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.

RENESAS

User's Manual

16

H8/3687Group

Hardware Manual

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

H8/3687N	HD64N3687G, HD6483687G,
H8/3687F	HD64F3687, HD64F3687G,
H8/3687	HD6433687, HD6433687G,
H8/3686	HD6433686, HD6433686G,
H8/3685	HD6433685, HD6433685G,
H8/3684F	HD64F3684, HD64F3684G,
H8/3684	HD6433684, HD6433684G,
H8/3683	HD6433683, HD6433683G,
H8/3682	HD6433682, HD6433682G

Renesas Electronics
www.renesas.com

Rev.5.00 2005.11

under any intellectual property rights, or any other rights, belonging to Renesas Technology C
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, algorith circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, progra algorithms represents information on products at the time of publication of these materials, a subject to change by Renesas Technology Corp. without notice due to product improvement other reasons. It is therefore recommended that customers contact Renesas Technology Co an authorized Renesas Technology Corp. product distributor for the latest product informatio 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.co
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 t system before making a final decision on the applicability of the information and products. R Technology Corp. assumes no responsibility for any damage, liability or other loss resulting f information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a d system that is used under circumstances in which human life is potentially at stake. Please Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor considering the use of a product contained herein for any specific purposes, such as appara 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 c 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 produ 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 imn
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 H8/3687 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:	Register name:	The following notation is used for cases when the sa similar function, e.g. serial communication interface implemented on more than one channel: XXX_N (XXX is the register name and N is the cha number)
	Bit order:	The MSB is on the left and the LSB is on the right.

Notes:

When using the on-chip emulator (E7, E8) for H8/3687 program development and debug following restrictions must be noted.

7. Use channel 1 of the SCI3 (P21/RXD, P22/TXD) in on-board programming mode by
mode.

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

H8/3687 Group manuals:

Document Title	Docume
H8/3687 Group Hardware Manual	This man
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

RENESAS

Section 2 CPU

Section 3 Exception Handling

Section 4 Address Break

Section 13 Timer Z

Section 17 I²C Bus Interface 2 (IIC2)

Appendix B I/O Port Block Diagrams

Section 12 Timer V

Section 13 Timer Z

Section 15 14-Bit PWM

Section 16 Serial Communication Interface 3 (SCI3)

Section 20 Power-On Reset and Low-Voltage Detection Circuits (Optional)

Section 23 Electrical Characteristics

— RTC (can be used as a free running counter)

— Timer B1 (8-bit timer)

— Timer V (8-bit timer)

— Timer Z (16-bit timer)

— 14-bit PWM

— Watchdog timer

— SCI (Asynchronous or clocked synchronous serial communication interface) × 2

— I²C Bus Interface (conforms to the I²C bus interface format that is advocated by P
Electronics)

— 10-bit A/D converter

		H8/3686	HD6433686	HD6433686G	48 kbytes	3 kbytes
		H8/3685	HD6433685	HD6433685G	40 kbytes	3 kbytes
		H8/3684	HD6433684	HD6433684G	32 kbytes	3 kbytes
		H8/3683	HD6433683	HD6433683G	24 kbytes	3 kbytes
		H8/3682	HD6433682	HD6433682G	16 kbytes	3 kbytes
EEPROM stacked version (512 bytes)	Flash memory version	H8/3687N	—	HD64N3687G	56 kbytes	4 kbytes
	Mask-ROM version		—	HD6483687G	56 kbytes	3 kbytes

- General I/O ports
  - I/O pins: 45 I/O pins (43 I/O pins for H8/3687N), 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 H8/3687N)
  - I^2C bus interface (conforms to the I^2C bus interface format that is advocated by Ph Electronics)
- Supports various power-down states

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

- Compact package

Package	Code	Body Size	Pin Pitch
LQFP-64	FP-64E	10.0 × 10.0 mm	0.5 mm
QFP-64	FP-64A	14.0 × 14.0 mm	0.8 mm

Only LQFP-64 (FP-64E) for H8/3687N package

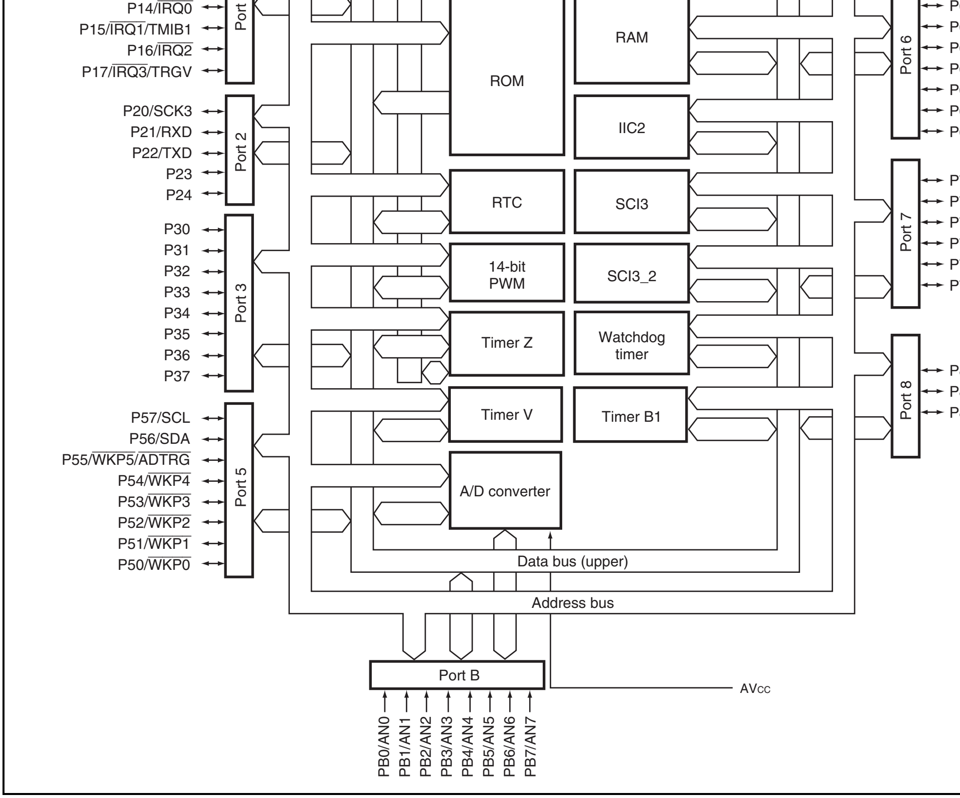
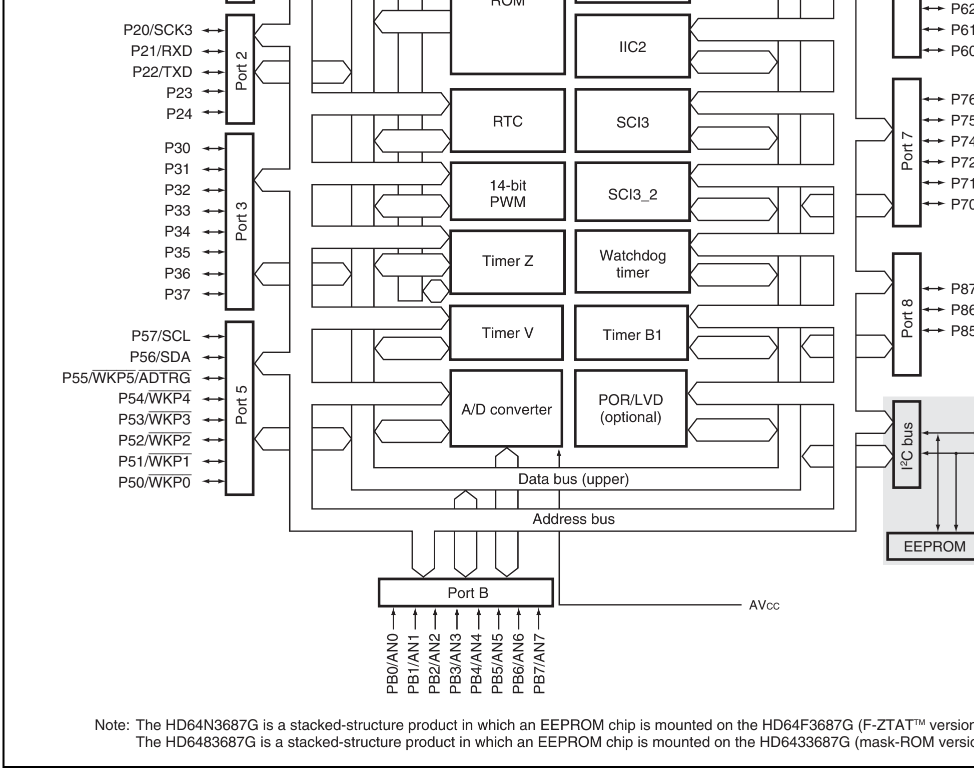


Figure 1.1 Internal Block Diagram of H8/3687 Group of F-ZTAT™ and Mask-ROM Versions



Note: The HD64N3687G is a stacked-structure product in which an EEPROM chip is mounted on the HD64F3687G (F-ZTAT™ version
The HD6483687G is a stacked-structure product in which an EEPROM chip is mounted on the HD6433687G (mask-ROM versio

Figure 1.2 Internal Block Diagram of H8/3687N (EEPROM Stacked Versio

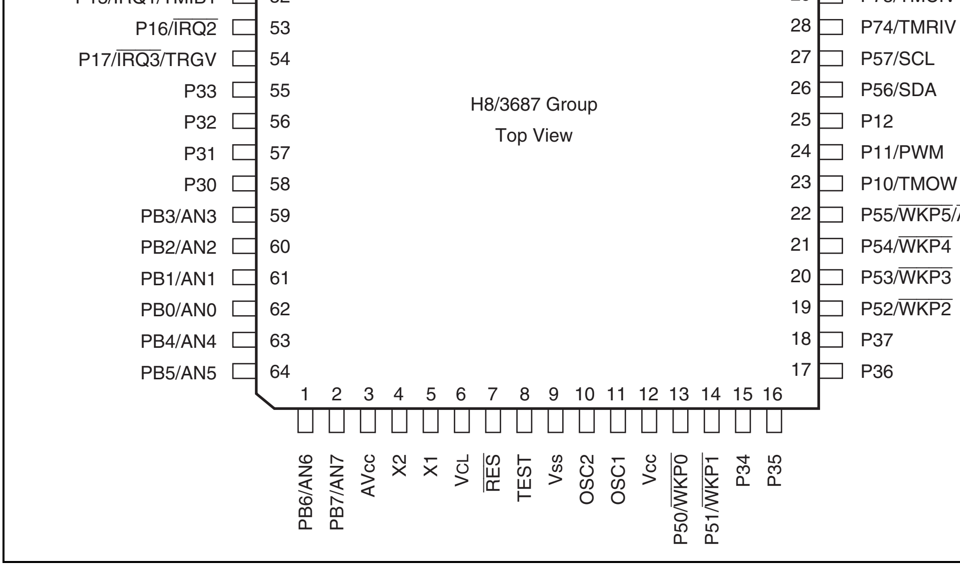


Figure 1.3 Pin Arrangement of H8/3687 Group of F-ZTAT™ and Mask-ROM V
(FP-64E, FP-64A)

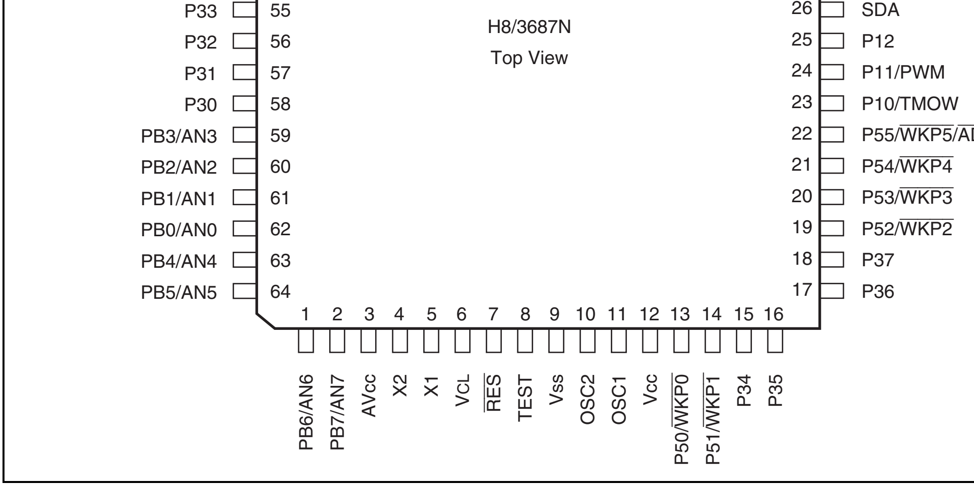


Figure 1.4 Pin Arrangement of H8/3687N (EEPROM Stacked Version) (FP-64E)

	AV_{CC}	3	Input	Analog power supply pin for the A/D c When the A/D converter is not used, c this pin to the system power supply.
	V_{CL}	6	Input	Internal step-down power supply pin. capacitor of around 0.1 μF between th the Vss pin for stabilization.
Clock pins	OSC1	11	Input	These pins connect with crystal or cer resonator for the system clock, or can to input an external clock. See section 5, Clock Pulse Generator typical connection.
	OSC2	10	Output	
	X1	5	Input	These pins connect with a 32.768 kHz resonator for the subclock. See sectio Pulse Generators, for a typical connec
	X2	4	Output	
System control	$\overline{\text{RES}}$	7	Input	Reset pin. The pull-up resistor (typ. 15 incorporated. When driven low, the ch
	TEST	8	Input	Test pin. Connect this pin to Vss.
Interrupt pins	$\overline{\text{NMI}}$	35	Input	Non-maskable interrupt request input sure to pull-up by a pull-up resistor.
	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$	51 to 54	Input	External interrupt request input pins. C the rising or falling edge.
	$\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$	13, 14, 19 to 22	Input	External interrupt request input pins. C the rising or falling edge.
RTC	TMOW	23	Output	This is an output pin for divided clocks
Timer B1	TMIB1	52	Input	External event input pin.

	FTIOB0	34	I/O	Output compare output/input capture input/PWM output pin
	FTIOC0	33	I/O	Output compare output/input capture input/PWM sync output pin (at a reset complementary PWM mode)
	FTIOD0	32	I/O	Output compare output/input capture input/PWM output pin
	FTIOA1	37	I/O	Output compare output/input capture input/PWM output pin (at a reset, complementary PWM mode)
	FTIOB1 to FTIOD1	38 to 40	I/O	Output compare output/input capture input/PWM output pin
14-bit PWM	PWM	24	Output	14-bit PWM square wave output pin
I^2C bus interface (IIC)	SDA*[1]	26	I/O	IIC data I/O pin. Can directly drive a b NMOS open-drain output. When using external pull-up resistance is required
	SCL*[1]	27	I/O (EEPROM: Input)	IIC clock I/O pin. Can directly drive a NMOS open-drain output. When using external pull-up resistance is required
Serial communication interface (SCI)	TXD, TXD_2	46, 50	Output	Transmit data output pin
	RXD, RXD_2	45, 49	Input	Receive data input pin
	SCK3, SCK3_2	44, 48	I/O	Clock I/O pin
A/D converter	AN7 to AN0	1, 2, 59 to 64	Input	Analog input pin
	$\overline{\text{ADTRG}}$	22	Input	A/D converter trigger input pin.

P57 to P50	13, 14, 19 to 22, 26*2, 27*2	I/O	8-bit I/O port
P67 to P60	32 to 34, 36, 37 to 40	I/O	8-bit I/O port
P76 to P74, P72 to P70	28 to 30, 48 to 50	I/O	6-bit I/O port
P87 to P85	41 to 43	I/O	3-bit I/O port.

Notes: 1. These pins are only available for the I^2C bus interface in the H8/3687N. Since bus is disabled after canceling a reset, the ICE bit in ICCR1 must be set to 1 the program.

2. The P57 and P56 pins are not available in the H8/3687N.

- 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

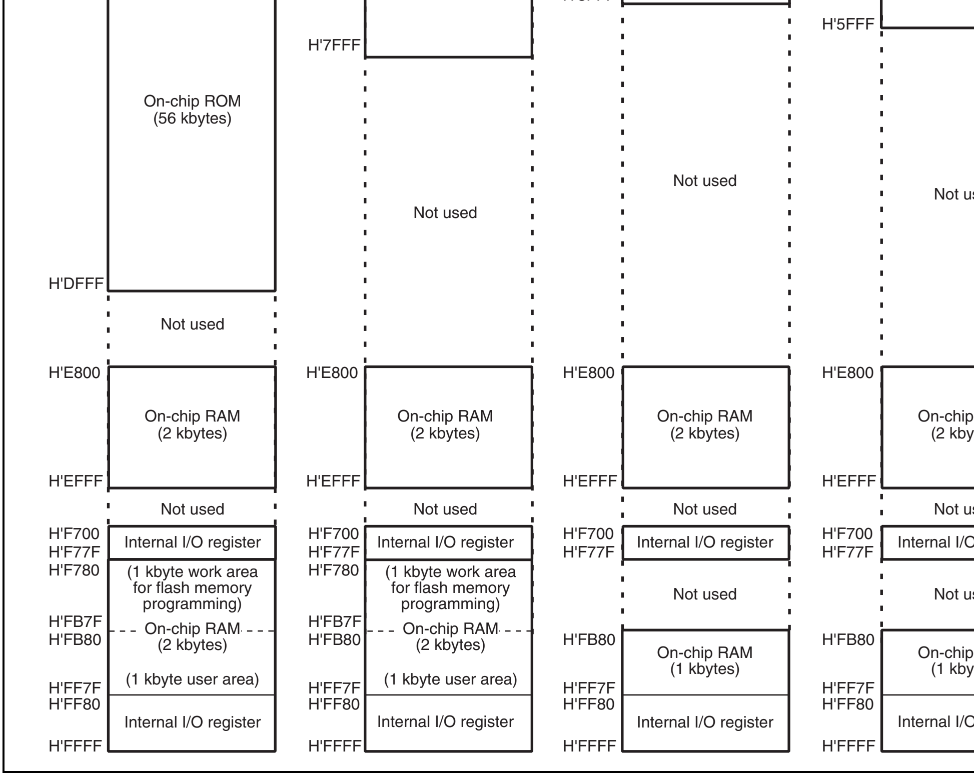


Figure 2.1 Memory Map (1)

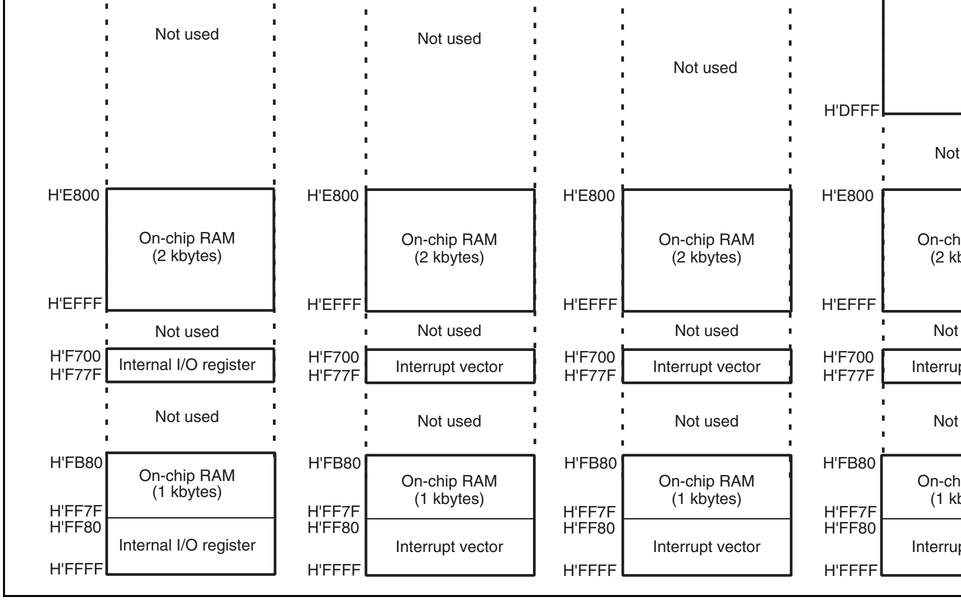


Figure 2.1 Memory Map (2)

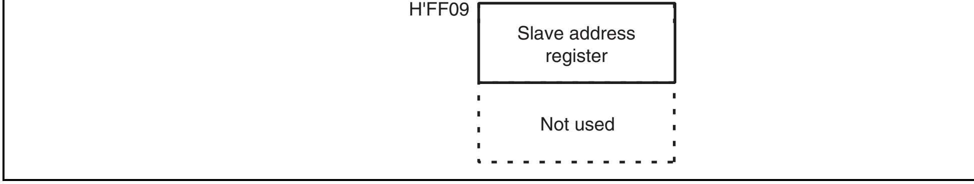


Figure 2.1 Memory Map (3)

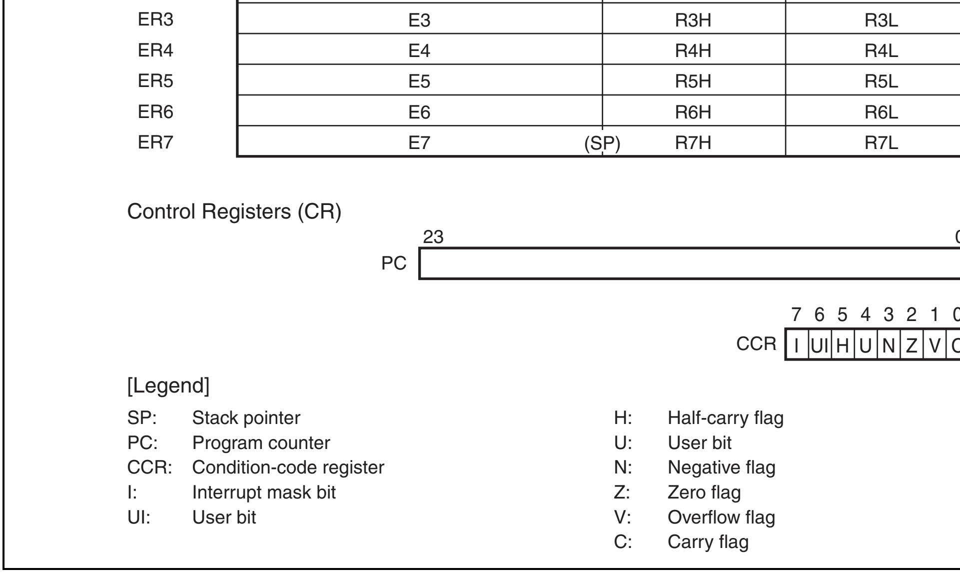


Figure 2.2 CPU Registers

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.

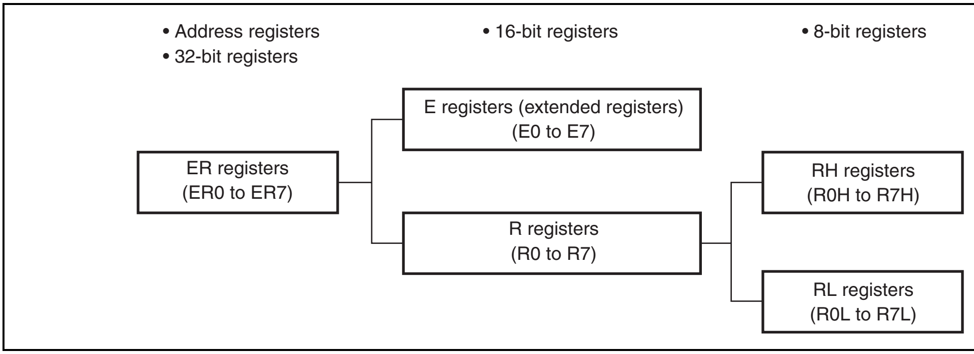


Figure 2.3 Usage of General Registers

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

Figure 2.4   Relationship between Stack Pointer and Stack Area

2.2.2 Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute. 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 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 cleared 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 ca The carry flag is also used as a bit accumulato manipulation instructions.

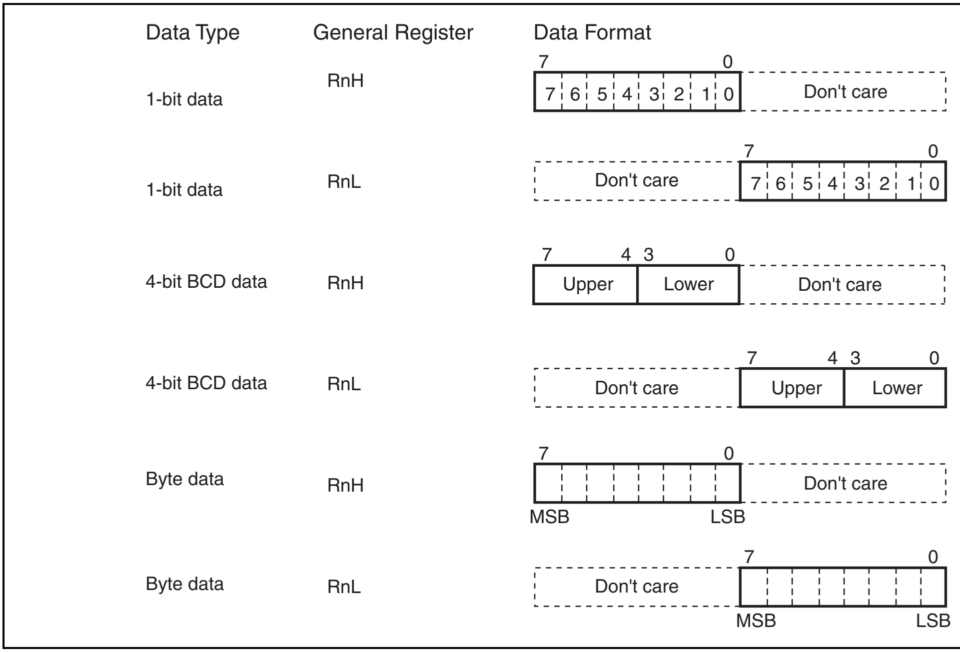


Figure 2.5 General Register Data Formats (1)

[Legend]

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

Figure 2.5 General Register Data Formats (2)

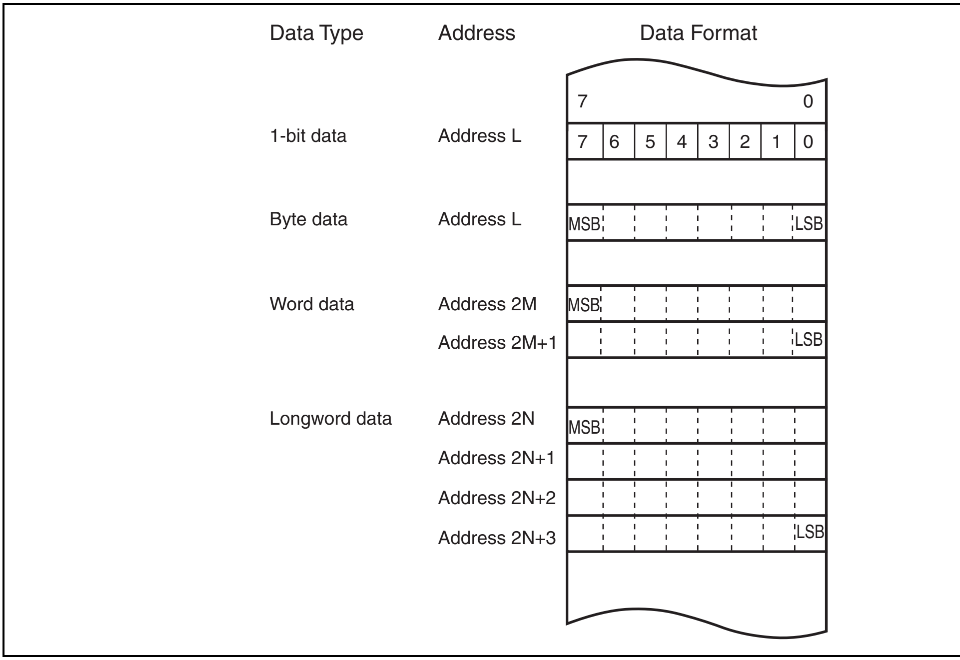


Figure 2.6 Memory Data Formats

Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register or address register)
(EAd)	Destination operand
(EAs)	Source operand
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical XOR
→	Move
¬	NOT (logical complement)
:3/:8/:16/:24	3-, 8-, 16-, or 24-bit length

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

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

Note: * Refers to the operand size.

B: Byte

W: Word

L: Longword

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

Note: * efers to the operand size.

B: Byte

W: Word

L: Longword

NEG	B/W/L	0 − Rd → Rd 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

BNOT	B	¬ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory operanc 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 gene register or memory operand and stores the result in the carry f 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 r 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 gener 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	$N \oplus V = 0$
BLT	Less than	$N \oplus V = 1$
BGT	Greater than	$Z \vee (N \oplus V) = 0$
BLE	Less or equal	$Z \vee (N \oplus V) = 1$

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

Note: * Bcc is the general name for conditional branch instructions.

		code register size is one byte, but in transfer to memory, data is by word access.
ANDC	B	CCR ∧ #IMM → CCR Logically ANDs the CCR with immediate data.
ORC	B	CCR ∨ #IMM → CCR Logically ORs the CCR with immediate data.
XORC	B	CCR ⊕ #IMM → CCR Logically XORs the CCR with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

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

else next;

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

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

Some instructions have two operation fields.

- Register Field

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

- Effective Address Extension

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

- Condition Field

  Specifies the branching condition of Bcc instructions.

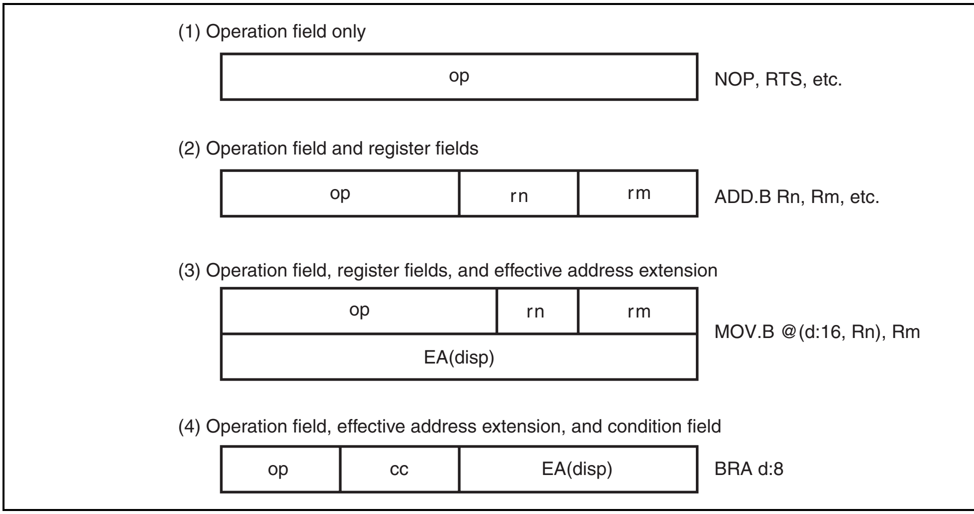


Figure 2.7 Instruction Formats

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

Table 2.10 Addressing Modes

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

Register Direct—Rn

The register field of the instruction specifies an 8-, 16-, or 32-bit general register contai 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 with post-increment—@ERn+

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

- Register indirect with pre-decrement—@-ERn

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

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

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

For an 8-bit absolute address, the upper 16 bits are all assumed to be 1 (H'FFFF). For a 16
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.

operand.

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

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

This mode is used in the BSR instruction. An 8-bit or 16-bit displacement contained in t
instruction is sign-extended and added to the 24-bit PC contents to generate a branch ad
PC value to which the displacement is added is the address of the first byte of the next in
so the possible branching range is –126 to +128 bytes (–63 to +64 words) or –32766 to
bytes (–16383 to +16384 words) from the branch instruction. The resulting value should
even number.

Memory Indirect—@@aa:8

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

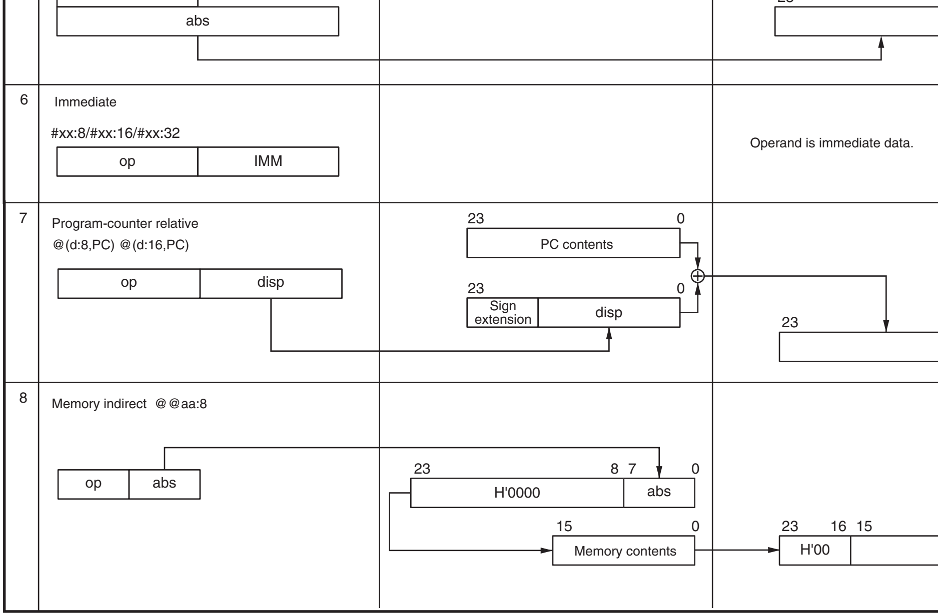
Note that the first part of the address range is also the exception vector area.

2.5.2 Effective Address Calculation

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

Table 2.12 Effective Address Calculation (1)

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



[Legend]

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

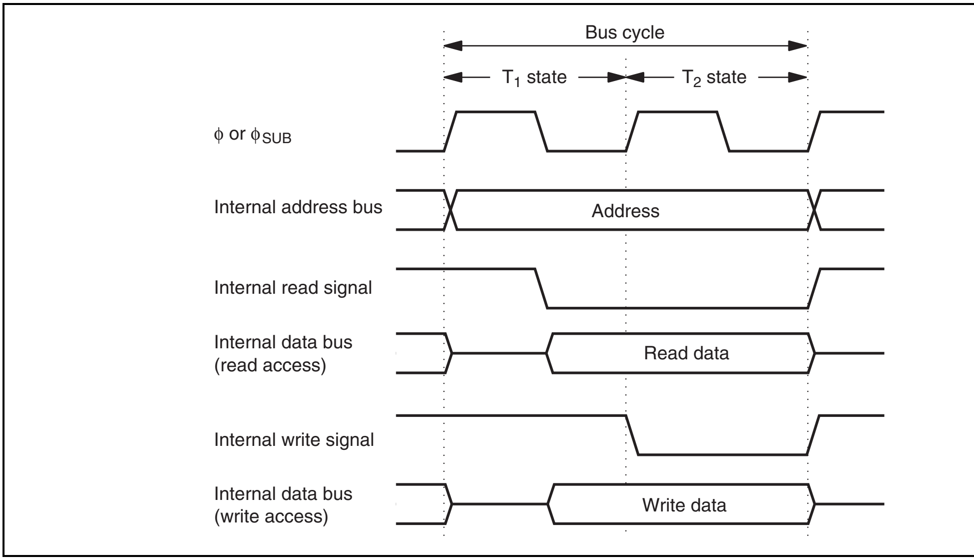


Figure 2.9 On-Chip Memory Access Cycle

module.

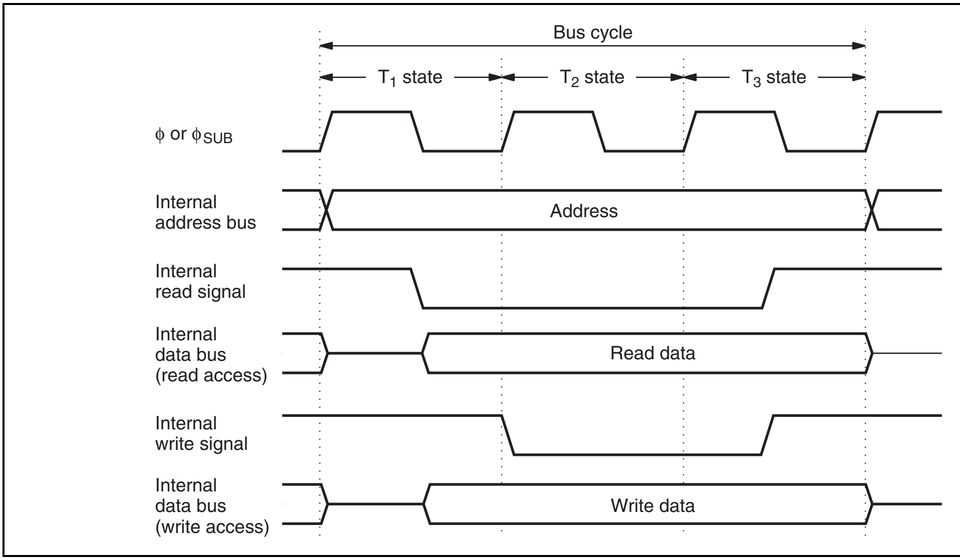


Figure 2.10 On-Chip Peripheral Module Access Cycle (3-State Access)

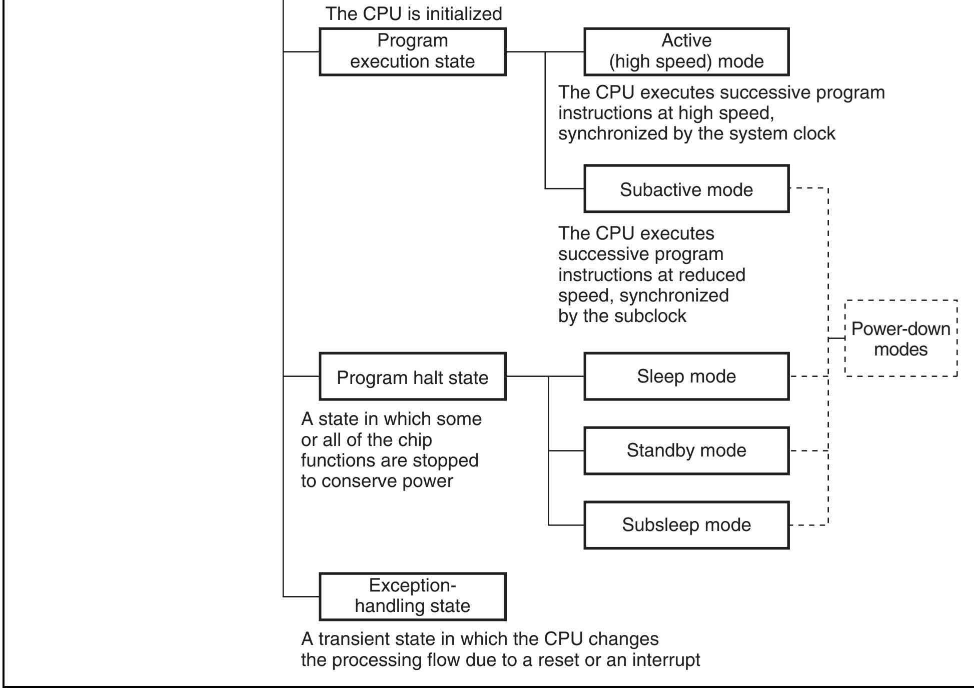


Figure 2.11 CPU Operation States

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 o
I/O registers areas available to the user. When data is transferred from CPU to empty are
transferred data will be lost. This action may also cause the CPU to malfunction. When d
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'FF
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.

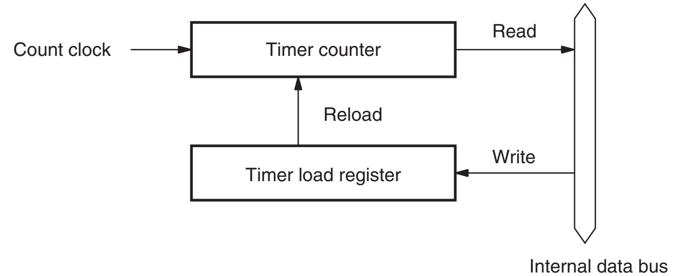


Figure 2.13 Example of Timer Configuration with Two Registers Allocated to S Address

