To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User's Manual

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

Renesas Electronics
www.renesas.com

Rev.3.00 2007.05

document, please confirm the latest product information with a Renesas sales office. Also, please pay and careful attention to additional and different information to be disclosed by Renesas such as that dis through our website. (http://www.renesas.com)

5. Renesas has used reasonable care in compiling the information included in this document, but Renesa assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the infor included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the informati light of the total system before deciding about the applicability of such information to the intended appli Renesas makes no representations, warranties or guaranties regarding the suitability of its products fo 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 o malfunction of which may cause a direct threat to human life or create a risk of human injury or which re especially high quality and reliability such as safety systems, or equipment or systems for transportatio traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea commu transmission. If you are considering the use of our products for such purposes, please contact a Renes sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth ab
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed
   (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 purchase elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless R Technology Corp., its affiliated companies and their officers, directors, and employees against any and damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with re 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 malfunc 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 spe 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 inj injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design 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 th evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final prod system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesa products are attached or affixed, the risk of accident such as swallowing by infants and small children is high. You should implement safety measures so that Renesas products may not be easily detached fro 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 writt approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in document, Renesas semiconductor products, or if you have any other inquiries.

RENESAS

are in their open states, intermediate levels are induced by noise in the vicinity, a 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 throughou chip and a low level is input on the reset pin. During the period where the states a undefined, the register settings and the output state of each pin are also undefined your system so that it does not malfunction because of processing while it is in th undefined state. For those products which have a reset function, reset the LSI imr 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 reg may have been be allocated to these addresses. Do not access these registers; the operation is not guaranteed if they are accessed.

- CPU and System-Control Modules
- On-Chip Peripheral Modules

The configuration of the functional description of each module differs according to t
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. Ea
includes notes in relation to the descriptions given, and usage notes are given, as require
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 ve
This does not include all of the revised contents. For details, see the actual locations in t
manual.

11. Index

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 o 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 in on the CPU, system control functions, peripheral functions, and electrical characterist

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 register. The addresses, bits, and initial values of the registers are summarized in secti List of Registers.

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

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

6. When the on-chip emulator is used, even though the on-chip oscillator is selected, co 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 we Please ensure you have the latest versions of all documents you rec http://www.renesas.com/

H8/38602R Group manuals:

Document Title	Docume
H8/38602R Group Hardware Manual	This mar
H8/300H Series Software Manual	REJ09B0

User's manuals for development tools:

Document Title	Docume
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	REJ10B0
Microcomputer Development Environment System H8S, H8/300 Series Simulator/Debugger User's Manual	ADE-702
H8S, H8/300 Series High-performance Embedded Workshop 3 Tutorial	REJ10B0
H8S, H8/300 Series High-performance Embedded Workshop 3 User's Manual	REJ10B0

Section 3 Exception Handling

Section 4 Clock Pulse Generators

Section 15 Synchronous Serial Communication Unit (SSU)

Appendix

Appendix

— 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 P
Electronics)*
— 10-bit A/D converter
— Comparators

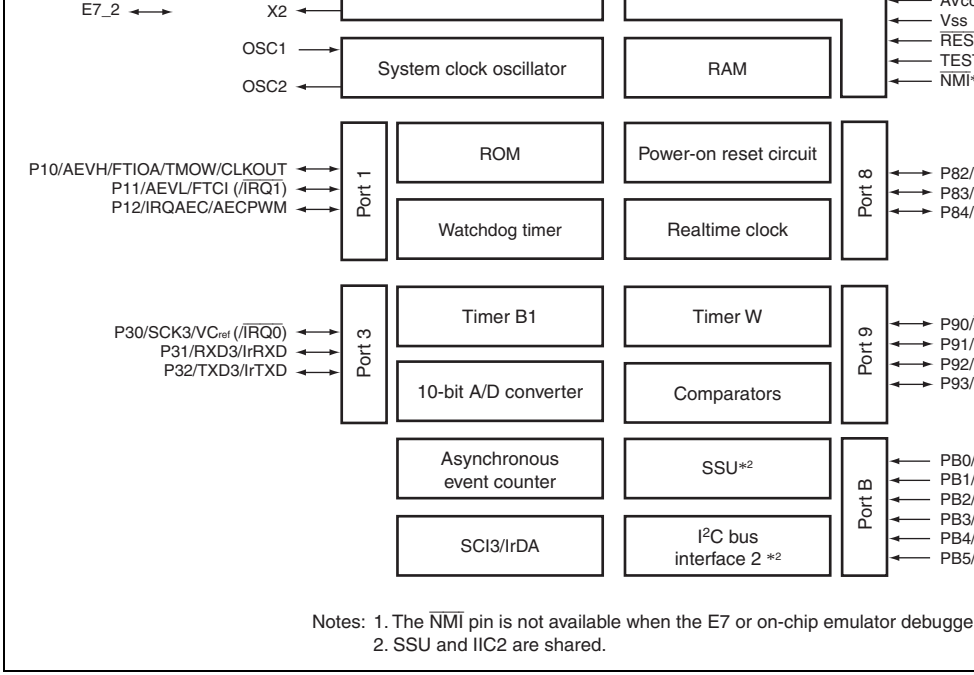
Note: * SSU and IIC2 are shared.

- 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.
  — Input-only pins: 6 input pins (also used as analog input pins)
- Supports various power-down states



Notes: 1. The NMI pin is not available when the E7 or on-chip emulator debugge
2. SSU and IIC2 are shared.

Figure 1.1 Internal Block Diagram of H8/38602R Group

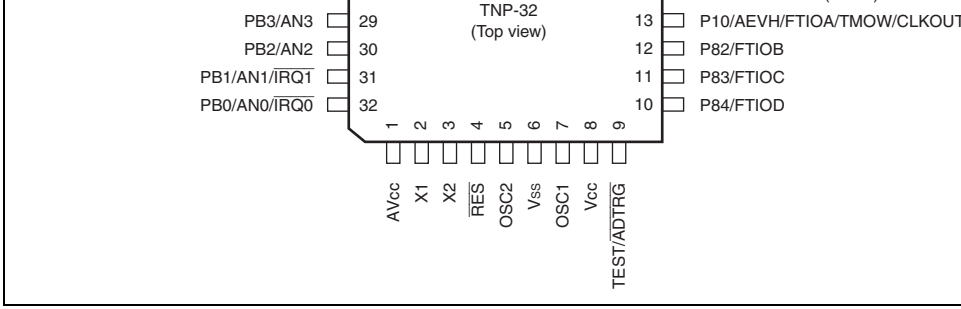


Figure 1.2 Pin Assignment of H8/38602R Group (TNP-32)

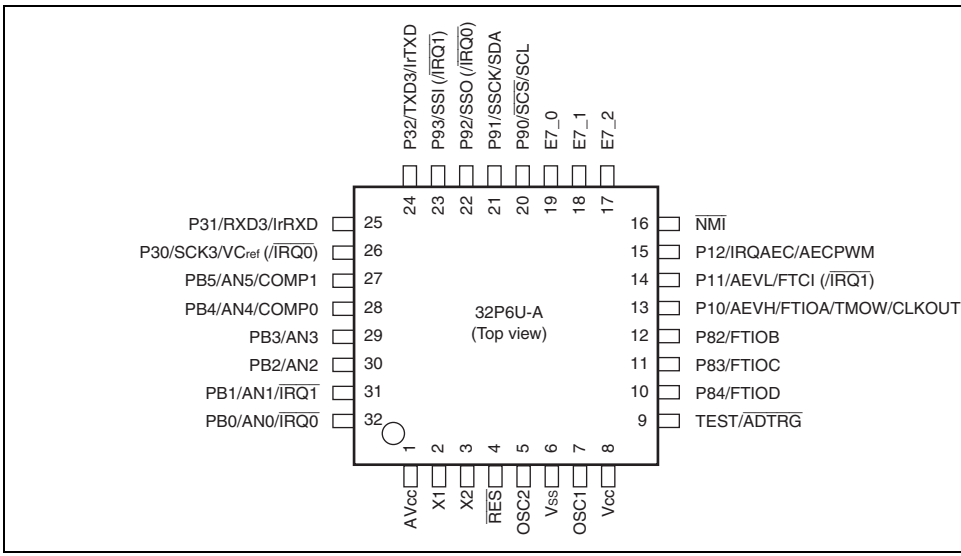


Figure 1.3 Pin Assignment of H8/38602R Group (32P6U-A)

				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 u input an external clock. See section 4, Clock Pulse Generators, for connection.
	OSC2	5	Output	
	X1	2	Input	These pins connect with a 32.768- or 38.4- crystal resonator for the subclock. See sect Clock Pulse Generators, for a typical conne
	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, t is reset.
	TEST	9	Input	Test pins. Also used as the $\overline{\text{ADTRG}}$ pin. W pin is not used as the $\overline{\text{ADTRG}}$ pin, users ca use this pin. Connect this pin to Vss. When is used as the $\overline{\text{ADTRG}}$ pin, see section 17.4 External Trigger Input Timing.

	IRQAEC	15	Input	Interrupt input pin for the asynchronous ev counter. This pin enables the asynchronous event
Timer W	FTCI	14	Input	External event input pin.
	FTIOA to FTIOD	13 to 10	I/O	Output compare output/input capture inpu output pins.
Asynchro-nous event counter (AEC)	AEVL	14	Input	Event input pins for input to the asynchron 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 communi-cation 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 format.
	TXD3/ IrTXD	24	Output	SCI3 data output pins or data output pins IrDA format.
Synchro-nous serial communi-cation unit (SSU)	$\overline{SCS}$	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.

				(PCR1).
	P30 to P32	26 to 24	I/O	3-bit I/O pins. Input or output can be design each bit by means of the port control regist (PCR3).
	P82 to P84	12 to 10	I/O	3-bit I/O pins. Input or output can be design each bit by means of the port control regist (PCR8).
	P90 to P93	20 to 23	I/O	4-bit I/O pins. Input or output can be design each bit by means of the port control regist (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 wh the on-chip oscillator is used. E7_2 is pulle down by a 100-kΩ resistance. For details, s section 4, Clock Pulse Generators.

- General-register architecture

  Sixteen 16-bit general registers also usable as sixteen 8-bit registers and eight 16-bit 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

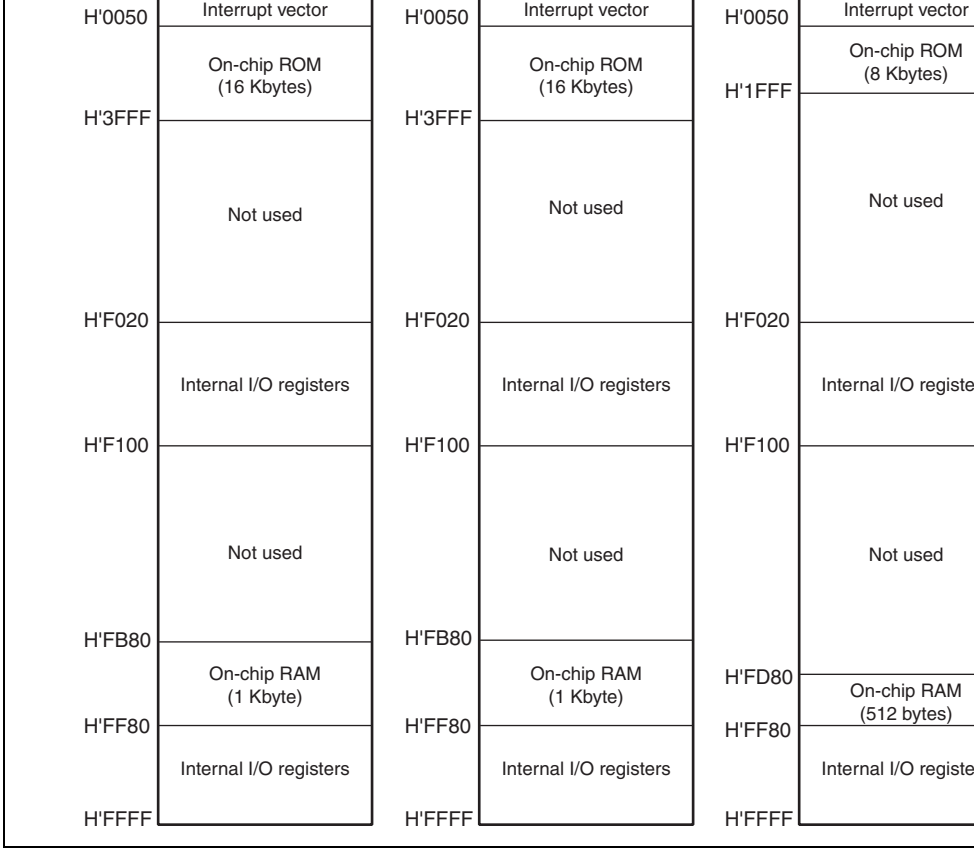
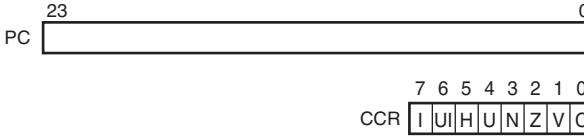


Figure 2.1 Memory Map

ER2	E2		R2H	R2L
ER3	E3		R3H	R3L
ER4	E4		R4H	R4L
ER5	E5		R5H	R5L
ER6	E6		R6H	R6L
ER7	E7	(SP)	R7H	R7L

Control Registers (CR)



[Legend]

SP:	Stack pointer	H:	Half-carry flag
PC:	Program counter	U:	User bit
CCR:	Condition-code register	N:	Negative flag
I:	Interrupt mask bit	Z:	Zero flag
UI:	User bit	V:	Overflow flag
		C:	Carry flag

Figure 2.2 CPU Registers

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

The usage of each register can be selected independently.

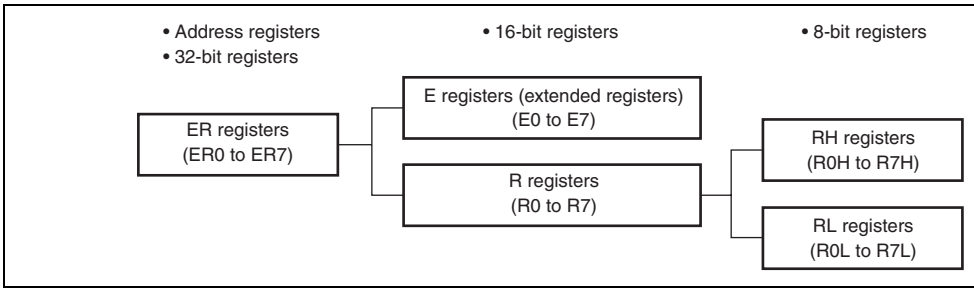


Figure 2.3 Usage of General Registers

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. Th
of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. (
instruction is fetched, the least significant PC bit is regarded as 0). The PC is initialized
start address is loaded by the vector address generated during reset exception-handling s

2.2.3 Condition-Code Register (CCR)

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

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

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

				there is a carry or borrow at bit 3, and cleared t otherwise. When the ADD.W, SUB.W, CMP.W NEG.W instruction is executed, the H flag is se there is a carry or borrow at bit 11, and clearec otherwise. When the ADD.L, SUB.L, CMP.L, o instruction is executed, the H flag is set to 1 if t carry or borrow at bit 27, and cleared to 0 othe
4	U	Undefined	R/W	User Bit Can be written and read by software using the STC, ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit of d sign bit.
2	Z	Undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to 0 indicate non-zero data.
1	V	Undefined	R/W	Overflow Flag Set to 1 when an arithmetic overflow occurs, a 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 c The carry flag is also used as a bit accumulato manipulation instructions.

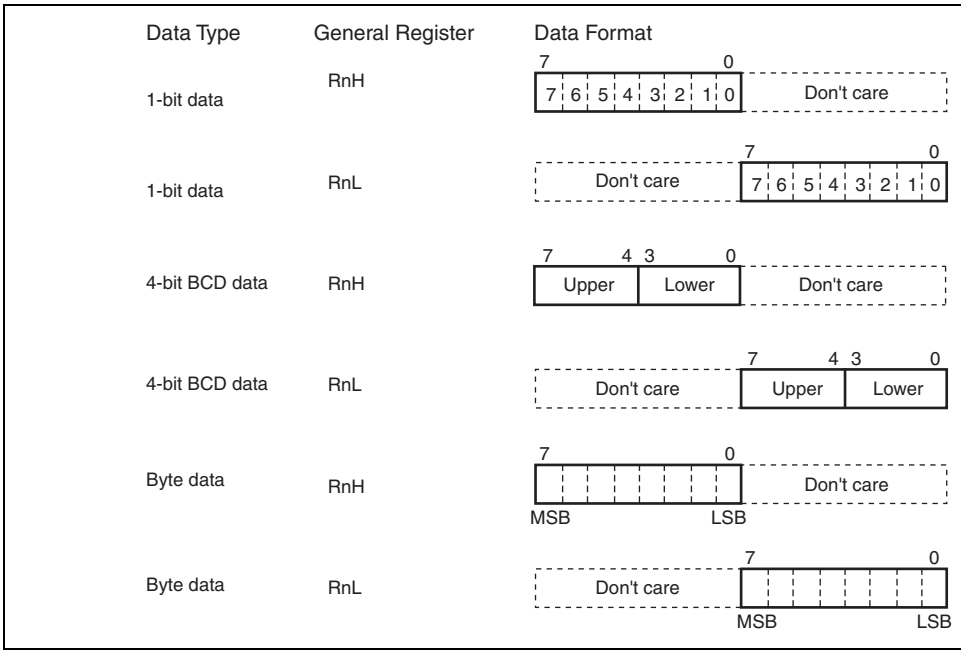


Figure 2.5 General Register Data Formats (1)

[Legend]

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

Figure 2.5 General Register Data Formats (2)

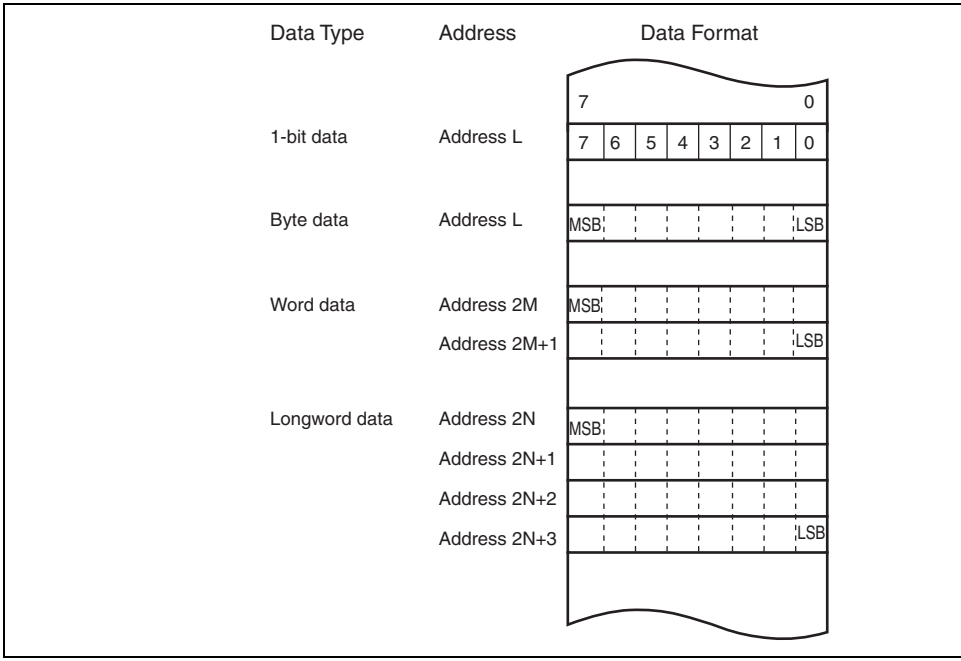


Figure 2.6 Memory Data Formats

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 regis
to R7, E0 to E7), and 32-bit registers/address register (ER0 to ER7).

		Pops a general register from the stack. POP.W Rn is identical MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+,
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is ident MOV.W Rn, @–SP. PUSH.L ERn is identical to MOV.L ERn,

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

INC DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (Byte op 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
DAA DAS	B	Rd (decimal adjust) → Rd Decimal-adjusts an addition or subtraction result in a general re referring to the CCR to produce 4-bit BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general registe 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: 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: eit bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 16-bit quotient and 16-bit remainder.

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

		Takes the two's complement (arithmetic complement) of data i general register.
EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the bits of a 32-bit register to longword size, by padding with zeros left.
EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the bits of a 32-bit register to longword size, by extending the sign

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

NOT	B/W/L	¬ (Rd) → (Rd) Takes the one's complement (logical complement) of general re 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

		Inverts a specified bit in a general register or memory operand number is specified by 3-bit immediate data or the lower three general register.
BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operand or clears the Z flag accordingly. The bit number is specified by 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 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 gen register or memory operand and stores the result in the carry 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 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 gene 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

		carry flag.
BILD	B	¬ (<bit-No.> of <EAd>) → C Transfers the inverse of a specified bit in a general register or m 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 regi memory operand.
BIST	B	¬ C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit in general register or memory operand. The bit number is specified by 3-bit immediate data.

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

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 ⊕ V = 0
BLT	Less than	N ⊕ V = 1
BGT	Greater than	Z∨(N ⊕ V) = 0
BLE	Less or equal	Z∨(N ⊕ 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.

		code register size is one byte, but in transfer to memory, data is 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

else next;

Transfers a data block. Starting from the address set in ER5, t
data for the number of bytes set in R4L or R4 to the address lo
in ER6.

Execution of the next instruction begins as soon as the transfe
completed.

Some instructions have two operation fields.

- Register Field

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

- Effective Address Extension

  8, 16, or 32 bits specifying immediate data, an absolute address, or a displacement. A 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.

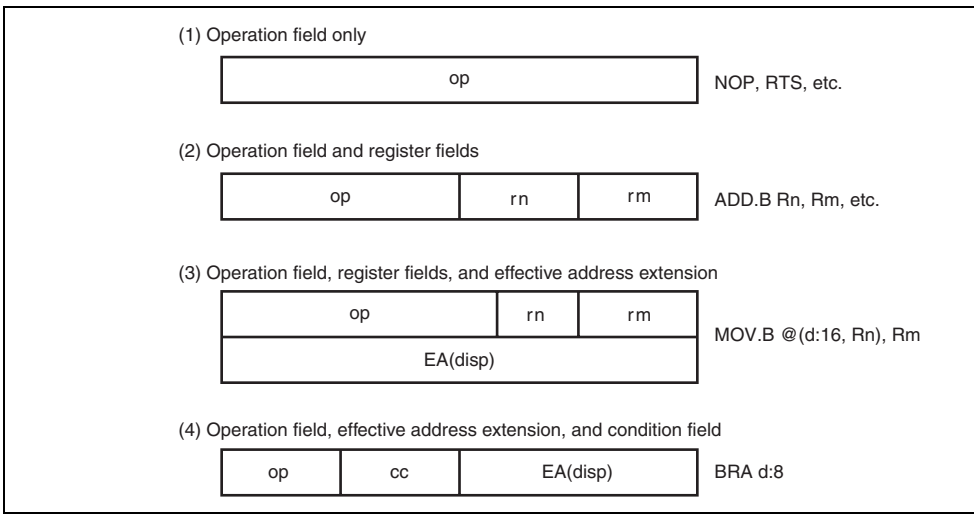


Figure 2.7 Instruction Formats

Modes.

Arithmetic and logic instructions can use the register direct and immediate modes. Data
instructions can use all addressing modes except program-counter relative and memory
Bit-manipulation instructions use register direct, register indirect, or the absolute addres
(@aa:8) to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST ins
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

A 16-bit or 24-bit displacement contained in the instruction is added to an address registe
specified by the register field of the instruction, and the lower 24 bits of the sum the addr
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
  of which contains the address of a memory operand. After the operand is accessed, 1,
  added to the address register contents (32 bits) and the sum is stored in the address reg
  The value added is 1 for byte access, 2 for word access, or 4 for longword access. For
  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 regis
  in the instruction code, and the lower 24 bits of the result is the address of a memory o
  The result is also stored in the address register. The value subtracted is 1 for byte acce
  word access, or 4 for longword access. For the word or longword access, the register v
  should be even.

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

The instruction code contains the absolute address of a memory operand. The absolute ad
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 1
absolute address the upper 8 bits are a sign extension. A 24-bit absolute address can acce
entire address space.

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 dat
operand.

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some
manipulation instructions contain 3-bit immediate data in the instruction code, specifyin
number. The TRAPA instruction contains 2-bit immediate data in its instruction code, sp
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 t
instruction is sign-extended and added to the 24-bit PC contents to generate a branch ad
PC value to which the displacement is added is the address of the first byte of the next in
so the possible branching range is –126 to +128 bytes (–63 to +64 words) or –32766 to +
bytes (–16383 to +16384 words) from the branch instruction. The resulting value should
even number.

Memory Indirect—@@aa:8

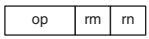
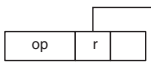
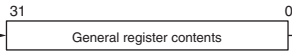
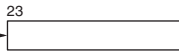
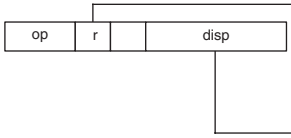
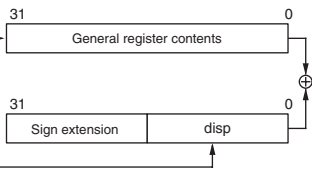
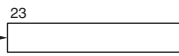
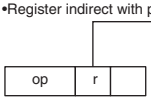
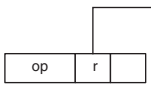
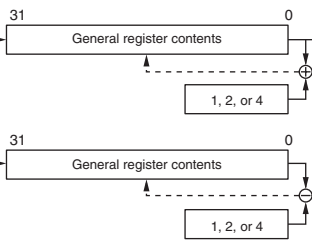
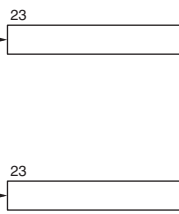
This mode can be used by the JMP and JSR instructions. The instruction code contains a
absolute address specifying a memory operand. This memory operand contains a branch
The memory operand is accessed by longword access. The first byte of the memory oper
ignored, generating a 24-bit branch address. Figure 2.8 shows how to specify branch add
memory indirect mode. The upper bits of the absolute address are all assumed to be 0, so
address range is 0 to 255 (H'0000 to H'00FF).

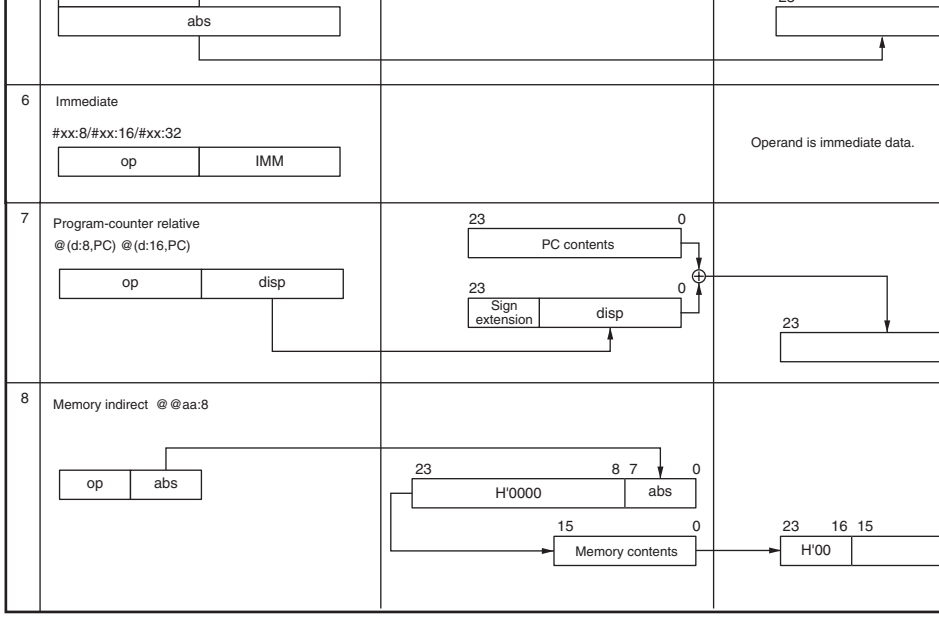
Note that the first part of the address range is also the exception vector area.

2.5.2 Effective Address Calculation

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

Table 2.12 Effective Address Calculation (1)

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



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

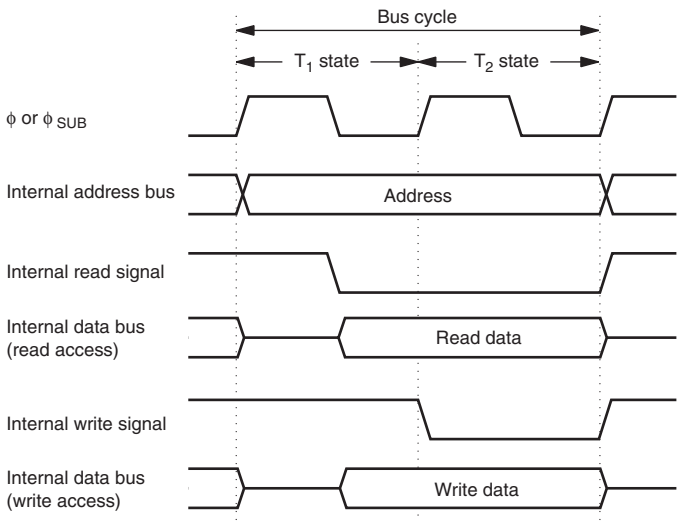


Figure 2.9 On-Chip Memory Access Cycle

module.

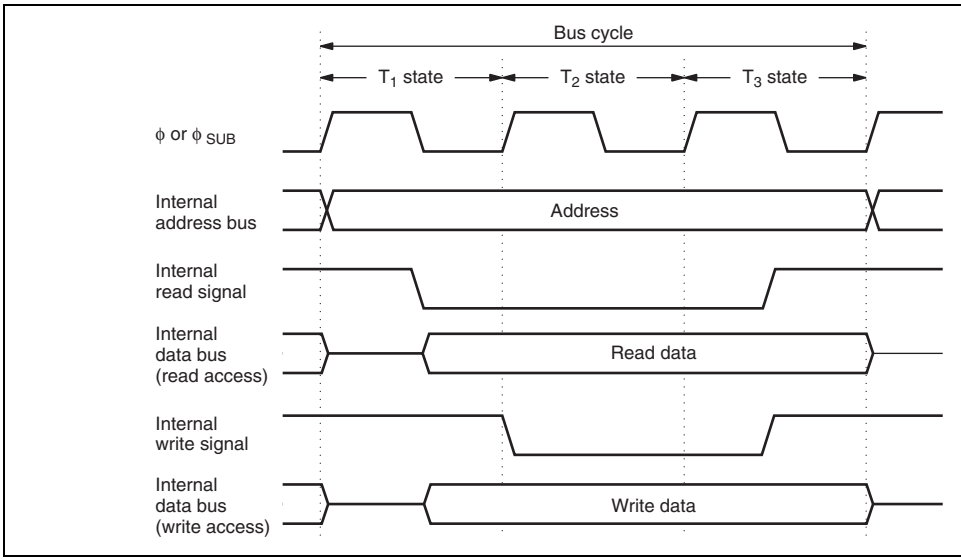


Figure 2.10 On-Chip Peripheral Module Access Cycle (3-State Access)

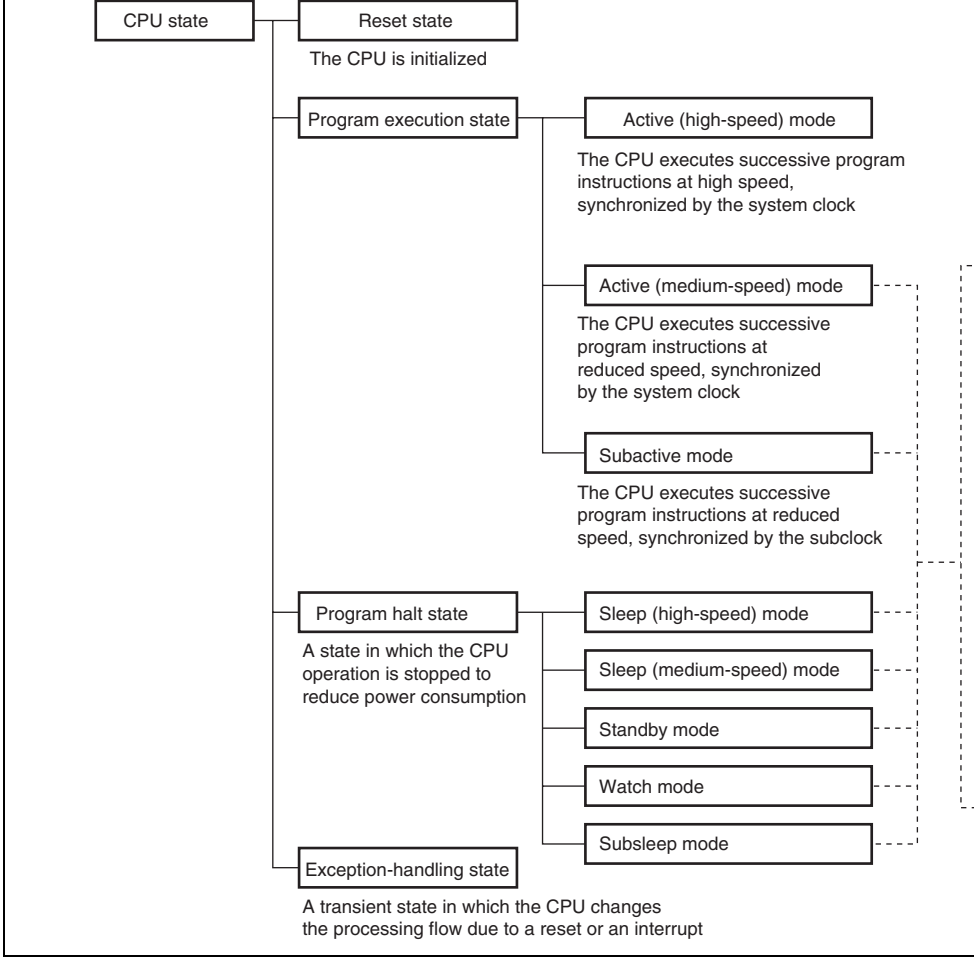


Figure 2.11 CPU Operating States

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
I/O registers areas available to the user. When data is transferred from CPU to empty ar
transferred data will be lost. This action may also cause the CPU to malfunction. When
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
R4L, which starts from the address indicated by R5, to the address indicated by R6. Set
and R6 so that the end address of the destination address (value of R6 + R4L or R6 + R4
exceed H'FFFF (the value of R6 must not change from H'FFFF to H'0000 during execut

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 address. When a bit-manipulation instruction accesses the timer load register and timer co a reloadable timer, since these two registers share the same address, the following operati 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 tim register. As a result, bits other than the intended bit in the timer counter may be modified modified value may be written to the timer load register.

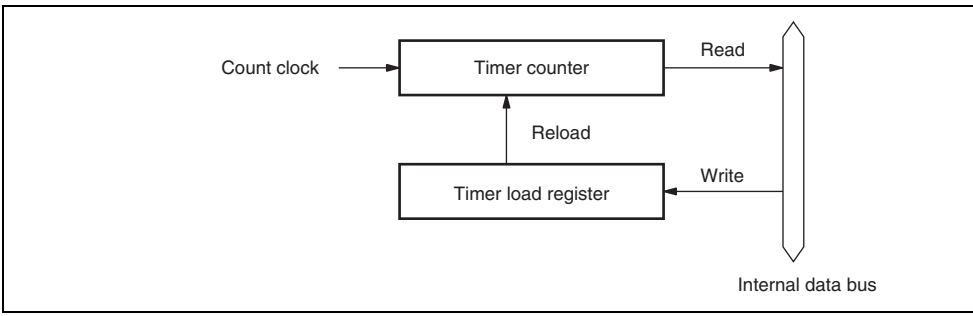


Figure 2.13 Example of Timer Configuration with Two Registers Allocated to Same Address

PCR5	0	0	1	1	1	1	1
PDR5	1	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**
Input/output	Input	Input	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1
PDR5	0	1	0	0	0	0	0

- 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-l input).
   P55 to P50 are output pins, so the CPU reads the value in PDR5. In this example PD 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.

	P57	P56	P55	P54	P53	P52	P51
Input/output	Input	Input	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1
PDR5	1	0	0	0	0	0	0
RAM0	1	0	0	0	0	0	0

- BSET instruction executed

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

The BSET instruction is executed designating the 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
Input/output	Input	Input	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1
PDR5	1	0	0	0	0	0	0
RAM0	1	0	0	0	0	0	0

Input/output	Input	Input	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1
PDR5	1	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**
Input/output	Output	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level
PCR5	1	1	1	1	1	1	1
PDR5	1	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 register, the CPU reads a value of H'FF, even though the PCR5 value is actually H'3
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. Ho bits 7 and 6 in PCR5 change to 1, so that P57 and P56 change from input pins to outp To prevent this problem, store a copy of the PDR5 data in a work area in memory an manipulate data of the bit in the work area, then write this data to PDR5.

PDR5	1	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1

- BCLR instruction executed

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

The BCLR instructions executed for the PCR5 wo
(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**
Input/output	Input	Input	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1
PDR5	1	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1

Exception handling starts when a trap instruction (TRAPA) is executed. A vector ad
corresponding to a vector number from 0 to 3 which are specified in the instruction c
generated. Exception handling can be executed at all times in the program execution
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
  and kept pending while the I bit is set to 1. Exception handling starts when the curre
  instruction or exception handling ends, if an interrupt is requested.

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

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

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

	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 LS the reset state. The internal state of the CPU and the registers of the on-chip peripheral m 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 characterist

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 transfer 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.

(1) Reset exception handling vector address (H'0000)
(2) Program start address
(3) Initial program instruction

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 initi
and CCR will not be pushed onto the stack correctly, resulting in program runaway. To
this, immediately after reset exception handling all interrupts are masked. For this reaso
initial program instruction is always executed immediately after a reset. This instruction
initialize the stack pointer (e.g. MOV.L #xx: 32, SP).

$\overline{\text{IRQ1}}$	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)

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 in 1: Detects a rising edge of the $\overline{\text{ADTRG}}$ pin in
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 inpu 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 inpu 1: Detects a rising edge of the $\overline{\text{IRQ0}}$ pin input

				The write value should always be 0.
2	IENEC2	0	R/W	IRQAEC Interrupt Request Enable The IRQAEC interrupt request is enabled when t set to 1.
1	IEN1	0	R/W	IRQ1 Interrupt Request Enable The IRQ1 interrupt request is enabled when this to 1.
0	IEN0	0	R/W	IRQ0 Interrupt Request Enable The IRQ0 interrupt request is enabled when this to 1.

				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 set to 1.
1	—	0	—	Reserved The write value should always be 0.
0	IENEC	0	R/W	Asynchronous Event Counter Interrupt Reques The asynchronous event counter interrupt requ enabled when this bit is set to 1.

				When the P12 pin is set to the IRQAEC/AECPW and the specified edge is detected as the pin sta
				[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 p 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 p the specified edge is detected
				[Clearing condition]
				When 0 is written to this bit

Note: * Only 0 can be written to clear the 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 [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 Reques [Setting condition] When the asynchronous event counter overflow [Clearing condition] When 0 is written to this bit

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

at a rising edge or a falling edge on the 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 gen
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
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 accep
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 (P
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 interru
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 correspond
IRR1 is set to 1 and an interrupt request is generated. For details, see section 13, Asynchr
Event Counter (AEC).

NMI interrupts are accepted at all times except in the reset state. In the case of IRQ inter on-chip peripheral module interrupts, an enable bit is provided for each interrupt. Clearin enable bit to 0 disables the corresponding interrupt request. Interrupt sources for which t 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 u interrupt acceptance.

Interrupt operation is described as follows.

1. If an interrupt source whose interrupt enable register bit is set to 1 occurs, an interrup is sent to the interrupt controller.
2. When the interrupt controller receives an interrupt request, it sets the interrupt reques
3. From among the interrupts with interrupt request flags set to 1, the interrupt controlle the interrupt request with the highest priority and holds the others pending (see table
4. The interrupt controller checks the I bit of CCR. If the I bit is 0, the selected interrup 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 comp both PC and CCR are pushed onto the stack. The state of the stack at this time is sho figure 3.5. The PC value pushed onto the stack is the address of the first instruction t 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 inter handling routine located at the address indicated by the contents of the vector addres executed.

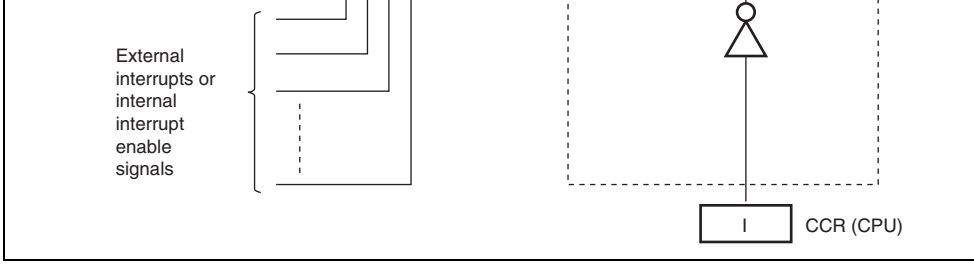
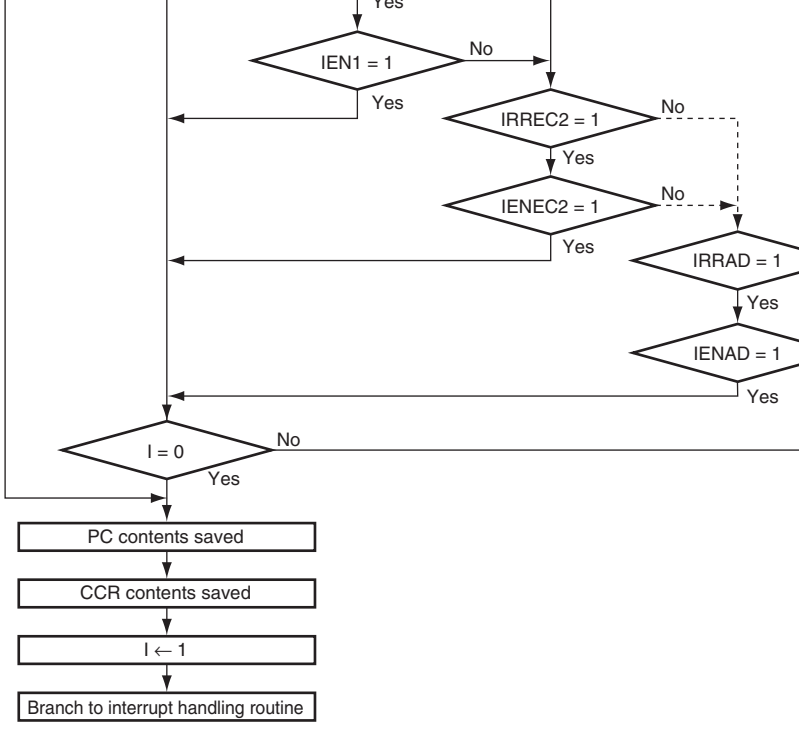


Figure 3.2 Block Diagram of Interrupt Controller



[Legend]
PC: Program counter
CCR: Condition code register
I: I bit of CCR

Figure 3.3 Flow up to Interrupt Acceptance

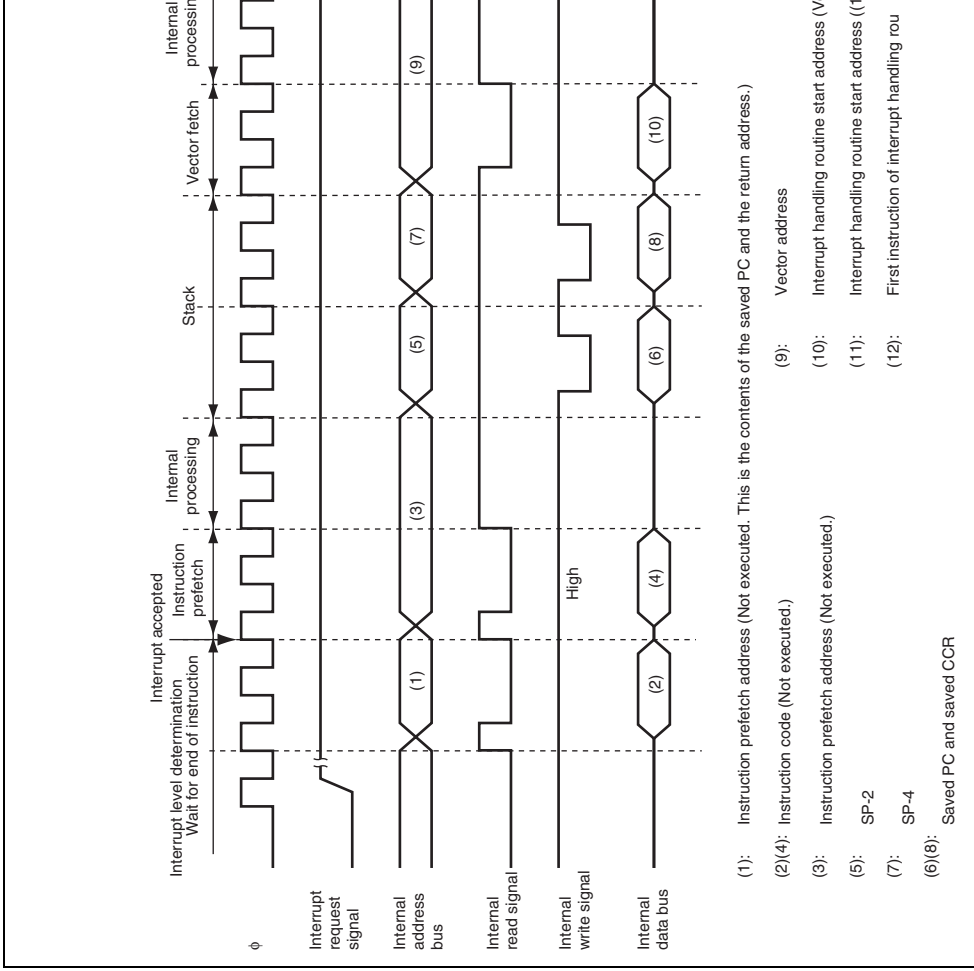


Figure 3.4 Interrupt Exception Handling Sequence

[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 f 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.

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

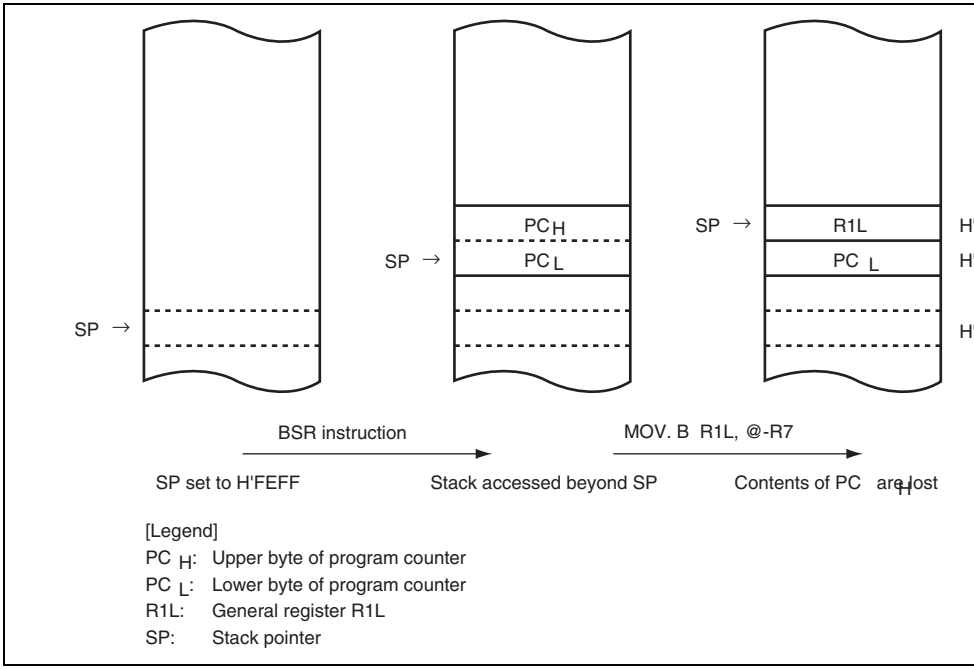


Figure 3.6 Operation when Odd Address is Set in SP

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

if a valid edge has not arrived on the selected IRQAEC or IECPWM (PWM output for t
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 int
request flag. This procedure also applies to AEGSR setting.

