs32 → ERd32	—	—	↕	↕	0	—	4	
XOR	XOR.B #xx:8, Rd	B	2									Rd8⊕#xx:8 → Rd8	—	—	↕	↕	0	—	2	
	XOR.B Rs, Rd	B		2								Rd8⊕Rs8 → Rd8	—	—	↕	↕	0	—	2	
	XOR.W #xx:16, Rd	W	4									Rd16⊕#xx:16 → Rd16	—	—	↕	↕	0	—	4	
	XOR.W Rs, Rd	W		2								Rd16⊕Rs16 → Rd16	—	—	↕	↕	0	—	2	
	XOR.L #xx:32, ERd	L	6									ERd32⊕#xx:32 → ERd32	—	—	↕	↕	0	—	6	
	XOR.L ERs, ERd	L		4								ERd32⊕ERs32 → ERd32	—	—	↕	↕	0	—	4	
NOT	NOT.B Rd	B		2								¬ Rd8 → Rd8	—	—	↕	↕	0	—	2	
	NOT.W Rd	W		2								¬ Rd16 → Rd16	—	—	↕	↕	0	—	2	
	NOT.L ERd	L		2								¬ Rd32 → Rd32	—	—	↕	↕	0	—	2	

4. Shift Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	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	SHAL.B Rd	B		2									—	—	↕	↕	↕	↕	2	
	SHAL.W Rd	W		2									—	—	↕	↕	↕	↕	2	
	SHAL.L ERd	L		2									—	—	↕	↕	↕	↕	2	
SHAR	SHAR.B Rd	B		2									—	—	↕	↕	0	↕	2	
	SHAR.W Rd	W		2									—	—	↕	↕	0	↕	2	
	SHAR.L ERd	L		2									—	—	↕	↕	0	↕	2	
SHLL	SHLL.B Rd	B		2									—	—	↕	↕	0	↕	2	
	SHLL.W Rd	W		2									—	—	↕	↕	0	↕	2	
	SHLL.L ERd	L		2									—	—	↕	↕	0	↕	2	
SHLR	SHLR.B Rd	B		2									—	—	↕	↕	0	↕	2	
	SHLR.W Rd	W		2									—	—	↕	↕	0	↕	2	
	SHLR.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTXL	ROTXL.B Rd	B		2									—	—	↕	↕	0	↕	2	
	ROTXL.W Rd	W		2									—	—	↕	↕	0	↕	2	
	ROTXL.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTXR	ROTXR.B Rd	B		2									—	—	↕	↕	0	↕	2	
	ROTXR.W Rd	W		2									—	—	↕	↕	0	↕	2	
	ROTXR.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTL	ROTL.B Rd	B		2									—	—	↕	↕	0	↕	2	
	ROTL.W Rd	W		2									—	—	↕	↕	0	↕	2	
	ROTL.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTR	ROTR.B Rd	B		2									—	—	↕	↕	0	↕	2	
	ROTR.W Rd	W		2									—	—	↕	↕	0	↕	2	
	ROTR.L ERd	L		2									—	—	↕	↕	0	↕	2	

5. Bit-Manipulation Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	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	BSET #xx:3, Rd	B		2								(#xx:3 of Rd8) ← 1	—	—	—	—	—	—	2	
	BSET #xx:3, @ERd	B			4							(#xx:3 of @ERd) ← 1	—	—	—	—	—	—	8	
	BSET #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) ← 1	—	—	—	—	—	—	8	
	BSET Rn, Rd	B		2								(Rn8 of Rd8) ← 1	—	—	—	—	—	—	2	
	BSET Rn, @ERd	B			4							(Rn8 of @ERd) ← 1	—	—	—	—	—	—	8	
	BSET Rn, @aa:8	B						4				(Rn8 of @aa:8) ← 1	—	—	—	—	—	—	8	
BCLR	BCLR #xx:3, Rd	B		2								(#xx:3 of Rd8) ← 0	—	—	—	—	—	—	2	
	BCLR #xx:3, @ERd	B			4							(#xx:3 of @ERd) ← 0	—	—	—	—	—	—	8	
	BCLR #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) ← 0	—	—	—	—	—	—	8	
	BCLR Rn, Rd	B		2								(Rn8 of Rd8) ← 0	—	—	—	—	—	—	2	
	BCLR Rn, @ERd	B			4							(Rn8 of @ERd) ← 0	—	—	—	—	—	—	8	
	BCLR Rn, @aa:8	B						4				(Rn8 of @aa:8) ← 0	—	—	—	—	—	—	8	
BNOT	BNOT #xx:3, Rd	B		2								(#xx:3 of Rd8) ← ¬ (#xx:3 of Rd8)	—	—	—	—	—	—	2	
	BNOT #xx:3, @ERd	B			4							(#xx:3 of @ERd) ← ¬ (#xx:3 of @ERd)	—	—	—	—	—	—	8	
	BNOT #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) ← ¬ (#xx:3 of @aa:8)	—	—	—	—	—	—	8	
	BNOT Rn, Rd	B		2								(Rn8 of Rd8) ← ¬ (Rn8 of Rd8)	—	—	—	—	—	—	2	
	BNOT Rn, @ERd	B			4							(Rn8 of @ERd) ← ¬ (Rn8 of @ERd)	—	—	—	—	—	—	8	
	BNOT Rn, @aa:8	B						4				(Rn8 of @aa:8) ← ¬ (Rn8 of @aa:8)	—	—	—	—	—	—	8	
BTST	BTST #xx:3, Rd	B		2								¬ (#xx:3 of Rd8) → Z	—	—	—	↕	—	—	2	
	BTST #xx:3, @ERd	B			4							¬ (#xx:3 of @ERd) → Z	—	—	—	↕	—	—	6	
	BTST #xx:3, @aa:8	B						4				¬ (#xx:3 of @aa:8) → Z	—	—	—	↕	—	—	6	
	BTST Rn, Rd	B		2								¬ (Rn8 of @Rd8) → Z	—	—	—	↕	—	—	2	
	BTST Rn, @ERd	B			4							¬ (Rn8 of @ERd) → Z	—	—	—	↕	—	—	6	
	BTST Rn, @aa:8	B						4				¬ (Rn8 of @aa:8) → Z	—	—	—	↕	—	—	6	
BLD	BLD #xx:3, Rd	B		2								(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	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	BLD #xx:3, @ERd	B			4							(#xx:3 of @ERd) → C	—	—	—	—	—	↕	6	
	BLD #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BILD	BILD #xx:3, Rd	B		2								¬ (#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BILD #xx:3, @ERd	B			4							¬ (#xx:3 of @ERd) → C	—	—	—	—	—	↕	6	
	BILD #xx:3, @aa:8	B						4				¬ (#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BST	BST #xx:3, Rd	B		2								C → (#xx:3 of Rd8)	—	—	—	—	—	—	2	
	BST #xx:3, @ERd	B			4							C → (#xx:3 of @ERd24)	—	—	—	—	—	—	8	
	BST #xx:3, @aa:8	B						4				C → (#xx:3 of @aa:8)	—	—	—	—	—	—	8	
BIST	BIST #xx:3, Rd	B		2								¬ C → (#xx:3 of Rd8)	—	—	—	—	—	—	2	
	BIST #xx:3, @ERd	B			4							¬ C → (#xx:3 of @ERd24)	—	—	—	—	—	—	8	
	BIST #xx:3, @aa:8	B						4				¬ C → (#xx:3 of @aa:8)	—	—	—	—	—	—	8	
BAND	BAND #xx:3, Rd	B		2								C∧(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BAND #xx:3, @ERd	B			4							C∧(#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BAND #xx:3, @aa:8	B						4				C∧(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BIAND	BIAND #xx:3, Rd	B		2								C∧ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BIAND #xx:3, @ERd	B			4							C∧ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BIAND #xx:3, @aa:8	B						4				C∧ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BOR	BOR #xx:3, Rd	B		2								C∨(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BOR #xx:3, @ERd	B			4							C∨(#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BOR #xx:3, @aa:8	B						4				C∨(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BIOR	BIOR #xx:3, Rd	B		2								C∨ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BIOR #xx:3, @ERd	B			4							C∨ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BIOR #xx:3, @aa:8	B						4				C∨ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BXOR	BXOR #xx:3, Rd	B		2								C⊕(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BXOR #xx:3, @ERd	B			4							C⊕(#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BXOR #xx:3, @aa:8	B						4				C⊕(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BIXOR	BIXOR #xx:3, Rd	B		2								C⊕ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BIXOR #xx:3, @ERd	B			4							C⊕ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BIXOR #xx:3, @aa:8	B						4				C⊕ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	

6. Branching Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|	Branch Condition	I	H	N	Z	V	C	Normal	Advanced
Bcc	BRA d:8 (BT d:8)	—							2			If condition is true then PC ← PC+d else next; Always	—	—	—	—	—	—	4	
	BRA d:16 (BT d:16)	—							4				—	—	—	—	—	—	6	
	BRN d:8 (BF d:8)	—							2			Never	—	—	—	—	—	—	4	
	BRN d:16 (BF d:16)	—							4				—	—	—	—	—	—	6	
	BHI d:8	—							2			C∨ Z = 0	—	—	—	—	—	—	4	
	BHI d:16	—							4				—	—	—	—	—	—	6	
	BLS d:8	—							2			C∨ Z = 1	—	—	—	—	—	—	4	
	BLS d:16	—							4				—	—	—	—	—	—	6	
	BCC d:8 (BHS d:8)	—							2			C = 0	—	—	—	—	—	—	4	
	BCC d:16 (BHS d:16)	—							4				—	—	—	—	—	—	6	
	BCS d:8 (BLO d:8)	—							2			C = 1	—	—	—	—	—	—	4	
	BCS d:16 (BLO d:16)	—							4				—	—	—	—	—	—	6	
	BNE d:8	—							2			Z = 0	—	—	—	—	—	—	4	
	BNE d:16	—							4				—	—	—	—	—	—	6	
	BEQ d:8	—							2			Z = 1	—	—	—	—	—	—	4	
	BEQ d:16	—							4				—	—	—	—	—	—	6	
	BVC d:8	—							2			V = 0	—	—	—	—	—	—	4	
	BVC d:16	—							4				—	—	—	—	—	—	6	
	BVS d:8	—							2			V = 1	—	—	—	—	—	—	4	
	BVS d:16	—							4				—	—	—	—	—	—	6	
	BPL d:8	—							2			N = 0	—	—	—	—	—	—	4	
	BPL d:16	—							4				—	—	—	—	—	—	6	
	BMI d:8	—							2			N = 1	—	—	—	—	—	—	4	
	BMI d:16	—							4				—	—	—	—	—	—	6	
	BGE d:8	—							2			N⊕V = 0	—	—	—	—	—	—	4	
	BGE d:16	—							4				—	—	—	—	—	—	6	
	BLT d:8	—							2			N⊕V = 1	—	—	—	—	—	—	4	
	BLT d:16	—							4				—	—	—	—	—	—	6	
	BGT d:8	—							2			Z∨ (N⊕V) = 0	—	—	—	—	—	—	4	
	BGT d:16	—							4				—	—	—	—	—	—	6	
	BLE d:8	—							2			Z∨ (N⊕V) = 1	—	—	—	—	—	—	4	
	BLE d:16	—							4				—	—	—	—	—	—	6	

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	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	JMP @ERn	—			2							PC ← ERn	—	—	—	—	—	—	4	
	JMP @aa:24	—						4				PC ← aa:24	—	—	—	—	—	—	6	
	JMP @@aa:8	—								2		PC ← @aa:8	—	—	—	—	—	—	8	10
BSR	BSR d:8	—							2			PC → @–SP PC ← PC+d:8	—	—	—	—	—	—	6	8
	BSR d:16	—							4			PC → @–SP PC ← PC+d:16	—	—	—	—	—	—	8	10
JSR	JSR @ERn	—			2							PC → @–SP PC ← ERn	—	—	—	—	—	—	6	8
	JSR @aa:24	—						4				PC → @–SP PC ← aa:24	—	—	—	—	—	—	8	10
	JSR @@aa:8	—								2		PC → @–SP PC ← @aa:8	—	—	—	—	—	—	8	12
RTS	RTS	—									2	PC ← @SP+	—	—	—	—	—	—	8	10

7. System Control Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	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	TRAPA #x:2	—									2	PC → @-SP CCR → @-SP <vector> → PC	1	—	—	—	—	—	14	16
RTE	RTE	—										CCR ← @SP+ PC ← @SP+	↕	↕	↕	↕	↕	↕	10	
SLEEP	SLEEP	—										Transition to power-down state	—	—	—	—	—	—	2	
LDC	LDC #xx:8, CCR	B	2									#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
	LDC Rs, CCR	B		2								Rs8 → CCR	↕	↕	↕	↕	↕	↕	2	
	LDC @ERs, CCR	W			4							@ERs → CCR	↕	↕	↕	↕	↕	↕	6	
	LDC @(d:16, ERs), CCR	W				6						@(d:16, ERs) → CCR	↕	↕	↕	↕	↕	↕	8	
	LDC @(d:24, ERs), CCR	W				10						@(d:24, ERs) → CCR	↕	↕	↕	↕	↕	↕	12	
	LDC @ERs+, CCR	W					4					@ERs → CCR ERs32+2 → ERs32	↕	↕	↕	↕	↕	↕	8	
	LDC @aa:16, CCR	W						6				@aa:16 → CCR	↕	↕	↕	↕	↕	↕	8	
	LDC @aa:24, CCR	W						8				@aa:24 → CCR	↕	↕	↕	↕	↕	↕	10	
STC	STC CCR, Rd	B		2								CCR → Rd8	—	—	—	—	—	—	2	
	STC CCR, @ERd	W			4							CCR → @ERd	—	—	—	—	—	—	6	
	STC CCR, @(d:16, ERd)	W				6						CCR → @(d:16, ERd)	—	—	—	—	—	—	8	
	STC CCR, @(d:24, ERd)	W				10						CCR → @(d:24, ERd)	—	—	—	—	—	—	12	
	STC CCR, @–ERd	W					4					ERd32–2 → ERd32 CCR → @ERd	—	—	—	—	—	—	8	
	STC CCR, @aa:16	W						6				CCR → @aa:16	—	—	—	—	—	—	8	
	STC CCR, @aa:24	W						8				CCR → @aa:24	—	—	—	—	—	—	10	
ANDC	ANDC #xx:8, CCR	B	2									CCR∧#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
ORC	ORC #xx:8, CCR	B	2									CCR∨#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
XORC	XORC #xx:8, CCR	B	2									CCR⊕#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
NOP	NOP	—									2	PC ← PC+2	—	—	—	—	—	—	2	

8. Block Data Transfer Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	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	EEPMOV. B	—									4	if R4L ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4L–1 → R4L until R4L=0 else next	—	—	—	—	—	—	8+4n*2	
	EEPMOV. W	—									4	if R4 ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4–1 → R4 until R4=0 else next	—	—	—	—	—	—	8+4n*2	

Notes: 1. The number of states in cases where the instruction code and its operands are located in on-chip memory is shown here. For other cases, see appendix A.3, Number of Execution States.

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.