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-le 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 manipulatio 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 I
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
an input pin by the BCLR instruction is shown below. It is assumed that a high-level sign
input to this input pin.

```
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 out 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 in
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.

—	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
CPU	Direct transition by executing the SLEEP instruction	13	H'001A to H'001B
External interrupt pin	IRQ0 Low-voltage detection interrupt*	14	H'001C to H'001D
	IRQ1	15	H'001E to H'001F
	IRQ2	16	H'0020 to H'0021
	IRQ3	17	H'0022 to H'0023
	WKP	18	H'0024 to H'0025
RTC	Overflow	19	H'0026 to H'0027
—	Reserved for system use	20	H'0028 to H'0029
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

	Timer Z overflow		
	Compare match/input capture A1 to D1 Timer Z overflow Timer Z underflow	27	H'0036 to H'0037
Timer B1	Timer B1 overflow	29	H'003A to H'003B
SCI3_2	Receive data full Transmit data empty Transmit end Receive error	32	H'0040 to H'0041

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

3.2 Register Descriptions

Interrupts are controlled by the following registers.

- Interrupt edge select register 1 (IEGR1)
- Interrupt edge select register 2 (IEGR2)
- Interrupt enable register 1 (IENR1)
- Interrupt enable register 2 (IENR2)
- Interrupt flag register 1 (IRR1)
- Interrupt flag register 2 (IRR2)
- Wakeup interrupt flag register (IWPR)

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

				enabled.
5	IENWP	0	R/W	Wakeup Interrupt Enable This bit is an enable bit, which is common to the $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$. When the bit is set to 1, interrup 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 the pin are enabled.
2	IEN2	0	R/W	IRQ2 Interrupt Enable When this bit is set to 1, interrupt requests of the pin are enabled.
1	IEN1	0	R/W	IRQ1 Interrupt Enable When this bit is set to 1, interrupt requests of the pin are enabled.
0	IEN0	0	R/W	IRQ0 Interrupt Enable When this bit is set to 1, interrupt requests of the 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 cl operations are performed while I = 0, and as a result a conflict arises between the clear in and an interrupt request, exception handling for the interrupt will be executed after the cl instruction has been executed.

4 to 0	—	All 1	—	Reserved These bits are always read as 1.

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 c
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.

3.2.5 Interrupt Flag Register 1 (IRR1)

IRR1 is a status flag register for direct transition interrupts, RTC interrupts, and $\overline{IRQ3}$ to
interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	IRRDT	0	R/W	Direct Transfer Interrupt Request Flag [Setting condition] When a direct transfer is made by executing a S instruction while DTON in SYSCR2 is set to 1. [Clearing condition] When IRRDT is cleared by writing 0
6	IRRTA	0	R/W	RTC Interrupt Request Flag [Setting condition] When the RTC counter value overflows [Clearing condition] When IRRTA is cleared by writing 0

2	IRRI2	0	R/W	IRQ2 Interrupt Request Flag [Setting condition] When $\overline{IRQ2}$ 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 $\overline{IRQ1}$ pin is designated for interrupt input a designated signal edge is detected. [Clearing condition] When IRRI1 is cleared by writing 0
0	IRRI0	0	R/W	IRQ0 Interrupt Request Flag [Setting condition] When $\overline{IRQ0}$ pin is designated for interrupt input a designated signal edge is detected. [Clearing condition] When IRRI0 is cleared by writing 0

Bit	Bit Name	Initial Value	R/W	Description
				When the timer B1 counter value overflows [Clearing condition] When IRRTB1 is cleared by writing 0
4 to 0	—	All 1	—	Reserved These bits are always read as 1.

3.2.7 Wakeup Interrupt Flag Register (IWPR)

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.

				When $\overline{\text{WKP2}}$ pin is designated for interrupt input designated signal edge is detected. [Clearing condition] When IWPF2 is cleared by writing 0.
1	IWPF1	0	R/W	WKP1 Interrupt Request Flag [Setting condition] 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.

The reset exception handling sequence is as follows:

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'000 data in that address is sent to the program counter (PC) as the start address, and prog execution starts from that address.

3.4 Interrupt Exception Handling

3.4.1 External Interrupts

As the external interrupts, there are NMI, IRQ3 to IRQ0, and WKP5 to WKP0 interrupt

NMI Interrupt

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

NMI is the highest-priority interrupt, and can always be accepted without depending 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 eithe 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 designa edge is input, the corresponding bit in IRR1 is set to 1, requesting the CPU of an inte These interrupts can be masked by setting bits IEN3 to IEN0 in IENR1.

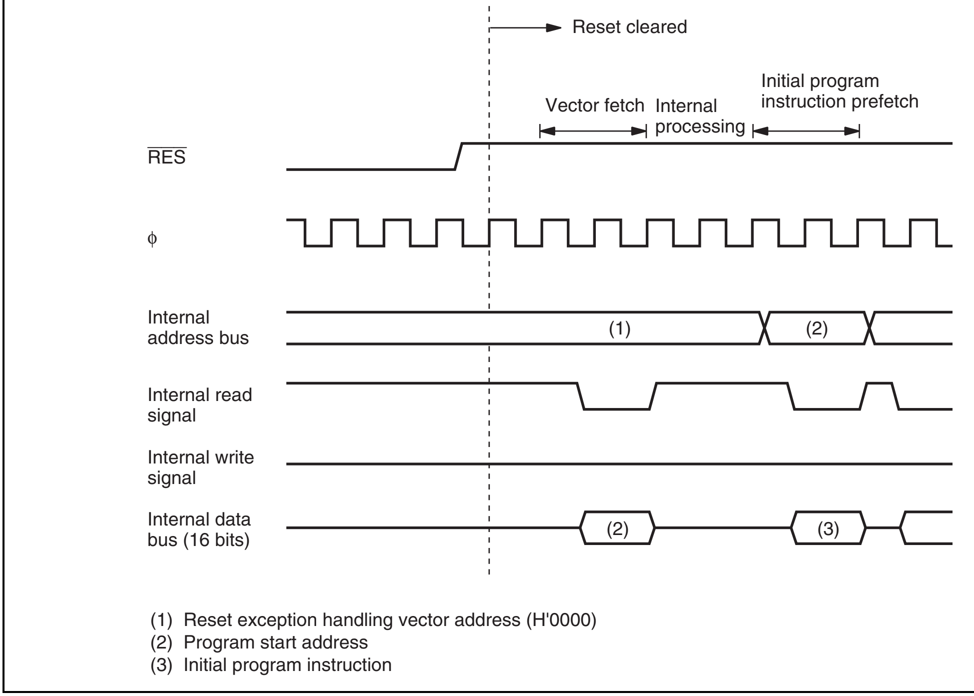


Figure 3.1 Reset Sequence

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 re signal is sent to the interrupt controller.
2. When multiple interrupt requests are generated, the interrupt controller requests to th the interrupt handling with the highest priority at that time according to table 3.1. Ot interrupt requests are held pending.
3. The CPU accepts the NMI and address break without depending on the I bit value. C interrupt requests are accepted, if the I bit is cleared to 0 in CCR; if the I bit is set to interrupt request is held pending.
4. If the CPU accepts the interrupt after processing of the current instruction is complet interrupt exception handling will begin. First, both PC and CCR are pushed onto the state of the stack at this time is shown in figure 3.2. The PC value pushed onto the st address of the first instruction to be executed upon return from interrupt handling.
5. Then, the I bit of CCR is set to 1, masking further interrupts excluding the NMI and 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 a 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 R 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 first 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.

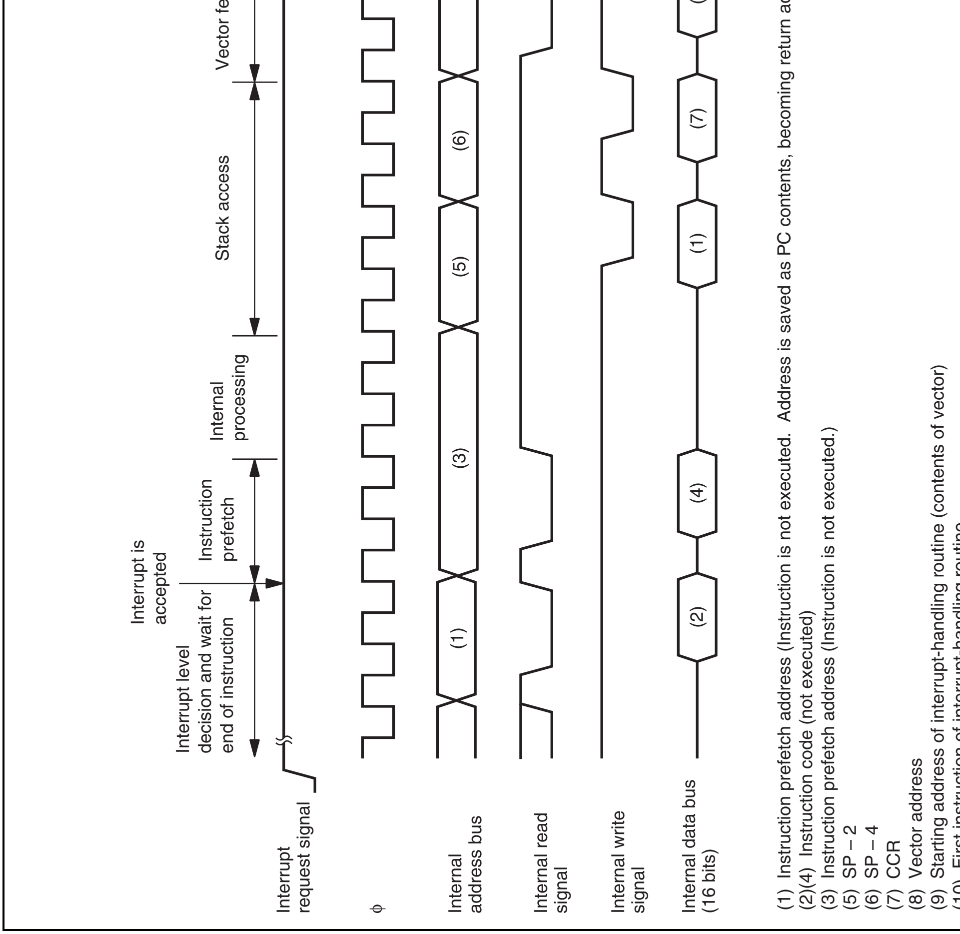


Figure 3.3 Interrupt Sequence

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

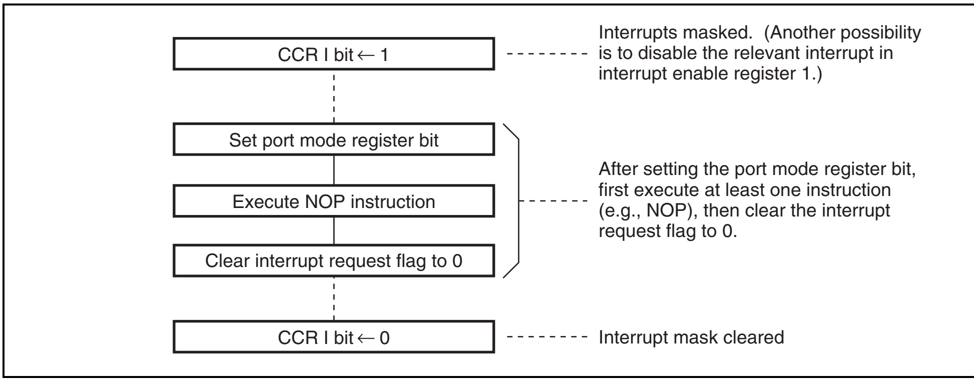


Figure 3.4 Port Mode Register Setting and Interrupt Request Flag Clearing Pro

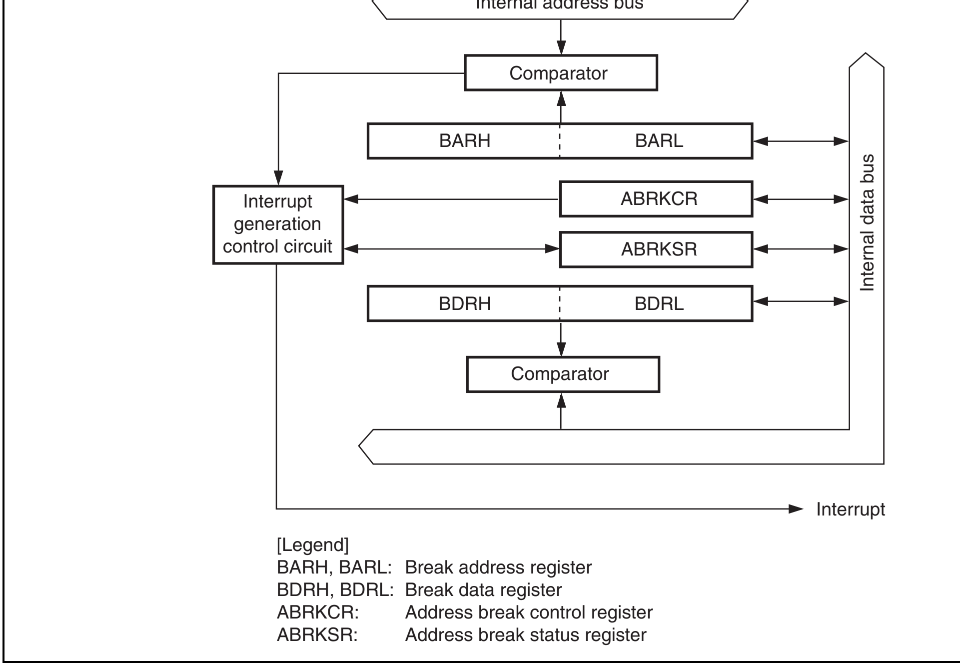


Figure 4.1 Block Diagram of Address Break

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)

				executing RTE is masked and then one instructio be executed. When this bit is 1, the interrupt is n masked.
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 betweer address set in BAR and the internal address bus
2	ACMP0	0	R/W	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 betweer 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.

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.

4.1.3 Break Address Registers (BARH, BARL)

BARH and BARL are 16-bit read/write registers that set the address for generating an ad break interrupt. When setting the address break condition to the instruction execution cy the first byte address of the instruction. The initial value of this register is H'FFFF.

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 execute
The address break interrupt is not masked by the I bit in CCR of the CPU.

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

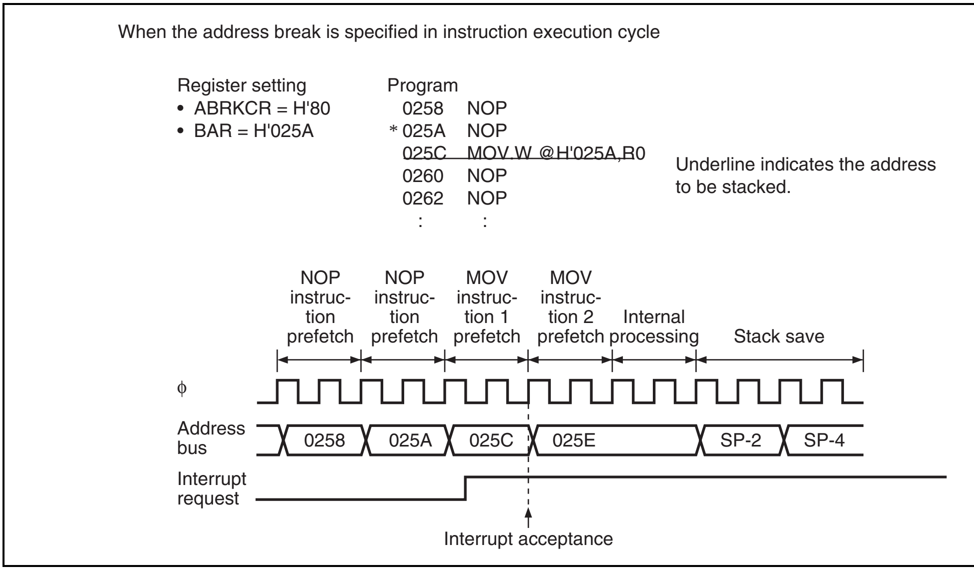


Figure 4.2 Address Break Interrupt Operation Example (1)

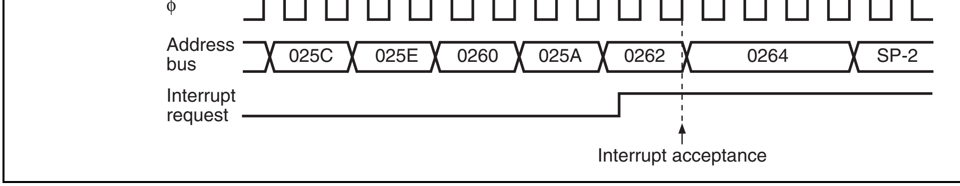


Figure 4.2 Address Break Interrupt Operation Example (2)

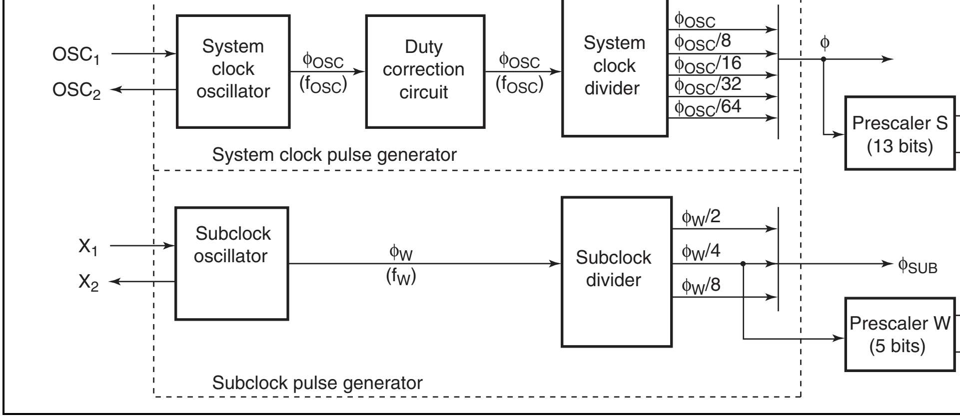


Figure 5.1 Block Diagram of Clock Pulse Generators

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.

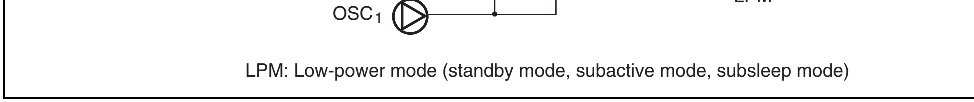


Figure 5.2 Block Diagram of System Clock Generator

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 reso resonator having the characteristics given in table 5.1 should be used.

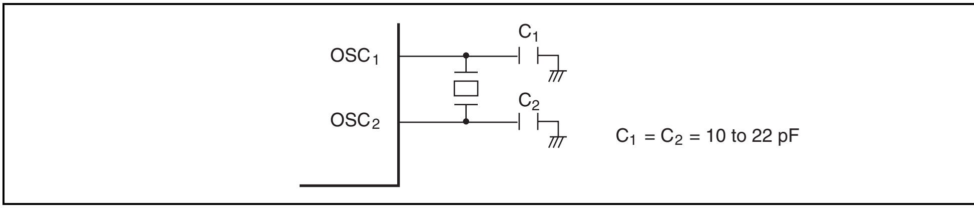


Figure 5.3 Typical Connection to Crystal Resonator

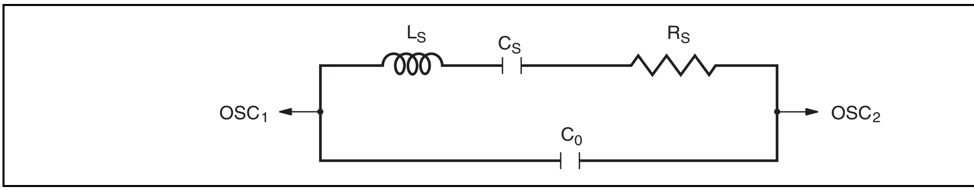


Figure 5.4 Equivalent Circuit of Crystal Resonator

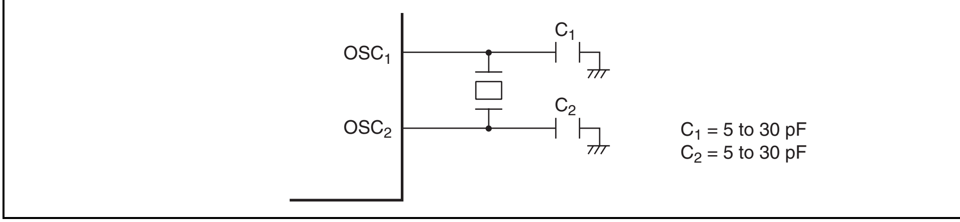


Figure 5.5 Typical Connection to Ceramic Resonator

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%.

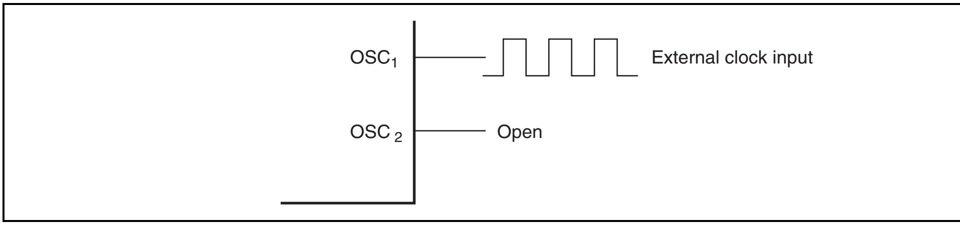


Figure 5.6 Example of External Clock Input

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-kH resonator.

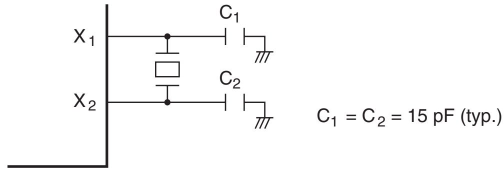
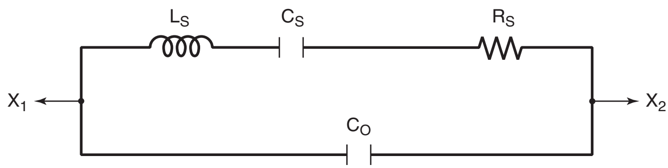


Figure 5.8 Typical Connection to 32.768-kHz Crystal Resonator



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 (ø) 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 ($ø_W$/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.

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

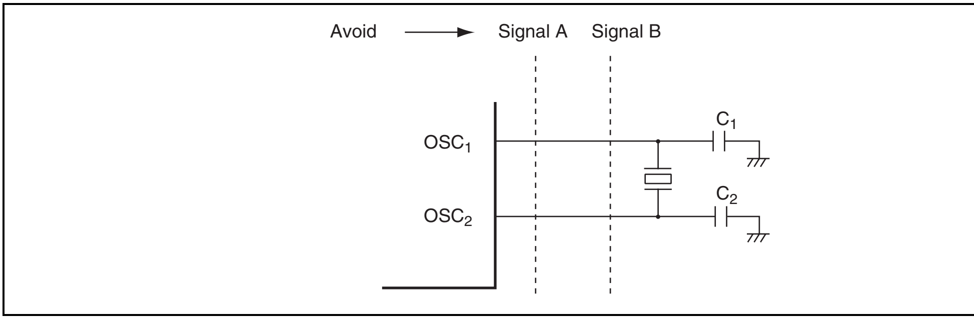


Figure 5.11 Example of Incorrect Board Design

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, the RTC 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)
- Module standby control register 2 (MSTCR2)

				1: Enters standby mode. For details, see table 6.2.
6 5 4	STS2 STS1 STS0	0 0 0	R/W R/W R/W	Standby Timer Select 2 to 0 These bits designate the time the CPU and perip modules wait for stable clock operation after exit standby mode, subactive mode, or subsleep mod 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.

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

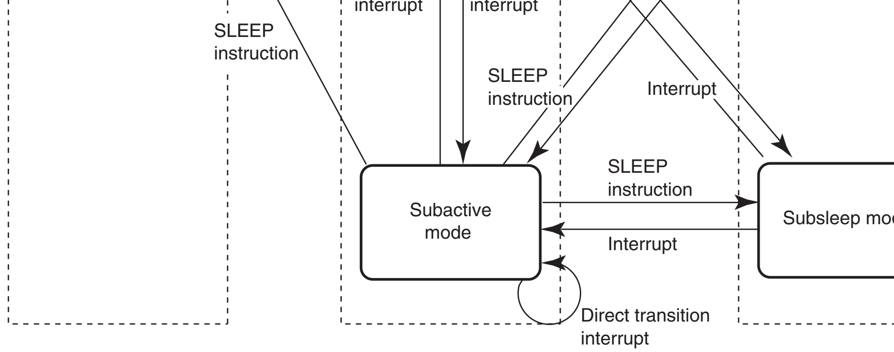
Legend: Don't care.

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	—	0	—	Reserved This bit is always read as 0.
1	MSTTV	0	R/W	Timer V Module Standby Timer V enters standby mode when this bit is s
0	MSTTA	0	R/W	RTC Module Standby RTC enters standby mode when this bit is set t

4	MSTTB1	0	R/W	Timer B1 Module Standby Timer B1 enters standby mode when this bit is s
3, 2	—	All 0	—	Reserved These bits are always read as 0.
1	MSTTZ	0	R/W	Timer Z Module Standby Timer Z enters standby mode when this bit is se
0	MSTPWM	0	R/W	PWM Module Standby PWM enters standby mode when this bit is set t

6.2 Mode Transitions and States of LSI

Figure 6.1 shows the possible transitions among these operating modes. A transition is m the program execution state to the program halt state by executing a SLEEP instruction. I allow for returning from the program halt state to the program execution state. A direct tr between active mode and subactive mode, which are both program execution states, can b without halting the program. The operating frequency can also be changed in the same m making a transition directly from active mode to active mode, and from subactive mode t subactive mode. $\overline{\text{RES}}$ input enables transitions from a mode to the reset state. Table 6.2 s transition conditions of each mode after the SLEEP instruction is executed and a mode to by an interrupt. Table 6.3 shows the internal states of the LSI in each mode.



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

Figure 6.1 Mode Transition Diagram

1	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, SCI3_2 A/D converter are reset, and all registers are set to their initial values. To use t functions after entering active mode, reset the registers.

						co reta out hig imp sta
External interrupts	IRQ3 to IRQ0	Functioning	Functioning	Functioning	Functioning	Fur
	WKP5 to WKP0	Functioning	Functioning	Functioning	Functioning	Fur
Peripheral functions	RTC	Functioning	Functioning	Functioning if the timekeeping time-l function is selected, and retained if r		
	Timer V	Functioning	Functioning	Reset	Reset	Re
	Watchdog timer	Functioning	Functioning	Retained (functioning if the internal o selected as a count clock*)		
	SCI3, SCI3_2	Functioning	Functioning	Reset	Reset	Res
	IIC2	Functioning	Functioning	Retained*	Retained	Ret
	Timer B1	Functioning	Functioning	Retained*	Retained	Ret
	Timer Z	Functioning	Functioning	Retained (the counter increments ac subclocks if the internal clock (ϕ) is s a count clock*)		
	A/D converter	Functioning	Functioning	Reset	Reset	Re

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 mod
functioning. However, as long as the rated voltage is supplied, the contents of CPU regist
chip RAM, and some on-chip peripheral module registers are retained. On-chip RAM con
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 clock
generator starts. After the time set in bits STS2 to STS0 in SYSCR1 has elapsed, and inte
exception handling starts. Standby mode is not cleared if the I bit of CCR is set to 1 or the
requested interrupt is disabled in the interrupt enable register.

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 functioni
$\overline{\text{RES}}$ pin must be kept low until the pulse generator output stabilizes. After the pulse gene
output has stabilized, the CPU starts reset exception handling if the $\overline{\text{RES}}$ pin is driven hig

6.2.3 Subsleep Mode

In subsleep mode, operation of the CPU and on-chip peripheral modules other than RTC
As long as a required voltage is applied, the contents of CPU registers, the on-chip RAM,
some registers of the on-chip peripheral modules are retained. I/O ports keep the same sta
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 i
or the requested interrupt is disabled in the interrupt enable register. After subsleep mode

The operating frequency of subactive mode is selected from $\phi_w/2$, $\phi_w/4$, and $\phi_w/8$ by the SA0 bits in SYSCR2. After the SLEEP instruction is executed, the operating frequency the frequency which is set before the execution. When the SLEEP instruction is execute subactive mode, a transition to sleep mode, subsleep mode, standby mode, active mode, subactive mode is made, depending on the combination of SYSCR1 and SYSCR2. Whe pin goes low, the system clock pulse generator starts. Since system clock signals are sup the entire chip as soon as the system clock pulse generator starts functioning, the $\overline{\text{RES}}$ pi kept low until the pulse generator output stabilizes. After the pulse generator output has 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 in SYSCR2. The operating frequency changes to the set frequency after SLEEP instructi 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 h
(the direct transition time) is calculated by equation (1).

Direct transition time = {(number of SLEEP instruction execution states) + (number of in
processing states)}× (tcyc before transition) + (number of interrupt exception handling sta
(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 h
(the direct transition time) is calculated by equation (2).

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

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

erased in turn.

- Reprogramming capability
  - 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 flash memory. The thick lines indicate eras the narrow lines indicate programming units, and the values are addresses. The 56-kbyt memory is divided into 1 kbyte × 4 blocks, 28 kbytes × 1 block, 16 kbytes × 1 block, an × 1 block. The 32-kbyte flash memory is divided into 1 kbyte × 4 blocks and 28 kbytes × Erasing is performed in these units. Programming is performed in 128-byte units startin address with lower eight bits H'00 or H'80.

Erase unit	H'0880	H'0881	H'0882		H'08FF
1 kbyte					
	H'0B80	H'0B81	H'0B82		H'0BFF
	H'0C00	H'0C01	H'0C02	← Programming unit: 128 bytes →	H'0C7F
Erase unit	H'0C80	H'0C81	H'0C82		H'0CFF
1 kbyte					
	H'0F80	H'0F81	H'0F82		H'0FFF
	H'1000	H'1001	H'1002	← Programming unit: 128 bytes →	H'107F
Erase unit	H'1080	H'1081	H'1082		H'10FF
28 kbytes					
	H'7F80	H'7F81	H'7F82		H'7FFF
	H'8000	H'8001	H'8002	← Programming unit: 128 bytes →	H'807F
Erase unit	H'8080	H'8081	H'8082		H'80FF
16 kbytes					
	H'BF80	H'BF81	H'BF82		H'BFFF
	H'C000	H'C001	H'C002	← Programming unit: 128 bytes →	H'C07F
Erase unit	H'C080	H'C081	H'C082		H'C0FF
8 kbytes					
	HDF80	H'DF81	H'DF82		H'DFFF

Figure 7.1 Flash Memory Block Configuration

7.2.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is a register that makes the flash memory change to program mode, program-v
mode, erase mode, or erase-verify mode. For details on register setting, refer to section
Memory Programming/Erasing.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.
6	SWE	0	R/W	Software Write Enable When this bit is set to 1, flash memory programming/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.

When this bit is set to 1 while SWE=1 and PSU= flash memory changes to program mode. When cleared to 0, program mode is cancelled.

7.2.2 Flash Memory Control Register 2 (FLMCR2)

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 section 7.5.3, Error Protection, for details.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

6	EB6	0	R/W	When this bit is set to 1, 8 bytes of H'C000 to H be erased.
5	EB5	0	R/W	When this bit is set to 1, 16 bytes of H'8000 to I will be erased.
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 registers 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	$\overline{NMI}$	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.

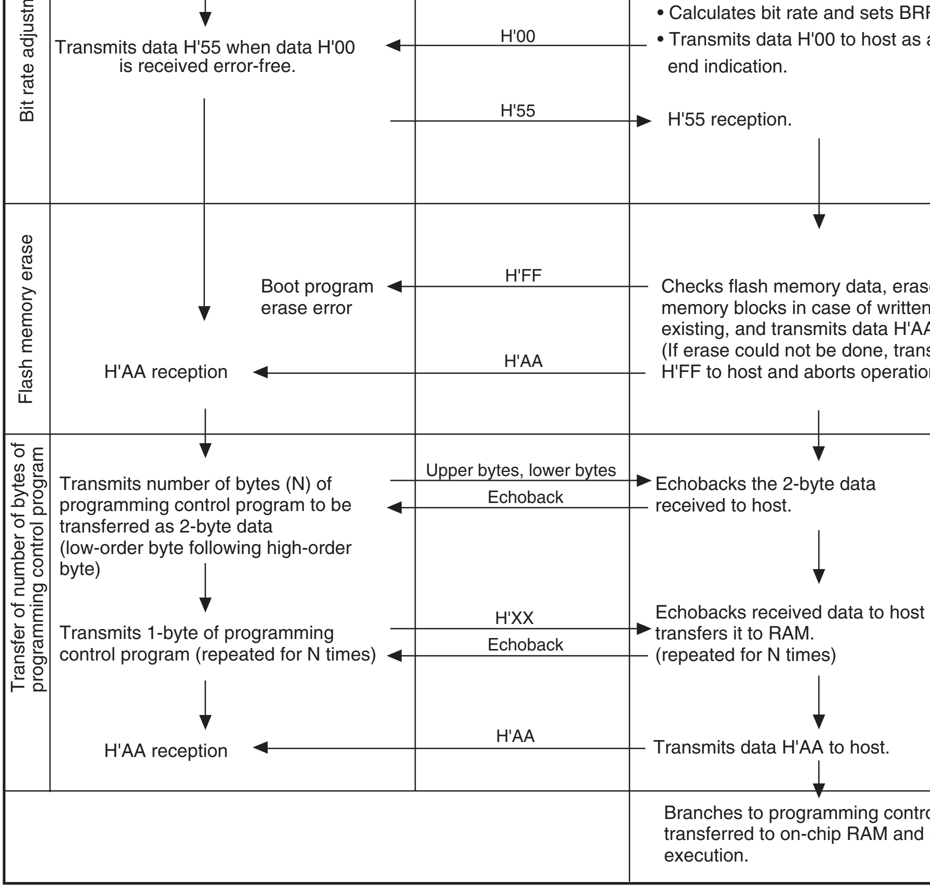
1. When boot mode is used, the flash memory programming control program must be p
the host beforehand. Prepare a programming control program in accordance with the
description in section 7.4, Flash Memory Programming/Erasing.
2. SCI3 should be set to asynchronous mode, and the transfer format as follows: 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 sh

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 program data or verify data with the host. The TxD pin is high (PCR22 = 1, P22 =
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 adjustn
Transmits data H'55 when data H'00 is received error-free.
H'00
• Calculates bit rate and sets BRF
• Transmits data H'00 to host as a end indication.
H'55
H'55 reception.
Flash memory erase
Boot program erase error
H'FF
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 operatior
H'AA reception
H'AA
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.

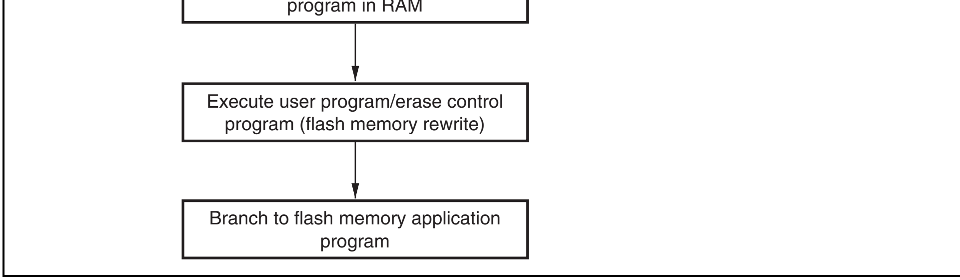
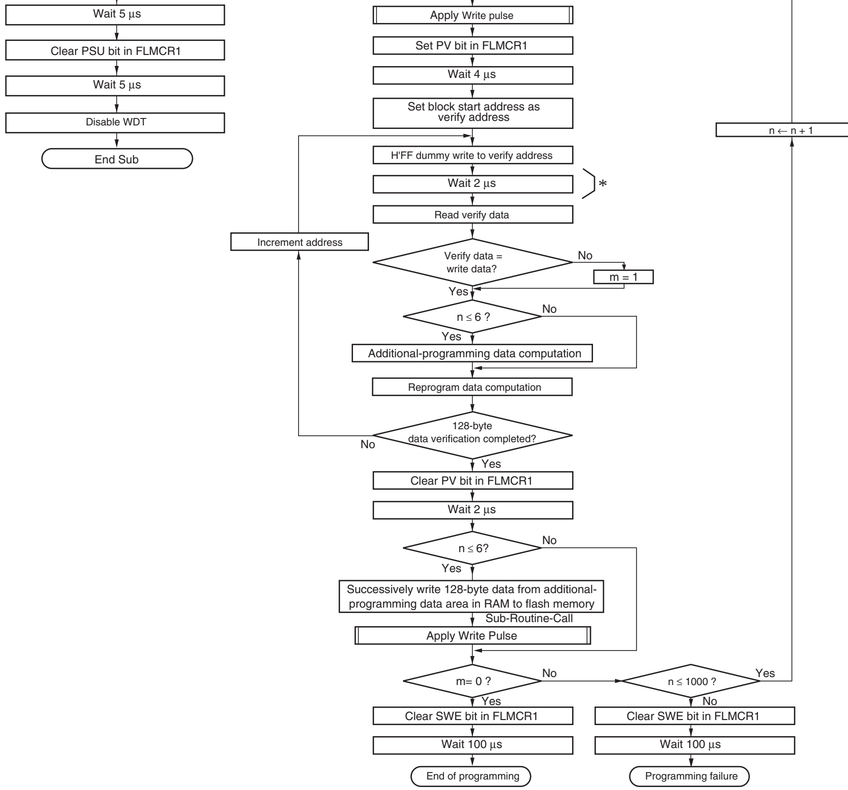


Figure 7.2 Programming/Erasing Flowchart Example in User Program Mo

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 whi 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 writte 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 I
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.

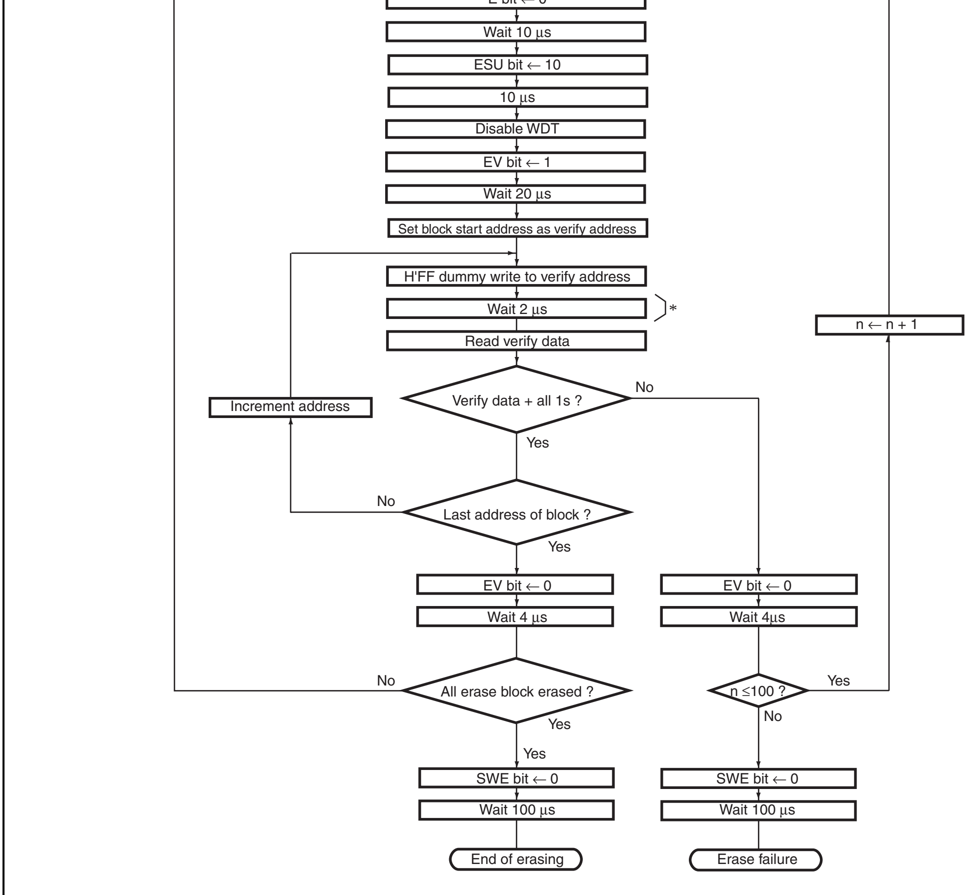
7.4.2 Erase/Erase-Verify

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

Figure 7.4 Erase/Erase-Verify Flowchart

entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In th
a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the A
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 forcibly aborted. Aborting the program/er
operation prevents damage to the flash memory due to overprogramming or overerasing

When the following errors are detected during programming/erasing of flash memory, th
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/erasing
(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 era
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.

Table 7.7 Flash Memory Operating States

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

		H8/3685	3 kbytes	H'E800 to H'EFFF, H'FB80 to H'F
		H8/3684	3 kbytes	H'E800 to H'EFFF, H'FB80 to H'F
		H8/3683	3 kbytes	H'E800 to H'EFFF, H'FB80 to H'F
		H8/3682	3 kbytes	H'E800 to H'EFFF, H'FB80 to H'F
EEPROM stacked version	Flash memory version	H8/3687N	4 kbytes	H'E800 to H'EFFF, H'F780 to H'F
	Mask-ROM version		3 kbytes	H'E800 to H'EFFF, H'FB80 to H'F

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, see appendix B.1, I/O Port Block Diagrams. For the executio manipulation instructions to the port control register and port data register, see section 2 Manipulation Instruction.

9.1 Port 1

Port 1 is a general I/O port also functioning as IRQ interrupt input pins, an RTC output p bit PWM output pin, a timer B1 input pin, and a timer V input pin. Figure 9.1 shows its configuration.

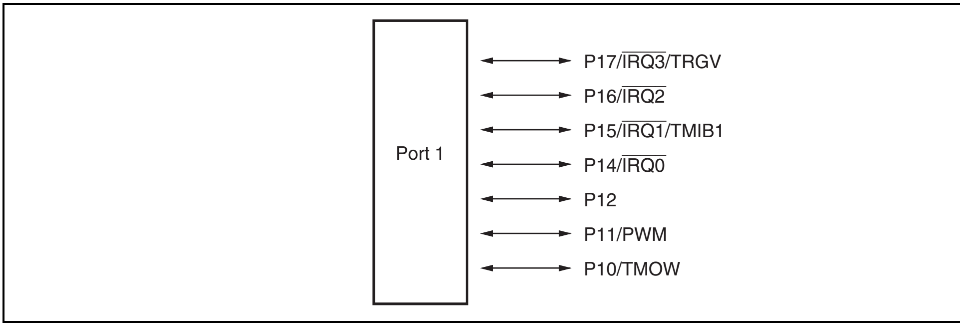


Figure 9.1 Port 1 Pin Configuration

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)

				0: General I/O port 1: $\overline{\text{IRQ2}}$ input pin
5	IRQ1	0	R/W	This bit selects the function of pin P15/$\overline{\text{IRQ1}}$/TMI 0: General I/O port 1: $\overline{\text{IRQ1}}$/TMIB1 input pin