When switching a pin function, mask the interrupt before setting the bit in PFCR and PM
AEGSR). After accessing PFCR and PMRB (or AEGSR), execute at least one instructio
NOP), then clear the interrupt request flag from 1 to 0. If the instruction to clear the flag
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 function
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 comp

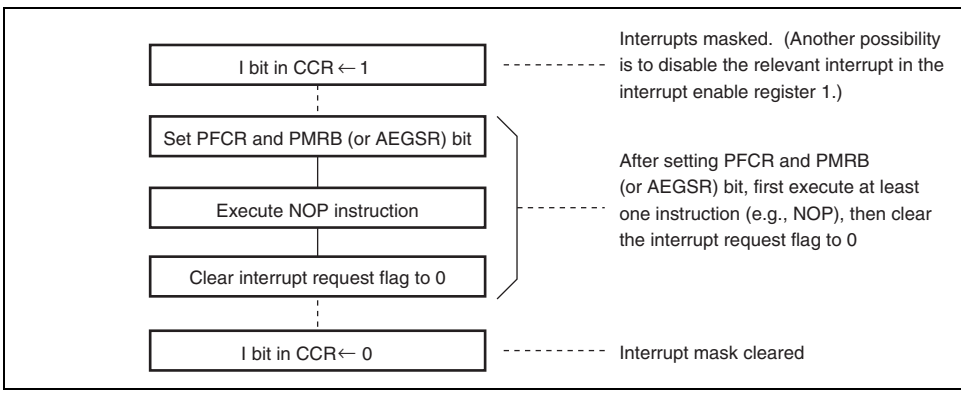


Figure 3.7 PFCR and PMRB (or AEGSR) Setting and Interrupt Request Flag Clearing Procedure

```
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 I (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 AN instruction is executing.

  The IRQ0 interrupt is disabled because, although the original objective is clearing IRI 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 ef after execution of the instruction. When an interrupt enable bit is cleared to 0 by an instru such as BCLR or MOV, and if an interrupt is generated during execution of the instructio 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. W interrupt source flag is cleared to 0, the interrupt concerned will be ignored.

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

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

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

Therefore, if an NMI interrupt is generated during execution of an EEPMOV.W instruct 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 th flag register is cleared, the interrupt request should be masked (I bit = 1). If the above op executed while the I bit is 0 and conflict between the instruction execution and the interr request generation occurs, exception handling, which corresponds to the interrupt reques generated after instruction execution of the above operation is completed, is executed.

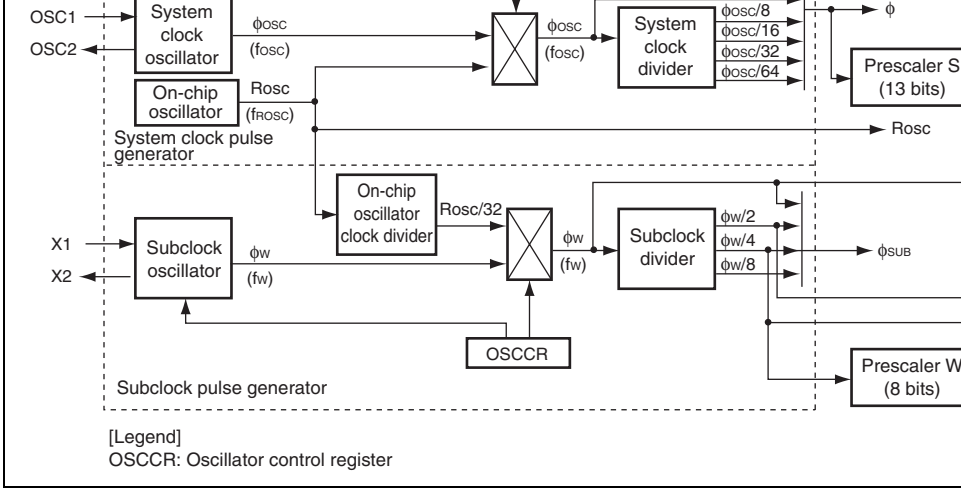


Figure 4.1 Block Diagram of Clock Pulse Generators

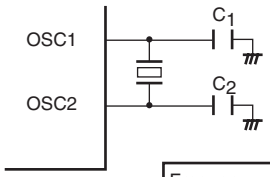
The reference clock signals that drive the CPU and on-chip peripheral modules are ϕ and system clock is divided by prescaler S to become a clock signal from $\phi/8192$ to $\phi/2$. $\phi_w/4$ 1/4th of the watch clock ϕ_w, is divided by prescaler W to become a clock signal from ϕ_w $\phi_w/8$. Both the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

7	SUBSTP	0	R/W	Subclock Oscillator Control Controls start and stop of the subclock oscillato 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 resistan system clock oscillator is used when an extern is input or when the on-chip oscillator is used. After setting this bit in the state in which an ext clock is input or the on-chip oscillator is used, temporarily transit to standby mode, watch mo subactive mode. The setting of whether the fee resistance in the system clock oscillator is use takes effect when standby mode, watch mode, 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 when the subclock is not being used.

				0: System clock oscillator operates 1: On-chip oscillator operates (system clock o 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.

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

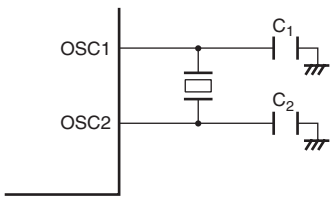


Frequency	Manufacturer	Product Type	C1, C2 Recommenda
4.194304 MHz	NIHON DEMPA KOGYO CO.,LTD.	NR-18	12 pF ±20%

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 conne
refer to section 4.5.2, Notes on Board Design.



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

Figure 4.3 Typical Connection to Ceramic Resonator

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

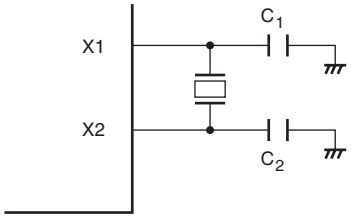
When the on-chip oscillator is selected, a resonator no longer needs to be connected to th and OSC2 pins. In such a case, fix the OSC1 pin to GND or leave it open, and leave the open.

Notes: 1. When programming or erasing the flash memory, such as performing on-boa programming, the system clock oscillator must be selected. When the on-chi is used, even though the on-chip oscillator is selected, connect a resonator or external clock.

2. When the on-chip debugger is connected, the value of the resistor should be l When not connected, it is specified according to the selected oscillator.

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

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



Frequency	Manufacturer	Products Name	Equivalent Series Resistanc
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 crystal resonator manufacture and connect it under the optimum condition. Even when resonator listed above or the equivalent is used, as the oscillation characteristics depen 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 t state (the $\overline{\text{RES}}$ pin is driven from low to high).

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, a
in figure 4.7.

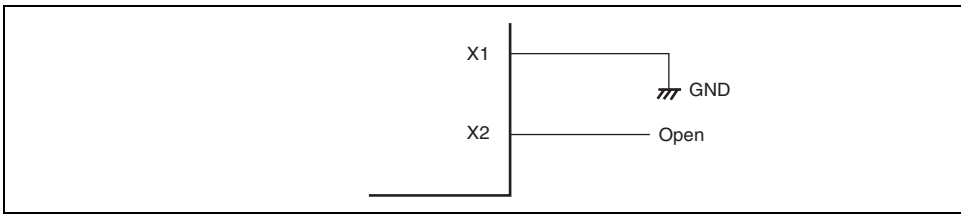


Figure 4.7 Pin Connection when not Using Subclock

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 oscil
selected, a resonator no longer needs to be connected to the X1 and X2 pins. In such a cas
X1 pin at GND.

4.4.1 Prescaler S

Prescaler S is a 13-bit counter using the system clock (ϕ) as its input clock. A divided ou
used as an internal clock of an on-chip peripheral module. Prescaler S is initialized to H'
reset, and starts counting up on exit from the reset state. In standby mode, watch mode, s
mode, and subsleep mode, prescaler S stops and is initialized to H'0000. The CPU canno
from or write to prescaler S.

The output from prescaler S is shared by the on-chip peripheral modules. In active (med
speed) mode and sleep (medium-speed) mode, the clock input to prescaler S is determin
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 in
A divided output is used as an internal clock of an on-chip peripheral module. Prescaler
initialized to H'00 at a reset, and starts counting up on exit from the reset state. In standb
prescaler W is halted. Even when transiting to watch mode, subactive mode, and subslee
prescaler W continues operation.

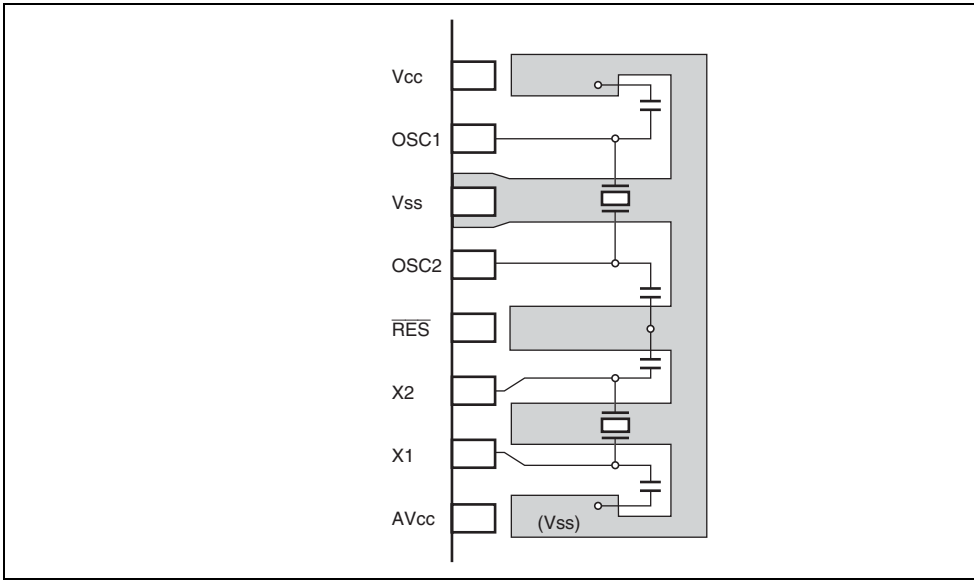


Figure 4.9 Example of Crystal and Ceramic Resonator Assignment

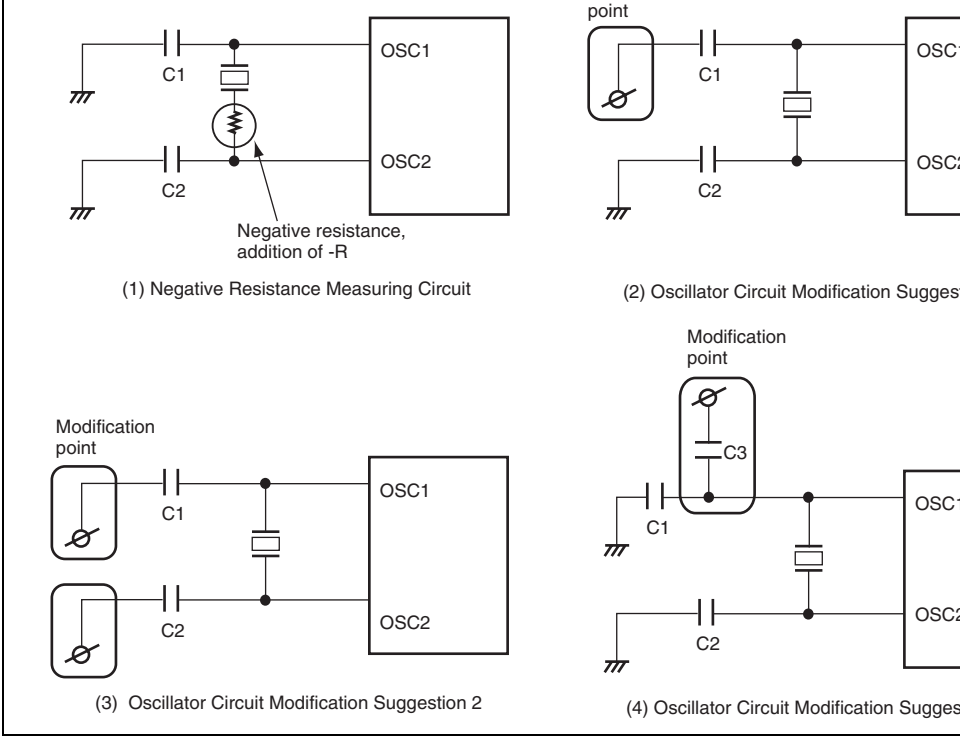


Figure 4.10 Negative Resistance Measurement and Circuit Modification Sugg

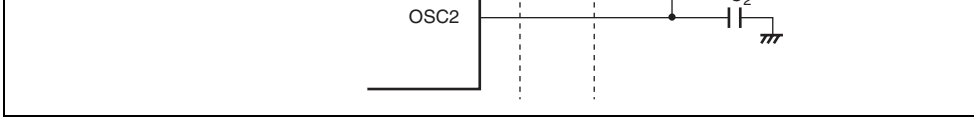


Figure 4.11 Example of Incorrect Board Design

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

4.5.3 Definition of Oscillation Stabilization Wait Time

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

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

(1) Oscillation Start Time

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

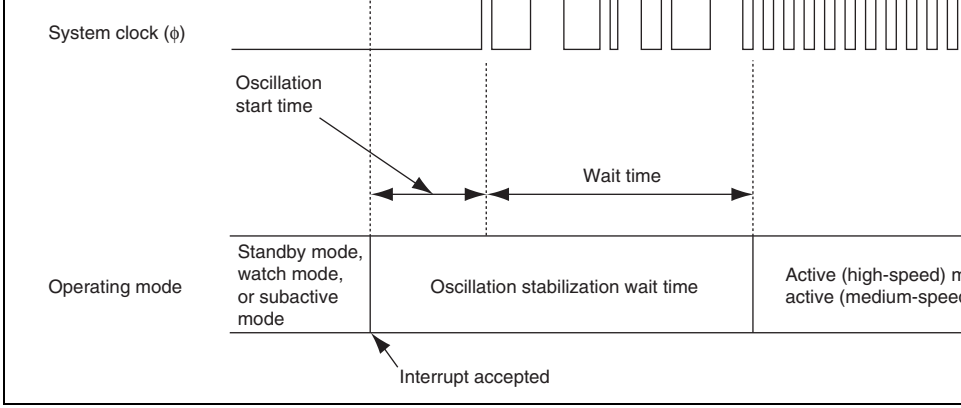


Figure 4.12 Oscillation Stabilization Wait Time

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

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

the power supply potential. In this state, the oscillation waveform may be disrupted, leadi
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 (ST
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 th
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
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 capacit
which is externally connected to the $\overline{\text{RES}}$ pin. The external capacitor capacitance should
adjusted to secure the oscillation stabilization time before reset clearing. For details, refer
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 vo
and temperature conditions. Therefore, when using the on-chip emulator, the resonator m
connected to the OSC1 and OSC2 pins or an external clock must be supplied. In this case
chip oscillator is used for user program execution, and the system clock is used for flash
programming/erasing. This control is handled when the E7_2 pin is fixed to high level du
reset by the on-chip emulator.

- Subactive mode

  The CPU and all on-chip peripheral modules are operable on the subclock. The subc
  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 sy
  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 su
  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-c
  peripheral modules that are not used in module units.

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

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 SLEEP instruction. 0: A transition is made to sleep mode or subsle mode. 1: A transition is made to standby mode or wat For details, see table 5.2.
6	STS2	0	R/W	Standby Timer Select 2 to 0
5	STS1	0	R/W	Designate the time the CPU and peripheral mo
4	STS0	0	R/W	wait for stable clock operation after exiting from mode, subactive mode, or watch mode to activ or sleep mode due to an interrupt. The designa should be made according to the operating fre so that the waiting time is at least equal to the oscillation stabilization time. The relationship b the specified value and the number of wait stat shown in table 5.1. When an external clock is to be used, the mini value (STS2 = 1, STS1 = 1, and STS0 = 1) is recommended. When the on-chip oscillator is t used, the minimum value (STS2 = 1, STS1 = 1 STS0 = 1) is recommended. If a setting other t recommended value is made, operation may s before the end of the waiting time.

0	MA0	1	R/W	Select the operating clock frequency in active speed) mode and sleep (medium-speed) mod MA1 and MA0 bits should be written to in acti 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
0	0	0	8,192 states	819.2	1,024.0*[1]	1,365.3*[1]	1,638.4	1953.3	2,730.7
		1	16,384 states	1,638.4	2,048.0	2,730.7	3,276.8	3906.5	5,461.3
	1	0	1,024 states	102.4	128.0	170.7	204.8	244.2	341.3
		1	2,048 states	204.8	256.0	341.3	409.6	488.3	682.7
1	0	0	4,096 states	409.6	512.0	682.7*[1]	819.2*[1]	976.6	1,365.3
		1	256 states	25.6	32.0	42.7*[2]	51.2*[2]	61.0	85.3*[2]
	1	0	512 states	51.2	64.0*[2]	85.3*[2]	102.4	122.1	170.7
		1	16 states	1.6	2.0	2.7	3.2	3.8	5.3

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

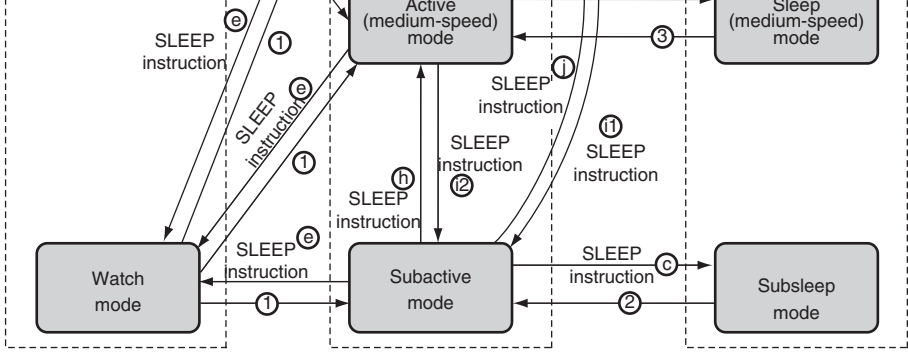
				The subclock pulse generator generates the w clock signal (ϕ_w) and the system clock pulse ge generates the oscillator clock (ϕ_{OSC}). This bit se sampling frequency of ϕ_{OSC} when ϕ_w is sampled system clock is used, clear this bit to 0.When t chip oscillator is selected, set this bit to 1. 0: Sampling 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 mad the SLEEP instruction is executed with bits SS 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 selects active (high-speed) or active (medium- 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 subacti subsleep modes. The operating clock frequenc 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

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

				The IIC2 enters standby mode when this bit is to 0.
4	SSUCKSTP	0	R/W	SSU Module Standby The SSU enters standby mode when this bit is to 0.
3	AECCKSTP	0	R/W	Asynchronous Event Counter Module Standby The asynchronous event counter enters standb when this bit is cleared to 0.
2	WDCKSTP	1	R/W*[3]	Watchdog Timer Module Standby The watchdog timer enters standby mode whe is cleared to 0.
1	COMPCKSTP	0	R/W	Comparator Module Standby The comparators enter standby mode when thi cleared to 0.
0	—	0	—	Reserved This bit is always read as 0 and cannot be mod

Notes: 1. When the SCI3 module standby is set, all registers in the SCI3 enter the reset
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 wh WDON bit is set to 1 (while the watchdog timer is operating), this bit is cleared However, the watchdog timer does not enter module standby mode and contin operating. When the WDON bit is cleared to 0 by software, this bit is valid and watchdog timer enters module standby mode.

is executed and a mode to return by an interrupt. Table 5.5 shows the internal states of th
each mode.



→ : Transition is made after exception handling is executed.

Power-down modes

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, W AEC, and timer B1
②	All interrupts except IIC2
③	All interrupts
④	NMI, IRQ0, IRQ1, IRQAEC, COMP, WDT, a 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

	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-spe
	0	1	0	x	0	Sleep (medium-speed) mode	Active (medium-
	0	0	1	0	0	Standby mode	Active (high-spe
	0	1	1	0	0	Standby mode	Active (medium-
	0	0	1	1	0	Watch mode	Active (high-spe
	0	1	1	1	0	Watch mode	Active (medium-
	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-spe
	0	1	1	1	0	Watch mode	Active (medium-
	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.

	Registers							
	I/O							
External interrupts	NMI	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
	Timer W					Retained	Functions/ Retained*3	Functions/ Retained*3
	WDT					Functions/ Retained*5	Functions/ Retained*5	Functions/ Retained*5
	RTC					Functions/ Retained*6	Functions/ Retained*6	Functions/ Retained*6
	Asynchro-nous event counter					Functions	Functions	Functions
	SCI3/ IrDA					Reset	Functions/ Retained*7	Functions/ Retained*7
	IIC2					Retained	Retained	Retained
	SSU					Retained	Functions/ Retained*8	Functions/ Retained*8
	A/D					Retained	Functions/ Retained*9	Functions/ Retained*9
	Comparator					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 reta otherwise.

5.2.1 Sleep Mode

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

Sleep mode is cleared by an interrupt. When an interrupt is requested, sleep mode is clea interrupt exception handling starts. Sleep mode is not cleared if the I bit in CCR is set to requested interrupt is disabled by the interrupt enable bit. After sleep mode is cleared, a is made from sleep (high-speed) mode to active (high-speed) mode or from sleep (mediu 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 interrupt request signal is synchronous with the system clock, the maximum time of 2/ϕ delayed from the point at which an interrupt request signal occurs until the interrupt exc handling is started.

to active (high-speed) or active (medium-speed) mode according to the MSON bit in SYS
Standby mode is not cleared if the I bit in CCR is set to 1 or the requested interrupt is disa
the interrupt enable bit.

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

5.2.3 Watch Mode

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

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

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

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

5.2.5 Subactive Mode

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

Subactive mode is cleared by the SLEEP instruction. When subactive mode is cleared, a to subsleep mode, active mode, or watch mode is made, depending on the combination o 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 generated by the $\overline{\text{RES}}$ pin, it must be kept low until the system clock oscillator output st and the t_{REL} period has elapsed. The CPU starts reset exception handling when the $\overline{\text{RES}}$ p driven high.

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

goes low, the CPU goes into the reset state and active (medium-sleep) mode is cleared.

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

When a SLEEP instruction is executed in active (high-speed) mode while the SSBY and bits in SYSCR1 are cleared to 0 and the MSON and DTON bits in SYSCR2 are set to 1, 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 (the direct transition time) is calculated by equation (1).

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

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

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

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

exception handling execution states) × (tsubcyc after transition).

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

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

When a SLEEP instruction is executed in active (medium-speed) mode while the SSBY a
LSON bits in SYSCR1 are cleared to 0, the MSON bit in SYSCR2 is cleared to 0, and the
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 h
(the direct transition time) is calculated by equation (3).

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

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

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

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

exception handling execution states) × (tsubcyc after transition)

Example: When ϕosc/8 and ϕw/8 are selected as the CPU operating clock before 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 b SYSCR1 are set to 1, the LSON bit in SYSCR1 is cleared to 0, the MSON bit in SYSCR cleared to 0, and the DTON bit in SYSCR2 is set to 1, a transition is made directly to ac speed) mode via watch mode after the waiting time set in bits STS2 to STS0 in SYSCR1 elapsed.

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

Direct transition time = {(Number of SLEEP instruction execution states) + (Number of processing states)} × (tsubcyc before transition) + (Wait time s STS2 to STS0) + (Number of interrupt exception handling exec states) × (tcyc after transition)……

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

processing states)} × (tsubcyc before transition) + (Wait time set
STS2 to STS0) + (Number of interrupt exception handling execu
states) × (tcyc after transition)……..……

Example: When ϕw/8 and ϕosc/8 are selected as the CPU operating clock before
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.

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 I
  Input Signal Changes before/after Standby Mode.

The on-chip oscillator can be used as the clock source for the watchdog timer (WDT), sub
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
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 circui
in standby mode and operates in other modes.

When the on-chip oscillator is used only as the clock source for the system clock generati
circuit, it operates in active and sleep modes but halts the operation in subactive, subsleep
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 (W
selects the on-chip oscillator as the clock source for the initial value.

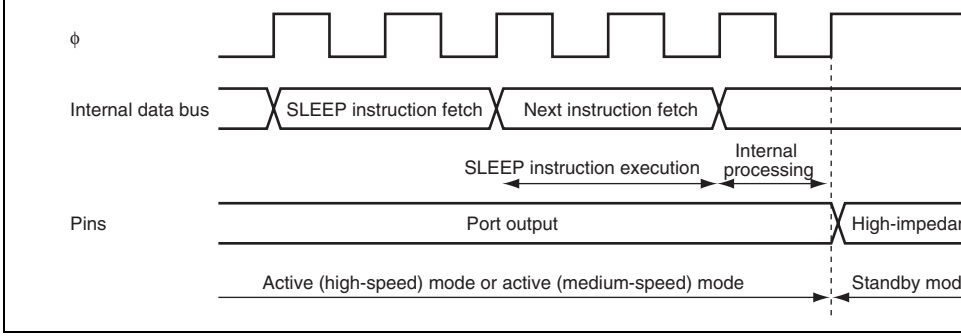


Figure 5.2 Standby Mode Transition and Pin States

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 M

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

least 2 t_{cyc} or 2 t_{subcyc} are necessary before a transition is made to standby mode or watch m
shown in "Capture possible: case 1."

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

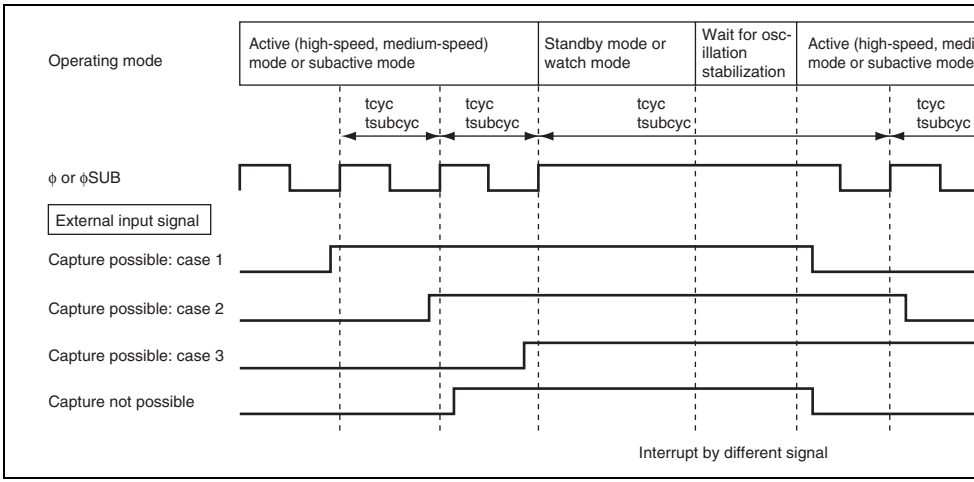


Figure 5.3 External Input Signal Capture when Signal Changes before/after Standby Mode or Watch Mode

(4) Input Pins to which these Notes Apply

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

On-board programming/erasure can be done in boot mode, in which the boot progra into this LSI is started to erase or program of the entire flash memory. In normal use 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 ma 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 re 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 inde when not used. (For details, refer to section 5.4, Module Standby Function.) Howeve using the on-chip emulator debugger, set the FROMCKSTP bit in clock halt register

Note: The system clock oscillator must be used when programming or erasing the flas

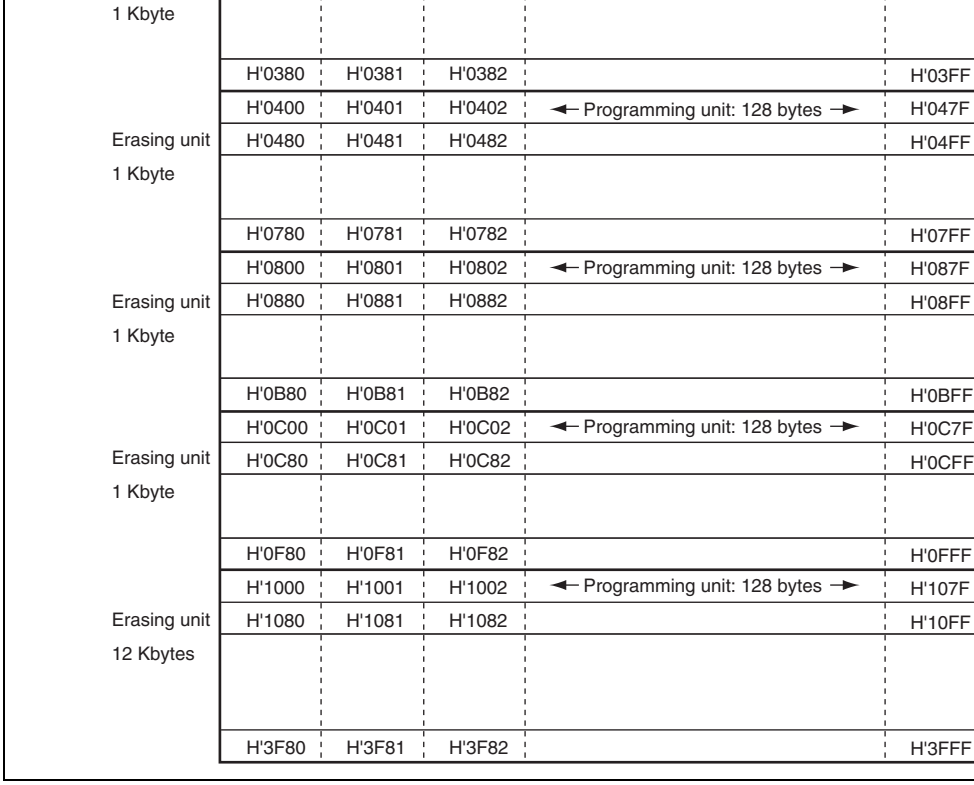


Figure 6.1 Flash Memory Block Configuration

6.2.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is a register that makes the flash memory enter the programming mode, progr verifying mode, erasing mode, or erasing-verifying mode. For details on register setting 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 to 0, other FLMCR1 register bits and all EBR1 b be set.
5	ESU	0	R/W	Erase Setup When this bit is set to 1, the flash memory ente erasure setup state. When it is cleared to 0, the setup state is released. Set this bit to 1 before s E bit in FLMCR1 to 1.
4	PSU	0	R/W	Program Setup When this bit is set to 1, the flash memory ente programming setup state. When it is cleared to programming setup state is released. Set this b before setting the P bit in FLMCR1 to 1.

				When this bit is set to 1 while SWE=1 and ESU= flash memory enters the erasing mode. When it 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= flash memory enters the programming mode. Wh 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. FLM 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 progr or erasing flash memory. When this bit is set to 1 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.

4	EB4	0	R/W	When this bit is set to 1, a12-Kbyte area of H'10 H'3FFF will be erased.
3	EB3	0	R/W	When this bit is set to 1, a 1-Kbyte area of H'0C H'0FFF will be erased.
2	EB2	0	R/W	When this bit is set to 1, a 1-Kbyte area of H'08 H'0BFF will be erased.
1	EB1	0	R/W	When this bit is set to 1, a 1-Kbyte area of H'04 H'07FF will be erased.
0	EB0	0	R/W	When this bit is set to 1, a 1-Kbyte area of H'00 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 enters the subactive mode. There are two modes: mode in which operation of the power circuit of flash memory is partly halted in power-down mode and flash memory can be r mode in which even if a transition is made to subactive mode, operation of the power su 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 su mode, the flash memory enters the power-down When this bit is 1, the flash memory remains in normal mode even after a transition is made to mode.
6 to 0	—	All 0	—	Reserved These bits are always read as 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 en on-board programming/erasure. On-board programming/erasure can also be performed in program mode. When this LSI starts after releasing the reset state, it enters a mode depen the signal levels on the TEST, $\overline{\text{NMI}}$, and E7_0 pins, as shown in table 6.1. The input leve 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 p transfers the programming control program from the externally-connected host to on-chip via SCI3. After erasing the entire flash memory, the programming control program is exe 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 mode, individual blocks can be erased and programmed by branching to the user programming/erasing control program prepared by the user.

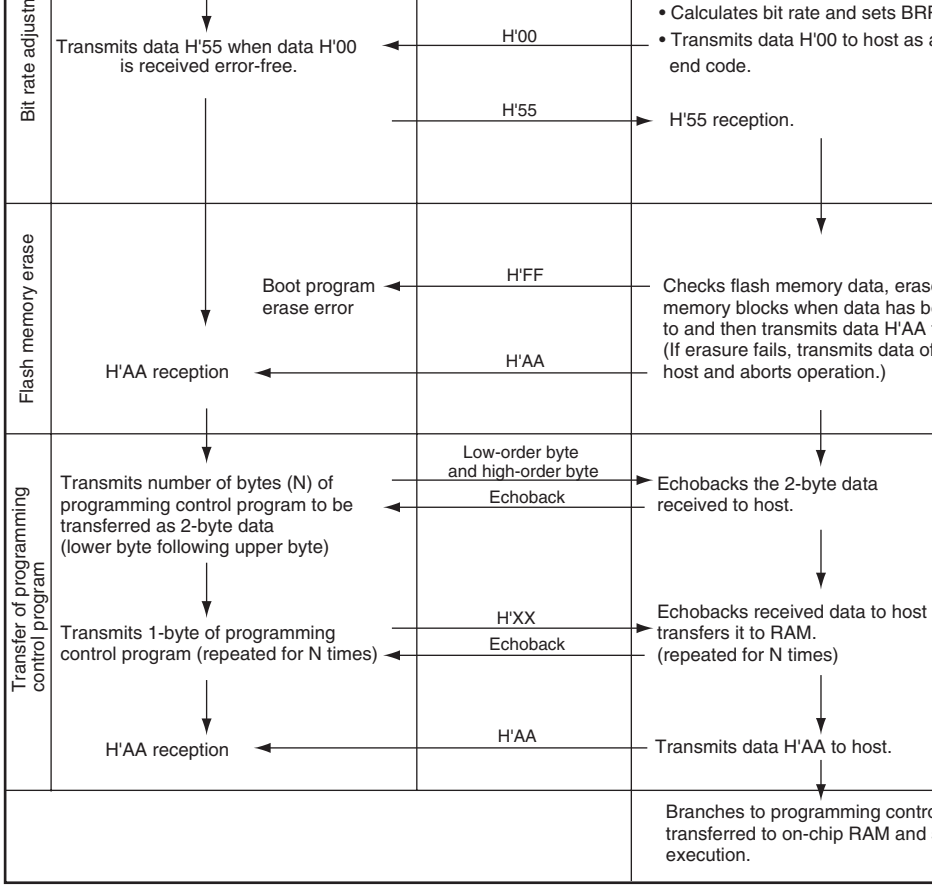
Table 6.1 Setting Programming Modes

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.

1. When the boot mode is used, the flash memory programming control program must
prepared in the host beforehand. Prepare a programming control program in accorda
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,
and no parity. The inversion function of the TXD and RXD pins by SPCR is set to "
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 th
This LSI then calculates the bit rate of the transfer from the host, and adjusts the SCI
to match that of the host. The reset signal should be negated while the RXD pin is dr
The RXD and TXD pins should be pulled up on the board if necessary. After the res
negated, it takes approximately 100 states before this LSI is ready to measure the lov
period.
4. After matching the bit rates, SCI3 transmits one byte of H'00 to the host to indicate t
completion of bit rate adjustment. The host should confirm that it has received this a
end code (H'00) normally and then transmit one byte of H'55 to this LSI. If reception
be performed normally, initiate the boot mode again by a reset. Depending on the ho
transfer bit rate and system clock frequency of this LSI, there will be a discrepancy b
the bit rates of the host and this LSI. To operate the SCI properly, set the host's trans
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 prog
control program transmitted from the host can be stored in the area from H'FB80 to
The boot program area cannot be used until control of the execution is switched from
program to the programming control program.

8. Do not change the TEST pin and $\overline{\text{NMI}}$ pin input levels in boot mode.


Bit rate adjustm
Transmits data H'55 when data H'00 is received error-free.
H'00
• Calculates bit rate and sets BRF
• Transmits data H'00 to host as a end code.
H'55
H'55 reception.
Flash memory erase
Boot program erase error
H'FF
Checks flash memory data, erase memory blocks when data has b to and then transmits data H'AA (If erasure fails, transmits data of host and aborts operation.)
H'AA reception
H'AA
Transfer of programming control program
Low-order byte and high-order byte
Transmits number of bytes (N) of programming control program to be transferred as 2-byte data (lower byte following upper byte)
Echoback
Echobacks the 2-byte data received to host.
H'XX
Transmits 1-byte of programming control program (repeated for N times)
Echoback
Echobacks received data to host transfers it to RAM. (repeated for N times)
H'AA reception
H'AA
Transmits data H'AA to host.
Branches to programming contro transferred to on-chip RAM and execution.

On-board programming/erasing of an individual flash memory block can also be perform
program mode by branching to a user programming/erasing control program. The user mu
prepare the settings for branching to the user programming/erasing control program and r
transfer programming data for on-board programming. The flash memory must contain th
programming/erasing control program or a program that transfer the user programming/e
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
in boot mode. Figure 6.2 shows a sample procedure for programming/erasure in user prog
mode. Prepare a user programming/erasing control program in accordance with the descri
section 6.4, Flash Memory Programming/Erasure.

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

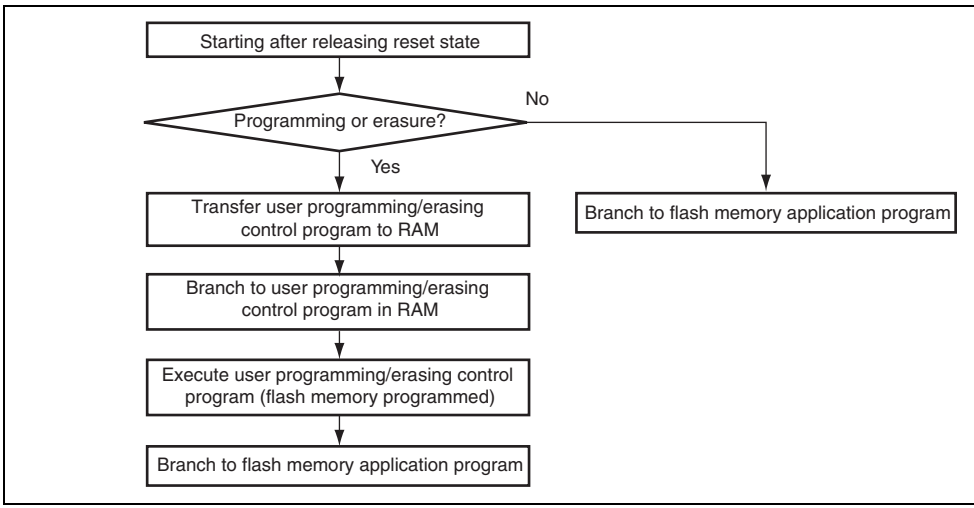
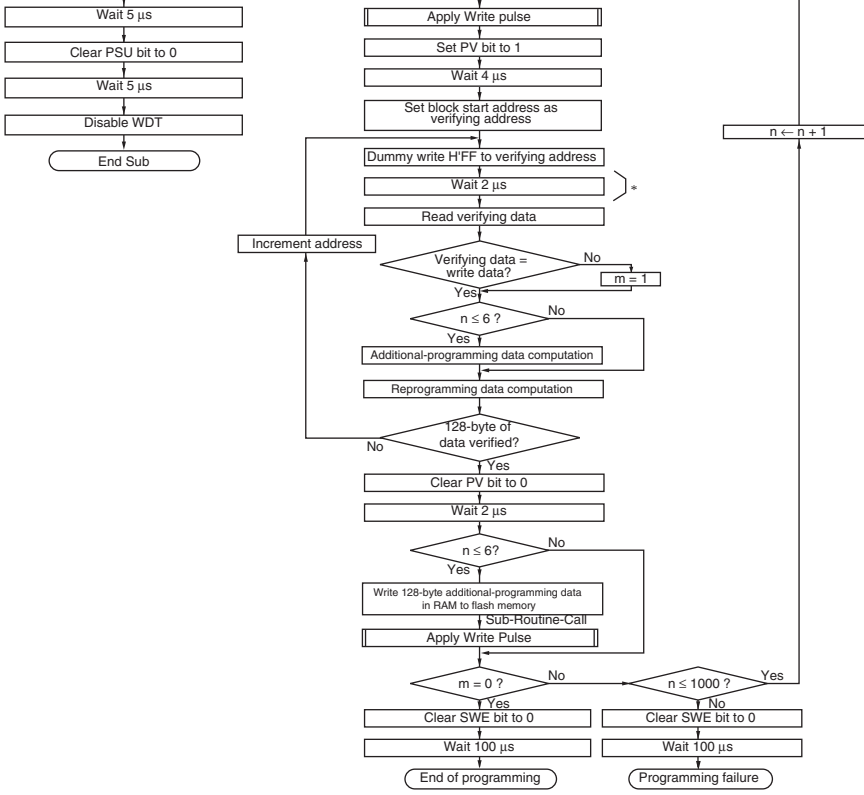


Figure 6.2 Programming/Erasing Flowchart Example in User Program Mo

6.4.1 Programming/Programming-Verifying

When writing data or programs to the flash memory, the programming/programming-ve
flowchart shown in figure 6.3 should be followed. Performing programming operations
to this flowchart will enable data or programs to be written to the flash memory without
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
   data has already been programmed.
2. Programming should be carried out 128 bytes at a time. A 128-byte data transfer mu
   performed even if programming fewer than 128 bytes. In this case, the remaining are
   filled with H'FF.
3. Prepare the following data storage areas in RAM: A 128-byte programming data are
   byte reprogramming data area, and a 128-byte additional-programming data area. Pe
   reprogramming data computation according to table 6.4, and additional programmin
   computation according to table 6.5.
4. Consecutively transfer 128 bytes of data in bytes from the reprogramming data area
   additional-programming data area to the flash memory. The programming address ar
   byte data are latched in the flash memory. The lower eight bits of the start address in
   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
   allowable programming times.
6. The watchdog timer (WDT) is set to prevent overprogramming due to program runa
   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 wh
   two bits are B'00. Verifying data can be read in words or in longwords from the addr
   which a dummy write was performed.



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

Reprogram Data	Verify Data	Additional-Program Data	Comments
0	0	0	Needs to be progra additionally
0	1	1	No additional progr
1	0	1	No additional progr
1	1	1	No additional progr

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.