A.2 Operation Code Map

Table A.2 Operation Code Map (1)

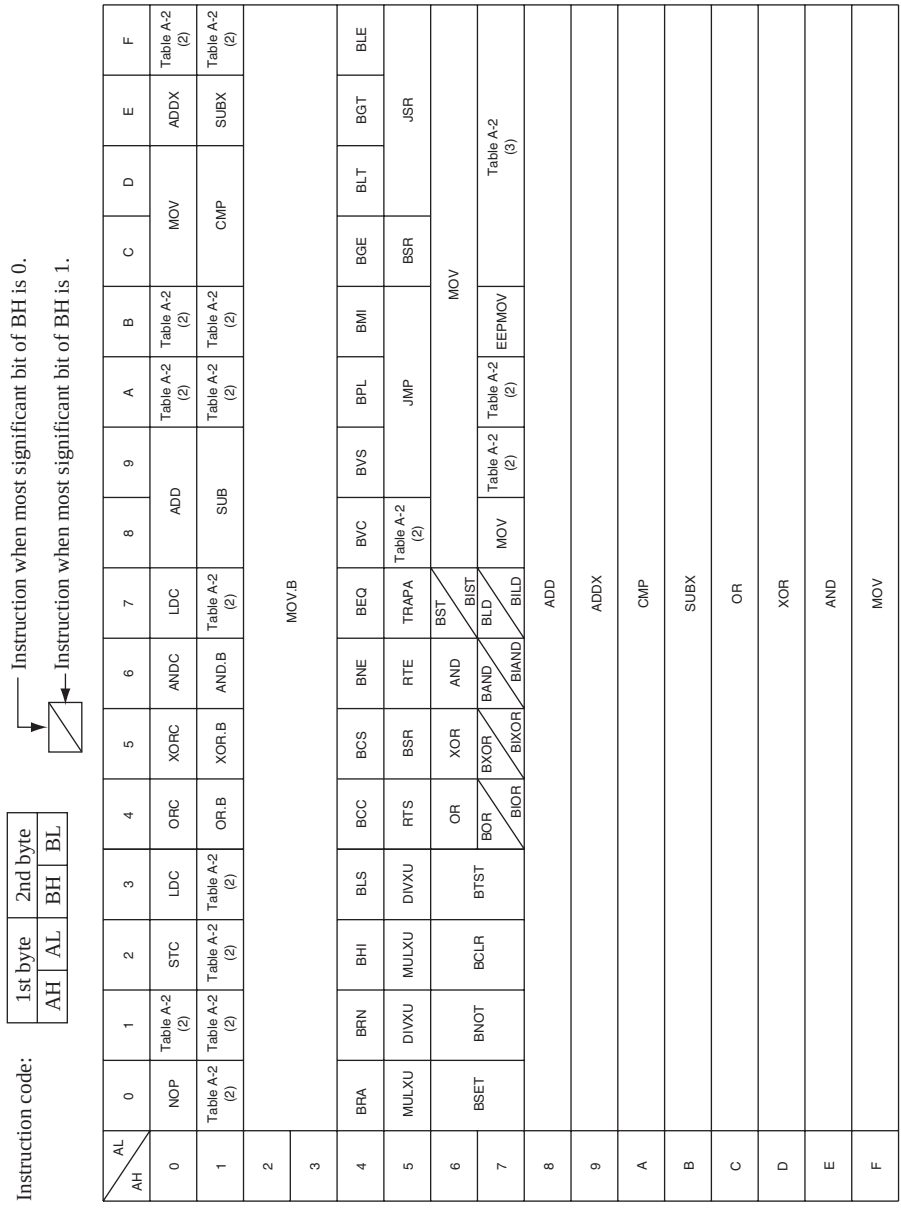


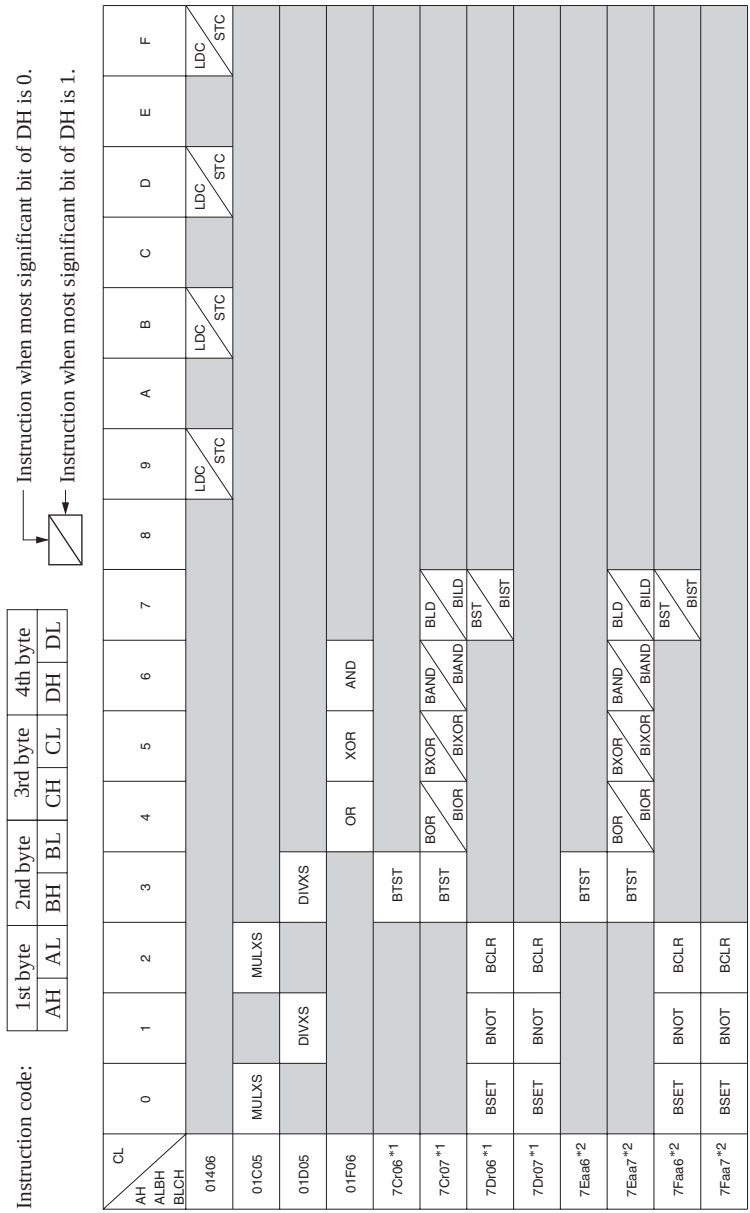
Table A.2 Operation Code Map (2)

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	SUBS					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									

Table A.2 Operation Code Map (3)



Notes: 1. r is the register designation field.
2. aa is the absolute address field.

A.3 Number of Execution States

The status of execution for each instruction of the H8/300H CPU and the method of calculating the number of states required for instruction execution are shown below. Table A.4 shows the number of cycles of each type occurring in each instruction, such as instruction fetch and data read/write. Table A.3 shows the number of states required for each cycle. The total number of states required for execution of an instruction can be calculated by the following expression:

$$\text{Execution states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

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 peripheral module is accessed. See section 20.1, Register Addresses (Address Order).

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 #xx:16, Rd	2					
	ADD.W Rs, Rd	1					
	ADD.L #xx:32, ERd	3					
	ADD.L ERs, ERd	1					
ADDS	ADDS #1/2/4, ERd	1					
ADDX	ADDX #xx:8, Rd	1					
	ADDX Rs, Rd	1					
AND	AND.B #xx:8, Rd	1					
	AND.B Rs, Rd	1					
	AND.W #xx:16, Rd	2					
	AND.W Rs, Rd	1					
	AND.L #xx:32, ERd	3					
	AND.L ERs, ERd	2					
ANDC	ANDC #xx:8, CCR	1					
BAND	BAND #xx:3, Rd	1					
	BAND #xx:3, @ERd	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					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
Bcc	BLT d:8	2					
	BGT d:8	2					
	BLE d:8	2					
	BRA d:16(BT d:16)	2					2
	BRN d:16(BF d:16)	2					2
	BHI d:16	2					2
	BLS d:16	2					2
	BCC d:16(BHS d:16)	2					2
	BCS d:16(BLO d:16)	2					2
	BNE d:16	2					2
	BEQ d:16	2					2
	BVC d:16	2					2
	BVS d:16	2					2
	BPL d:16	2					2
	BMI d:16	2					2
	BGE d:16	2					2
	BLT d:16	2					2
	BGT d:16	2					2
	BLE d:16	2					2
BCLR	BCLR #xx:3, Rd	1					
	BCLR #xx:3, @ERd	2			2		
	BCLR #xx:3, @aa:8	2			2		
	BCLR Rn, Rd	1					
	BCLR Rn, @ERd	2			2		
	BCLR Rn, @aa:8	2			2		
BIAND	BIAND #xx:3, Rd	1					
	BIAND #xx:3, @ERd	2			1		
	BIAND #xx:3, @aa:8	2			1		
BILD	BILD #xx:3, Rd	1					
	BILD #xx:3, @ERd	2			1		
	BILD #xx:3, @aa:8	2			1		

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BIOR	BIOR #xx:3, Rd	1					
	BIOR #xx:3, @ERd	2			1		
	BIOR #xx:3, @aa:8	2			1		
BIST	BIST #xx:3, Rd	1					
	BIST #xx:3, @ERd	2			2		
	BIST #xx:3, @aa:8	2			2		
BIXOR	BIXOR #xx:3, Rd	1					
	BIXOR #xx:3, @ERd	2			1		
	BIXOR #xx:3, @aa:8	2			1		
BLD	BLD #xx:3, Rd	1					
	BLD #xx:3, @ERd	2			1		
	BLD #xx:3, @aa:8	2			1		
BNOT	BNOT #xx:3, Rd	1					
	BNOT #xx:3, @ERd	2			2		
	BNOT #xx:3, @aa:8	2			2		
	BNOT Rn, Rd	1					
	BNOT Rn, @ERd	2			2		
	BNOT Rn, @aa:8	2			2		
BOR	BOR #xx:3, Rd	1					
	BOR #xx:3, @ERd	2			1		
	BOR #xx:3, @aa:8	2			1		
BSET	BSET #xx:3, Rd	1					
	BSET #xx:3, @ERd	2			2		
	BSET #xx:3, @aa:8	2			2		
	BSET Rn, Rd	1					
	BSET Rn, @ERd	2			2		
	BSET Rn, @aa:8	2			2		
BSR	BSR d:8	2		1			
	BSR d:16	2		1			2
BST	BST #xx:3, Rd	1					
	BST #xx:3, @ERd	2			2		
	BST #xx:3, @aa:8	2			2		

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BTST	BTST #xx:3, Rd	1					
	BTST #xx:3, @ERd	2			1		
	BTST #xx:3, @aa:8	2			1		
	BTST Rn, Rd	1					
	BTST Rn, @ERd	2			1		
	BTST Rn, @aa:8	2			1		
BXOR	BXOR #xx:3, Rd	1					
	BXOR #xx:3, @ERd	2			1		
	BXOR #xx:3, @aa:8	2			1		
CMP	CMP.B #xx:8, Rd	1					
	CMP.B Rs, Rd	1					
	CMP.W #xx:16, Rd	2					
	CMP.W Rs, Rd	1					
	CMP.L #xx:32, ERd	3					
	CMP.L ERs, ERd	1					
DAA	DAA Rd	1					
DAS	DAS Rd	1					
DEC	DEC.B Rd	1					
	DEC.W #1/2, Rd	1					
	DEC.L #1/2, ERd	1					
DIVXS	DIVXS.B Rs, Rd	2					12
	DIVXS.W Rs, ERd	2					20
DIVXU	DIVXU.B Rs, Rd	1					12
	DIVXU.W Rs, ERd	1					20
EEPMOV	EEPMOV.B	2			2n+2*[1]		
	EEPMOV.W	2			2n+2*[1]		
EXTS	EXTS.W Rd	1					
	EXTS.L ERd	1					
EXTU	EXTU.W Rd	1					
	EXTU.L ERd	1					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
INC	INC.B Rd	1					
	INC.W #1/2, Rd	1					
	INC.L #1/2, ERd	1					
JMP	JMP @ERn	2					
	JMP @aa:24	2					2
	JMP @@aa:8	2	1				2
JSR	JSR @ERn	2		1			
	JSR @aa:24	2		1			2
	JSR @@aa:8	2	1	1			
LDC	LDC #xx:8, CCR	1					
	LDC Rs, CCR	1					
	LDC@ERs, CCR	2				1	
	LDC@(d:16, ERs), CCR	3				1	
	LDC@(d:24,ERs), CCR	5				1	
	LDC@ERs+, CCR	2				1	2
	LDC@aa:16, CCR	3				1	
	LDC@aa:24, CCR	4				1	
MOV	MOV.B #xx:8, Rd	1					
	MOV.B Rs, Rd	1					
	MOV.B @ERs, Rd	1			1		
	MOV.B @(d:16, ERs), Rd	2			1		
	MOV.B @(d:24, ERs), Rd	4			1		
	MOV.B @ERs+, Rd	1			1		2
	MOV.B @aa:8, Rd	1			1		
	MOV.B @aa:16, Rd	2			1		
	MOV.B @aa:24, Rd	3			1		
	MOV.B Rs, @Erd	1			1		
	MOV.B Rs, @(d:16, ERd)	2			1		
	MOV.B Rs, @(d:24, ERd)	4			1		
	MOV.B Rs, @-ERd	1			1		2
	MOV.B Rs, @aa:8	1			1		

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MOV	MOV.B Rs, @aa:16	2			1		
	MOV.B Rs, @aa:24	3			1		
	MOV.W #xx:16, Rd	2					
	MOV.W Rs, Rd	1					
	MOV.W @ERs, Rd	1				1	
	MOV.W @(d:16,ERs), Rd	2				1	
	MOV.W @(d:24,ERs), Rd	4				1	
	MOV.W @ERs+, Rd	1				1	2
	MOV.W @aa:16, Rd	2				1	
	MOV.W @aa:24, Rd	3				1	
	MOV.W Rs, @ERd	1				1	
	MOV.W Rs, @(d:16,ERd)	2				1	
	MOV.W Rs, @(d:24,ERd)	4				1	
	MOV.W Rs, @-ERd	1				1	2
	MOV.W Rs, @aa:16	2				1	
	MOV.W Rs, @aa:24	3				1	
	MOV.L #xx:32, ERd	3					
	MOV.L ERs, ERd	1					
	MOV.L @ERs, ERd	2				2	
	MOV.L @(d:16,ERs), ERd	3				2	
	MOV.L @(d:24,ERs), ERd	5				2	
	MOV.L @ERs+, ERd	2				2	2
	MOV.L @aa:16, ERd	3				2	
	MOV.L @aa:24, ERd	4				2	
	MOV.L ERs,@ERd	2				2	
	MOV.L ERs, @(d:16,ERd)	3				2	
	MOV.L ERs, @(d:24,ERd)	5				2	
	MOV.L ERs, @-ERd	2				2	2
	MOV.L ERs, @aa:16	3				2	
	MOV.L ERs, @aa:24	4				2	
MOVFPE	MOVFPE @aa:16, Rd*[2]	2			1		
MOVTPE	MOVTPE Rs,@aa:16*[2]	2			1		