4	IRQ0	0	R/W	This bit selects the function of pin P14/$\overline{\text{IRQ0}}$. 0: General I/O port 1: $\overline{\text{IRQ0}}$ input pin
3	TXD2	0	R/W	This bit selects the function of pin P72/TXD_2. 0: General I/O port 1: TXD_2 output pin
2	PWM	0	R/W	This bit selects the function of pin P11/PWM. 0: General I/O port 1: PWM output pin
1	TXD	0	R/W	This bit selects the function of pin P22/TXD. 0: General I/O port 1: TXD output pin
0	TMOW	0	R/W	This bit selects the function of pin P10/TMOW. 0: General I/O port 1: TMOW output pin

3	—	—	—	
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 stored in PDR1 are read. If PDR1 is read while are cleared to 0, the pin states are read regardl value stored in PDR1.
5	P15	0	R/W	
4	P14	0	R/W	
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.

Register	PMR1	PCR1	
Bit Name	**IRQ1**	**PCR15**	**Pin Function**
Setting value	0	0	P15 input pin
		1	P15 output pin
	1	X	$\overline{\text{IRQ1}}$ input/TMIB1 input pin

Legend: X: Don't care.

P14/$\overline{\text{IRQ0}}$ pin

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

Register	PMR1	PCR1	
Bit Name	**TMOW**	**PCR10**	**Pin Function**
Setting value	0	0	P10 input pin
		1	P10 output pin
	1	X	TMOW output pin

Legend: X: Don't care.

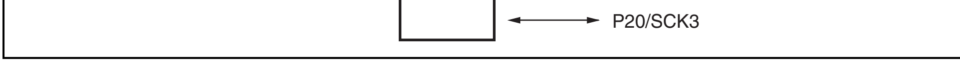


Figure 9.2 Port 2 Pin Configuration

Port 2 has the following registers.

- Port control register 2 (PCR2)
- Port data register 2 (PDR2)
- Port mode register 3 (PMR3)

9.2.1 Port Control Register 2 (PCR2)

PCR2 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 2

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	—	—	Reserved
4	PCR24	0	W	When each of the port 2 pins P24 to P20 functi general I/O port, setting a PCR2 bit to 1 makes corresponding pin an output port, while clearing 0 makes the pin an input port.
3	PCR23	0	W	
2	PCR22	0	W	
1	PCR21	0	W	
0	PCR20	0	W	

2	P22	0	R/W	stored in PDR2 is read. If PDR2 is read while PC
1	P21	0	R/W	are cleared to 0, the pin states are read regardle
0	P20	0	R/W	value stored in PDR2.

9.2.3 Port Mode Register 3 (PMR3)

PMR3 selects the CMOS output or NMOS open-drain output for port 2.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	—	Reserved These bits are always read as 0.
4	POF24	0	R/W	When the bit is set to 1, the corresponding pin is by PMOS and it functions as the NMOS open-dr output. When cleared to 0, the pin functions as t output.
3	POF23	0	R/W	
2 to 0	—	All 1	—	Reserved These bits are always read as 1.

P23 pin

Register	PCR2	
Bit Name	**PCR23**	**Pin Function**
Setting Value	0	P23 input pin
	1	P23 output pin

P22/TXD pin

Register	PMR1	PCR2	
Bit Name	**TXD**	**PCR22**	**Pin Function**
Setting Value	0	0	P22 input pin
		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.

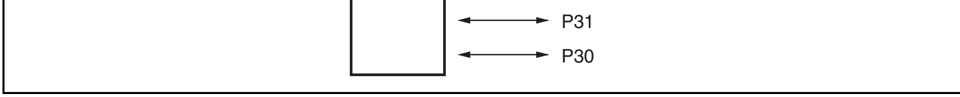


Figure 9.3 Port 3 Pin Configuration

Port 3 has the following registers.

- Port control register 3 (PCR3)
- Port data register 3 (PDR3)

9.3.1 Port Control Register 3 (PCR3)

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

Bit	Bit Name	Initial Value	R/W	Description
7	PCR37	0	W	Setting a PCR3 bit to 1 makes the correspondi output port, while clearing the bit to 0 makes th input port.
6	PCR36	0	W	
5	PCR35	0	W	
4	PCR34	0	W	
3	PCR33	0	W	
2	PCR32	0	W	
1	PCR31	0	W	
0	PCR30	0	W	

3	P33	0	R/W
2	P32	0	R/W
1	P31	0	R/W
0	P30	0	R/W

9.3.3 Pin Functions

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

P37 pin

Register	PCR3	
Bit Name	**PCR37**	**Pin Function**
Setting Value	0	P37 input pin
	1	P37 output pin

P36 pin

Register	PCR3	
Bit Name	**PCR36**	**Pin Function**
Setting Value	0	P36 input pin
	1	P36 output pin

Bit Name	PCR34	Pin Function
Setting Value	0	P34 input pin
	1	P34 output pin

P33 pin

Register	PCR3	
Bit Name	**PCR33**	**Pin Function**
Setting Value	0	P33 input pin
	1	P33 output pin

P32 pin

Register	PCR3	
Bit Name	**PCR32**	**Pin Function**
Setting Value	0	P32 input pin
	1	P32 output pin

P31 pin

Register	PCR3	
Bit Name	**PCR31**	**Pin Function**
Setting Value	0	P31 input pin
	1	P31 output pin

pin, and wakeup interrupt input pin. Each pin of the port 5 is shown in figure 9.4. The reg
setting of the I^{2}C bus interface register has priority for functions of the pins P57/SCL and
P56/SDA. Since the output buffer for pins P56 and P57 has the NMOS push-pull structur
differs from an output buffer with the CMOS structure in the high-level output characteri
section 23, Electrical Characteristics).

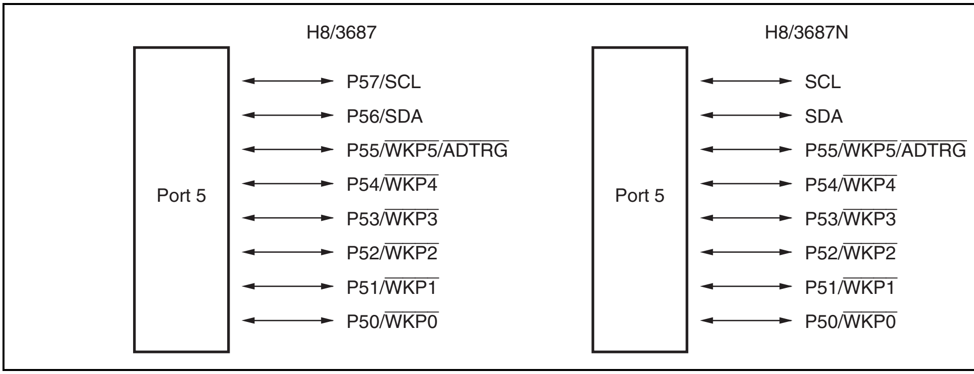


Figure 9.4 Port 5 Pin Configuration

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	This bit selects the function of pin P54/$\overline{\text{WKP4}}$. 0: General I/O port 1: $\overline{\text{WKP4}}$ input pin
3	WKP3	0	R/W	This bit selects the function of pin P53/$\overline{\text{WKP3}}$. 0: General I/O port 1: $\overline{\text{WKP3}}$ input pin
2	WKP2	0	R/W	This bit selects the function of pin P52/$\overline{\text{WKP2}}$. 0: General I/O port 1: $\overline{\text{WKP2}}$ input pin
1	WKP1	0	R/W	This bit selects the function of pin P51/$\overline{\text{WKP1}}$. 0: General I/O port 1: $\overline{\text{WKP1}}$ input pin
0	WKP0	0	R/W	This bit selects the function of pin P50/$\overline{\text{WKP0}}$. 0: General I/O port 1: $\overline{\text{WKP0}}$ input pin

Bit	Bit Name	Initial Value	R/W	Description
3	PCR53	0	W	in the H8/3687N.
2	PCR52	0	W	
1	PCR51	0	W	
0	PCR50	0	W	

9.4.3 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, the stored in PDR5 are read. If PDR5 is read while F are cleared to 0, the pin states are read regardle value stored in PDR5.
5	P55	0	R/W	
4	P54	0	R/W	
3	P53	0	R/W	Note: The P57 and P56 bits should not be set to H8/3687N.
2	P52	0	R/W	
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	

9.4.5 Pin Functions

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

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.

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.

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/$\overline{\text{WKP1}}$ 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.

P50/$\overline{\text{WKP0}}$ pin

Register	PMR5	PCR5	
Bit Name	**WKP0**	**PCR50**	**Pin Function**
Setting Value	0	0	P50 input pin
		1	P50 output pin
	1	X	$\overline{\text{WKP0}}$ input pin

Legend: X: Don't care.

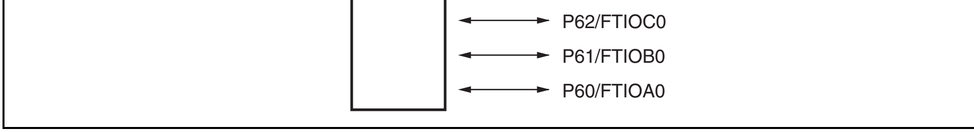


Figure 9.5 Port 6 Pin Configuration

Port 6 has the following registers.

- Port control register 6 (PCR6)
- Port data register 6 (PDR6)

9.5.1 Port Control Register 6 (PCR6)

PCR6 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 6.

Bit	Bit Name	Initial Value	R/W	Description
7	PCR67	0	W	When each of the port 6 pins P67 to P60 functio general I/O port, setting a PCR6 bit to 1 makes t corresponding pin an output port, while clearing 0 makes the pin an input port.
6	PCR66	0	W	
5	PCR65	0	W	
4	PCR64	0	W	
3	PCR63	0	W	
2	PCR62	0	W	
1	PCR61	0	W	
0	PCR60	0	W	

3	P63	0	R/W
2	P62	0	R/W
1	P61	0	R/W
0	P60	0	R/W

9.5.3 Pin Functions

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

P67/FTIOD1 pin

Register	TOER	TFCR	TPMR	TIORC1	PCR6	
Bit Name	**ED1**	**CMD1 and CMD0**	**PWMD1**	**IOD2 to IOD0**	**PCR67**	**Pin Function**
Setting Value	1	00	0	000 or 1XX	0	P67 input/FTIOD1 in
					1	P67 output pin
	0	00	0	001 or 01X	X	FTIOD1 output pin
			1	XXX		
		Other than 00	X	XXX		

Legend: X: Don't care.

		Other than 00	X	XXX		

Legend: X: Don't care.

P65/FTIOB1 pin

Register	TOER	TFCR	TPMR	TIORA1	PCR6	
Bit Name	**EB1**	**CMD1 to CMD0**	**PWMB1**	**IOB2 to IOB0**	**PCR65**	**Pin Function**
Setting Value	1	00	0	000 or 1XX	0	P65 input/FTIOB1 inp
					1	P65 output pin
	0	00	0	001 or 01X	X	FTIOB1 output pin
			1	XXX		
		Other than 00	X	XXX		

Legend: X: Don't care.

P63/FTIOD0 pin

Register	TOER	TFCR	TPMR	TIORC0	PCR6	
Bit Name	**ED0**	**CMD1 to CMD0**	**PWMD0**	**IOD2 to IOD0**	**PCR63**	**Pin Function**
Setting Value	1	00	0	000 or 1XX	0	P63 input/FTIOD0 in
					1	P63 output pin
	0	00	0	001 or 01X	X	FTIOD0 output pin
			1	XXX		
		Other than 00	X	XXX		

Legend: X: Don't care.

			1	XXX		
		Other than 00	X	XXX		

Legend: X: Don't care.

P61/FTIOB0 pin

Register	TOER	TFCR	TPMR	TIORA0	PCR6	
Bit Name	**EB0**	**CMD1 to CMD0**	**PWMB0**	**IOB2 to IOB0**	**PCR61**	**Pin Function**
Setting Value	1	00	0	000 or 1XX	0	P61 input/FTIOB0 inp
					1	P61 output pin
	0	00	0	001 or 01X	X	FTIOB0 output pin
			1	XXX		
		Other than 00	X	XXX		

Legend: X: Don't care.

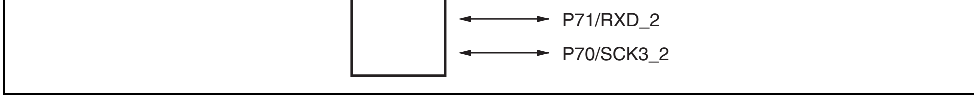


Figure 9.6 Port 7 Pin Configuration

Port 7 has the following registers.

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

9.6.1 Port Control Register 7 (PCR7)

PCR7 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 7.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	When each of the port 7 pins P76 to P74 and P7 functions as a general I/O port, setting a PCR7 b makes the corresponding pin an output port, whi clearing the bit to 0 makes the pin an input port. Bits 7 and 3 are reserved bits.
6	PCR76	0	W	
5	PCR75	0	W	
4	PCR74	0	W	
3	—	—	—	
2	PCR72	0	W	
1	PCR71	0	W	
0	PCR70	0	W	

3	—	1	—	Bits 7 and 3 are reserved bits. These bits are a as 1.
2	P72	0	R/W	
1	P71	0	R/W	
0	P70	0	R/W	

9.6.3 Pin Functions

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

P76/TMOV pin

Register	TCSRV	PCR7	
Bit Name	**OS3 to OS0**	**PCR76**	**Pin Function**
Setting Value	0000	0	P76 input pin
		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

Bit Name	TXD2	PCR72	Pin Function
Setting Value	0	0	P72 input pin
		1	P72 output pin
	1	X	TXD_2 output pin

Legend: X: Don't care.

P71/RXD_2 pin

Register	SCR3_2	PCR7	
Bit Name	**RE**	**PCR71**	**Pin Function**
Setting Value	0	0	P71 input pin
		1	P71 output pin
	1	X	RXD_2 input pin

Legend: X: Don't care.

P70/SCK3_2 pin

Register	SCR3_2		SMR2	PCR7	
Bit Name	**CKE1**	**CKE0**	**COM**	**PCR70**	**Pin Function**
Setting Value	0	0	0	0	P70 input pin
				1	P70 output pin
	0	0	1	X	SCK3_2 output pin
	0	1	X	X	SCK3_2 output pin
	1	X	X	X	SCK3_2 input pin

Legend: X: Don't care.

Port 8 has the following registers.

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

9.7.1 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 P85 functi
6	PCR86	0	W	general I/O port, setting a PCR8 bit to 1 makes
5	PCR85	0	W	corresponding pin an output port, while clearing 0 makes the pin an input port.
4 to 0	—	—	—	Reserved

9.7.2 Port Data Register 8 (PDR8)

PDR8 is a general I/O port data register of port 8.

Bit	Bit Name	Initial Value	R/W	Description
7	P87	0	R/W	PDR8 stores output data for port 8 pins.
6	P86	0	R/W	If PDR8 is read while PCR8 bits are set to 1, th
5	P85	0	R/W	stored in PDR8 is read. If PDR8 is read while P are cleared to 0, the pin states are read regardl value stored in PDR8.
4 to 0	—	All 1	—	Reserved These bits are always read as 1.

P86 pin

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

P85 pin

Register	PCR8	
Bit Name	**PCR85**	**Pin Function**
Setting Value	0	P85 input pin
	1	P85 output pin

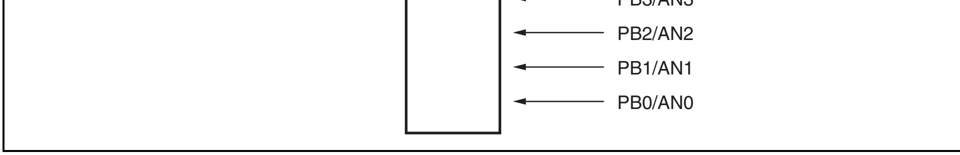


Figure 9.8 Port B Pin Configuration

Port B has the following register.

- Port data register B (PDRB)

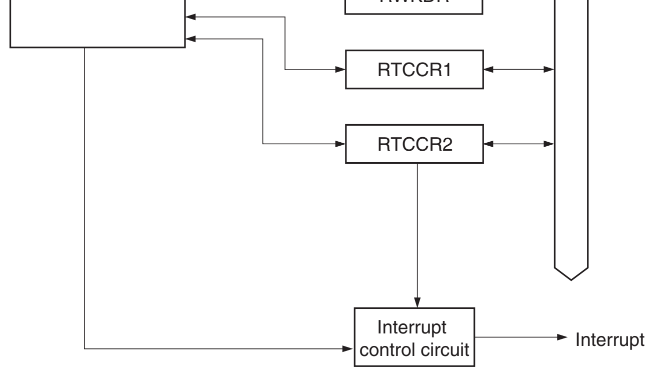
9.8.1 Port Data Register B (PDRB)

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 t register. However, if a port B pin is designated as an ana 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	

- Readable/writable counter of seconds, minutes, hours, and day-of-week with BCD c
- Periodic (seconds, minutes, hours, days, and weeks) interrupts
- 8-bit free running counter
- Selection of clock source

RTC3000A_000120030300



[Legend]

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

Figure 10.1 Block Diagram of RTC

10.2 Input/Output Pin

Table 10.1 shows the RTC input/output pin.

Table 10.1 Pin Configuration

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

- Clock source select register (RTCCSR)

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

RSECDR counts the BCD-coded second value. The setting range is decimal 00 to 59. It read register used as a counter, when it operates as a free running counter. For more info on reading seconds, minutes, hours, and day-of-week, see section 10.4.3, Data Reading

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

				hour, and day-of-week data registers must be ac
6	MN12	—	R/W	Counting ten's position of minutes
5	MN11	—	R/W	Counts on 0 to 5 for 60-minute counting.
4	MN10	—	R/W	
3	MN03	—	R/W	Counting one's position of minutes
2	MN02	—	R/W	Counts on 0 to 9 once per minute. When a carry generated, 1 is added to the ten's position.
1	MN01	—	R/W	
0	MN00	—	R/W	

				registers. When this bit is 0, the values of secor hour, and day-of-week data registers must be a
6	—	0	—	Reserved This bit is always read as 0.
5	HR11	—	R/W	Counting ten's position of hours
4	HR10	—	R/W	Counts on 0 to 2 for ten's position of hours.
3	HR03	—	R/W	Counting one's position of hours
2	HR02	—	R/W	Counts on 0 to 9 once per hour. When a carry generated, 1 is added to the ten's position.
1	HR01	—	R/W	
0	HR00	—	R/W	

				hour, and day-of-week data registers must be ac
6 to 3	—	All 0	—	Reserved These bits are always read as 0.
2	WK2	—	R/W	Day-of-week counting
1	WK1	—	R/W	Day-of-week is indicated with a binary code
0	WK0	—	R/W	000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: Reserved (setting prohibited)

6	12/24	—	R/W	Operating mode 0: RTC operates in 12-hour mode. RHRDR co to 11. 1: RTC operates in 24-hour mode. RHRDR co to 23.
5	PM	—	R/W	A.m./p.m. 0: Indicates a.m. when RTC is in the 12-hour m 1: Indicates p.m. when RTC is in the 12-hour m
4	RST	0	R/W	Reset 0: Normal operation 1: Resets registers and control circuits except F and this bit. Clear this bit to 0 after having be 1.
3 to 0	—	All 0	—	Reserved These bits are always read as 0.

Noon (between 11 and 12)

24-hour count	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
12-hour count	0	1	2	3	4	5	6	7	8	9	10	11	0	1	2	3	4	5
PM	0 (Morning)												1 (Afternoon)					

24-hour count	18	19	20	21	22	23	0
12-hour count	6	7	8	9	10	11	0
PM	1 (Afternoon)						0

Figure 10.2 Definition of Time Expression

				These bits are always read as 0.
5	FOIE	—	R/W	Free Running Counter Overflow Interrupt Enable 0: Disables an overflow interrupt 1: Enables an overflow interrupt
4	WKIE	—	R/W	Week Periodic Interrupt Enable 0: Disables a week periodic interrupt 1: Enables a week periodic interrupt
3	DYIE	—	R/W	Day Periodic Interrupt Enable 0: Disables a day periodic interrupt 1: Enables a day periodic interrupt
2	HRIE	—	R/W	Hour Periodic Interrupt Enable 0: Disables an hour periodic interrupt 1: Enables an hour periodic interrupt
1	MNIE	—	R/W	Minute Periodic Interrupt Enable 0: Disables a minute periodic interrupt 1: Enables a minute periodic interrupt
0	SEIE	—	R/W	Second Periodic Interrupt Enable 0: Disables a second periodic interrupt 1: Enables a second periodic interrupt

7	—	0	—	Reserved This bit is always read as 0.
6	RCS6	0	R/W	Clock output selection
5	RCS5	0	R/W	Selects a clock output from the TMOW pin whe TMOW in PMR1 to 1. 00: φ/4 01: φ/8 10: φ/16 11: φ/32
4	—	0	—	Reserved This bit is always read as 0.
3	RCS3	1	R/W	Clock source selection
2	RCS2	0	R/W	0000: φ/8·················· Free running counter opera
1	RCS1	0	R/W	0001: φ/32················ Free running counter opera
0	RCS0	0	R/W	0010: φ/128·············· Free running counter opera 0011: φ/256·············· Free running counter opera 0100: φ/512·············· Free running counter opera 0101: φ/2048············ Free running counter opera 0110: φ/4096············ Free running counter opera 0111: φ/8192············ Free running counter opera 1XXX: 32.768 kHz·····RTC operation

Legend: X: Don't care.

Figure 10.3 shows the procedure for the initial setting of the RTC. To set the RTC again, follow this procedure.

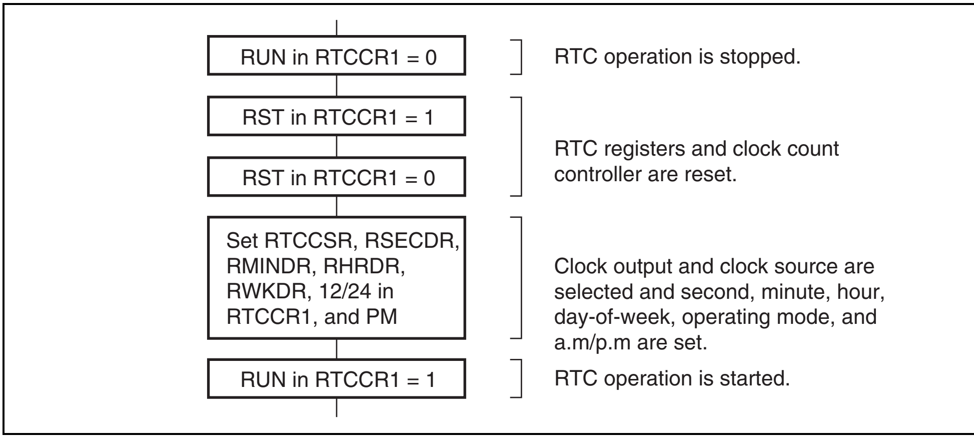


Figure 10.3 Initial Setting Procedure

bit is set to 1, the registers are updated, and the BSY bit is cleared to 0.

2. Making use of interrupts, read from the second, minute, hour, and day-of week regis the IRRTA flag in IRR1 is set to 1 and the BSY bit is confirmed to be 0.
3. Read from the second, minute, hour, and day-of week registers twice in a row, and if no change in the read data, the read data is used.

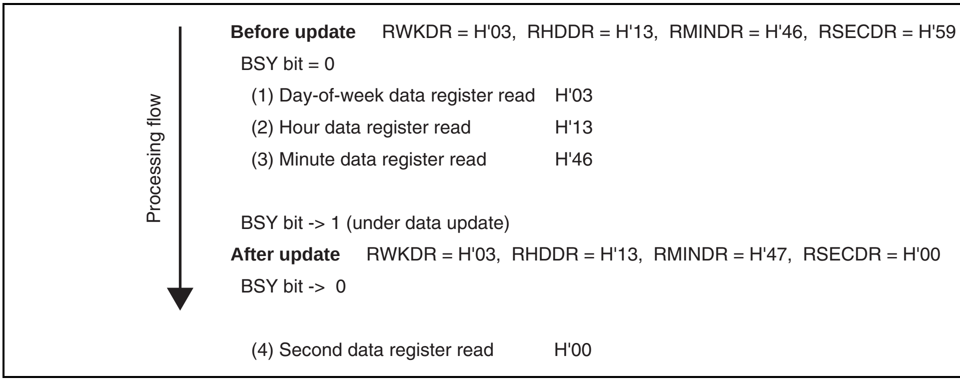
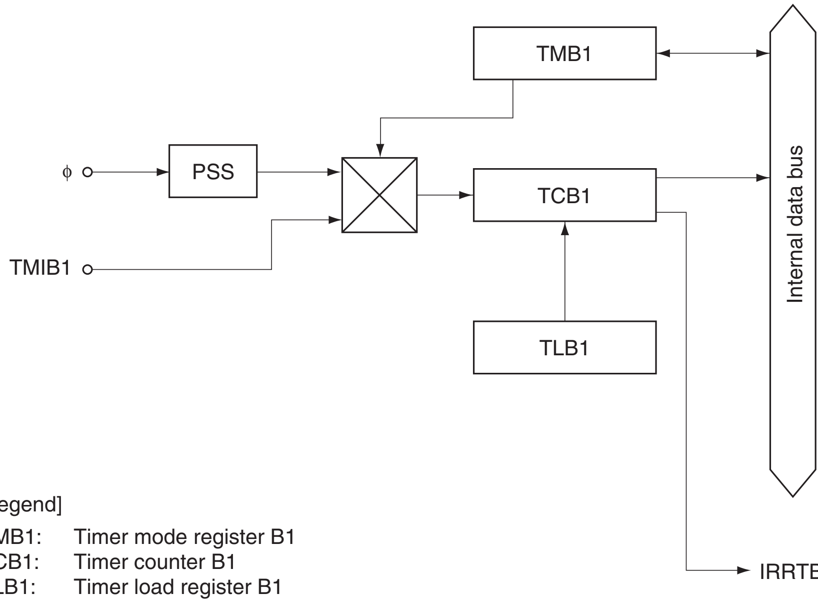


Figure 10.4 Example: Reading of Inaccurate Time Data

Table 10.2 Interrupt Source

Interrupt Name	Interrupt Source	Interrupt En
Overflow interrupt	Occurs when the free running counter is overflown.	FOIE
Week periodic interrupt	Occurs every week when the day-of-week date register value becomes 0.	WKIE
Day periodic interrupt	Occurs every day when the day-of-week date register is counted.	DYIE
Hour periodic interrupt	Occurs every hour when the hour date register is counted.	HRIE
Minute periodic interrupt	Occurs every minute when the minute date register is counted.	MNIE
Second periodic interrupt	Occurs every second when the second date register is counted.	SCIE



[Legend]

TMB1: Timer mode register B1
TCB1: Timer counter B1
TLB1: Timer load register B1
IRRTB1: Timer B1 interrupt request flag
PSS: Prescaler S
TMIB1: Timer B1 event input

Figure 11.1 Block Diagram of Timer B1

TIM08B0A_000020020200

The timer B1 has the following registers.

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

				These bits are always read as 1.
2	TMB12	0	R/W	Clock select
1	TMB11	0	R/W	000: Internal clock: ϕ/8192
0	TMB10	0	R/W	001: Internal clock: ϕ/2048
				010: Internal clock: ϕ/512
				011: Internal clock: ϕ/256
				100: Internal clock: ϕ/64
				101: Internal clock: ϕ/16
				110: Internal clock: ϕ/4
				111: External event (TMIB1): rising or falling ed
				Note: * The edge of the external event signal by bit IEG1 in the interrupt edge selec (IEGR1). See section 3.2.1, Interrupt E Select Register 1 (IEGR1), for details. setting TMB12 to TMB10 to 1, IRQ1 in mode register 1 (PMR1) should be set

11.3.2 Timer Counter B1 (TCB1)

TCB1 is an 8-bit read-only up-counter, which is incremented by internal clock input. Th source for input to this counter is selected by bits TMB12 to TMB10 in TMB1. TCB1 v be read by the CPU at any time. When TCB1 overflows from H'FF to H'00 or to the valu TLB1, the IRRTB1 flag in IRR2 is set to 1. TCB1 is allocated to the same address as TL is initialized to H'00.

11.4.1 Interval Timer Operation

When bit TMB17 in TMB1 is cleared to 0, timer B1 functions as an 8-bit interval timer. U reset, TCB1 is cleared to H'00 and bit TMB17 is cleared to 0, so up-counting and interval resume immediately. The operating clock of timer B1 is selected from seven internal clo output by prescaler S, or an external clock input at pin TMB1. The selection is made by TMB12 to TMB10 in TMB1.

After the count value in TMB1 reaches H'FF, the next clock signal input causes timer B1 overflow, setting flag IRRTB1 in IRR2 to 1. If IENTB1 in IENR2 is 1, an interrupt is req the CPU.

At overflow, TCB1 returns to H'00 and starts counting up again. During interval timer op (TMB17 = 0), when a value is set in TLB1, the same value is set in TCB1.

11.4.2 Auto-Reload Timer Operation

Setting bit TMB17 in TMB1 to 1 causes timer B1 to function as an 8-bit auto-reload time a reload value is set in TLB1, the same value is loaded into TCB1, becoming the value fr TCB1 starts its count. After the count value in TCB1 reaches H'FF, the next clock signal causes timer B1 to overflow. The TLB1 value is then loaded into TCB1, and the count co from that value. The overflow period can be set within a range from 1 to 256 input clock depending on the TLB1 value.

The clock sources and interrupts in auto-reload mode are the same as in interval mode. In reload mode (TMB17 = 1), when a new value is set in TLB1, the TLB1 value is also load TCB1.

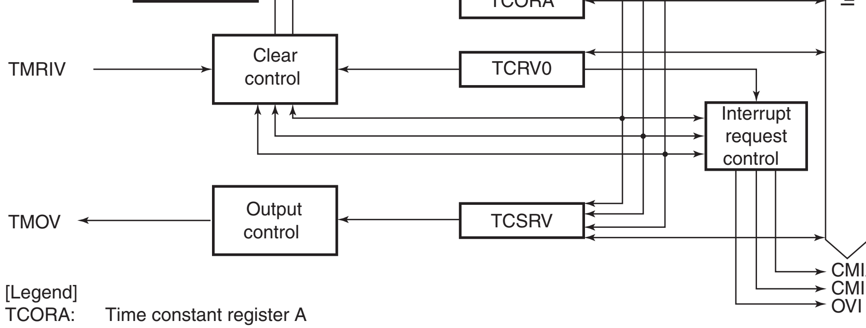
Table 11.2 shows the timer B1 operating modes.

Table 11.2 Timer B1 Operating Modes

Operating Mode		Reset	Active	Sleep	Subactive	Subsleep	S
TCB1	Interval	Reset	Functions	Functions	Halted	Halted	H
	Auto-reload	Reset	Functions	Functions	Halted	Halted	H
TMB1		Reset	Functions	Retained	Retained	Retained	R

- 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.

TIM08V0A_000120030300



[Legend]

TCORA: Time constant register A
TCORB: Time constant register B
TCNTV: Timer counter V
TCSRV: Timer control/status register V
TCRV0: Timer control register V0
TCRV1: Timer control register V1
PSS: Prescaler S
CMIA: Compare-match interrupt A
CMIB: Compare-match interrupt B
OVI: Overflow interupt

Figure 12.1 Block Diagram of Timer V

12.3 Register Descriptions

Time V has the following registers.

- Timer counter V (TCNTV)
- Timer constant register A (TCORA)
- Timer constant register B (TCORB)
- Timer control register V0 (TCRV0)
- Timer control/status register V (TCSRV)
- Timer control register V1 (TCRV1)

12.3.1 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.

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

