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

H8/3694 Group

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer
H8 Family/H8/300H Tiny Series

H8/3694N	HD64N3694G, HD6483694G,
H8/3694F	HD64F3694, HD64F3694G,
H8/3694	HD6433694, HD6433694G,
H8/3693	HD6433693, HD6433693G,
H8/3692	HD6433692, HD6433692G,
H8/3691	HD6433691, HD6433691G,
H8/3690	HD6433690, HD6433690G

Renesas Electronics
www.renesas.com

Rev.5.00 2005.11

a third party.

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

Generally, the input pins of CMOS products are high-impedance input pins. If un
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 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 H8/3694 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 characteristi
- 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: Bit order: The MSB is on the left and the LSB is on the right.

Notes:

When using the on-chip emulator (E7, E8) for H8/3694 program development and debug following restrictions must be noted.

1. The $\overline{\text{NMI}}$ pin is reserved for the E7 or E8, and cannot be used.
2. Pins P85, P86, and P87 cannot be used. In order to use these pins, additional hardware provided on the user board.
3. Area H'7000 to H'7FFF is used by the E7 or E8, and is not available to the user.
4. Area H'F780 to H'FB7F must on no account be accessed.

Document Title	Docume
H8/3694 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

Application notes:

Document Title	Docume
H8S, H8/300 Series C/C++ Compiler Package Application Note	REJ05B0
Single Power Supply F-ZTAT™ On-Board Programming	ADE-502

Section 2 CPU

Section 3 Exception Handling

Section 4 Address Break

Section 15 I^2C Bus Interface 2 (IIC2)

— Timer A (can be used as a time base for a clock)
— Timer V (8-bit timer)
— Timer W (16-bit timer)
— Watchdog timer
— SCI (Asynchronous or clocked synchronous serial communication interface)
— I^2C Bus Interface (conforms to the I^2C bus interface format that is advocated by P
Electronics)
— 10-bit A/D converter

Mask ROM version		H8/3694	HD6433694	HD6433694G	32 kbytes	1,024 bytes
		H8/3693	HD6433693	HD6433693G	24 kbytes	1,024 bytes
		H8/3692	HD6433692	HD6433692G	16 kbytes	512 bytes
		H8/3691	HD6433691	HD6433691G	12 kbytes	512 bytes
		H8/3690	HD6433690	HD6433690G	8 kbytes	512 bytes
EEPROM stacked version (512 bytes)	Flash memory version	H8/3694N	—	HD64N3694G	32 kbytes	2,048 bytes
	Mask-ROM version		—	HD6483694G	32 kbytes	1,024 bytes

- General I/O ports
  - — I/O pins: 29 I/O pins (27 I/O pins for H8/3694N), including 8 large current ports (mA, @V_{OL} = 1.5 V)
  - — Input-only pins: 8 input pins (also used for analog input)
- EEPROM interface (only for H8/3694N)
  - — I^2C bus interface (conforms to the I^2C bus interface format that is advocated by Ph Electronics)
- Supports various power-down modes

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

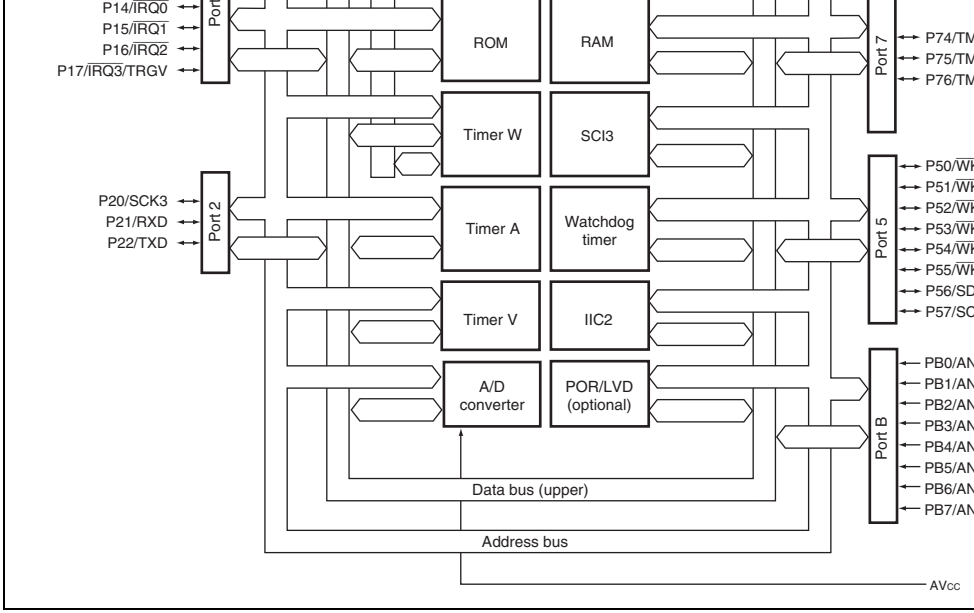
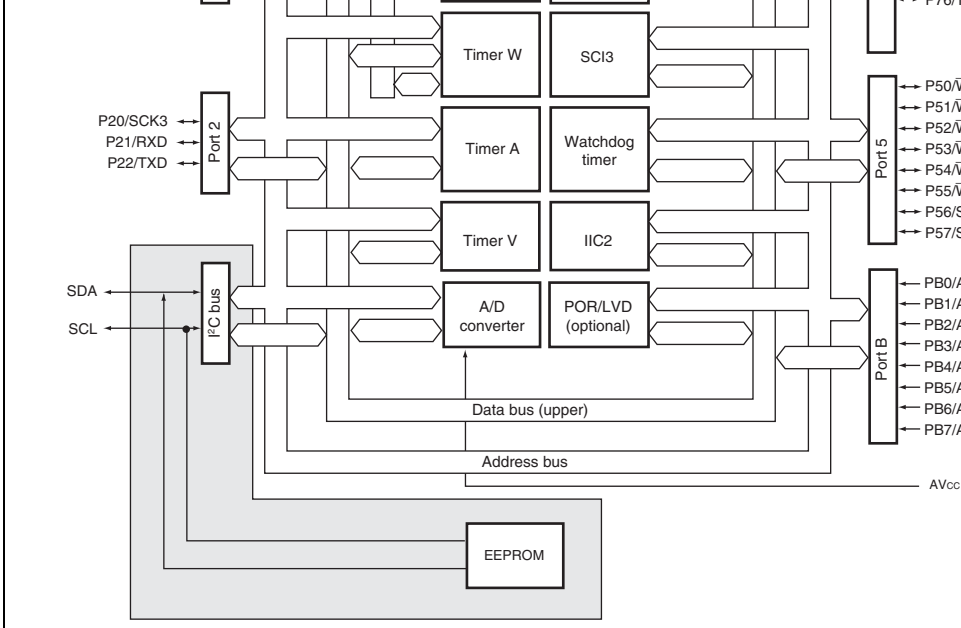
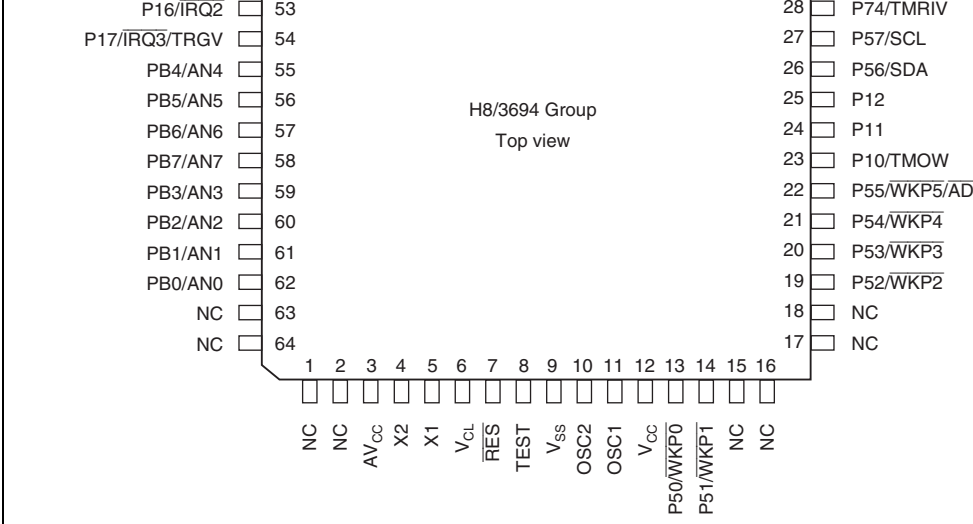


Figure 1.1 Internal Block Diagram of H8/3694 Group of F-ZTAT™ and Mask-ROM Versions



Note: The HD64N3694G is a stacked-structure product in which an EEPROM chip is mounted on the HD64F3694G (F-ZTAT™ vers
The HD6483694G is a stacked-structure product in which an EEPROM chip is mounted on the HD6433694G (mask-ROM ve

Figure 1.2 Internal Block Diagram of H8/3694N (EEPROM Stacked Versi



Note: Do not connect NC pins (these pins are not connected to the internal circuitry).

Figure 1.3 Pin Arrangement of H8/3694 Group of F-ZTAT™ and Mask-ROM V
(FP-64E, FP-64A)

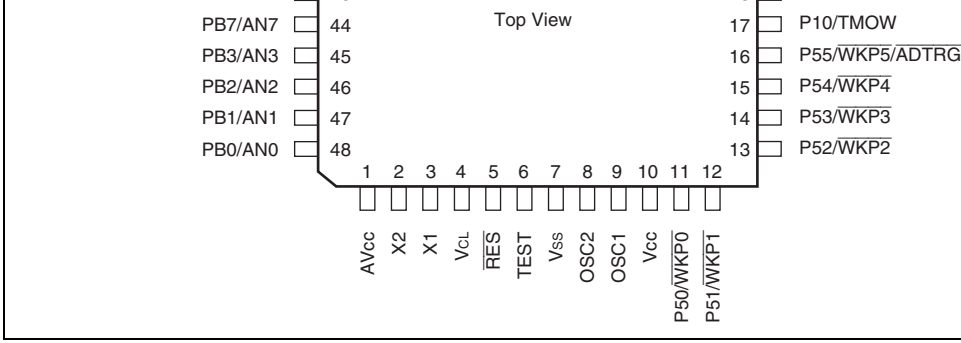
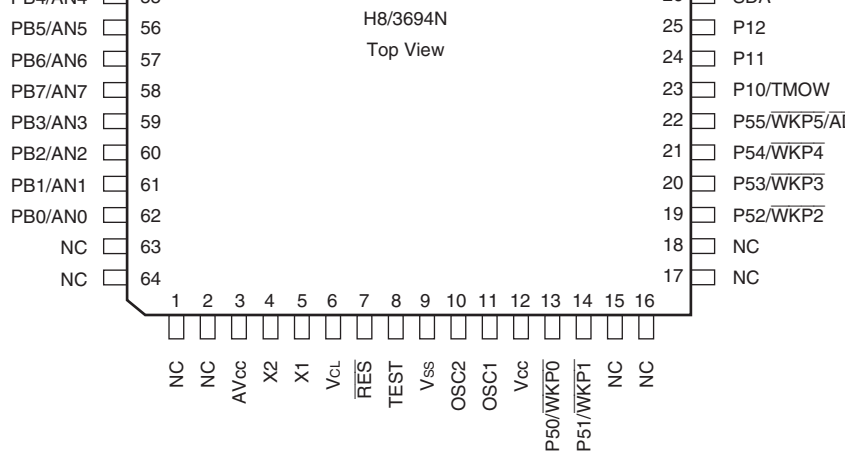


Figure 1.4 Pin Arrangement of H8/3694 Group of F-ZTAT™ and Mask-ROM V
(FP-48F, FP-48B, TNP-48)



Note: Do not connect NC pins.

Figure 1.5 Pin Arrangement of H8/3694N (EEPROM Stacked Version) (FP-64E)

					power supply (0V).
	AV_{CC}	3	1	Input	Analog power supply pin for the A/D c When the A/D converter is not used, this pin to the system power supply.
	V_{CL}	6	4	Input	Internal step-down power supply pin. capacitor of around 0.1 µF between t and the Vss pin for stabilization.
Clock pins	OSC1	11	9	Input	These pins connect with crystal or ce resonator for the system clock, or can to input an external clock. See section 5, Clock Pulse Generato typical connection.
	OSC2	10	8	Output	
	X1	5	3	Input	These pins connect with a 32.768 kH resonator for the subclock. See sectio Clock Pulse Generators, for a typical connection.
	X2	4	2	Output	
System control	$\overline{RES}$	7	5	Input	Reset pin. The pull-up resistor (typ. 1 incorporated. When driven low, the c reset.
	TEST	8	6	Input	Test pin. Connect this pin to Vss.
Interrupt pins	$\overline{NMI}$	35	25	Input	Non-maskable interrupt request input sure to pull-up by a pull-up resistor.
	$\overline{IRQ0}$ to $\overline{IRQ3}$	51 to 54	37 to 40	Input	External interrupt request input pins. the rising or falling edge.
	$\overline{WKP0}$ to $\overline{WKP5}$	13, 14, 19 to 22	11 to 16	Input	External interrupt request input pins. the rising or falling edge.

Timer W	FTCI	36	26	Input	External event input pin.
	FTIOA to FTIOD	37 to 40	27 to 30	I/O	Output compare output/input captu PWM output pin
I^2C bus interface (IIC)	SDA	26*[1]	20	I/O	IIC data I/O pin. Can directly drive NMOS open-drain output.
	SCL	27*[1]	21	I/O (EEPROM: Input)	IIC clock I/O pin. Can directly drive by NMOS open-drain output.
Serial communi-cation interface (SCI)	TXD	46	36	Output	Transmit data output pin
	RXD	45	35	Input	Receive data input pin
	SCK3	44	34	I/O	Clock I/O pin
A/D converter	AN7 to AN0	55 to 62	41 to 48	Input	Analog input pin
	$\overline{\text{ADTRG}}$	22	16	Input	A/D converter trigger input pin.
I/O ports	PB7 to PB0	55 to 62	41 to 48	Input	8-bit input port.
	P17 to P14, P12 to P10	51 to 54, 23 to 25	37 to 40 17 to 19	I/O	7-bit I/O port.
	P22 to P20	44 to 46	34 to 36	I/O	3-bit I/O port.
	P57 to P50	13, 14, 19 to 22, 26*[2], 27*[2]	20, 21, 13 to 16, 11, 12	I/O	8-bit I/O port

2. The P57 and P56 pins are not available in the H8/3694N.

- General-register architecture
  - Sixteen 16-bit general registers also usable as sixteen 8-bit registers and eight 16 registers, or eight 32-bit registers
- Sixty-two basic instructions
  - 8/16/32-bit data transfer and arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d:24,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 64-kbyte address space
- High-speed operation
  - All frequently-used instructions execute in one or two states
  - 8/16/32-bit register-register add/subtract : 2 state
  - 8 × 8-bit register-register multiply : 14 states
  - 16 ÷ 8-bit register-register divide : 14 states
  - 16 × 16-bit register-register multiply : 22 states
  - 32 ÷ 16-bit register-register divide : 22 states
- Power-down state
  - Transition to power-down state by SLEEP instruction

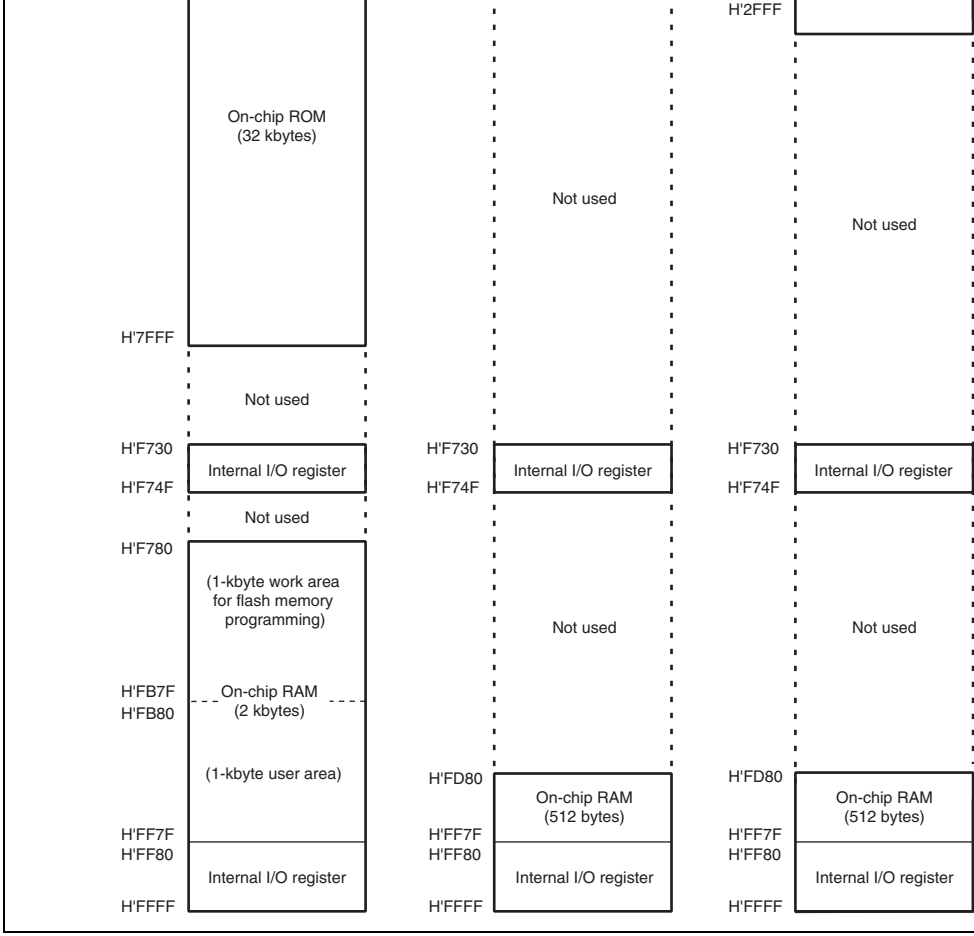


Figure 2.1 Memory Map (1)

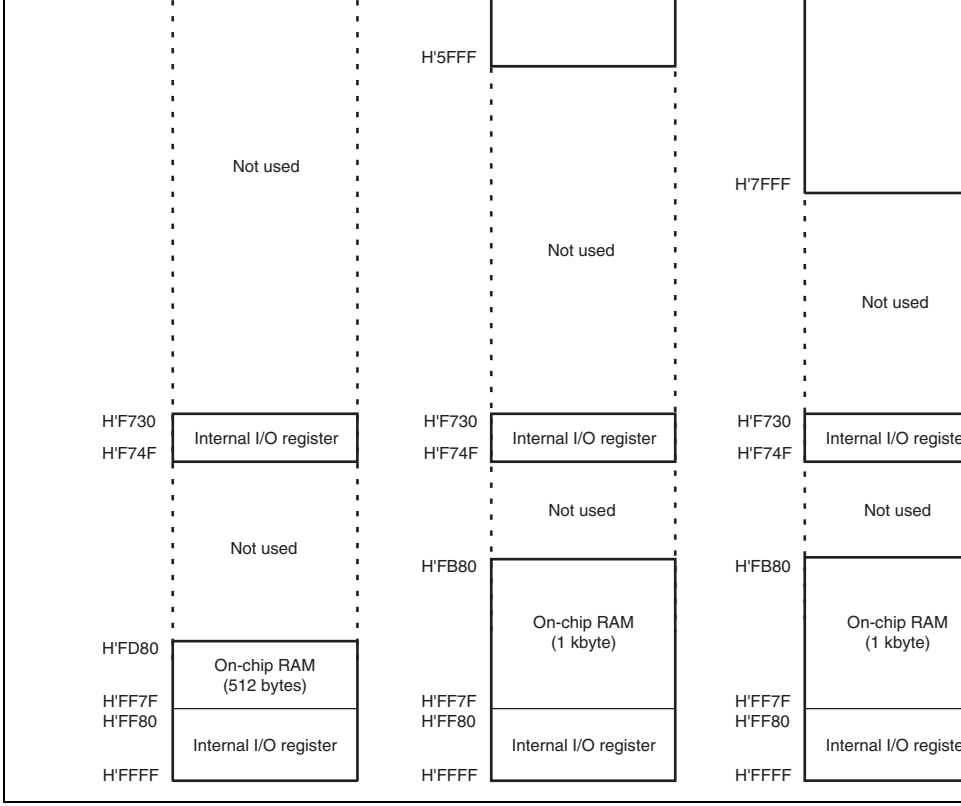


Figure 2.1 Memory Map (2)

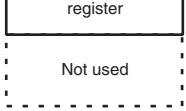


Figure 2.1 Memory Map (3)

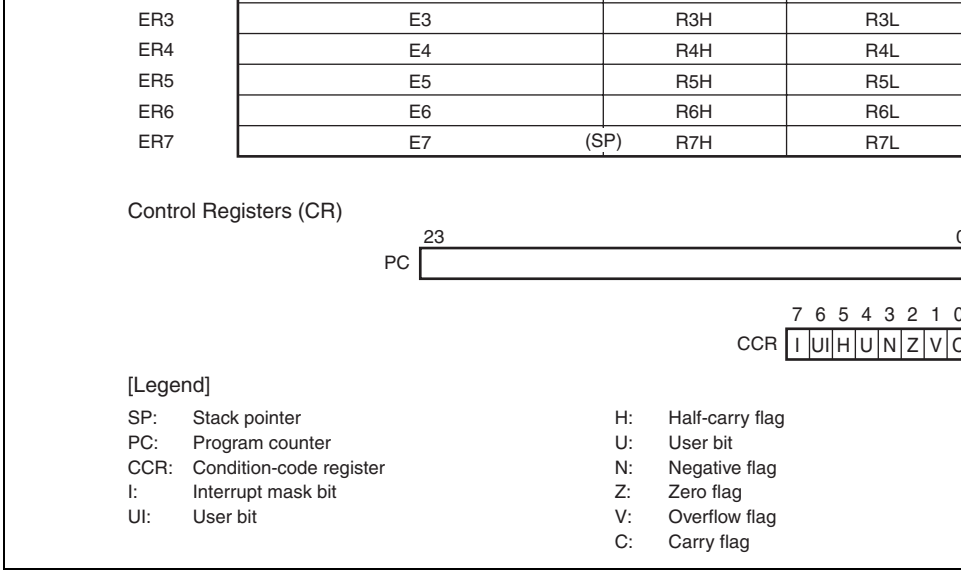


Figure 2.2 CPU Registers

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.

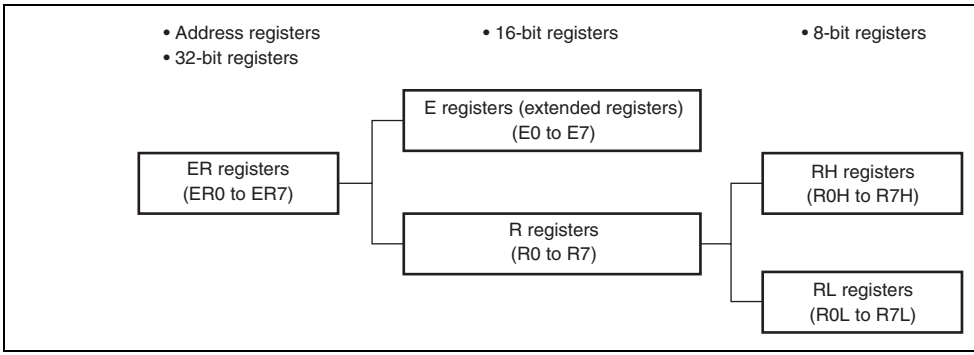


Figure 2.3 Usage of General Registers

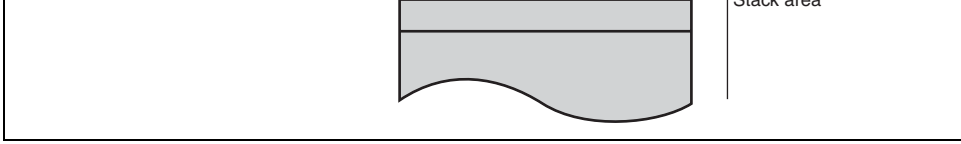


Figure 2.4 Relationship between Stack Pointer and Stack Area

2.2.2 Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute. T
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.

				When the ADD.B, ADDX.B, SUB.B, SUBX.B, C or NEG.B instruction is executed, this flag is se 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 da 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.

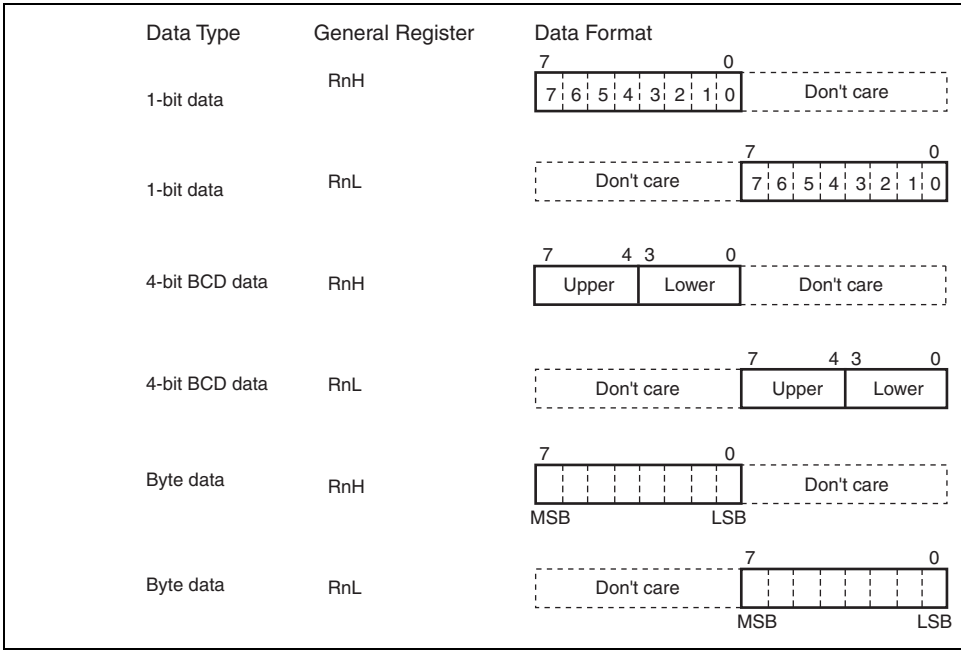
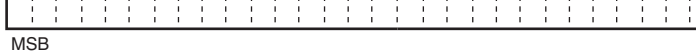


Figure 2.5 General Register Data Formats (1)



[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)

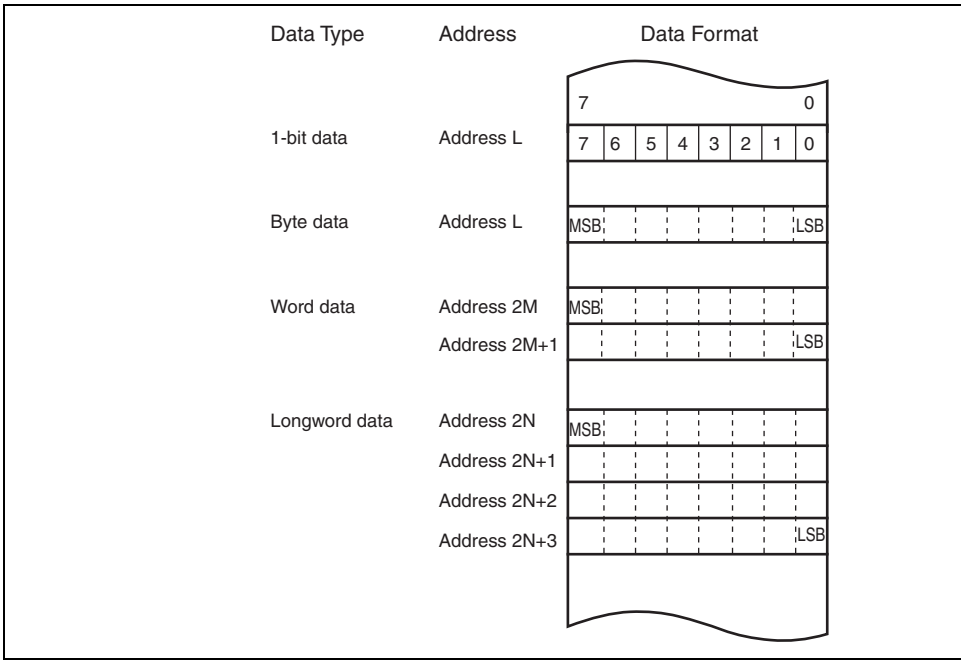


Figure 2.6 Memory Data Formats

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).

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

		another general register or immediate data.
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.

2.4.2 Basic Instruction Formats

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

Figure 2.7 shows examples of instruction formats.

- Operation Field

  Indicates the function of the instruction, the addressing mode, and the operation to b out on the operand. The operation field always includes the first four bits of the instr Some instructions have two operation fields.
- Register Field

  Specifies a general register. Address registers are specified by 3 bits, and data registe 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.

(4) Operation field, effective address extension, and condition field

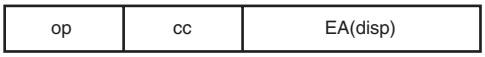


Figure 2.7 Instruction Formats

2.5 Addressing Modes and Effective Address Calculation

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

2.5.1 Addressing Modes

The H8/300H CPU supports the eight addressing modes listed in table 2.10. Each instruc
a subset of these addressing modes. Addressing modes that can be used differ depending
instruction. For details, refer to appendix A.4, Combinations of Instructions and Addressi
Modes.

Arithmetic and logic instructions can use the register direct and immediate modes. Data t
instructions can use all addressing modes except program-counter relative and memory i
Bit manipulation instructions use register direct, register indirect, or the absolute addressi
(@aa:8) to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instr
or immediate (3-bit) addressing mode to specify a bit number in the operand.

7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@@aa:8

Register Direct—Rn

The register field of the instruction specifies an 8-, 16-, or 32-bit general register contain operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn), the lower 2 which contain the address of the operand on memory.

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

A 16-bit or 24-bit displacement contained in the instruction is added to an address regist specified by the register field of the instruction, and the lower 24 bits of the sum the add 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 lowe 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 r The value added is 1 for byte access, 2 for word access, or 4 for longword access. Fo or longword access, the register value should be even.

For an 8-bit absolute address, the upper 16 bits are all assumed to be 1 (H'FFFF). For a 16
absolute address the upper 8 bits are a sign extension. A 24-bit absolute address can acces
entire address space.

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

Table 2.11 Absolute Address Access Ranges

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

Immediate—#xx:8, #xx:16, or #xx:32

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

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some l
manipulation instructions contain 3-bit immediate data in the instruction code, specifying
number. The TRAPA instruction contains 2-bit immediate data in its instruction code, spe
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 th
instruction is sign-extended and added to the 24-bit PC contents to generate a branch addr
PC value to which the displacement is added is the address of the first byte of the next ins

address range is 0 to 255 (H'0000 to H'00FF).

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

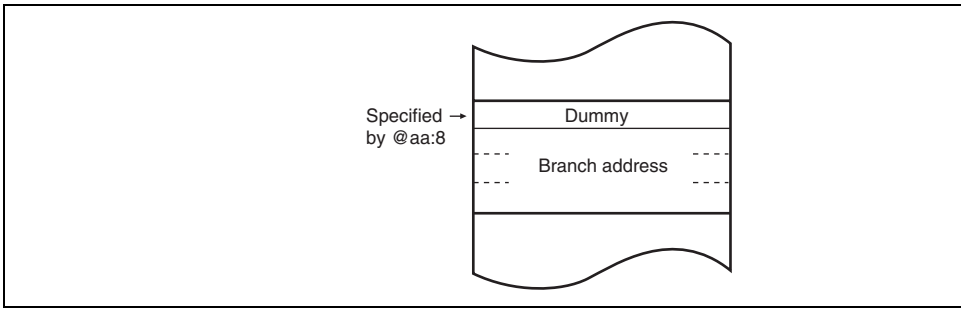
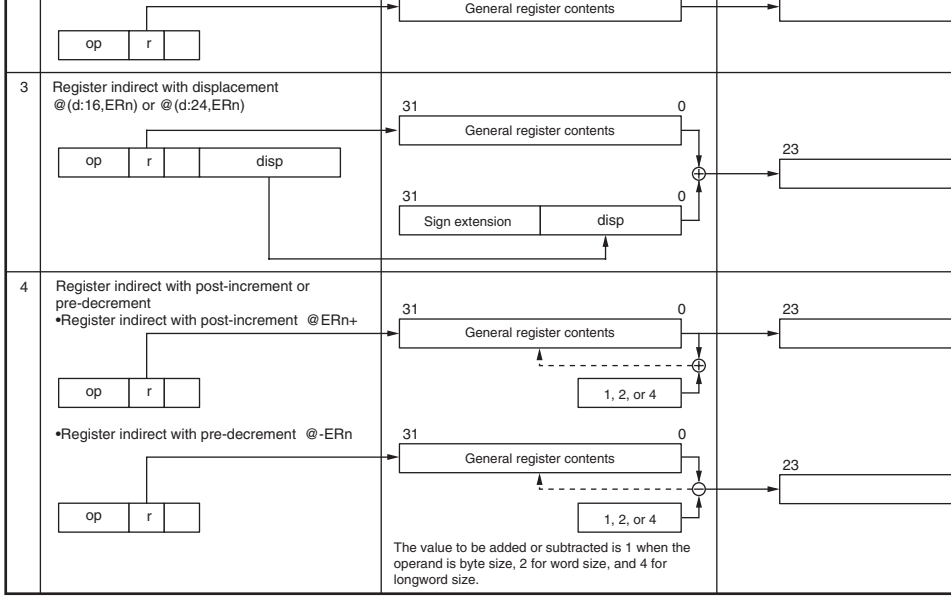
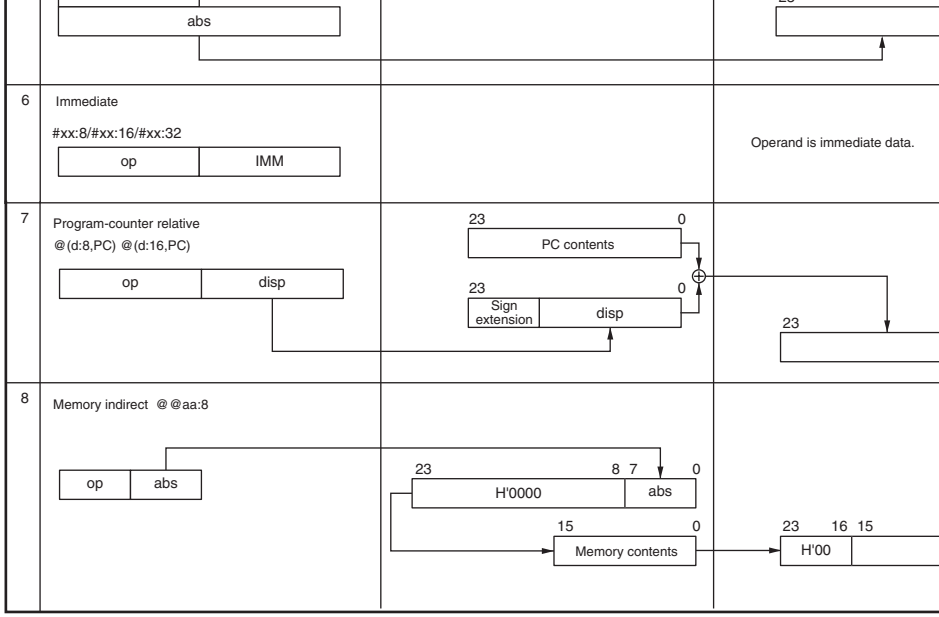


Figure 2.8 Branch Address Specification in Memory Indirect Mode


General register contents
op
r
3
Register indirect with displacement
@(d:16,ERn) or @(d:24,ERn)
31
0
General register contents
op
r
disp
23
31
0
Sign extension
disp
4
Register indirect with post-increment or
pre-decrement
•Register indirect with post-increment @ERn+
31
0
General register contents
23
op
r
1, 2, or 4
•Register indirect with pre-decrement @-ERn
31
0
General register contents
23
op
r
1, 2, or 4
The value to be added or subtracted is 1 when the operand is byte size, 2 for word size, and 4 for longword size.



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

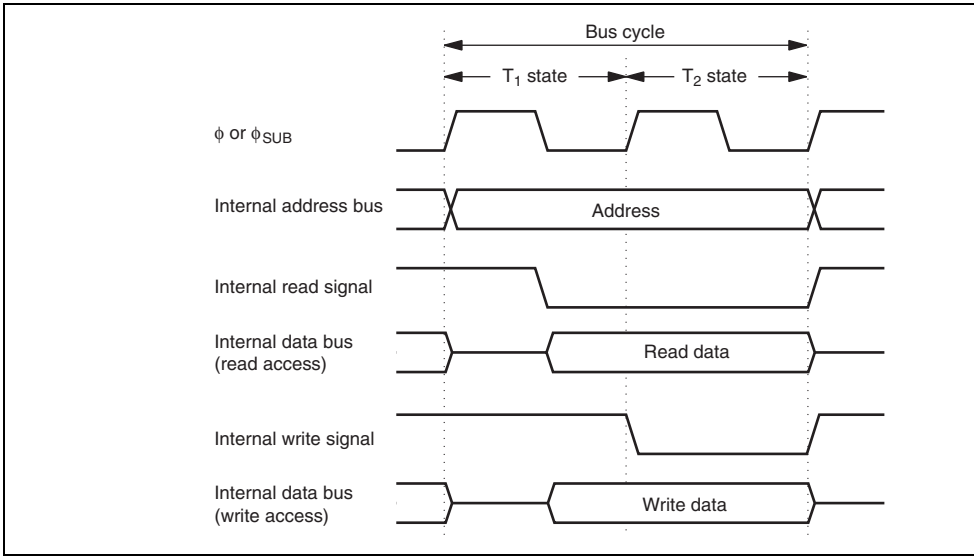


Figure 2.9 On-Chip Memory Access Cycle

module.

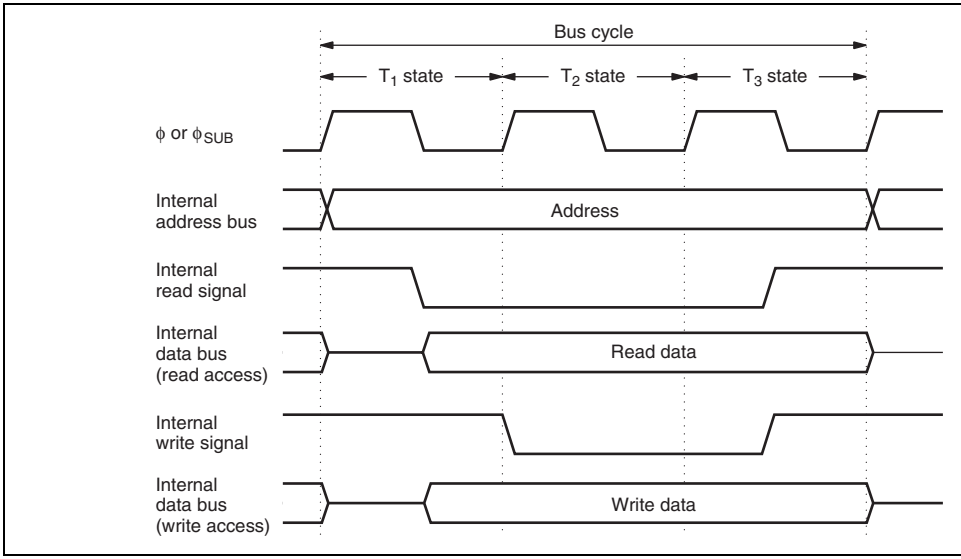


Figure 2.10 On-Chip Peripheral Module Access Cycle (3-State Access)

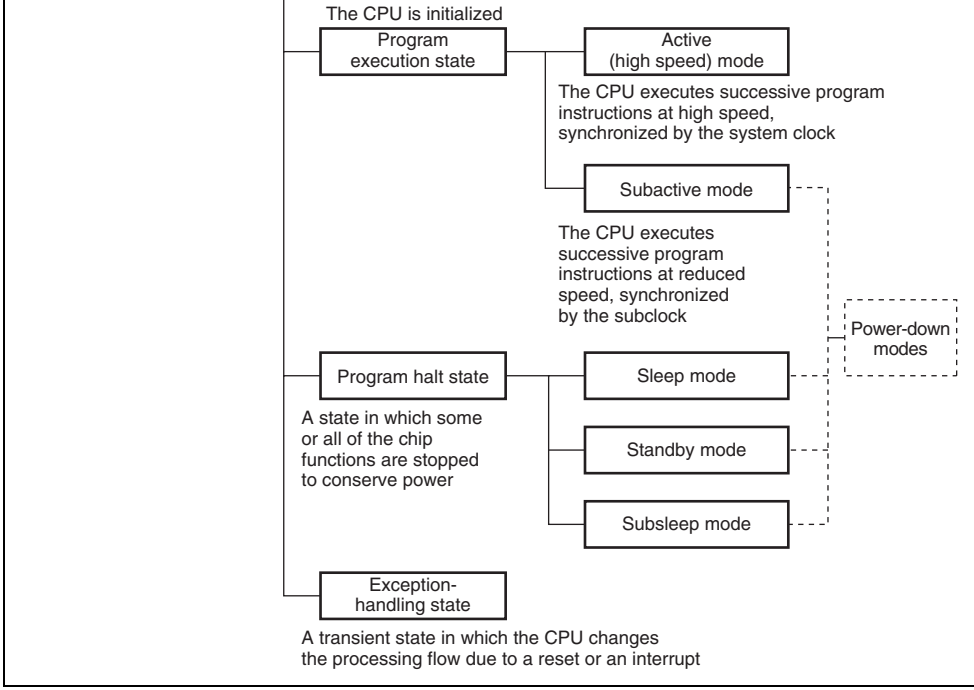


Figure 2.11 CPU Operation States

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
which starts from the address indicated by R5, to the address indicated by R6. Set R4L a
that the end address of the destination address (value of R6 + R4L) does not exceed H'F
value of R6 must not change from H'FFFF to H'0000 during execution).

2.8.3 Bit Manipulation Instruction

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

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.

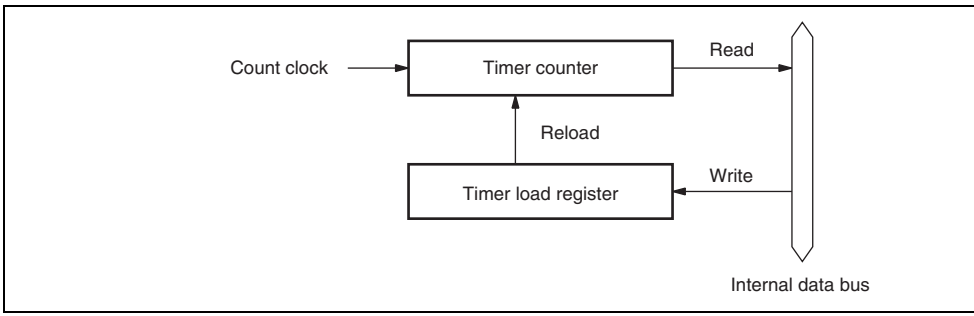


Figure 2.13 Example of Timer Configuration with Two Registers Allocated to S Address

Example 2: The BSET instruction is executed for port 5.