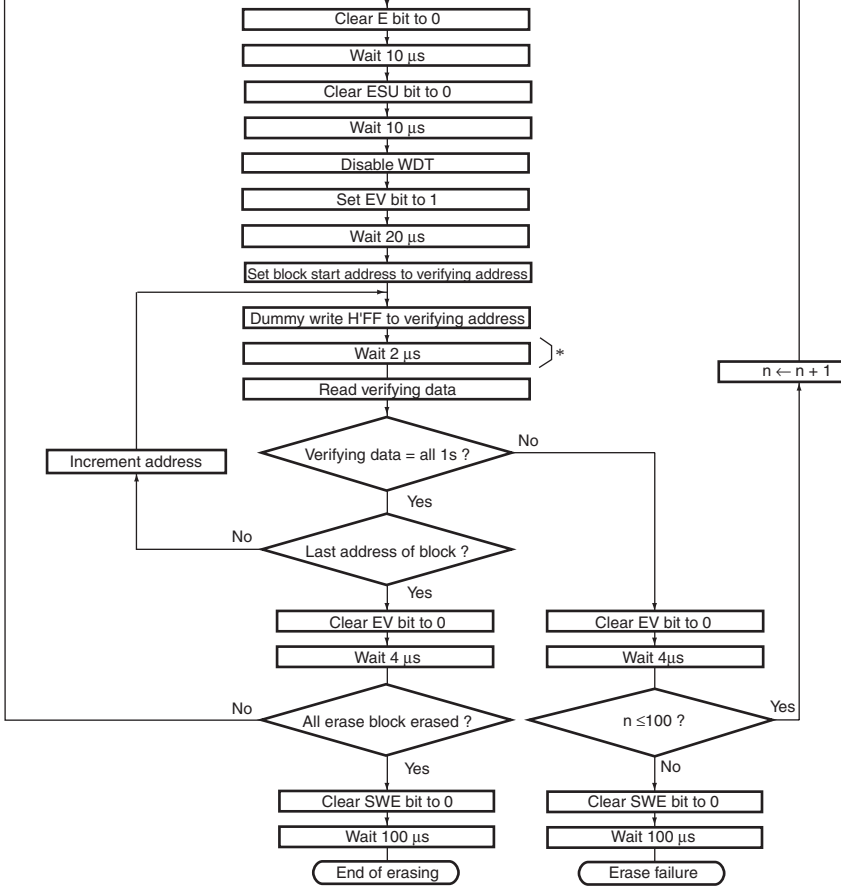
crush, etc. An overflow cycle of approximately 19.8 ms is adequate.

5. For writing dummy data to a verifying address, write one byte of data H'FF to a whose lower two bits are B'00. Verifying data can be read in longwords from the a 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 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 progr 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 o 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 can 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 ve

Figure 6.4 Erase/Erase-Verify Flowchart

register 2 (FLMCR2), and erase block register 1 (EBR1) are initialized. For a reset by the pin, the reset state is entered when the $\overline{\text{RES}}$ signal is held low until oscillation stabilizes a switching on. For a reset during operation, hold the $\overline{\text{RES}}$ signal low for the $\overline{\text{RES}}$ pulse wi specified in the AC Characteristics section.

6.5.2 Software Protection

Software protection can protect programming/erasing of all flash memory blocks by clea SWE bit in FLMCR1. When software protection is enabled, setting the P or E bit in FLM does not cause a transition to programming mode or erasing mode. By setting the erase b register 1 (EBR1), erasing protection can be set for individual blocks. When EBR1 is set 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 erro detected because CPU crush occurs during flash memory programming/erasure, or operat performed in accordance with the programming/erasing algorithm. Aborting programmin prevents damage to the flash memory due to overprogramming or overerasing.

When the following errors are detected during programming/erasing of flash memory, the 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 vecto and instruction fetch)
- Exception handling excluding a reset is started during programming/erasure
- When the SLEEP instruction is executed during programming/erasure

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 me 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 fl memory. In subactive mode, the flash memory can be set to operate in power-down mo PDWND bit in FLPWCR. When the flash memory returns to its normal operating state power-down mode or standby mode, a period to stabilize operation of the power supply that were stopped is needed. When the flash memory returns to its normal operating stat STS2 to STS0 in SYSCR1 must be set to provide a wait time of at least 20 µs, even whe 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

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

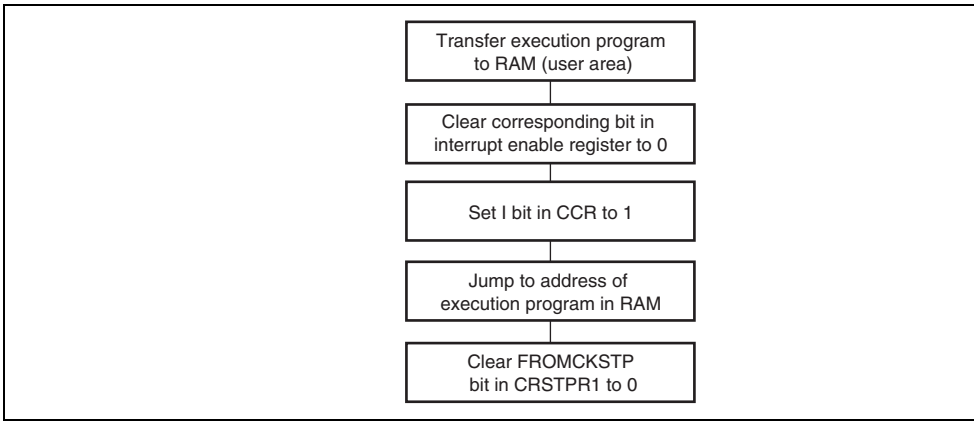


Figure 6.5 Module Standby Mode Setting

For details on the execution of bit manipulation instructions to the port data register (PD
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
RTC output pin, CLKOUT output pin, and interrupt input pin. Figure 8.1 shows its pin
configuration.

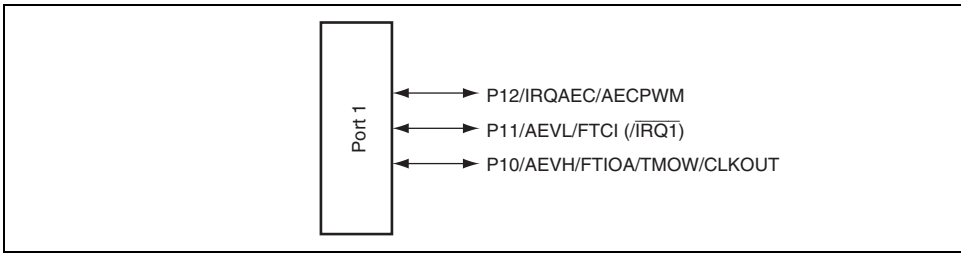


Figure 8.1 Port 1 Pin Configuration

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)

1	P11	0	R/W	states. If port 1 is read while PCR1 bits are cle
0	P10	0	R/W	the pin states are read.

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 canno modified.
2	PCR12	0	W	Setting a PCR1 bit to 1 makes the correspondi
1	PCR11	0	W	(P12 to P10) an output pin, while clearing the b
0	PCR10	0	W	makes the pin an input pin. The settings in PCI 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.

0	PUCR10	0	R/W	for the corresponding pin, while clearing the b turns off the pull-up MOS.

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 cann 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

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 b
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 p
		0	x	IRQAEC input pin

[Legend] x: Don't care.

[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 i
				1	x	x	1	P10 output/FTIOA
			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 pi
			1	x	x	x	x	CLKOUT output pi
		1	0	x	x	x	x	CLKOUT output pi

[Legend] x: Don't care.

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

[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 volt and interrupt pin. Figure 8.2 shows its pin configuration.

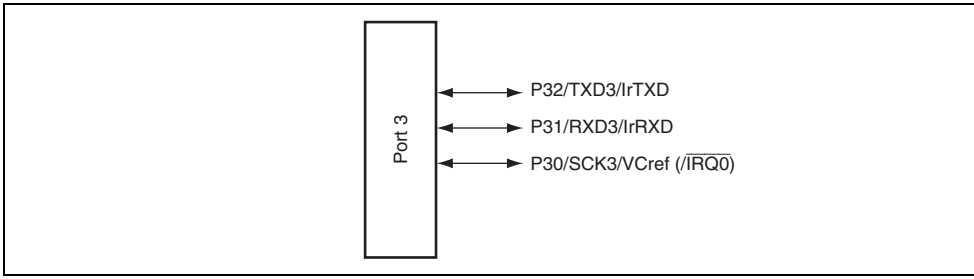


Figure 8.2 Port 3 Pin Configuration

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)

1	P31	0	R/W	states. If port 3 is read while PCR3 bits are cl
0	P30	0	R/W	the pin states are read.

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 cann modified.
2	PCR32	0	W	Setting a PCR3 bit to 1 makes the correspond
1	PCR31	0	W	(P32 to P30) an output pin, while clearing the makes the pin an input pin. The settings in PC
0	PCR30	0	W	PDR3 are valid when the corresponding pin is designated as a general I/O pin. PCR3 is a write-only register. The read value undefined.

1	PUCR31	0	R/W	for the corresponding pin, while clearing the bit
0	PUCR30	0	R/W	turns off the pull-up MOS.

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 canno 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

		1	x	IrTXD output pin

[Legend] x: Don't care.

- P31/RXD3/IrRXD pin

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

[Legend] x: Don't care.

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

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

[Legend] x: Don't care.

[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 config

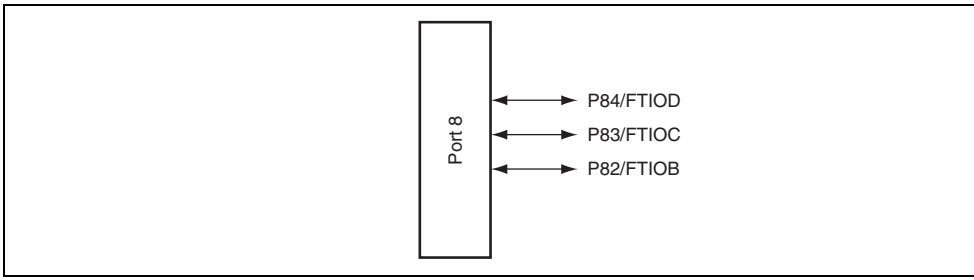


Figure 8.3 Port 8 Pin Configuration

Port 8 has the following registers.

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

3	P83	0	R/W	states. If port 8 is read while PCR8 bits are cl
2	P82	0	R/W	the pin states are read.
1, 0	—	—	—	Reserved The read value is undefined. These bits cann 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 cann modified.
4	PCR84	0	W	Setting a PCR8 bit to 1 makes the correspond
3	PCR83	0	W	(P84 to P82) an output pin, while clearing the
2	PCR82	0	W	makes the pin an input pin. The settings in PC PDR8 are valid when the corresponding pin is designated as a general I/O pin. PCR8 is a write-only register. The read value undefined.
1, 0	—	—	—	Reserved The read value is undefined. These bits cann modified.

3	PUCR83	0	R/W	for the corresponding pin, while clearing the bit
2	PUCR82	0	R/W	turns off the pull-up MOS.
1, 0	—	—	—	Reserved The read value is undefined. These bits canno 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
			1	x	x	FTIOD output
		1	x	x	0	P84 input/FTI pin
					1	P84 output/FT input pin
	1	x	x	x	x	FTIOD output

[Legend] x: Don't care.

					1	P83 output/F input pin
	1	x	x	x	x	FTIOC outpu

[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 pi
				1	x	FTIOB outpu
			1	x	x	FTIOB outpu
		1	x	x	0	P82 input/FT pin
					1	P82 output/F input pin
	1	x	x	x	x	FTIOB outpu

[Legend] x: Don't care.

[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. Fi
shows its pin configuration.

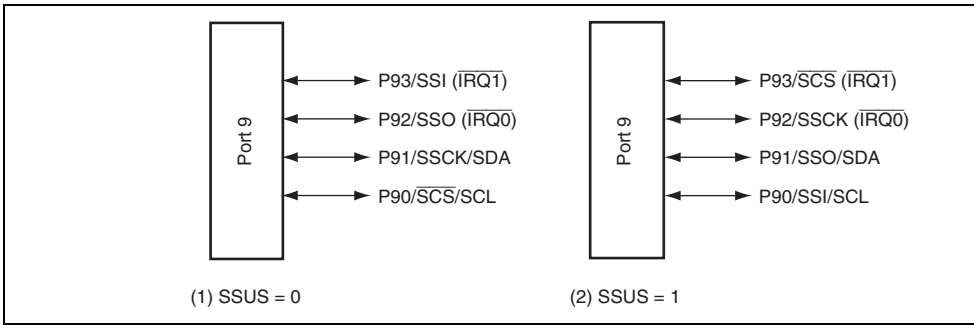


Figure 8.4 Port 9 Pin Configuration

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)

2	P92	0	R/W	states. If port 9 is read while PCR9 bits are cl
1	P91	0	R/W	the pin states are read.
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 cann modified.
3	PCR93	0	W	Setting a PCR9 bit to 1 makes the correspond
2	PCR92	0	W	output pin, while clearing the bit to 0 makes th
1	PCR91	0	W	input pin. The settings in PCR9 and in PDR9 when the corresponding pin is designated as
0	PCR90	0	W	I/O pin. PCR9 is a write-only register. The read value undefined.

2	P92ODR	0	R/W	as the NMOS open-drain output. When cleared
1	P91ODR	0	R/W	corresponding pin functions as the CMOS outp
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 canno modified.
3	PUCR93	0	R/W	When a PCR9 bit is cleared to 0, setting the corresponding PUCR9 bit to 1 turns on the pul for the corresponding pin, while clearing the bit turns off the pull-up MOS.
2	PUCR92	0	R/W	
1	PUCR91	0	R/W	
0	PUCR90	0	R/W	

4. When the pins for communication data (SSI and SSO) are set to open-drain output w
SOOS bit in the SSCRH register of SSU, they are set to open-drain output regardless
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 requir
details, see section 15.4.4, Communication Modes and Pin Functions, and appen
Port 9 Related Register Settings and Pin Functions.

[Legend] x: Don't care.

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

Note: When this pin is used as the $\overline{SCS}$/SSI pin, register settings of the SSU are requir
details, see section 15.4.4, Communication Modes and Pin Functions, and appen
Port 9 Related Register Settings and Pin Functions. When this pin is used as the
register settings of the IIC2 are required. For details, see section 16.3.1, I^2C Bus
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. W
PCR9 bit is cleared to 0, setting the corresponding PUCR9 bit to 1 turns on the input pu
for that pin. The input pull-up MOS function is in the off state after a reset.

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

[Legend] x: Don't care.

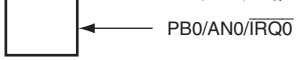


Figure 8.5 Port B Pin Configuration

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 canno modified.
5	PB5	Undefined	R	Reading PDRB always gives the pin states. Ho
4	PB4	Undefined	R	a port B pin is selected as an analog input cha
3	PB3	Undefined	R	the CH3 to CH0 bits in AMR of the A/D conver
2	PB2	Undefined	R	pin is read as 0 regardless of the input voltage
1	PB1	Undefined	R	
0	PB0	Undefined	R	

				Selects whether pin TEST/ADTRG is used as as ADTRG. 0: TEST pin 1: ADTRG input pin For details on the setting of the ADTRG input to section 17.4.2, External Trigger Input Timin
2	—	—	—	Reserved The read value is undefined. This bit cannot b modified.
1	IRQ1	0	R/W	PB1/AN1/IRQ1 Pin Function Switch Selects whether pin PB1/AN1/IRQ1 is used a PB1/AN1 or as IRQ1. 0: PB1/AN1 input pin 1: IRQ1 input pin*
0	IRQ0	0	R/W	PB0/AN0/IRQ0 Pin Function Switch Selects whether pin PB0/AN0/IRQ0 is used a PB0/AN0 or as IRQ0. 0: PB0/AN0 input pin 1: IRQ0 input pin*

Note: * When the IRQnS1 and IRQnS0 (n = 1 or 0) bits in PFCR are set to a value ot B'00, these bits should not be set since the IRQn pin is assigned to another p

[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
	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.

[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.

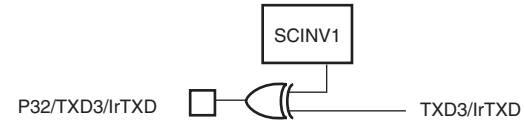


Figure 8.6 Input/Output Data Inversion Function

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 as TXD3/IrTXD. 0: P32 I/O pin 1: TXD3/IrTXD output pin* Note: * Set the TE bit in SCR3 after setting this
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 TXD3/IrTXD pin is to be inverted or not. 0: TXD3/IrTXD output data is not inverted 1: TXD3/IrTXD output data is inverted

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 c

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	—	Reserved These bits are always read as 0. These bits c 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 SSCK is assigned to P91 $\overline{\text{SCS}}$ is assigned to P90 1: SSI is assigned to P90 SSO is assigned to P91 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

— For a pin also used by the A/D converter, pull it up to AVcc with an external resis
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 resis
  approximately 100 kΩ.
  — Set the output of the unused pin to low and pull it down to GND with an external r
  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 FTI
FTCI, SSCK, $\overline{\text{SCS}}$, SDA, and SCL are selected, the corresponding pins have the schmitt-
input characteristics, which are different from the ones when they are used as the port inp

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

- Use of module standby mode enables this module to be placed in standby mode inde
  when not used. (Timer B1 is halted as the initial value. For details, refer to section 5.
  Standby Function.)

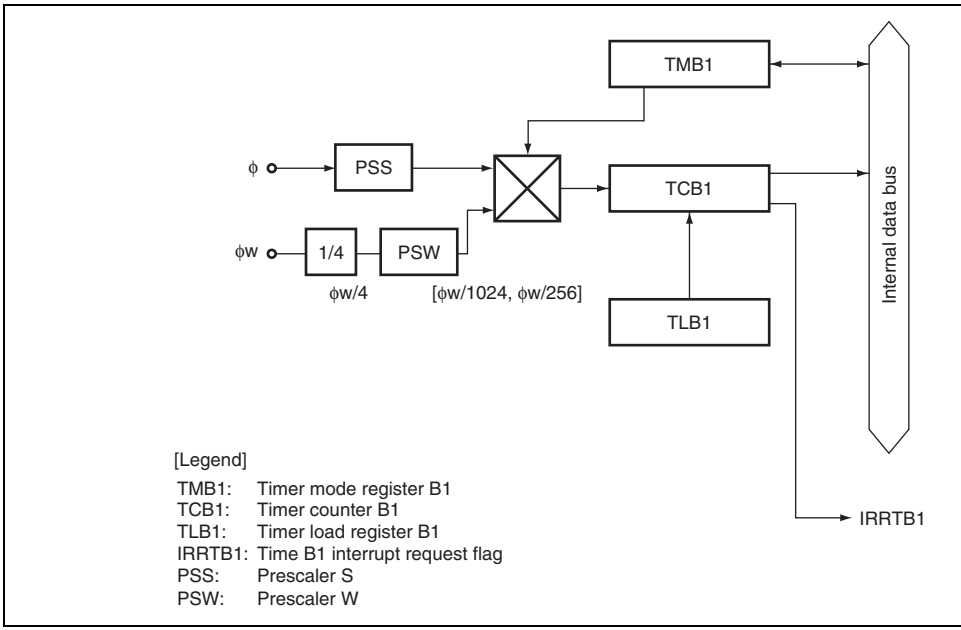


Figure 9.1 Block Diagram of Timer B1

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

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

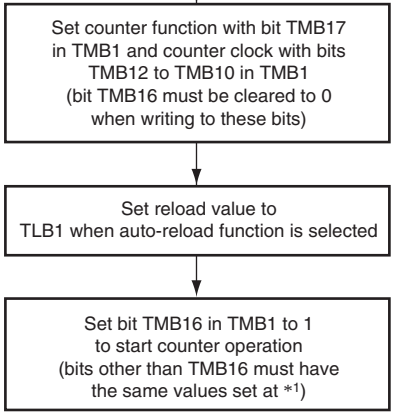


Figure 9.2 Timer B1 Initial Setting Flow

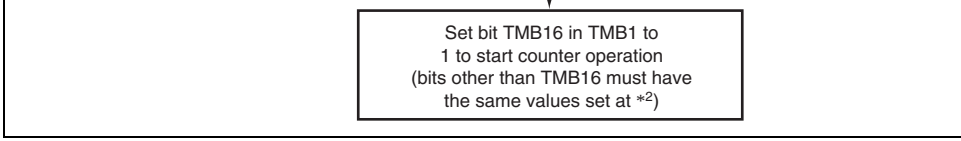


Figure 9.3 Processing Flow When Changing Setting during Counter Opera

After bit TMB16 in TMB1 is set to 1 to start the counter operation and the count value in reaches H'FF, the next clock signal input causes timer B1 to overflow, setting flag IRRTB 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 tim operation (TMB17 = 0) is selected, when a value is set in TLB1 with bit TMB16 in TMB 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 time a reload value is set in TLB1 with bit TMB16 in TMB1 cleared to 0, the same value is loa TCB1. After bit TMB16 in TMB1 is set to 1 to start the counter operation and the count v TCB1 reaches H'FF, the next clock signal input causes timer B1 to overflow. The TLB1 v then loaded into TCB1, and the count continues from that value. The overflow period can 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 new value in TLB1 in auto-reload mode (TMB17 = 1), clear bit TMB16 in TMB1 to 0 be making the new setting.

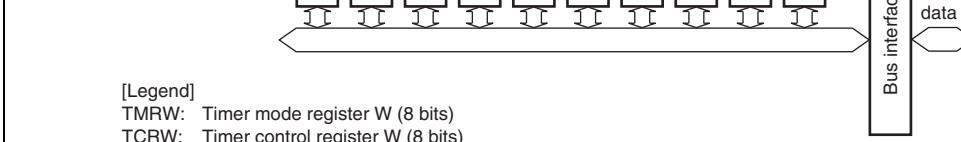
ϕw/1024											
ϕ/4, ϕ/16, ϕ/64, ϕ/256, ϕ/2048, ϕ/8192	o	o	o	o	×	×	×	×	×	×	

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

- 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 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 inde when not used. (The timer W is halted as the initial value. For details, refer to sectio Module Standby Function.)

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

		compare match	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	Com mat cap



[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

compare B			input pin for GRB input captu PWM output pin in PWM mod
Input capture/output compare C	FTIOC	Input/output	Output pin for GRC output co input pin for GRC input captu PWM output pin in PWM mod
Input capture/output compare D	FTIOD	Input/output	Output pin for GRD output co input pin for GRD input captu PWM output pin in PWM mod

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)

5	BUFEB	0	R/W	Buffer Operation B Selects the GRD function. 0: GRD operates as an input capture/output register 1: GRD operates as the buffer register for G
4	BUFEA	0	R/W	Buffer Operation A Selects the GRC function. 0: GRC operates as an input capture/output register 1: GRC operates as the buffer register for G
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 compar 1: PWM output
1	PWMC	0	R/W	PWM Mode C Selects the output mode of the FTIOC pin. 0: FTIOC operates normally (output compar 1: PWM output
0	PWMB	0	R/W	PWM Mode B Selects the output mode of the FTIOB pin. 0: FTIOB operates normally (output compar 1: PWM output

3	CKS1	0	R/W	Select the TCNT clock source.
4	CKS0	0	R/W	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 ev (FTCI) With a setting of 0xx, the timer W can be used active mode or sleep mode. Do not make this subactive mode or subsleep mode. When 100 is set in subactive mode or subslee the timer W can be used only when ϕ_w is selec the CPU operating clock. When 101 is set in subactive mode or subslee the timer W can be used only when ϕ_w or $\phi_w/2$ selected as the CPU operating clock.
3	TOD	0	R/W	Timer Output Level Setting D Sets the output value of the FTIOD pin until the 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 compare match C is generated. 0: Output value is 0* 1: Output value is 1*

				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 requ 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 Ena When this bit is set to 1, IMID interrupt reque IMFD flag in TSRW is enabled.
2	IMIEC	0	R/W	Input Capture/Compare Match Interrupt Ena When this bit is set to 1, IMIC interrupt reque IMFC flag in TSRW is enabled.
1	IMIEB	0	R/W	Input Capture/Compare Match Interrupt Ena When this bit is set to 1, IMIB interrupt reque IMFB flag in TSRW is enabled.
0	IMIEA	0	R/W	Input Capture/Compare Match Interrupt Ena When this bit is set to 1, IMIA interrupt reque IMFA flag in TSRW is enabled.

				Read OVF when OVF = 1, then write 0 in OV
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 compare register • The TCNT value is transferred to GRD by capture signal when GRD functions as an capture register [Clearing condition] Read IMFD when IMFD = 1, then write 0 in IM
2	IMFC	0	R/(W)*	Input Capture/Compare Match Flag C [Setting conditions] • TCNT = GRC when GRC functions as an compare register • The TCNT value is transferred to GRC by capture signal when GRC functions as an capture register [Clearing condition] Read IMFC when IMFC = 1, then write 0 in IM

				Read IMFB when IMFB = 1, then write 0 in IM
0	IMFA	0	R/(W)*	Input Capture/Compare Match Flag A [Setting conditions] • TCNT = GRA when GRA functions as an compare register • The TCNT value is transferred to GRA by capture signal when GRA functions as an capture register [Clearing condition] Read IMFA when IMFA = 1, then write 0 in IM

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 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 regis 1: GRB functions as an input capture register

				01: Input capture at falling edge at the FTIOB p
				1x: Input capture at rising and falling edges of 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 registe 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 10: 1 output to the FTIOA pin at GRA compare 11: Output toggles to the FTIOA pin at GRA co match When IOA2 = 1, 00: Input capture at rising edge of the FTIOA p 01: Input capture at falling edge of the FTIOA p 1x: Input capture at rising and falling edges of FTIOA pin

[Legend] x: Don't care.

				0: GRD functions as an output compare registe
				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
				10: 1 output to the FTIOD pin at GRD compare
				11: Output toggles to the FTIOD pin at GRD co match
				When IOD2 = 1,
				00: Input capture at rising edge at the FTIOD p
				01: Input capture at falling edge at the FTIOD p
				1x: Input capture at rising and falling edges at 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 registe
				1: GRC functions as an input capture register

01: Input capture to GRC at falling edge of the F
1x: Input capture to GRC at rising and falling ed
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
CKS0 in TCRW. TCNT can be cleared to H'0000 through a compare match with GRA by
the CCLR in TCRW to 1. When TCNT overflows (changes from H'FFFF to H'0000), the
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 allowe
TCNT is initialized to H'0000 by a reset.

When a general register is used as an input-capture register, an external input-capture sig
detected and the current TCNT value is stored in the general register. The corresponding
(IMFA, IMFB, IMFC, or IMFD) in TSRW is set to 1. If the corresponding interrupt-ena
(IMIEA, IMIEB, IMIEC, or IMIED) in TIERW is set to 1 at this time, an interrupt reque
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 settin
and BUFEB in TMRW.

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

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

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

When the count overflows from H'FFFF to H'0000, the OVF flag in TSRW is set to 1. If
in TIERW is set to 1, an interrupt request is generated. Figure 10.2 shows free-running co

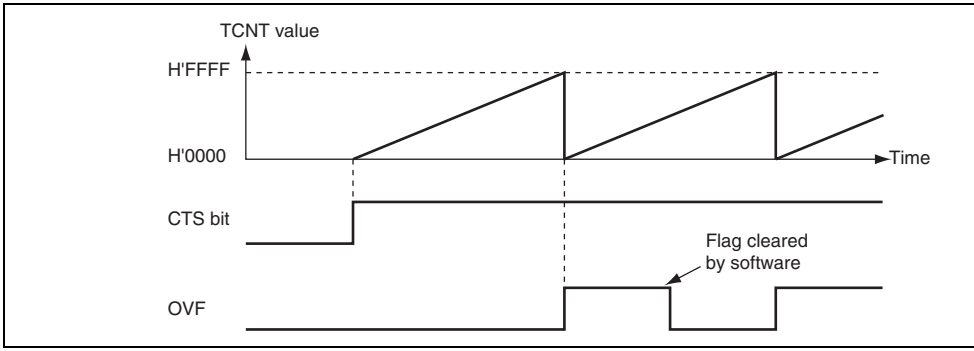


Figure 10.2 Free-Running Counter Operation

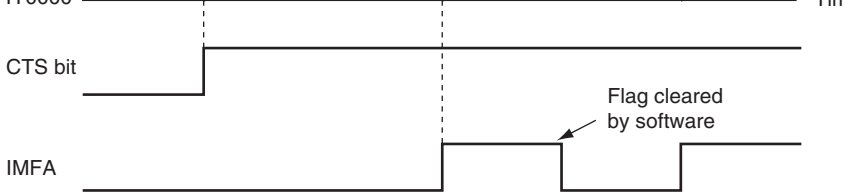


Figure 10.3 Periodic Counter Operation

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

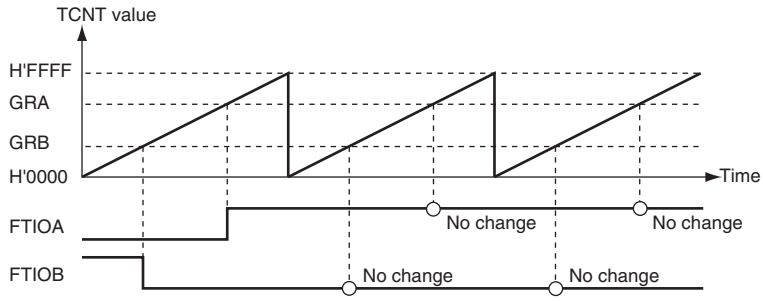


Figure 10.4 0 and 1 Output Example (TOA = 0, TOB = 1)

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 c
cleared by compare match A. Toggle output is selected for both compare match A and B.

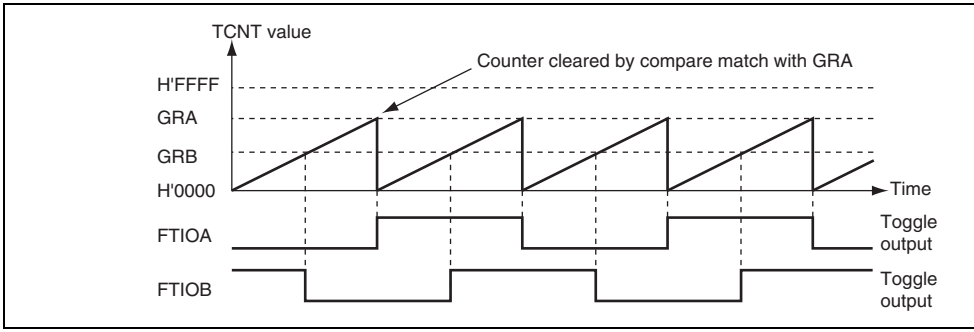


Figure 10.6 Toggle Output Example (TOA = 0, TOB = 1)

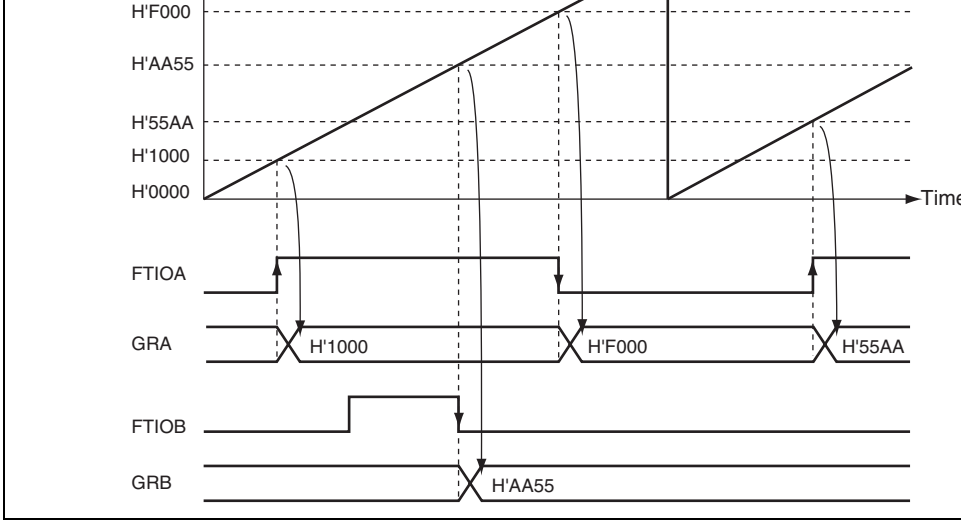


Figure 10.7 Input Capture Operating Example

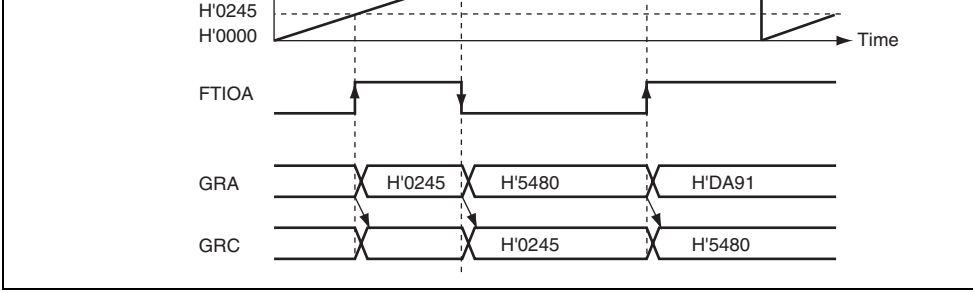


Figure 10.8 Buffer Operation Example (Input Capture)

10.4.2 PWM Operation

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

Figure 10.9 shows an example of operation in PWM mode. The output signals go to 1 an is cleared at compare match A, and the output signals go to 0 at compare match B, C, and (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
TCNT is cleared at compare match A, and the output signals go to 1 at compare match E
D (TOB, TOC, and TOD = 0: initial output values are set to 0).

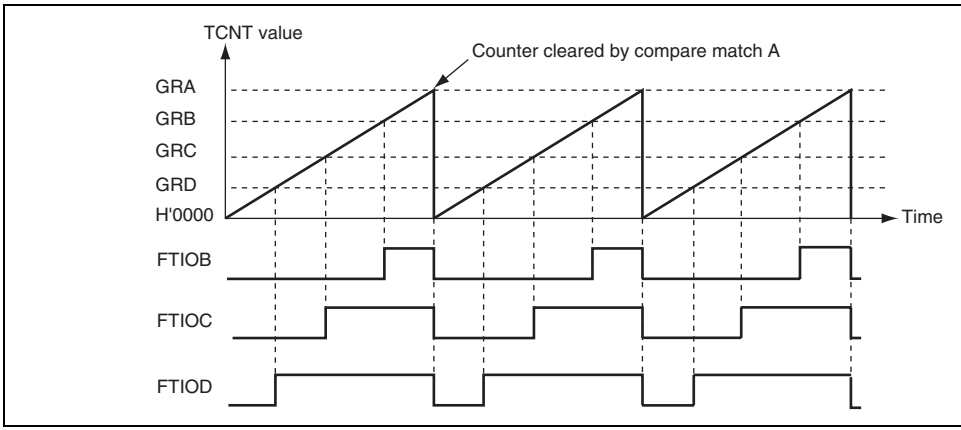


Figure 10.10 PWM Mode Example (2)

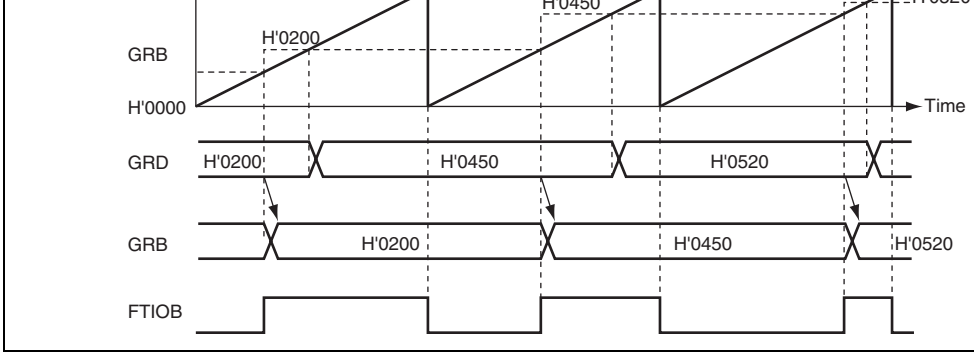


Figure 10.11 Buffer Operation Example (Output Compare)

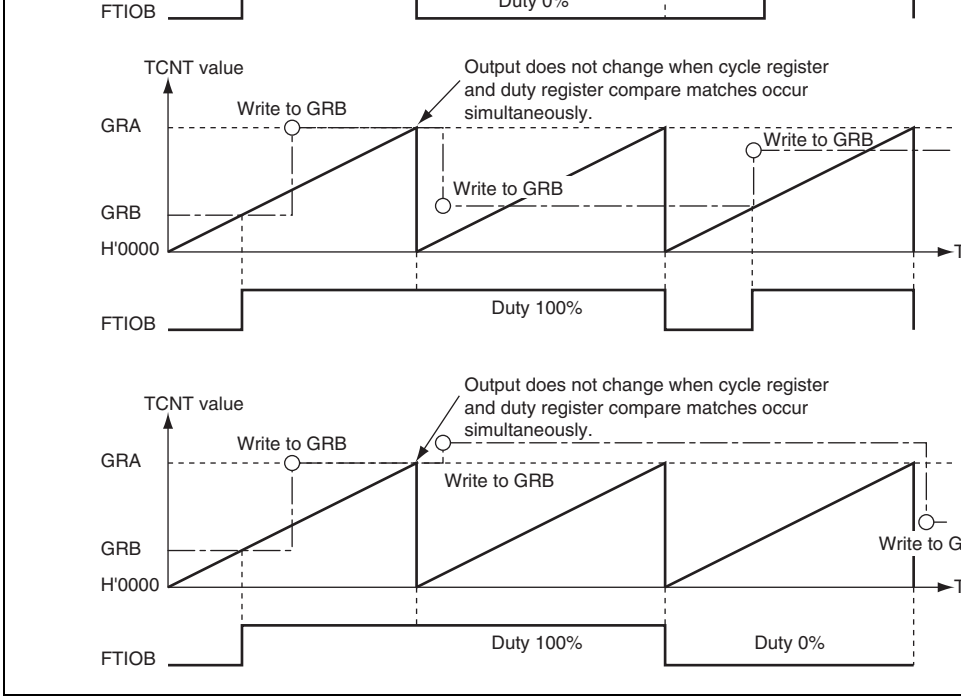


Figure 10.12 PWM Mode Example
(TOB, TOC, and TOD = 0: initial output values are set to 0)

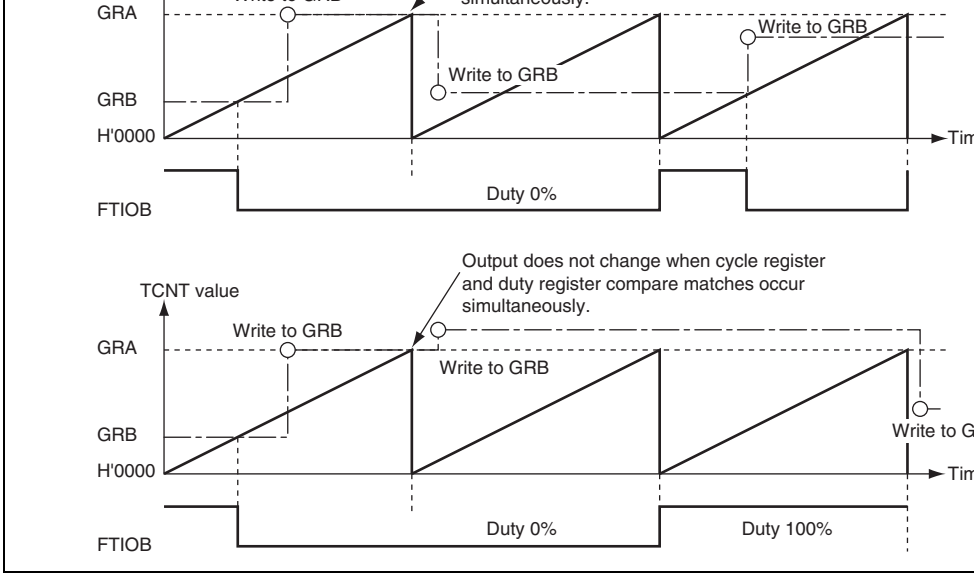


Figure 10.13 PWM Mode Example
(TOB, TOC, and TOD = 1: initial output values are set to 1)

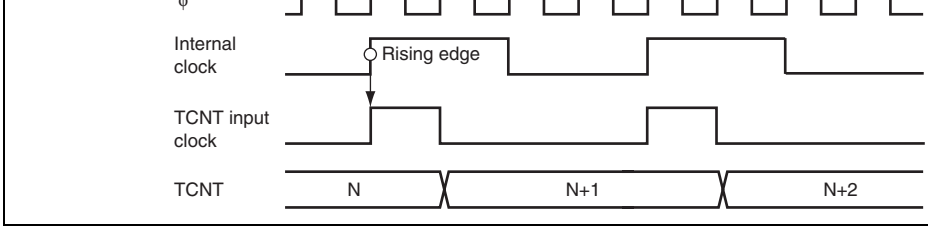


Figure 10.14 Count Timing for Internal Clock Source

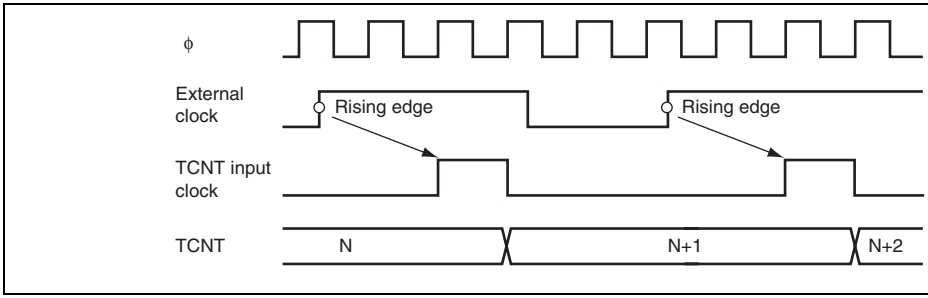


Figure 10.15 Count Timing for External Clock Source

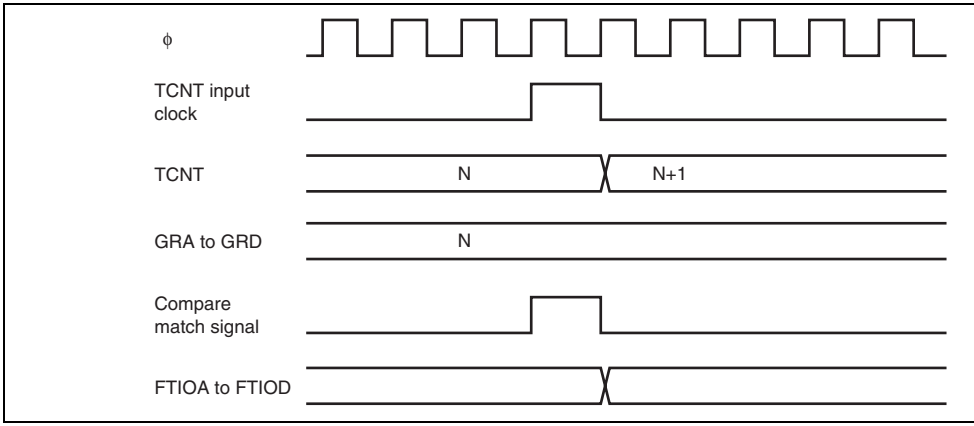


Figure 10.16 Output Compare Output Timing

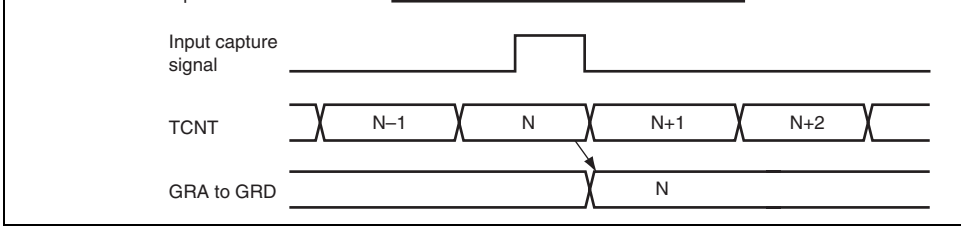


Figure 10.17 Input Capture Input Signal Timing

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 t
value is N, the counter counts from 0 to N, and its cycle is N + 1.

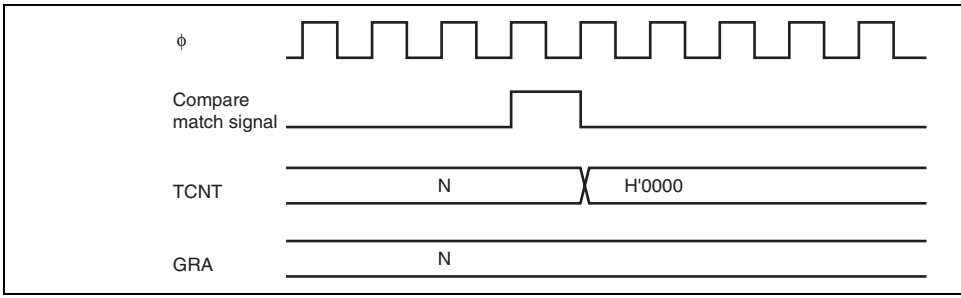


Figure 10.18 Timing of Counter Clearing by Compare Match

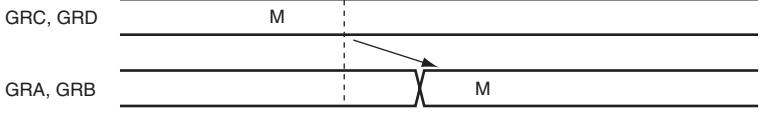


Figure 10.19 Buffer Operation Timing (Compare Match)

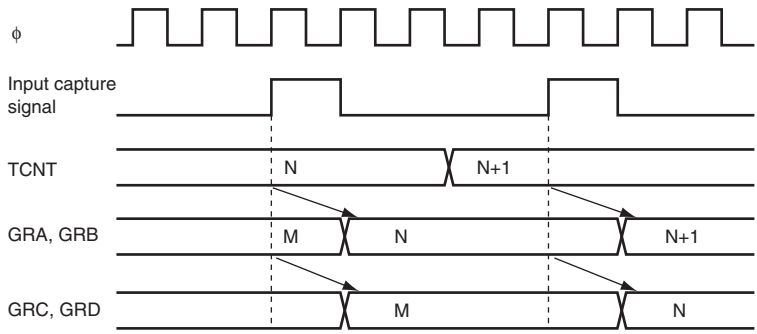


Figure 10.20 Buffer Operation Timing (Input Capture)

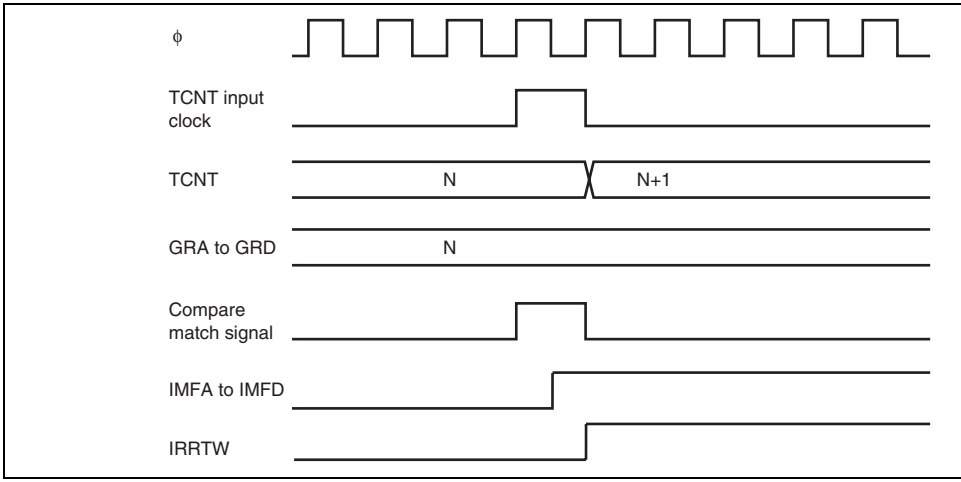


Figure 10.21 Timing of IMFA to IMFD Flag Setting at Compare Match

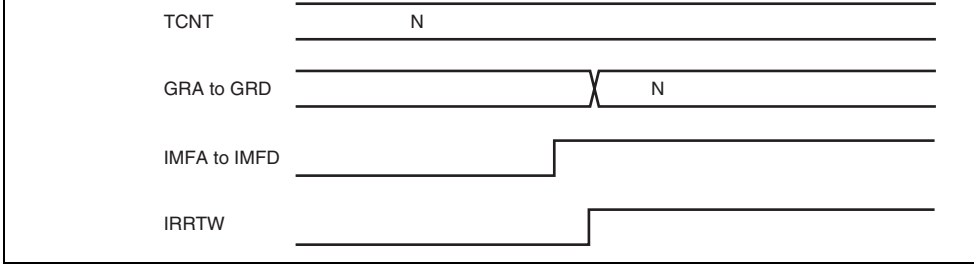


Figure 10.22 Timing of IMFA to IMFD Flag Setting at Input Capture

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 s
is cleared. Figure 10.23 shows the status flag clearing timing.

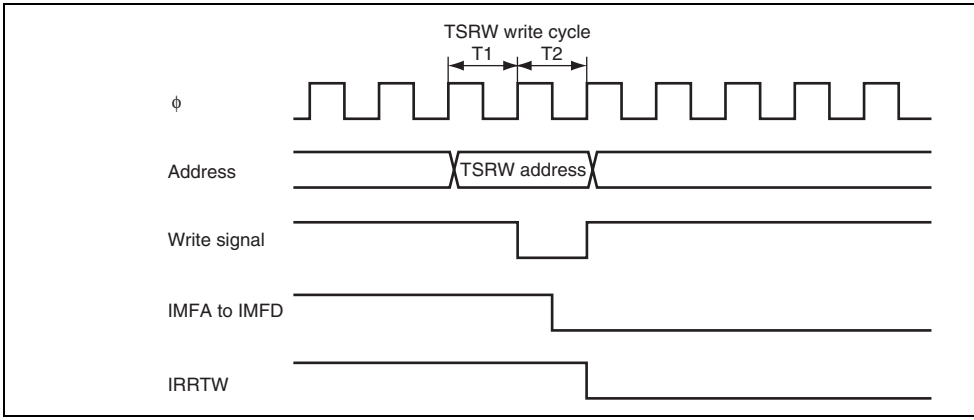


Figure 10.23 Timing of Status Flag Clearing by CPU

ϕw/16										
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 leas system clock cycles; shorter pulses will not be detected correctly. The system clock here indicates the clock set for the CPU operation. For example, in the ϕw/8 operatio 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 c takes priority and the write is not performed, as shown in figure 10.24. If counting-u generated in the TCNT write cycle to contend with the TCNT counting-up, writing t precedence.
3. Depending on the timing, TCNT may be incremented by a switch between different clock sources. When TCNT is internally clocked, an increment pulse is generated fr rising edge of an internal clock signal, that is, the divided system clock (ϕ). Therefor shown in figure 10.25 the switch is from a low clock signal to a high clock signal, th 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 i request cannot be cleared. Before entering module standby mode, disable interrupt re

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 in GRA or GRB before capturing.)

9. When the compare match timing conflicts with the GRA to GRD write timing as the c 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 match operation in buffer mode,
    a. the written values are reflected in GRA or GRB. (The values in GRA or GRB are 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 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 o in buffer mode, FTIOC or FTIOD output changes by the GRC or GRD compare matc
13. When ϕw, $\phi w/4$, $\phi w/16$, or FTCI input is selected as the count clock, counting is enab in subactive and subsleep modes. Counting is disabled during the oscillation stabilizat 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 enab active and sleep modes although counting may be misaligned by one in transition fron active to subactive mode.

Figure 10.24 Contention between TCNT Write and Clear

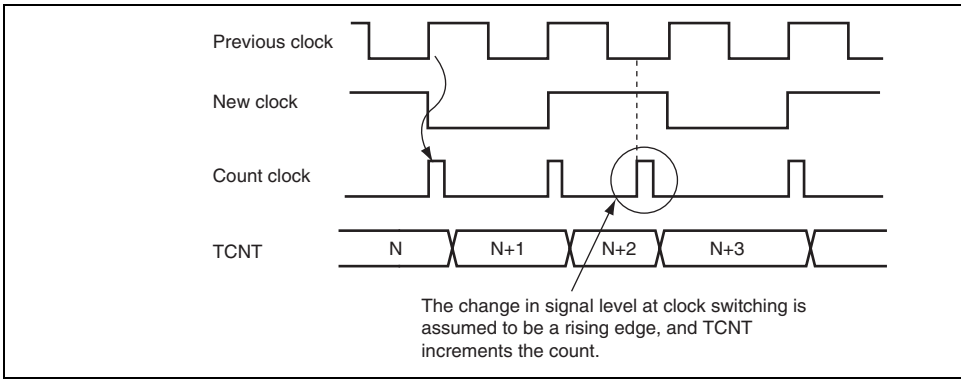
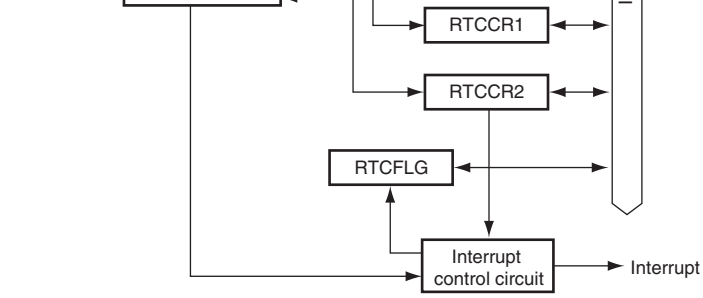


Figure 10.25 Internal Clock Switching and TCNT Operation

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



[Legend]

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

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

- 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
read register used as a counter, when it operates as a free running counter. For more info
on reading seconds, minutes, hours, and day-of-week, see section 11.4.3, Data Reading

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 (o the values of second, minute, hour, and day-of- registers. When this bit is 0, the values of secon hour, and day-of-week data registers must be a
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 carr 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.

				hour, and day-of-week data registers must be ac
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
1	MN01	—/(0)*	R/W	generated, 1 is added to the ten's position.
0	MN00	—/(0)*	R/W	

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

				registers. When this bit is 0, the values of secon hour, and day-of-week data registers must be a
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 i 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.

				hour, and day-of-week data registers must be ac
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.

				0: RTC operates in 12-hour mode. RHRDR co to 11.
				1: RTC operates in 24-hour mode. RHRDR co to 23.
5	PM	—/(0)*	R/W	A.M./P.M.
				0: Indicates a.m. when RTC is in the 12-hour m
				1: Indicates p.m. when RTC is in the 12-hour m
4	RST	0	R/W	Reset
				0: Normal operation
				1: Resets registers and control circuits except R and this bit. Clear this bit to 0 after having be 1.
3	INT	—/(0)*	R/W	Interrupt Occurrence Timing
				0: Periodic interrupts of second, minute, hour, a week occur during the RTC busy period.
				1: Periodic interrupts of second, minute, hour, a week occur immediately after the RTC busy 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.

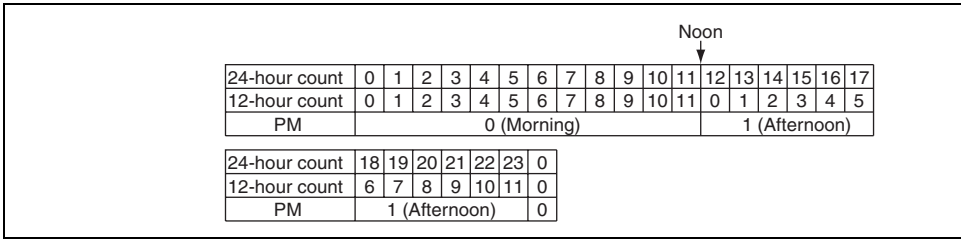


Figure 11.2 Definition of Time Expression

				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.

Bit	Bit Name	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
4	SUB32K	0	R/W	TMOW output in PMR1. 000: ϕ/4 010: ϕ/8 100: ϕ/16 110: ϕ/32 xx1: ϕ_w
3	RCS3	1	R/W	Clock Source Selection
2	RCS2	0	R/W	0000: ϕ/8····· Free running counter opera
1	RCS1	0	R/W	0001: ϕ/32····· Free running counter opera
0	RCS0	0	R/W	0010: ϕ/128····· Free running counter opera 0011: ϕ/256····· Free running counter opera 0100: ϕ/512····· Free running counter opera 0101: ϕ/2048····· Free running counter opera 0110: ϕ/4096····· Free running counter opera 0111: ϕ/8192····· Free running counter opera 1000: ϕ_w/4····· RTC operation 1001 to 1111: Setting prohibited

				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

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

11.4.2 Initial Setting Procedure

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

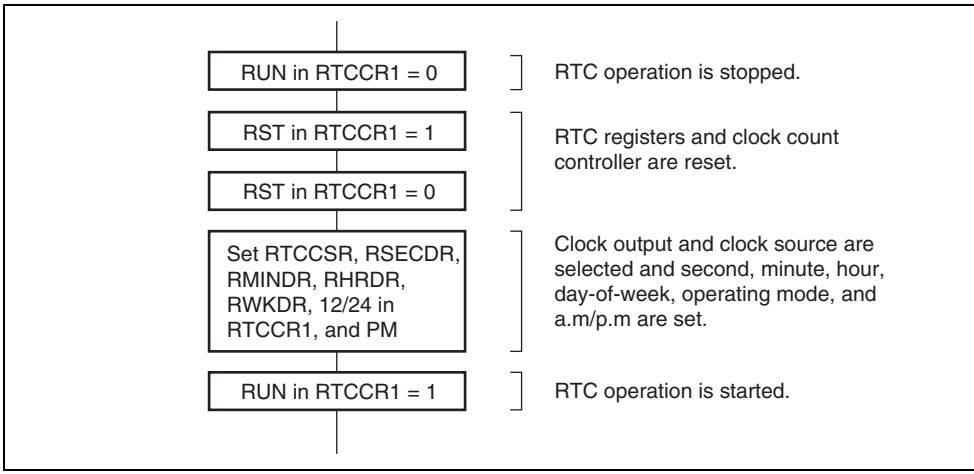


Figure 11.3 Initial Setting Procedure

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, hour, and day-of-week registers after the relevant flag in RTCFLG is set to 1 and the is confirmed to be 0.