TCORA and TCORB have the same function.

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

TCRV0 selects the input clock signals of TCNTV, specifies the clearing conditions of TC
and controls each interrupt request.

Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	Compare Match Interrupt Enable B When this bit is set to 1, interrupt request from th bit in TCSRV is enabled.
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 counting condition in combination with ICKS0 in
0	CKS0	0	R/W	Refer to table 12.2.

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

				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 2	OS3 OS2	0 0	R/W R/W	Output Select 3 and 2 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 0	OS1 OS0	0 0	R/W R/W	Output Select 1 and 0 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 3	TVEG1 TVEG0	0 0	R/W R/W	TRGV Input Edge Select 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 12.2.

will be set. The timing at this time is shown in figure 12.4. 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 or (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 12.6 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 12.7 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 12.8 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.

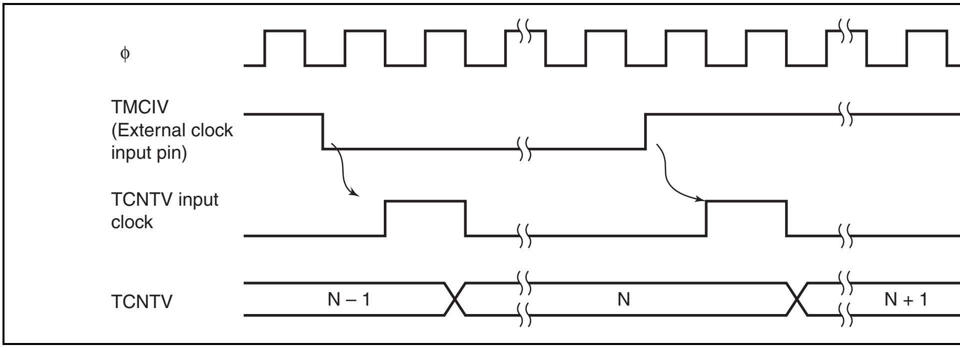


Figure 12.3 Increment Timing with External Clock

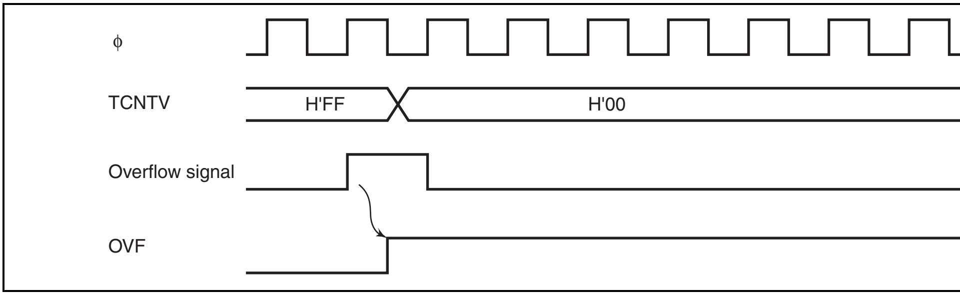


Figure 12.4 OVF Set Timing

Figure 12.5 CMFA and CMFB Set Timing

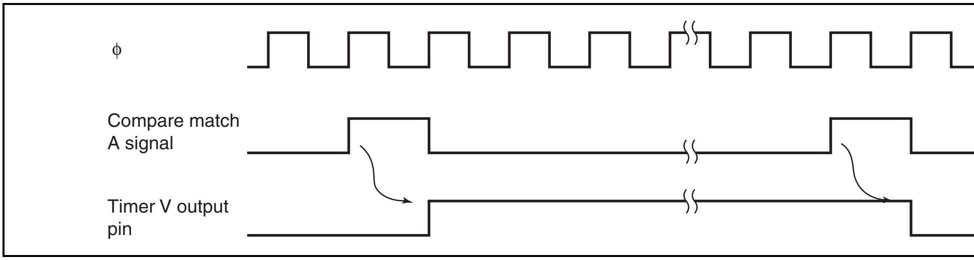


Figure 12.6 TMOV Output Timing

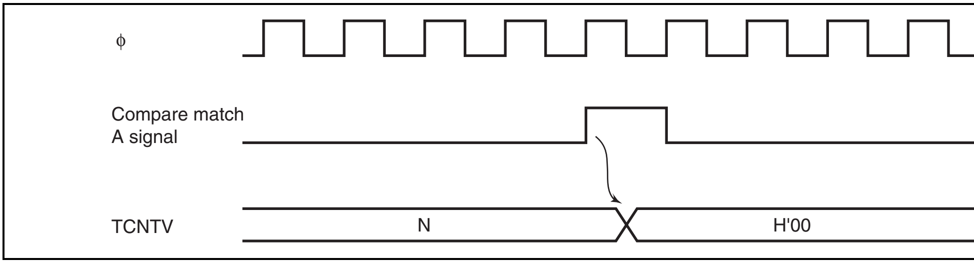


Figure 12.7 Clear Timing by Compare Match

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.

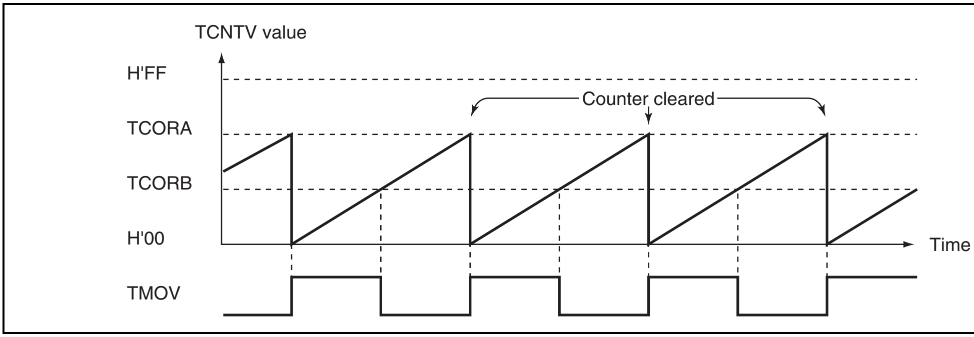


Figure 12.9 Pulse Output Example

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 interve with a delay determined by TCORA from the TRGV input, and a pulse width determ (TCORB – TCORA).

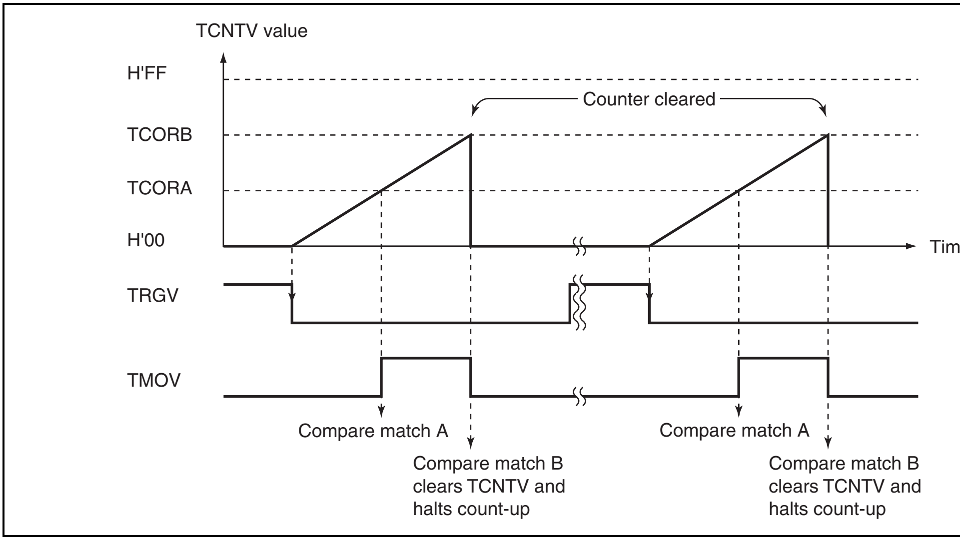


Figure 12.10 Example of Pulse Output Synchronized to TRGV Input

12.12 shows the timing.

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 12.3 the switch is from a high clock signal to a low clock signal, the switche seen as a falling edge, causing TCNTV to increment. TCNTV can also be incremente switch between internal and external clocks.

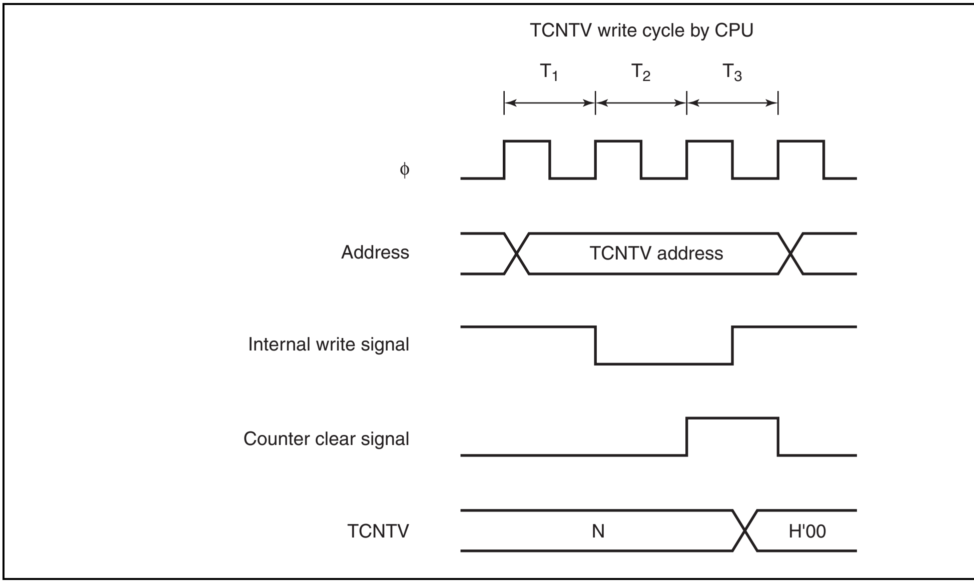


Figure 12.11 Contention between TCNTV Write and Clear

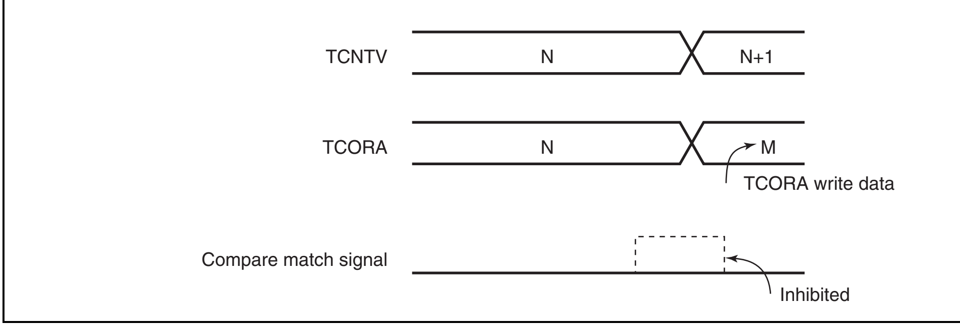


Figure 12.12 Contention between TCORA Write and Compare Match

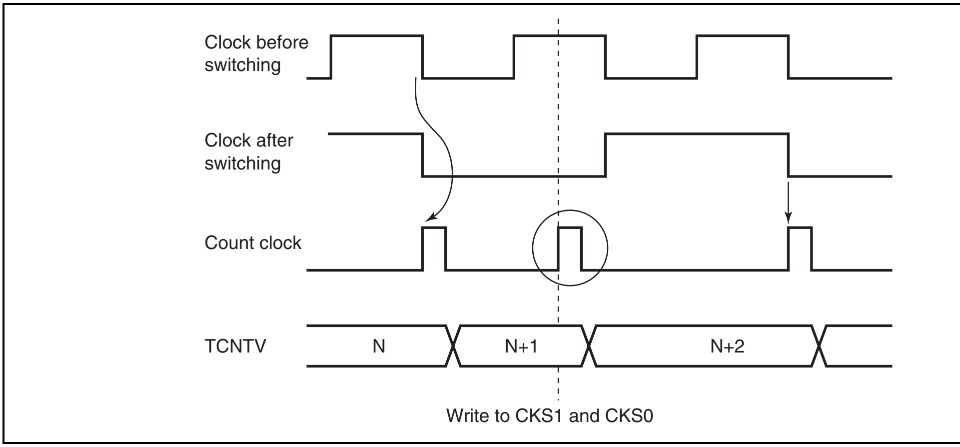
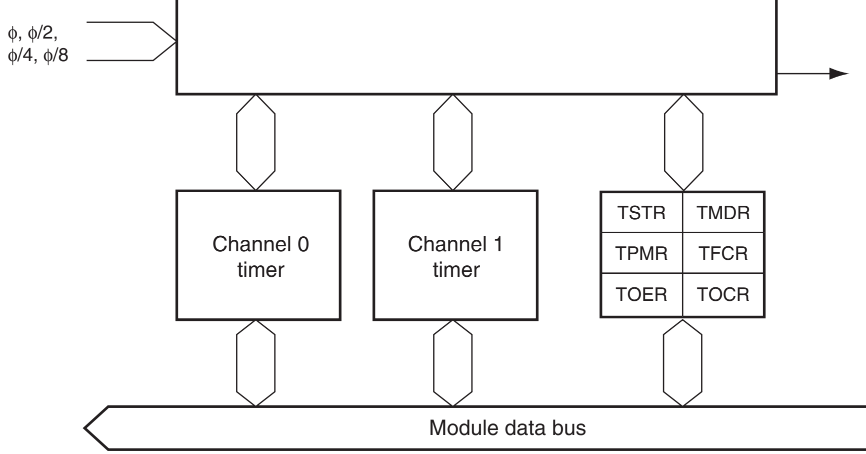


Figure 12.13 Internal Clock Switching and TCNTV Operation

— Independently assignable output compare or input capture functions

- Selection of five counter clock sources: four internal clocks (ϕ, ϕ/2, ϕ/4, and ϕ/8) and
  external clock
- Seven selectable operating modes
  - Output compare function
    Selection of 0 output, 1 output, or toggle output
  - Input capture function
    Rising edge, falling edge, or both edges
  - Synchronous operation
    Timer counters_0 and _1 (TCNT_0 and TCNT_1) can be written simultaneously
    Simultaneous clearing by compare match or input capture is possible.
  - PWM mode
    Up to six-phase PWM output can be provided with desired duty ratio.
  - Reset synchronous PWM mode
    Three-phase PWM output for normal and counter phases
  - Complementary PWM mode
    Three-phase PWM output for non-overlapped normal and counter phases
    The A/D conversion start trigger can be set for PWM cycles.
  - Buffer operation
    The input capture register can be consisted of double buffers.
    The output compare register can automatically be modified.
- High-speed access by the internal 16-bit bus
  - 16-bit TCNT and GR registers can be accessed in high speed by a 16-bit bus inte
- Any initial timer output value can be set
- Output of the timer is disabled by external trigger

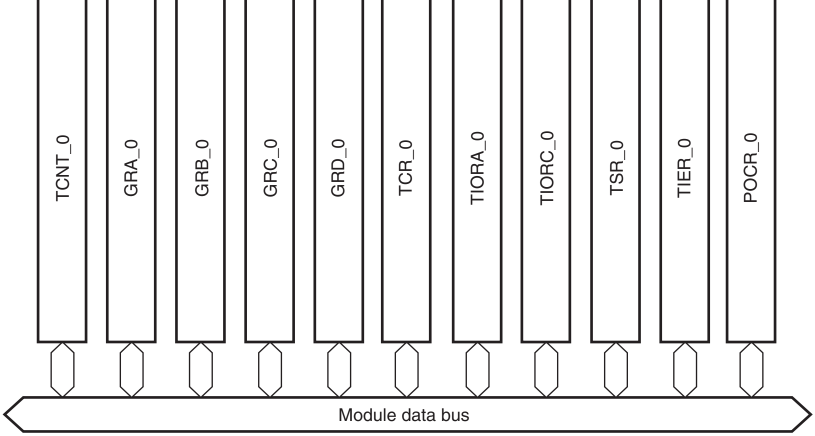
Buffer register		GRC_0, GRD_0	GRC_1, GRD_1
I/O pins		FTIOA0, FTIOB0, FTIOC0, FTIOD0	FTIOA1, FTIOB1, FTIOC FTIOD1
Counter clearing function		Compare match/input capture of GRA_0, GRB_0, GRC_0, or GRD_0	Compare match/input cap GRA_1, GRB_1, GRC_1, GRD_1
Compare match output	0 output	Yes	Yes
	1 output	Yes	Yes
	output	Yes	Yes
Input capture function		Yes	Yes
Synchronous operation		Yes	Yes
PWM mode		Yes	Yes
Reset synchronous PWM mode		Yes	Yes
Complementary PWM mode		Yes	Yes
Buffer function		Yes	Yes
Interrupt sources		Compare match/input capture A0 to D0 Overflow	Compare match/input cap to D1 Overflow Underflow



[Legend]

TSTR : Timer start register (8 bits)
TMDR : Timer mode register (8 bits)
TPMR : Timer PWM mode register (8 bits)
TFCR : Timer function control register (8 bits)
TOER : Timer output master enable register (8 bits)
TOCR : Timer output control register (8 bits)
$\overline{\text{ADTRG}}$: A/D conversion start trigger output signal
ITMZ0 : Channel 0 interrupt
ITMZ1 : Channel 1 interrupt

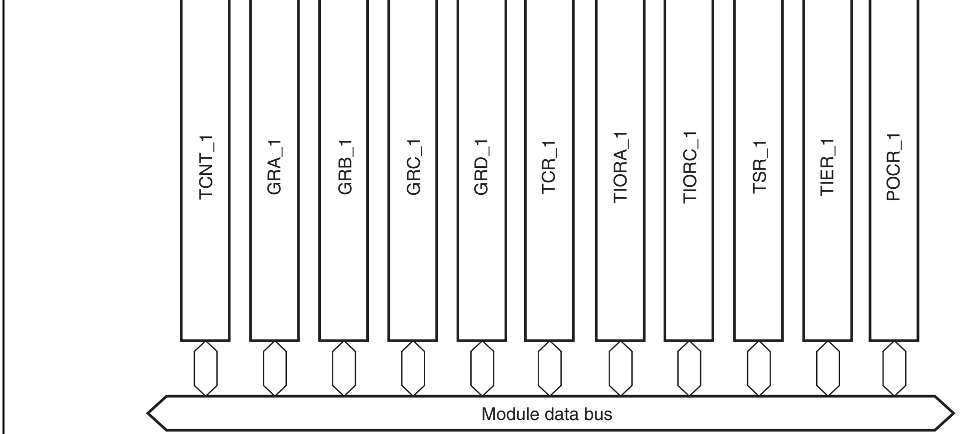
Figure 13.1 Timer Z Block Diagram



[Legend]

TCNT_0 :	Timer counter_0 (16 bits)
GRA_0, GRB_0: GRC_0, GRD_0 :	General registers A_0, B_0, C_0, and D_0 (input capture/output compare registers: 16 bits × 4)
TCR_0 :	Timer control register_0 (8 bits)
TIORA_0 :	Timer I/O control register A_0 (8 bits)
TIORC_0 :	Timer I/O control register C_0 (8 bits)
TSR_0 :	Timer status register_0 (8 bits)
TIER_0 :	Timer interrupt enable register_0 (8 bits)
POCR_0 :	PWM mode output level control register_0 (8 bits)
ITMZ0 :	Channel 0 interrupt

Figure 13.2 Timer Z (Channel 0) Block Diagram



[Legend]

TCNT_1 :	Timer counter_1 (16 bits)
GRA_1, GRB_1: GRC_1, GRD_1 :	General registers A_1, B_1, C_1, and D_1 (input capture/output compare registers: 16 bits × 4)
TCR_1 :	Timer control register_1 (8 bits)
TIORA_1 :	Timer I/O control register A_1 (8 bits)
TIORC_1 :	Timer I/O control register C_1 (8 bits)
TSR_1 :	Timer status register_1 (8 bits)
TIER_1 :	Timer interrupt enable register_1 (8 bits)
POCR_1 :	PWM mode output level control register_1 (8 bits)
ITMZ1 :	Channel 1 interrupt

Figure 13.3 Timer Z (Channel 1) Block Diagram

compare B0			input capture input, or PWM ou
Input capture/output compare C0	FTIOC0	Input/output	GRC_0 output compare output input capture input, or PWM synchronous output (in reset synchronous PWM and comple PWM modes)
Input capture/output compare D0	FTIOD0	Input/output	GRD_0 output compare output input capture input, or PWM ou
Input capture/output compare A1	FTIOA1	Input/output	GRA_1 output compare output input capture input, or PWM ou reset synchronous PWM and complementary PWM modes)
Input capture/output compare B1	FTIOB1	Input/output	GRB_1 output compare output input capture input, or PWM ou
Input capture/output compare C1	FTIOC1	Input/output	GRC_1 output compare output input capture input, or PWM ou
Input capture/output compare D1	FTIOD1	Input/output	GRD_1 output compare output input capture input, or PWM ou

- Timer output control register (TOCR)

Channel 0

- Timer control register_0 (TCR_0)
- Timer I/O control register A_0 (TIORA_0)
- Timer I/O control register C_0 (TIORC_0)
- Timer status register_0 (TSR_0)
- Timer interrupt enable register_0 (TIER_0)
- PWM mode output level control register_0 (POCR_0)
- Timer counter_0 (TCNT_0)
- General register A_0 (GRA_0)
- General register B_0 (GRB_0)
- General register C_0 (GRC_0)
- General register D_0 (GRD_0)

Channel 1

- Timer control register_1 (TCR_1)
- Timer I/O control register A_1 (TIORA_1)
- Timer I/O control register C_1 (TIORC_1)
- Timer status register_1 (TSR_1)
- Timer interrupt enable register_1 (TIER_1)
- PWM mode output level control register_1 (POCR_1)
- Timer counter_1 (TCNT_1)
- General register A_1 (GRA_1)
- General register B_1 (GRB_1)

				These bits are always read as 1, and cannot be
1	STR1	0	R/W	Channel 1 Counter Start 0: TCNT_1 halts counting 1: TCNT_1 starts counting
0	STR0	0	R/W	Channel 0 Counter Start 0: TCNT_0 halts counting 1: TCNT_0 starts counting

6	BFC1	0	R/W	Buffer Operation C1 0: GRC_1 operates normally 1: GRA_1 and GRD_1 are used together for bu operation
5	BFD0	0	R/W	Buffer Operation D0 0: GRD_0 operates normally 1: GRB_0 and GRD_0 are used together for bu operation
4	BFC0	0	R/W	Buffer Operation C0 0: GRC_0 operates normally 1: GRA_0 and GRC_0 are used together for bu operation
3 to 1	—	All 1	—	Reserved These bits are always read as 1, and cannot be
0	SYNC	0	R/W	Timer Synchronization 0: TCNT_1 and TCNT_0 operate as a different 1: TCNT_1 and TCNT_0 are synchronized TCNT_1 and TCNT_0 can be pre-set or clea synchronously

				1: FTIOD1 operates in PWM mode
5	PWMC1	0	R/W	PWM Mode C1 0: FTIOC1 operates normally 1: FTIOC1 operates in PWM mode
4	PWMB1	0	R/W	PWM Mode B1 0: FTIOB1 operates normally 1: FTIOB1 operates in PWM mode
3	—	1	—	Reserved This bit is always read as 1, and cannot be modi
2	PWMD0	0	R/W	PWM Mode D0 0: FTIOD0 operates normally 1: FTIOD0 operates in PWM mode
1	PWMC0	0	R/W	PWM Mode C0 0: FTIOC0 operates normally 1: FTIOC0 operates in PWM mode
0	PWMB0	0	R/W	PWM Mode B0 0: FTIOB0 operates normally 1: FTIOB0 operates in PWM mode

				1: External clock input is enabled
5	ADEG	0	R/W	A/D Trigger Edge Select A/D module should be set to start an A/D conve the external trigger 0: A/D trigger at the crest in complementary PW 1: A/D trigger at the trough in complementary P
4	ADTRG	0	R/W	External Trigger Disable 0: A/D trigger for PWM cycles is disabled in complementary PWM mode 1: A/D trigger for PWM cycles is enabled in complementary PWM mode
3	OLS1	0	R/W	Output Level Select 1 Selects the counter-phase output levels in rese synchronous PWM mode or complementary PW 0: Initial output is high and the active level is lo 1: Initial output is low and the active level is hig
2	OLS0	0	R/W	Output Level Select 0 Selects the normal-phase output levels in reset synchronous PWM mode or complementary PW 0: Initial output is high and the active level is lo 1: Initial output is low and the active level is hig Figure 13.4 shows an example of outputs in res synchronous PWM mode and complementary P mode when OLS1 = 0 and OLS0 = 0.

the crest)

Note: When reset synchronous PWM mode or complementary PWM mode is selected by bits, this setting has the priority to the setti PWM mode by each bit in TPMR. Stop TC and TCNT_1 before making settings for re synchronous PWM mode or complementa mode.

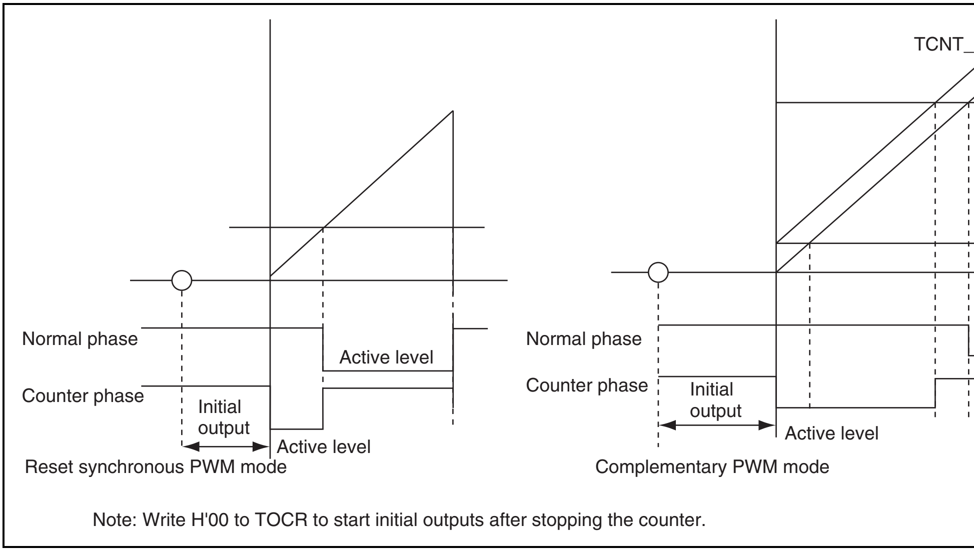


Figure 13.4 Example of Outputs in Reset Synchronous PWM Mode and Complementary PWM Mode

				1: FTIOD1 pin output is disabled regardless of t TFCR, and TIORC_1 settings (FTIOD1 pin is as an I/O port).
6	EC1	1	R/W	Master Enable C1 0: FTIOC1 pin output is enabled according to th TFCR, and TIORC_1 settings 1: FTIOC1 pin output is disabled regardless of t TFCR, and TIORC_1 settings (FTIOC1 pin is as an I/O port).
5	EB1	1	R/W	Master Enable B1 0: FTIOB1 pin output is enabled according to th TFCR, and TIORA_1 settings 1: FTIOB1 pin output is disabled regardless of t TFCR, and TIORA_1 settings (FTIOB1 pin is as an I/O port).
4	EA1	1	R/W	Master Enable A1 0: FTIOA1 pin output is enabled according to th TFCR, and TIORA_1 settings 1: FTIOA1 pin output is disabled regardless of t TFCR, and TIORA_1 settings (FTIOA1 pin is as an I/O port).
3	ED0	1	R/W	Master Enable D0 0: FTIOD0 pin output is enabled according to th TFCR, and TIORC_0 settings 1: FTIOD0 pin output is disabled regardless of t TFCR, and TIORC_0 settings (FTIOD0 pin is as an I/O port).

				1: FTIOB0 pin output is disabled regardless of th TFCR, and TIORA_0 settings (FTIOB0 pin is as an I/O port).
0	EA0	1	R/W	Master Enable A0 0: FTIOA0 pin output is enabled according to the TFCR, and TIORA_0 settings 1: FTIOA0 pin output is disabled regardless of th TFCR, and TIORA_0 settings (FTIOA0 pin is as an I/O port).

13.3.6 Timer Output Control Register (TOCR)

TOCR selects the initial outputs before the first occurrence of a compare match. Note tha OLS1 and OLS0 in TFCR set these initial outputs in reset synchronous PWM mode and complementary PWM mode.

Bit	Bit Name	Initial Value	R/W	Description
7	TOD1	0	R/W	Output Level Select D1 0: 0 output at the FTIOD1 pin* 1: 1 output at the FTIOD1 pin*
6	TOC1	0	R/W	Output Level Select C1 0: 0 output at the FTIOC1 pin* 1: 1 output at the FTIOC1 pin*
5	TOB1	0	R/W	Output Level Select B1 0: 0 output at the FTIOB1 pin* 1: 1 output at the FTIOB1 pin*

				1: 1 output at the FTIOC0 pin*
1	TOB0	0	R/W	Output Level Select B0 0: 0 output at the FTIOB0 pin* 1: 1 output at the FTIOB0 pin*
0	TOA0	0	R/W	Output Level Select A0 0: 0 output at the FTIOA0 pin* 1: 1 output at the FTIOA0 pin*

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

13.3.7 Timer Counter (TCNT)

The timer Z has two TCNT counters (TCNT_0 and TCNT_1), one for each channel. The counters are 16-bit readable/writable registers that increment/decrement according to inp Input clocks can be selected by bits TPSC2 to TPSC0 in TCR. TCNT0 and TCNT 1 increment/decrement in complementary PWM mode, while they only increment in other

The TCNT counters are initialized to H'0000 by compare matches with corresponding G GRC, or GRD, or input captures to GRA, GRB, GRC, or GRD (counter clearing functio the TCNT counters overflow, an OVF flag in TSR for the corresponding channel is set t TCNT_1 underflows, an UDF flag in TSR is set to 1. The TCNT counters cannot be acc bit units; they must always be accessed as a 16-bit unit.

external signals. At this point, IMFA to IMFD flags in the corresponding TSR are set to 1. Detection edges for input capture signals can be selected by TIORA and TIORC.

When PWM mode, complementary PWM mode, or reset synchronous PWM mode is sele values in TIORA and TIORC are ignored. Upon reset, the GR registers are set as output c registers (no output) and initialized to H'FFFF. The GR registers cannot be accessed in 8- they must always be accessed as a 16-bit unit.

				capture
				010: Clears TCNT by GRB compare match/inpu capture*[1]
				011: Synchronization clear; Clears TCNT in syr with counter clearing of the other channel'
				100: Disables TCNT clearing
				101: Clears TCNT by GRC compare match/inpu capture*[1]
				110: Clears TCNT by GRD compare match/inpu capture*[1]
				111: Synchronization clear; Clears TCNT in syr with counter clearing of the other channel'
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	00: Count at rising edge
				01: Count at falling edge
				1X: Count at both edges
2	TPSC2	0	R/W	Time Prescaler 2 to 0
1	TPSC1	0	R/W	000: Internal clock: count by ϕ
0	TPSC0	0	R/W	001: Internal clock: count by $\phi/2$
				010: Internal clock: count by $\phi/4$
				011: Internal clock: count by $\phi/8$
				1XX: External clock: count by FTIOA0 (TCLK) p

Notes: 1. When GR functions as an output compare register, TCNT is cleared by comp
When GR functions as input capture, TCNT is cleared by input capture.

2. Synchronous operation is set by TMDR.

3. X: Don't care

Bit	Bit Name	Initial value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1.
6 5 4	IOB2 IOB1 IOB0	0 0 0	R/W R/W R/W	I/O Control B2 to B0 GRB is an output compare register: 000: Disables pin output by compare match 001: 0 output by GRB compare match 010: 1 output by GRB compare match 011: Toggle output by GRB compare match GRB is an input capture register: 100: Input capture to GRB at the rising edge 101: Input capture to GRB at the falling edge 11X: Input capture to GRB at both rising and falli
3	—	1	—	Reserved This bit is always read as 1.
2 1 0	IOA2 IOA1 IOA0	0 0 0	R/W R/W R/W	I/O Control A2 to A0 GRA is an output compare register: 000: Disables pin output by compare match 001: 0 output by GRA compare match 010: 1 output by GRA compare match 011: Toggle output by GRA compare match GRA is an input capture register: 100: Input capture to GRA at the rising edge 101: Input capture to GRA at the falling edge 11X: Input capture to GRA at both rising and falli

Legend: X: Don't care

5	IOD1	0	R/W	GRD is an output compare register:
4	IOD0	0	R/W	000: Disables pin output by compare match 001: 0 output by GRD compare match 010: 1 output by GRD compare match 011: Toggle output by GRD compare match GRD is an input capture register: 100: Input capture to GRD at the rising edge 101: Input capture to GRD at the falling edge 11X: Input capture to GRD at both rising and fa edges
3	—	1	—	Reserved This bit is always read as 1.
2	IOC2	0	R/W	I/O Control C2 to C0
1	IOC1	0	R/W	GRC is an output compare register:
0	IOC0	0	R/W	000: Disables pin output by compare match 001: 0 output by GRC compare match 010: 1 output by GRC compare match 011: Toggle Output by GRC compare match GRC is an input capture register: 100: Input capture to GRC at the rising edge 101: Input capture to GRC at the falling edge 11X: Input capture to GRC at both rising and fa edges

Legend: X: Don't care

5	UDF*	0	R/W	Underflow Flag [Setting condition] • When TCNT_1 underflows [Clearing condition] • When 0 is written to UDF after reading UDF
4	OVF	0	R/W	Overflow Flag [Setting condition] • When the TCNT value underflows [Clearing condition] • When 0 is written to OVF after reading OVF
3	IMFD	0	R/W	Input Capture/Compare Match Flag D [Setting conditions] • When TCNT = GRD and GRD is functioning compare register • When TCNT value is transferred to GRD by i capture signal and GRD is functioning as inp capture register [Clearing condition] • When 0 is written to IMFD after reading IMFD

				• When 0 is written to IMFC after reading IMF
1	IMFB	0	R/W	Input Capture/Compare Match Flag B [Setting conditions] • When TCNT = GRB and GRB is functioning compare register • When TCNT value is transferred to GRB by capture signal and GRB is functioning as in capture register [Clearing condition] • When 0 is written to IMFB after reading IMF
0	IMFA	0	R/W	Input Capture/Compare Match Flag A [Setting conditions] • When TCNT = GRA and GRA is functioning compare register • When TCNT value is transferred to GRA by capture signal and GRA is functioning as in capture register [Clearing condition] • When 0 is written to IMFA after reading IMF

Note: Bit 5 is not the UDF flag in TSR_0. It is a reserved bit. It is always read as 1.

				0: Interrupt requests (OVI) by OVF or UDF flag a disabled 1: Interrupt requests (OVI) by OVF or UDF flag a enabled
3	IMIED	0	R/W	Input Capture/Compare Match Interrupt Enable D 0: Interrupt requests (IMID) by IMFD flag are dis 1: Interrupt requests (IMID) by IMFD flag are ena
2	IMIEC	0	R/W	Input Capture/Compare Match Interrupt Enable C 0: Interrupt requests (IMIC) by IMFC flag are dis 1: Interrupt requests (IMIC) by IMFC flag are ena
1	IMIEB	0	R/W	Input Capture/Compare Match Interrupt Enable B 0: Interrupt requests (IMIB) by IMFB flag are dis 1: Interrupt requests (IMIB) by IMFB flag are ena
0	IMIEA	0	R/W	Input Capture/Compare Match Interrupt Enable A 0: Interrupt requests (IMIA) by IMFA flag are dis 1: Interrupt requests (IMIA) by IMFA flag are ena

				0: The output level of FTIOD is low-active 1: The output level of FTIOD is high-active
1	POLC	0	R/W	PWM Mode Output Level Control C 0: The output level of FTIOC is low-active 1: The output level of FTIOC is high-active
0	POLB	0	R/W	PWM Mode Output Level Control B 0: The output level of FTIOB is low-active 1: The output level of FTIOB is high-active

13.3.14 Interface with CPU

1. 16-bit register

   TCNT and GR are 16-bit registers. Reading/writing in a 16-bit unit is enabled but di an 8-bit unit since the data bus with the CPU is 16-bit width. These registers must al accessed in a 16-bit unit. Figure 13.5 shows an example of accessing the 16-bit regis

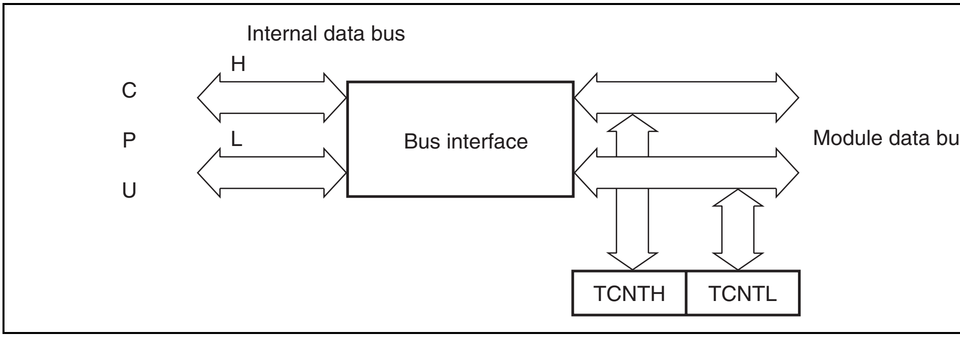


Figure 13.5 Accessing Operation of 16-Bit Register (between CPU and TCNT (

Figure 13.6 Accessing Operation of 8-Bit Register (between CPU and TSTR (8

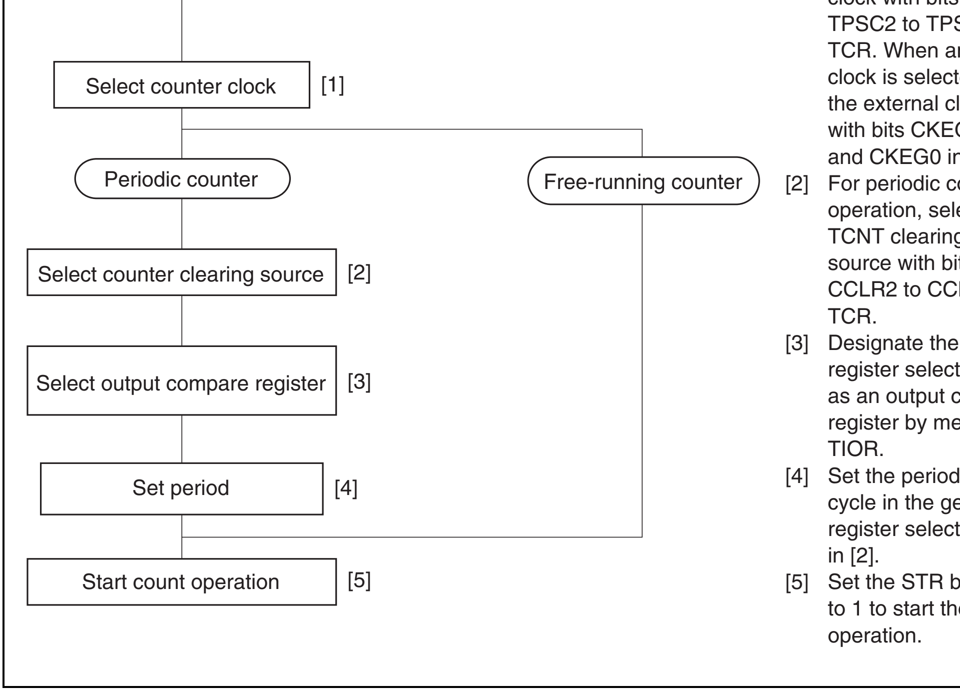


Figure 13.7 Example of Counter Operation Setting Procedure

1. Free-running count operation and periodic count operation

   Immediately after a reset, the TCNT counters for channels 0 and 1 are all designated running counters. When the relevant bit in TSTR is set to 1, the corresponding TCNT starts an increment operation as a free-running counter. When TCNT overflows, the in TSR is set to 1. If the value of the OVIE bit in the corresponding TIER is 1 at this timer Z requests an interrupt. After overflow, TCNT starts an increment operation ag H'0000.

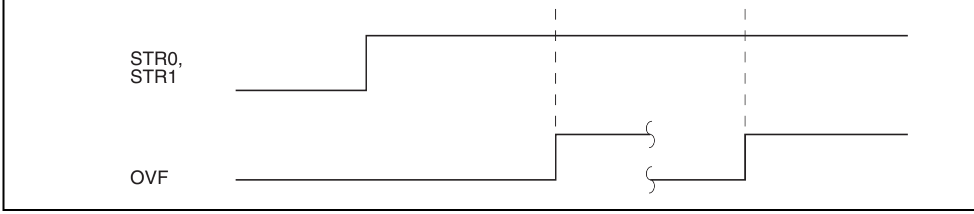


Figure 13.8 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the channel performs periodic count operation. The GR registers for setting the period are de as output compare registers, and counter clearing by compare match is selected by means CCLR1 and CCLR0 in TCR. After the settings have been made, TCNT starts an increme operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the value matches the value in GR, the IMFA, IMFB, IMFC, or IMFD flag in TSR is set to 1 TCNT is cleared to H'0000.