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MULXS	MULXS.B Rs, Rd	2					12
	MULXS.W Rs, ERd	2					20
MULXU	MULXU.B Rs, Rd	1					12
	MULXU.W Rs, ERd	1					20
NEG	NEG.B Rd	1					
	NEG.W Rd	1					
	NEG.L ERd	1					
NOP	NOP	1					
NOT	NOT.B Rd	1					
	NOT.W Rd	1					
	NOT.L ERd	1					
OR	OR.B #xx:8, Rd	1					
	OR.B Rs, Rd	1					
	OR.W #xx:16, Rd	2					
	OR.W Rs, Rd	1					
	OR.L #xx:32, ERd	3					
	OR.L ERs, ERd	2					
ORC	ORC #xx:8, CCR	1					
POP	POP.W Rn	1				1	2
	POP.L ERn	2				2	2
PUSH	PUSH.W Rn	1				1	2
	PUSH.L ERn	2				2	2
ROTL	ROTL.B Rd	1					
	ROTL.W Rd	1					
	ROTL.L ERd	1					
ROTR	ROTR.B Rd	1					
	ROTR.W Rd	1					
	ROTR.L ERd	1					
ROTXL	ROTXL.B Rd	1					
	ROTXL.W Rd	1					
	ROTXL.L ERd	1					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ROTXR	ROTXR.B Rd	1					
	ROTXR.W Rd	1					
	ROTXR.L ERd	1					
RTE	RTE	2		2			2
RTS	RTS	2		1			2
SHAL	SHAL.B Rd	1					
	SHAL.W Rd	1					
	SHAL.L ERd	1					
SHAR	SHAR.B Rd	1					
	SHAR.W Rd	1					
	SHAR.L ERd	1					
SHLL	SHLL.B Rd	1					
	SHLL.W Rd	1					
	SHLL.L ERd	1					
SHLR	SHLR.B Rd	1					
	SHLR.W Rd	1					
	SHLR.L ERd	1					
SLEEP	SLEEP	1					
STC	STC CCR, Rd	1					
	STC CCR, @ERd	2				1	
	STC CCR, @(d:16,ERd)	3				1	
	STC CCR, @(d:24,ERd)	5				1	
	STC CCR,@-ERd	2				1	2
	STC CCR, @aa:16	3				1	
	STC CCR, @aa:24	4				1	
SUB	SUB.B Rs, Rd	1					
	SUB.W #xx:16, Rd	2					
	SUB.W Rs, Rd	1					
	SUB.L #xx:32, ERd	3					
	SUB.L ERs, ERd	1					
SUBS	SUBS #1/2/4, ERd	1					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
SUBX	SUBX #xx:8, Rd	1					
	SUBX. Rs, Rd	1					
TRAPA	TRAPA #xx:2	2	1	2			
XOR	XOR.B #xx:8, Rd	1					
	XOR.B Rs, Rd	1					
	XOR.W #xx:16, Rd	2					
	XOR.W Rs, Rd	1					
	XOR.L #xx:32, ERd	3					
	XOR.L ERs, ERd	2					
XORC	XORC #xx:8, CCR	1					

Notes: 1. n: Specified value in R4L and R4. The source and destination operands are accessed n+1 times respectively.
2. It cannot be used in this LSI.

A.4 Combinations of Instructions and Addressing Modes

Table A.5 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

B. I/O Ports

B.1 I/O Port Block Diagrams

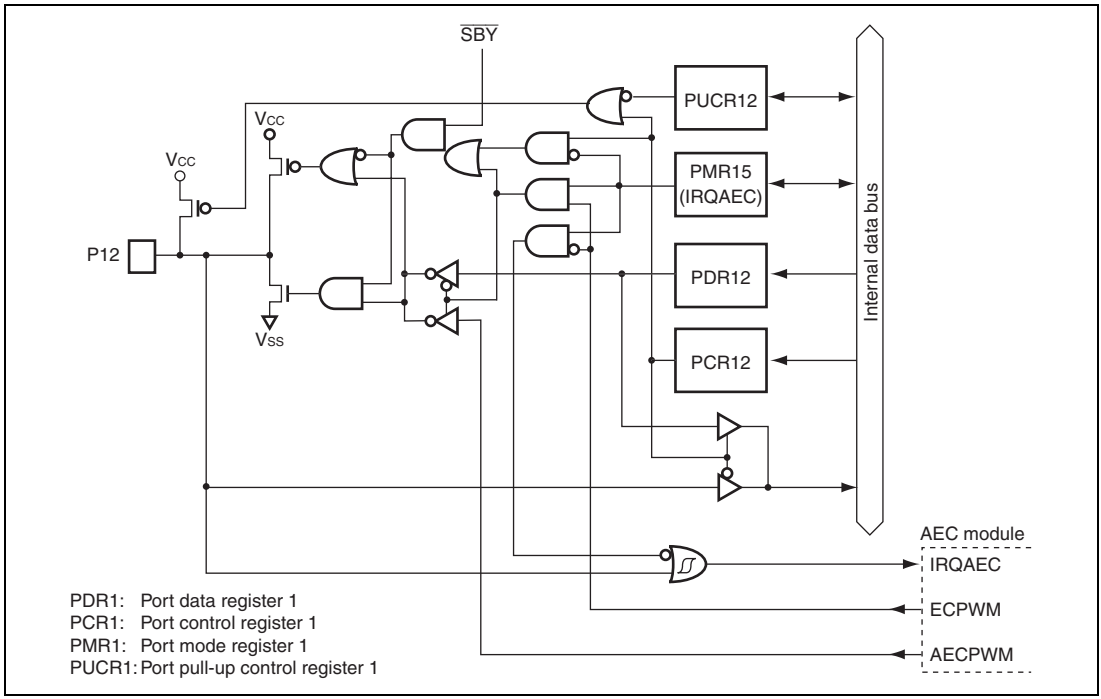


Figure B.1 (a) Port 1 Block Diagram (P12)

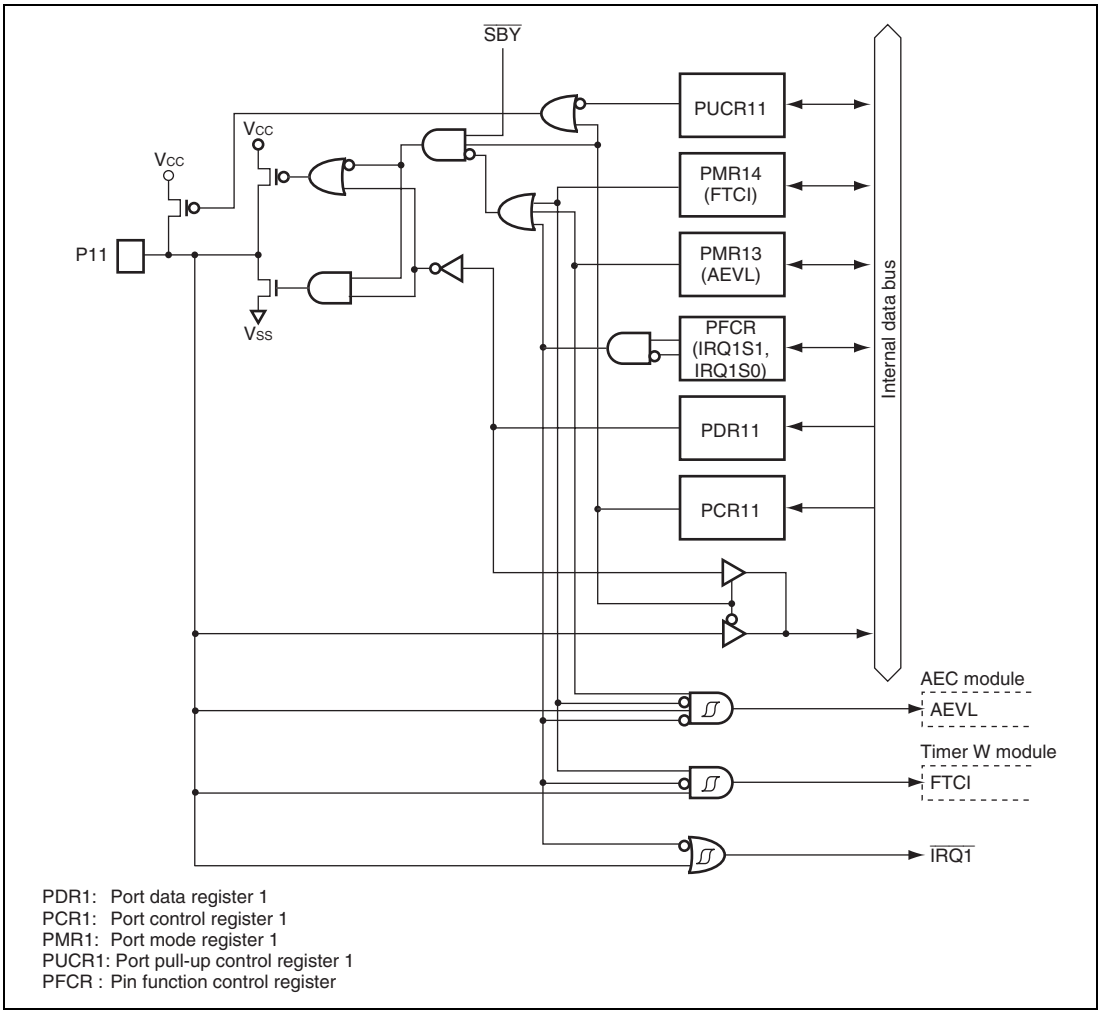


Figure B.1 (b) Port 1 Block Diagram (P11)

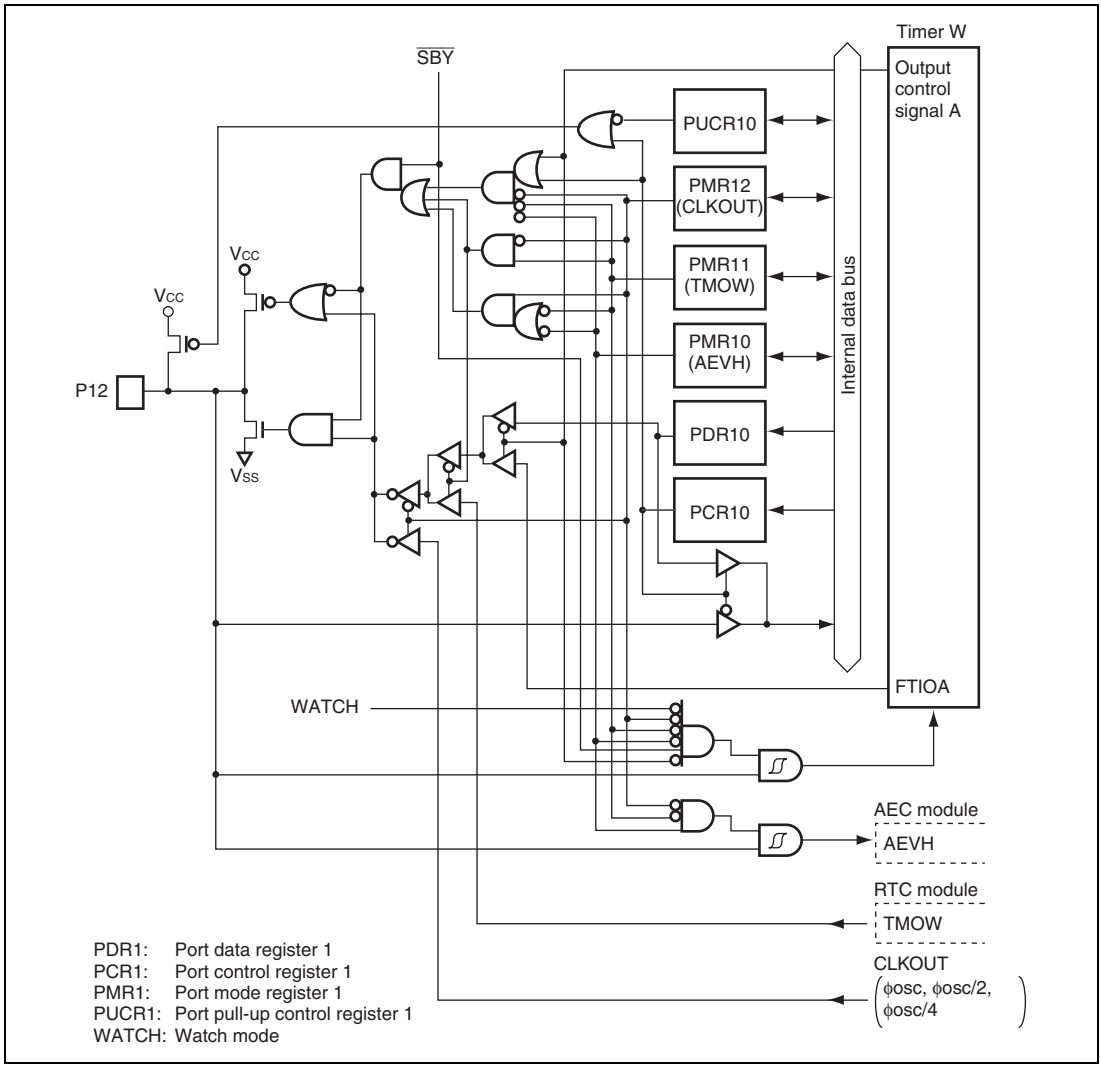


Figure B.1 (c) Port 1 Block Diagram (P10)

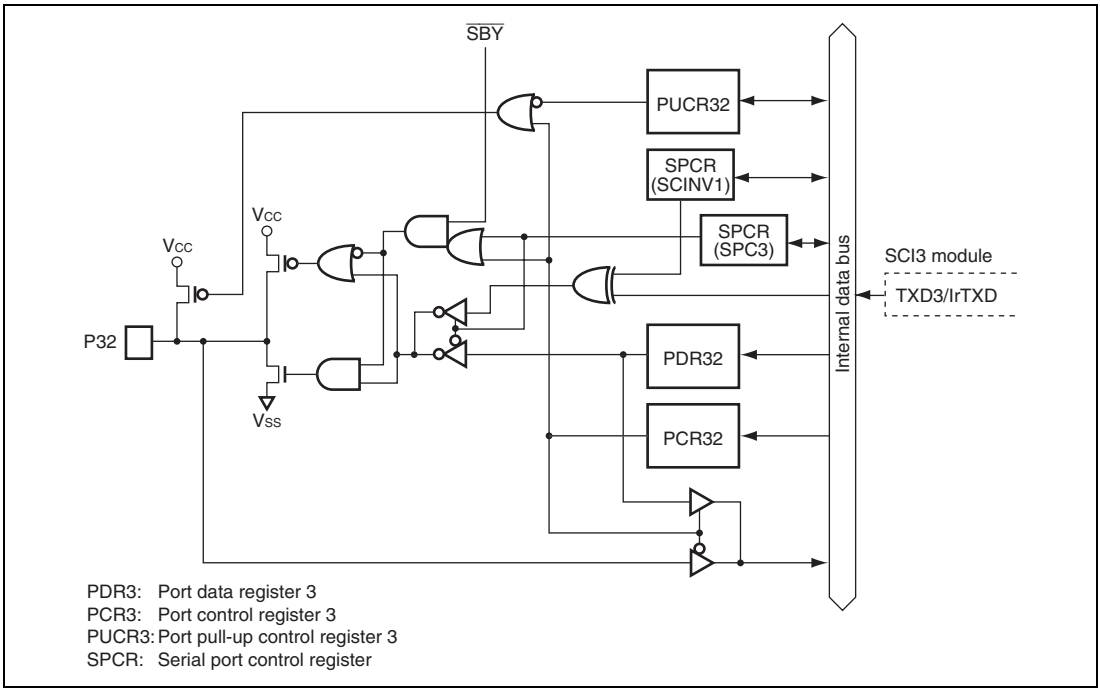


Figure B.2 (a) Port 3 Block Diagram (P32)