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

```
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.
   As a result of the BSET instruction, bit 0 in PDR5 becomes 1, and P50 outputs a hig signal. However, bits 7 and 6 of PDR5 end up with different values. To prevent this store a copy of the PDR5 data in a work area in memory. Perform the bit manipulati data in the work area, then write this data to PDR5.

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

Bit Manipulation in a Register Containing a Write-Only Bit

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

P57 and P56 are input pins, with a low-level signal input at P57 and a high-level signal in
P56. P55 to P50 are output pins that output low-level signals. An example of setting the P

- 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. The TRAPA ins
generates a vector address corresponding to a vector number from 0 to 3, as specified in
instruction code. Exception handling can be executed at all times in the program executi
regardless of the setting of the I bit in CCR.

- Interrupts

External interrupts other than NMI and internal interrupts other than address break are m
the I bit in CCR, and kept masked while the I bit is set to 1. Exception handling starts wh
current instruction or exception handling ends, if an interrupt request has been issued.

3.1 Exception Sources and Vector Address

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

Table 3.1 Exception Sources and Vector Address

Relative Module	Exception Sources	Vector Number	Vector Address
$\overline{\text{RES}}$ pin Watchdog timer	Reset	0	H'0000 to H'0001
—	Reserved for system use	1 to 6	H'0002 to H'000D
External interrupt pin	NMI	7	H'000E to H'000F
CPU	Trap instruction (#0)	8	H'0010 to H'0011
	(#1)	9	H'0012 to H'0013
	(#2)	10	H'0014 to H'0015
	(#3)	11	H'0016 to H'0017
Address break	Break conditions satisfied	12	H'0018 to H'0019

	WKP	18	H'0024 to H'0025
Timer A	Overflow	19	H’0026 to H’0027
—	Reserved for system use	20	H’0028 to H’0029
Timer W	Timer W input capture A /compare match A Timer W input capture B /compare match B Timer W input capture C /compare match C Timer W input capture D /compare match D Timer W overflow	21	H’002A to H’002B
Timer V	Timer V compare match A Timer V compare match B Timer V overflow	22	H'002C to H'002D
SCI3	SCI3 receive data full SCI3 transmit data empty SCI3 transmit end SCI3 receive error	23	H'002E to H'002F
IIC2	Transmit data empty Transmit end Receive data full Arbitration lost/Overrun error NACK detection Stop conditions detected	24	H'0030 to H'0031
A/D converter	A/D conversion end	25	H'0032 to H'0033

Note * A low-voltage detection interrupt is enabled only in the product with an on-chip on reset and low-voltage detection circuit.

IEGR1 selects the direction of an edge that generates interrupt requests of pins $\overline{\text{NMI}}$ and $\overline{\text{IRQ0}}$.

Bit	Bit Name	Initial Value	R/W	Description
7	NMIEG	0	R/W	NMI Edge Select 0: Falling edge of $\overline{\text{NMI}}$ pin input is detected 1: Rising edge of $\overline{\text{NMI}}$ pin input is detected
6 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	IEG3	0	R/W	IRQ3 Edge Select 0: Falling edge of $\overline{\text{IRQ3}}$ pin input is detected 1: Rising edge of $\overline{\text{IRQ3}}$ pin input is detected
2	IEG2	0	R/W	IRQ2 Edge Select 0: Falling edge of $\overline{\text{IRQ2}}$ pin input is detected 1: Rising edge of $\overline{\text{IRQ2}}$ pin input is detected
1	IEG1	0	R/W	IRQ1 Edge Select 0: Falling edge of $\overline{\text{IRQ1}}$ pin input is detected 1: Rising edge of $\overline{\text{IRQ1}}$ pin input is detected
0	IEG0	0	R/W	IRQ0 Edge Select 0: Falling edge of $\overline{\text{IRQ0}}$ pin input is detected 1: Rising edge of $\overline{\text{IRQ0}}$ pin input is detected

				0: Falling edge of WKP5(ADTRG) pin input is de
				1: Rising edge of WKP5(ADTRG) pin input is det
4	WPEG4	0	R/W	WKP4 Edge Select
				0: Falling edge of WKP4 pin input is detected
				1: Rising edge of WKP4 pin input is detected
3	WPEG3	0	R/W	WKP3 Edge Select
				0: Falling edge of WKP3 pin input is detected
				1: Rising edge of WKP3 pin input is detected
2	WPEG2	0	R/W	WKP2 Edge Select
				0: Falling edge of WKP2 pin input is detected
				1: Rising edge of WKP2 pin input is detected
1	WPEG1	0	R/W	WKP1Edge Select
				0: Falling edge of WKP1 pin input is detected
				1: Rising edge of WKP1 pin input is detected
0	WPEG0	0	R/W	WKP0 Edge Select
				0: Falling edge of WKP0 pin input is detected
				1: Rising edge of WKP0 pin input is detected

				requests are enabled.
5	IENWP	0	R/W	Wakeup Interrupt Enable This bit is an enable bit, which is common to th $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$. When the bit is set to 1, interru requests are enabled.
4	—	1	—	Reserved This bit is always read as 1.
3	IEN3	0	R/W	IRQ3 Interrupt Enable When this bit is set to 1, interrupt requests of th pin are enabled.
2	IEN2	0	R/W	IRQ2 Interrupt Enable When this bit is set to 1, interrupt requests of th pin are enabled.
1	IEN1	0	R/W	IRQ1 Interrupt Enable When this bit is set to 1, interrupt requests of th pin are enabled.
0	IEN0	0	R/W	IRQ0 Interrupt Enable When this bit is set to 1, interrupt requests of th pin are enabled.

When disabling interrupts by clearing bits in an interrupt enable register, or when clearin an interrupt flag register, always do so while interrupts are masked (I = 1). If the above o operations are performed while I = 0, and as a result a conflict arises between the clear i and an interrupt request, exception handling for the interrupt will be executed after the c instruction has been executed.

				[Clearing condition] When IRRDT is cleared by writing 0
6	IRRTA	0	R/W	Timer A Interrupt Request Flag [Setting condition] When the timer A counter value overflows [Clearing condition] When IRRTA is cleared by writing 0
5, 4	—	All 1	—	Reserved These bits are always read as 1.
3	IRRI3	0	R/W	IRQ3 Interrupt Request Flag [Setting condition] When $\overline{\text{IRQ3}}$ pin is designated for interrupt input a designated signal edge is detected. [Clearing condition] When IRRI3 is cleared by writing 0
2	IRRI2	0	R/W	IRQ2 Interrupt Request Flag [Setting condition] When $\overline{\text{IRQ2}}$ pin is designated for interrupt input a designated signal edge is detected. [Clearing condition] When IRRI2 is cleared by writing 0
1	IRRI1	0	R/W	IRQ1 Interrupt Request Flag [Setting condition] When $\overline{\text{IRQ1}}$ pin is designated for interrupt input a designated signal edge is detected. [Clearing condition] When IRRI1 is cleared by writing 0

IWPR is a status flag register for $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$ interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1.
5	IWPF5	0	R/W	WKP5 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP5}}$ pin is designated for interrupt inpu designated signal edge is detected. [Clearing condition] When IWPF5 is cleared by writing 0.
4	IWPF4	0	R/W	WKP4 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP4}}$ pin is designated for interrupt inpu designated signal edge is detected. [Clearing condition] When IWPF4 is cleared by writing 0.
3	IWPF3	0	R/W	WKP3 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP3}}$ pin is designated for interrupt inpu designated signal edge is detected. [Clearing condition] When IWPF3 is cleared by writing 0.

				When $\overline{\text{WKP1}}$ pin is designated for interrupt input designated signal edge is detected. [Clearing condition] When IWPF1 is cleared by writing 0.
0	IWPF0	0	R/W	WKP0 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP0}}$ pin is designated for interrupt input designated signal edge is detected. [Clearing condition] When IWPF0 is cleared by writing 0.

3.3 Reset Exception Handling

When the $\overline{\text{RES}}$ pin goes low, all processing halts and this LSI enters the reset. The interna the CPU and the registers of the on-chip peripheral modules are initialized by the reset. T that this LSI is reset at power-up, hold the $\overline{\text{RES}}$ pin low until the clock pulse generator ou stabilizes. To reset the chip during operation, hold the $\overline{\text{RES}}$ pin low for at least 10 system cycles. When the $\overline{\text{RES}}$ pin goes high after being held low for the necessary time, this LSI reset exception handling. The reset exception handling sequence is shown in figure 3.1.

The reset exception handling sequence is as follows. However, for the reset exception ha sequence of the product with on-chip power-on reset circuit, refer to section 18, Power-O and Low-Voltage Detection Circuits (Optional).

1. Set the I bit in the condition code register (CCR) to 1.
2. The CPU generates a reset exception handling vector address (from H'0000 to H'0001 data in that address is sent to the program counter (PC) as the start address, and progr execution starts from that address.

bit value in CCR.

IRQ3 to IRQ0 Interrupts

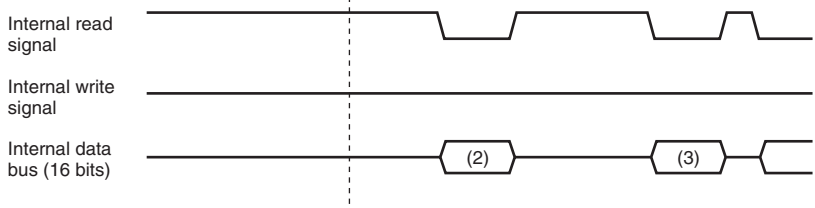
IRQ3 to IRQ0 interrupts are requested by input signals to pins $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$. These interrupts are given different vector addresses, and are detected individually by either rising edge sensing or falling edge sensing, depending on the settings of bits IEG3 to IEG0.

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

WKP5 to WKP0 Interrupts

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

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



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

Figure 3.1 Reset Sequence

3.4.2 Internal Interrupts

Each on-chip peripheral module has a flag to show the interrupt request status and the ena
enable or disable the interrupt. For timer A interrupt requests and direct transfer interrupt
generated by execution of a SLEEP instruction, this function is included in IRR1 and IEN

When an on-chip peripheral module requests an interrupt, the corresponding interrupt req
status flag is set to 1, requesting the CPU of an interrupt. These interrupts can be masked
writing 0 to clear the corresponding enable bit.

3.4.3 Interrupt Handling Sequence

Interrupts are controlled by an interrupt controller.

Interrupt operation is described as follows.

1. If an interrupt occurs while the NMI or interrupt enable bit is set to 1, an interrupt req
   signal is sent to the interrupt controller.

5. Then, the I bit of CCR is set to 1, masking further interrupts excluding the NMI and
break. Upon return from interrupt handling, the values of I bit and other bits in CCR
restored and returned to the values prior to the start of interrupt exception handling.
6. Next, the CPU generates the vector address corresponding to the accepted interrupt,
transfers the address to PC as a start address of the interrupt handling-routine. Then
starts executing from the address indicated in PC.

Figure 3.3 shows a typical interrupt sequence where the program area is in the on-chip F
the stack area is in the on-chip RAM.

[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.
3. Ignored when returning from the interrupt handling routine.

Figure 3.2 Stack Status after Exception Handling

3.4.4 Interrupt Response Time

Table 3.2 shows the number of wait states after an interrupt request flag is set until the fir instruction of the interrupt handling-routine is executed.

Table 3.2 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: * Not including EEPMOV instruction.

Figure 3.3 Interrupt Sequence

3.5.2 Notes on Stack Area Use

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

3.5.3 Notes on Rewriting Port Mode Registers

When a port mode register is rewritten to switch the functions of external interrupt pins, $\overline{\text{I}}$
$\overline{\text{IRQ0}}$, and $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$, the interrupt request flag may be set to 1.

When switching a pin function, mask the interrupt before setting the bit in the port mode
After accessing the port mode register, execute at least one instruction (e.g., NOP), then c
interrupt request flag from 1 to 0.

Figure 3.4 shows a port mode register setting and interrupt request flag clearing procedur

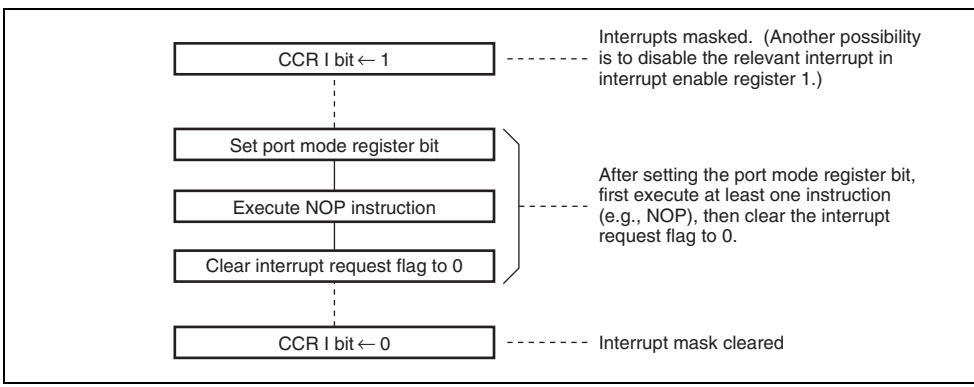


Figure 3.4 Port Mode Register Setting and Interrupt Request Flag Clearing Pro

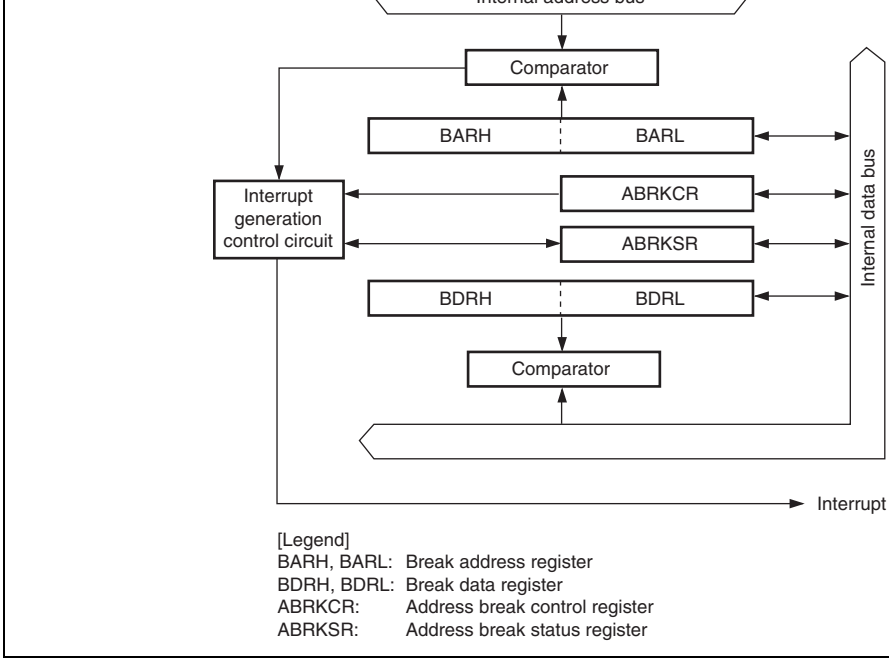


Figure 4.1 Block Diagram of Address Break

4.1 Register Descriptions

Address break has the following registers.

- Address break control register (ABRKCR)
- Address break status register (ABRKSR)
- Break address register (BARH, BARL)
- Break data register (BDRH, BDRL)

6	CSEL1	0	R/W	Condition Select 1 and 0
5	CSEL0	0	R/W	These bits set address break conditions. 00: Instruction execution cycle 01: CPU data read cycle 10: CPU data write cycle 11: CPU data read/write cycle
4	ACMP2	0	R/W	Address Compare Condition Select 2 to 0
3	ACMP1	0	R/W	These bits set the comparison condition between
2	ACMP0	0	R/W	address set in BAR and the internal address bus 000: Compares 16-bit addresses 001: Compares upper 12-bit addresses 010: Compares upper 8-bit addresses 011: Compares upper 4-bit addresses 1XX: Reserved (setting prohibited)
1	DCMP1	0	R/W	Data Compare Condition Select 1 and 0
0	DCMP0	0	R/W	These bits set the comparison condition between set in BDR and the internal data bus. 00: No data comparison 01: Compares lower 8-bit data between BDRL a bus 10: Compares upper 8-bit data between BDRH a bus 11: Compares 16-bit data between BDR and dat

Legend: X: Don't care.

ROM space	Upper 8 bits	Lower 8 bits	Upper 8 bits	Uppe
RAM space	Upper 8 bits	Lower 8 bits	Upper 8 bits	Uppe
I/O register with 8-bit data bus width	Upper 8 bits	Upper 8 bits	Upper 8 bits	Uppe
I/O register with 16-bit data bus width	Upper 8 bits	Lower 8 bits	—	—

4.1.2 Address Break Status Register (ABRKSR)

ABRKSR consists of the address break interrupt flag and the address break interrupt ena

Bit	Bit Name	Initial Value	R/W	Description
7	ABIF	0	R/W	Address Break Interrupt Flag [Setting condition] When the condition set in ABRKCR is satisfied [Clearing condition] When 0 is written after ABIF=1 is read
6	ABIE	0	R/W	Address Break Interrupt Enable When this bit is 1, an address break interrupt re enabled.
5 to 0	—	All 1	—	Reserved These bits are always read as 1.

even and odd addresses in the data transmission. Therefore, comparison data must be set
BDRH for byte access. For word access, the data bus used depends on the address. See s
4.1.1, Address Break Control Register (ABRKCR), for details. The initial value of this re
undefined.

4.2 Operation

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

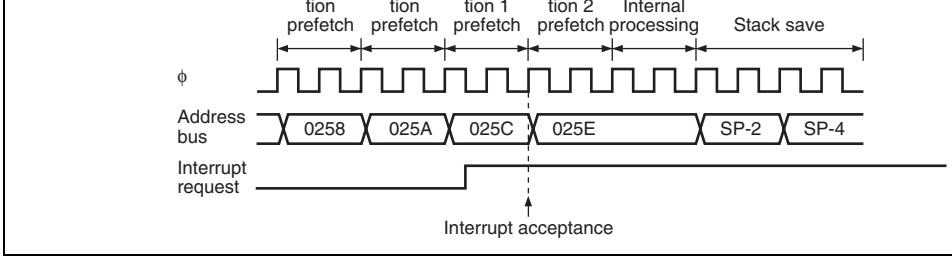


Figure 4.2 Address Break Interrupt Operation Example (1)

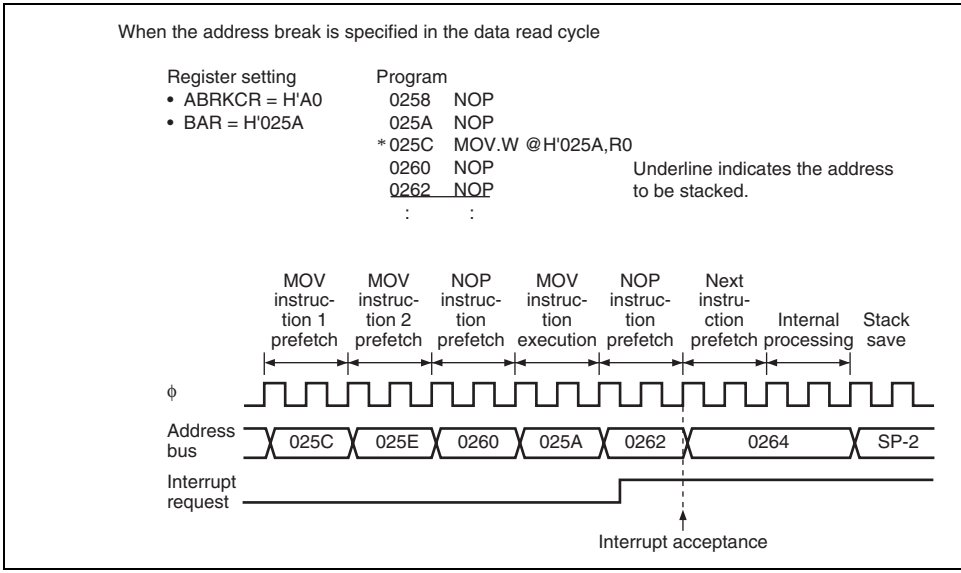


Figure 4.2 Address Break Interrupt Operation Example (2)

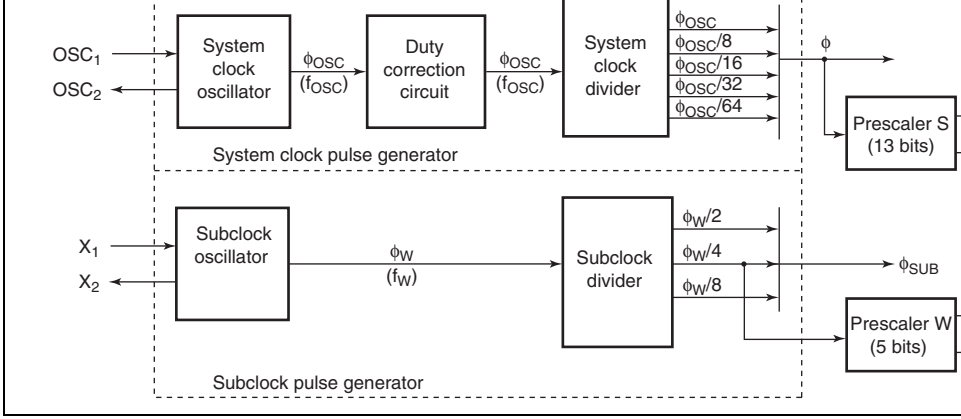


Figure 5.1 Block Diagram of Clock Pulse Generators

The basic clock signals that drive the CPU and on-chip peripheral modules are ϕ and ϕ_{SU} system clock is divided by prescaler S to become a clock signal from $\phi/8192$ to $\phi/2$, and subclock is divided by prescaler W to become a clock signal from $\phi w/128$ to $\phi w/8$. Both system clock and subclock signals are provided to the on-chip peripheral modules.

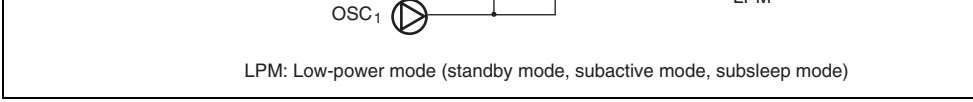


Figure 5.2 Block Diagram of System Clock Generator

5.1.1 Connecting Crystal Resonator

Figure 5.3 shows a typical method of connecting a crystal resonator. An AT-cut parallel-r
crystal resonator should be used. Figure 5.4 shows the equivalent circuit of a crystal reson
resonator having the characteristics given in table 5.1 should be used.

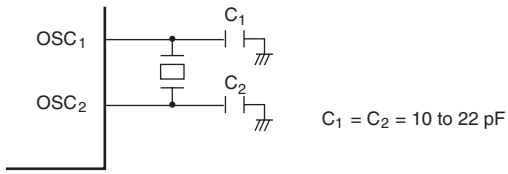


Figure 5.3 Typical Connection to Crystal Resonator

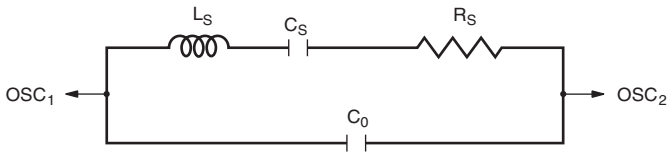


Figure 5.4 Equivalent Circuit of Crystal Resonator

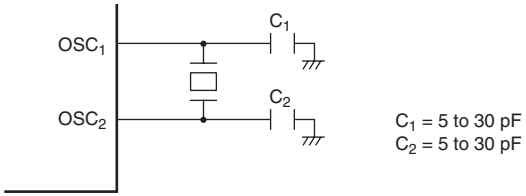


Figure 5.5 Typical Connection to Ceramic Resonator

5.1.3 External Clock Input Method

Connect an external clock signal to pin OSC_1, and leave pin OSC_2 open. Figure 5.6 show
connection. The duty cycle of the external clock signal must be 45 to 55%.

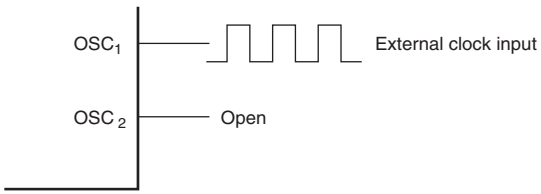


Figure 5.6 Example of External Clock Input

Note : Registance is a reference value.

Figure 5.7 Block Diagram of Subclock Generator

5.2.1 Connecting 32.768-kHz Crystal Resonator

Clock pulses can be supplied to the subclock divider by connecting a 32.768-kHz crystal resonator, as shown in figure 5.8. Figure 5.9 shows the equivalent circuit of the 32.768-k resonator.

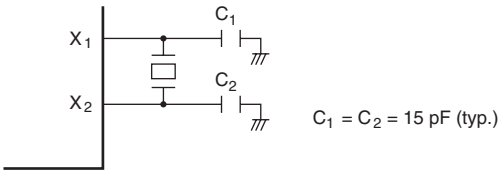
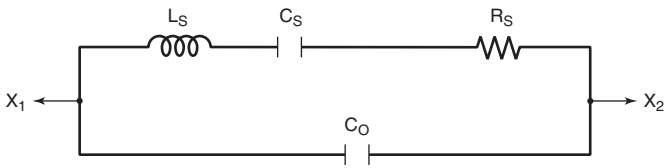


Figure 5.8 Typical Connection to 32.768-kHz Crystal Resonator



C_O = 1.5 pF (typ.)
R_S = 14 kΩ (typ.)
f_W = 32.768 kHz

Note: Constants are reference values.

Figure 5.9 Equivalent Circuit of 32.768-kHz Crystal Resonator

Figure 5.10 Pin Connection when not Using Subclock

5.3 Prescalers

5.3.1 Prescaler S

Prescaler S is a 13-bit counter using the system clock (ϕ) as its input clock. It is increme per clock period. Prescaler S is initialized to H'0000 by a reset, and starts counting on ex the reset state. In standby mode, subactive mode, and subsleep mode, the system clock p generator stops. Prescaler S also stops and is initialized to H'0000. The CPU cannot read prescaler S. The output from prescaler S is shared by the on-chip peripheral modules. Th ratio can be set separately for each on-chip peripheral function. In active mode and sleep the clock input to prescaler S is determined by the division factor designated by MA2 to SYSCR2.

5.3.2 Prescaler W

Prescaler W is a 5-bit counter using a 32.768 kHz signal divided by 4 ($\phi_w/4$) as its input divided output is used for clock time base operation of timer A. Prescaler W is initialize by a reset, and starts counting on exit from the reset state. Even in standby mode, subact or subsleep mode, prescaler W continues functioning so long as clock signals are suppli X_1 and X_2. Prescaler W can be reset by setting 1s in bits TMA3 and TMA2 of timer mo A (TMA).

5.4.2 Notes on Board Design

When using a crystal resonator (ceramic resonator), place the resonator and its load capac
close as possible to the OSC_1 and OSC_2 pins. Other signal lines should be routed away fro
resonator circuit to prevent induction from interfering with correct oscillation (see figure

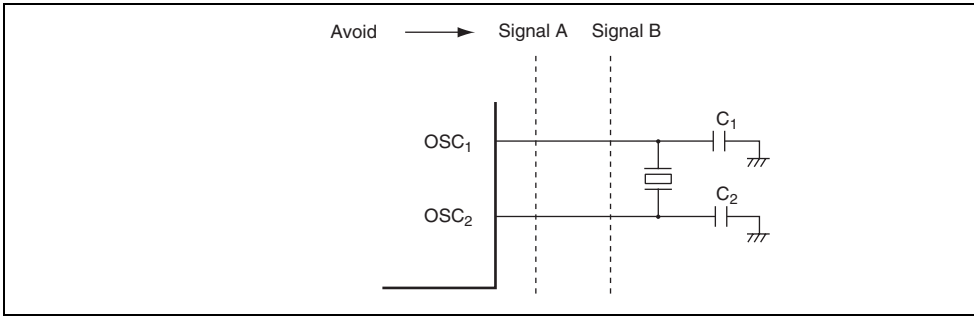


Figure 5.11 Example of Incorrect Board Design

The CPU and all on-chip peripheral modules are operable on the subclock. The subc
frequency can be selected from ϕw/2, ϕw/4, and ϕw/8.

- Sleep mode

  The CPU halts. On-chip peripheral modules are operable on the system clock.
- Subsleep mode

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

  The CPU and all on-chip peripheral modules halt. When the clock time-base functio
  selected, timer A is operable.
- Module standby mode

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

6.1 Register Descriptions

The registers related to power-down modes are listed below.

- System control register 1 (SYSCR1)
- System control register 2 (SYSCR2)
- Module standby control register 1 (MSTCR1)

				1: a transition is made to standby mode. For details, see table 6.2.
6	STS2	0	R/W	Standby Timer Select 2 to 0
5	STS1	0	R/W	These bits designate the time the CPU and perip
4	STS0	0	R/W	modules wait for stable clock operation after exit standby mode, subactive mode, or subsleep mo active mode or sleep mode due to an interrupt. T designation should be made according to the clo frequency so that the waiting time is at least 6.5 relationship between the specified value and the of wait states is shown in table 6.1. When an ext clock is to be used, the minimum value (STS2 = STS0 = 1) is recommended.
3	NESEL	0	R/W	Noise Elimination Sampling Frequency Select The subclock pulse generator generates the wat signal (ϕ_w) and the system clock pulse generator generates the oscillator clock (ϕ_{osc}). This bit sele sampling frequency of the oscillator clock when t clock signal (ϕ_w) is sampled. When ϕ_{osc} = 4 to 20 clear NESEL to 0. 0: Sampling rate is ϕ_{osc}/16 1: Sampling rate is ϕ_{osc}/4
2 to 0	—	All 0	—	Reserved These bits are always read as 0.

	1	0	128 states	0.00	0.00	0.01	0.02	0.03	0.06	0.13
		1	16 states	0.00	0.00	0.00	0.00	0.00	0.01	0.02

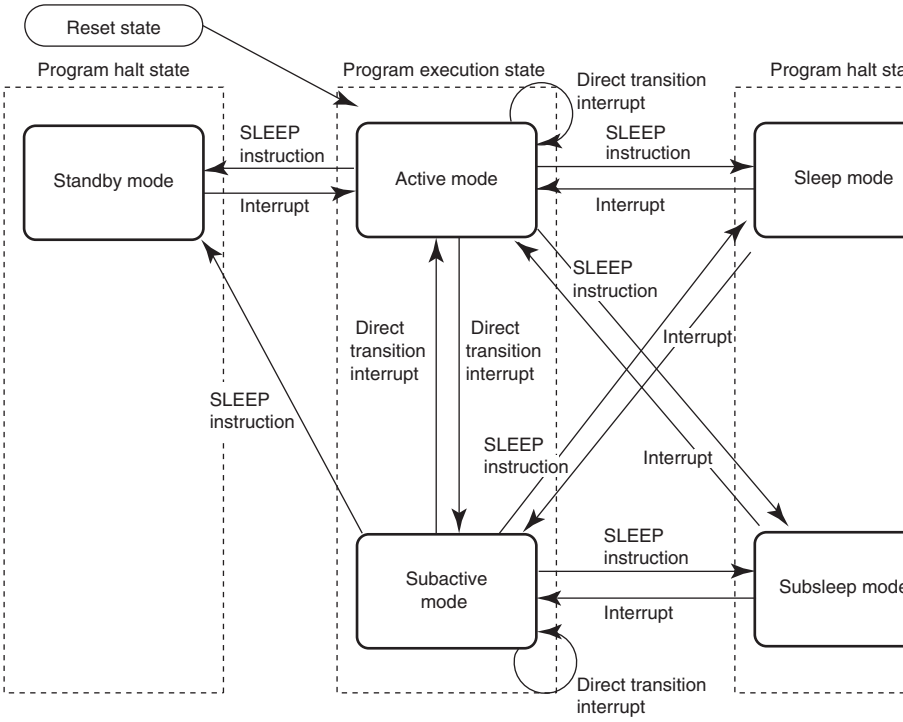
Note: Time unit is ms.

				of a SLEEP instruction, as well as bit SSBY of S For details, see table 6.2.
4	MA2	0	R/W	Active Mode Clock Select 2 to 0
3	MA1	0	R/W	These bits select the operating clock frequency i
2	MA0	0	R/W	and sleep modes. The operating clock frequency changes to the set frequency after the SLEEP in is executed. 0XX: ϕ_{osc} 100: ϕ_{osc}/8 101: ϕ_{osc}/16 110: ϕ_{osc}/32 111: ϕ_{osc}/64
1	SA1	0	R/W	Subactive Mode Clock Select 1 and 0
0	SA0	0	R/W	These bits select the operating clock frequency i subactive and subsleep modes. The operating cl frequency changes to the set frequency after the instruction is executed. 00: ϕ_w/8 01: ϕ_w/4 1X: ϕ_w/2

Legend: X : Don't care.

5	MSTS3	0	R/W	SCI3 Module Standby SCI3 enters standby mode when this bit is set t
4	MSTAD	0	R/W	A/D Converter Module Standby A/D converter enters standby mode when this b 1
3	MSTWD	0	R/W	Watchdog Timer Module Standby Watchdog timer enters standby mode when this to 1.When the internal oscillator is selected for watchdog timer clock, the watchdog timer opera regardless of the setting of this bit
2	MSTTW	0	R/W	Timer W Module Standby Timer W enters standby mode when this bit is s
1	MSTTV	0	R/W	Timer V Module Standby Timer V enters standby mode when this bit is s
0	MSTTA	0	R/W	Timer A Module Standby Timer A enters standby mode when this bit is s

each mode.



Notes: 1. To make a transition to another mode by an interrupt, make sure interrupt handling is after the int
is accepted.
2. Details on the mode transition conditions are given in table 6.2.

Figure 6.1 Mode Transition Diagram

	X	0*	0	Active mode (direct transition)	—
	X	X	1	Subactive mode (direct transition)	—

Legend: X : Don’t care.

* When a state transition is performed while SMSEL is 1, timer V, SCI3, and th converter are reset, and all registers are set to their initial values. To use thes functions after entering active mode, reset the registers.

External interrupts	IRQ3 to IRQ0	Functioning	Functioning	Functioning	Functioning	Func
	WKP5 to WKP0	Functioning	Functioning	Functioning	Functioning	Func
Peripheral functions	Timer A	Functioning	Functioning	Functioning if the timekeeping time-ba function is selected, and retained if no		
	Timer V	Functioning	Functioning	Reset	Reset	Rese
	Timer W	Functioning	Functioning	Retained (if internal clock ϕ is selected as a count clock, the counter is incremented by a subclock*)		Reta
	Watchdog timer	Functioning	Functioning	Retained (functioning if the internal os selected as a count clock*)		
	SCI3	Functioning	Functioning	Reset	Reset	Rese
	IIC	Functioning	Functioning	Retained*	Retained	Reta
	A/D converter	Functioning	Functioning	Reset	Reset	Rese

Note: * Registers can be read or written in subactive mode.

6.2.2 Standby Mode

In standby mode, the clock pulse generator stops, so the CPU and on-chip peripheral mo functioning. However, as long as the rated voltage is supplied, the contents of CPU regis chip RAM, and some on-chip peripheral module registers are retained. On-chip RAM co will be retained as long as the voltage set by the RAM data retention voltage is provided ports go to the high-impedance state.

Standby mode is cleared by an interrupt. When an interrupt is requested, the system cloc generator starts. After the time set in bits STS2–STS0 in SYSCR1 has elapsed, and inter exception handling starts. Standby mode is not cleared if the I bit of CCR is set to 1 or th requested interrupt is disabled in the interrupt enable register.

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

6.2.3 Subsleep Mode

In subsleep mode, operation of the CPU and on-chip peripheral modules other than time halted. As long as a required voltage is applied, the contents of CPU registers, the on-ch and some registers of the on-chip peripheral modules are retained. I/O ports keep the sam as before the transition.

Subsleep mode is cleared by an interrupt. When an interrupt is requested, subsleep mode and interrupt exception handling starts. Subsleep mode is not cleared if the I bit of CCR or the requested interrupt is disabled in the interrupt enable register. After subsleep mod

SA0 bits in SYSCR2. After the SLEEP instruction is executed, the operating frequency c
the frequency which is set before the execution. When the SLEEP instruction is executed
subactive mode, a transition to sleep mode, subsleep mode, standby mode, active mode, c
subactive mode is made, depending on the combination of SYSCR1 and SYSCR2. When
pin goes low, the system clock pulse generator starts. Since system clock signals are supp
the entire chip as soon as the system clock pulse generator starts functioning, the $\overline{\text{RES}}$ pin
kept low until the pulse generator output stabilizes. After the pulse generator output has s
the CPU starts reset exception handling if the $\overline{\text{RES}}$ pin is driven high.

6.3 Operating Frequency in Active Mode

Operation in active mode is clocked at the frequency designated by the MA2, MA1, and M
in SYSCR2. The operating frequency changes to the set frequency after SLEEP instructio
execution.

by means of an interrupt.

6.4.1 Direct Transition from Active Mode to Subactive 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 i
processing states)}× (tcyc before transition) + (number of interrupt exception handling s
(tsubcyc after transition) (1)

Example
Direct transition time = (2 + 1) × tosc + 14 × 8tw = 3tosc + 112tw