   When INT in RTCCR1 is set to 1 and an interrupt is used, read from the second, min 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 i no change in the read data, the read data is used.

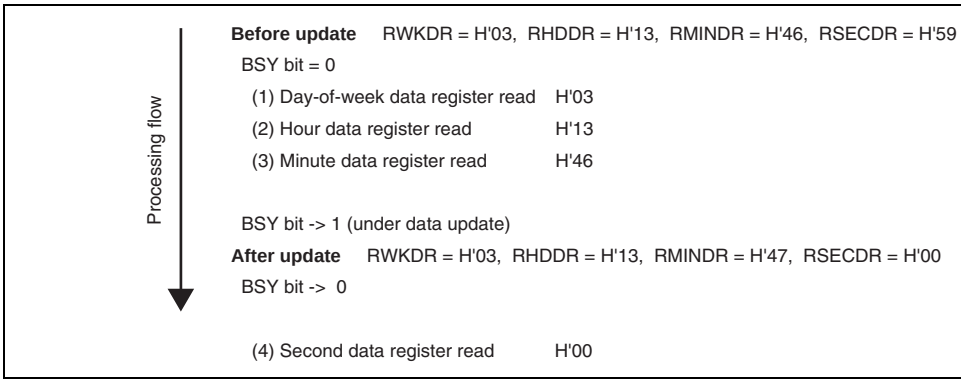


Figure 11.4 Example: Reading of Inaccurate Time Data

Table 11.2 Interrupt Sources

Interrupt Name	Interrupt Source	Interrupt En
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

their values are undefined after power-on.

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

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/819 and ϕ_W/256) or the on-chip oscillator (R_{OSC}/2048) can be selected as the timer-counte
- 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 inde when not used. (The WDT is operating as the initial value. For details, refer to sectio Module Standby Function.)

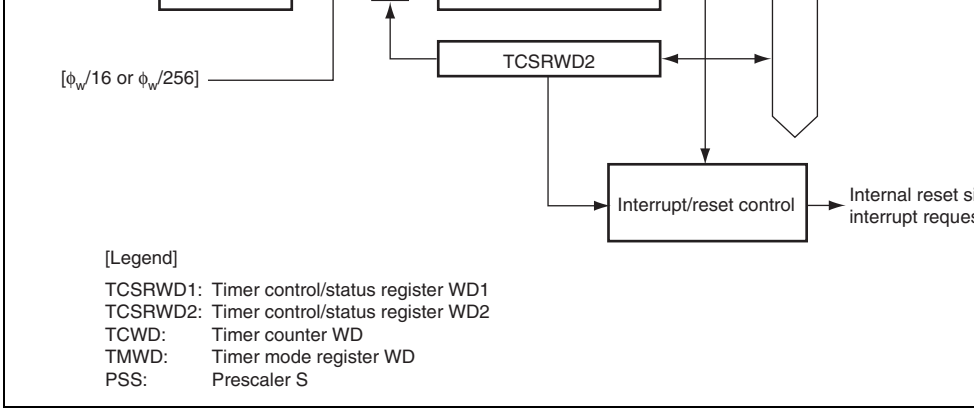


Figure 12.1 Block Diagram of Watchdog Timer

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)

				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 se When writing data to this bit, the write value for be 0.
5	B4WI	1	R/W	Bit 4 Write Inhibit The TCSRWE bit can be written only when the value of the B4WI bit is 0. This bit is always rea
4	TCSRWE	0	R/W	Timer Control/Status Register WD Write Enable The WDON and WRST bits can be written whe TCSRWE bit is set to 1. When writing data to this bit, the write value for be 0.
3	B2WI	1	R/W	Bit 2 Write Inhibit The WDON bit can be written only when the wr of the B2WI bit is 0. This bit is always read as 1

				• When 0 is written to the WDON bit and 0 to t 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 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 watchdo is generated. This bit is not cleared by a reset ca the watchdog timer. [Setting condition] When TCWD overflows and an internal reset sig generated [Clearing conditions] • Reset by $\overline{\text{RES}}$ pin • When 0 is written to the WRST bit and 0 to th bit while the TCSRWE bit is 1

				[Setting condition] When TCWD overflows (changes from H'FF to When internal reset request generation is selec watchdog timer mode, this bit is cleared automa the internal reset after it has been set. [Clearing condition] • When TCSRWD2 is read when OVF = 1, th written to OVF
6	B5WI	1	R/(W)*2	Bit 5 Write Inhibit The WT/$\overline{IT}$ bit can be written only when the writ 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 watchdo 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 wri 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 reque interval timer mode. 0: Disables an overflow interrupt 1: Enables an overflow interrupt

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

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, se 21, Electrical Characteristics.
				In active (medium-speed), sleep (medium-spe subactive, and subsleep modes, the 00xx valu interval timer mode cannot be set simultaneou
				In subactive and subsleep modes, when the s frequency is ϕ_w/8, the 010x value and the inte mode cannot be set simultaneously.

[Legend] x: Don't care.

reset signal is output for a period of 512 clock cycles by the on-chip oscillator (R_{OSC}). TCW
writable counter, and when a value is set in TCWD, the count-up starts from that value. A
overflow period in the range of 1 to 256 input clock cycles can therefore be set, according
TCWD set value.

Figure 12.2 shows an example of watchdog timer operation.

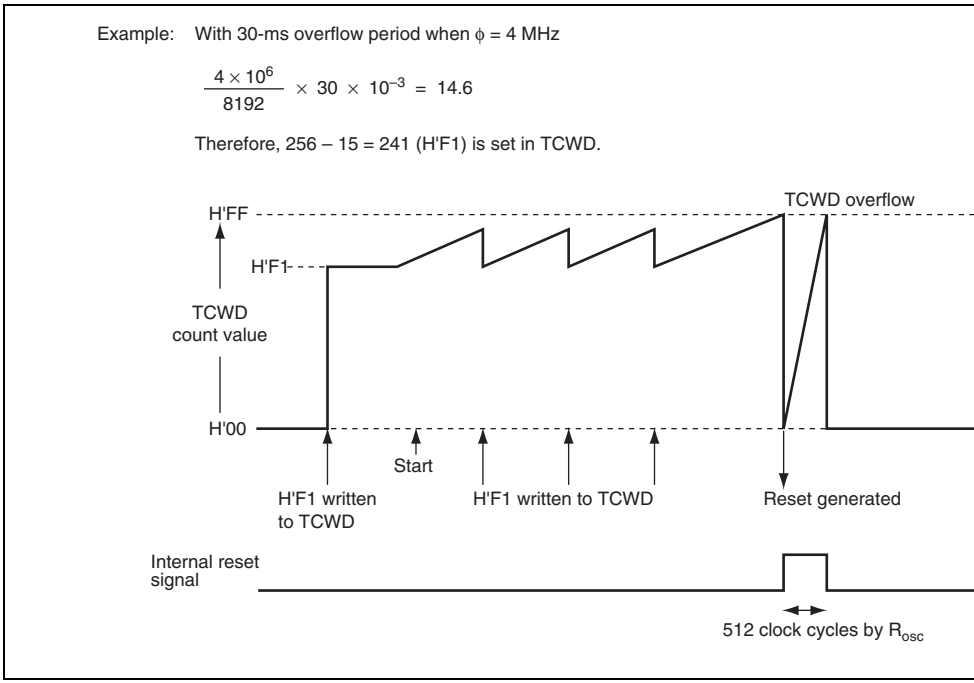


Figure 12.2 Example of Watchdog Timer Operation

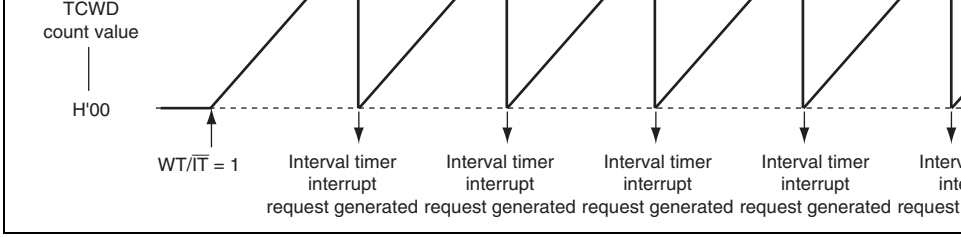


Figure 12.3 Interval Timer Mode Operation

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 se
TCWD overflows. At the same time, a reset signal is output in watchdog timer mode an
interval timer interrupt is generated in interval timer mode.

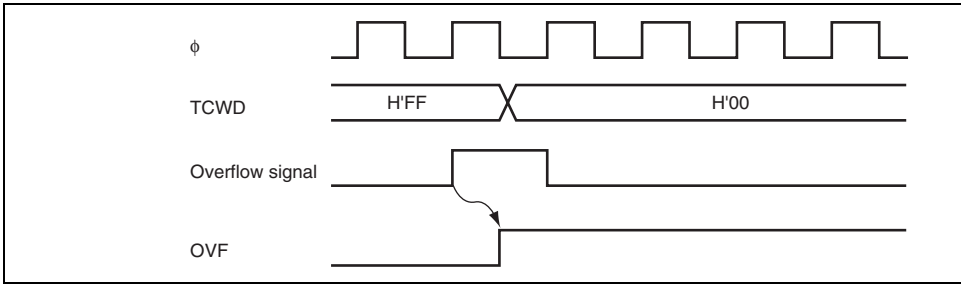


Figure 12.4 Timing of OVF Flag Setting

If modes are switched between watchdog timer and interval timer, while the WDT is oper
error may occur in the count value. Software must stop the watchdog timer (by clearing th
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
WD1 (TCSRWD1) is cleared to 0. The WDCKSTP bit can be cleared to 0 while the WD
set to 1 (while the watchdog timer is operating). However, the watchdog timer does not e
module standby mode but continues operating. When the WDON bit is cleared to 0 by so
after the watchdog timer stops operating, the WDCKSTP bit is valid at the same time and
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 (TCSRW
0, the corresponding bit may not be cleared to 0 depending on the program address. In pa
if lower two bits in the address of the MOV.B instruction to transfer a value to TCSRWD
B'10, the WT/$\overline{\text{IT}}$ or IEOVF bit is successfully cleared to 0, whereas if lower two bits in th
are B'00, the WT/$\overline{\text{IT}}$ or IEOVF bit may not be cleared to 0. To avoid this failure, make su
the assembly program shown in table 12.1, when clearing the WT/$\overline{\text{IT}}$ or IEOVF bit to 0. S
TCSRWD2 by the 8-bit absolute address, and LABEL by the 16-bit absolute address. Do
change nor add instructions. The value of "xx" in line 1 and line 4 must be set according t
12.2. Use an arbitrary 8-bit general register for Rn and Rm. In addition, Address1 in tabl
shows an example when the WT/$\overline{\text{IT}}$ or IEOVF bit is cleared to 0 successfully by the MOV
instruction in line 2. Address2 in table 12.1 shows an example when the WT/$\overline{\text{IT}}$ or IEOVF
to be cleared to 0 by the MOV.B instruction in line 2, but cleared to 0 by the MOV.B inst
in line 6.

H'00AE	H'0240	LABEL	NOP

Table 12.2 The Value of "xx"

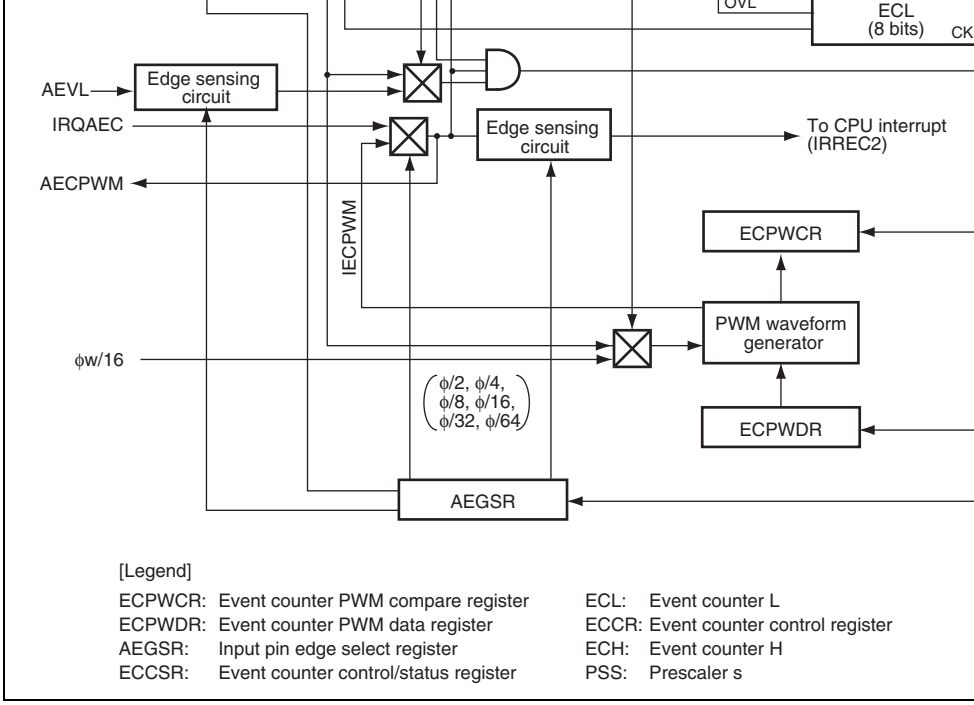
Bit(s) Cleared to 0	The Value of "xx" in Line 1	The Value of "xx" in
Both WT/$\overline{IT}$ and IEOVF	07	28
Only WT/$\overline{IT}$	17	20
Only IEOVF	47	08

clocks (ϕ) or subclocks (ϕ_{SUB}).

- Can be used as two-channel independent 8-bit event counter or single-channel indep bit event counter.
- Event/clock input is enabled when IRQAEC goes high or event counter PWM outpu (IECPWM) goes high.
- Both rising and falling edge sensing can be used for IRQAEC or event counter PWM (IECPWM) interrupts. When the asynchronous counter is not used, they can be used independent interrupts.
- When an event counter PWM is used, event clock input enabling/disabling can be co 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 inde when not used. (The asynchronous event counter is halted as the initial value. For de to section 5.4, Module Standby Function.)



[Legend]

ECPWCR:	Event counter PWM compare register	ECL:	Event counter L
ECPWDR:	Event counter PWM data register	ECCR:	Event counter control register
AEGSR:	Input pin edge select register	ECH:	Event counter H
ECCSR:	Event counter control/status register	PSS:	Prescaler s

Figure 13.1 Block Diagram of Asynchronous Event Counter

Event input enable interrupt input	IRQAEC	Input	Input pin for interrupt enabling event in
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)

11	ECPWCR11	1	R/W	When changing the conversion period, the eve counter PWM must be halted by clearing the E bit in AEGSR to 0 before modifying ECPWCR.
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	

11	ECPWDR11	0	W	When changing the conversion cycle, the eve counter PWM must be halted by clearing the bit in AEGSR to 0 before modifying ECPWDR
10	ECPWDR10	0	W	
9	ECPWDR9	0	W	
8	ECPWDR8	0	W	The read value is undefined.
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	

				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 tl 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 tl 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 s 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 sh written to.

				10: ϕ/4
				11: ϕ/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: ϕ/2
				10: ϕ/4
				11: ϕ/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: ϕ/2
				001: ϕ/4
				010: ϕ/8
				011: ϕ/16
				100: ϕ/32
				101: ϕ/64
				110: ϕ_w/16
				111: Setting prohibited
				When changing the event counter PWM clock ECPWME bit in AEGSR must be cleared to 0 the PWM before rewriting this setting.
0	—	0	R/W	Reserved
				Although this bit is readable/writable, only 0 s written to.

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

				1: ECH reset is cleared and count-up function 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 enabled

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

5	ECH5	0	R	ECH can be cleared to H'00 when the CRCH b
4	ECH4	0	R	ECCSR is cleared to 0.
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. also operates as the lower 8-bit up-counter of a 16-bit event counter configured in combi with ECH.

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

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

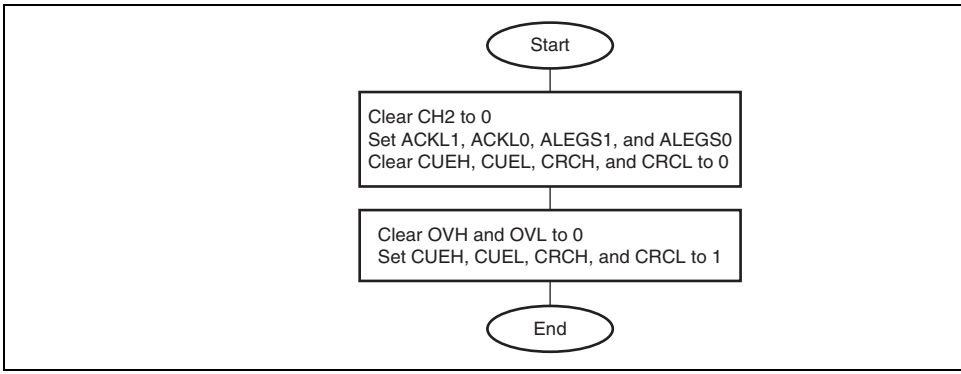


Figure 13.2 Software Procedure when Using ECH and ECL as 16-Bit Event C

As CH2 is cleared to 0 by a reset, ECH and ECL operate as a 16-bit event counter after
and as ACKL1 and ACKL0 are cleared to B′00, the operating clock is asynchronous eve
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,
ECL overflow from H'FFFF to H'0000, the OVH flag is set to 1 in ECCSR, the ECH an
count values each return to H'00, and counting up is restarted. When an overflow occurs
IRREC bit is set to 1 in IRR2. If the IENEC bit in IENR2 is 1 at this time, an interrupt r
sent to the CPU.

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

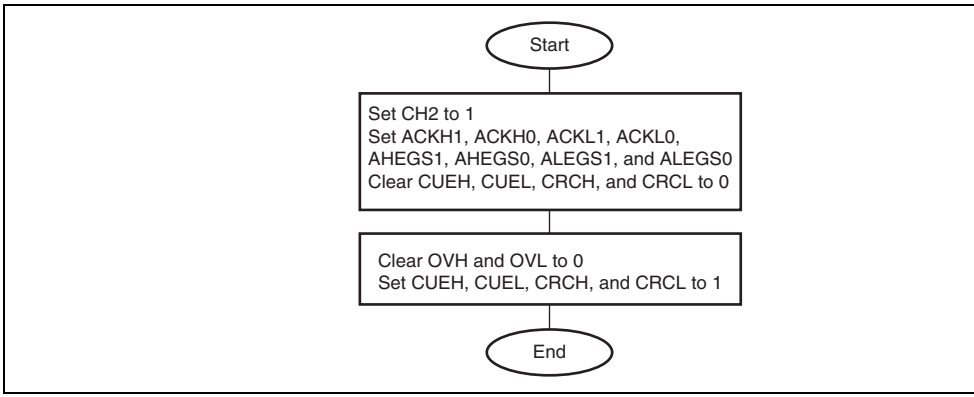


Figure 13.3 Software Procedure when Using ECH and ECL as 8-Bit Event Cou

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

interrupt request is sent to the CPU.

Rising, falling, or both edge sensing can be selected for the IRQAEC input pin with bits
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 wh
counter PWM output (IECPWM) is high. When IECPWM is low, the input clocks are n
the counters, and so ECH and ECL do not count. ECH and ECL count operations can th
controlled cyclically by controlling event counter PWM. In this case, ECH and ECL car
controlled individually.

IECPWM can also operate as an interrupt source.

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

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

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-sp
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 = toff
ϕ/2	0.5 µs	H'7A11	H'16E3	2.93 ms	15.625 ms	12.69
ϕ/4	1 µs	D'31249	D'5859	5.86 ms	31.25 ms	25.39
ϕ/8	2 µs			11.72 ms	62.5 ms	50.78
ϕ/16	4 µs			23.44 ms	125.0 ms	101.5
ϕ/32	8 µs			46.88 ms	250.0 ms	203.1
ϕ/64	16 µs			93.76 ms	500.0 ms	406.2
ϕ$_w$/16	488 µs			2861.59 ms	15260.19 ms	1239

Note: * toff minimum width

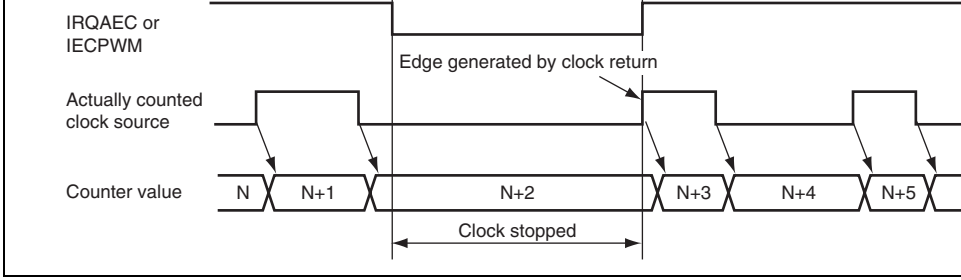


Figure 13.5 Example of Clock Control Operation

ECL	AEVL											
	ϕ/2, ϕ/4, ϕ/8	o	o	o	o	×	×	×	×	×	×	×
PWM	ϕw/16	o	o	o	o	o	o	o	×	×	o	o
	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.

Table 13.4 Maximum Clock Frequency

Mode		Maximum Clock Fre Input to AEVH/AEVI
Active (high-speed), sleep (high-speed)		4 to 10 MHz (2.7 to 3
		2 to 4.2 MHz (1.8 to 3
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 EC second, or set both CUEH and CRCH to 1 at same time before clock input. When A operating on 16-bit mode, do not change CUEH. Otherwise, ECH will be miscounte
4. When ECPWME in AEGSR is 1, the event counter PWM is operating and therefore and ECPWDR should not be modified.

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

14.1 Features

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

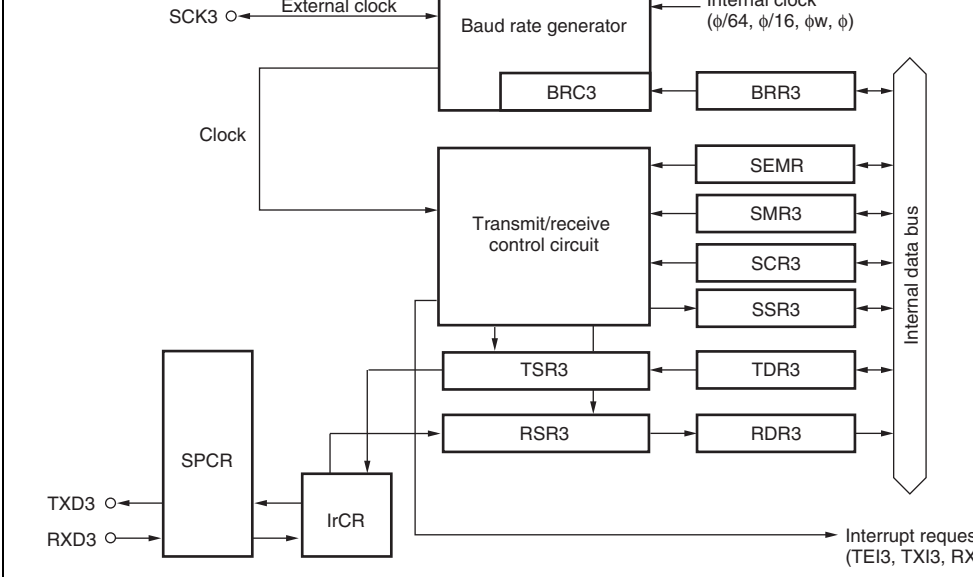
  The transmitter and receiver are mutually independent, enabling transmission and re
  be executed simultaneously.

  Double-buffering is used in both the transmitter and the receiver, enabling continuou
  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 tra
  clock source.
- Six interrupt sources

  Transmit-end, transmit-data-empty, receive-data-full, overrun error, framing error, a
  error.
- Use of module standby mode enables this module to be placed in standby mode inde
  when not used. (The SCI3 is halted as the initial value. For details, refer to section 5.
  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
  framing error



[Legend]
RSR3: Receive shift register 3
RDR3: Receive data register 3
TSR3: Transmit shift register 3
TDR3: Transmit data register 3
SMR3: Serial mode register 3
SCR3: Serial control register 3
SSR3: Serial status register 3
BRR3: Bit rate register 3
BRC3: Bit rate counter 3
SPCR: Serial port control register
IrCR: IrDA control register
SEMR: Serial extended mode register

Figure 14.1 Block Diagram of SCI3

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, BRR in the text.

receive-enabled. As RSR and RDR function as a double buffer in this way, continuous re
operations are possible. After confirming that the RDRF bit in SSR is set to 1, read RDR
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

14.3.3 Transmit Shift Register (TSR)

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

14.3.4 Transmit Data Register (TDR)

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

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

				1: Clock synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchron 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) i not transmitted. To select 5 bits as the data le 1 to both the PE and MP bits. The three most bits (bits 7, 6, and 5) in TDR are not transmitt clock synchronous mode, the data length is fi bits regardless of the CHR bit setting.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous When this bit is set to 1, the parity bit is adde transmit data before transmission, and the pa checked in reception. In clock synchronous m parity bit addition and checking is not perform regardless of the PE bit setting.

				an even number. When odd parity is selected, a parity bit is add transmission so that the total number of 1 bits transmit data plus the parity bit is an odd numb reception, a check is carried out to confirm tha number of 1bits in the receive data plus the pa an odd number. If parity bit addition and checking is disabled in synchronous mode and asynchronous mode, t setting is invalid.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous 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 is 0, it is treated as the start bit of the next tran character.
2	MP	0	R/W	5-Bit Communication When this bit is set to 1, the 5-bit communicati is enabled. Make sure to set bit 5 (PF) to 1 wh this bit (MP) to 1.

	subsleep mode, the SCI3 can be used only w selected for the CPU operating clock. For the relationship between the bit rate regis and the baud rate, see section 14.3.8, Bit Rat (BRR). n is the decimal representation of the in BRR (see section 14.3.8, Bit Rate Register

14.3.6 Serial Control Register (SCR)

SCR enables or disables SCI3 transfer operations and interrupt requests, and selects the 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

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 req enabled. TXI3 can be released by clearing the or TI bit to 0.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, the RXI3 and ERI3 in requests are enabled. RXI3 and ERI3 can be released by clearing th bit or the FER, PER, or OER error flag to 0, o clearing the RIE bit to 0.

				When this bit is set to 1, reception is enabled. I state, serial data reception is started when a st detected in asynchronous mode or serial clock detected in clock synchronous mode. Be sure t out the SMR settings to decide the reception fo before setting bit RE to 1. Note that the RDRF, FER, PER, and OER flag are not affected when bit RE is cleared to 0, an 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 requ enabled. TEI3 can be released by clearing bit T 0 and clearing bit TEND to 0 in SSR, or by clea TEIE to 0.

times the bit rate from the SCK3 pin)

11: Reserved

Clock synchronous mode:

00: Internal clock (SCK3 pin functions as cloc

01: Reserved

10: External clock (SCK3 pin functions as clo

11: Reserved

				[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 TD • 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 RD [Setting condition] • When serial reception ends normally and re data is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RD When data is read from RDR If an error is detected in reception, or if the RE SCR has been cleared to 0, RDR and bit RDR affected and retain their previous state. Note that if data reception is completed while b is still set to 1, an overrun error (OER) will occu the receive data will be lost.

				data it held before the overrun error occurred, received after the error is lost. Reception can continued with bit OER set to 1, and in clock synchronous mode, transmission cannot be c 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 FE When bit RE in SCR is cleared to 0, bit FER i affected and retains its previous state. Note that, in 2-stop-bit mode, only the first sto checked for a value of 1, and the second stop checked. When a framing error occurs, the re is transferred to RDR but bit RDRF is not set. Reception cannot be continued with bit FER s clock synchronous mode, neither transmissio reception is possible when bit FER is set to 1.

				still transferred to RDR, but bit RDRF is not Reception cannot be continued with bit PER 1. In clock synchronous mode, neither trans nor reception is possible when bit PER is s
2	TEND	1	R	Transmit End [Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last 1-byte serial transmit character [Clearing conditions] • When 0 is written to TDRE after reading TD • When the transmit data is written to TDR
1	MPBR	0	R	Reserved This bit is always read as 0 and cannot be mod
0	MPBT	0	R/W	Reserved The write value should always be 0.

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

operating frequencies and bit rates can be obtained by the following formulas.

[Asynchronous Mode and ABCS Bit is 0]

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

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

[Asynchronous Mode and ABCS Bit is 1]

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

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

[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)

600	—	—	—	0	1	0.00	0	103	0.16	0	108
1200	—	—	—	0	0	0.00	0	51	0.16	0	54
2400	—	—	—	—	—	—	0	25	0.16	0	26
4800	—	—	—	—	—	—	0	12	0.16	0	13
9600	—	—	—	—	—	—	—	—	—	0	6
19200	—	—	—	—	—	—	—	—	—	—	—
31250	—	—	—	—	—	—	0	1	0.00	—	—
38400	—	—	—	—	—	—	—	—	—	—	—

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

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

600	0	217	0.21	0	255	0.00	2	15	1.73	2	19
1200	0	108	0.21	0	127	0.00	0	129	0.16	0	155
2400	0	54	–0.70	0	63	0.00	0	64	0.16	0	77
4800	0	26	1.14	0	31	0.00	0	32	–1.36	0	38
9600	0	13	–2.48	0	15	0.00	0	15	1.73	0	19
19200	0	6	–2.48	0	7	0.00	0	7	1.73	0	9
31250	—	—	—	0	4	–1.70	0	4	0.00	0	5
38400	—	—	—	0	3	0.00	0	3	1.73	0	4

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

	6.144 MHz			7.3728 MHz			8 MHz			9.8304 MHz			1	
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	2	108	0.08	2	130	–0.07	2	141	0.03	2	174	–0.26	2	1
150	2	79	0.00	2	95	0.00	2	103	0.16	2	127	0.00	2	1
200	2	59	0.00	2	71	0.00	2	77	0.16	2	95	0.00	2	9
250	2	47	0.00	2	57	–0.69	2	62	-0.79	2	76	-0.26	2	7
300	2	39	0.00	2	47	0.00	2	51	0.16	2	63	0.00	2	6
600	2	19	0.00	2	23	0.00	2	25	0.16	2	31	0.00	2	3
1200	0	159	0.00	0	191	0.00	0	207	0.16	0	255	0.00	2	1
2400	0	79	0.00	0	95	0.00	0	103	0.16	0	127	0.00	0	1
4800	0	39	0.00	0	47	0.00	0	51	0.16	0	63	0.00	0	6
9600	0	19	0.00	0	23	0.00	0	25	0.16	0	31	0.00	0	3
19200	0	9	0.00	0	11	0.00	0	12	0.16	0	15	0.00	0	1
31250	0	5	2.40	—	—	—	0	7	0.00	0	9	–1.70	0	9
38400	0	4	0.00	0	5	0.00	—	—	—	0	7	0.00	0	7

600	—	—	—	0	3	0.00	0	207	0.16	0	217
1200	—	—	—	0	1	0.00	0	103	0.16	0	108
2400	—	—	—	0	0	0.00	0	51	0.16	0	54
4800	—	—	—	—	—	—	0	25	0.16	0	26
9600	—	—	—	—	—	—	0	12	0.16	0	13
19200	—	—	—	—	—	—	—	—	—	0	6
31250	—	—	—	—	—	—	0	3	0.00	—	—
38400	—	—	—	—	—	—	—	—	—	—	—

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

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

600	2	26	1.14	2	31	0.00	2	32	–1.36	2	38
1200	0	217	0.21	0	255	0.00	2	15	1.73	2	19
2400	0	108	0.21	0	127	0.00	0	129	0.16	0	155
4800	0	54	–0.70	0	63	0.00	0	64	0.16	0	77
9600	0	26	1.14	0	31	0.00	0	32	–1.36	0	38
19200	0	13	–2.48	0	15	0.00	0	15	1.73	0	19
31250	—	—	—	0	9	–1.70	0	9	0.00	0	11
38400	0	6	-2.48	0	7	0.00	0	7	1.73	0	9

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

	6.144 MHz			7.3728 MHz			8 MHz			9.8304 MHz			1	
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	2	217	0.08	3	64	0.70	3	70	0.03	3	86	0.31	3	8
150	2	159	0.00	2	191	0.00	2	207	0.16	2	255	0.00	3	6
200	2	119	0.00	2	143	0.00	2	155	0.16	2	191	0.00	2	1
250	2	95	0.00	2	114	0.17	2	124	0.00	2	153	-0.26	2	1
300	2	79	0.00	2	95	0.00	2	103	0.16	2	127	0.00	2	1
600	2	39	0.00	2	47	0.00	2	51	0.16	2	63	0.00	2	6
1200	2	19	0.00	2	23	0.00	2	25	0.16	2	31	0.00	2	3
2400	0	159	0.00	0	191	0.00	0	207	0.16	0	255	0.00	2	1
4800	0	79	0.00	0	95	0.00	0	103	0.16	0	127	0.00	0	1
9600	0	39	0.00	0	47	0.00	0	51	0.16	0	63	0.00	0	6
19200	0	19	0.00	0	23	0.00	0	25	0.16	0	31	0.00	0	3
31250	0	11	2.40	0	14	–1.70	0	15	0.00	0	19	–1.70	0	1
38400	0	9	0.00	0	11	0.00	0	12	0.16	0	15	0.00	0	1

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

5k	—	—	—	—	—	—	0	99	0.0
10k	—	—	—	—	—	—	0	49	0.0
25k	—	—	—	—	—	—	0	19	0.0
50k	—	—	—	—	—	—	0	9	0.0
100k	—	—	—	—	—	—	0	4	0.0
250k	—	—	—	—	—	—	0	1	0.0
500k	—	—	—	—	—	—	0*	0*	0.0
1M	—	—	—	—	—	—	—	—	—

Note: * Continuous transmission/reception is not possible.

10k	0	99	0.00	0	199	0.00	0	249	0.0
25k	0	39	0.00	0	79	0.00	0	99	0.0
50k	0	19	0.00	0	39	0.00	0	49	0.0
100k	0	9	0.00	0	19	0.00	0	24	0.0
250k	0	3	0.00	0	7	0.00	0	9	0.0
500k	0	1	0.00	0	3	0.00	0	4	0.0
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 clock is ϕ_w.

5		0		Reserved This bit is always read as 0 and cannot be mo
4	SPC3	0	R/W	P32/TXD3/IrTXD Pin Function Switch Selects whether pin P32/TXD3/IrTXD is used 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 b modified.
1	SCINV1	0	R/W	TXD3/IrTXD Pin Output Data Inversion Switch Selects whether output data of the TXD3/IrTX inverted or not. 0: Output data of TXD3/IrTXD pin is not inver 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 inverted or not. 0: Input data of RXD3/IrRXD pin is not inverte 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 point is inverted immediately after the modification, and an invalid data change is output. When modifying the serial port control register, modification must be mad in which data changes are invalidated.

				1: TXD3/IrTXD and RXD3/IrRXD pins function and IrRXD
6	IrCKS2	0	R/W	IrDA Clock Select
5	IrCKS1	0	R/W	If the IrDA function is enabled, these bits set th
4	IrCKS0	0	R/W	pulse width when encoding the IrTXD output p
				000: Bit rate × 3/16
				001: ϕ/2
				010: ϕ/4
				011: ϕ/8
				100: ϕ/16
				101: Setting prohibited
				11x: Setting prohibited
3 to 0	—	All 0	—	Reserved
				These bits are always read as 0 and cannot be modified.

[Legend] x: Don't care.

				Selects the basic clock for the bit period in asynchronous mode. This setting is enabled only in asynchronous (COM bit in SMR3 is 0). 0: Operates on a basic clock with a frequency times the transfer rate 1: Operates on a basic clock with a frequency times the transfer rate Clear the ABCS bit to 0, when the IrDA functi enabled.
2 to 0	—	All 0	—	Reserved These bits are always read as 0 and cannot b modified.

14.4 Operation in Asynchronous Mode

Figure 14.2 shows the general format for asynchronous serial communication. One fram of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low l finally stop bits (high level). In asynchronous mode, synchronization is performed at the edge of the start bit during reception. The data is sampled on the 8th pulse of a clock wit frequency 16 times the bit period, so that the transfer data is latched at the center of each 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 ind units, enabling full duplex. Both the transmitter and the receiver also have a double-buff structure, so data can be read or written during transmission or reception, enabling conti transfer. Table 14.8 shows the 16 data transfer formats that can be set in asynchronous m format is selected by the settings in SMR as shown in table 14.9.

the SCK3 pin can be selected as the SCI3's serial clock source, according to the setting o
COM bit in SMR and the CKE0 and CKE1 bits in SCR. When an external clock is input
SCK3 pin, the clock frequency should be 16 times the bit rate used (when the ABCS bit i
is 1, the clock frequency should be eight times the bit rate used). For details on selection
clock source, see table 14.10. When the SCI3 is operated on an internal clock, the clock c
output from the SCK3 pin. The frequency of the clock output in this case is equal to the b
and the phase is such that the rising edge of the clock is in the middle of the transfer data,
shown in figure 14.3.

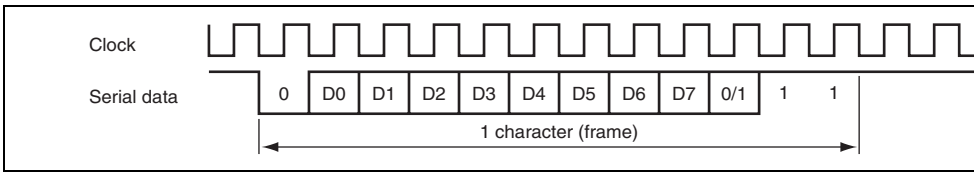
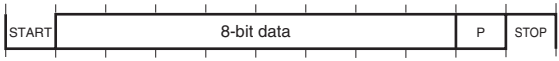
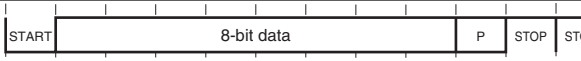
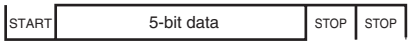
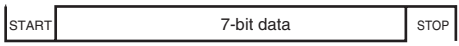
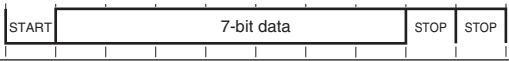
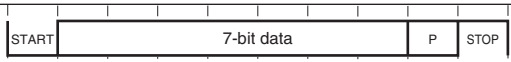
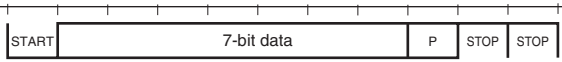
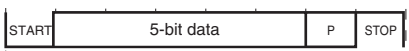
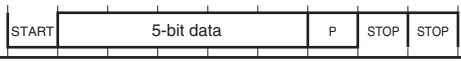


Figure 14.3 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode) (Example with 8-Bit Data, Parity, Two Stop Bits)

0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	Setting prohibited
1	0	1	1	Setting prohibited
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

[Legend]

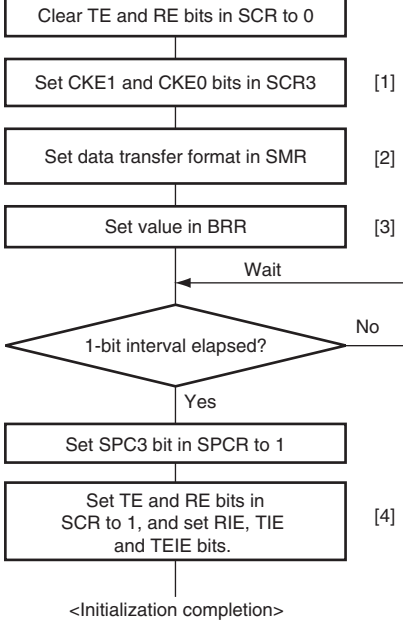
START: Start bit
STOP: Stop bit
P: Parity bit

				1			
			1	0			Yes
				1			
	0	1	0	0			Setting prohibited
				1			
			1	0		5-bit data	No
				1			
	1		0	0			Setting prohibited
				1			
			1	0		5-bit data	Yes
				1			
1	x	0	x	x	Clock synchronous mode	8-bit data	No

[Legend] x: Don't care.

1	0	0	Clock synchronous mode	Internal	Outputs the serial cloc
	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 rate.



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

transmit data to TDR before transmission of the current transmit data has been comp

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, a 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 state" is entered, in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a interrupt request is generated.
6. Figure 14.6 shows a sample flowchart for transmission in asynchronous mode.

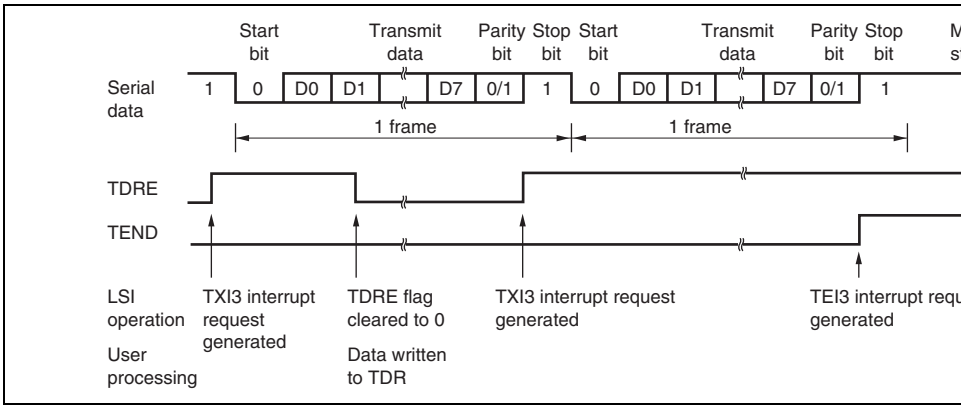
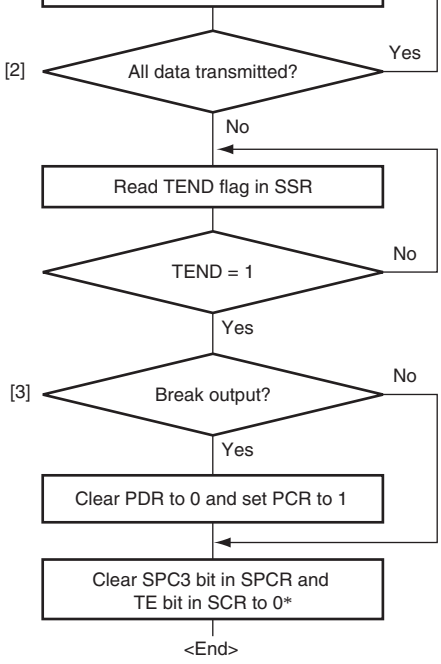


Figure 14.5 Example SCI3 Operation in Transmission in Asynchronous M (8-Bit Data, Parity, One Stop Bit)



[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 the pin functions as an I/O port.

Figure 14.6 Sample Serial Transmission Flowchart (Asynchronous Mode)

- Stop bit check

  The SCI3 checks that the stop bit is 1. If two stop bits are used, only the first is chec
- Status check

  The SCI3 checks that bit RDRF is set to 0, indicating that the receive data can be tra from RSR to RDR.

2. If an overrun error occurs (when reception of the next data is completed while the R 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 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 transfe RDR. If the RIE bit in SCR is set to 1 at this time, an ERI3 interrupt request is gener
4. If a framing error is detected (when the stop bit is 0), the FER bit in SSR is set to 1 a data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI3 inte request is generated.
5. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive d transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI3 interrupt re generated. Continuous reception is possible because the RXI3 interrupt routine reads receive data transferred to RDR before reception of the next receive data has been co

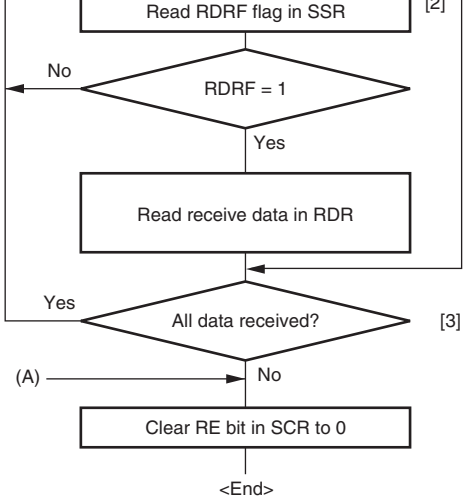
Figure 14.7 Example SCI3 Operation in Reception in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)

Table 14.11 shows the states of the SSR status flags and receive data handling when a rec error is detected. If a receive error is detected, the RDRF flag retains its state before recei data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clea OER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 14.8 shows a sam 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 + frami
1	1	0	1	Lost	Overrun error + parity
0	0	1	1	Transferred to RDR	Framing error + parity
1	1	1	1	Lost	Overrun error + frami + parity error

Note: * The RDRF flag retains the state it had before data reception. However, note th is read after an overrun error has occurred in a frame because reading of the r data in the previous frame was delayed, the RDRF flag will be cleared to 0.



the error. After performing the
appropriate error processing, e
that the OER, PER, and FER fl
all cleared to 0. Reception can
resumed if any of these flags a
1. In the case of a framing erro
break can be detected by readi
value of the input port correspo
the RXD3 pin.

Figure 14.8 Sample Serial Data Reception Flowchart (Asynchronous Mode)

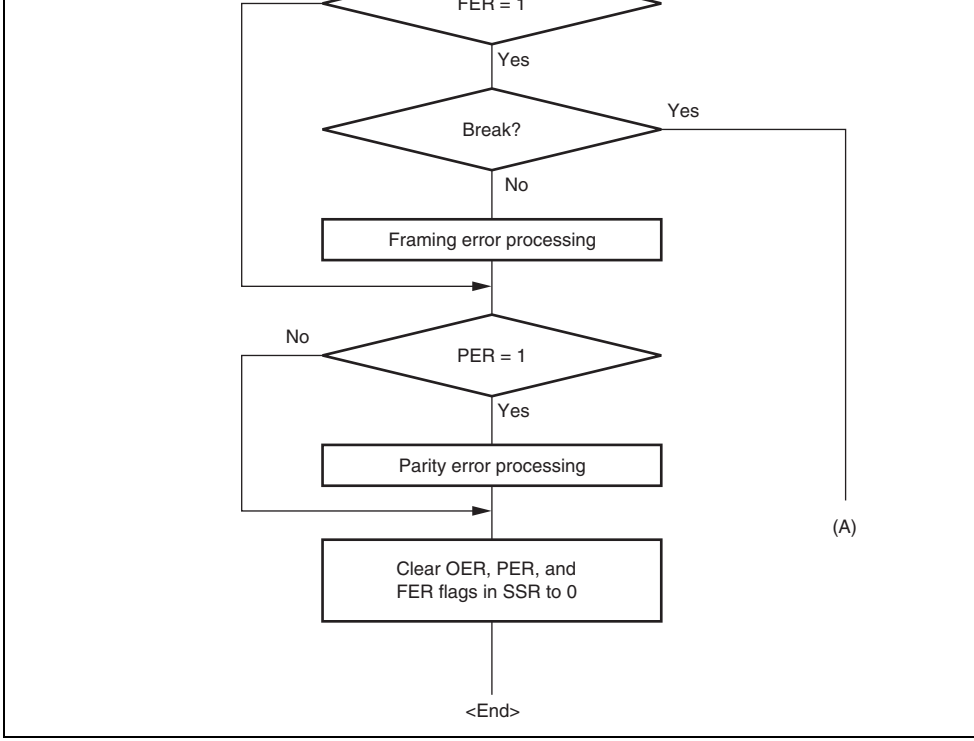


Figure 14.8 Sample Serial Data Reception Flowchart (Asynchronous Mode)

read or written during transmission or reception, enabling continuous data transfer.

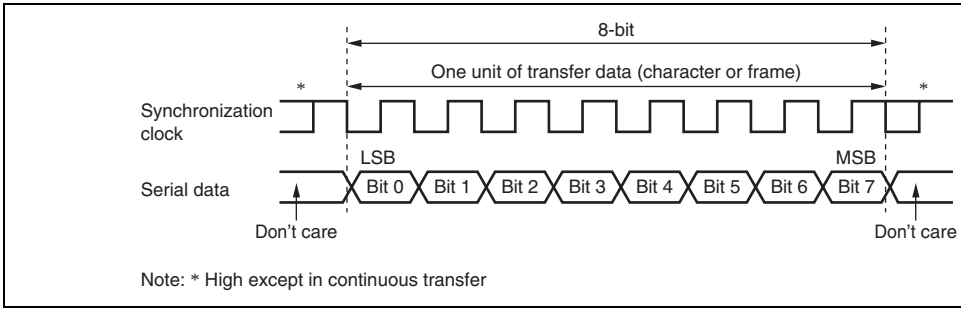


Figure 14.9 Data Format in Clock Synchronous Communication

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 bit in SMR and CKE0 and CKE1 bits in SCR. When the SCI3 is operated on an internal serial clock is output from the SCK3 pin. Eight serial clock pulses are output in the trans 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 s flowchart in figure 14.4.

mode has been specified, and synchronized with the input clock when use of an exter
has been specified. Serial data is transmitted sequentially from the LSB (bit 0), from t
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 tran
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 ma
the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI3 in
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 fla
cleared to 0, transmission will not start while a receive error flag (OER, FER, or PER) is
Make sure that the receive error flags are cleared to 0 before starting transmission.

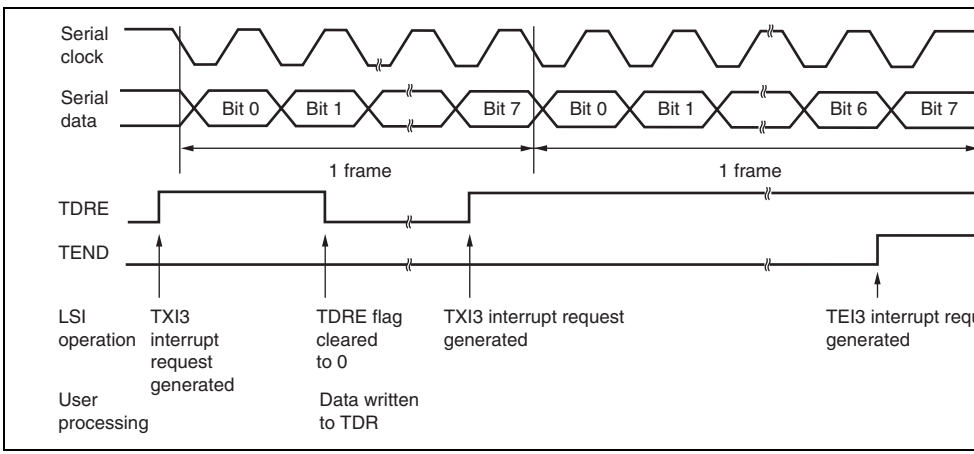
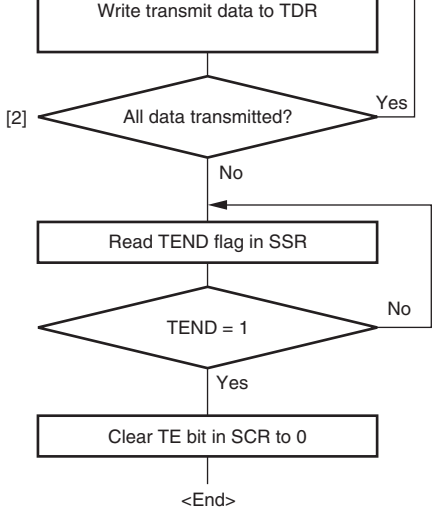


Figure 14.10 Example of SCI3 Operation in Transmission in Clock Synchronous



When data is written to TDR, the TDRE f
is automatically cleared to 0.

Figure 14.11 Sample Serial Transmission Flowchart (Clock Synchronous M

time, an ERI3 interrupt request is generated, receive data is not transferred to RDR, a
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 da
transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI3 interrupt re
generated.

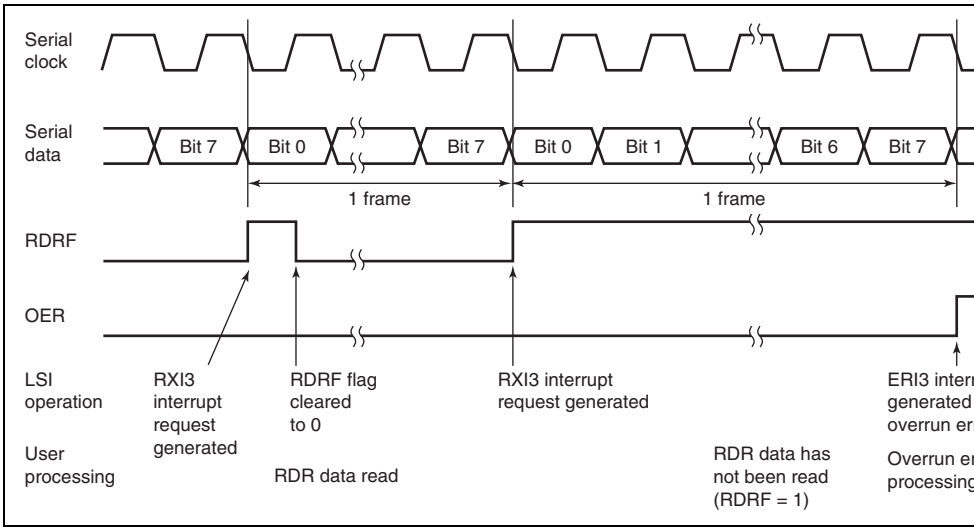
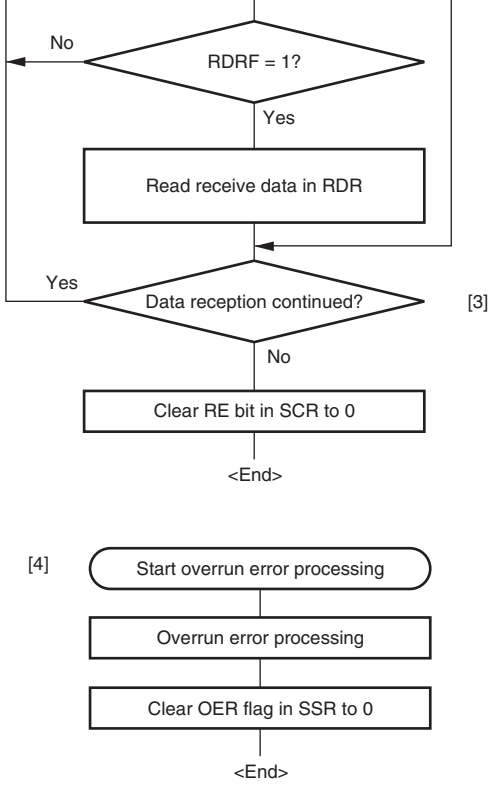


Figure 14.12 Example of SCI3 Reception Operation in Clock Synchronous M

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the
FER, PER, and RDRF bits to 0 before resuming reception. Figure 14.13 shows a sample
for serial data reception.



cleared to 0.

[4] If an overrun error occurs, read the
flag in SSR, and after performing th
appropriate error processing, clear t
flag to 0. Reception cannot be resu
the OER flag is set to 1.

Figure 14.13 Sample Serial Reception Flowchart (Clock Synchronous Mod

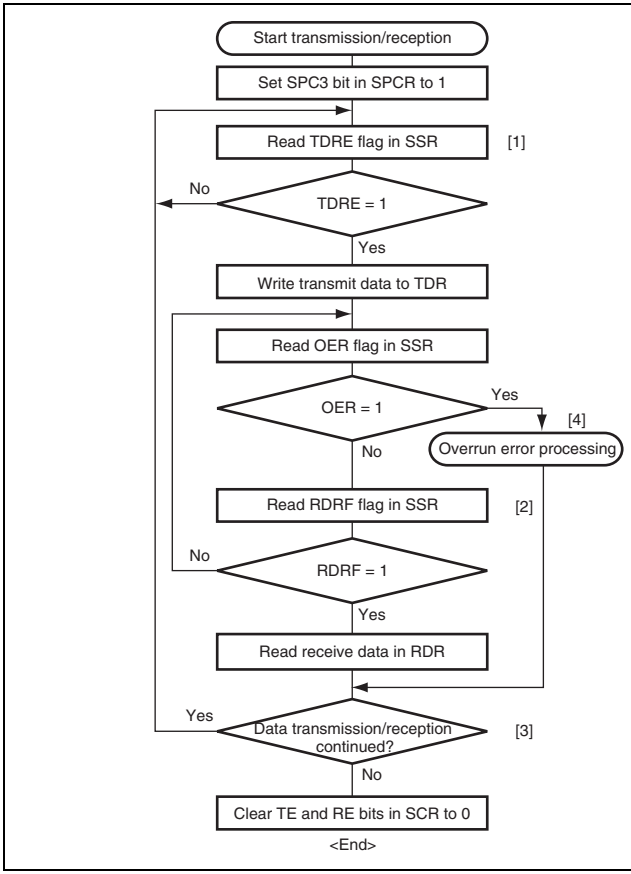


Figure 14.14 Sample Flowchart of Simultaneous Serial Transmit and Receive Op (Clock Synchronous Mode)

of 9600 bps, which can be modified as required. The IrDA interface provided by this LS
incorporate the capability of automatic modification of the transfer rate; the transfer rate
modified through programming.

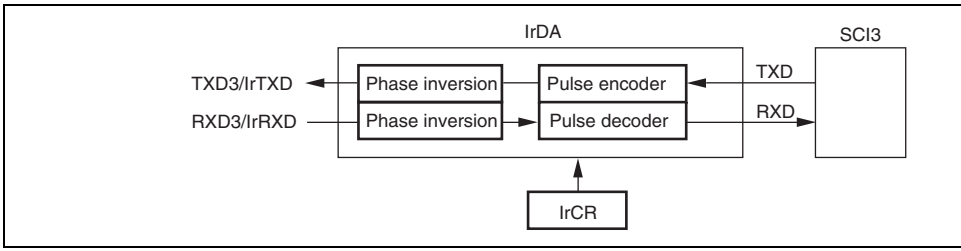


Figure 14.15 IrDA Block Diagram

frequency of system clock ϕ is 10 MHz, being equal to or greater than 1.41 µs, the high-l
width at minimum can be specified as 1.6 µs.

For serial data of level 1, no pulses are output.

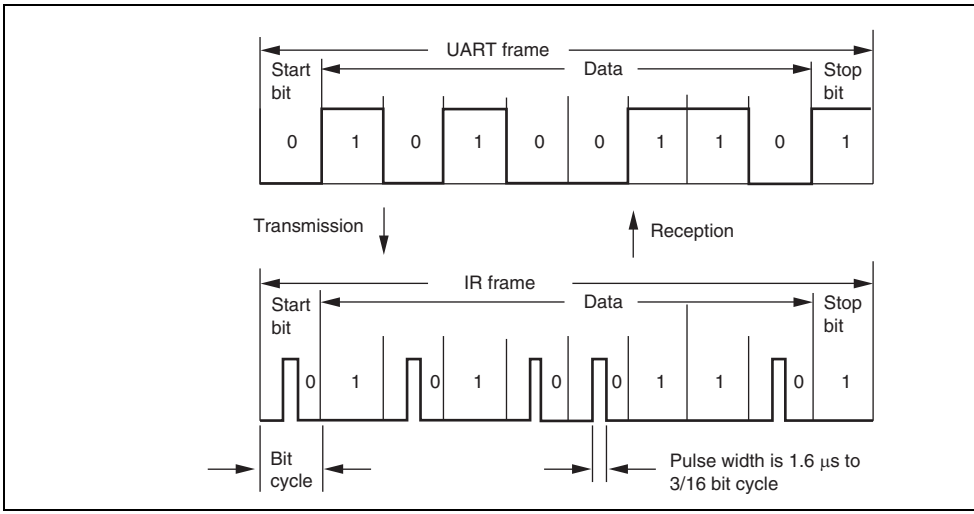


Figure 14.16 IrDA Transmission and Reception

Table 14.12 shows possible settings for bits IrCKS2 to IrCKS0 (minimum pulse width), LSI's operating frequencies and bit rates, for making the pulse width shorter than 3/16 ti rate in transmission.

Table 14.12 IrCKS2 to IrCKS0 Bit Settings

Operating	Bit Rate (bps) (Upper Row) / Bit Interval × 3/16 (µs) (Lower R			
Frequency	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

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 is set to 1, a TEI3 interrupt is requested. These two interrupts are generated during transm

The initial value of the TDRE flag in SSR is 1. Thus, when the TIE bit in SCR is set to 1 transferring the transmit data to TDR, a TXI3 interrupt request is generated even if the tra data is not ready. The initial value of the TEND flag in SSR is 1. Thus, when the TEIE bi is set to 1 before transferring the transmit data to TDR, a TEI3 interrupt request is genera if the transmit data has not been sent. It is possible to make use of the most of these interr requests efficiently by transferring the transmit data to TDR in the interrupt routine. To p the generation of these interrupt requests (TXI3 and TEI3), clear the enable bits (TIE and 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 O 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 TXI3.

	TIE	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).)	transmit data to TDR and clears to 0. Continuous transmission c performed by repeating the abo operations until the data transfe 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 tr data has not been written to TD the last bit of the transmit chara TSR is transmitted.

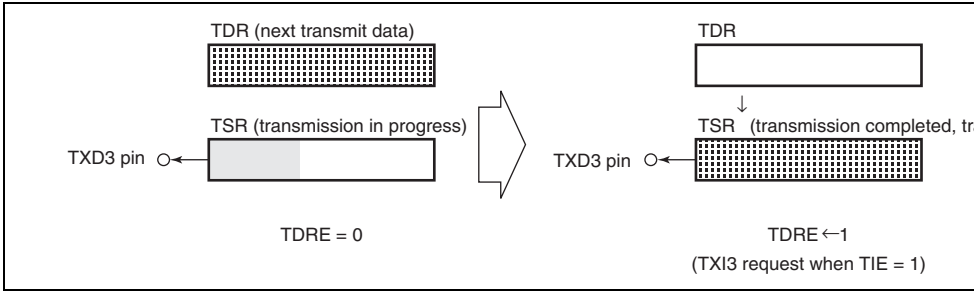


Figure 14.17 (b) TDRE Setting and TXI Interrupt

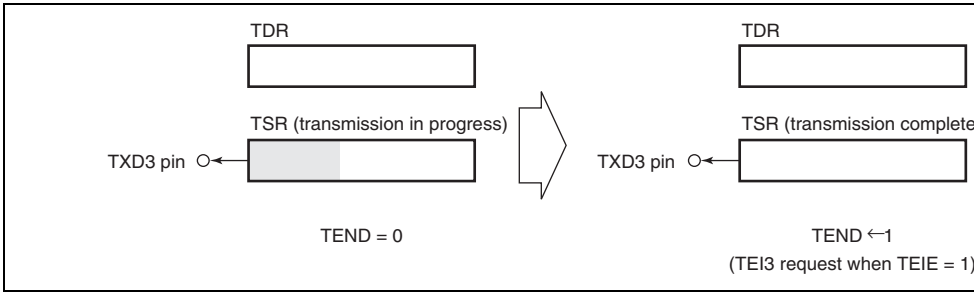


Figure 14.17 (c) TEND Setting and TEI Interrupt

When the SPC3 bit in SPCR is 0, the TXD3 pin functions as an I/O port whose direction output) and level are determined by PCR and PDR, regardless of the TE setting. This ca to set the TXD3 pin to the mark state (high level) or send a break during data transmissi maintain the communication line at the mark state until the SPC3 bit in SPCR is set to 1 PCR and PDR to 1. As the SPC3 bit in SPCR is cleared to 0 at this point, the TXD3 pin as an I/O port, and 1 is output from the TXD3 pin. To send a break during data transmis set PCR to 1 and PDR to 0, and then clear the SPC3 and TE bits to 0. When the TE bit is to 0 directly after the SPC3 bit is cleared to 0, the transmitter is initialized regardless of transmission state after the TE bit is cleared, the TXD3 pin functions as an I/O port after bit is cleared, and 0 is output from the TXD3 pin.

14.8.3 Receive Error Flags and Transmit Operations (Clock Synchronous Mo

Transmission cannot be started when a receive error flag (OER, PER, or FER) is set to 1 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 to 0.

... 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.
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 allowe
system design.

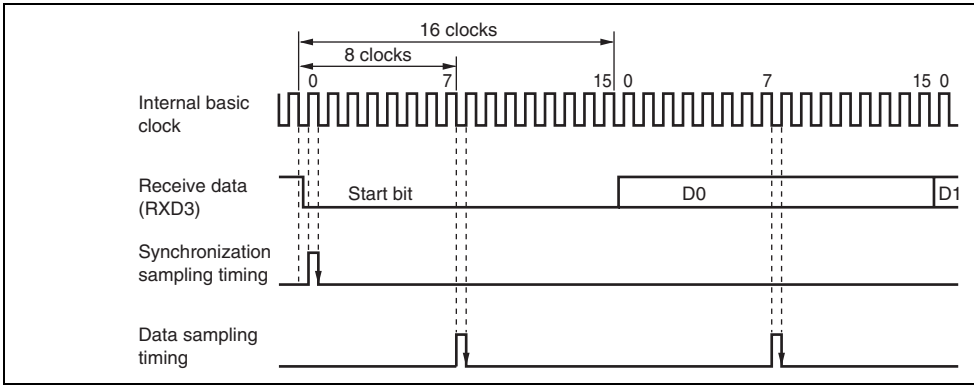


Figure 14.18 Receive Data Sampling Timing in Asynchronous Mode

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 b
as a general input/output pin. To avoid an intermediate level of voltage from being appli
SCK3 pin, the line connected to the SCK3 pin should be pulled up to the V_{CC} level via a
supplied with output from an external device.

(2) When SCK3 Pin Function is Switched from Clock Output to General Input/O

When stopping data transfer,

1. Issue one instruction to clear bits TE and RE to 0 and to set bits CKE1 and CKE0 in
   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
   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 se
transmission has not been prepared in TDR. When data is written to TDR, bit TDRE is c
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 writ
TDR while bit TDRE is cleared to 0, the data previously stored in TDR will be lost if it
been transferred to TSR. Accordingly, to ensure that serial transmission is performed de
you should first check that bit TDRE is set to 1, then write the transmit data to TDR onl
two or more times).

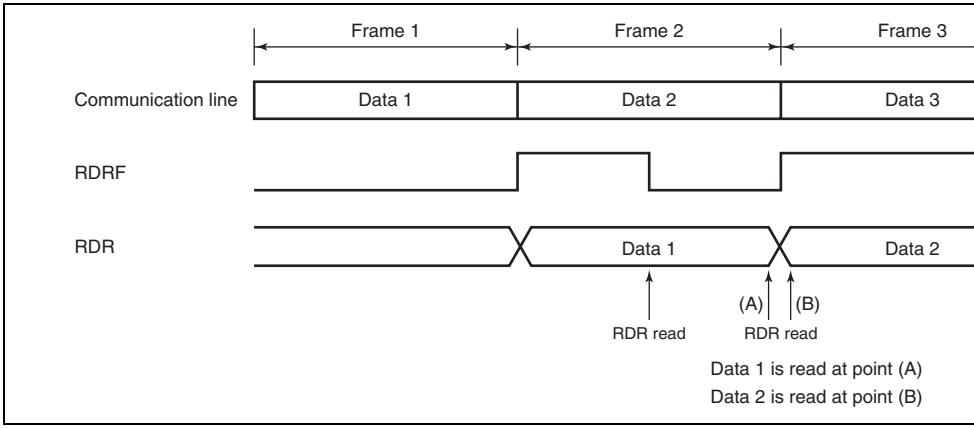


Figure 14.19 Relation between RDR Read Timing and Data

In this case, only a single RDR read operation (not two or more) should be performed aft checking that bit RDRF is set to 1. If two or more reads are performed, the data read the f 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 complet precise in terms of timing, the RDR read should be completed before bit 7 is transferred i synchronous mode, or before the STOP bit is transferred in asynchronous mode.

14.8.10 Oscillator when Serial Communication Interface 3 is Used

When serial communication interface 3 is used, the system clock oscillator or subclock o
must be used. Do not use the on-chip oscillator. For details on selecting the system clock
or on-chip oscillator, see section 4.2.4, On-Chip Oscillator Selection Method. For details
selecting the subclock oscillator or on-chip oscillator, see section 4.1.1, Oscillator Contr
Register (OSCCR).

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) a 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 conflict error
- Continuous transmission and reception of serial data are enabled since both transmit receiver have buffer structure
- Use of module standby mode enables this module to be placed in standby mode inde when not used. (The SSU is halted as the initial value. For details, refer to section 5.4 Standby Function.)

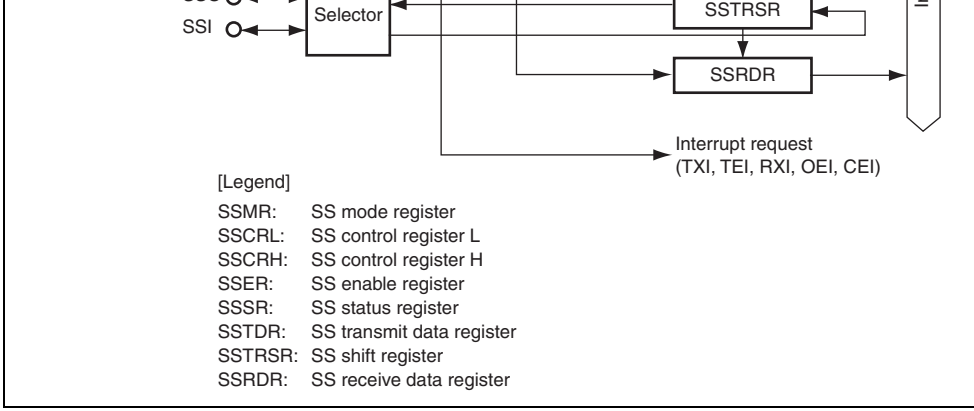


Figure 15.1 Block Diagram of SSU

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

- 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, s
open-drain output of the serial data output pin, selects an output value of the serial data
selects the SSCK pin, and selects the $\overline{SCS}$ 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 mast or a slave device. When this module is used as device, transfer clock is output from the SSCK the CE bit in SSSR is set, this bit is automatica 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 th pin are both used or only one pin is used. For refer to section 15.4.3, Relationship between D Input/Output and Shift Register. When the SSU SSCRL is 0, this setting is invalid. 0: Normal mode. Communication is performed two pins. 1: Bidirectional mode. Communication is perfo using only one pin.

				Although the value in the last bit of transmit data retained in the serial data output after the end of transmission, the output level of serial data can changed by manipulating this bit before or after transmission. When the output level is changed, SOLP bit should be cleared to 0 and the MOV in should be used. If this bit is written during data t erroneous operation may occur. Therefore this b not be manipulated during transmission. 0: Shows serial data output level to low in readin Changes serial data output level to low in writ 1: Shows serial data output level to high in readi Changes serial data output level to high in wr
3	SOLP	1	R/W	SOL Write Protect When output level of serial data is changed, the instruction is used to set the SOL bit to 1 and cle bit to 0 or to clear the SOL bit and this bit to 0. 0: In writing, output level can be changed accord the value of the SOL bit. 1: In reading, this bit is always read as 1. In writi output level cannot be changed. (See section Usage Note.)

				0, the SCS pin functions as a port regardless o 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, func 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 SS SCS pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.

				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 seque forcibly reset. Then this bit is automatically clear 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 CMO 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 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.

				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 the internal clock is selected.
0	CKS0	0	R/W	The system clock (ϕ) is halted in subactive mod subsleep mode. Select ϕ_{SUB}/2 in these modes. 000: ϕ/256 001: ϕ/128 010: ϕ/64 011: ϕ/32 100: ϕ/16 101: ϕ/8 110: ϕ/4 111: ϕ_{SUB}/2

5	RSSTP	0	R/W	Receive single stop When this bit is 1, receive operation is complete 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 enabled.
2	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, a TXI interrupt request enabled.
1	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, an RXI and an OEI inte requests are enabled.
0	CEIE	0	R/W	Conflict Error Interrupt Enable When this bit is set to 1, a CEI interrupt request enabled.

				reception because an overrun error has occurr SSRDR retains received data before the overru occurs and the received data after the overrun occurs is lost. When this bit is set to 1, subseq reception cannot be continued. When the MSS SSCRH is 1, this is also applied to serial transr [Setting condition] • When the next serial reception is complete 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 is 1 [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 receive data is transferred from SSTRSR to [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 = 1 and MSS =1, the $\overline{\text{SCS}}$ pin input is low • When the $\overline{\text{SCS}}$ pin level changes from low to during transfer while SSUMS = 1 and MSS = [Clearing condition] • When 0 is written to this bit after reading 1

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

SSTDR is an 8-bit register that stores serial data for transmission. SSTDR can be read o
to by the CPU at all times. When the SSU detects that SSTRSR is empty, it transfers the
data written in SSTDR to SSTRSR and starts serial transmission. If the next transmit da
already been written to SSTDR during serial transmission, continuous serial transmissio
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
from SSTDR to SSTRSR, bit 0 in SSTDR is transferred to bit 0 in SSTRSR while the M
SSMR is 0 (LSB-first transfer) and bit 7 in SSTDR is transferred to bit 0 in SSTRSR wh
MLS bit in SSMR is 1 (MSB-first transfer). SSTRSR cannot be directly accessed by the

15.4 Operation

15.4.1 Transfer Clock

Transfer clock can be selected from eight internal clocks and an external clock. When th
is used, the SSCK pin must be selected as a serial clock by setting the SCKS bit in SSCR
When the MSS bit in SSCRH is 1, an internal clock is selected and the SSCK pin is in th
state. If transfer is started, the SSCK pin outputs clocks of the transfer rate set in the CK
CKS0 bits in SSMR. When the MSS bit is 0, an external clock is selected and the SSCK
the input state.

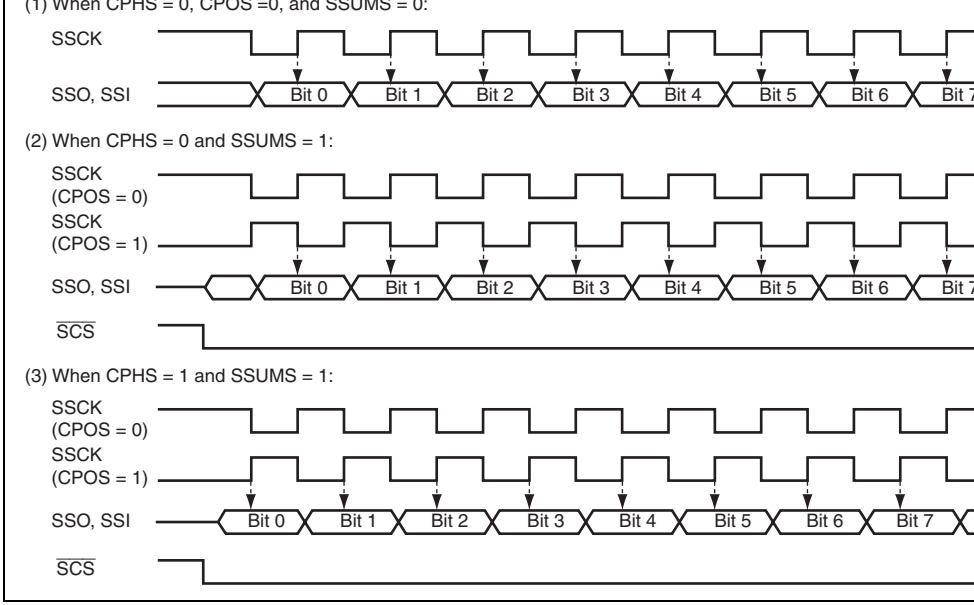


Figure 15.2 Relationship between Clock Polarity and Phase, and Data

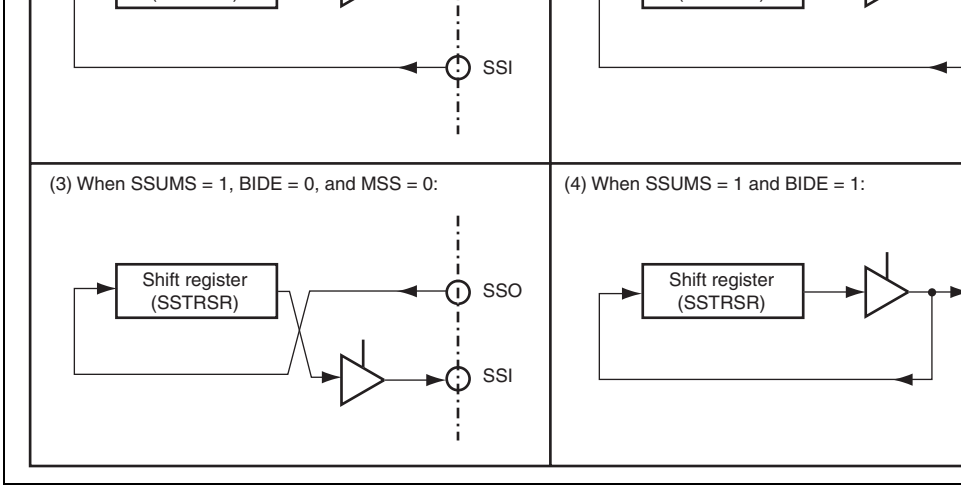


Figure 15.3 Relationship between Data Input/Output Pin and Shift Regist

Clocked Synchronous Communication Mode	0	x	0	0	1	In	—
				1	0	—	Out
					1	In	Out
			1	0	1	In	—
				1	0	—	Out
					1	In	Out
Four-Line Bus Communication Mode	1	0	0	0	1	—	In
				1	0	Out	—
					1	Out	In
			1	0	1	In	—
				1	0	—	Out
					1	In	Out
Four-Line Bus (Bidirectional) Communication Mode	1	1	0	0	1	—	In
				1	0	—	Out
			1	0	1	—	In
				1	0	—	Out

[Legend] x: Don't care.

—: Can be used as a general I/O port.

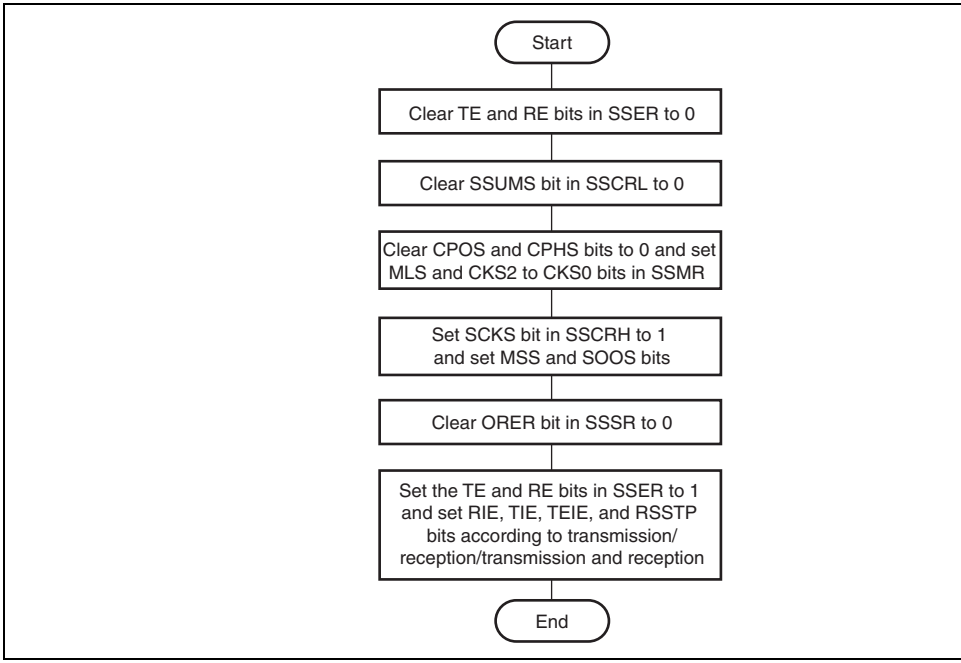


Figure 15.4 Initialization in Clocked Synchronous Communication Mod

When the TDRE flag is 0 and one frame of data has transferred, data is transferred from S
SSTRSR and serial transmission of the next frame is started. If the eighth bit is transmitte
the TDRE flag is 1, the TEND bit in SSSR is set to 1 and the state is retained. If the TEIE
SSER is set to 1 at this time, a TEI is generated. After transmission is ended, the SSCK p
fixed high.

While the ORER bit in SSSR is set to 1, transmission cannot be performed. Therefore cor
the ORER bit is cleared to 0 before transmission.

Figure 15.6 shows a sample flowchart for serial data transmission.

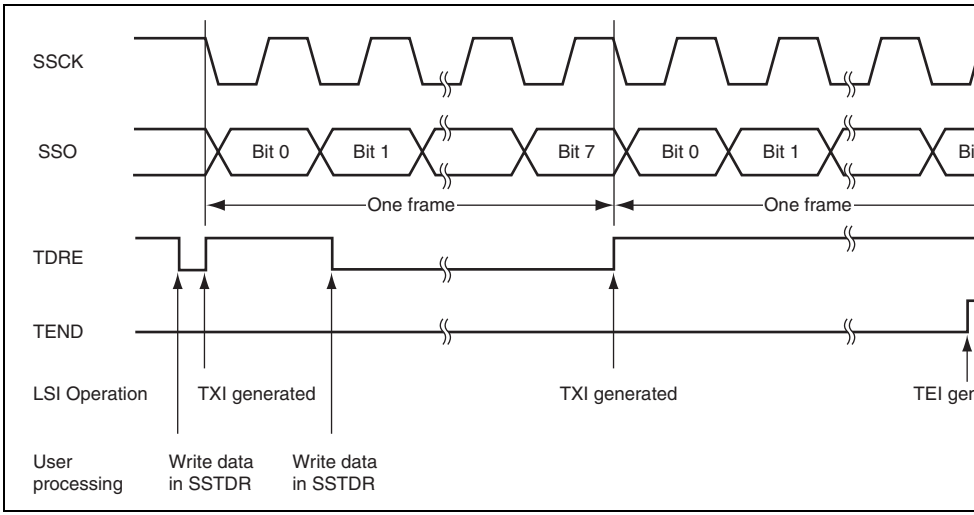


Figure 15.5 Example of Operation in Data Transmission

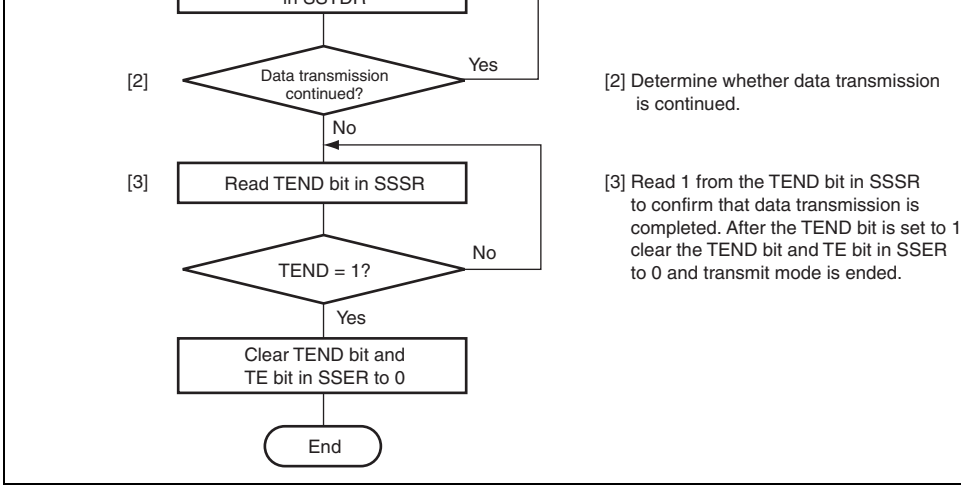


Figure 15.6 Sample Serial Transmission Flowchart

When the SSU is set as a master device and reception is ended, received data is read after the RSSTP bit in SSER to 1. Then the SSU outputs eight bits of clocks and operation is st After that, the RE and RSSTP bits are cleared to 0 and the last received data is read. Note 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 a overrun error (OEI) is generated and operation is stopped. When the ORER bit in SSSR i reception cannot be performed. Therefore confirm that the ORER bit is cleared to 0 befor reception.

Figure 15.8 shows a sample flowchart for serial data reception.

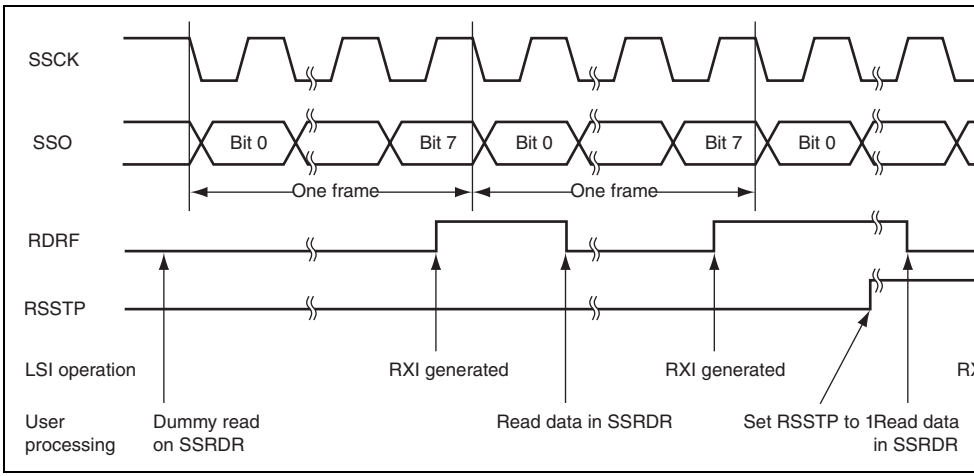
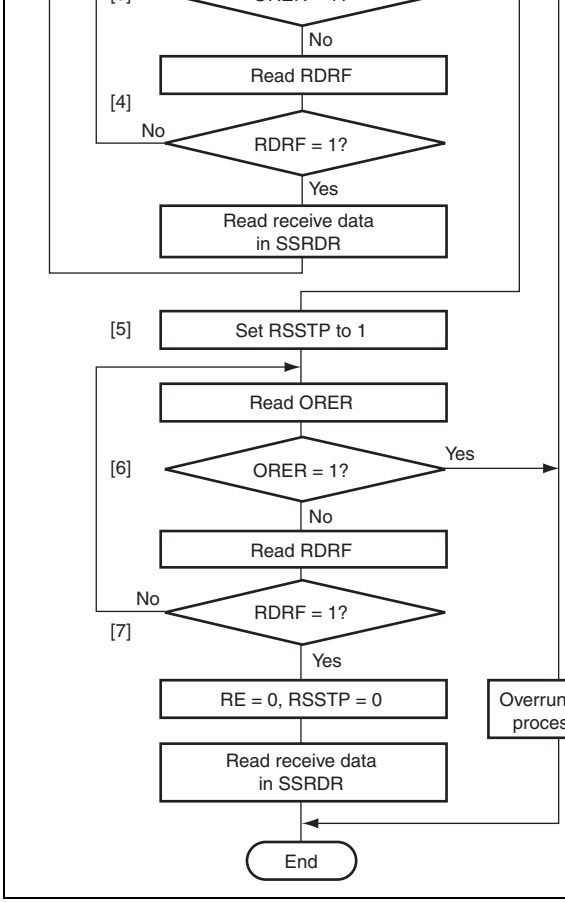


Figure 15.7 Example of Operation in Data Reception (MSS = 1)



[3][6] When a receive error occurs, clear t
ORER flag to 0 after the ORER flag
SSSR is read and an appropriate er
processing is performed. When the
flag is set to 1, transmission/receptio
cannot be started again.

[4] Confirm that the RDRF bit is 1. If the R
bit is 1, receive data in SSRDR is read
SSRDR bit is read, the RDRF bit is aut
cleared.

[5] Before the last one byte of data is rece
set the RSSTP bit to 1 and reception is
after the data is received.

[7] Confirm that the RDRF bit is 1. To end
reception, clear the RE and RSSTP bit
0 and then read the last receive data. I
SSRDR bit is read before clearing the
reception is started again.

Figure 15.8 Sample Serial Reception Flowchart (MSS = 1)

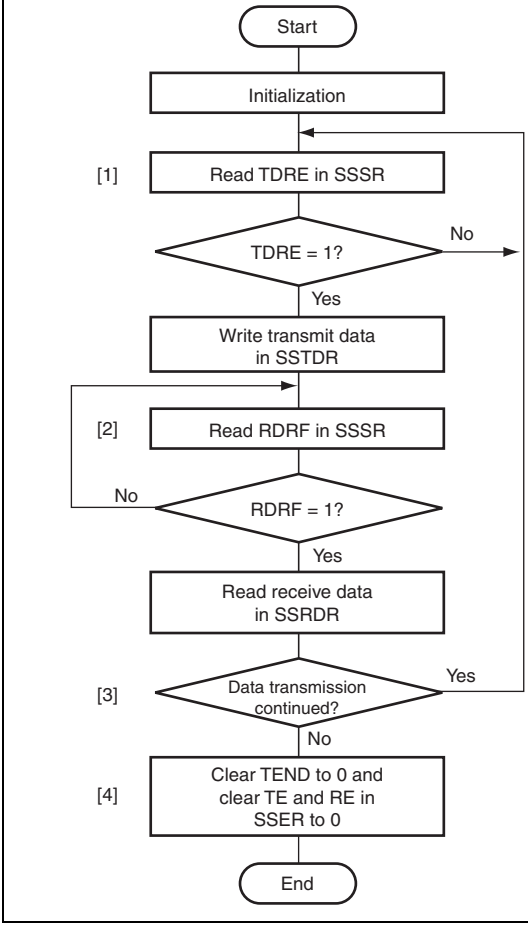


Figure 15.9 Sample Flowchart for Serial Transmit and Receive Operations

When the SSU is set as a master device, the chip select line controls output. When the S
as a slave device, the chip select line controls input. When the SSU is set as a master de
chip select line controls output of the $\overline{\text{SCS}}$ pin or controls output of a general port by set
CSS1 bit in SSCRH to 1. When the SSU is set as a slave device, the chip select line sets
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 p
in MSB-first order.

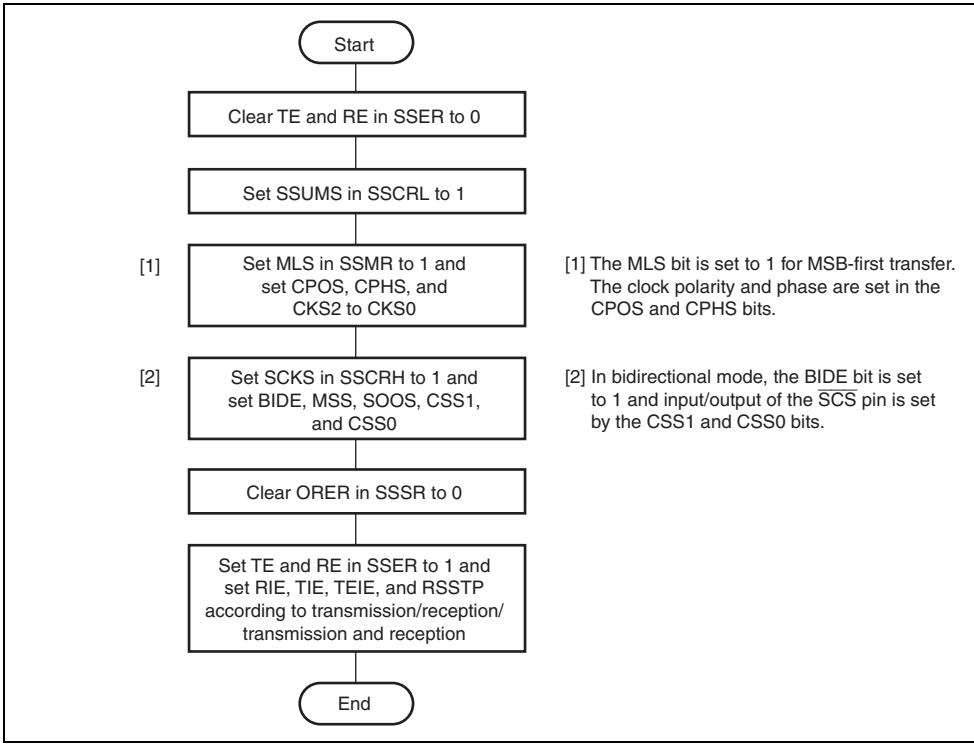


Figure 15.10 Initialization in Four-Line Bus Communication Mode

TDRE flag to 1 and starts transmission. If the TIE bit in SSER is set to 1 at this time, a T
generated.

When the TDRE flag is 0 and one frame of data has transferred, data is transferred from
SSTRSR and serial transmission of the next frame is started. If the eighth bit is transmit
the TDRE flag is 1, the TEND bit in SSSR is set to 1 and the state is retained. If the TEI
SSER is set to 1 at this time, a TEI is generated. After transmission is ended, the SSCK
fixed high and the $\overline{\text{SCS}}$ pin goes high. When continuous transmission is performed with
pin low, the next data should be written to SSTDR before transmitting the eighth bit of t

While the ORER bit in SSSR is set to 1, transmission cannot be performed. Therefore co
the ORER bit is cleared to 0 before transmission.

The difference between this mode and clocked synchronous communication mode is as
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
Z state and when the SSU is set as a slave device, the SSI pin is in the Hi-Z state if the $\overline{\text{S}}$
in the high-input state. The sample flowchart for serial data transmission is the same as t
clocked synchronous communication mode.

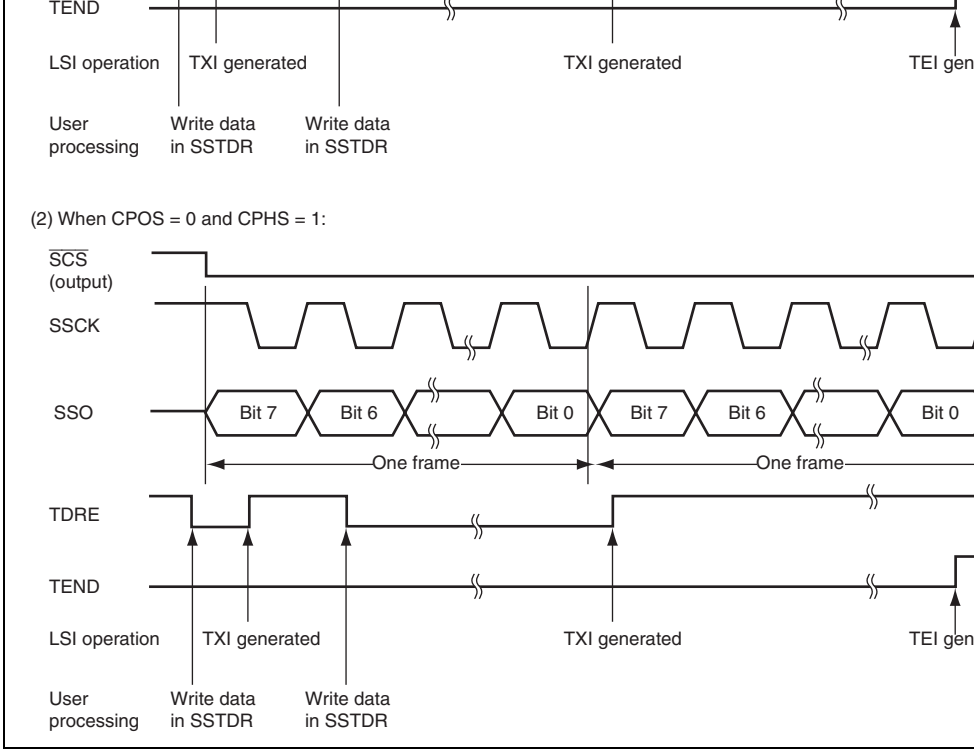


Figure 15.11 Example of Operation in Data Transmission (MSS = 1)

SSRDR. If the RIE bit in SSER is set to 1 at this time, an RXI is generated. If SSRDR is
RDRF bit is automatically cleared to 0.

When the SSU is set as a master device and reception is ended, received data is read afte
the RSSTP bit in SSER to 1. Then the SSU outputs eight bits of clocks and operation is
After that, the RE and RSSTP bits are cleared to 0 and the last received data is read. Not
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
overrun error (OEI) is generated and operation is stopped. When the ORER bit in SSSR
reception cannot be performed. Therefore confirm that the ORER bit is cleared to 0 befo
reception.

The set timings of the RDRF and ORER flags differ according to the CPHS setting. The
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 synchronou
communication mode.

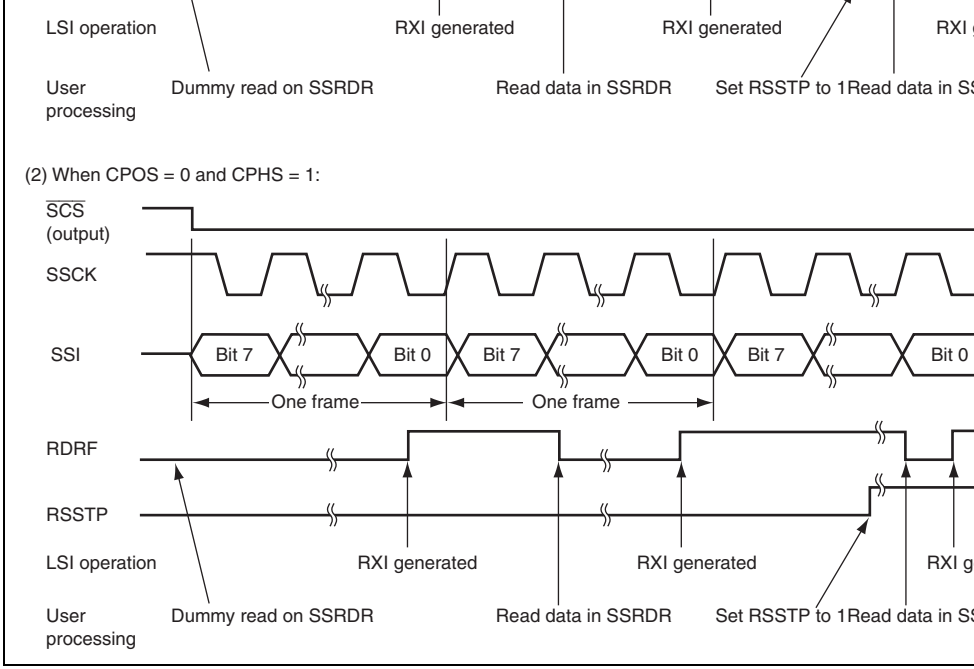


Figure 15.12 Example of Operation in Data Reception (MSS = 1)

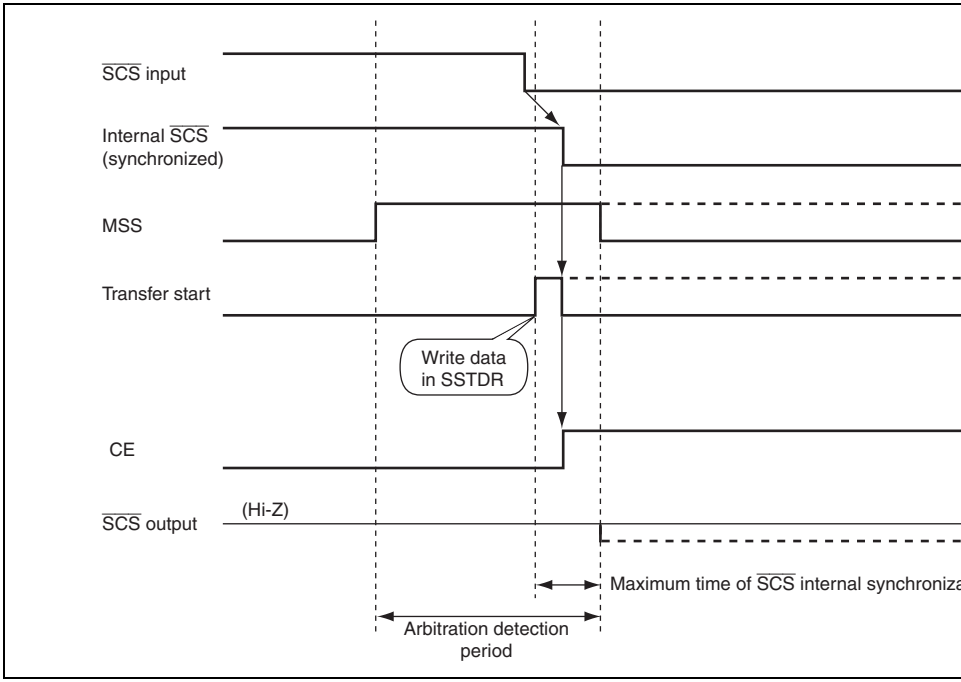


Figure 15.13 Arbitration Check Timing

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 e
the interrupt exception handling. Each interrupt source must be cleared during the excepti
handling. Note that the TDRE and TEND bits are automatically cleared by writing transm
SSTDR and the RDRF bit is automatically cleared by reading SSRDR. When transmit da
written in SSTDR, the TDRE bit is set again at the same time. Then if the TDRE bit is cl
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 (t
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
the SOL bit to itself. With this procedure, the write protect can be performed on the SOL

- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independe each other, the continuous transmission/reception can be performed.
- Use of module standby mode enables this module to be placed in standby mode inde when not used. (The IIC2 is halted as the initial value. For details, refer to section 5.4 Standby Function.)

I^2C 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 ar completed.
- Six interrupt sources

  Transmit data empty (including slave-address match), transmit end, receive data full 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 port/serial function is selected) and NMOS outputs when the bus drive function is se

Clock synchronous format

- Four interrupt sources

  Transmit-data-empty, transmit-end, receive-data-full, and overrun error

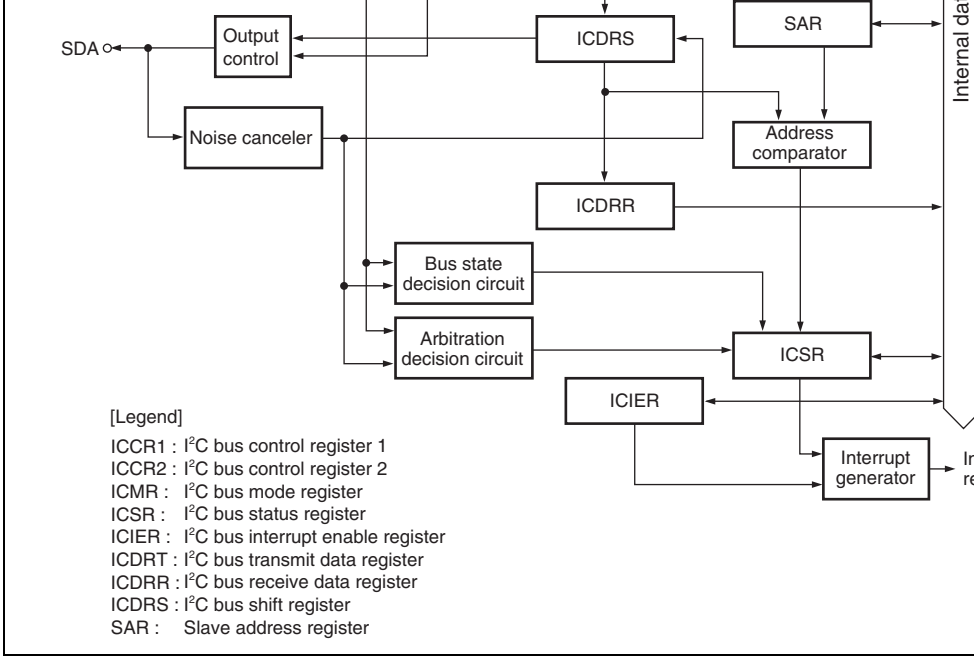


Figure 16.1 Block Diagram of I²C Bus Interface 2

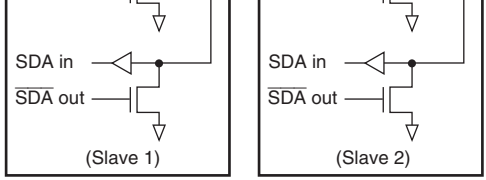


Figure 16.2 External Circuit Connections of I/O Pins

16.2 Input/Output Pins

Table 16.1 shows the pin configuration of the I^2C 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

- 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
master or slave mode, transmission or reception, and transfer clock frequency in master m

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 a the port/serial function.) 1: This bit is enabled for transfer operations. (S SDA pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable This bit enables or disables the next operation TRS is 0 and ICDRR is read. 0: Enables next reception 1: Disables next reception

				agree with the slave address that is set to SA eighth bit is 1, TRS is automatically set to 1. overrun error occurs in master mode with the synchronous serial format, MST is cleared to slave receive mode is entered. Operating modes are described below accord MST and TRS combination. When clock sync serial format is selected and MST is 1, clock i 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
2	CKS2	0	R/W	These bits are valid only in master mode and set according to the necessary transfer rate (table 16.2). These bit are used to specify the time in slave transmit mode. The data setup t secured for 10tcyc when CKS3 = 0 and for 20 CKS3 = 1.
1	CKS1	0	R/W	
0	CKS0	0	R/W	

		1	0	φ/112	17.9 kHz	44.6 kHz	89.3 k
			1	φ/128	15.6 kHz	39.1 kHz	78.1 k
1	0	0	0	φ/56	35.7 kHz	89.3 kHz	179 k
			1	φ/80	25.0 kHz	62.5 kHz	125 k
		1	0	φ/96	20.8 kHz	52.1 kHz	104 k
			1	φ/128	15.6 kHz	39.1 kHz	78.1 k
	1	0	0	φ/160	12.5 kHz	31.3 kHz	62.5 k
			1	φ/200	10.0 kHz	25.0 kHz	50.0 k
		1	0	φ/224	8.9 kHz	22.3 kHz	44.6 k
			1	φ/256	7.8 kHz	19.5 kHz	39.1 k

				format, this bit has no meaning. With the I²C b this bit is set to 1 when the SDA level change high to low under the condition of SCL = high that the start condition has been issued. This cleared to 0 when the SDA level changes fro high under the condition of SCL = high, assu the stop condition has been issued. Write 1 to and 0 to SCP to issue a start condition. Follo procedure when also re-transmitting a start co Write 0 in BBSY and 0 in SCP to issue a stop To issue start/stop conditions, use the MOV i
6	SCP	1	R/W	Start/Stop Issue Condition Disable The SCP bit controls the issue of start/stop co master mode. To issue a start condition, write 1 in BBSY an SCP. A repeated start condition is issued in th way. To issue a stop condition, write 0 in BBS SCP. This bit is always read as 1. If 1 is writte data is not stored.
5	SDAO	1	R/W	SDA Output Value Control This bit is used with SDAOP when modifying level of SDA. This bit should not be manipulat transfer. 0: When reading, SDA pin outputs low. When writing, SDA pin is changed to outpu 1: When reading, SDA pin outputs high. When writing, SDA pin is changed to outpu (outputs high by external pull-up resistanc

2	—	1	—	Reserved This bit is always read as 1, and cannot be mo
1	IICRST	0	R/W	IIC Control Part Reset This bit resets the control part except for I^2C re this bit is set to 1 when hang-up occurs becaus communication failure during I^2C operation, I^2C part can be reset without setting ports and initia registers.
0	—	1	—	Reserved This bit is always read as 1, and cannot be mo

				Set this bit to 0 when the I²C bus format is use
6	WAIT	0	R/W	Wait Insertion Bit In master mode with the I²C bus format, this b whether to insert a wait after data transfer exc acknowledge bit. When WAIT is set to 1, after the clock for the final data bit, low period is ex two transfer clocks. If WAIT is cleared to 0, da acknowledge bits are transferred consecutive wait inserted. The setting of this bit is invalid in slave mode I²C bus format or with the clock synchronous format.
5, 4	—	All 1	—	Reserved These bits are always read as 1, and cannot modified.
3	BCWP	1	R/W	BC Write Protect This bit controls the BC2 to BC0 modifications modifying BC2 to BC0, this bit should be clea and use the MOV instruction. In clock synchro serial mode, BC should not be modified. 0: When writing, values of BC2 to BC0 are se 1: When reading, 1 is always read. When writing, settings of BC2 to BC0 are i

the clock synchronous serial format, these bits
not be modified.

I^2C Bus Format	Clock Synchronous Seria
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

				disabled. 1: Transmit data empty interrupt request (TXI enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable This bit enables or disables the transmit end i (TEI) at the rising of the ninth clock while the in ICSR is 1. TEI can be canceled by clearing bit or the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disa 1: Transmit end interrupt request (TEI) is ena