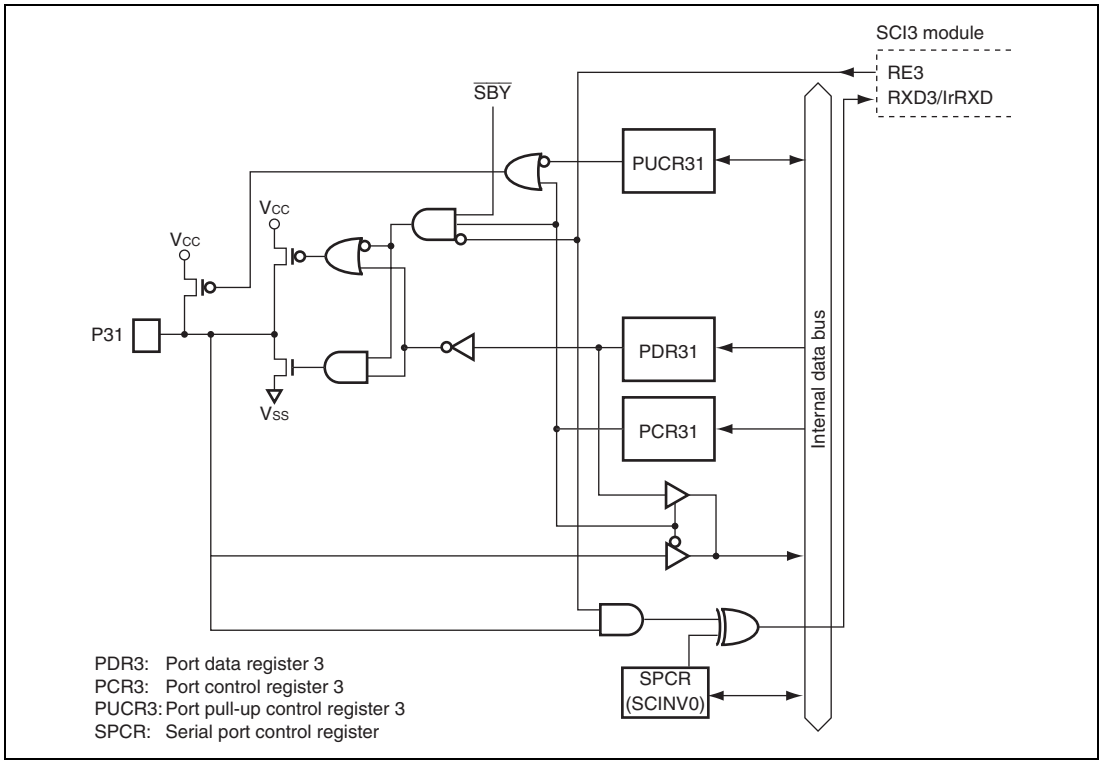


Figure B.2 (b) Port 3 Block Diagram (P31)

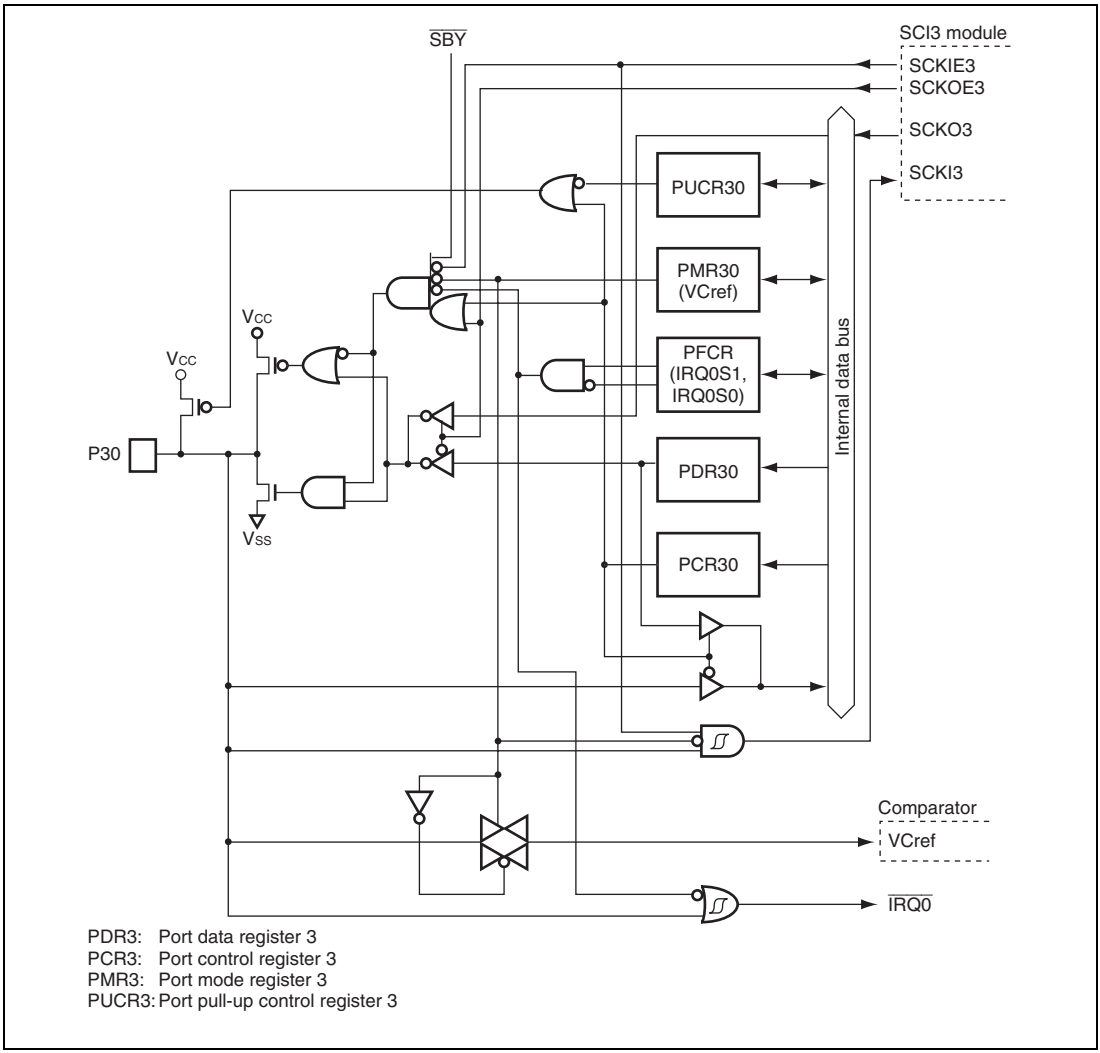


Figure B.2 (c) Port 3 Block Diagram (P30)

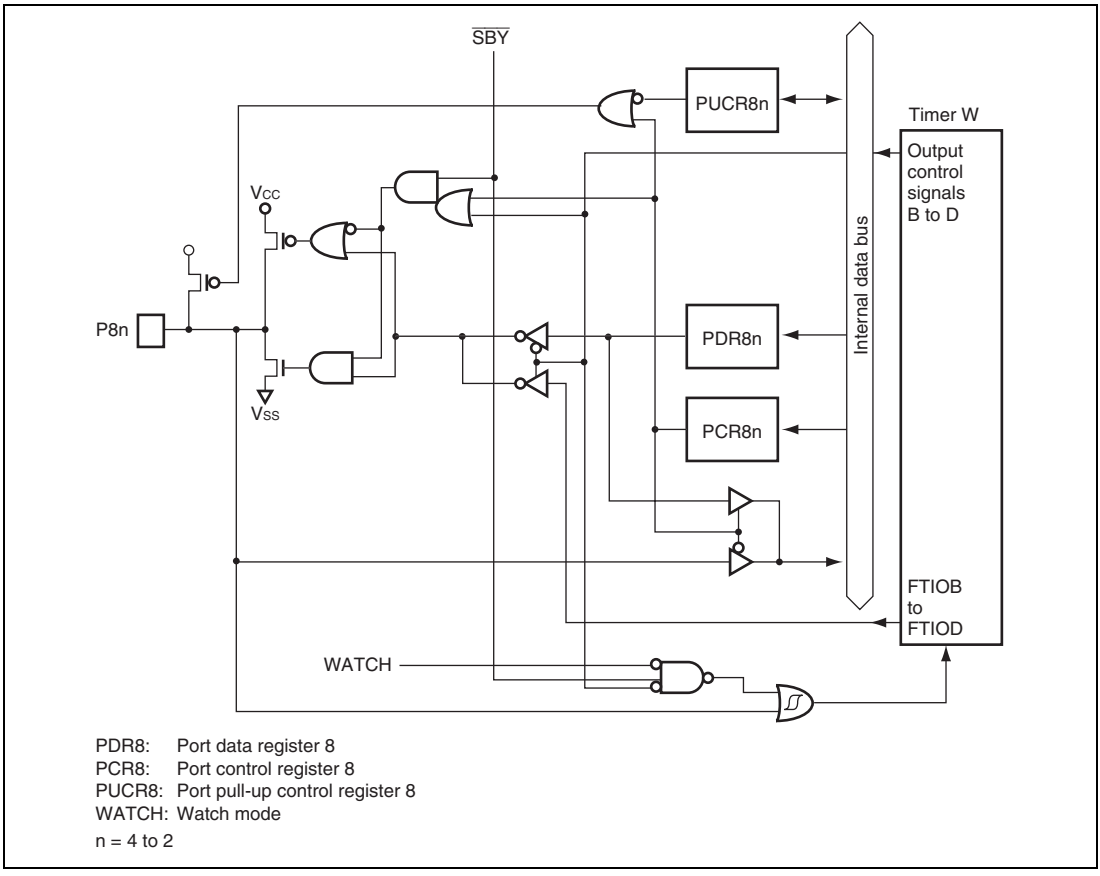


Figure B.3 (a) Port 8 Block Diagram (P84 to P82)

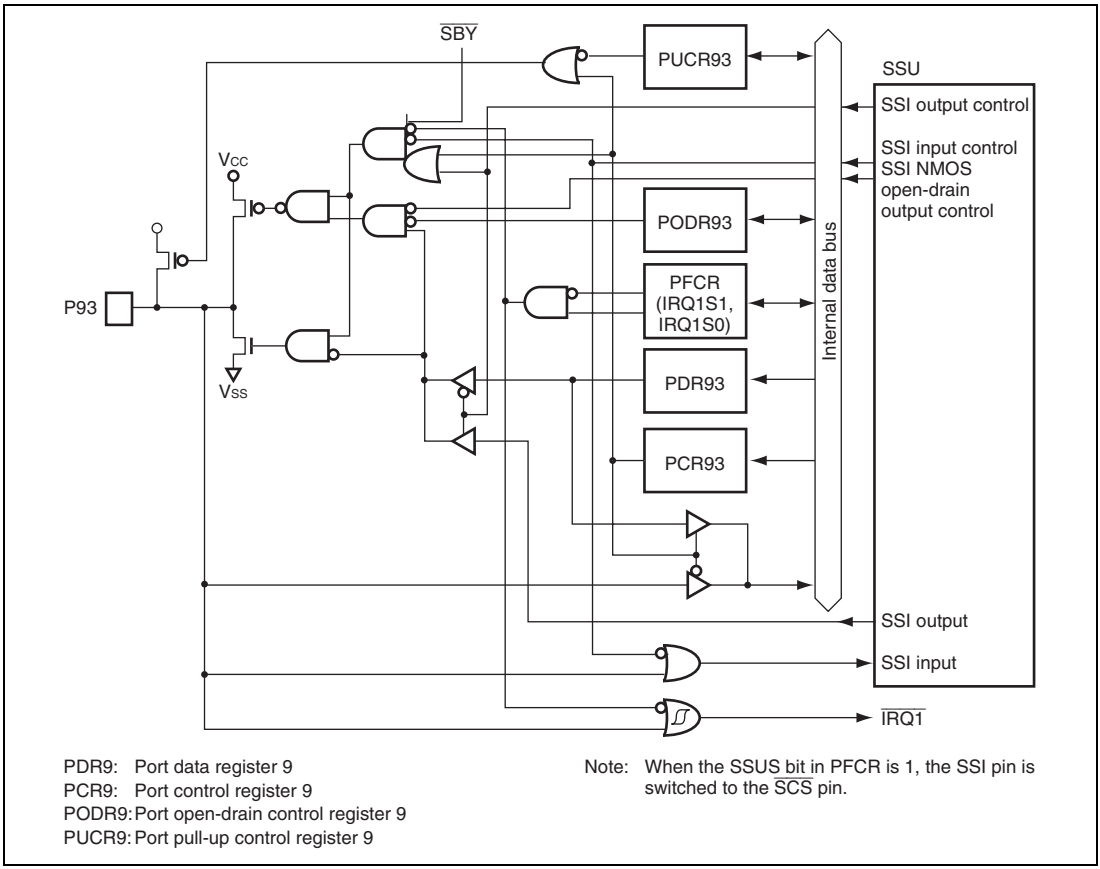


Figure B.4 (a) Port 9 Block Diagram (P93)