(when the CPU operating clock of $\phi_{osc} \rightarrow \phi_w/8$ is selected)

Legend
tosc: OSC clock cycle time
tw: watch clock cycle time
tcyc: system clock (ϕ) cycle time
tsubcyc: subclock (ϕ_{SUB}) cycle time

6.4.2 Direct Transition from Subactive Mode to Active Mode

The time from the start of SLEEP instruction execution to the end of interrupt exception
(the direct transition time) is calculated by equation (2).

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

The module-standby function can be set to any peripheral module. In module standby mo
clock supply to modules stops to enter the power-down mode. Module standby mode ena
on-chip peripheral module to enter the standby state by setting a bit that corresponds to e
module to 1 and cancels the mode by clearing the bit to 0.

— The flash memory can be reprogrammed up to 1,000 times.

- On-board programming
  — On-board programming/erasing can be done in boot mode, in which the boot pro into the chip is started to erase or program of the entire flash memory. In normal program mode, individual blocks can be erased or programmed.
- Programmer mode
  — Flash memory can be programmed/erased in programmer mode using a PROM programmer, as well as in on-board programming mode.
- Automatic bit rate adjustment
  — For data transfer in boot mode, this LSI's bit rate can be automatically adjusted to the transfer bit rate of the host.
- Programming/erasing protection
  — Sets software protection against flash memory programming/erasing.
- Power-down mode
  — Operation of the power supply circuit can be partly halted in subactive mode. As flash memory can be read with low power consumption.

7.1 Block Configuration

Figure 7.1 shows the block configuration of 32-kbyte flash memory. The thick lines ind erasing units, the narrow lines indicate programming units, and the values are addresses. memory is divided into 1 kbyte × 4 blocks and 28 kbytes × 1 block. Erasing is performe units. Programming is performed in 128-byte units starting from an address with lower H'00 or H'80.

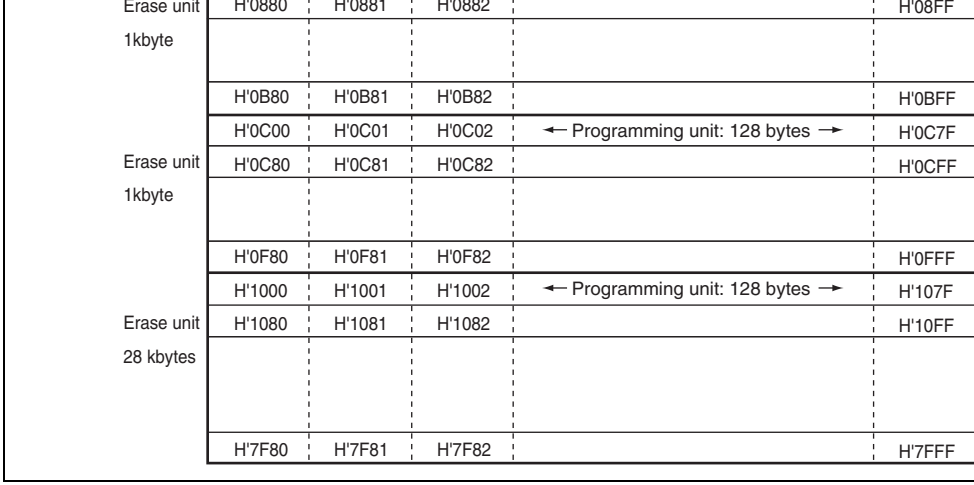


Figure 7.1 Flash Memory Block Configuration

7.2 Register Descriptions

The flash memory has the following registers.

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

6	SWE	0	R/W	Software Write Enable When this bit is set to 1, flash memory programming/erasing 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 char erase setup state. When it is cleared to 0, the e setup state is cancelled. Set this bit to 1 before E bit to 1 in FLMCR1.
4	PSU	0	R/W	Program Setup When this bit is set to 1, the flash memory char program setup state. When it is cleared to 0, th setup state is cancelled. Set this bit to 1 before the P bit in FLMCR1.
3	EV	0	R/W	Erase-Verify When this bit is set to 1, the flash memory char erase-verify mode. When it is cleared to 0, era mode is cancelled.
2	PV	0	R/W	Program-Verify When this bit is set to 1, the flash memory char program-verify mode. When it is cleared to 0, p verify mode is cancelled.
1	E	0	R/W	Erase When this bit is set to 1, and while the SWE=1 ESU=1 bits are 1, the flash memory changes to mode. When it is cleared to 0, erase mode is c

FLMCR2 is a register that displays the state of flash memory programming/erasing. 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 an op on flash memory (programming or erasing). Wh is set to 1, flash memory goes to the error-protec state. See 7.5.3, 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, 28 kbytes of H'1000 to will be erased.
3	EB3	0	R/W	When this bit is set to 1, 1 kbyte of H'0C00 to H be erased.
2	EB2	0	R/W	When this bit is set to 1, 1 kbyte of H'0800 to H be erased.
1	EB1	0	R/W	When this bit is set to 1, 1 kbyte of H'0400 to H be erased.
0	EB0	0	R/W	When this bit is set to 1, 1 kbyte of H'0000 to H be erased.

				When this bit is 0 and a transition is made to sub mode, the flash memory enters the power-down When this bit is 1, the flash memory remains in t normal mode even after a transition is made to s mode.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

7.2.5 Flash Memory Enable Register (FENR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	FLSHE	0	R/W	Flash Memory Control Register Enable Flash memory control registers can be accessed this bit is set to 1. Flash memory control register be accessed when this bit is set to 0.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

via SCI3. After erasing the entire flash memory, the programming control program is ex
This can be used for programming initial values in the on-board state or for a forcible re
programming/erasing can no longer be done in user program mode. In user program mo
individual blocks can be erased and programmed by branching to the user program/erase
program prepared by the user.

Table 7.1 Setting Programming Modes

TEST	$\overline{\text{NMI}}$	P85	PB0	PB1	PB2	LSI State after Reset End
0	1	X	X	X	X	User Mode
0	0	1	X	X	X	Boot Mode
1	X	X	0	0	0	Programmer Mode

Legend: X : Don't care.

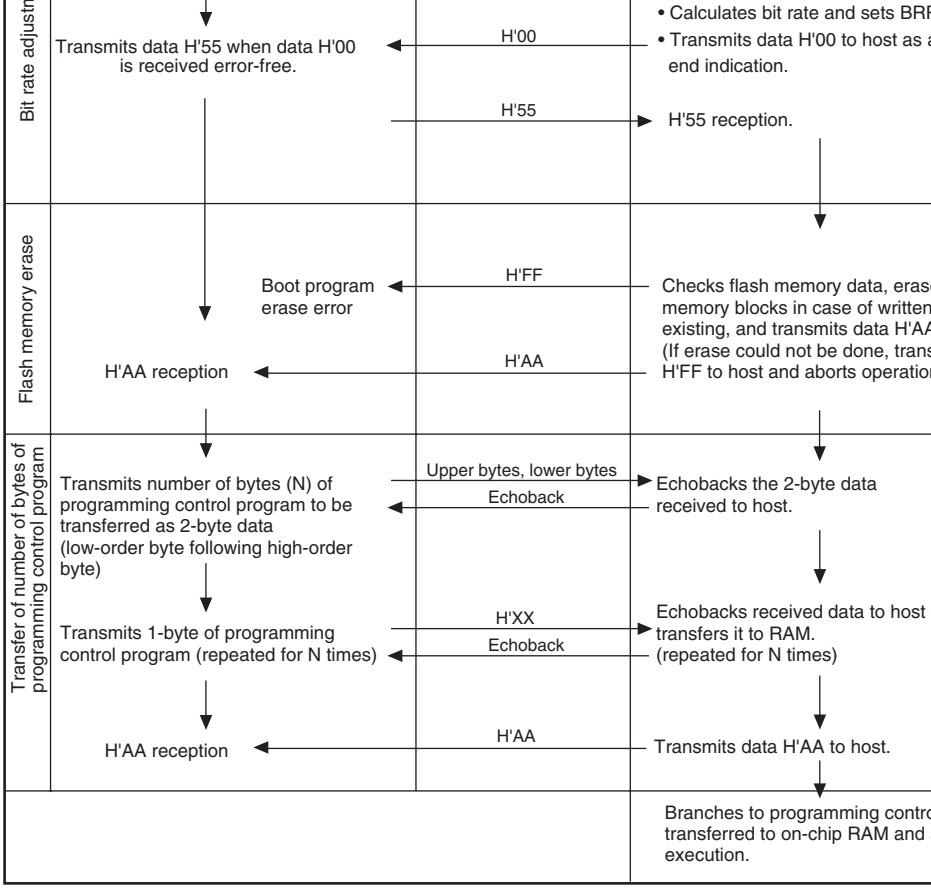
7.3.1 Boot Mode

Table 7.2 shows the boot mode operations between reset end and branching to the progr
control program.

1. When boot mode is used, the flash memory programming control program must be p
the host beforehand. Prepare a programming control program in accordance with the
description in section 7.4, Flash Memory Programming/Erasing.
2. SCI3 should be set to asynchronous mode, and the transfer format as follows: 8-bit d
bit, and no parity.
3. When the boot program is initiated, the chip measures the low-level period of asynch
SCI communication data (H'00) transmitted continuously from the host. The chip the
calculates the bit rate of transmission from the host, and adjusts the SCI3 bit rate to m
of the host. The reset should end with the RxD pin high. The RxD and TxD pins sho

RAMEND is the area to which the programming control program is transferred from the
The boot program area cannot be used until the execution state in boot mode switches
programming control program.

6. Before branching to the programming control program, the chip terminates transfer op
by SCI3 (by clearing the RE and TE bits in SCR to 0), however the adjusted bit rate v
remains set in BRR. Therefore, the programming control program can still use it for t
of write data or verify data with the host. The TxD pin is high (PCR22 = 1, P22 = 1).
contents of the CPU general registers are undefined immediately after branching to th
programming control program. These registers must be initialized at the beginning of
programming control program, as the stack pointer (SP), in particular, is used implicit
subroutine calls, etc.
7. Boot mode can be cleared by a reset. End the reset after driving the reset pin low, wa
least 20 states, and then setting the $\overline{\text{NMI}}$ pin. Boot mode is also cleared when a WDT
occurs.
8. Do not change the TEST pin and NMI pin input levels in boot mode.


Bit rate adjustm
Transmits data H'55 when data H'00 is received error-free.
H'00
H'55
• Calculates bit rate and sets BRF
• Transmits data H'00 to host as a end indication.
H'55 reception.
Flash memory erase
Boot program erase error
H'FF
H'AA reception
H'AA
Checks flash memory data, erase memory blocks in case of written existing, and transmits data H'AA (If erase could not be done, trans H'FF to host and aborts operatio
Transfer of number of bytes of programming control program
Transmits number of bytes (N) of programming control program to be transferred as 2-byte data (low-order byte following high-order byte)
Upper bytes, lower bytes
Echoback
Echobacks the 2-byte data received to host.
Transmits 1-byte of programming control program (repeated for N times)
H'XX
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.

7.3.2 Programming/Erasing in User Program Mode

On-board programming/erasing of an individual flash memory block can also be perform program mode by branching to a user program/erase control program. The user must set b conditions and provide on-board means of supplying programming data. The flash memo contain the user program/erase control program or a program that provides the user progr control program from external memory. As the flash memory itself cannot be read during programming/erasing, transfer the user program/erase control program to on-chip RAM, mode. Figure 7.2 shows a sample procedure for programming/erasing in user program mo Prepare a user program/erase control program in accordance with the description in sectio Flash Memory Programming/Erasing.

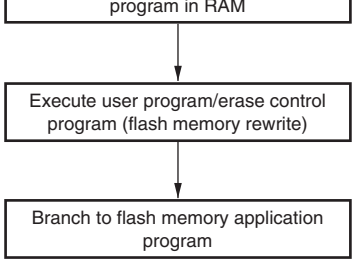


Figure 7.2 Programming/Erasing Flowchart Example in User Program Mo

7.4.1 Program/Program-Verify

When writing data or programs to the flash memory, the program/program-verify flowcha in figure 7.3 should be followed. Performing programming operations according to this f will enable data or programs to be written to the flash memory without subjecting the chi voltage stress or sacrificing program data reliability.

1. Programming must be done to an empty address. Do not reprogram an address to wh programming has already been performed.
2. Programming should be carried out 128 bytes at a time. A 128-byte data transfer mus performed even if writing fewer than 128 bytes. In this case, H'FF data must be writt extra addresses.
3. Prepare the following data storage areas in RAM: A 128-byte programming data area byte reprogramming data area, and a 128-byte additional-programming data area. Per reprogramming data computation according to table 7.4, and additional programming computation according to table 7.5.
4. Consecutively transfer 128 bytes of data in byte units from the reprogramming data ar additional-programming data area to the flash memory. The program address and 128 data are latched in the flash memory. The lower 8 bits of the start address in the flash 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 7.6 shows allowable programming times.
6. The watchdog timer (WDT) is set to prevent overprogramming due to program runaw An overflow cycle of approximately 6.6 ms is allowed.
7. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lo are B'00. Verify data can be read in words or in longwords from the address to which dummy write was performed.

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

Figure 7.3 Program/Program-Verify Flowchart

Reprogram Data	Verify Data	Additional-Program Data	Comments
0	0	0	Additional-program l
0	1	1	No additional progra
1	0	1	No additional progra
1	1	1	No additional progra

Table 7.6 Programming Time

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

Note: Time shown in µs.

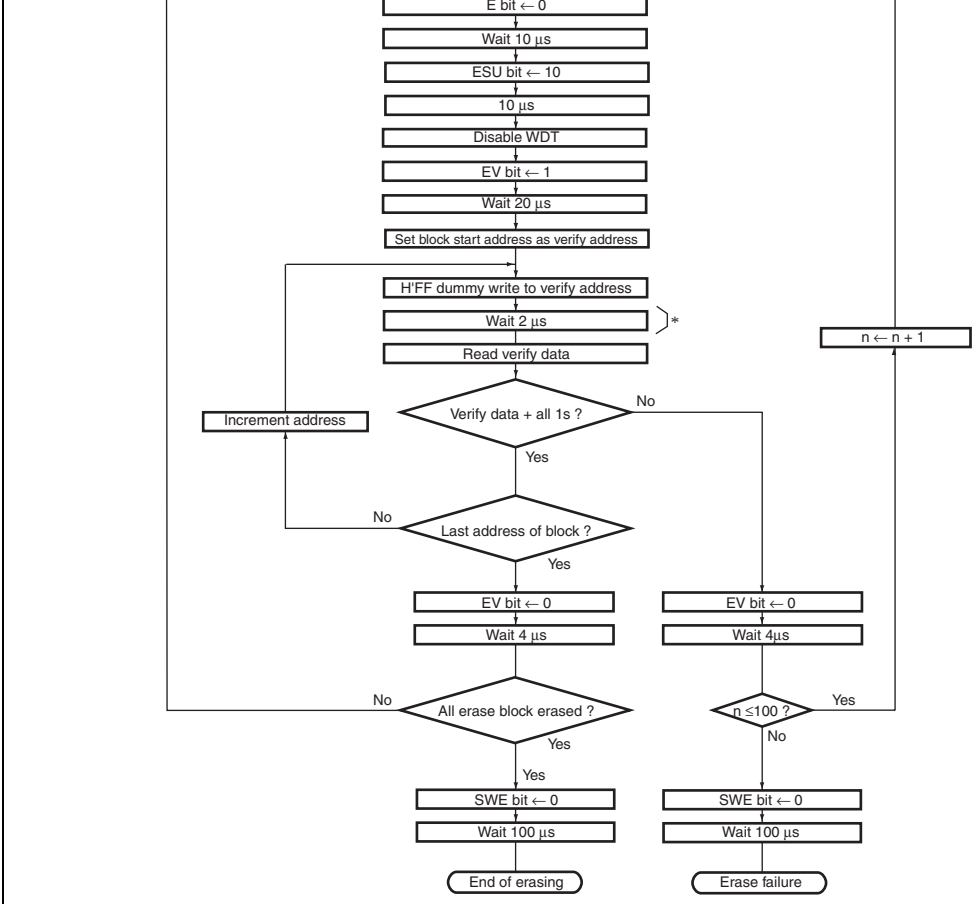
7.4.2 Erase/Erase-Verify

When erasing flash memory, the erase/erase-verify flowchart shown in figure 7.4 should
followed.

1. Prewriting (setting erase block data to all 0s) is not necessary.
2. Erasing is performed in block units. Make only a single-bit specification in the erase
   register (EBR1). To erase multiple blocks, each block must be erased in turn.
3. The time during which the E bit is set to 1 is the flash memory erase time.
4. The watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc
   overflow cycle of approximately 19.8 ms is allowed.

or erased, or while the boot program is executing, for the following three reasons.

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



Note: *The RTS instruction must not be used during a period between dummy writing of H'FF to a verify address and verify data readi

Figure 7.4 Erase/Erase-Verify Flowchart

entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In t
a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the
Characteristics section.

7.5.2 Software Protection

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

7.5.3 Error Protection

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

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

- When the flash memory of the relevant address area is read during programming/era
(including vector read and instruction fetch)
- Immediately after exception handling excluding a reset during programming/erasing
- When a SLEEP instruction is executed during programming/erasing

The FLMCR1, FLMCR2, and EBR1 settings are retained, however program mode or er
is aborted at the point at which the error occurred. Program mode or erase mode cannot

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

- Normal operating mode

  The flash memory can be read and written to at high speed.
- Power-down operating mode

  The power supply circuit of flash memory can be partly halted. As a result, flash mem
  be read with low power consumption.
- Standby mode

  All flash memory circuits are halted.

Table 7.7 shows the correspondence between the operating modes of this LSI and the flas
memory. In subactive mode, the flash memory can be set to operate in power-down mode
PDWND bit in FLPWCR. When the flash memory returns to its normal operating state fr
power-down mode or standby mode, a period to stabilize operation of the power supply c
that were stopped is needed. When the flash memory returns to its normal operating state
STS2 to STS0 in SYSCR1 must be set to provide a wait time of at least 20 µs, even when
external clock is being used.

		H8/3692	512 kbytes	H'FD80 to H'FF7F
		H8/3691	512 kbytes	H'FD80 to H'FF7F
		H8/3690	512 kbytes	H'FD80 to H'FF7F
EEPROM stacked version	Flash memory version	H8/3694N	2 kbytes	H'F780 to H'FF7F*
	Mask-ROM version		1 kbyte	H'FB80 to H'FF7F

Note: * When the E7 or E8 is used, area H'F780 to H'FB7F must not be accessed.

storing output data and can select inputs/outputs in bit units. For functions in each port, s
appendix B.1, I/O Port Block Diagrams. For the execution of bit manipulation instructio
port control register and port data register, see section 2.8.3, Bit Manipulation Instructio

9.1 Port 1

Port 1 is a general I/O port also functioning as IRQ interrupt input pins, a timer A outpu
a timer V input pin. Figure 9.1 shows its pin configuration.

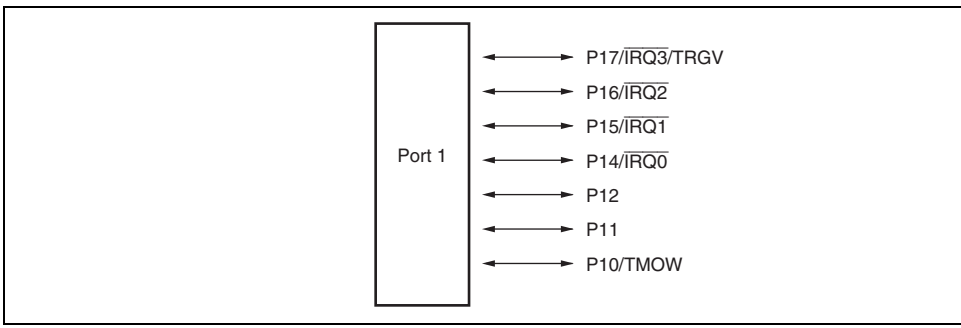


Figure 9.1 Port 1 Pin Configuration

Port 1 has the following registers.

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

6	IRQ2	0	R/W	P16/$\overline{\text{IRQ2}}$ Pin Function Switch This bit selects whether pin P16/$\overline{\text{IRQ2}}$ is used as as $\overline{\text{IRQ2}}$. 0: General I/O port 1: $\overline{\text{IRQ2}}$ input pin
5	IRQ1	0	R/W	P15/$\overline{\text{IRQ1}}$ Pin Function Switch This bit selects whether pin P15/$\overline{\text{IRQ1}}$ is used as as $\overline{\text{IRQ1}}$. 0: General I/O port 1: $\overline{\text{IRQ1}}$ input pin
4	IRQ0	0	R/W	P14/$\overline{\text{IRQ0}}$ Pin Function Switch This bit selects whether pin P14/$\overline{\text{IRQ0}}$ is used as as $\overline{\text{IRQ0}}$. 0: General I/O port 1: $\overline{\text{IRQ0}}$ input pin
3, 2	—	All 1	—	Reserved These bits are always read as 1.
1	TXD	0	R/W	P22/TXD Pin Function Switch This bit selects whether pin P22/TXD is used as as TXD. 0: General I/O port 1: TXD output pin
0	TMOW	0	R/W	P10/TMOW Pin Function Switch This bit selects whether pin P10/TMOW is used as TMOW. 0: General I/O port 1: TMOW output pin

2	PCR12	0	W
1	PCR11	0	W
0	PCR10	0	W

9.1.3 Port Data Register 1 (PDR1)

PDR1 is a general I/O port data register of port 1.

Bit	Bit Name	Initial Value	R/W	Description
7	P17	0	R/W	PDR1 stores output data for port 1 pins.
6	P16	0	R/W	If PDR1 is read while PCR1 bits are set to 1, th
5	P15	0	R/W	stored in PDR1 are read. If PDR1 is read while
4	P14	0	R/W	are cleared to 0, the pin states are read regardl value stored in PDR1.
3	—	1	—	Bit 3 is a reserved bit. This bit is always read as
2	P12	0	R/W	
1	P11	0	R/W	
0	P10	0	R/W	

3	—	1	—
2	PUCR12	0	R/W
1	PUCR11	0	R/W
0	PUCR10	0	R/W

9.1.5 Pin Functions

The correspondence between the register specification and the port functions is shown be

P17/$\overline{\text{IRQ3}}$/TRGV pin

Register	PMR1	PCR1	
Bit Name	**IRQ3**	**PCR17**	**Pin Function**
Setting value	0	0	P17 input pin
		1	P17 output pin
	1	X	$\overline{\text{IRQ3}}$ input/TRGV input pin

Legend: X: Don't care.

P16/$\overline{\text{IRQ2}}$ pin

Register	PMR1	PCR1	
Bit Name	**IRQ2**	**PCR16**	**Pin Function**
Setting value	0	0	P16 input pin
		1	P16 output pin
	1	X	$\overline{\text{IRQ2}}$ input pin

Legend: X: Don't care.

Register	PMR1	PCR1	
Bit Name	**IRQ0**	**PCR14**	**Pin Function**
Setting value	0	0	P14 input pin
		1	P14 output pin
	1	X	$\overline{\text{IRQ0}}$ input pin

Legend: X: Don't care.

P12 pin

Register	PCR1	
Bit Name	**PCR12**	**Pin Function**
Setting value	0	P12 input pin
	1	P12 output pin

P11 pin

Register	PCR1	
Bit Name	**PCR11**	**Pin Function**
Setting value	0	P11 input pin
	1	P11 output pin

9.2 Port 2

Port 2 is a general I/O port also functioning as a SCI3 I/O pin. Each pin of the port 2 is sh
figure 9.2. The register settings of PMR1 and SCI3 have priority for functions of the pins
uses.

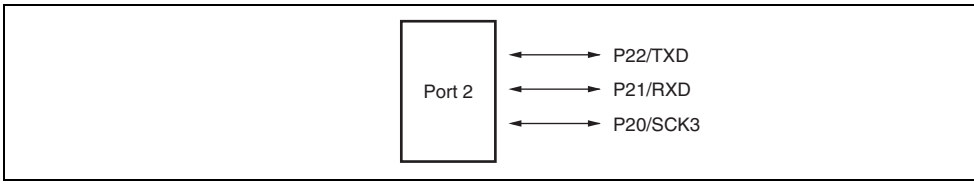


Figure 9.2 Port 2 Pin Configuration

Port 2 has the following registers.

- Port control register 2 (PCR2)
- Port data register 2 (PDR2)

9.2.2 Port Data Register 2 (PDR2)

PDR2 is a general I/O port data register of port 2.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 1	—	Reserved These bits are always read as 1.
2	P22	0	R/W	PDR2 stores output data for port 2 pins.
1	P21	0	R/W	If PDR2 is read while PCR2 bits are set to 1, th
0	P20	0	R/W	stored in PDR2 is read. If PDR2 is read while P are cleared to 0, the pin states are read regardl value stored in PDR2.

		1	P22 output pin
	1	X	TXD output pin

Legend: X: Don't care.

P21/RXD pin

Register	SCR3	PCR2	
Bit Name	**RE**	**PCR21**	**Pin Function**
Setting Value	0	0	P21 input pin
		1	P21 output pin
	1	X	RXD input pin

Legend: X: Don't care.

P20/SCK3 pin

Register	SCR3		SMR	PCR2	
Bit Name	**CKE1**	**CKE0**	**COM**	**PCR20**	**Pin Function**
Setting Value	0	0	0	0	P20 input pin
				1	P20 output pin
	0	0	1	X	SCK3 output pin
	0	1	X	X	SCK3 output pin
	1	X	X	X	SCK3 input pin

Legend: X: Don't care.

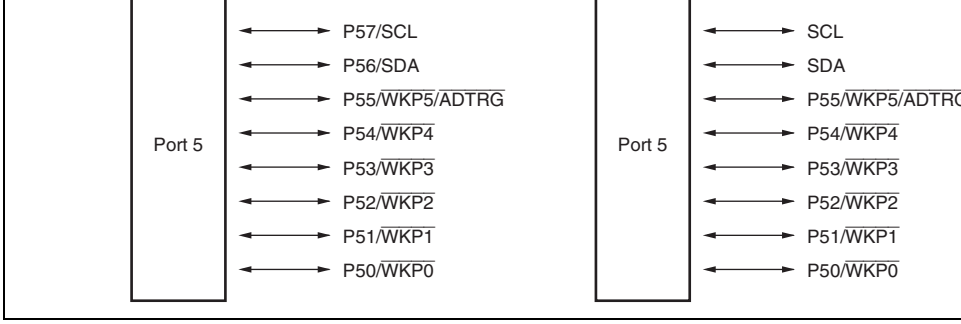


Figure 9.3 Port 5 Pin Configuration

Port 5 has the following registers.

- Port mode register 5 (PMR5)
- Port control register 5 (PCR5)
- Port data register 5 (PDR5)
- Port pull-up control register 5 (PUCR5)

				0: General I/O port 1: $\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ input pin
4	WKP4	0	R/W	P54/$\overline{\text{WKP4}}$ Pin Function Switch Selects whether pin P54/$\overline{\text{WKP4}}$ is used as P54 o $\overline{\text{WKP4}}$. 0: General I/O port 1: $\overline{\text{WKP4}}$ input pin
3	WKP3	0	R/W	P53/$\overline{\text{WKP3}}$ Pin Function Switch Selects whether pin P53/$\overline{\text{WKP3}}$ is used as P53 o $\overline{\text{WKP3}}$. 0: General I/O port 1: $\overline{\text{WKP3}}$ input pin
2	WKP2	0	R/W	P52/$\overline{\text{WKP2}}$ Pin Function Switch Selects whether pin P52/$\overline{\text{WKP2}}$ is used as P52 o $\overline{\text{WKP2}}$. 0: General I/O port 1: $\overline{\text{WKP2}}$ input pin
1	WKP1	0	R/W	P51/$\overline{\text{WKP1}}$ Pin Function Switch Selects whether pin P51/$\overline{\text{WKP1}}$ is used as P51 o $\overline{\text{WKP1}}$. 0: General I/O port 1: $\overline{\text{WKP1}}$ input pin
0	WKP0	0	R/W	P50/$\overline{\text{WKP0}}$ Pin Function Switch Selects whether pin P50/$\overline{\text{WKP0}}$ is used as P50 o $\overline{\text{WKP0}}$. 0: General I/O port 1: $\overline{\text{WKP0}}$ input pin

2	PCR52	0	W	
1	PCR51	0	W	
0	PCR50	0	W	

9.3.3 Port Data Register 5 (PDR5)

PDR5 is a general I/O port data register of port 5.

Bit	Bit Name	Initial Value	R/W	Description
7	P57	0	R/W	Stores output data for port 5 pins.
6	P56	0	R/W	If PDR5 is read while PCR5 bits are set to 1, th
5	P55	0	R/W	stored in PDR5 are read. If PDR5 is read while
4	P54	0	R/W	are cleared to 0, the pin states are read regardl value stored in PDR5.
3	P53	0	R/W	Note: The P57 and P56 bits should not be set t
2	P52	0	R/W	H8/3694N.
1	P51	0	R/W	
0	P50	0	R/W	

3	PUCR53	0	R/W	these bits are cleared to 0.
2	PUCR52	0	R/W	
1	PUCR51	0	R/W	
0	PUCR50	0	R/W	

9.3.5 Pin Functions

The correspondence between the register specification and the port functions is shown be

P57/SCL pin

Register	ICCR1	PCR5	
Bit Name	**ICE**	**PCR57**	**Pin Function**
Setting Value	0	0	P57 input pin
		1	P57 output pin
	1	X	SCL I/O pin

Legend: X: Don't care.

SCL performs the NMOS open-drain output, that enables a direct bus drive.

SDA performs the NMOS open-drain output, that enables a direct bus drive.

P55/$\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ pin

Register	PMR5	PCR5	
Bit Name	**WKP5**	**PCR55**	**Pin Function**
Setting Value	0	0	P55 input pin
		1	P55 output pin
	1	X	$\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ input pin

Legend: X: Don't care.

P54/$\overline{\text{WKP4}}$ pin

Register	PMR5	PCR5	
Bit Name	**WKP4**	**PCR54**	**Pin Function**
Setting Value	0	0	P54 input pin
		1	P54 output pin
	1	X	$\overline{\text{WKP4}}$ input pin

Legend: X: Don't care.

P52/WKP2 pin

Register	PMR5	PCR5	
Bit Name	**WKP2**	**PCR52**	**Pin Function**
Setting Value	0	0	P52 input pin
		1	P52 output pin
	1	X	$\overline{\text{WKP2}}$ input pin

Legend: X: Don't care.

P51/WKP1 pin

Register	PMR5	PCR5	
Bit Name	**WKP1**	**PCR51**	**Pin Function**
Setting Value	0	0	P51 input pin
		1	P51 output pin
	1	X	$\overline{\text{WKP1}}$ input pin

Legend: X: Don't care.

9.4 Port 7

Port 7 is a general I/O port also functioning as a timer V I/O pin. Each pin of the port 7 i
in figure 9.4. The register setting of TCSRV in timer V has priority for functions of pin
P76/TMOV. The pins, P75/TMCIV and P74/TMRIV, are also functioning as timer V in
that are connected to the timer V regardless of the register setting of port 7.

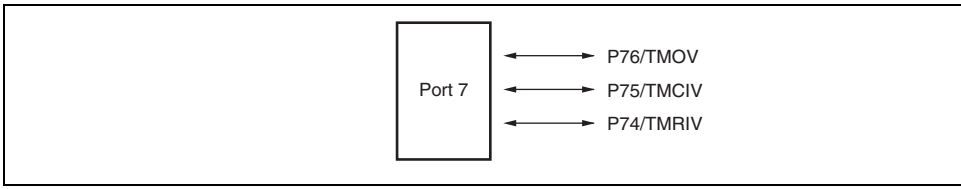


Figure 9.4 Port 7 Pin Configuration

Port 7 has the following registers.

- Port control register 7 (PCR7)
- Port data register 7 (PDR7)

				P76/TMOV pin.
3 to 0	—	—	—	Reserved

9.4.2 Port Data Register 7 (PDR7)

PDR7 is a general I/O port data register of port 7.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1.
6	P76	0	R/W	PDR7 stores output data for port 7 pins.
5	P75	0	R/W	If PDR7 is read while PCR7 bits are set to 1, the
4	P74	0	R/W	stored in PDR7 is read. If PDR7 is read while PC are cleared to 0, the pin states are read regardle value stored in PDR7.
3 to 0	—	All 1	—	Reserved These bits are always read as 1.

		P76 output pin
	Other than the above values	X	TMOV output pin

Legend: X: Don't care.

P75/TMCIV pin

Register	PCR7	
Bit Name	**PCR75**	**Pin Function**
Setting Value	0	P75 input/TMCIV input pin
	1	P75 output/TMCIV input pin

P74/TMRIV pin

Register	PCR7	
Bit Name	**PCR74**	**Pin Function**
Setting Value	0	P74 input/TMRIV input pin
	1	P74 output/TMRIV input pin

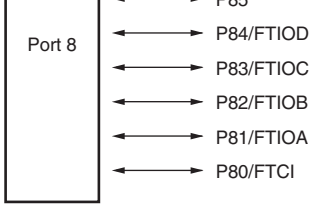


Figure 9.5 Port 8 Pin Configuration

Port 8 has the following registers.

- Port control register 8 (PCR8)
- Port data register 8 (PDR8)

9.5.1 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	PCR87	0	W	When each of the port 8 pins P87 to P80 functio
6	PCR86	0	W	general I/O port, setting a PCR8 bit to 1 makes t
5	PCR85	0	W	corresponding pin an output port, while clearing
4	PCR84	0	W	0 makes the pin an input port.
3	PCR83	0	W	
2	PCR82	0	W	
1	PCR81	0	W	
0	PCR80	0	W	

3	P83	0	R/W
2	P82	0	R/W
1	P81	0	R/W
0	P80	0	R/W

9.5.3 Pin Functions

The correspondence between the register specification and the port functions is shown b

P87 pin

Register	PCR8	
Bit Name	**PCR87**	**Pin Function**
Setting Value	0	P87 input pin
	1	P87 output pin

P86 pin

Register	PCR8	
Bit Name	**PCR86**	**Pin Function**
Setting Value	0	P86 input pin
	1	P86 output pin

Register	TIOR1			PCR8	
Bit Name	**IOD2**	**IOD1**	**IOD0**	**PCR84**	**Pin Function**
Setting Value	0	0	0	0	P84 input/FTIOD input pin
				1	P84 output/FTIOD input pin
	0	0	1	X	FTIOD output pin
	0	1	X	X	FTIOD output pin
	1	X	X	0	P84 input/FTIOD input pin
				1	P84 output/FTIOD input pin

Legend: X: Don't care.

P83/FTIOC pin

Register	TIOR1			PCR8	
Bit Name	**IOC2**	**IOC1**	**IOC0**	**PCR83**	**Pin Function**
Setting Value	0	0	0	0	P83 input/FTIOC input pin
				1	P83 output/FTIOC input pin
	0	0	1	X	FTIOC output pin
	0	1	X	X	FTIOC output pin
	1	X	X	0	P83 input/FTIOC input pin
				1	P83 output/FTIOC input pin

Legend: X: Don't care.

				1	P82 output/FTIOB input pin

Legend: X: Don't care.

P81/FTIOA pin

Register	TIOR0			PCR8	
Bit Name	**IOA2**	**IOA1**	**IOA0**	**PCR81**	**Pin Function**
Setting Value	0	0	0	0	P81 input/FTIOA input pin
				1	P81 output/FTIOA input pin
	0	0	1	X	FTIOA output pin
	0	1	X	X	FTIOA output pin
	1	X	X	0	P81 input/FTIOA input pin
				1	P81 output/FTIOA input pin

Legend: X: Don't care.

P80/FTCI pin

Register	PCR8	
Bit Name	**PCR80**	**Pin Function**
Setting Value	0	P80 input/FTCI input pin
	1	P80 output/FTCI input pin

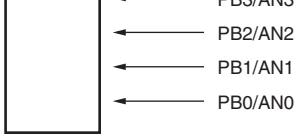


Figure 9.6 Port B Pin Configuration

Port B has the following register.

- Port data register B (PDRB)

9.6.1 Port Data Register B (PDRB)

PDRB is a general input-only port data register of port B.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7	—	R	The input value of each pin is read by reading th register. However, if a port B pin is designated as an anal channel by ADCSR in A/D converter, 0 is read.
6	PB6	—	R	
5	PB5	—	R	
4	PB4	—	R	
3	PB3	—	R	
2	PB2	—	R	
1	PB1	—	R	
0	PB0	—	R	

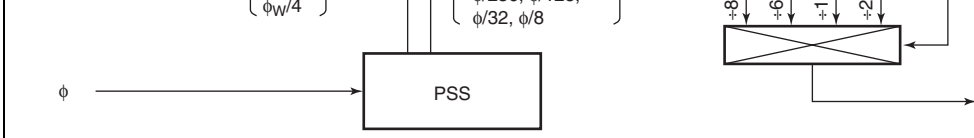
- Any of eight clock signals can be output from pin TMOW: 32.768 kHz divided by 3
  4 (1 kHz, 2 kHz, 4 kHz, 8 kHz), or the system clock divided by 32, 16, 8, or 4.

Interval Timer

- Choice of eight internal clock sources (ϕ/8192, ϕ/4096, ϕ/2048, ϕ/512, ϕ/256, ϕ/128,

Clock Time Base

- Choice of four overflow periods (1 s, 0.5 s, 0.25 s, 31.25 ms) when timer A is used a