5	RIE	0	R/W	Receive Interrupt Enable This bit enables or disables the receive data f interrupt request (RXI) and the overrun error i request (ERI) with the clock synchronous forn a receive data is transferred from ICDRS to IC the RDRF bit in ICSR is set to 1. RXI can be c by clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) an error interrupt request (ERI) with the clock synchronous format are disabled. 1: Receive data full interrupt request (RXI) an error interrupt request (ERI) with the clock synchronous format are enabled.

3	STIE	0	R/W	Stop Condition Detection Interrupt Enable 0: Stop condition detection interrupt request (S disabled. 1: Stop condition detection interrupt request (S enabled.
2	ACKE	0	R/W	Acknowledge Bit Judgment Select 0: The value of the receive acknowledge bit is and continuous transfer is performed. 1: If the receive acknowledge bit is 1, continuo transfer is halted.
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowled that are returned by the receive device. This bi 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 the acknowledge timing. 0: 0 is sent at the acknowledge timing. 1: 1 is sent at the acknowledge timing.

				• When TRS is set • When a start condition (including re-transf been issued • When transmit mode is entered from rece in slave mode [Clearing conditions] • When 0 is written in TDRE after reading T • When data is written to ICDRT with an ins
6	TEND	0	R/(W)*	Transmit End [Setting conditions] • When the ninth clock of SCL rises with the format while the TDRE flag is 1 • When the final bit of transmit frame is sen clock synchronous serial format [Clearing conditions] • When 0 is written in TEND after reading T • When data is written to ICDRT with an ins
5	RDRF	0	R/(W)*	Receive Data Register Full [Setting condition] • When a receive data is transferred from IC ICDRR [Clearing conditions] • When 0 is written in RDRF after reading R • When ICDRR is read with an instruction

3	STOP	0	R/(W)*	Stop Condition Detection Flag [Setting conditions] • In master mode, when a stop condition is d after the completion of frame transfer • In slave mode, when a stop condition is det after the slave address of the first byte, follo general call and the detection of the start c matches the address set in SAR [Clearing condition] • When 0 is written in STOP after reading ST

				[Setting conditions] • If the internal SDA and SDA pin disagree of SCL in master transmit mode • When the SDA pin outputs high in master while a start condition is detected • When the final bit is received with the cloc synchronous format while RDRF = 1 [Clearing condition] • When 0 is written in AL/OVE after reading = 1
1	AAS	0	R/(W)*	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if th frame following a start condition matches bits SVA0 in SAR. [Setting conditions] • When the slave address is detected in sla mode • When the general call address is detected receive mode. [Clearing condition] • When 0 is written in AAS after reading AA

16.3.6 Slave Address Register (SAR)

SAR selects the communication format and sets the slave address. When the chip is in sla
with the I²C bus format, if the upper 7 bits of SAR match the upper 7 bits of the first fram
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 t differing form the addresses of other slave dev 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 d
space in the shift register (ICDRS), it transfers the transmit data which is written in ICDR
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 se
and when the data is written to ICDRT, the MSB/LSB inverted data is read. The initial va
ICDRT is H'FF. The initial value of ICDRT is H'FF.

ICDRS to ICDRR after data of one byte is received. This register cannot be read directly
CPU.

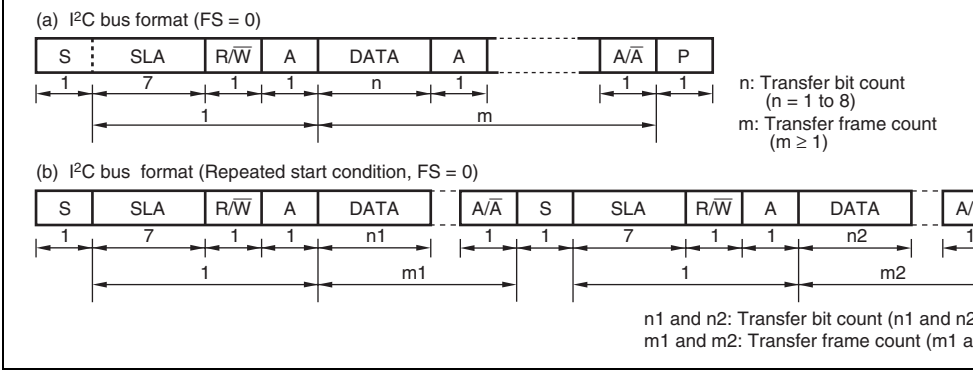


Figure 16.3 I²C Bus Formats

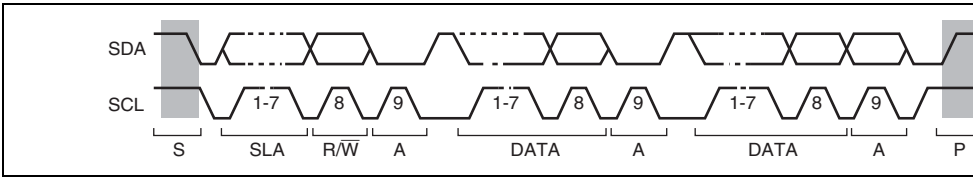


Figure 16.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is hig

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 hig

ICCR1 to select master transmit mode. Then, write 1 to BBSY and 0 to SCP using M
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 b
show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cle
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 i
at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confir
slave device has been selected. Then, write second byte data to ICDRT. When ACK
the slave device has not been acknowledged, so issue the stop condition. To issue the
condition, write 0 to BBSY and SCP using MOV instruction. SCL is fixed low until
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 e
byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1)
receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TE
NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mo

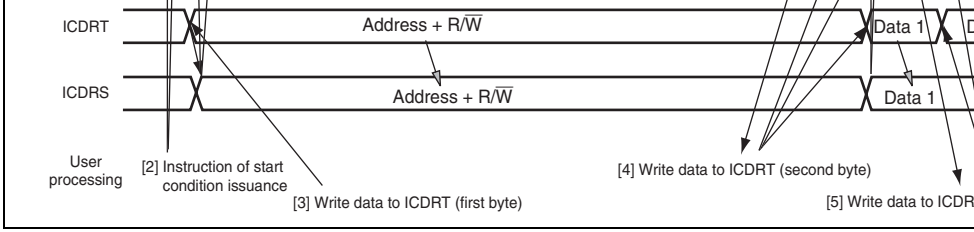


Figure 16.5 Master Transmit Mode Operation Timing (1)

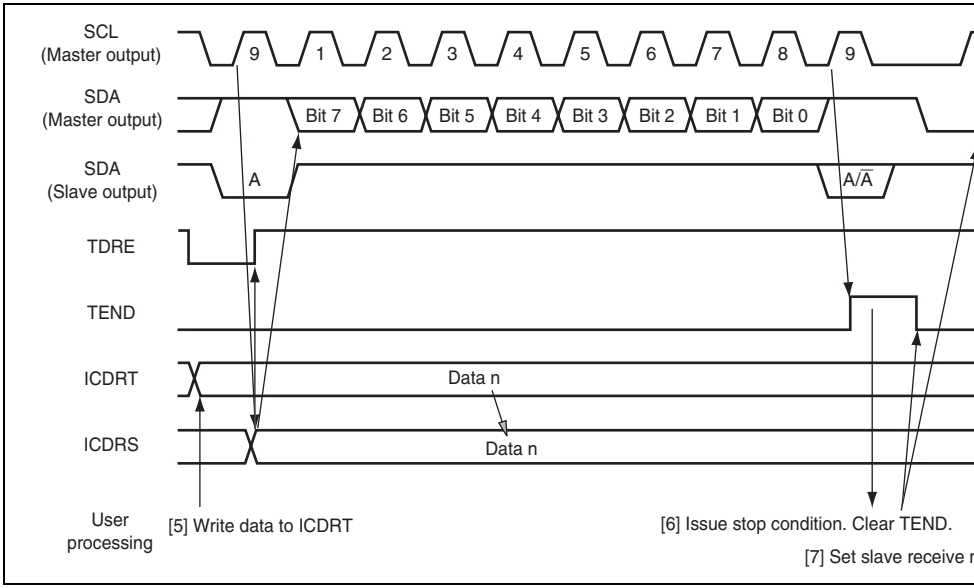


Figure 16.6 Master Transmit Mode Operation Timing (2)

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 of 9th receive clock pulse. At this time, the receive data is read by reading ICDRR, a is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. I receive clock pulse falls after reading ICDRR by the other processing while RDRF i 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 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 c
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.

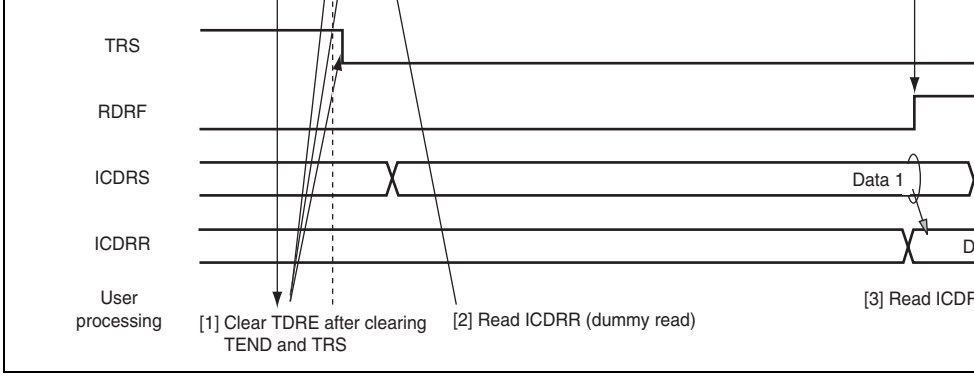


Figure 16.7 Master Receive Mode Operation Timing (1)

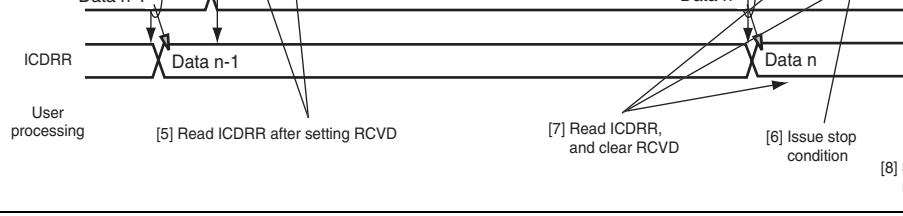


Figure 16.8 Master Receive Mode Operation Timing (2)

16.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master devi the receive clock and returns an acknowledge signal. For slave transmit mode operation 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 t bits in ICCR1 to 1. (Initial setting) Set the MST and TRS bits in ICCR1 to select sla mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start co the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise clock pulse. At this time, if the 8th bit data (R/$\overline{W}$) is 1, the TRS and ICSR bits in IC set to 1, and the mode changes to slave transmit mode automatically. The continuou transmission is performed by writing transmit data to ICDRT every time TDRE is se
3. If TDRE is set after writing last transmit data to ICDRT, wait until TEND in ICSR is 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.

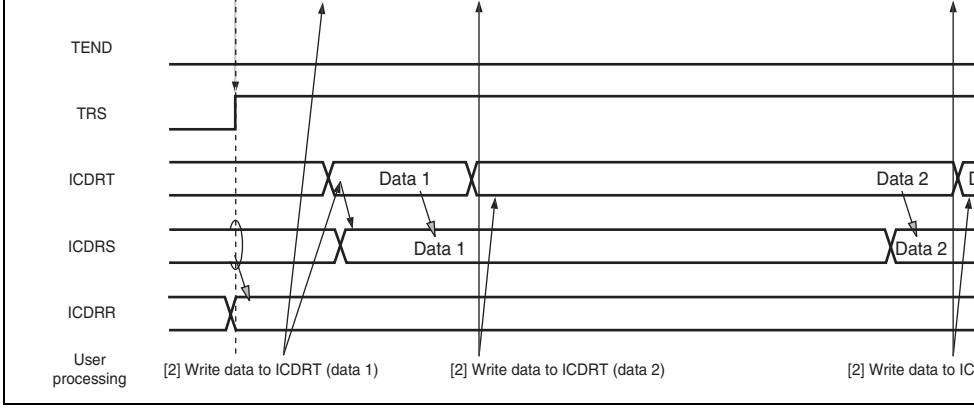


Figure 16.9 Slave Transmit Mode Operation Timing (1)

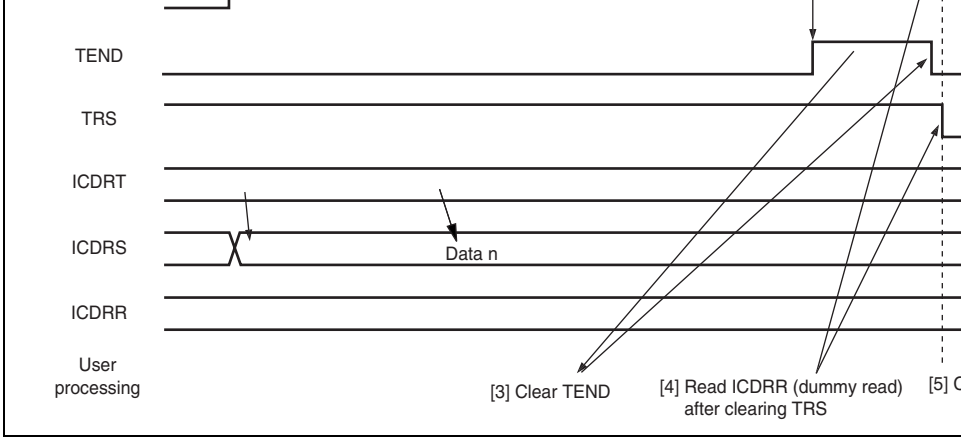


Figure 16.10 Slave Transmit Mode Operation Timing (2)

2. When the slave address matches in the first frame following detection of the start cond the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise o clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (S 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 fixed low until ICDRR is read. The change of the acknowledge before reading ICDRR returned to the master device, is reflected to the next transmit frame.
4. The last byte data is read by reading ICDRR.

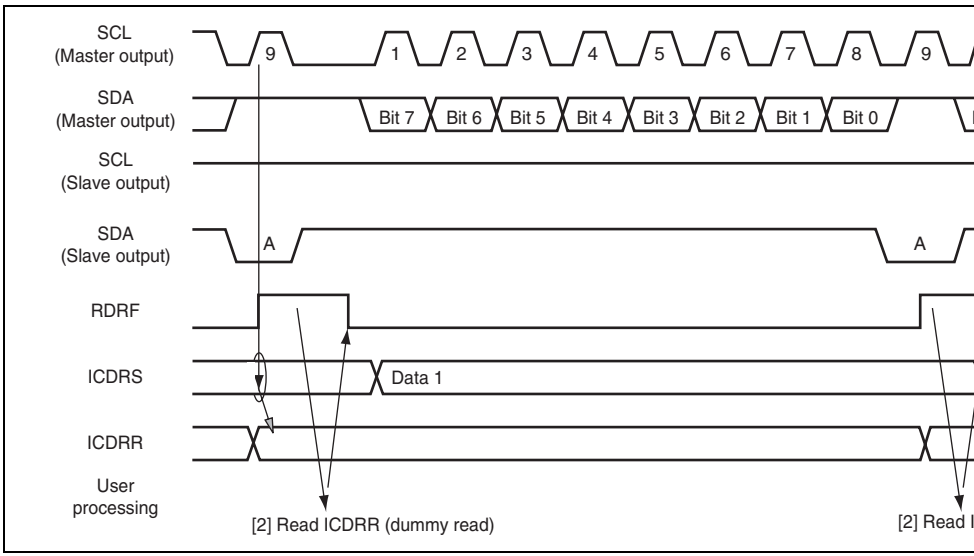


Figure 16.11 Slave Receive Mode Operation Timing (1)

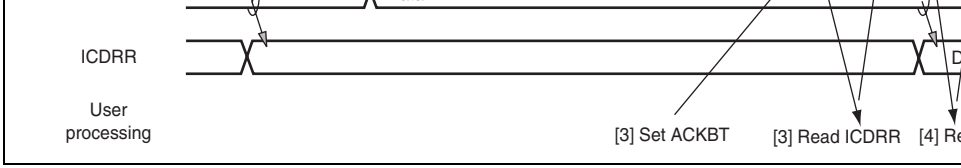


Figure 16.12 Slave Receive Mode Operation Timing (2)

16.4.6 Clock Synchronous Serial Format

This module can be operated with the clock synchronous serial format, by setting the FS SAR to 1. When the MST bit in ICCR1 is 1, the transfer clock output from SCL is selec 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 r of the SCL clock is guaranteed. The MLS bit in ICMR sets the order of data transfer, in MSB first or LSB first. The output level of SDA can be changed during the transfer wai SDAO bit in ICCR2.

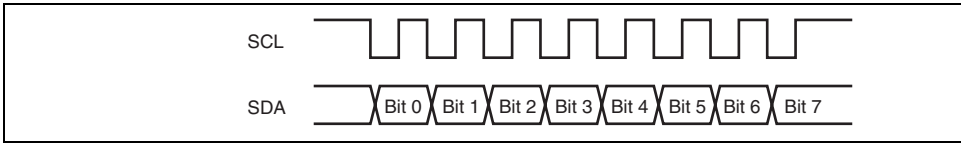


Figure 16.13 Clock Synchronous Serial Transfer Format

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 c from transmit mode to receive mode, clear TRS while TDRE is 1.

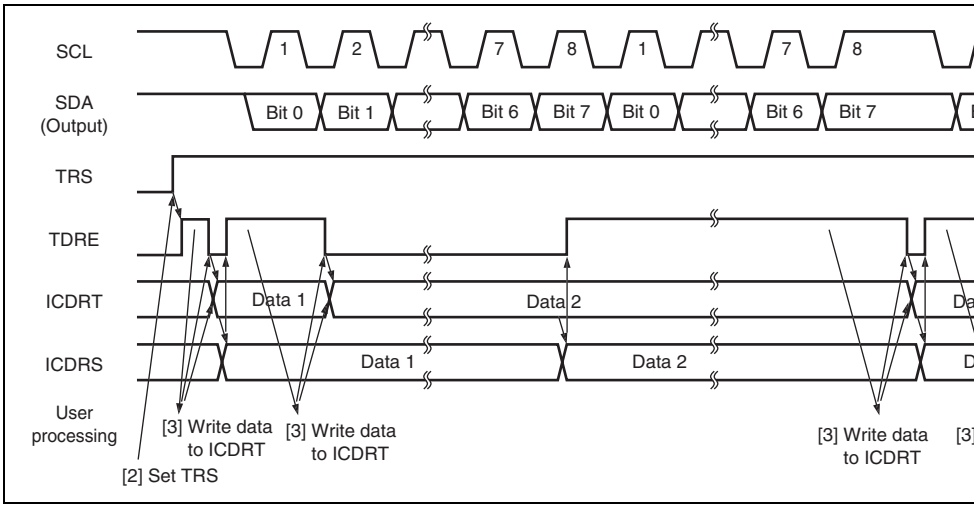


Figure 16.14 Transmit Mode Operation Timing

continually output. The continuous reception is performed by reading ICDRR every
RDRF is set. When the 8th clock is risen while RDRF is 1, the overrun is detected a
AL/OVE in ICSR is set. At this time, the previous reception data is retained in ICDR

4. To stop receiving when MST = 1, set RCVD in ICCR1 to 1, then read ICDRR. Then
fixed high after receiving the next byte data.

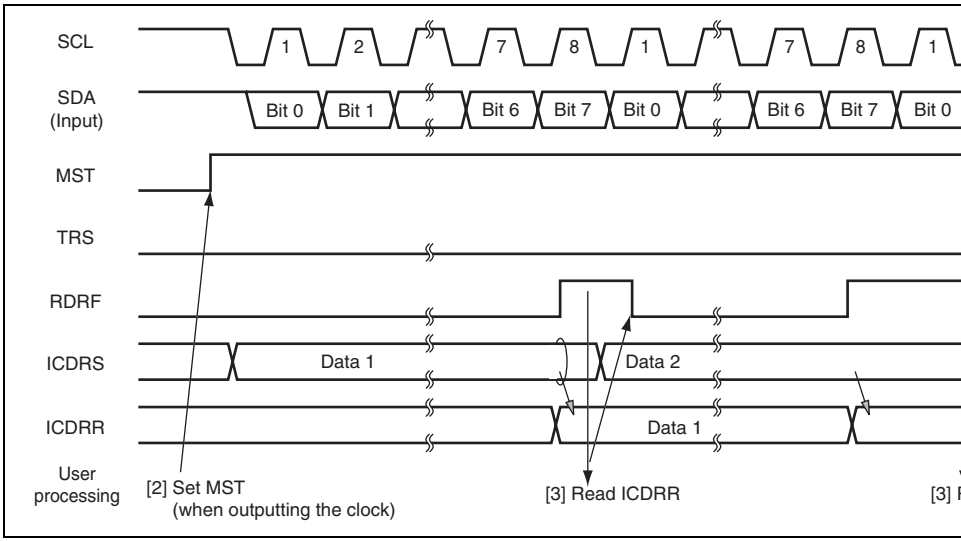


Figure 16.15 Receive Mode Operation Timing

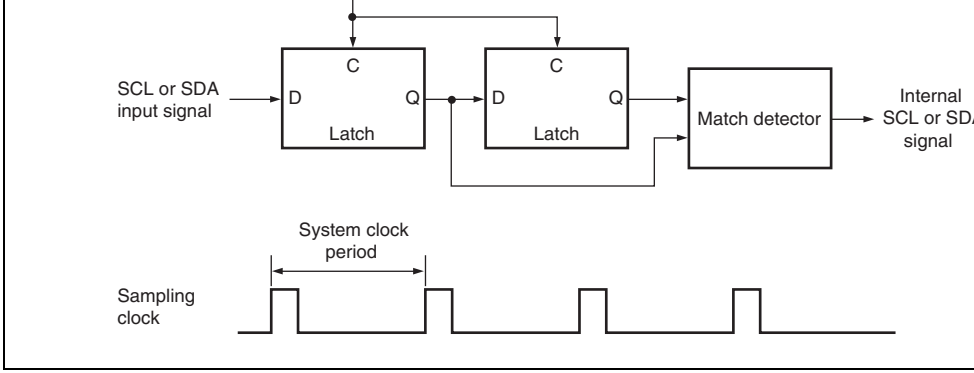


Figure 16.16 Block Diagram of Noise Conceler

16.4.8 Example of Use

Flowcharts in respective modes that use the I^2C bus interface 2 are shown in figures 16.17 16.20.

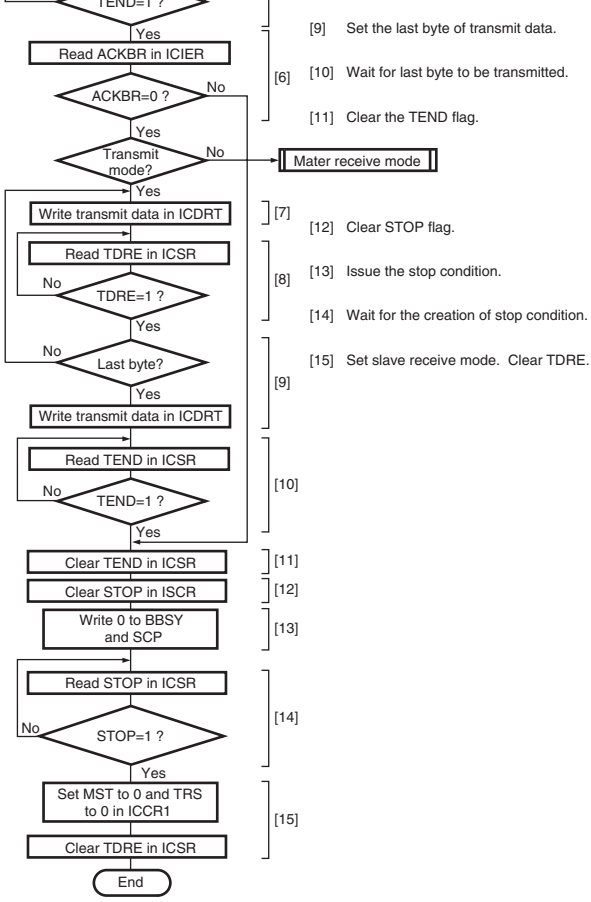
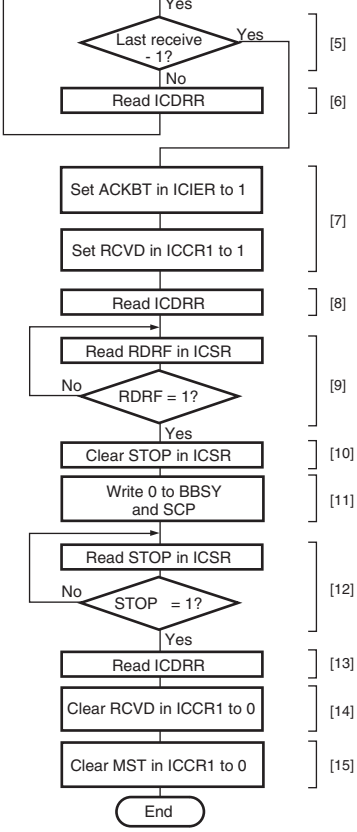


Figure 16.17 Sample Flowchart for Master Transmit Mode



[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.

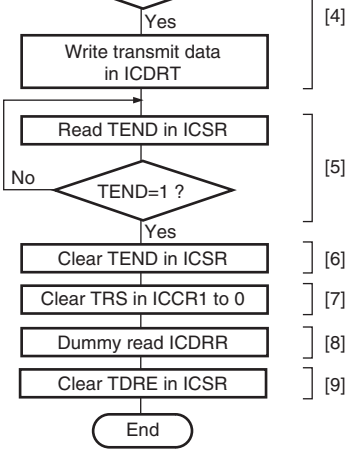
[14] Clear RCVD.

[15] Set slave receive mode.

Note: * Do not activate an interrupt during the execution of steps [1] to [3

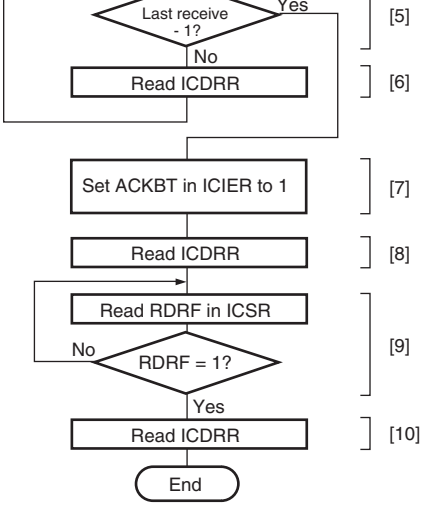
For single-byte reception, skip steps [2] to [6] after step [1]. Ther step [7]. Dummy-read in step [8].

Figure 16.18 Sample Flowchart for Master Receive Mode



[9] Clear the TDRE flag.

Figure 16.19 Sample Flowchart for Slave Transmit Mode



[8] Read the (last byte - 1) of receive data.

[9] Wait the last byte to be received.

[10] Read for the last byte of receive data.

For single-byte reception, skip steps [2] to [6] after step [1]. Then, jump to step [7]. Dummy-read in step

Figure 16.20 Sample Flowchart for Slave Receive 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 CP executes interrupt exception processing. Interrupt sources should be cleared in the excep processing. TDRE and TEND are automatically cleared to 0 by writing the transmit data ICDRT. RDRF are automatically cleared to 0 by reading ICDRR. TDRE is set to 1 agai same time when transmit data is written to ICDRT. When TDRE is cleared to 0, then an data of one byte may be transmitted.

Figure 16.21 shows the timing of the bit synchronous circuit and table 16.4 shows the tim
SCL output changes from low to Hi-Z then SCL is monitored.

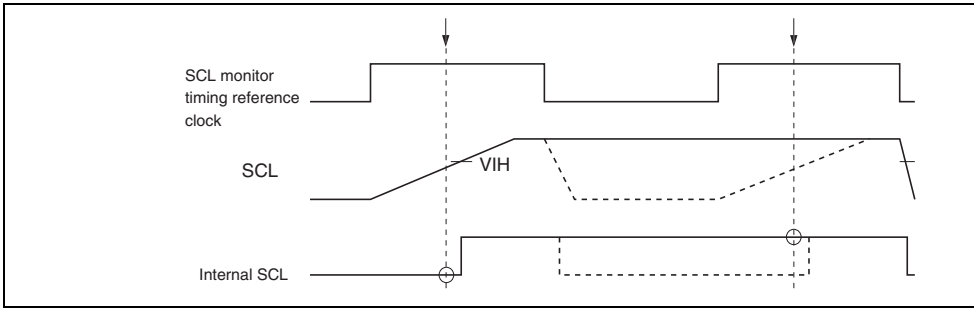


Figure 16.21 Timing of Bit Synchronous Circuit

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

1. The rising edge of the SCL becomes less sharp and longer due to the SCL bus load (
   capacitor and pull-up resistor) than the period defined in section 16.6, Bit Synchron
2. When the slave device elongates the low level period between the eighth and ninth c
   activates the bit synchronous circuit.

16.7.2 Note on Setting WAIT Bit in I2C Bus Mode Register (ICMR)

The WAIT bit in the I^2C bus mode register (ICMR) should be set to 0. Note that if the W
set to 1, when a slave device holds the SCL signal low more than one transfer clock cyc
the eighth clock, the high level period of the ninth clock may be shorter than a given per

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 o
masters, SCL may be output with an unexpected width. To avoid this phenomenon, set t
rate by 1/1.8 or faster than the fastest rate of the other masters. For example, if the faste
rate of the other masters is set to 400 kbps, the IIC transfer rate in this LSI should be set
kbps (= 400/1.18) or more.

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 R
is set to 1. When ICDRR is read around the falling edge of the 8th clock, the clock is only
low in the 8th clock of the next round of data reception. The SCL is then released from its
state without reading ICDRR and the 9th clock is output. As a result, some receive data is
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 by

- 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 inde when not used. (The A/D converter is halted as the initial value. For details, refer to 5.4, Module Standby Function.)

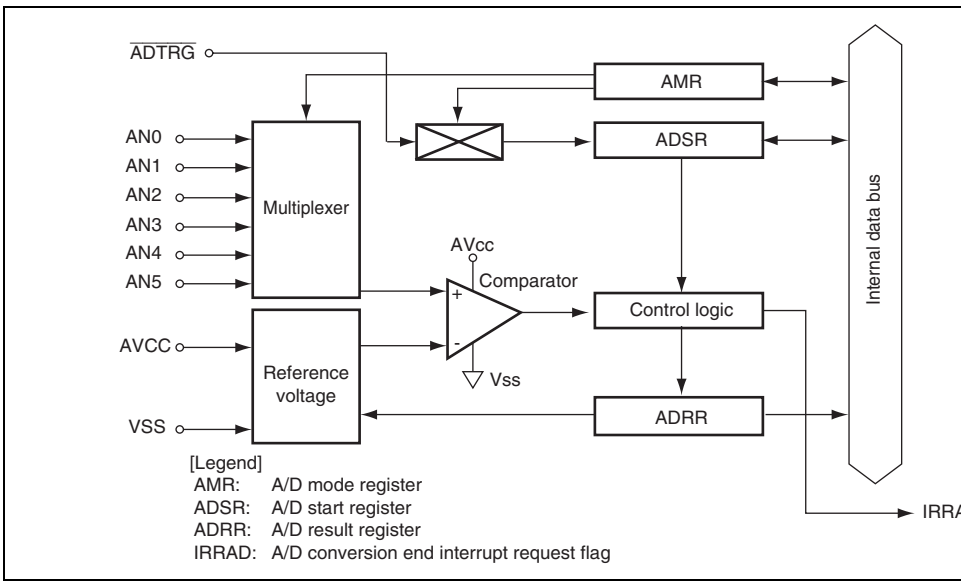


Figure 17.1 Block Diagram of A/D Converter

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 control 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 the upper 10 bits of ADRR. ADRR can be read by the CPU at any time, but the ADRR va during A/D conversion is undefined. After A/D conversion is completed, the conversion r stored as 10-bit data, and this data is retained until the next conversion operation starts. T value of ADRR is undefined. This register must be read in words.

				external trigger input. 0: Disables the A/D conversion start by the ex trigger input. 1: Starts A/D conversion at the rising or falling 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 subsleep mode, the A/D converter can be use when the CPU operating clock is ϕw.

				1001: AN5 101x: No channel selected 11xx: No channel selected The channel selection should be made while th 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 star When conversion is completed, the converted set in ADRR and at the same time this bit is cle 0. If this bit is written to 0, A/D conversion can forcibly terminated.
6	LADS	0	R/W	Ladder Resistance Select 0: Ladder resistance operates while the A/D co is idle. 1: Ladder resistance is halted while the A/D co is idle. The ladder resistance is always halted in stand mode, watch mode, or module standby mode, reset.
5 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

2. When A/D conversion is completed, the result is transferred to the A/D result registe
3. On completion of conversion, the IRRAD flag in IRR2 is set to 1. If the IENAD bit i 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, 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. I trigger input is enabled at the $\overline{\text{ADTRG}}$ pin when the ADTSTCHG bit in PMRB is set to TRGE bit in AMR is set to 1. Then when the input signal edge designated in the ADTRG in IEGR is detected at the $\overline{\text{ADTRG}}$ pin, the ADSF bit in ADSR will be set to 1, starting conversion.

Figure 17.2 shows the timing.

Note: * The $\overline{\text{ADTRG}}$ input pin is shared with the TEST pin. Therefore when the pin the $\overline{\text{ADTRG}}$ pin, reset should be cleared while the 0-fixed signal is input to t pin. Then the ADTSTCHG bit should be set to 1 after the TEST signal is fix

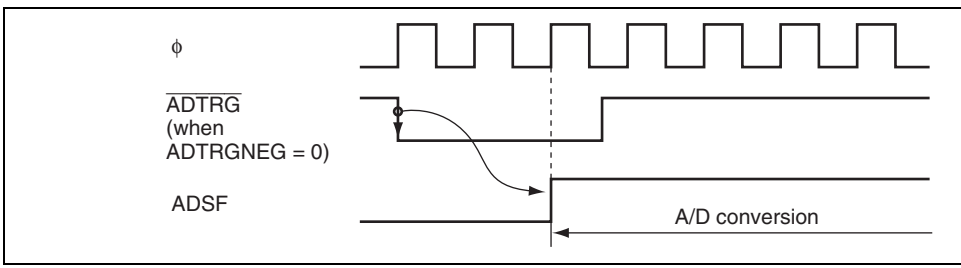


Figure 17.2 External Trigger Input Timing

					Retained*2	Retained*2		
ADRR	Retained*1	Functions	Functions	Retained	Functions/ Retained*2	Functions/ Retained*2	Retained	

Notes: 1. Undefined at a power-on reset.
2. Function if ϕw/2 is selected as the internal clock. Halted and retained otherwis

17.5 Example of Use

An example of how the A/D converter can be used is given below, using channel 1 (pin A 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 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 r stored in ADRR. At the same time bit ADSF is cleared to 0, and the A/D converter g 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 p

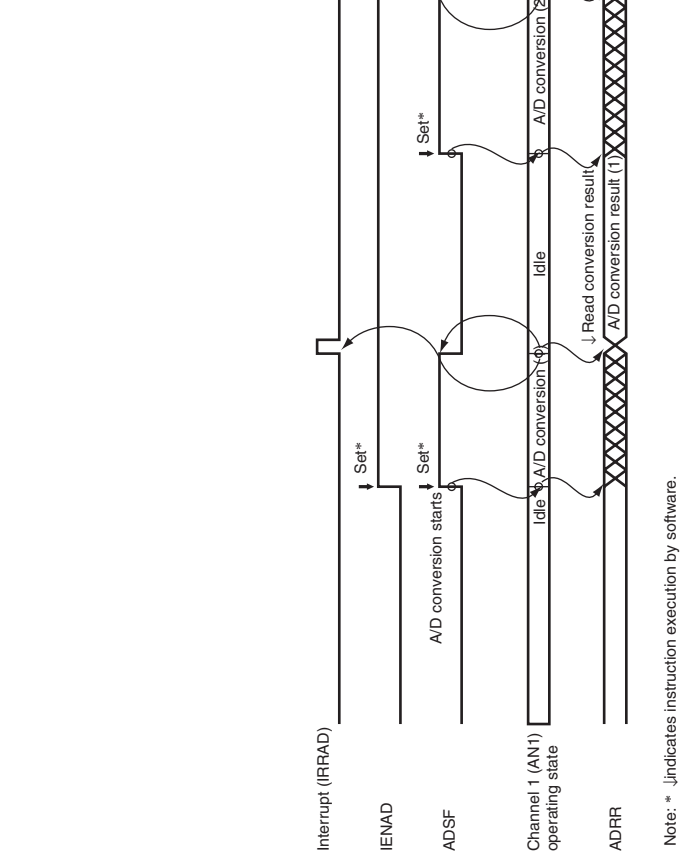


Figure 17.3 Example of A/D Conversion Operation

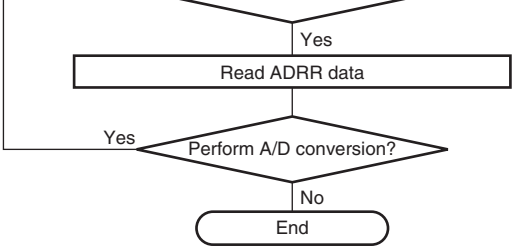


Figure 17.4 Flowchart of Procedure for Using A/D Converter (Polling by Softv

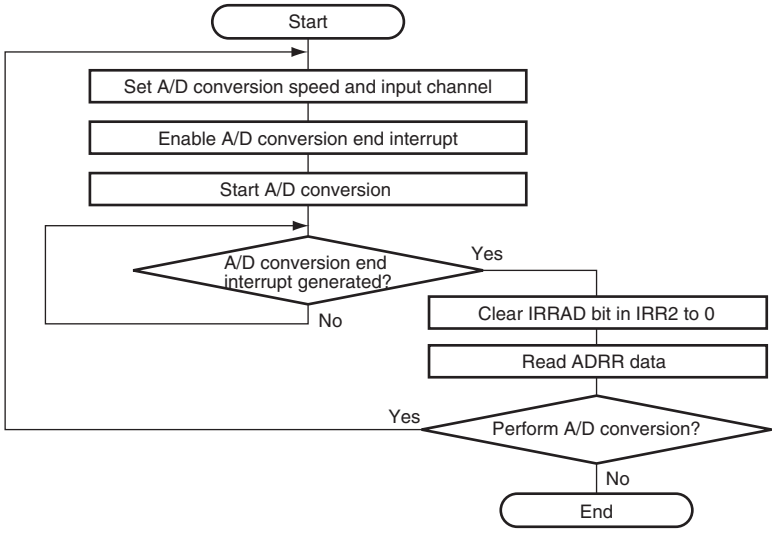


Figure 17.5 Flowchart of Procedure for Using A/D Converter (Interrupts Us

when the digital output changes from the minimum voltage value 0000000000 to 00
(see figure 17.7).

- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion chara
  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 volta
  full-scale voltage. Does not include offset error, full-scale error, or quantization erro

- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes offset e
  scale error, quantization error, and nonlinearity error.

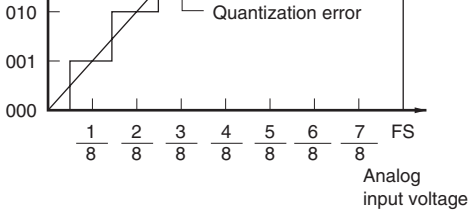


Figure 17.6 A/D Conversion Accuracy Definitions (1)

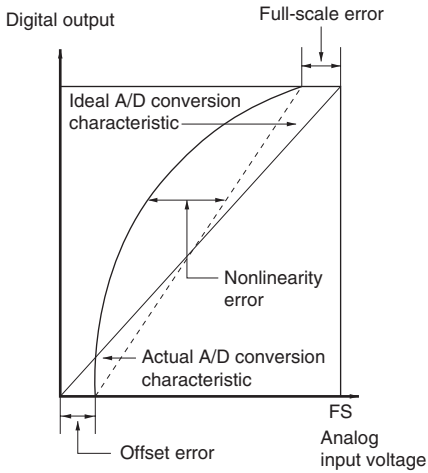


Figure 17.7 A/D Conversion Accuracy Definitions (2)

the signal source impedance is ignored. However, as a low-pass filter effect is obtained i
case, it may not be possible to follow an analog signal with a large differential coefficien
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 adv
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 a
antennas on the mounting board.

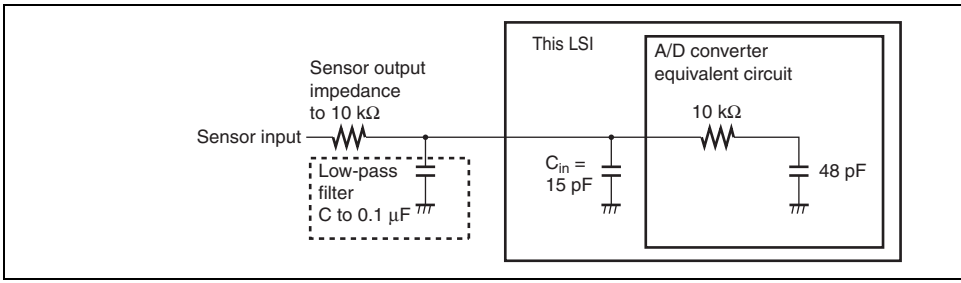


Figure 17.8 Example of Analog Input Circuit

sixteen types of voltages is possible.

- When the internal power supply is selected, the hysteresis characteristics of the comp 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 inde when not used. (A comparator is halted as the initial value. For details, refer to sectio Module Standby Function.)

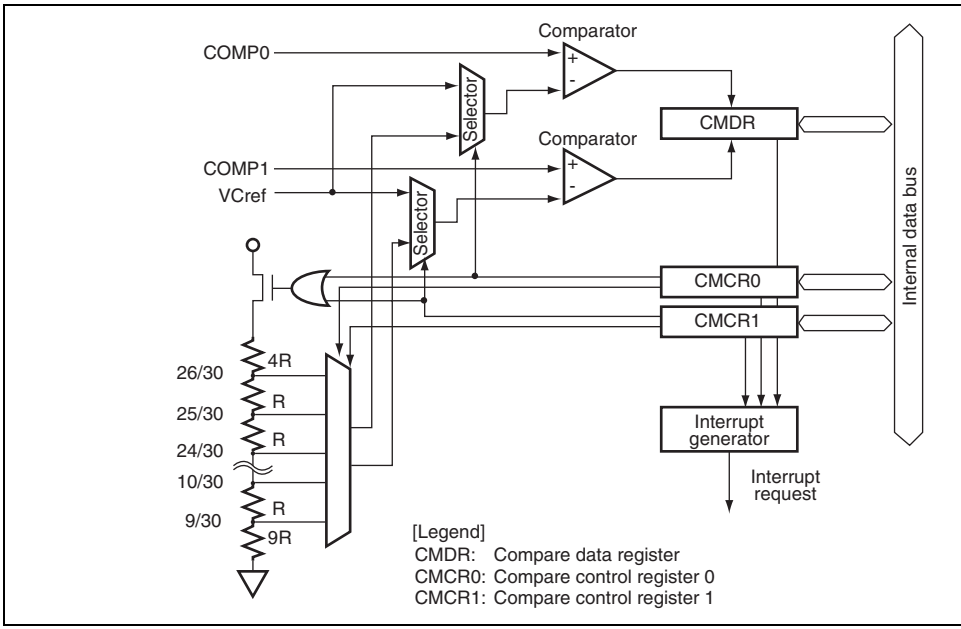


Figure 18.1 Block Diagram of Comparators

18.3 Register Descriptions

The comparators have the following registers. For details on register addresses and regist 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 v 1: Reference voltage is input from VCref pin For the combination of the CMR and CMLS bit

0	CRS0	0	R/W	When CMR = 0 and CMES = 1, V_{IL} will be as When CMR = 1, CRS3 to CRS0 settings are 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 21, Electrical Characteristics.

		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.

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 r
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 hyster
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 Character
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.

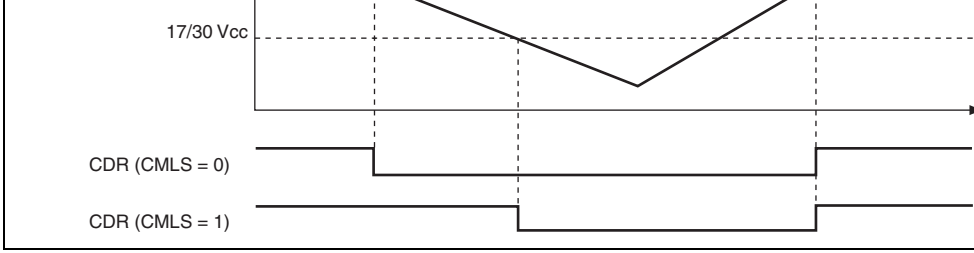


Figure 18.2 Hysteresis/Non-Hysteresis Selection by CDR

18.4.3 Interrupt Setting

When the CDR bit is read while the comparator interrupt is enabled and both the CME ar
bits are set to 1, it is latched in the internal latch. When a difference occurs between the o
the latch and the CDR bit, the interrupt is generated. While the CDR bit is being read, the
is masked.

[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.

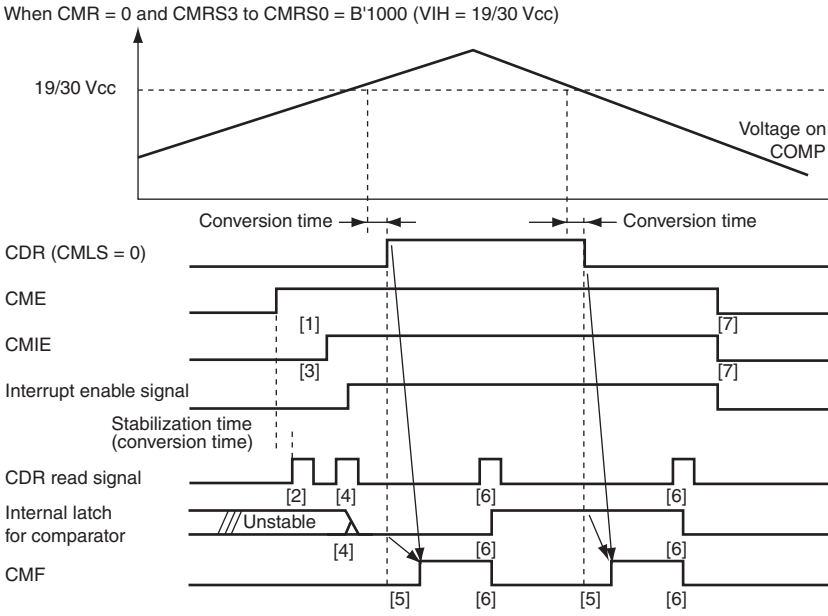


Figure 18.3 Procedure for Setting Interrupt (1)

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 an 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 V cannot be used for any other function.
3. To stop the operation of a comparator, clear the CME0 and CME1 bits in CMCR0 an 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, th operation of the comparator is maintained. Since the comparator operates even in stan mode or watch mode, it returns to the same mode after the specified interrupt is cance 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 is canceled and the current consumption needs to be reduced, stop the comparator by the CME0 and CME1 bits in CMCR0 and CMCR1 to 0 before shifting the mode.

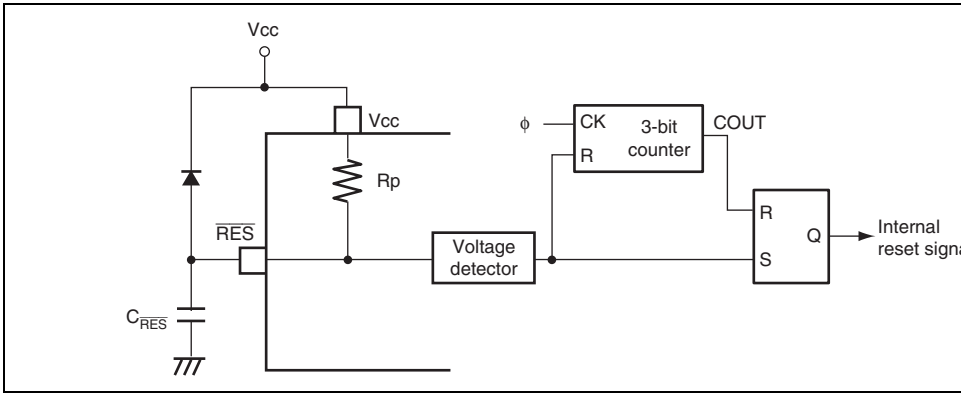


Figure 19.1 Power-On Reset Circuit

The capacitance ($C_{\overline{RES}}$) which is connected to the $\overline{RES}$ pin can be computed using the foll formula; where the $\overline{RES}$ rising time is t. For the on-chip resistor (Rp), see section 21, Ele Characteristics. The power supply rising time (t_vtr) should be shorter than half the $\overline{RES}$ time (t). The $\overline{RES}$ rising time (t) is also should be longer than the oscillation stabilization (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 c the $\overline{RES}$ pin is removed. To remove charge on the $\overline{RES}$ pin, it is recommended that the di should be placed near Vcc. If the power supply voltage (Vcc) rises from the point above V power-on reset may not occur.

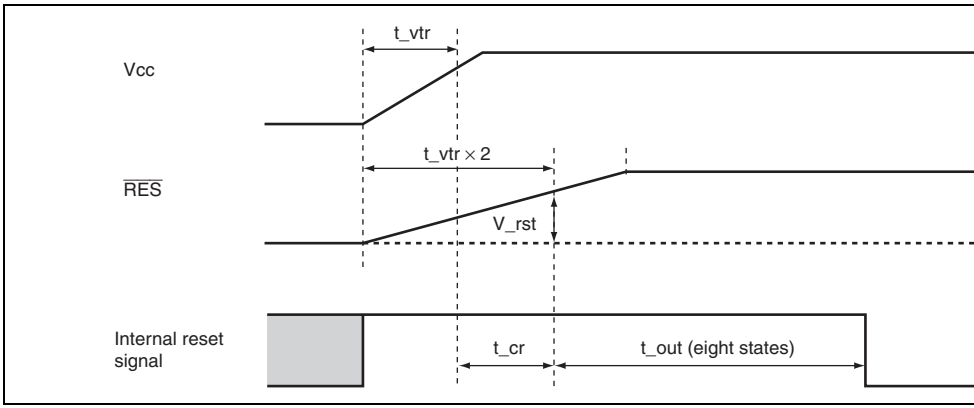


Figure 19.2 Power-On Reset Circuit Operation Timing

- The number of access states is indicated.

2. Register bits
- Bit configurations of the registers are described in the same order as the register addr
- 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 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 spec for an on-chip peripheral module, refer to the section on that on-chip peripheral mod

Flash memory power control register	FLPWCR	H'F022	ROM	8
Erase block register 1	EBR1	H'F023	ROM	8
Flash memory enable register	FENR	H'F02B	ROM	8
RTC interrupt flag register	RTCFLG	H'F067	RTC	8
Second data register/free running counter data register	RSECDR	H'F068	RTC	8
Minute data register	RMINDR	H'F069	RTC	8
Hour data register	RHRDR	H'F06A	RTC	8
Day-of-week data register	RWKDR	H'F06B	RTC	8
RTC control register 1	RTCCR1	H'F06C	RTC	8
RTC control register 2	RTCCR2	H'F06D	RTC	8
Clock source select register	RTCCSR	H'F06F	RTC	8
I^2C bus control register 1	ICCR1	H'F078	IIC2	8
I^2C bus control register 2	ICCR2	H'F079	IIC2	8
I^2C bus mode register	ICMR	H'F07A	IIC2	8
I^2C bus interrupt enable register	ICIER	H'F07B	IIC2	8
I^2C bus status register	ICSR	H'F07C	IIC2	8
Slave address register	SAR	H'F07D	IIC2	8
I^2C bus transmit data register	ICDRT	H'F07E	IIC2	8
I^2C bus receive data register	ICDRR	H'F07F	IIC2	8

Compare control register 1	CMCR1	H'F0DD	Comparator	8
Compare data register	CMDR	H'F0DE	Comparator	8
SS control register H	SSCRH	H'F0E0	SSU*[1]	8
SS control register L	SSCRL	H'F0E1	SSU*[1]	8
SS mode register	SSMR	H'F0E2	SSU*[1]	8
SS enable register	SSER	H'F0E3	SSU*[1]	8
SS status register	SSSR	H'F0E4	SSU*[1]	8
SS receive data register	SSRDR	H'F0E9	SSU*[1]	8
SS transmit data register	SSTDR	H'F0EB	SSU*[1]	8
Timer mode register W	TMRW	H'F0F0	Timer W	8
Timer control register W	TCRW	H'F0F1	Timer W	8
Timer interrupt enable register W	TIERW	H'F0F2	Timer W	8
Timer status register W	TSRW	H'F0F3	Timer W	8
Timer I/O control register 0	TIOR0	H'F0F4	Timer W	8
Timer I/O control register 1	TIOR1	H'F0F5	Timer W	8
Timer counter	TCNT	H'F0F6	Timer W	16
General register A	GRA	H'F0F8	Timer W	16
General register B	GRB	H'F0FA	Timer W	16
General register C	GRC	H'F0FC	Timer W	16
General register D	GRD	H'F0FE	Timer W	16

Event counter L	ECL	H'FF97	AEC	8
Serial mode register 3	SMR3	H'FF98	SCI3	8
Bit rate register 3	BRR3	H'FF99	SCI3	8
Serial control register 3	SCR3	H'FF9A	SCI3	8
Transmit data register 3	TDR3	H'FF9B	SCI3	8
Serial status register 3	SSR3	H'FF9C	SCI3	8
Receive data register 3	RDR3	H'FF9D	SCI3	8
Serial extended mode register	SEMR	H'FFA6	SCI3	8
IrDA control register	IrCR	H'FFA7	IrDA	8
Timer mode register WD	TMWD	H'FFB0	WDT*[3]	8
Timer control/status register WD1	TCSRWD1	H'FFB1	WDT*[3]	8
Timer control/status register WD2	TCSRWD2	H'FFB2	WDT*[3]	8
Timer counter WD	TCWD	H'FFB3	WDT*[3]	8
A/D result register	ADRR	H'FFBC	A/D converter	16
A/D mode register	AMR	H'FFBE	A/D converter	8
A/D start register	ADSR	H'FFBF	A/D converter	8

Port data register B	PDRB	H'FFDE	I/O ports	8
Port pull-up control register 1	PUCR1	H'FFE0	I/O ports	8
Port pull-up control register 3	PUCR3	H'FFE1	I/O ports	8
Port control register 1	PCR1	H'FFE4	I/O ports	8
Port control register 3	PCR3	H'FFE6	I/O ports	8
Port control register 8	PCR8	H'FFEB	I/O ports	8
Port control register 9	PCR9	H'FFEC	I/O ports	8
System control register 1	SYSCR1	H'FFF0	System	8
System control register 2	SYSCR2	H'FFF1	System	8
Interrupt edge select register	IEGR	H'FFF2	Interrupts	8
Interrupt enable register 1	IENR1	H'FFF3	Interrupts	8
Interrupt enable register 2	IENR2	H'FFF4	Interrupts	8
Oscillator control register	OSCCR	H'FFF5	System	8
Interrupt flag register 1	IRR1	H'FFF6	Interrupts	8
Interrupt flag register 2	IRR2	H'FFF7	Interrupts	8
Clock stop register 1	CKSTPR1	H'FFFA	System	8
Clock stop register 2	CKSTPR2	H'FFFB	System	8

Notes: 1. SSU: Synchronous serial communication unit
2. AEC: Asynchronous event counter
3. WDT: Watchdog timer

EBR1	—	—	—	EB4	EB3	EB2	EB1	EB0
FENR	FLSHE	—	—	—	—	—	—	—
RTCFLG	FOIFG	WKIFG	DYIFG	HRIFG	MNIFG	1SEIFG	05SEIFG	025SEIFG
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
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
PUCR8	—	—	—	PUCR84	PUCR83	PUCR82	—	—
PUCR9	—	—	—	—	PUCR93	PUCR92	PUCR91	PUCR90
PODR9	—	—	—	—	P93ODR	P92ODR	P91ODR	P90ODR
TMB1	TMB17	TMB16	—	—	—	TMB12	TMB11	TMB10
TCB1 (R)/ TLB1 (W)	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
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
	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

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	—	—	—	—
TMWD	—	—	—	—	CKS3	CKS2	CKS1	CKS0
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
	ADR1	ADR0	—	—	—	—	—	—
AMR	—	TRGE	CKS1	CKS0	CH3	CH2	CH1	CH0
ADSR	ADSF	LADS	—	—	—	—	—	—
PMR1	—	—	IRQAEC	FTCI	AEVL	CLKOUT	TMOW	AEVH
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

IENR1	IENRTC	—	—	—	—	IENEC2	IEN1	IEN0
IENR2	—	IENAD	—	—	—	IENTB1	—	IENEC
OSCCR	SUBSTP	RFCUT	SUBSEL	—	—	—	OSCF	—
IRR1	—	—	—	—	—	IRREC2	IRRI1	IRRI0
IRR2	—	IRRAD	—	—	—	IRRTB1	—	IRREC
CKSTPR1	—	S3CKSTP	—	ADCKSTP	—	TB1CKSTP	FROMCKSTP	RTCCKST
CKSTPR2	—	TWCKSTP	IICCKSTP	SSUCKSTP	AECCKSTP	WDCKSTP	COMPCKSTP	—

Notes: 1. SSU: Synchronous serial communication unit
2. AEC: Asynchronous event counter
3. WDT: Watchdog timer

RSECDR	—	—	—	—	—	—	—	
RMINDR	—	—	—	—	—	—	—	
RHRDR	—	—	—	—	—	—	—	
RWKDR	—	—	—	—	—	—	—	
RTCCR1	—	—	—	—	—	—	—	
RTCCR2	—	—	—	—	—	—	—	
RTCCSR	Initialized	—	—	—	—	—	—	
ICCR1	Initialized	—	—	—	—	—	—	IIC
ICCR2	Initialized	—	—	—	—	—	—	
ICMR	Initialized	—	—	—	—	—	—	
ICIER	Initialized	—	—	—	—	—	—	
ICSR	Initialized	—	—	—	—	—	—	
SAR	Initialized	—	—	—	—	—	—	
ICDRT	Initialized	—	—	—	—	—	—	
ICDRR	Initialized	—	—	—	—	—	—	
PFCR	Initialized	—	—	—	—	—	—	Sy
PUCR8	Initialized	—	—	—	—	—	—	I/O
PUCR9	Initialized	—	—	—	—	—	—	
PODR9	Initialized	—	—	—	—	—	—	
TMB1	Initialized	—	—	—	—	—	—	Tin
TCB1/TLB1	Initialized	—	—	—	—	—	—	

SSRDR	Initialized	—	—	—	—	—	—	
SSTDR	Initialized	—	—	—	—	—	—	
TMRW	Initialized	—	—	—	—	—	—	T
TCRW	Initialized	—	—	—	—	—	—	
TIERW	Initialized	—	—	—	—	—	—	
TSRW	Initialized	—	—	—	—	—	—	
TIOR0	Initialized	—	—	—	—	—	—	
TIOR1	Initialized	—	—	—	—	—	—	
TCNT	Initialized	—	—	—	—	—	—	
GRA	Initialized	—	—	—	—	—	—	
GRB	Initialized	—	—	—	—	—	—	
GRC	Initialized	—	—	—	—	—	—	
GRD	Initialized	—	—	—	—	—	—	
ECPWCR	Initialized	—	—	—	—	—	—	A
ECPWDR	Initialized	—	—	—	—	—	—	
SPCR	Initialized	—	—	—	—	—	—	S
AEGSR	Initialized	—	—	—	—	—	—	A
ECCR	Initialized	—	—	—	—	—	—	
ECCSR	Initialized	—	—	—	—	—	—	
ECH	Initialized	—	—	—	—	—	—	
ECL	Initialized	—	—	—	—	—	—	

TMWD	Initialized	—	—	—	—	—	—	WD
TCSRWD1	Initialized	—	—	—	—	—	—	
TCSRWD2	Initialized	—	—	—	—	—	—	
TCWD	Initialized	—	—	—	—	—	—	
ADRR	—	—	—	—	—	—	—	A/D
AMR	Initialized	—	—	—	—	—	—	co
ADSR	Initialized	—	—	—	—	—	—	
PMR1	Initialized	—	—	—	—	—	—	I/O
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	—	—	—	—	—	—	

CKSTPR1	Initialized	—	—	—	—	—	—	S
CKSTPR2	Initialized	—	—	—	—	—	—	

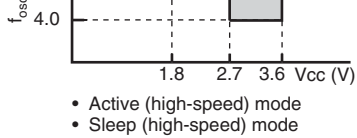
Notes: — is not initialized.

1. SSU: Synchronous serial communication unit
2. AEC: Asynchronous event counter
3. WDT: Watchdog timer

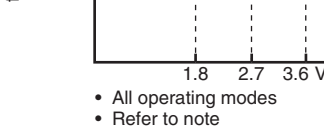
Analog power supply voltage		AV_{CC}	–0.3 to +4.3	V
Input voltage	Other than port B	V_{in}	–0.3 to V_{CC} +0.3	V
	Port B	AV_{in}	–0.3 to AV_{CC} +0.3	V
Operating temperature		T_{opr}	–20 to +75 (general specifications)*[2]	°C
			–40 to +85 (wide temperature range specifications)*[2]	
Storage temperature		T_{stg}	–55 to +125	°C

Notes: 1. Permanent damage may occur to the LSI if absolute maximum ratings are ex
Normal operation should be under the conditions specified in Electrical Chara
Exceeding these values can result in incorrect operation and reduced reliabili

2. The operating temperature range for flash memory programming/erasing is T
+75°C.

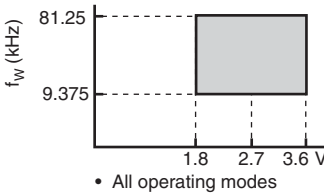


- Active (high-speed) mode
- Sleep (high-speed) mode



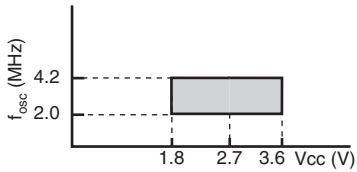
- All operating modes
- Refer to note

R_{osc}/32 used (reference value)

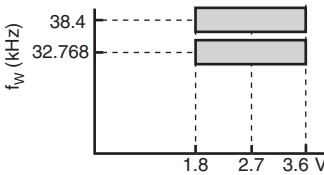


- All operating modes

(2) System clock oscillator selected (4-MHz version)

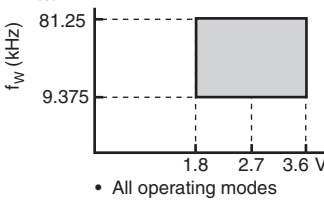


- Active (high-speed) mode
- Sleep (high-speed) mode



- All operating modes
- Refer to note

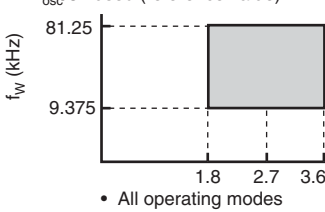
R_{osc}/32 used (reference value)



- All operating modes

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)



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 21.2 Power Supply Voltage and Oscillation Frequency Range (2)

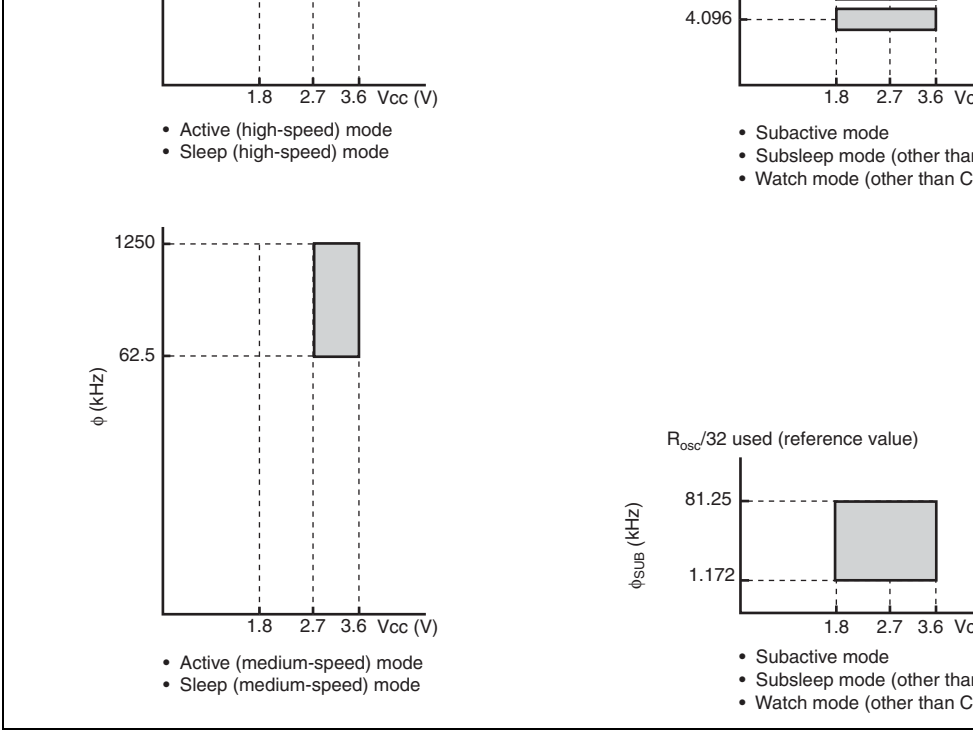


Figure 21.3 Power Supply Voltage and Operating Frequency Range (1)

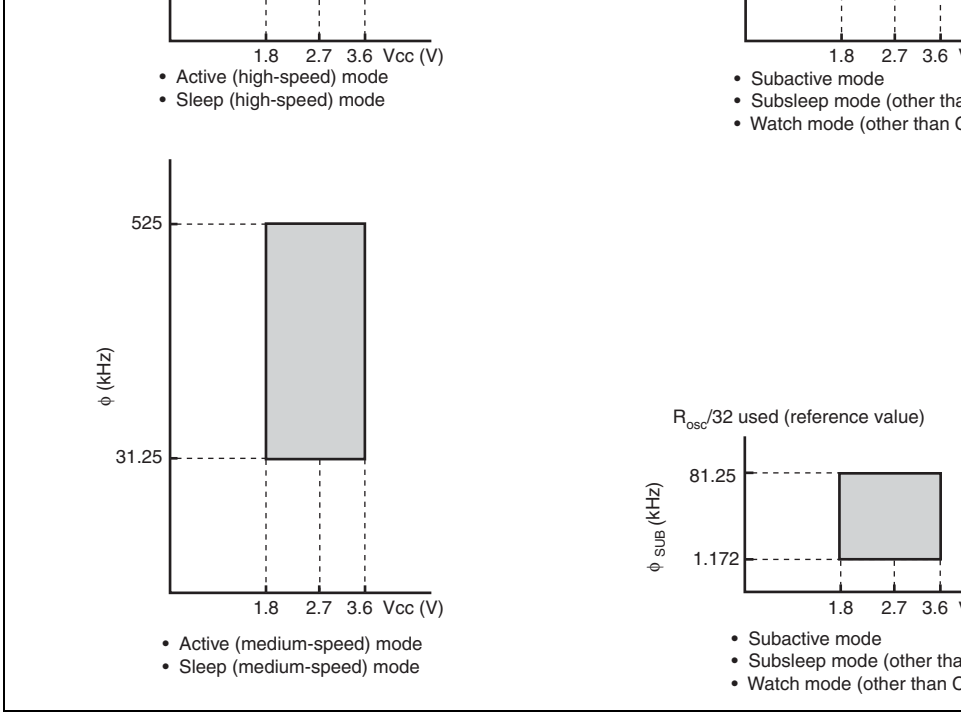


Figure 21.4 Power Supply Voltage and Operating Frequency Range (2)

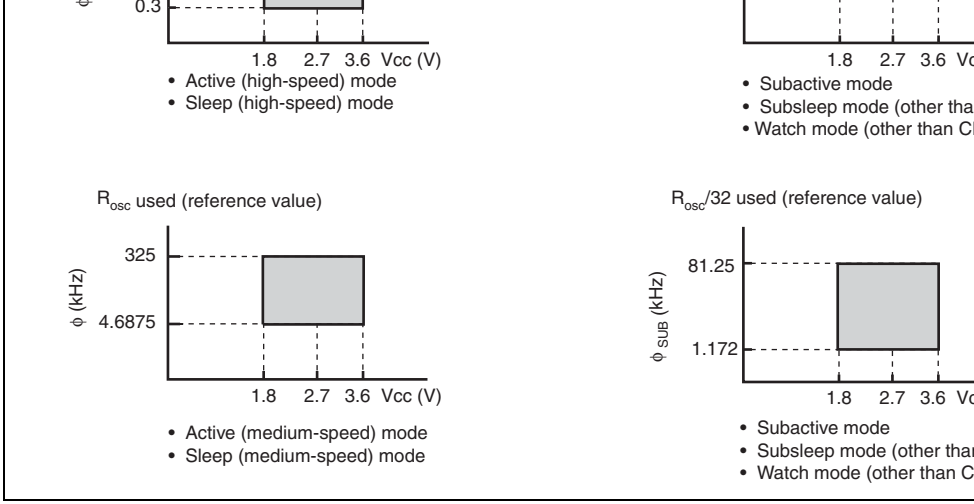


Figure 21.5 Power Supply Voltage and Operating Frequency Range (3)

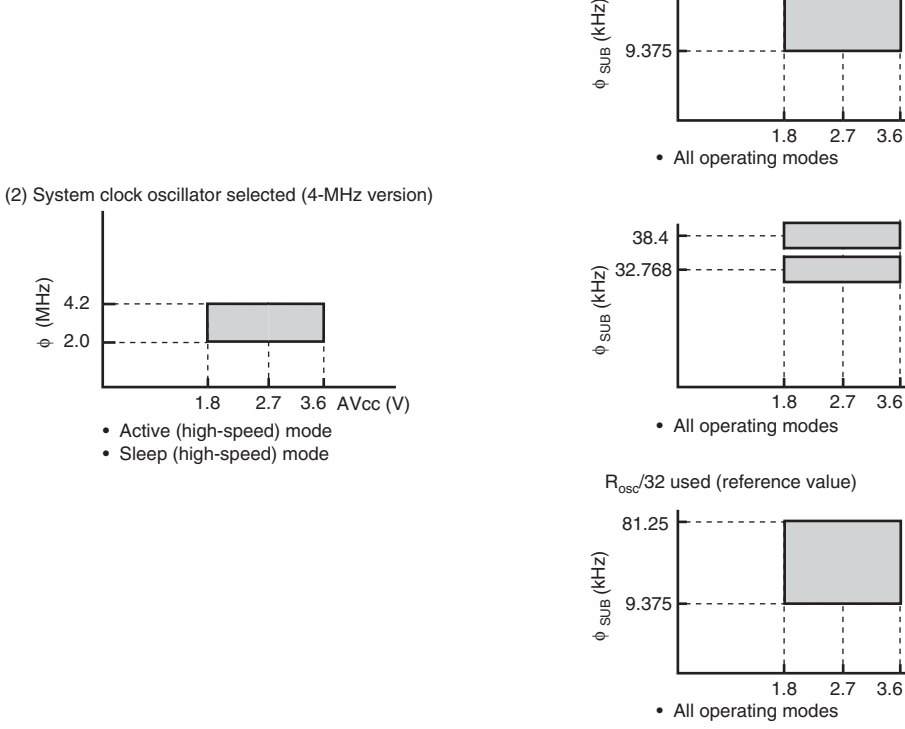


Figure 21.6 Analog Power Supply Voltage and Operating Frequency Rang
A/D Converter (1)

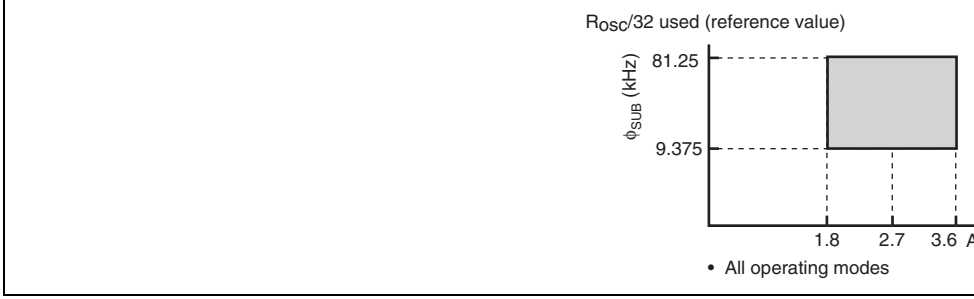


Figure 21.7 Analog Power Supply Voltage and Operating Frequency Range A/D Converter (2)

voltage	$\overline{NMI}$*3, AEVL, AEVH, $\overline{ADTRG}$, SCK3, IRQAEC			
	$\overline{IRQ0}$*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

		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	

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
			Active (high-speed) mode, V_{CC} = 3 V, f_{OSC} = R_{OSC}	—	1.2	—	
			Active (high-speed) mode, V_{CC} = 3 V, f_{OSC} = 4.2 MHz	—	2.6	4.0	
			Active (high-speed) mode, V_{CC} = 3 V, f_{OSC} = 10 MHz	—	6.0	10.0	

			Active (medium-speed) mode, V_{CC} = 3 V, f_{OSC} = 10 MHz, ϕ_{osc}/64	—	0.8	1.3	
Sleep mode supply current	I_{SLEEP}	V_{CC}	V_{CC} = 1.8 V, f_{OSC} = 2 MHz	—	0.9	—	mA
			V_{CC} = 3 V, f_{OSC} = 4.2 MHz	—	2.0	3.2	
			V_{CC} = 3 V, f_{OSC} = 10 MHz	—	4.2	6.4	
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
			V_{CC} = 2.7 V, 32-kHz crystal resonator (ϕ_{SUB} = ϕ_W/2)	—	25	—	
			V_{CC} = 2.7 V, on-chip oscillator/32 (ϕ_{SUB} = ϕ_W = R_{OSC}/32)	—	80	—	
			V_{CC} = 2.7 V, 32-kHz crystal resonator (ϕ_{SUB} = ϕ_W)	—	45	75	

			32-kHz crystal resonator (ϕ_{SUB} = ϕ_W)				
Watch mode supply current	I_{WATCH}	V_{CC}	V_{CC} = 1.8 V, Ta = 25°C, 32-kHz crystal resonator	—	0.5	—	µA
			V_{CC} = 2.7 V, 32-kHz crystal resonator	—	1.5	5.0	
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
			32-kHz crystal resonator not used	—	1.0	5.0	
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	

Active (high-speed) mode (I_{OPE1})	V_{CC}	Only CPU operates	V_{CC}	System cloc oscillator: Cr resonator
Active (medium-speed) mode (I_{OPE2})				Subclock os Pin X1 = GN
Sleep mode	V_{CC}	Only on-chip timers operate	V_{CC}	
Subactive mode	V_{CC}	Only CPU operates	V_{CC}	System cloc oscillator: Cr 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}	Subclock os Crystal reso
Standby mode	V_{CC}	CPU and timers both stop, SUBSTP = 1	V_{CC}	System cloc oscillator: Cr resonator
				Subclock os Pin X1 = Cry 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 releas
4. When bits IRQ0S1 and IRQ0S0 are set to B'01 or B'10, and bits IRQ1S1 and I are set to B'01 or B'10, the maximum value is given V_{CC} + 0.3 (V).

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
			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
On-chip oscillator clock cycle time	t_{ROSC}			0.38	—	3.3	µs
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
Subclock (ϕ_{SUB}) cycle time	t_{subcyc}			1	—	8	t_W
Instruction cycle time				2	—	—	t_{cyc} t_{subcyc}

			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	F a
			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	F 2
			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	F 2
			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	F 2
			V_{CC} = 1.8 V to 3.6 V (4-MHz version)	—	—	24		
		X1		—	—	55.0	ns	

			mode				
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}
		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}
		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 figu

(clock synchronous)	t_{TXD}				t_{cyc} or t_{subcyc}	
Receive data setup time (clock synchronous)	t_{RXS}	400.0	—	—	ns	Fig
Receive data hold time (clock synchronous)	t_{RXH}	400.0	—	—	ns	Fig

	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

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

supply voltage								
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 Ref
	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	Oth sub ope
			AV_{CC} = 2.0 V to 3.6 V V_{CC} = 2.0 V to 3.6 V	—	—	±5.5		
			Subclock operating	—	—	±5.5		Sub sub con = 3
			Other than above	—	—	±7.5		*4
Quantization error				—	—	±0.5	LSB	

Conversion time	AV_{CC} = 2.7 V to 3.6 V V_{CC} = 2.7 V to 3.6 V	12.4	—	124	µs	Syster oscilla selecte
		31	62	124		On-chi is sele Refere (f_{ROSC} =
		—	807	—		ϕ_{SUB} = 3
		—	945	—		ϕ_{SUB} = 3
		—	992	—		ϕ_{SUB} = I Refere (f_{ROSC} =
	Other than AV_{CC} = 2.7 V to 3.6 V V_{CC} = 2.7 V to 3.6 V	29.5	—	124		Syster oscilla selecte
		31	62	124		On-chi is sele Refere (f_{ROSC} =
		—	807	—		ϕ_{SUB} = 3
		—	945	—		ϕ_{SUB} = 3
		—	992	—		ϕ_{SUB} = I Refere (f_{ROSC} =

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

3. AI_{STOP2} is the current flowing at a reset, in standby mode or watch mode, throug ladder resistor while the A/D converter is idle.

4. Conversion time is 29.5 µs.

Conversion time		—	—	15	µs	
External input reference voltage	VCref pin	0.9	—	$0.9 \times V_{CC}$	V	
Internal resistance compare voltage		0.9	—	$26/30 \times V_{CC}$	V	
Comparator input voltage	COMP0 and COMP1 pins	–0.3	—	$AV_{CC} + 0.3$	V	
Ladder resistance		—	3	—	MΩ	R

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.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit
				Min.	Typ.	Max.	
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 cou
reaches 255 and an internal reset occurs while the on-chip oscillator is select

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 is selecte (referenc
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	

Item		Symbol	Condition	Min.	Typ.	Max
Programming time (per 128 bytes)*1*2*4		t_P		—	7	200
Erasing time (per block)*1*3*6		t_E		—	100	1200
Maximum programming count		N_{WEC}		1000 *8*11	10000 *9	—
				100 *8*12	10000 *9	—
Data retention time		t_{DRP}		10*10	—	—
Programming	Wait time after setting SWE bit*1	x		1	—	—
	Wait time after setting PSU bit*1	y		50	—	—
	Wait time after setting P bit*1*4	z1	$1 \le n \le 6$	28	30	32
		z2	$7 \le n \le 1000$	198	200	202
		z3	Additional-programming	8	10	12
	Wait time after clearing P bit*1	α		5	—	—
	Wait time after clearing PSU bit*1	β		5	—	—
	Wait time after setting PV bit*1	γ		4	—	—
	Wait time after dummy write*1	ε		2	—	—
	Wait time after clearing PV bit*1	η		2	—	—
	Wait time after clearing SWE bit*1	θ		100	—	—
	Maximum programming count*1*4*5	N		—	—	1000

Wait time after clearing SWE bit	θ		100		
Maximum erasing count*[1]*[6]*[7]	N		—	—	120

Notes: 1. Make the time settings in accordance with the programming/erasing algorithm

2. The programming time for 128 bytes. (Indicates the total time for which the P flash memory control register 1 (FLMCR1) is set. The programming-verifying not included.)

3. The time required to erase one block. (Indicates the total time for which the E flash memory control register 1 (FLMCR1) is set. The erasing-verifying time is 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 o and z3, so that it does not exceed the programming time maximum value (t_P (The wait time after setting P bit (z1, z2) should be changed as follows accord 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) × ma erasing count (N)

7. Set the maximum erasing count (N) according to the actual set value of (z), s does not exceed the erasing time maximum value (t_E (max.)).

8. The minimum number of times in which all characteristics are guaranteed foll reprogramming. (The guarantee covers the range from 1 to the minimum valu

9. Reference value at 25°C. (Guideline showing programming count over which functioning will be retained under normal circumstances.)

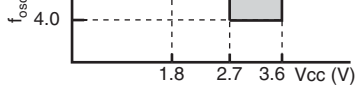
10. Data retention characteristics within the range indicated in the specifications, the minimum programming count.

11. Applies to an operating voltage range when reading data of 2.7 to 3.6 V.

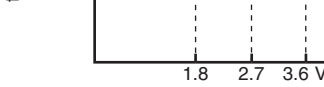
12. Applies to an operating voltage range when reading data of 1.8 to 3.6 V.

	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 ex
Normal operation should be under the conditions specified in Electrical Chara
Exceeding these values can result in incorrect operation and reduced reliabili

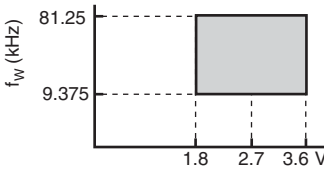


- Active (high-speed) mode
- Sleep (high-segment) mode



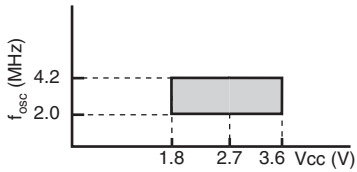
- All operating modes
- Refer to note

R_{osc}/32 used (reference value)

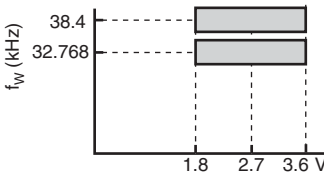


- All operating modes

(2) System clock oscillator selected (4-MHz version)

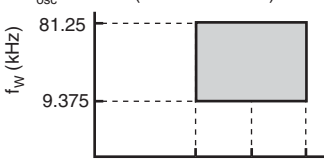


- Active (high-speed) mode
- Sleep (high-speed) mode



- All operating modes
- Refer to note

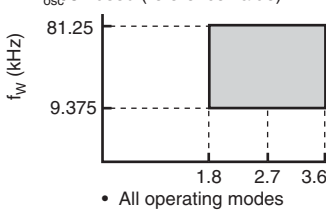
R_{osc}/32 used (reference value)



- All operating modes

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 21.8 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)

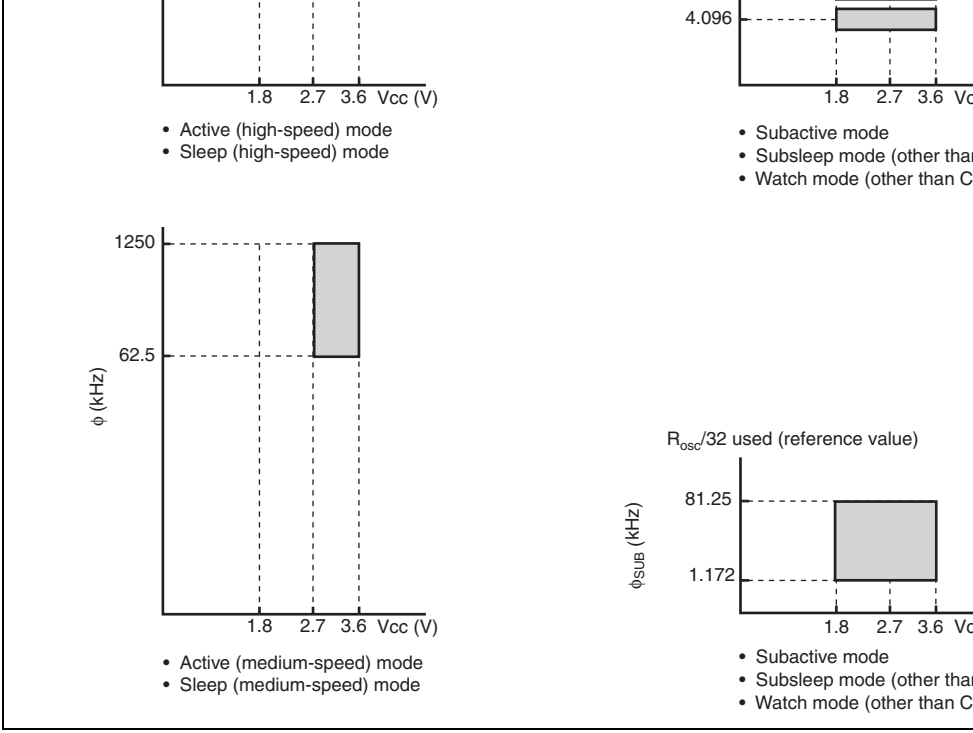


Figure 21.10 Power Supply Voltage and Operating Frequency Range (1)

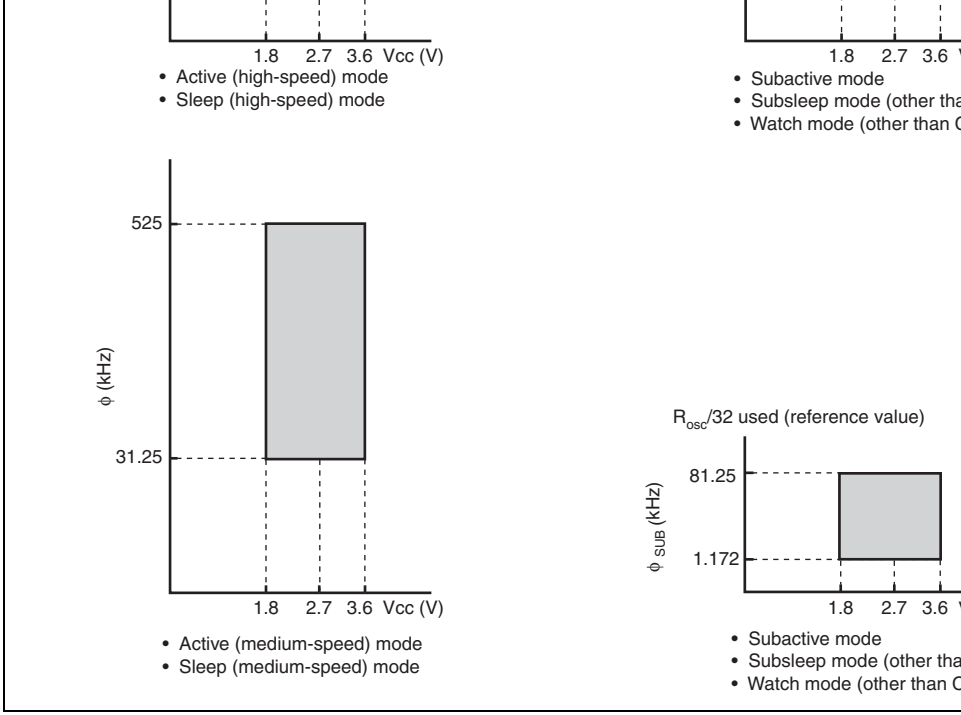


Figure 21.11 Power Supply Voltage and Operating Frequency Range (2

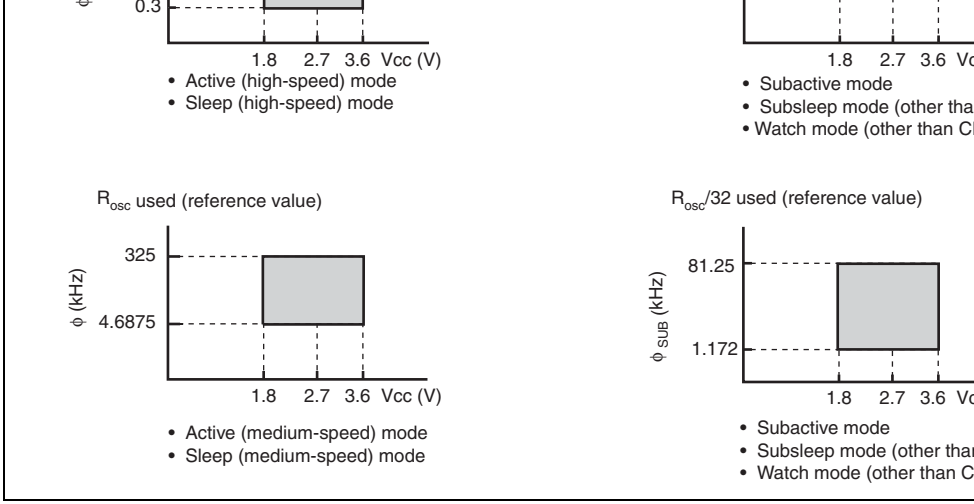


Figure 21.12 Power Supply Voltage and Operating Frequency Range (3)

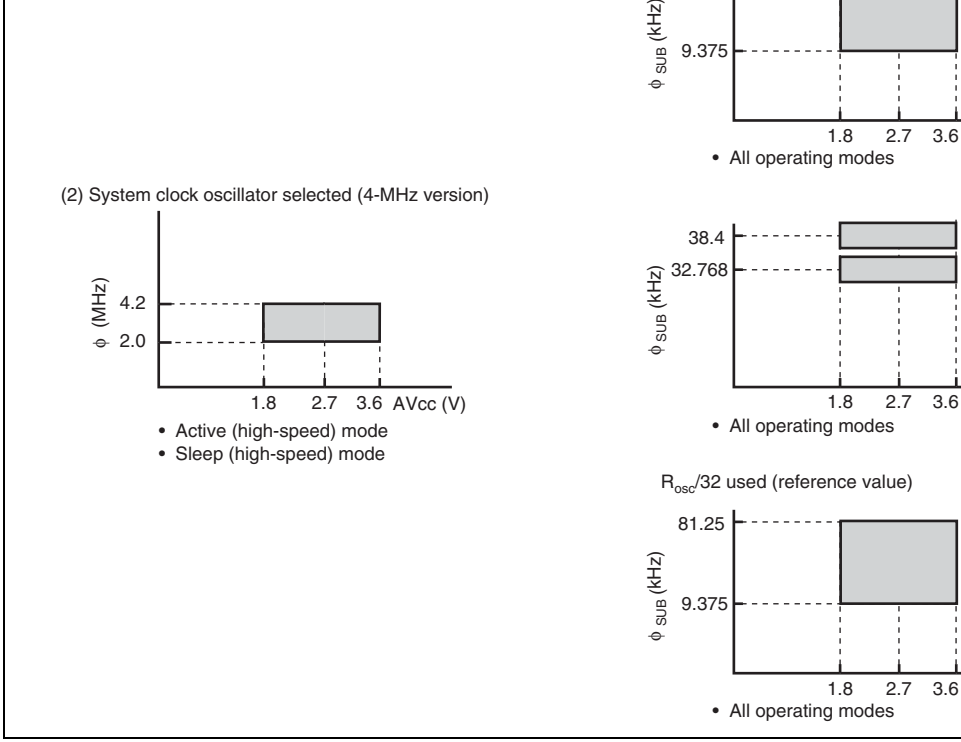


Figure 21.13 Analog Power Supply Voltage and Operating Frequency Rang
A/D Converter (1)

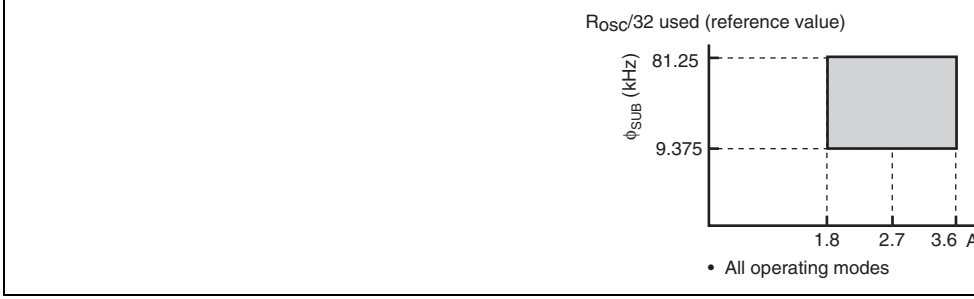


Figure 21.14 Analog Power Supply Voltage and Operating Frequency Range A/D Converter (2)

voltage	$\overline{NMI}$, AEVL, AEVH, $\overline{ADTRG}$, SCK3, IRQAEC			
	$\overline{IRQ0}$*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

		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	

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 = 1.
			Active (high-speed) mode, V_{CC} = 3 V, f_{OSC} = R_{OSC}	—	0.6	—		Max = 1. Ref
			Active (high-speed) mode, V_{CC} = 3 V, f_{OSC} = 4.2 MHz	—	2.0	3.0		*1*2 4-M
			Active (high-speed) mode, V_{CC} = 3 V, f_{OSC} = 10 MHz	—	4.5	6.8		*1*2 10-I

			ϕ_{osc}/64					
			Active (medium-speed) mode, V_{CC} = 3 V, f_{OSC} = 10 MHz, ϕ_{osc}/64	—	0.5	0.7		*1*2 10-MH
Sleep mode supply current	I_{SLEEP}	V_{CC}	V_{CC} = 1.8 V, f_{OSC} = 2 MHz	—	0.3	—	mA	Max. = 1.1
			V_{CC} = 3 V, f_{OSC} = 4.2 MHz	—	1.0	1.5		*1*2 4-MH
			V_{CC} = 3 V, f_{OSC} = 10 MHz	—	1.8	2.7		*1*2 10-MH
Subactive mode supply current	I_{SUB}	V_{CC}	V_{CC} = 1.8 V, 32-kHz crystal resonator (ϕ_{SUB} = ϕ_W/2)	—	4.0	—	µA	*1*2 Refer
			V_{CC} = 2.7 V, 32-kHz crystal resonator (ϕ_{SUB} = ϕ_W/8)	—	3.6	—		*1*2 Refer
			V_{CC} = 2.7 V, 32-kHz crystal resonator (ϕ_{SUB} = ϕ_W/2)	—	7.4	—		*1*2 Refer
			V_{CC} = 2.7 V, on-chip oscillator/32 (ϕ_{SUB} = ϕ_W = R_{OSC}/32)	—	40	—		*1*2 Refer
			V_{CC} = 2.7 V, 32-kHz crystal resonator (ϕ_{SUB} = ϕ_W)	—	13	25		*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
			V_{CC} = 2.7 V, 32-kHz crystal resonator	—	1.5	5.0	
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
			32-kHz crystal resonator not used	—	1.0	5.0	
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	

Active (high-speed) mode (I_{OPE1})	V_{CC}	Only CPU operates	V_{CC}	System cloc oscillator: Cr resonator
Active (medium-speed) mode (I_{OPE2})				Subclock os Pin X1 = GN
Sleep mode	V_{CC}	Only on-chip timers operate	V_{CC}	
Subactive mode	V_{CC}	Only CPU operates	V_{CC}	System cloc oscillator: Cr 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}	Subclock os Crystal reso
Standby mode	V_{CC}	CPU and timers both stop, SUBSTP = 1	V_{CC}	System cloc oscillator: Cr resonator
				Subclock os Pin X1 = Cry 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 I are set to B'01 or B'10, the maximum value is given V_{CC} + 0.3 (V).

Item	Symbol	Pins	Test Condition	Min.	Typ.	Max.	Unit
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
			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
On-chip oscillator clock cycle time	t_{ROSC}			0.38	—	3.3	µs
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

			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
			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
			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
			V_{cc} = 1.8 V to 3.6 V (4-MHz version)	95	—	—	
		X1		—	15.26 or 13.02	—	µs

			(4-MHz version)				
		X1		—	—	55.0	ns
$\overline{\text{RES}}$ pin low width	t_{REL}	$\overline{\text{RES}}$	At switching on or other than below	t_{rc} + 20 × t_{cyc}	—	—	µs
			Active mode or sleep mode	20			t_{cyc}
Input pin high width	t_{IH}	$\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, $\overline{\text{NMI}}$, IRQAEC, $\overline{\text{ADTRG}}$, FTCI, FTIOA, FTIOB, FTIOC, FTIOD		2	—	—	t_{cyc} t_{subcyc}
		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}	$\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, $\overline{\text{NMI}}$, IRQAEC, $\overline{\text{ADTRG}}$, FTCI, FTIOA, FTIOB, FTIOC, FTIOD		2	—	—	t_{cyc} t_{subcyc}
		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 figu

(clock synchronous)	t_{TXD}				t_{cyc} or t_{subcyc}	
Receive data setup time (clock synchronous)	t_{RXS}	400.0	—	—	ns	Fi
Receive data hold time (clock synchronous)	t_{RXH}	400.0	—	—	ns	Fi

	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

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

supply voltage								
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 Refe valu
	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	Othe subc oper
			AV_{CC} = 2.0 V to 3.6 V V_{CC} = 2.0 V to 3.6 V	—	—	±5.5		
			Subclock operating	—	—	±5.5		Suba subs conv 31/$\phi_{}$
			Other than above	—	—	±7.5		*4
Quantization error				—	—	±0.5	LSB	

Conversion time	AV_{CC} = 2.7 V to 3.6 V V_{CC} = 2.7 V to 3.6 V	12.4	—	124	µs	Syster oscilla selecte
		31	62	124		On-chi is sele Refere (f_{ROSC} =
		—	807	—		ϕ_{SUB} = 3
		—	945	—		ϕ_{SUB} = 3
		—	992	—		ϕ_{SUB} = I Refere (f_{ROSC} =
	Other than AV_{CC} = 2.7 V to 3.6 V V_{CC} = 2.7 V to 3.6 V	29.5	—	124		Syster oscilla selecte
		31	62	124		On-chi is sele Refere (f_{ROSC} =
		—	807	—		ϕ_{SUB} = 3
		—	945	—		ϕ_{SUB} = 3
		—	992	—		ϕ_{SUB} = I Refere (f_{ROSC} =

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
3. AI_{STOP2} is the current flowing at a reset, in standby mode or watch mode, throug ladder resistor while the A/D converter is idle.
4. Conversion time is 29.5 µs.

Conversion time		—	—	15	µs	
External input reference voltage	VCref pin	0.9	—	$0.9 \times V_{CC}$	V	
Internal resistance compare voltage		0.9	—	$26/30 \times V_{CC}$	V	
Comparator input voltage	COMP0 and COMP1 pins	–0.3	—	$AV_{CC} + 0.3$	V	
Ladder resistance		—	3	—	MΩ	

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**
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 cou reaches 255 and an internal reset occurs while the on-chip oscillator is select

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 $\overline{\text{RES}}$ rising time.				
Reset count time	t_out		0.8	—	4.0	µs	
			3.2	—	26.7		On-chip is selecte (referenc
Count start time	t_cr		Adjustable by the value of the external capacitor connected to the $\overline{\text{RES}}$ pin.				
Pull-up resistance	R_P		60	100	—	kΩ	

Figure 21.15 Clock Input Timing

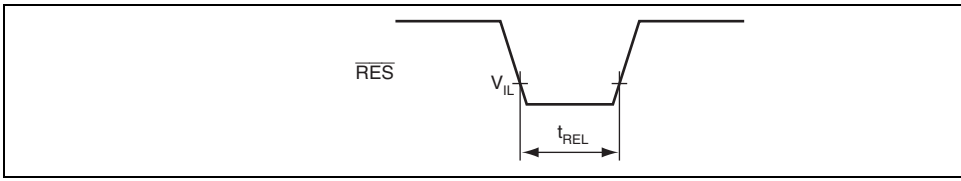


Figure 21.16 RES Low Width Timing

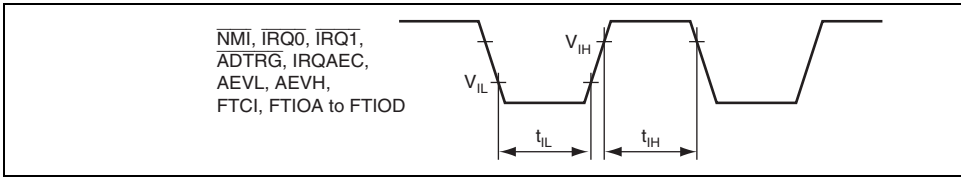


Figure 21.17 Input Timing

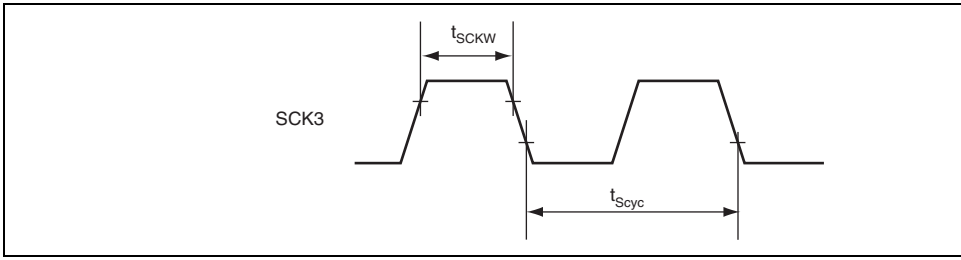


Figure 21.18 SCK3 Input Clock Timing

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

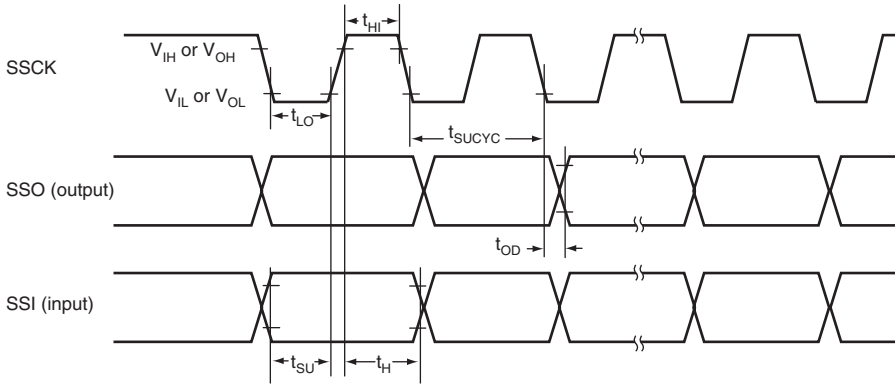


Figure 21.20 SSU Input/Output Timing in Clock Synchronous Mode

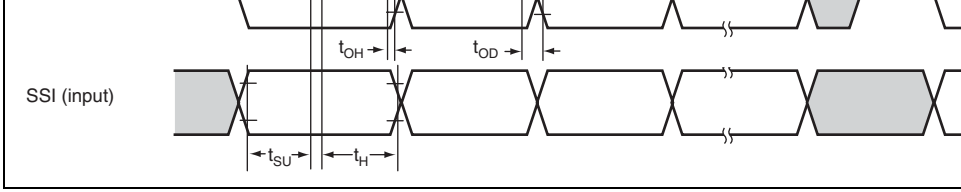


Figure 21.21 SSU Input/Output Timing
(Four-Line Bus Communication Mode, Master, CPHS = 1)

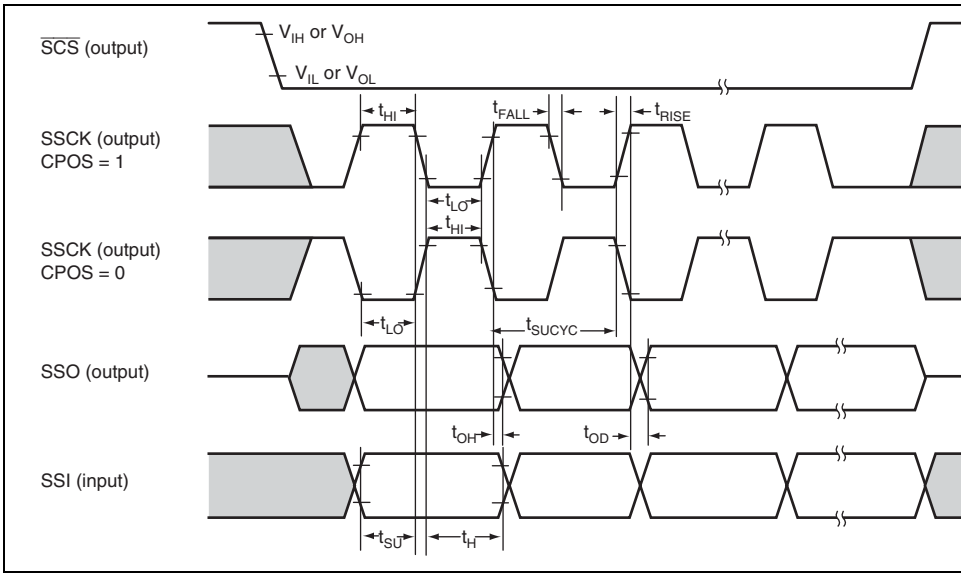


Figure 21.22 SSU Input/Output Timing
(Four-Line Bus Communication Mode, Master, CPHS = 0)

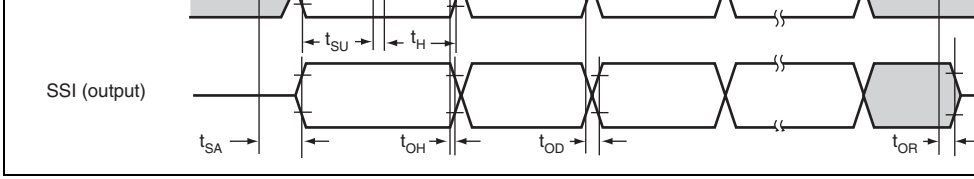


Figure 21.23 SSU Input/Output Timing
(Four-Line Bus Communication Mode, Slave, CPHS = 1)

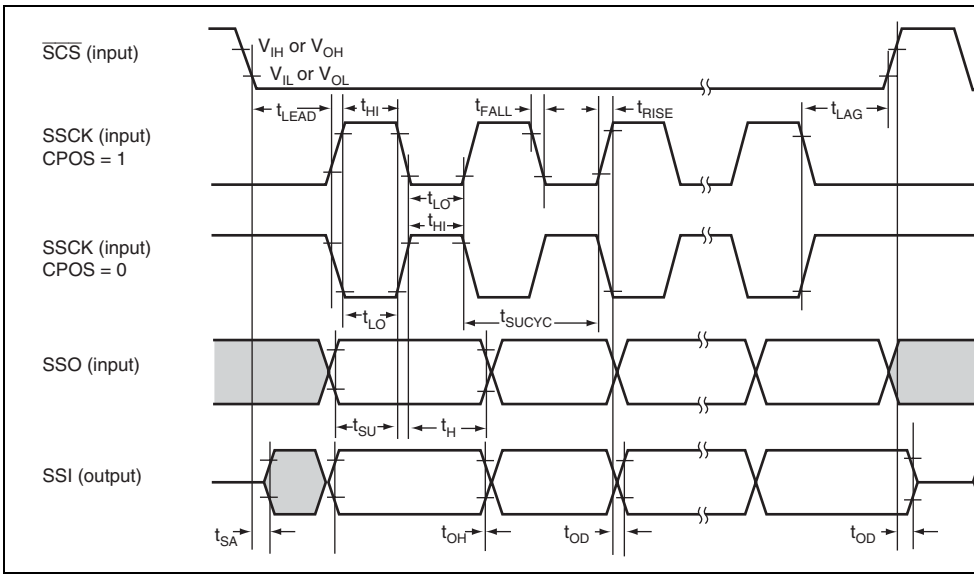


Figure 21.24 SSU Input/Output Timing
(Four-Line Bus Communication Mode, Slave, CPHS = 0)

Note: * S, P, and Sr represent the following:
S: Start condition
P: Stop condition
Sr: Repeated start condition

Figure 21.25 I^2C Bus Interface Input/Output Timing

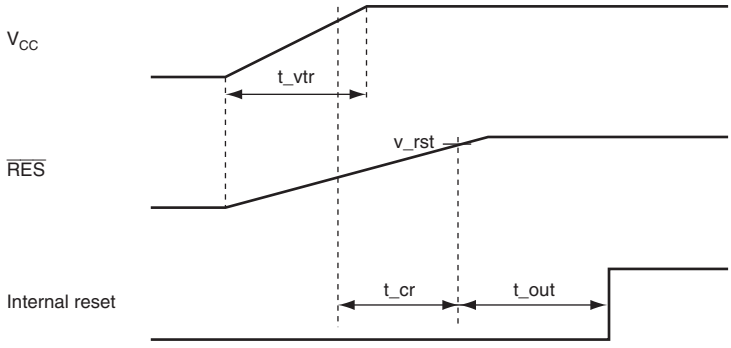


Figure 21.26 Power-On Reset Circuit Reset Timing

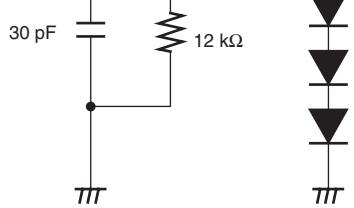


Figure 21.27 Output Load Condition

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

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 trans 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 (R0 to R7 and E0 to E7).

	MOV.B Rs, Rd	B		2							Rs8 → Rd8	—	—	↕	↕	0
	MOV.B @ERs, Rd	B			2						@ERs → Rd8	—	—	↕	↕	0
	MOV.B @(d:16, ERs), Rd	B				4					@(d:16, ERs) → Rd8	—	—	↕	↕	0
	MOV.B @(d:24, ERs), Rd	B				8					@(d:24, ERs) → Rd8	—	—	↕	↕	0
	MOV.B @ERs+, Rd	B					2				@ERs → Rd8 ERs32+1 → ERs32	—	—	↕	↕	0
	MOV.B @aa:8, Rd	B						2			@aa:8 → Rd8	—	—	↕	↕	0
	MOV.B @aa:16, Rd	B						4			@aa:16 → Rd8	—	—	↕	↕	0
	MOV.B @aa:24, Rd	B						6			@aa:24 → Rd8	—	—	↕	↕	0
	MOV.B Rs, @ERd	B			2						Rs8 → @ERd	—	—	↕	↕	0
	MOV.B Rs, @(d:16, ERd)	B				4					Rs8 → @(d:16, ERd)	—	—	↕	↕	0
	MOV.B Rs, @(d:24, ERd)	B				8					Rs8 → @(d:24, ERd)	—	—	↕	↕	0
	MOV.B Rs, @–ERd	B					2				ERd32–1 → ERd32 Rs8 → @ERd	—	—	↕	↕	0
	MOV.B Rs, @aa:8	B						2			Rs8 → @aa:8	—	—	↕	↕	0
	MOV.B Rs, @aa:16	B						4			Rs8 → @aa:16	—	—	↕	↕	0
	MOV.B Rs, @aa:24	B						6			Rs8 → @aa:24	—	—	↕	↕	0
	MOV.W #xx:16, Rd	W	4								#xx:16 → Rd16	—	—	↕	↕	0
	MOV.W Rs, Rd	W		2							Rs16 → Rd16	—	—	↕	↕	0
	MOV.W @ERs, Rd	W			2						@ERs → Rd16	—	—	↕	↕	0
	MOV.W @(d:16, ERs), Rd	W				4					@(d:16, ERs) → Rd16	—	—	↕	↕	0
	MOV.W @(d:24, ERs), Rd	W				8					@(d:24, ERs) → Rd16	—	—	↕	↕	0
	MOV.W @ERs+, Rd	W					2				@ERs → Rd16 ERs32+2 → @ERd32	—	—	↕	↕	0
	MOV.W @aa:16, Rd	W						4			@aa:16 → Rd16	—	—	↕	↕	0
	MOV.W @aa:24, Rd	W						6			@aa:24 → Rd16	—	—	↕	↕	0
	MOV.W Rs, @ERd	W			2						Rs16 → @ERd	—	—	↕	↕	0
	MOV.W Rs, @(d:16, ERd)	W				4					Rs16 → @(d:16, ERd)	—	—	↕	↕	0
	MOV.W Rs, @(d:24, ERd)	W				8					Rs16 → @(d:24, ERd)	—	—	↕	↕	0

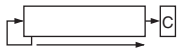
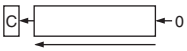
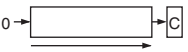
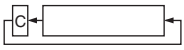
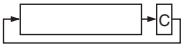
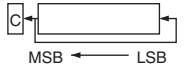
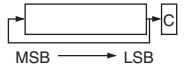
	MOV.L ERs, ERd	L		2								ERs32 → ERd32	—	—	↕	↕	0	
	MOV.L @ERs, ERd	L			4							@ERs → ERd32	—	—	↕	↕	0	
	MOV.L @(d:16, ERs), ERd	L				6						@(d:16, ERs) → ERd32	—	—	↕	↕	0	
	MOV.L @(d:24, ERs), ERd	L				10						@(d:24, ERs) → ERd32	—	—	↕	↕	0	
	MOV.L @ERs+, ERd	L					4					@ERs → ERd32 ERs32+4 → ERs32	—	—	↕	↕	0	
	MOV.L @aa:16, ERd	L						6				@aa:16 → ERd32	—	—	↕	↕	0	
	MOV.L @aa:24, ERd	L						8				@aa:24 → ERd32	—	—	↕	↕	0	
	MOV.L ERs, @ERd	L			4							ERs32 → @ERd	—	—	↕	↕	0	
	MOV.L ERs, @(d:16, ERd)	L				6						ERs32 → @(d:16, ERd)	—	—	↕	↕	0	
	MOV.L ERs, @(d:24, ERd)	L				10						ERs32 → @(d:24, ERd)	—	—	↕	↕	0	
	MOV.L ERs, @–ERd	L					4					ERd32–4 → ERd32 ERs32 → @ERd	—	—	↕	↕	0	
	MOV.L ERs, @aa:16	L						6				ERs32 → @aa:16	—	—	↕	↕	0	
	MOV.L ERs, @aa:24	L						8				ERs32 → @aa:24	—	—	↕	↕	0	
POP	POP.W Rn	W									2	@SP → Rn16 SP+2 → SP	—	—	↕	↕	0	
	POP.L ERn	L									4	@SP → ERn32 SP+4 → SP	—	—	↕	↕	0	
PUSH	PUSH.W Rn	W									2	SP–2 → SP Rn16 → @SP	—	—	↕	↕	0	
	PUSH.L ERn	L									4	SP–4 → SP ERn32 → @SP	—	—	↕	↕	0	
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					

	ADD.W Rs, Rd	W		2								Rd16+Rs16 → Rd16	—	(1)	↕	↕	↕
	ADD.L #xx:32, ERd	L	6									ERd32+#xx:32 → ERd32	—	(2)	↕	↕	↕
	ADD.L ERs, ERd	L		2								ERd32+ERs32 → ERd32	—	(2)	↕	↕	↕
ADDX	ADDX.B #xx:8, Rd	B	2									Rd8+#xx:8 +C → Rd8	—	↕	↕	(3)	↕
	ADDX.B Rs, Rd	B		2								Rd8+Rs8 +C → Rd8	—	↕	↕	(3)	↕
ADDS	ADDS.L #1, ERd	L		2								ERd32+1 → ERd32	—	—	—	—	—
	ADDS.L #2, ERd	L		2								ERd32+2 → ERd32	—	—	—	—	—
	ADDS.L #4, ERd	L		2								ERd32+4 → ERd32	—	—	—	—	—
INC	INC.B Rd	B		2								Rd8+1 → Rd8	—	—	↕	↕	↕
	INC.W #1, Rd	W		2								Rd16+1 → Rd16	—	—	↕	↕	↕
	INC.W #2, Rd	W		2								Rd16+2 → Rd16	—	—	↕	↕	↕
	INC.L #1, ERd	L		2								ERd32+1 → ERd32	—	—	↕	↕	↕
	INC.L #2, ERd	L		2								ERd32+2 → ERd32	—	—	↕	↕	↕
DAA	DAA Rd	B		2								Rd8 decimal adjust → Rd8	—	*	↕	↕	*
SUB	SUB.B Rs, Rd	B		2								Rd8–Rs8 → Rd8	—	↕	↕	↕	↕
	SUB.W #xx:16, Rd	W	4									Rd16–#xx:16 → Rd16	—	(1)	↕	↕	↕
	SUB.W Rs, Rd	W		2								Rd16–Rs16 → Rd16	—	(1)	↕	↕	↕
	SUB.L #xx:32, ERd	L	6									ERd32–#xx:32 → ERd32	—	(2)	↕	↕	↕
	SUB.L ERs, ERd	L		2								ERd32–ERs32 → ERd32	—	(2)	↕	↕	↕
SUBX	SUBX.B #xx:8, Rd	B	2									Rd8–#xx:8–C → Rd8	—	↕	↕	(3)	↕
	SUBX.B Rs, Rd	B		2								Rd8–Rs8–C → Rd8	—	↕	↕	(3)	↕
SUBS	SUBS.L #1, ERd	L		2								ERd32–1 → ERd32	—	—	—	—	—
	SUBS.L #2, ERd	L		2								ERd32–2 → ERd32	—	—	—	—	—
	SUBS.L #4, ERd	L		2								ERd32–4 → ERd32	—	—	—	—	—
DEC	DEC.B Rd	B		2								Rd8–1 → Rd8	—	—	↕	↕	↕
	DEC.W #1, Rd	W		2								Rd16–1 → Rd16	—	—	↕	↕	↕
	DEC.W #2, Rd	W		2								Rd16–2 → Rd16	—	—	↕	↕	↕

												(unsigned multiplication)						
	MULXU. W Rs, ERd	W		2								Rd16 × Rs16 → ERd32 (unsigned multiplication)	—	—	—	—	—	—
MULXS	MULXS. B Rs, Rd	B		4								Rd8 × Rs8 → Rd16 (signed multiplication)	—	—	↕	↕	—	—
	MULXS. W Rs, ERd	W		4								Rd16 × Rs16 → ERd32 (signed multiplication)	—	—	↕	↕	—	—
DIVXU	DIVXU. B Rs, Rd	B		2								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	(6)	(7)	—	—
	DIVXU. W Rs, ERd	W		2								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	(6)	(7)	—	—
DIVXS	DIVXS. B Rs, Rd	B		4								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	(8)	(7)	—	—
	DIVXS. W Rs, ERd	W		4								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	(8)	(7)	—	—
CMP	CMP.B #xx:8, Rd	B	2									Rd8–#xx:8	—	↕	↕	↕	↕	
	CMP.B Rs, Rd	B		2								Rd8–Rs8	—	↕	↕	↕	↕	
	CMP.W #xx:16, Rd	W	4									Rd16–#xx:16	—	(1)	↕	↕	↕	
	CMP.W Rs, Rd	W		2								Rd16–Rs16	—	(1)	↕	↕	↕	
	CMP.L #xx:32, ERd	L	6									ERd32–#xx:32	—	(2)	↕	↕	↕	
	CMP.L ERs, ERd	L		2								ERd32–ERs32	—	(2)	↕	↕	↕	

	EXTU.L ERd	L		2							0 → (<bits 31 to 16> of ERd32)	—	—	0	↕	0
EXTS	EXTS.W Rd	W		2							(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)	—	—	↕	↕	0
	EXTS.L ERd	L		2							(<bit 15> of ERd32) → (<bits 31 to 16> of ERd32)	—	—	↕	↕	0

	AND.W Rs, Rd	W		2								Rd16∧Rs16 → Rd16	—	—	↕	↕	0	-
	AND.L #xx:32, ERd	L	6									ERd32∧#xx:32 → ERd32	—	—	↕	↕	0	-
	AND.L ERs, ERd	L		4								ERd32∧ERs32 → ERd32	—	—	↕	↕	0	-
OR	OR.B #xx:8, Rd	B	2									Rd8∨#xx:8 → Rd8	—	—	↕	↕	0	-
	OR.B Rs, Rd	B		2								Rd8∨Rs8 → Rd8	—	—	↕	↕	0	-
	OR.W #xx:16, Rd	W	4									Rd16∨#xx:16 → Rd16	—	—	↕	↕	0	-
	OR.W Rs, Rd	W		2								Rd16∨Rs16 → Rd16	—	—	↕	↕	0	-
	OR.L #xx:32, ERd	L	6									ERd32∨#xx:32 → ERd32	—	—	↕	↕	0	-
	OR.L ERs, ERd	L		4								ERd32∨ERs32 → ERd32	—	—	↕	↕	0	-
XOR	XOR.B #xx:8, Rd	B	2									Rd8⊕#xx:8 → Rd8	—	—	↕	↕	0	-
	XOR.B Rs, Rd	B		2								Rd8⊕Rs8 → Rd8	—	—	↕	↕	0	-
	XOR.W #xx:16, Rd	W	4									Rd16⊕#xx:16 → Rd16	—	—	↕	↕	0	-
	XOR.W Rs, Rd	W		2								Rd16⊕Rs16 → Rd16	—	—	↕	↕	0	-
	XOR.L #xx:32, ERd	L	6									ERd32⊕#xx:32 → ERd32	—	—	↕	↕	0	-
	XOR.L ERs, ERd	L		4								ERd32⊕ERs32 → ERd32	—	—	↕	↕	0	-
NOT	NOT.B Rd	B		2								¬ Rd8 → Rd8	—	—	↕	↕	0	-
	NOT.W Rd	W		2								¬ Rd16 → Rd16	—	—	↕	↕	0	-
	NOT.L ERd	L		2								¬ Rd32 → Rd32	—	—	↕	↕	0	-

SHAR	SHAR.B Rd	B		2									—	—	↕	↕	0
	SHAR.W Rd	W		2									—	—	↕	↕	0
	SHAR.L ERd	L		2									—	—	↕	↕	0
SHLL	SHLL.B Rd	B		2									—	—	↕	↕	0
	SHLL.W Rd	W		2									—	—	↕	↕	0
	SHLL.L ERd	L		2									—	—	↕	↕	0
SHLR	SHLR.B Rd	B		2									—	—	↕	↕	0
	SHLR.W Rd	W		2									—	—	↕	↕	0
	SHLR.L ERd	L		2									—	—	↕	↕	0
ROTXL	ROTXL.B Rd	B		2									—	—	↕	↕	0
	ROTXL.W Rd	W		2									—	—	↕	↕	0
	ROTXL.L ERd	L		2									—	—	↕	↕	0
ROTXR	ROTXR.B Rd	B		2									—	—	↕	↕	0
	ROTXR.W Rd	W		2									—	—	↕	↕	0
	ROTXR.L ERd	L		2									—	—	↕	↕	0
ROTL	ROTL.B Rd	B		2									—	—	↕	↕	0
	ROTL.W Rd	W		2									—	—	↕	↕	0
	ROTL.L ERd	L		2									—	—	↕	↕	0
ROTR	ROTR.B Rd	B		2									—	—	↕	↕	0
	ROTR.W Rd	W		2									—	—	↕	↕	0
	ROTR.L ERd	L		2									—	—	↕	↕	0

	BSET Rn, Rd	B		2								(Rn8 of Rd8) ← 1	—	—	—	—	—	
	BSET Rn, @ERd	B			4							(Rn8 of @ERd) ← 1	—	—	—	—	—	
	BSET Rn, @aa:8	B						4				(Rn8 of @aa:8) ← 1	—	—	—	—	—	
BCLR	BCLR #xx:3, Rd	B		2								(#xx:3 of Rd8) ← 0	—	—	—	—	—	
	BCLR #xx:3, @ERd	B			4							(#xx:3 of @ERd) ← 0	—	—	—	—	—	
	BCLR #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) ← 0	—	—	—	—	—	
	BCLR Rn, Rd	B		2								(Rn8 of Rd8) ← 0	—	—	—	—	—	
	BCLR Rn, @ERd	B			4							(Rn8 of @ERd) ← 0	—	—	—	—	—	
	BCLR Rn, @aa:8	B						4				(Rn8 of @aa:8) ← 0	—	—	—	—	—	
BNOT	BNOT #xx:3, Rd	B		2								(#xx:3 of Rd8) ← ¬ (#xx:3 of Rd8)	—	—	—	—	—	
	BNOT #xx:3, @ERd	B			4							(#xx:3 of @ERd) ← ¬ (#xx:3 of @ERd)	—	—	—	—	—	
	BNOT #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) ← ¬ (#xx:3 of @aa:8)	—	—	—	—	—	
	BNOT Rn, Rd	B		2								(Rn8 of Rd8) ← ¬ (Rn8 of Rd8)	—	—	—	—	—	
	BNOT Rn, @ERd	B			4							(Rn8 of @ERd) ← ¬ (Rn8 of @ERd)	—	—	—	—	—	
	BNOT Rn, @aa:8	B						4				(Rn8 of @aa:8) ← ¬ (Rn8 of @aa:8)	—	—	—	—	—	
BTST	BTST #xx:3, Rd	B		2								¬ (#xx:3 of Rd8) → Z	—	—	—	↕	—	
	BTST #xx:3, @ERd	B			4							¬ (#xx:3 of @ERd) → Z	—	—	—	↕	—	
	BTST #xx:3, @aa:8	B						4				¬ (#xx:3 of @aa:8) → Z	—	—	—	↕	—	
	BTST Rn, Rd	B		2								¬ (Rn8 of @Rd8) → Z	—	—	—	↕	—	
	BTST Rn, @ERd	B			4							¬ (Rn8 of @ERd) → Z	—	—	—	↕	—	
	BTST Rn, @aa:8	B						4				¬ (Rn8 of @aa:8) → Z	—	—	—	↕	—	
BLD	BLD #xx:3, Rd	B		2								(#xx:3 of Rd8) → C	—	—	—	—	—	

BST	BST #xx:3, Rd	B		2								C → (#xx:3 of Rd8)	—	—	—	—	—
	BST #xx:3, @ERd	B			4							C → (#xx:3 of @ERd24)	—	—	—	—	—
	BST #xx:3, @aa:8	B						4				C → (#xx:3 of @aa:8)	—	—	—	—	—
BIST	BIST #xx:3, Rd	B		2								¬ C → (#xx:3 of Rd8)	—	—	—	—	—
	BIST #xx:3, @ERd	B			4							¬ C → (#xx:3 of @ERd24)	—	—	—	—	—
	BIST #xx:3, @aa:8	B						4				¬ C → (#xx:3 of @aa:8)	—	—	—	—	—
BAND	BAND #xx:3, Rd	B		2								C∧(#xx:3 of Rd8) → C	—	—	—	—	—
	BAND #xx:3, @ERd	B			4							C∧(#xx:3 of @ERd24) → C	—	—	—	—	—
	BAND #xx:3, @aa:8	B						4				C∧(#xx:3 of @aa:8) → C	—	—	—	—	—
BIAND	BIAND #xx:3, Rd	B		2								C∧ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—
	BIAND #xx:3, @ERd	B			4							C∧ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—
	BIAND #xx:3, @aa:8	B						4				C∧ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—
BOR	BOR #xx:3, Rd	B		2								C∨(#xx:3 of Rd8) → C	—	—	—	—	—
	BOR #xx:3, @ERd	B			4							C∨(#xx:3 of @ERd24) → C	—	—	—	—	—
	BOR #xx:3, @aa:8	B						4				C∨(#xx:3 of @aa:8) → C	—	—	—	—	—
BIOR	BIOR #xx:3, Rd	B		2								C∨ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—
	BIOR #xx:3, @ERd	B			4							C∨ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—
	BIOR #xx:3, @aa:8	B						4				C∨ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—
BXOR	BXOR #xx:3, Rd	B		2								C⊕(#xx:3 of Rd8) → C	—	—	—	—	—
	BXOR #xx:3, @ERd	B			4							C⊕(#xx:3 of @ERd24) → C	—	—	—	—	—
	BXOR #xx:3, @aa:8	B						4				C⊕(#xx:3 of @aa:8) → C	—	—	—	—	—
BIXOR	BIXOR #xx:3, Rd	B		2								C⊕ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—
	BIXOR #xx:3, @ERd	B			4							C⊕ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—
	BIXOR #xx:3, @aa:8	B						4				C⊕ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—

	BRN d:16 (BF d:16)	—							4							—	—	—	—	—	
	BHI d:8	—							2			C∨ Z = 0				—	—	—	—	—	
	BHI d:16	—							4							—	—	—	—	—	
	BLS d:8	—							2			C∨ Z = 1				—	—	—	—	—	
	BLS d:16	—							4							—	—	—	—	—	
	BCC d:8 (BHS d:8)	—							2			C = 0				—	—	—	—	—	
	BCC d:16 (BHS d:16)	—							4							—	—	—	—	—	
	BCS d:8 (BLO d:8)	—							2			C = 1				—	—	—	—	—	
	BCS d:16 (BLO d:16)	—							4							—	—	—	—	—	
	BNE d:8	—							2			Z = 0				—	—	—	—	—	
	BNE d:16	—							4							—	—	—	—	—	
	BEQ d:8	—							2			Z = 1				—	—	—	—	—	
	BEQ d:16	—							4							—	—	—	—	—	
	BVC d:8	—							2			V = 0				—	—	—	—	—	
	BVC d:16	—							4							—	—	—	—	—	
	BVS d:8	—							2			V = 1				—	—	—	—	—	
	BVS d:16	—							4							—	—	—	—	—	
	BPL d:8	—							2			N = 0				—	—	—	—	—	
	BPL d:16	—							4							—	—	—	—	—	
	BMI d:8	—							2			N = 1				—	—	—	—	—	
	BMI d:16	—							4							—	—	—	—	—	
	BGE d:8	—							2			N⊕V = 0				—	—	—	—	—	
	BGE d:16	—							4							—	—	—	—	—	
	BLT d:8	—							2			N⊕V = 1				—	—	—	—	—	
	BLT d:16	—							4							—	—	—	—	—	
	BGT d:8	—							2			Z∨ (N⊕V) = 0				—	—	—	—	—	
	BGT d:16	—							4							—	—	—	—	—	
	BLE d:8	—							2			Z∨ (N⊕V) = 1				—	—	—	—	—	
	BLE d:16	—							4							—	—	—	—	—	

	BSR d:16	—							4			PC → @–SP PC ← PC+d:16	—	—	—	—	—
JSR	JSR @ERn	—			2							PC → @–SP PC ← ERn	—	—	—	—	—
	JSR @aa:24	—						4				PC → @–SP PC ← aa:24	—	—	—	—	—
	JSR @@aa:8	—								2		PC → @–SP PC ← @aa:8	—	—	—	—	—
RTS	RTS	—									2	PC ← @SP+	—	—	—	—	—

											PC ← @SP+						
SLEEP	SLEEP	—									Transition to power-down state	—	—	—	—	—	
LDC	LDC #xx:8, CCR	B	2								#xx:8 → CCR	↕	↕	↕	↕	↕	
	LDC Rs, CCR	B		2							Rs8 → CCR	↕	↕	↕	↕	↕	
	LDC @ERs, CCR	W			4						@ERs → CCR	↕	↕	↕	↕	↕	
	LDC @(d:16, ERs), CCR	W				6					@(d:16, ERs) → CCR	↕	↕	↕	↕	↕	
	LDC @(d:24, ERs), CCR	W				10					@(d:24, ERs) → CCR	↕	↕	↕	↕	↕	
	LDC @ERs+, CCR	W					4				@ERs → CCR ERs32+2 → ERs32	↕	↕	↕	↕	↕	
	LDC @aa:16, CCR	W						6			@aa:16 → CCR	↕	↕	↕	↕	↕	
	LDC @aa:24, CCR	W						8			@aa:24 → CCR	↕	↕	↕	↕	↕	
STC	STC CCR, Rd	B		2							CCR → Rd8	—	—	—	—	—	
	STC CCR, @ERd	W			4						CCR → @ERd	—	—	—	—	—	
	STC CCR, @(d:16, ERd)	W				6					CCR → @(d:16, ERd)	—	—	—	—	—	
	STC CCR, @(d:24, ERd)	W				10					CCR → @(d:24, ERd)	—	—	—	—	—	
	STC CCR, @–ERd	W					4				ERd32–2 → ERd32 CCR → @ERd	—	—	—	—	—	
	STC CCR, @aa:16	W						6			CCR → @aa:16	—	—	—	—	—	
	STC CCR, @aa:24	W						8			CCR → @aa:24	—	—	—	—	—	
ANDC	ANDC #xx:8, CCR	B	2								CCR∧#xx:8 → CCR	↕	↕	↕	↕	↕	
ORC	ORC #xx:8, CCR	B	2								CCR∨#xx:8 → CCR	↕	↕	↕	↕	↕	
XORC	XORC #xx:8, CCR	B	2								CCR⊕#xx:8 → CCR	↕	↕	↕	↕	↕	
NOP	NOP	—								2	PC ← PC+2	—	—	—	—	—	

											R4L–1 → R4L until R4L=0 else next						
	EEPMOV. W	—								4	if R4 ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4–1 → R4 until R4=0 else next	—	—	—	—	—	

Notes: 1. The number of states in cases where the instruction code and its operands ar in on-chip memory is shown here. For other cases, see appendix A.3, Numbe 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 prev
   (5) The number of states required for execution of an instruction that transfer 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.

Instruction code:

1st byte		2nd byte	
AH	AL	BH	BL

Instruction when most significant bit of BH is 0.
Instruction when most significant bit of BH is 1.

AH \ AL	0	1	2	3	4	5	6	7	8	9	A	B	C
0	NOP	Table A-2 (2)	STC	LDC	ORC	XORC	ANDC	LDC	ADD		Table A-2 (2)	Table A-2 (2)	
1	Table A-2 (2)	Table A-2 (2)	Table A-2 (2)	Table A-2 (2)	OR.B	XOR.B	AND.B	Table A-2 (2)	SUB		Table A-2 (2)	Table A-2 (2)	
2	MOV.B												
3													
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE
5	MULXU	DIVXU	MULXU	DIVXU	RTS	BSR	RTE	TRAPA	Table A-2 (2)	JMP			BSR
6	BSET	BNOT	BCLR	BTST	OR	XOR	AND	BST / BIST	MOV				
7					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD	MOV	Table A-2 (2)	Table A-2 (2)	EEPMOV	
8	ADD												
9	ADDX												
A	CMP												
B	SUBX												
C	OR												
D	XOR												
E	AND												
F	MOV												

Instruction code:

1st byte		2nd byte	
AH	AL	BH	BL

AH AL \ BH	0	1	2	3	4	5	6	7	8	9	A	B
01	MOV				LDC/STC				SLEEP			
0A	INC											
0B	ADDS					INC		INC	ADDS			
0F	DAA											
10	SHLL			SHLL					SHAL			SH
11	SHLR			SHLR					SHAR			SH
12	ROTXL			ROTXL					ROTL			RO
13	ROTXR			ROTXR					ROTR			RO
17	NOT			NOT		EXTU		EXTU	NEG			NE
1A	DEC											
1B	SUBS					DEC		DEC	SUBS			
1F	DAS											
58	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BM
79	MOV	ADD	CMP	SUB	OR	XOR	AND					
7A	MOV	ADD	CMP	SUB	OR	XOR	AND					

Instruction code:

1st byte		2nd byte		3rd byte		4th byte	
AH	AL	BH	BL	CH	CL	DH	DL

Instruction when m
Instruction when m

AH / ALBH / BLCH \ CL	0	1	2	3	4	5	6	7	8	9	A	B
01406										LDC / STC		LDC / STC
01C05	MULXS		MULXS									
01D05		DIVXS		DIVXS								
01F06					OR	XOR	AND					
7Cr06 *1				BTST								
7Cr07 *1				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD				
7Dr06 *1	BSET	BNOT	BCLR					BST / BIST				
7Dr07 *1	BSET	BNOT	BCLR									
7Eaa6 *2				BTST								
7Eaa7 *2				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD				
7Faa6 *2	BSET	BNOT	BCLR					BST / BIST				
7Faa7 *2	BSET	BNOT	BCLR									

Notes: 1. r is the register designation field.
2. aa is the absolute address field.

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 RO
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 = $2 \times 2 + 1 \times 2 + 1 \times 2 = 8$

Note: * Depends on which on-chip peripheral module is accessed. See section 20.1, F
Addresses (Address Order).

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		

	BCS d:16(BLO d:16)	2		
	BNE d:16	2		
	BEQ d:16	2		
	BVC d:16	2		
	BVS d:16	2		
	BPL d:16	2		
	BMI d:16	2		
	BGE d:16	2		
	BLT d:16	2		
	BGT d:16	2		
	BLE d:16	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

	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	
BST	BST #xx:3, Rd	1		
	BST #xx:3, @ERd	2		2
	BST #xx:3, @aa:8	2		2

	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		
	DIVXS.W Rs, ERd	2		
DIVXU	DIVXU.B Rs, Rd	1		
	DIVXU.W Rs, ERd	1		
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		

	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
	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	
	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	
	MOV.B Rs, @aa:8	1				1	

	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
	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
	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
	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	

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
	POP.L ERn	2		2
PUSH	PUSH.W Rn	1		1
	PUSH.L ERn	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		

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
	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		

	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 a
n+1 times respectively.
2. It cannot be used in this LSI.

instructions	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		—	—	—	—	—	—	—	—	—	—	—

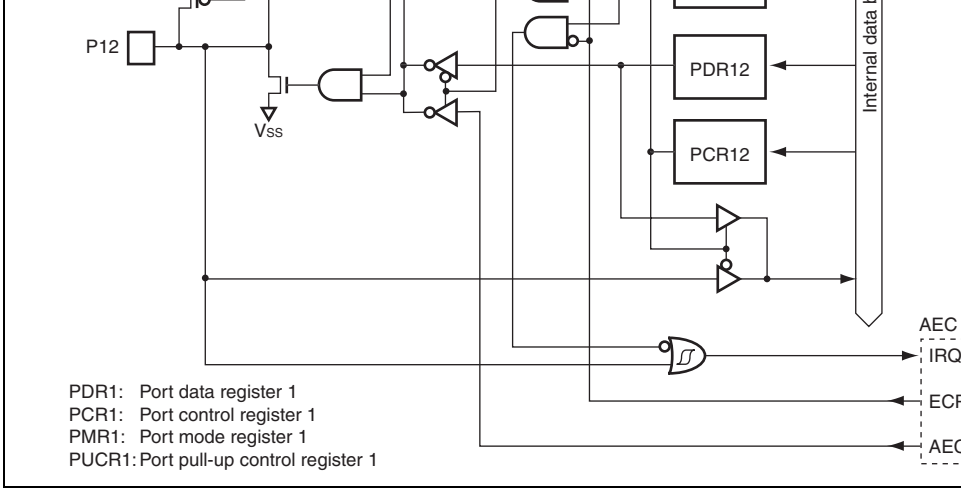
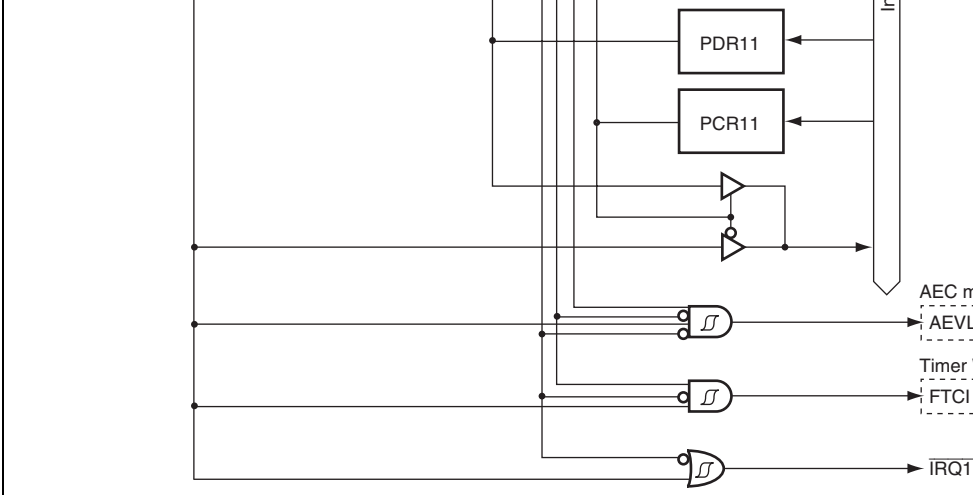


Figure B.1 (a) Port 1 Block Diagram (P12)



PDR1: Port data register 1
PCR1: Port control register 1
PMR1: Port mode register 1
PUCR1: Port pull-up control register 1
PFCR : Pin function control register

Figure B.1 (b) Port 1 Block Diagram (P11)

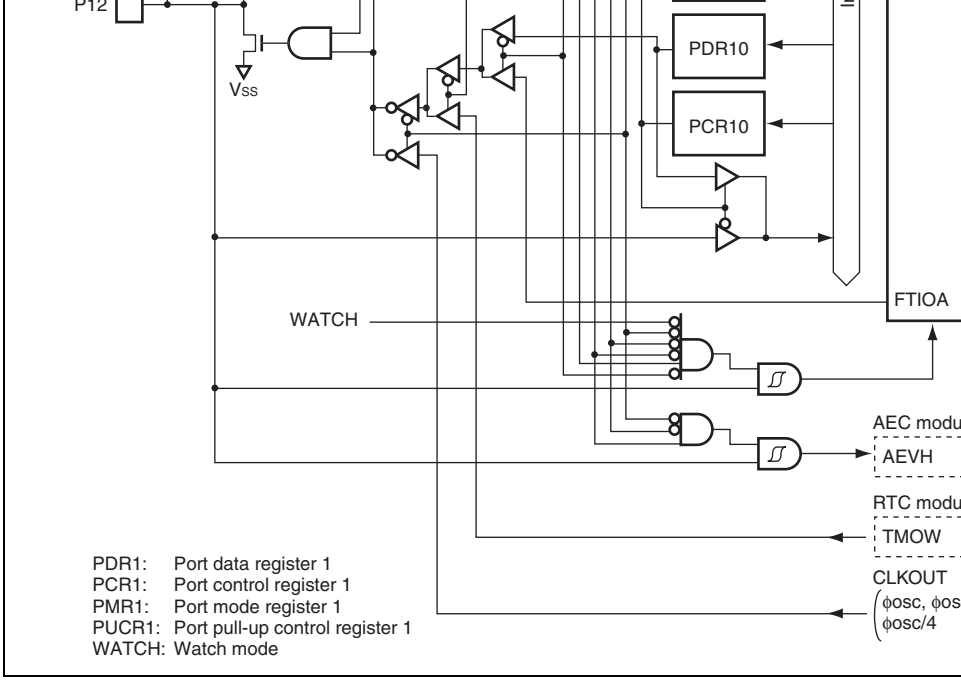
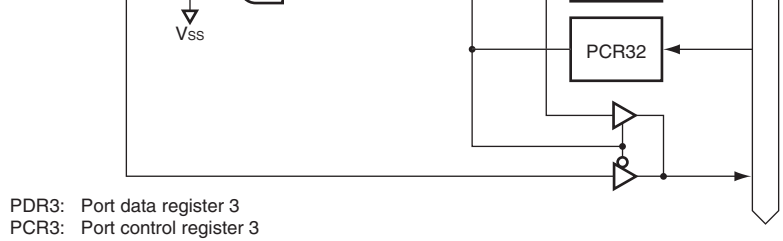


Figure B.1 (c) Port 1 Block Diagram (P10)



PDR3: Port data register 3
PCR3: Port control register 3
PUCR3: Port pull-up control register 3
SPCR: Serial port control register

Figure B.2 (a) Port 3 Block Diagram (P32)

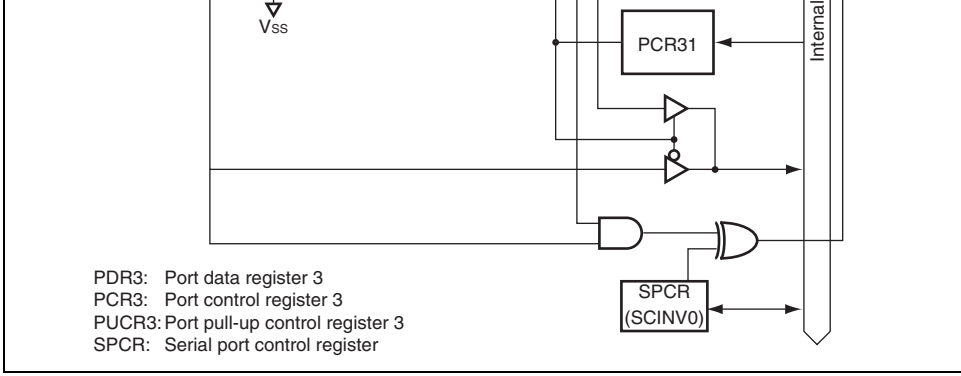


Figure B.2 (b) Port 3 Block Diagram (P31)

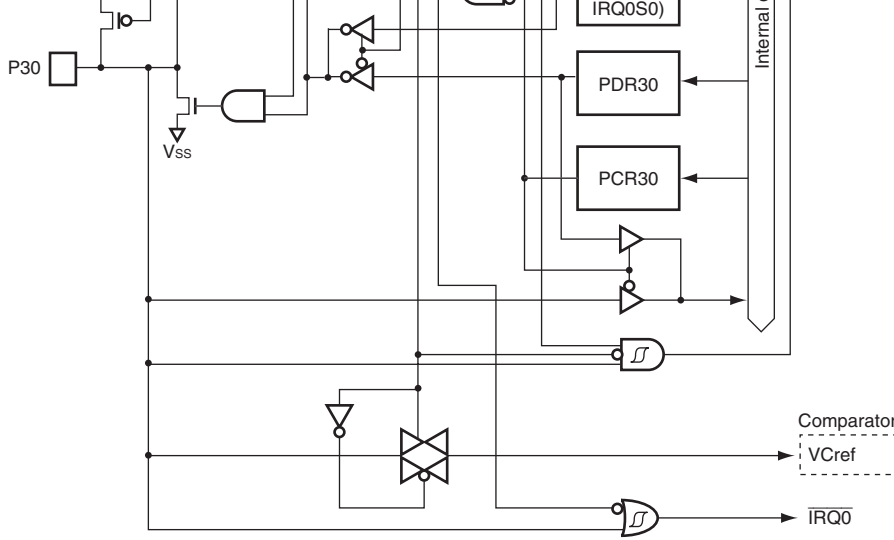
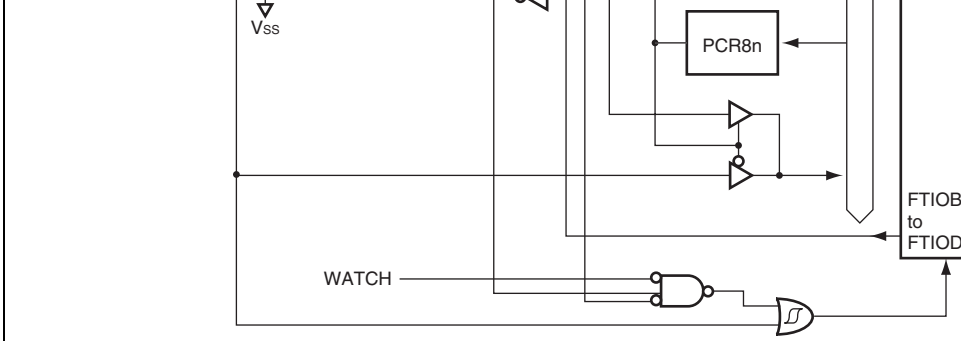
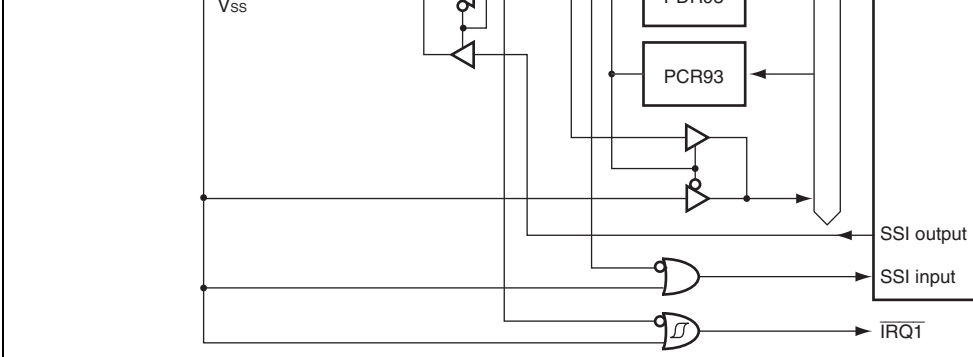


Figure B.2 (c) Port 3 Block Diagram (P30)



PDR8: Port data register 8
PCR8: Port control register 8
PUCR8: Port pull-up control register 8
WATCH: Watch mode
n = 4 to 2

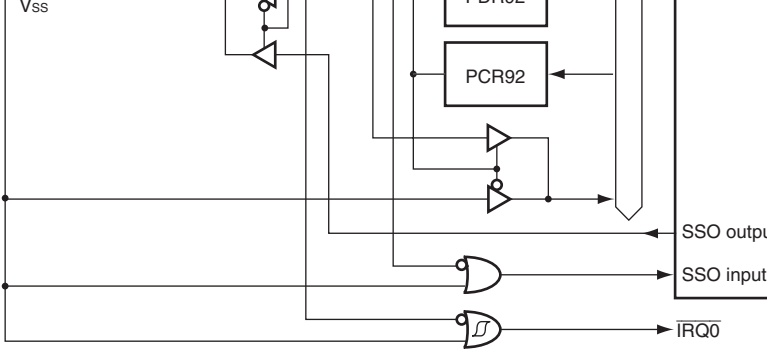
Figure B.3 (a) Port 8 Block Diagram (P84 to P82)



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

Note: When the SSUS bit in PFCR is 1, the SS
switched to the $\overline{SCS}$ pin.

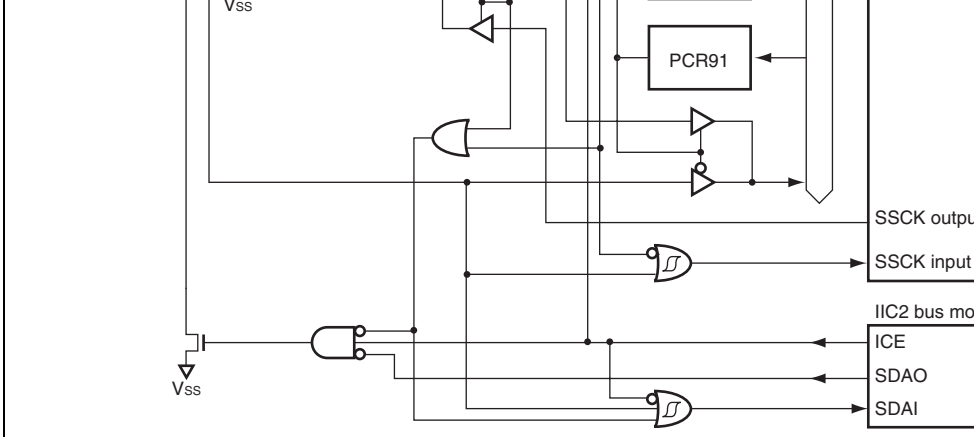
Figure B.4 (a) Port 9 Block Diagram (P93)



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

Note: When the SSUS bit in PFCR is 1, the S
switched to the SSCK pin.

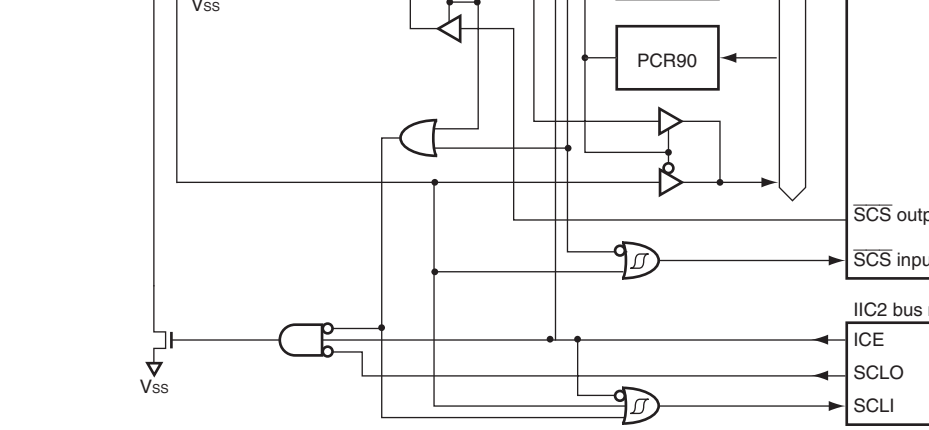
Figure B.4 (b) Port 9 Block Diagram (P92)



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

Note: Priority: SSU pin > IIC2 pin > P91
When the SSUS bit in PFCR is 1, the SSC
switched to the SSO pin.

Figure B.4 (c) Port 9 Block Diagram (P91)



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

Note: Priority: SSU pin > IIC2 pin > P90
When the SSUS bit in PFCR is 1, the S
switched to the SSI pin.

Figure B.4 (d) Port 9 Block Diagram (P90)

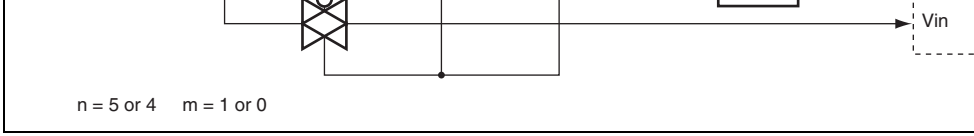


Figure B.5 (a) Port B Block Diagram (PB5 or PB4)

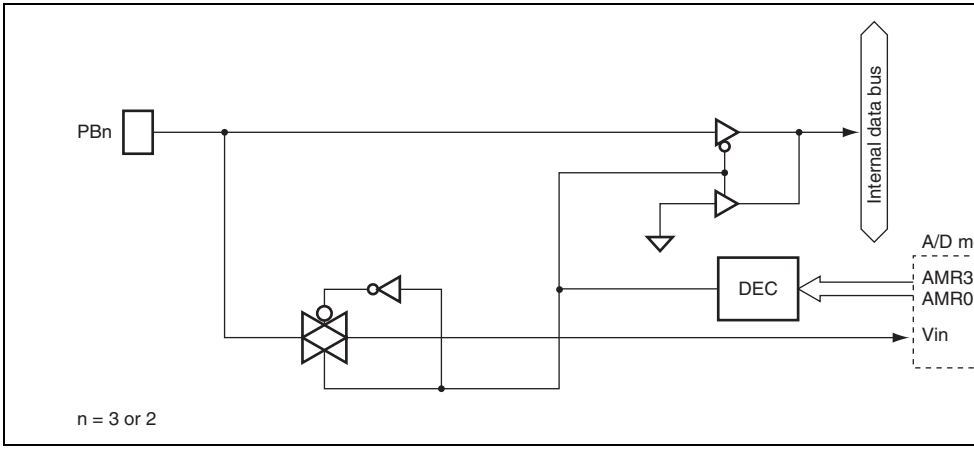
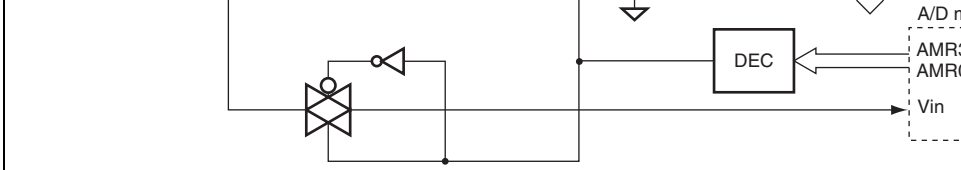


Figure B.5 (b) Port B Block Diagram (PB3 or PB2)



PMRB: Port mode register B
PFCR: Pin function control register

n = 1 or 0

Figure B.5 (c) Port B Block Diagram (PB1 or PB0)

	impedance			impedance*[1]*[2]			
P93 to P90	High impedance	Retained	Retained	High impedance*[1]*[2]	Functions	Functions	F
PB5 to PB0	High impedance	High impedance	High impedance	High impedance*[1]	High impedance	High impedance	H i

Notes: 1. Registers are retained and output level is high impedance.
2. High-level output when the pull-up MOS is turned on.

						1	Other then 01	Other then 01	P93 I/O	SSCK input	P91 I
							01	Other then 01	IRQ1N input	SSCK input	P91 I
			1 (Transmit)	0	0 (IIC2 not used)	0	Other then 01	Other then 01	P93 I/O	SSO output	SSCK input
							01	Other then 01	IRQ1N input	SSO output	SSCK input
						1	Other then 01	Other then 01	P93 I/O	SSCK input	SSO outpu
							01	Other then 01	IRQ1N input	SSCK input	SSO outpu
			1 (Transmit)	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	SSI input	SSO output	SSCK input
						1	Other then 01	Other then 01	P93 I/O	SSCK input	SSO outpu
							01	Other then 01	IRQ1N input	SSCK input	SSO outpu

			(Transmit)		(IIC2 not used)		01	then 01		output	output
							01	Other then 01	IRQ1N input	SSO output	SSCK output
						1	Other then 01	Other then 01	P93 I/O	SSCK output	SSO output
							01	Other then 01	IRQ1N input	SSCK output	SSO output
			1 (Transmit)	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	SSI input	SSO output	SSCK output
						1	Other then 01	Other then 01	P93 I/O	SSCK output	SSO output
							01	Other then 01	IRQ1N input	SSCK output	SSO output

							then 01		output	input	in
						1	Other then 01	Other then 01	SCS input	SSCK input	P9
			1 (Transmit)	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	SSI output	SSO input	SS in
						1	Other then 01	Other then 01	SCS input	SSCK input	SS in
		1 (Master)	0	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	SSI input	P92 I/O	SS ou
							Other then 01	01	SSI input	IRQ0N input	SS ou
						1	Other then 01	Other then 01	SCS output	SSCK output	P9
			1 (Transmit)	0	0 (IIC2 not used)	0	Other then 01	Other then 01	P93 I/O	SSO output	SS ou
							01	Other then 01	IRQ1N input	SSO output	SS ou
						1	Other then 01	Other then 01	SCS output	SSCK output	SS ou
			1 (Transmit)	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	SSI input	SSO output	SS ou
						1	Other then 01	Other then 01	SCS output	SSCK output	SS ou

								then 01	input	output	inpu
						1	Other then 01	Other then 01	SCS input	SSCK input	SSO outp
		1 (Master)	0	1 (Receive)	0 (IIC2 not used)	0	Other then 01	Other then 01	P93 I/O	SSO input	SSO outp
							01	Other then 01	IRQ1N input	SSO input	SSO outp
						1	Other then 01	Other then 01	SCS output	SSCK output	SSO inpu
			1 (Transmit)	0	0 (IIC2 not used)	0	Other then 01	Other then 01	P93 I/O	SSO output	SSO outp
							01	Other then 01	IRQ1N input	SSO output	SSO outp
						1	Other then 01	Other then 01	SCS output	SSCK output	SSO outp
0 (SSU not used)	0	0	0	0	1 (IIC2 used)	*	Other then 01	Other then 01	P93 I/O	P92 I/O	SDA
							Other then 01	01	P93 I/O	IRQ0N input	SDA
							01	Other then 01	IRQ1N input	P92 I/O	SDA
							01	01	IRQ1N input	IRQ0N input	SDA
					0 (IIC2 not used)	*	Other then 01	Other then 01	P93 I/O	P92 I/O	P91
							Other then 01	01	P93 I/O	IRQ0N input	P91
							01	Other then 01	IRQ1N input	P92 I/O	P91
							01	01	IRQ1N input	IRQ0N input	P91

[Legend] *: Don't care.

		HD64338602RFH	38602R(***)	3
H8/38600R	Masked ROM version	HD64338600RFT	38600R(***)	3
		HD64338600RFH	38600R(***)	3

[Legend]
(***): ROM code

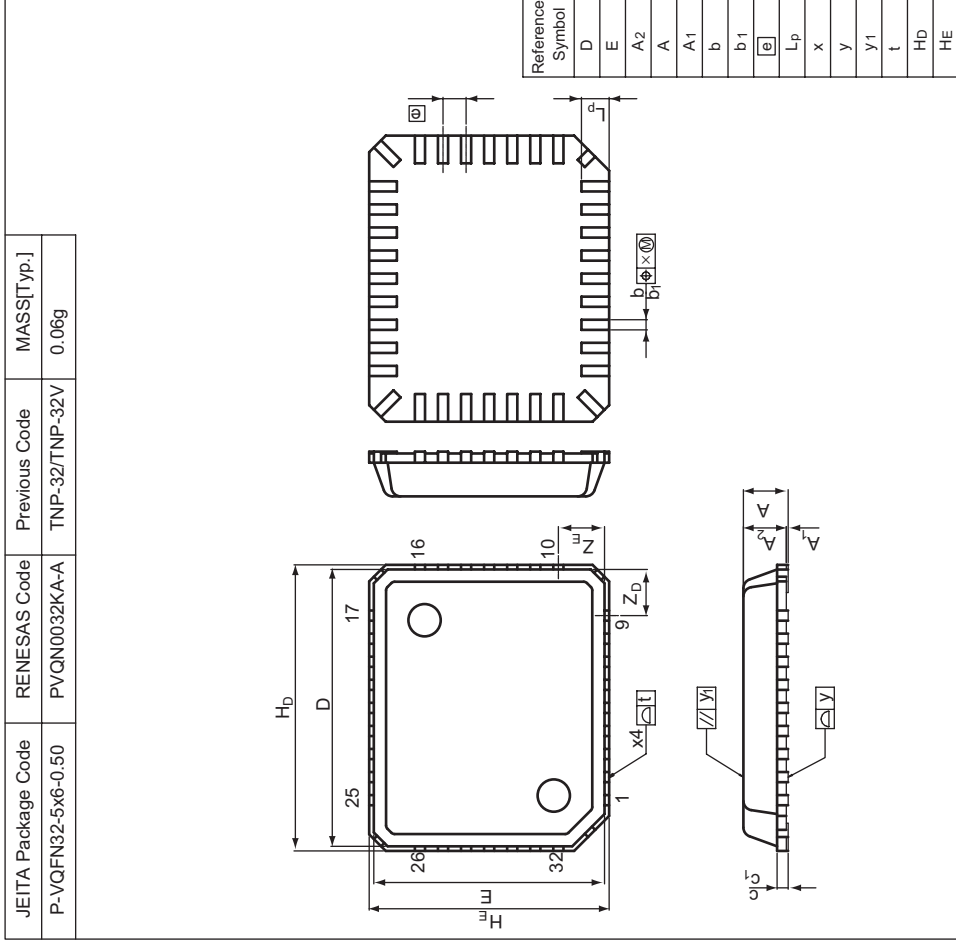


Figure D.1 Package Dimensions (TNP-32)

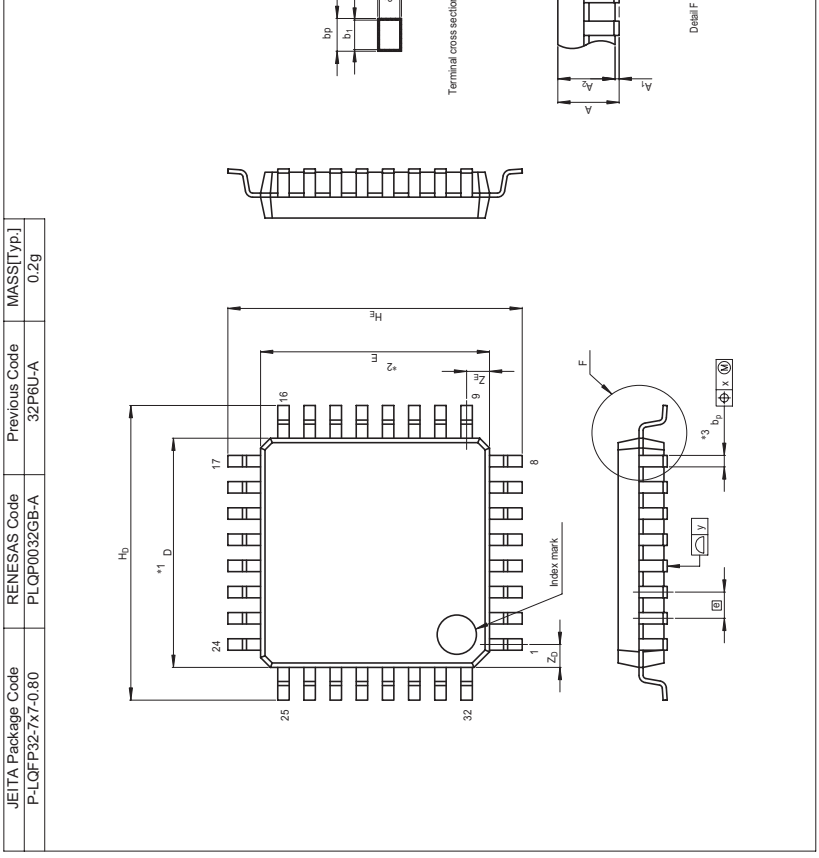
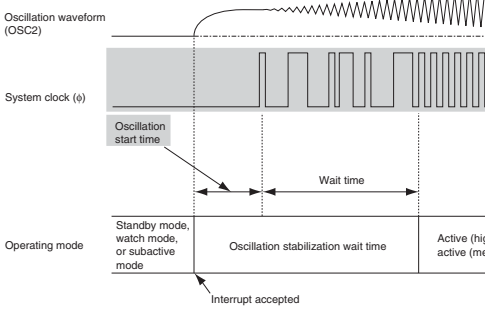


Figure D.2 Package Dimensions (32P6U-A)

		indicated by R6. Set R4, R4L, and R6 so that address of the destination address (value of R or R6 + R4) does not exceed H'FFFF (the val must not change from H'FFFF to H'0000 durin 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. T 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 nev indicate an odd address. To save register valu PUSH.W Rn (MOV.W Rn, @–SP) or PUSH.L (MOV.L ERn, @–SP). To restore register valu POP.W Rn (MOV.W @SP+, Rn) or POP.L ER (MOV.L @SP+, ERn).

		Notes: 1. …… 2. When the on-chip debugger is connected, th of the resistor should be high. When not con it is specified according to the selected oscil
Figure 4.5 Typical Connection to 32.768-kHz/38.4-kHz Crystal Resonator	68	Modified <table><tr><td>Frequency</td><td>Manufacturer</td><td>Products Name</td><td>Equivalent Se</td></tr><tr><td>38.4 kHz</td><td>EPSON TOYOCOM CORPORATION</td><td>C-4-TYPE</td><td>30 kΩ (max.)</td></tr><tr><td>32.768 kHz</td><td>EPSON TOYOCOM CORPORATION</td><td>C-001R</td><td>35 kΩ (max.)</td></tr><tr><td colspan="4">$C_1 = C_2$ = 7 pF (typ.)</td></tr></table> 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 a used, perform matching evaluation with the resonator manufacture and connect it under optimum condition. Even when the resonato above or the equivalent is used, as the oscil characteristics depend on the board specific perform matching evaluation on the mountin 2. Perform matching evaluation in the reset sta RES pin is low) and on exit from the reset st RES pin is driven from low to high).
4.4.1 Prescaler S	71	Deleted The output from prescaler S is shared by the o peripheral modules. ~~The division ratio can be c separately for each on-chip peripheral function~~



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 clear time by the capacitor capacitance, which externally connected to the $\overline{\text{RES}}$ pin. The exte capacitor capacitance should be adjusted to s oscillation stabilization time before reset clear details, refer to section 19, Power-On Reset C
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 operatin~~ WDON bit is cleared to 0 by software, this bit and the watchdog timer enters module standb
Table 5.3 Internal State in Each Operating Mode	86	The note is modified. Notes: 6. Functions if the 32.768-kHz RTC is selecte internal clock. Halted and retained otherwise.

		oscillator output stabilizes and the t_{RFI} period ha elapsed. The CPU starts reset exception handl when the $\overline{RES}$ pin is driven high.
5.2.3 Watch Mode	88	Modified … or the requested interrupt is disabled by the enable register. When a reset source is generated in watch mo system clock oscillator starts. If a reset is gene the $\overline{RES}$ pin, it must be kept low until the syste oscillator output stabilizes. The CPU starts res exception handling when the $\overline{RES}$ pin is driven
5.2.4 Subsleep Mode	89	Modified … or the requested interrupt is disabled by the enable register. When a reset source is generated in subsleep the system clock oscillator starts. If a reset is g by the $\overline{RES}$ pin, it must be kept low until the sy clock oscillator output stabilizes. The CPU star exception handling when the $\overline{RES}$ pin is driven
5.2.5 Subactive Mode	89	Modified … on the combination of bits SSBY, LSON, an in SYSCR1 and bits MSON and DTON in SYS ~~Subactive mode is not cleared if the I bit in CC~~ ~~to 1 or the requested interrupt is disabled by th~~ ~~interrupt enable register.~~ When a reset source is generated in subactive the system clock oscillator starts. If a reset is g by the $\overline{RES}$ pin, it must be kept low until the sy clock oscillator output stabilizes and the t_{RFI} pe elapsed. The CPU starts reset exception handl when the $\overline{RES}$ pin is driven high.

		made depending on the combination of bits S TMA3 in SYSCR1, or a transition to sleep mo made depending on the combination of bits S LSON in SYSCR1. Moreover, a transition to a (high-speed) mode or subactive mode is mad direct transition. ~~Active (medium-sleep) mode~~ ~~entered if the I bit in CCR is set to 1 or the req~~ ~~interrupt is disabled in the interrupt enable reg~~ When the $\overline{\text{RES}}$ pin goes low, the CPU goes in reset state and active (medium-sleep) mode i ~~In active (medium-speed) mode, the on-chip p~~ ~~modules function at the clock set by the MA1~~ ~~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 acti speed) mode while the SSBY and LSON bits SYSCR1 are cleared to 0 and the MSON and bits in SYSCR2 are set to 1, a transition is ma active (medium-speed) mode via sleep mode The time from the start of SLEEP instruction e to the end of interrupt exception handling (the transition time) is calculated by equation (1). Example: When ϕosc/8 is selected as the C operating clock after the transition Direct transition time = $(2 + 1) \times 1$tosc + 14 115tosc For the legend of symbols used above, refe section 21, Electrical Characteristics.

		operating clock after the transition Direct transition time = (2 + 1) × 1tosc + 14 × 3tosc + 112tw For the legend of symbols used above, refer 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 activ (medium-speed) mode while the SSBY and LS in SYSCR1 are cleared to 0, the MSON bit in S is cleared to 0, and the DTON bit in SYSCR2 is 1, a transition is made to active (high-speed) m sleep mode. The time from the start of SLEEP instruction ex to the end of interrupt exception handling (the d transition time) is calculated by equation (3). Example: When ϕosc/8 is selected as the CP operating clock before the transition Direct transition time = (2 + 1) × 8tosc + 14 × 38tosc For the legend of symbols used above, refer section 21, Electrical Characteristics.

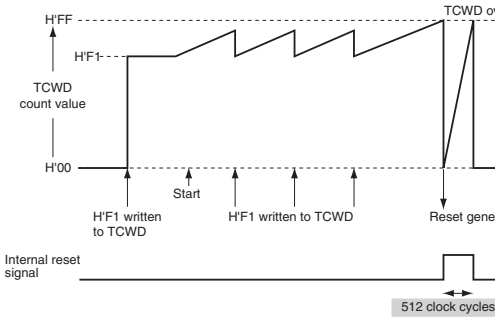
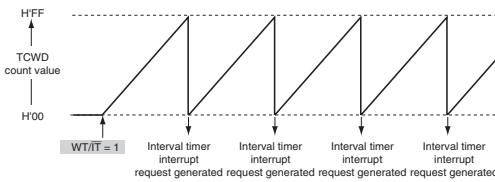
		CPU operating clock before and after the tr respectively Direct transition time = (2 + 1) × 8tosc + 14 24tosc + 112tw For the legend of symbols used above, refe section 21, Electrical Characteristics.
5.3.5 Direct Transition from Subactive Mode to Active (High-Speed) Mode	93	Added and modified When a SLEEP instruction is executed in sub mode while the SSBY and TMA3 bits in SYSC set to 1, the LSON bit in SYSCR1 is cleared t MSON bit in SYSCR2 is cleared to 0, and the in SYSCR2 is set to 1, a transition is made di active (high-speed) mode via watch mode aft waiting time set in bits STS2 to STS0 in SYSC elapsed. The time from the start of SLEEP instruction e to the end of interrupt exception handling (the transition time) is calculated by equation (5). Direct transition time = {(Number of SLEEP in execution states) + (Number of internal proce states)} × (tsubcyc before transition) + (Wait t bits STS2 to STS0) + (Number of interrupt ex handling execution states) × (tcyc after transit Example: When ϕw/8 is selected as the CP operating clock after the transition and wait 8192 states Direct transition time = (2 + 1) × 8tw + (8192 ×1tosc = 24tw + 8206tosc For the legend of symbols used above, refe section 21, Electrical Characteristics.

		transition time) is calculated by equation (3). Example: When ϕw/8 and ϕosc/8 are selecte CPU operating clock before and after the tra respectively, and wait time = 8192 states Direct transition time = (2 + 1) × 8tw + 8192 × 14 × 8tosc = 24tw + 8304tosc For the legend of symbols used above, refer 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 ent module standby mode. If an interrupt is generated in module standby r the vector address cannot be fetched. As a res 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}/ CLKOUT output must be performed wh CLKOUT output is halted (CLKOUT = 0 When making a transition to a power-d mode wherein the system clock oscillat halted, the output level is retained. (In s mode, output is the high-impedance sta When making a transition from a powe mode wherein the system clock oscillat halted, to the active mode wherein the clock oscillator operates, halt CLKOUT (CLKOUT = 0) before the transition.
8.7.2 Input Characteristics Difference due to Pin Function	124	This section is newly added.

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	0000: ϕ/8 ····· Free running counter
1	RCS1	0001: ϕ/32 ····· Free running counter
0	RCS0	0010: ϕ/128 ····· Free running counter
		0011: ϕ/256 ····· Free running counter
		0100: ϕ/512 ····· Free running counter
		0101: ϕ/2048 ····· Free running counter
		0110: ϕ/4096 ····· Free running counter
		0111: ϕ/8192 ····· Free running counter
		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, day-of-week data, control registers, and interr registers are not reset by a RES input, or by a source caused by a watchdog timer.
11.5 Interrupt Sources	198	Modified … When using an interrupt, set the IENRTC (interrupt request enable) bit in IENR1 to 1 las other registers are set.
11.6.2 Note when Using RTC Interrupts	199	Added

12.3.1 Watchdog Timer Mode	208	Modified … When a clock pulse is input after the TCWD value has reached H'FF, the watchdog timer ov and an internal reset signal is generated. The i reset signal is output for a period of 512 clock c 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

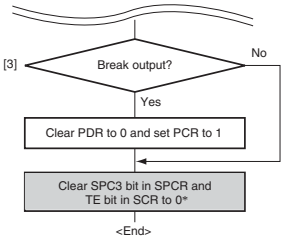
		1	ECH1
		0	ECH0

13.3.7 Event Counter L (ECL)	222	Modified		
		Bit	**Bit Name**	**Description**
		7	ECL7	Either the external asynchronous event
		6	ECL6	$\phi/2$, $\phi/4$, or $\phi/8$ can be selected as the i
		5	ECL5	source. ECL can be cleared to H'00 wh
		4	ECL4	CRCL bit in ECCSR is cleared to 0.
		3	ECL3	
		2	ECL2	
		1	ECL1	
		0	ECL0	

Section 14 Serial Communication Interface 3 (SCI3, IrDA)	231	Deleted The serial communication interface 3 (SCI3) c both asynchronous and clock synchronous se communication. In the asynchronous method, data communication can be carried out using asynchronous communication chips such as a Asynchronous Receiver/Transmitter (UART) o Asynchronous Communication Interface Adap (ACIA). ~~A function is also provided for serial communication between processors (multipro communication function).~~

14.3.5 Serial Mode Register (SMR)	235	Modified		
		Bit	**Bit Name**	**Description**
		2	MP	5-Bit CommunicationWhen this bit is se 5-bit communication format is enabled. to set bit 5 (PF) to 1 when setting this b

		Bit **Bit Name** **Description**
		7 IrE IrDA EnableSelects whether the SCI3 I/O function as the SCI3 or IrDA.0: TXD3/IrT RXD3/IrRXD pins function as TXD3 and TXD3/IrTXD and RXD3/IrRXD pins funct IrTXD and IrRXD
14.3.11 Serial Extended Mode Register (SEMR)	253	Added **Bit** **Bit Name** **Description** 3 ABCS Asynchronous Mode Basic Clock Select Selects the basic clock for the bit period asynchronous mode. This setting is enabled only in asynchron (COM bit in SMR3 is 0). 0: Operates on a basic clock with a frequ 16 times the transfer rate 1: Operates on a basic clock with a frequ eight times the transfer rate Clear the ABCS bit to 0, when the IrDA f enabled.
Table 14.8 Data Transfer Formats (Asynchronous Mode)	255	The formats are modified.
Table 14.9 SMR Settings and Corresponding Data Transfer Formats	256	The settings are modified.

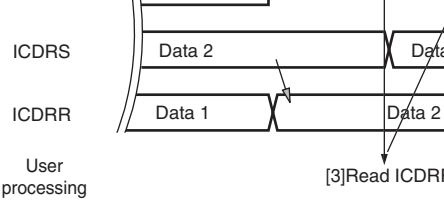
Figure 14.6 Sample Serial Transmission Flowchart (Asynchronous Mode)	260	Modified  [3] To output a break in serial t after setting PCR to 1 and P clear the SPC3 bit in SPCR bit in SCR to 0. Note: * When the SPC3 bit in SI the pin functions as an I/
14.5 Operation in Clock Synchronous Mode	264	Deleted … After 8-bit data is output, the transmission the MSB state. In clock synchronous mode, n ~~multiprocessor~~ bit is added.
14.6 Multiprocessor Communication Function	270	This section is deleted.

TXD3 pin. To send a break during data transm
first set PCR to 1 and PDR to 0, and then clea
SPC3 and TE bits to 0. When the TE bit is clea
directly after the SPC3 bit is cleared to 0, the t
is initialized regardless of the current transmiss
after the TE bit is cleared, the TXD3 pin functio
I/O port after the SPC3 bit is cleared, and 0 is
from the TXD3 pin.

Section 16 I2C Bus Interface 2 (IIC2)	323	Modified

16.3.5 I^2C Bus Status Register (ICSR)

Bit	Bit Name	Description
3	STOP	Stop Condition Detection Flag[Set conditions] • In master mode, when a stop is detected after the completi frame transfer • In slave mode, when a stop c detected, after the slave addr first byte, following the genera the detection of the start cond matches the address set in S



Section 17 A/D Converter	350	Modified
17.3.1 A/D Result Register (ADRR)		ADRR is a 16-bit read-only register that stores results of A/D conversion. The data is stored in upper 10 bits of ADRR. ADRR can be read by at any time, …
17.7.3 Usage Notes	360	Deleted 3. When A/D conversion is started after clearing standby mode, wait for 10ϕ clock cycles before starting A/D conversion. ~~4. When the LADS bit in ADSR is changed as halting to operating, wait for 10ϕ clock cycles starting A/D conversion.~~

		needs to be reduced, stop the comparator b the CME0 and CME1 bits in CMCR0 and CM 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 circ shown in figure 19.2. As the power supply volta the capacitor, which is externally connected to pin, is gradually charged through the on-chip p resistor (Rp).
Section 21 Electrical Characteristics Table 21.3 Control Signal Timing	411	Modified

Item	Symbol	Applicable Pins	Test Condition	Values Min.	
Oscillation stabilization time	trc	OSC1,OSC2	Ceramic resonator (V_{cc} = 2.2 V to 3.6 V)	—	2
			Ceramic resonator (Other than above)	—	8
			Crystal resonator (V_{cc} = 2.7 V to 3.6 V)	—	3
			Crystal resonator (V_{cc} = 2.2 V to 3.6 V)	—	

		2.7 V to 3.6 V) Crystal resonator(VCC = — 2.2 V to 3.6 V)

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	Cond I	H
DAA		B	2	2							Rd8 decimal adjust → Rd8	—	*

C. Product Part No. Lineup	491	The list is modified.
D. Package Dimensions Figure D.2 Package Dimensions (32P6U-A)	492	Added

C

E

F

G

I

S

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 SALES OFFICES

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,
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8/38602R Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0152-0300

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 16-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

MB90F036APMC-GSE1 MB90F342CASPMC-GSE1 MB90F345CESPMC-GE1 MB90F349CAPFR-GSE1 MB90F428GCPFR-GSE1 MB90F462APFM-GE1 MB90F462APMC-G-SNE1 MB90F497GPF-GE1 MB90F546GSPFR-GE1 MB90F947APFR-GS-SPE1 MB96F346RSBPMC-GS-N2E2 MB96F683RBPMC-GSAE1 R5F11BGEAFB#30 DF3026XBL25V S912ZVFP64F1VLL R4F24268NVRFQV R5F107DEGSP#X0 R5F11B7EANA#U0 R5F21172DSP#U0 M30622F8PGP#U3C MB90092PF-G-BNDE1 MB90F335APMC1-G-SPE1 MB90F342CASPFR-GS-N2E1 MB90F345CAPFR-GSE1 MB90F543GPF-GE1 MB90F546GSPF-GE1 MB90F568PMCR-GE1 MB90F594APFR-GE1 MB90F882ASPMC-GE1 MB96F346RSAPQCR-GS-N2E2 MB96F387RSBPMC-GSE2 MB96F387RSBPMC-GS-N2E2 MB96F395RSAPMC-GSE2 MB96F623RBPMC1-GSE1 MB96F646RBPMC-GSE1 XE167F96F66LACFXUMA1 MB96F696RBPMC-GSAE1 MB96F018RBPMC-GSE1 MB90F962SPMCR-GE1 MB90F867ASPFR-GE1 MB90F543GPF-G-FLE1 MB90F345CESPF-GE1 M30290FCHP#U3A DF2239FA20IV HD64F3672FPV R5F104AEASP#V0 R5F100BCANA#U0 R5F100BFANA#U0 S9S12H256J2VFVER R5F100ACASP#V0