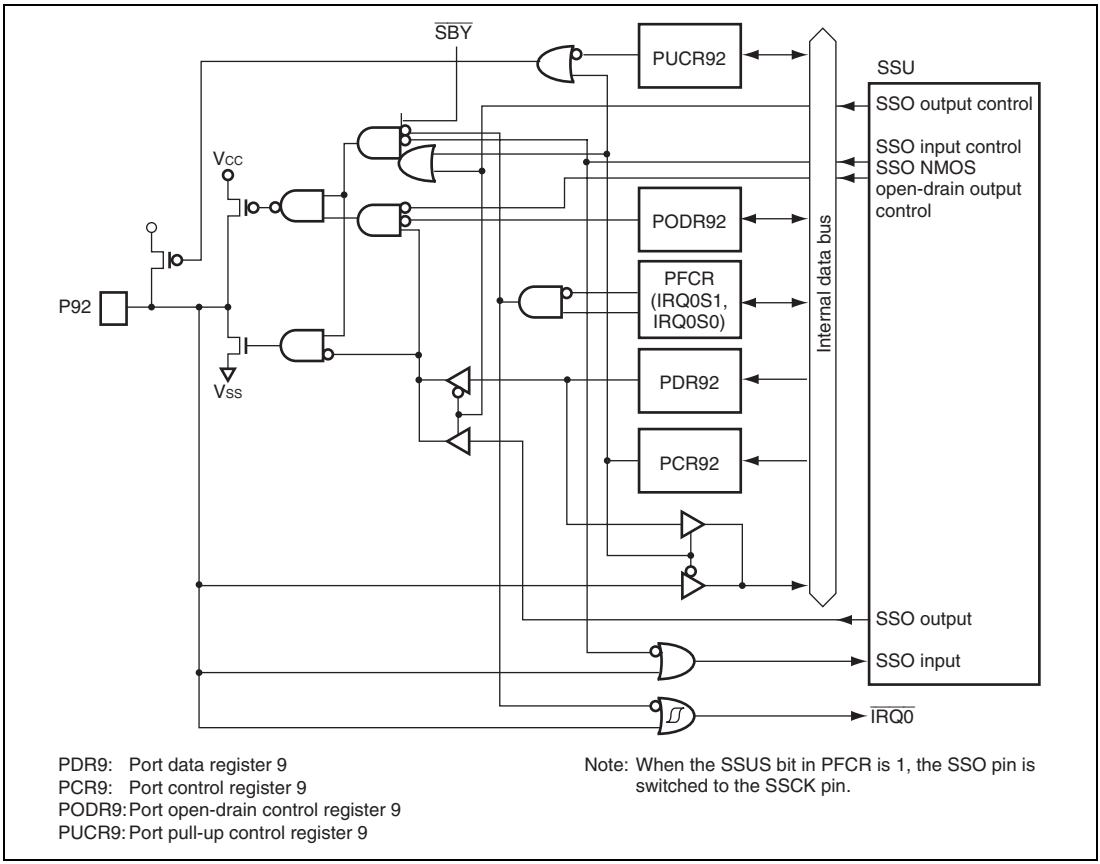


Figure B.4 (b) Port 9 Block Diagram (P92)

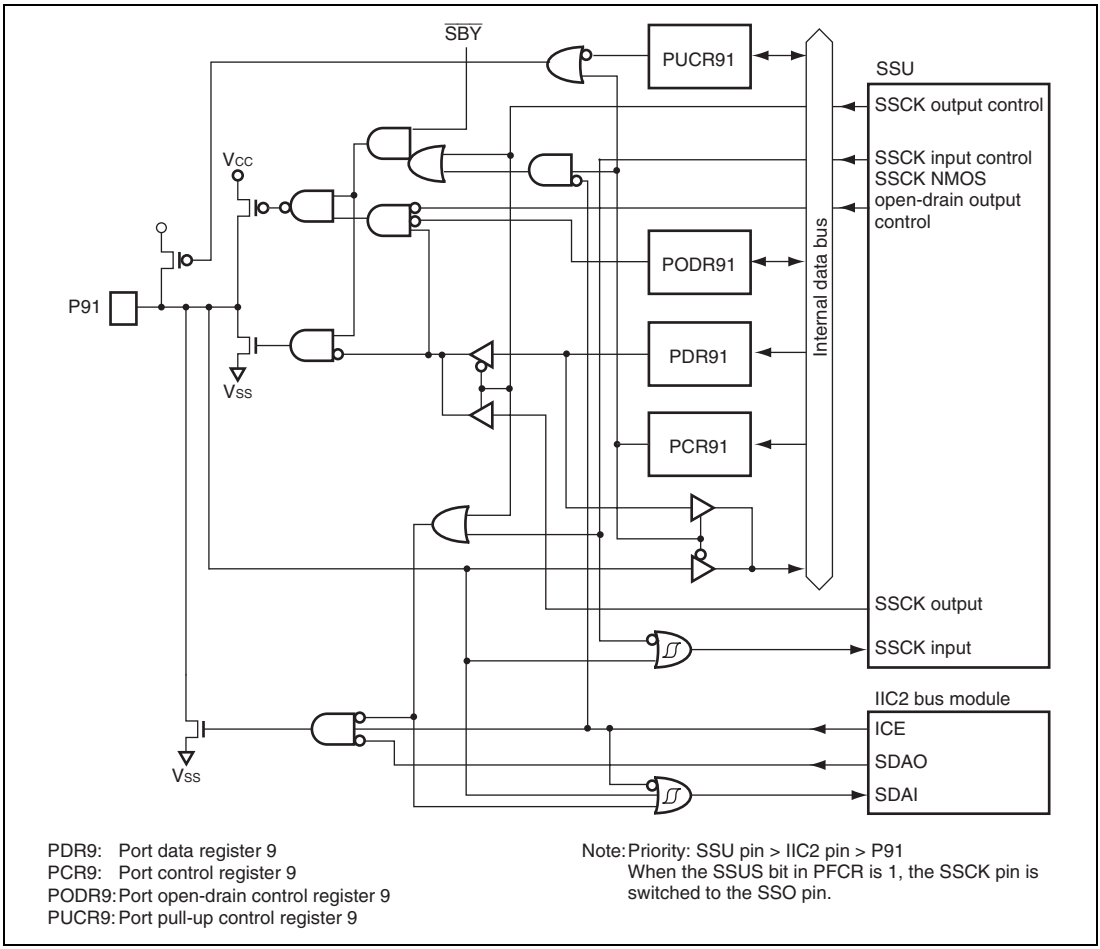


Figure B.4 (c) Port 9 Block Diagram (P91)

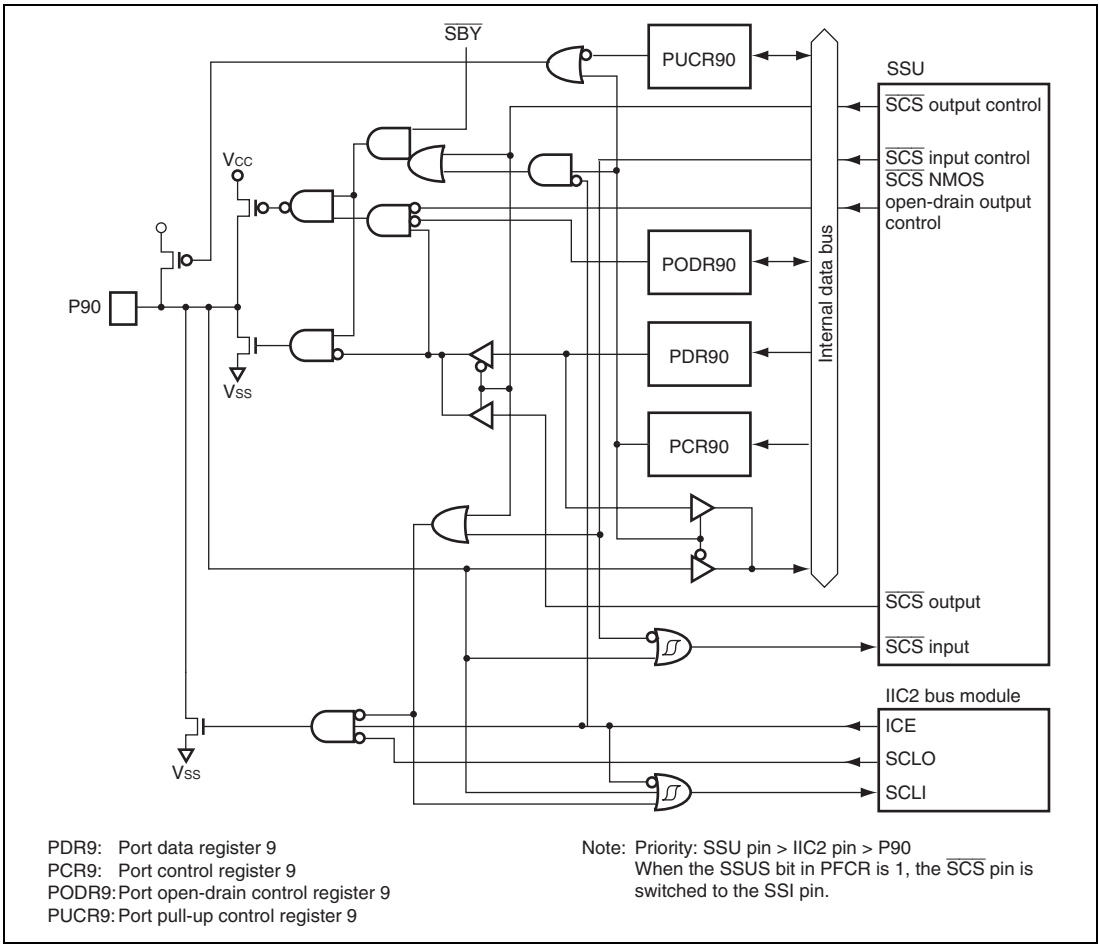


Figure B.4 (d) Port 9 Block Diagram (P90)

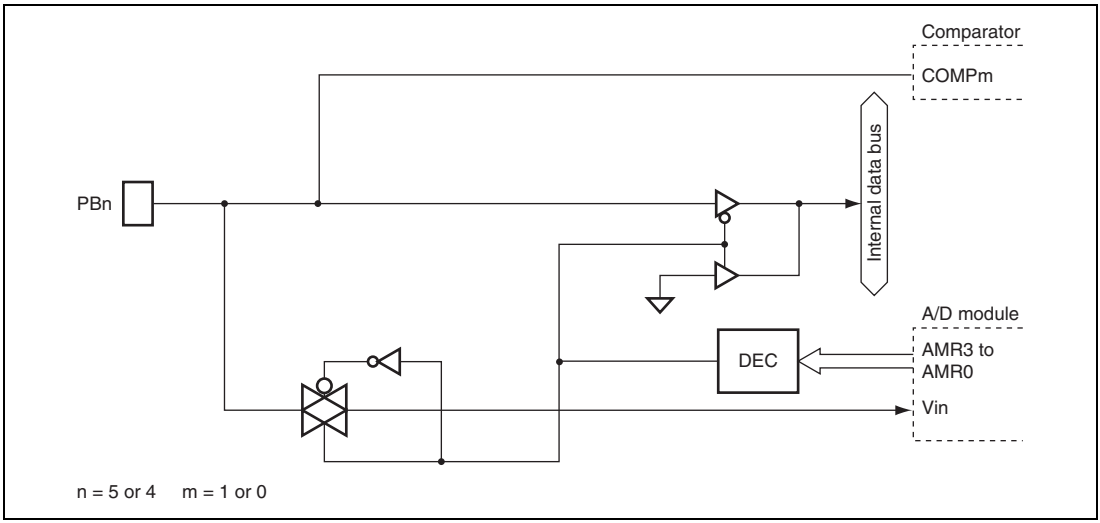


Figure B.5 (a) Port B Block Diagram (PB5 or PB4)

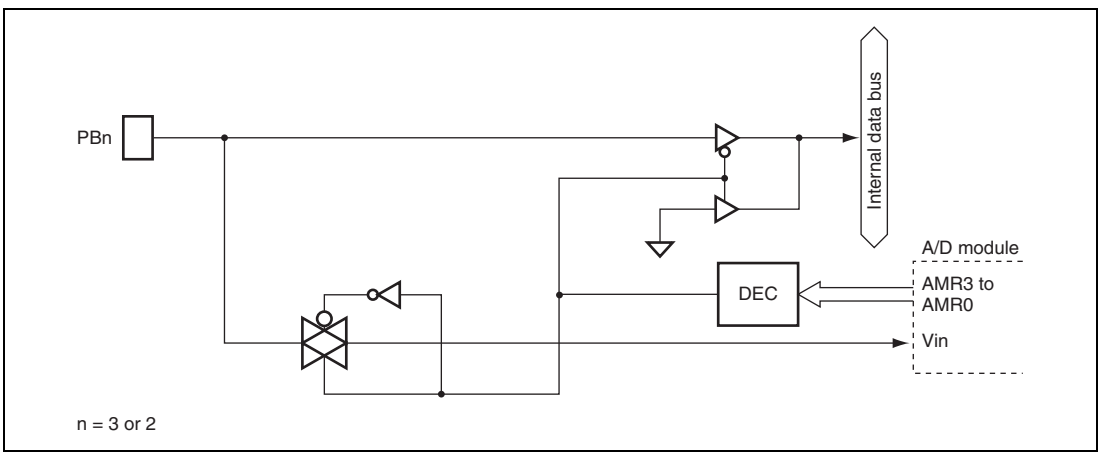


Figure B.5 (b) Port B Block Diagram (PB3 or PB2)

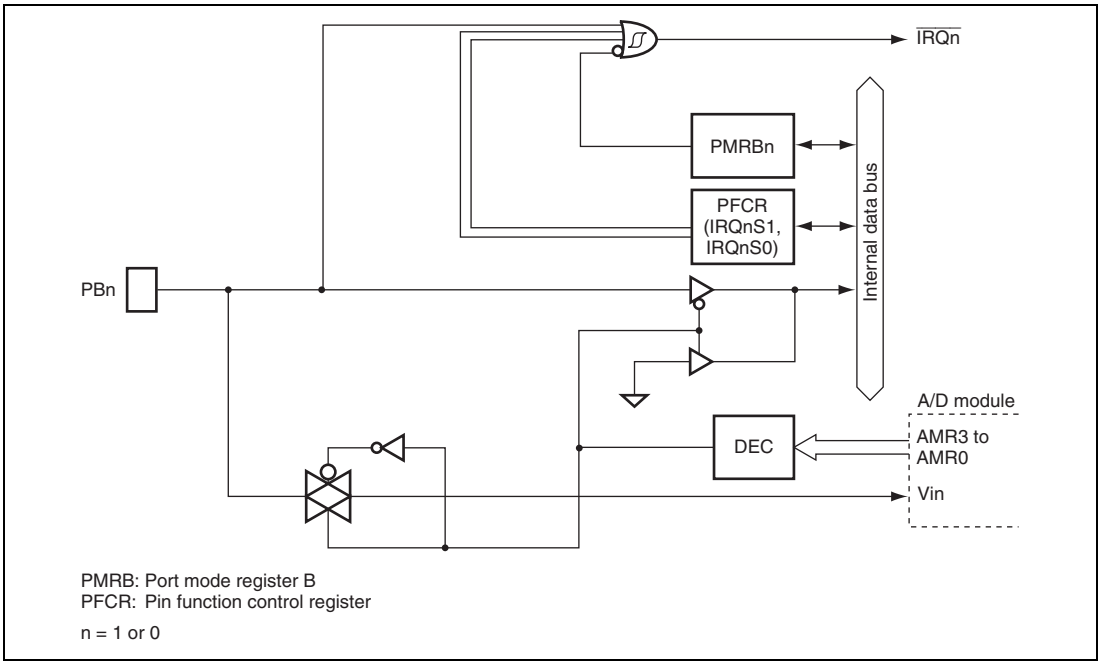


Figure B.5 (c) Port B Block Diagram (PB1 or PB0)

B.2 Port States in Each Operating State

Port	Reset	Sleep (High-Speed/ Medium-Speed)	Subsleep	Standby	Subactive	Active (High-Speed/ Medium-Speed)	Watch
P12 to P10	High impedance	Retained	Retained	High impedance*1*2	Functions	Functions	Retained
P32 to P30	High impedance	Retained	Retained	High impedance*1*2	Functions	Functions	Retained
P84 to P82	High impedance	Retained	Retained	High impedance*1*2	Functions	Functions	Retained
P93 to P90	High impedance	Retained	Retained	High impedance*1*2	Functions	Functions	Retained
PB5 to PB0	High impedance	High impedance	High impedance	High impedance*1	High impedance	High impedance	High impedance

Notes: 1. Registers are retained and output level is high impedance.
2. High-level output when the pull-up MOS is turned on.