If the value of the corresponding IMIEA, IMIEB, IMIEC, or IMIED bit in TIER is 1 at th the timer Z requests an interrupt. After a compare match, TCNT starts an increment opera again from H'0000.

Figure 13.9 illustrates periodic counter operation.

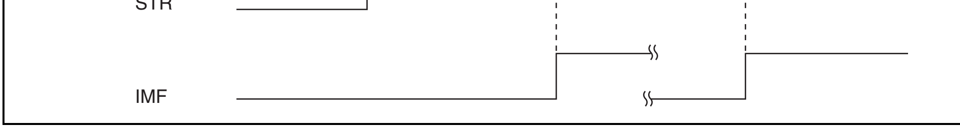


Figure 13.9 Periodic Counter Operation

2. TCNT count timing

A. Internal clock operation

A system clock (ϕ) or three types of clocks (ϕ/2, ϕ/4, or ϕ/8) that divides the syst
can be selected by bits TPSC2 to TPSC0 in TCR.

Figure 13.10 illustrates this timing.

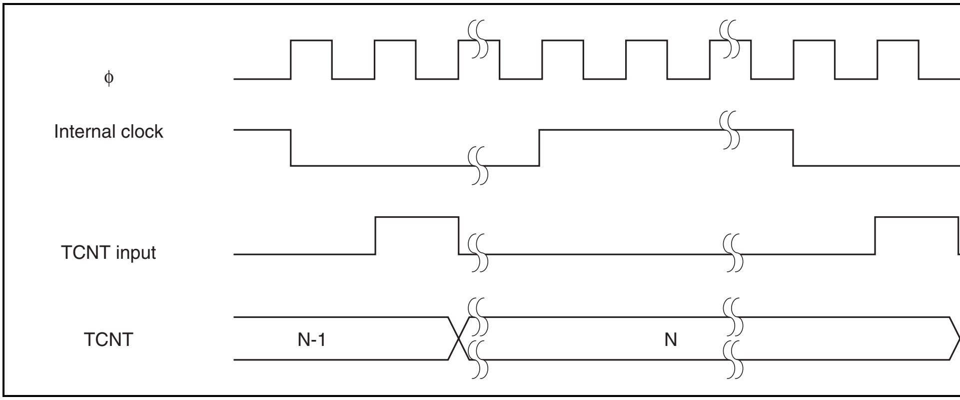


Figure 13.10 Count Timing at Internal Clock Operation

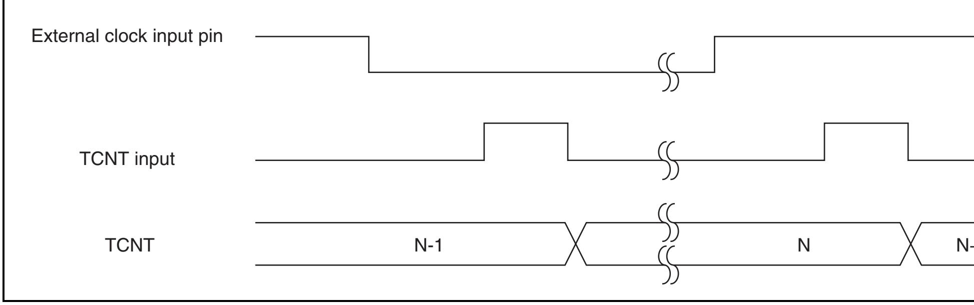
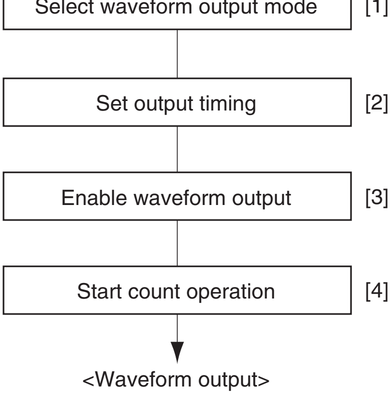


Figure 13.11   Count Timing at External Clock Operation (Both Edges Detect



match occurs.

[2] Set the timing for compare match generation in GRA/GRB/GRC/GRD.

[3] Enable or disable the timer output by TOER.

[4] Set the STR bit in TSTR to 1 to start the TCNT count operation.

Figure 13.12 Example of Setting Procedure for Waveform Output by Compare

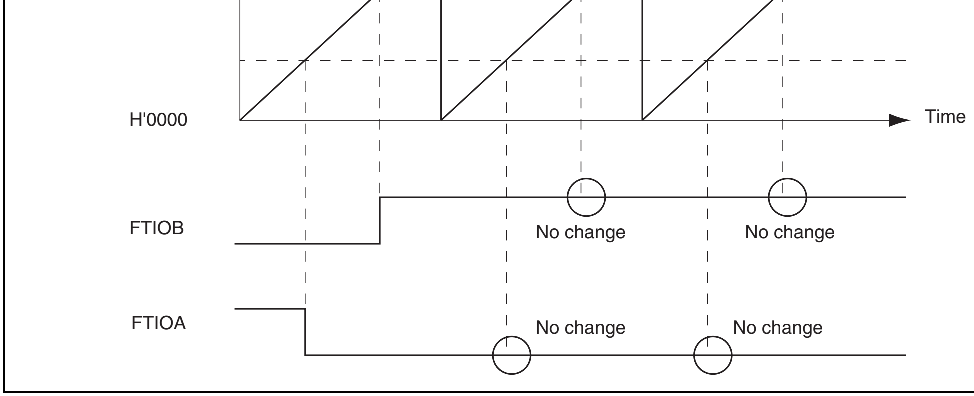


Figure 13.13 Example of 0 Output/1 Output Operation

Figure 13.14 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing by compare match B), and settings have been made such that the output is toggled by both compare match A and compare match B.

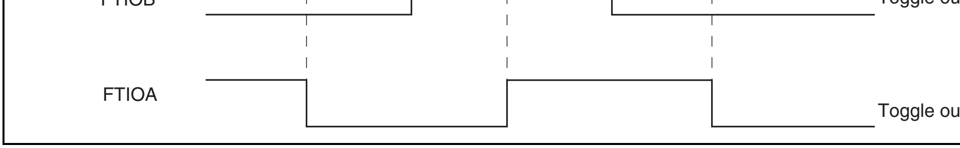


Figure 13.14 Example of Toggle Output Operation

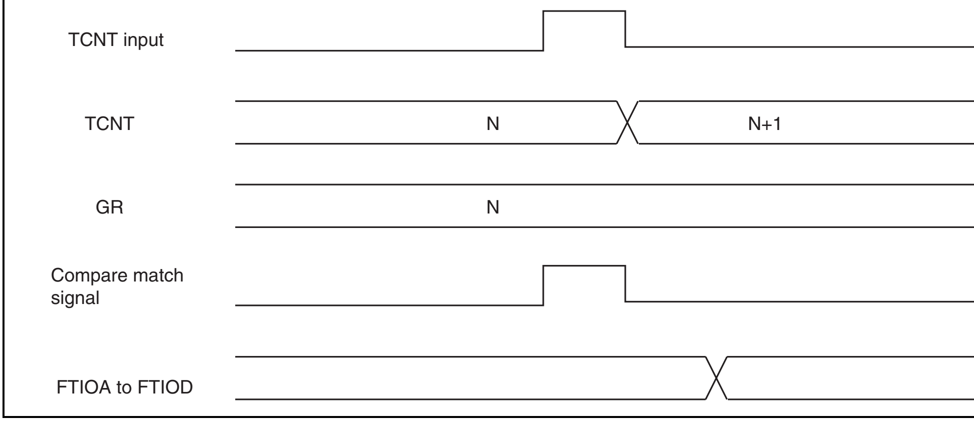


Figure 13.15 Output Compare Timing

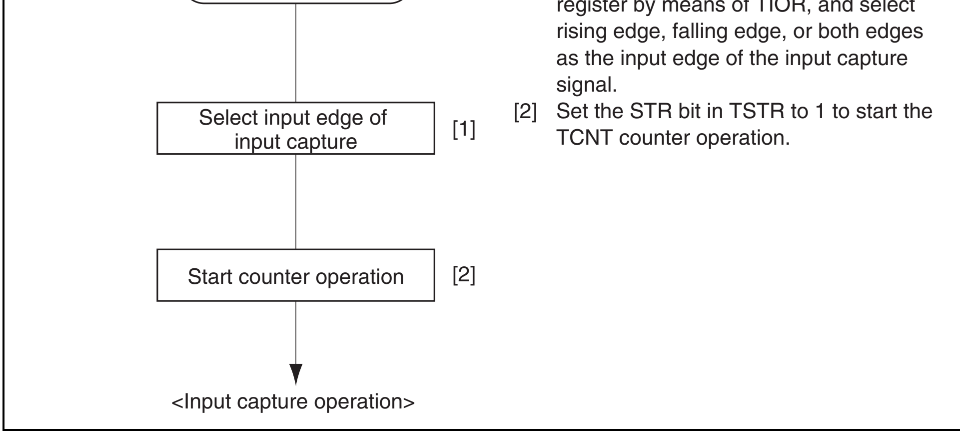


Figure 13.16 Example of Input Capture Operation Setting Procedure

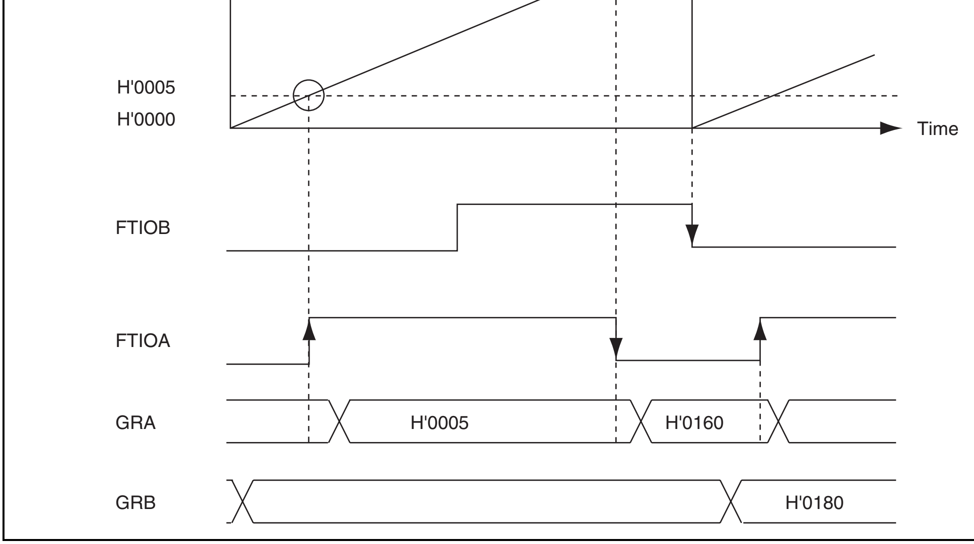


Figure 13.17 Example of Input Capture Operation

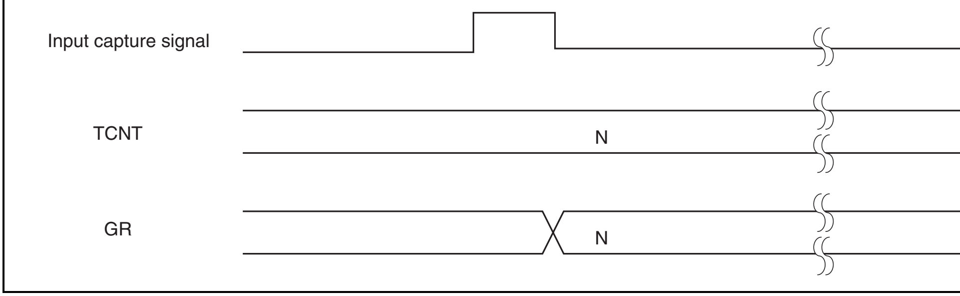
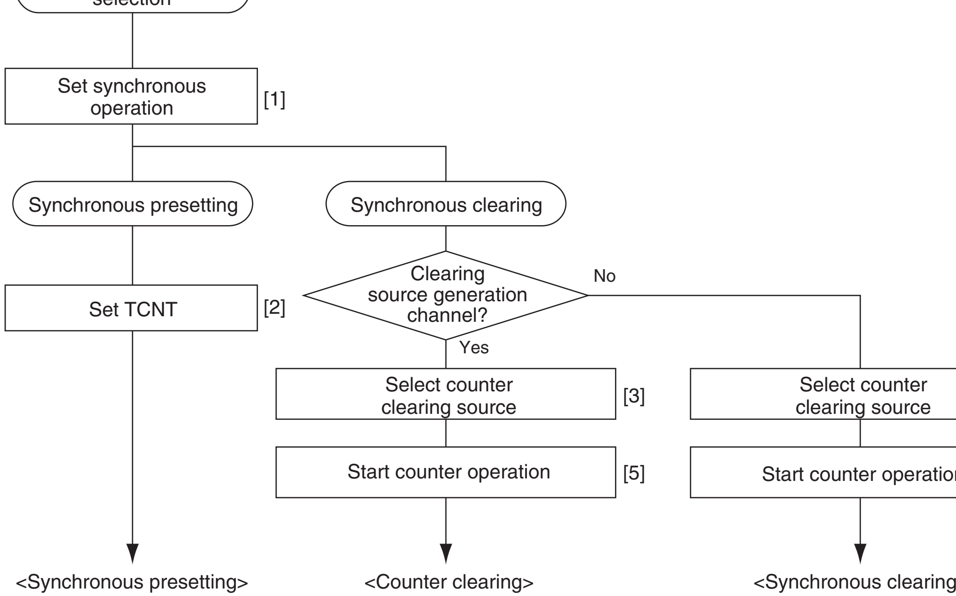


Figure 13.18 Input Capture Signal Timing



[1] Set the SYNC bits in TMDR to 1.

[2] When a value is written to either of the TCNT counters, the same value is simultaneously written to th other TCNT counter.

[3] Set bits CCLR1 and CCLR0 in TCR to specify counter clearing by compare match/input capture.

[4] Set bits CCLR1 and CCLR0 in TCR to designate synchronous clearing for the counter clearing source

[5] Set the STR bit in TSTR to 1 to start the count operation.

Figure 13.19 Example of Synchronous Operation Setting Procedure

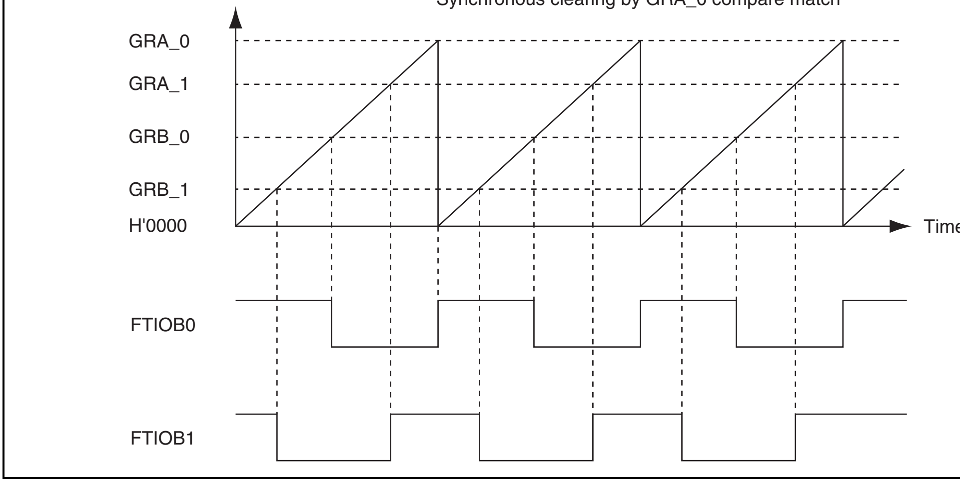


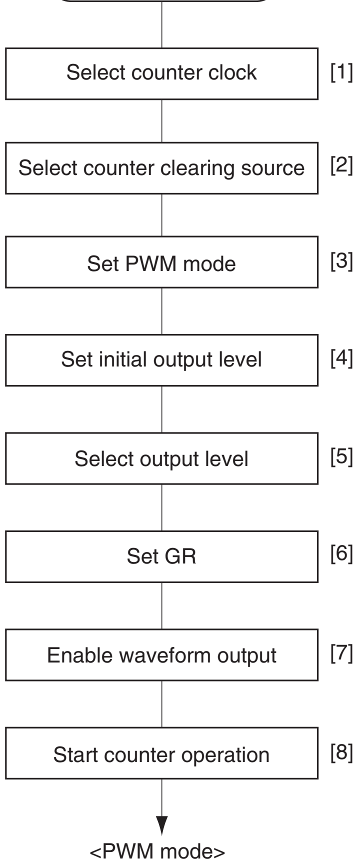
Figure 13.20 Example of Synchronous Operation

13.4.5 PWM Mode

In PWM mode, PWM waveforms are output from the FTIOB, FTIOC, and FTIOD outp with GRA as a cycle register and GRB, GRC, and GRD as duty registers. The initial ou of the corresponding pin depends on the setting values of TOCR and POCR. Table 13.3 example of the initial output level of the FTIOB0 pin.

The output level is determined by the POLB to POLD bits corresponding to POCR. Wh is 0, the FTIOB output pin is set to 0 by compare match B and set to 1 by compare matc When POLB is 1, the FTIOB output pin is set to 1 by compare match B and cleared to 0 compare match A. In PWM mode, maximum 6-phase PWM outputs are possible.

Figure 13.21 shows an example of the PWM mode setting procedure.



[1] Select the counter clock with bits TPSC2 to TOSC0 in TCR. When an external clock is selected, select the external clock edge with bits CKEG1 and CKEG0 in TCR.
[2] Use bits CCLR1 and CCLR0 in TCR to select the counter clearing source.
[3] Select the PWM mode with bits PWMB0 to PWMD0 and PWMB1 to PWMD1 in TPMR.
[4] Set the initial output value with bits TOB0 to TOD0 and TOB1 to TOD1 in TOCR.
[5] Set the output level with bits POLB to POLD in POCR.
[6] Set the cycle in GRA, and set the duty in the other GR.
[7] Enable or disable the timer output by TOER.
[8] Set the STR bit in TSTR to 1 and start the counter operation.

Figure 13.21 Example of PWM Mode Setting Procedure

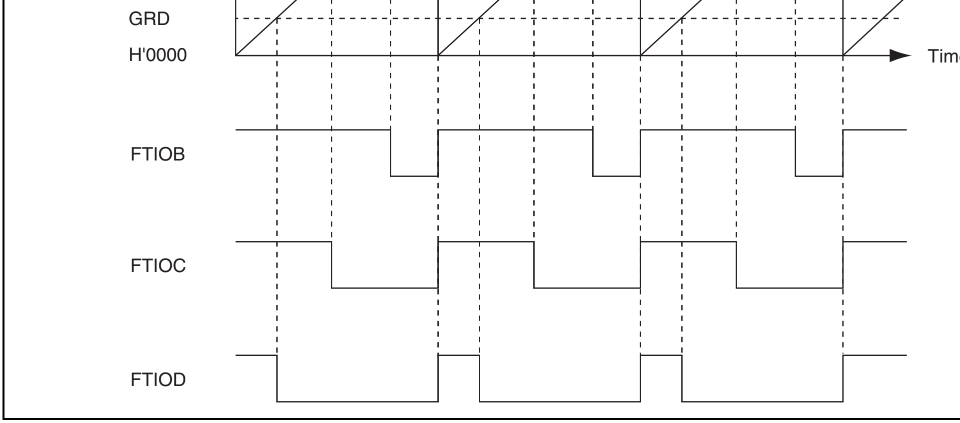


Figure 13.22 Example of PWM Mode Operation (1)

Figure 13.23 shows another example of operation in PWM mode. The output signals go
TCNT is reset at compare match A, and the output signals go to 1 at compare match B,
(TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 1).

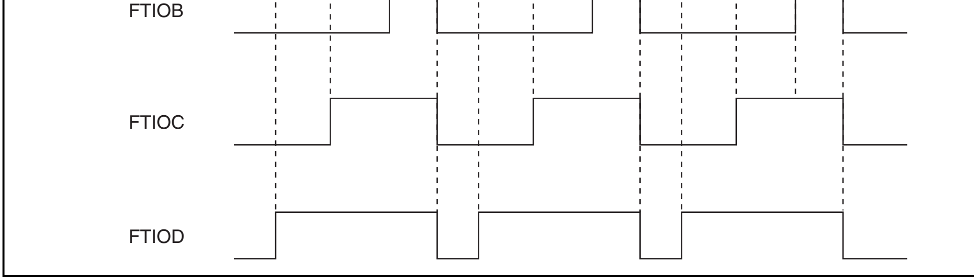


Figure 13.23   Example of PWM Mode Operation (2)

Figures 13.24 (when TOB, TOC, and TOD = 1, POLB, POLC, and POLD = 0) and 13.25
TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 1) show examples of the output o
waveforms with duty cycles of 0% and 100% in PWM mode.

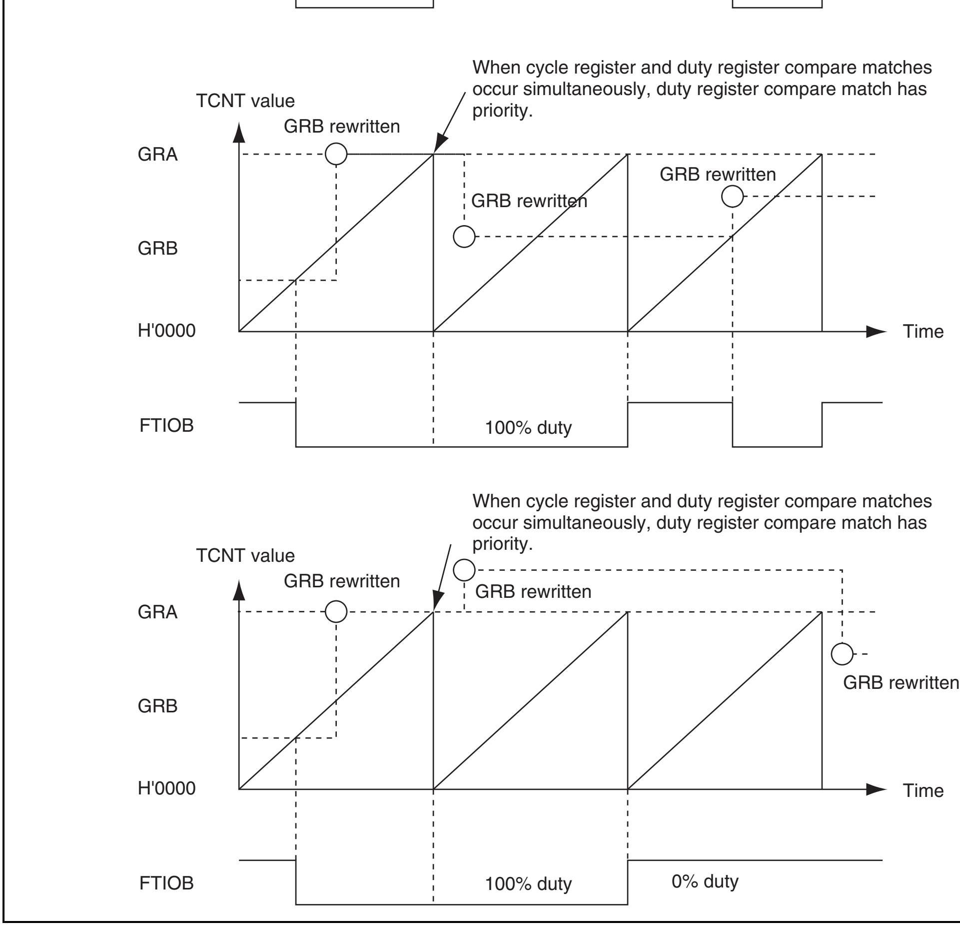


Figure 13.24 Example of PWM Mode Operation (3)

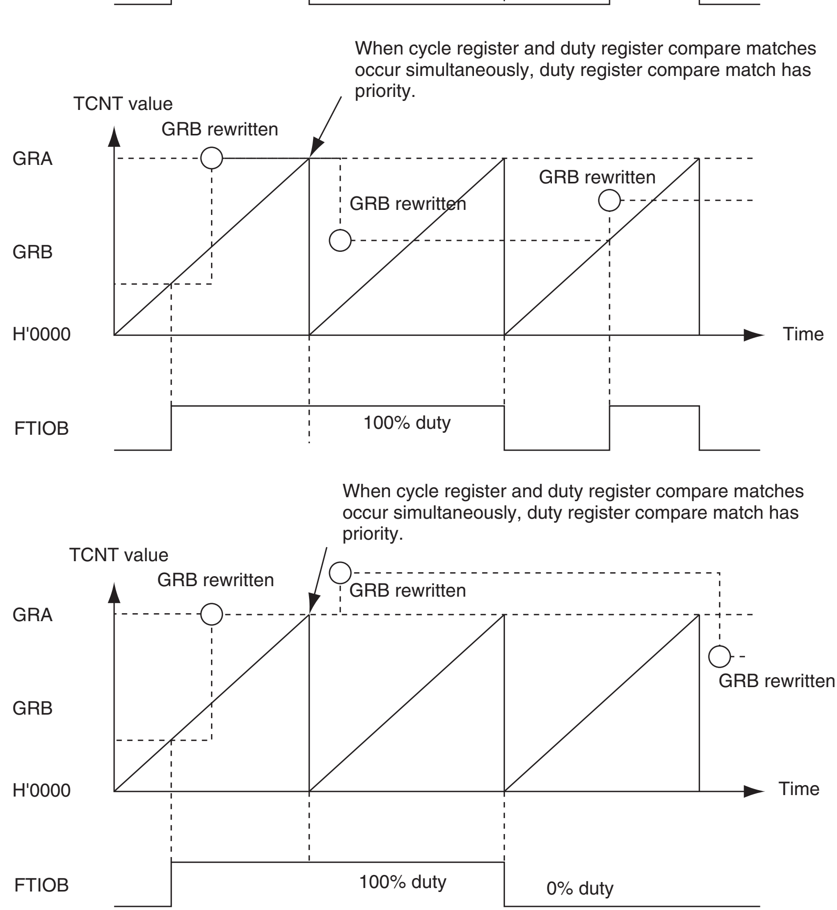
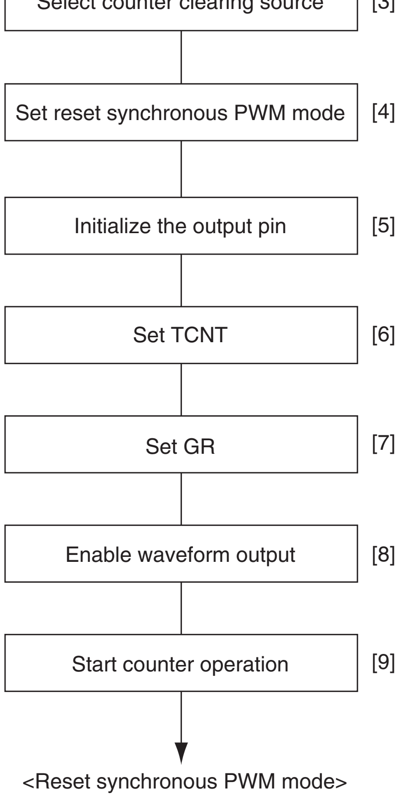


Figure 13.25 Example of PWM Mode Operation (4)

Channel	Pin Name	Input/Output	Pin Function
0	FTIOC0	Output	Toggle output in synchronous with PWM cyc
0	FTIOB0	Output	PWM output 1
0	FTIOD0	Output	PWM output 1 (counter-phase waveform of P output 1)
1	FTIOA1	Output	PWM output 2
1	FTIOC1	Output	PWM output 2 (counter-phase waveform of P output 2)
1	FTIOB1	Output	PWM output 3
1	FTIOD1	Output	PWM output 3 (counter-phase waveform of P output 3)

Table 13.5 Register Settings in Reset Synchronous PWM Mode

Register	Description
TCNT_0	Initial setting of H'0000
TCNT_1	Not used (independently operates)
GRA_0	Sets counter cycle of TCNT_0
GRB_0	Set a changing point of the PWM waveform output from pins FTIOB0 a FTIOD0.
GRA_1	Set a changing point of the PWM waveform output from pins FTIOA1 a FTIOC1.
GRB_1	Set a changing point of the PWM waveform output from pins FTIOB1 a FTIOD1.



[4] Select the reset synchronous PWM mo with bits CMD1 and CMD0 in TFCR. FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 become PWM output pins automatically.

[5] Set H'00 to TOCR.

[6] Set TCNT_0 as H'0000. TCNT1 does need to be set.

[7] GRA_0 is a cycle register. Set a cycle GRA_0. Set the changing point timing the PWM output waveform for GRB_0, GRA_1, and GRB_1.

[8] Enable or disable the timer output by TOER.

[9] Set the STR bit in TSTR to 1 and start counter operation.

Figure 13.26 Example of Reset Synchronous PWM Mode Setting Procedur

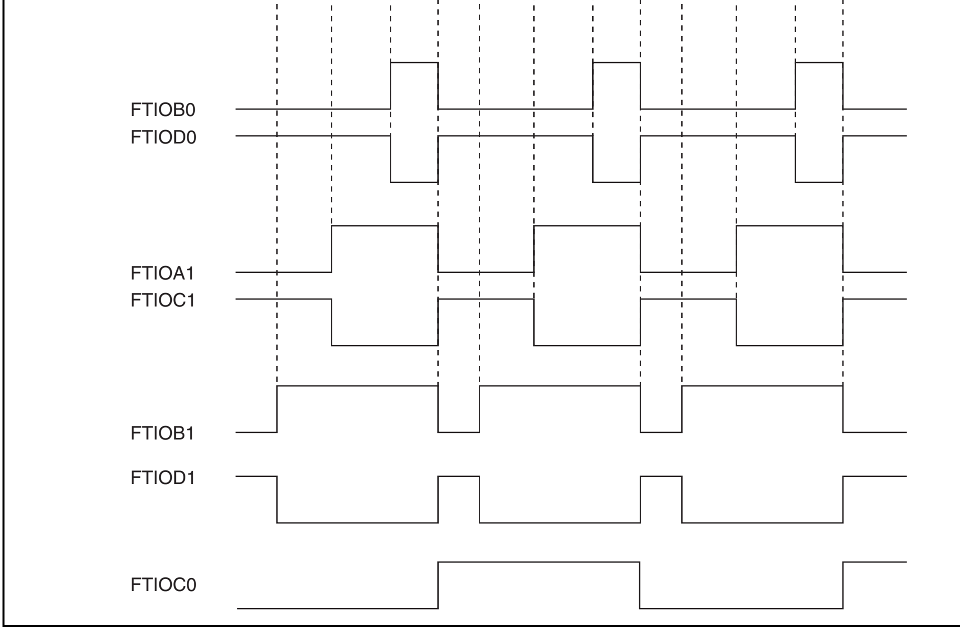


Figure 13.27 Example of Reset Synchronous PWM Mode Operation (OLS0 = O

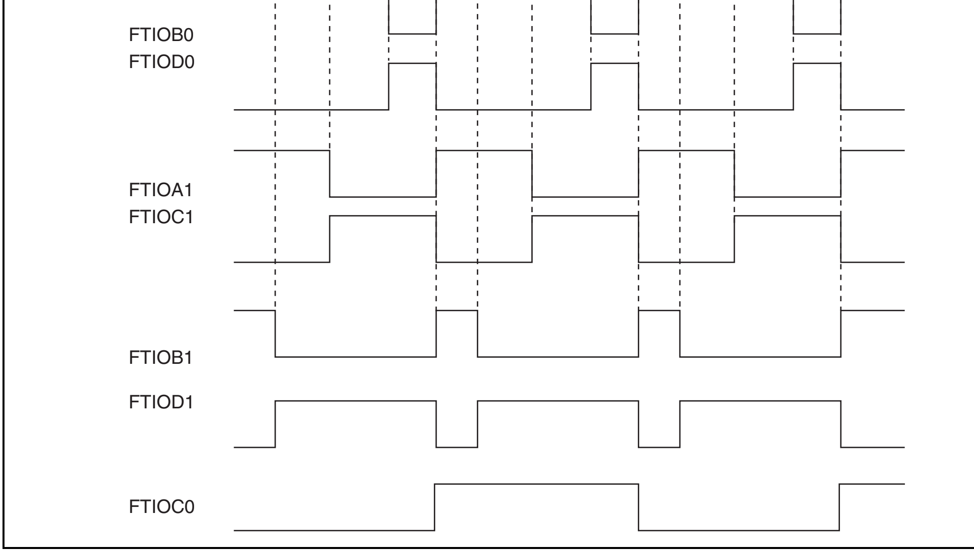


Figure 13.28 Example of Reset Synchronous PWM Mode Operation (OLS0 = OL

In reset synchronous PWM mode, TCNT_0 and TCNT_1 perform increment and indeper operations, respectively. However, GRA_1 and GRB_1 are separated from TCNT_1. W compare match occurs between TCNT_0 and GRA_0, a counter is cleared and an increm operation is restarted from H'0000.

The PWM pin outputs 0 or 1 whenever a compare match between GRB_0, GRA_1, GRB TCNT_0 or counter clearing occur.

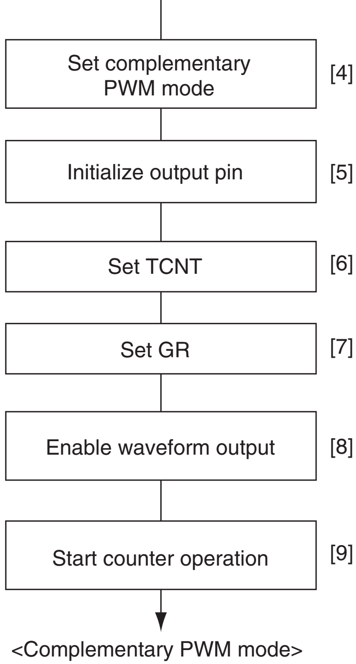
For details on operations when reset synchronous PWM mode and buffer operation are simultaneously set, refer to section 13.4.8, Buffer Operation.

Table 13.6 Output Pins in Complementary PWM Mode

Channel	Pin Name	Input/Output	Pin Function
0	FTIOC0	Output	Toggle output in synchronous with PWM cyc
0	FTIOB0	Output	PWM output 1
0	FTIOD0	Output	PWM output 1 (counter-phase waveform nor overlapped with PWM output 1)
1	FTIOA1	Output	PWM output 2
1	FTIOC1	Output	PWM output 2 (counter-phase waveform nor overlapped with PWM output 2)
1	FTIOB1	Output	PWM output 3
1	FTIOD1	Output	PWM output 3 (counter-phase waveform nor overlapped with PWM output 3)

Table 13.7 Register Settings in Complementary PWM Mode

Register	Description
TCNT_0	Initial setting of non-overlapped periods (non-overlapped periods are di with TCNT_1)
TCNT_1	Initial setting of H'0000
GRA_0	Sets (upper limit value – 1) of TCNT_0
GRB_0	Set a changing point of the PWM waveform output from pins FTIOB0 a FTIOD0.
GRA_1	Set a changing point of the PWM waveform output from pins FTIOA1 a FTIOC1.
GRB_1	Set a changing point of the PWM waveform output from pins FTIOB1 a FTIOD1.



[4] Use bits CMD1 and CMD0 in TFCR to set complementary PWM mode. FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 automatically become PWM output pins.

[5] Set H'00 to TOCR.

[6] TCNT_1 must be H'0000. Set a non-overlapped period to TCNT_0.

[7] GRA_0 is a cycle register. Set the cycle to GRA_0. Set the timing to change the PWM output waveform to GRB_0, GRA_1, and GRB_1. Note that the timing must be set within the range of compare match carried out for TCNT_0 and TCNT_1. For GR settings, see 3. Setting GR Value in Complementary PWM Mode in section 13.4.7.

[8] Use TOER to enable or disable the timer output.

[9] Set the STR0 and STR1 bits in TSTR to 1 to start the count operation.

Note: To re-enter complementary PWM mode, first, enter a mode other than the complement PWM mode. After that, repeat the setting procedures from step [1]. For settings of waveform outputs with a duty cycle of 0% and 100%, see the settings sh in 2. Examples of Complementary PWM Mode Operation and 3. Setting GR Value in Complementary PWM Mode in section 13.4.7.

Figure 13.29 Example of Complementar y PWM Mode Setting Procedure

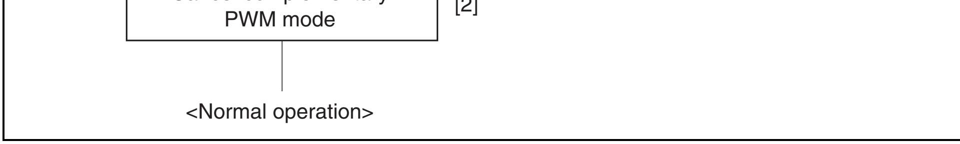


Figure 13.30 Canceling Procedure of Complementary PWM Mode

2. Examples of Complementary PWM Mode Operation: Figure 13.31 shows an exampl complementary PWM mode operation. In complementary PWM mode, TCNT_0 anc perform an increment or decrement operation. When TCNT_0 and GRA_0 are comp their contents match, the counter is decremented, and when TCNT_1 underflows, the is incremented. In GRA_0, GRA_1, and GRB_1, compare match is carried out in the TCNT_0 → TCNT_1 → TCNT_1 → TCNT_0 and PWM waveform is output, durin cycle of a up/down counter. In this mode, the initial setting will be TCNT_0 > TCNT

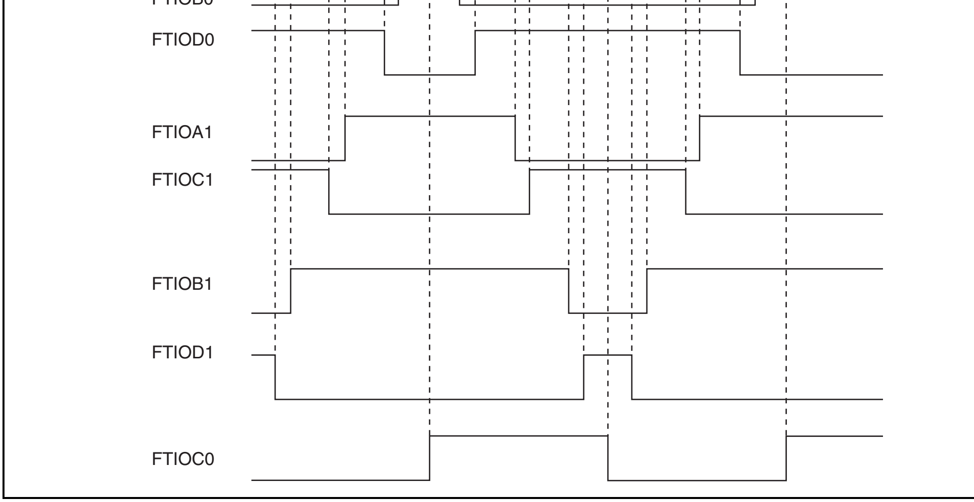


Figure 13.31 Example of Complementary PWM Mode Operation (1)

Figure 13.32 (1) and (2) show examples of PWM waveform output with 0% duty and 10
in complementary PWM mode (for one phase).

- TPSC2 = TPSC1 = TPSC0 = 0

  Set GRB_0 to H'0000 or a value equal to or more than GRA_0. The waveform with a cycle of 0% and 100% can be output. When buffer operation is used together, the dut can easily be changed, including the above settings, during operation. For details on b operation, refer to section 13.4.8, Buffer Operation.
- Other than TPSC2 = TPSC1 = TPSC0 = 0

  Set GRB_0 to satisfy the following expression: GRA_0 + 1 < GRB_0 < H'FFFF. The waveform with a duty cycle of 0% and 100% can be output. For details on 0%- and 1 cycle waveform output, see 3. C., Outputting a waveform with a duty cycle of 0% and section 13.4.7.

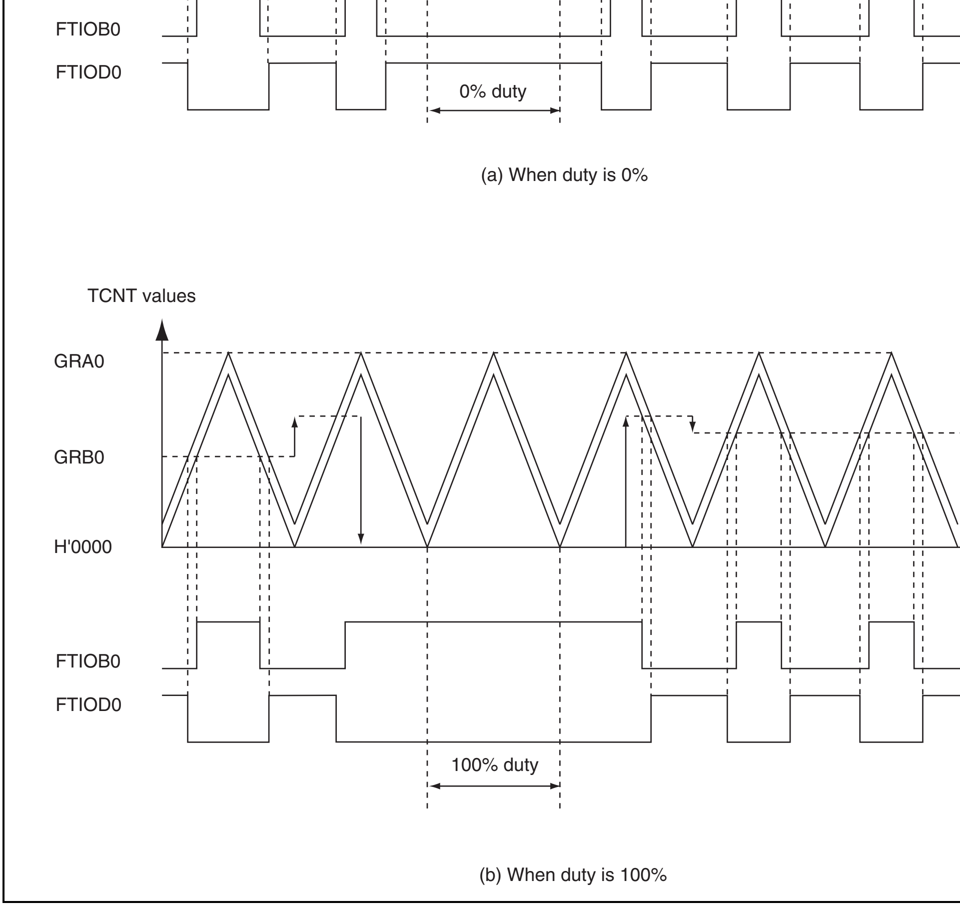
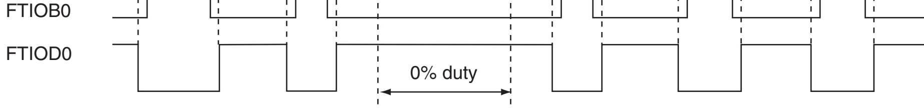
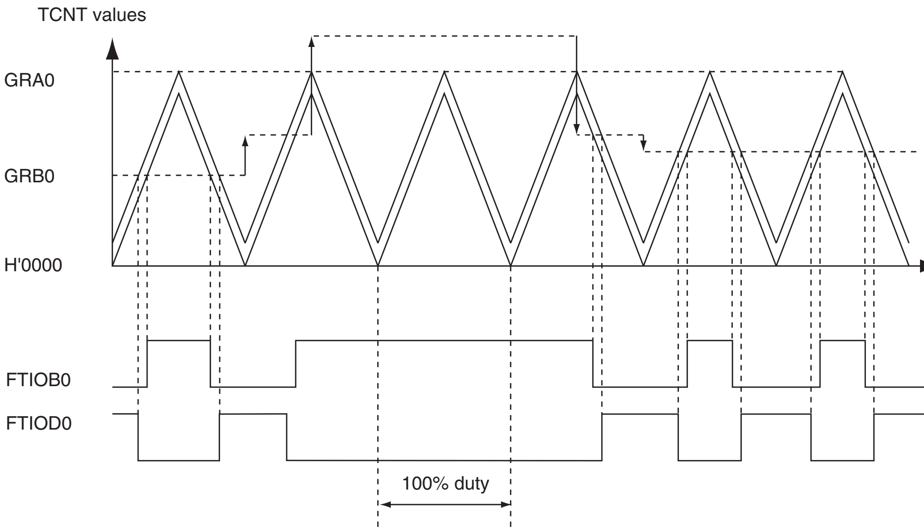


Figure 13.32 (1) Example of Complementary PWM Mode Operation (TPSC2 = TPSC1 = TPSC0 = 0) (2)



(a) When duty is 0%



(b) When duty is 100%

Figure 13.32 (2) Example of Complementary PWM Mode Operation (TPSC2 = TPSC1 = TPSC0 ≠ 0) (3)

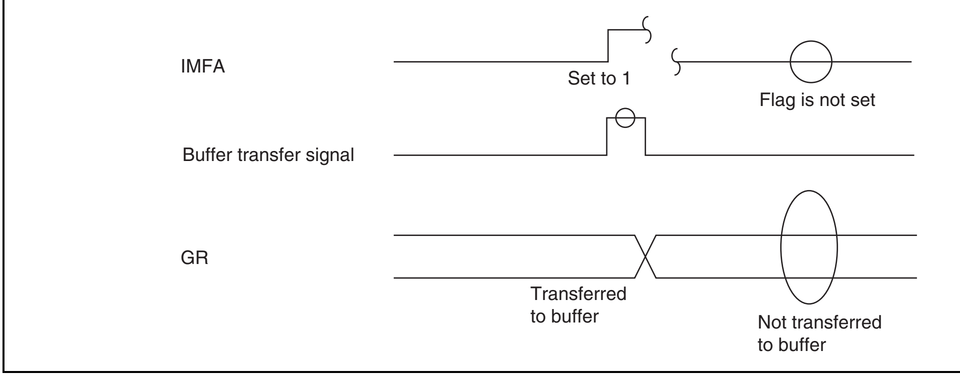


Figure 13.33 Timing of Overshooting

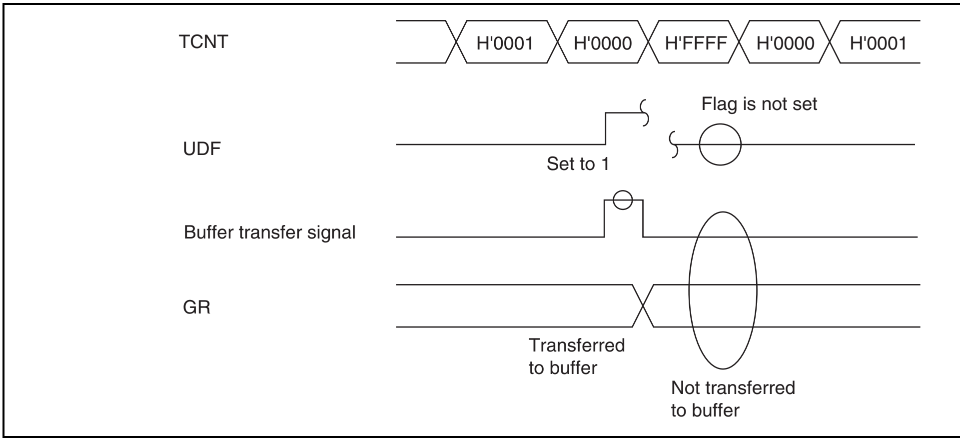


Figure 13.34 Timing of Undershooting

H'FFFC or less. When TPSC2 = TPSC1 = TPSC0 = 0, the GRA_0 value can b
H'FFFF or less.

b. H'0000 to T – 1 (T: Initial value of TCNT0) must not be set for the initial valu

c. GRA_0 – (T – 1) or more must not be set for the initial value.

d. When using buffer operation, the same values must be set in the buffer register corresponding general registers.

B. Modifying the setting value

a. Writing to GR directly must be performed while the TCNT_1 and TCNT_0 va should satisfy the following expression: H'0000 ≤ TCNT_1 < previous GR val previous GR value < TCNT_0 ≤ GRA_0. Otherwise, a waveform is not output correctly. For details on outputting a waveform with a duty cycle of 0% and 10 C., Outputting a waveform with a duty cycle of 0% and 100%.

b. Do not write the following values to GR directly. When writing the values, a w is not output correctly.

H'0000 ≤ GR ≤ T – 1 and GRA_0 – (T – 1) ≤ GR < GRA_0 when TPSC2 = T TPSC0 = 0

H'0000 < GR ≤ T – 1 and GRA_0 – (T – 1) ≤ GR < GRA_0 + 1 when TPSC2 = TPSC0 = 0

c. Do not change settings of GRA_0 during operation.

C. Outputting a waveform with a duty cycle of 0% and 100%

a. Buffer operation is not used and TPSC2 = TPSC1 = TPSC0 = 0

Write H'0000 or a value equal to or more than the GRA_0 value to GR directly timing shown below.

- To output a 0%-duty cycle waveform, write a value equal to or more than the G value while H'0000 ≤ TCNT_1 < previous GR value
- To output a 100%-duty cycle waveform, write H'0000 while previous GR valu TCNT_0 ≤ GRA_0

- To output a 0%-duty cycle waveform, write a value equal to or more than the value to the buffer register
- To output a 100%-duty cycle waveform, write H'0000 to the buffer register

  For details on buffer operation, see section 13.4.8, Buffer Operation.

c. Buffer operation is not used and other than TPSC2 = TPSC1 = TPSC0 = 0

  Write a value which satisfies GRA_0 + 1 < GR < H'FFFF to GR directly at th shown below.

- To output a 0%-duty cycle waveform, write the value while H'0000 ≤ TCNT_ previous GR value
- To output a 100%-duty cycle waveform, write the value while previous GR v TCNT_0 ≤ GRA_0

  To change duty cycles while a waveform with a duty cycle of 0% and 100% i output, the following procedure must be followed.

- To change duty cycles while a 0%-duty cycle waveform is being output, writ while H'0000 ≤ TCNT_1 < previous GR value
- To change duty cycles while a 100%-duty cycle waveform is being output, w while previous GR value< TCNT_0 ≤ GRA_0

  Note that changing from a 0%-duty cycle waveform to a 100%-duty cycle wa and vice versa is not possible.

d. Buffer operation is used and other than TPSC2 = TPSC1 = TPSC0 = 0

  Write a value which satisfies GRA_0 + 1 < GR < H'FFFF to the buffer regist waveform with a duty cycle of 0% can be output. However, a waveform with cycle of 100% cannot be output using the buffer operation. Also, the buffer o cannot be used to change duty cycles while a waveform with a duty cycle of 1 being output. For details on buffer operation, see section 13.4.8, Buffer Opera

GRA	GRC
GRB	GRD

1. When GR is an output compare register

   When a compare match occurs, the value in the buffer register of the corresponding cl transferred to the general register.

   This operation is illustrated in figure 13.35.

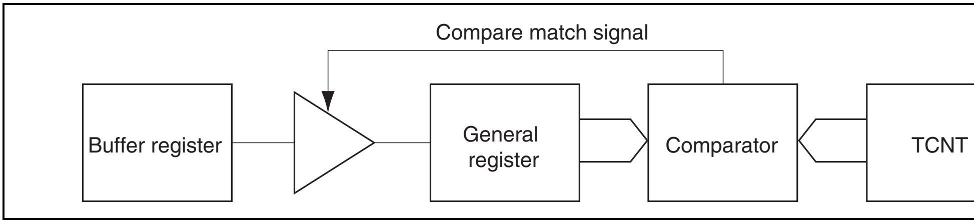


Figure 13.35 Compare Match Buffer Operation

2. When GR is an input capture register

   When an input capture occurs, the value in TCNT is transferred to the general register value previously stored in the general register is transferred to the buffer register.

   This operation is illustrated in figure 13.36.

buffer register is transferred to the general register. Here, the value of the buffer regi
transferred to the general register in the following timing:

A. When TCNT_0 and GRA_0 are compared and their contents match

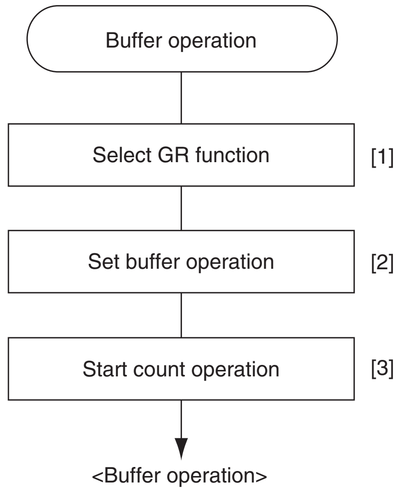
B. When TCNT_1 underflows

4. Reset Synchronous PWM Mode

   The value of the buffer register is transferred from compare match A0 to the general

5. Example of Buffer Operation Setting Procedure

   Figure 13.37 shows an example of the buffer operation setting procedure.



[1] Designate GR as an input capture registe or output compare register by means of TIOR.
[2] Designate GR for buffer operation with b BFD1, BFC1, BFD0, or BFC0 in TMDR.
[3] Set the STR bit in TSTR to 1 to start the count operation of TCNT.

Figure 13.37 Example of Buffer Operation Setting Procedure

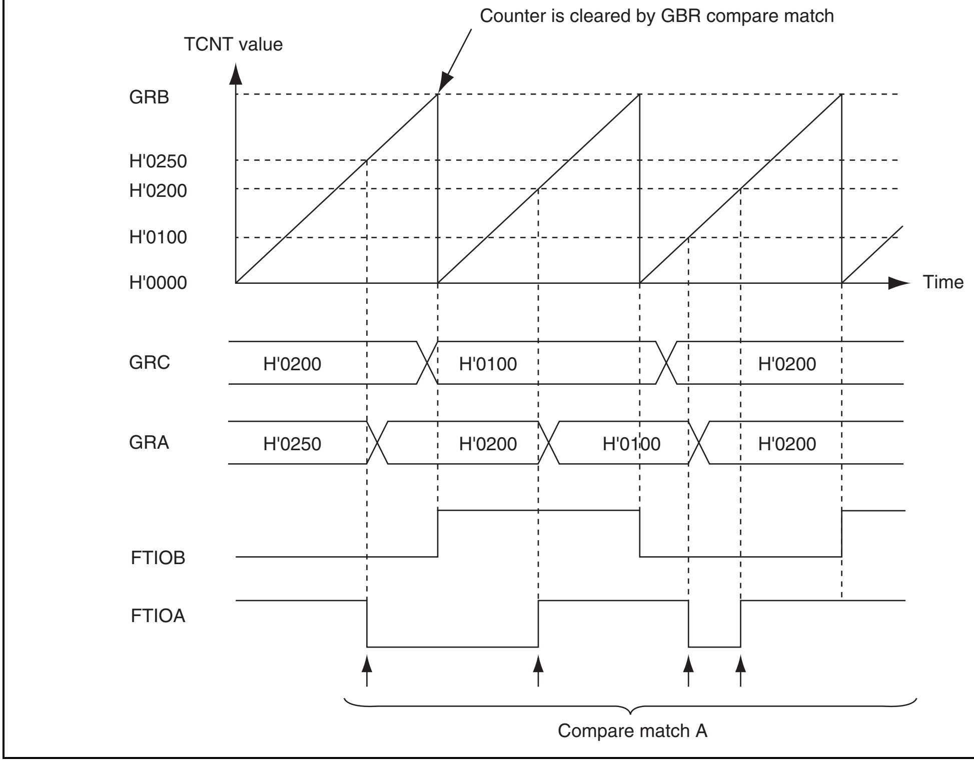


Figure 13.38 Example of Buffer Operation (1) (Buffer Operation for Output Compare Register)

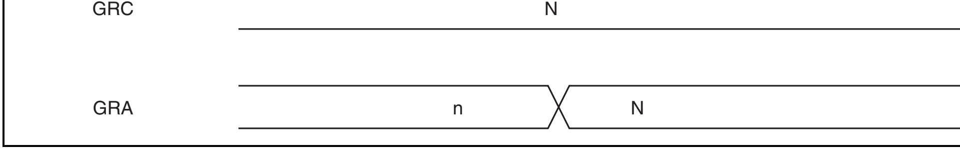


Figure 13.39 Example of Compare Match Timing for Buffer Operation

Figure 13.40 shows an operation example in which GRA has been designated as an inpu register, and buffer operation has been designated for GRA and GRC.

Counter clearing by input capture B has been set for TCNT, and falling edges have been as the FIOCB pin input capture input edge. And both rising and falling edges have been as the FIOCA pin input capture input edge.

As buffer operation has been set, when the TCNT value is stored in GRA upon the occu input capture A, the value previously stored in GRA is simultaneously transferred to GR transfer timing is shown in figure 13.41.

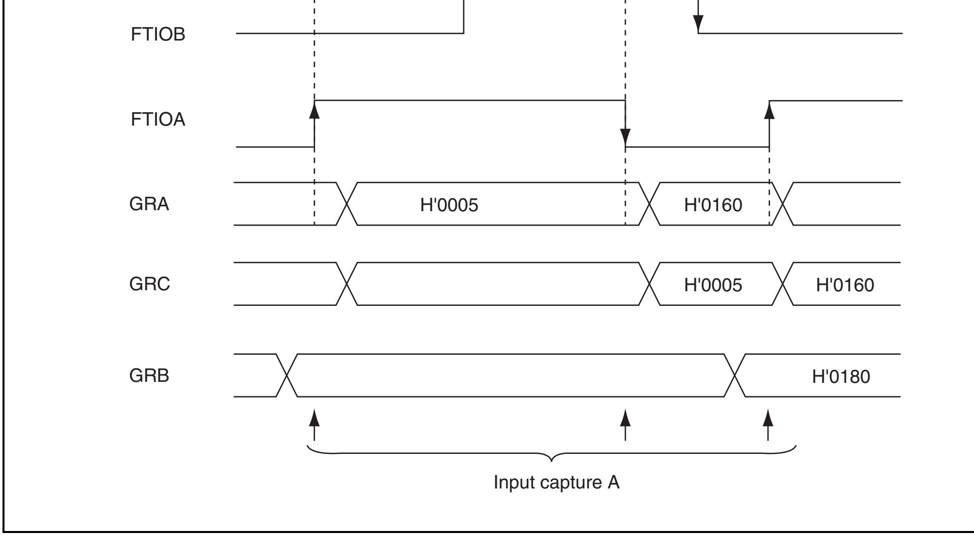


**Figure 13.40 Example of Buffer Operation (2)
(Buffer Operation for Input Capture Register)**

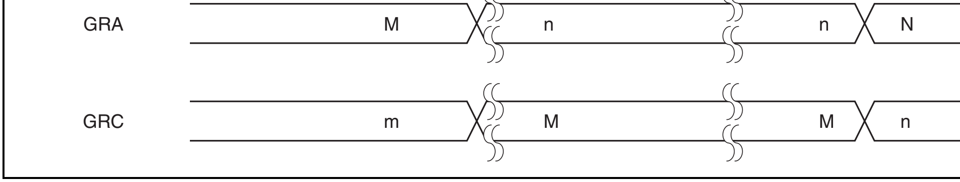


Figure 13.41 Input Capture Timing of Buffer Operation

Figures 13.42 and 13.43 show the operation examples when buffer operation has been d
for GRB_0 and GRD_0 in complementary PWM mode. These are examples when a PW
waveform of 0% duty is created by using the buffer operation and performing GRD_0 ≥
Data is transferred from GRD_0 to GRB_0 according to the settings of CMD_0 and CM
TCNT_0 and GRA_0 are compared and their contents match or when TCNT_1 underflo
However, when GRD_0 ≥ GRA_0, data is transferred from GRD_0 to GRB_0 when TC
underflows regardless of the setting of CMD_0 and CMD_1. When GRD_0 = H'0000, d
transferred from GRD_0 to GRB_0 when TCNT_0 and GRA_0 are compared and their
match regardless of the settings of CMD_0 and CMD_1.

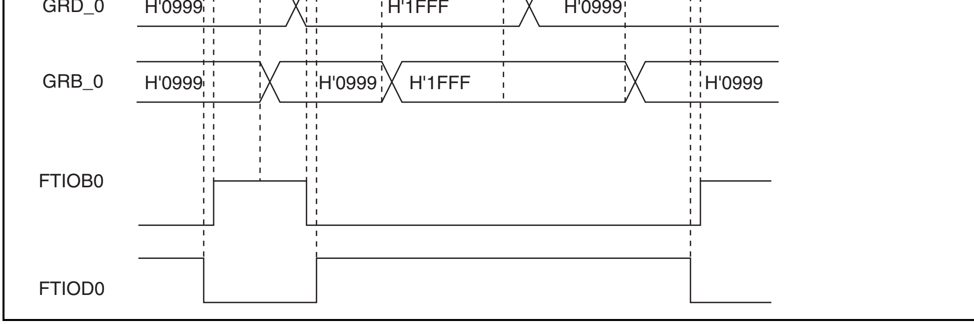


Figure 13.42 Buffer Operation (3)
(Buffer Operation in Complementary PWM Mode CMD1 = CMD0 = 1)

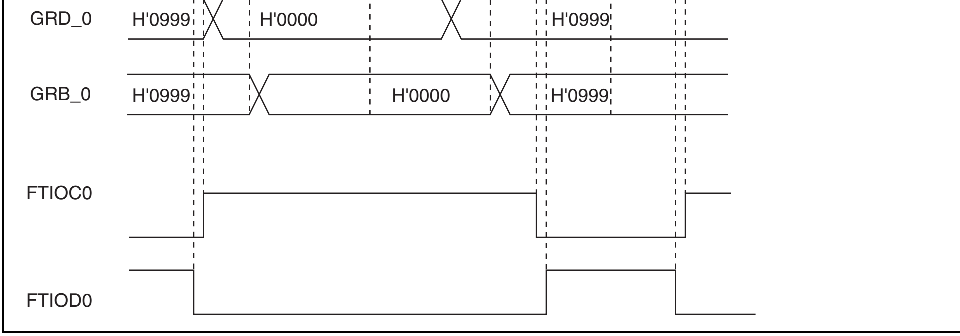


Figure 13.43 Buffer Operation (4)
(Buffer Operation in Complementary PWM Mode CMD1 = CMD0 = 1)

13.4.9 Timer Z Output Timing

The outputs of channels 0 and 1 can be disabled or inverted by the settings of TOER anc
and the external level.

1. Output Disable/Enable Timing of Timer Z by TOER: Setting the master enable bit ir
   1 disables the output of timer Z. By setting the PCR and PDR of the corresponding I
   beforehand, any value can be output. Figure 13.44 shows the timing to enable or disa
   output of timer Z by TOER.

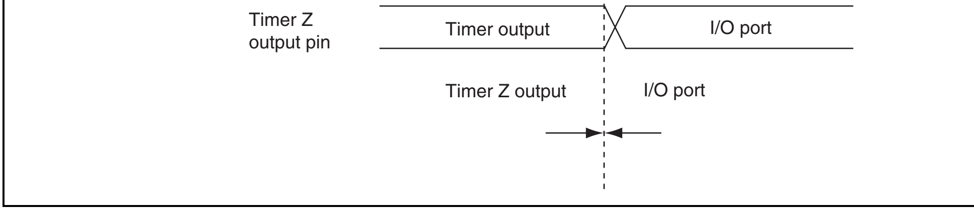


Figure 13.44 Example of Output Disable Timing of Timer Z by Writing to TC

2. Output Disable Timing of Timer Z by External Trigger: When P54/$\overline{\text{WKP4}}$ is set as a input pin, and low level is input to $\overline{\text{WKP4}}$, the master enable bit in TOER is set to 1 a output of timer Z will be disabled.

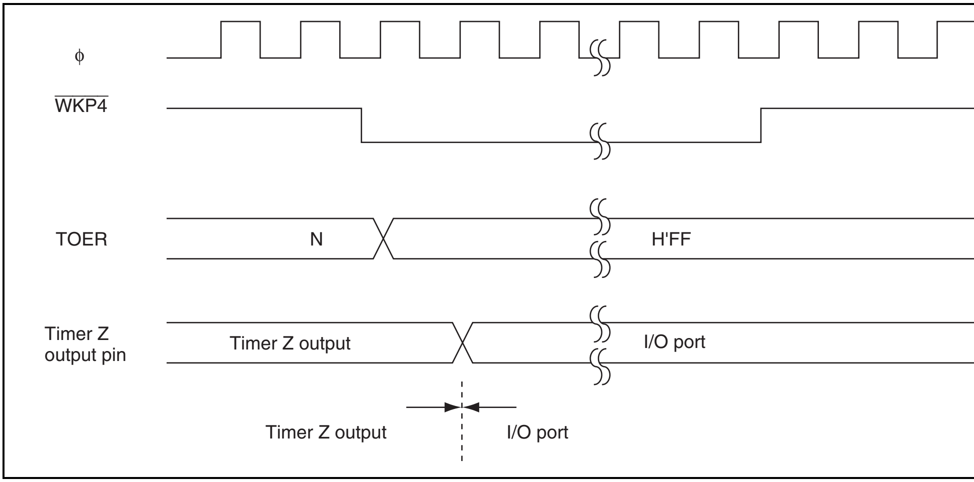


Figure 13.45 Example of Output Disable Timing of Timer Z by External Trig

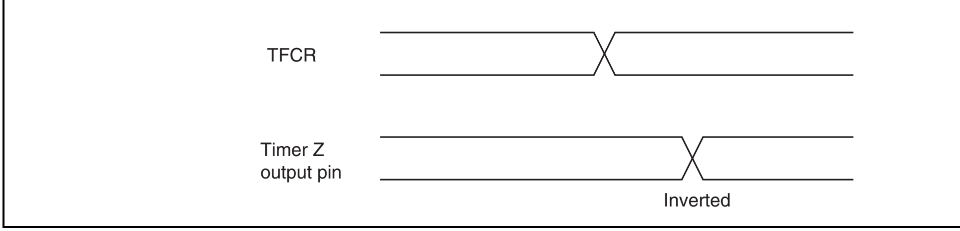


Figure 13.46 Example of Output Inverse Timing of Timer Z by Writing to T

4. Output Inverse Timing by POCR: The output level can be inverted by inverting the POLC, and POLB bits in POCR in PWM mode. Figure 13.47 shows the timing.

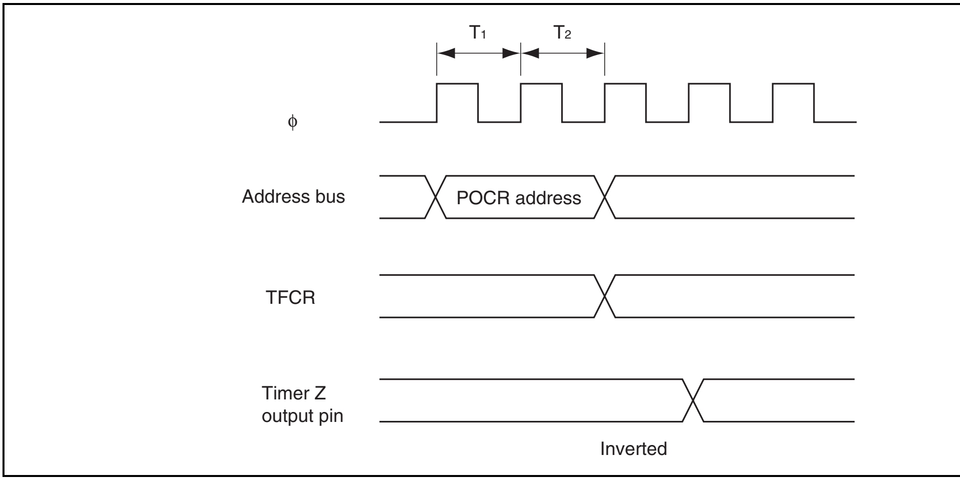


Figure 13.47 Example of Output Inverse Timing of Timer Z by Writing to P

when the TCNT and GR matches, the compare match signal will not be generated un
TCNT input clock is generated. Figure 13.48 shows the timing to set the IMF flag.

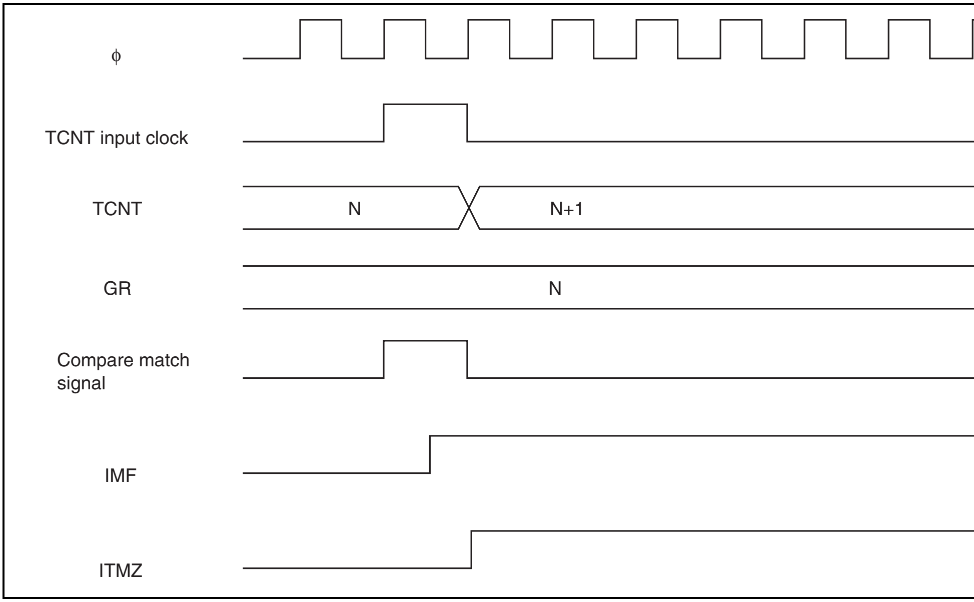


Figure 13.48 IMF Flag Set Timing when Compare Match Occurs

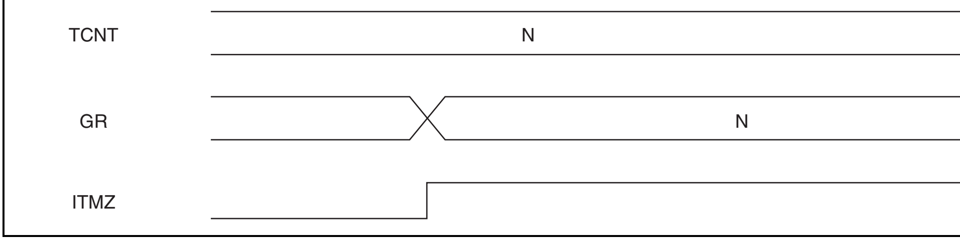


Figure 13.49 IMF Flag Set Timing at Input Capture

3. Overflow Flag (OVF) Set Timing: The overflow flag is set to 1 when the TCNT ove
Figure 13.50 shows the timing.

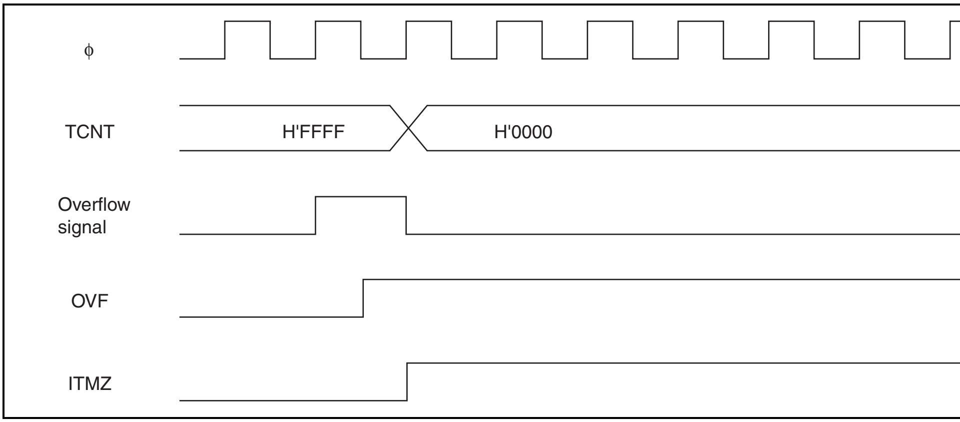


Figure 13.50 OVF Flag Set Timing

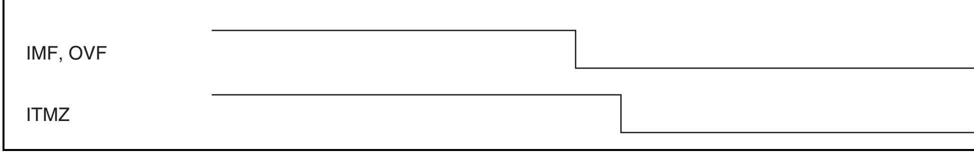


Figure 13.51 Status Flag Clearing Timing

13.6 Usage Notes

1. Contention between TCNT Write and Clear Operations: If a counter clear signal is ge in the T_2 state of a TCNT write cycle, TCNT clearing has priority and the TCNT write performed. Figure 13.52 shows the timing in this case.

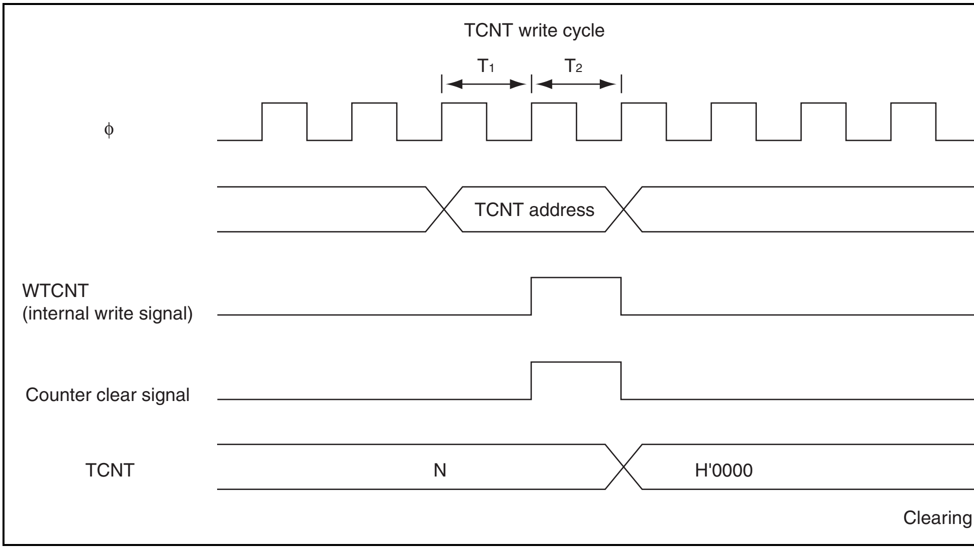


Figure 13.52 Contention between TCNT Write and Clear Operations

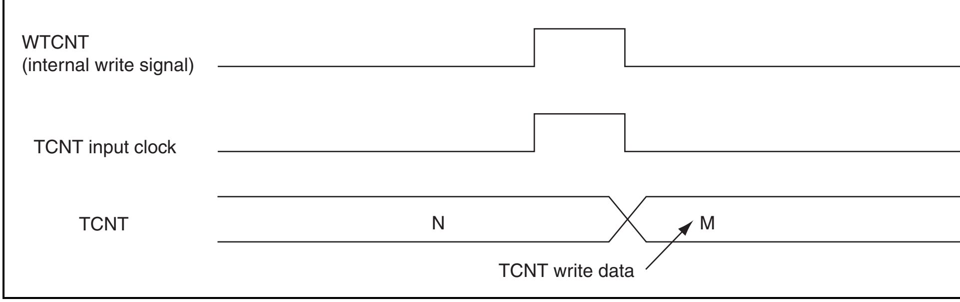


Figure 13.53   Contention between TCNT Write and Increment Operation

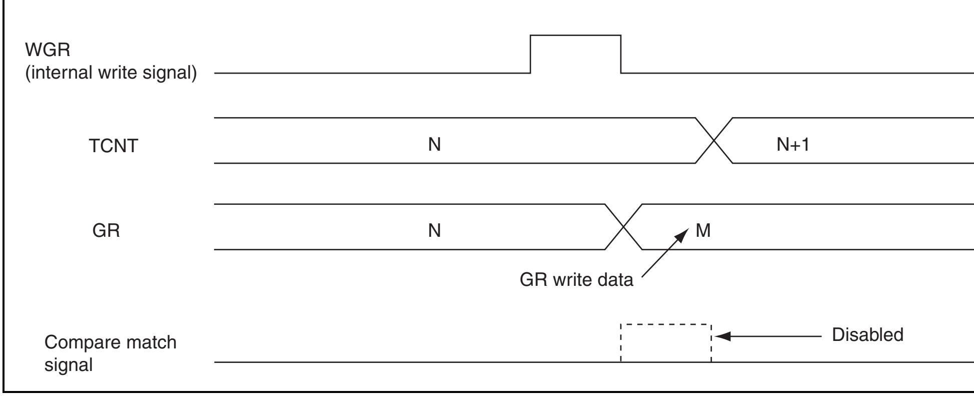


Figure 13.54 Contention between GR Write and Compare Match

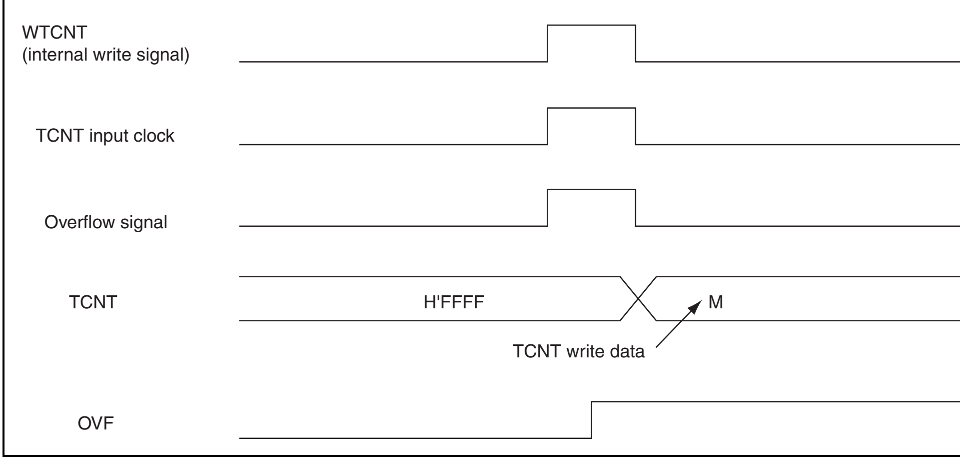


Figure 13.55 Contention between TCNT Write and Overflow

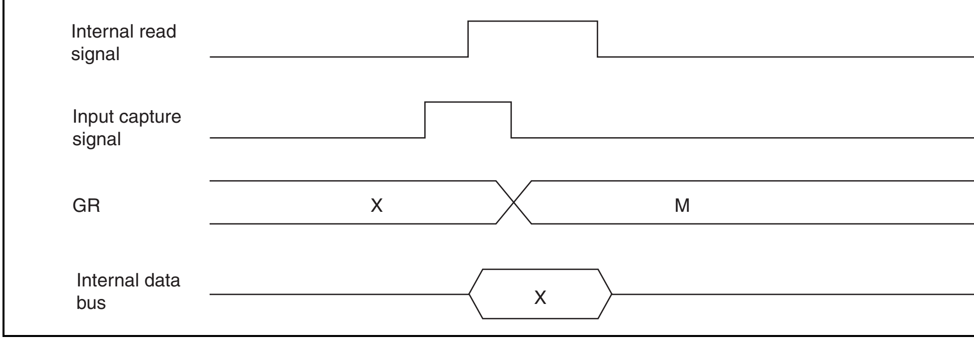


Figure 13.56 Contention between GR Read and Input Capture

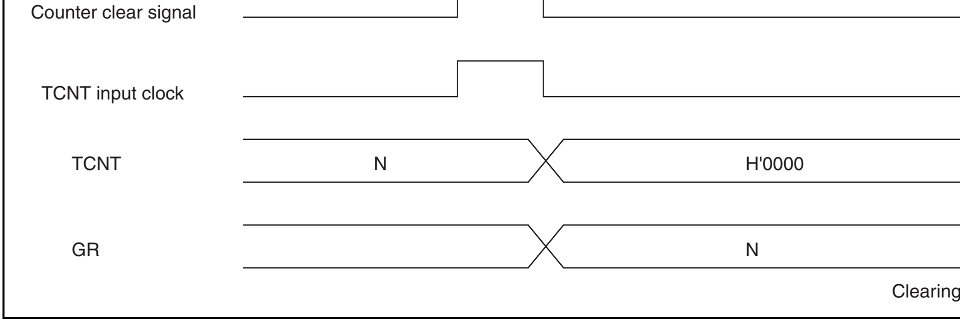


Figure 13.57 Contention between Count Clearing and Increment Operatio by Input Capture

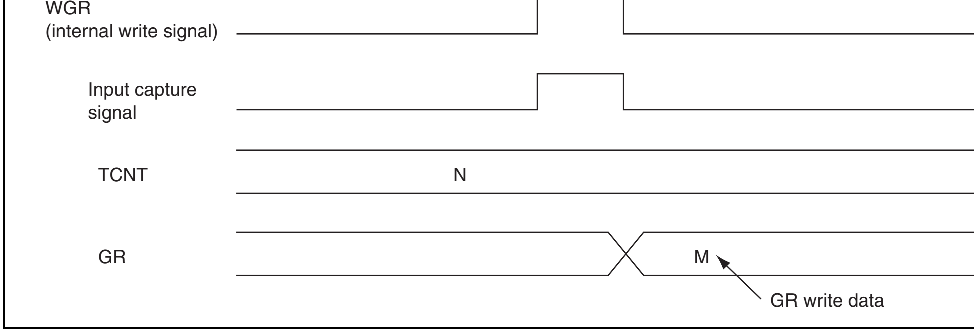


Figure 13.58 Contention between GR Write and Input Capture

instruction must be executed. Note that this note is only applied to the F-ZTAT vers
problem has already been solved in the mask ROM version.

Example: When clearing bit 4 (OVF) in TSR

```
MOV.B @TSR,R0L
MOV.B #B'11101111, R0L  <------ Only the bit to be cleared is 0 and
                                the other bits are all set to 1.
MOV.B R0L,@TSR
```

10. Note on Writing to the TOA0 to TOD0 Bits and the TOA1 to TOD1 Bits in TOCR:

The TOA0 to TOD0 bits and the TOA1 to TOD1 bits in TOCR decide the value of t pin, which is output until the first compare match occurs. Once a compare match occ this compare match changes the values of FTIOA0 to FTIOD0 and FTIOA1 to FTIC output, the values of the FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pin output and values read from the TOA0 to TOD0 and TOA1 to TOD1 bits may differ. Moreover writing to TOCR and the generation of the compare match A0 to D0 and A1 to D1 o same timing, the writing to TOCR has the priority. Thus, output change due to the co match is not reflected to the FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pins. Ther when bit manipulation instruction is used to write to TOCR, the values of the FTIOA FTIOD0 and FTIOA1 to FTIOD1 pin output may result in an unexpected result. Wh is to be written to while compare match is operating, stop the counter once before ac TOCR, read the port 6 state to reflect the values of FTIOA0 to FTIOD0 and FTIOA1 FTIOD1 output, to TOA0 to TOD0 and TOA1 to TOD1, and then restart the counter 13.59 shows an example when the compare match and the bit manipulation instructio TOCR occur at the same timing.

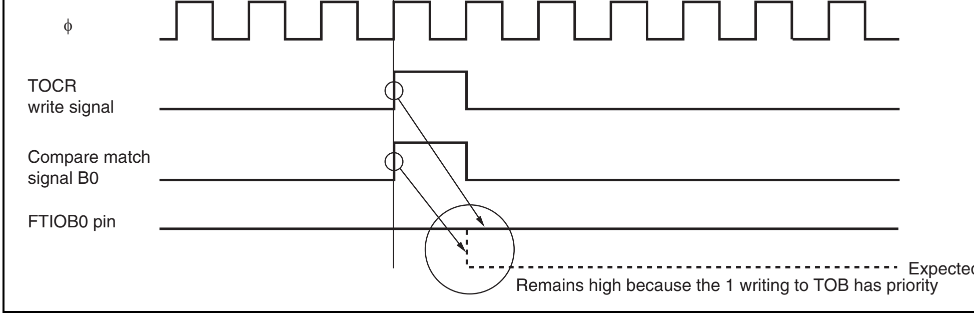


Figure 13.59 When Compare Match and Bit Manipulation Instruction to TO Occur at the Same Timing

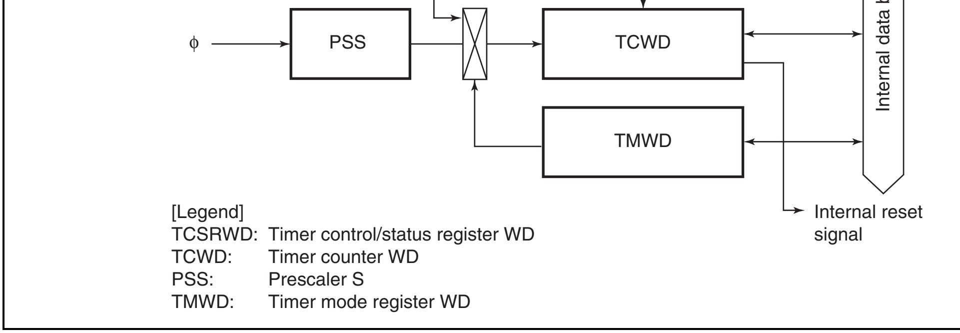


Figure 14.1 Block Diagram of Watchdog Timer

14.1 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.

14.2 Register Descriptions

The watchdog timer has the following registers.

- Timer control/status register WD (TCSRWD)
- Timer counter WD (TCWD)
- Timer mode register WD (TMWD)

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.
2	WDON	0	R/W	Watchdog Timer On TCWD starts counting up when WDON is set to halts when WDON is cleared to 0. [Setting condition] When 1 is written to the WDON bit while writing B2WI bit when the TCSRWE bit=1 [Clearing conditions] • Reset by $\overline{RES}$ pin • When 0 is written to the WDON bit while writi 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 whe write value of the B0WI bit is 0. This bit is always 1.

14.2.2 Timer Counter WD (TCWD)

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

14.2.3 Timer Mode Register WD (TMWD)

TMWD selects the input clock.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	CKS3	1	R/W	Clock Select 3 to 0
2	CKS2	1	R/W	Select the clock to be input to TCWD.
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 23, Electrical Characteristics.

Legend: X: Don't care.

Figure 14.2 shows an example of watchdog timer operation.

Example: With 30ms overflow period when ϕ = 4 MHz

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

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

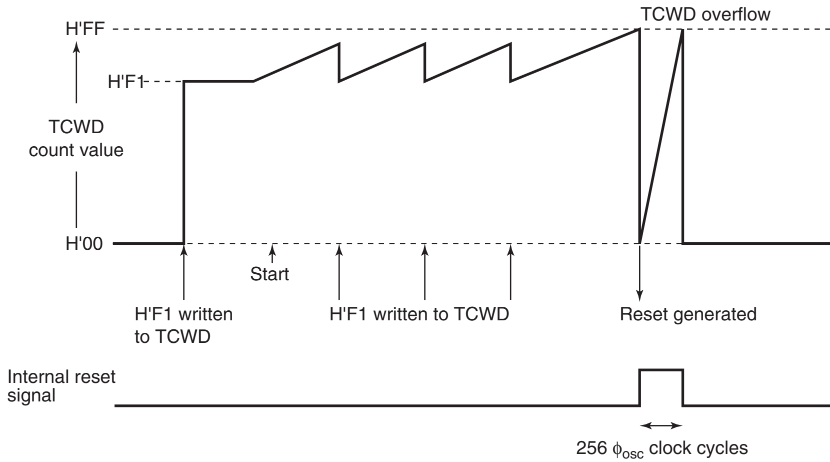


Figure 14.2 Watchdog Timer Operation Example

- Pulse division method for less ripple

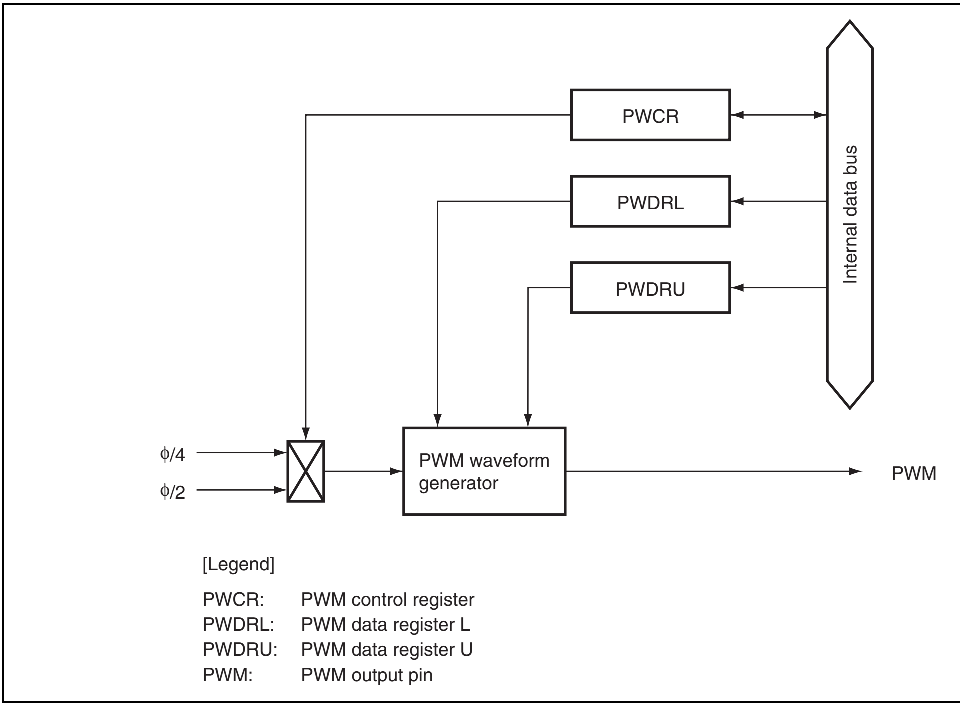


Figure 15.1 Block Diagram of 14-Bit PWM

The 14-bit PWM has the following registers.

- PWM control register (PWCR)
- PWM data register U (PWDRU)
- PWM data register L (PWDRL)

15.3.1 PWM Control Register (PWCR)

PWCR selects the conversion period.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 1	—	Reserved These bits are always read as 1, and cannot be
0	PWCR0	0	R/W	Clock Select 0: The input clock is φ/2 (tφ = 2/φ) — The conversion period is 16384/φ, with a modulation width of 1/φ 1: The input clock is φ/4 (tφ = 4/φ) — The conversion period is 32768/φ, with a modulation width of 2/φ

Legend: tφ: Period of PWM clock input

PWDRU and PWDRL are initialized to H'C000.

15.4 Operation

When using the 14-bit PWM, set the registers in this sequence:

1. Set the PWM bit in the port mode register 1 (PMR1) to set the P11/PWM pin to func
   PWM output pin.
2. Set the PWCR0 bit in PWCR to select a conversion period of either.
3. Set the output waveform data in PWDRU and PWDRL. Be sure to write byte data fi
   PWDRL and then to PWDRU. When the data is written in PWDRU, the contents of
   registers are latched in the PWM waveform generator, and the PWM waveform gene
   data is updated in synchronization with internal signals.

One conversion period consists of 64 pulses, as shown in figure 15.2. The total high-lev during this period (T_H) corresponds to the data in PWDRU and PWDRL. This relation c expressed as follows:

$$T_H = (\text{data value in PWDRU and PWDRL} + 64) \times t\phi/2$$

where $t\phi$ is the period of PWM clock input: $2/\phi$ (bit PWCR0 = 0) or $4/\phi$ (bit PWCR0 = 1
If the data value in PWDRU and PWDRL is from H'FFC0 to H'FFFF, the PWM output
When the data value is H'C000, T_H is calculated as follows:

$$T_H = 64 \times t\phi/2 = 32\ t\phi$$

t f1 = t f2 = t f3 = ··· = t f64

Figure 15.2 Waveform Output by 14-Bit PWM

SCI3. Since pin functions are identical for each of the two channels (SCI3 and SCI3_2),
explanations are not given in this section.

16.1 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

			TDR	H'FFAB
			SSR	H'FFAC
			RDR	H'FFAD
			RSR	—
			TSR	—
Channel 2	SCI3_2	SCK3_2 RXD_2 TXD_2	SMR_2	H'F740
			BRR_2	H'F741
			SCR3_2	H'F742
			TDR_2	H'F743
			SSR_2	H'F744
			RDR_2	H'F745
			RSR_2	—
			TSR_2	—

Note: * The channel 1 of the SCI3 is used in on-board programming mode by boot mo

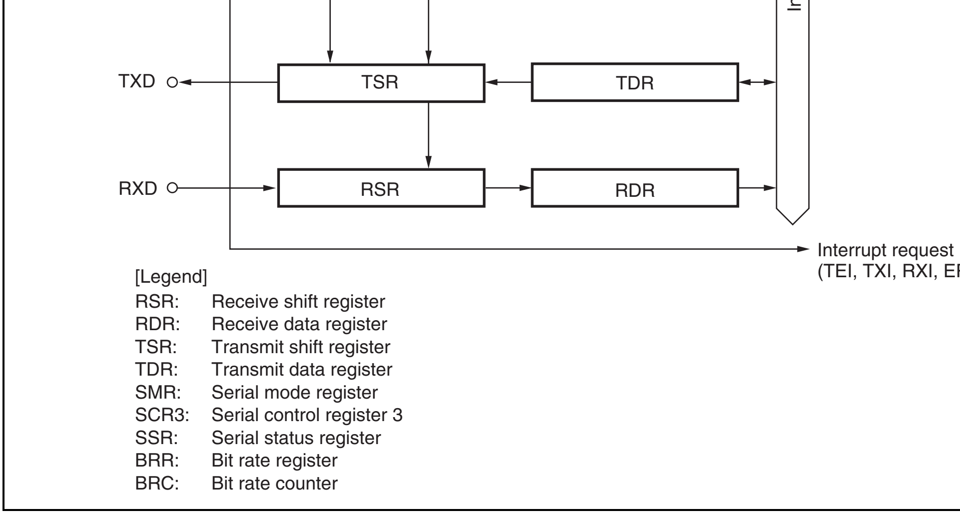


Figure 16.1 Block Diagram of SCI3

16.3 Register Descriptions

The SCI3 has the following registers for each channel.

- 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 r
operations are possible. After confirming that the RDRF bit in SSR is set to 1, read RDF
once. RDR cannot be written to by the CPU. RDR is initialized to H'00.

16.3.3 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the transfers transmit data from TDR to TSR automatically, then sends the data that starts fr LSB to the TXD pin. TSR cannot be directly accessed by the CPU.

16.3.4 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for transmission. When the SCI3 detects that TS empty, it transfers the transmit data written in TDR to TSR and starts transmission. The buffered structure of TDR and TSR enables continuous serial transmission. If the next tr data has already been written to TDR during transmission of one-frame data, the SCI3 tr 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. TI initialized to H'FF.

6	CHR	0	R/W	Character Length (enabled only in asynchronous 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 mo When this bit is set to 1, the parity bit is added to data before transmission, and the parity bit is che reception.
4	PM	0	R/W	Parity Mode (enabled only when the PE bit is 1 i asynchronous mode) 0: Selects even parity. 1: Selects odd parity.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous r Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits For reception, only the first stop bit is checked, r of the value in the bit. If the second stop bit is 0, treated as the start bit of the next transmit charac
2	MP	0	R/W	Multiprocessor Mode When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit a bit settings are invalid in multiprocessor mode. In synchronous mode, clear this bit to 0.

and the baud rate, see section 16.3.8, Bit Rate
(BRR). n is the decimal representation of the va
BRR (see section 16.3.8, Bit Rate Register (BR

16.3.6 Serial Control Register 3 (SCR3)

SCR3 is a register that enables or disables SCI3 transfer operations and interrupt request
also used to select the transfer clock source. For details on interrupt requests, refer to sec
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 reque enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt r are enabled.
5	TE	0	R/W	Transmit Enable When this bit s 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, TEI interrupt request is
1	CKE1	0	R/W	Clock Enable 0 and 1
0	CKE0	0	R/W	Selects the clock source. • Asynchronous mode 00: On-chip baud rate generator 01: On-chip 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: On-chip clock (SCK3 pin functions as clock o 01: Reserved 10: External clock (SCK3 pin functions as clock i 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 TD • When the transmit data is written to TDR
6	RDRF	0	R/W	Receive Data Register Full Indicates that the received data is stored in RD [Setting condition] • When serial reception ends normally and re is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RD • 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 OEF
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 b frame serial transmit character [Clearing conditions] • When 0 is written to TDRE after reading TDF • When the transmit data is written to TDR
1	MPBR	0	R	Multiprocessor Bit Receive MPBR stores the multiprocessor bit in the receiv character data. When the RE bit in SCR3 is clea its 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]

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

Legend B: Bit rate (bit/s)

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

ϕ: Operating frequency (MHz)

n: CSK1 and CSK0 settings in SMR ($0 \le n \le 3$)

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	—	—

	Operating Frequency φ (MHz)										
	3.6864			**4**			**4.9152**			**5**	
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	129
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	129
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

Legend:

—: A setting is available but error occurs

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

	Operating Frequency φ (MHz)										
	8			9.8304			10				
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	21
150	2	103	0.16	2	127	0.00	2	129	0.16	2	15
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77
600	1	103	0.16	1	127	0.00	1	129	0.16	1	15
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	15
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11
38400	0	6	-6.99	0	7	0.00	0	7	1.73	0	9

Legend:

—: A setting is available but error occurs.

1200	1	79	0.00	1	90	0.16	1	95	0.00	1	103
2400	0	159	0.00	0	181	0.16	0	191	0.00	0	207
4800	0	79	0.00	0	90	0.16	0	95	0.00	0	103
9600	0	39	0.00	0	45	–0.93	0	47	0.00	0	51
19200	0	19	0.00	0	22	–0.93	0	23	0.00	0	25
31250	0	11	2.40	0	13	0.00	0	14	–1.70	0	15
38400	0	9	0.00	—	—	—	0	11	0.00	0	12

	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.

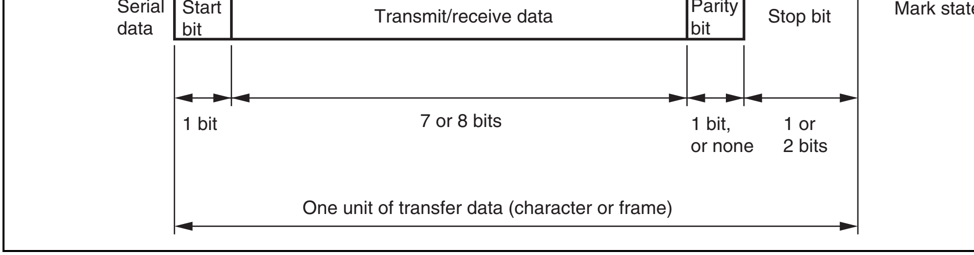


Figure 16.2 Data Format in Asynchronous Communication

16.4.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock the SCK3 pin can be selected as the SCI3's serial clock, according to the setting of the C SMR and the CKE0 and CKE1 bits in SCR3. When an external clock is input at the SCK 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 p frequency of the clock output in this case is equal to the bit rate, and the phase is such tha rising edge of the clock is in the middle of the transmit data, as shown in figure 16.3.

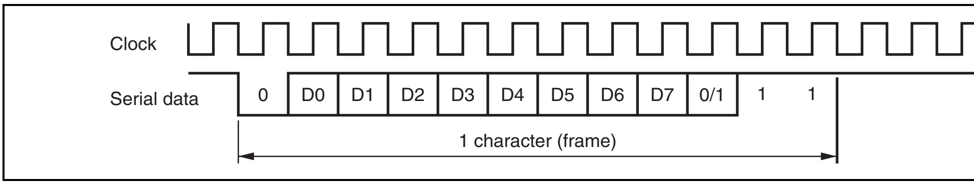
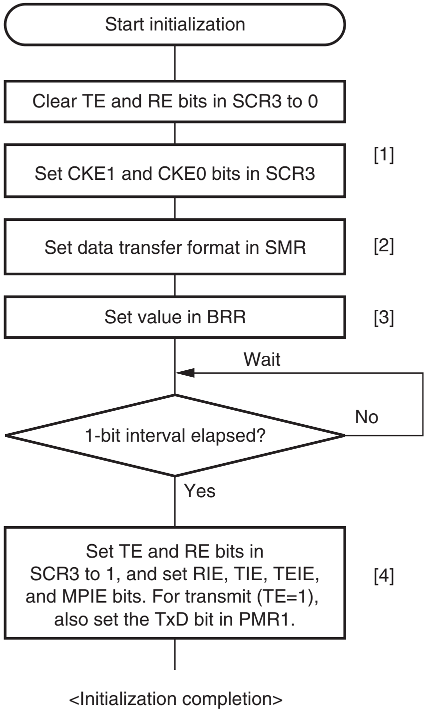


Figure 16.3 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode) (Example with 8-Bit Data, Parity, Two Stop Bits)



[1] Set the clock selection in SCR3. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock output is selected in asynchronous mode, clock is output immediately after CKE1 and CKE0 settings are made. When the clock output is selected at reception in 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 16.4 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, an 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 t state" is entered, in which 1 is output. If the TEIE bit in SCR3 is set to 1 at this time, a interrupt request is generated.
6. Figure 16.6 shows a sample flowchart for transmission in asynchronous mode.

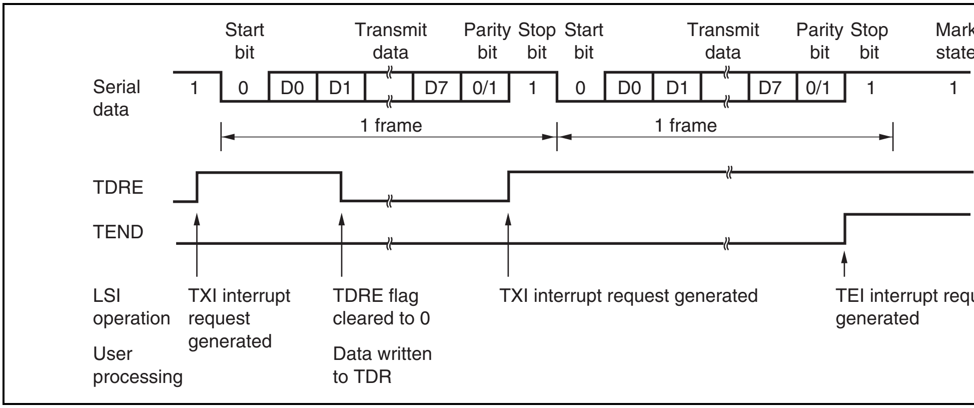


Figure 16.5 Example of SCI3 Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)

and PDR to 0, clear TxD in PMR
to 0, then clear the TE bit in SCR
to 0.

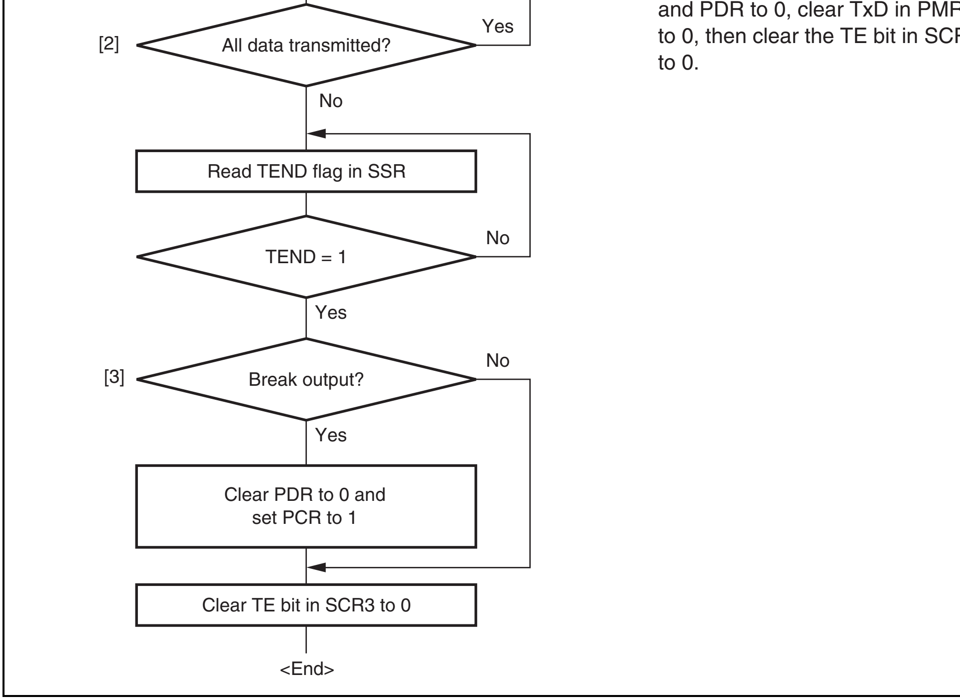


Figure 16.6 Sample Serial Transmission Data Flowchart (Asynchronous Mo

3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferr
RDR. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is genera

4. If a framing error is detected (when the stop bit is 0), the FER bit in SSR is set to 1 an data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an ERI inter request is generated.

5. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive da transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt re generated. Continuous reception is possible because the RXI interrupt routine reads th data transferred to RDR before reception of the next receive data has been completed.

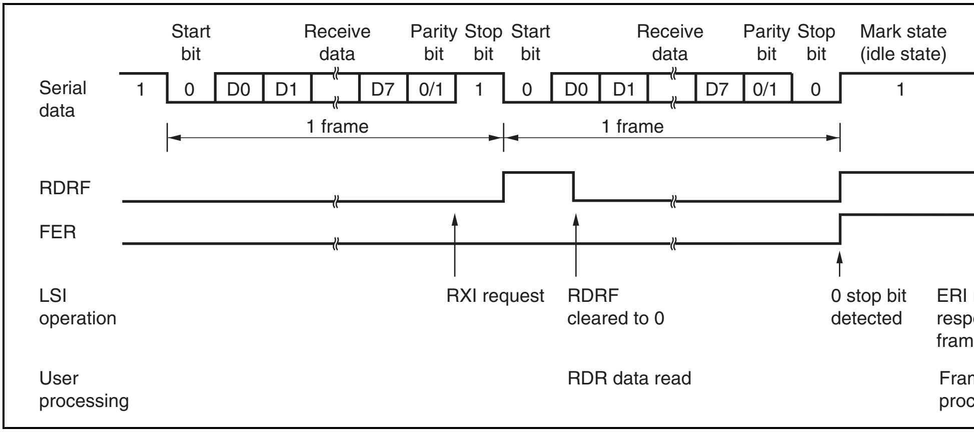
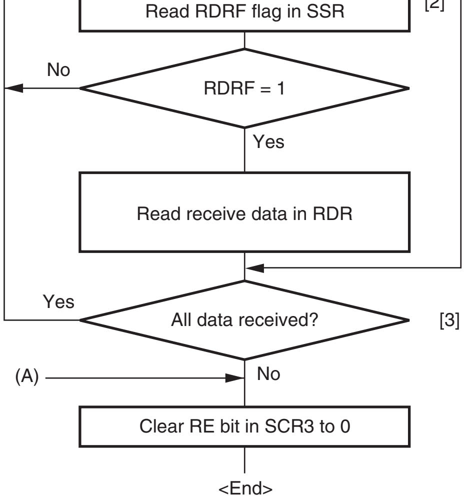


Figure 16.7 Example of SCI3 Reception in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)

Table 16.6 shows the states of the SSR status flags and receive data handling when a rece is detected. If a receive error is detected, the RDRF flag retains its state before receiving Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the

1	1	1	0	Lost	Overrun error + fram
1	1	0	1	Lost	Overrun error + parit
0	0	1	1	Transferred to RDR	Framing error + parit
1	1	1	1	Lost	Overrun error + fram parity error

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



the error. After performing the appropriate error processing, en that the OER, PER, and FER fla all cleared to 0. Reception cann resumed if any of these flags are 1. In the case of a framing error break can be detected by readin value of the input port correspon the RxD pin.

Figure 16.8 Sample Serial Reception Data Flowchart (Asynchronous Mode)

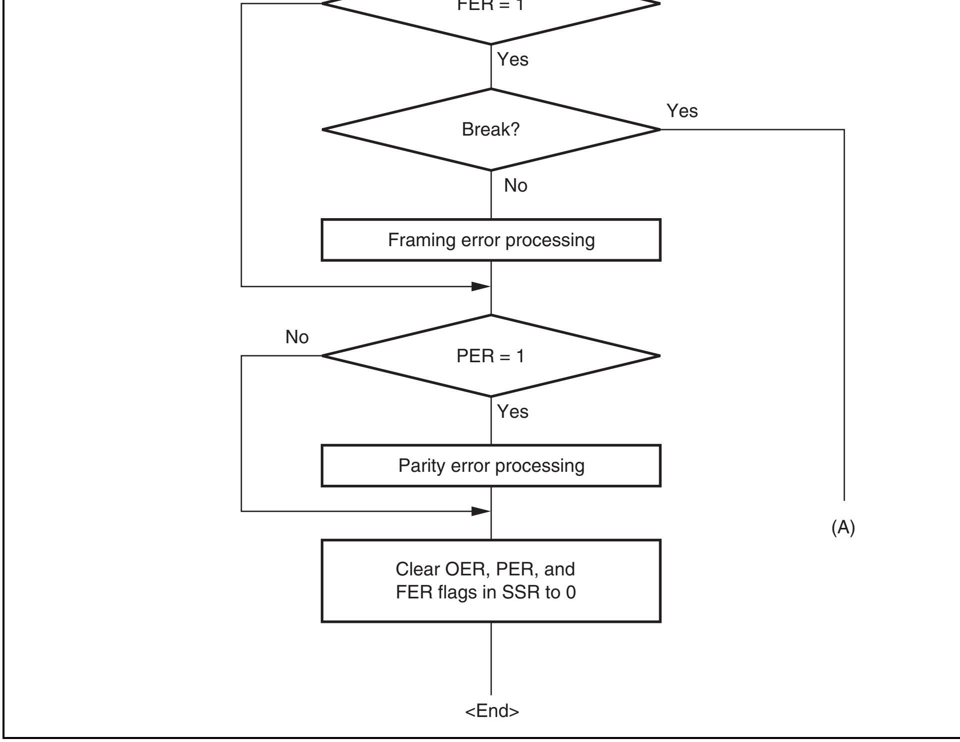


Figure 16.8 Sample Serial Reception Data Flowchart (Asynchronous Mode)

duplex communication through the use of a common clock. Both the transmitter and the r
also have a double-buffered structure, so data can be read or written during transmission o
reception, enabling continuous data transfer.

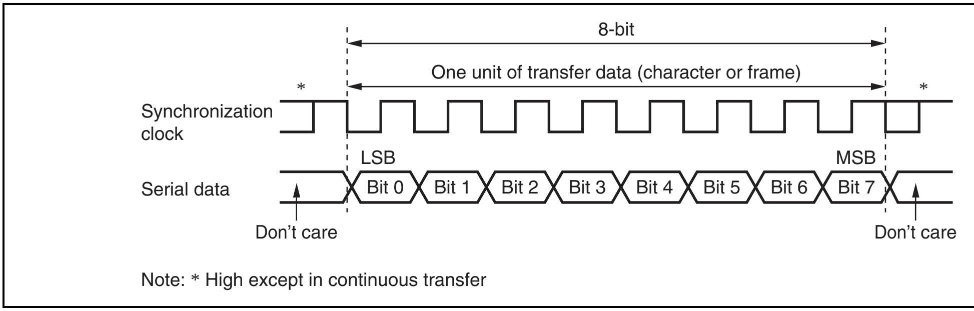


Figure 16.9 Data Format in Clocked Synchronous Communication

16.5.1 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 th bit in SMR and CKE0 and CKE1 bits in SCR3. When the SCI3 is operated on an internal the synchronization clock is output from the SCK3 pin. Eight synchronization clock pulse output in the transfer of one character, and when no transfer is performed the clock is fixe

16.5.2 SCI3 Initialization

Before transmitting and receiving data, the SCI3 should be initialized as described in a sa flowchart in figure 16.4.

mode has been specified, and synchronized with the input clock when use of an exte
has been specified. Serial data is transmitted sequentially from the LSB (bit 0), from
pin.

4. The SCI3 checks the TDRE flag at the timing for sending the MSB (bit 7).
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial tra of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TDRE flag ma output state of the last bit. If the TEIE bit in SCR3 is set to 1 at this time, a TEI inter request is generated.
7. The SCK3 pin is fixed high at the end of transmission.

Figure 16.11 shows a sample flow chart for serial data transmission. Even if the TDRE f 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 16.10 Example of SCI3 Transmission in Clocked Synchronous Mod

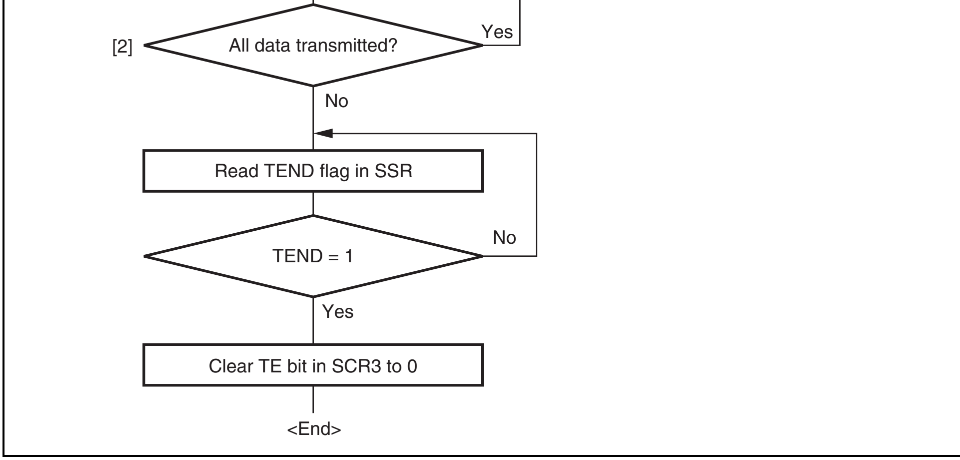


Figure 16.11 Sample Serial Transmission Flowchart (Clocked Synchronous M

time, an ERI interrupt request is generated, receive data is not transferred to RDR, and RDRF flag remains to be set to 1.

4. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive da transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt req generated.

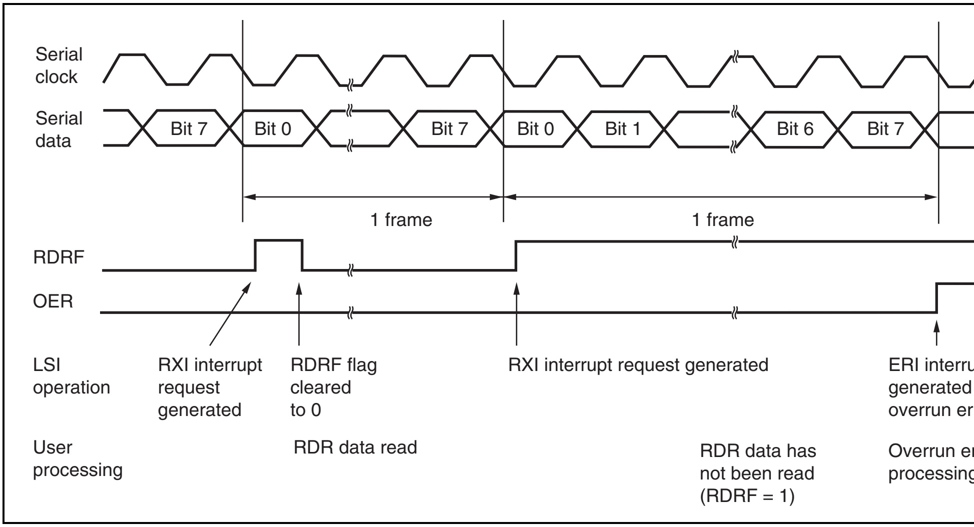
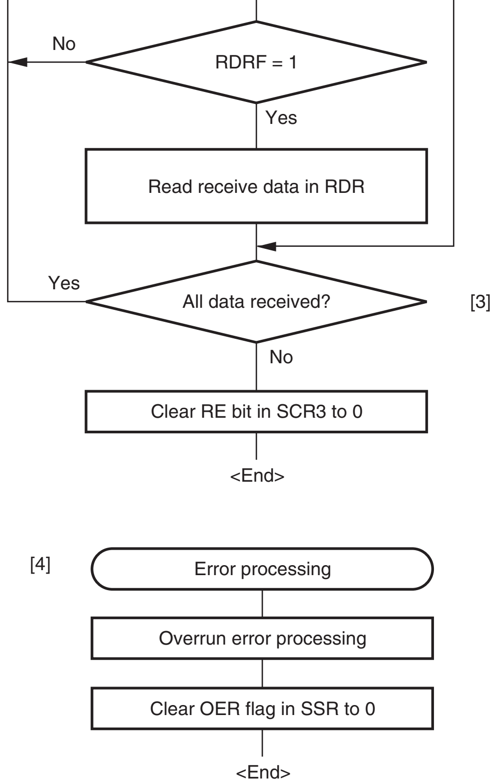


Figure 16.12 Example of SCI3 Reception in Clocked Synchronous Mode

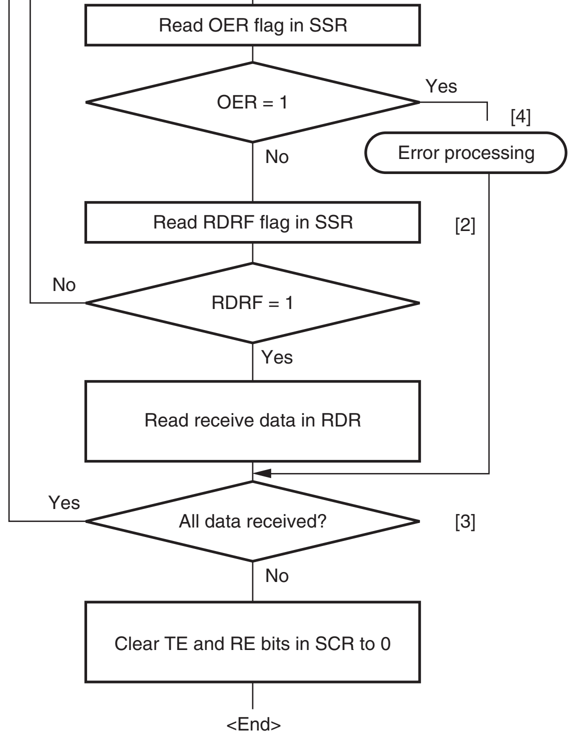
Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the FER, PER, and RDRF bits to 0 before resuming reception. Figure 16.13 shows a sample chart for serial data reception.

cleared to 0.



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

Figure 16.13 Sample Serial Reception Flowchart (Clocked Synchronous Mo



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

[4] If an overrun error occurs, read
OER flag in SSR, and after
performing the appropriate erro
processing, clear the OER flag
Transmission/reception cannot
resumed if the OER flag is set
For overrun error processing, s
figure 16.13.

Figure 16.14 Sample Flowchart of Simultaneous Serial Transmit and Receive Op
(Clocked Synchronous Mode)

cycle is a data transmission cycle. Figure 16.13 shows an example of inter-processor
communication using the multiprocessor format. The transmitting station first sends the I
of the receiving station with which it wants to perform serial communication as data with
multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit a
When data with a 1 multiprocessor bit is received, the receiving station compares that dat
own ID. The station whose ID matches then receives the data sent next. Stations whose II
match continue to skip data until data with a 1 multiprocessor bit is again received.

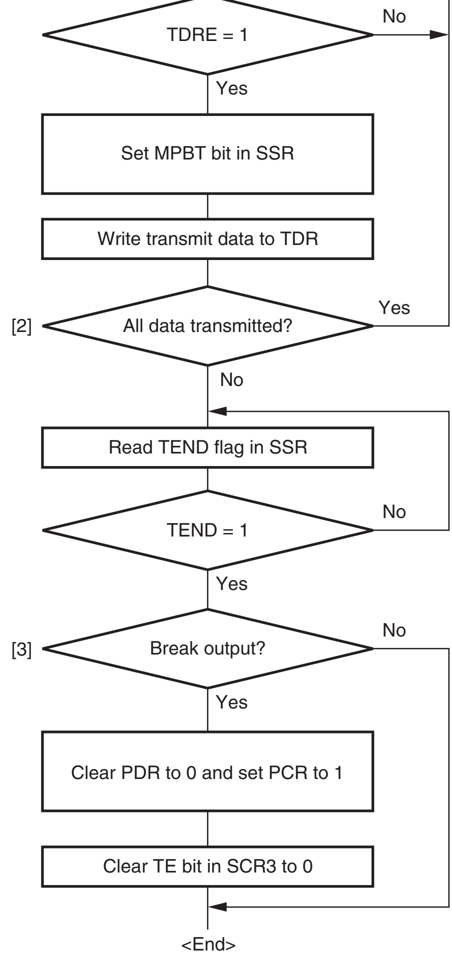
The SCI3 uses the MPIE bit in SCR3 to implement this function. When the MPIE bit is se
transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status
RDRF, FER, and OER, to 1, are inhibited until data with a 1 multiprocessor bit is receive
reception of a receive character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

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

specification receiving station specified by ID

[Legend]
MPB: Multiprocessor bit

Figure 16.15 Example of Inter-Processor Communication Using Multiprocessor (Transmission of Data H'AA to Receiving Station A)

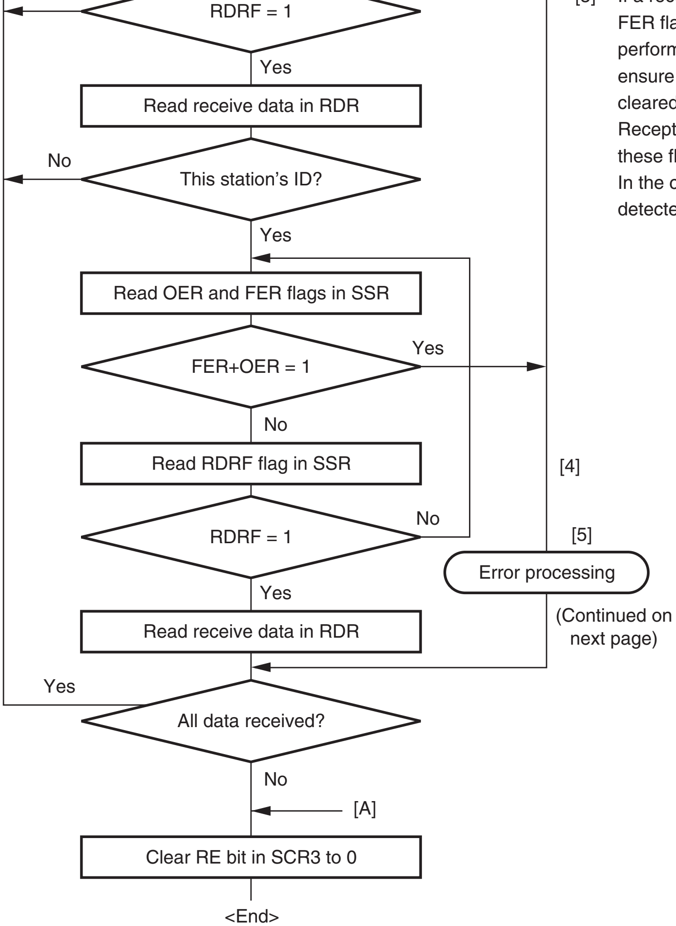


TDR, the TDRE flag is automatically cleared to 0.

[2] To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR. When data is written to TDR, the TDRE flag is automatically cleared to 0.

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

Figure 16.16 Sample Multiprocessor Serial Transmission Flowchart



[3] If a receive error occurs, read the OE
FER flags in SSR to identify the error.
performing the appropriate error proce
ensure that the OER and FER flags a
cleared to 0.
Reception cannot be resumed if eithe
these flags is set to 1.
In the case of a framing error, a break
detected by reading the RxD pin value

Figure 16.17 Sample Multiprocessor Serial Reception Flowchart (1)

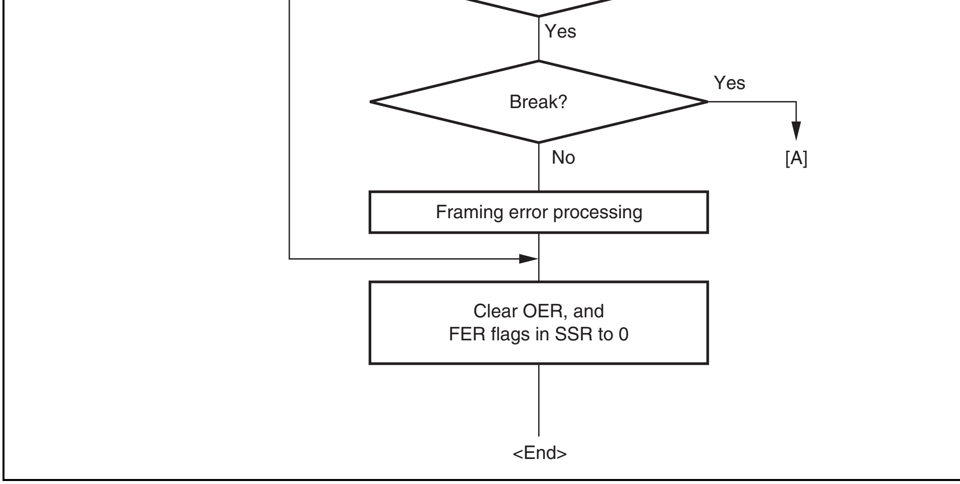
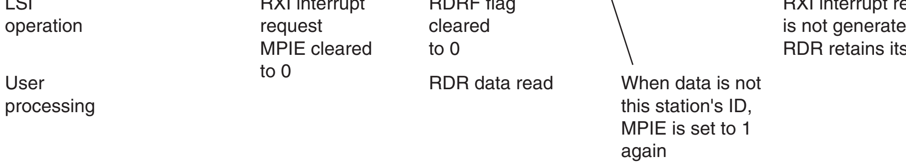
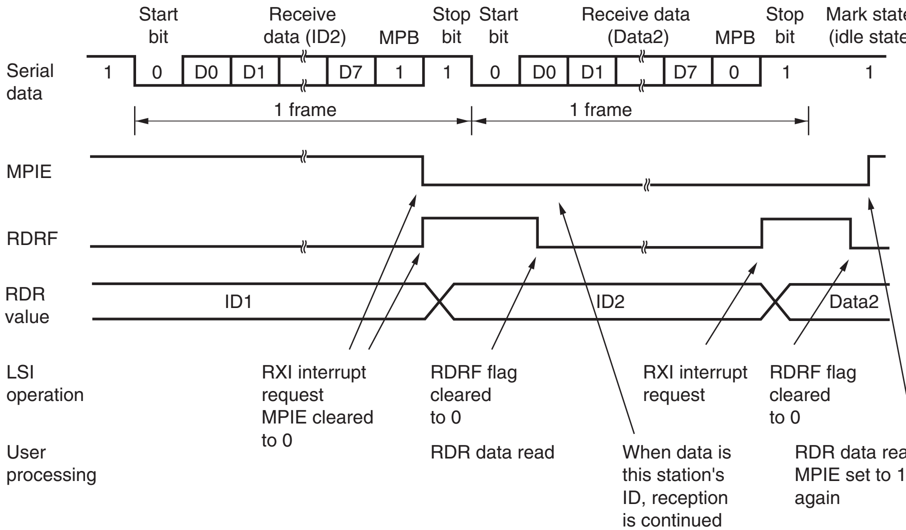


Figure 16.17 Sample Multiprocessor Serial Reception Flowchart (2)



(a) When data does not match this receiver's ID



(b) When data matches this receiver's ID

Figure 16.18 Example of SCI3 Reception Using Multiprocessor Format (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 transferring the transmit data to TDR, a TXI interrupt request is generated even if the tra is not ready. The initial value of the TEND flag in SSR is 1. Thus, when the TEIE bit in 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 interru requests efficiently by transferring the transmit data to TDR in the interrupt routine. To generation of these interrupt requests (TXI and TEI), set the enable bits (TIE and TEIE) 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 le
determined by PCR and PDR. This can be used to set the TxD pin to mark state (high lev
send a break during serial data transmission. To maintain the communication line at mark
until TE is set to 1, set both PCR and PDR to 1. As TE is cleared to 0 at this point, the Tx
becomes an I/O port, and 1 is output from the TxD pin. To send a break during serial tran
first set PCR to 1 and clear PDR to 0, and then clear TE to 0. When TE is cleared to 0, the
transmitter is initialized regardless of the current transmission state, the TxD pin becomes
port, and 0 is output from the TxD pin.

16.8.3 Receive Error Flags and Transmit Operations (Clocked Synchronous Mod

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 is to 0.

... Formula (1)

Legend 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 allow
system design.

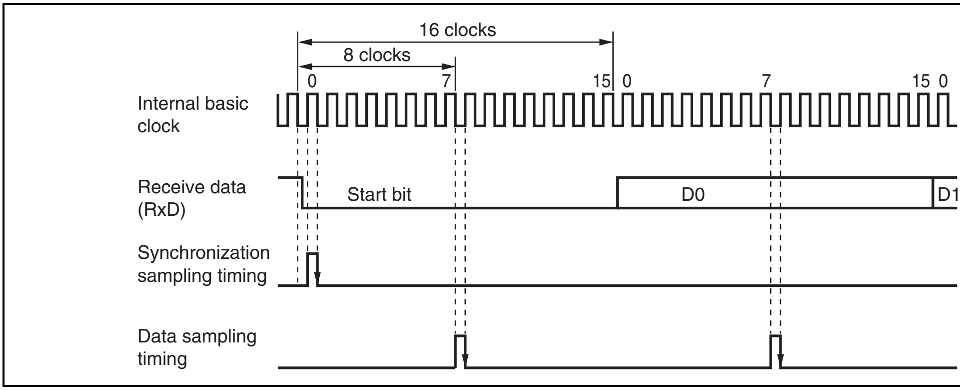


Figure 16.19 Receive Data Sampling Timing in Asynchronous Mode

17.1 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 are 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 c function is selected.

Clocked synchronous format

- Four interrupt sources

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

IFIIC10A_000020020200

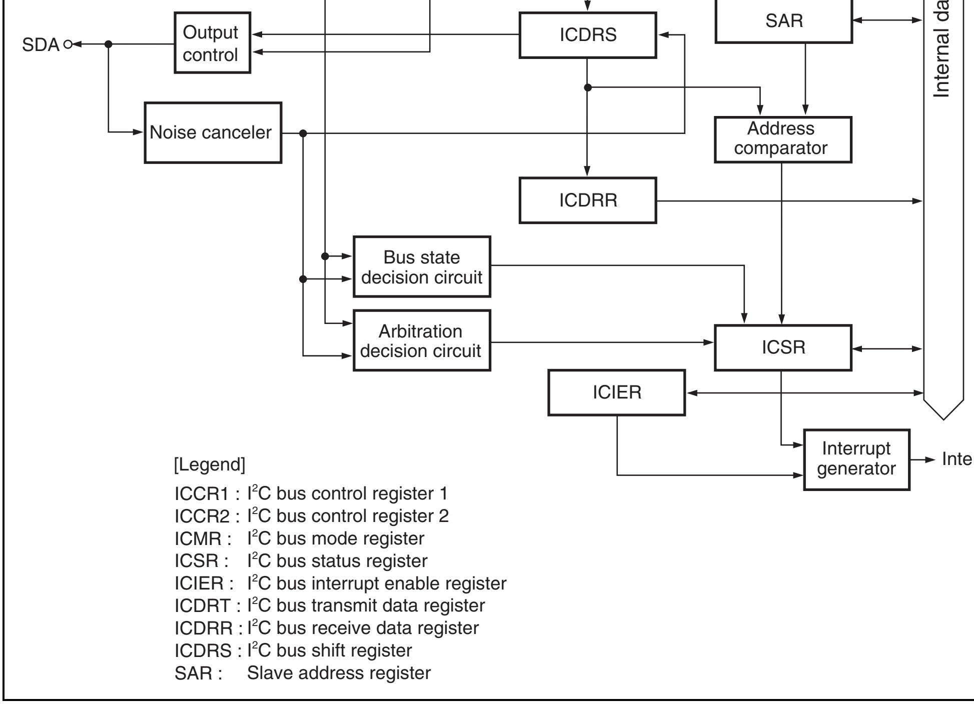


Figure 17.1 Block Diagram of I²C Bus Interface 2

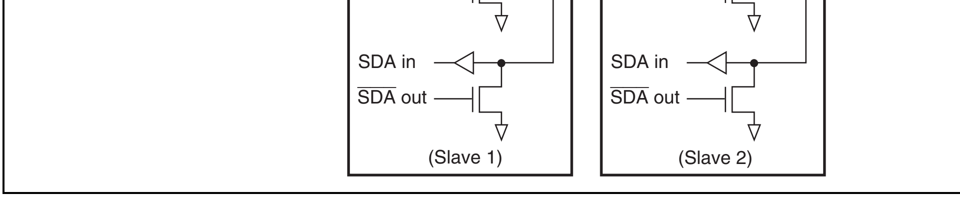


Figure 17.2 External Circuit Connections of I/O Pins

17.2 Input/Output Pins

Table 17.1 summarizes the input/output pins used by the I^{2}C bus interface 2.

Table 17.1 I^{2}C 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

17.3 Register Descriptions

The I^{2}C bus interface 2 has the following registers:

- I^{2}C bus control register 1 (ICCR1)
- I^{2}C bus control register 2 (ICCR2)
- I^{2}C bus mode register (ICMR)
- I^{2}C bus interrupt enable register (ICIER)
- I^{2}C bus status register (ICSR)
- I^{2}C bus slave address register (SAR)
- I^{2}C bus transmit data register (ICDRT)

				0: This module is halted. (SCL and SDA pins are port function.) 1: This bit is enabled for transfer operations. (SC SDA pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable This bit enables or disables the next operation w is 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 ar is lost, MST and TRS are both reset by hardware causing a transition to slave receive mode. Modi of the TRS bit should be made between transfer After data receive has been started in slave rece mode, when the first seven bits of the receive da with the slave address that is set to SAR and the bit is 1, TRS is automatically set to 1. If an overr occurs in master mode with the clock synchrono format, MST is cleared to 0 and slave receive mo entered. Operating modes are described below according and TRS combination. When clocked synchrono 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

17.3.3 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 t the clock for the final data bit, low period is exte two transfer clocks. If WAIT is cleared to 0, data 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 ser

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 next. When read, the remaining number of trans indicated. With the I²C bus format, the data is tra with one addition acknowledge bit. Bit BC2 to BC settings should be made during an interval betwe transfer frames. If bits BC2 to BC0 are set to a v other than 000, the setting should be made while pin is low. The value returns to 000 at the end of transfer, including the acknowledge bit. With the synchronous serial format, these bits should not modified.
0	BC0	0	R/W	I²C 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 of 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-transfer 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 format while the TDRE flag is 1 • When the final bit of transmit frame is sent 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 IC 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 the general call address or the first byte slave 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

				master
				[Setting conditions]
				• If the internal SDA and SDA pin disagree at SCL in master transmit mode
				• When the SDA pin outputs high in master m a start condition is detected
				• When the final bit is received with the clock synchronous format while RDRF = 1
				[Clearing condition]
				• When 0 is written in AL/OVE after reading A
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if the 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

17.3.6 Slave Address Register (SAR)

SAR selects the communication format and sets the slave address. When the chip is in sla with the I^2C bus format, if the upper 7 bits of SAR match the upper 7 bits of the first fram received after a start condition, the chip operates as the slave device.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA6 to SVA0	All 0	R/W	Slave Address 6 to 0 These bits set a unique address in bits SVA6 to differing form the addresses of other slave devic connected to the I^2C bus.
0	FS	0	R/W	Format Select 0: I^2C 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 receivec 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.

17.3.9 I^{2}C 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 transferrec ICDRS to ICDRR after data of one byte is received. This register cannot be read directly CPU.

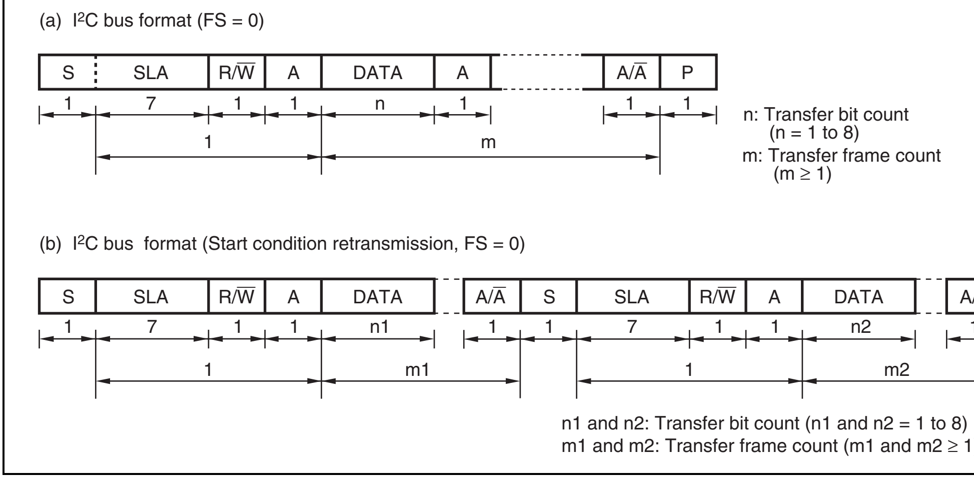


Figure 17.3 I²C Bus Formats

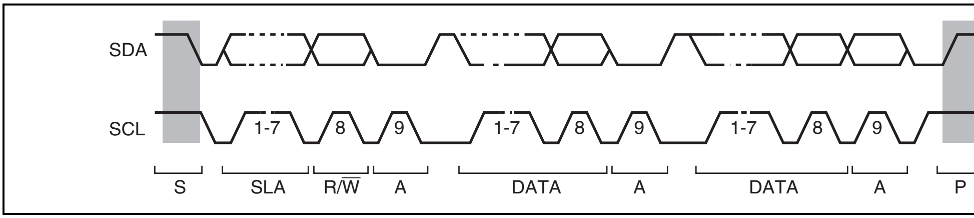


Figure 17.4 I²C Bus Timing

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/$\overline{\text{W}}$) 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) f
   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

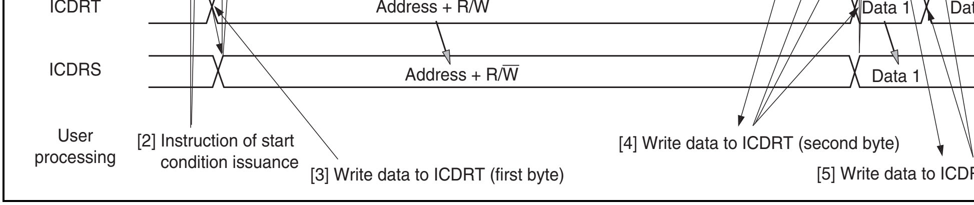


Figure 17.5 Master Transmit Mode Operation Timing (1)

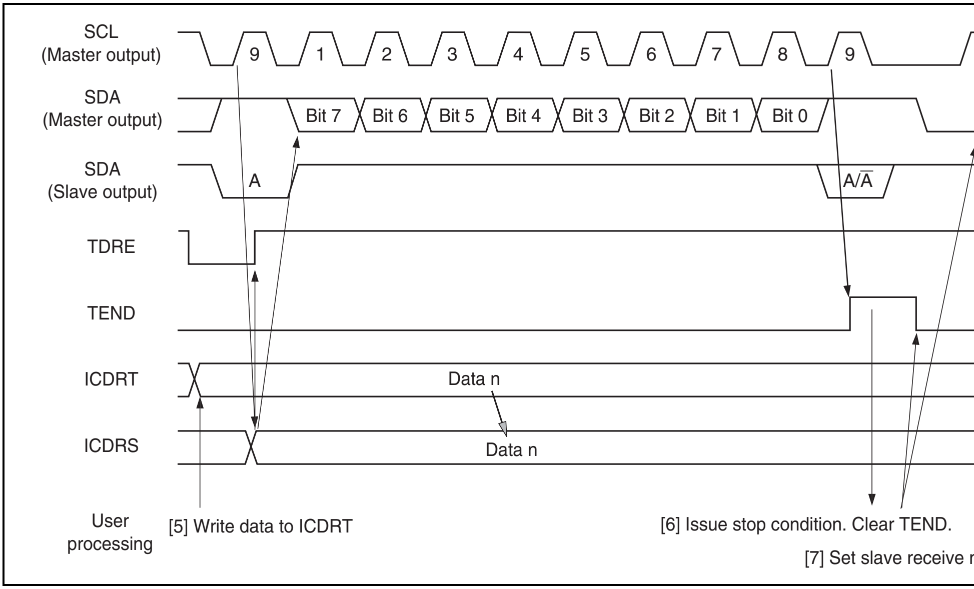


Figure 17.6 Master Transmit Mode Operation Timing (2)

and data received, in synchronization with the internal clock. The master device outp
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 is fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1 before reading 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.

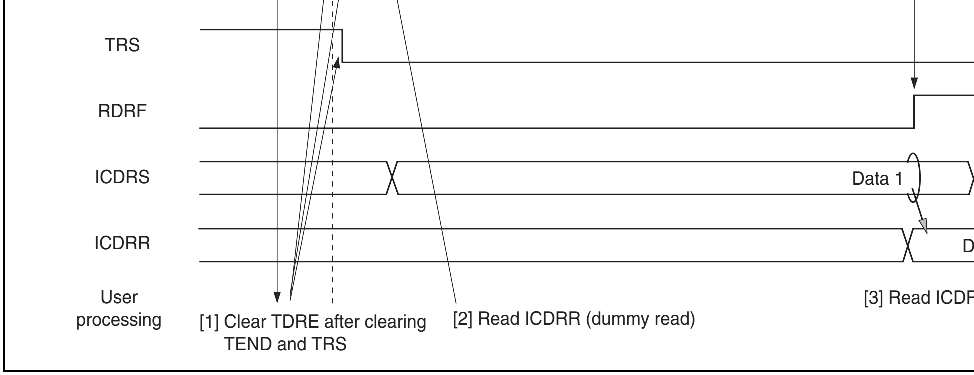


Figure 17.7   Master Receive Mode Operation Timing (1)

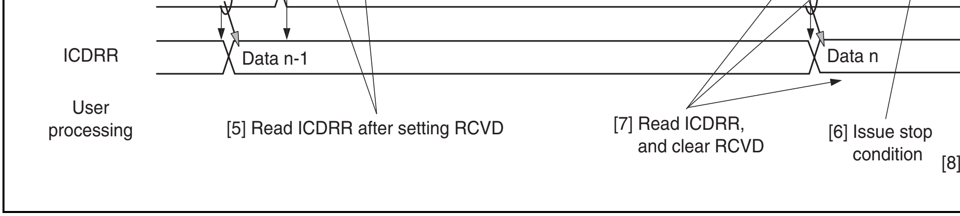


Figure 17.8 Master Receive Mode Operation Timing (2)

17.4.4 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 17.9 and 17.10.

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 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 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.

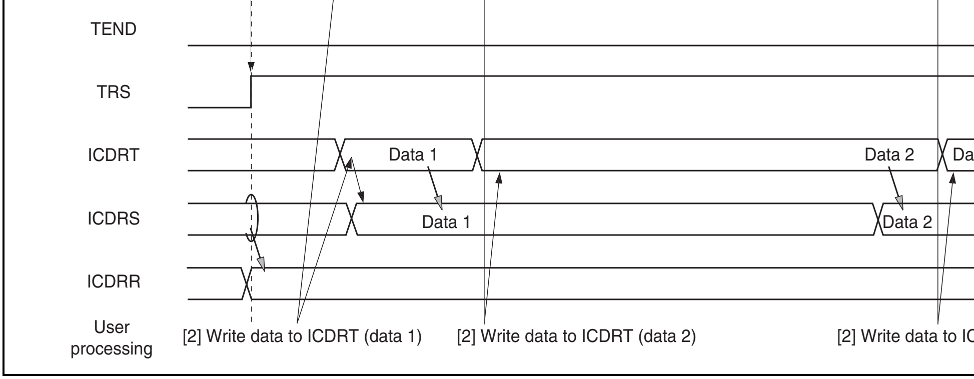


Figure 17.9 Slave Transmit Mode Operation Timing (1)

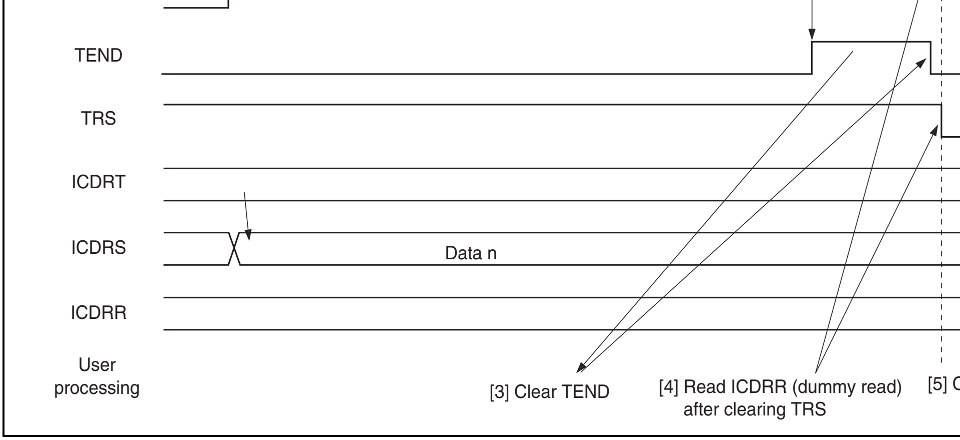


Figure 17.10 Slave Transmit Mode Operation Timing (2)

17.4.5 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 17.11 and 17.12. 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 con
   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/$\overline{\text{W}}$, it is not used.)

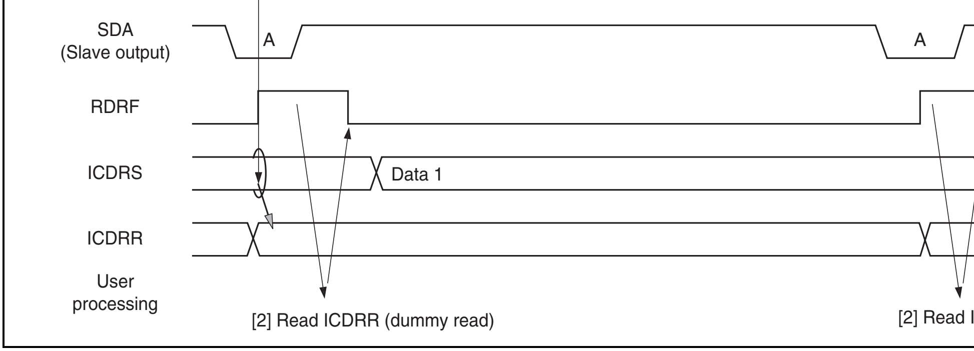


Figure 17.11 Slave Receive Mode Operation Timing (1)

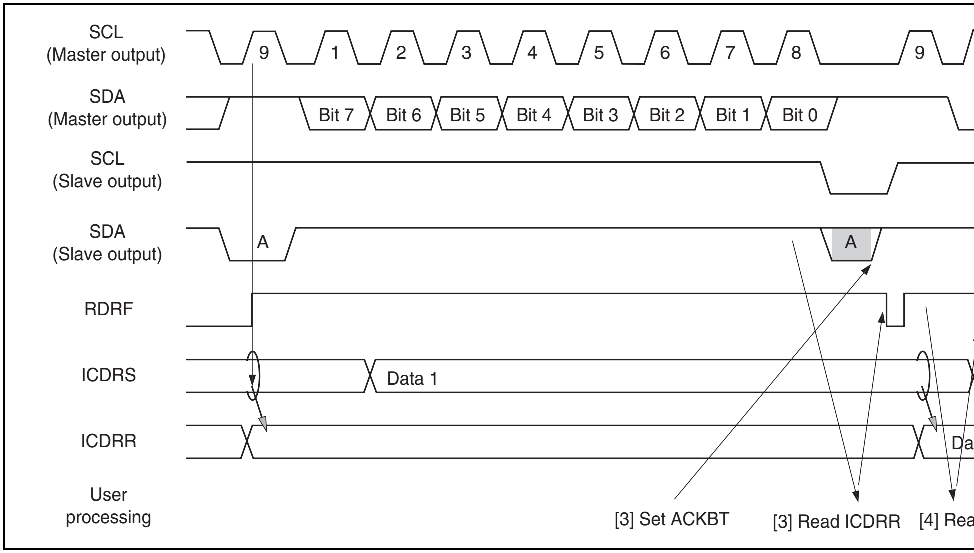


Figure 17.12 Slave Receive Mode Operation Timing (2)

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

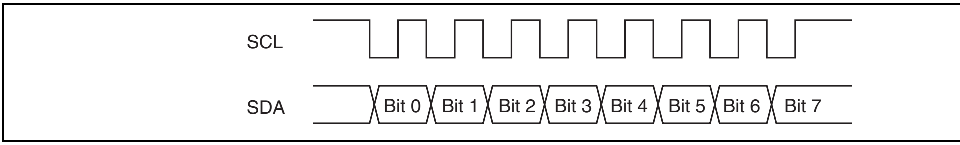


Figure 17.13 Clocked Synchronous Serial Transfer Format

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 17.14. 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.

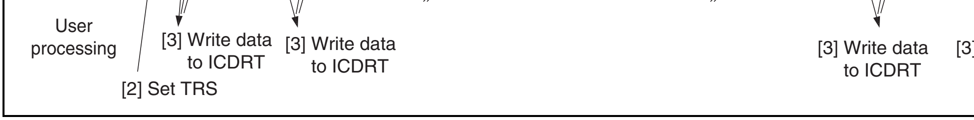


Figure 17.14 Transmit Mode Operation Timing

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 17.15. 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 anc 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.

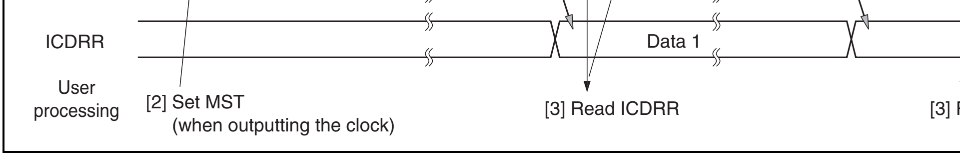


Figure 17.15 Receive Mode Operation Timing

17.4.7 Noise Canceler

The logic levels at the SCL and SDA pins are routed through noise cancelers before bei internally. Figure 17.16 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.

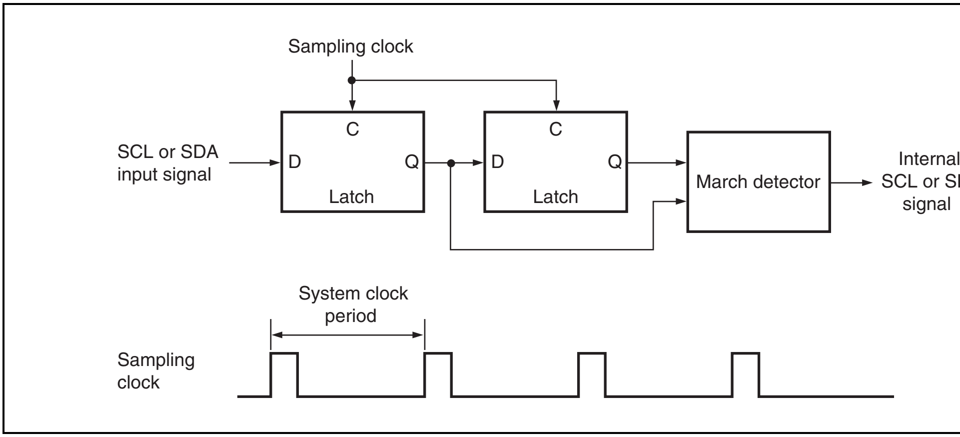
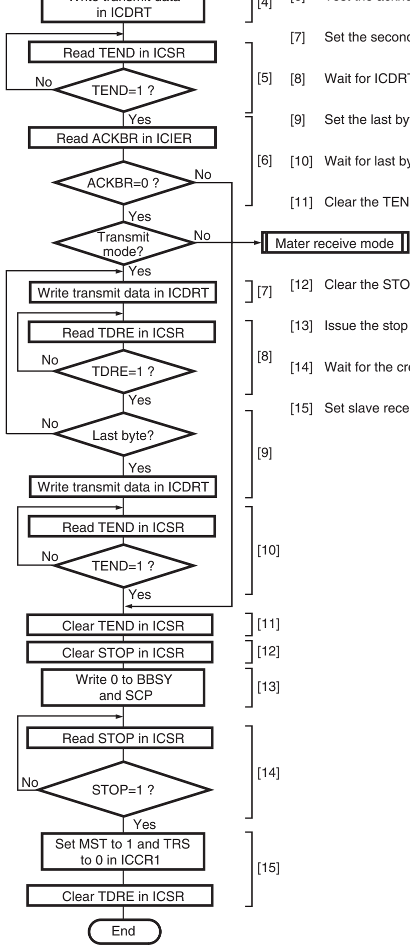


Figure 17.16 Block Diagram of Noise Conceler



[7] Set the second and subsequent bytes (except for the final byte) of transmit data.

[8] Wait for ICDRT empty.

[9] Set the last byte of transmit data.

[10] Wait for last byte to be transmitted.

[11] Clear the TEND flag.

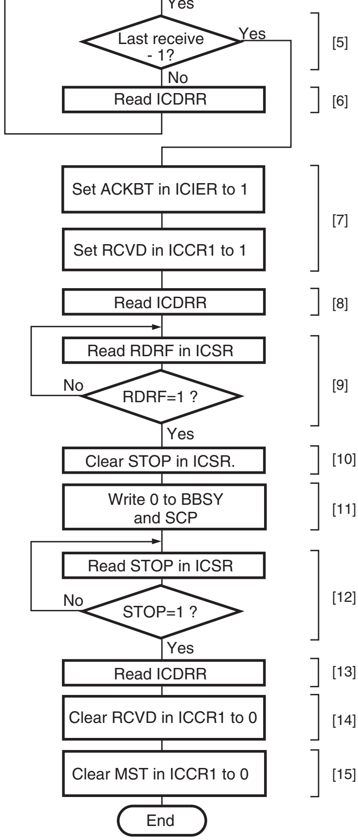
[12] Clear the STOP flag.

[13] Issue the stop condition.

[14] Wait for the creation of stop condition.

[15] Set slave receive mode. Clear TDRE.

Figure 17.17 Sample Flowchart for Master Transmit Mode



[9] Wait for the last byte to be receive.

[10] Clear the 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]
skipped after step [1], before jumping to st
The step [8] is dummy-read in ICDRR.

Figure 17.18 Sample Flowchart for Master Receive Mode

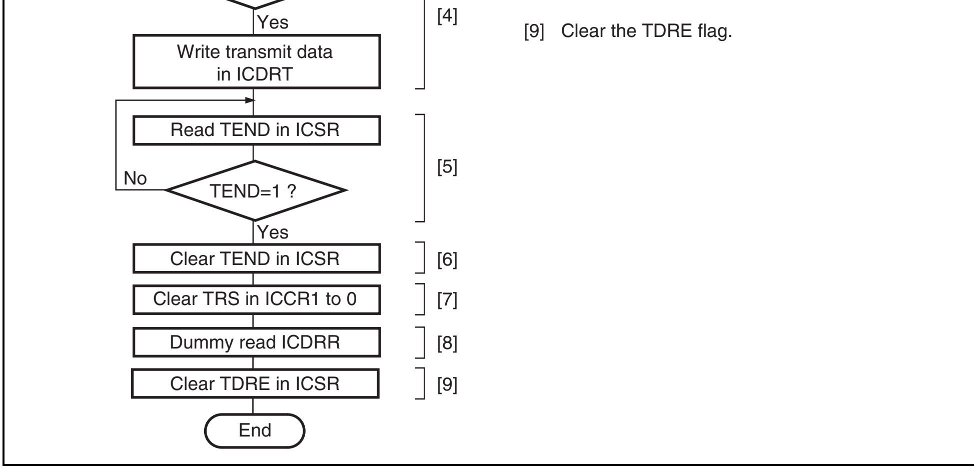
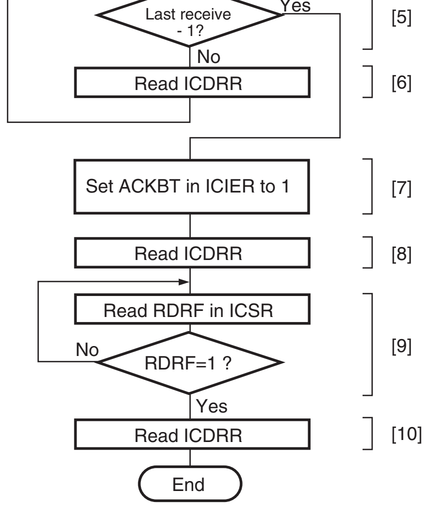


Figure 17.19 Sample Flowchart for Slave Transmit Mode



[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 17.20 Sample Flowchart for Slave Receive Mode

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 Error			○	○

When interrupt conditions described in table 17.3 are 1 and the I bit in CCR is 0, the CPU executes an interrupt exception processing. Interrupt sources should be cleared in the exce 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 17.21 shows the timing of the bit synchronous circuit and table 17.4 shows the ti
SCL output changes from low to Hi-Z then SCL is monitored.

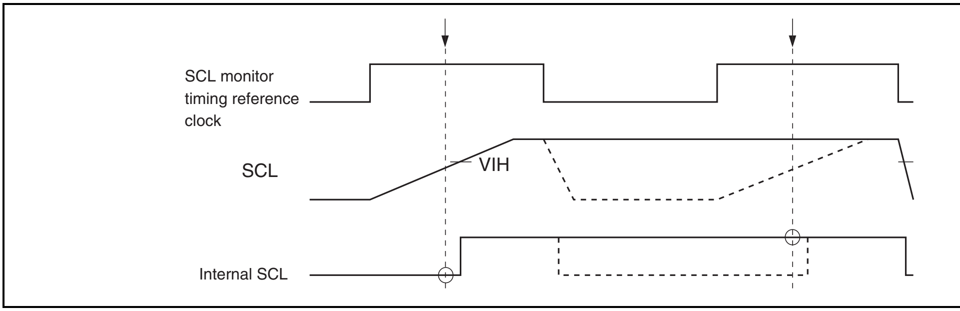


Figure 17.21 The Timing of the Bit Synchronous Circuit

Table 17.4 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 ar clocks, that is driven by the slave device

17.7.2 WAIT Setting in I^{2}C 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

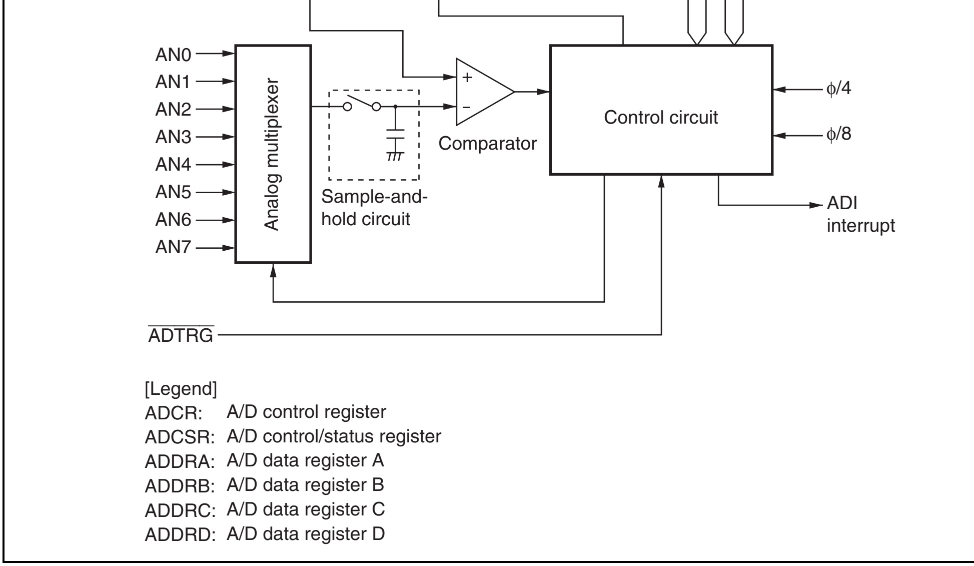


Figure 18.1 Block Diagram of A/D Converter

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 A/D conversion

18.3.1 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 inp
channel, are shown in table 18.2.

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 re
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 18.2 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

				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 en 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 sing this bit is cleared to 0 automatically when conve the specified channel is complete. In scan mode conversion continues sequentially on the specifi channels until this bit is cleared to 0 by software or a transition to standby mode.
4	SCAN	0	R/W	Scan Mode Selects single mode or scan mode as the A/D co 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 con time.

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

18.3.3 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 and t edge of the external trigger signal ($\overline{\text{ADTRG}}$) wher is set to 1. The selection between the falling edge and rising the external trigger pin ($\overline{\text{ADTRG}}$) conforms to the bit in the interrupt edge select register 2 (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/

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.

18.4.2 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 18.3 apply to the first conversion time. In the seco subsequent conversions, the conversion time is 128 states (fixed) when CKS = 0 and 66 s (fixed) when CKS = 1.

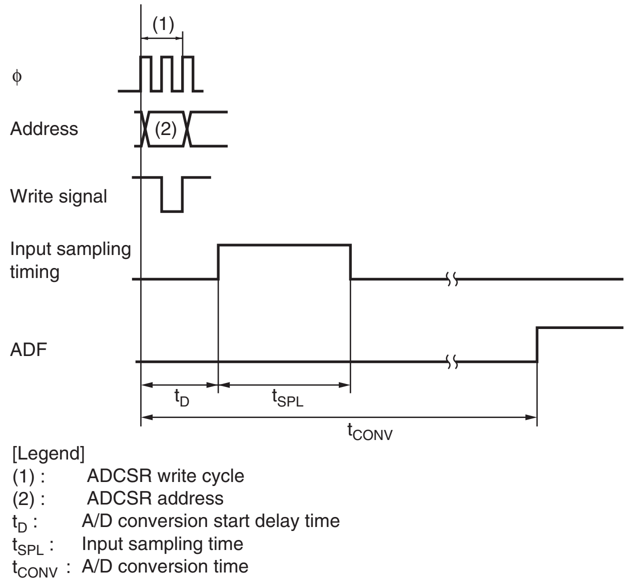


Figure 18.2 A/D Conversion Timing

18.4.4 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 bot
and scan modes, are the same as when the bit ADST has been set to 1 by software. Figu
shows the timing.

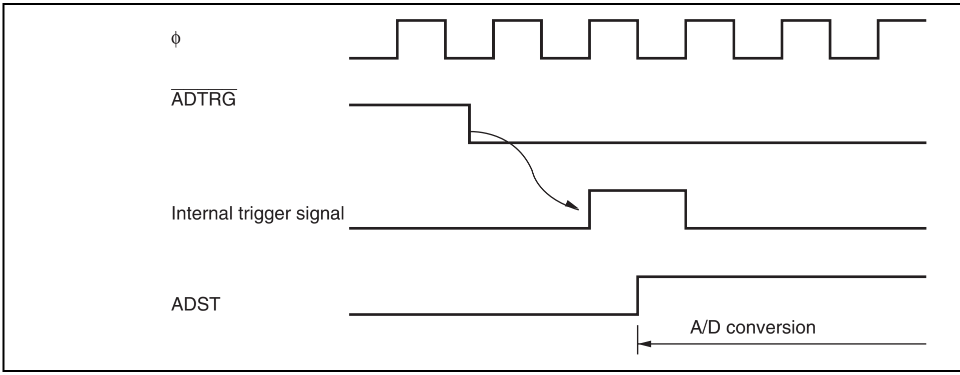


Figure 18.3 External Trigger Input Timing

when the digital output changes from the minimum voltage value 0000000000 to 000
(see figure 18.5).

- 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 18.5).

- 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.

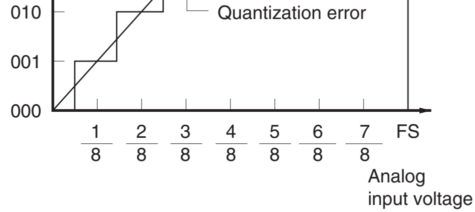


Figure 18.4 A/D Conversion Accuracy Definitions (1)

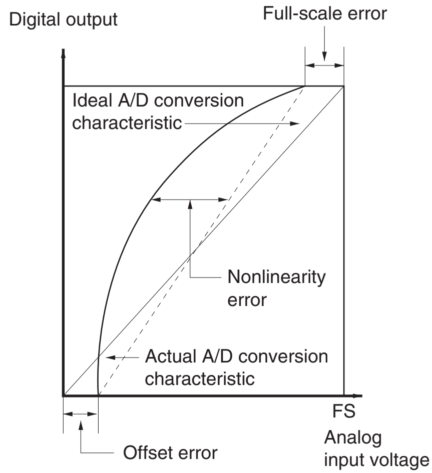


Figure 18.5 A/D Conversion Accuracy Definitions (2)

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 18.6). When converting a hig
analog signal or converting in scan mode, a low-impedance buffer should be inserted.

18.6.2 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.

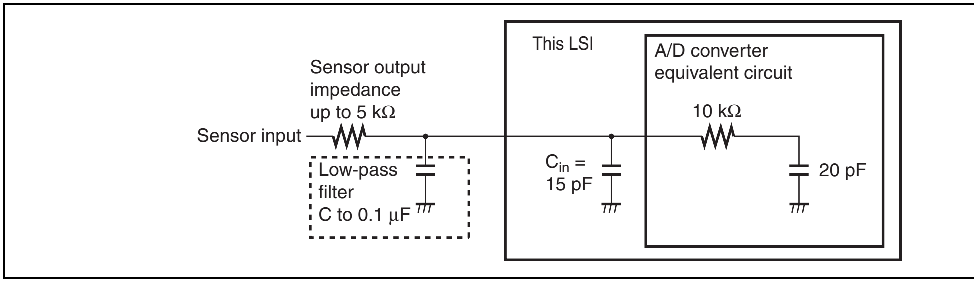


Figure 18.6 Analog Input Circuit Example

- 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

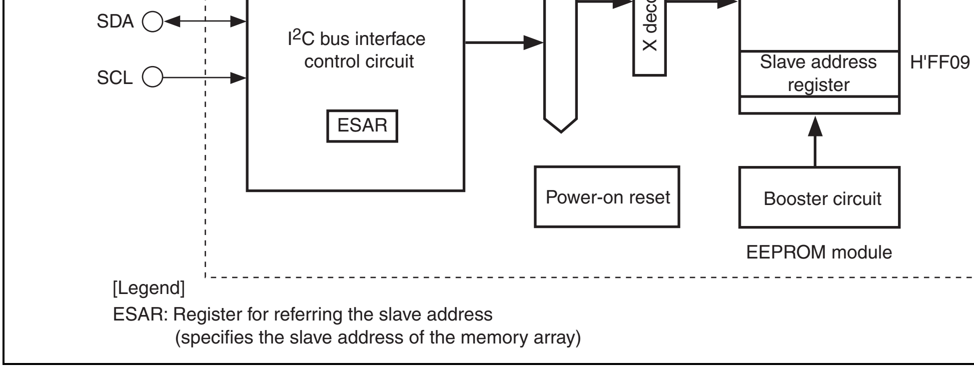


Figure 19.1 Block Diagram of EEPROM

			is open-drain driven structure of the I²C pin. proper resistor value for your system by con V_{OL}, I_{OL}, and the C_{IN} pin capacitance in sectio DC Characteristics and in section 23.2.3, AC Characteristics. Maximum clock frequency kHz.
Serial data pin	SDA	Input/Output	The SDA pin is bidirectional for serial data t The SDA pin needs to be pulled up by resis pin is open-drain driven structure. Use prop value for your system by considering V_{OL}, I_{OL} C_{IN} pin capacitance in section 23.2.2, DC Characteristics and in section 23.2.3, AC Characteristics. Except for a start condition stop condition which will be discussed later, to-low and low-to-high change of SDA input done during SCL low periods.

19.3 Register Description

The EEPROM has a following register.

- EEPROM key register (EKR)

19.3.1 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.

19.4.2 Bus Format and Timing

The I^2C bus format and the I^2C bus timing follow section 17.4.1, I^2C 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 c
   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 19.2.

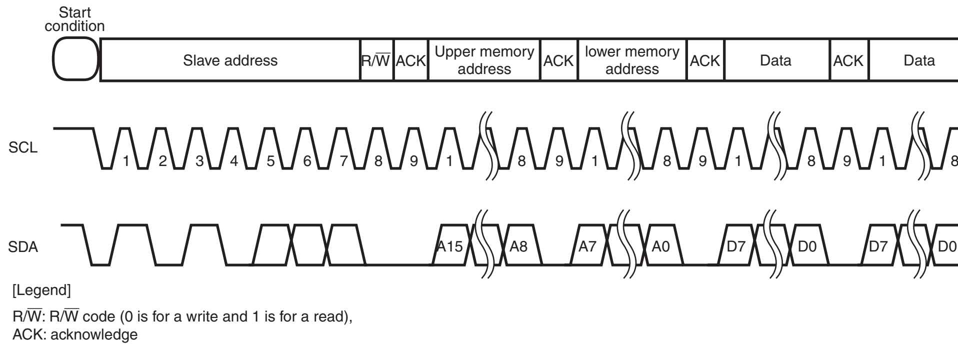


Figure 19.2 EEPROM Bus Format and Bus Timing

19.4.3 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.

19.4.6 Slave Addressing

The EEPROM device receives a 7-bit slave address and a 1-bit R/$\overline{\text{W}}$ 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 19.2. 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.

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 c
2	Slave address code A1	0	A1	The initial value can be c
1	Slave address code A0	0	A0	The initial value can be c

19.4.7 Write Operations

There are two types write operations; byte write operation and page write operation. To i the write operation, input 0 to R/$\overline{\text{W}}$ code following the slave address.

1. Byte Write

   A write operation requires an 8-bit data of a 7-bit slave address with R/$\overline{\text{W}}$ code = "0". the EEPROM sends acknowledgement "0" at the ninth bit. This enters the write mod two bytes of the memory address are received from the MSB side in the order of uppe lower. Upon receipt of one-byte memory address, the EEPROM sends acknowledgen and receives a following a one-byte write data. After receipt of write data, the EEPR(acknowledgement "0". If the EEPROM receives a stop condition, the EEPROM enter 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 comple the write cycle.

   The byte write operation is shown in figure 19.3.

Figure 19.3 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.

Addresses in the page are incremented at each receipt of the write data and the write
be input up to 8 bytes. If the LSB 3 bits (A2 to A0) in the EEPROM address reach t
address of the page, the address will roll over to the first address of the same page. W
address is rolled over, write data is received twice or more to the same address, howe
last received data is valid. At the receipt of the stop condition, write data reception i
terminated and the write operation is entered.

The page write operation is shown in figure 19.4.

Figure 19.4 Page Write Operation

19.4.8 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 an
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.

19.4.9 Read Operation

There are three read operations; current address read, random address read, and sequentia
Read operations are initiated in the same way as write operations with the exception of R

1. Current Address Read

   The internal address counter maintains the (n+1) address that is made by the last addr
   accessed during the last read or write operation, with incremented by one. Current ad
   read accesses the (n+1) address kept by the internal address counter.

   After receiving in the order of a start condition and the slave address + $R/\overline{W}$ code (R/V
   the EEPROM outputs the 1-byte data of the (n+1) address from the most significant b
   following acknowledgement "0". If the EEPROM receives in the order of acknowledg
   "1" and a following stop condition, the EEPROM stops the read operation and is turne
   standby state.

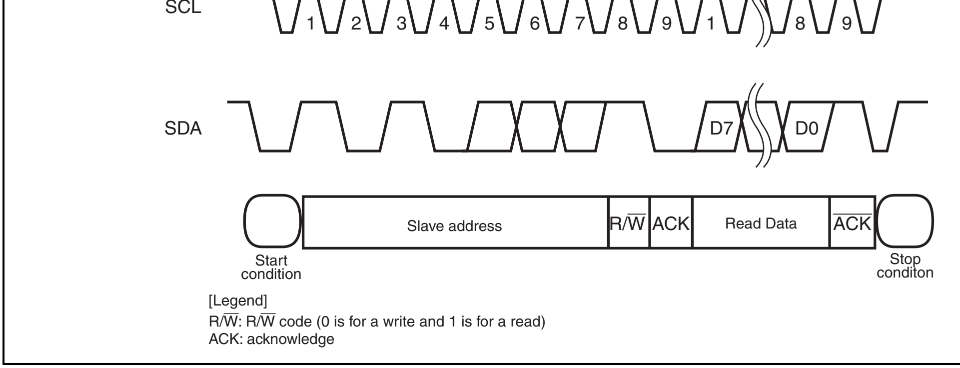


Figure 19.5 Current Address Read Operation

2. Random Address Read

This is a read operation with defined read address. A random address read requires a write to set read address. The EEPROM receives a start condition, slave address + R (R/$\overline{W}$ = 0), memory address (upper) and memory address (lower) sequentially. The outputs acknowledgement "0" after receiving memory address (lower) then enters a address read with receiving a start condition again. The EEPROM outputs the read address which was defined in the dummy write operation. After receiving acknowle "1" and a following stop condition, the EEPROM stops the random read operation a to a standby state.

The random address read operation is shown in figure 19.6.

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 19.7.

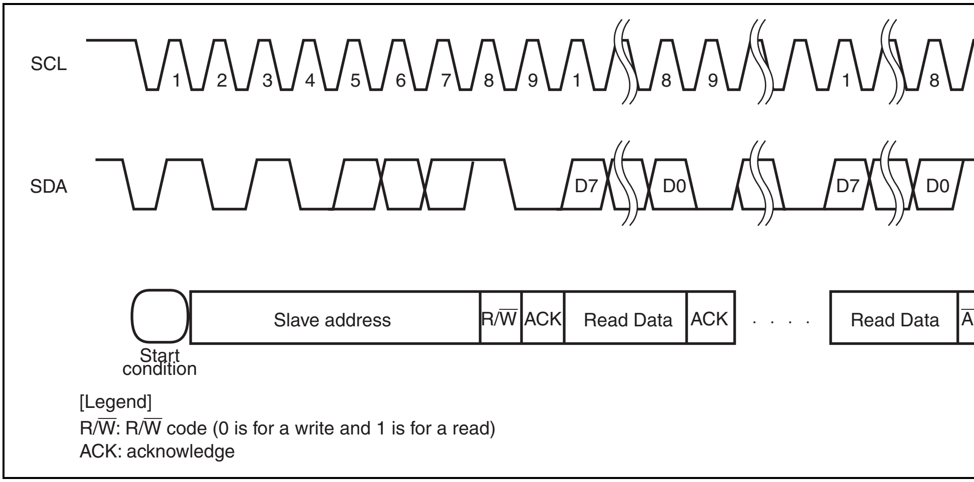


Figure 19.7 Sequential Read Operation (when current address read is used

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

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

19.5.2 Write/Erase Endurance

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

19.5.3 Noise Suppression Time

This EEPROM has a noise suppression function at SCL and SDA inputs, that cuts noise less than 50 ns. Be careful not to allow noise of width more than 50 ns because the nois 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 ac is automatically entered.

Figure 20.1 is a block diagram of the power-on reset circuit and the low-voltage detectio

20.1 Features

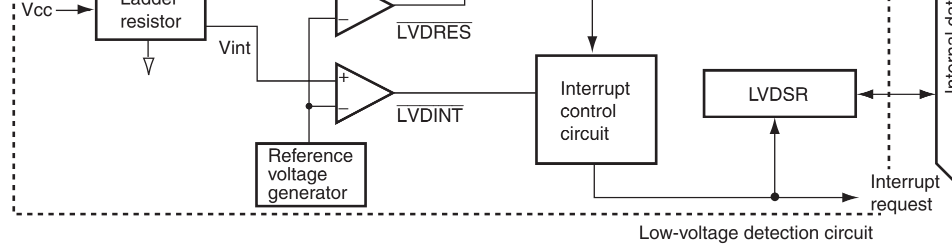
- Power-on reset circuit

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

20.2 Register Descriptions

The low-voltage detection circuit has the following registers.

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

20.2.1 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.

Bit	Bit Name	Initial Value	R/W	Description
				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.

20.2.2 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.

LSI,072 clock (φ) cycles. The noise cancellation circuit of approximately 100 ns is inco
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 forn

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

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

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.

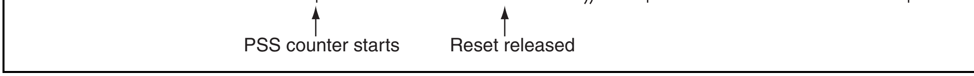


Figure 20.2 Operational Timing of Power-On Reset Circuit

20.3.2 Low-Voltage Detection Circuit

LVDR (Reset by Low Voltage Detect) Circuit:

Figure 20.3 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), the 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 rises 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.

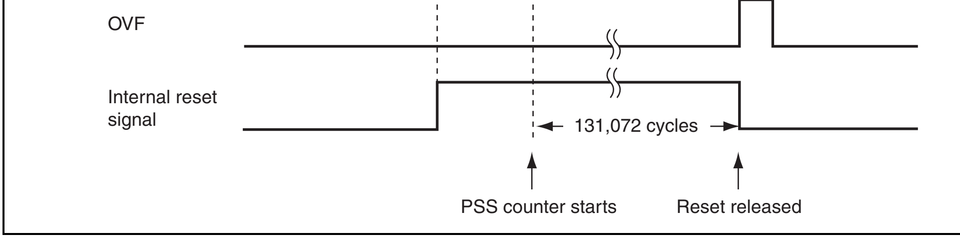


Figure 20.3 Operational Timing of LVDR Circuit

LVDI (Interrupt by Low Voltage Detect) Circuit:

Figure 20.4 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 cle
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

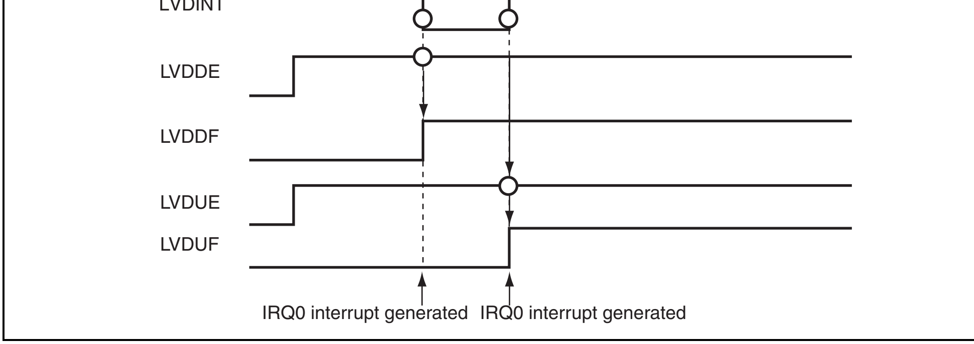


Figure 20.4 Operational Timing of LVDI Circuit

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

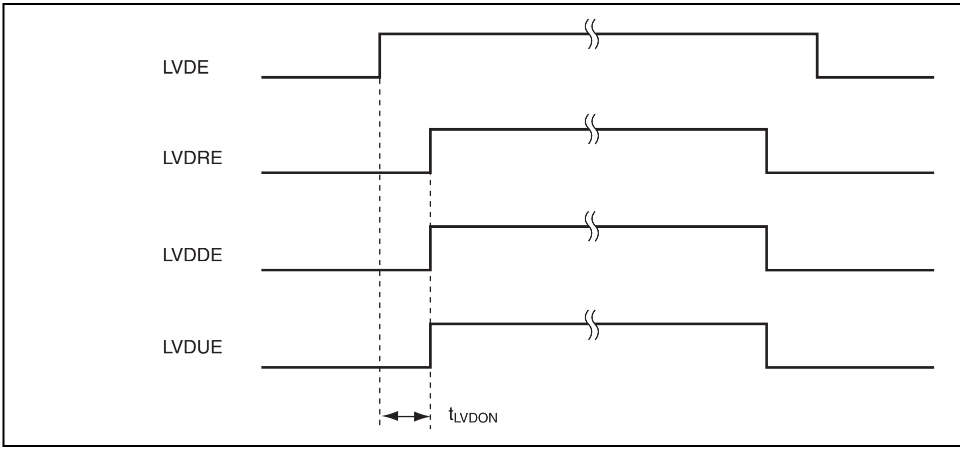


Figure 20.5 Timing for Operation/Release of Low-Voltage Detection Circu

21.1 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 21.1. 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, anc
level is that for the low level. The A/D converter analog power supply is not affected by
internal step-down circuit.

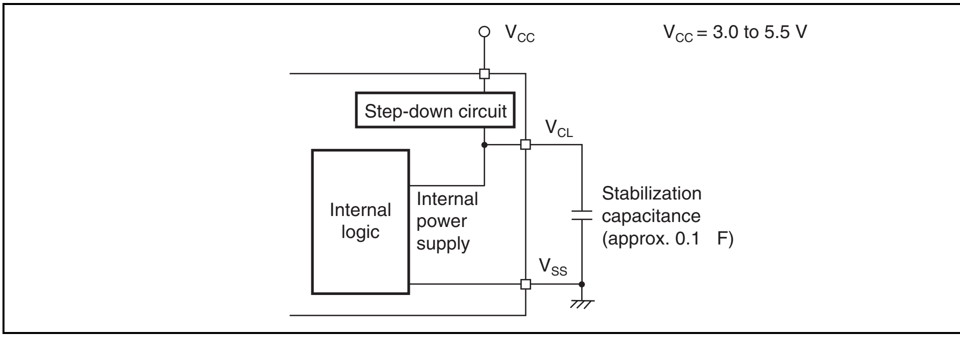


Figure 21.1 Power Supply Connection when Internal Step-Down Circuit is U

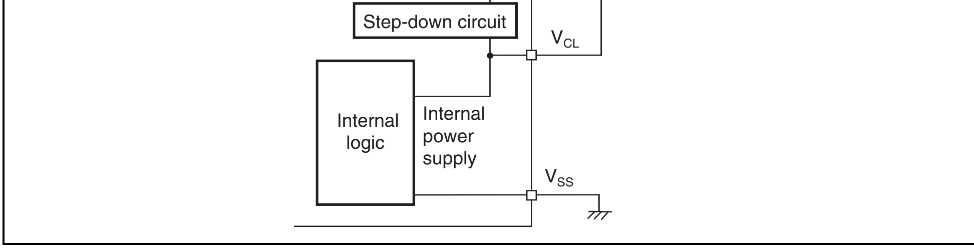


Figure 21.2 Power Supply Connection when Internal Step-Down Circuit is Not

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'F6FF	—	—
Timer control register_0	TCR_0	8	H'F700	Timer Z	8
Timer I/O control register A_0	TIORA_0	8	H'F701	Timer Z	8
Timer I/O control register C_0	TIORC_0	8	H'F702	Timer Z	8
Timer status register_0	TSR_0	8	H'F703	Timer Z	8
Timer interrupt enable register_0	TIER_0	8	H'F704	Timer Z	8
PWM mode output level control register_0	POCR_0	8	H'F705	Timer Z	8
Timer counter_0	TCNT_0	16	H'F706	Timer Z	16
General register A_0	GRA_0	16	H'F708	Timer Z	16
General register B_0	GRB_0	16	H'F70A	Timer Z	16
General register C_0	GRC_0	16	H'F70C	Timer Z	16
General register D_0	GRD_0	16	H'F70E	Timer Z	16
Timer control register_1	TCR_1	8	H'F710	Timer Z	8
Timer I/O control register A_1	TIORA_1	8	H'F711	Timer Z	8
Timer I/O control register C_1	TIORC_1	8	H'F712	Timer Z	8
Timer status register_1	TSR_1	8	H'F713	Timer Z	8
Timer interrupt enable register_1	TIER_1	8	H'F714	Timer Z	8
PWM mode output level control register_1	POCR_1	8	H'F715	Timer Z	8
Timer counter_1	TCNT_1	16	H'F716	Timer Z	16
General register A_1	GRA_1	16	H'F718	Timer Z	16
General register B_1	GRB_1	16	H'F71A	Timer Z	16

Timer output control register	TOCR	8	H'F725	Timer Z	8
—	—	—	H'F726, H'F727	Timer Z	—
Second data register/free running counter data register	RSECDR	8	H'F728	RTC	8
Minute data register	RMINDR	8	H'F729	RTC	8
Hour data register	RHRDR	8	H'F72A	RTC	8
Day-of-week data register	RWKDR	8	H'F72B	RTC	8
RTC control register 1	RTCCR1	8	H'F72C	RTC	8
RTC control register 2	RTCCR2	8	H'F72D	RTC	8
—	—	—	H'F72E	RTC	—
Clock source select register	RTCCSR	8	H'F72F	RTC	8
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'F73F	—	—
Serial mode register_2	SMR_2	8	H'F740	SCI3_2	8
Bit rate register_2	BRR_2	8	H'F741	SCI3_2	8
Serial control register 3_2	SCR3_2	8	H'F742	SCI3_2	8
Transmit data register_2	TDR_2	8	H'F743	SCI3_2	8
Serial status register_2	SSR_2	8	H'F744	SCI3_2	8

I2C status register	ICSR	8	H'F74C	IIC2	8
Slave address register	SAR	8	H'F74D	IIC2	8
I2C bus transmit data register	ICDRT	8	H'F74E	IIC2	8
I2C bus receive data register	ICDRR	8	H'F74F	IIC2	8
—	—	—	H'F750 to H'F75F	—	—
Timer mode register B1	TMB1	8	H'F760	Timer B1	8
Timer counter B1	TCB1	8	H'F761	Timer B1	8
—	—	—	H'F762 to H'FF8F	—	—
Flash memory control register 1	FLMCR1	8	H'FF90	ROM	8
Flash memory control register 2	FLMCR2	8	H'FF91	ROM	8
Flash memory power control register	FLPWCR	8	H'FF92	ROM	8
Erase block register 1	EBR1	8	H'FF93	ROM	8
—	—	—	H'FF94 to H'FF9A	ROM	—
Flash memory enable register	FENR	8	H'FF9B	ROM	8
—	—	—	H'FF9C to H'FF9F	ROM	—
Timer control register V0	TCRV0	8	H'FFA0	Timer V	8
Timer control/status register V	TCSRV	8	H'FFA1	Timer V	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	SCI3	—
A/D data register	ADDRA	16	H'FFB0	A/D converter	8
A/D data register	ADDRB	16	H'FFB2	A/D converter	8
A/D data register	ADDRC	16	H'FFB4	A/D converter	8
A/D data register	ADDRD	16	H'FFB6	A/D converter	8
A/D control/status register	ADCSR	8	H'FFB8	A/D converter	8
A/D control register	ADCR	8	H'FFB9	A/D converter	8
—	—	—	H'FFBA, H'FFBB	—	—
PWM data register L	PWDRL	8	H'FFBC	14-bit PWM	8
PWM data register U	PWDRU	8	H'FFBD	14-bit PWM	8
PWM control register	PWCR	8	H'FFBE	14-bit PWM	8
—	—	—	H'FFBF	14-bit PWM	—

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
Port pull-up control register 1	PUCR1	8	H'FFD0	I/O port	8
Port pull-up control register 5	PUCR5	8	H'FFD1	I/O port	8
—	—	—	H'FFD2, H'FFD3	I/O port	—
Port data register 1	PDR1	8	H'FFD4	I/O port	8
Port data register 2	PDR2	8	H'FFD5	I/O port	8
Port data register 3	PDR3	8	H'FFD6	I/O port	8
—	—	—	H'FFD7	I/O port	—
Port data register 5	PDR5	8	H'FFD8	I/O port	8
Port data register 6	PDR6	8	H'FFD9	I/O port	8
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 control register 1	PCR1	8	H'FFE4	I/O port	8
Port control register 2	PCR2	8	H'FFE5	I/O port	8
Port control register 3	PCR3	8	H'FFE6	I/O port	8
—	—	—	H'FFE7	I/O port	—
Port control register 5	PCR5	8	H'FFE8	I/O port	8
Port control register 6	PCR6	8	H'FFE9	I/O port	8
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	—
System control register 1	SYSCR1	8	H'FFF0	Low power	8
System control register 2	SYSCR2	8	H'FFF1	Low power	8
Interrupt edge select register 1	IEGR1	8	H'FFF2	Interrupt	8
Interrupt edge select register 2	IEGR2	8	H'FFF3	Interrupt	8
Interrupt enable register 1	IENR1	8	H'FFF4	Interrupt	8
Interrupt enable register 2	IENR2	8	H'FFF5	Interrupt	8
Interrupt flag register 1	IRR1	8	H'FFF6	Interrupt	8
Interrupt flag register 2	IRR2	8	H'FFF7	Interrupt	8
Wakeup interrupt flag register	IWPR	8	H'FFF8	Interrupt	8
Module standby control register 1	MSTCR1	8	H'FFF9	Low power	8
Module standby control register 2	MSTCR2	8	H'FFFA	Low power	8
—	—	—	H'FFEB	Low power	—

EEPROM slave address register	—	8	H'FF09	EEPROM	—	–
EEPROM key register	EKR	8	H'FF10	EEPROM	8	2

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

TIORC_0	—	IOD2	IOD1	IOD0	—	IOC2	IOC1	IOC0
TSR_0	—	—	—	OVF	IMFD	IMFC	IMFB	IMFA
TIER_0	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
POCR_0	—	—	—	—	—	POLD	POLC	POLB
TCNT_0	TCNT0H7	TCNT0H6	TCNT0H5	TCNT0H4	TCNT0H3	TCNT0H2	TCNT0H1	TCNT0H0
	TCNT0L7	TCNT0L6	TCNT0L5	TCNT0L4	TCNT0L3	TCNT0L2	TCNT0L1	TCNT0L0
GRA_0	GRA0H7	GRA0H6	GRA0H5	GRA0H4	GRA0H3	GRA0H2	GRA0H1	GRA0H0
	GRA0L7	GRA0L6	GRA0L5	GRA0L4	GRA0L3	GRA0L2	GRA0L1	GRA0L0
GRB_0	GRB0H7	GRB0H6	GRB0H5	GRB0H4	GRB0H3	GRB0H2	GRB0H1	GRB0H0
	GRB0L7	GRB0L6	GRB0L5	GRB0L4	GRB0L3	GRB0L2	GRB0L1	GRB0L0
GRC_0	GRC0H7	GRC0H6	GRC0H5	GRC0H4	GRC0H3	GRC0H2	GRC0H1	GRC0H0
	GRC0L7	GRC0L6	GRC0L5	GRC0L4	GRC0L3	GRC0L2	GRC0L1	GRC0L0
GRD_0	GRD0H7	GRD0H6	GRD0H5	GRD0H4	GRD0H3	GRD0H2	GRD0H1	GRD0H0
	GRD0L7	GRD0L6	GRD0L5	GRD0L4	GRD0L3	GRD0L2	GRD0L1	GRD0L0
TCR_1	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TIORA_1	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
TIORC_1	—	IOD2	IOD1	IOD0	—	IOC2	IOC1	IOC0
TSR_1	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA
TIER_1	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
POCR_1	—	—	—	—	—	POLD	POLC	POLB
TCNT_1	TCNT1H7	TCNT1H6	TCNT1H5	TCNT1H4	TCNT1H3	TCNT1H2	TCNT1H1	TCNT1H0
	TCNT1L7	TCNT1L6	TCNT1L5	TCNT1L4	TCNT1L3	TCNT1L2	TCNT1L1	TCNT1L0
GRA_1	GRA1H7	GRA1H6	GRA1H5	GRA1H4	GRA1H3	GRA1H2	GRA1H1	GRA1H0
	GRA1L7	GRA1L6	GRA1L5	GRA1L4	GRA1L3	GRA1L2	GRA1L1	GRA1L0

TMDR	BFD1	BFC1	BFD0	BFC0				SYNC
TPMR	—	PWMD1	PWMC1	PWMB1	—	PWMD0	PWMC0	PWMB0
TFCR	—	STCLK	ADEG	ADTRG	OLS1	OLS0	CMD1	CMD0
TOER	ED1	EC1	EB1	EA1	ED0	EC0	EB0	EA0
TOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0
RSECDR	BSY	SC12	SC11	SC10	SC03	SC02	SC01	SC00
RMINDR	BSY	MN12	MN11	MN10	MN03	MN02	MN01	MN00
RHRDR	BSY	—	HR11	HR10	HR03	HR02	HR01	HR00
RWKDR	BSY	—	—	—	—	WK2	WK1	WK0
RTCCR1	RUN	12/24	PM	RST	—	—	—	—
RTCCR2	—	—	FOIE	WKIE	DYIE	HRIE	MNIE	SEIE
RTCCSR	—	RCS6	RCS5	—	RCS3	RCS2	RCS1	RCS0
LVDCR	LVDE	—	—	—	LVDSEL	LVDRE	LVDDE	LVDUE
LVDSR	—	—	—	—	—	—	LVDDF	LVDUF
—	—	—	—	—	—	—	—	—
SMR_2	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0
BRR_2	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
SCR3_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_2	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
SSR_2	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT
RDR_2	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
ICCR1	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0
ICCR2	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—
ICMR	MLS	WAIT	—	—	BCWP	BC2	BC1	BC0

TCB1	TCB17	TCB16	TCB15	TCB14	TCB13	TCB12	TCB11	TCB10
—	—	—	—	—	—	—	—	—
FLMCR1	—	SWE	ESU	PSU	EV	PV	E	P
FLMCR2	FLER	—	—	—	—	—	—	—
FLPWCR	PDWND	—	—	—	—	—	—	—
EBR1	—	EB6	EB5	EB4	EB3	EB2	EB1	EB0
FENR	FLSHE	—	—	—	—	—	—	—
TCRV0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCSRV	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
TCORA	TCORA7	TCORA6	TCORA5	TCORA4	TCORA3	TCORA2	TCORA1	TCORA0
TCORB	TCORB7	TCORB6	TCORB5	TCORB4	TCORB3	TCORB2	TCORB1	TCORB0
TCNTV	TCNTV7	TCNTV6	TCNTV5	TCNTV4	TCNTV3	TCNTV2	TCNTV1	TCNTV0
TCRV1	—	—	—	TVEG1	TVEG0	TRGE	—	ICKS0
—	—	—	—	—	—	—	—	—
SMR	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0
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
	AD1	AD0	—	—	—	—	—	—

ADCR	TRGE							
—	—	—	—	—	—	—	—	—
PWDRL	PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	PWDRL0
PWDRU	—	—	PWDRU5	PWDRU4	PWDRU3	PWDRU2	PWDRU1	PWDRU0
PWCR	—	—	—	—	—	—	—	PWCR0
TCSRWD	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI	WRST
TCWD	TCWD7	TCWD6	TCWD5	TCWD4	TCWD3	TCWD2	TCWD1	TCWD0
TMWD	—	—	—	—	CKS3	CKS2	CKS1	CKS0
—	—	—	—	—	—	—	—	—
ABRKCR	RTINTE	CSEL1	CSEL0	ACMP2	ACMP1	ACMP0	DCMP1	DCMP0
ABRKSR	ABIF	ABIE	—	—	—	—	—	—
BARH	BARH7	BARH6	BARH5	BARH4	BARH3	BARH2	BARH1	BARH0
BARL	BARL7	BARL6	BARL5	BARL4	BARL3	BARL2	BARL1	BARL0
BDRH	BDRH7	BDRH6	BDRH5	BDRH4	BDRH3	BDRH2	BDRH1	BDRH0
BDRL	BDRL7	BDRL6	BDRL5	BDRL4	BDRL3	BDRL2	BDRL1	BDRL0
—	—	—	—	—	—	—	—	—
PUCR1	PUCR17	PUCR16	PUCR15	PUCR14	—	PUCR12	PUCR11	PUCR10
PUCR5	—	—	PUCR55	PUCR54	PUCR53	PUCR52	PUCR51	PUCR50
PDR1	P17	P16	P15	P14	—	P12	P11	P10
PDR2	—	—	—	P24	P23	P22	P21	P20
PDR3	P37	P36	P35	P34	P33	P32	P31	P30
PDR5	P57*[3]	P56*[3]	P55	P54	P53	P52	P51	P50
PDR6	P67	P66	P65	P64	P63	P62	P61	P60
PDR7	—	P76	P75	P74	—	P72	P71	P70

PCR3	PCR37	PCR36	PCR35	PCR34	PCR33	PCR32	PCR31	PCR30
PCR5	PCR57*[3]	PCR56*[3]	PCR55	PCR54	PCR53	PCR52	PCR51	PCR50
PCR6	PCR67	PCR66	PCR65	PCR64	PCR63	PCR62	PCR61	PCR60
PCR7	—	PCR76	PCR75	PCR74	—	PCR72	PCR71	PCR70
PCR8	PCR87	PCR86	PCR85	—	—	—	—	—
SYSCR1	SSBY	STS2	STS1	STS0	NESEL	—	—	—
SYSCR2	SMSEL	LSON	DTON	MA2	MA1	MA0	SA1	SA0
IEGR1	NMIEG	—	—	—	IEG3	IEG2	IEG1	IEG0
IEGR2	—	—	WPEG5	WPEG4	WPEG3	WPEG2	WPEG1	WPEG0
IENR1	IENDT	IENTA	IENWP	—	IEN3	IEN2	IEN1	IEN0
IENR2	—	—	IENTB1	—	—	—	—	—
IRR1	IRRDT	IRRTA	—	—	IRRI3	IRRI2	IRRI1	IRRI0
IRR2	—	—	IRRTB1	—	—	—	—	—
IWPR	—	—	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0
MSTCR1	—	MSTIIC	MSTS3	MSTAD	MSTWD	—	MSTTV	MSTTA
MSTCR2	MSTS3_2	—	—	MSTTB1	—	—	MSTTZ	MSTPWM
—	—	—	—	—	—	—	—	—

- EEPROM

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	M N
EKR									E

Notes: 1. LVDC: Low-voltage detection circuits (optional)
2. WDT: Watchdog timer
3. These bits are reserved in the EEPROM stacked F-ZTAT™ and mask-ROM v

POCR_0	Initialized						
TCNT_0	Initialized	—	—	—	—	—	
GRA_0	Initialized	—	—	—	—	—	
GRB_0	Initialized	—	—	—	—	—	
GRC_0	Initialized	—	—	—	—	—	
GRD_0	Initialized	—	—	—	—	—	
TCR_1	Initialized	—	—	—	—	—	
TIORA_1	Initialized	—	—	—	—	—	
TIORC_1	Initialized	—	—	—	—	—	
TSR_1	Initialized	—	—	—	—	—	
TIER_1	Initialized	—	—	—	—	—	
POCR_1	Initialized	—	—	—	—	—	
TCNT_1	Initialized	—	—	—	—	—	
GRA_1	Initialized	—	—	—	—	—	
GRB_1	Initialized	—	—	—	—	—	
GRC_1	Initialized	—	—	—	—	—	
GRD_1	Initialized	—	—	—	—	—	
TSTR	Initialized	—	—	—	—	—	
TMDR	Initialized	—	—	—	—	—	
TPMR	Initialized	—	—	—	—	—	
TFCR	Initialized	—	—	—	—	—	
TOER	Initialized	—	—	—	—	—	
TOCR	Initialized	—	—	—	—	—	
RSECDR	—	—	—	—	—	—	RTC

LVDSR	Initialized						(?)
SMR_2	Initialized	—	—	Initialized	Initialized	Initialized	SCI3
BRR_2	Initialized	—	—	Initialized	Initialized	Initialized	
SCR3_2	Initialized	—	—	Initialized	Initialized	Initialized	
TDR_2	Initialized	—	—	Initialized	Initialized	Initialized	
SSR_2	Initialized	—	—	Initialized	Initialized	Initialized	
RDR_2	Initialized	—	—	Initialized	Initialized	Initialized	
ICCR1	Initialized	—	—	—	—	—	IIC2
ICCR2	Initialized	—	—	—	—	—	
ICMR	Initialized	—	—	—	—	—	
ICIER	Initialized	—	—	—	—	—	
ICSR	Initialized	—	—	—	—	—	
SAR	Initialized	—	—	—	—	—	
ICDRT	Initialized	—	—	—	—	—	
ICDRR	Initialized	—	—	—	—	—	
TMB1	Initialized	—	—	—	—	—	Time
TCB1	Initialized	—	—	—	—	—	
FLMCR1	Initialized	—	—	Initialized	Initialized	Initialized	ROM
FLMCR2	Initialized	—	—	—	—	—	
FLPWCR	Initialized	—	—	—	—	—	
EBR1	Initialized	—	—	Initialized	Initialized	Initialized	
FENR	Initialized	—	—	—	—	—	
TCRV0	Initialized	—	—	Initialized	Initialized	Initialized	Time
TCSRV	Initialized	—	—	Initialized	Initialized	Initialized	

TDR	Initialized			Initialized	Initialized	Initialized	
SSR	Initialized	—	—	Initialized	Initialized	Initialized	
RDR	Initialized	—	—	Initialized	Initialized	Initialized	
ADDRA	Initialized	—	—	Initialized	Initialized	Initialized	A/D c
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	
PWDRL	Initialized	—	—	—	—	—	14bit
PWDRU	Initialized	—	—	—	—	—	
PWCR	Initialized	—	—	—	—	—	
TCSRWD	Initialized	—	—	—	—	—	WDT*
TCWD	Initialized	—	—	—	—	—	
TMWD	Initialized	—	—	—	—	—	
ABRKCR	Initialized	—	—	—	—	—	Addre
ABRKSR	Initialized	—	—	—	—	—	
BARH	Initialized	—	—	—	—	—	
BARL	Initialized	—	—	—	—	—	
BDRH	Initialized	—	—	—	—	—	
BDRL	Initialized	—	—	—	—	—	
PUCR1	Initialized	—	—	—	—	—	I/O po
PUCR5	Initialized	—	—	—	—	—	
PDR1	Initialized	—	—	—	—	—	

PMR1	Initialized						
PMR5	Initialized	—	—	—	—	—	
PMR3	Initialized	—	—	—	—	—	
PCR1	Initialized	—	—	—	—	—	
PCR2	Initialized	—	—	—	—	—	
PCR3	Initialized	—	—	—	—	—	
PCR5	Initialized	—	—	—	—	—	
PCR6	Initialized	—	—	—	—	—	
PCR7	Initialized	—	—	—	—	—	
PCR8	Initialized	—	—	—	—	—	
SYSCR1	Initialized	—	—	—	—	—	Low
SYSCR2	Initialized	—	—	—	—	—	
IEGR1	Initialized	—	—	—	—	—	Inte
IEGR2	Initialized	—	—	—	—	—	
IENR1	Initialized	—	—	—	—	—	
IENR2	Initialized	—	—	—	—	—	
IRR1	Initialized	—	—	—	—	—	
IRR2	Initialized	—	—	—	—	—	
IWPR	Initialized	—	—	—	—	—	
MSTCR1	Initialized	—	—	—	—	—	Low
MSTCR2	Initialized	—	—	—	—	—	

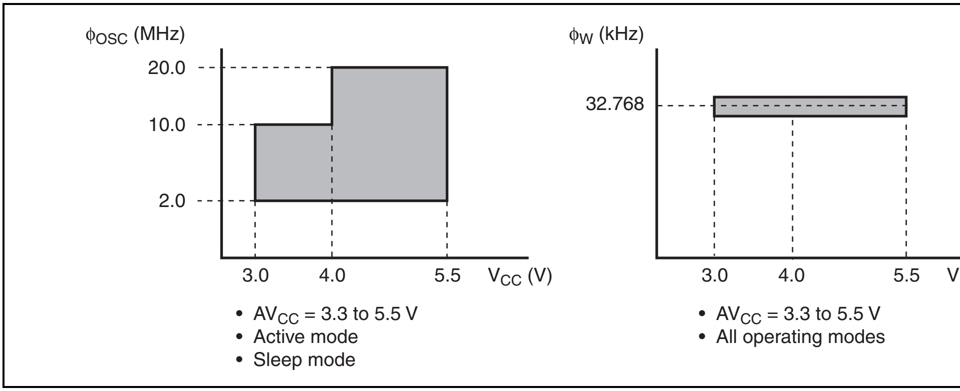
	and X1	IN	CC	
	Port B		–0.3 to AV_{CC} +0.3	V
	X1		–0.3 to 4.3	V
Operating temperature		T_{opr}	–20 to +75	°C
Storage temperature		T_{stg}	–55 to +125	°C

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.

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

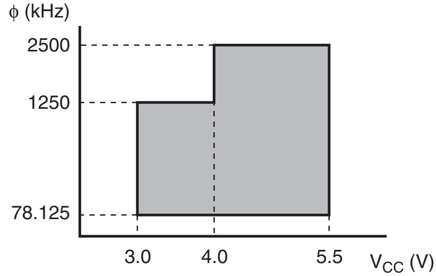
23.2.1 Power Supply Voltage and Operating Ranges

Power Supply Voltage and Oscillation Frequency Range



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

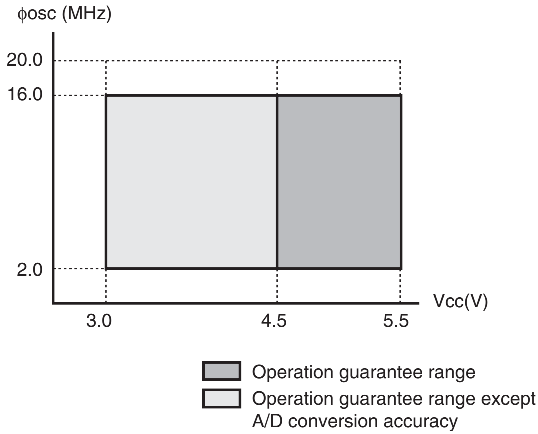
- AV_{CC} = 3.3 to 5.5 V
- Subactive mode
- Subsleep mode



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

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

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



		$\overline{ADTRG}$, TMIB1, TMRIV,					
		TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1,SCK3, SCK3_2, TRGV		$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	
		RXD, RXD_2, SCL, SDA, P10 to P12, P14 to P17, P20 to P24, P30 to P37,	V_{CC} = 4.0 to 5.5 V	$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V
		P50 to P57, P60 to P67, P70 to P72 P74 to P76, P85 to P87		$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	
		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 $\overline{IRQ3}$, $\overline{ADTRG}$, TMIB1, TMRIV,	V_{CC} = 4.0 to 5.5 V	–0.3	—	$V_{CC} \times 0.2$	V
		TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, SCK3, SCK3_2, TRGV		–0.3	—	$V_{CC} \times 0.1$	

Note: Connect the TEST pin to Vss.

		PB0 to PB7					
		OSC1	V_{CC} = 4.0 to 5.5 V	–0.3	—	0.5	V
				–0.3	—	0.3	
Output high voltage	V_{OH}	P10 to P12, P14 to P17, P20 to P24, P30 to P37, P50 to P55, P60 to P67, P70 to P72, P74 to P76, P85 to P87,	V_{CC} = 4.0 to 5.5 V $-I_{OH}$ = 1.5 mA	V_{CC} – 1.0	—	—	V
			$-I_{OH}$ = 0.1 mA	V_{CC} – 0.5	—	—	
		P56, P57	V_{CC} = 4.0 to 5.5 V $-I_{OH}$ = 0.1 mA	V_{CC} – 2.5	—	—	V
			V_{CC} = 3.0 to 4.0 V $-I_{OH}$ = 0.1 mA	V_{CC} – 2.0	—	—	
Output low voltage	V_{OL}	P10 to P12, P14 to P17, P20 to P24, P30 to P37, P50 to P57, P70 to P72, P74 to P76, P85 to P87	V_{CC} = 4.0 to 5.5 V I_{OL} = 1.6 mA	—	—	0.6	V
			I_{OL} = 0.4 mA	—	—	0.4	
		P60 to P67	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	

		FTIOD0, FTIOA1 to FTIOD1 RXD, SCK3, RXD_2, SCK3_2, SCL, SDA						
		P10 to P12, P14 to P17, P20 to P24, P30 to P37, P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 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	
Pull-up MOS current	$–I_p$	P10 to P12, P14 to P17, P50 to P55	V_{CC} = 5.0 V, V_{IN} = 0.0 V	50.0	—	300.0	μA	
			V_{CC} = 3.0 V, V_{IN} = 0.0 V	—	60.0	—		R va
Input capaci-tance	C_{in}	All input pins except power supply pins	f = 1 MHz, V_{IN} = 0.0 V, T_a = 25°C	—	—	15.0	pF	
Active mode current consump-tion	I_{OPE1}	V_{CC}	Active mode 1 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	21.0	30.0	mA	*
			Active mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	9.0	—		* R va
	I_{OPE2}	V_{CC}	Active mode 2 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	1.8	3.0	mA	*
			Active mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	1.2	—		* R va

			V_{CC} = 3.0 V, f_{OSC} = 10 MHz				
Subactive mode current consump-tion	I_{SUB}	V_{CC}	V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	35.0	70.0	μA
			V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/8$)	—	25.0	—	
Subsleep mode current consump-tion	I_{SUBSP}	V_{CC}	V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	25.0	50.0	μA
Standby mode current consump-tion	I_{STBY}	V_{CC}	32-kHz crystal resonator not used	—	—	5.0	μA
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).

				crystal resonator
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	Main clock: ceramic or crysta Subclock: Pin X1 = V_{SS}

Table 23.2 DC Characteristics (2)

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

Item	Symbol	Applicable Pins	Test Condition	Values			Unit
				Min	Typ	Max	
EEPROM current consump-tion	I_{EEW}	V_{CC}	V_{CC} = 5.0 V, t_{SCL} = 2.5 µs (when writing)	—	—	2.0	mA
	I_{EER}	V_{CC}	V_{CC} = 5.0 V, t_{SCL} = 2.5 µs (when reading)	—	—	0.3	mA
	I_{EESTBY}	V_{CC}	V_{CC} = 5.0 V, t_{SCL} = 2.5 µs (at standby)	—	—	3.0	µA

Note: * The current consumption of the EEPROM chip is shown.

For the current consumption of H8/3687N, add the above current values to the consumption of H8/3687F.

		SDA				
		Port 6		—	—	10.0
		SCL, SDA		—	—	6.0
Allowable output low current (total)	$\sum I_{OL}$	Output pins except port 6, SCL, and SDA	V_{CC} = 4.0 to 5.5 V	—	—	40.0
		Port 6, SCL, and SDA		—	—	80.0
		Output pins except port 6, SCL, and SDA		—	—	20.0
		Port 6, 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)	$\lvert -\sum I_{OH} \rvert$	All output pins	V_{CC} = 4.0 to 5.5 V	—	—	30.0
				—	—	8.0

System clock (ϕ) cycle time	t_{cyc}			1	—	64	t_{OSC}	*
				—	—	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		

		$\overline{WKP5}$, TMCIV, TMRIV, TRGV, $\overline{ADTRG}$, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1				
Input pin low width	t_{IL}	$\overline{NMI}$, TMIB1, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{WKP0}$ to $\overline{WKP5}$, TMCIV, TMRIV, TRGV, $\overline{ADTRG}$, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1	2	—	—	t_{cyc} t_{subcyc}

Notes: 1. When an external clock is input, the minimum system clock oscillation freque
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	

Input clock pulse width	t_{SCKW}	SCK3		0.4		0.6	t_{Scyc}	
Transmit data delay time (clocked synchronous)	t_{TXD}	TXD	V_{CC} = 4.0 to 5.5 V	—	—	1	t_{cyc}	F
				—	—	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	—	—		

		AN7						
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	* F V
	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 w
A/D converter is idle.

23.2.5 Watchdog Timer Characteristics

Table 23.7 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.

Reprogramming count		N_{WEC}		1000	10000	
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 ir memory control register 1 (FLMCR1) is set. The erase-verify time is not incluc

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 c and z3, so that it does not exceed the programming time maximum value (t_P(r 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) × maxi 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).

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.

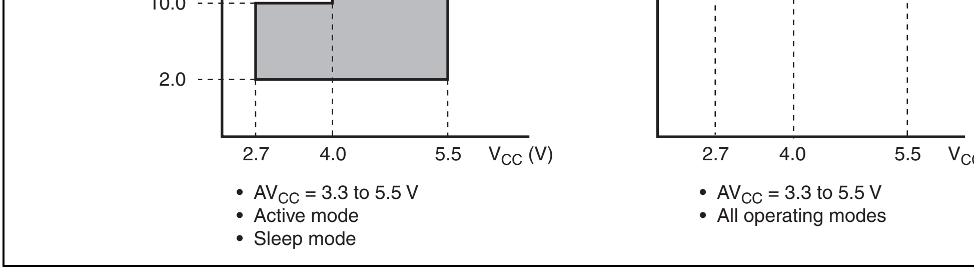
23.2.9 Power-On Reset Circuit Characteristics (Optional)

Table 23.11 Power-On Reset Circuit Characteristics

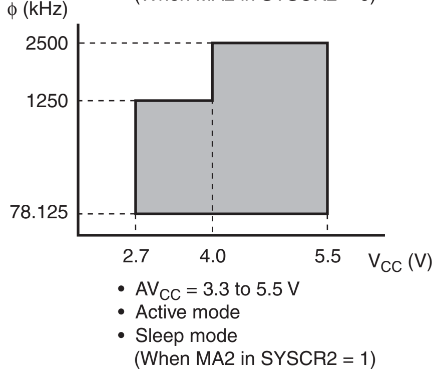
V_{SS} = 0.0 V, T_a = –20 to +75°C, unless otherwise indicated.

			Values		
Item	**Symbol**	**Test Condition**	**Min**	**Typ**	**Max**
Pull-up resistance of $\overline{RES}$ pin	R_{RES}		100	150	—
Power-on reset start voltage*	V_{por}		—	—	100

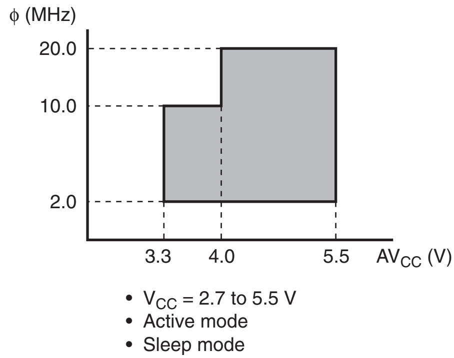
Note: * The power-supply voltage (Vcc) must fall below Vpor = 100 mV and then rise charge of the $\overline{RES}$ pin is removed completely. In order to remove charge of th pin, it is recommended that the diode be placed in the Vcc side. If the power-s voltage (Vcc) rises from the point over 100 mV,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



Power Supply Voltage and Operating Frequency Range



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



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

Operation guarantee range

Operation guarantee range except A/D conversion accuracy

ADTRG, TMIB1, TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, SCK3, SCK3_2, TRGV					
		$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	
RXD, RXD_2 SCL, SDA, P10 to P12, P14 to P17, P20 to P24, P30 to P37	V_{CC} = 4.0 to 5.5 V	$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V
P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 to P87		$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	
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
		$V_{CC} - 0.3$	—	$V_{CC} + 0.3$	

Note: Connect the TEST pin to Vss.

		RXD, RXD_2, SCL, SDA, P10 to P12, P14 to P17, P20 to P24, P30 to P37, P50 to P57, P60 to P67,. P70 to P72, P74 to P76, P85 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	
Output high voltage	V_{OH}	P10 to P12, P14 to P17, P20 to P24, P30 to P37, P50 to P55, P60 to P67, P70 to P72, P74 to P76, P85 to P87	V_{CC} = 4.0 to 5.5 V $-I_{OH}$ = 1.5 mA	V_{CC} – 1.0	—	—	V
			$-I_{OH}$ = 0.1 mA	V_{CC} – 0.5	—	—	
		P56, P57	V_{CC} = 4.0 to 5.5 V $-I_{OH}$ = 0.1 mA	V_{CC} – 2.5	—	—	V
			V_{CC} =2.7 to 4.0 V $-I_{OH}$ = 0.1 mA	V_{CC} – 2.0	—	—	

			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	$\| I_{IL} \|$	OSC1, TMIB1, $\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$, TRGV, TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, RXD, SCK3, RXD_2, SCK3_2, SCL, SDA	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	—	—	1.0	μA
		P10 to P12, P14 to P17, P20 to P24, P30 to P37, P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 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

consump- tion			Active mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	9.0	—		* F v
	I_{OPE2}	V_{CC}	Active mode 2 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	1.8	3.0	mA	*
			Active mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	1.2	—		* F v
Sleep mode current consump- tion	I_{SLEEP1}	V_{CC}	Sleep mode 1 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	17.5	22.5	mA	*
			Sleep mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	7.5	—		* F v
	I_{SLEEP2}	V_{CC}	Sleep mode 2 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	1.7	2.7	mA	*
			Sleep mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	1.1	—		* F v
Subactive mode current consump- tion	I_{SUB}	V_{CC}	V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	35.0	70.0	μA	*
			V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/8$)	—	25.0	—		* F v
Subsleep mode current consump- tion	I_{SUBSP}	V_{CC}	V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	25.0	50.0	μA	*

Mode	$\overline{\text{RES}}$ Pin	Internal State	Other Pins	Oscillator Pi
Active mode 1	V_{CC}	Operates	V_{CC}	Main clock: ceramic or cry 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 cry resonator
Subsleep mode	V_{CC}	Only timers operate	V_{CC}	Subclock reso crystal
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	Main clock: ceramic or cry resonator Subclock: Pin X1 = V_{SS}

Note: * The current consumption of the EEPROM chip is shown.
For the current consumption of H8/3687N, add the above current values to the
consumption of H8/3687.

		SCL, and SDA				
		Port 6		—	—	10.0
		SCL, SDA		—	—	6.0
Allowable output low current (total)	$\sum I_{OL}$	Output pins except port 6, SCL, and SDA	V_{CC} = 4.0 to 5.5 V	—	—	40.0
		Port 6, SCL, and SDA		—	—	80.0
		Output pins except port 6, SCL, and SDA		—	—	20.0
		Port 6, 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)	$\lvert -\sum I_{OH} \rvert$	All output pins	V_{CC} = 4.0 to 5.5 V	—	—	30.0
				—	—	8.0

System clock (ϕ) cycle time	t_{cyc}			1	—	64	t_{OSC}	*2
				—	—	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	*2
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		

		$\overline{\text{WKP5}}$, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1					
Input pin low width	t_{IL}	$\overline{\text{NMI}}$, TMIB1, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1		2	—	—	t_{cyc} t_{subcyc}

Notes: 1. When an external clock is input, the minimum system clock oscillation freque
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								
Transmit data delay time (clocked synchronous)	t_{TXD}	TXD	V_{CC} = 4.0 to 5.5 V	—	—	1	t_{cyc}	F
				—	—	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	—	—		

		AN7		0.3			
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 w
A/D converter is idle.

23.3.5 Watchdog Timer Characteristics

Table 23.17 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			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.

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

23.4 Operation Timing

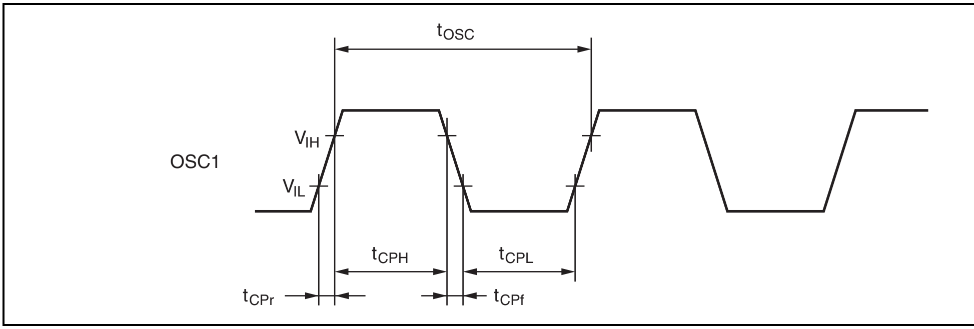


Figure 23.1 System Clock Input Timing

Figure 23.2 $\overline{\text{RES}}$ Low Width Timing

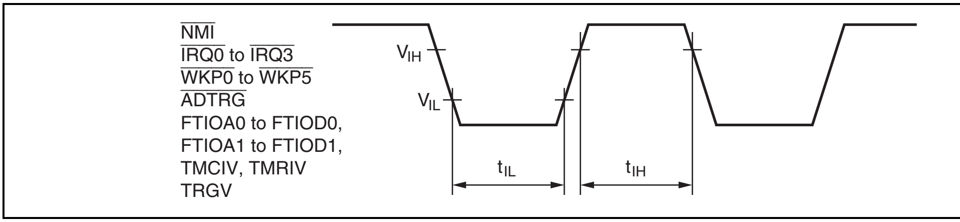
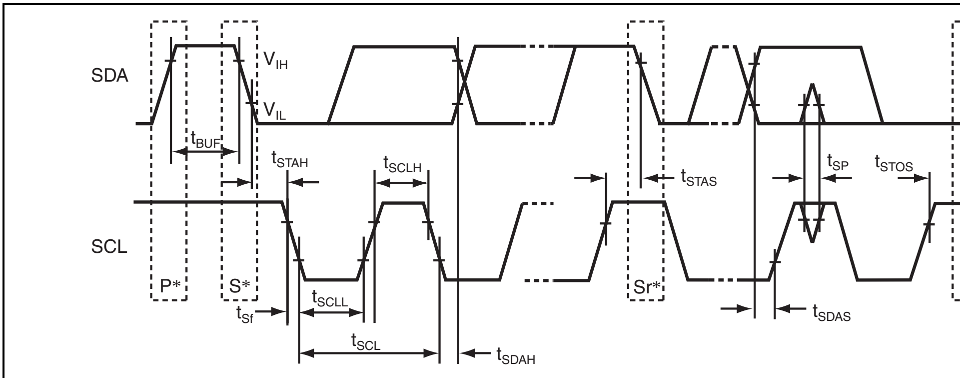


Figure 23.3 Input Timing



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

Figure 23.4 I²C Bus Interface Input/Output Timing

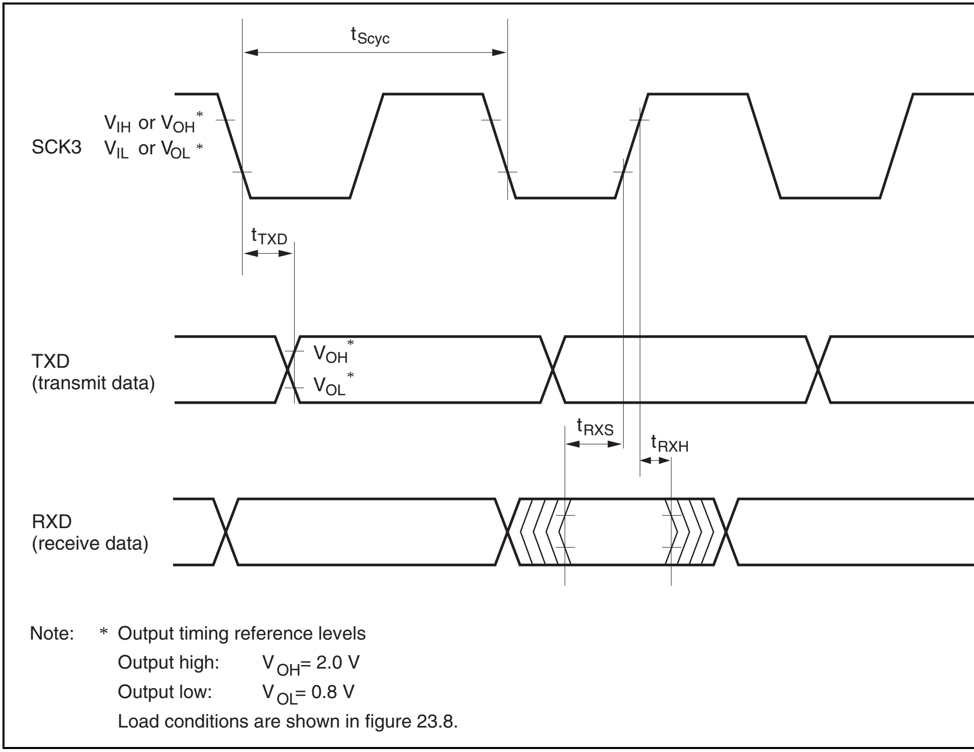


Figure 23.6 SCI Input/Output Timing in Clocked Synchronous Mode

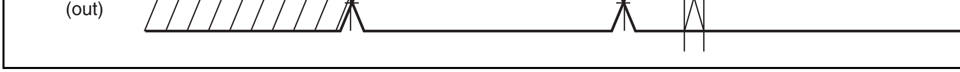


Figure 23.7 EEPROM Bus Timing

23.5 Output Load Condition

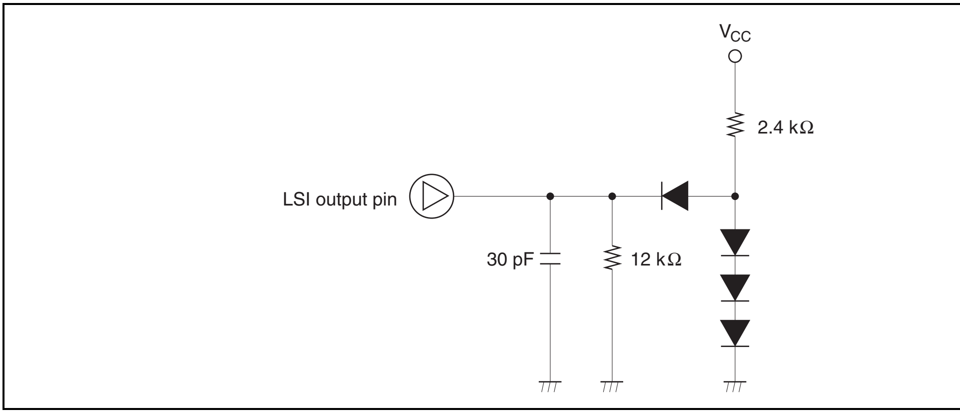


Figure 23.8 Output Load Circuit

Rn	General register
ERd	General destination register (address register or 32-bit register)
ERs	General source register (address register or 32-bit register)
ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
PC	Program counter
SP	Stack pointer
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
disp	Displacement
→	Transfer from the operand on the left to the operand on the right, or trans the state on the left to the state on the right
+	Addition of the operands on both sides
–	Subtraction of the operand on the right from the operand on the left
×	Multiplication of the operands on both sides
÷	Division of the operand on the left by the operand on the right
∧	Logical AND of the operands on both sides
∨	Logical OR of the operands on both sides
⊕	Logical exclusive OR of the operands on both sides
¬	NOT (logical complement)
( ), < >	Contents of operand

—	Not affected by execution of the instruction
Δ	Varies depending on conditions, described in notes

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

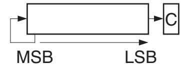
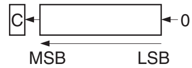
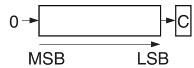
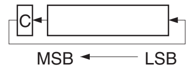
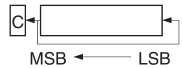
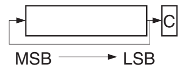
	MOV.L ERs, ERd	L		2								ERs32 → ERd32	—	—	↕	↕	0	—
	MOV.L @ERs, ERd	L			4							@ERs → ERd32	—	—	↕	↕	0	—
	MOV.L @(d:16, ERs), ERd	L				6						@(d:16, ERs) → ERd32	—	—	↕	↕	0	—
	MOV.L @(d:24, ERs), ERd	L				10						@(d:24, ERs) → ERd32	—	—	↕	↕	0	—
	MOV.L @ERs+, ERd	L					4					@ERs → ERd32 ERs32+4 → ERs32	—	—	↕	↕	0	—
	MOV.L @aa:16, ERd	L						6				@aa:16 → ERd32	—	—	↕	↕	0	—
	MOV.L @aa:24, ERd	L						8				@aa:24 → ERd32	—	—	↕	↕	0	—
	MOV.L ERs, @ERd	L			4							ERs32 → @ERd	—	—	↕	↕	0	—
	MOV.L ERs, @(d:16, ERd)	L				6						ERs32 → @(d:16, ERd)	—	—	↕	↕	0	—
	MOV.L ERs, @(d:24, ERd)	L				10						ERs32 → @(d:24, ERd)	—	—	↕	↕	0	—
	MOV.L ERs, @–ERd	L					4					ERd32–4 → ERd32 ERs32 → @ERd	—	—	↕	↕	0	—
	MOV.L ERs, @aa:16	L						6				ERs32 → @aa:16	—	—	↕	↕	0	—
	MOV.L ERs, @aa:24	L						8				ERs32 → @aa:24	—	—	↕	↕	0	—
POP	POP.W Rn	W									2	@SP → Rn16 SP+2 → SP	—	—	↕	↕	0	—
	POP.L ERn	L									4	@SP → ERn32 SP+4 → SP	—	—	↕	↕	0	—
PUSH	PUSH.W Rn	W									2	SP–2 → SP Rn16 → @SP	—	—	↕	↕	0	—
	PUSH.L ERn	L									4	SP–4 → SP ERn32 → @SP	—	—	↕	↕	0	—
MOVFPE	MOVFPE @aa:16, Rd	B						4				Cannot be used in this LSI	Cannot be used in this LSI					
MOVTPE	MOVTPE Rs, @aa:16	B						4				Cannot be used in this LSI	Cannot be used in this LSI					

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

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

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

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

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

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

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

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

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

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 (upper left of a split cell)
Instruction when n (lower right of a split cell)

AH ALBH BLCH \ CL	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:

$I = L = 2, \quad J = K = M = N = 0$

From table A.3:

$S_I = 2, \quad S_L = 2$

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:

$I = 2, \quad J = K = 1, \quad L = M = N = 0$

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 22.1, R
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 @ERs+, Rd	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	

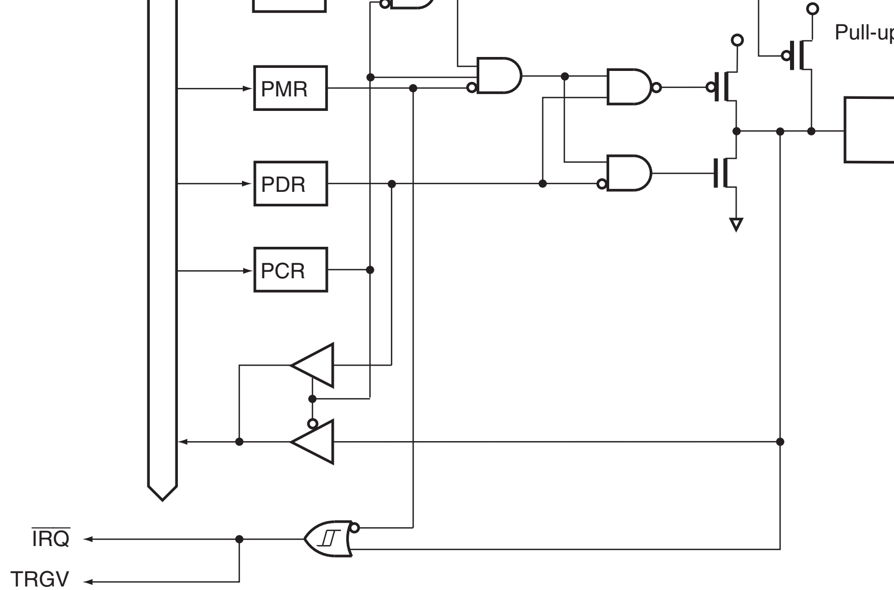
	SHAL.L ERd	1	
SHAR	SHAR.B Rd	1	
	SHAR.W Rd	1	
	SHAR.L ERd	1	
SHLL	SHLL.B Rd	1	
	SHLL.W Rd	1	
	SHLL.L ERd	1	
SHLR	SHLR.B Rd	1	
	SHLR.W Rd	1	
	SHLR.L ERd	1	
SLEEP	SLEEP	1	
STC	STC CCR, Rd	1	
	STC CCR, @ERd	2	1
	STC CCR, @(d:16,ERd)	3	1
	STC CCR, @(d:24,ERd)	5	1
	STC CCR,@-ERd	2	1
	STC CCR, @aa:16	3	1
	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 and R4. The source and destination operands are a
n+1 times respectively.

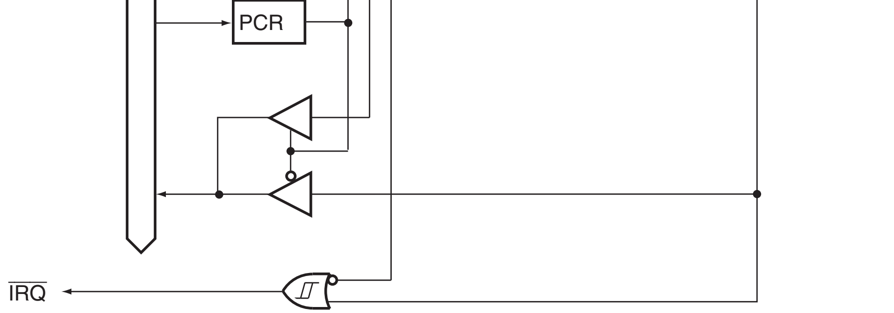
2. Cannot be used in this LSI.

	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 (P14, P16)

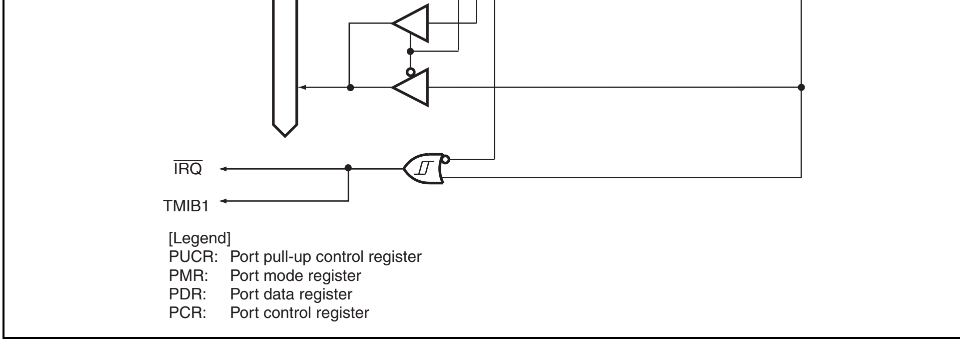
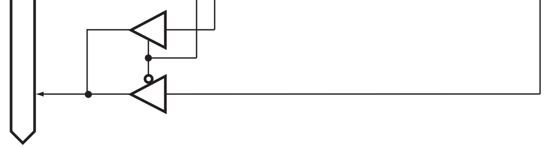
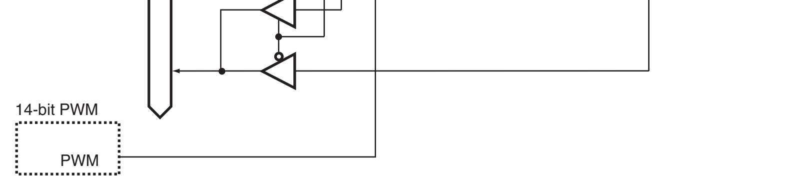


Figure B.3 Port 1 Block Diagram (P15)



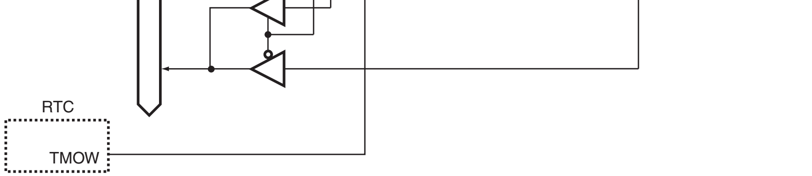
[Legend]
PUCR: Port pull-up control register
PDR: Port data register
PCR: Port control register

Figure B.4 Port 1 Block Diagram (P12)



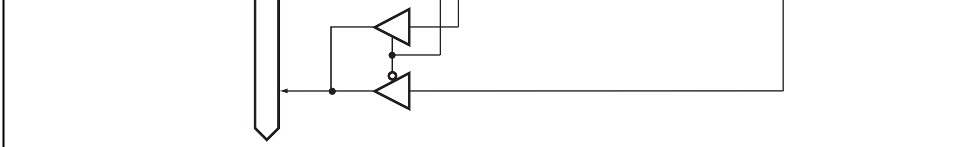
[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.5 Port 2 Block Diagram (P11)



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

Figure B.6 Port 1 Block Diagram (P10)



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

Figure B.7 Port 2 Block Diagram (P24, P23)

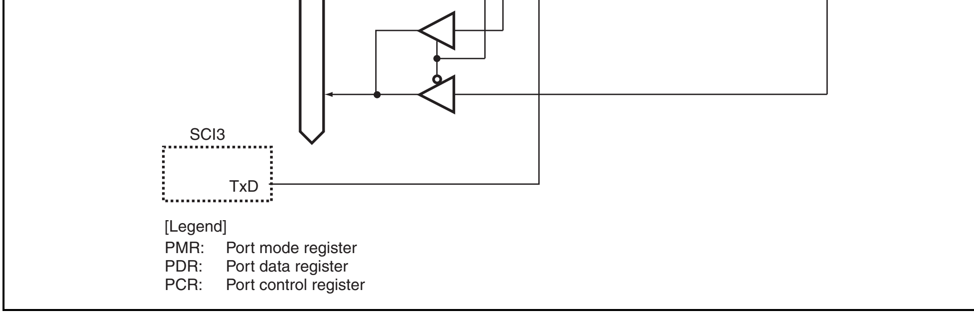


Figure B.8 Port 2 Block Diagram (P22)

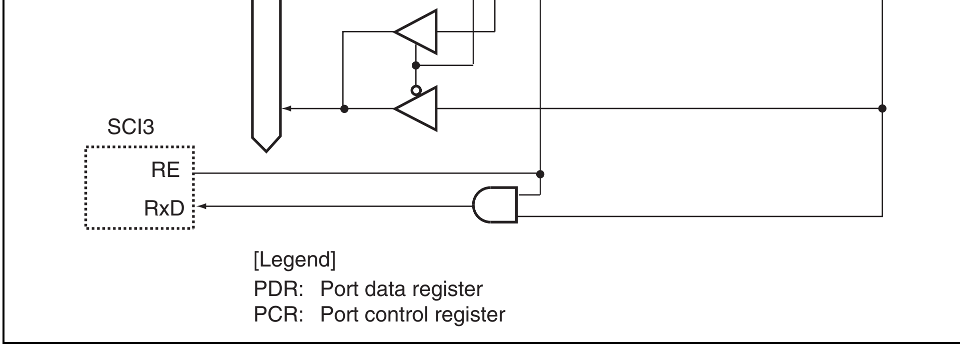


Figure B.9 Port 2 Block Diagram (P21)

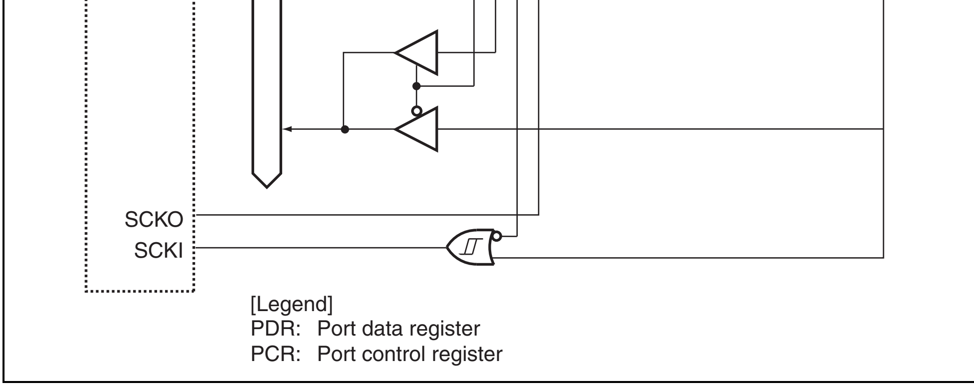
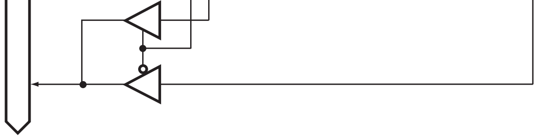


Figure B.10 Port 2 Block Diagram (P20)

RENESAS



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

Figure B.11 Port 3 Block Diagram (P37 to P30)

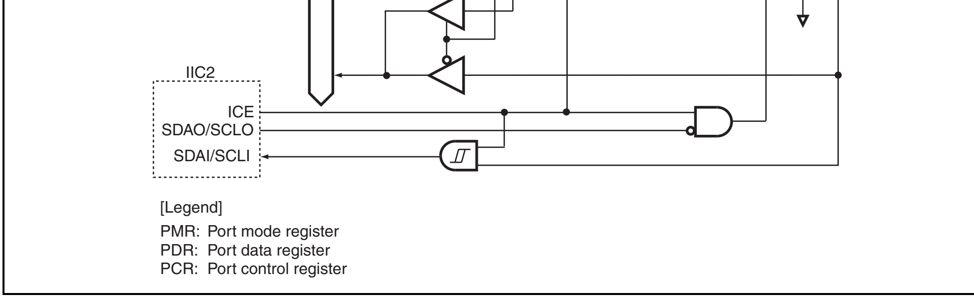
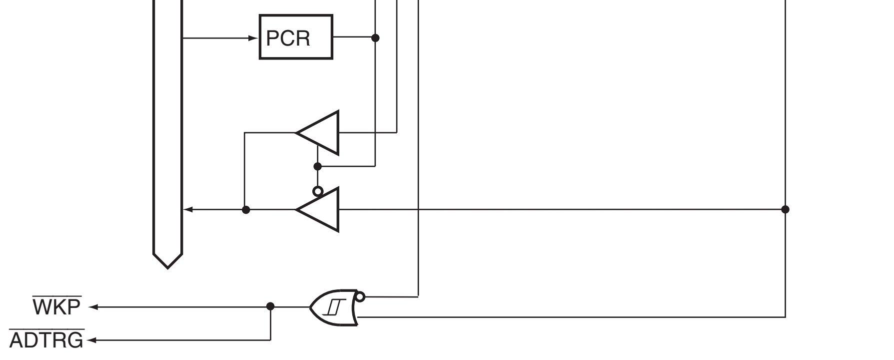


Figure B.12 Port 5 Block Diagram (P57, P56)*

Note: * This diagram is applied to the SCL and SDA pins in the H8/3687N.



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

Figure B.13 Port 5 Block Diagram (P55)

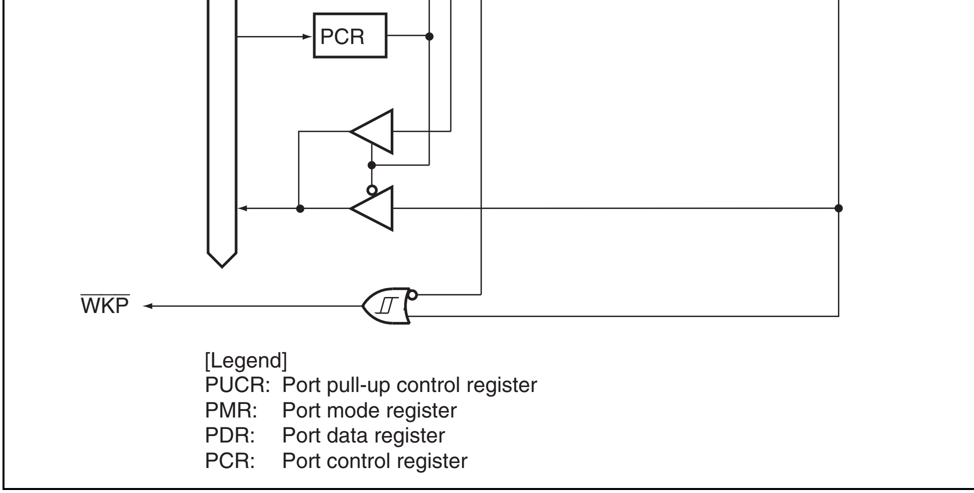


Figure B.14 Port 5 Block Diagram (P54 to P50)

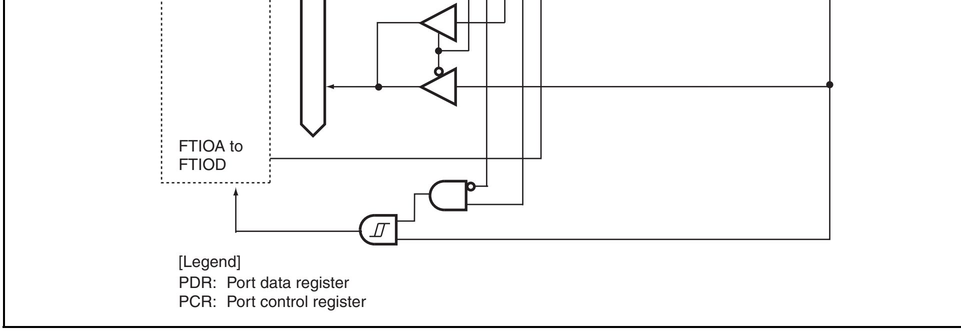


Figure B.15 Port 6 Block Diagram (P67 to P60)

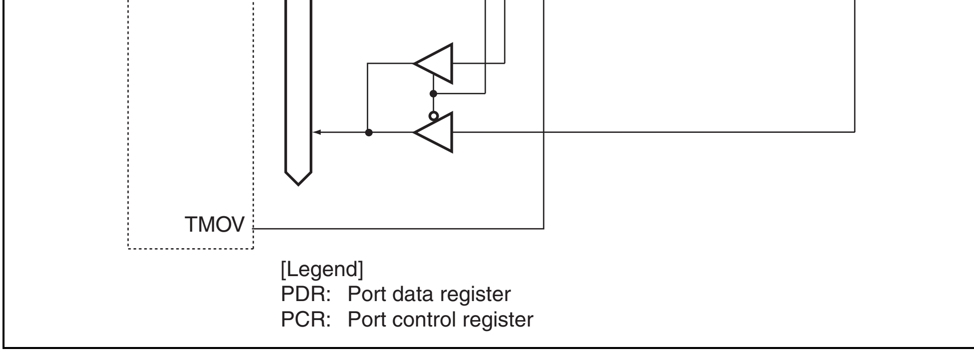


Figure B.16 Port 7 Block Diagram (P76)

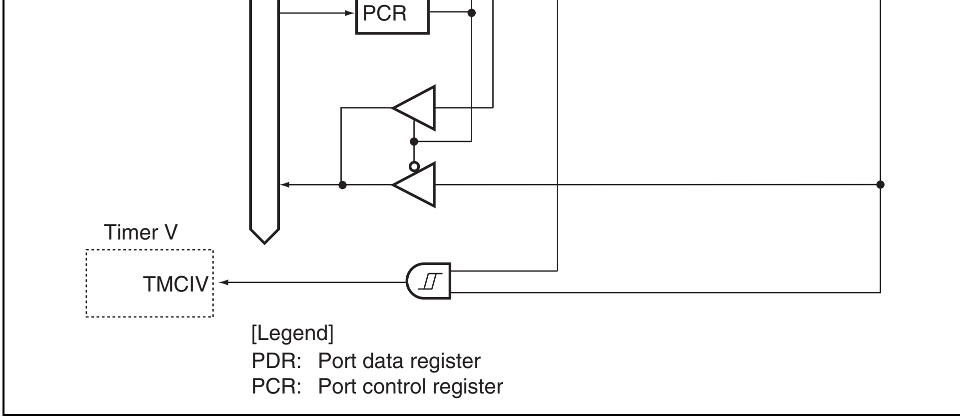


Figure B.17 Port 7 Block Diagram (P75)

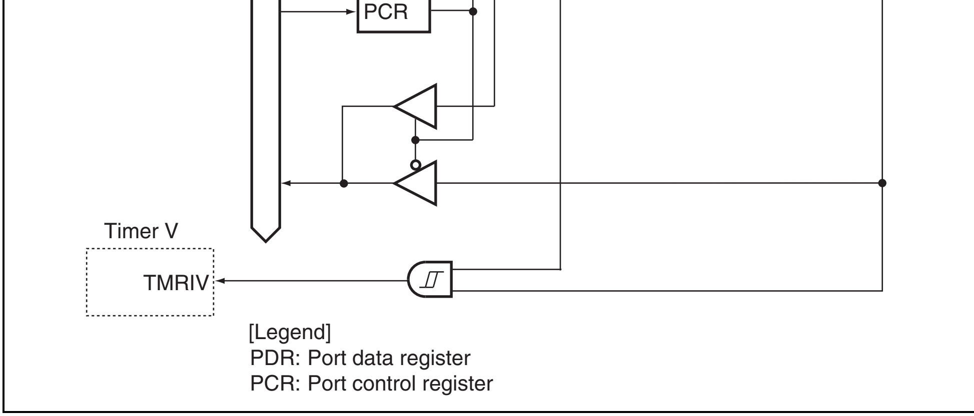


Figure B.18 Port 7 Block Diagram (P74)

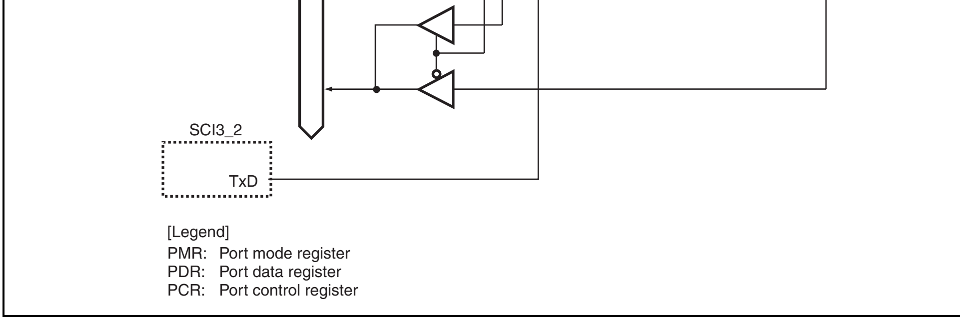


Figure B.19 Port 7 Block Diagram (P72)

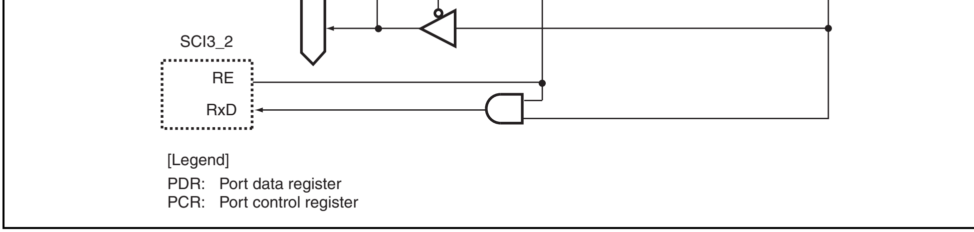


Figure B.20 Port 7 Block Diagram (P71)

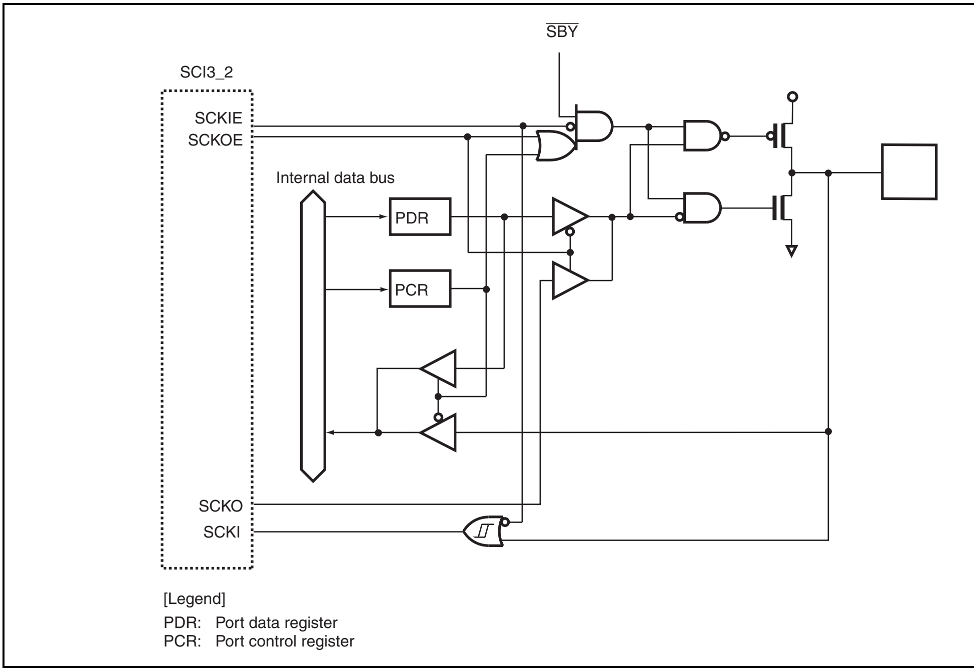


Figure B.21 Port 7 Block Diagram (P70)

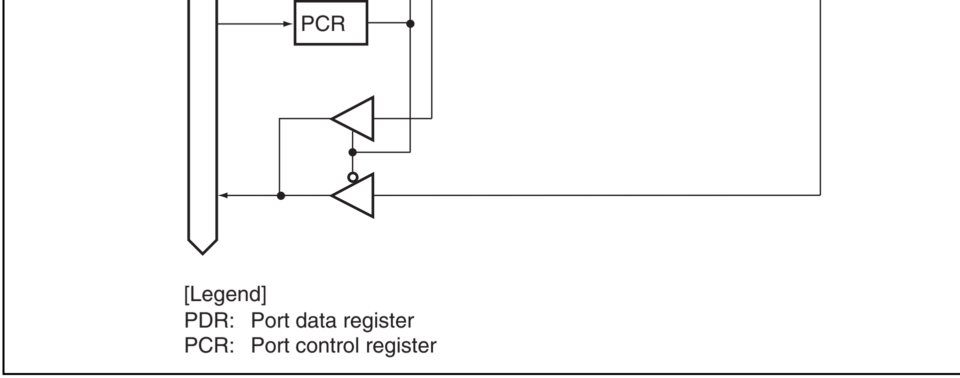


Figure B.22 Port 8 Block Diagram (P87 to P85)

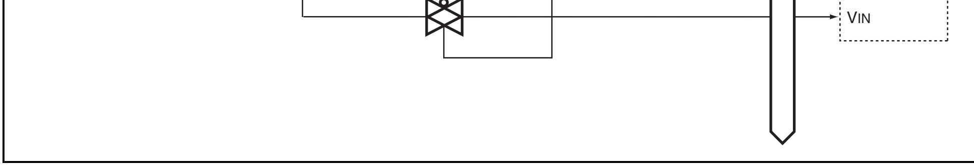


Figure B.23 Port B Block Diagram (PB7 to PB0)

P67 to P60	High impedance	Retained	Retained	High impedance	Functioning	Fu
P76 to P74, P72 to P70	High impedance	Retained	Retained	High impedance	Functioning	Fu
P87 to P85	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 H8/3687N.

		Product with POR & LVDC	HD6433687GH	HD6433687G(***)H	QFP-64 (F
			HD6433687GFP	HD6433687G(***)FP	LQFP-64 (
H8/3686	Mask ROM version	Standard product	HD6433686H	HD6433686(***)H	QFP-64 (F
			HD6433686FP	HD6433686(***)FP	LQFP-64 (
		Product with POR & LVDC	HD6433686GH	HD6433686G(***)H	QFP-64 (F
			HD6433686GFP	HD6433686G(***)FP	LQFP-64 (
H8/3685	Mask ROM version	Standard product	HD6433685H	HD6433685(***)H	QFP-64 (F
			HD6433685FP	HD6433685(***)FP	LQFP-64 (
		Product with POR & LVDC	HD6433685GH	HD6433685G(***)H	QFP-64 (F
			HD6433685GFP	HD6433685G(***)FP	LQFP-64 (
H8/3684	Flash memory version	Standard product	HD64F3684H	HD64F3684H	QFP-64 (F
			HD64F3684FP	HD64F3684FP	LQFP-64 (
		Product with POR & LVDC	HD64F3684GH	HD64F3684GH	QFP-64 (F
			HD64F3684GFP	HD64F3684GFP	LQFP-64 (
	Mask ROM version	Standard product	HD6433684H	HD6433684(***)H	QFP-64 (F
			HD6433684FP	HD6433684(***)FP	LQFP-64 (
		Product with POR & LVDC	HD6433684GH	HD6433684G(***)H	QFP-64 (F
			HD6433684GFP	HD6433684G(***)FP	LQFP-64 (
H8/3683	Mask ROM version	Standard product	HD6433683H	HD6433683(***)H	QFP-64 (F
			HD6433683FP	HD6433683(***)FP	LQFP-64 (
		Product with POR & LVDC	HD6433683GH	HD6433683G(***)H	QFP-64 (F
			HD6433683GFP	HD6433683G(***)FP	LQFP-64 (
H8/3682	Mask ROM version	Standard product	HD6433682H	HD6433682(***)H	QFP-64 (F
			HD6433682FP	HD6433682(***)FP	LQFP-64 (
		Product with POR & LVDC	HD6433682GH	HD6433682G(***)H	QFP-64 (F
			HD6433682GFP	HD6433682G(***)FP	LQFP-64 (

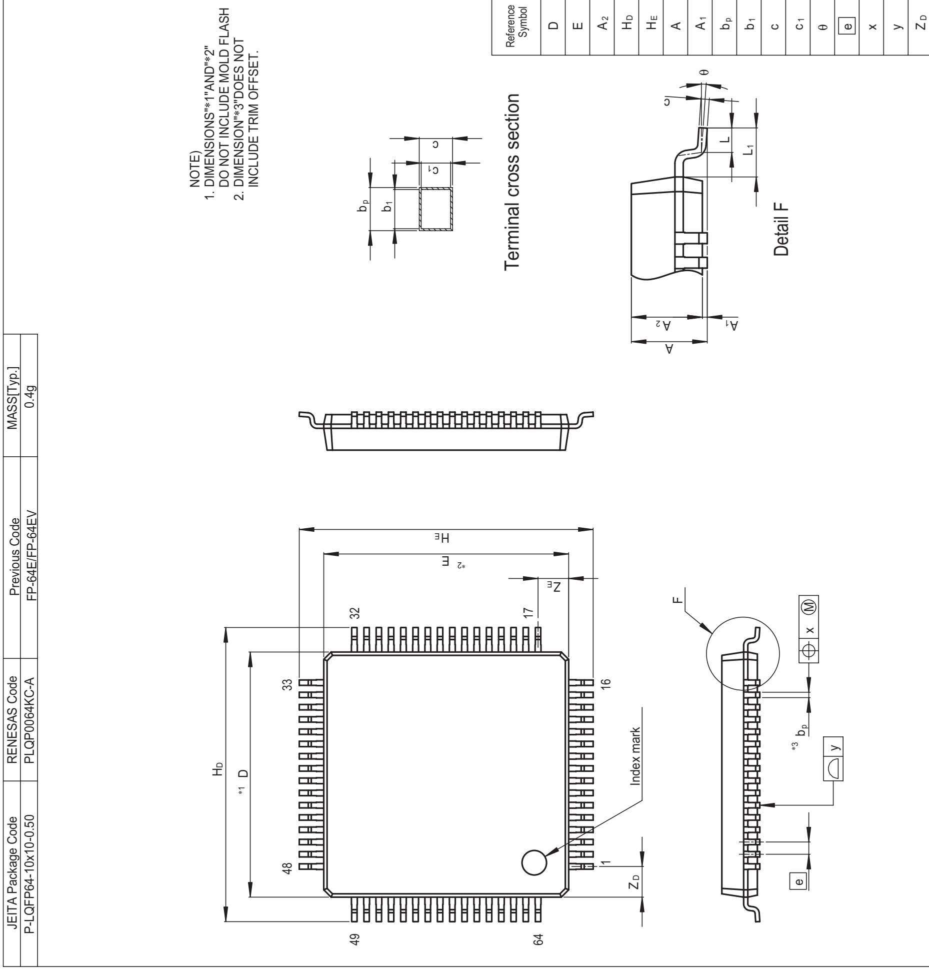


Figure D.1 FP-64E Package Dimensions

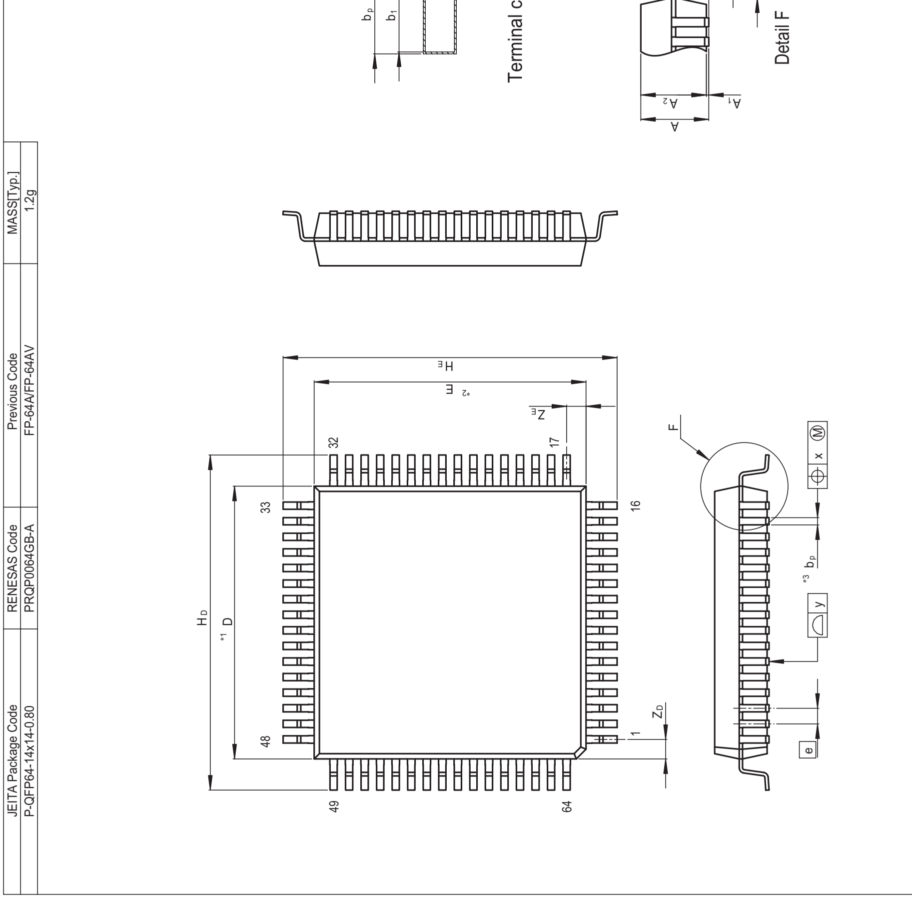


Figure D.2 FP-64A Package Dimensions

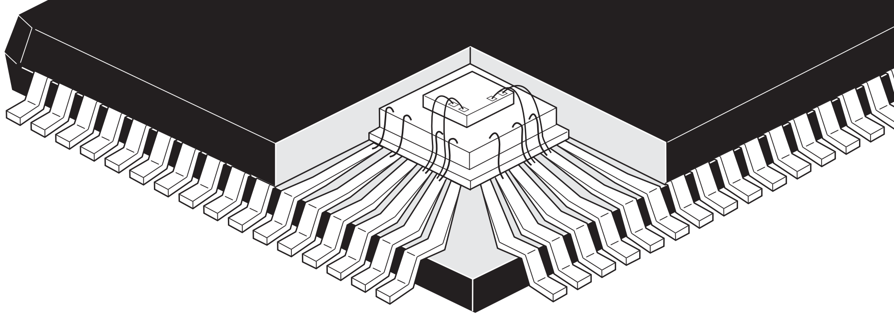
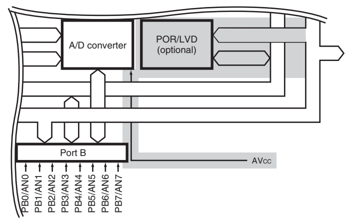
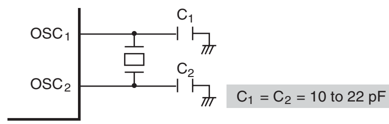
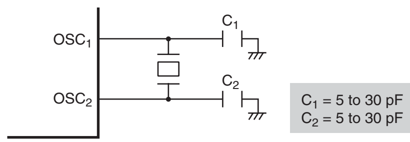
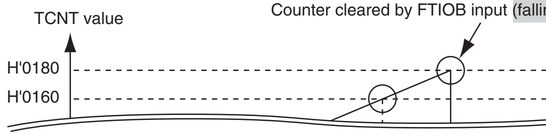


Figure E.1 EEPROM Stacked-Structure Cross-Sectional View

		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 o 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. 7. Use channel 1 of the SCI3 (P21/RXD, P22/TXD) i board programming mode by boot mode. Note has been deleted.
Section 1 Overview Figure 1.2 Internal Block Diagram of H8/3687N (EEPROM Stacked Version)	4	
Section 5 Clock Pulse Generators Figure 5.3 Typical Connection to Crystal Resonator	70	
Figure 5.5 Typical Connection to Ceramic Resonator	71	

		ϕ_{OSC} = 4 to 20 MHz, clear NESEL to (
Section 8 RAM	107	Note: * When the E7 or E8 is used, area H'F780 to H'F must not be accessed.
Section 13 Timer Z Figure 13.17 Example of Input Capture Operation	208	
13.4.4 Synchronous Operation	211	Figure 13.20 shows an example of synchronous opera this example, synchronous operation has been selecte FTIOB0 and FTIOB1 have been designated for PWM r GRA_0 compare match has been set as the channel 0 clearing source, and synchronous clearing has been s channel 1 counter clearing source. In addition, the san clock has been set as the counter input clock for chanr channel 1. Two-phase PWM waveforms are output fro FTIOB0 and FTIOB1.
13.4.5 PWM Mode	213	Figure 13.22 shows an example of operation in PWM r The output signals go to 1 and TCNT is reset at compa match A, and the output signals go to 0 at compare ma C, and D (TOB, TOC, and TOD = 1, POLB, POLC, anc = 0).
	214	Figures 13.24 (when TOB, TOC, and TOD = 1, POLB, and POLD = 0) and 13.25 (when TOB, TOC, and TOD POLB, POLC, and POLD = 1) show examples of the o PWM waveforms with duty cycles of 0% and 100% in l mode.

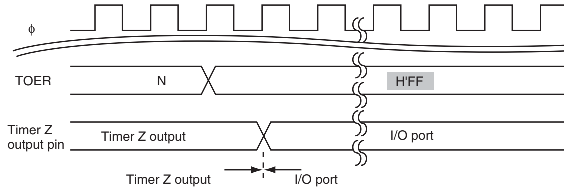
<table>
<tr><td>Figure 13.45 Example of Output Disable Timing of Timer Z by External Trigger</td><td>238</td><td>
</td></tr>
<tr><td>Section 14 Watchdog Timer
14.2.1 Timer Control/Status Register WD (TCSRWD)</td><td>252</td><td><table><tr><th>Bit</th><th>Bit Name</th><th>Description</th></tr><tr><td>4</td><td>TCSRWE</td><td>Timer Control/Status Register WD W
Enable</td></tr></table></td></tr>
<tr><td>Section 17 I²C Bus Interface 2 (IIC2)
17.3.5 I²C Bus Status Register (ICSR)</td><td>314</td><td><table><tr><th>Bit</th><th>Bit Name</th><th>Description</th></tr><tr><td>3</td><td>STOP</td><td>Stop Condition Detection Flag
[Setting conditions]
• In master mode, when a stop co
detected after frame transfer
• In slave mode, when a stop con
detected after the general call a
the first byte slave address, nex
detection of start condition, acc
the address set in SAR</td></tr></table></td></tr>
<tr><td>17.7 Usage Notes</td><td>336</td><td>Added</td></tr>
<tr><td>Section 18 A/D Converter
18.3.1 A/D Data Registers A to D (ADDRA to ADDRD)</td><td>340</td><td>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.</td></tr>
</table>

Table 23.2 DC Characteristics (1)

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, RXD2, SCL, SDA, P10 to P12, : P85 to P87, PB0 to PB7	V_{CC} = 4.0 to 5.5 V	–
				–

398

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 opera
Sleep mode 2		Only timers opera (φOSC/64)

PB0 to PB7

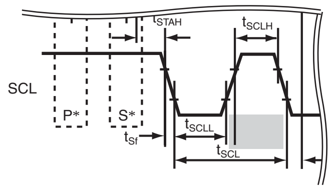
417

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 ope
Sleep mode 2		Only timers ope (ϕOSC/64)

Table 23.16 A/D Converter Characteristics — 424

Item	Test Condition	V
		M
Conversion time (single mode)	AV_{CC} = 3.3 to 5.5 V	13

Figure 23.4 I²C Bus Interface Input/Output Timing — 429



Appendix D Package Dimensions — 488, 489 — Swapped with new ones.

C

E

F

L

O

P

R

S

V

W

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8/3687 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.

RenesasTechnology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, J

RENESAS SALES OFFICES

http://www.ren

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

H8/3687Group
Hardware Manual

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

REJ09B0027-0500