  time base (using a 32.768 kHz crystal oscillator).



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

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

Figure 10.1 Block Diagram of Timer A

10.2 Input/Output Pins

Table 10.1 shows the timer A input/output pin.

Table 10.1 Pin Configuration

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

Bit	Bit Name	Initial Value	R/W	Description
7	TMA7	0	R/W	Clock Output Select 7 to 5
6	TMA6	0	R/W	These bits select the clock output at the TMOW
5	TMA5	0	R/W	000: ϕ/32
				001: ϕ/16
				010: ϕ/8
				011: ϕ/4
				100: ϕ_w/32
				101: ϕ_w/16
				110: ϕ_w/8
				111: ϕ_w/4
				For details on clock outputs, see section 10.4.3 Output.
4	—	1	—	Reserved
				This bit is always read as 1.
3	TMA3	0	R/W	Internal Clock Select 3
				This bit selects the operating mode of the timer
				0: Functions as an interval timer to count the ou prescaler S.
				1: Functions as a clock-time base to count the prescaler W.

	110: ϕ/32 111: ϕ/8
	These bits select the overflow period when TMA (when a 32.768 kHz crystal oscillator with is use 000: 1s 001: 0.5 s 010: 0.25 s 011: 0.03125 s 1XX: Both PSW and TCA are reset

Legend: X: Don't care.

10.3.2 Timer Counter A (TCA)

TCA is an 8-bit readable up-counter, which is incremented by internal clock input. The cl source for input to this counter is selected by bits TMA3 to TMA0 in TMA. TCA values read by the CPU in active mode, but cannot be read in subactive mode. When TCA overf IRRTA bit in interrupt request register 1 (IRR1) is set to 1. TCA is cleared by setting bits and TMA2 in TMA to B’11. TCA is initialized to H'00.

After the count value in TCA reaches H'FF, the next clock signal input causes timer A to
overflow, setting bit IRRTA to 1 in interrupt Flag Register 1 (IRR1). If IENTA = 1 in int
enable register 1 (IENR1), a CPU interrupt is requested. At overflow, TCA returns to H'
starts counting up again. In this mode timer A functions as an interval timer that generat
overflow output at intervals of 256 input clock pulses.

10.4.2 Clock Time Base Operation

When bit TMA3 in TMA is set to 1, timer A functions as a clock-timer base by counting
signals output by prescaler W. When a clock signal is input after the TCA counter value
become H'FF, timer A overflows and IRRTA in IRR1 is set to 1. At that time, an interru
is generated to the CPU if IENTA in the interrupt enable register 1 (IENR1) is 1. The ov
period of timer A is set by bits TMA1 and TMA0 in TMA. A choice of four periods is a
In clock time base operation (TMA3 = 1), setting bit TMA2 to 1 clears both TCA and pr
to H'00.

10.4.3 Clock Output

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

10.5 Usage Note

When the clock time base function is selected as the internal clock of TCA in active mo
mode, the internal clock is not synchronous with the system clock, so it is synchronized
synchronizing circuit. This may result in a maximum error of 1/ϕ (s) in the count cycle.

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

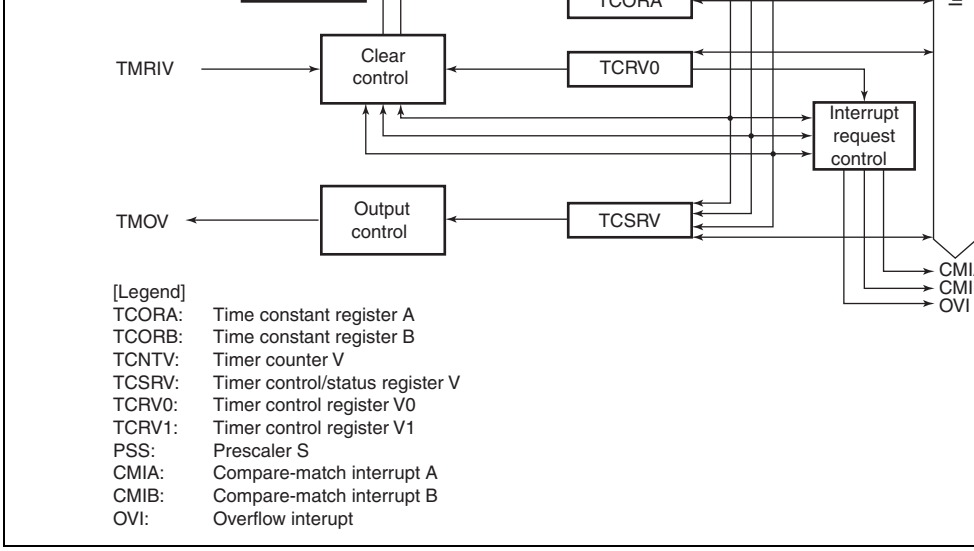


Figure 11.1 Block Diagram of Timer V

11.2 Input/Output Pins

Table 11.1 shows the timer V pin configuration.

Table 11.1 Pin Configuration

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

11.3.1 Timer Counter V (TCNTV)

TCNTV is an 8-bit up-counter. The clock source is selected by bits CKS2 to CKS0 in ti
control register V0 (TCRV0). The TCNTV value can be read and written by the CPU at
TCNTV can be cleared by an external reset input signal, or by compare match A or B. T
clearing signal is selected by bits CCLR1 and CCLR0 in TCRV0.

When TCNTV overflows, OVF is set to 1 in timer control/status register V (TCSRV).

TCNTV is initialized to H'00.

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

TCORA and TCORB have the same function.

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

TCORA and TCNTV are compared at all times. When the TCORA and TCNTV conten
CMFA is set to 1 in TCSRV. If CMIEA is also set to 1 in TCRV0, a CPU interrupt is re
Note that they must not be compared during the T3 state of a TCORA write cycle.

Timer output from the TMOV pin can be controlled by the identifying signal (compare m
and the settings of bits OS3 to OS0 in TCSRV.

TCORA and TCORB are initialized to H'FF.

6	CMIEA	0	R/W	Compare Match Interrupt Enable A When this bit is set to 1, interrupt request from th bit in TCSRV is enabled.
5	OVIE	0	R/W	Timer Overflow Interrupt Enable When this bit is set to 1, interrupt request from th bit in TCSRV is enabled.
4	CCLR1	0	R/W	Counter Clear 1 and 0
3	CCLR0	0	R/W	These bits specify the clearing conditions of TCN 00: Clearing is disabled 01: Cleared by compare match A 10: Cleared by compare match B 11: Cleared on the rising edge of the TMRIV pin. operation of TCNTV after clearing depends o in TCRV1.
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	These bits select clock signals to input to TCNTV
0	CKS0	0	R/W	counting condition in combination with ICKS0 in Refer to table 11.2.

		1	0	Internal clock: counts on ϕ/64, falling
			1	Internal clock: counts on ϕ/128, falling
1	0	0	—	Clock input prohibited
		1	—	External clock: counts on rising edge
	1	0	—	External clock: counts on falling edge
		1	—	External clock: counts on rising and fa

				After reading CMFB = 1, cleared by writing 0 to C
6	CMFA	0	R/W	Compare Match Flag A Setting condition: When the TCNTV value matches the TCORA va Clearing condition: After reading CMFA = 1, cleared by writing 0 to C
5	OVF	0	R/W	Timer Overflow Flag Setting condition: When TCNTV overflows from H'FF to H'00 Clearing condition: After reading OVF = 1, cleared by writing 0 to OV
4	—	1	—	Reserved This bit is always read as 1.
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits select an output method for the TMOV the compare match of TCORB and TCNTV. 00: No change 01: 0 output 10: 1 output 11: Output toggles
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits select an output method for the TMOV the compare match of TCORA and TCNTV. 00: No change 01: 0 output 10: 1 output 11: Output toggles

7 to 5	—	All 1	—	Reserved These bits are always read as 1.
4	TVEG1	0	R/W	TRGV Input Edge Select
3	TVEG0	0	R/W	These bits select the TRGV input edge. 00: TRGV trigger input is prohibited 01: Rising edge is selected 10: Falling edge is selected 11: Rising and falling edges are both selected
2	TRGE	0	R/W	TCNT starts counting up by the input of the edg selected by TVEG1 and TVEG0. 0: Disables starting counting-up TCNTV by the the TRGV pin and halting counting-up TCNT TCNTV is cleared by a compare match. 1: Enables starting counting-up TCNTV by the i the TRGV pin and halting counting-up TCNT TCNTV is cleared by a compare match.
1	—	1	—	Reserved This bit is always read as 1.
0	ICKS0	0	R/W	Internal Clock Select 0 This bit selects clock signals to input to TCNTV combination with CKS2 to CKS0 in TCRV0. Refer to table 11.2.

will be set. The timing at this time is shown in figure 11.4. An interrupt request is sen
CPU when OVIE in TCRV0 is 1.

3. TCNTV is constantly compared with TCORA and TCORB. Compare match flag A o
(CMFA or CMFB) is set to 1 when TCNTV matches TCORA or TCORB, respectivel
compare-match signal is generated in the last state in which the values match. Figure
shows the timing. An interrupt request is generated for the CPU when CMIEA or CM
TCRV0 is 1.
4. When a compare match A or B is generated, the TMOV responds with the output valu
selected by bits OS3 to OS0 in TCSRV. Figure 11.6 shows the timing when the outpu
toggled by compare match A.
5. When CCLR1 or CCLR0 in TCRV0 is 01 or 10, TCNTV can be cleared by the corres
compare match. Figure 11.7 shows the timing.
6. When CCLR1 or CCLR0 in TCRV0 is 11, TCNTV can be cleared by the rising edge
input of TMRIV pin. A TMRIV input pulse-width of at least 1.5 system clocks is nec
Figure 11.8 shows the timing.
7. When a counter-clearing source is generated with TRGE in TCRV1 set to 1, the coun
halted as soon as TCNTV is cleared. TCNTV resumes counting-up when the edge sel
TVEG1 or TVEG0 in TCRV1 is input from the TGRV pin.

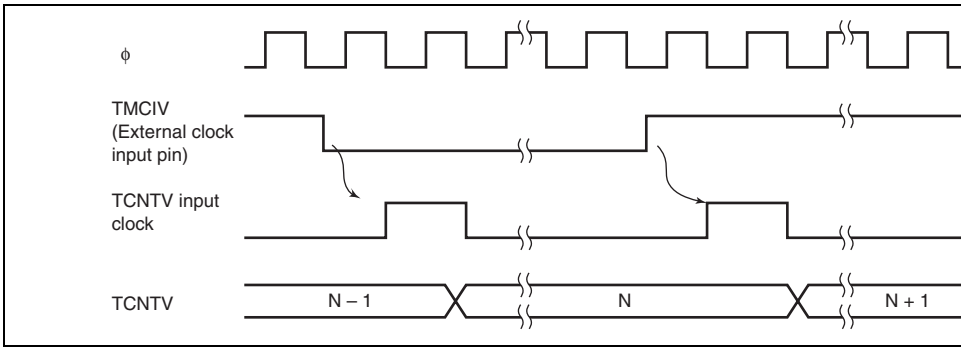


Figure 11.3 Increment Timing with External Clock

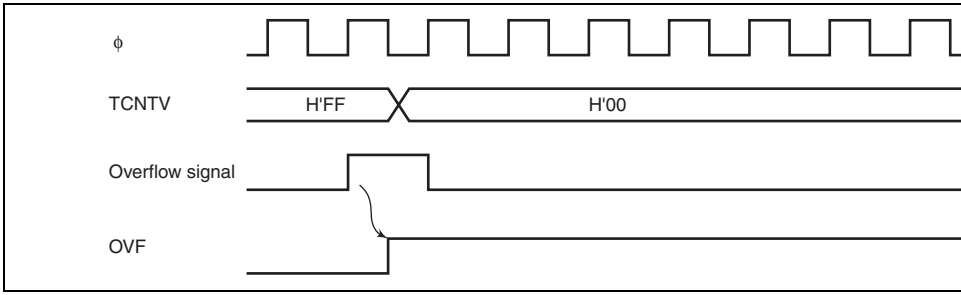


Figure 11.4 OVF Set Timing

Figure 11.5 CMFA and CMFB Set Timing

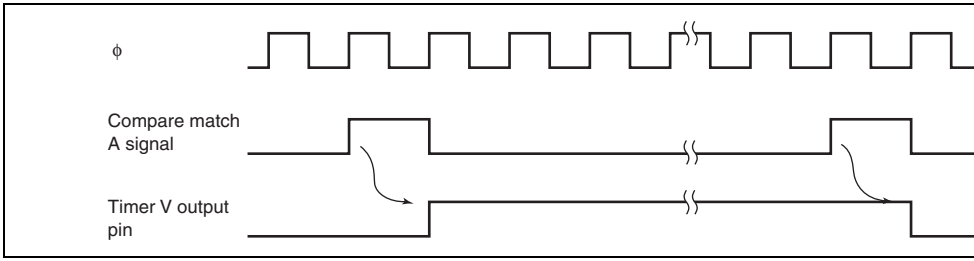


Figure 11.6 TMOV Output Timing

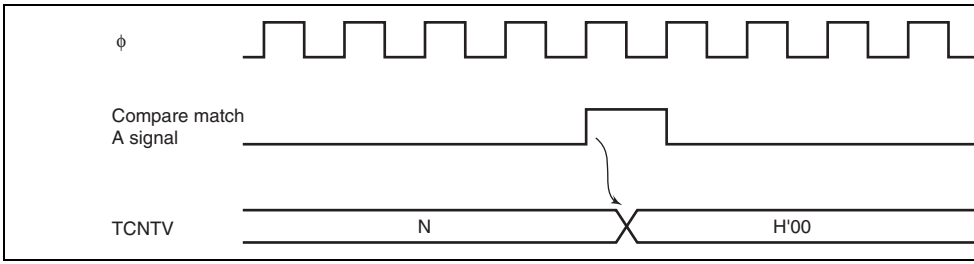


Figure 11.7 Clear Timing by Compare Match

3. Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired cloc
4. With these settings, a waveform is output without further software intervention, with a
determined by TCORA and a pulse width determined by TCORB.

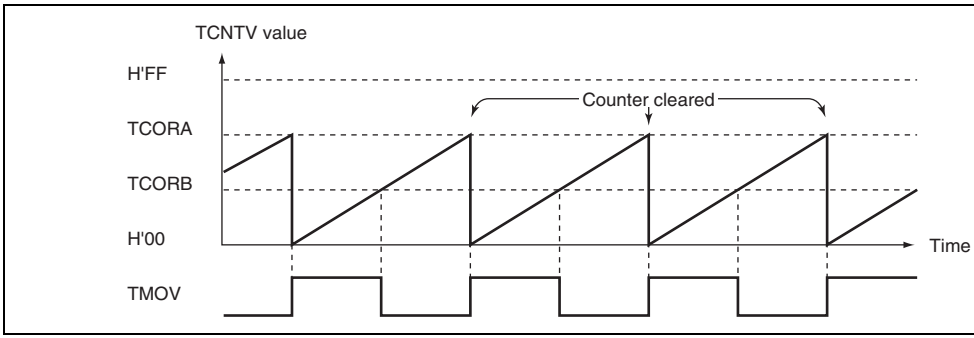


Figure 11.9 Pulse Output Example

input.

4. Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clo
5. After these settings, a pulse waveform will be output without further software interv with a delay determined by TCORA from the TRGV input, and a pulse width determ (TCORB – TCORA).

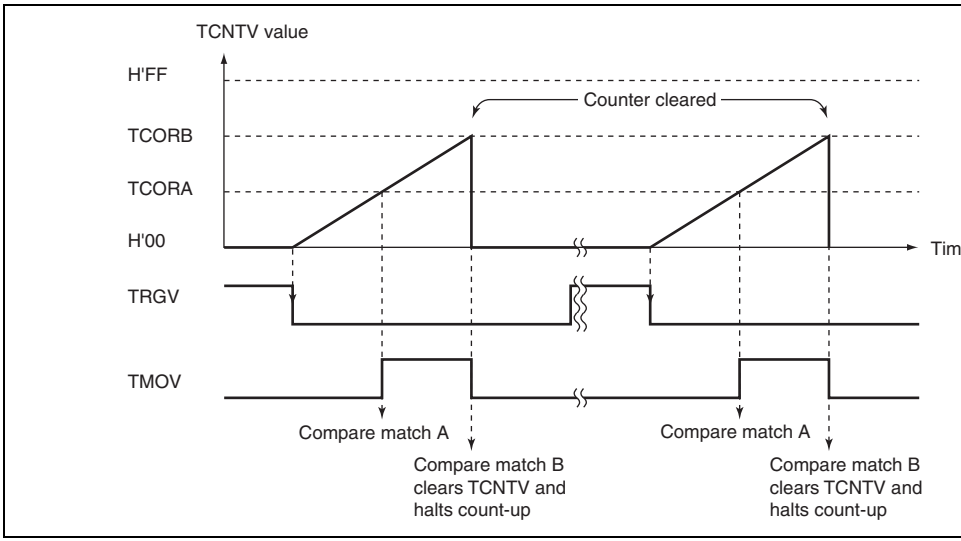


Figure 11.10 Example of Pulse Output Synchronized to TRGV Input

3. If compare matches A and B occur simultaneously, any conflict between the output s for compare match A and compare match B is resolved by the following priority: tog output > output 1 > output 0.
4. Depending on the timing, TCNTV may be incremented by a switch between differen clock sources. When TCNTV is internally clocked, an increment pulse is generated f falling edge of an internal clock signal, that is divided system clock (ϕ). Therefore, a in figure 11.3 the switch is from a high clock signal to a low clock signal, the switch seen as a falling edge, causing TCNTV to increment. TCNTV can also be incremente switch between internal and external clocks.

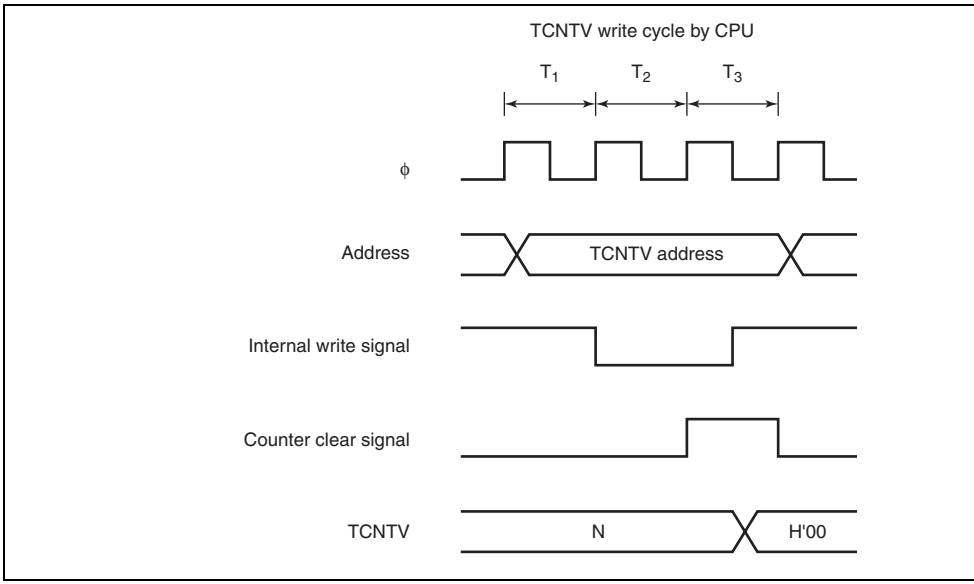


Figure 11.11 Contention between TCNTV Write and Clear

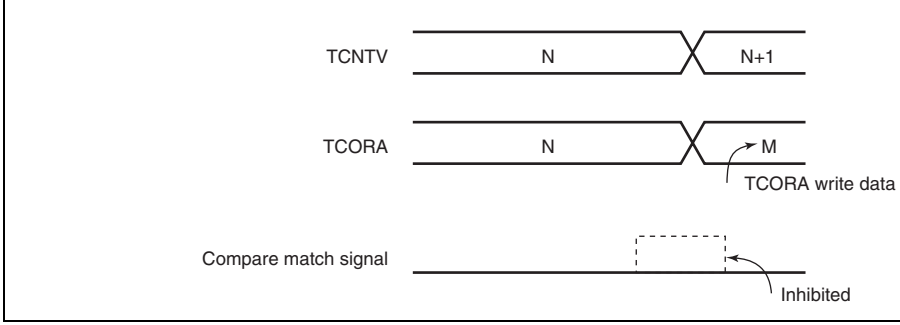


Figure 11.12 Contention between TCORA Write and Compare Match

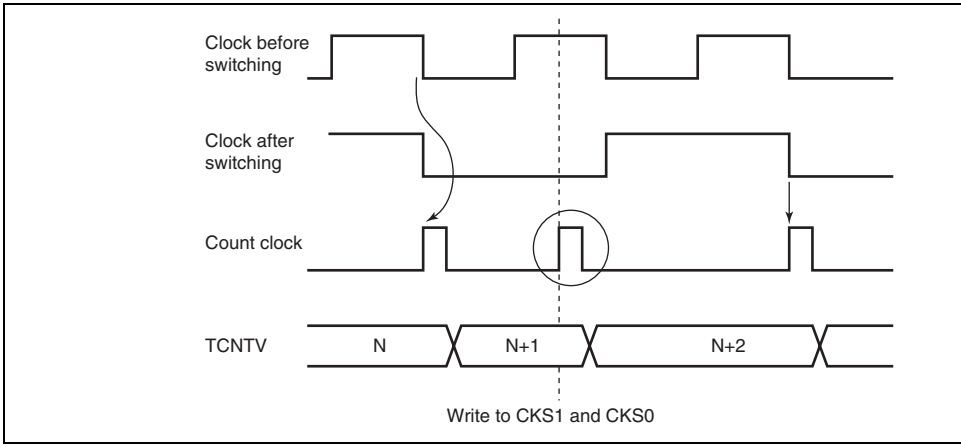
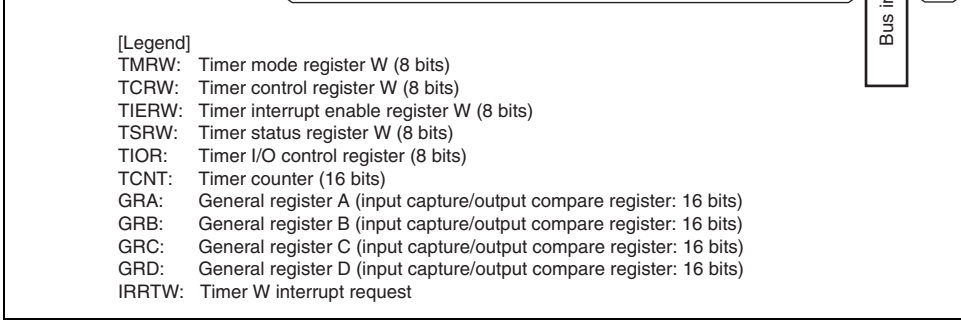


Figure 11.13 Internal Clock Switching and TCNTV Operation

- 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.

Table 12.1 summarizes the timer W functions, and figure 12.1 shows a block diagram of 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 capt



[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 12.1 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

12.3 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)

				This bit is always read as 1.
5	BUFEB	0	R/W	Buffer Operation B Selects the GRD function. 0: GRD operates as an input capture/output co register 1: GRD operates as the buffer register for GRB
4	BUFEA	0	R/W	Buffer Operation A Selects the GRC function. 0: GRC operates as an input capture/output co register 1: GRC operates as the buffer register for GRA
3	—	1	—	Reserved This bit is always read as 1.
2	PWMD	0	R/W	PWM Mode D Selects the output mode of the FTIOD pin. 0: FTIOD operates normally (output compare o 1: PWM output
1	PWMC	0	R/W	PWM Mode C Selects the output mode of the FTIOC pin. 0: FTIOC operates normally (output compare o 1: PWM output
0	PWMB	0	R/W	PWM Mode B Selects the output mode of the FTIOB pin. 0: FTIOB operates normally (output compare o 1: PWM output

5	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 $\phi/8$
				1XX: Counts on rising edges of the external ever
				When the internal clock source (ϕ) is selected, su sources are counted in subactive and subsleep m
3	TOD	0	R/W	Timer Output Level Setting D
				Sets the output value of the FTIOD pin until the f 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 f compare match C is generated.
				0: Output value is 0*
				1: Output value is 1*
1	TOB	0	R/W	Timer Output Level Setting B
				Sets the output value of the FTIOB pin until the f compare match B is generated.
				0: Output value is 0*
				1: Output value is 1*

12.3.3 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 request flag in TSRW is enabled.
6 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	IMIED	0	R/W	Input Capture/Compare Match Interrupt Enable When this bit is set to 1, IMID interrupt requeste IMFD flag in TSRW is enabled.
2	IMIEC	0	R/W	Input Capture/Compare Match Interrupt Enable When this bit is set to 1, IMIC interrupt requeste IMFC flag in TSRW is enabled.
1	IMIEB	0	R/W	Input Capture/Compare Match Interrupt Enable When this bit is set to 1, IMIB interrupt requeste IMFB flag in TSRW is enabled.
0	IMIEA	0	R/W	Input Capture/Compare Match Interrupt Enable When this bit is set to 1, IMIA interrupt requeste IMFA flag in TSRW is enabled.

				Read OVF when OVF = 1, then write 0 in OVF
6 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	IMFD	0	R/W	Input Capture/Compare Match Flag D [Setting conditions] • TCNT = GRD when GRD functions as an out compare register • The TCNT value is transferred to GRD by an capture signal when GRD functions as an inp capture register [Clearing condition] Read IMFD when IMFD = 1, then write 0 in IMFD
2	IMFC	0	R/W	Input Capture/Compare Match Flag C [Setting conditions] • TCNT = GRC when GRC functions as an out compare register • The TCNT value is transferred to GRC by an capture signal when GRC functions as an inp capture register [Clearing condition] Read IMFC when IMFC = 1, then write 0 in IMFC

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

12.3.5 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 registe 1: GRB functions as an input capture register

				00: Input capture at rising edge at the FTIOB pin
				01: Input capture at falling edge at the FTIOB pi
				1X: Input capture at rising and falling edges of th pin
3	—	1	—	Reserved
				This bit is always read as 1.
2	IOA2	0	R/W	I/O Control A2
				Selects the GRA function.
				0: GRA functions as an output compare register
				1: GRA functions as an input capture register
1	IOA1	0	R/W	I/O Control A1 and A0
0	IOA0	0	R/W	When IOA2 = 0,
				00: No output at compare match
				01: 0 output to the FTIOA pin at GRA compare m
				10: 1 output to the FTIOA pin at GRA compare m
				11: Output toggles to the FTIOA pin at GRA com match
				When IOA2 = 1,
				00: Input capture at rising edge of the FTIOA pin
				01: Input capture at falling edge of the FTIOA pi
				1X: Input capture at rising and falling edges of th pin

Legend: X: Don't care.

				Selects the GRD function. 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 pi 01: Input capture at falling edge at the FTIOD p 1X: Input capture at rising and falling edges at t 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

00: Input capture to GRC at rising edge of the FT

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.

12.3.7 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.

12.3.8 General Registers A to D (GRA to GRD)

Each general register is a 16-bit readable/writable register that can function as either an o compare register or an input-capture register. The function is selected by settings in TIOR TIOR1.

When a general register is used as an input-compare register, its value is constantly comp the TCNT value. When the two values match (a compare match), the corresponding flag IMFB, IMFC, or IMFD) in TSRW is set to 1. An interrupt request is generated at this tim IMIEA, IMIEB, IMIEC, or IMIED is set to 1. Compare match output can be selected in T

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-enab

Value in TCNT is transferred to GRA and the value in the buffer register GRC is transfe
GRA whenever an input capture is generated.

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

12.4 Operation

The timer W has the following operating modes.

- Normal Operation
- PWM Operation

12.4.1 Normal Operation

TCNT performs free-running or periodic counting operations. After a reset, TCNT is set
running counter. When the CTS bit in TMRW is set to 1, TCNT starts incrementing the
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 12.2 shows free-running c

Periodic counting operation can be performed when GRA is set as an output compare reg
bit CCLR in TCRW is set to 1. When the count matches GRA, TCNT is cleared to H'000
IMFA flag in TSRW is set to 1. If the corresponding IMIEA bit in TIERW is set to 1, an
request is generated. TCNT continues counting from H'0000. Figure 12.3 shows periodic
counting.

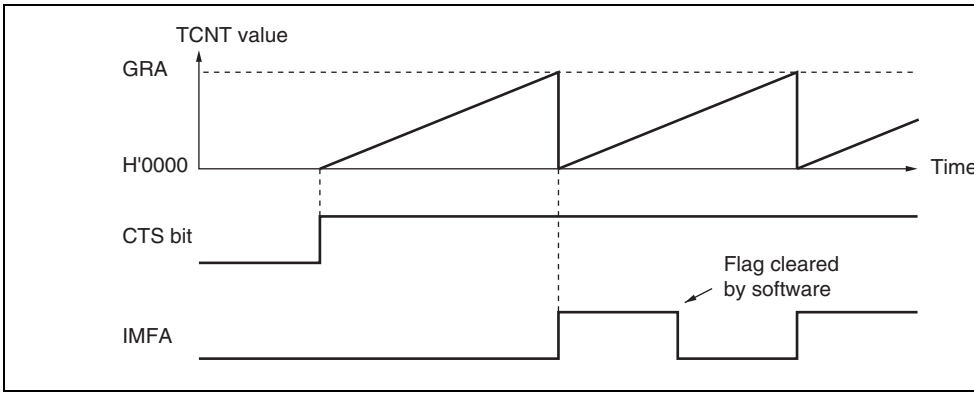


Figure 12.3 Periodic Counter Operation

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

Figure 12.5 shows an example of toggle output when TCNT operates as a free-running c
and toggle output is selected for both compare match A and B.

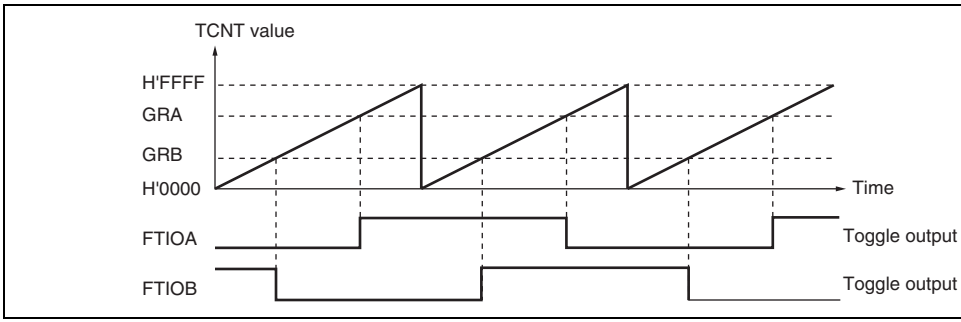


Figure 12.5 Toggle Output Example (TOA = 0, TOB = 1)

Figure 12.6 shows another example of toggle output when TCNT operates as a periodic
cleared by compare match A. Toggle output is selected for both compare match A and B

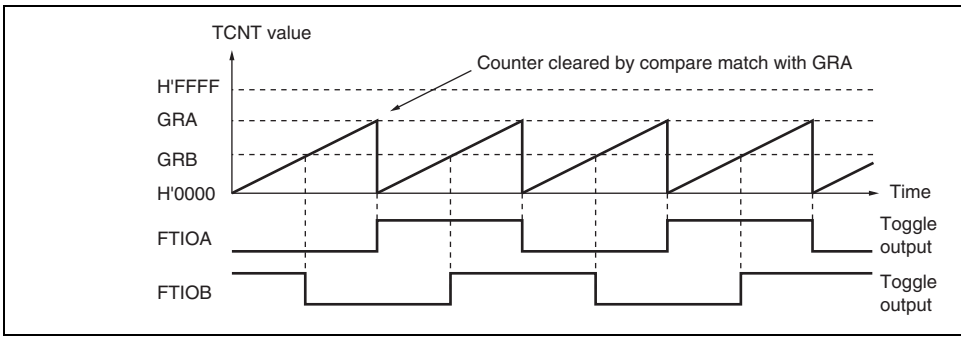


Figure 12.6 Toggle Output Example (TOA = 0, TOB = 1)

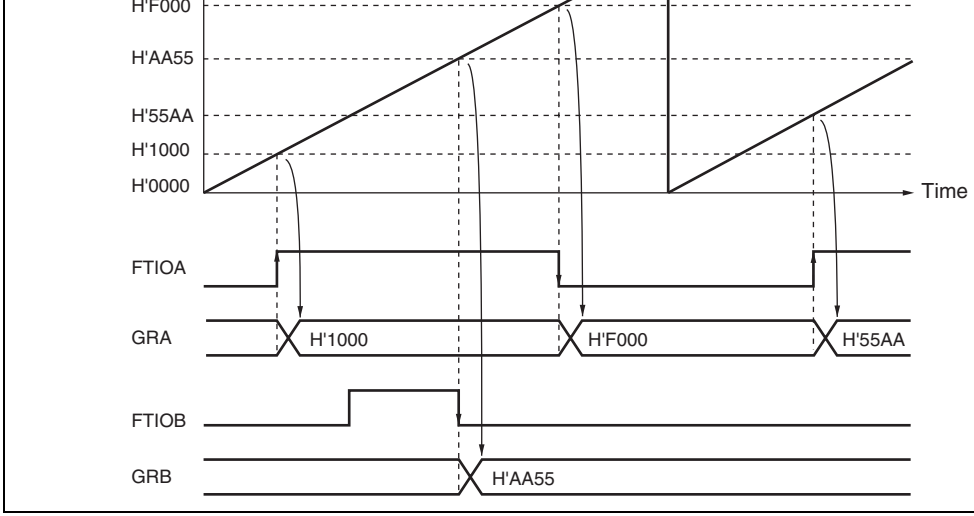


Figure 12.7 Input Capture Operating Example

Figure 12.8 shows an example of buffer operation when the GRA is set as an input-captu
register and GRC is set as the buffer register for GRA. TCNT operates as a free-running c
and FTIOA captures both rising and falling edge of the input signal. Due to the buffer op
the GRA value is transferred to GRC by input-capture A and the TCNT value is stored in

Figure 12.8 Buffer Operation Example (Input Capture)

12.4.2 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 gener functions as an output compare register automatically. The output level of each pin depe corresponding timer output level set bit (TOB, TOC, and TOD) in TCRW. When TOB i FTIOB output goes to 1 at compare match A and to 0 at compare match B. When TOB i FTIOB output goes to 0 at compare match A and to 1 at compare match B. Thus the con 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 12.9 shows an example of operation in PWM mode. The output signals go to 1 ar is cleared at compare match A, and the output signals go to 0 at compare match B, C, an TOC, and TOD = 1: initial output values are set to 1).

Figure 12.9 PWM Mode Example (1)

Figure 12.10 shows another example of operation in PWM mode. The output signals go t
TCNT is cleared at compare match A, and the output signals go to 1 at compare match B,
D (TOB, TOC, and TOD = 0: initial output values are set to 1).

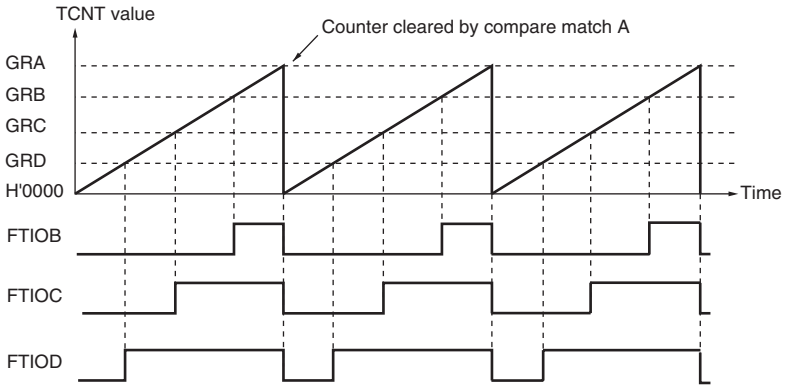


Figure 12.10 PWM Mode Example (2)

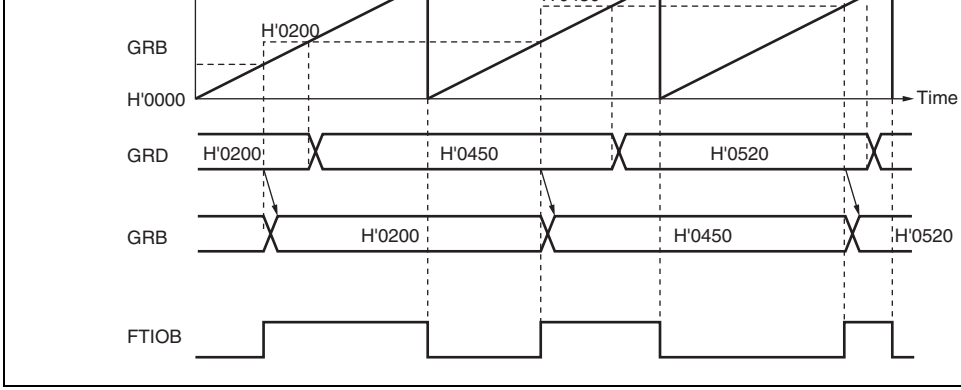


Figure 12.11 Buffer Operation Example (Output Compare)

Figures 12.12 and 12.13 show examples of the output of PWM waveforms with duty cy
and 100%.

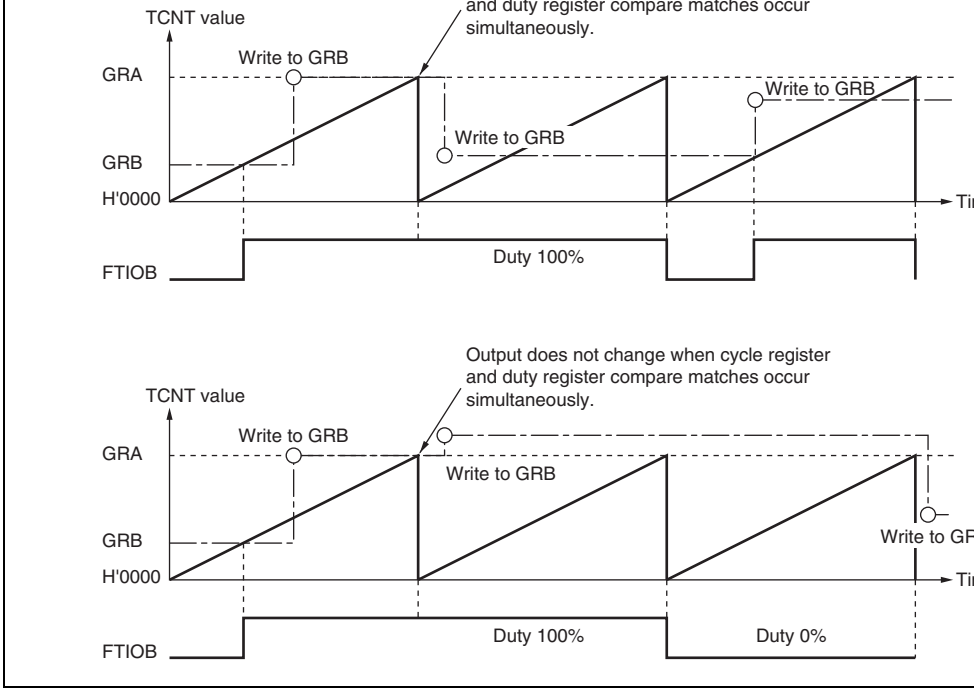


Figure 12.12 PWM Mode Example
(TOB, TOC, and TOD = 0: initial output values are set to 0)

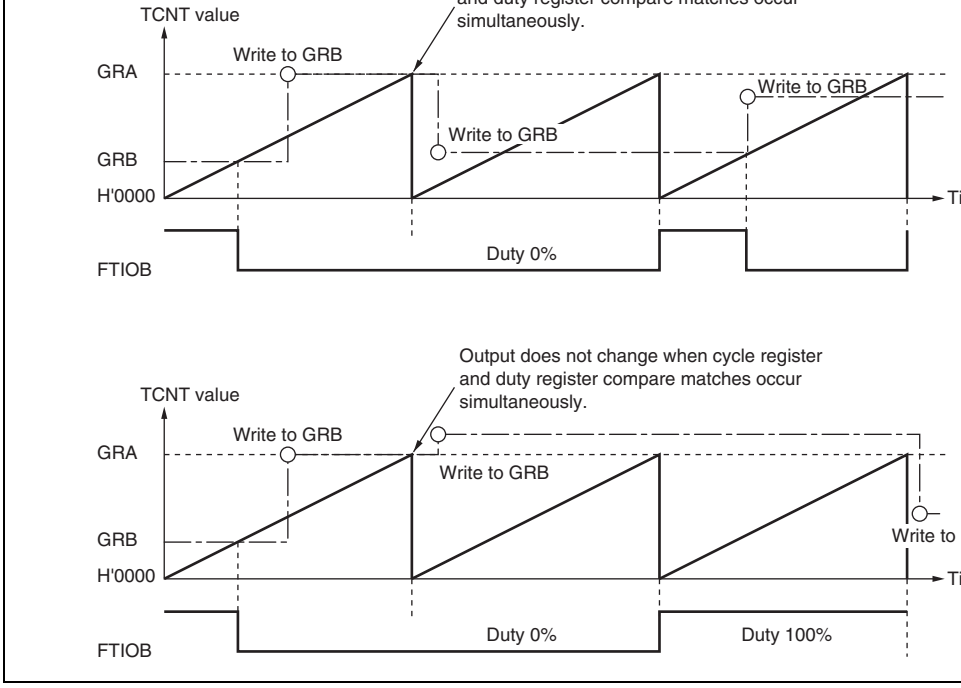


Figure 12.13 PWM Mode Example
(TOB, TOC, and TOD = 1: initial output values are set to 1)

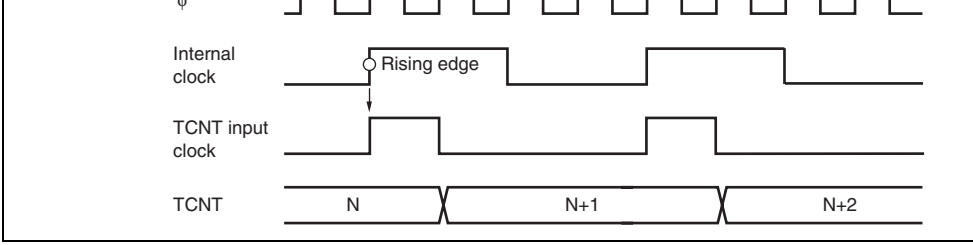


Figure 12.14 Count Timing for Internal Clock Source

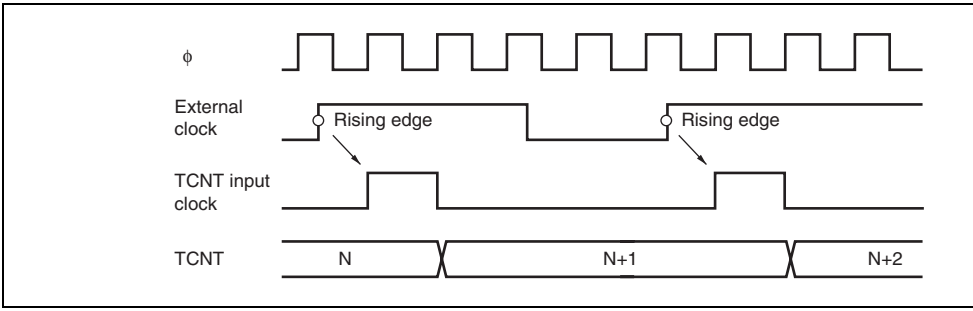


Figure 12.15 Count Timing for External Clock Source

12.5.2 Output Compare Output Timing

The compare match signal is generated in the last state in which TCNT and GR match (w
TCNT changes from the matching value to the next value). When the compare match sign
generated, the output value selected in TIOR is output at the compare match output pin (F
FTIOB, FTIOC, or FTIOD).

When TCNT matches GR, the compare match signal is generated only after the next coun
pulse is input.

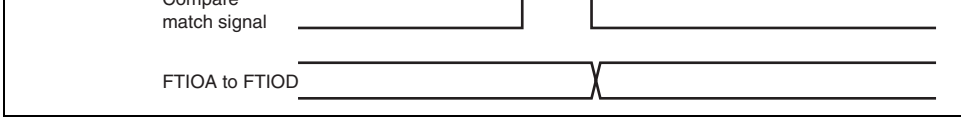


Figure 12.16 Output Compare Output Timing

12.5.3 Input Capture Timing

Input capture on the rising edge, falling edge, or both edges can be selected through sett
TIOR0 and TIOR1. Figure 12.17 shows the timing when the falling edge is selected. Th
width of the input capture signal must be at least two system clock (ϕ) cycles; shorter pu
not be detected correctly.

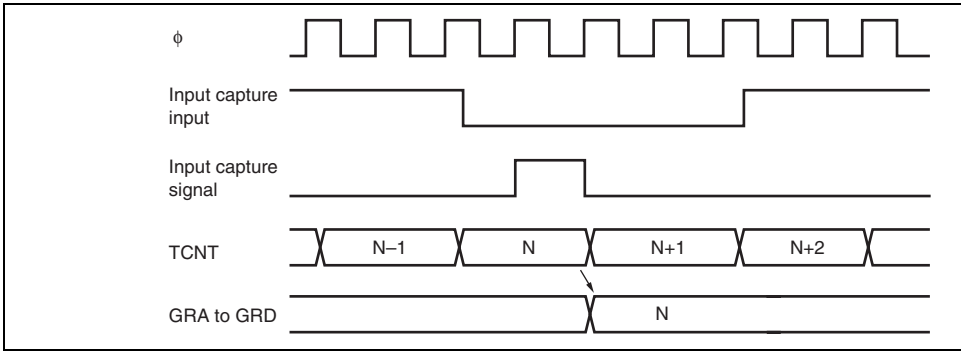


Figure 12.17 Input Capture Input Signal Timing

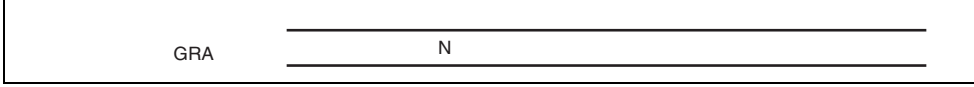


Figure 12.18 Timing of Counter Clearing by Compare Match

12.5.5 Buffer Operation Timing

Figures 12.19 and 12.20 show the buffer operation timing.

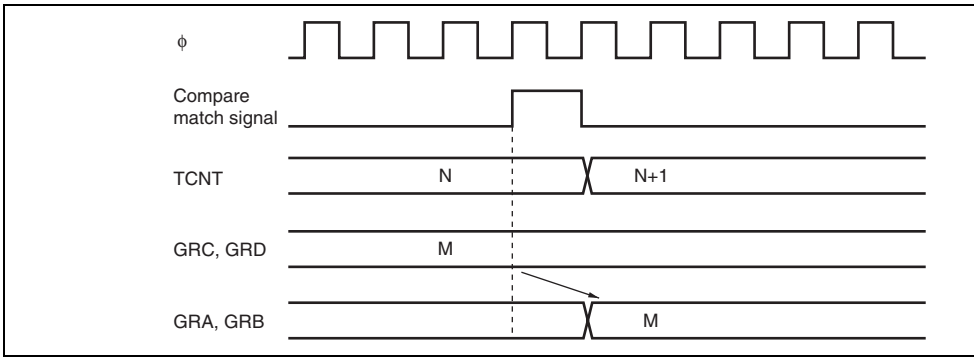


Figure 12.19 Buffer Operation Timing (Compare Match)

Figure 12.20 Buffer Operation Timing (Input Capture)

12.5.6 Timing of IMFA to IMFD Flag Setting at Compare Match

If a general register (GRA, GRB, GRC, or GRD) is used as an output compare register, t
corresponding IMFA, IMFB, IMFC, or IMFD flag is set to 1 when TCNT matches the g
register.

The compare match signal is generated in the last state in which the values match (when
updated from the matching count to the next count). Therefore, when TCNT matches a g
register, the compare match signal is generated only after the next TCNT clock pulse is

Figure 12.21 shows the timing of the IMFA to IMFD flag setting at compare match.

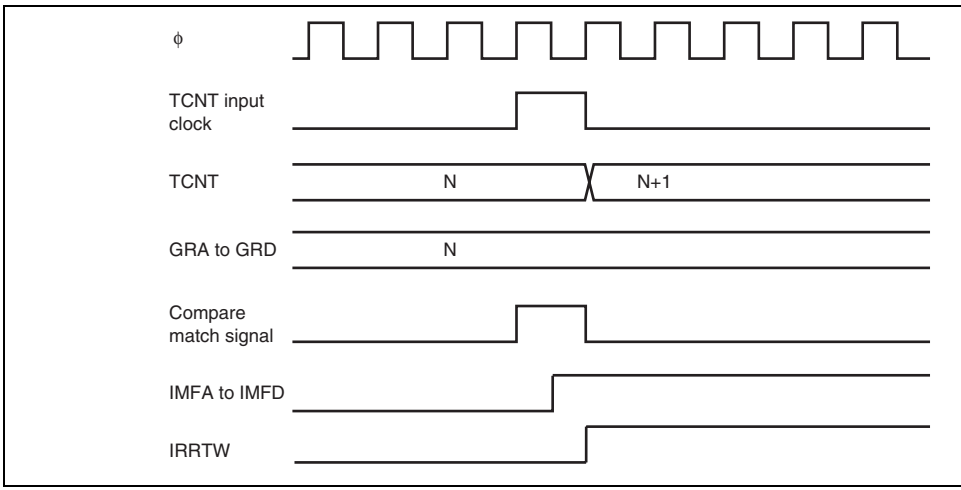


Figure 12.21 Timing of IMFA to IMFD Flag Setting at Compare Match

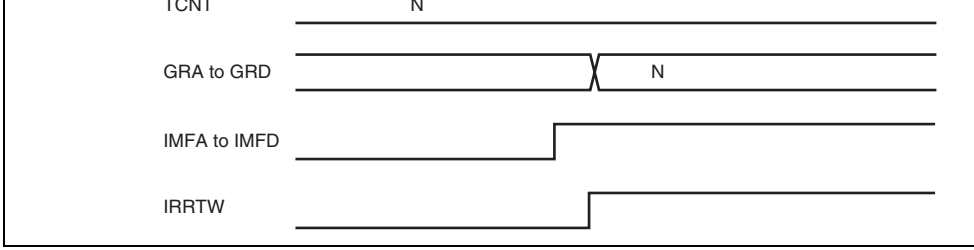


Figure 12.22 Timing of IMFA to IMFD Flag Setting at Input Capture

12.5.8 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 12.23 shows the status flag clearing timing.

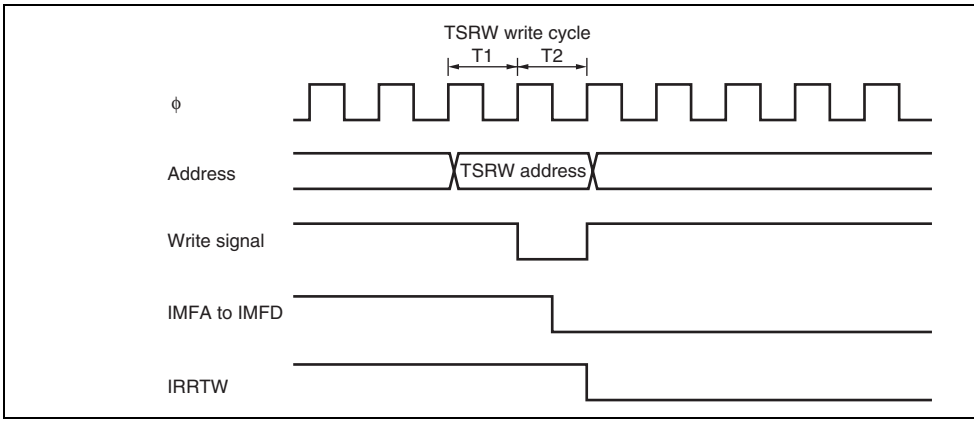


Figure 12.23 Timing of Status Flag Clearing by CPU

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 divided system clock (ϕ). Therefore, as figure 12.25 the switch is from a low clock signal to a high clock signal, the switcho 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

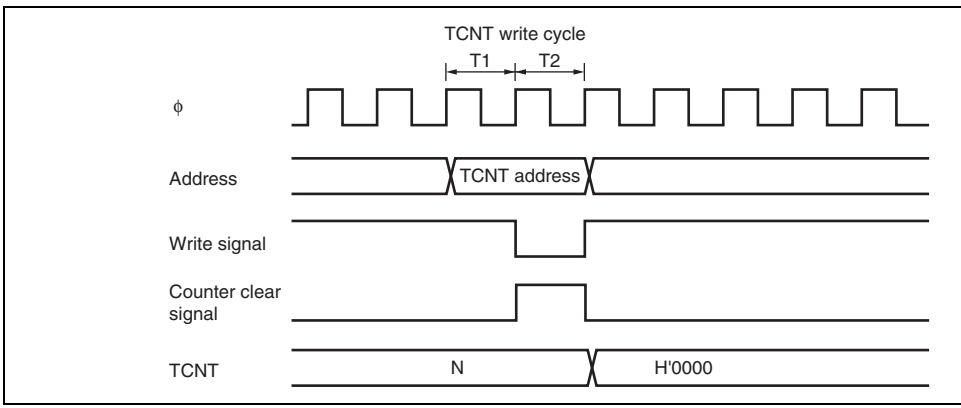


Figure 12.24 Contention between TCNT Write and Clear

Figure 12.25 Internal Clock Switching and TCNT Operation

bit manipulation instruction to TCRW occur at the same timing.

TCRW has been set to H'06. Compare match B and compare match C are used. The FTIOB pin is in the 1 output state and is set to the toggle output or the 0 output by compare match B.
When BCLR#2, @TCRW is executed to clear the TOC bit (the FTIOC signal is low) and compare match B occurs at the same timing as shown below, the H'02 writing to TCRW has priority and compare match B does not drive the FTI the FTIOB signal remains high.

Bit	7	6	5	4	3	2	1	0
TCRW	CCLR	CKS2	CKS1	CKS0	TOD	TOC	TOB	TOA
Set value	0	0	0	0	0	1	1	0

BCLR#2, @TCRW
(1) TCRW read operation: Read H'06
(2) Modify operation: Modify H'06 to H'02
(3) Write operation to TCRW: Write H'02

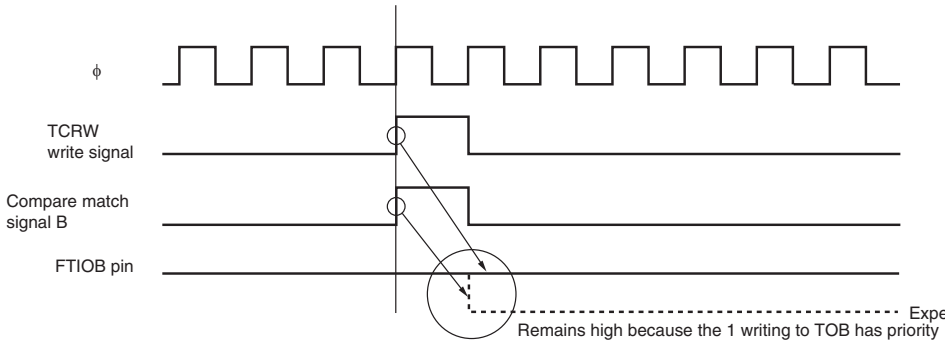


Figure 12.26 When Compare Match and Bit Manipulation Instruction to TC Occur at the Same Timing

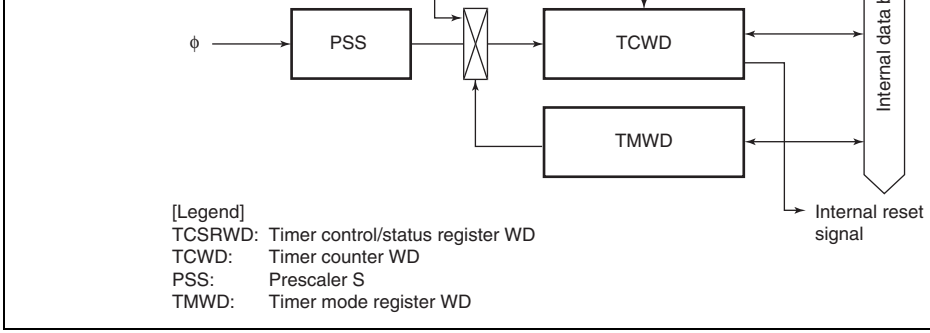


Figure 13.1 Block Diagram of Watchdog Timer

13.1 Features

- Selectable from nine counter input clocks.

  Eight clock sources (φ/64, φ/128, φ/256, φ/512, φ/1024, φ/2048, φ/4096, and φ/8192) internal oscillator can be selected as the timer-counter clock. When the internal oscil selected, it can operate as the watchdog timer in any operating mode.
- Reset signal generated on counter overflow

  An overflow period of 1 to 256 times the selected clock can be set.

watchdog timer operation and indicates the operating state. TCSRWD must be rewritten l
the MOV instruction. The bit manipulation instruction cannot be used to change the settin

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

				• When 0 is written to the WDON bit while wri the B2WI when the TCSRWE bit=1
1	B0WI	1	R/W	Bit 0 Write Inhibit This bit can be written to the WRST bit only wh write value of the B0WI bit is 0. This bit is alway 1.
0	WRST	0	R/W	Watchdog Timer Reset [Setting condition] When TCWD overflows and an internal reset si generated [Clearing conditions] • Reset by $\overline{\text{RES}}$ pin • When 0 is written to the WRST bit while wri the B0WI bit when the TCSRWE bit=1

13.2.2 Timer Counter WD (TCWD)

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 TCSRWD is set to 1. TCWD is in H'00.

1	CKS1	1	R/W	1000: Internal clock: counts on ϕ/64
0	CKS0	1	R/W	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
				0XXX: Internal oscillator
				For the internal oscillator overflow periods, see 21, Electrical Characteristics.

Legend: X: Don't care.

Figure 13.2 shows an example of watchdog timer operation.

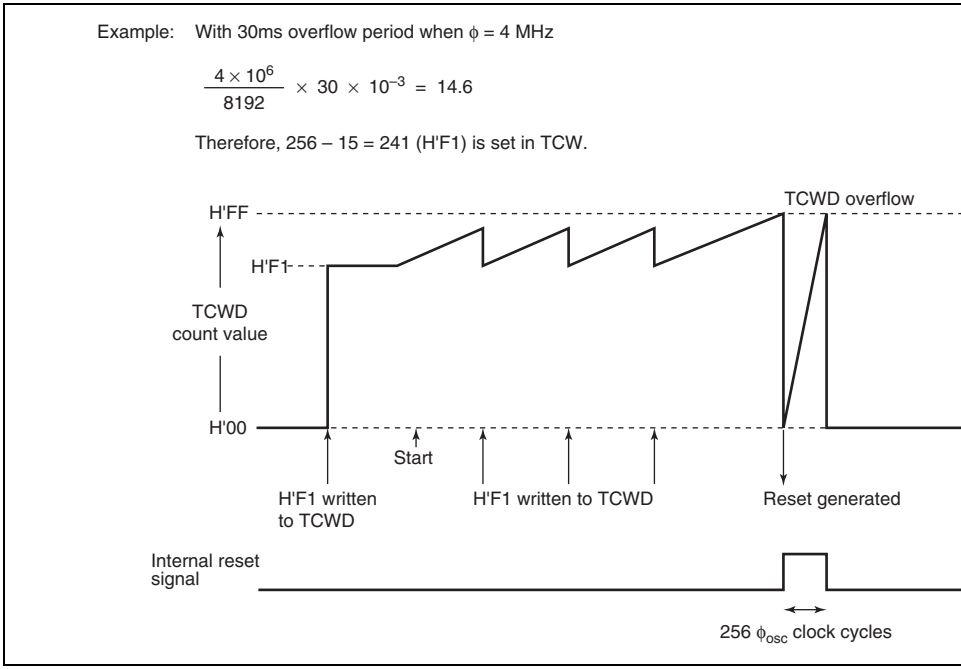


Figure 13.2 Watchdog Timer Operation Example

14.1 Features

- Choice of asynchronous or clocked 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
- External clock or on-chip baud rate generator can be selected as a transfer clock sou
- Six interrupt sources

  Transmit-end, transmit-data-empty, receive-data-full, overrun error, framing error, a
  error.

Asynchronous mode

- Data length: 7 or 8 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 RxD pin level directly in the c
  framing error

Clocked synchronous mode

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

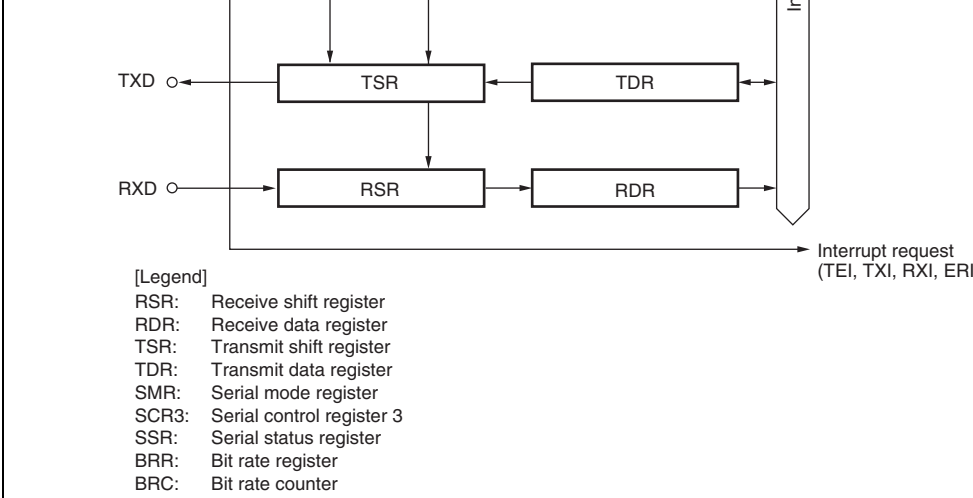


Figure 14.1 Block Diagram of SCI3

14.3 Register Descriptions

The SCI3 has the following registers.

- Receive shift register (RSR)
- Receive data register (RDR)
- Transmit shift register (TSR)
- Transmit data register (TDR)
- Serial mode register (SMR)
- Serial control register 3 (SCR3)
- Serial status register (SSR)
- Bit rate register (BRR)

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.

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 TXD pin. TSR cannot be directly accessed by the CPU.

14.3.4 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for transmission. When the SCI3 detects that 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.

6	CHR	0	R/W	Character Length (enabled only in asynchronou 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous m When this bit is set to 1, the parity bit is added t data before transmission, and the parity bit is ch reception.
4	PM	0	R/W	Parity Mode (enabled only when the PE bit is 1 asynchronous mode) 0: Selects even parity. 1: Selects odd parity.
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, of the value in the bit. If the second stop bit is 0 treated as the start bit of the next transmit chara
2	MP	0	R/W	Multiprocessor Mode When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit bit settings are invalid. In clocked synchronous bit should be cleared to 0.

(BRR). n is the decimal representation of the val
BRR (see section 14.3.8, Bit Rate Register (BRR

14.3.6 Serial Control Register 3 (SCR3)

SCR3 is a register that enables or disables SCI3 transfer operations and interrupt requests also used to select the transfer clock source. For details on interrupt requests, refer to sect Interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, the TXI interrupt reques enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt re are enabled.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.

2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, the TEI interrupt reque enabled.
1	CKE1	0	R/W	Clock Enable 0 and 1
0	CKE0	0	R/W	Selects the clock source. Asynchronous mode: 00: Internal baud rate generator 01: Internal baud rate generator Outputs a clock of the same frequency as th from the SCK3 pin. 10: External clock Inputs a clock with a frequency 16 times the from the SCK3 pin. 11: Reserved Clocked synchronous mode: 00: Internal clock (SCK3 pin functions as clock 01: Reserved 10: External clock (SCK3 pin functions as clock 11: Reserved

				• When the TE bit in SCR3 is 0 • When data is transferred from TDR to TSR [Clearing conditions] • When 0 is written to TDRE after reading TDR • When the transmit data is written to TDR
6	RDRF	0	R/W	Receive Data Register Full Indicates that the received data is stored in RDR [Setting condition] • When serial reception ends normally and rec is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDR • When data is read from RDR
5	OER	0	R/W	Overrun Error [Setting condition] • When an overrun error occurs in reception [Clearing condition] • When 0 is written to OER after reading OER
4	FER	0	R/W	Framing Error [Setting condition] • When a framing error occurs in reception [Clearing condition] • When 0 is written to FER after reading FER =

				• When TDRE = 1 at transmission of the last byte serial transmit character [Clearing conditions] • When 0 is written to TEND after reading TE • When the transmit data is written to TDR
1	MPBR	0	R	Multiprocessor Bit Receive MPBR stores the multiprocessor bit in the recei character data. When the RE bit in SCR3 is cle its previous state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be added transmit character data.

[Asynchronous Mode]

$$N = \frac{\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

[Clocked Synchronous Mode]

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Note: B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
ϕ: Operating frequency (MHz)
n: CKS1 and CKS0 setting for SMR ($0 \leq N \leq 3$)

1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	77
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	19
9600	0	6	–6.99	0	6	–2.48	0	7	0.00	0	9
19200	0	2	8.51	0	2	13.78	0	3	0.00	0	4
31250	0	1	0.00	0	1	4.86	0	1	22.88	0	2
38400	0	1	–18.62	0	1	–14.67	0	1	0.00	—	—

Legend:

— : A setting is available but error occurs

	Operating Frequency ϕ (MHz)										
	3.6864			4			4.9152				
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64
300	1	95	0.00	1	103	0.16	1	127	0.00	1	12
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	12
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15
19200	0	5	0.00	0	6	–6.99	0	7	0.00	0	7
31250	—	—	—	0	3	0.00	0	4	–1.70	0	4
38400	0	2	0.00	0	2	8.51	0	3	0.00	0	3

1200	0	155	0.16	0	159	0.00	0	191	0.00	0	207
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	25
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	12
31250	0	5	0.00	0	5	2.40	0	6	5.33	0	7
38400	0	4	–2.34	0	4	0.00	0	5	0.00	0	6

	Operating Frequency ϕ (MHz)										
	9.8304			10			12			12.8	
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9

1200	1	90	0.16	1	95	0.00	1	103	0.16
2400	0	181	0.16	0	191	0.00	0	207	0.16
4800	0	90	0.16	0	95	0.00	0	103	0.16
9600	0	45	–0.93	0	47	0.00	0	51	0.16
19200	0	22	–0.93	0	23	0.00	0	25	0.16
31250	0	13	0.00	0	14	–1.70	0	15	0.00
38400	—	—	—	0	11	0.00	0	12	0.16

	Operating Frequency ϕ (MHz)					
	18			20		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)
110	3	79	–0.12	3	88	–0.25
150	2	233	0.16	3	64	0.16
300	2	116	0.16	2	129	0.16
600	1	233	0.16	2	64	0.16
1200	1	116	0.16	1	129	0.16
2400	0	233	0.16	1	64	0.16
4800	0	116	0.16	0	129	0.16
9600	0	58	–0.96	0	64	0.16
19200	0	28	1.02	0	32	–1.36
31250	0	17	0.00	0	19	0.00
38400	0	14	–2.34	0	15	1.73

Legend:

—: A setting is available but error occurs.

4.9152	153600	0	0	14.7456	460800	0
5	156250	0	0	16	500000	0
6	187500	0	0	17.2032	537600	0
6.144	192000	0	0	18	562500	0
7.3728	230400	0	0	20	625000	0

2.5k	0	199	1	99	1	199	1	249	2
5k	0	99	0	199	1	99	1	124	1
10k	0	49	0	99	0	199	0	249	1
25k	0	19	0	39	0	79	0	99	0
50k	0	9	0	19	0	39	0	49	0
100k	0	4	0	9	0	19	0	24	0
250k	0	1	0	3	0	7	0	9	0
500k	0	0*	0	1	0	3	0	4	0
1M			0	0*	0	1	—	—	0
2M					0	0*	—	—	0
2.5M							0	0*	—
4M									0

Legend:

Blank : No setting is available.

— : A setting is available but error occurs.

* : Continuous transfer is not possible.

2.5k	2	112	2	124
5k	1	224	1	249
10k	1	112	1	124
25k	0	179	0	199
50k	0	89	0	99
100k	0	44	0	49
250k	0	17	0	19
500k	0	8	0	9
1M	0	4	0	4
2M	—	—	—	—
2.5M	—	—	0	1
4M	—	—	—	—

Legend:

Blank : No setting is available.

— : A setting is available but error occurs.

* : Continuous transfer is not possible.

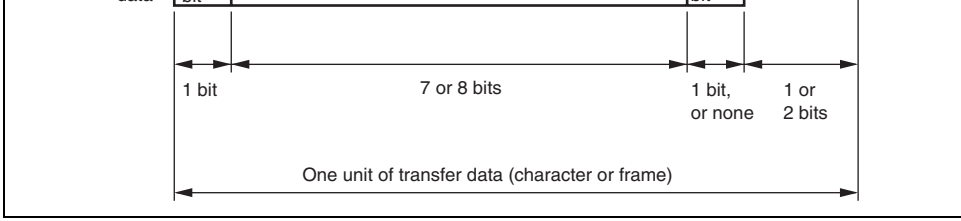


Figure 14.2 Data Format in Asynchronous Communication

14.4.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external cloc
the SCK3 pin can be selected as the SCI3's serial clock source, according to the setting
COM bit in SMR and the CKE0 and CKE1 bits in SCR3. When an external clock is inp
SCK3 pin, the clock frequency should be 16 times the bit rate used.

When the SCI3 is operated on an internal clock, the clock can be output from the SCK3
frequency of the clock output in this case is equal to the bit rate, and the phase is such th
rising edge of the clock is in the middle of the transmit data, as shown in figure 14.3.

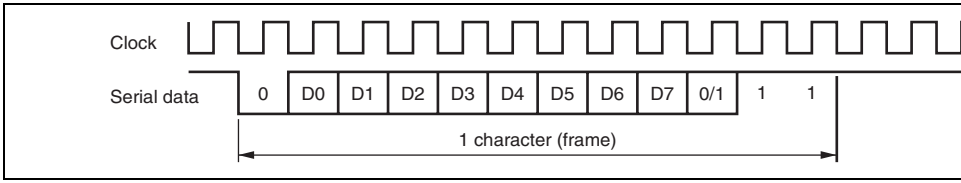
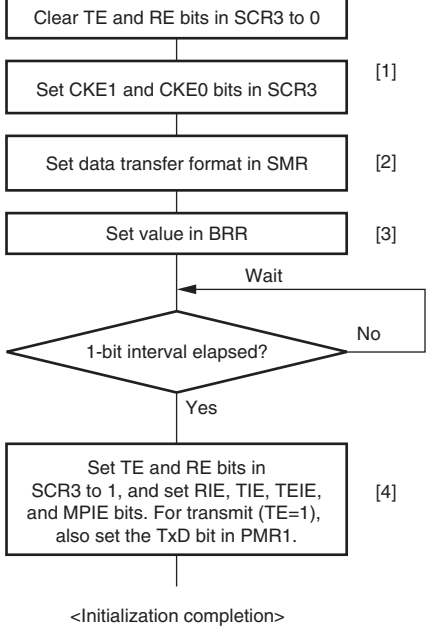


**Figure 14.3 Relationship between Output Clock and Transfer Data Phas
(Asynchronous Mode) (Example with 8-Bit Data, Parity, Two Stop Bits)**



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 clocked 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 SCR3 to 1. RE settings enable the RXD pin to be used. For transmission, set the TXD bit in PMR1 to 1 to enable the TXD output pin to be used. Also set the RIE, TIE, TEIE, and MPIE 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

to TDR before transmission of the current transmit data has been completed.

3. The SCI3 checks the TDRE flag at the timing for sending the stop bit.
4. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, 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 SCR3 is set to 1 at this time, interrupt request is generated.
6. Figure 14.6 shows a sample flowchart for transmission in asynchronous mode.

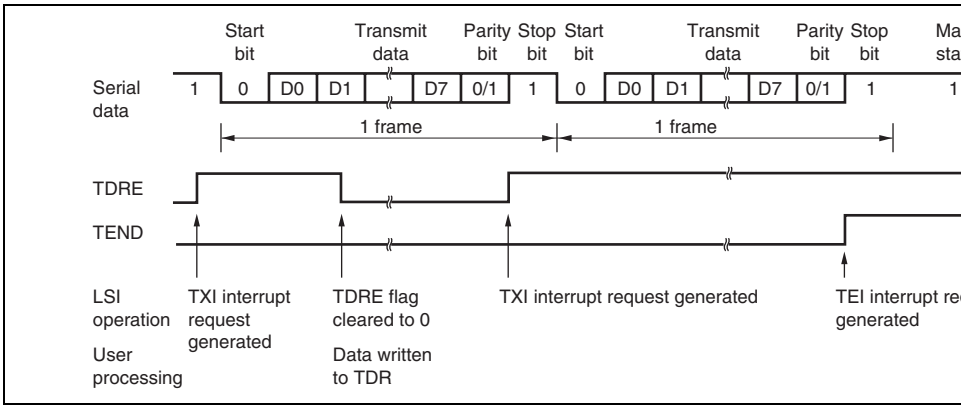
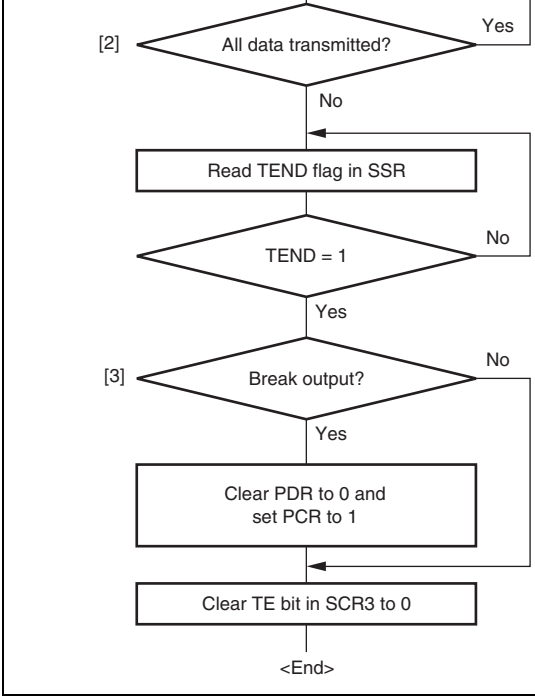


Figure 14.5 Example SCI3 Operation in Transmission in Asynchronous M (8-Bit Data, Parity, One Stop Bit)



and PDR to 0, clear TxD in PMR1
to 0, then clear the TE bit in SCR3
to 0.

Figure 14.6 Sample Serial Transmission Flowchart (Asynchronous Mode)

3. If a parity error is detected, the FER bit in SSR is set to 1 and receive data is transfer
RDR. If the RIE bit in SCR3 is set to 1 at this time, an ERI 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 SCR3 is set to 1 at this time, an ERI 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 SCR3 is set to 1 at this time, an RXI interrupt re
generated. Continuous reception is possible because the RXI interrupt routine reads t
data transferred to RDR before reception of the next receive data has been completed

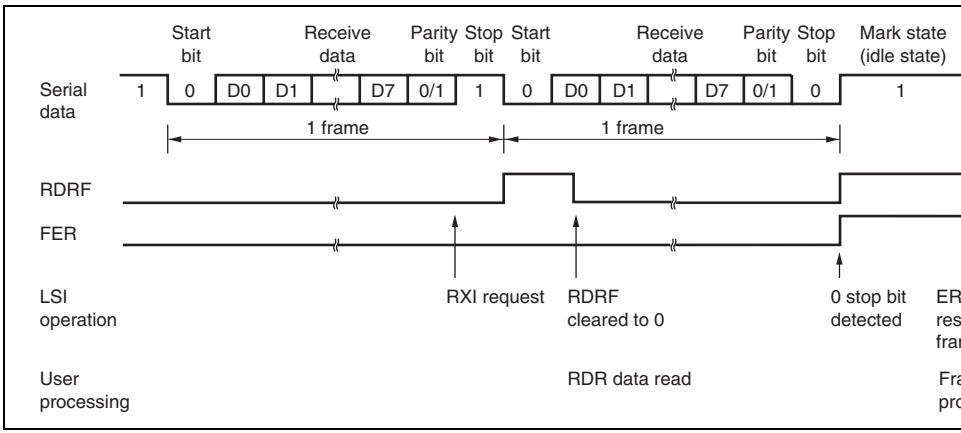
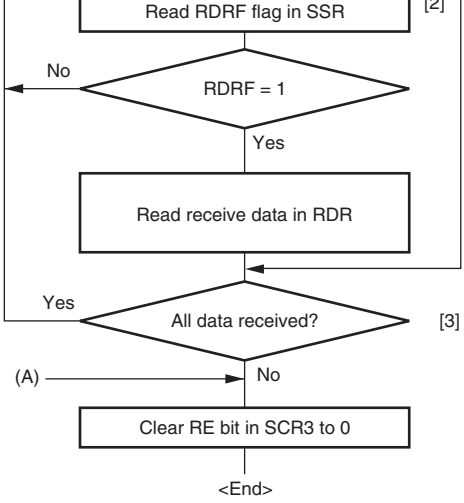


**Figure 14.7 Example SCI3 Operation in Reception in Asynchronous Mod
(8-Bit Data, Parity, One Stop Bit)**

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 + framin
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 + framin parity error

Note: * The RDRF flag retains the state it had before data reception.



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 RxD pin.

Figure 14.8 Sample Serial Data Reception Flowchart (Asynchronous mode)

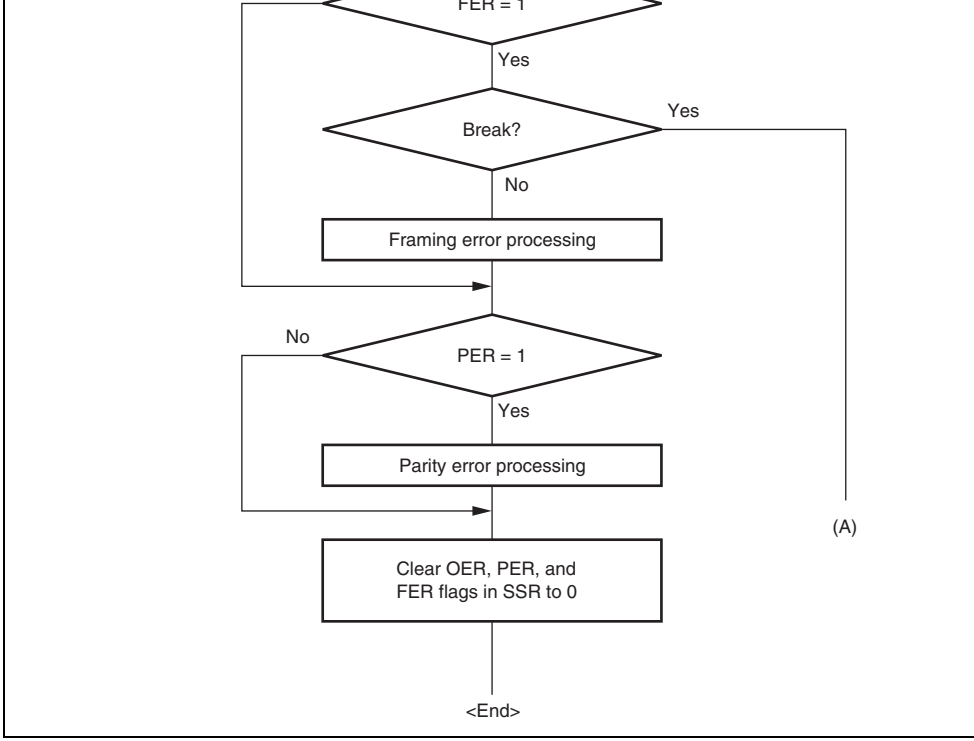


Figure 14.8 Sample Serial Reception Data Flowchart (2)

through the use of a common clock. Both the transmitter and the receiver also have a do
buffered structure, so data can be read or written during transmission or reception, enabl
continuous data transfer.

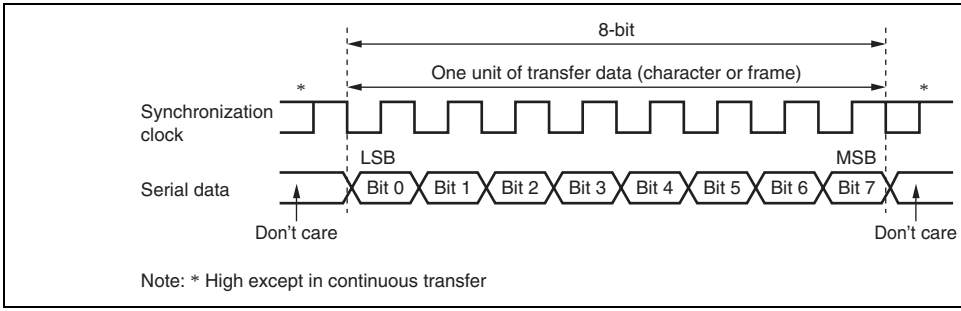


Figure 14.9 Data Format in Clocked Synchronous Communication

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 SCR3. When the SCI3 is operated on an interna the serial clock is output from the SCK3 pin. Eight serial clock pulses are output in the t one character, and when no transfer is performed the clock is fixed high.

14.5.2 SCI3 Initialization

Before transmitting and receiving data, the SCI3 should be initialized as described in a 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 SCI 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 trans
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 mai
output state of the last bit. If the TEIE bit in SCR3 is set to 1 at this time, a TEI interr
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 Clocked Synchronc

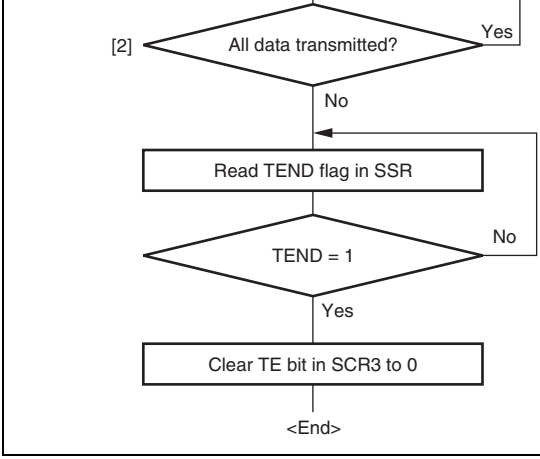


Figure 14.11 Sample Serial Transmission Flowchart (Clocked Synchronous M

time, an ERI 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 d
transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt re
generated.

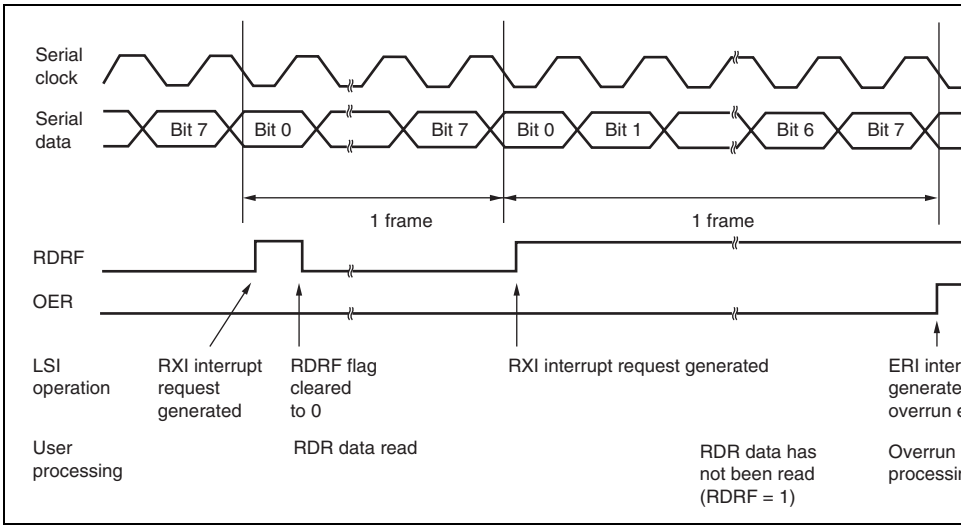
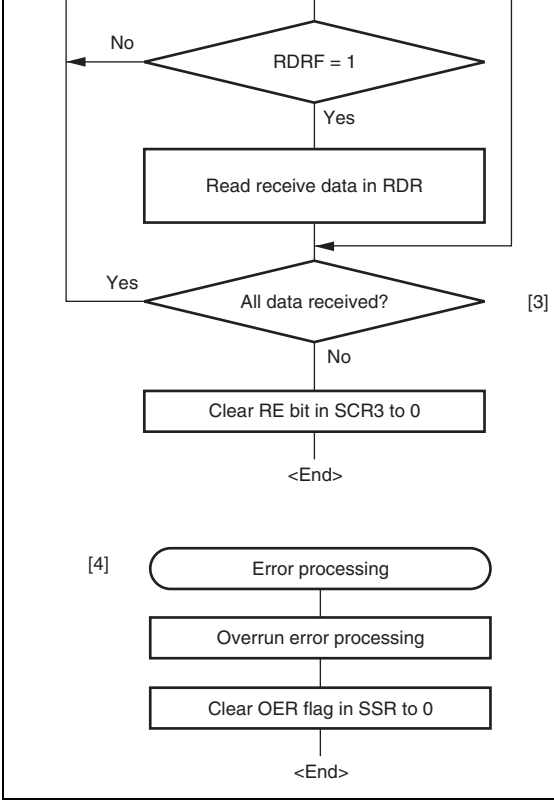


Figure 14.12 Example of SCI3 Reception Operation in Clocked Synchronous

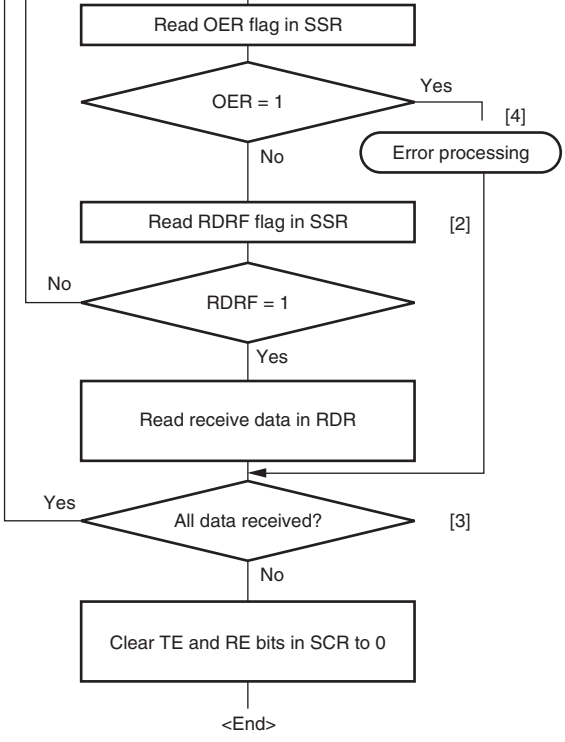
Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear th
FER, PER, and RDRF bits to 0 before resuming reception. Figure 14.13 shows a sample
for serial data reception.



cleared to 0.

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

Figure 14.13 Sample Serial Reception Flowchart (Clocked Synchronous Mod



reading the RDRF flag, reading
Also, before the MSB (bit 7) of th
current frame is transmitted, rea
from the TDRE flag to confirm th
writing is possible. Then write d
TDR.
When data is written to TDR, the
TDRE flag is automatically clear
0. When data is read from RDR,
RDRF flag is automatically clear
0.

[4] If an overrun error occurs, read t
OER flag in SSR, and after
performing the appropriate error
processing, clear the OER flag t
Transmission/reception cannot b
resumed if the OER flag is set to
For overrun error processing, se
figure 14.13.

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

cycle is a data transmission cycle. Figure 14.13 shows an example of inter-processor
communication using the multiprocessor format. The transmitting station first sends the
of the receiving station with which it wants to perform serial communication as data wit
multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit
When data with a 1 multiprocessor bit is received, the receiving station compares that da
own ID. The station whose ID matches then receives the data sent next. Stations whose
match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI3 uses the MPIE bit in SCR3 to implement this function. When the MPIE bit is
transfer of receive data from RSR to RDR, error flag detection, and setting the SSR statu
RDRF, FER, and OER to 1, are inhibited until data with a 1 multiprocessor bit is receive
reception of a receive character with a 1 multiprocessor bit, the MPBR bit in SSR is set
the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit i
set to 1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is rendered invalid. Al
settings are the same as those in normal asynchronous mode. The clock used for multipr
communication is the same as that in normal asynchronous mode.

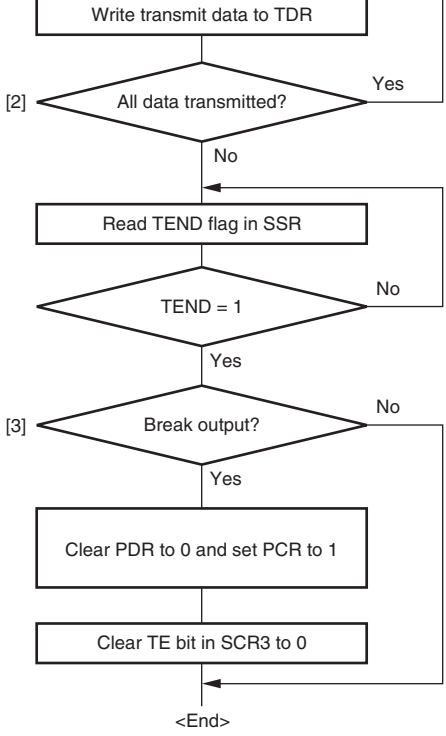
specification receiving station specified by ID

[Legend]
MPB: Multiprocessor bit

Figure 14.15 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

14.6.1 Multiprocessor Serial Data Transmission

Figure 14.16 shows a sample flowchart for multiprocessor serial data transmission. For a
transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmis
cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI3 operations are
as those in asynchronous mode.



[3] To output a break in serial transmission, set the port PCR to 1, clear PDR to 0, then clear the TE bit in SCR3 to 0.

Figure 14.16 Sample Multiprocessor Serial Transmission Flowchart

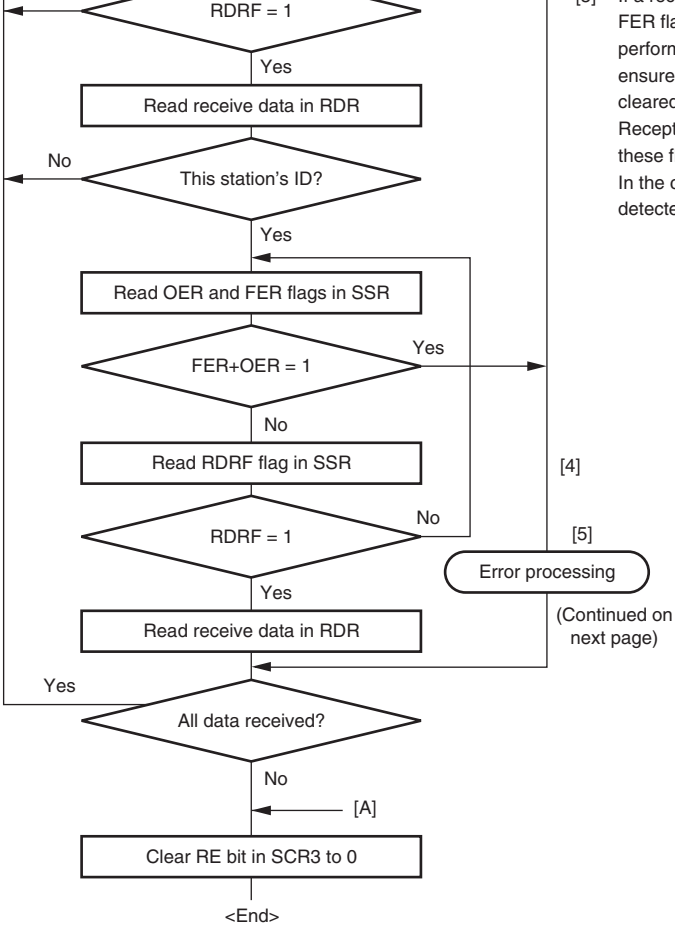


Figure 14.17 Sample Multiprocessor Serial Reception Flowchart (1)

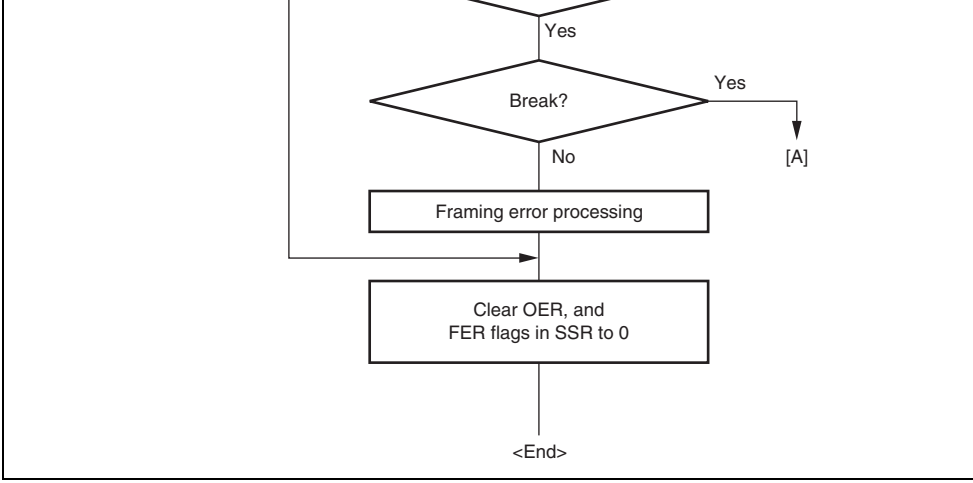


Figure 14.17 Sample Multiprocessor Serial Reception Flowchart (2)

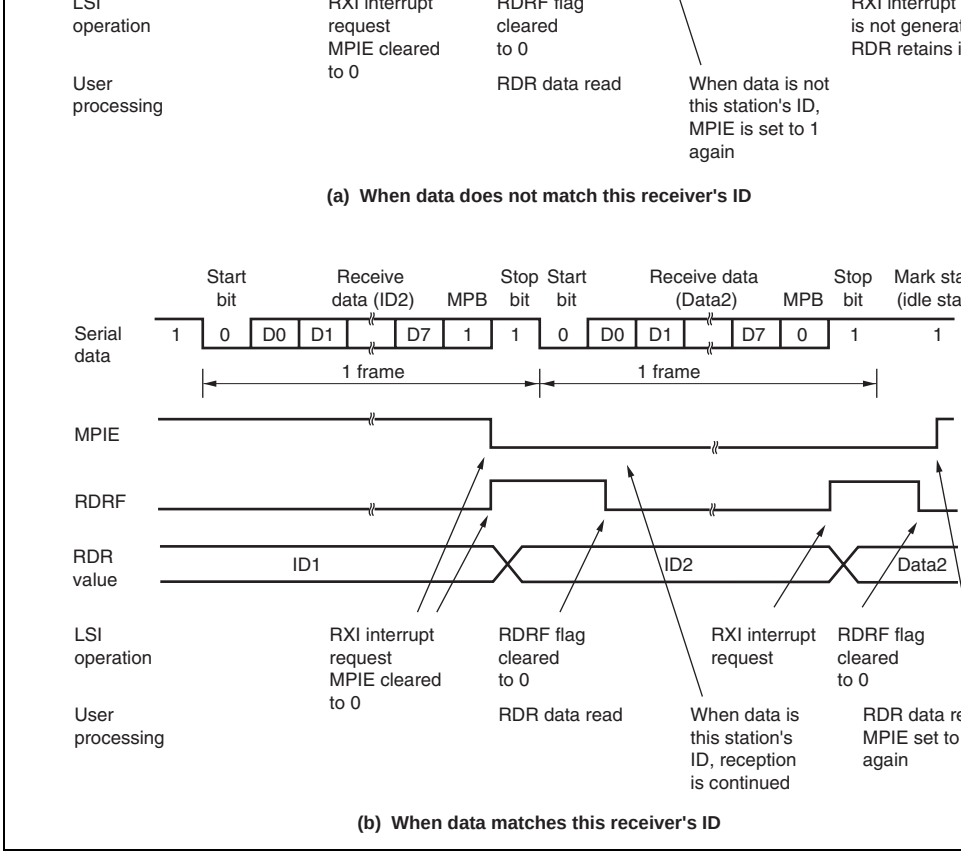


Figure 14.18 Example of SCI3 Operation in Reception Using Multiprocessor F (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

Transmission End	TEI	Setting TEND in SSR
Receive Error	ERI	Setting OER, FER, and PER in SSR

The initial value of the TDRE flag in SSR is 1. Thus, when the TIE bit in SCR3 is set to 1 transferring the transmit data to TDR, a TXI interrupt request is generated even if the tran is not ready. The initial value of the TEND flag in SSR is 1. Thus, when the TEIE bit in S set to 1 before transferring the transmit data to TDR, a TEI interrupt request is generated the transmit data has not been sent. It is possible to make use of the most of these interrup requests efficiently by transferring the transmit data to TDR in the interrupt routine. To p generation of these interrupt requests (TXI and TEI), set the enable bits (TIE and TEIE) t correspond to these interrupt requests to 1, after transferring the transmit data to TDR.

When TE is 0, the TxD pin is used as an I/O port whose direction (input or output) and l
determined by PCR and PDR. This can be used to set the TxD pin to mark state (high le
send a break during serial data transmission. To maintain the communication line at mar
until TE is set to 1, set both PCR and PDR to 1. As TE is cleared to 0 at this point, the T
becomes an I/O port, and 1 is output from the TxD pin. To send a break during serial tra
first set PCR to 1 and PDR to 0, and then clear TE to 0. When TE is cleared to 0, the tra
initialized regardless of the current transmission state, the TxD pin becomes an I/O port,
output from the TxD pin.

14.8.3 Receive Error Flags and Transmit Operations (Clocked 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.

[2N N]

... 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.

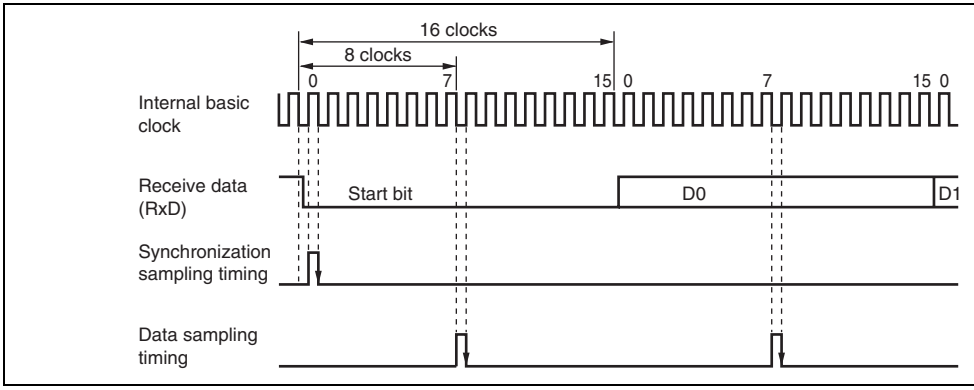


Figure 14.19 Receive Data Sampling Timing in Asynchronous Mode

15.1 Features

- Selection of I^2C format or clocked synchronous serial format
- 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.

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 NMOS open-drain outputs when the bus d function is selected.

Clocked synchronous format

- Four interrupt sources

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

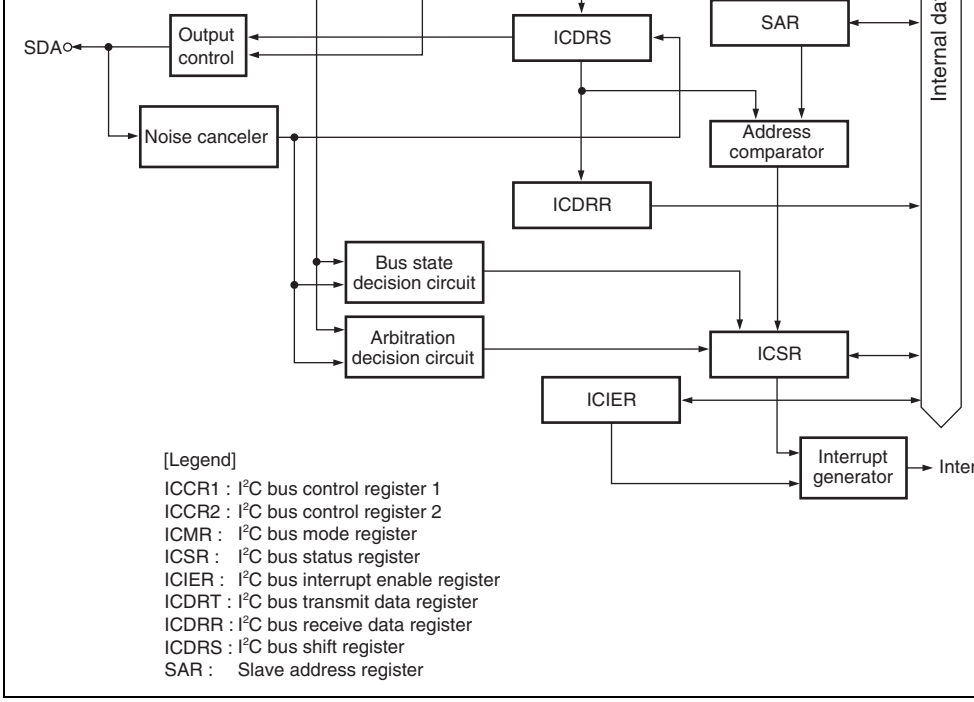


Figure 15.1 Block Diagram of I²C Bus Interface 2

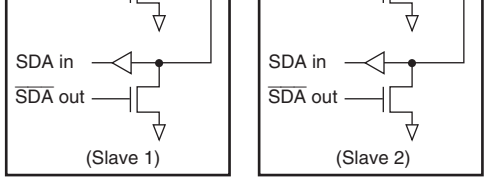


Figure 15.2 External Circuit Connections of I/O Pins

15.2 Input/Output Pins

Table 15.1 summarizes the input/output pins used by the I²C bus interface 2.

Table 15.1 I²C Bus Interface Pins

Name	Abbreviation	I/O	Function
Serial clock	SCL	I/O	IIC serial clock input/output
Serial data	SDA	I/O	IIC serial data input/output

15.3 Register Descriptions

The I²C bus interface 2 has the following registers:

- I²C bus control register 1 (ICCR1)
- I²C bus control register 2 (ICCR2)
- I²C bus mode register (ICMR)
- I²C bus interrupt enable register (ICIER)
- I²C bus status register (ICSR)
- I²C bus slave address register (SAR)

7	ICE	0	R/W	I^2C Bus Interface Enable 0: This module is halted. (SCL and SDA pins are s function.) 1: This bit is enabled for transfer operations. (SCL pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable This bit enables or disables the next operation wh 0 and ICDRR is read. 0: Enables next reception 1: Disables next reception
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select In master mode with the I^2C bus format, when arbi lost, MST and TRS are both reset by hardware, ca transition to slave receive mode. Modification of th should be made between transfer frames. After data receive has been started in slave receiv when the first seven bits of the receive data agree slave address that is set to SAR and the eighth bit TRS is automatically set to 1. If an overrun error o master mode with the clock synchronous serial for MST is cleared to 0 and slave receive mode is ent Operating modes are described below according t and TRS combination. When clocked synchronous format is selected and MST is 1, clock is output. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode

Bit 3	Bit 2	Bit 1	Bit 0		Transfer Rate			
CKS3	CKS2	CKS1	CKS0	Clock	ϕ = 5 MHz	ϕ = 8 MHz	ϕ = 10 MHz	ϕ = 16 MHz
0	0	0	0	ϕ/28	179 kHz	286 kHz	357 kHz	571 kHz
			1	ϕ/40	125 kHz	200 kHz	250 kHz	400 kHz
		1	0	ϕ/48	104 kHz	167 kHz	208 kHz	333 kHz
			1	ϕ/64	78.1 kHz	125 kHz	156 kHz	250 kHz
	1	0	0	ϕ/80	62.5 kHz	100 kHz	125 kHz	200 kHz
			1	ϕ/100	50.0 kHz	80.0 kHz	100 kHz	160 kHz
		1	0	ϕ/112	44.6 kHz	71.4 kHz	89.3 kHz	143 kHz
			1	ϕ/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz
1	0	0	0	ϕ/56	89.3 kHz	143 kHz	179 kHz	286 kHz
			1	ϕ/80	62.5 kHz	100 kHz	125 kHz	200 kHz
		1	0	ϕ/96	52.1 kHz	83.3 kHz	104 kHz	167 kHz
			1	ϕ/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz
	1	0	0	ϕ/160	31.3 kHz	50.0 kHz	62.5 kHz	100 kHz
			1	ϕ/200	25.0 kHz	40.0 kHz	50.0 kHz	80.0 kHz
		1	0	ϕ/224	22.3 kHz	35.7 kHz	44.6 kHz	71.4 kHz
			1	ϕ/256	19.5 kHz	31.3 kHz	39.1 kHz	62.5 kHz

				format, this bit has no meaning. With the I^2C bus this bit is set to 1 when the SDA level changes fr to low under the condition of SCL = high, assum the start condition has been issued. This bit is cl 0 when the SDA level changes from low to high condition of SCL = high, assuming that the stop has been issued. Write 1 to BBSY and 0 to SCP a start condition. Follow this procedure when als transmitting a start condition. Write 0 in BBSY ar SCP to issue a stop condition. To issue start/sto conditions, use the MOV instruction.
6	SCP	1	W	Start/Stop Issue Condition Disable The SCP bit controls the issue of start/stop cond master mode. To issue a start condition, write 1 in BBSY and 0 A retransmit start condition is issued in the same issue a stop condition, write 0 in BBSY and 0 in This bit is always read as 1. If 1 is written, the da stored.
5	SDAO	1	R/W	SDA Output Value Control This bit is used with SDAOP when modifying out of SDA. This bit should not be manipulated durin transfer. 0: When reading, SDA pin outputs low. When writing, SDA pin is changed to output lo 1: When reading, SDA pin outputs high. When writing, SDA pin is changed to output H (outputs high by external pull-up resistance).

				This bit is always read as 1, and cannot be mod
1	IICRST	0	R/W	IIC Control Part Reset This bit resets the control part except for I²C reg this bit is set to 1 when hang-up occurs becaus communication failure during I²C operation, I²C part can be reset without setting ports and initia registers.
0	—	1	—	Reserved This bit is always read as 1, and cannot be mod

15.3.3 I²C Bus Mode Register (ICMR)

ICMR selects whether the MSB or LSB is transferred first, performs master mode wait c and selects the transfer bit count.

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I²C bus format is used
6	WAIT	0	R/W	Wait Insertion Bit In master mode with the I²C bus format, this bit whether to insert a wait after data transfer exce acknowledge bit. When WAIT is set to 1, after the clock for the final data bit, low period is ext two transfer clocks. If WAIT is cleared to 0, dat acknowledge bits are transferred consecutively wait inserted. The setting of this bit is invalid in slave mode w bus format or with the clocked synchronous se

When writing, settings of BC2 to BC0 are inva

2	BC2	0	R/W	Bit Counter 2 to 0
1	BC1	0	R/W	These bits specify the number of bits to be trans
0	BC0	0	R/W	next. When read, the remaining number of trans indicated. With the I^2C bus format, the data is tra with one addition acknowledge bit. Bit BC2 to B settings should be made during an interval betw transfer frames. If bits BC2 to BC0 are set to a v other than 000, the setting should be made whil SCL pin is low. The value returns to 000 at the e data transfer, including the acknowledge bit. Wit clock synchronous serial format, these bits shou modified.

I^2C Bus Format	Clock Synchronous Serial
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

				1: Transmit data empty interrupt request (TXI) i
6	TEIE	0	R/W	Transmit End Interrupt Enable This bit enables or disables the transmit end int (TEI) at the rising of the ninth clock while the TD ICSR is 1. TEI can be canceled by clearing the or the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabl 1: Transmit end interrupt request (TEI) is enabl
5	RIE	0	R/W	Receive Interrupt Enable This bit enables or disables the receive data ful request (RXI) and the overrun error interrupt re (ERI) with the clocked synchronous format, whe receive data is transferred from ICDRS to ICDR RDRF bit in ICSR is set to 1. RXI can be cance clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) and error interrupt request (ERI) with the clocked synchronous format are disabled. 1: Receive data full interrupt request (RXI) and error interrupt request (ERI) with the clocked synchronous format are enabled.
4	NAKIE	0	R/W	NACK Receive Interrupt Enable This bit enables or disables the NACK receive i request (NAKI) and the overrun error (setting o bit in ICSR) interrupt request (ERI) with the cloc synchronous format, when the NACKF and AL ICSR are set to 1. NAKI can be canceled by cle NACKF, OVE, or NAKIE bit to 0. 0: NACK receive interrupt request (NAKI) is dis 1: NACK receive interrupt request (NAKI) is ena

				1: If the receive acknowledge bit is 1, continuous is halted.
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowledge that are returned by the receive device. This bit c 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 se 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-transfe been issued • When transmit mode is entered from receiv slave mode [Clearing conditions] • When 0 is written in TDRE after reading TD • When data is written to ICDRT with an instr
6	TEND	0	R/W	Transmit End [Setting conditions] • When the ninth clock of SCL rises with the I format while the TDRE flag is 1 • When the final bit of transmit frame is sent v clock synchronous serial format [Clearing conditions] • When 0 is written in TEND after reading TE • When data is written to ICDRT with an instr
5	RDRF	0	R/W	Receive Data Register Full [Setting condition] • When a receive data is transferred from ICD ICDRR [Clearing conditions] • When 0 is written in RDRF after reading RD • When ICDRR is read with an instruction

[Setting conditions]

- In master mode, when a stop condition is det
  after frame transfer
- In slave mode, when a stop condition is dete
  after the general call address or the first byte
  address, next to detection of start condition, a
  with the address set in SAR

[Clearing condition]

- When 0 is written in STOP after reading STO

				[Setting conditions] • If the internal SDA and SDA pin disagree at t SCL in master transmit mode • When the SDA pin outputs high in master mo a start condition is detected • When the final bit is received with the clocke synchronous format while RDRF = 1 [Clearing condition] • When 0 is written in AL/OVE after reading AL
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if the f following a start condition matches bits SVA6 to SAR. [Setting conditions] • When the slave address is detected in slave mode • When the general call address is detected in receive mode. [Clearing condition] • When 0 is written in AAS after reading AAS =
0	ADZ	0	R/W	General Call Address Recognition Flag This bit is valid in I^2C bus format slave receive m [Setting condition] • When the general call address is detected in receive mode [Clearing condition] • When 0 is written in ADZ after reading ADZ =

				connected to the I²C bus.
0	FS	0	R/W	Format Select 0: I²C bus format is selected. 1: Clocked synchronous serial format is selected

ICDRR is an 8-bit register that stores the receive data. When data of one byte is received transfers the receive data from ICDRS to ICDRR and the next data can be received. ICD receive-only register, therefore the CPU cannot write to this register. The initial value o is H'FF.

15.3.9 I^2C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferr ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred ICDRS to ICDRR after data of one byte is received. This register cannot be read directly CPU.

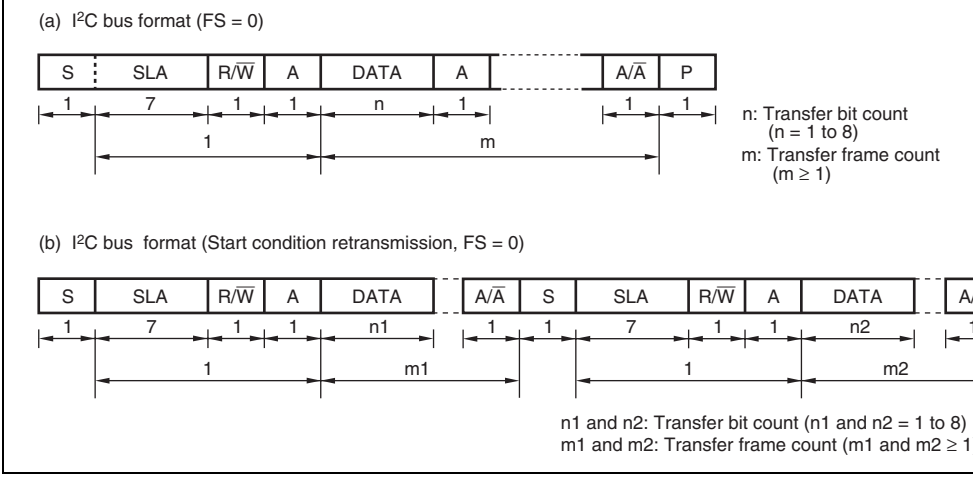


Figure 15.3 I²C Bus Formats

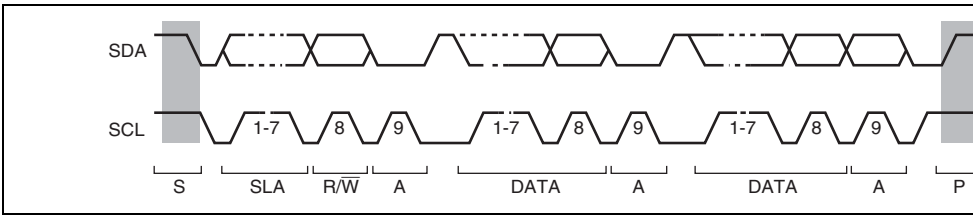


Figure 15.4 I²C Bus Timing

Legend

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

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)
2. Read the BBSY flag in ICCR2 to confirm that the bus is free. Set the MST and TRS 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

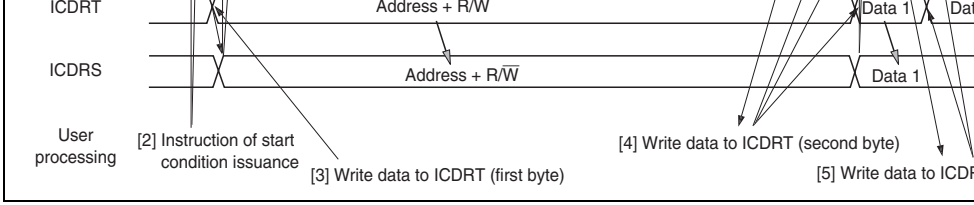


Figure 15.5 Master Transmit Mode Operation Timing (1)

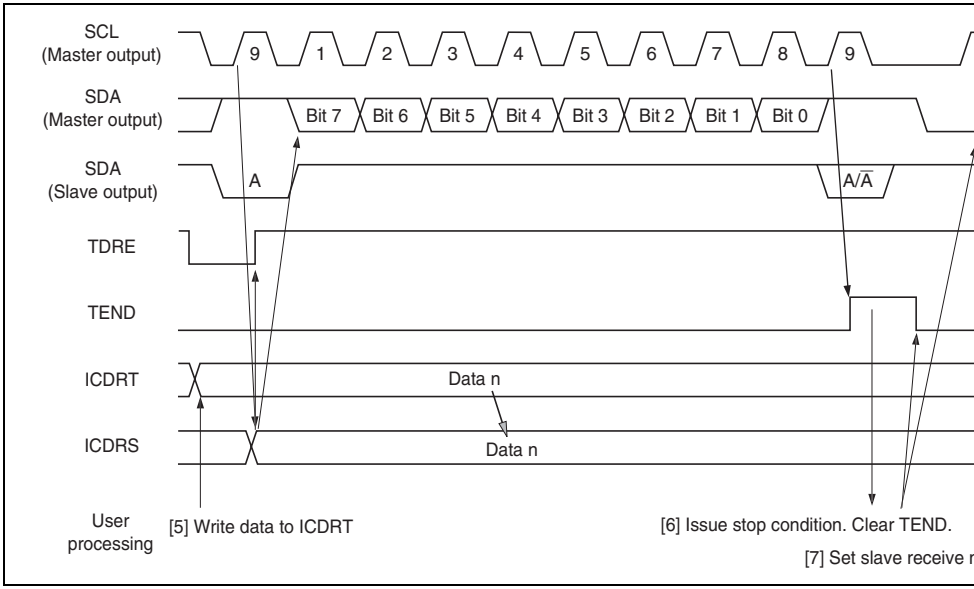


Figure 15.6 Master Transmit Mode Operation Timing (2)

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.

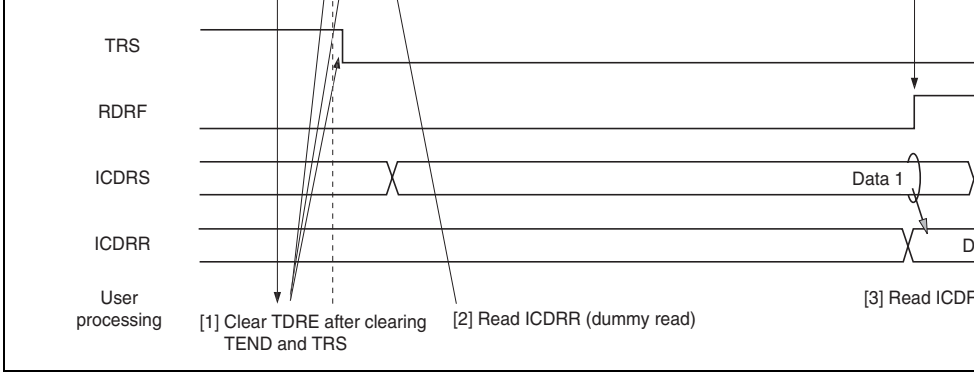


Figure 15.7 Master Receive Mode Operation Timing (1)

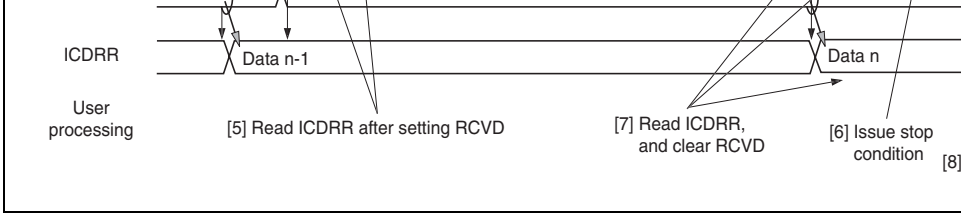


Figure 15.8 Master Receive Mode Operation Timing (2)

15.4.4 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 15.9 and 15.10.

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.

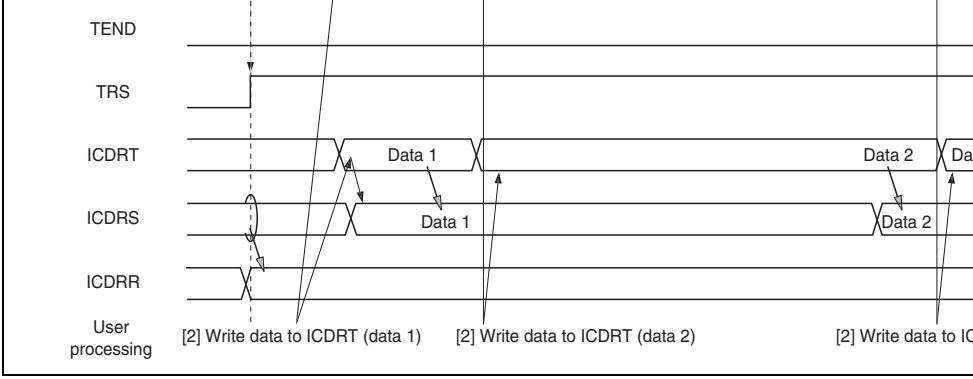


Figure 15.9 Slave Transmit Mode Operation Timing (1)

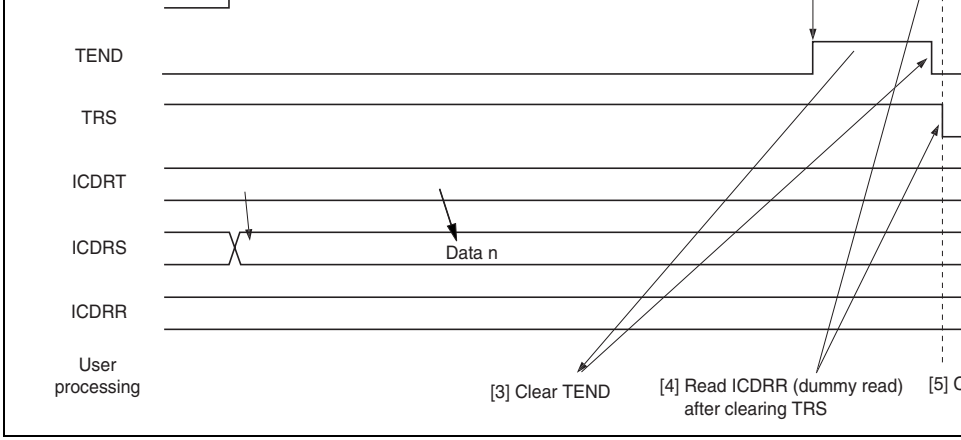


Figure 15.10 Slave Transmit Mode Operation Timing (2)

15.4.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, ar slave device returns an acknowledge signal. For slave receive mode operation timing, re figures 15.11 and 15.12. The reception procedure and operations in slave receive mode a described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 bits in ICCR1 to 1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slav 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 the same time, RDRF in ICSR is set to read ICDRR (dummy read). (read data show the slave address and R/$\overline{W}$, it is not used.)

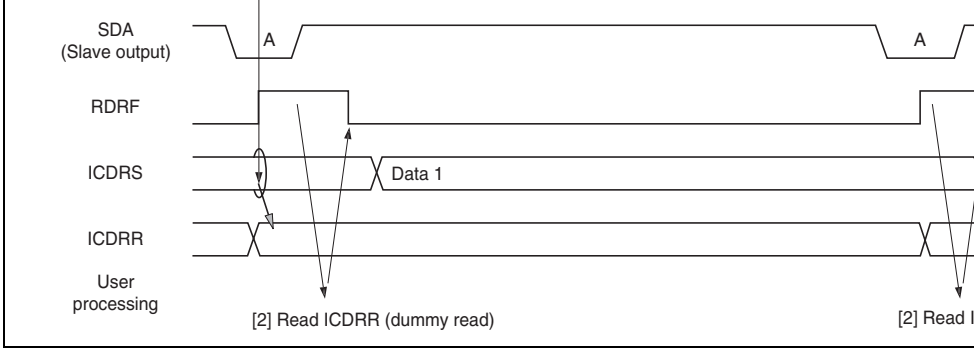


Figure 15.11 Slave Receive Mode Operation Timing (1)

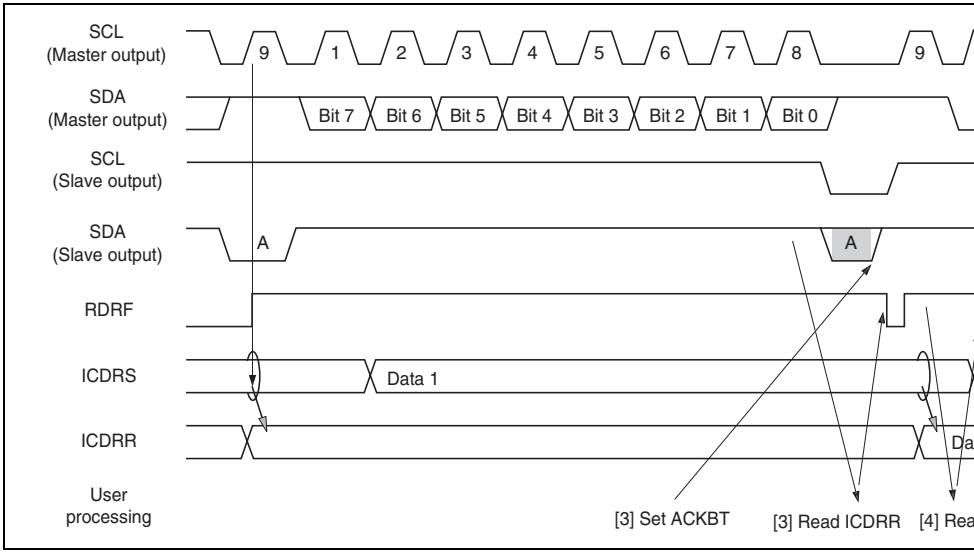


Figure 15.12 Slave Receive Mode Operation Timing (2)

MSB first or LSB first. The output level of SDA can be changed during the transfer wai
SDAO bit in ICCR2.

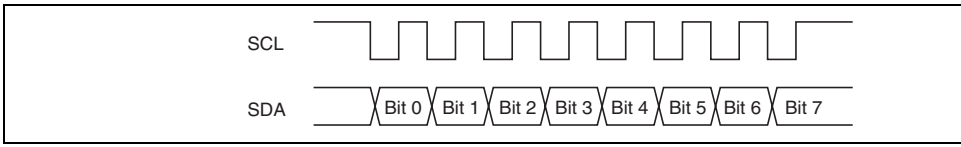


Figure 15.13 Clocked Synchronous Serial Transfer Format

Transmit Operation

In transmit mode, transmit data is output from SDA, in synchronization with the fall of t
clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is
transmit mode operation timing, refer to figure 15.14. The transmission procedure and o
in transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1 to 1. (
   setting)
2. Set the TRS bit in ICCR1 to select the transmit mode. Then, TDRE in ICSR is set.
3. Confirm that TDRE has been set. Then, write the transmit data to ICDRT. The data i
   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
   from transmit mode to receive mode, clear TRS while TDRE is 1.

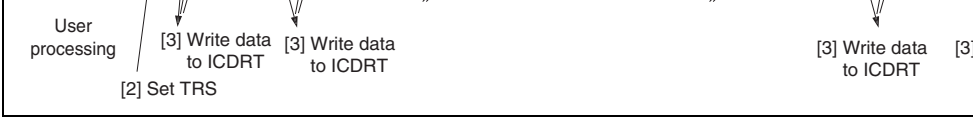


Figure 15.14 Transmit Mode Operation Timing

Receive Operation

In receive mode, data is latched at the rise of the transfer clock. The transfer clock is outp MST in ICCR1 is 1, and is input when MST is 0. For receive mode operation timing, refe figure 15.15. The reception procedure and operations in receive mode are described belov

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1 to 1. (I setting)
2. When the transfer clock is output, set MST to 1 to start outputting the receive clock.
3. When the receive operation is completed, data is transferred from ICDRS to ICDRR a RDRF in ICSR is set. When MST = 1, the next byte can be received, so the clock is continually output. The continuous reception is performed by reading ICDRR every ti RDRF is set. When the 8th clock is risen while RDRF is 1, the overrun is detected and AL/OVE in ICSR is set. At this time, the previous reception data is retained in ICDRI
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.

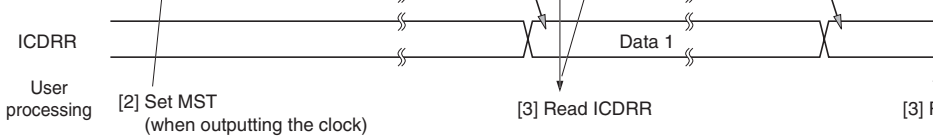


Figure 15.15 Receive Mode Operation Timing

15.4.7 Noise Canceler

The logic levels at the SCL and SDA pins are routed through noise cancelers before bei
internally. Figure 15.16 shows a block diagram of the noise canceler circuit.

The noise canceler consists of two cascaded latches and a match detector. The SCL (or S
input signal is sampled on the system clock, but is not passed forward to the next circuit
outputs of both latches agree. If they do not agree, the previous value is held.

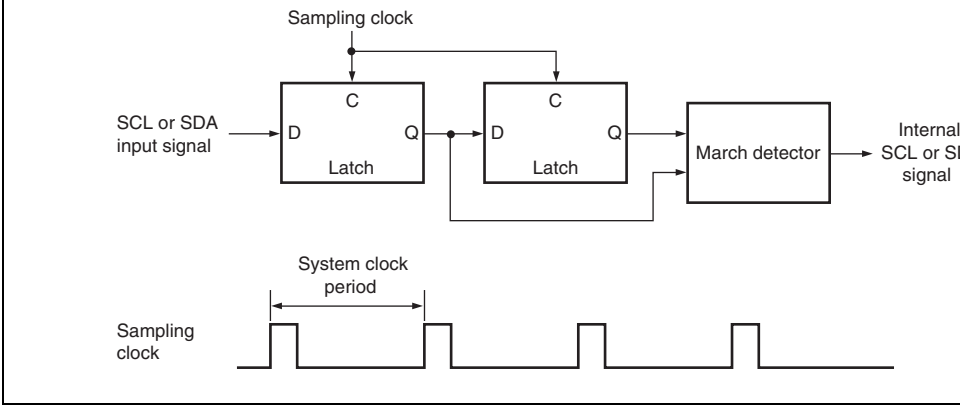


Figure 15.16 Block Diagram of Noise Conceler

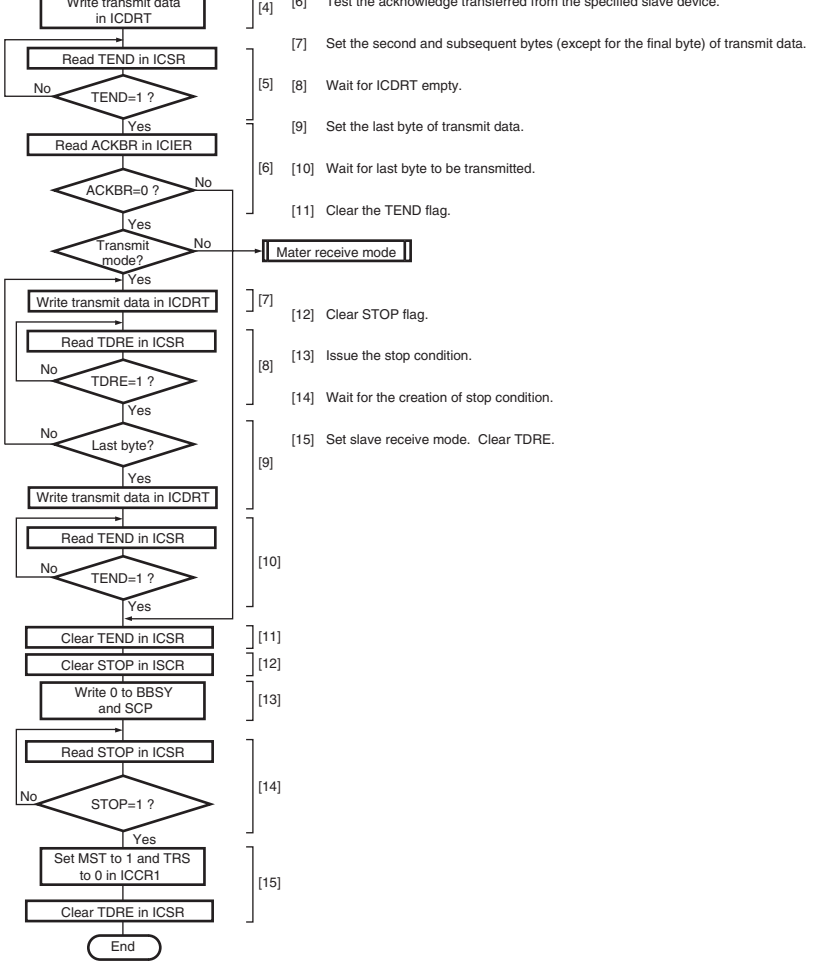
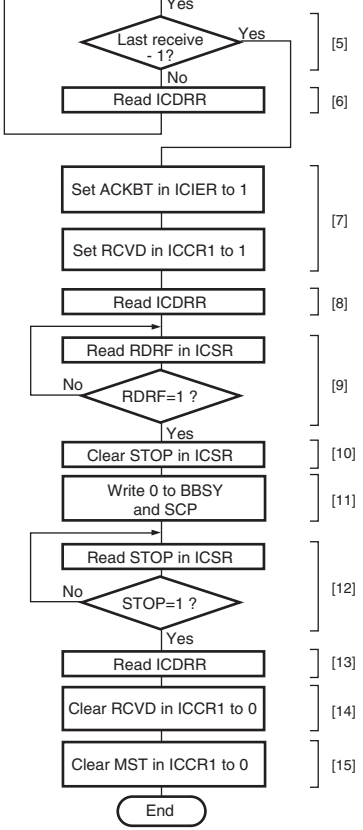


Figure 15.17 Sample Flowchart for Master Transmit Mode



[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.

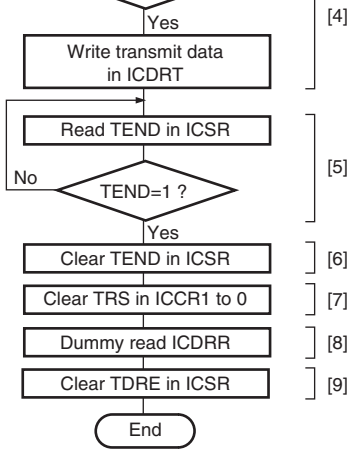
[14] Clear RCVD.

[15] Set slave receive mode.

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

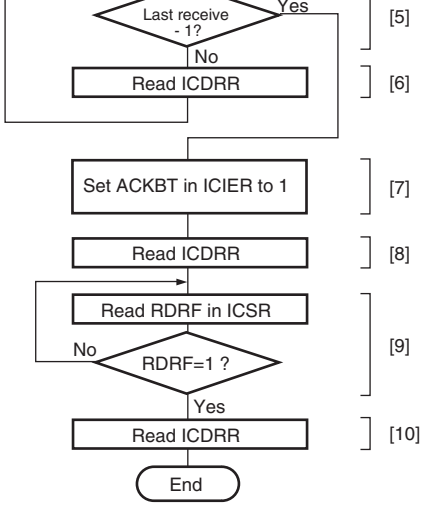
Supplementary explanation: When one byte is received, steps [2] to [6] a
skipped after step [1], before jumping to ste
The step [8] is dummy-read in ICDRR.

Figure 15.18 Sample Flowchart for Master Receive Mode



[9] Clear the TDRE flag.

Figure 15.19 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.

Supplementary explanation: When one byte is received, steps [2] to [6] are skipped after step [1], before jumping to step [7]. The step [8] is dummy-read in ICDRR.

Figure 15.20 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 15.3 are 1 and the I bit in CCR is 0, the CPU executes an interrupt exception processing. Interrupt sources should be cleared in the exc 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 again same time when transmit data is written to ICDRT. When TDRE is cleared to 0, then an e data of one byte may be transmitted.

Figure 15.21 shows the timing of the bit synchronous circuit and table 15.4 shows the ti
SCL output changes from low to Hi-Z then SCL is monitored.

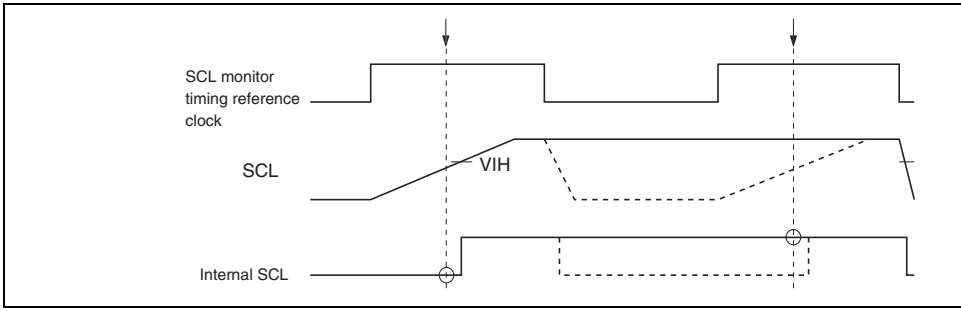


Figure 15.21 The Timing of the Bit Synchronous Circuit

Table 15.4 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

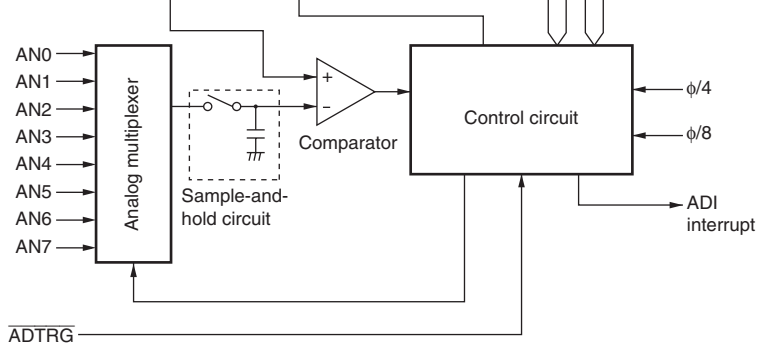
Circuit, by the load of the SCL bus (load capacitance or pull-up resistance)

2. When the bit synchronous circuit is activated by extending the low period of eighth an clocks, that is driven by the slave device

15.7.2 WAIT Setting in I²C Bus Mode Register (ICMR)

If the WAIT bit is set to 1, and the SCL signal is driven low for two or more transfer cloc slave device at the eighth and ninth clocks, the high period of ninth clock may be shorten avoid this, set the WAIT bit in ICMR to 0.

- Conversion time: at least 3.5 µs per channel (at 20-MHz operation)
- Two operating modes
  - Single mode: Single-channel A/D conversion
  - Scan mode: Continuous A/D conversion on 1 to 4 channels
- Four data registers
  - Conversion results are held in a data register for each channel
- Sample-and-hold function
- Two conversion start methods
  - Software
  - External trigger signal
- Interrupt request
  - An A/D conversion end interrupt request (ADI) can be generated



[Legend]
ADCR: A/D control register
ADCSR: A/D control/status register
ADDRA: A/D data register A
ADDRB: A/D data register B
ADDRC: A/D data register C
ADDRD: A/D data register D

Figure 16.1 Block Diagram of A/D Converter

Analog input pin 0	AN0	Input	Group 0 analog input
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	Group 1 analog input
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for s A/D conversion

16.3.1 A/D Data Registers A to D (ADDRA to ADDRD)

There are four 16-bit read-only ADDR registers; ADDRA to ADDRD, used to store the r
A/D conversion. The ADDR registers, which store a conversion result for each analog in
channel, are shown in table 16.2.

The converted 10-bit data is stored in bits 15 to 6. The lower 6 bits are always read as 0.

The data bus width between the CPU and the A/D converter is 8 bits. The upper byte can
directly from the CPU, however the lower byte should be read via a temporary register. T
temporary register contents are transferred from the ADDR when the upper byte data is r
Therefore byte access to ADDR should be done by reading the upper byte first then the lo
Word access is also possible. ADDR is initialized to H'0000.

Table 16.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel		
Group 0	**Group 1**	**A/D Data Register to Be Stored Results of A/D Conversi**
AN0	AN4	ADDRA
AN1	AN5	ADDRB
AN2	AN6	ADDRC
AN3	AN7	ADDRD

				channels selected in scan mode [Clearing condition] • When 0 is written after reading ADF = 1
6	ADIE	0	R/W	A/D Interrupt Enable A/D conversion end interrupt request (ADI) is by ADF when this bit is set to 1
5	ADST	0	R/W	A/D Start Setting this bit to 1 starts A/D conversion. In mode, this bit is cleared to 0 automatically wh conversion on the specified channel is compl scan mode, conversion continues sequentiall specified channels until this bit is cleared to 0 software, a reset, or a transition to standby m
4	SCAN	0	R/W	Scan Mode Selects single mode or scan mode as the A/D conversion operating mode. 0: Single mode 1: Scan mode
3	CKS	0	R/W	Clock Select Selects the A/D conversions time. 0: Conversion time = 134 states (max.) 1: Conversion time = 70 states (max.) Clear the ADST bit to 0 before switching the c time.

	101: AN5	101: AN4 and AN5
	110: AN6	110: AN4 to AN6
	111: AN7	111: AN4 to AN7

16.3.3 A/D Control Register (ADCR)

ADCR enables A/D conversion started by an external trigger signal.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGE	0	R/W	Trigger Enable A/D conversion is started at the falling edge an rising edge of the external trigger signal ($\overline{\text{ADTF}}$ when this bit is set to 1. The selection between the falling edge and risi of the external trigger pin ($\overline{\text{ADTRG}}$) conforms t WPEG5 bit in the interrupt edge select register (IEGR2)
6 to 1	—	All 1	—	Reserved These bits are always read as 1.
0	—	0	R/W	Reserved Do not set this bit to 1, though the bit is readable/writable.

channel as follows:

1. A/D conversion is started when the ADST bit in ADCSR is set to 1, according to sof external trigger input.
2. When A/D conversion is completed, the result is transferred to the corresponding A/ register of the channel.
3. On completion of conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit is s this time, an ADI interrupt request is generated.
4. The ADST 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.

16.4.2 Scan Mode

In scan mode, A/D conversion is performed sequentially for the analog input of the spec channels (four channels maximum) as follows:

1. When the ADST bit in ADCSR is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN0 when CH2 = 0, AN4 when C
2. When A/D conversion for each channel is completed, the result is sequentially transf the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF flag in ADCSR If the ADIE bit is set to 1 at this time, an ADI interrupt requested is generated. A/D c starts again on the first channel in the group.
4. The ADST bit is not automatically cleared to 0. Steps [2] and [3] are repeated as lon ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops

In scan mode, the values given in table 16.3 apply to the first conversion time. In the secc
subsequent conversions, the conversion time is 128 states (fixed) when CKS = 0 and 66 s
(fixed) when CKS = 1.

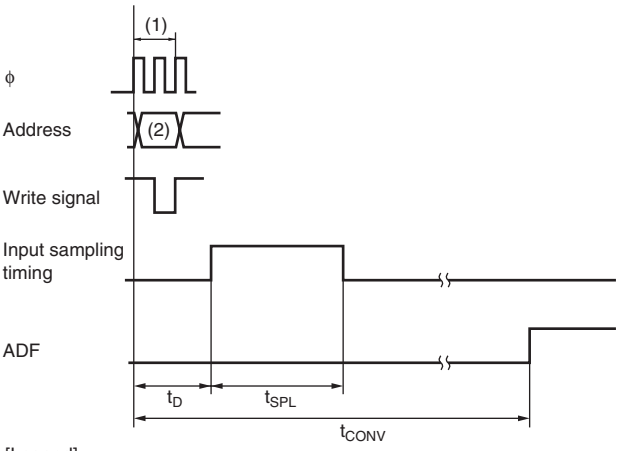


Figure 16.2 A/D Conversion Timing

16.4.4 External Trigger Input Timing

A/D conversion can also be started by an external trigger input. When the TRGE bit in A set to 1, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{ADTR}}$ pin sets the ADST bit in ADCSR to 1, starting A/D conversion. Other operations, in bo and scan modes, are the same as when the bit ADST has been set to 1 by software. Figu shows the timing.

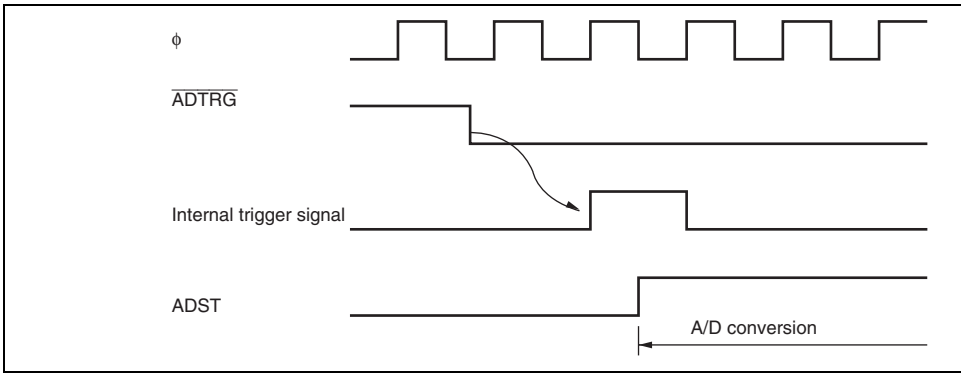


Figure 16.3 External Trigger Input Timing

when the digital output changes from the minimum voltage value 0000000000 to 000
(see figure 16.5).

- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion charac
  when the digital output changes from 1111111110 to 1111111111 (see figure 16.5).

- Nonlinearity error

  The deviation from the ideal A/D conversion characteristic as the voltage changes fro
  full scale. This does not include the offset error, full-scale error, or quantization error

- Absolute accuracy

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

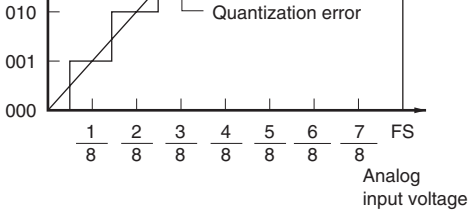


Figure 16.4 A/D Conversion Accuracy Definitions (1)

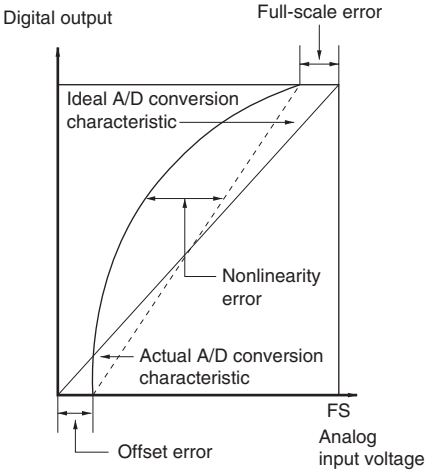


Figure 16.5 A/D Conversion Accuracy Definitions (2)

input resistance of 10 kΩ, and the signal source impedance is ignored. However, as a low filter effect is obtained in this case, it may not be possible to follow an analog signal with differential coefficient (e.g., 5 mV/µs or greater) (see figure 16.6). When converting a hi analog signal or converting in scan mode, a low-impedance buffer should be inserted.

16.6.2 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adve 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 ac antennas on the mounting board.

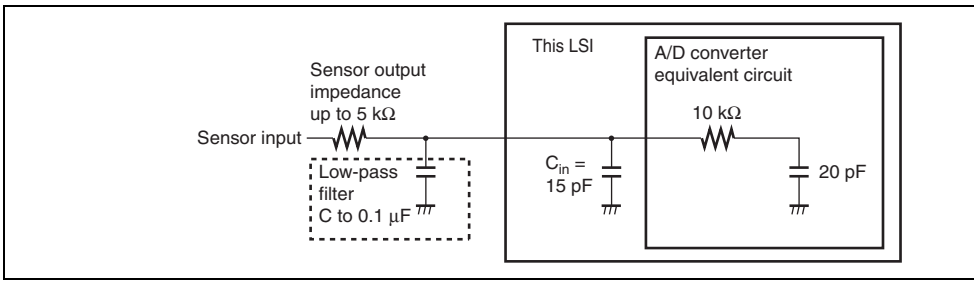


Figure 16.6 Analog Input Circuit Example

- Three reading methods:
  Current address read
  Random address read
  Sequential read
- Acknowledge polling possible
- Write cycle time:
  10 ms (power supply voltage Vcc = 2.7 V or more)
- Write/Erase endurance:
  10^4 cycles/byte (byte write mode), 10^5 cycles/page (page write mode)
- Data retention:
  10 years after the write cycle of 10^4 cycles (page write mode)
- Interface with the CPU
  I^2C bus interface (complies with the standard of Philips Corporation)
  Device code 1010
  Sleep address code can be changed (initial value: 000)
  The I^2C bus is open to the outside, so the EEPROM can be directly accessed from th

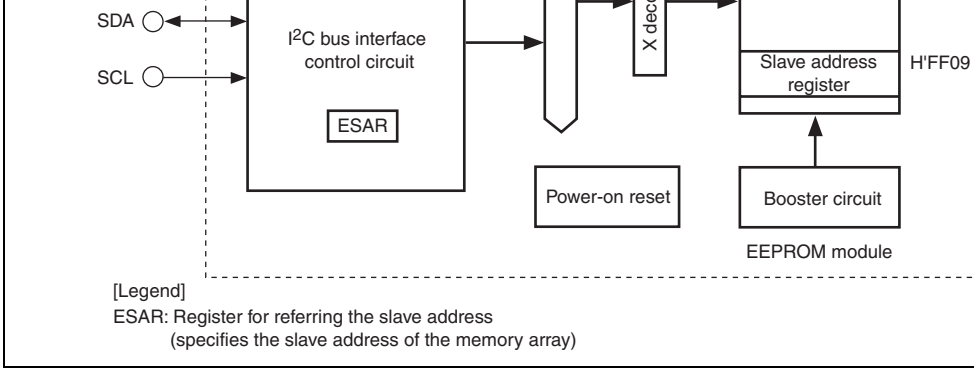


Figure 17.1 Block Diagram of EEPROM

			pin is open-drain driven structure of the I²C proper resistor value for your system by co V_{OL}, I_{OL}, and the C_{IN} pin capacitance in secti DC Characteristics and in section 21.2.3, A Characteristics. Maximum clock frequency kHz.
Serial data pin	SDA	Input/Output	The SDA pin is bidirectional for serial data The SDA pin needs to be pulled up by resis pin is open-drain driven structure. Use pro resistor value for your system by considerin and the C_{IN} pin capacitance in section 21.2 Characteristics and in section 21.2.3, AC Characteristics. Except for a start condition stop condition which will be discussed later to-low and low-to-high change of SDA inpu be done during SCL low periods.

17.3 Register Description

The EEPROM has a following register.

- EEPROM key register (EKR)

17.3.1 EEPROM Key Register (EKR)

EKR is an 8-bit readable/writable register, which changes the slave address code written
EEPROM. The slave address code is changed by writing H'5F in EKR and then writing
H'00 to H'07 as an address code to the H'FF09 address in the EEPROM by the byte writ
EKR is initialized to H'FF.

17.4.2 Bus Format and Timing

The I²C bus format and the I²C bus timing follow section 15.4.1, I²C Bus Format. The bu
specific for the EEPROM are the following two.

1. The EEPROM address is configured of two bytes, the write data is transferred in the o
   upper address and lower address from each MSB side.
2. The write data is transmitted from the MSB side.

The bus format and bus timing of the EEPROM are shown in figure 17.2.

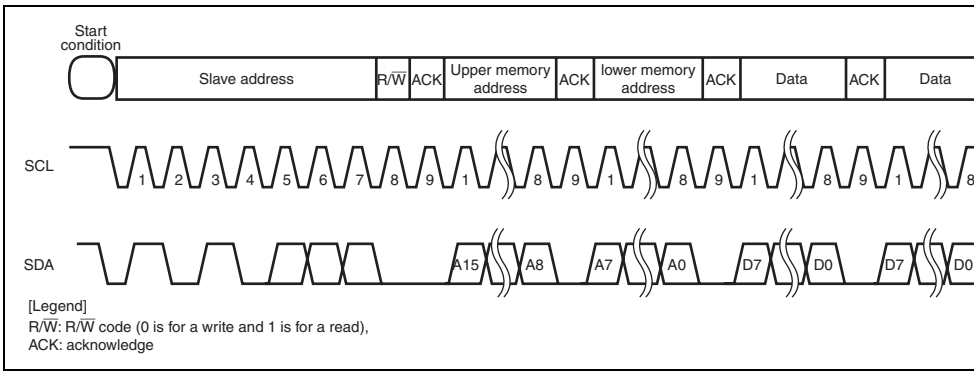


Figure 17.2 EEPROM Bus Format and Bus Timing

17.4.3 Start Condition

A high-to-low transition of the SDA input with the SCL input high is needed to generate
condition for starting read, write operation.

All address data and serial data such as read data and write data are transmitted to and fr
bit unit. The acknowledgement is the signal that indicates that this 8-bit data is normally
transmitted to and from.

In the write operation, EEPROM sends "0" to acknowledge in the ninth cycle after recei
data. In the read operation, EEPROM sends a read data following the acknowledgemen
receiving the data. After sending read data, the EEPROM enters the bus open state. If t
EEPROM receives "0" as an acknowledgement, it sends read data of the next address. I
EEPROM does not receive acknowledgement "0" and receives a following stop conditio
the read operation and enters a standby mode. If the EEPROM receives neither acknow
"0" nor a stop condition, the EEPROM keeps bus open without sending read data.

17.4.6 Slave Addressing

The EEPROM device receives a 7-bit slave address and a 1-bit R/$\overline{W}$ code following the
of the start conditions. The EEPROM enables the chip for a read or a write operation w
operation.

The slave address consists of a former 4-bit device code and latter 3-bit slave address as
table 17.2. The device code is used to distinguish device type and this LSI uses "1010"
in the same manner as in a general-purpose EEPROM. The slave address code selects o
out of all devices with device code 1010 (8 devices in maximum) which are connected t
bus. This means that the device is selected if the inputted slave address code received in
of A2, A1, A0 is equal to the corresponding slave address reference register (ESAR).

The slave address code is stored in the address H'FF09 in the EEPROM. It is transferred
from the slave address register in the memory array during 10 ms after the reset is releas
access to the EEPROM is not allowed during transfer.

7	Device code D3	—	1	
6	Device code D2	—	0	
5	Device code D1	—	1	
4	Device code D0	—	0	
3	Slave address code A2	0	A2	The initial value can be changed
2	Slave address code A1	0	A1	The initial value can be changed
1	Slave address code A0	0	A0	The initial value can be changed

acknowledgement "0". If the EEPROM receives a stop condition, the EEPROM ent
internally controlled write cycle and terminates receipt of SCL and SDA inputs until
completion of the write cycle. The EEPROM returns to a standby mode after compl
the write cycle.

The byte write operation is shown in figure 17.3.

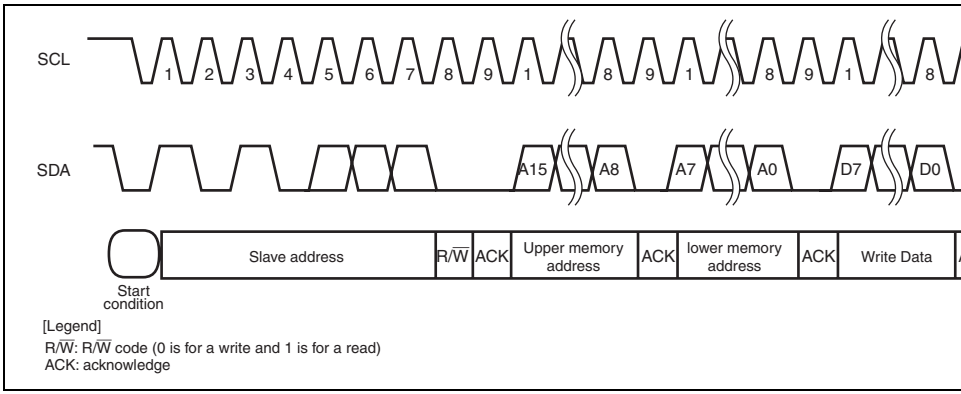


Figure 17.3 Byte Write Operation

2. Page Write

   This LSI is capable of the page write operation which allows any number of bytes up
   to be written in a single write cycle. The write data is input in the same sequence as
   write in the order of a start condition, slave address + R/$\overline{W}$ code, memory address (n
   write data (Dn) with every ninth bit acknowledgement "0" output. The EEPROM en
   page write operation if the EEPROM receives more write data (Dn+1) is input instea
   receiving a stop condition after receiving the write data (Dn). LSB 3 bits (A2 to A0)
   EEPROM address are automatically incremented to be the (n+1) address upon receiv
   data (Dn+1). Thus the write data can be received sequentially.

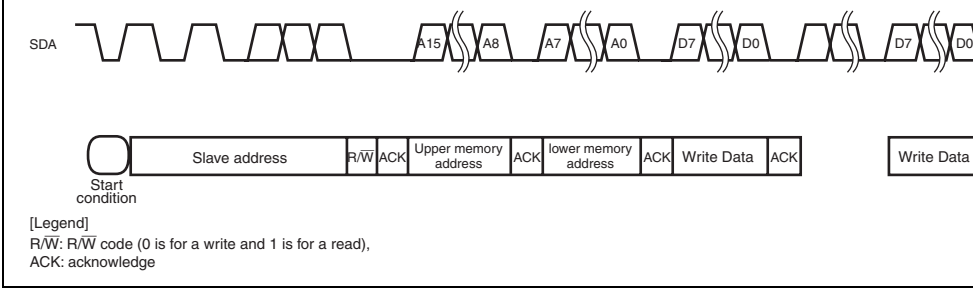


Figure 17.4 Page Write Operation

17.4.8 Acknowledge Polling

Acknowledge polling feature is used to show if the EEPROM is in an internally-timed wr or not. This feature is initiated by the input of the 8-bit slave address + R/$\overline{W}$ code followi start condition during an internally-timed write cycle. Acknowledge polling will operate code = "0". The ninth acknowledgement judges if the EEPROM is an internally-timed w or not. Acknowledgement "1" shows the EEPROM is in a internally-timed write cycle ar acknowledgement "0" shows the internally-timed write cycle has been completed. The acknowledge polling starts to function after a write data is input, i.e., when the stop condi input.

the EEPROM outputs the 1-byte data of the (n+1) address from the most significant
following acknowledgement "0". If the EEPROM receives in the order of acknowled
"1" and a following stop condition, the EEPROM stops the read operation and is turn
standby state.

In case the EEPROM has accessed the last address H'01FF at previous read operatio
current address will roll over and returns to zero address. In case the EEPROM has
the last address of the page at previous write operation, the current address will roll o
page addressing and returns to the first address in the same page.

The current address is valid while power is on. The current address after power on w
undefined. After power is turned on, define the address by the random address read
described below is necessary.

The current address read operation is shown in figure 17.5.

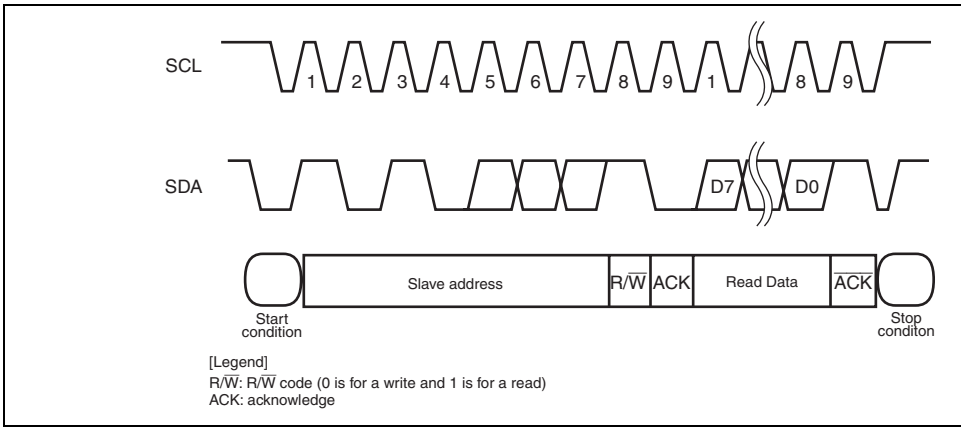


Figure 17.5 Current Address Read Operation

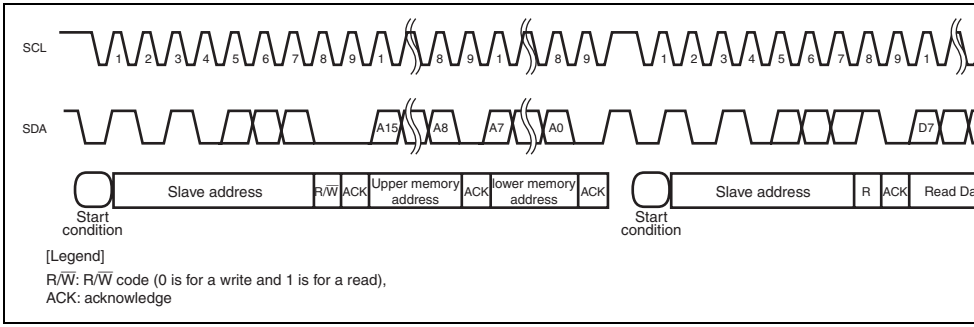


Figure 17.6 Random Address Read Operation

3. Sequential Read

   This is a mode to read the data sequentially. Data is sequential read by either a curren read or a random address read. If the EEPROM receives acknowledgement "0" after read data is output, the read address is incremented and the next 1-byte read data are o out. Data is output sequentially by incrementing addresses as long as the EEPROM re acknowledgement "0" after the data is output. The address will roll over and returns a zero if it reaches the last address H'01FF. The sequential read can be continued after The sequential read is terminated if the EEPROM receives acknowledgement "1" and following stop condition as the same manner as in the random address read.

   The condition of a sequential read when the current address read is used is shown in f 17.7.

[Legend]
R/$\overline{W}$: R/$\overline{W}$ code (0 is for a write and 1 is for a read)
ACK: acknowledge

Figure 17.7 Sequential Read Operation (when current address read is use

turned on from the ground level (V_{ss}).

4. V_{cc} turn on speed should be longer than 10 us.

17.5.2 Write/Erase Endurance

The endurance is 10^5 cycles/page (1% cumulative failure rate) in case of page programmi 10^4 cycles/byte in case of byte programming. The data retention time is more than 10 yea device is page-programmed less than 10^4 cycles.

17.5.3 Noise Suppression Time

This EEPROM has a noise suppression function at SCL and SDA inputs, that cuts noise o less than 50 ns. Be careful not to allow noise of width more than 50 ns because the noise more than 50 ms is recognized as an active pulse.

power supply voltage rises again.

Even if the power supply voltage falls, the unstable state when the power supply voltage below the guaranteed operating voltage can be removed by entering standby mode when exceeding the guaranteed operating voltage and during normal operation. Thus, system s can be improved. If the power supply voltage falls more, the reset state is automatically the power supply voltage rises again, the reset state is held for a specified period, then a is automatically entered.

Figure 18.1 is a block diagram of the power-on reset circuit and the low-voltage detectio

18.1 Features

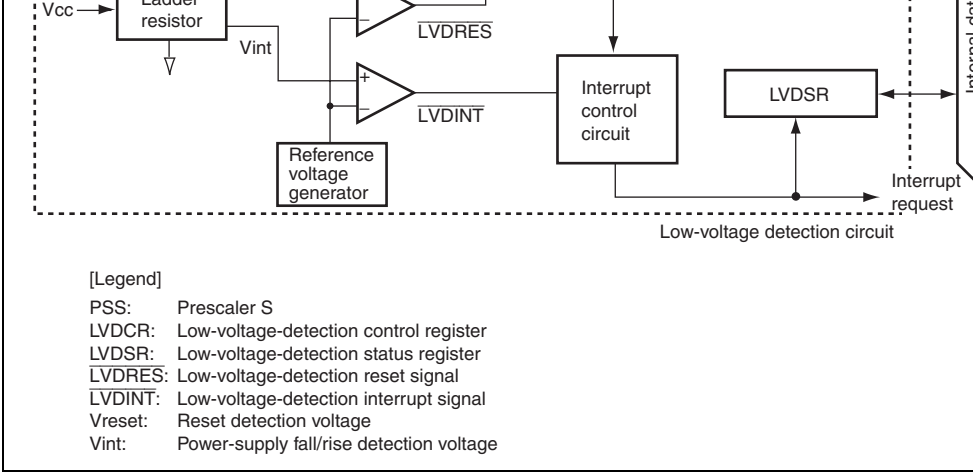
- Power-on reset circuit

  Uses an external capacitor to generate an internal reset signal when power is first sup
- Low-voltage detection circuit

  LVDR: Monitors the power-supply voltage, and generates an internal reset signal wh voltage falls below a specified value.

  LVDI: Monitors the power-supply voltage, and generates an interrupt when the volta below or rises above respective specified values.

  Two pairs of detection levels for reset generation voltage are available: when only th circuit is used, or when the LVDI and LVDR circuits are both used.



[Legend]

PSS:	Prescaler S
LVDCR:	Low-voltage-detection control register
LVDSR:	Low-voltage-detection status register
$\overline{\text{LVDRES}}$:	Low-voltage-detection reset signal
$\overline{\text{LVDINT}}$:	Low-voltage-detection interrupt signal
Vreset:	Reset detection voltage
Vint:	Power-supply fall/rise detection voltage

Figure 18.1 Block Diagram of Power-On Reset Circuit and Low-Voltage Detection

18.2 Register Descriptions

The low-voltage detection circuit has the following registers.

- Low-voltage-detection control register (LVDCR)
- Low-voltage-detection status register (LVDSR)

18.2.1 Low-Voltage-Detection Control Register (LVDCR)

LVDCR is used to enable or disable the low-voltage detection circuit, set the detection le the LVDR function, enable or disable the LVDR function, and enable or disable generatio interrupt when the power-supply voltage rises above or falls below the respective levels.

				These bits are always read as 1, and cannot be
3	LVDSEL	0*	R/W	LVDR Detection Level Select 0: Reset detection voltage is 2.3 V (typ.) 1: Reset detection voltage is 3.6 V (typ.) When the falling or rising voltage detection inte used, reset detection voltage of 2.3 V (typ.) sho used. When only a reset detection interrupt is u detection voltage of 3.6 V (typ.) should be used
2	LVDRE	0*	R/W	LVDR Enable 0: Disables the LVDR function 1: Enables the LVDR function
1	LVDDE	0	R/W	Voltage-Fall-Interrupt Enable 0: Interrupt on the power-supply voltage falling selected detection level disabled 1: Interrupt on the power-supply voltage falling selected detection level enabled
0	LVDUE	0	R/W	Voltage-Rise-Interrupt Enable 0: Interrupt on the power-supply voltage rising a selected detection level disabled 1: Interrupt on the power-supply voltage rising a selected detection level enabled

Note: * Not initialized by LVDR but initialized by a power-on reset or WDT reset.

1	0	1	1	1	O	O	O	O

Legend: *: means invalid.

18.2.2 Low-Voltage-Detection Status Register (LVDSR)

LVDSR indicates whether the power-supply voltage falls below or rises above the respec
specified values.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 1	—	Reserved These bits are always read as 1, and cannot be
1	LVDDF	0*	R/W	LVD Power-Supply Voltage Fall Flag [Setting condition] When the power-supply voltage falls below Vint = 3.7 V) [Clearing condition] Writing 0 to this bit after reading it as 1
0	LVDUF	0*	R/W	LVD Power-Supply Voltage Rise Flag [Setting condition] When the power supply voltage falls below Vint the LVDUE bit in LVDCR is set to 1, then rises a (U) (typ. = 4.0 V) before falling below Vreset1 (ty V) [Clearing condition] Writing 0 to this bit after reading it as 1

Note: * Initialized by LVDR.

prevent the incorrect operation of the chip by noise on the $\overline{RES}$ pin.

To achieve stable operation of this LSI, the power supply needs to rise to its full level ar within the specified time. The maximum time required for the power supply to rise and s power has been supplied (t_{PWON}) is determined by the oscillation frequency (f_{OSC}) and capa which is connected to $\overline{RES}$ pin ($C_{\overline{RES}}$). If t_{PWON} means the time required to reach 90 % of supply voltage, the power supply circuit should be designed to satisfy the following form

$$t_{PWON}\ (ms) \leq 90 \times C_{\overline{RES}}\ (\mu F) + 162/f_{OSC}\ (MHz)$$

($t_{PWON} \leq 3000$ ms, $C_{\overline{RES}} \geq 0.22$ µF, and $f_{OSC} = 10$ in 2-MHz to 10-MHz operatio

Note that the power supply voltage (Vcc) must fall below Vpor = 100 mV and rise after the $\overline{RES}$ pin is removed. To remove charge on the $\overline{RES}$ pin, it is recommended that the c should be placed near Vcc. If the power supply voltage (Vcc) rises from the point above power-on reset may not occur.

Figure 18.2 Operational Timing of Power-On Reset Circuit

18.3.2 Low-Voltage Detection Circuit

LVDR (Reset by Low Voltage Detect) Circuit:

Figure 18.3 shows the timing of the LVDR function. The LVDR enters the module-stand after a power-on reset is canceled. To operate the LVDR, set the LVDE bit in LVDCR to for 50 µs (t_{LVDON}) until the reference voltage and the low-voltage-detection power supply h stabilized by a software timer, etc., then set the LVDRE bit in LVDCR to 1. After that, th settings of ports must be made. To cancel the low-voltage detection circuit, first the LVD should be cleared to 0 and then the LVDE bit should be cleared to 0. The LVDE and LVI must not be cleared to 0 simultaneously because incorrect operation may occur.

When the power-supply voltage falls below the Vreset voltage (typ. = 2.3 V or 3.6 V), th clears the $\overline{\text{LVDRES}}$ signal to 0, and resets the prescaler S. The low-voltage detection rese remains in place until a power-on reset is generated. When the power-supply voltage rise the Vreset voltage again, the prescaler S starts counting. It counts 131,072 clock (ϕ) cycle then releases the internal reset signal. In this case, the LVDE, LVDSEL, and LVDRE bits LVDCR are not initialized.

Note that if the power supply voltage (Vcc) falls below $V_{LVDRmin}$ = 1.0 V and then rises fro point, the low-voltage detection reset may not occur.

If the power supply voltage (Vcc) falls below Vpor = 100 mV, a power-on reset occurs.

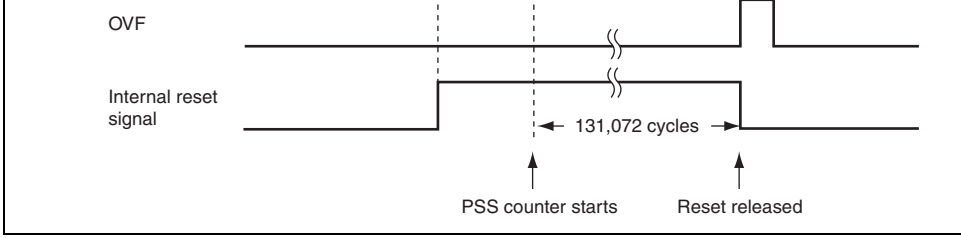


Figure 18.3 Operational Timing of LVDR Circuit

LVDI (Interrupt by Low Voltage Detect) Circuit:

Figure 18.4 shows the timing of LVDI functions. The LVDI enters the module-standby s a power-on reset is canceled. To operate the LVDI, set the LVDE bit in LVDCR to 1, w µs (t_{LVDON}) until the reference voltage and the low-voltage-detection power supply have s by a software timer, etc., then set the LVDDE and LVDUE bits in LVDCR to 1. After th output settings of ports must be made. To cancel the low-voltage detection circuit, first t LVDDE and LVDUE bits should all be cleared to 0 and then the LVDE bit should be cl The LVDE bit must not be cleared to 0 at the same timing as the LVDDE and LVDUE b because incorrect operation may occur.

When the power-supply voltage falls below Vint (D) (typ. = 3.7 V) voltage, the LVDI cl $\overline{\text{LVDINT}}$ signal to 0 and the LVDDF bit in LVDSR is set to 1. If the LVDDE bit is 1 at an IRQ0 interrupt request is simultaneously generated. In this case, the necessary data m saved in the external EEPROM, etc, and a transition must be made to standby mode or s mode. Until this processing is completed, the power supply voltage must be higher than limit of the guaranteed operating voltage.

When the power-supply voltage does not fall below Vreset1 (typ. = 2.3 V) voltage but ri Vint (U) (typ. = 4.0 V) voltage, the LVDI sets the $\overline{\text{LVDINT}}$ signal to 1. If the LVDUE b

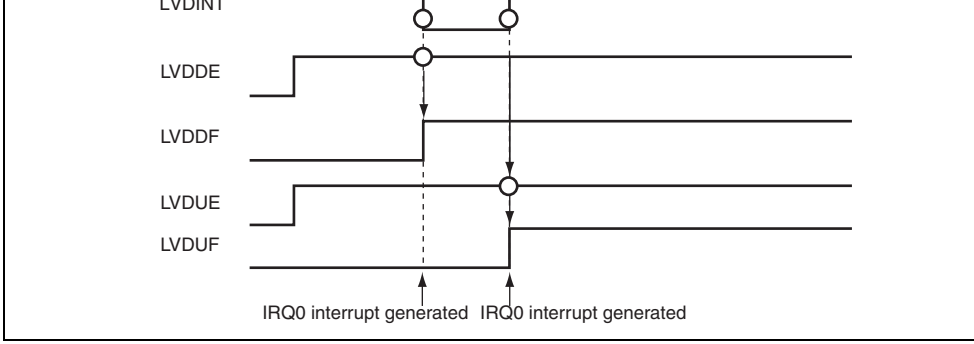


Figure 18.4 Operational Timing of LVDI Circuit

LVDUE bits to 0. Then clear the LVDE bit to 0. The LVDE bit must not be cleared t
same timing as the LVDRE, LVDDE, and LVDUE bits because incorrect operation

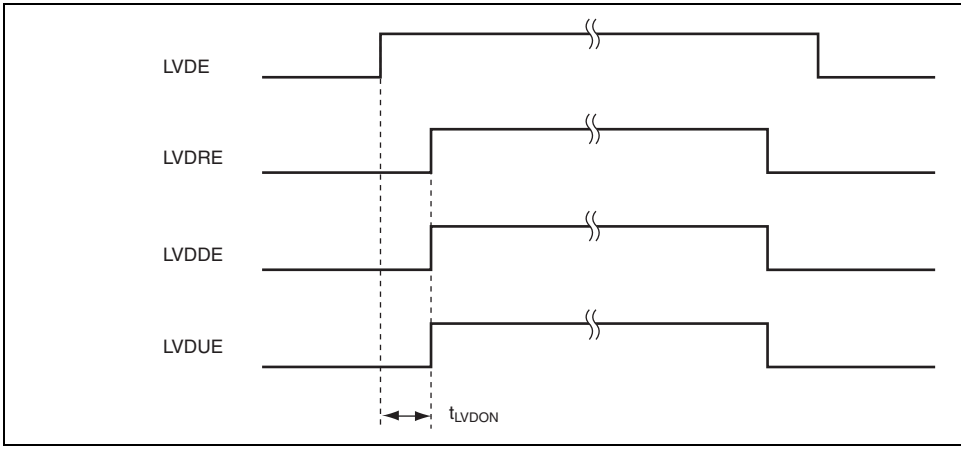


Figure 18.5 Timing for Operation/Release of Low-Voltage Detection Circu

19.1 When Using Internal Power Supply Step-Down Circuit

Connect the external power supply to the V_{CC} pin, and connect a capacitance of approxin
µF between V_{CL} and V_{SS}, as shown in figure 19.1. The internal step-down circuit is made
simply by adding this external circuit. In the external circuit interface, the external powe
voltage connected to V_{CC} and the GND potential connected to V_{SS} are the reference levels
example, for port input/output levels, the V_{CC} level is the reference for the high level, and
level is that for the low level. The A/D converter analog power supply is not affected by
internal step-down circuit.

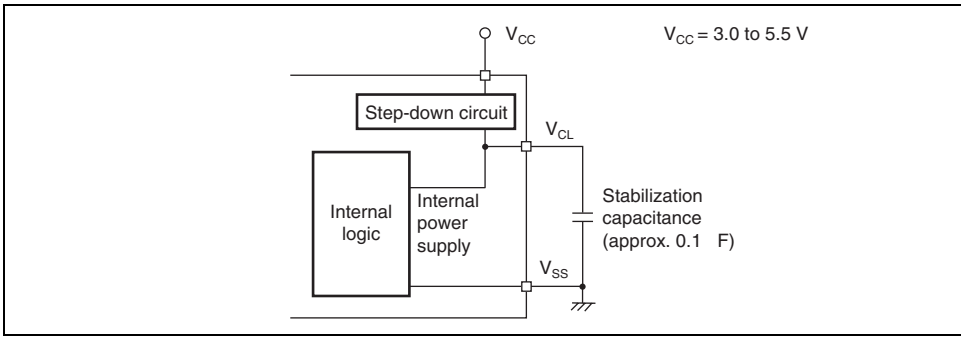


Figure 19.1 Power Supply Connection when Internal Step-Down Circuit is U

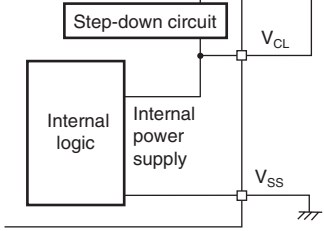


Figure 19.2 Power Supply Connection when Internal Step-Down Circuit is Not

Do not attempt to access reserved addresses.

- When the address is 16-bit wide, the address of the upper byte is given in the list.
- Registers are classified by functional modules.
- The data bus width is indicated.
- The number of access states is indicated.

2. Register bits

- Bit configurations of the registers are described in the same order as the register addr
- Reserved bits are indicated by — in the bit name column.
- 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

—	—	—	H'F000 to H'F72F	—	—
Low-voltage detection control register	LVDCR	8	H'F730	LVDC*[1]	8
Low-voltage detection status register	LVDSR	8	H'F731	LVDC*[1]	8
—	—	—	H'F732 to H'F747	—	—
I^2C bus control register 1	ICCR1	8	H'F748	IIC2	8
I^2C bus control register 2	ICCR2	8	H'F749	IIC2	8
I^2C bus mode register	ICMR	8	H'F74A	IIC2	8
I^2C bus interrupt enable register	ICIER	8	H'F74B	IIC2	8
I^2C bus status register	ICSR	8	H'F74C	IIC2	8
Slave address register	SAR	8	H'F74D	IIC2	8
I^2C bus transmit data register	ICDRT	8	H'F74E	IIC2	8
I^2C bus receive data register	ICDRR	8	H'F74F	IIC2	8
—	—	—	H'F750 to H'FF7F	—	—
Timer mode register W	TMRW	8	H'FF80	Timer W	8
Timer control register W	TCRW	8	H'FF81	Timer W	8
Timer interrupt enable register W	TIERW	8	H'FF82	Timer W	8
Timer status register W	TSRW	8	H'FF83	Timer W	8
Timer I/O control register 0	TIOR0	8	H'FF84	Timer W	8
Timer I/O control register 1	TIOR1	8	H'FF85	Timer W	8
Timer counter	TCNT	16	H'FF86	Timer W	16*[2]
General register A	GRA	16	H'FF88	Timer W	16*[2]

—	—	—	H'FF94 to H'FF9A	—	—
Flash memory enable register	FENR	8	H'FF9B	ROM	8
—	—	—	H'FF9C to H'FF9F	—	—
Timer control register V0	TCRV0	8	H'FFA0	Timer V	8
Timer control/status register V	TCSRV	8	H'FFA1	Timer V	8
Timer constant register A	TCORA	8	H'FFA2	Timer V	8
Timer constant register B	TCORB	8	H'FFA3	Timer V	8
Timer counter V	TCNTV	8	H'FFA4	Timer V	8
Timer control register V1	TCRV1	8	H'FFA5	Timer V	8
Timer mode register A	TMA	8	H'FFA6	Timer A	8
Timer counter A	TCA	8	H'FFA7	Timer A	8
Serial mode register	SMR	8	H'FFA8	SCI3	8
Bit rate register	BRR	8	H'FFA9	SCI3	8
Serial control register 3	SCR3	8	H'FFAA	SCI3	8
Transmit data register	TDR	8	H'FFAB	SCI3	8
Serial status register	SSR	8	H'FFAC	SCI3	8
Receive data register	RDR	8	H'FFAD	SCI3	8
—	—	—	H'FFAE, H'FFAF	—	—
A/D data register A	ADDRA	16	H'FFB0	A/D converter	8

A/D control register	ADCR	8	H'FFB9	A/D converter	8
—	—	—	H'FFBA to H'FFBF	—	—
Timer control/status register WD	TCSRWD	8	H'FFC0	WDT*[3]	8
Timer counter WD	TCWD	8	H'FFC1	WDT*[3]	8
Timer mode register WD	TMWD	8	H'FFC2	WDT*[3]	8
—	—	—	H'FFC3	—	—
—	—	—	H'FFC4 to H'FFC7	—	—
Address break control register	ABRKCR	8	H'FFC8	Address break	8
Address break status register	ABRKSR	8	H'FFC9	Address break	8
Break address register H	BARH	8	H'FFCA	Address break	8
Break address register L	BARL	8	H'FFCB	Address break	8
Break data register H	BDRH	8	H'FFCC	Address break	8
Break data register L	BDRL	8	H'FFCD	Address break	8
—	—	—	H'FFCE, H'FFCF	—	—

			H'FFD7		
Port data register 5	PDR5	8	H'FFD8	I/O port	8
—	—	—	H'FFD9	I/O port	—
Port data register 7	PDR7	8	H'FFDA	I/O port	8
Port data register 8	PDR8	8	H'FFDB	I/O port	8
—	—	—	H'FFDC	I/O port	—
Port data register B	PDRB	8	H'FFDD	I/O port	8
—	—	—	H'FFDE, H'FFDF	I/O port	—
Port mode register 1	PMR1	8	H'FFE0	I/O port	8
Port mode register 5	PMR5	8	H'FFE1	I/O port	8
—	—	—	H'FFE2, H'FFE3	I/O port	—
Port control register 1	PCR1	8	H'FFE4	I/O port	8
Port control register 2	PCR2	8	H'FFE5	I/O port	8
—	—	—	H'FFE6, H'FFE7	I/O port	—
Port control register 5	PCR5	8	H'FFE8	I/O port	8
—	—	—	H'FFE9	I/O port	—
Port control register 7	PCR7	8	H'FFEA	I/O port	8
Port control register 8	PCR8	8	H'FFEB	I/O port	8
—	—	—	H'FFEC to H'FFEF	I/O port	—

—	—	—	H'FFE7	I/O port	—
Wake-up interrupt flag register	IWPR	8	H'FFF8	Interrupts	8
Module standby control register 1	MSTCR1	8	H'FFF9	Power-down	8
—	—	—	H'FFFA to H'FFFF	—	—

- EEPROM

Register Name	Abbre-viation	Bit No	Address	Module Name	Data Bus Width
EEPROM slave address register	—	8	H'FF09	EEPROM	—
EEPROM key register	EKR	8	H'FF10	EEPROM	—

Notes: 1. LVDC: Low-voltage detection circuits (optional)
2. Only word access can be used.
3. WDT: Watchdog timer

ICCR1	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	II
ICCR2	BBSY	SCP	SDAO	SDAOP	SCKO	—	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	
—	—	—	—	—	—	—	—	—	–
TMRW	CTS	—	BUFEB	BUFEA	—	PWMD	PWMC	PWMB	T
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	

TCORB	TCORB7	TCORB6	TCORB5	TCORB4	TCORB3	TCORB2	TCORB1	TCORB0	
TCNTV	TCNTV7	TCNTV6	TCNTV5	TCNTV4	TCNTV3	TCNTV2	TCNTV1	TCNTV0	
TCRV1	—	—	—	TVEG1	TVEG0	TRGE	—	ICKS0	
TMA	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0	Tir
TCA	TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0	
SMR	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0	SC
BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0	
SCR3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	
SSR	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT	
RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0	
ADDRA	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/I
	AD1	AD0	—	—	—	—	—	—	co
ADDRB	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRC	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRD	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0	
ADCR	TRGE	—	—	—	—	—	—	—	
—	—	—	—	—	—	—	—	—	—
TCSRWD	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI	WRST	WI
TCWD	TCWD7	TCWD6	TCWD5	TCWD4	TCWD3	TCWD2	TCWD1	TCWD0	

BDRE	BDRE7	BDRE6	BDRE5	BDRE4	BDRE3	BDRE2	BDRE1	BDRE0	
—	—	—	—	—	—	—	—	—	—
PUCR1	PUCR17	PUCR16	PUCR15	PUCR14	—	PUCR12	PUCR11	PUCR10	I/
PUCR5	—	—	PUCR55	PUCR54	PUCR53	PUCR52	PUCR51	PUCR50	
PDR1	P17	P16	P15	P14	—	P12	P11	P10	
PDR2	—	—	—	—	—	P22	P21	P20	
PDR5	P57*[3]	P56*[3]	P55	P54	P53	P52	P51	P50	
PDR7	—	P76	P75	P74	—	—	—	—	
PDR8	P87	P86	P85	P84	P83	P82	P81	P80	
PDRB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
PMR1	IRQ3	IRQ2	IRQ1	IRQ0	—	—	TXD	TMOW	
PMR5	—	—	WKP5	WKP4	WKP3	WKP2	WKP1	WKP0	
PCR1	PCR17	PCR16	PCR15	PCR14	—	PCR12	PCR11	PCR10	
PCR2	—	—	—	—	—	PCR22	PCR21	PCR20	
PCR5	PCR57*[3]	PCR56*[3]	PCR55	PCR54	PCR53	PCR52	PCR51	PCR50	
PCR7	—	PCR76	PCR75	PCR74	—	—	—	—	
PCR8	PCR87	PCR86	PCR85	PCR84	PCR83	PCR82	PCR81	PCR80	
SYSCR1	SSBY	STS2	STS1	STS0	NESEL	—	—	—	P
SYSCR2	SMSEL	LSON	DTON	MA2	MA1	MA0	SA1	SA0	
IEGR1	NMIEG	—	—	—	IEG3	IEG2	IEG1	IEG0	Ir
IEGR2	—	—	WPEG5	WPEG4	WPEG3	WPEG2	WPEG1	WPEG0	
IENR1	IENDT	IENTA	IENWP	—	IEN3	IEN2	IEN1	IEN0	
IRR1	IRRDT	IRRTA	—	—	IRRI3	IRRI2	IRRI1	IRRI0	
IWPR	—	—	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0	
MSTCR1	—	MSTIIC	MSTS3	MSTAD	MSTWD	MSTTW	MSTTV	MSTTA	P
—	—	—	—	—	—	—	—	—	—

ICSR	Initialized	—	—	—	—	—	
SAR	Initialized	—	—	—	—	—	
ICDRT	Initialized	—	—	—	—	—	
ICDRR	Initialized	—	—	—	—	—	
TMRW	Initialized	—	—	—	—	—	Timer W
TCRW	Initialized	—	—	—	—	—	
TIERW	Initialized	—	—	—	—	—	
TSRW	Initialized	—	—	—	—	—	
TIOR0	Initialized	—	—	—	—	—	
TIOR1	Initialized	—	—	—	—	—	
TCNT	Initialized	—	—	—	—	—	
GRA	Initialized	—	—	—	—	—	
GRB	Initialized	—	—	—	—	—	
GRC	Initialized	—	—	—	—	—	
GRD	Initialized	—	—	—	—	—	
FLMCR1	Initialized	—	—	Initialized	Initialized	Initialized	ROM
FLMCR2	Initialized	—	—	—	—	—	
FLPWCR	Initialized	—	—	—	—	—	
EBR1	Initialized	—	—	Initialized	Initialized	Initialized	
FENR	Initialized	—	—	—	—	—	
TCRV0	Initialized	—	—	Initialized	Initialized	Initialized	Timer V
TCSRV	Initialized	—	—	Initialized	Initialized	Initialized	
TCORA	Initialized	—	—	Initialized	Initialized	Initialized	
TCORB	Initialized	—	—	Initialized	Initialized	Initialized	

SSR	Initialized	—	—	Initialized	Initialized	Initialized	
RDR	Initialized	—	—	Initialized	Initialized	Initialized	
ADDRA	Initialized	—	—	Initialized	Initialized	Initialized	A/D con
ADDRB	Initialized	—	—	Initialized	Initialized	Initialized	
ADDRC	Initialized	—	—	Initialized	Initialized	Initialized	
ADDRD	Initialized	—	—	Initialized	Initialized	Initialized	
ADCSR	Initialized	—	—	Initialized	Initialized	Initialized	
ADCR	Initialized	—	—	Initialized	Initialized	Initialized	
TCSRWD	Initialized	—	—	—	—	—	WDT*[2]
TCWD	Initialized	—	—	—	—	—	
TMWD	Initialized	—	—	—	—	—	
ABRKCR	Initialized	—	—	—	—	—	Address
ABRKSR	Initialized	—	—	—	—	—	
BARH	Initialized	—	—	—	—	—	
BARL	Initialized	—	—	—	—	—	
BDRH	Initialized	—	—	—	—	—	
BDRL	Initialized	—	—	—	—	—	
PUCR1	Initialized	—	—	—	—	—	I/O port
PUCR5	Initialized	—	—	—	—	—	
PDR1	Initialized	—	—	—	—	—	
PDR2	Initialized	—	—	—	—	—	
PDR5	Initialized	—	—	—	—	—	
PDR7	Initialized	—	—	—	—	—	
PDR8	Initialized	—	—	—	—	—	
PDRB	Initialized	—	—	—	—	—	

SYSCR2	Initialized	—	—	—	—	—	Power-
IEGR1	Initialized	—	—	—	—	—	Interru
IEGR2	Initialized	—	—	—	—	—	Interru
IENR1	Initialized	—	—	—	—	—	Interru
IRR1	Initialized	—	—	—	—	—	Interru
IWPR	Initialized	—	—	—	—	—	Interru
MSTCR1	Initialized	—	—	—	—	—	Power-

- EEPROM

Register Name	Reset	Active	Sleep	Subactive	Subsleep	Standby	Modul
EKR	—	—	—	—	—	—	EEPRO

Notes: — is not initialized

1. LVDC: Low-voltage detection circuits (optional)
2. WDT: Watchdog timer

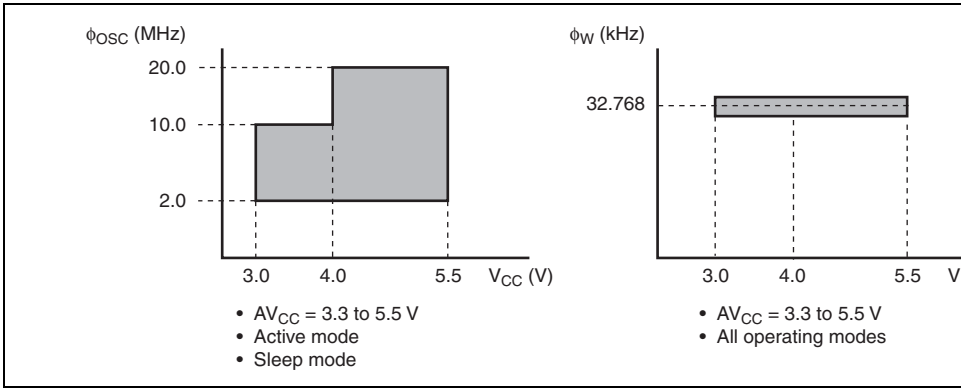
	B and X1			
	Port B		–0.3 to AV_{CC} +0.3	V
	X1		–0.3 to 4.3	V
Operating temperature		T_{opr}	–20 to +75	°C
Storage temperature		T_{stg}	–55 to +125	°C

Note: * Permanent damage may result if maximum ratings are exceeded. Normal ope should be under the conditions specified in Electrical Characteristics. Exceed values can result in incorrect operation and reduced reliability.

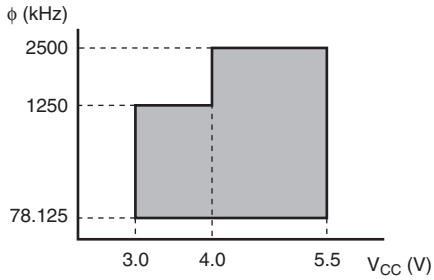
21.2 Electrical Characteristics (F-ZTAT™ Version, EEPROM Sta F-ZTAT™ Version)

21.2.1 Power Supply Voltage and Operating Ranges

Power Supply Voltage and Oscillation Frequency Range



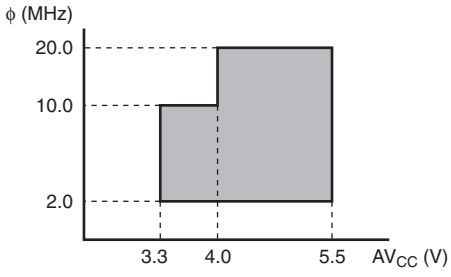
• AV_{CC} = 3.3 to 5.5 V
• Active mode
• Sleep mode
(When MA2 in SYSCR2 = 0)



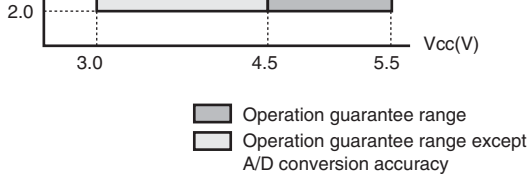
• AV_{CC} = 3.3 to 5.5 V
• Active mode
• Sleep mode
(When MA2 in SYSCR2 = 1)

• AV_{CC} = 3.3 to 5.5 V
• Subactive mode
• Subsleep mode

Analog Power Supply Voltage and A/D Converter Accuracy Guarantee Range



• V_{CC} = 3.0 to 5.5 V
• Active mode
• Sleep mode


2.0
Vcc(V)
3.0
4.5
5.5
Operation guarantee range
Operation guarantee range except
A/D conversion accuracy

		TMCIV, FTCI, FTIOA to FTIOD, SCK3, TRGV		$V_{cc} \times 0.9$	—	$V_{cc} + 0.3$	
		RXD, SCL, SDA, P10 to P12, P14 to P17, P20 to P22, P50 to P57, P74 to P76, P80 to P87	V_{cc} = 4.0 to 5.5 V	$V_{cc} \times 0.7$	—	$V_{cc} + 0.3$	V
				$V_{cc} \times 0.8$	—	$V_{cc} + 0.3$	
		PB0 to PB7	V_{cc} = 4.0 to 5.5 V	$V_{cc} \times 0.7$	—	$AV_{cc} + 0.3$	V
				$V_{cc} \times 0.8$	—	$AV_{cc} + 0.3$	
		OSC1	V_{cc} = 4.0 to 5.5 V	$V_{cc} - 0.5$	—	$V_{cc} + 0.3$	V
				$V_{cc} - 0.3$	—	$V_{cc} + 0.3$	
Input low voltage	V_{IL}	$\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$,TMRIV, TMCIV, FTCI, FTIOA to FTIOD, SCK3, TRGV	V_{cc} = 4.0 to 5.5 V	–0.3	—	$V_{cc} \times 0.2$	V
				–0.3	—	$V_{cc} \times 0.1$	
		RXD, SCL, SDA, P10 to P12, P14 to P17, P20 to P22, P50 to P57, P74 to P76, P80 to P87 PB0 to PB7	V_{cc} = 4.0 to 5.5 V	–0.3	—	$V_{cc} \times 0.3$	V
				–0.3	—	$V_{cc} \times 0.2$	
		OSC1	V_{cc} = 4.0 to 5.5 V	–0.3	—	0.5	V
				–0.3	—	0.3	

			$-I_{OH}$ = 0.1 mA				
Output low voltage	V_{OL}	P10 to P12, P14 to P17, P20 to P22, P50 to P57, P74 to P76	V_{CC} = 4.0 to 5.5 VI_{OL} = 1.6 mA	—	—	0.6	V
			I_{OL} = 0.4 mA	—	—	0.4	
		P80 to P87	V_{CC} = 4.0 to 5.5 V I_{OL} = 20.0 mA	—	—	1.5	V
			V_{CC} = 4.0 to 5.5 V I_{OL} = 10.0 mA	—	—	1.0	
			V_{CC} = 4.0 to 5.5 V I_{OL} = 1.6 mA	—	—	0.4	
			I_{OL} = 0.4 mA	—	—	0.4	
		SCL, SDA	V_{CC} = 4.0 to 5.5 V I_{OL} = 6.0 mA	—	—	0.6	V
			I_{OL} = 3.0 mA	—	—	0.4	
Input/ output leakage current	$\lvert I_{IL} \rvert$	OSC1, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$, TRGV, TMRIV, TMCIV, FTCI, FTIOA to FTIOD, RXD, SCK3, SCL, SDA	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	—	—	1.0	µA
		P10 to P12, P14 to P17, P20 to P22, P50 to P57, P74 to P76, P80 to P87	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	—	—	1.0	µA
		PB0 to PB7	V_{IN} = 0.5 V to (AV_{CC} – 0.5 V)	—	—	1.0	µA

mode current consumption			V_{CC} = 5.0 V, f_{OSC} = 20 MHz					
			Active mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	8.0	—		* Refe valu
	I_{OPE2}	V_{CC}	Active mode 2 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	2.0	3.0	mA	*
			Active mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	1.2	—		* Refe valu
Sleep mode current consumption	I_{SLEEP1}	V_{CC}	Sleep mode 1 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	16.0	22.5	mA	*
			Sleep mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	8.0	—		* Refe valu
	I_{SLEEP2}	V_{CC}	Sleep mode 2 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	1.8	2.7	mA	*
			Sleep mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	1.2	—		* Refe valu
Subactive mode current consumption	I_{SUB}	V_{CC}	V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	40.0	70.0	µA	*
			V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/8$)	—	30.0	—		* Refe valu

RAM data retaining voltage	V_{RAM}	V_{CC}	2.0	—	—	V

Note: * Pin states during current consumption measurement are given below (exclud in the pull-up MOS transistors and output buffers).

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pi
Active mode 1	V_{CC}	Operates	V_{CC}	Main clock: ceramic or cr resonator
Active mode 2		Operates (ϕOSC/64)		Subclock: Pin X1 = V_{SS}
Sleep mode 1	V_{CC}	Only timers operate	V_{CC}	
Sleep mode 2		Only timers operate (ϕOSC/64)		
Subactive mode	V_{CC}	Operates	V_{CC}	Main clock: ceramic or cr resonator
Subsleep mode	V_{CC}	Only timers operate	V_{CC}	Subclock: crystal reson
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	Main clock: ceramic or cr resonator
				Subclock: Pin X1 = V_{SS}

Note: * The current consumption of the EEPROM chip is shown.
For the current consumption of H8/3694N, add the above current values to the
consumption of H8/3694F.

		SCL and SDA		—	—	6.0
		Output pins except port 8, SCL, and SDA		—	—	0.5
Allowable output low current (total)	ΣI_{OL}	Output pins except port 8, SCL, and SDA	V_{cc} = 4.0 to 5.5 V	—	—	40.0
		Port 8, SCL, and SDA		—	—	80.0
		Output pins except port 8, SCL, and SDA		—	—	20.0
		Port 8, SCL, and SDA		—	—	40.0
Allowable output high current (per pin)	$\lvert -I_{OH} \rvert$	All output pins	V_{cc} = 4.0 to 5.5 V	—	—	2.0
				—	—	0.2
Allowable output high current (total)	$\lvert -\Sigma I_{OH} \rvert$	All output pins	V_{cc} = 4.0 to 5.5 V	—	—	30.0
				—	—	8.0

cycle time				—	—	12.8	µs	
Subclock oscillation frequency	f_W	X1, X2		—	32.768	—	kHz	
Watch clock (ϕ_W) cycle time	t_W	X1, X2		—	30.5	—	µs	
Subclock (ϕ_{SUB}) cycle time	t_{subcyc}			2	—	8	t_W	*
Instruction cycle time				2	—	—	t_{cyc} t_{subcyc}	
Oscillation stabilization time (crystal resonator)	t_{rc}	OSC1, OSC2		—	—	10.0	ms	
Oscillation stabilization time (ceramic resonator)	t_{rc}	OSC1, OSC2		—	—	5.0	ms	
Oscillation stabilization time	t_{rcx}	X1, X2		—	—	2.0	s	
External clock high width	t_{CPH}	OSC1	V_{CC} = 4.0 to 5.5 V	20.0	—	—	ns	F
				40.0	—	—		
External clock low width	t_{CPL}	OSC1	V_{CC} = 4.0 to 5.5 V	20.0	—	—	ns	
				40.0	—	—		
External clock rise time	t_{CPr}	OSC1	V_{CC} = 4.0 to 5.5 V	—	—	10.0	ns	
				—	—	15.0		
External clock fall time	t_{CPf}	OSC1	V_{CC} = 4.0 to 5.5 V	—	—	10.0	ns	
				—	—	15.0		
RES pin low width	t_{REL}	$\overline{RES}$	At power-on and in modes other than those below	t_{rc}	—	—	ms	F
			In active mode and sleep mode operation	200	—	—	ns	

		FTIOD				
Input pin low width	t_{IL}	$\overline{NMI}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{WKP0}$ to $\overline{WKP5}$, TMCIV, TMRIV, TRGV, $\overline{ADTRG}$, FTCI, FTIOA to FTIOD	2	—	—	t_{cyc} t_{subcyc}

Notes: 1. When an external clock is input, the minimum system clock oscillation frequ
1.0 MHz.
2. Determined by MA2, MA1, MA0, SA1, and SA0 of system control register 2 (S

SCL and SDA input spike pulse removal time	t_{SP}		—	—	$1t_{cyc}$	ns
SDA input bus-free time	t_{BUF}		$5t_{cyc}$	—	—	ns
Start condition input hold time	t_{STAH}		$3t_{cyc}$	—	—	ns
Retransmission start condition input setup time	t_{STAS}		$3t_{cyc}$	—	—	ns
Setup time for stop condition input	t_{STOS}		$3t_{cyc}$	—	—	ns
Data-input setup time	t_{SDAS}		$1t_{cyc}$+20	—	—	ns
Data-input hold time	t_{SDAH}		0	—	—	ns
Capacitive load of SCL and SDA	c_b		0	—	400	pF
SCL and SDA output fall time	t_{Sf}	V_{CC} = 4.0 to 5.5 V	—	—	250	ns
			—	—	300	

width	t_{SCKW}						t_{Scyc}	
Transmit data delay time (clocked synchronous)	t_{TXD}	TXD	V_{CC} = 4.0 V to 5.5 V	—	—	1	t_{cyc}	F
				—	—	1		
Receive data setup time (clocked synchronous)	t_{RXS}	RXD	V_{CC} = 4.0 V to 5.5 V	50.0	—	—	ns	
				100.0	—	—		
Receive data hold time (clocked synchronous)	t_{RXH}	RXD	V_{CC} = 4.0 V to 5.5 V	50.0	—	—	ns	
				100.0	—	—		

Analog power supply current	AI_{OPE}	AV_{CC}	AV_{CC} = 5.0 V f_{OSC} = 20 MHz	—	—	2.0	mA
	AI_{STOP1}	AV_{CC}		—	50	—	µA
	AI_{STOP2}	AV_{CC}		—	—	5.0	µA
Analog input capacitance	C_{AIN}	AN0 to AN7		—	—	30.0	pF
Allowable signal source impedance	R_{AIN}	AN0 to AN7		—	—	5.0	kΩ
Resolution (data length)				10	10	10	bit
Conversion time (single mode)			AV_{CC} = 3.3 to 5.5 V	134	—	—	t_{cyc}
Nonlinearity error				—	—	±7.5	LSB
Offset error				—	—	±7.5	LSB
Full-scale error				—	—	±7.5	LSB
Quantization error				—	—	±0.5	LSB
Absolute accuracy				—	—	±8.0	LSB
Conversion time (single mode)			AV_{CC} = 4.0 to 5.5 V	70	—	—	t_{cyc}
Nonlinearity error				—	—	±7.5	LSB
Offset error				—	—	±7.5	LSB
Full-scale error				—	—	±7.5	LSB
Quantization error				—	—	±0.5	LSB
Absolute accuracy				—	—	±8.0	LSB

2. AI_{STOP1} is the current in active and sleep modes while the A/D converter is idle
3. AI_{STOP2} is the current at reset and in standby, subactive, and subsleep modes
A/D converter is idle.

21.2.5 Watchdog Timer Characteristics

Table 21.7 Watchdog Timer Characteristics

V_{CC} = 3.0 to 5.5 V, V_{SS} = 0.0 V, T_a = –20 to +75°C, unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit
				Min	Typ	Max	
On-chip oscillator overflow time	t_{OVF}			0.2	0.4	—	s

Note: * Shows the time to count from 0 to 255, at which point an internal reset is gen
when the internal oscillator is selected.

Programming	Wait time after SWE bit setting*[1]	x		1	—	—
	Wait time after PSU bit setting*[1]	y		50	—	—
	Wait time after P bit setting *[1]*[4]	z1	1 ≤ n ≤ 6	28	30	32
		z2	7 ≤ n ≤ 1000	198	200	202
		z3	Additional-programming	8	10	12
	Wait time after P bit clear*[1]	α		5	—	—
	Wait time after PSU bit clear*[1]	β		5	—	—
	Wait time after PV bit setting*[1]	γ		4	—	—
	Wait time after dummy write*[1]	ε		2	—	—
	Wait time after PV bit clear*[1]	η		2	—	—
	Wait time after SWE bit clear*[1]	θ		100	—	—
	Maximum programming count *[1]*[4]*[5]	N		—	—	1000

	Wait time after EV bit setting*[1]	γ	20	—	—
	Wait time after dummy write*[1]	ε	2	—	—
	Wait time after EV bit clear*[1]	η	4	—	—
	Wait time after SWE bit clear*[1]	θ	100	—	—
	Maximum erase count *[1]*[6]*[7]	N	—	—	120

Notes: 1. Make the time settings in accordance with the program/erase algorithms.

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

3. The time required to erase one block. (Indicates the time for which the E bit i memory control register 1 (FLMCR1) is set. The erase-verify time is not inclu

4. Programming time maximum value (t_P (max.)) = wait time after P bit setting (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 P bit setting (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. Erase time maximum value (t_E (max.)) = wait time after E bit setting (z) × max erase count (N)

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

SCL, SDA input spike pulse removal time	t_{SP}	—	—	50	ns
SDA input bus-free time	t_{BUF}	1200	—	—	ns
Start condition input hold time	t_{STAH}	600	—	—	ns
Retransmit start condition input setup time	t_{STAS}	600	—	—	ns
Stop condition input setup time	t_{STOS}	600	—	—	ns
Data input setup time	t_{SDAS}	160	—	—	ns
Data input hold time	t_{SDAH}	0	—	—	ns
SCL, SDA input fall time	t_{Sf}	—	—	300	ns
SDA input rise time	t_{Sr}	—	—	300	ns
Data output hold time	t_{DH}	50	—	—	ns
SCL, SDA capacitive load	C_b	0	—	400	pF
Access time	t_{AA}	100	—	900	ns
Cycle time at writing*	t_{WC}	—	—	10	ms
Reset release time	t_{RES}	—	—	13	ms

Note: * Cycle time at writing is a time from the stop condition to write completion (inter control).

Voltage					
Reset detection voltage 1*[1]	Vreset1	LVDSEL = 0	—	2.3	2.7
Reset detection voltage 2*[2]	Vreset2	LVDSEL = 1	3.0	3.6	4.2
Lower-limit voltage of LVDR operation*[3]	$V_{LVDRmin}$		1.0	—	—
LVD stabilization time	t_{LVDON}		50	—	—
Current consumption in standby mode	I_{STBY}	LVDE = 1, Vcc = 5.0 V, When a 32-kHz crystal resonator is not used	—	—	350

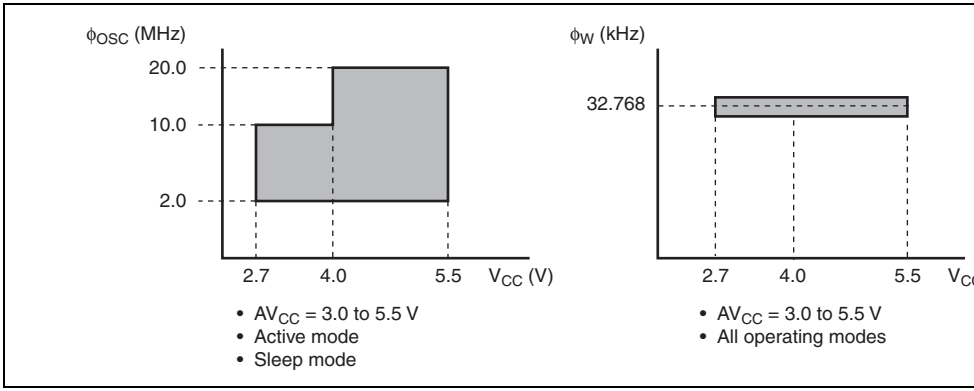
Notes: 1. This voltage should be used when the falling and rising voltage detection func used.
2. Select the low-voltage reset 2 when only the low-voltage detection reset is us
3. When the power-supply voltage (Vcc) falls below $V_{LVDRmin}$ = 1.0 V and then rise may not occur. Therefore sufficient evaluation is required.

Note: The power supply voltage (Vcc) must fall below Vpor = 100 mV and then rise a charge of the $\overline{RES}$ pin is removed completely. In order to remove charge of the pin, it is recommended that the diode be placed in the Vcc side. If the power-su voltage (Vcc) rises from the point over 100 mV, a power-on reset may not occu

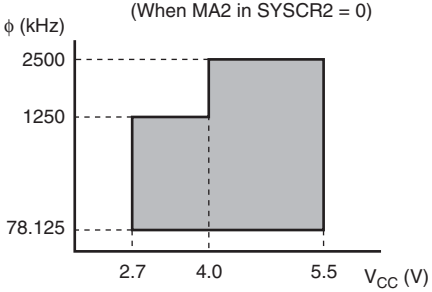
21.3 Electrical Characteristics (Mask-ROM Version, EEPROM Sta Mask-ROM Version)

21.3.1 Power Supply Voltage and Operating Ranges

Power Supply Voltage and Oscillation Frequency Range



- AV_{CC} = 3.0 to 5.5 V
- Active mode
- Sleep mode
  (When MA2 in SYSCR2 = 0)

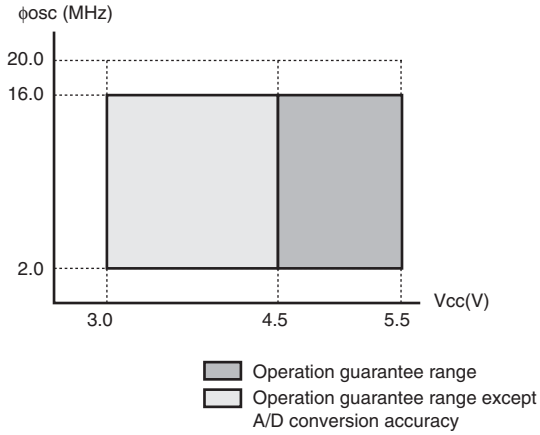


- AV_{CC} = 3.0 to 5.5 V
- Active mode
- Sleep mode
  (When MA2 in SYSCR2 = 1)

- AV_{CC} = 3.0 to 5.5 V
- Subactive mode
- Subsleep mode

- V_{CC} = 2.7 to 5.5 V
- Active mode
- Sleep mode

Range of Power Supply Voltage and Oscillation Frequency when Low-Voltage Dete Circuit is Used



		TMCIV, FTCI, FTIOA to FTIOD, SCK3, TRGV		$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	
		RXD, SCL, SDA, P10 to P12, P14 to P17, P20 to P22, P50 to P57, P74 to P76, P80 to P87	V_{CC} = 4.0 to 5.5 V	$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V
				$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	
		PB0 to PB7	V_{CC} = 4.0 to 5.5 V	$V_{CC} \times 0.7$	—	$AV_{CC} + 0.3$	V
				$V_{CC} \times 0.8$	—	$AV_{CC} + 0.3$	
		OSC1	V_{CC} = 4.0 to 5.5 V	$V_{CC} - 0.5$	—	$V_{CC} + 0.3$	V
				$V_{CC} - 0.3$	—	$V_{CC} + 0.3$	
Input low voltage	V_{IL}	$\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$,TMRIV, TMCIV, FTCI, FTIOA to FTIOD, SCK3, TRGV	V_{CC} = 4.0 to 5.5 V	–0.3	—	$V_{CC} \times 0.2$	V
				–0.3	—	$V_{CC} \times 0.1$	
		RXD, SCL, SDA, P10 to P12, P14 to P17, P20 to P22, P50 to P57, P74 to P76, P80 to P87, PB0 to PB7	V_{CC} = 4.0 to 5.5 V	–0.3	—	$V_{CC} \times 0.3$	V
				–0.3	—	$V_{CC} \times 0.2$	
		OSC1	V_{CC} = 4.0 to 5.5 V	–0.3	—	0.5	V
				–0.3	—	0.3	

			$-I_{OH}$ = 0.1 mA				
Output low voltage	V_{OL}	P10 to P12, P14 to P17, P20 to P22,	V_{CC} = 4.0 to 5.5 V I_{OL} = 1.6 mA	—	—	0.6	V
		P50 to P57, P74 to P76	I_{OL} = 0.4 mA	—	—	0.4	
		P80 to P87	V_{CC} = 4.0 to 5.5 V I_{OL} = 20.0 mA	—	—	1.5	V
			V_{CC} = 4.0 to 5.5 V I_{OL} = 10.0 mA	—	—	1.0	
			V_{CC} = 4.0 to 5.5 V I_{OL} = 1.6 mA	—	—	0.4	
			I_{OL} = 0.4 mA	—	—	0.4	
		SCL, SDA	V_{CC} = 4.0 to 5.5 V I_{OL} = 6.0 mA	—	—	0.6	V
			I_{OL} = 3.0 mA	—	—	0.4	
Input/ output leakage current	$\lvert I_{IL} \rvert$	OSC1, $\overline{\text{NMI}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{ADTRG}}$, TRGV, TMRIV, TMCIV, FTCI, FTIOA to FTIOD, RXD, SCK3, SCL, SDA	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	—	—	1.0	µA
		P10 to P12, P14 to P17, P20 to P22, P50 to P57, P74 to P76, P80 to P87	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	—	—	1.0	µA
		PB0 to PB7	V_{IN} = 0.5 V to (AV_{CC} – 0.5 V)	—	—	1.0	µA

mode current consump-tion			V_{CC} = 5.0 V, f_{OSC} = 20 MHz					
			Active mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	8.0	—		* Re va
	I_{OPE2}	V_{CC}	Active mode 2 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	2.0	3.0	mA	*
			Active mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	1.2	—		* Re va
Sleep mode current consump-tion	I_{SLEEP1}	V_{CC}	Sleep mode 1 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	10.0	17.5	mA	*
			Sleep mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	5.5	—		* Re va
	I_{SLEEP2}	V_{CC}	Sleep mode 2 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	1.6	2.4	mA	*
			Sleep mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	0.8	—		* Re va
Subactive mode current consump-tion	I_{SUB}	V_{CC}	V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	40.0	70.0	µA	*
			V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/8$)	—	30.0	—		* Re va

RAM data retaining voltage	V_{RAM}	V_{CC}	2.0	—	—	V

Note: * Pin states during current consumption measurement are given below (excludin in the pull-up MOS transistors and output buffers).

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pin
Active mode 1	V_{CC}	Operates	V_{CC}	Main clock: ceramic or crys resonator
Active mode 2		Operates (ϕOSC/64)		Subclock: Pin X1 = V_{SS}
Sleep mode 1	V_{CC}	Only timers operate	V_{CC}	
Sleep mode 2		Only timers operate (ϕOSC/64)		
Subactive mode	V_{CC}	Operates	V_{CC}	Main clock: ceramic or crys resonator
Subsleep mode	V_{CC}	Only timers operate	V_{CC}	Subclock: crystal resonat
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	Main clock: ceramic or crys resonator
				Subclock: Pin X1 = V_{SS}

Note: * The current consumption of the EEPROM chip is shown.
For the current consumption of H8/3694N, add the above current values to th
consumption of H8/3694.

		Output pins except port 8, SCL, and SDA		—	—	0.5
Allowable output low current (total)	ΣI_{OL}	Output pins except port 8, SCL, and SDA	V_{CC} = 4.0 to 5.5 V	—	—	40.0
		Port 8, SCL, and SDA		—	—	80.0
		Output pins except port 8, SCL, and SDA		—	—	20.0
		Port 8, SCL, and SDA		—	—	40.0
Allowable output high current (per pin)	$\lvert -I_{OH} \rvert$	All output pins	V_{CC} = 4.0 to 5.5 V	—	—	2.0
				—	—	0.2
Allowable output high current (total)	$\lvert -\Sigma I_{OH} \rvert$	All output pins	V_{CC} = 4.0 to 5.5 V	—	—	30.0
				—	—	8.0

cycle time	t_{cyc}			—	—	12.8	µs	
Subclock oscillation frequency	f_W	X1, X2		—	32.768	—	kHz	
Watch clock (ϕ_W) cycle time	t_W	X1, X2		—	30.5	—	µs	
Subclock (ϕ_{SUB}) cycle time	t_{subcyc}			2	—	8	t_W	
Instruction cycle time				2	—	—	t_{cyc} t_{subcyc}	
Oscillation stabilization time (crystal resonator)	t_{rc}	OSC1, OSC2		—	—	10.0	ms	
Oscillation stabilization time (ceramic resonator)	t_{rc}	OSC1, OSC2		—	—	5.0	ms	
Oscillation stabilization time	t_{rcx}	X1, X2		—	—	2.0	s	
External clock high width	t_{CPH}	OSC1	V_{CC} = 4.0 to 5.5 V	20.0	—	—	ns	
				40.0	—	—		
External clock low width	t_{CPL}	OSC1	V_{CC} = 4.0 to 5.5 V	20.0	—	—	ns	
				40.0	—	—		
External clock rise time	t_{CPr}	OSC1	V_{CC} = 4.0 to 5.5 V	—	—	10.0	ns	
				—	—	15.0		
External clock fall time	t_{CPf}	OSC1	V_{CC} = 4.0 to 5.5 V	—	—	10.0	ns	
				—	—	15.0		
$\overline{RES}$ pin low width	t_{REL}	$\overline{RES}$	At power-on and in modes other than those below	t_{rc}	—	—	ms	
			In active mode and sleep mode operation	200	—	—	ns	

		FTIOD				
Input pin low width	t_{IL}	$\overline{NMI}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{WKP0}$ to $\overline{WKP5}$, TMCIV, TMRIV, TRGV, $\overline{ADTRG}$, FTCI, FTIOA to FTIOD	2	—	—	t_{cyc} t_{subcyc}

Notes: 1. When an external clock is input, the minimum system clock oscillation frequen
1.0 MHz.

2. Determined by MA2, MA1, MA0, SA1, and SA0 of system control register 2 (S

SCL and SDA input spike pulse removal time	t_{SP}		—	—	$1t_{cyc}$	ns
SDA input bus-free time	t_{BUF}		$5t_{cyc}$	—	—	ns
Start condition input hold time	t_{STAH}		$3t_{cyc}$	—	—	ns
Retransmission start condition input setup time	t_{STAS}		$3t_{cyc}$	—	—	ns
Setup time for stop condition input	t_{STOS}		$3t_{cyc}$	—	—	ns
Data-input setup time	t_{SDAS}		$1t_{cyc}$+20	—	—	ns
Data-input hold time	t_{SDAH}		0	—	—	ns
Capacitive load of SCL and SDA	c_b		0	—	400	pF
SCL and SDA output fall time	t_{Sf}	V_{CC} = 4.0 to 5.5 V	—	—	250	ns
			—	—	300	

Transmit data delay time (clocked synchronous)	t_{TXD}	TXD	V_{CC} = 4.0 to 5.5 V	—	—	1	t_{cyc}	Fig
				—	—	1		
Receive data setup time (clocked synchronous)	t_{RXS}	RXD	V_{CC} = 4.0 to 5.5 V	50.0	—	—	ns	
				100.0	—	—		
Receive data hold time (clocked synchronous)	t_{RXH}	RXD	V_{CC} = 4.0 to 5.5 V	50.0	—	—	ns	
				100.0	—	—		

Analog power supply current	AI_{OPE}	AV_{CC}	AV_{CC} = 5.0 V f_{OSC} = 20 MHz	—	—	2.0	mA
	AI_{STOP1}	AV_{CC}		—	50	—	µA
	AI_{STOP2}	AV_{CC}		—	—	5.0	µA
Analog input capacitance	C_{AIN}	AN0 to AN7		—	—	30.0	pF
Allowable signal source impedance	R_{AIN}	AN0 to AN7		—	—	5.0	kΩ
Resolution (data length)				10	10	10	bit
Conversion time (single mode)			AV_{CC} = 3.0 to 5.5 V	134	—	—	t_{cyc}
Nonlinearity error				—	—	±7.5	LSB
Offset error				—	—	±7.5	LSB
Full-scale error				—	—	±7.5	LSB
Quantization error				—	—	±0.5	LSB
Absolute accuracy				—	—	±8.0	LSB
Conversion time (single mode)			AV_{CC} = 4.0 to 5.5 V	70	—	—	t_{cyc}
Nonlinearity error				—	—	±7.5	LSB
Offset error				—	—	±7.5	LSB
Full-scale error				—	—	±7.5	LSB
Quantization error				—	—	±0.5	LSB
Absolute accuracy				—	—	±8.0	LSB

2. AI_{STOP1} is the current in active and sleep modes while the A/D converter is idle.
3. AI_{STOP2} is the current at reset and in standby, subactive, and subsleep modes w A/D converter is idle.

21.3.5 Watchdog Timer Characteristics

Table 21.17 Watchdog Timer Characteristics

V_{CC} = 2.7 to 5.5 V, V_{SS} = 0.0 V, T_a = –20 to +75°C, unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	R Fi
On-chip oscillator overflow time	t_{OVF}			0.2	0.4	—	s	*

Note: * Shows the time to count from 0 to 255, at which point an internal reset is gene when the internal oscillator is selected.

SCL input low pulse width	t_{SCLL}	1200	—	—	ns
SCL, SDA input spike pulse removal time	t_{SP}	—	—	50	ns
SDA input bus-free time	t_{BUF}	1200	—	—	ns
Start condition input hold time	t_{STAH}	600	—	—	ns
Retransmit start condition input setup time	t_{STAS}	600	—	—	ns
Stop condition input setup time	t_{STOS}	600	—	—	ns
Data input setup time	t_{SDAS}	160	—	—	ns
Data input hold time	t_{SDAH}	0	—	—	ns
SCL, SDA input fall time	t_{Sf}	—	—	300	ns
SDA input rise time	t_{Sr}	—	—	300	ns
Data output hold time	t_{DH}	50	—	—	ns
SCL, SDA capacitive load	C_b	0	—	400	pF
Access time	t_{AA}	100	—	900	ns
Cycle time at writing*	t_{WC}	—	—	10	ms
Reset release time	t_{RES}	—	—	13	ms

Note: * Cycle time at writing is a time from the stop condition to write completion (inte control).

voltage					
Reset detection voltage 1*[1]	Vreset1	LVDSEL = 0	—	2.3	2.7
Reset detection voltage 2*[2]	Vreset2	LVDSEL = 1	3.0	3.6	4.2
Lower-limit voltage of LVDR operation*[3]	$V_{LVDRmin}$		1.0	—	—
LVD stabilization time	t_{LVDON}		50	—	—
Current consumption in standby mode	I_{STBY}	LVDE = 1, Vcc = 5.0 V, When a 32-kHz crystal resonator is not used	—	—	350

Notes: 1. This voltage should be used when the falling and rising voltage detection funct used.
2. Select the low-voltage reset 2 when only the low-voltage detection reset is use
3. When the power-supply voltage (Vcc) falls below $V_{LVDRmin}$ = 1.0 V and then rises may not occur. Therefore sufficient evaluation is required.

Note: The power supply voltage (Vcc) must fall below Vpor = 100 mV and then rise
charge of the $\overline{RES}$ pin is removed completely. In order to remove charge of th
pin, it is recommended that the diode be placed in the Vcc side. If the power-
voltage (Vcc) rises from the point over 100 mV, a power-on reset may not occ

21.4 Operation Timing

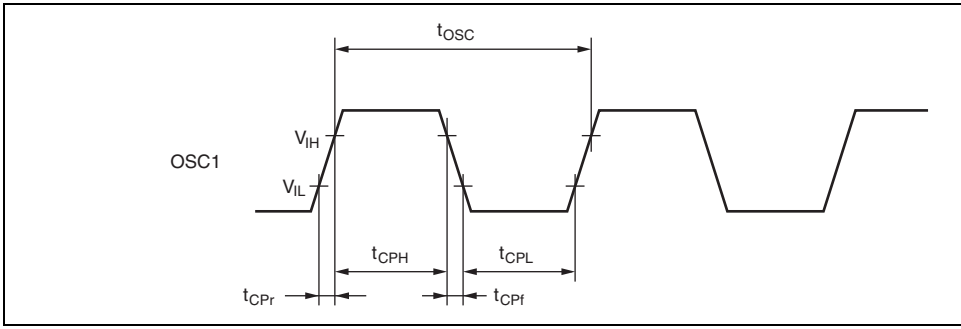


Figure 21.1 System Clock Input Timing

Figure 21.2 $\overline{\text{RES}}$ Low Width Timing

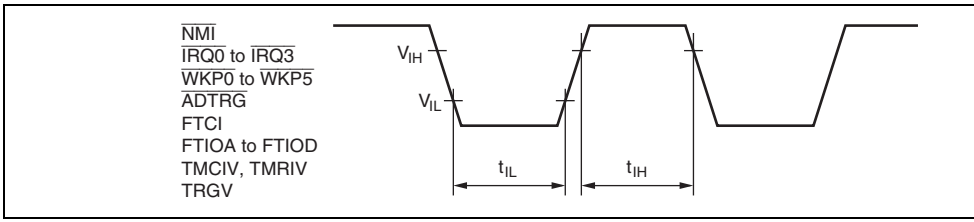
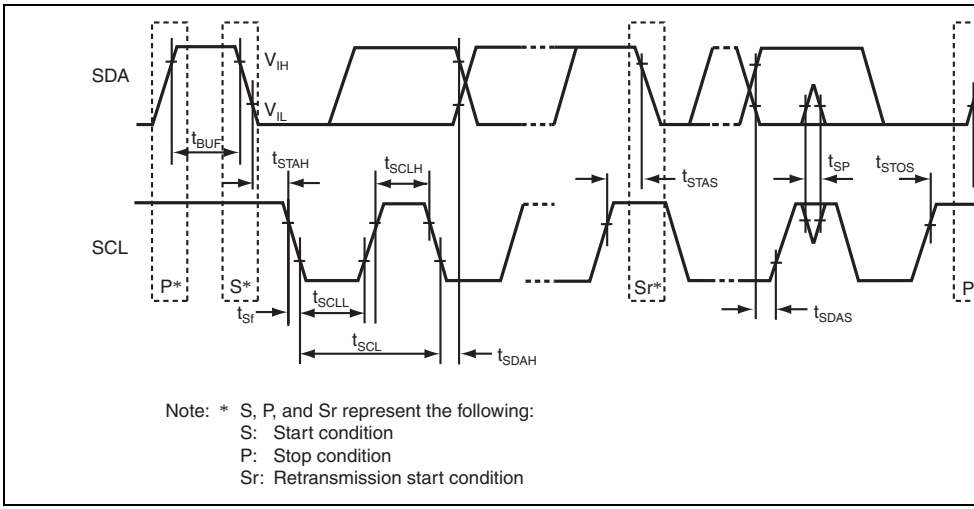
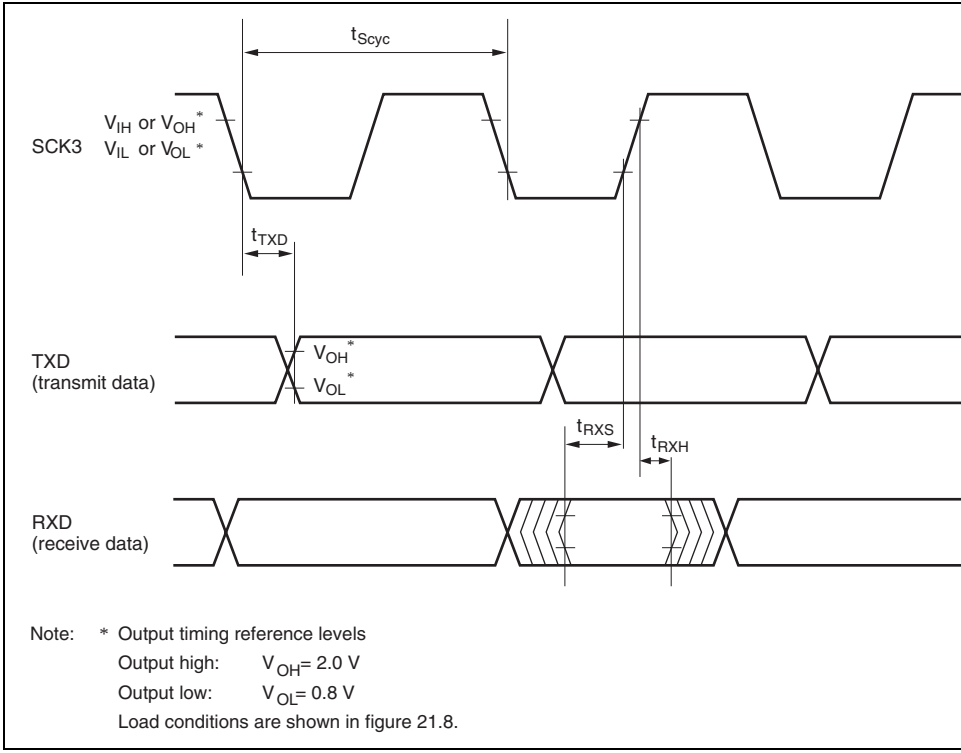


Figure 21.3 Input Timing



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

Figure 21.4 I²C Bus Interface Input/Output Timing



Note: * Output timing reference levels
Output high: V_{OH}= 2.0 V
Output low: V_{OL}= 0.8 V
Load conditions are shown in figure 21.8.

Figure 21.6 SCI Input/Output Timing in Clocked Synchronous Mode

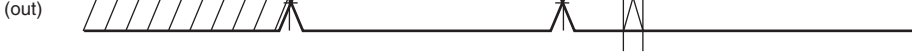


Figure 21.7 EEPROM Bus Timing

21.5 Output Load Condition

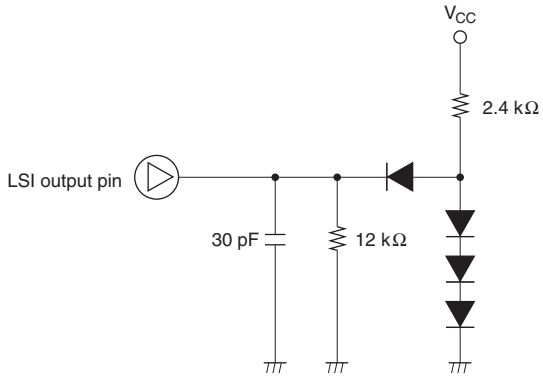


Figure 21.8 Output Load Circuit

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

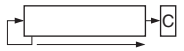
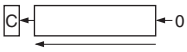
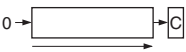
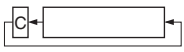
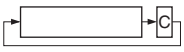
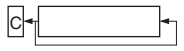
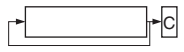
	MOV.L @ERs, ERd	L			4						@ERs → ERd32	—	—	↕	↕	0	-
	MOV.L @(d:16, ERs), ERd	L				6					@(d:16, ERs) → ERd32	—	—	↕	↕	0	-
	MOV.L @(d:24, ERs), ERd	L				10					@(d:24, ERs) → ERd32	—	—	↕	↕	0	-
	MOV.L @ERs+, ERd	L					4				@ERs → ERd32 ERs32+4 → ERs32	—	—	↕	↕	0	-
	MOV.L @aa:16, ERd	L						6			@aa:16 → ERd32	—	—	↕	↕	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	—	—	↕	↕	↕

	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)	↕	↕	↕	

											of ERd32)						
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					—	—	—	—	—				

													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+	—	—	—	—	—

RTE	RTE	—										CCR ← @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 i

Instruction when most significant bit of BH i

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												

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			SHAL
11	SHLR			SHLR					SHAR			SHAR
12	ROTXL			ROTXL					ROTL			ROTL
13	ROTXR			ROTXR					ROTR			ROTR
17	NOT			NOT		EXTU		EXTU	NEG			NEG
1A	DEC											
1B	SUBS					DEC		DEC	SUB			
1F	DAS											
58	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI
79	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 n
Instruction when n

CL / AH / ALBH / BLCH	0	1	2	3	4	5	6	7	8	9	A	B
01406										LDC / STC		LDC / ST
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).

ADDS	ADDS #1/2/4, ERd	1		
ADDX	ADDX #xx:8, Rd	1		
	ADDX Rs, Rd	1		
AND	AND.B #xx:8, Rd	1		
	AND.B Rs, Rd	1		
	AND.W #xx:16, Rd	2		
	AND.W Rs, Rd	1		
	AND.L #xx:32, ERd	3		
	AND.L ERs, ERd	2		
ANDC	ANDC #xx:8, CCR	1		
BAND	BAND #xx:3, Rd	1		
	BAND #xx:3, @ERd	2		1
	BAND #xx:3, @aa:8	2		1
Bcc	BRA d:8 (BT d:8)	2		
	BRN d:8 (BF d:8)	2		
	BHI d:8	2		
	BLS d:8	2		
	BCC d:8 (BHS d:8)	2		
	BCS d:8 (BLO d:8)	2		
	BNE d:8	2		
	BEQ d:8	2		
	BVC d:8	2		
	BVS d:8	2		
	BPL d:8	2		
	BMI d:8	2		
	BGE d:8	2		

	BCC d:16(BHS d:16)	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, @ERd	2		1
	BIXOR #xx:3, @aa:8	2		1
BLD	BLD #xx:3, Rd	1		
	BLD #xx:3, @ERd	2		1
	BLD #xx:3, @aa:8	2		1
BNOT	BNOT #xx:3, Rd	1		
	BNOT #xx:3, @ERd	2		2
	BNOT #xx:3, @aa:8	2		2
	BNOT Rn, Rd	1		
	BNOT Rn, @ERd	2		2
	BNOT Rn, @aa:8	2		2
BOR	BOR #xx:3, Rd	1		
	BOR #xx:3, @ERd	2		1
	BOR #xx:3, @aa:8	2		1
BSET	BSET #xx:3, Rd	1		
	BSET #xx:3, @ERd	2		2
	BSET #xx:3, @aa:8	2		2
	BSET Rn, Rd	1		
	BSET Rn, @ERd	2		2
	BSET Rn, @aa:8	2		2
BSR	BSR d:8	2	1	
	BSR d:16	2	1	
BST	BST #xx:3, Rd	1		
	BST #xx:3, @ERd	2		2
	BST #xx:3, @aa:8	2		2

	BXOR #xx:3, @ERd	2		1
	BXOR #xx:3, @aa:8	2		1
CMP	CMP.B #xx:8, Rd	1		
	CMP.B Rs, Rd	1		
	CMP.W #xx:16, Rd	2		
	CMP.W Rs, Rd	1		
	CMP.L #xx:32, ERd	3		
	CMP.L ERs, ERd	1		
DAA	DAA Rd	1		
DAS	DAS Rd	1		
DEC	DEC.B Rd	1		
	DEC.W #1/2, Rd	1		
	DEC.L #1/2, ERd	1		
DUVXS	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:24	2			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	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	

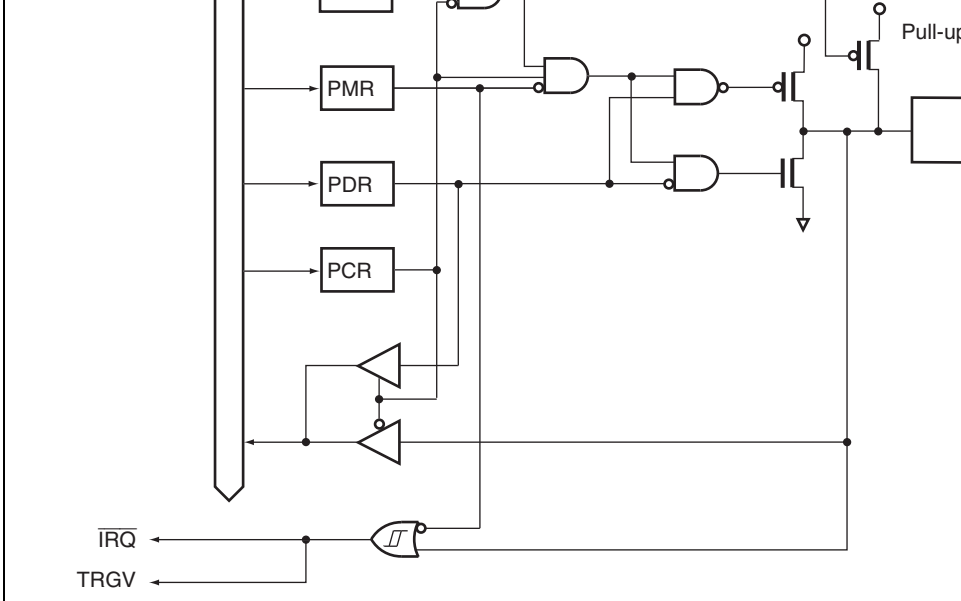
NOP	NOP	1		
NOT	NOT.B Rd	1		
	NOT.W Rd	1		
	NOT.L ERd	1		
OR	OR.B #xx:8, Rd	1		
	OR.B Rs, Rd	1		
	OR.W #xx:16, Rd	2		
	OR.W Rs, Rd	1		
	OR.L #xx:32, ERd	3		
	OR.L ERs, ERd	2		
ORC	ORC #xx:8, CCR	1		
POP	POP.W Rn	1		1
	POP.L ERn	2		2
PUSH	PUSH.W Rn	1		1
	PUSH.L ERn	2		2
ROTL	ROTL.B Rd	1		
	ROTL.W Rd	1		
	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		

	SHAL.L ERd	1		
SHAR	SHAR.B Rd	1		
	SHAR.W Rd	1		
	SHAR.L ERd	1		
SHLL	SHLL.B Rd	1		
	SHLL.W Rd	1		
	SHLL.L ERd	1		
SHLR	SHLR.B Rd	1		
	SHLR.W Rd	1		
	SHLR.L ERd	1		
SLEEP	SLEEP	1		
STC	STC CCR, Rd	1		
	STC CCR, @ERd	2		1
	STC CCR, @(d:16,ERd)	3		1
	STC CCR, @(d:24,ERd)	5		1
	STC CCR,@-ERd	2		1
	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 #xx:32, ERd	3
	XOR.L ERs, ERd	2
XORC	XORC #xx:8, CCR	1

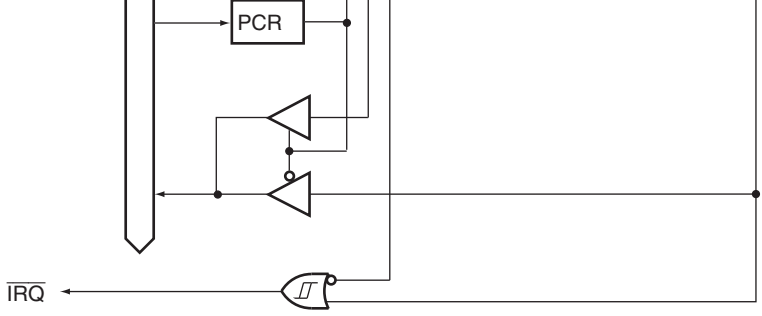
Notes: 1. n: Specified value in R4L. The source and destination operands are accessed
times respectively.
2. It can not be used in this LSI.

	MOVFPE, MOVTPE	—	—	—	—	—	—	—	—	—	—	—	—	—
Arithmetic operations	ADD, CMP	BWL	BWL	—	—	—	—	—	—	—	—	—	—	—
	SUB	WL	BWL	—	—	—	—	—	—	—	—	—	—	—
	ADDX, SUBX	B	B	—	—	—	—	—	—	—	—	—	—	—
	ADDS, SUBS	—	L	—	—	—	—	—	—	—	—	—	—	—
	INC, DEC	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	DAA, DAS	—	B	—	—	—	—	—	—	—	—	—	—	—
	MULXU, MULXS, DIVXU, DIVXS	—	BW	—	—	—	—	—	—	—	—	—	—	—
	NEG	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	EXTU, EXTS	—	WL	—	—	—	—	—	—	—	—	—	—	—
Logical operations	AND, OR, XOR	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	NOT	—	BWL	—	—	—	—	—	—	—	—	—	—	—
Shift operations		—	BWL	—	—	—	—	—	—	—	—	—	—	—
Bit manipulations		—	B	B	—	—	—	B	—	—	—	—	—	—
Branching instructions	BCC, BSR	—	—	—	—	—	—	—	—	—	—	—	—	—
	JMP, JSR	—	—	◯	—	—	—	—	—	—	◯	◯	—	—
	RTS	—	—	—	—	—	—	—	—	◯	—	—	◯	—
System control instructions	TRAPA	—	—	—	—	—	—	—	—	—	—	—	—	◯
	RTE	—	—	—	—	—	—	—	—	—	—	—	—	◯
	SLEEP	—	—	—	—	—	—	—	—	—	—	—	—	◯
	LDC	B	B	W	W	W	W	—	W	W	—	—	—	◯
	STC	—	B	W	W	W	W	—	W	W	—	—	—	—
	ANDC, ORC, XORC	B	—	—	—	—	—	—	—	—	—	—	—	—
	NOP	—	—	—	—	—	—	—	—	—	—	—	—	◯
Block data transfer instructions		—	—	—	—	—	—	—	—	—	—	—	—	BW



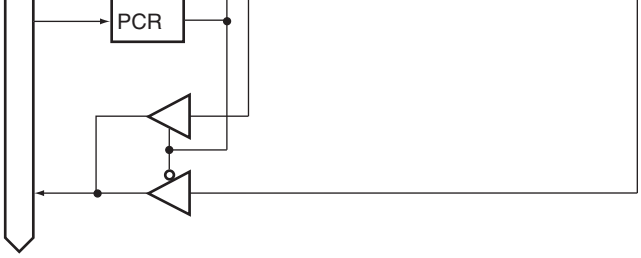
[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.1 Port 1 Block Diagram (P17)



[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.2 Port 1 Block Diagram (P16 to P14)



[Legend]
PUCR: Port pull-up control register
PDR: Port data register
PCR: Port control register

Figure B.3 Port 1 Block Diagram (P12, P11)

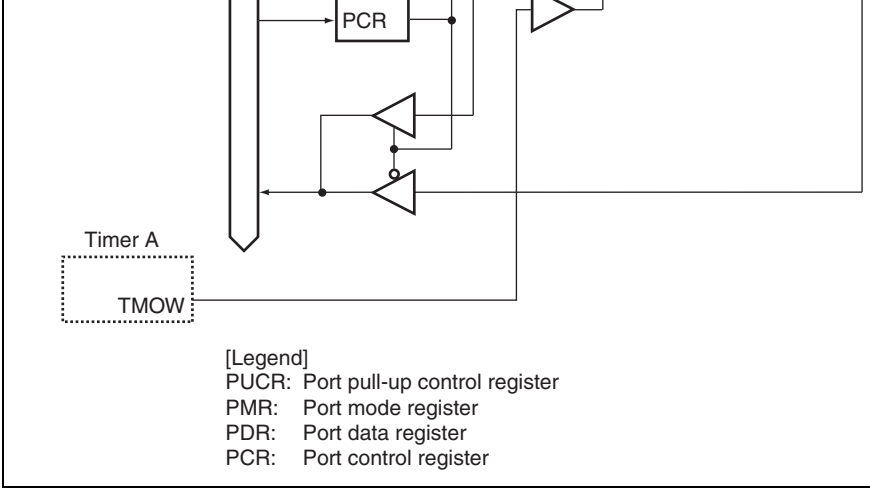
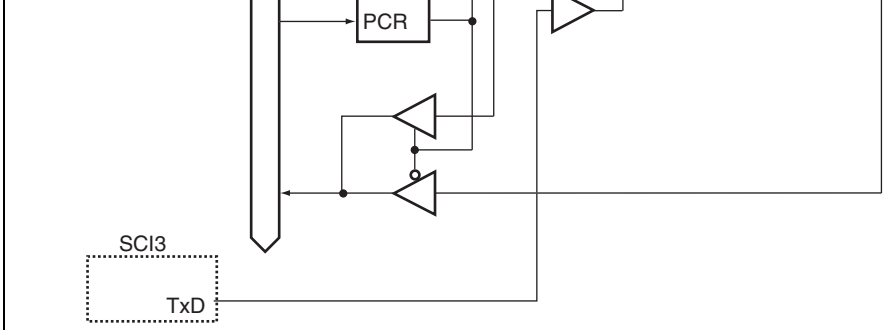


Figure B.4 Port 1 Block Diagram (P10)



[Legend]
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.5 Port 2 Block Diagram (P22)

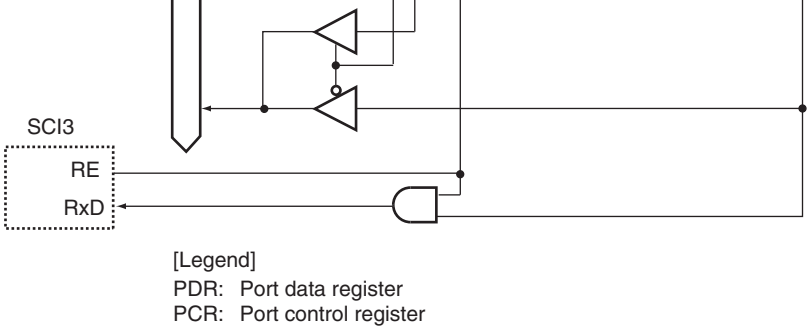
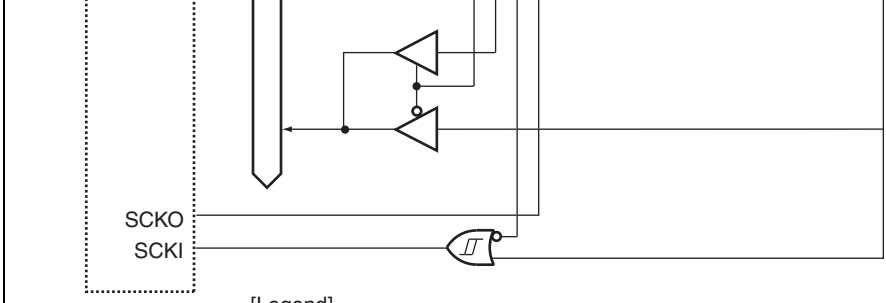


Figure B.6 Port 2 Block Diagram (P21)



[Legend]
PDR: Port data register
PCR: Port control register

Figure B.7 Port 2 Block Diagram (P20)

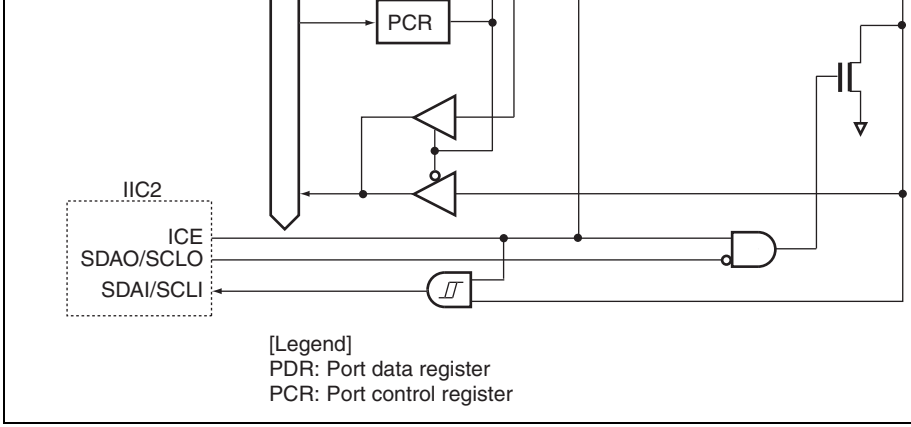
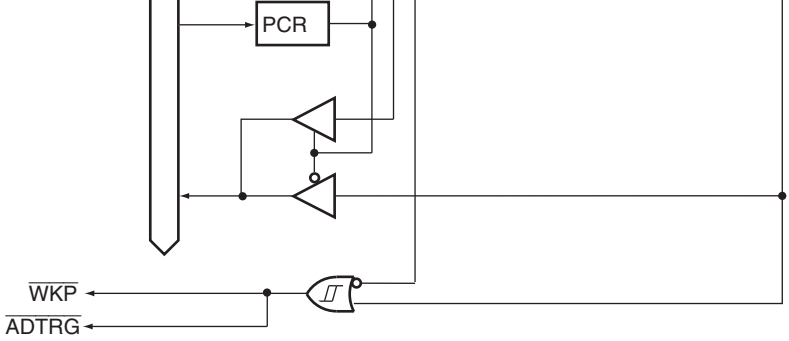


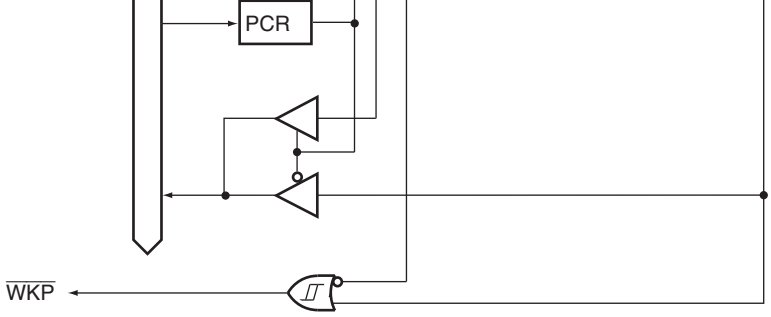
Figure B.8 Port 5 Block Diagram (P57, P56)*

Note: * This diagram is applied to the SCL and SDA pins in the H8/3694N.



[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.9 Port 5 Block Diagram (P55)



[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.10 Port 5 Block Diagram (P54 to P50)

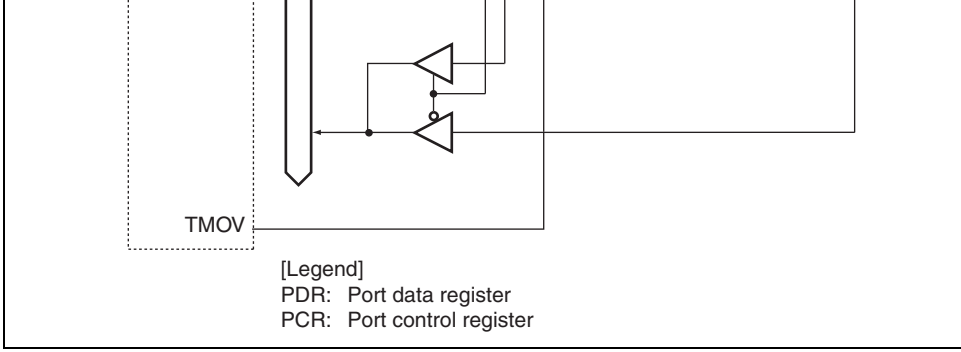


Figure B.11 Port 7 Block Diagram (P76)

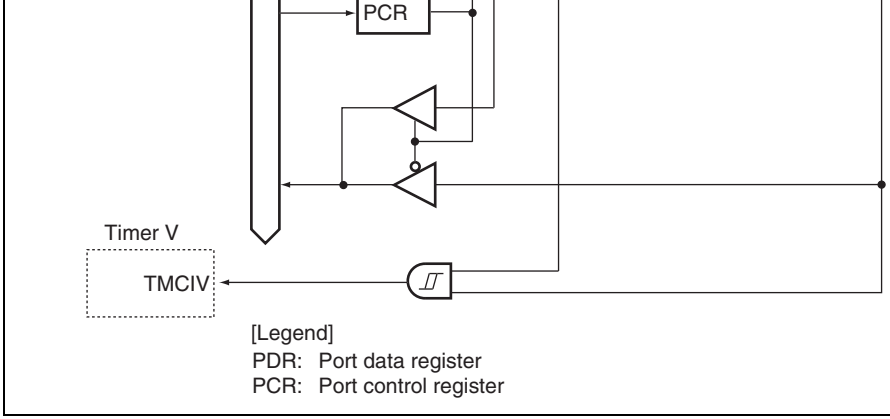
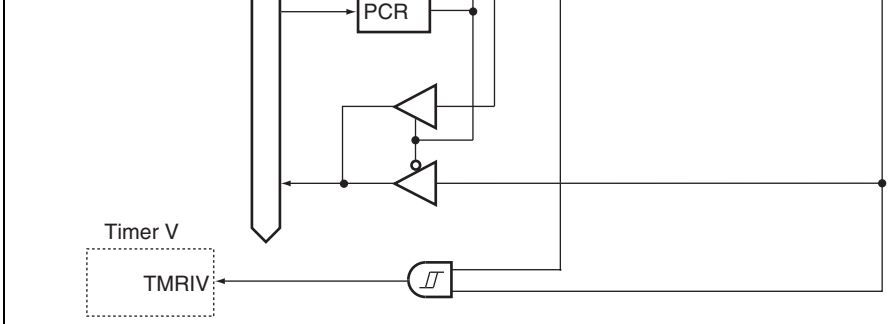
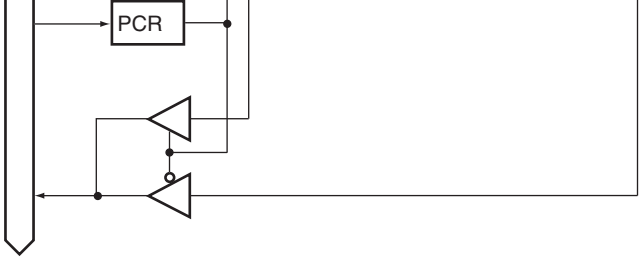


Figure B.12 Port 7 Block Diagram (P75)



[Legend]
PDR: Port data register
PCR: Port control register

Figure B.13 Port 7 Block Diagram (P74)



[Legend]
PDR: Port data register
PCR: Port control register

Figure B.14 Port 8 Block Diagram (P87 to P85)

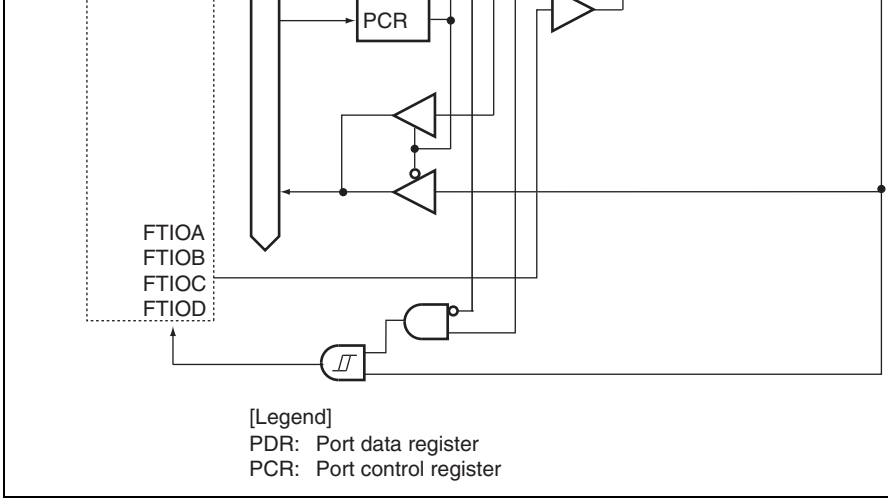


Figure B.15 Port 8 Block Diagram (P84 to P81)

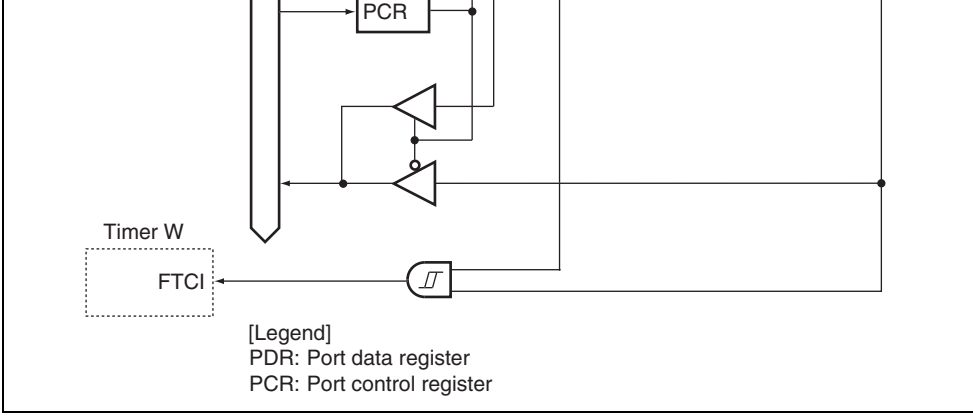


Figure B.16 Port 8 Block Diagram (P80)

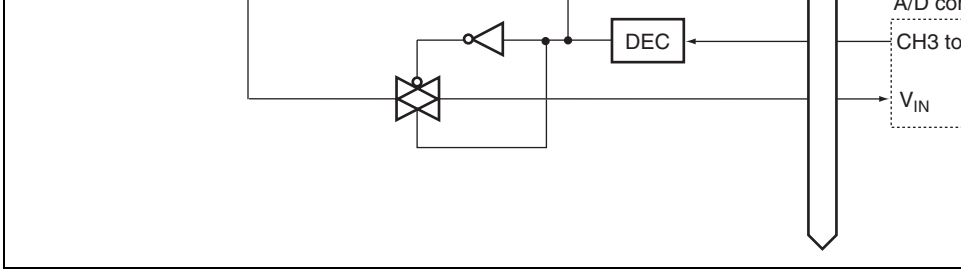


Figure B.17 Port B Block Diagram (PB7 to PB0)

B.2 Port States in Each Operating State

Port	Reset	Sleep	Subsleep	Standby	Subactive	Ac
P17 to P14, P12 to P10	High impedance	Retained	Retained	High impedance*[1]	Functioning	Fu
P22 to P20	High impedance	Retained	Retained	High impedance	Functioning	Fu
P57 to P50*[2]	High impedance	Retained	Retained	High impedance*[1]	Functioning	Fu
P76 to P74	High impedance	Retained	Retained	High impedance	Functioning	Fu
P87 to P80	High impedance	Retained	Retained	High impedance	Functioning	Fu
PB7 to PB0	High impedance	High impedance	High impedance	High impedance	High impedance	Hig im

Notes: 1. High level output when the pull-up MOS is in on state.
2. The P55 to P50 pins are applied to the H8/3694N.

		POR & LVDC	HD64F3694GFP	HD64F3694GFP	LQFP-64 (
			HD64F3694GFX	HD64F3694GFX	LQFP-48 (
			HD64F3694GFY	HD64F3694GFY	LQFP-48 (
			HD64F3694GFT	HD64F3694GFT	QFN-48(TI
	Mask ROM version	Standard product	HD6433694H	HD6433694(***)H	QFP-64 (F
			HD6433694FP	HD6433694(***)FP	LQFP-64 (
			HD6433694FX	HD6433694(***)FX	LQFP-48 (
			HD6433694FY	HD6433694(***)FY	LQFP-48 (
			HD6433694FT	HD6433694(***)FT	QFN-48(TI
		Product with POR & LVDC	HD6433694GH	HD6433694G(***)H	QFP-64 (F
			HD6433694GFP	HD6433694G(***)FP	LQFP-64 (
			HD6433694GFX	HD6433694G(***)FX	LQFP-48 (
			HD6433694GFY	HD6433694G(***)FY	LQFP-48 (
			HD6433694GFT	HD6433694G(***)FT	QFN-48(TI
H8/3693	Mask ROM version	Standard product	HD6433693H	HD6433693(***)H	QFP-64 (F
			HD6433693FP	HD6433693(***)FP	LQFP-64 (
			HD6433693FX	HD6433693(***)FX	LQFP-48 (
			HD6433693FY	HD6433693(***)FY	LQFP-48 (
			HD6433693FT	HD6433693(***)FT	QFN-48(TI
		Product with POR & LVDC	HD6433693GH	HD6433693G(***)H	QFP-64 (F
			HD6433693GFP	HD6433693G(***)FP	LQFP-64 (
			HD6433693GFX	HD6433693G(***)FX	LQFP-48 (
			HD6433693GFY	HD6433693G(***)FY	LQFP-48 (
			HD6433693GFT	HD6433693G(***)FT	QFN-48(TI

			HD6433692GFY	HD6433692G(***)FY	LQFP-4
			HD6433692GFT	HD6433692G(***)FT	QFN-48
H8/3691	Mask ROM version	Standard product	HD6433691H	HD6433691(***)H	QFP-64
			HD6433691FP	HD6433691(***)FP	LQFP-6
			HD6433691FX	HD6433691(***)FX	LQFP-4
			HD6433691FY	HD6433691(***)FY	LQFP-4
			HD6433691FT	HD6433691(***)FT	QFN-48
		Product with POR & LVDC	HD6433691GH	HD6433691G(***)H	QFP-64
			HD6433691GFP	HD6433691G(***)FP	LQFP-6
			HD6433691GFX	HD6433691G(***)FX	LQFP-4
			HD6433691GFY	HD6433691G(***)FY	LQFP-4
			HD6433691GFT	HD6433691G(***)FT	QFN-48
H8/3690	Mask ROM version	Standard product	HD6433690H	HD6433690(***)H	QFP-64
			HD6433690FP	HD6433690(***)FP	LQFP-6
			HD6433690FX	HD6433690(***)FX	LQFP-4
			HD6433690FY	HD6433690(***)FY	LQFP-4
			HD6433690FT	HD6433690(***)FT	QFN-48
		Product with POR & LVDC	HD6433690GH	HD6433690G(***)H	QFP-64
			HD6433690GFP	HD6433690G(***)FP	LQFP-6
			HD6433690GFX	HD6433690G(***)FX	LQFP-4
			HD6433690GFY	HD6433690G(***)FY	LQFP-4
			HD6433690GFT	HD6433690G(***)FT	QFN-48

Figure D.1 FP-64E Package Dimensions

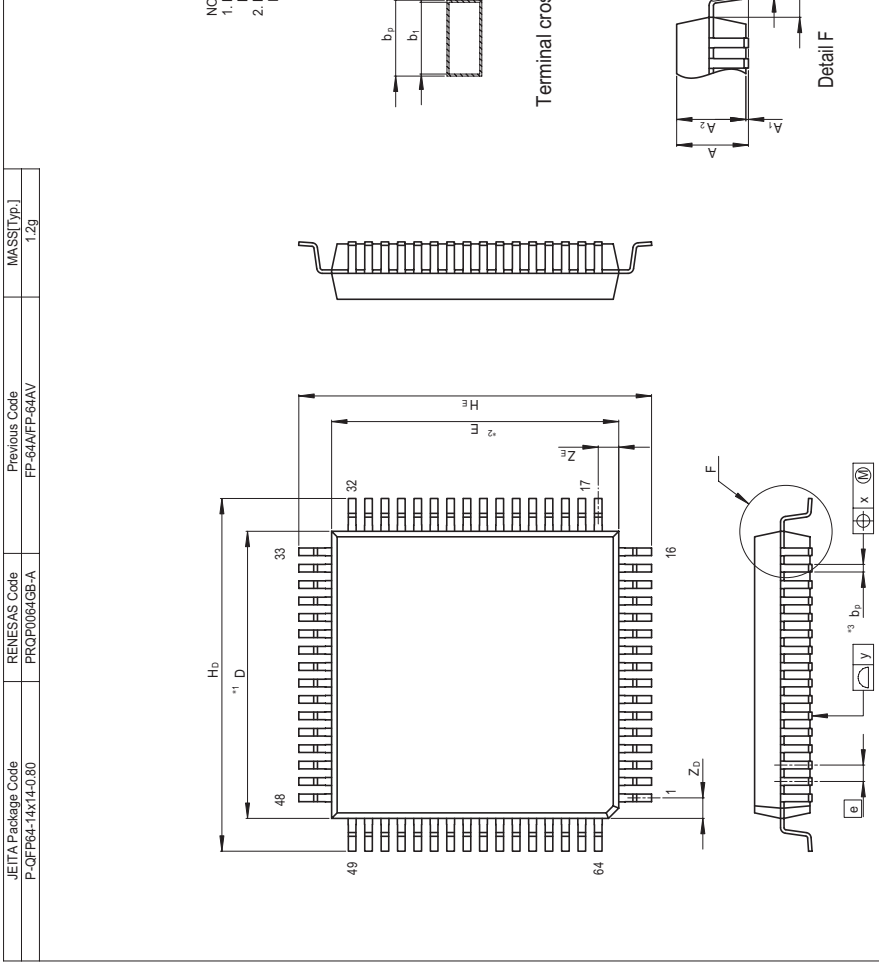


Figure D.2 FP-64A Package Dimensions

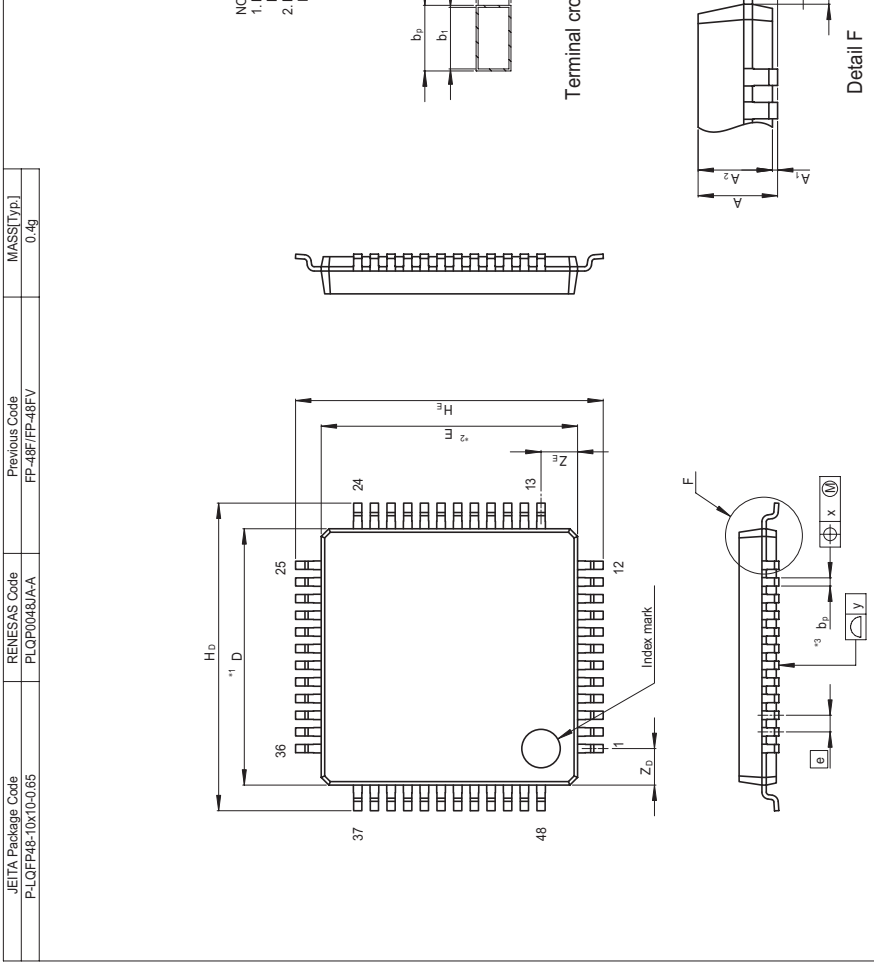


Figure D.3 FP-48F Package Dimensions

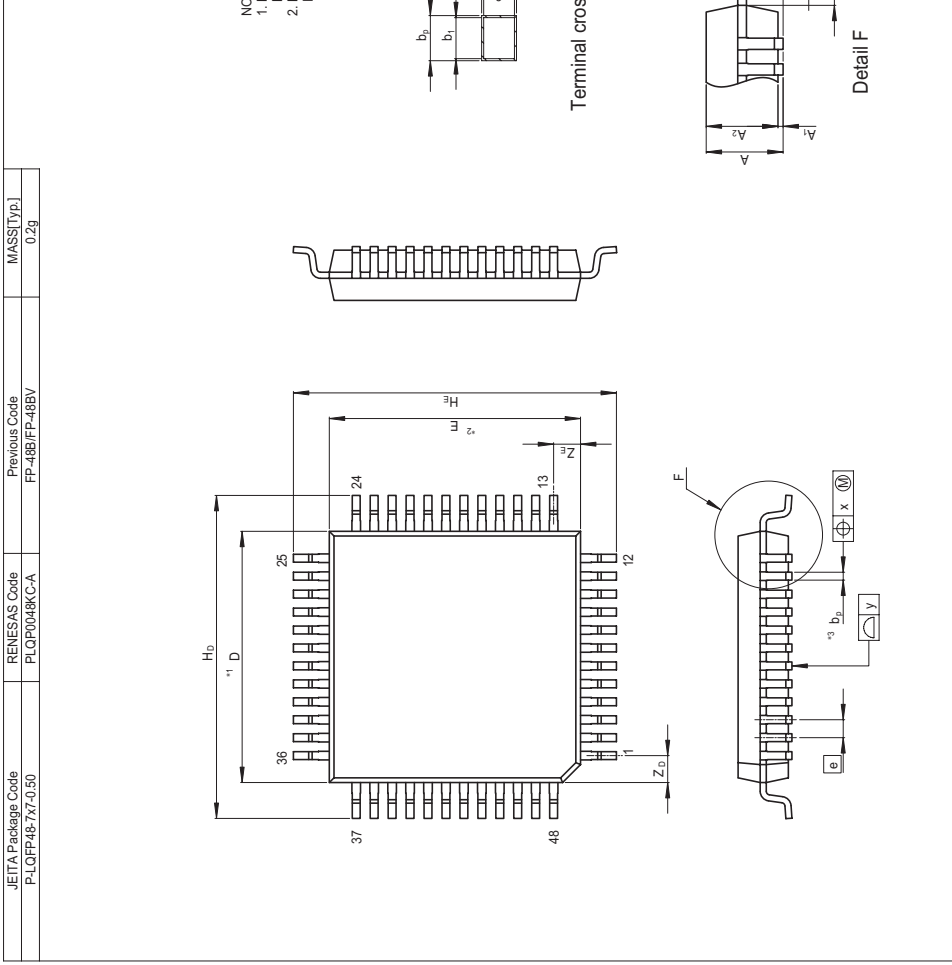


Figure D.4 FP-48B Package Dimensions

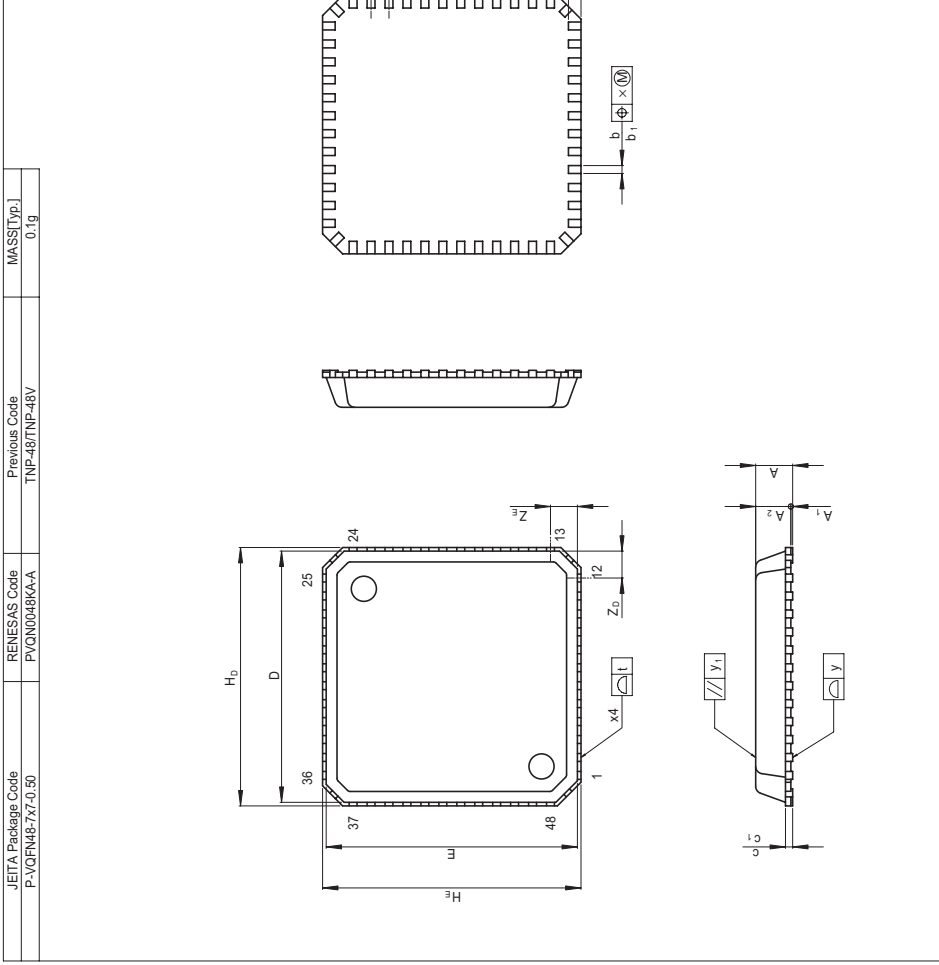


Figure D.5 TNP-48 Package Dimensions

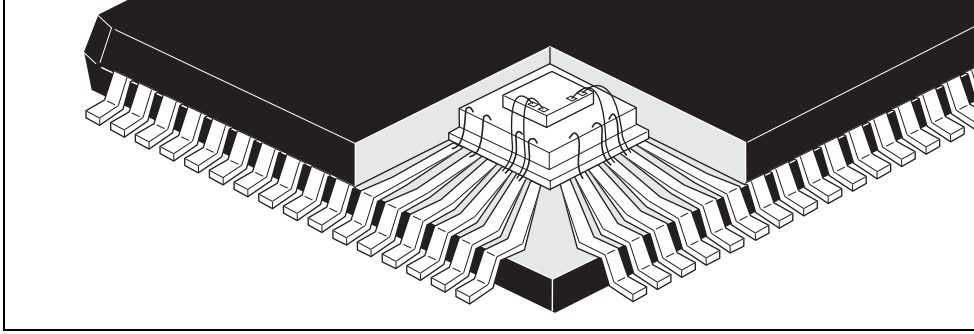
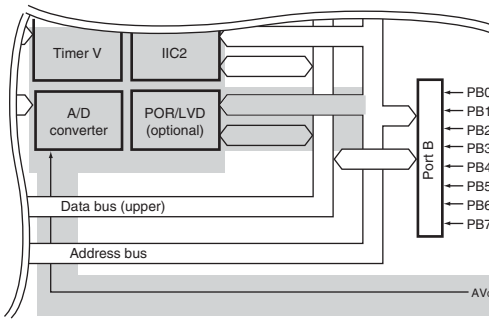


Figure E.1 EEPROM Stacked-Structure Cross-Sectional View

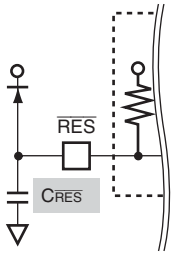
available to the user.

5. When the E7 or E8 is used, address breaks can b
either available to the user or for use by the E7 or
address breaks are set as being used by the E7 c
address break control registers must not be acces
6. When the E7 or E8 is used, $\overline{\text{NMI}}$ is an input/outpu
(open-drain in output mode), P85 and P87 are inp
and P86 is an output pin.

Note has been deleted.

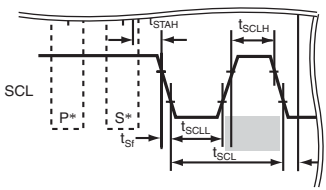
Section 1 Overview Figure 1.1 Internal Block Diagram of H8/3694 Group of F-ZTAT™ and Mask-ROM Versions, Figure 1.2 Internal Block Diagram of H8/3694N (EEPROM Stacked Version)	4, 5	

Section 6 Power-Down Modes 6.1.1 System Control Register 1 (SYSCR1)	76	**Bit**	**Bit Name**	**Description**
		3	NESEL	Noise Elimination Sampling Frequen Select The subclock pulse generator genera watch clock signal (ϕ_w) and the syste pulse generator generates the oscilla clock (ϕ_{osc}). This bit selects the samp frequency of the oscillator clock whe watch clock signal (ϕ_w) is sampled. W ϕ_{osc} = 4 to 20 MHz, clear NESEL to 0
Section 8 RAM	107	Note: * When the E7 or E8 is used, area H'F780 to H'F must not be accessed.		
Section 13 Watchdog Timer 13.2.1 Timer Control/Status Register WD (TCSRWD)	184	**Bit**	**Bit Name**	**Description**
		4	TCSRWE	Timer Control/Status Register WD Write

		the address set in SAR
15.7 Usage Notes	264	Added
Section 16 A/D Converter 16.3.1 A/D Data Registers A to D (ADDRA to ADDRD)	268	Therefore byte access to ADDR should be done by re upper byte first then the lower one. Word access is al possible. ADDR is initialized to H'0000.
Section 18 Power-On Reset and Low-Voltage Detection Circuits (Optional) Figure 18.1 Block Diagram of Power-On Reset Circuit and Low-Voltage Detection Circuit	290	
Section 21 Electrical Characteristics Table 21.2 DC Characteristics (1)	318	

Item	Symbol	Applicable Pins	Test Condition
Input high voltage	V_{IH}	PB0 to PB7	V_{CC} = 4.0 to 5.5 V
Input low voltage	V_{IL}	RXD,SCL, SDA, P10 to P12, : P80 to P87 PB0 to PB7	V_{CC} = 4.0 to 5.5 V

Item	Symbol	Pins	Test Condition	M
Input high voltage	V_{IH}	PB0 to PB7	V_{CC} = 4.0 to 5.5 V	V
				V
Input low voltage	V_{IL}	RXD, SCL, SDA	V_{CC} = 4.0 to 5.5 V	–
		P10 to P12, : P80 to P87		–
		PB0 to PB7		

	340	**Mode**	**$\overline{RES}$ Pin**	**Internal State**
		Active mode 1	V_{CC}	Operates
		Active mode 2		Operates (ϕOSC/64)
		Sleep mode 1	V_{CC}	Only timers oper
		Sleep mode 2		Only timers oper (ϕOSC/64)
Figure 21.4 I^2C Bus Interface Input/Output Timing	352			
Appendix D Package Dimensions	405 to 409	Swapped with new ones.		

C

E

F

G

I

L

R

S

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8/3694 Group

Publication Date: 1st Edition, Jul, 2001
Rev.5.00, Nov. 02, 2005
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, United Kingdom
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit2607 Ruijing Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

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

H8/3694Group
Hardware Manual

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

REJ09B0028-0500

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