B.3 Port 9 Related Register Settings and Pin Functions

Table B.1 Port 9 Related Register Settings and Pin Functions

SSU Setting					IIC2 Setting	PFCR Setting			Pin Functions			
SSUMS	BIDE	MSS	TE	RE	ICE	SSUS	IRQ1S1, IRQ1S0	IRQ0S1, IRQ0S0	P93	P92	P91	P90
0 (Clock synchro-nous)	*	0 (Slave)	0	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	SSI input	P92 I/O	SSCK input	P90 I/O
							Other then 01	01	SSI input	IRQ0N input	SSCK input	P90 I/O
						1	Other then 01	Other then 01	P93 I/O	SSCK input	P91 I/O	SSI input
							01	Other then 01	IRQ1N input	SSCK input	P91 I/O	SSI input
			1 (Transmit)	0	0 (IIC2 not used)	0	Other then 01	Other then 01	P93 I/O	SSO output	SSCK input	P90 I/O
							01	Other then 01	IRQ1N input	SSO output	SSCK input	P90 I/O
						1	Other then 01	Other then 01	P93 I/O	SSCK input	SSO output	P90 I/O
							01	Other then 01	IRQ1N input	SSCK input	SSO output	P90 I/O
			1 (Transmit)	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	SSI input	SSO output	SSCK input	P90 I/O
						1	Other then 01	Other then 01	P93 I/O	SSCK input	SSO output	SSI input
							01	Other then 01	IRQ1N input	SSCK input	SSO output	SSI input

SSU Setting					IIC2 Setting	PFCR Setting			Pin Functions			
SSUMS	BIDE	MSS	TE	RE	ICE	SSUS	IRQ1S1, IRQ1S0	IRQ0S1, IRQ0S0	P93	P92	P91	P90
0 (Clock synchro-nous)	*	1 (Master)	0	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	SSI input	P92 I/O	SSCK output	P90 I/O
							Other then 01	01	SSI input	IRQ0N input	SSCK output	P90 I/O
						1	Other then 01	Other then 01	P93 I/O	SSCK output	P91 I/O	SSI input
							01	Other then 01	IRQ1N input	SSCK output	P91 I/O	SSI input
			1 (Transmit)	0	0 (IIC2 not used)	0	Other then 01	Other then 01	P93 I/O	SSO output	SSCK output	P90 I/O
							01	Other then 01	IRQ1N input	SSO output	SSCK output	P90 I/O
						1	Other then 01	Other then 01	P93 I/O	SSCK output	SSO output	P90 I/O
							01	Other then 01	IRQ1N input	SSCK output	SSO output	P90 I/O
			1 (Transmit)	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	SSI input	SSO output	SSCK output	P90 I/O
						1	Other then 01	Other then 01	P93 I/O	SSCK output	SSO output	SSI input
							01	Other then 01	IRQ1N input	SSCK output	SSO output	SSI input

SSU Setting					IIC2 Setting	PFCR Setting			Pin Functions			
SSUMS	BIDE	MSS	TE	RE	ICE	SSUS	IRQ1S1, IRQ1S0	IRQ0S1, IRQ0S0	P93	P92	P91	P90
1 (Four-line bus communication)	0 (One-way)	0 (Slave)	0	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	P93 I/O	SSO input	SSCK input	SCS input
							01	Other then 01	IRQ1N input	SSO input	SSCK input	SCS input
						1	Other then 01	Other then 01	SCS input	SSCK input	SSO input	P90 I/O
			1 (Transmit)	0	0 (IIC2 not used)	0	Other then 01	Other then 01	SSI output	P92 I/O	SSCK input	SCS input
							Other then 01	01	SSI output	IRQ0N input	SSCK input	SCS input
						1	Other then 01	Other then 01	SCS input	SSCK input	P91 I/O	SSI output
			1 (Transmit)	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	SSI output	SSO input	SSCK input	SCS input
						1	Other then 01	Other then 01	SCS input	SSCK input	SSO input	SSI output
		1 (Master)	0	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	SSI input	P92 I/O	SSCK output	SCS output
							Other then 01	01	SSI input	IRQ0N input	SSCK output	SCS output
						1	Other then 01	Other then 01	SCS output	SSCK output	P91 I/O	SSI input
			1 (Transmit)	0	0 (IIC2 not used)	0	Other then 01	Other then 01	P93 I/O	SSO output	SSCK output	SCS output
							01	Other then 01	IRQ1N input	SSO output	SSCK output	SCS output
						1	Other then 01	Other then 01	SCS output	SSCK output	SSO output	P90 I/O
			1 (Transmit)	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	SSI input	SSO output	SSCK output	SCS output
						1	Other then 01	Other then 01	SCS output	SSCK output	SSO output	SSI input

SSU Setting					IIC2 Setting	PFCR Setting			Pin Functions			
SSUMS	BIDE	MSS	TE	RE	ICE	SSUS	IRQ1S1, IRQ1S0	IRQ0S1, IRQ0S0	P93	P92	P91	P90
1 (Four-line bus communication)	1 (Bidirectional)	0 (Slave)	0	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	P93 I/O	SSO input	SSCK input	SCS input
							01	Other then 01	IRQ1N input	SSO input	SSCK input	SCS input
						1	Other then 01	Other then 01	SCS input	SSCK input	SSO input	P90 I/O
			1 (Transmit)	0	0 (IIC2 not used)	0	Other then 01	Other then 01	P93 I/O	SSO output	SSCK input	SCS input
							01	Other then 01	IRQ1N input	SSO output	SSCK input	SCS input
						1	Other then 01	Other then 01	SCS input	SSCK input	SSO output	P90 I/O
		1 (Master)	0	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	P93 I/O	SSO input	SSCK output	SCS output
							01	Other then 01	IRQ1N input	SSO input	SSCK output	SCS output
						1	Other then 01	Other then 01	SCS output	SSCK output	SSO input	P90 I/O
			1 (Transmit)	0	0 (IIC2 not used)	0	Other then 01	Other then 01	P93 I/O	SSO output	SSCK output	SCS output
							01	Other then 01	IRQ1N input	SSO output	SSCK output	SCS output
						1	Other then 01	Other then 01	SCS output	SSCK output	SSO output	P90 I/O
0 (SSU not used)	0	0	0	0	1 (IIC2 used)	*	Other then 01	Other then 01	P93 I/O	P92 I/O	SDA I/O	SCL I/O
							Other then 01	01	P93 I/O	IRQ0N input	SDA I/O	SCL I/O
							01	Other then 01	IRQ1N input	P92 I/O	SDA I/O	SCL I/O
							01	01	IRQ1N input	IRQ0N input	SDA I/O	SCL I/O
					0 (IIC2 not used)	*	Other then 01	Other then 01	P93 I/O	P92 I/O	P91 I/O	P90 I/O
							Other then 01	01	P93 I/O	IRQ0N input	P91 I/O	P90 I/O
							01	Other then 01	IRQ1N input	P92 I/O	P91 I/O	P90 I/O
							01	01	IRQ1N input	IRQ0N input	P91 I/O	P90 I/O

[Legend] *: Don't care.

C. Product Part No. Lineup

Product Classification				Product Part No.	Model Marking	Package (Package Code)
H8/38602R Group	H8/38602R	Flash memory version	(10 MHz)	HD64F38602RFT10	38602R10	32-pin QFN (TNP-32)
			(4 MHz)	HD64F38602RFT4	38602R4	
			(10 MHz)	HD64F38602RFH10	F38602RFH10	32-pin LQFP (32P6U-A)
			(4 MHz)	HD64F38602RFH4	F38602RFH4	
		Masked ROM version		HD64338602RFT	38602R(***)	32-pin QFN (TNP-32)
				HD64338602RFH	38602R(***)	32-pin LQFP (32P6U-A)
	H8/38600R	Masked ROM version		HD64338600RFT	38600R(***)	32-pin QFN (TNP-32)
				HD64338600RFH	38600R(***)	32-pin LQFP (32P6U-A)

[Legend]
(***): ROM code

D. Package Dimensions

The package dimensions that are shown in the Renesas Semiconductor Packages Data Book have priority.

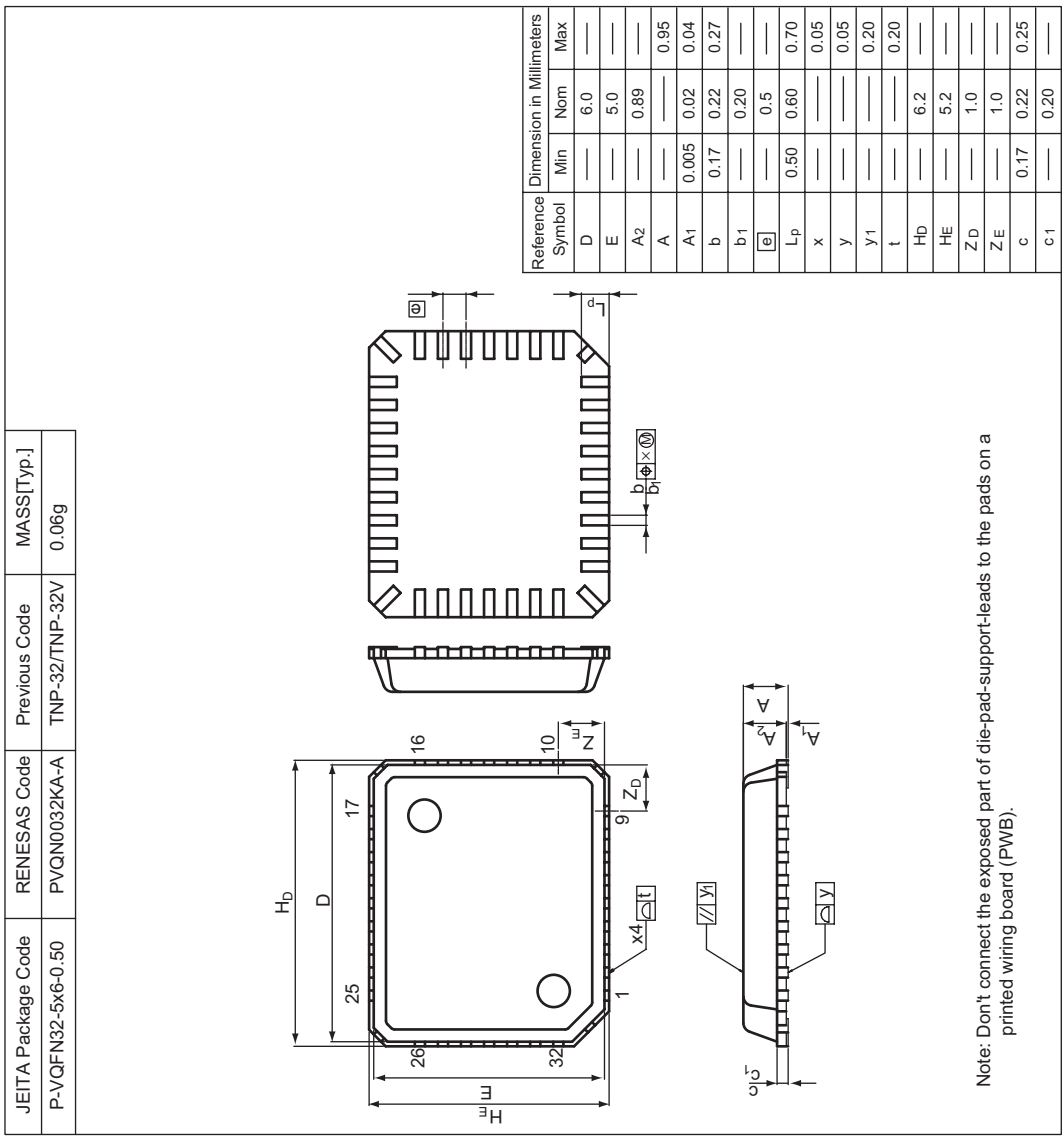


Figure D.1 Package Dimensions (TNP-32)

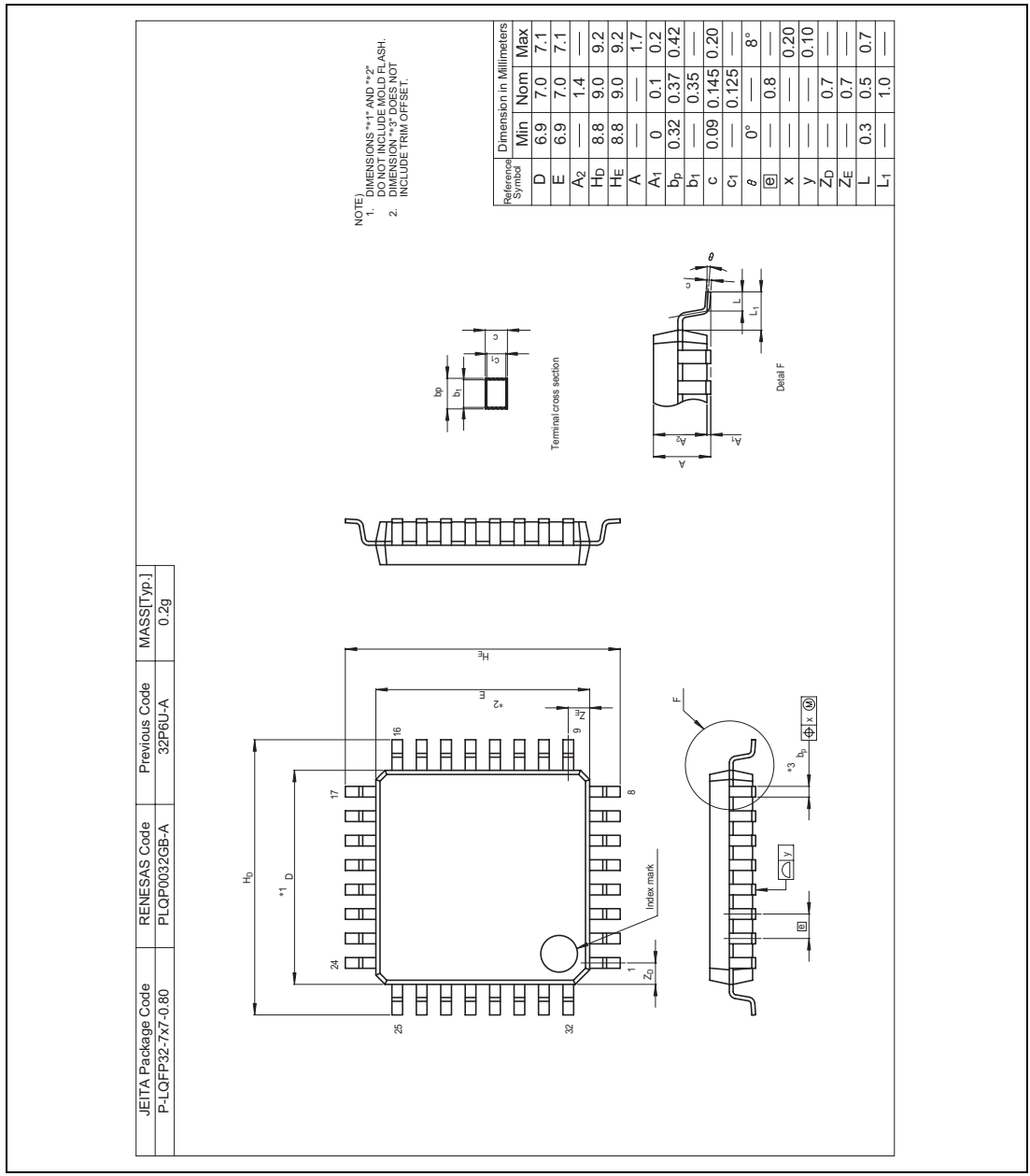
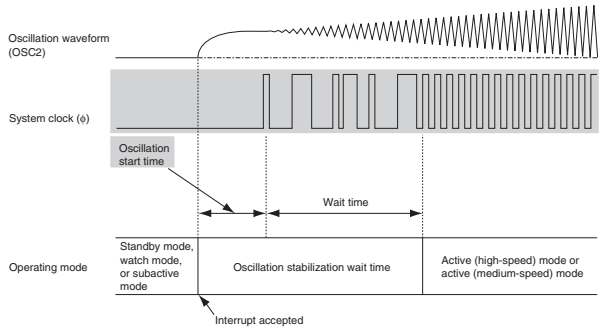


Figure D.2 Package Dimensions (32P6U-A)

Main Revisions and Additions in this Edition

Item	Page	Revisions (See Manual for Details)
Section 1 Overview 1.1 Features • Compact package	1	The description on the package, P-LQFP-32, is added.
Section 2 CPU 2.8.2 EEPMOV Instruction	36	Modified EEPMOV is a block-transfer instruction and transfers the byte size of data indicated by R4 or R4L, which starts from the address indicated by R5, to the address indicated by R6. Set R4, R4L, and R6 so that the end address of the destination address (value of R6 + R4L or R6 + R4) does not exceed H'FFFF (the value of R6 must not change from H'FFFF to H'0000 during execution).
Section 3 Exception Handling 3.2 Reset	44	Modified A reset has the highest exception priority. There are three sources to generate a reset. Table 3.2 lists the reset sources.
Table 3.2 Reset Sources	44	Added
3.2.1 Reset Exception Handling	44	The description in this section is modified.
3.8.1 Notes on Stack Area Use	58	Modified ……, so the stack pointer (SP: R7) should never indicate an odd address. To save register values, use PUSH.W Rn (MOV.W Rn, @–SP) or PUSH.L ERn (MOV.L ERn, @–SP). To restore register values, use POP.W Rn (MOV.W @SP+, Rn) or POP.L ERn (MOV.L @SP+, ERn).

Item	Page	Revisions (See Manual for Details)
Section 4 Clock Pulse Generators 4.2.4 On-Chip Oscillator Selection Method	67	Modified …… The input level on the E7_2 pin during a reset is pulled up or down using a resistor according to the selected oscillator, and fixed on exit from the reset state. When the on-chip oscillator is selected, a resonator no longer needs to be connected to the OSC1 and OSC2 pins. In such a case, fix the OSC1 pin to GND or leave it open, and leave the OSC2 pin open.
		Note is added. Notes: 1. …… 2. When the on-chip debugger is connected, the value of the resistor should be high. When not connected, it is specified according to the selected oscillator.
Figure 4.5 Typical Connection to 32.768-kHz/38.4-kHz Crystal Resonator	68	Modified Frequency / Manufacturer / Products Name / Equivalent Series Resistance 38.4 kHz / EPSON TOYOCOM CORPORATION / C-4-TYPE / 30 kΩ (max.) 32.768 kHz / EPSON TOYOCOM CORPORATION / C-001R / 35 kΩ (max.) $C_1 = C_2$ = 7 pF (typ.) Note: Consult with the crystal resonator manufacturer to determine the parameters.
4.3.1 Connecting 32.768-kHz/38.4-kHz Crystal Resonator	68	Added 1. When the resonator other than ones listed above is used, perform matching evaluation with the crystal resonator manufacture and connect it under the optimum condition. Even when the resonator listed above or the equivalent is used, as the oscillation characteristics depend on the board specification, perform matching evaluation on the mounting board. 2. Perform matching evaluation in the reset state (the RES pin is low) and on exit from the reset state (the RES pin is driven from low to high).
4.4.1 Prescaler S	71	Deleted The output from prescaler S is shared by the on-chip peripheral modules. ~~The division ratio can be set separately for each on-chip peripheral function.~~

Item	Page	Revisions (See Manual for Details)
4.5.1 Note on Resonators and Resonator Circuits	72	Modified Resonator characteristics are closely related to board design. Therefore, resonators should be assigned after being carefully evaluated by the user in the masked ROM version and flash memory version, with referring to the examples shown in this section.
4.5.3 Definition of Oscillation Stabilization Wait Time	74	The description in this section is modified.
Figure 4.12 Oscillation Stabilization Wait Time	75	Modified 
4.5.5 Note on the Oscillation Stabilization of Resonators	76	The title modified
4.5.6 Note on Using Power-On Reset	76	Modified The power-on reset circuit in this LSI adjusts the reset clear time by the capacitor capacitance, which is externally connected to the $\overline{\text{RES}}$ pin. The external capacitor capacitance should be adjusted to secure the oscillation stabilization time before reset clearing. For details, refer to section 19, Power-On Reset Circuit.
Section 5 Power-Down Modes 5.1.3 Clock Halt Registers 1 and 2 (CKSTPR1 and CKSTPR2) • CKSTPR2	81	The note is modified. Notes: 3. … When ~~the watchdog timer stops operating and~~ the WDON bit is cleared to 0 by software, this bit is valid and the watchdog timer enters module standby mode.
Table 5.3 Internal State in Each Operating Mode	86	The note is modified. Notes: 6. Functions if the 32.768-kHz RTC is selected as an internal clock. Halted and retained otherwise.

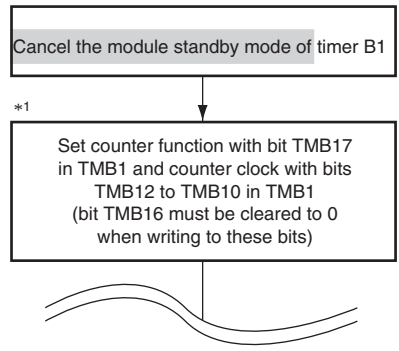
Item	Page	Revisions (See Manual for Details)
5.2.2 Standby Mode	88	Modified … However, as long as the rated voltage is supplied, the contents of CPU registers~~, on-chip RAM,~~ and some on-chip peripheral module registers are retained. …
		Modified … or the requested interrupt is disabled by the interrupt enable bit. When a reset source is generated in standby mode, the system clock oscillator starts. If a reset is generated by the $\overline{\text{RES}}$ pin, it must be kept low until the system clock oscillator output stabilizes and the t_{RFI} period has elapsed. The CPU starts reset exception handling when the $\overline{\text{RES}}$ pin is driven high.
5.2.3 Watch Mode	88	Modified … or the requested interrupt is disabled by the interrupt enable register. When a reset source is generated in watch mode, the system clock oscillator starts. If a reset is generated by the $\overline{\text{RES}}$ pin, it must be kept low until the system clock oscillator output stabilizes. The CPU starts reset exception handling when the $\overline{\text{RES}}$ pin is driven high.
5.2.4 Subsleep Mode	89	Modified … or the requested interrupt is disabled by the interrupt enable register. When a reset source is generated in subsleep mode, the system clock oscillator starts. If a reset is generated by the $\overline{\text{RES}}$ pin, it must be kept low until the system clock oscillator output stabilizes. The CPU starts reset exception handling when the $\overline{\text{RES}}$ pin is driven high.
5.2.5 Subactive Mode	89	Modified … on the combination of bits SSBY, LSON, and TMA3 in SYSCR1 and bits MSON and DTON in SYSCR2. ~~Subactive mode is not cleared if the I bit in CCR is set to 1 or the requested interrupt is disabled by the interrupt enable register.~~ When a reset source is generated in subactive mode, the system clock oscillator starts. If a reset is generated by the $\overline{\text{RES}}$ pin, it must be kept low until the system clock oscillator output stabilizes and the t_{RFI} period has elapsed. The CPU starts reset exception handling when the $\overline{\text{RES}}$ pin is driven high.

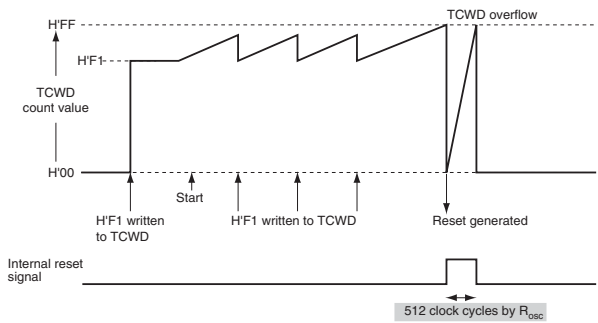
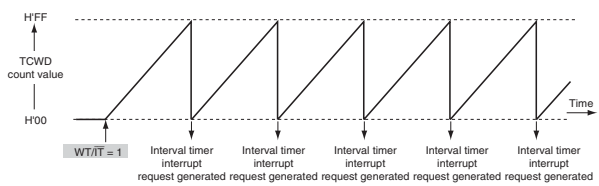
Item	Page	Revisions (See Manual for Details)
5.2.6 Active (Medium-Speed) Mode	90	Modified In active (medium-speed) mode, the clock set by the MA1 and MA0 bits in SYSCR1 is used as the system clock, and the CPU and on-chip peripheral modules function. Active (medium-speed) mode is cleared by the SLEEP instruction. When active (medium-speed) mode is cleared, a transition to standby mode is made depending on the combination of bits SSBY, LSON, and TMA3 in SYSCR1, a transition to watch mode is made depending on the combination of bits SSBY and TMA3 in SYSCR1, or a transition to sleep mode is made depending on the combination of bits SSBY and LSON in SYSCR1. Moreover, a transition to active (high-speed) mode or subactive mode is made by a direct transition. ~~Active (medium-sleep) mode is not entered if the I bit in CCR is set to 1 or the requested interrupt is disabled in the interrupt enable register.~~ When the $\overline{\text{RES}}$ pin goes low, the CPU goes into the reset state and active (medium-sleep) mode is cleared. ~~In active (medium-speed) mode, the on-chip peripheral modules function at the clock set by the MA1 and MA0 bits in SYSCR1.~~
5.3 Direct Transition	91	The description in this section is modified.
5.3.1 Direct Transition from Active (High-Speed) Mode to Active (Medium-Speed) Mode	91	Added When a SLEEP instruction is executed in active (high-speed) mode while the SSBY and LSON bits in SYSCR1 are cleared to 0 and the MSON and DTON bits in SYSCR2 are set to 1, a transition is made to active (medium-speed) mode via sleep 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). Example: When ϕosc/8 is selected as the CPU operating clock after the transition Direct transition time = $(2 + 1) \times 1$tosc + 14×8tosc = 115tosc For the legend of symbols used above, refer to section 21, Electrical Characteristics.

Item	Page	Revisions (See Manual for Details)
5.3.2 Direct Transition from Active (High-Speed) Mode to Subactive Mode	92	Added When a SLEEP instruction is executed in active (high-speed) mode while the SSBY, TMA3, and LSON bits in SYSCR1 are set to 1 and the DTON bit in SYSCR2 is set to 1, a transition is made to subactive mode via watch 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). Example: When $\phi w/8$ is selected as the subactive operating clock after the transition Direct transition time = $(2 + 1) \times 1tosc + 14 \times 8tw = 3tosc + 112tw$ For the legend of symbols used above, refer to section 21, Electrical Characteristics.
5.3.3 Direct Transition from Active (Medium-Speed) Mode to Active (High-Speed) Mode	92	Added When a SLEEP instruction is executed in active (medium-speed) mode while the SSBY and LSON bits in SYSCR1 are cleared to 0, the MSON bit in SYSCR2 is cleared to 0, and the DTON bit in SYSCR2 is set to 1, a transition is made to active (high-speed) mode via sleep mode. The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (3). Example: When $\phi osc/8$ is selected as the CPU operating clock before the transition Direct transition time = $(2 + 1) \times 8tosc + 14 \times 1tosc = 38tosc$ For the legend of symbols used above, refer to section 21, Electrical Characteristics.

Item	Page	Revisions (See Manual for Details)
5.3.4 Direct Transition from Active (Medium-Speed) Mode to Subactive Mode	93	Added When a SLEEP instruction is executed in active (medium-speed) mode while the SSBY, LSON, and TMA3 bits in SYSCR1 are set to 1 and the DTON bit in SYSCR2 is set to 1, a transition is made to subactive mode via watch 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 (4). Example: When ϕosc/8 and ϕw/8 are selected as the CPU operating clock before and after the transition, respectively Direct transition time = (2 + 1) × 8tosc + 14 × 8tw = 24tosc + 112tw For the legend of symbols used above, refer to section 21, Electrical Characteristics.
5.3.5 Direct Transition from Subactive Mode to Active (High-Speed) Mode	93	Added and modified When a SLEEP instruction is executed in subactive mode while the SSBY and TMA3 bits in SYSCR1 are set to 1, the LSON bit in SYSCR1 is cleared to 0, the MSON bit in SYSCR2 is cleared to 0, and the DTON bit in SYSCR2 is set to 1, a transition is made directly to active (high-speed) mode via watch mode after the waiting time set in bits STS2 to STS0 in SYSCR1 has elapsed. The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (5). Direct transition time = {(Number of SLEEP instruction execution states) + (Number of internal processing states)} × (tsubcyc before transition) + (Wait time set in bits STS2 to STS0) + (Number of interrupt exception handling execution states) × (tcyc after transition)…..(5) Example: When ϕw/8 is selected as the CPU operating clock after the transition and wait time = 8192 states Direct transition time = (2 + 1) × 8tw + (8192 + 14) ×1tosc = 24tw + 8206tosc For the legend of symbols used above, refer to section 21, Electrical Characteristics.

Item	Page	Revisions (See Manual for Details)
5.3.6 Direct Transition from Subactive Mode to Active (Medium-Speed) Mode	94	Added When a SLEEP instruction is executed in subactive mode while the SSBY and TMA3 bits in SYSCR1 are set to 1, the LSON bit in SYSCR1 is cleared to 0, and the MSON and DTON bits in SYSCR2 are set to 1, a transition is made directly to active (medium-speed) mode via watch mode after the waiting time set in bits STS2 to STS0 in SYSCR1 has elapsed. The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (6). Example: When ϕw/8 and ϕosc/8 are selected as the CPU operating clock before and after the transition, respectively, and wait time = 8192 states Direct transition time = (2 + 1) $\times$ 8tw + 8192 $\times$ 1tosc + 14 $\times$ 8tosc = 24tw + 8304tosc For the legend of symbols used above, refer to section 21, Electrical Characteristics.
Section 6 ROM 6.7 Notes on Setting Module Standby Mode	117	Modified … Then the flash memory should be set to enter the module standby mode. If an interrupt is generated in module standby mode, the vector address cannot be fetched. As a result, the program may run away.
Section 8 I/O Ports 8.1.5 Pin Functions • P10/AEVH/FTIOA/TMOW/CLKOUT pin	122	Added Note: * Switching the clock (ϕ_{osc}, ϕ_{osc}/2, or ϕ_{osc}/4) for CLKOUT output must be performed when CLKOUT output is halted (CLKOUT = 0). When making a transition to a power-down mode wherein the system clock oscillator is halted, the output level is retained. (In standby mode, output is the high-impedance state.) When making a transition from a power-down mode wherein the system clock oscillator is halted, to the active mode wherein the system clock oscillator operates, halt CLKOUT output (CLKOUT = 0) before the transition.
8.7.2 Input Characteristics Difference due to Pin Function	124	This section is newly added.

Item	Page	Revisions (See Manual for Details)
Section 9 Timer B1 Figure 9.2 Timer B1 Initial Setting Flow	148	Modified 
Section 11 Realtime Clock (RTC) 11.3.7 Clock Source Select Register (RTCCSR)	193	Modified **Bit / Bit Name / Description** 3 RCS3 — Clock Source Selection 2 RCS2 1 RCS1 0 RCS0 0000: ϕ/8 ····· Free running counter operation 0001: ϕ/32 ····· Free running counter operation 0010: ϕ/128 ····· Free running counter operation 0011: ϕ/256 ····· Free running counter operation 0100: ϕ/512 ····· Free running counter operation 0101: ϕ/2048 ····· Free running counter operation 0110: ϕ/4096 ····· Free running counter operation 0111: ϕ/8192 ····· Free running counter operation 1000: ϕ_w/4 ····· RTC operation 1001 to 1111: Setting prohibited
11.4.1 Initial Settings of Registers after Power-On	196	Modified The RTC registers that store second, minute, hour, and day-of-week data, control registers, and interrupt registers are not reset by a RES input, or by a reset source caused by a watchdog timer.
11.5 Interrupt Sources	198	Modified … When using an interrupt, set the IENRTC (RTC interrupt request enable) bit in IENR1 to 1 last after other registers are set.
11.6.2 Note when Using RTC Interrupts	199	Added

Item	Page	Revisions (See Manual for Details)
Section 12 Watchdog Timer 12.2.1 Timer Control/Status Register WD1 (TCSRWD1)	203	Added **Bit** **Bit Name** **Description** 0 WRST Watchdog Timer Reset Indicates whether a reset caused by the watchdog timer is generated. This bit is not cleared by a reset caused by the watchdog timer. [Setting condition] When TCWD overflows and an internal reset signal is generated
12.3.1 Watchdog Timer Mode	208	Modified … 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. The internal reset signal is output for a period of 512 clock cycles by the on-chip oscillator (R_{OSC}).
Figure 12.2 Example of Watchdog Timer Operation	208	Modified 
Figure 12.3 Interval Timer Mode Operation	209	Modified 
12.5.3 Clearing the WT/IT or IEOVF Bit in TCSRWD2 to 0	210	Added
Table 12.1 Assembly Program for Clearing WT/IT or IEOVF Bit to 0	211	Added

Item	Page	Revisions (See Manual for Details)
Table 12.2 The Value of "xx"	211	Added
Section 13 Asynchronous Event Counter (AEC) 13.3.6 Event Counter H (ECH)	222	Modified

Bit	Bit Name	Description
7	ECH7	Either the external asynchronous event AEVH pin, φ/2, φ/4, or φ/8, or the overflow signal from lower 8-bit counter ECL can be selected as the input clock source. ECH can be cleared to H'00 when the CRCH bit in ECCSR is cleared to 0.
6	ECH6	
5	ECH5	
4	ECH4	
3	ECH3	
2	ECH2	
1	ECH1	
0	ECH0	

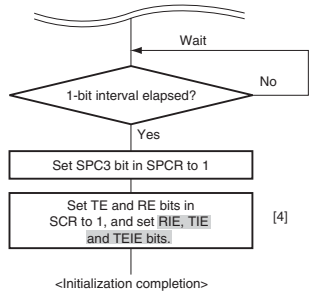
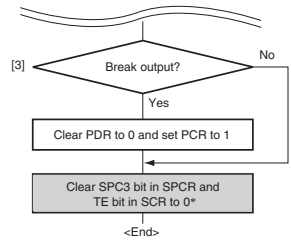
13.3.7 Event Counter L (ECL)	222	Modified

Bit	Bit Name	Description
7	ECL7	Either the external asynchronous event AEVL pin, φ/2, φ/4, or φ/8 can be selected as the input clock source. ECL can be cleared to H'00 when the CRCL bit in ECCSR is cleared to 0.
6	ECL6	
5	ECL5	
4	ECL4	
3	ECL3	
2	ECL2	
1	ECL1	
0	ECL0	

Section 14 Serial Communication Interface 3 (SCI3, IrDA)	231	Deleted The serial communication interface 3 (SCI3) can handle both asynchronous and clock synchronous serial communication. In the asynchronous method, serial data communication can be carried out using standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or an Asynchronous Communication Interface Adapter (ACIA). ~~A function is also provided for serial communication between processors (multiprocessor communication function).~~
14.3.5 Serial Mode Register (SMR)	235	Modified

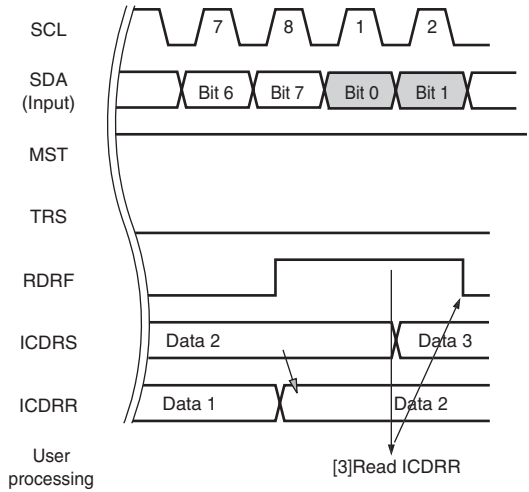
Bit	Bit Name	Description
2	MP	5-Bit CommunicationWhen this bit is set to 1, the 5-bit communication format is enabled. Make sure to set bit 5 (PF) to 1 when setting this bit (MP) to 1.

<table>
<tr><th>Item</th><th>Page</th><th>Revisions (See Manual for Details)</th></tr>
<tr><td>14.3.6 Serial Control Register (SCR)</td><td>237</td><td>Bit 3 is reserved.</td></tr>
<tr><td>14.3.7 Serial Status Register (SSR)</td><td>240</td><td>Deleted
SSR consists of status flags of the SCI3~~ and multiprocessor bits for transfer~~. 1 cannot be written to flags TDRE, RDRF, OER, PER, and FER; they can only be cleared.</td></tr>
<tr><td></td><td>240</td><td>Bits 1 and 0 are reserved.</td></tr>
<tr><td>14.3.10 IrDA Control Register (IrCR)</td><td>252</td><td>Modified
<table><tr><th>Bit</th><th>Bit Name</th><th>Description</th></tr><tr><td>7</td><td>IrE</td><td>IrDA EnableSelects whether the SCI3 I/O pins function as the SCI3 or IrDA.0: TXD3/IrTXD and RXD3/IrRXD pins function as TXD3 and RXD31: TXD3/IrTXD and RXD3/IrRXD pins function as IrTXD and IrRXD</td></tr></table></td></tr>
<tr><td>14.3.11 Serial Extended Mode Register (SEMR)</td><td>253</td><td>Added
<table><tr><th>Bit</th><th>Bit Name</th><th>Description</th></tr><tr><td>3</td><td>ABCS</td><td>Asynchronous Mode Basic Clock Select
Selects the basic clock for the bit period in asynchronous mode.
This setting is enabled only in asynchronous mode (COM bit in SMR3 is 0).
0: Operates on a basic clock with a frequency of 16 times the transfer rate
1: Operates on a basic clock with a frequency of eight times the transfer rate
Clear the ABCS bit to 0, when the IrDA function is enabled.</td></tr></table></td></tr>
<tr><td>Table 14.8 Data Transfer Formats (Asynchronous Mode)</td><td>255</td><td>The formats are modified.</td></tr>
<tr><td>Table 14.9 SMR Settings and Corresponding Data Transfer Formats</td><td>256</td><td>The settings are modified.</td></tr>
</table>

Item	Page	Revisions (See Manual for Details)
Figure 14.4 Sample SCI3 Initialization Flowchart	258	Modified  [[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Setting bits TE and RE enables the TXD3 and RXD3 pins to be used. Also set the RIE, TIE, and TEIE bits, depending on whether interrupts are required. In asynchronous mode, the bits are marked at transmission and idled at reception to wait for the start bit.
Figure 14.6 Sample Serial Transmission Flowchart (Asynchronous Mode)	260	Modified  [3] To output a break in serial transmission, after setting PCR to 1 and PDR to 0, clear the SPC3 bit in SPCR and the TE bit in SCR to 0. Note: * When the SPC3 bit in SPCR is cleared to 0, the pin functions as an I/O port.
14.5 Operation in Clock Synchronous Mode	264	Deleted … After 8-bit data is output, the transmission line holds the MSB state. In clock synchronous mode, no parity ~~or multiprocessor~~ bit is added.
14.6 Multiprocessor Communication Function	270	This section is deleted.

Item	Page	Revisions (See Manual for Details)
14.8.2 Mark State and Break Sending	277	Modified When the SPC3 bit in SPCR is 0, the TXD3 pin functions as an I/O port whose direction (input or output) and level are determined by PCR and PDR, regardless of the TE setting. This can be used to set the TXD3 pin to the mark state (high level) or send a break during data transmission. To maintain the communication line at the mark state until the SPC3 bit in SPCR is set to 1, set both PCR and PDR to 1. As the SPC3 bit in SPCR is cleared to 0 at this point, the TXD3 pin functions as an I/O port, and 1 is output from the TXD3 pin. To send a break during data transmission, first set PCR to 1 and PDR to 0, and then clear the SPC3 and TE bits to 0. When the TE bit is cleared to 0 directly after the SPC3 bit is cleared to 0, the transmitter is initialized regardless of the current transmission state after the TE bit is cleared, the TXD3 pin functions as an I/O port after the SPC3 bit is cleared, and 0 is output from the TXD3 pin.
Section 16 I2C Bus Interface 2 (IIC2) 16.3.5 I^2C Bus Status Register (ICSR)	323	Modified

Bit	Bit Name	Description
3	STOP	Stop Condition Detection Flag[Setting conditions] • In master mode, when a stop condition is detected after the completion of frame transfer • In slave mode, when a stop condition is detected, after the slave address of the first byte, following the general call and the detection of the start condition, matches the address set in SAR

Item	Page	Revisions (See Manual for Details)
Figure 16.15 Receive Mode Operation Timing	339	Modified 
Section 17 A/D Converter 17.3.1 A/D Result Register (ADRR)	350	Modified ADRR is a 16-bit read-only register that stores the results of A/D conversion. The data is stored in the upper 10 bits of ADRR. ADRR can be read by the CPU at any time, …
17.7.3 Usage Notes	360	Deleted 3. When A/D conversion is started after clearing module standby mode, wait for 10ϕ clock cycles before starting A/D conversion. 4. ~~When the LADS bit in ADSR is changed as from halting to operating, wait for 10ϕ clock cycles before starting A/D conversion.~~

Item	Page	Revisions (See Manual for Details)
Section 18 Comparators 18.5 Usage Notes	368	Modified 4. If the LSI enters the ~~software~~ standby mode or watch mode when a comparator is operating, the internal operation of the comparator is maintained. Since the comparator operates even in ~~software~~ standby mode or watch mode, it returns to the same mode after the specified interrupt is canceled, though the current for the comparator is consumed. If a comparator is not required to return to the ~~software~~ standby mode or watch mode when an interrupt is canceled and the current consumption needs to be reduced, stop the comparator by clearing the CME0 and CME1 bits in CMCR0 and CMCR1 to 0 before shifting the mode.
Section 19 Power-On Reset Circuit 19.2.1 Power-On Reset Circuit	370	Modified The operation timing of the power-on reset circuit is shown in figure 19.2. As the power supply voltage rises, the capacitor, which is externally connected to the $\overline{\text{RES}}$ pin, is gradually charged through the on-chip pull-up resistor (Rp).
Section 21 Electrical Characteristics Table 21.3 Control Signal Timing	411	Modified

Item	Symbol	Applicable Pins	Test Condition	Values		
				Min.	Typ.	Max.
Oscillation stabilization time	trc	OSC1,OSC2	Ceramic resonator (V_{cc} = 2.2 V to 3.6 V)	—	20	45
			Ceramic resonator (Other than above)	—	80	—
			Crystal resonator (V_{cc} = 2.7 V to 3.6 V)	—	300	800
			Crystal resonator (V_{cc} = 2.2 V to 3.6 V)	—	600	1000

Item	Page	Revisions (See Manual for Details)
Table 21.14 Control Signal Timing	425	Modified

Item	Symbol	Applicable Pins	Test Condition	Values		
				Min.	Typ.	Max.
Oscillation stabilization time	trc	OSC1,OSC2	Ceramic resonator(VCC = 2.2 V to 3.6 V)	—	20	45
			Ceramic resonator(Other than above)	—	80	—
			Crystal resonator(VCC = 2.7 V to 3.6 V)	—	300	800
			Crystal resonator(VCC = 2.2 V to 3.6 V)	—	600	1000

Appendix A.1 Instruction List 2. Arithmetic Instructions	447	Modified

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes) #xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa		Operation	Condition Code I	H	N	Z	V	C	No. of States*1 Normal	Advanced
DAA		B	2	2								Rd8 decimal adjust → Rd8	—	*	↕	↕	*	↕		2

C. Product Part No. Lineup	491	The list is modified.
D. Package Dimensions Figure D.2 Package Dimensions (32P6U-A)	492	Added

Index

L

M

O

P

R

S

T

V

W

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8/38602R Group

Publication Date:	Rev.1.00, May 20, 2004
	Rev.3.00, May 15, 2007
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Customer Support Department
	Global Strategic Communication Div.
	Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8/38602R Group
Hardware Manual

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan