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

H8/36024Group, H8/36014Group

Hardware Manual

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

H8/36024F	HD64F36024, HD64F36024G,
H8/36022F	HD64F36022, HD64F36022G,
H8/36014F	HD64F36014, HD64F36014G,
H8/36012F	HD64F36012, HD64F36012G,
H8/36024	HD64336024, HD64336024G,
H8/36023	HD64336023, HD64336023G,
H8/36022	HD64336022, HD64336022G,
H8/36014	HD64336014, HD64336014G,
H8/36013	HD64336013, HD64336013G,
H8/36012	HD64336012, HD64336012G,
H8/36011	HD64336011, HD64336011G,
H8/36010	HD64336010, HD64336010G

Renesas Electronics
www.renesas.com

Rev.4.00 2005.09

Generally, the input pins of CMOS products are high-impedance input pins. If un
are in their open states, intermediate levels are induced by noise in the vicinity, a
through current flows internally, and a malfunction may occur.

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

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

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

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

i) Feature

ii) Input/Output Pin

iii) Register Description

iv) Operation

v) Usage Note

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

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

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

11. Index

microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8/36024 Group and H8/36014 Group to the target 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 i on the CPU, system control functions, peripheral functions and electrical characteristi
- In order to understand the details of the CPU's functions

  Read the H8/300H Series Software Manual.
- In order to understand the details of a register when its name is known

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

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

Notes:

When using the on-chip emulator (E7, E8) for H8/36014 program development and debu the following restrictions must be noted.

1. The $\overline{\text{NMI}}$ pin is reserved for the E7 or E8, and cannot be used.
2. Area H’7000 to H’7FFF is used by the E7 or E8, and is not available to the user.
3. Area H’F780 to H’FB7F must on no account be accessed.

H8/36024 Group and H8/36014 Group manuals:

Document Title	Docume
H8/36024 Group, H8/36014 Group Hardware Manual	This ma
H8/300H Series Software Manual	REJ09B

User's manuals for development tools:

Document Title	Docume
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	REJ10B
Microcomputer Development Environment System H8S, H8/300 Series Simulator/Debugger User's Manual	ADE-70
H8S, H8/300 Series High-Performance Embedded Workshop 3, Tutorial	REJ10B
H8S, H8/300 Series High-Performance Embedded Workshop 3, User's Manual	REJ10B

Application notes:

Document Title	Docume
H8S, H8/300 Series C/C++ Compiler Package Application Note	REJ05B
Single Power Supply F-ZTAT™ On-Board Programming	ADE-50

Section 9 I/O Ports

Section 10 Timer V

Section 11 Timer W

Section 12 Watchdog Timer

Section 13 Serial Communication Interface 3 (SCI3)

Section 13 Serial Communication Interface 3 (SCI3)

Section 14 A/D Converter

Section 15 Power-On Reset and Low-Voltage Detection Circuits (Optional)

Section 18 Electrical Characteristics

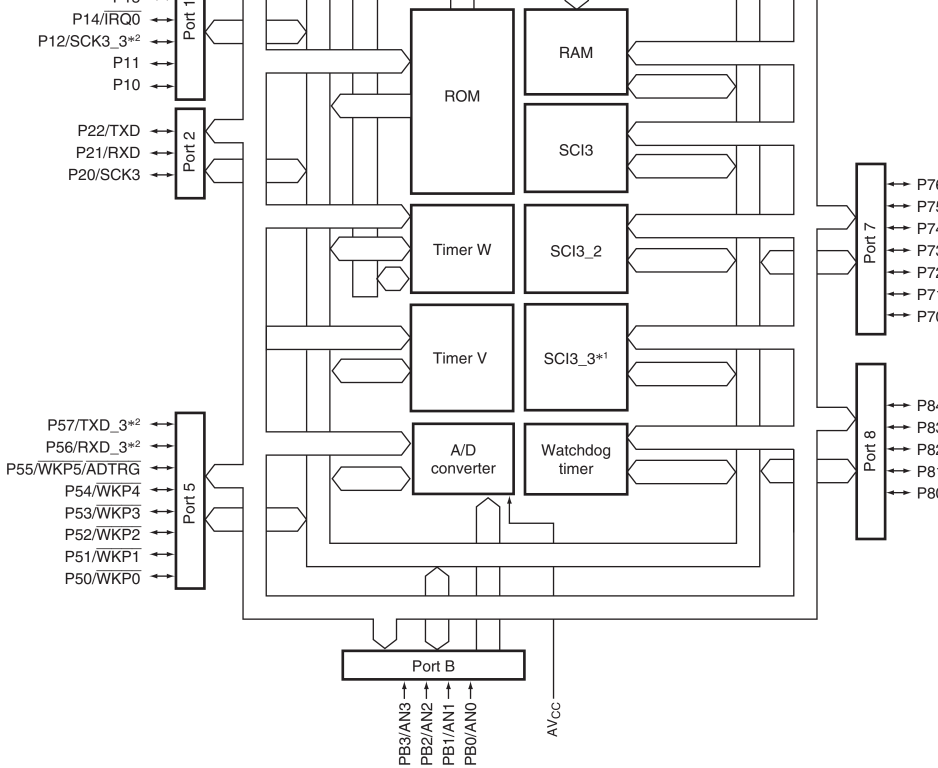
Appendix

— Timer V (8-bit timer)
— Timer W (16-bit timer)
— Watchdog timer
— SCI3 (Asynchronous or clocked synchronous serial communication interface)
— 10-bit A/D converter

- On-chip memory

Product Classification		Model		ROM	RAM
		Standard Version	On-Chip Power-On Reset and Low-Voltage Detecting Circuit Version		
Flash memory version (F-ZTAT™ version)	H8/36024F	HD64F36024	HD64F36024G	32 kbytes	2,048
	H8/36022F	HD64F36022	HD64F36022G	16 kbytes	2,048
	H8/36014F	HD64F36014	HD64F36014G	32 kbytes	2,048
	H8/36012F	HD64F36012	HD64F36012G	16 kbytes	2,048
Masked ROM version	H8/36024	HD64336024	HD64336024G	32 kbytes	1,024
	H8/36023	HD64336023	HD64336023G	24 kbytes	1,024
	H8/36022	HD64336022	HD64336022G	16 kbytes	512 by
	H8/36014	HD64336014	HD64336014G	32 kbytes	1,024
	H8/36013	HD64336013	HD64336013G	24 kbytes	1,024
	H8/36012	HD64336012	HD64336012G	16 kbytes	512 by
	H8/36011	HD64336011	HD64336011G	12 kbytes	512 by
	H8/36010	HD64336010	HD64336010G	8 kbytes	512 by

LQFP-48	FP-48F	10.0 × 10.0 mm	0.65 mm
LQFP-48	FP-48B	7.0 × 7.0 mm	0.5 mm
QFN-48	TNP-48	7.0 × 7.0 mm	0.5 mm



Notes: 1. The SCI3_3 function is incorporated in the H8/36024.
2. Since the SCI3_3 function is not incorporated in the H8/36014, the SCK3_3, RXD_3, and TXD_3 pins are not r
3. Can also be used for the E7 or E8 emulator.

Figure 1.1 Internal Block Diagram

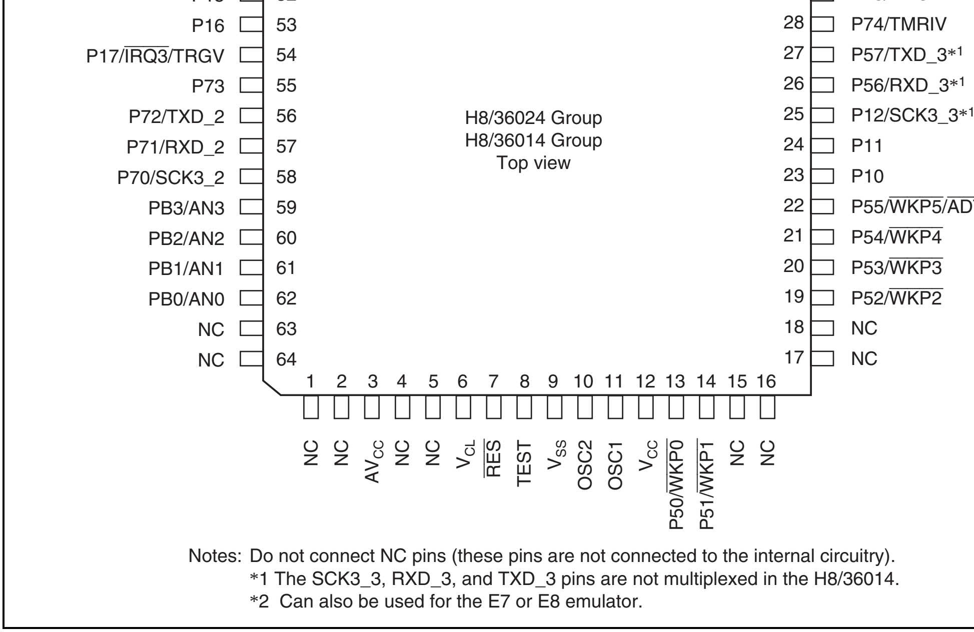
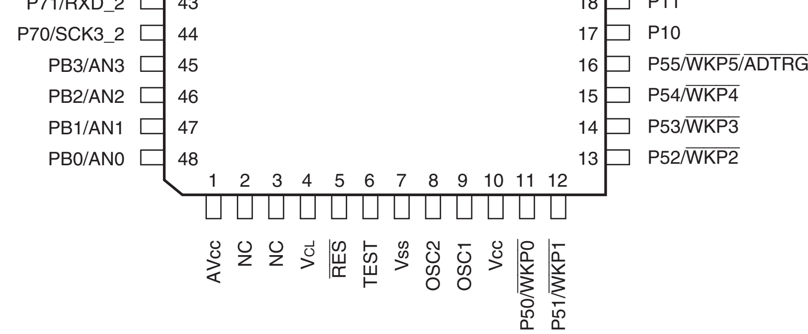


Figure 1.2 Pin Arrangement (FP-64E)



Notes: Do not connect NC pins (these pins are not connected to the internal circuitry).
*1 The SCK3_3, RXD_3, and TXD_3 pins are not multiplexed in the H8/36014.
*2 Can also be used for the E7 or E8 emulator.

Figure 1.3 Pin Arrangement (FP-48F, FP-48B, TNP-48)

					system power supply (0V).
	AV_{CC}	3	1	Input	Analog power supply pin fo converter. When the A/D c is not used, connect this pi system power supply.
	V_{CL}	6	4	Input	Internal step-down power s pin. Connect a capacitor o 0.1 μF between this pin an pin for stabilization.
Clock pins	OSC1	11	9	Input	These pins connect to a cr ceramic resonator for syste clocks, or can be used to i external clock. See section 5, Clock Pulse Generators, for a typical connection.
	OSC2	10	8	Output	
System control	$\overline{\text{RES}}$	7	5	Input	Reset pin. The pull-up resi 150 kΩ) is incorporated. W driven low, the chip is rese
	TEST	8	6	Input	Test pin. Connect this pin t
Interrupt pins	$\overline{\text{NMI}}$	35	25	Input	Non-maskable interrupt re input pin. Be sure to pull-u pull-up resistor.
	$\overline{\text{IRQ0}}$, $\overline{\text{IRQ3}}$	51, 54	37, 40	Input	External interrupt request i pins. Can select the rising edge.
	$\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$	13, 14, 19 to 22	11 to 16	Input	External interrupt request i pins. Can select the rising edge.

Timer W	FTCI	36	26	Input	External event input pin.
	FTIOA to FTIOD	37 to 40	27 to 30	I/O	Output compare output/ i capture input/ PWM outpu
Serial communication interface (SCI)	TXD, TXD_2, TXD_3*	46, 56, 27	36, 42, 21	Output	Transmit data output pin
	RXD, RXD_2, RXD_3*	45, 57, 26	35, 43, 20	Input	Receive data input pin
	SCK3, SCK3_2, SCK3_3*	44, 58, 25	34, 44, 19	I/O	Clock I/O pin
A/D converter	AN3 to AN0	59 to 62	45 to 48	Input	Analog input pin
	$\overline{\text{ADTRG}}$	22	16	Input	A/D converter trigger inpu
I/O ports	PB3 to PB0	59 to 62	45 to 48	Input	4-bit input port.
	P17 to P14, P12 to P10	54 to 51, 25 to 23	40 to 37, 19 to 17	I/O	7-bit I/O port.
	P22 to P20	46 to 44	36 to 34	I/O	3-bit I/O port.
	P57 to P50	27, 26, 22 to 19, 14, 13	21, 20, 16 to 11	I/O	8-bit I/O port
	P76 to P70	30 to 28, 55 to 58	24 to 22, 41 to 44	I/O	7-bit I/O port
	P84 to P80	40 to 36	30 to 26	I/O	5-bit I/O port.
E10T	E10T _0, E10T _1, E10T _2	41, 42, 43	31, 32, 33		Interface pin for the E10T E7 emulator

Note: * The SCK3_3, RXD_3, and TXD_3 pins are not multiplexed in the H8/36014.

- General-register architecture
  - Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-b
- 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

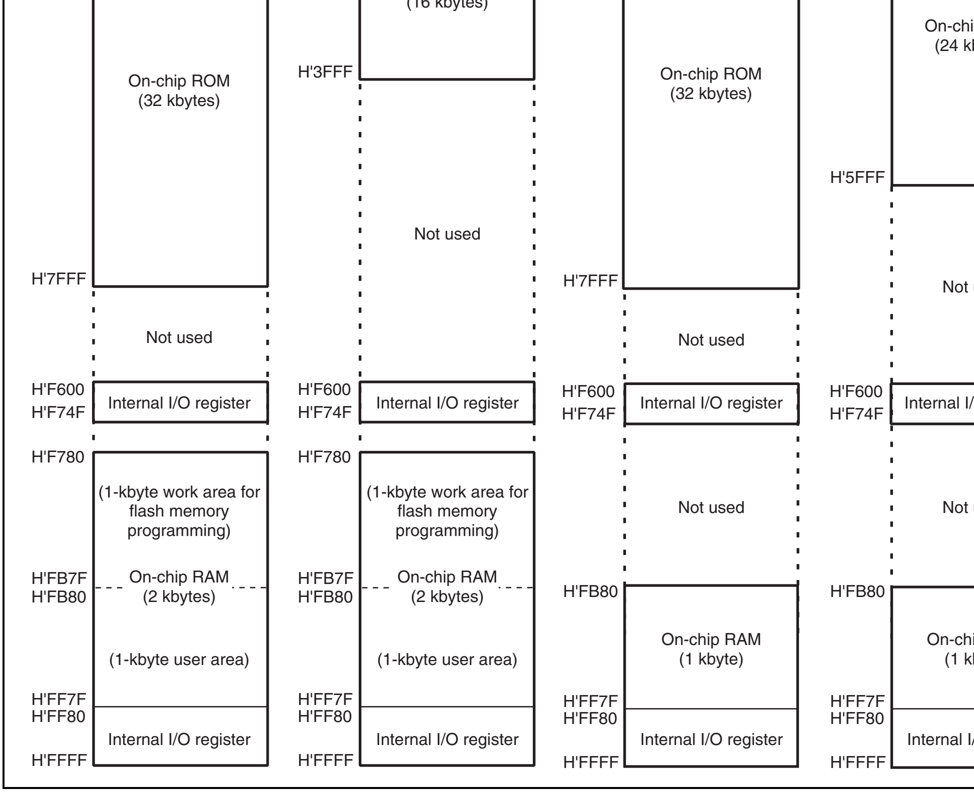


Figure 2.1 Memory Map (1)

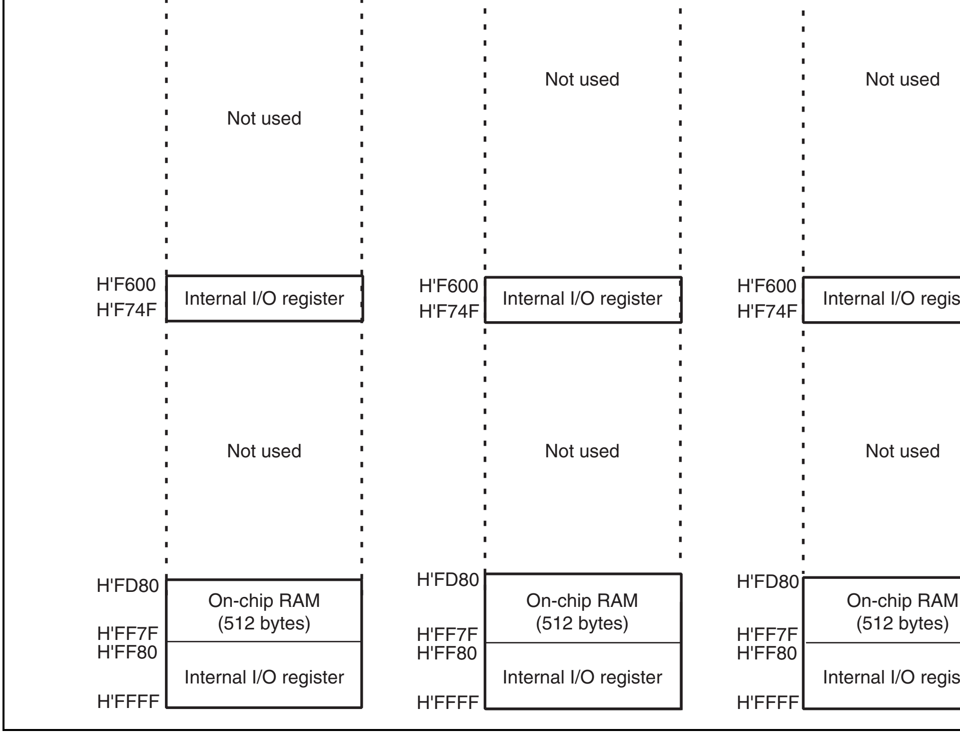


Figure 2.1 Memory Map (2)

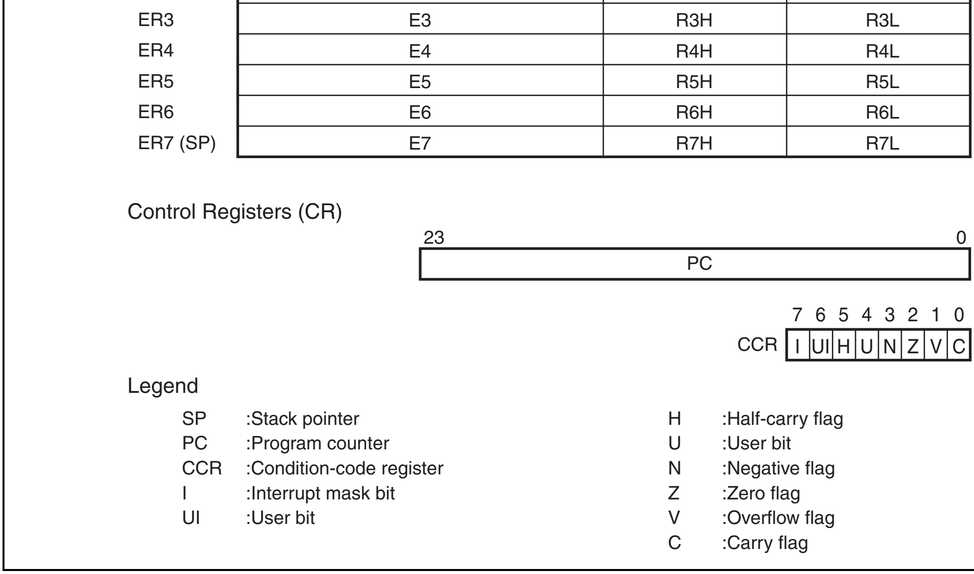


Figure 2.2 CPU Registers

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

The usage of each register can be selected independently.

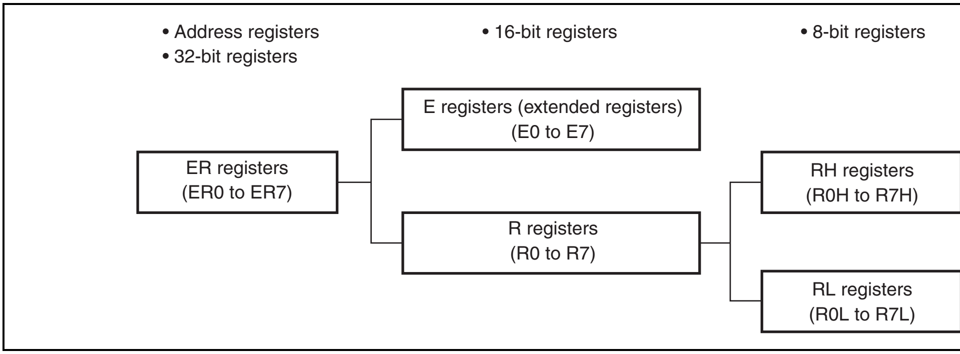


Figure 2.3 Usage of General Registers

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

Figure 2.4 Relationship between Stack Pointer and Stack Area

2.2.2 Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute. Th
of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. (W
instruction is fetched, the least significant PC bit is regarded as 0). The PC is initialized w
start address is loaded by the vector address generated during reset exception-handling se

2.2.3 Condition-Code Register (CCR)

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

Some instructions leave flag bits unchanged. Operations can be performed on the CCR bi
LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used as b
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, or NEG.B instruction is executed, this flag is s 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 s there is a carry or borrow at bit 11, and cleare otherwise. When the ADD.L, SUB.L, CMP.L, instruction is executed, the H flag is set to 1 if carry or borrow at bit 27, and cleared to 0 oth
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 sign bit.
2	Z	Undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to indicate non-zero data.
1	V	Undefined	R/W	Overflow Flag Set to 1 when an arithmetic overflow occurs, cleared to 0 at other times.
0	C	Undefined	R/W	Carry Flag Set to 1 when a carry occurs, and cleared to otherwise. Used by: • Add instructions, to indicate a carry • Subtract instructions, to indicate a borrow • Shift and rotate instructions, to indicate a The carry flag is also used as a bit accumulat manipulation instructions.

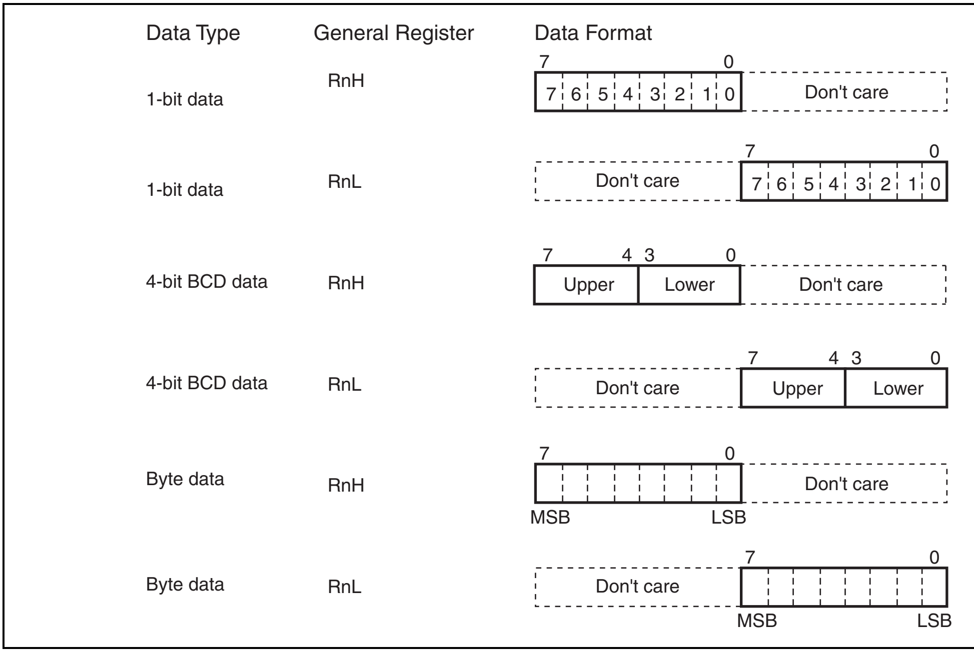


Figure 2.5 General Register Data Formats (1)

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)

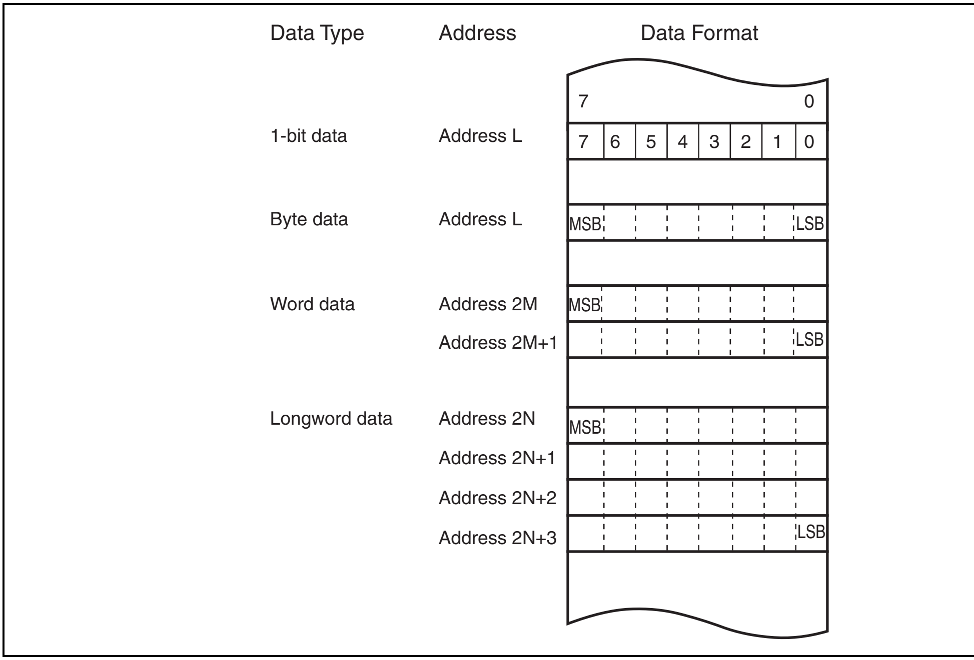


Figure 2.6 Memory Data Formats

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)

MOVFPE	B	(EAs) → Rd, Cannot be used in this LSI.
MOVTPE	B	Rs → (EAs) Cannot be used in this LSI.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical t MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, E
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is identic 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 o 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-b
DAA DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction result in a general r referring to the CCR to produce 4-bit BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general regist 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: e bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 1 16-bit quotient and 16-bit remainder.

Note: * Refers to the operand size.

B: Byte

W: Word

L: Longword

Instruction	Size*	Function
NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic complement) of data ir 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 b

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

Table 2.4 Logic Operations Instructions

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

SHLR		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 operand. number is specified by 3-bit immediate data or the lower three b general register.
BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operand ar or clears the Z flag accordingly. The bit number is specified by 3 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 r 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 gener register or memory operand and stores the result in the carry fla 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 m 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 genera 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 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 reg memory operand.
BIST	B	¬ C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit i 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 by word access.
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR with immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR with immediate data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR 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, tra data for the number of bytes set in R4L or R4 to the address loc in ER6.

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

2.4.2 Basic Instruction Formats

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

Figure 2.7 shows examples of instruction formats.

- Operation Field

  Indicates the function of the instruction, the addressing mode, and the operation to be out on the operand. The operation field always includes the first four bits of the instru 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. A2 address or displacement is treated as a 32-bit data in which the first 8 bits are 0 (H'00)
- Condition Field

  Specifies the branching condition of Bcc instructions.

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

op	cc	EA(disp)	BRA d:8

Figure 2.7 Instruction Formats

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

Table 2.10 Addressing Modes

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

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

(4) Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn

- Register indirect with post-increment—@ERn+

  The register field of the instruction code specifies an address register (ERn) the lowe
  of which contains the address of a memory operand. After the operand is accessed, 1
  added to the address register contents (32 bits) and the sum is stored in the address re
  The value added is 1 for byte access, 2 for word access, or 4 for longword access. Fo
  or longword access, the register value should be even.
- Register indirect with pre-decrement—@-ERn

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

(5) Absolute Address—@aa:8, @aa:16, @aa:24

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

(6) Immediate—#xx:8, #xx:16, or #xx:32

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

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

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

This mode is used in the BSR instruction. An 8-bit or 16-bit displacement contained in th
instruction is sign-extended and added to the 24-bit PC contents to generate a branch add
PC value to which the displacement is added is the address of the first byte of the next ins
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.

(8) 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).

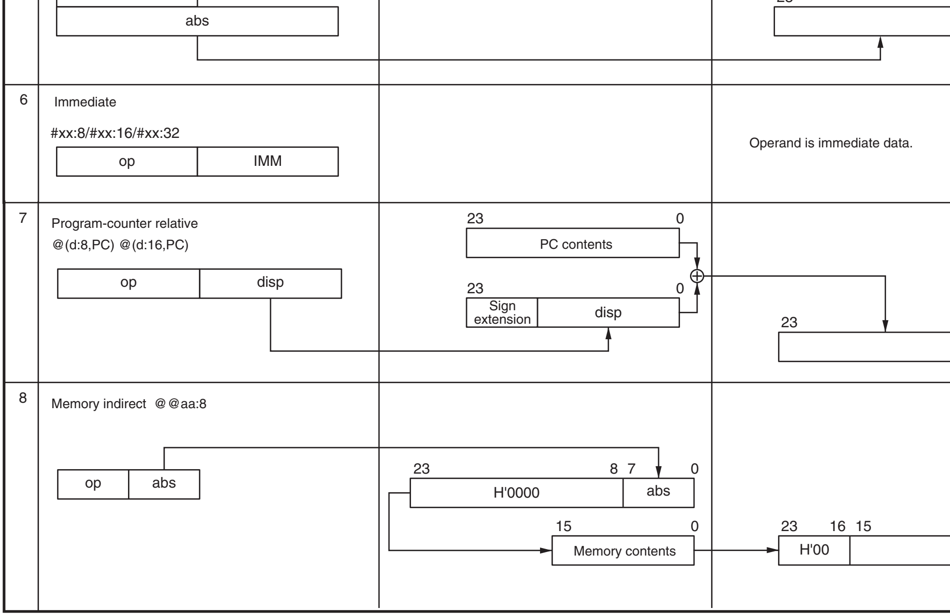
Note that the first part of the address range is also the exception vector area.

2.5.2 Effective Address Calculation

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

Table 2.12 Effective Address Calculation (1)

No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (E
1	Register direct(Rn) op | rm | rn		Operand is general register con
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

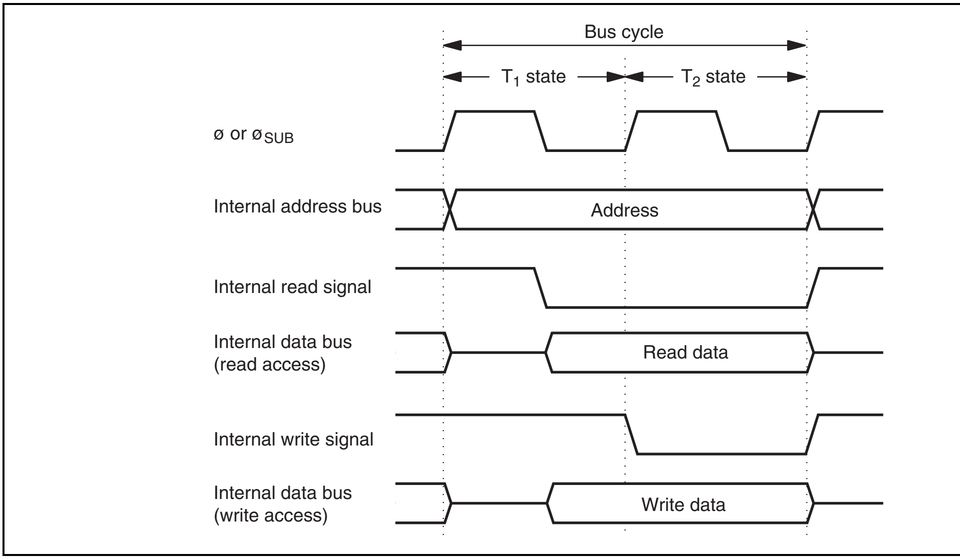


Figure 2.9 On-Chip Memory Access Cycle

module.

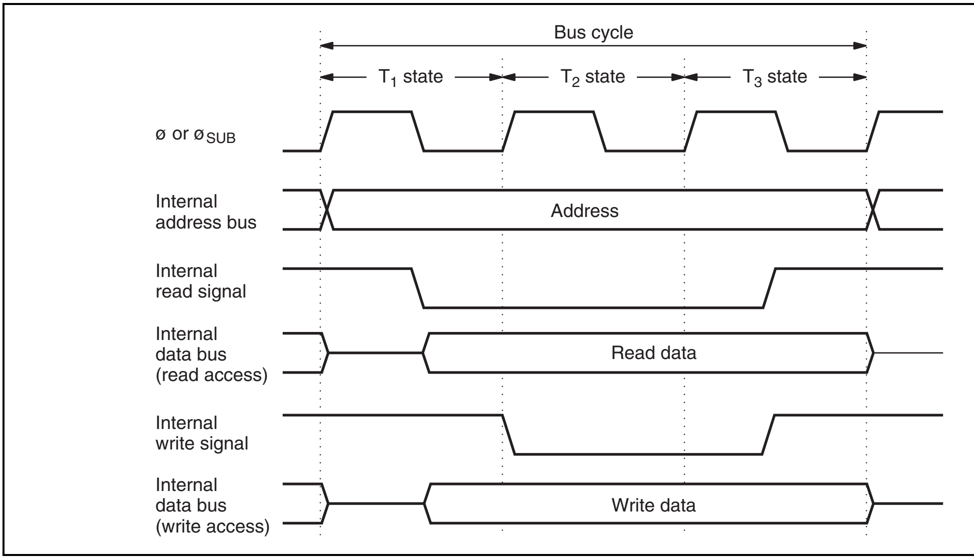


Figure 2.10 On-Chip Peripheral Module Access Cycle (3-State Access)

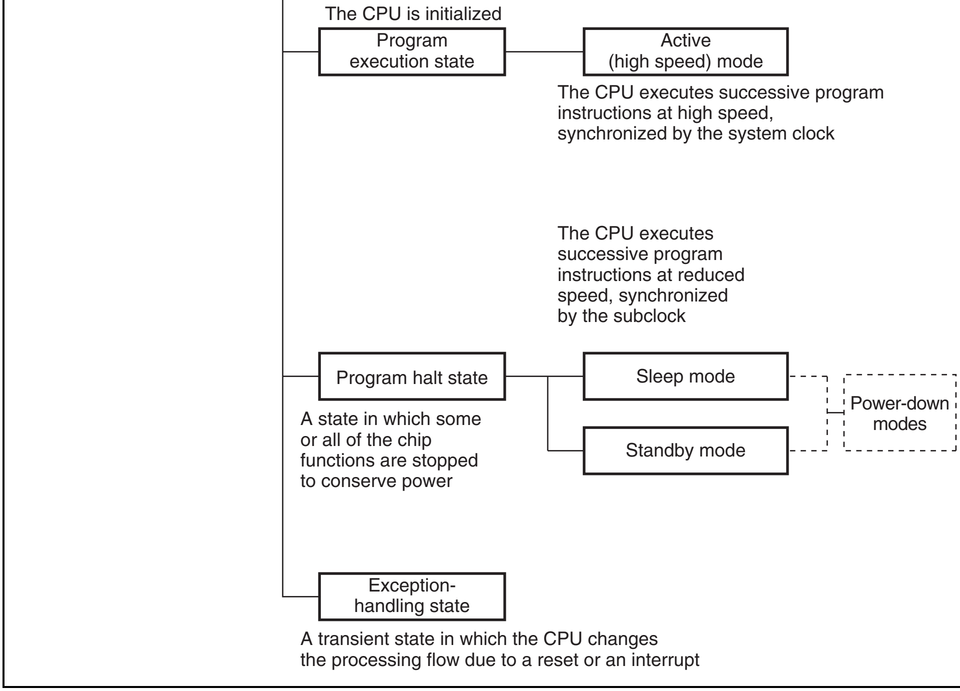


Figure 2.11 CPU Operation States

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 or I/O registers areas available to the user. When data is transferred from CPU to empty area 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 R which starts from the address indicated by R5, to the address indicated by R6. Set R4L an 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 addr byte units, manipulate the data of the target bit, and write data to the same address again i units. Special care is required when using these instructions in cases where two registers a assigned to the same address or when a bit is directly manipulated for a port, because this 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 instructio
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 ti register. As a result, bits other than the intended bit in the timer counter may be modifie modified value may be written to the timer load register.

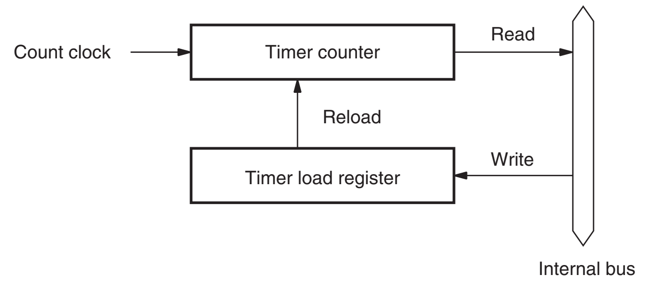


Figure 2.13 Example of Timer Configuration with Two Registers Allocate Same Address

PCR5	0	0	1	1	1	1	1
PDR5	1	0	0	0	0	0	0

- BSET instruction executed

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

The BSET instruction is executed for port 5.

- After executing BSET

	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 PDR 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.

Input/output	Input	Input	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1
PDR5	1	0	0	0	0	0	0
RAM0	1	0	0	0	0	0	0

- BSET instruction executed

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

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

- After executing BSET

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

The work area (RAM0) value is written to PDR5

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

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

- BCLR instruction executed

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

The BCLR instruction is executed for PCR5.

- After executing BCLR

	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 w register, the CPU reads a value of H'FF, even though the PCR5 value is actually H'3F
2. Next, the CPU clears bit 0 in the read data to 0, changing the data to H'FE.
3. Finally, H'FE is written to PCR5 and BCLR instruction execution ends.

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

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 w
(RAM0).

- After executing BCLR

```
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

- Interrupts

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

3.1 Exception Sources and Vector Address

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

Table 3.1 Exception Sources and Vector Address

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

	/compare match A Timer W input capture B /compare match B Timer W input capture C /compare match C Timer W input capture D /compare match D Timer W overflow		
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
A/D converter	A/D conversion end	25	H'0032 to H'0033
SCI3_2	SCI3_2 receive data full SCI3_2 transmit data empty SCI3_2 transmit end SCI3_2 receive error	32	H'0040 to H'0041
SCI3_3*[2]	SCI3_3 receive data full SCI3_3 transmit data empty SCI3_3 transmit end SCI3_3 receive error	34	H'0044 to H'0045

Notes: 1. A low-voltage detection interrupt is enabled only in the product with an on-chip on reset and low-voltage detection circuit.
2. The SCI3_3 function is incorporated in the H8/36024.

3.2.1 Interrupt Edge Select Register 1 (IEGR1)

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	–	Reserved This bit is always read as 0.
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, 1	—	All 0	—	Reserved These bits are always read as 0.
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 $\overline{\text{WKP5}}$ ($\overline{\text{ADTRG}}$) pin input is de 1: Rising edge of $\overline{\text{WKP5}}$ ($\overline{\text{ADTRG}}$) pin input is de
4	WPEG4	0	R/W	WKP4 Edge Select 0: Falling edge of $\overline{\text{WKP4}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP4}}$ pin input is detected
3	WPEG3	0	R/W	WKP3 Edge Select 0: Falling edge of $\overline{\text{WKP3}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP3}}$ pin input is detected
2	WPEG2	0	R/W	WKP2 Edge Select 0: Falling edge of $\overline{\text{WKP2}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP2}}$ pin input is detected
1	WPEG1	0	R/W	WKP1Edge Select 0: Falling edge of $\overline{\text{WKP1}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP1}}$ pin input is detected
0	WPEG0	0	R/W	WKP0 Edge Select 0: Falling edge of $\overline{\text{WKP0}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP0}}$ pin input is detected

5	IENWP	0	R/W	Wakeup Interrupt Enable This bit is an enable bit, which is common to the WKP5 to WKP0. When the bit is set to 1, interru requests are enabled.
4	—	1	—	Reserved This bit is always read as 1.
3	IEN3	0	R/W	IRQ3 Interrupt Enable When this bit is set to 1, interrupt requests of th are enabled.
2, 1	—	All 0	—	Reserved These bits are always read as 0.
0	IEN0	0	R/W	IRQ0 Interrupt Enable When this bit is set to 1, interrupt requests of th are enabled.

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

				[Clearing condition] When IRRDT is cleared by writing 0
6	—	0	—	Reserved This bit is always read as 0.
5, 4	—	All 1	—	Reserved These bits are always read as 1.
3	IRRI3	0	R/W	IRQ3 Interrupt Request Flag [Setting condition] When $\overline{IRQ3}$ pin is designated for interrupt input a designated signal edge is detected. [Clearing condition] When IRRI3 is cleared by writing 0
2, 1	—	All 0	—	Reserved These bits are always read as 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

				[Clearing condition] When IWPF5 is cleared by writing 0.
4	IWPF4	0	R/W	WKP4 Interrupt Request Flag [Setting condition] When $\overline{WKP4}$ pin is designated for interrupt inpu designated signal edge is detected. [Clearing condition] When IWPF4 is cleared by writing 0.
3	IWPF3	0	R/W	WKP3 Interrupt Request Flag [Setting condition] When $\overline{WKP3}$ pin is designated for interrupt inpu designated signal edge is detected. [Clearing condition] When IWPF3 is cleared by writing 0.
2	IWPF2	0	R/W	WKP2 Interrupt Request Flag [Setting condition] When $\overline{WKP2}$ pin is designated for interrupt inpu 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 $\overline{WKP1}$ pin is designated for interrupt inpu designated signal edge is detected. [Clearing condition] When IWPF1 is cleared by writing 0.

3.3 Reset Exception Handling

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

The reset exception handling sequence is as follows. However, for the reset exception ha
sequence of the product with on-chip power-on reset circuit, refer to section 15, Power-O
and Low-Voltage Detection Circuits (Optional).

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

CCR.

(2) IRQ3 to IRQ0 Interrupts

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

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

(3) WKP5 to WKP0 Interrupts

WKP5 to WKP0 interrupts are requested by input signals to pins $\overline{\text{WKP}}$5 to $\overline{\text{WKP}}$0. Thes
interrupts have the same vector addresses, and are detected individually by either rising
sensing or falling edge sensing, depending on the settings of bits WPEG5 to WPEG0 in

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

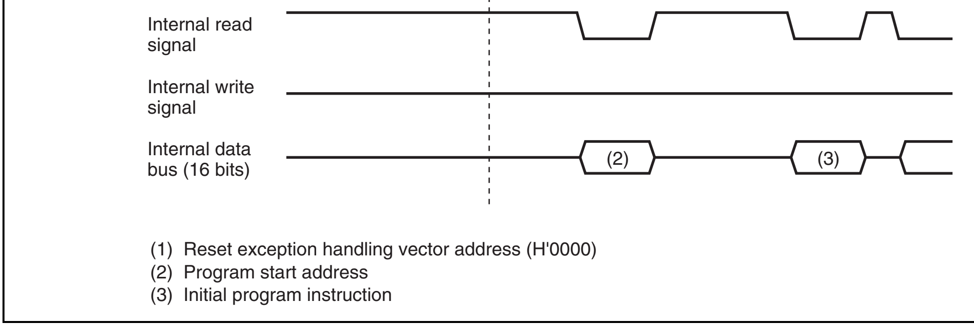


Figure 3.1 Reset Sequence

3.4.2 Internal Interrupts

Each on-chip peripheral module has a flag to show the interrupt request status and the ena
enable or disable the interrupt. For direct transfer interrupt requests generated by executio
SLEEP instruction, this function is included in IRR1 and IENR1.

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

3. The CPU accepts the NMI or address break without depending on the I bit value. Ot 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 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 fir instruction of the interrupt handling-routine is executed.

Table 3.2 Interrupt Wait States

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

Note: * Not including EEPMOV instruction.

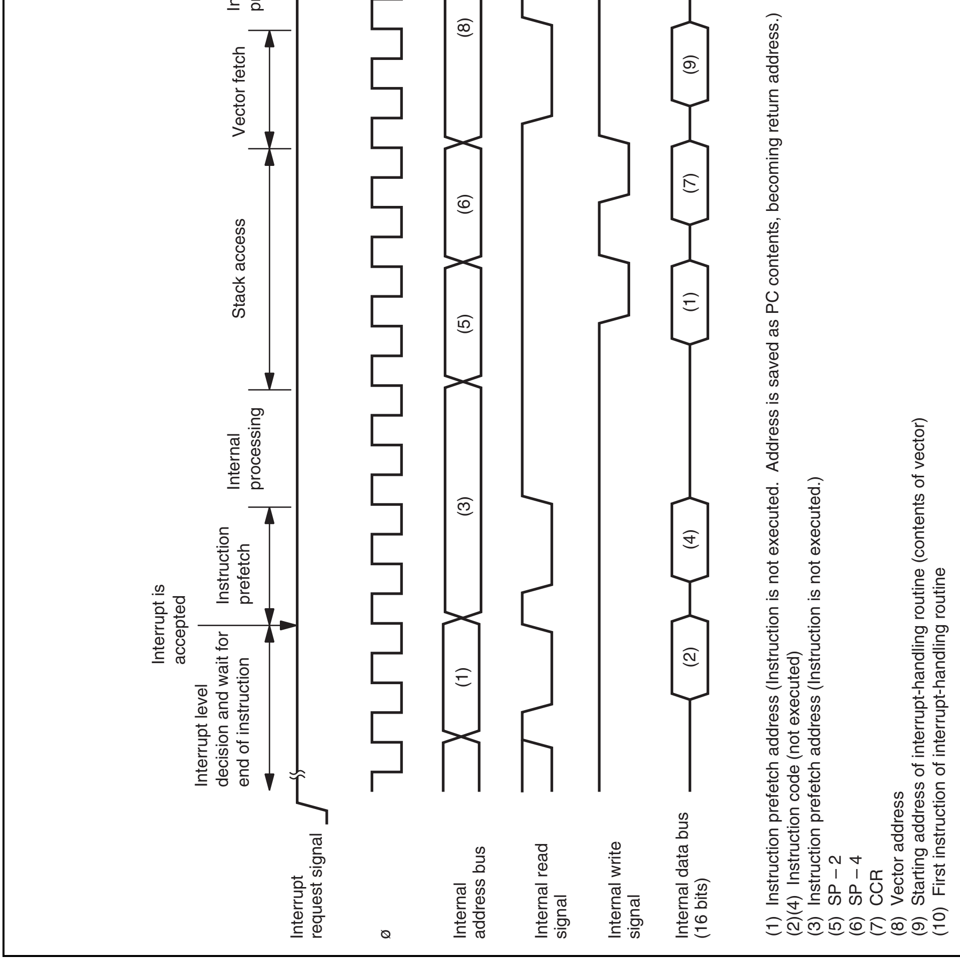


Figure 3.3 Interrupt Sequence

3.5.2 Notes on Stack Area Use

When word data is accessed, the least significant bit of the address is regarded as 0. Acce stack always takes place in word size, so the stack pointer (SP: R7) should never indicate address. Use PUSH Rn (MOV.W Rn, @–SP) or POP Rn (MOV.W @SP+, Rn) to save o 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, I $\overline{\text{IRQ0}}$, and $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$, the interrupt request flag may be set to 1.

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

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.

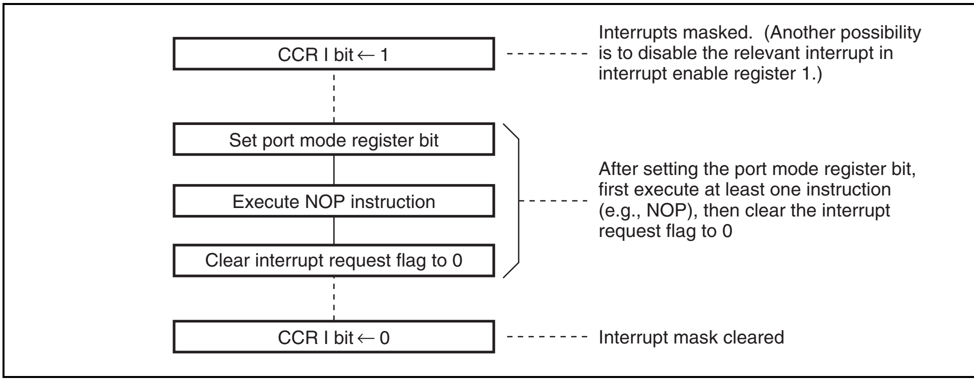


Figure 3.4 Port Mode Register Setting and Interrupt Request Flag Clearing Pro

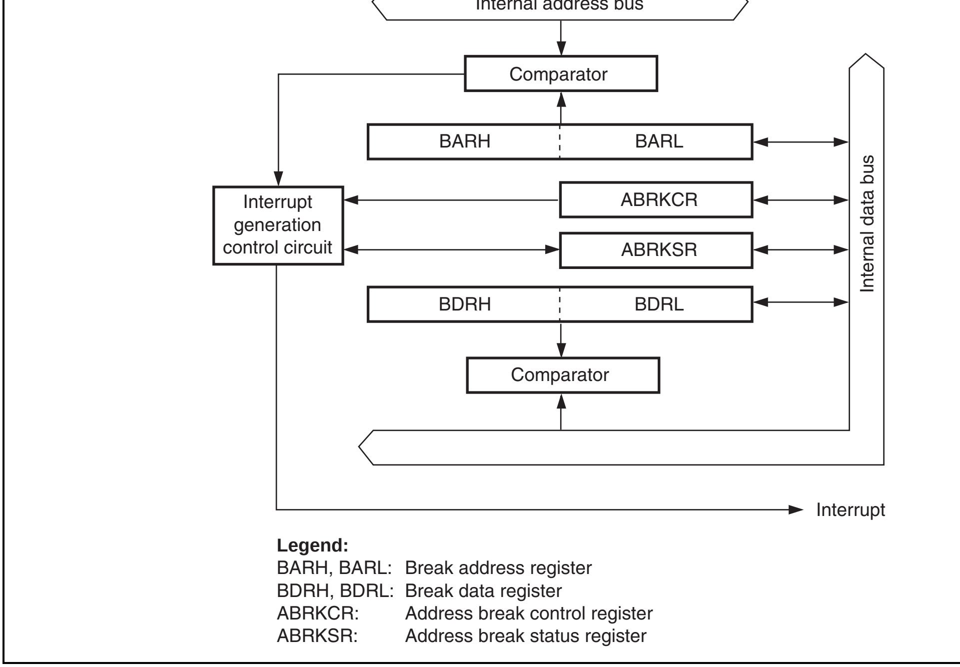


Figure 4.1 Block Diagram of Address Break

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)

				masked.
6 5	CSEL1 CSEL0	0 0	R/W R/W	Condition Select 1 and 0 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 3 2	ACMP2 ACMP1 ACMP0	0 0 0	R/W R/W R/W	Address Compare Condition Select 2 to 0 These bits comparison condition between the ad in BAR and the internal address bus. 000: Compares 16-bit addresses 001: Compares upper 12-bit addresses 010: Compares upper 8-bit addresses 011: Compares upper 4-bit addresses 1XX: Reserved (setting prohibited)
1 0	DCMP1 DCMP0	0 0	R/W R/W	Data Compare Condition Select 1 and 0 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 ar 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.

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

4.2 Operation

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

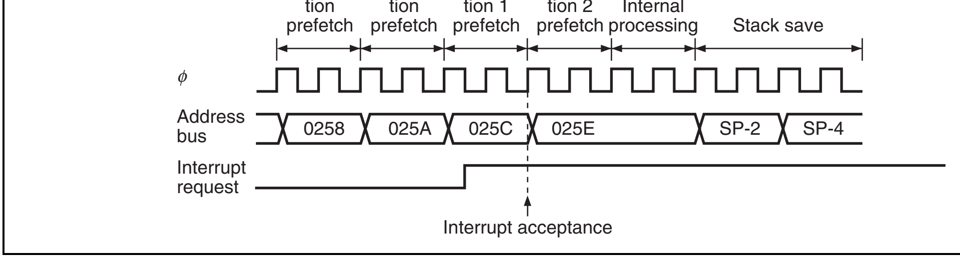


Figure 4.2 Address Break Interrupt Operation Example (1)

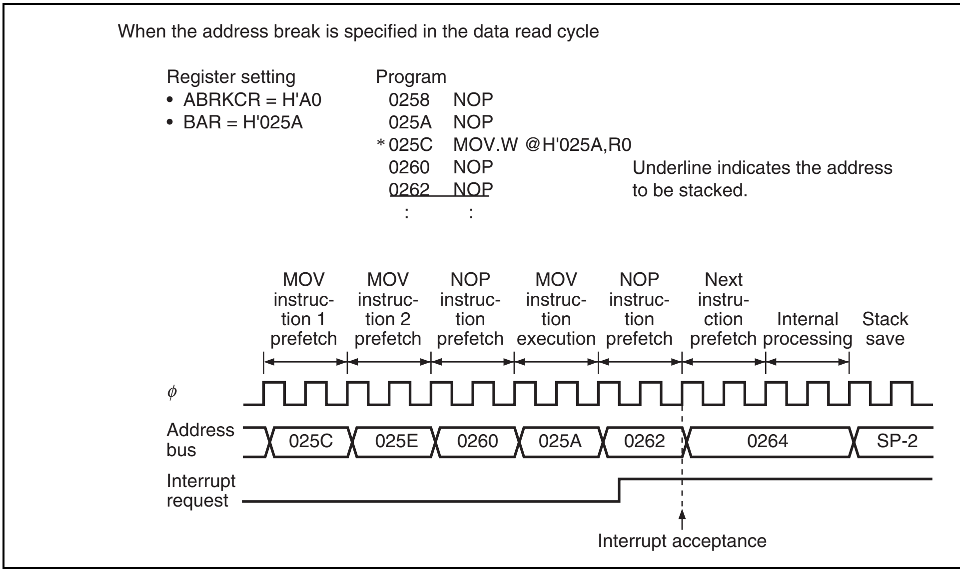


Figure 4.2 Address Break Interrupt Operation Example (2)

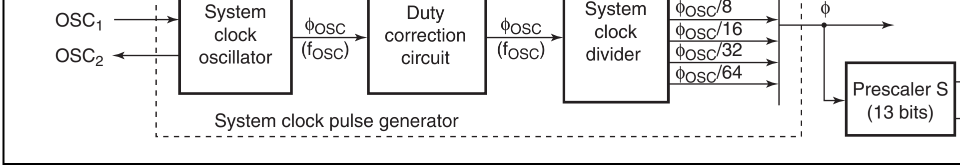


Figure 5.1 Block Diagram of Clock Pulse Generators

The basic clock signals that drive the CPU and on-chip peripheral modules are system c
The system clock is divided into φ/8192 to φ/2 by prescaler S and they are supplied to re
peripheral modules.

5.1 System Clock Generator

Clock pulses can be supplied to the system clock divider either by connecting a crystal o
resonator, or by providing external clock input. Figure 5.2 shows a block diagram of the
clock generator.

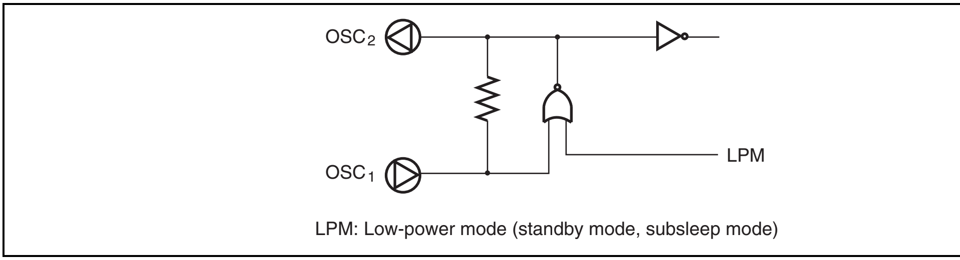


Figure 5.2 Block Diagram of System Clock Generator

Figure 5.3 Typical Connection to Crystal Resonator

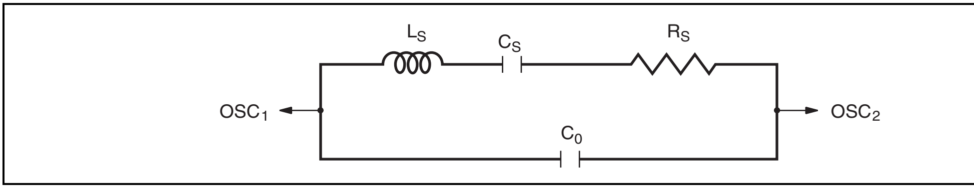


Figure 5.4 Equivalent Circuit of Crystal Resonator

Table 5.1 Crystal Resonator Parameters

Frequency (MHz)	2	4	8	10	16	20
R_s (max)	500 Ω	120 Ω	80 Ω	60 Ω	50 Ω	40 Ω
C_0 (max)	7 pF	7 pF	7 pF	7 pF	7 pF	7 pF

5.1.2 Connecting Ceramic Resonator

Figure 5.5 shows a typical method of connecting a ceramic resonator.

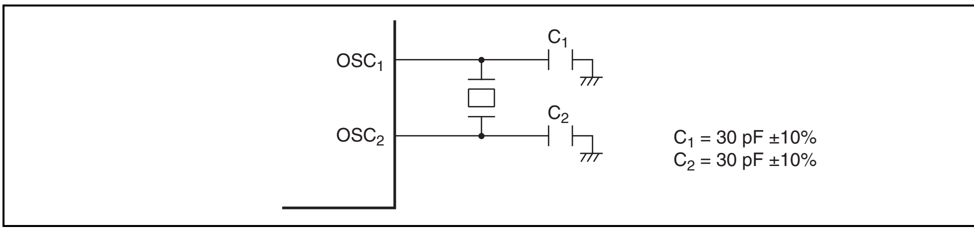


Figure 5.5 Typical Connection to Ceramic Resonator

Figure 5.6 Example of External Clock Input

5.2 Prescalers

5.2.1 Prescaler S

Prescaler S is a 13-bit counter using the system clock (ϕ) as its input clock. It is increme
per clock period. Prescaler S is initialized to H'0000 by a reset, and starts counting on ex
the reset state. In standby mode and subsleep mode, the system clock pulse generator sto
Prescaler S also stops and is initialized to H'0000. The CPU cannot read or write prescal

The output from prescaler S is shared by the on-chip peripheral modules. The divider ra
set separately for each on-chip peripheral function. In active mode and sleep mode, the c
to prescaler S is determined by the division factor designated by MA2 to MA0 in SYSC

5.3 Usage Notes

5.3.1 Note on Resonators

Resonator characteristics are closely related to board design and should be carefully eva
the user, referring to the examples shown in this section. Resonator circuit constants wil
depending on the resonator element, stray capacitance in its interconnecting circuit, and
factors. Suitable constants should be determined in consultation with the resonator elem
manufacturer. Design the circuit so that the resonator element never receives voltages ex
its maximum rating.

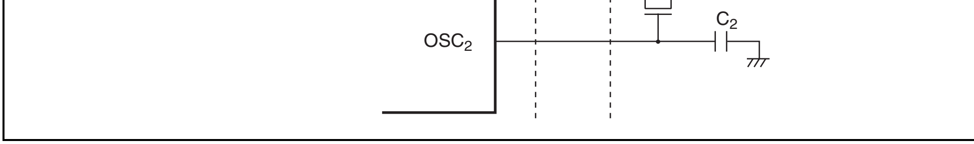


Figure 5.7 Example of Incorrect Board Design

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

- Standby mode

  The CPU and all on-chip peripheral modules halt.

- Subsleep mode

  The CPU and all on-chip peripheral modules halt. I/O ports keep the same states as b
  transition.

- 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: a transition is made to standby mode.
				For details, see table 6.2.
6	STS2	0	R/W	Standby Timer Select 2 to 0
5	STS1	0	R/W	These bits designate the time the CPU and perip
4	STS0	0	R/W	modules wait for stable clock operation after exit standby mode, to active mode or sleep mode du interrupt. The designation should be made accor the clock frequency so that the waiting time is at ms. The relationship between the specified value number of wait states is shown in table 6.1. Whe external clock is to be used, the minimum value STS1 = STS0 =1) is recommended.
3 to 0	—	All 0	—	Reserved These bits are always read as 0.

Table 6.1 Operating Frequency and Waiting Time

Bit Name				Operating Frequency						
STS2	STS1	STS0	Waiting Time	20 MHz	16 MHz	10 MHz	8 MHz	4 MHz	2 MHz	1 MHz
0	0	0	8,192 states	0.4	0.5	0.8	1.0	2.0	4.1	8.1
		1	16,384 states	0.8	1.0	1.6	2.0	4.1	8.2	16.4
	1	0	32,768 states	1.6	2.0	3.3	4.1	8.2	16.4	32.8
		1	65,536 states	3.3	4.1	6.6	8.2	16.4	32.8	65.5
1	0	0	131,072 states	6.6	8.2	13.1	16.4	32.8	65.5	131.1
		1	1,024 states	0.05	0.06	0.10	0.13	0.26	0.51	1.02
	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.

				This bit is always read as 0.
5	DTON	0	R/W	Direct Transfer on Flag This bit selects the mode to transit after the exe a SLEEP instruction, as well as bit SSBY of SY 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 and sleep modes. The operating clock frequenc changes to the set frequency after the SLEEP i is executed. 0XX: ϕ_{OSC} 100: $\phi_{OSC}/8$ 101: $\phi_{OSC}/16$ 110: $\phi_{OSC}/32$ 111: $\phi_{OSC}/64$
1, 0	—	All 0	—	Reserved These bits are always read as 0.

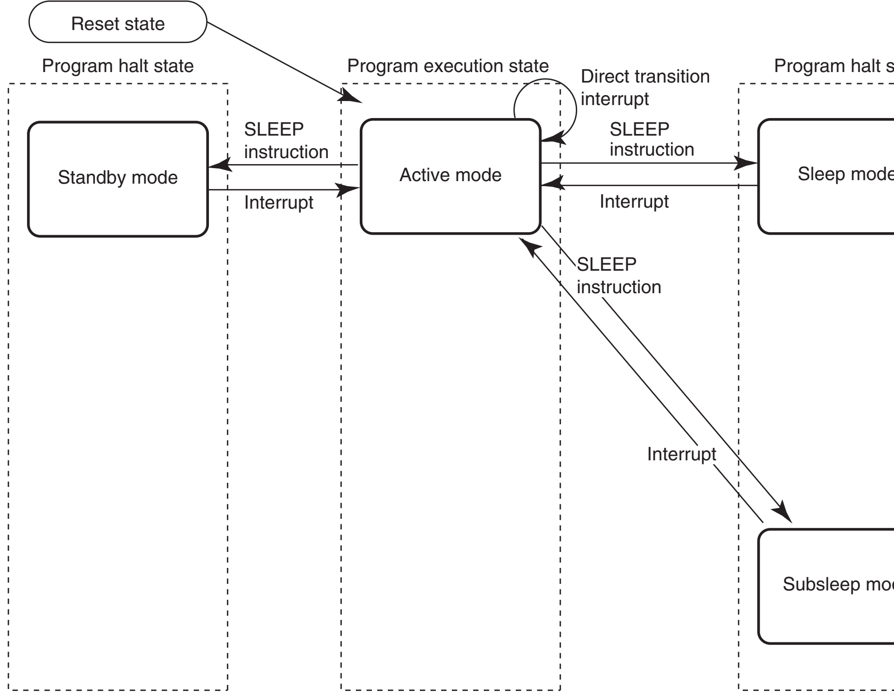
Legend: X : Don't care.

Bit	Bit Name	Initial Value	R/W	Description
4	MSTAD	0	R/W	A/D Converter Module Standby A/D converter enters standby mode when this bi 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 th watchdog timer clock, the watchdog timer operat regardless of the setting of this bit.
2	MSTTW	0	R/W	Timer W Module Standby Timer W enters standby mode when this bit is se
1	MSTTV	0	R/W	Timer V Module Standby Timer V enters standby mode when this bit is se
0	—	0	—	Reserved This bit is always read as 0.

6.1.4 Module Standby Control Register 2 (MSTCR2)

MSTCR2 allows the on-chip peripheral modules to enter a standby state in module units.

Bit	Bit Name	Initial Value	R/W	Description
7	MSTS3_2	0	R/W	SCI3_2 Module Standby SCI3_2 enters standby mode when this bit is set
6 to 0	—	All 0	—	Reserved These bits are always read as 0.



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

Figure 6.1 Mode Transition Diagram

initial values. To use these functions after entering active mode, reset the regis

Table 6.3 Internal State in Each Operating Mode

Function		Active Mode	Sleep Mode	Subsleep Mode	Standby Mode
System clock oscillator		Functioning	Functioning	Halted	Halted
CPU operations	Instructions	Functioning	Halted	Halted	Halted
	Registers	Functioning	Retained	Retained	Retained
RAM		Functioning	Retained	Retained	Retained
IO ports		Functioning	Retained	Retained	Register conten retained, but out high-impedance
External interrupts	IRQ3, IRQ0	Functioning	Functioning	Functioning	Functioning
	WKP5 to WKP0	Functioning	Functioning	Functioning	Functioning
Peripheral functions	Timer V	Functioning	Functioning	Reset	Reset
	Timer W	Functioning	Functioning	Retained	Retained (if inte ϕ is selected as clock, the counte incremented by subclock)
	Watchdog timer	Functioning	Functioning	Retained	Retained (functi internal oscillato selected as a co
	SCI3	Functioning	Functioning	Reset	Reset
	A/D converter	Functioning	Functioning	Reset	Reset

6.2.2 Standby Mode

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

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

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

6.2.3 Subsleep Mode

In subsleep mode, the system clock oscillator is halted, and operation of the CPU and on
peripheral modules is halted. As long as a required voltage is applied, the contents of CP
registers, the on-chip RAM, and some registers of the on-chip peripheral modules are re
ports keep the same states as before the transition.

Subsleep mode is cleared by an interrupt. When an interrupt is requested, the system clo
oscillator starts to oscillate. Subsleep mode is cleared and an interrupt exception handlin
when the time set in bits STS2 to STS0 in SYSCR1 elapses. Subsleep mode is not cleare
bit of CCR is 1 or the interrupt is disabled in the interrupt enable bit.

executing a SLEEP instruction while the DTON bit in SYSCR2 is set to 1. The direct tra
also enables operating frequency modification in active mode. After the mode transition,
transition interrupt exception handling starts.

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

6.5 Module Standby Function

The module-standby function can be set to any peripheral module. In module standby mo
clock supply to modules stops to enter the power-down mode. Module standby mode ena
on-chip peripheral module to enter the standby state by setting a bit that corresponds to ea
module in MSTCR1 and MSTCR2 to 1 and cancels the mode by clearing the bit to 0.

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

- On-board programming

— On-board programming/erasing can be done in boot mode, in which the boot pro into the chip is started to erase or program of the entire flash memory. In normal program mode, individual blocks can be erased or programmed.

- 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.

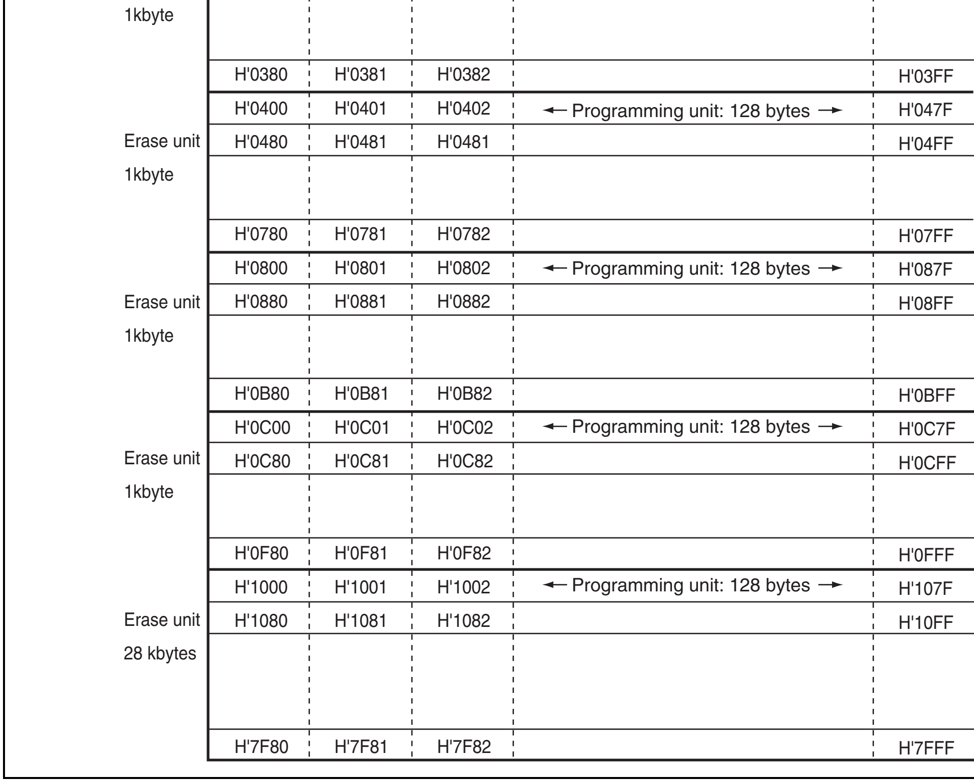


Figure 7.1 Flash Memory Block Configuration

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 7
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 state is cancelled. Set this bit to 1 before setting 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, the setup state is cancelled. Set this bit to 1 before 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, eras mode is cancelled.

When this bit is set to 1, and while the SWE = 1
= 1 bits are 1, the flash memory changes to prog
mode. When it is 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). Whe is set to 1, flash memory goes to the error-protec state. See 7.5.3, Error Protection, for details.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

4	EB4	0	R/W	When this bit is set to 1, 28 kbytes of H'1000 to will be erased.
3	EB3	0	R/W	When this bit is set to 1, 1 kbyte of H'0C00 to H be erased.
2	EB2	0	R/W	When this bit is set to 1, 1 kbyte of H'0800 to H be erased.
1	EB1	0	R/W	When this bit is set to 1, 1 kbyte of H'0400 to H be erased.
0	EB0	0	R/W	When this bit is set to 1, 1 kbyte of H'0000 to H be erased.

7.2.4 Flash Memory Enable Register (FENR)

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

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

This can be used for programming initial values in the on-board state or for a forcible retu
programming/erasing can no longer be done in user program mode. In user program mod
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	$\overline{NMI}$	E10T_0	PB0	PB1	PB2	LSI State after Reset End
0	1	X	X	X	X	User Mode
0	0	1	X	X	X	Boot Mode

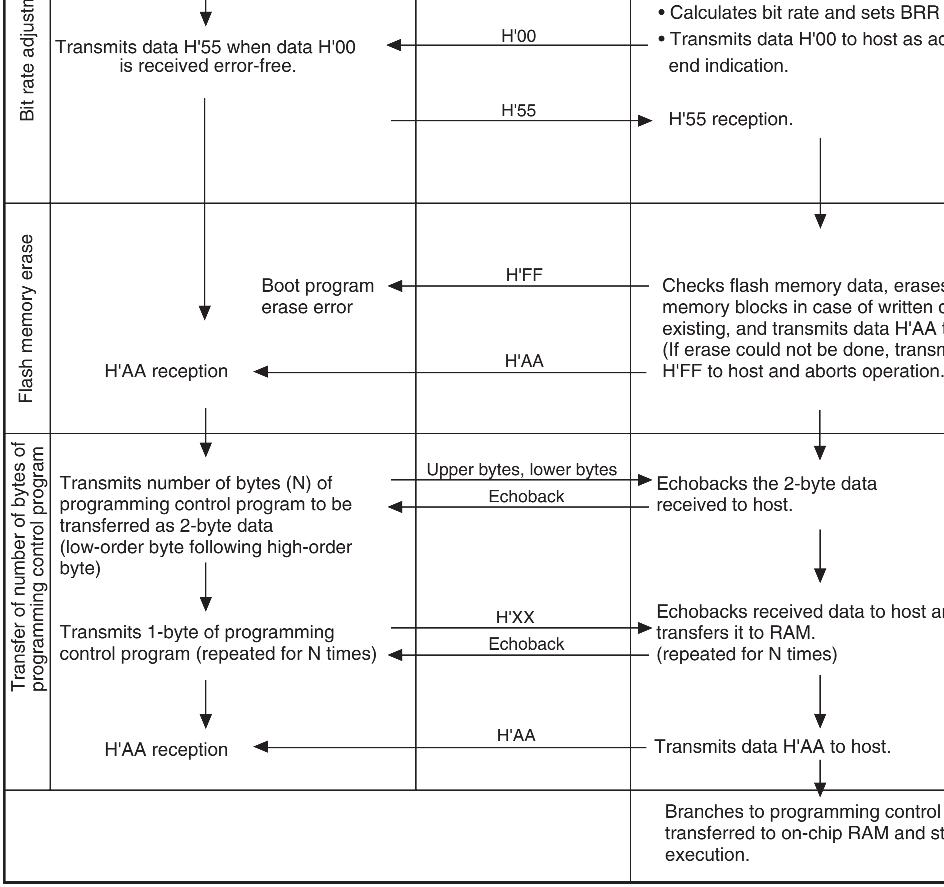
Legend: X: Don't care.

7.3.1 Boot Mode

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

1. When boot mode is used, the flash memory programming control program must be pr
   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 da
   bit, and no parity.
3. When the boot program is initiated, the chip measures the low-level period of asynchr
   SCI communication data (H'00) transmitted continuously from the host. The chip ther
   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 shou
   pulled up on the board if necessary. After the reset is complete, it takes approximately
   states before the chip is ready to measure the low-level period.

6. Before branching to the programming control program, the chip terminates transfer o
by SCI3 (by clearing the RE and TE bits in SCR3 to 0), however the adjusted bit rate
remains set in BRR. Therefore, the programming control program can still use it for
write data or verify data with the host. The TxD pin is high (PCR22 = 1, P22 = 1). T
contents of the CPU general registers are undefined immediately after branching to t
programming control program. These registers must be initialized at the beginning o
programming control program, as the stack pointer (SP), in particular, is used implic
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 $\overline{\text{NMI}}$ pin input levels in boot mode.


Bit rate adjustm
Transmits data H'55 when data H'00 is received error-free.
H'00
• Calculates bit rate and sets BRR
• Transmits data H'00 to host as ad end indication.
H'55
H'55 reception.
Flash memory erase
Boot program erase error
H'FF
Checks flash memory data, erases memory blocks in case of written d existing, and transmits data H'AA t (If erase could not be done, transm H'FF to host and aborts operation.
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 an transfers it to RAM. (repeated for N times)
H'AA reception
H'AA
Transmits data H'AA to host.
Branches to programming control transferred to on-chip RAM and st execution.

7.3.2 Programming/Erasing in User Program Mode

On-board programming/erasing of an individual flash memory block can also be perforr
program mode by branching to a user program/erase control program. The user must set
conditions and provide on-board means of supplying programming data. The flash mem
contain the user program/erase control program or a program that provides the user prog
control program from external memory. As the flash memory itself cannot be read durin
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 m
Prepare a user program/erase control program in accordance with the description in sect
Flash Memory Programming/Erasing.

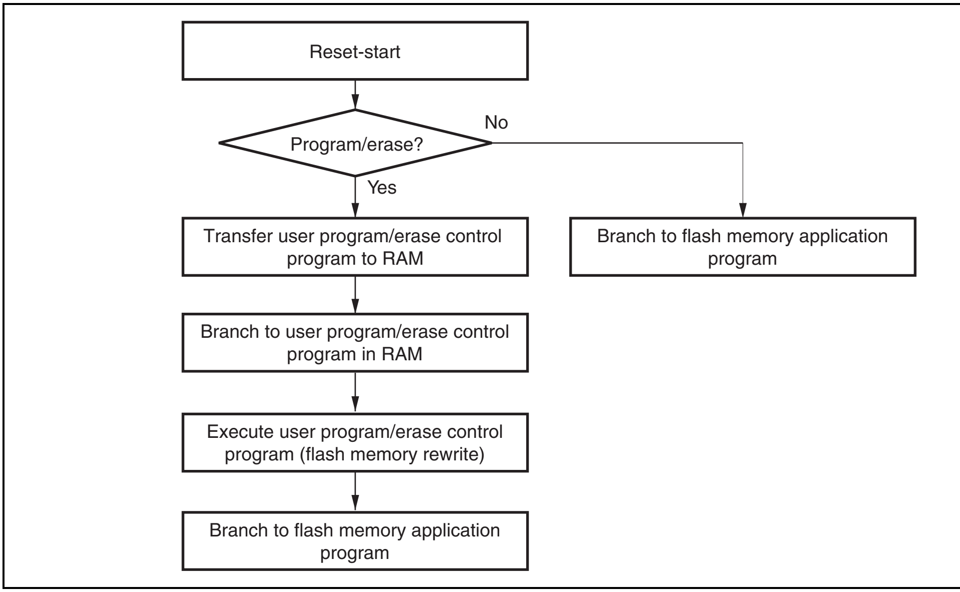
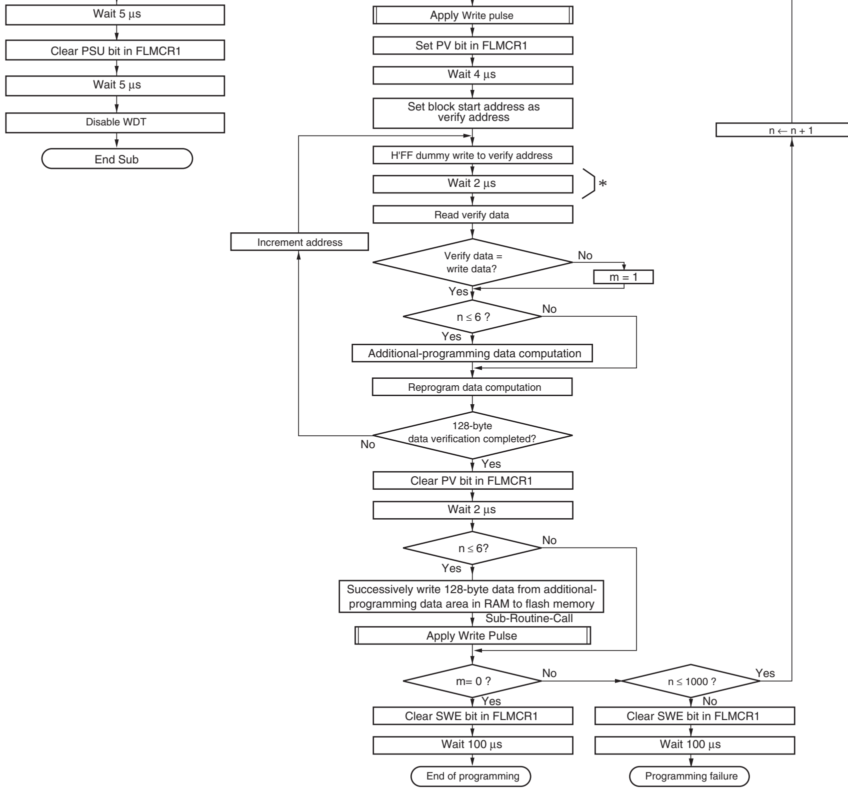


Figure 7.2 Programming/Erasing Flowchart Example in User Program Mo

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 fl 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 a 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 bits are B'00. Verify data can be read in words or in longwords from the address to w dummy write was performed.



Note: *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
0	1	1	No additional progra
1	0	1	No additional progra
1	1	1	No additional progra

Table 7.6 Programming Time

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

Note: Time shown in μs.

7.4.2 Erase/Erase-Verify

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

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

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

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

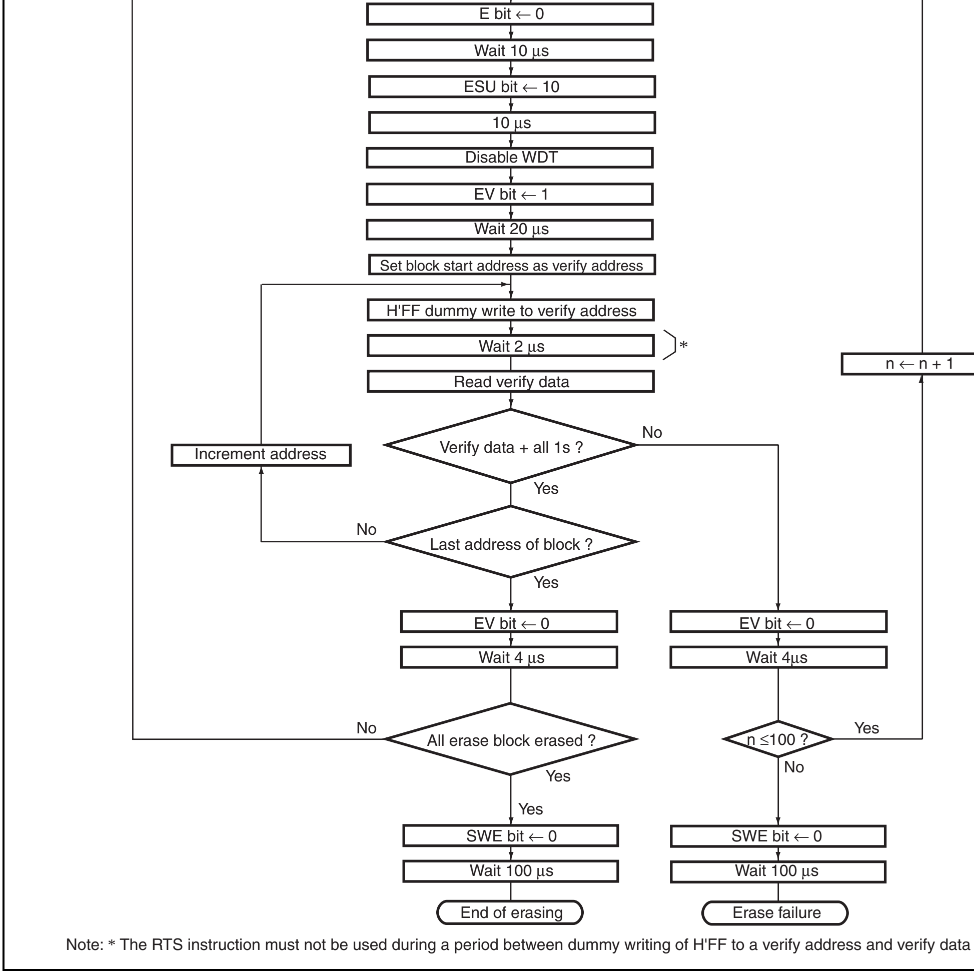


Figure 7.4 Erase/Erase-Verify Flowchart

unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case o
during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC 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 I
in FLMCR1 does not cause a transition to program mode or erase mode. By setting the e
register 1 (EBR1), erase protection can be set for individual blocks. When EBR1 is set t
erase protection is set for all blocks.

7.5.3 Error Protection

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

When the following errors are detected during programming/erasing of flash memory, 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/era (including vector read and instruction fetch)
- Immediately after exception handling excluding a reset during programming/erasing
- When a SLEEP instruction is executed during programming/erasing

H8/36022, H8/36012	512 bytes	H'FD80 to H'FF7F
H8/36011	512 bytes	H'FD80 to H'FF7F
H8/36010	512 bytes	H'FD80 to H'FF7F

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

functions in each port, see Appendix B.1, I/O Port Block Diagrams. For the execution o
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, timer V input pi
SCI3 I/O pin. Figure 9.1 shows its pin configuration.

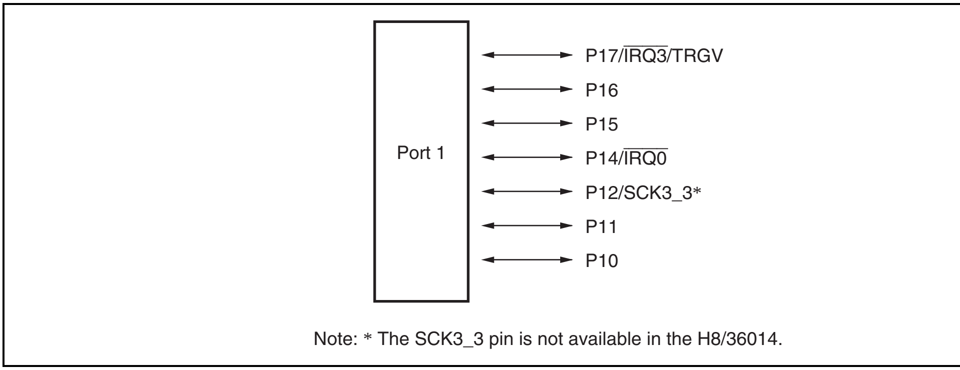


Figure 9.1 Port 1 Pin Configuration

Port 1 has the following registers.

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

				1: $\overline{\text{IRQ3}}$/TRGV input pin
6, 5	—	All 0	—	Reserved These bits are always read as 0.
4	IRQ0	0	R/W	P14/$\overline{\text{IRQ0}}$ Pin Function Switch This bit selects whether pin P14/$\overline{\text{IRQ0}}$ is used as as $\overline{\text{IRQ0}}$. 0: General I/O port 1: $\overline{\text{IRQ0}}$ input pin
3	TXD2	0	R/W	P72/TXD_2 Pin Function Switch This bit selects whether pin P72/TXD_2 is used a as TXD_2. 0: General I/O port 1: TXD_2 output pin
2	—	0	R/W	Reserved This bit must always be cleared to 0 (setting to 1 disabled).
1	TXD	0	R/W	P22/TXD Pin Function Switch This bit selects whether pin P22/TXD is used as as TXD. 0: General I/O port 1: TXD output pin
0	—	0	—	Reserved This bit is always read as 0.

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.

- P16 pin

Register	PCR1	
Bit Name	**PCR16**	**Pin Function**
Setting value	0	P16 input pin
	1	P16 output pin

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/SCK3_3* pin

Register	SCR3_3*		SMR_3*	PCR1	
Bit Name	**CKE1**	**CKE0**	**COM**	**PCR12**	**Pin Function**
Setting value	0	0	0	0	P12 input pin
				1	P12 output pin
			1	X	SCK3_3 output pin*
	0	1	X	X	SCK3_3 output pin*
	1	X	X	X	SCK3_3 input pin*

Legend X: Don't care.

Note: * Not available in the H8/36014.

- P11 pin

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

figure 9.2. The register settings of PMR1 and SCI3 have priority for functions of the pins
uses.

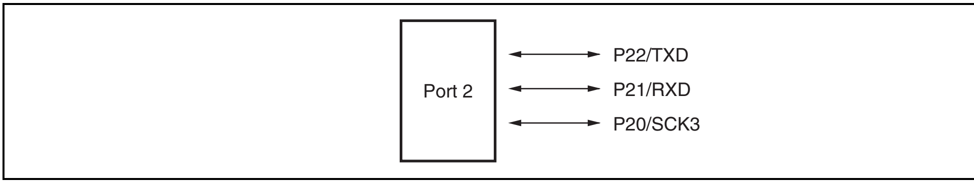


Figure 9.2 Port 2 Pin Configuration

Port 2 has the following registers.

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

9.2.1 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 3	—	—	—	Reserved
2	PCR22	0	W	When each of the port 2 pins P22 to P20 functio general I/O port, setting a PCR2 bit to 1 makes t corresponding pin an output port, while clearing 0 makes the pin an input port.
1	PCR21	0	W	
0	PCR20	0	W	

0	P20	0	R/W	stored in PDR2 is read. If PDR2 is read while P… are cleared to 0, the pin states are read regardl… value stored in PDR2.

9.2.3 Pin Functions

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

- 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.

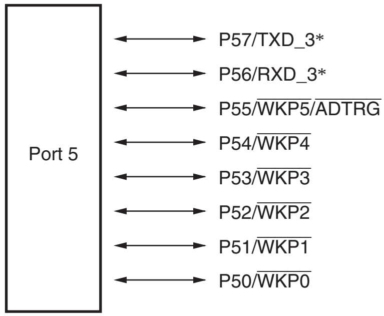
- P21/RXD pin

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.

9.3 Port 5

Port 5 is a general I/O port also functioning as an SCI3 I/O pins, A/D trigger input pin, ar
wakeup interrupt input pins. Each pin of the port 5 is shown in figure 9.3.



Note: * The TXD_3 and RXD_3 pins are not available in the H8/36014.

Figure 9.3 Port 5 Pin Configuration

Port 5 has the following registers.

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

				0: General I/O port 1: NMOS open-drain output
5	WKP5	0	R/W	P55/$\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ Pin Function Switch Selects whether pin P55/$\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ is used as $\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ input. 0: General I/O port 1: $\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ input pin
4	WKP4	0	R/W	P54/$\overline{\text{WKP4}}$ Pin Function Switch Selects whether pin P54/$\overline{\text{WKP4}}$ is used as P54 or a 0: General I/O port 1: $\overline{\text{WKP4}}$ input pin
3	WKP3	0	R/W	P53/$\overline{\text{WKP3}}$ Pin Function Switch Selects whether pin P53/$\overline{\text{WKP3}}$ is used as P53 or a 0: General I/O port 1: $\overline{\text{WKP3}}$ input pin
2	WKP2	0	R/W	P52/$\overline{\text{WKP2}}$ Pin Function Switch Selects whether pin P52/$\overline{\text{WKP2}}$ is used as P52 or a 0: General I/O port 1: $\overline{\text{WKP2}}$ input pin
1	WKP1	0	R/W	P51/$\overline{\text{WKP1}}$ Pin Function Switch Selects whether pin P51/$\overline{\text{WKP1}}$ is used as P51 or a 0: General I/O port 1: $\overline{\text{WKP1}}$ input pin

PCR5 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 5.

Bit	Bit Name	Initial Value	R/W	Description
7	PCR57	0	W	When each of the port 5 pins P57 to P50 functions as general I/O port, setting a PCR5 bit to 1 makes the corresponding pin an output port, while clearing the b makes the pin an input port.
6	PCR56	0	W	
5	PCR55	0	W	
4	PCR54	0	W	
3	PCR53	0	W	
2	PCR52	0	W	
1	PCR51	0	W	
0	PCR50	0	W	

3	P53	0	R/W	
2	P52	0	R/W	
1	P51	0	R/W	
0	P50	0	R/W	

9.3.4 Port Pull-Up Control Register 5 (PUCR5)

PUCR5 controls the pull-up MOS in bit units of the pins set as the input ports.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0.
5	PUCR55	0	R/W	Only bits for which PCR5 is cleared are valid. T MOS of the corresponding pins enter the on-sta these bits are set to 1, while they enter the off-s these bits are cleared to 0.
4	PUCR54	0	R/W	
3	PUCR53	0	R/W	
2	PUCR52	0	R/W	
1	PUCR51	0	R/W	
0	PUCR50	0	R/W	

	1	X	TXD_3 output pin*

Legend X: Don't care.

Note: * Not available in the H8/36014.

- P56/RXD_3* pin

Register	SCR3_3*	PCR5	
Bit Name	**RE**	**PCR56**	**Pin Function**
Setting Value	0	0	P56 input pin
		1	P56 output pin
	1	X	RXD_3 input pin*

Legend X: Don't care.

Note: * Not available in the H8/36014.

- 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.

Register	PMR5	PCR5	
Bit Name	**WKP3**	**PCR53**	**Pin Function**
Setting Value	0	0	P53 input pin
		1	P53 output pin
	1	X	$\overline{WKP3}$ input pin

Legend X: Don't care.

- P52/$\overline{WKP2}$ pin

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

Legend X: Don't care.

- P51/$\overline{WKP1}$ pin

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

Legend X: Don't care.

9.4 Port 7

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

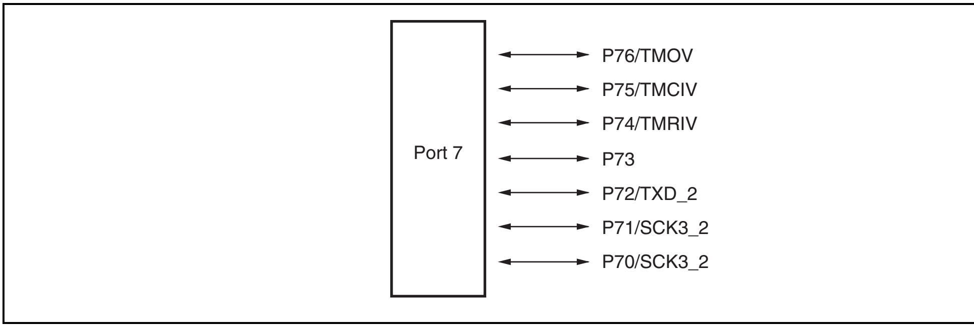


Figure 9.4 Port 7 Pin Configuration

Port 7 has the following registers.

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

3	PCR73	0	W	P76/TMOV pin.
2	PCR72	0	W	
1	PCR71	0	W	
0	PCR70	0	W	

9.4.2 Port Data Register 7 (PDR7)

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1.
6	P76	0	R/W	PDR7 stores output data for port 7 pins.
5	P75	0	R/W	If PDR7 is read while PCR7 bits are set to 1, th stored in PDR7 is read. If PDR7 is read while P are cleared to 0, the pin states are read regardl value stored in PDR7.
4	P74	0	R/W	
3	P73	0	R/W	
2	P72	0	R/W	
1	P71	0	R/W	
0	P70	0	R/W	

	Other than the above values	X	TMOV output pin

Legend X: Don't care.

- P75/TMCIV pin

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

- P74/TMRIV pin

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

- P73 pin

Register	PCR7	
Bit Name	**PCR73**	**Pin Function**
Setting Value	0	P73 input pin
	1	P73 output 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		SMR_2	PCR7	
Bit Name	CKE1	CKE0	COM	PCR70	Pin Function
Setting Value	0	0	0	0	P70 input pin
				1	P70 output pin
			1	X	SCK3_2 outpu
	0	1	X	X	SCK3_2 outpu
	1	X	X	X	SCK3_2 input p

Legend X: Don't care.

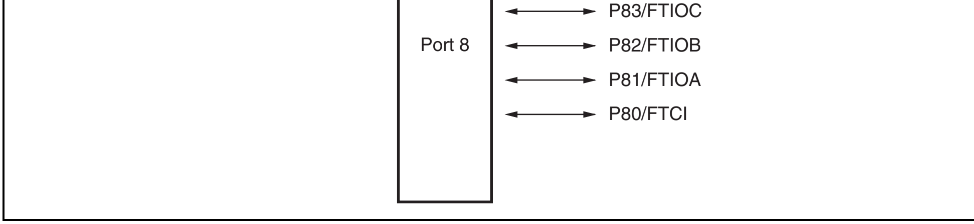


Figure 9.5 Port 8 Pin Configuration

Port 8 has the following registers.

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

9.5.1 Port Control Register 8 (PCR8)

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	—	—	Reserved
4	PCR84	0	W	When each of the port 8 pins P84 to P80 functio general I/O port, setting a PCR8 bit to 1 makes t corresponding pin an output port, while clearing 0 makes the pin an input port.
3	PCR83	0	W	
2	PCR82	0	W	
1	PCR81	0	W	
0	PCR80	0	W	

1	P81	0	R/W	value stored in PDR8.
0	P80	0	R/W	

9.5.3 Pin Functions

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

- P84/FTIOD pin

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

Legend X: Don't care.

Legend X: Don't care.

- P82/FTIOB pin

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

Legend X: Don't care.

- P81/FTIOA pin

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

Legend X: Don't care.

B is shown in figure 9.6.

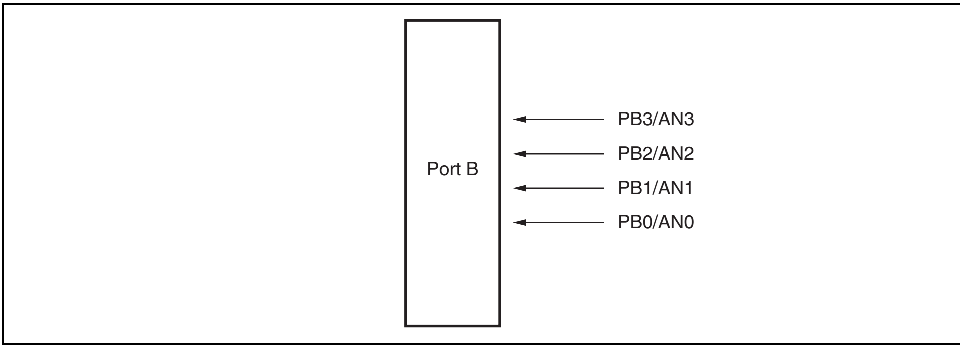


Figure 9.6 Port B Pin Configuration

Port B has the following register.

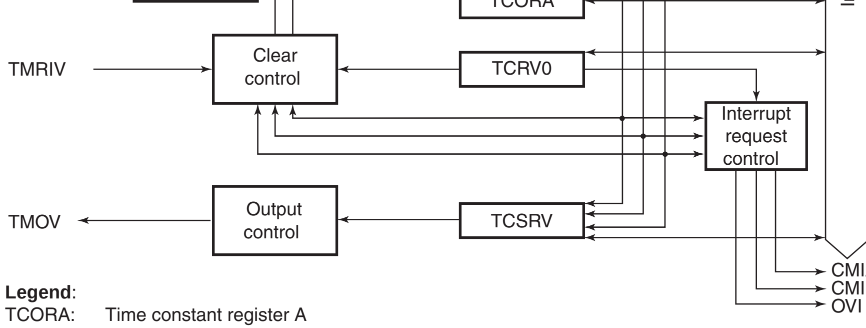
- Port data register B (PDRB)

0	PB0	—	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 10.1 Block Diagram of Timer V

10.3 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)

10.3.1 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.

and the settings of bits OS3 to OS0 in TCSRV.

TCORA and TCORB are initialized to H'FF.

6	CMIEA	0	R/W	Compare Match Interrupt Enable A When this bit is set to 1, interrupt request from t bit in TCSRV is enabled.
5	OVIE	0	R/W	Timer Overflow Interrupt Enable When this bit is set to 1, interrupt request from t 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 TC 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 pir operation of TCNTV after clearing depends 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 TCNT counting condition in combination with ICKS0 ir
0	CKS0	0	R/W	Refer to table 10.2.

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

				After reading CMFB = 1, cleared by writing 0 to
6	CMFA	0	R/W	Compare Match Flag A Setting condition: When the TCNTV value matches the TCORA v Clearing condition: After reading CMFA = 1, cleared by writing 0 to
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 C
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 TMC the compare match of TCORB and TCNTV. 00: No change 01: 0 output 10: 1 output 11: Output toggles

OS3 and OS2 select the output level for compare match B. OS1 and OS0 select the outpu
for compare match A. The two output levels can be controlled independently. After a rese
timer output is 0 until the first compare match.

3	TVEG0	0	R/W	These bits select the TRGV input edge. 00: TRGV trigger input is prohibited 01: Rising edge is selected 10: Falling edge is selected 11: Rising and falling edges are both selected
2	TRGE	0	R/W	TCNT starts counting up by the input of the edg selected by TVEG1 and TVEG0. 0: Disables starting counting-up TCNTV by the the TRGV pin and halting counting-up TCNT TCNTV is cleared by a compare match. 1: Enables starting counting-up TCNTV by the i the TRGV pin and halting counting-up TCNT TCNTV is cleared by a compare match.
1	—	1	—	Reserved This bit is always read as 1.
0	ICKS0	0	R/W	Internal Clock Select 0 This bit selects clock signals to input to TCNTV combination with CKS2 to CKS0 in TCRV0. Refer to table 10.2.

will be set. The timing at this time is shown in figure 10.4. 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 10.6 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 10.7 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 10.8 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.

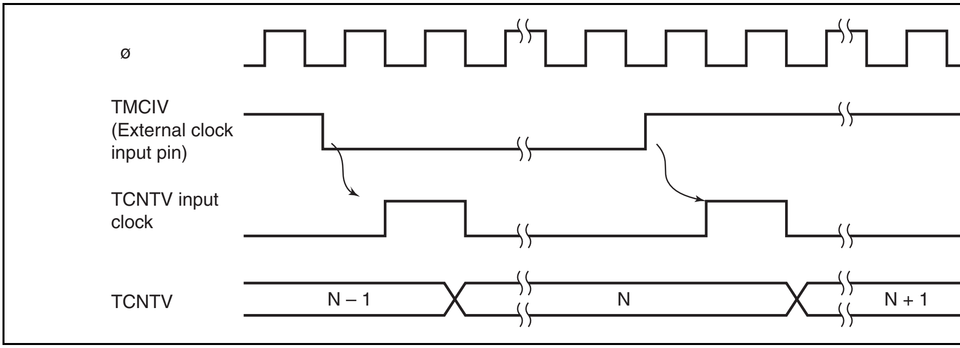


Figure 10.3 Increment Timing with External Clock

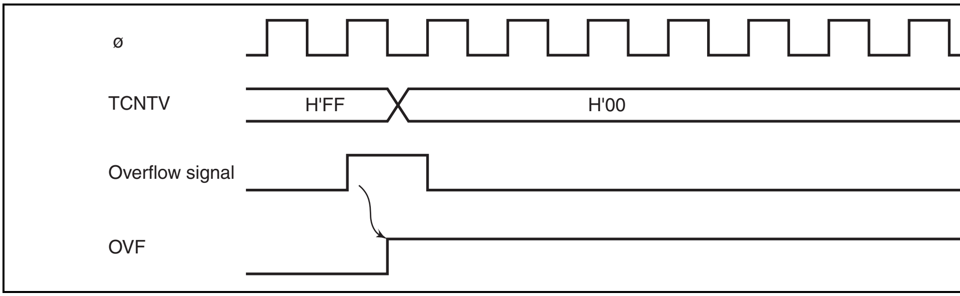


Figure 10.4 OVF Set Timing

Figure 10.5   CMFA and CMFB Set Timing

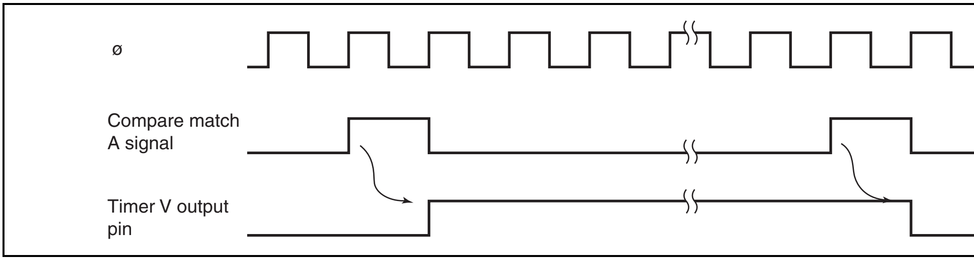


Figure 10.6   TMOV Output Timing

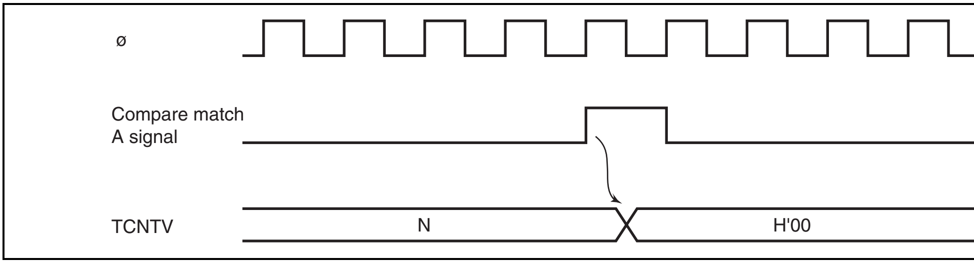


Figure 10.7   Clear Timing by Compare Match

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.

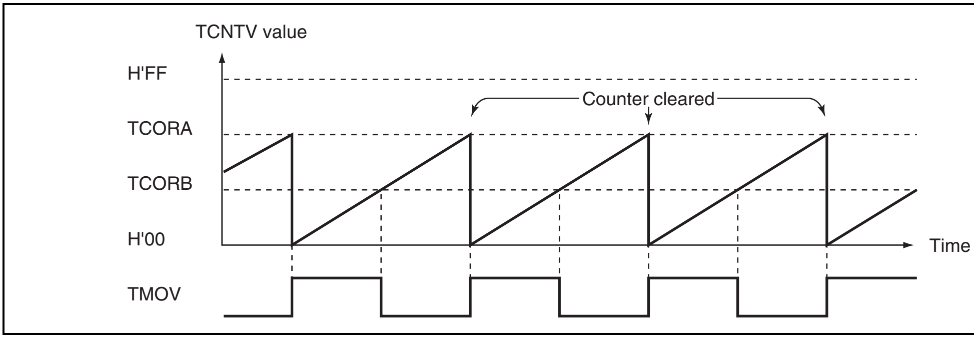


Figure 10.9 Pulse Output Example

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).

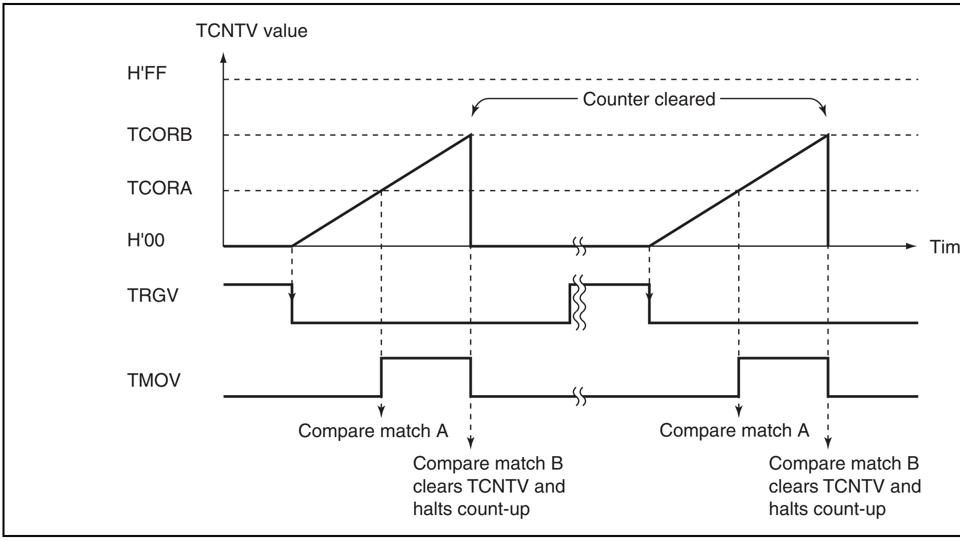


Figure 10.10 Example of Pulse Output Synchronized to TRGV Input

10.12 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 10.3 the switch is from a high clock signal to a low clock signal, the switch
seen as a falling edge, causing TCNTV to increment. TCNTV can also be increment
switch between internal and external clocks.

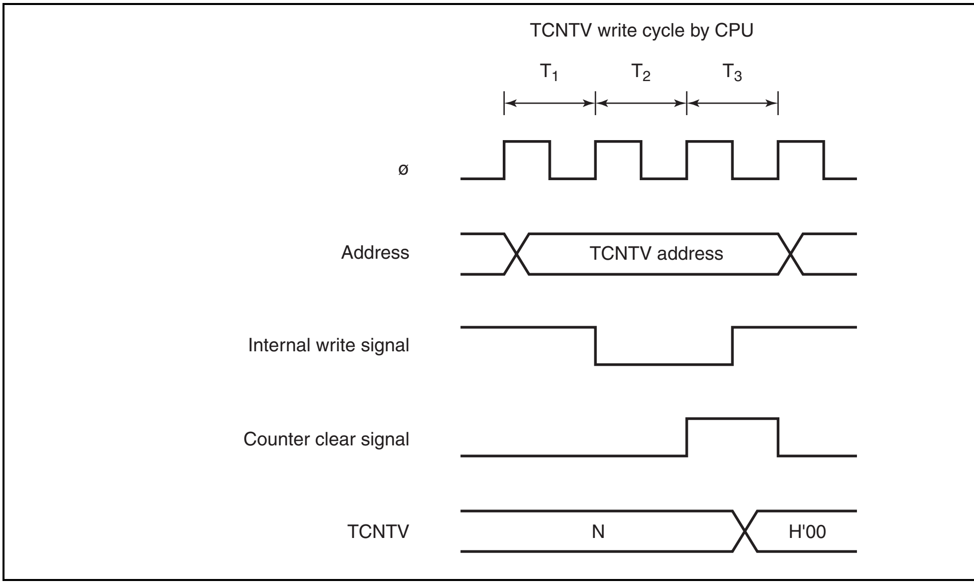


Figure 10.11 Contention between TCNTV Write and Clear

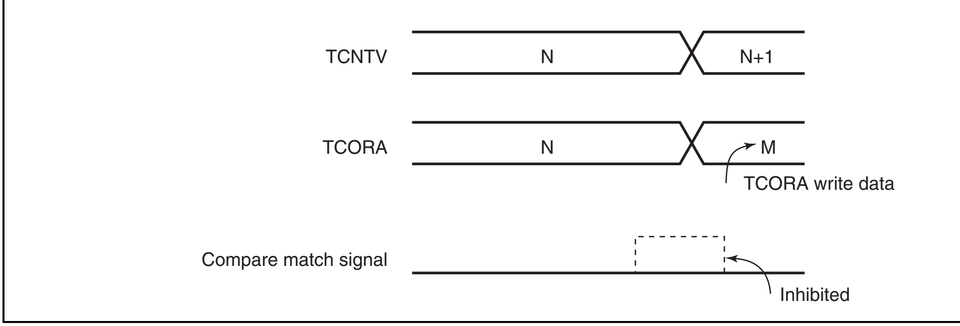


Figure 10.12 Contention between TCORA Write and Compare Match

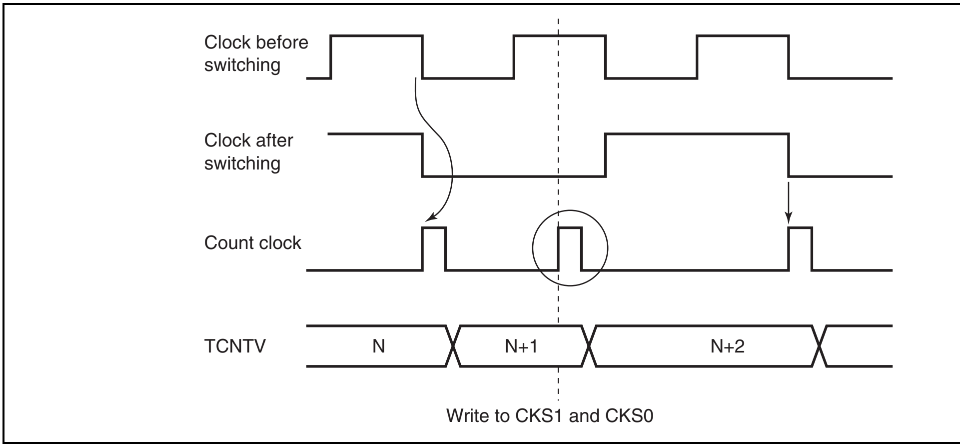
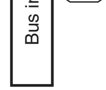


Figure 10.13 Internal Clock Switching and TCNTV Operation

- Capability to process up to four pulse outputs or four pulse inputs
- Four general registers:
  - Independently assignable output compare or input capture functions
  - Usable as two pairs of registers; one register of each pair operates as a buffer for
    compare or input capture register
- Four selectable operating modes :
  - Waveform output by compare match
    Selection of 0 output, 1 output, or toggle output
  - Input capture function
    Rising edge, falling edge, or both edges
  - Counter clearing function
    Counters can be cleared by compare match
  - PWM mode
    Up to three-phase PWM output can be provided with desired duty ratio.
- Any initial timer output value can be set
- Five interrupt sources
  Four compare match/input capture interrupts and an overflow interrupt.

Table 11.1 summarizes the timer W functions, and figure 11.1 shows a block diagram o
W.

		compare match	compare match			
Initial output value setting function		—	Yes	Yes	Yes	Yes
Buffer function		—	Yes	Yes	—	—
Compare match output	0	—	Yes	Yes	Yes	Yes
	1	—	Yes	Yes	Yes	Yes
	Toggle	—	Yes	Yes	Yes	Yes
Input capture function		—	Yes	Yes	Yes	Yes
PWM mode		—	—	Yes	Yes	Yes
Interrupt sources		Overflow	Compare match/input capture	Compare match/input capture	Compare match/input capture	Con mat capt



Legend:

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

Figure 11.1 Timer W Block Diagram

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

11.3 Register Descriptions

The timer W has the following registers.

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

This bit is always read as 1.

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

5	CKS1	0	R/W	Select the TCNT clock source.
4	CKS0	0	R/W	000: Internal clock: counts on φ 001: Internal clock: counts on φ/2 010: Internal clock: counts on φ/4 011: Internal clock: counts on φ/8 1XX: Counts on rising edges of the external ever When the internal clock source (φ) is selected, su sources are counted in subactive and subsleep r
3	TOD	0	R/W	Timer Output Level Setting D Sets the output value of the FTIOD pin until the f compare match D is generated. 0: Output value is 0* 1: Output value is 1*
2	TOC	0	R/W	Timer Output Level Setting C Sets the output value of the FTIOC pin until the f compare match C is generated. 0: Output value is 0* 1: Output value is 1*
1	TOB	0	R/W	Timer Output Level Setting B Sets the output value of the FTIOB pin until the f compare match B is generated. 0: Output value is 0* 1: Output value is 1*

11.3.3 Timer Interrupt Enable Register W (TIERW)

TIERW controls the timer W interrupt request.

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

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

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

				0: GRB functions as an output compare register 1: GRB functions as an input capture register
5 4	IOB1 IOB0	0 0	R/W R/W	I/O Control B1 and B0 When IOB2 = 0, 00: No output at compare match 01: 0 output to the FTIOB pin at GRB compare n 10: 1 output to the FTIOB pin at GRB compare n 11: Output toggles to the FTIOB pin at GRB com match When IOB2 = 1, 00: Input capture at rising edge at the FTIOB pin 01: Input capture at falling edge at the FTIOB pi 1X: Input capture at rising and falling edges of th pin
3	—	1	—	Reserved This bit is always read as 1.
2	IOA2	0	R/W	I/O Control A2 Selects the GRA function. 0: GRA functions as an output compare register 1: GRA functions as an input capture register

				00: Input capture at rising edge of the FTIOA pi
				01: Input capture at falling edge of the FTIOA p
				1X: Input capture at rising and falling edges of t pin

Legend X: Don't care.

11.3.6 Timer I/O Control Register 1 (TIOR1)

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1.
6	IOD2	0	R/W	I/O Control D2 Selects the GRD function. 0: GRD functions as an output compare registe 1: GRD functions as an input capture register

				00: Input capture at rising edge at the FTIOD pin 01: Input capture at falling edge at the FTIOD pi 1X: Input capture at rising and falling edges at th pin
3	—	1	—	Reserved This bit is always read as 1.
2	IOC2	0	R/W	I/O Control C2 Selects the GRC function. 0: GRC functions as an output compare register 1: GRC functions as an input capture register
1 0	IOC1 IOC0	0 0	R/W R/W	I/O Control C1 and C0 When IOC2 = 0, 00: No output at compare match 01: 0 output to the FTIOC pin at GRC compare r 10: 1 output to the FTIOC pin at GRC compare r 11: Output toggles to the FTIOC pin at GRC com match When IOC2 = 1, 00: Input capture to GRC at rising edge of the F 01: Input capture to GRC at falling edge of the F 1X: Input capture to GRC at rising and falling edg the FTIOC pin

Legend X: Don't care.

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

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

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

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

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

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

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

running counter. When the CTS bit in TMRW is set to 1, TCNT starts incrementing the c
When the count overflows from H'FFFF to H'0000, the OVF flag in TSRW is set to 1. If t
in TIERW is set to 1, an interrupt request is generated. Figure 11.2 shows free-running co

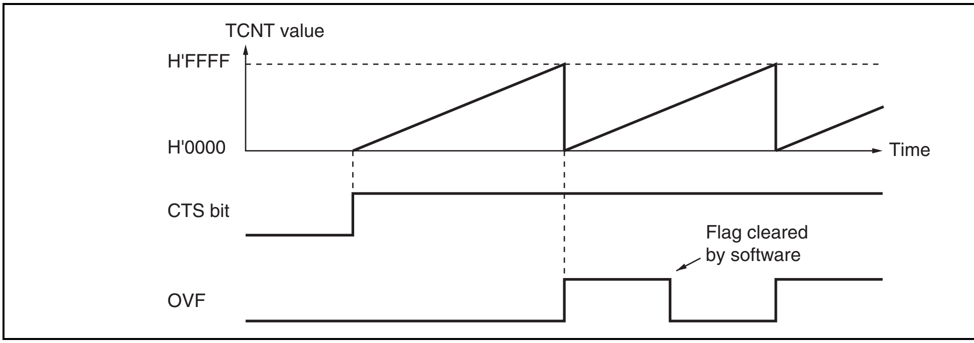


Figure 11.2   Free-Running Counter Operation

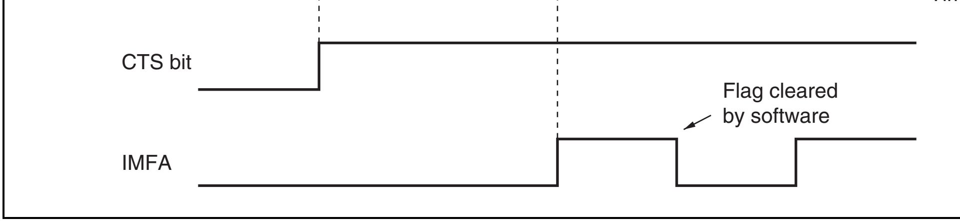


Figure 11.3 Periodic Counter Operation

By setting a general register as an output compare register, compare match A, B, C, or D the output at the FTIOA, FTIOB, FTIOC, or FTIOD pin to output 0, output 1, or toggle. 11.4 shows an example of 0 and 1 output when TCNT operates as a free-running counte is selected for compare match A, and 0 output is selected for compare match B. When si already at the selected output level, the signal level does not change at compare match.

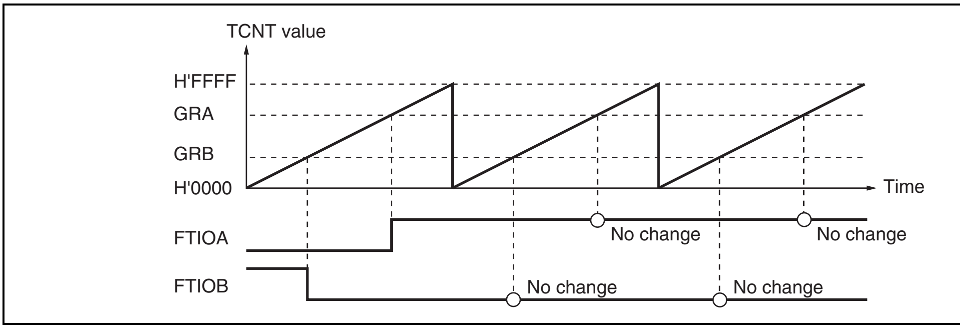


Figure 11.4 0 and 1 Output Example (TOA = 0, TOB = 1)

Figure 11.5 Toggle Output Example (TOA = 0, TOB = 1)

Figure 11.6 shows another example of toggle output when TCNT operates as a periodic c
cleared by compare match A. Toggle output is selected for both compare match A and B.

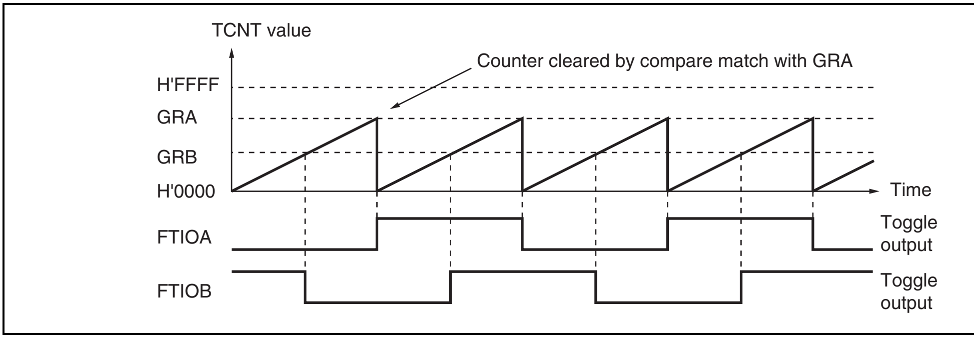


Figure 11.6 Toggle Output Example (TOA = 0, TOB = 1)

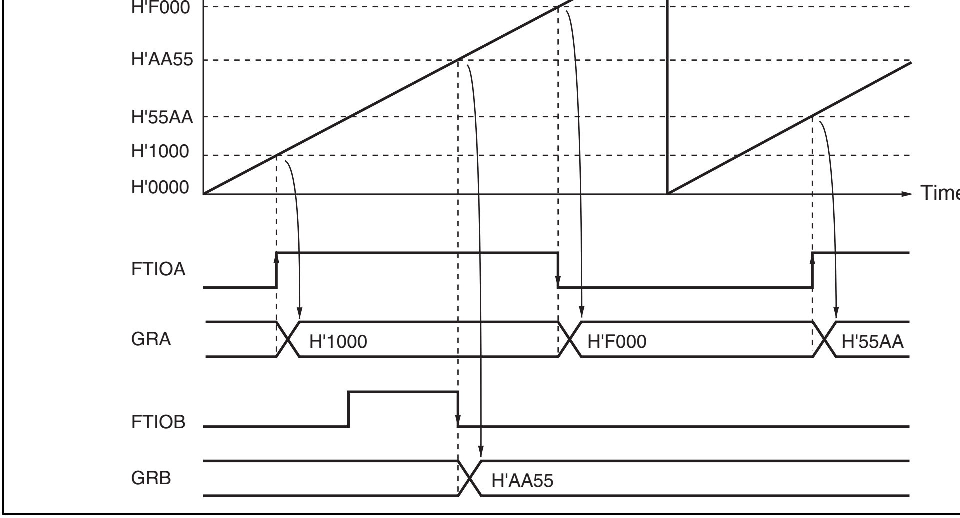


Figure 11.7 Input Capture Operating Example

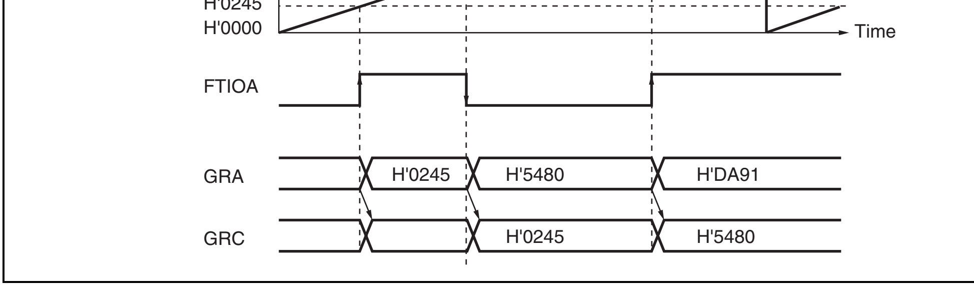


Figure 11.8 Buffer Operation Example (Input Capture)

If the same value is set in the cycle register and the duty register, the output does not ch
a compare match occurs.

Figure 11.9 shows an example of operation in PWM mode. The output signals go to 1 ar
is cleared at compare match A, and the output signals go to 0 at compare match B, C, an
TOC, and TOD = 1: initial output values are set to 1).

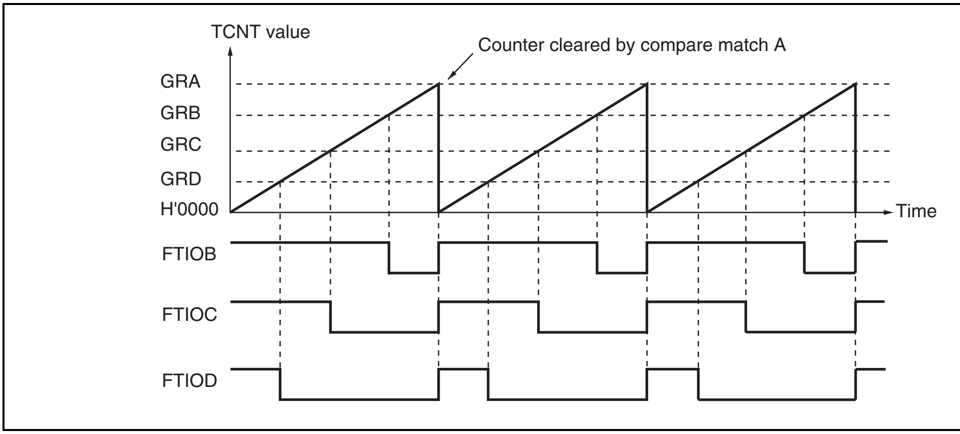


Figure 11.9 PWM Mode Example (1)

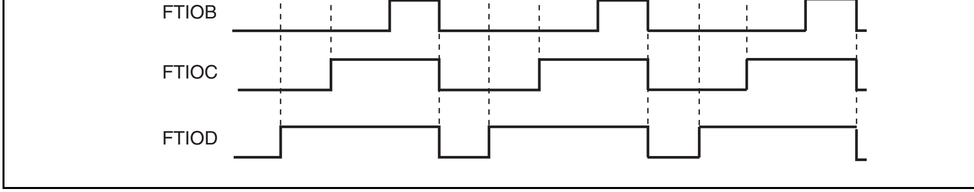


Figure 11.10 PWM Mode Example (2)

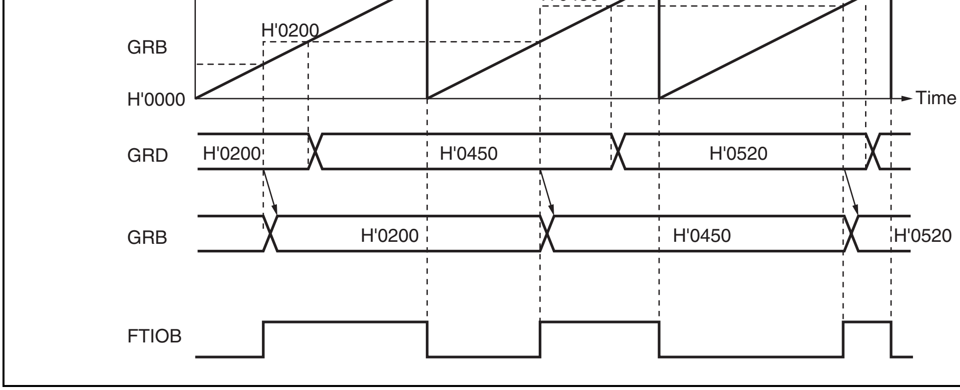


Figure 11.11 Buffer Operation Example (Output Compare)

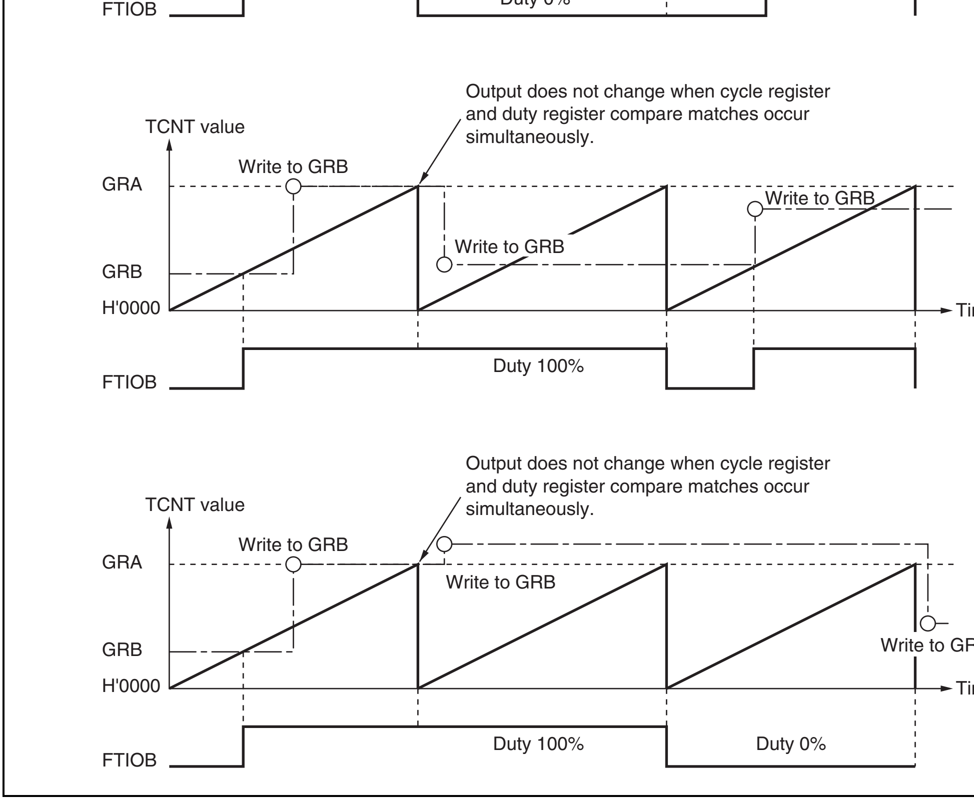


Figure 11.12 PWM Mode Example
(TOB, TOC, and TOD = 0: initial output values are set to 0)

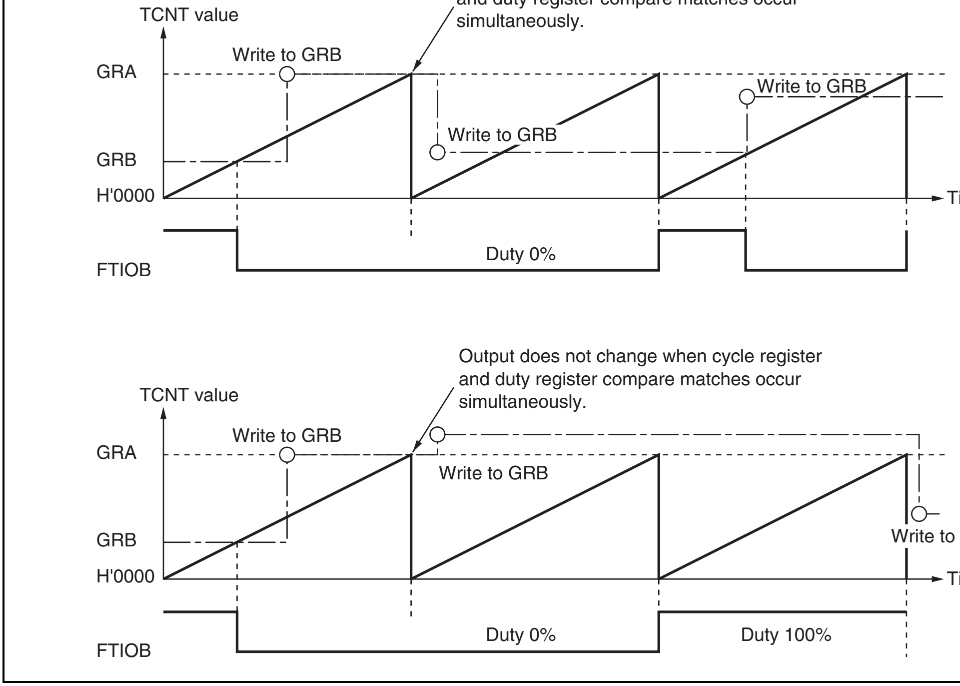


Figure 11.13 PWM Mode Example
(TOB, TOC, and TOD = 1: initial output values are set to 1)

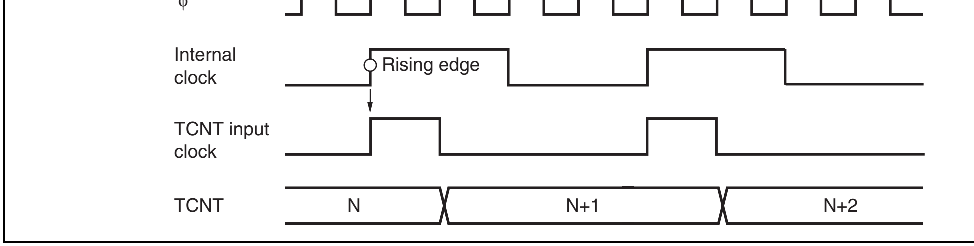


Figure 11.14 Count Timing for Internal Clock Source

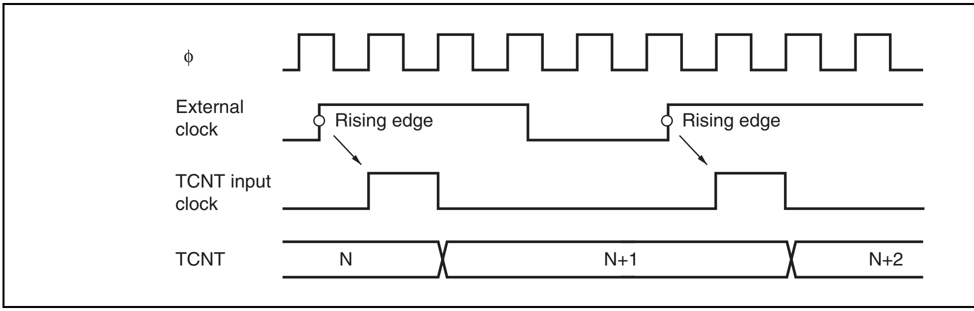


Figure 11.15 Count Timing for External Clock Source

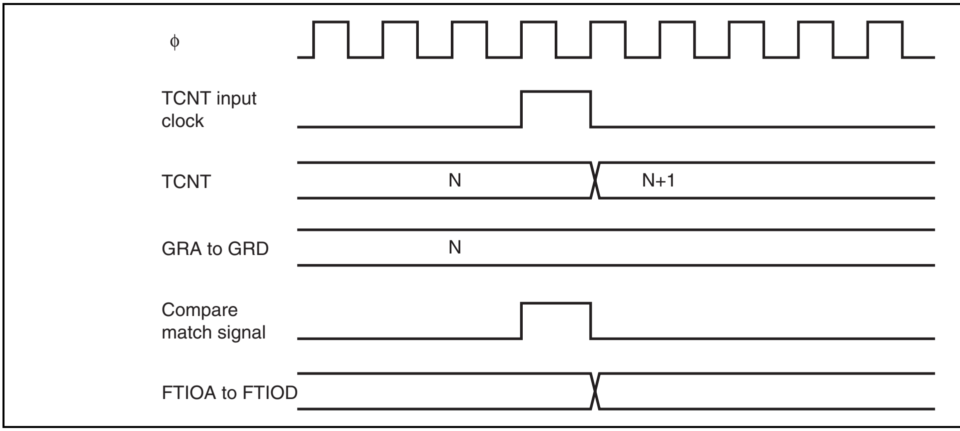


Figure 11.16 Output Compare Output Timing

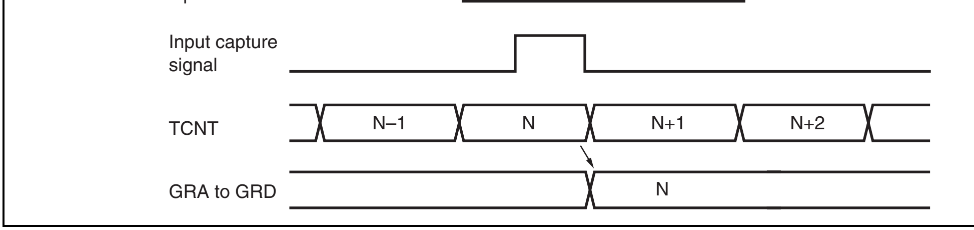


Figure 11.17 Input Capture Input Signal Timing

11.5.4 Timing of Counter Clearing by Compare Match

Figure 11.18 shows the timing when the counter is cleared by compare match A. When th
value is N, the counter counts from 0 to N, and its cycle is N + 1.

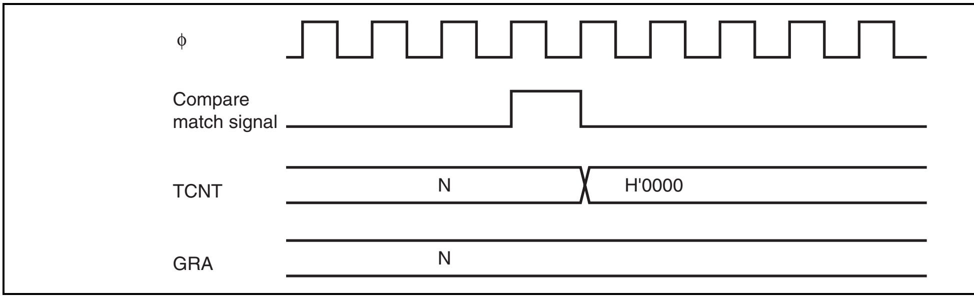


Figure 11.18 Timing of Counter Clearing by Compare Match

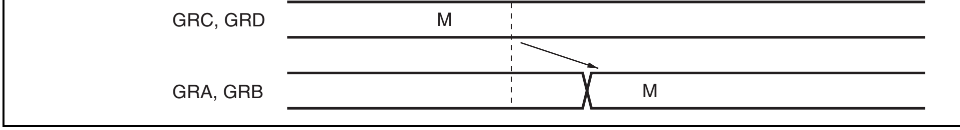


Figure 11.19 Buffer Operation Timing (Compare Match)

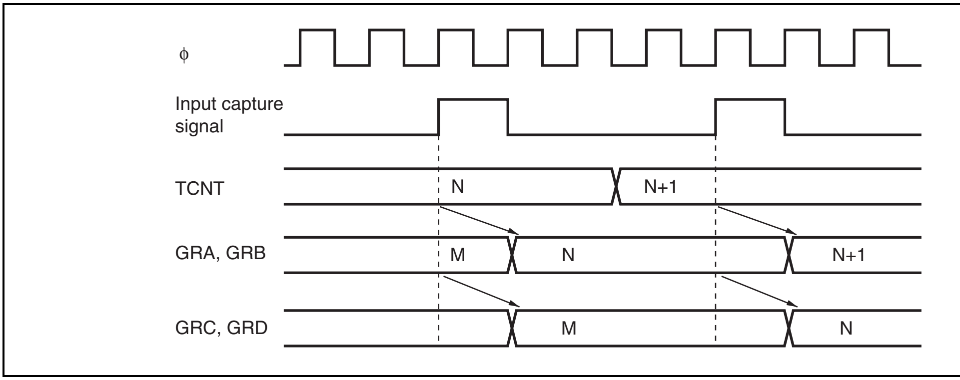


Figure 11.20 Buffer Operation Timing (Input Capture)

11.5.6 Timing of IMFA to IMFD Flag Setting at Compare Match

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

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

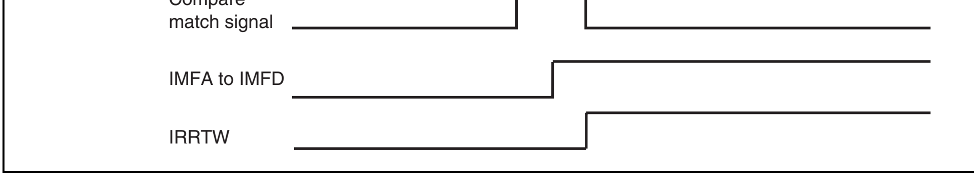


Figure 11.21 Timing of IMFA to IMFD Flag Setting at Compare Match

11.5.7 Timing of IMFA to IMFD Setting at Input Capture

If a general register (GRA, GRB, GRC, or GRD) is used as an input capture register, the corresponding IMFA, IMFB, IMFC, or IMFD flag is set to 1 when an input capture occur 11.22 shows the timing of the IMFA to IMFD flag setting at input capture.

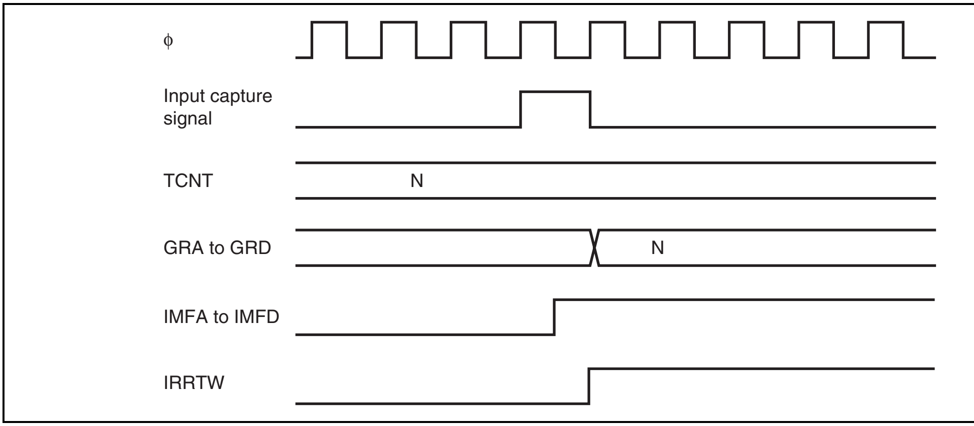


Figure 11.22 Timing of IMFA to IMFD Flag Setting at Input Capture

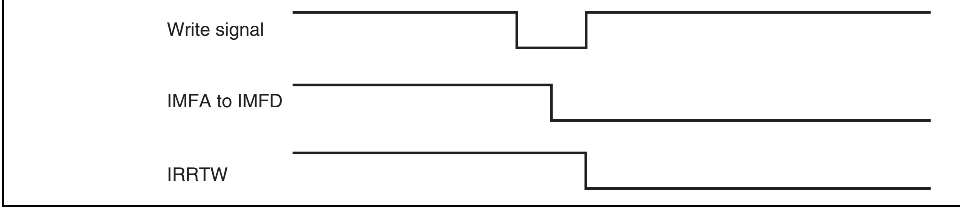


Figure 11.23 Timing of Status Flag Clearing by CPU

11.6 Usage Notes

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

1. The pulse width of the input clock signal and the input capture signal must be at leas system clock (ϕ) cycles; shorter pulses will not be detected correctly.
2. Writing to registers is performed in the T2 state of a TCNT write cycle.

   If counter clear signal occurs in the T2 state of a TCNT write cycle, clearing of the c takes priority and the write is not performed, as shown in figure 11.24. If counting-u generated in the TCNT write cycle to contend with the TCNT counting-up, writing t precedence.
3. Depending on the timing, TCNT may be incremented by a switch between different clock sources. When TCNT is internally clocked, an increment pulse is generated fro rising edge of an internal clock signal, that is divided system clock (ϕ). Therefore, as figure 11.25 the switch is from a low clock signal to a high clock signal, the switcho as a rising edge, causing TCNT to increment.
4. If timer W enters module standby mode while an interrupt request is generated, the i request cannot be cleared. Before entering module standby mode, disable interrupt re

Figure 11.24   Contention between TCNT Write and Clear

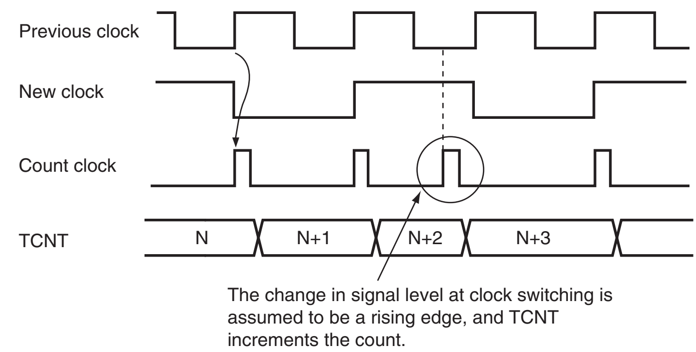


Figure 11.25   Internal Clock Switching and TCNT Operation

bit manipulation instruction to TCRW occur at the same timing.

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

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

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

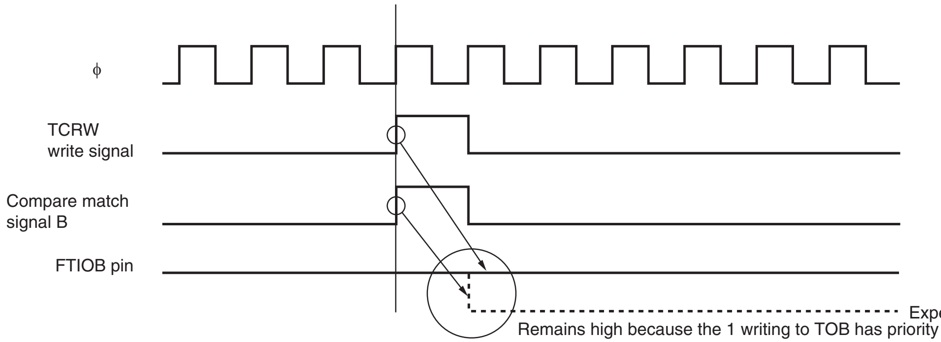


Figure 11.26 When Compare Match and Bit Manipulation Instruction to TC Occur at the Same Timing

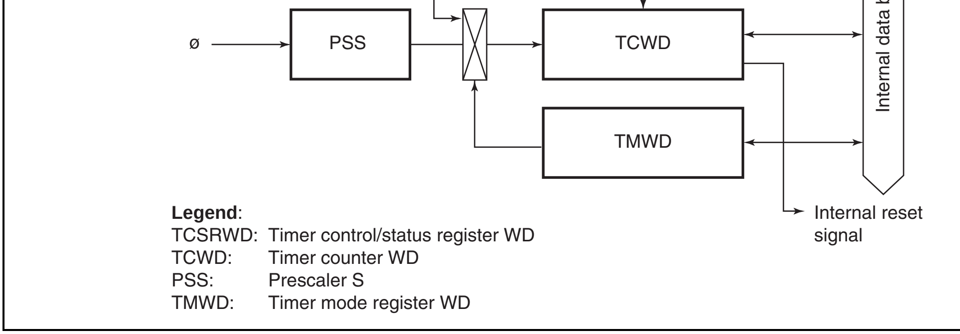


Figure 12.1 Block Diagram of Watchdog Timer

12.1 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.

WDT0110A_000020020200

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

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

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

12.2.2 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.

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 s 18, Electrical Characteristics.

Legend X: Don't care.

Figure 12.2 shows an example of watchdog timer operation.

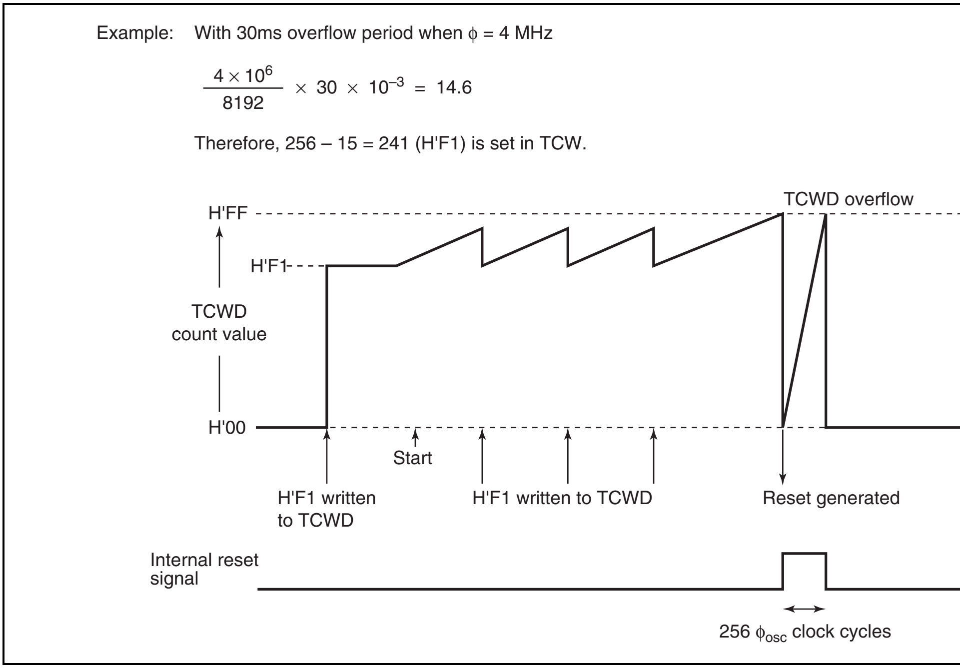


Figure 12.2 Watchdog Timer Operation Example

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

13.1 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 framing error

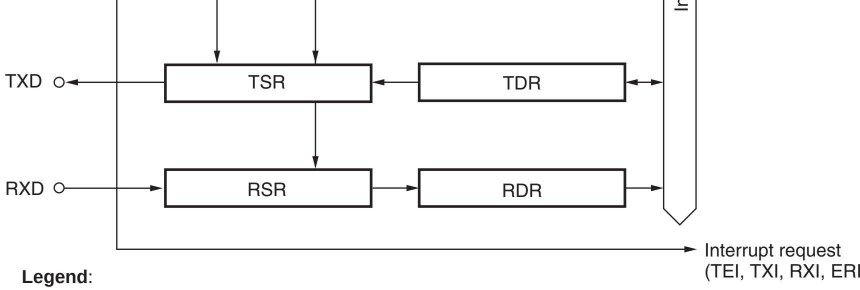
SCI0010A_000120030300

			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	—
Channel 3*[2]	SCI3_3	SCK3_3*[3] RXD_3 TXD_3	SMR_3	H'F600
			BRR_3	H'F601
			SCR3_3	H'F602
			TDR_3	H'F603
			SSR_3	H'F604
			RDR_3	H'F605
			RSR_3	—
			TSR_3	—
			SMCR	H'F608

Notes: 1. The channel 1 of the SCI3 is used in on-board programming mode by boot m

2. The SCI3_3 function is incorporated in the H8/36024.

3. When this pin is used as the SCI3_3 function with the emulator used, the cor PCR value must be cleared to 0.



Legend:

RSR: Receive shift register
RDR: Receive data register
TSR: Transmit shift register
TDR: Transmit data register
SMR: Serial mode register
SCR3: Serial control register 3
SSR: Serial status register
BRR: Bit rate register
BRC: Bit rate counter

Figure 13.1 Block Diagram of SCI3

13.3 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)
- SCI3_3 Module Control Register (SMCR)

receive-enabled. As RSR and RDR function as a double buffer in this way, continuous re
operations are possible. After confirming that the RDRF bit in SSR is set to 1, read RDR
once. RDR cannot be written to by the CPU. RDR is initialized to H'00.

13.3.3 Transmit Shift Register (TSR)

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

13.3.4 Transmit Data Register (TDR)

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

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

and the baud rate, see section 13.3.8, Bit Rate R
(BRR). n is the decimal representation of the val
BRR (see section 13.3.8, Bit Rate Register (BRR

13.3.6 Serial Control Register 3 (SCR3)

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

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, the TXI interrupt reques enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt re are enabled.
5	TE	0	R/W	Transmit Enable When this bit 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 t 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 01:Reserved 10: External clock (SCK3 pin functions as clock 11:Reserved

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

				• When TDRE = 1 at transmission of the last frame serial transmit character [Clearing conditions] • When 0 is written to TDRE after reading TD • When the transmit data is written to TDR
1	MPBR	0	R	Multiprocessor Bit Receive MPBR stores the multiprocessor bit in the recei character data. When the RE bit in SCR3 is cle its state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be addec 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 ($0 \leq N \leq 255$)

ϕ: Operating frequency (MHz)

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

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

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			12	
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	212
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155
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	155
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	155
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	10
2400	0	159	0.00	0	181	0.16	0	191	0.00	0	20
4800	0	79	0.00	0	90	0.16	0	95	0.00	0	10
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.

				used, these bits must be cleared to 0.
1	TXD_3	0	R/W	TXD_3 Output Select Selects the function of the P57/TXD_3 pin. 0: General I/O port 1: TXD_3 output pin
0	MSTS3_3	0	R/W	SCI3_3 Module Standby When this bit is set to 1, the SCI3_3 enters th state.

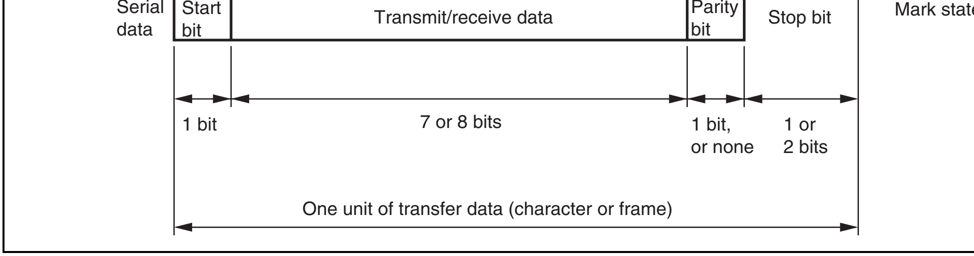


Figure 13.2 Data Format in Asynchronous Communication

13.4.1 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 13.3.

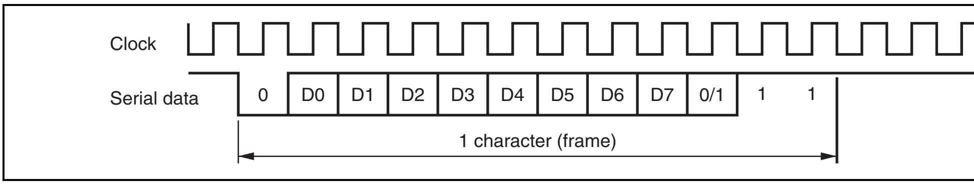
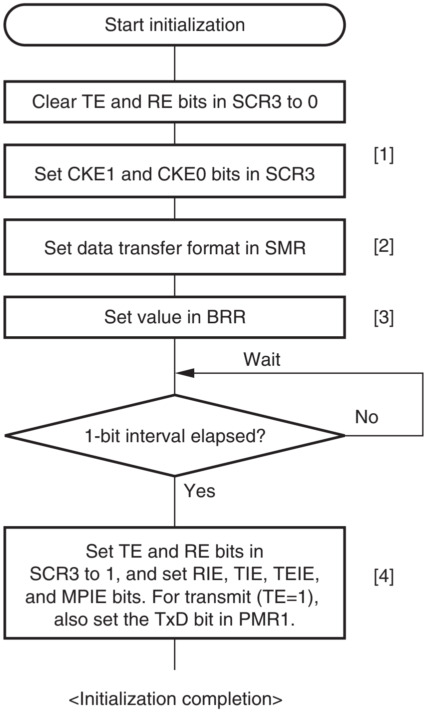


Figure 13.3 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode)(Example with 8-Bit Data, Parity, Two Stop Bits)



[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 13.4 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 13.6 shows a sample flowchart for transmission in asynchronous mode.

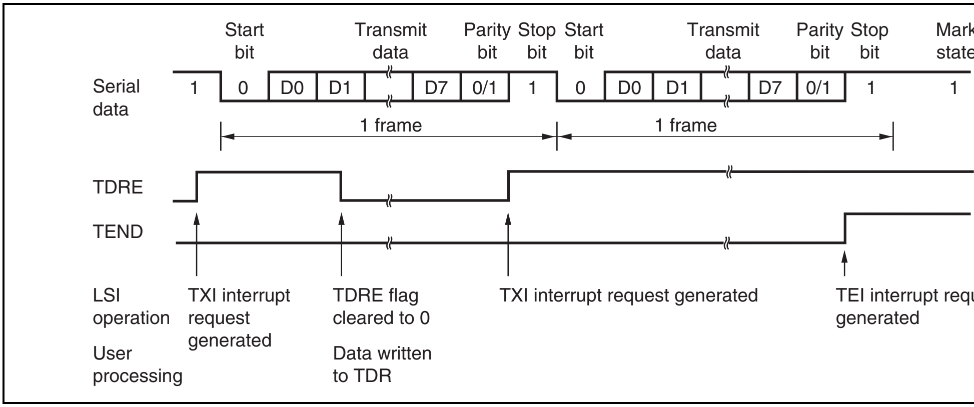


Figure 13.5 Example of SCI3 Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)

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

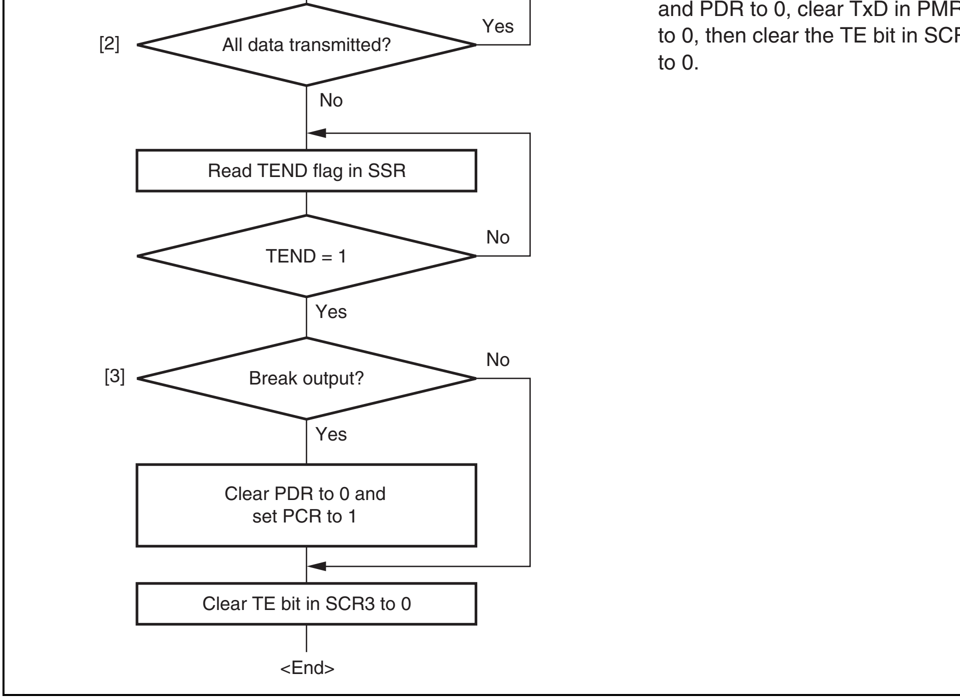


Figure 13.6 Sample Serial Transmission Data Flowchart (Asynchronous Mo

3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to 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.

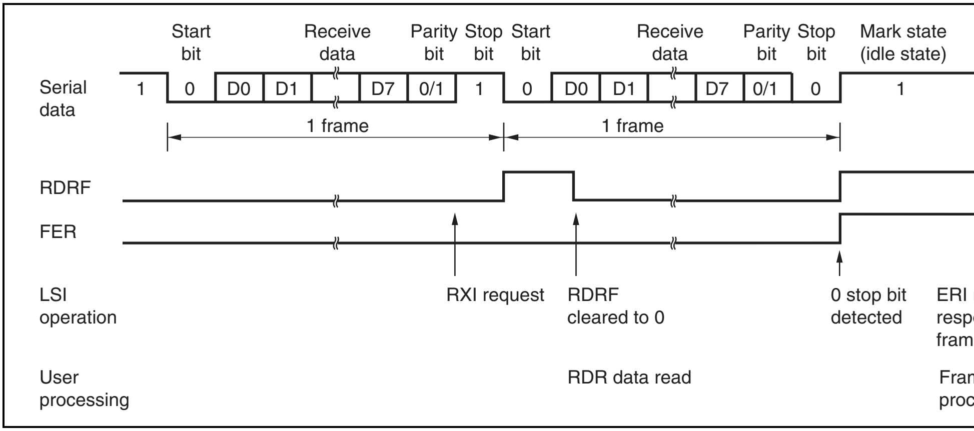
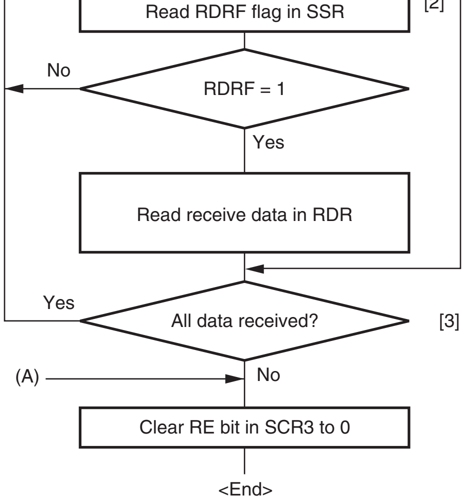


Figure 13.7 Example of SCI3 Reception in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)

0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + frami
1	1	0	1	Lost	Overrun error + parity
0	0	1	1	Transferred to RDR	Framing error + parity
1	1	1	1	Lost	Overrun error + frami parity error

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



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 13.8 Sample Serial Reception Data Flowchart (Asynchronous Mode)(

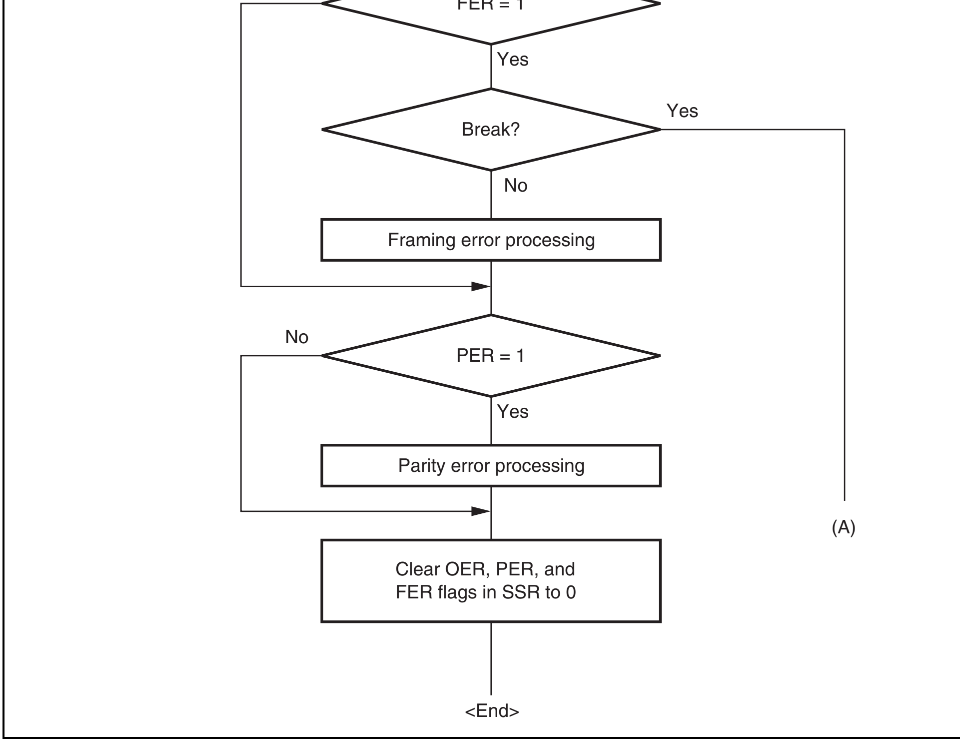


Figure 13.8 Sample Serial Reception Data Flowchart (Asynchronous Mode

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.

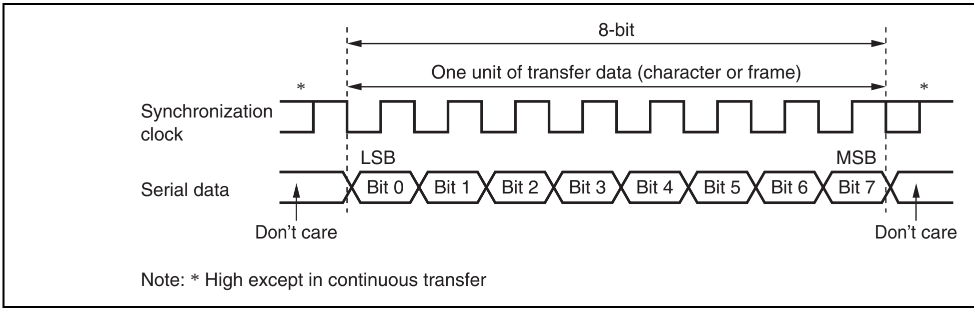


Figure 13.9 Data Format in Clocked Synchronous Communication

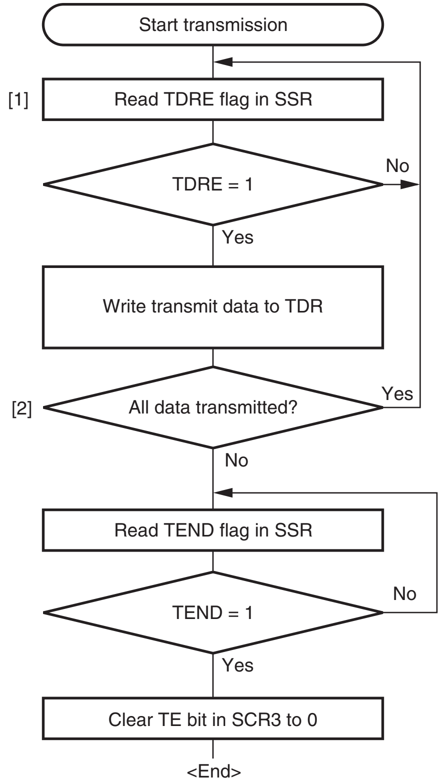
13.5.1 Clock

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

1. The SCI3 monitors the TDRE flag in SSR, and if the flag is 0, the SCI3 recognizes t
has been written to TDR, and transfers the data from TDR to TSR.
2. The SCI3 sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR3 is s
this time, a transmit data empty interrupt (TXI) is generated.
3. 8-bit data is sent from the TXD pin synchronized with the output clock when output
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 tran
of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TDRE flag m
the output state of the last bit. If the TEIE bit in SCR3 is set to 1 at this time, a TEI
request is generated.
7. The SCK3 pin is fixed high at the end of transmission.

Figure 13.11 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 13.10 Example of SCI3 Transmission in Clocked Synchronous Mod



[1] Read SSR and check that the TDRE flag i
set to 1, then write transmit data to TDR.
When data is written to TDR, the TDRE fla
is automatically cleared to 0 and clocks ar
output to start the data transmission.

[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 fla
is automatically cleared to 0.

Figure 13.11 Sample Serial Transmission Flowchart (Clocked Synchronous M

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

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

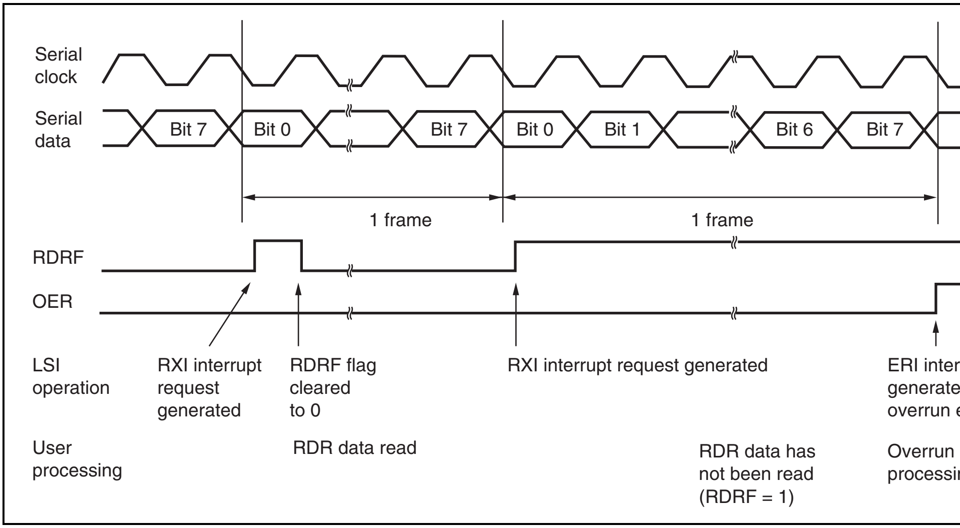
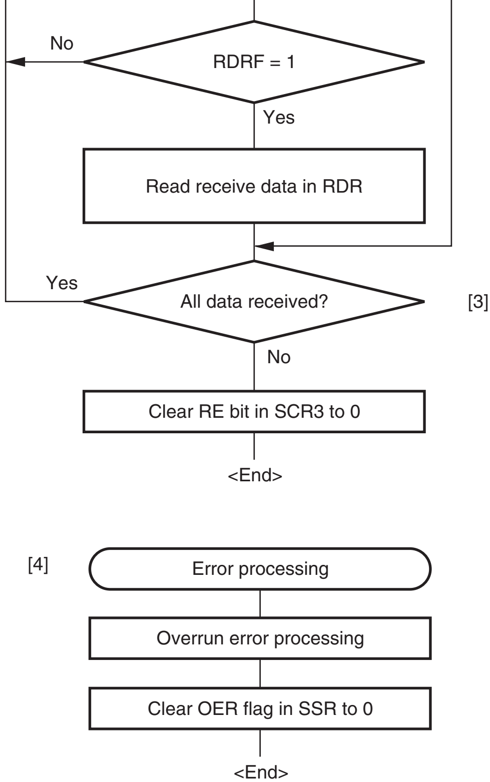


Figure 13.12 Example of SCI3 Reception in Clocked Synchronous Mode

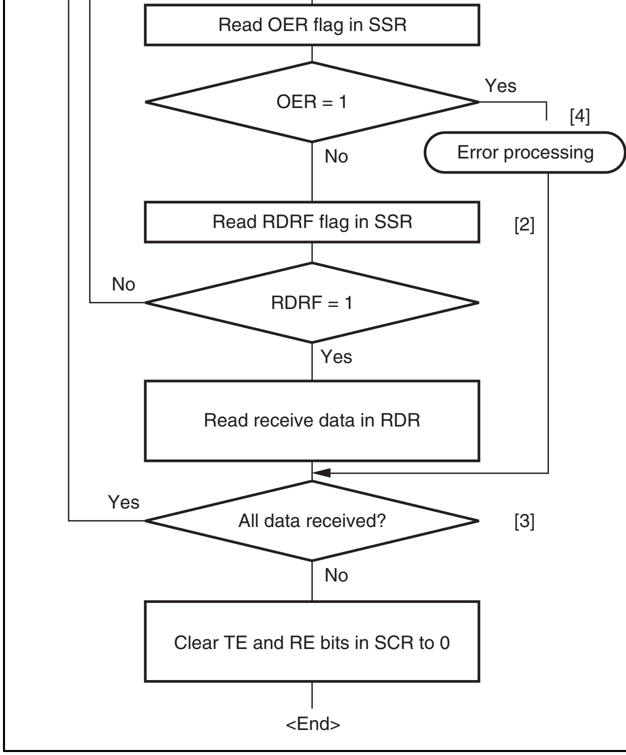
Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the FER, PER, and RDRF bits to 0 before resuming reception. Figure 13.13 shows a sample chart for serial data reception.



cleared to 0.

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

Figure 13.13 Sample Serial Reception Flowchart (Clocked Synchronous Mo



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

[4] If an overrun error occurs, read t
OER flag in SSR, and after
performing the appropriate error
processing, clear the OER flag t
Transmission/reception cannot b
resumed if the OER flag is set to
For overrun error processing, se
figure 13.13.

Figure 13.14 Sample Flowchart of Simultaneous Serial Transmit and Receive Ope
(Clocked Synchronous Mode)

cycle is a data transmission cycle. Figure 13.13 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends the
of the receiving station with which it wants to perform serial communication as data wit
multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit
When data with a 1 multiprocessor bit is received, the receiving station compares that da
own ID. The station whose ID matches then receives the data sent next. Stations whose
match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI3 uses the MPIE bit in SCR3 to implement this function. When the MPIE bit is
transfer of receive data from RSR to RDR, error flag detection, and setting the SSR statu
RDRF, FER, and OER, to 1, are inhibited until data with a 1 multiprocessor bit is receiv
reception of a receive character with a 1 multiprocessor bit, the MPBR bit in SSR is set
the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit 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. Al
settings are the same as those in normal asynchronous mode. The clock used for multipr
communication is the same as that in normal asynchronous mode.

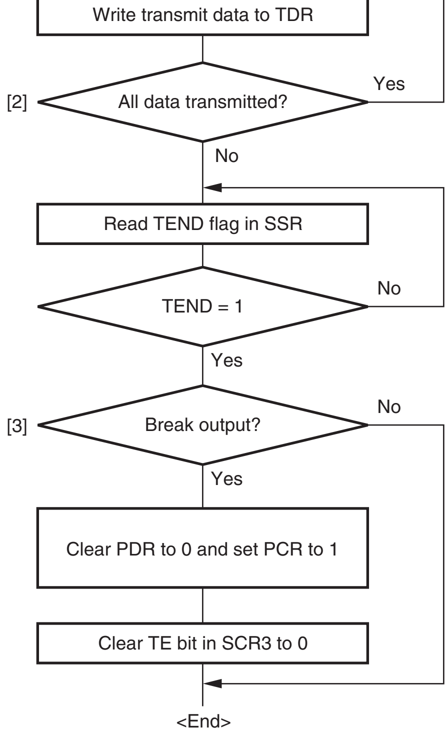
specification receiving station specified by ID

Legend
MPB: Multiprocessor bit

Figure 13.15 Example of Inter-Processor Communication Using Multiprocessor F
(Transmission of Data H'AA to Receiving Station A)

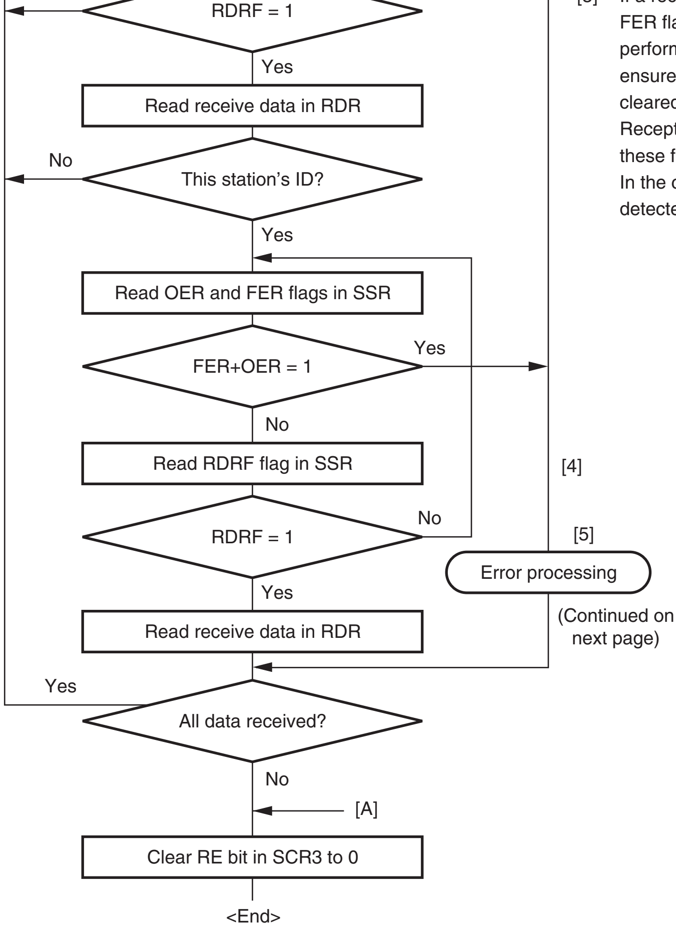
13.6.1 Multiprocessor Serial Data Transmission

Figure 13.16 shows a sample flowchart for multiprocessor serial data transmission. For an transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmis cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI3 operations are as those in asynchronous mode.



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

Figure 13.16 Sample Multiprocessor Serial Transmission Flowchart



[5] If a receive error occurs, read the OER and FER flags in SSR to identify the erro performing the appropriate error pro ensure that the OER and FER flags cleared to 0.
Reception cannot be resumed if eith these flags is set to 1.
In the case of a framing error, a brea detected by reading the RxD pin valu

Figure 13.17 Sample Multiprocessor Serial Reception Flowchart (1)

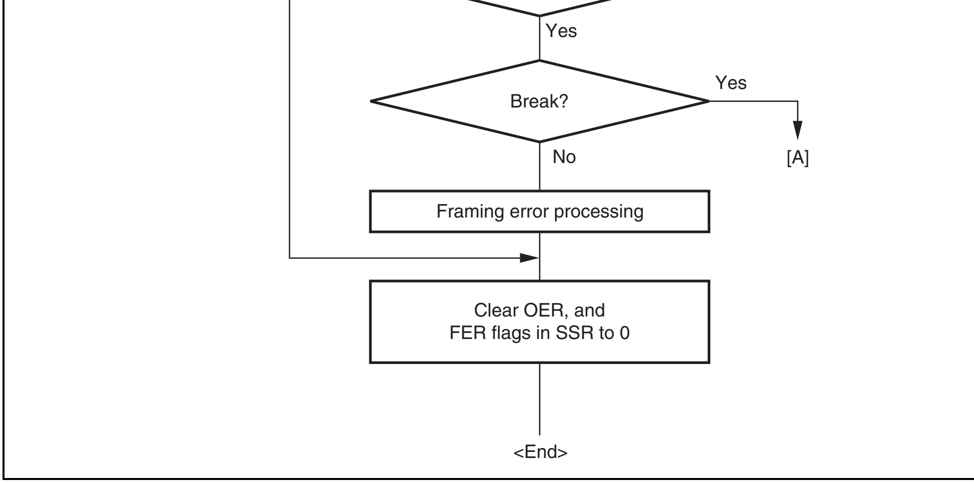


Figure 13.17 Sample Multiprocessor Serial Reception Flowchart (2)

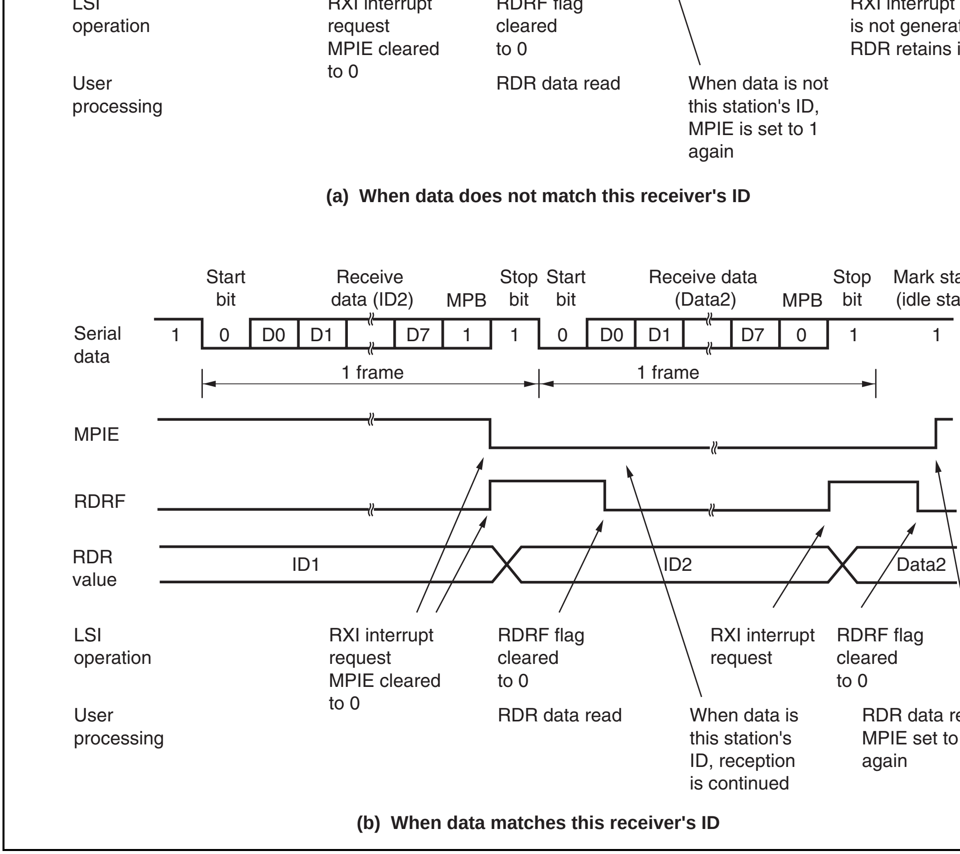


Figure 13.18 Example of SCI3 Reception Using Multiprocessor Format
(Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

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

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

13.8.3 Receive Error Flags and Transmit Operations (Clocked Synchronous Mo

Transmission cannot be started when a receive error flag (OER, PER, or FER) is set to 1
the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting
transmission. Note also that receive error flags cannot be cleared to 0 even if the RE bit
to 0.

... 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 allowe
system design.

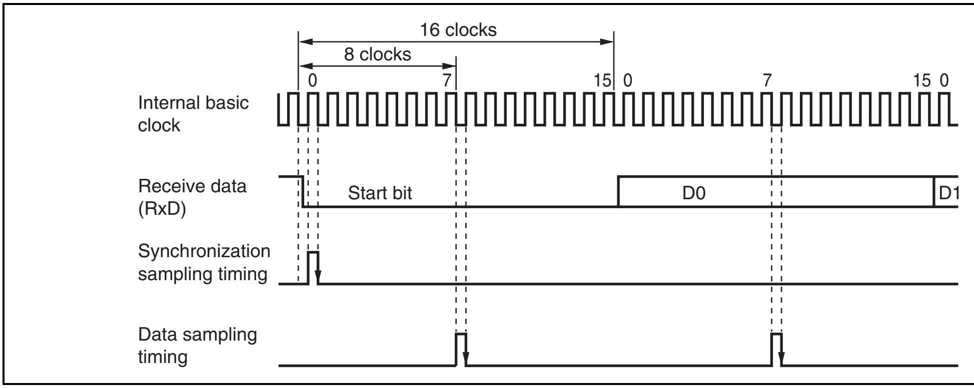


Figure 13.19 Receive Data Sampling Timing in Asynchronous Mode

- 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 16-bit 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

ADCMS32A_000020020200

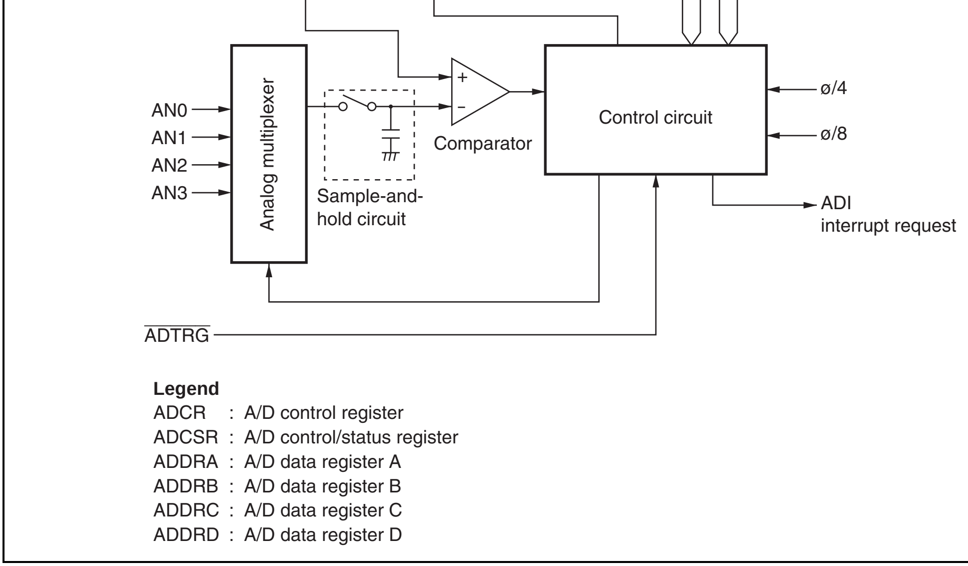


Figure 14.1 Block Diagram of A/D Converter

Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
A/D external trigger input pin	$\overline{ADTRG}$	Input	External trigger input pin for sta conversion

14.3.1 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 channel,
shown in table 14.2.

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

The data bus between the CPU and the A/D converter is 8 bits wide. The upper byte can b
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 14.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel	A/D Data Register to Be Stored Results of A/D Conversio
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD

				selected in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1
6	ADIE	0	R/W	A/D Interrupt Enable A/D conversion end interrupt (ADI) request enab ADF when 1 is set
5	ADST	0	R/W	A/D Start Setting this bit to 1 starts A/D conversion. In sin 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.

14.3.3 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 from the first channel when the ADST bit in ADCSR is se
   according to software or external trigger input.
2. When A/D conversion is completed, the result is transferred to the corresponding A
   register to the channel.
3. On completion of conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit is
   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,
   ADST bit is automatically cleared to 0 and the A/D converter enters the wait state.

14.4.2 Scan Mode

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

1. When the ADST bit is set to 1 by software, or external trigger input, A/D conversion
   the first channel in the group.
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 is requested. Conversion of t
   channel in the group starts again.
4. The ADST bit is not automatically cleared to 0. Steps [2] to [3] are repeated as long
   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 14.3 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.

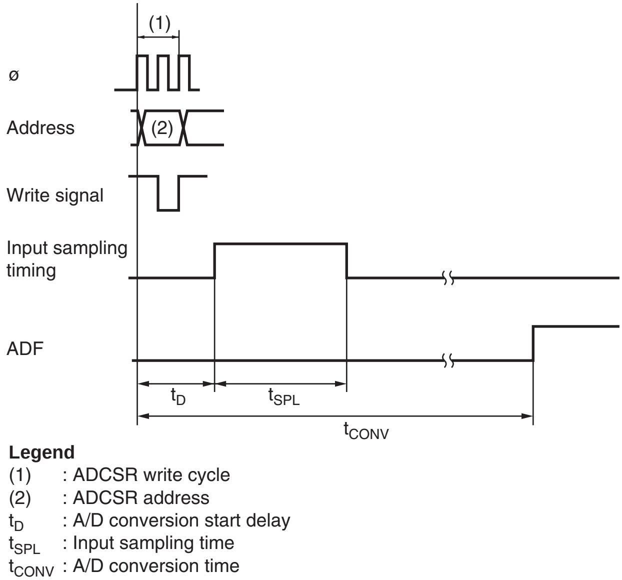


Figure 14.2 A/D Conversion Timing

14.4.4 External Trigger Input Timing

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

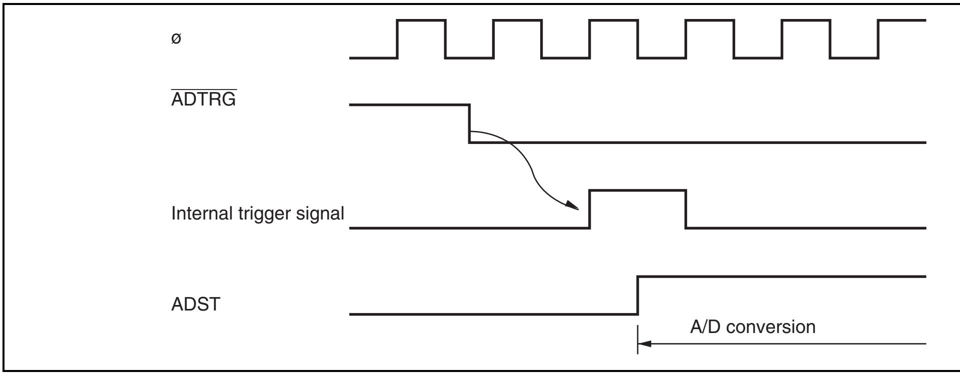


Figure 14.3 External Trigger Input Timing

when the digital output changes from the minimum voltage value 0000000000 to 000
(see figure 14.5).

- 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 14.5).

- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristics between zero voltag
  full-scale voltage. Does not include offset error, full-scale error, or quantization error.

- Absolute accuracy

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

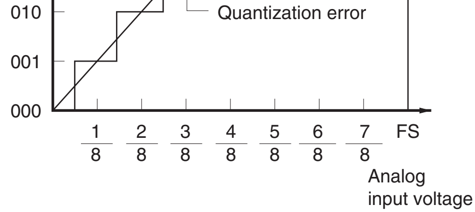


Figure 14.4 A/D Conversion Accuracy Definitions (1)

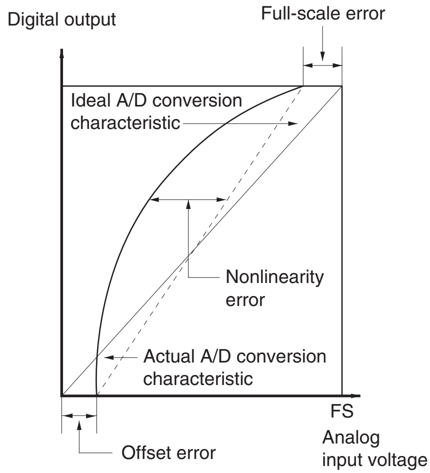


Figure 14.5 A/D Conversion Accuracy Definitions (2)

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

14.6.2 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.

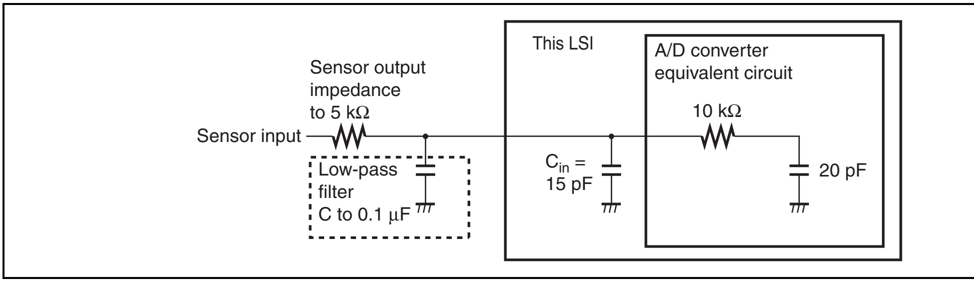


Figure 14.6 Analog Input Circuit Example

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 wher
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 15.1 is a block diagram of the power-on reset circuit and the low-voltage detectio

15.1 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.

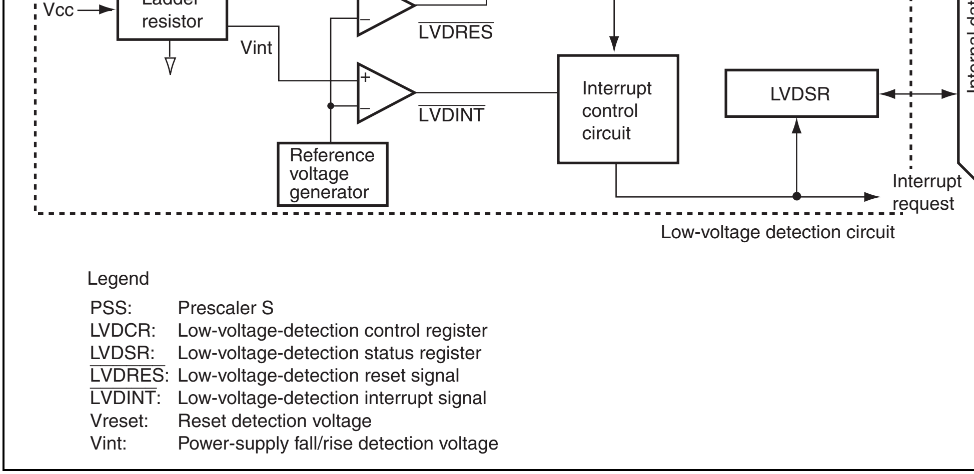


Figure 15.1 Block Diagram of Power-On Reset Circuit and Low-Voltage Detection

15.2 Register Descriptions

The low-voltage detection circuit has the following registers.

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

7	LVDE	0*	R/W	LVD Enable 0: The low-voltage detection circuit is not used standby mode) 1: The low-voltage detection circuit is used
6 to 4	—	All 1	—	Reserved 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.

15.2.2 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 m
1	LVDDF	0*	R/W	LVD Power-Supply Voltage Fall Flag [Setting condition] When the power-supply voltage falls below Vint (D 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 (D the LVDUE bit in LVDCR is set to 1, then rises ab (U) (typ. = 4.0 V) before falling below Vreset1 (typ. [Clearing condition] Writing 0 to this bit after reading it as 1

Note: * Initialized by LVDR.

131,072 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 cap
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 forr

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

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

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.

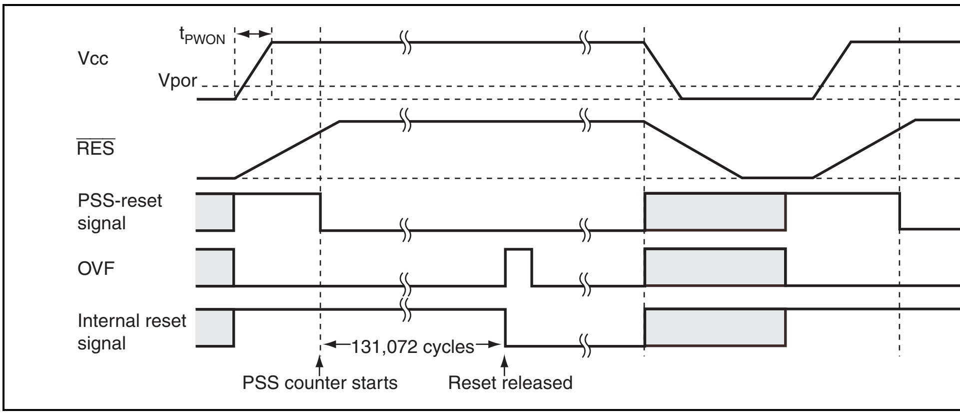


Figure 15.2 Operational Timing of Power-On Reset Circuit

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 rise the Vreset voltage again, the prescaler S starts counting. It counts 131,072 clock (ϕ) cycle then releases the internal reset signal. In this case, the LVDE, LVDSEL, and LVDRE bit 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.

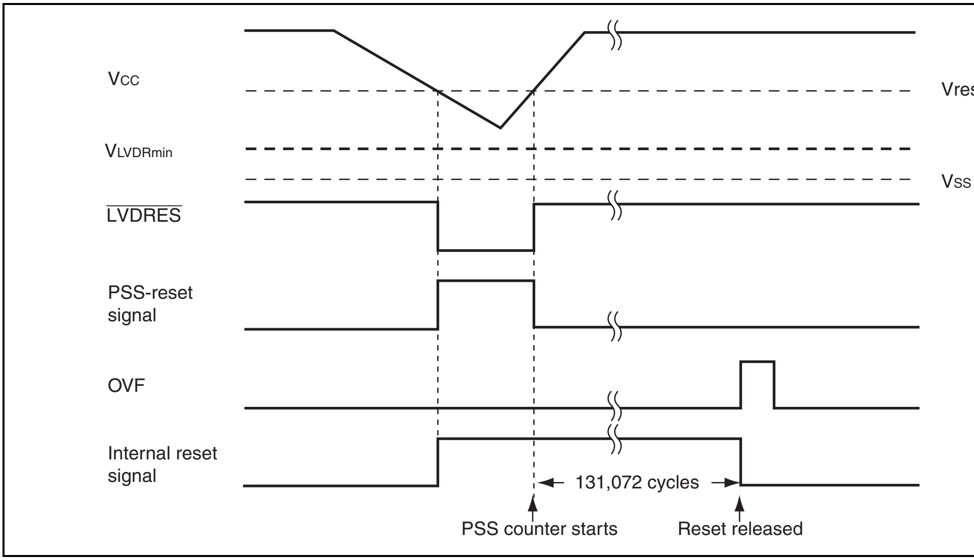


Figure 15.3 Operational Timing of LVDR Circuit

When the power-supply voltage falls below Vint (D) (typ. = 3.7 V) voltage, the LVDI c
$\overline{\text{LVDINT}}$ 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
this time, the LVDUF bit in LVDSR is set to 1 and an IRQ0 interrupt request is simultar
generated.

If the power supply voltage (Vcc) falls below Vreset1 (typ. = 2.3 V) voltage, the LVDR
is performed.

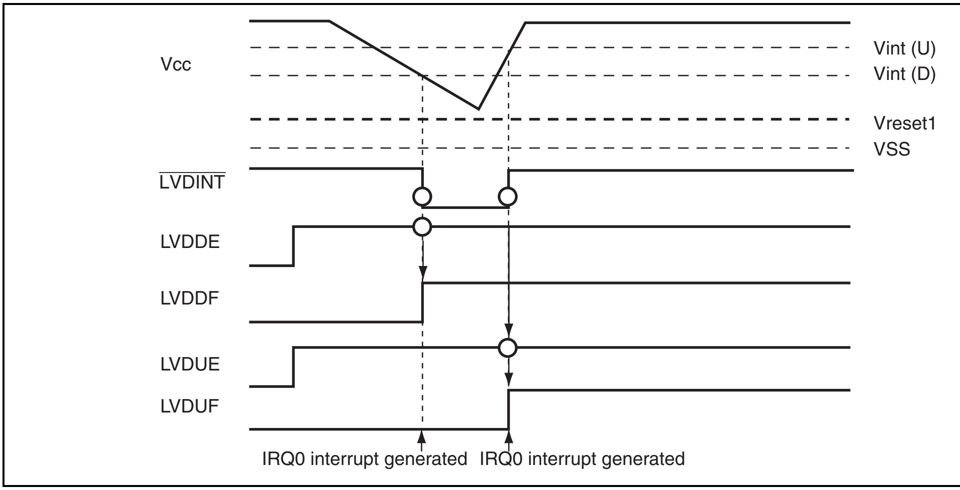


Figure 15.4 Operational Timing of LVDI Circuit

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

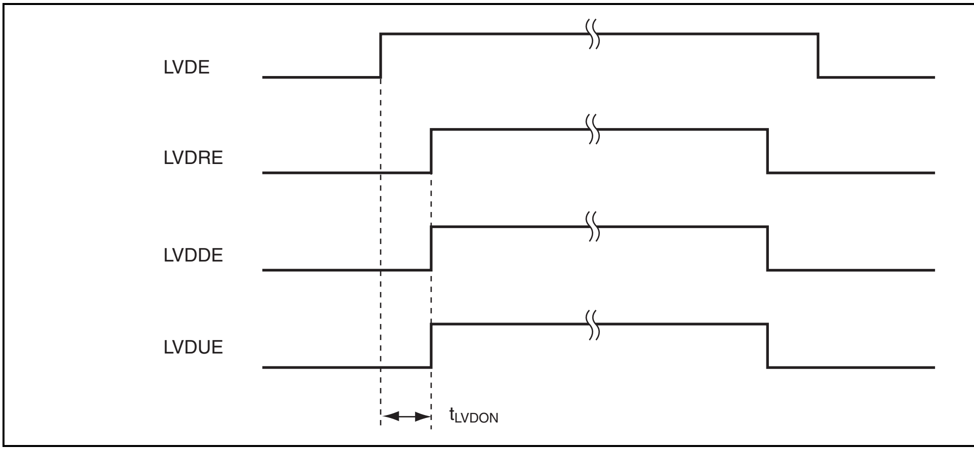


Figure 15.5 Timing for Operation/Release of Low-Voltage Detection Circui

16.1 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 16.1. The internal step-down circuit is made
simply by adding this external circuit. In the external circuit interface, the external powe
voltage connected to V_{CC} and the GND potential connected to V_{SS} are the reference levels
example, for port input/output levels, the V_{CC} level is the reference for the high level, and
level is that for the low level. The A/D converter analog power supply is not affected by
internal step-down circuit.

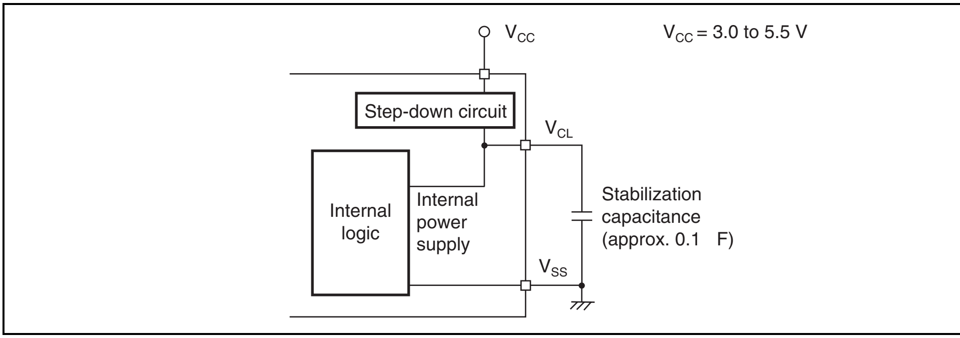


Figure 16.1 Power Supply Connection when Internal Step-Down Circuit is U

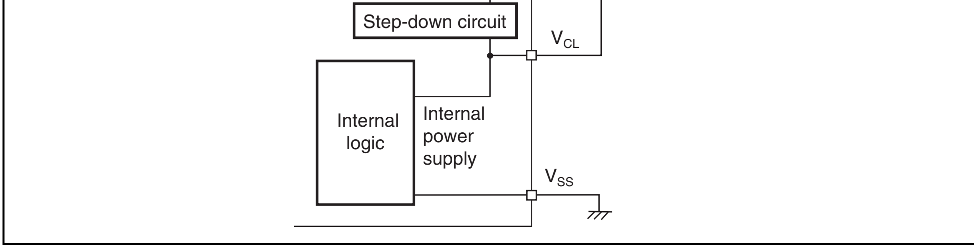


Figure 16.2 Power Supply Connection when Internal Step-Down Circuit is Not

- 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

Serial control register 3_3	SCR3_3	8	H'F602	SCI3_3	8
Transmit data register_3	TDR_3	8	H'F603	SCI3_3	8
Serial status register_3	SSR_3	8	H'F604	SCI3_3	8
Receive data register_3	RDR_3	8	H'F605	SCI3_3	8
—	—	—	H'F606, H'F607	SCI3_3	—
SCI3_3 module control register	SMCR	8	H'F608	SCI3_3	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
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
Receive data register_2	RDR_2	8	H'F745	SCI3_2	8
Timer mode register W	TMRW	8	H'FF80	Timer W	8
Timer control register W	TCRW	8	H'FF81	Timer W	8
Timer interrupt enable register W	TIERW	8	H'FF82	Timer W	8
Timer status register W	TSRW	8	H'FF83	Timer W	8
Timer I/O control register 0	TIOR0	8	H'FF84	Timer W	8
Timer I/O control register 1	TIOR1	8	H'FF85	Timer W	8
Timer counter	TCNT	16	H'FF86	Timer W	16*2
General register A	GRA	16	H'FF88	Timer W	16*2

Timer control register V0	TCRV0	8	H'FFA0	Timer V	8
Timer control/status register V	TCSRV	8	H'FFA1	Timer V	8
Timer constant register A	TCORA	8	H'FFA2	Timer V	8
Timer constant register B	TCORB	8	H'FFA3	Timer V	8
Timer counter V	TCNTV	8	H'FFA4	Timer V	8
Timer control register V1	TCRV1	8	H'FFA5	Timer V	8
Serial mode register	SMR	8	H'FFA8	SCI3	8
Bit rate register	BRR	8	H'FFA9	SCI3	8
Serial control register 3	SCR3	8	H'FFAA	SCI3	8
Transmit data register	TDR	8	H'FFAB	SCI3	8
Serial status register	SSR	8	H'FFAC	SCI3	8
Receive data register	RDR	8	H'FFAD	SCI3	8
A/D data register A	ADDRA	16	H'FFB0	A/D converter	8
A/D data register B	ADDRB	16	H'FFB2	A/D converter	8
A/D data register C	ADDRC	16	H'FFB4	A/D converter	8
A/D data register D	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
Timer control/status register WD	TCSRWD	8	H'FFC0	WDT*3	8
Timer counter WD	TCWD	8	H'FFC1	WDT*3	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
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 5	PDR5	8	H'FFD8	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
Port data register B	PDRB	8	H'FFDD	I/O port	8
Port mode register 1	PMR1	8	H'FFE0	I/O port	8
Port mode register 5	PMR5	8	H'FFE1	I/O port	8
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 5	PCR5	8	H'FFE8	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
System control register 1	SYSCR1	8	H'FFF0	Power-down	8
System control register 2	SYSCR2	8	H'FFF1	Power-down	8
Interrupt edge select register 1	IEGR1	8	H'FFF2	Interrupts	8

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

TDR_3	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	
SSR_3	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT	
RDR_3	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0	
SMCR	–	–	–	–	–	–	TXD_3	MSTS3_3	
LVDCR	LVDE	–	–	–	LVDSEL	LVDRE	LVDDE	LVDUE	LV (op
LVDSR	–	–	–	–	–	–	LVDDF	LVDUF	
SMR_2	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0	SC
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	
TMRW	CTS	—	BUFEB	BUFEA	—	PWMD	PWMC	PWMB	Tir
TCRW	CCLR	CKS2	CKS1	CKS0	TOD	TOC	TOB	TOA	
TIERW	OVIE	—	—	—	IMIED	IMIEC	IMIEB	IMIEA	
TSRW	OVF	—	—	—	IMFD	IMFC	IMFB	IMFA	
TIOR0	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0	
TIOR1	—	IOD2	IOD1	IOD0	—	IOC2	IOC1	IOC0	
TCNT	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8	
	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0	
GRA	GRA15	GRA14	GRA13	GRA12	GRA11	GRA10	GRA9	GRA8	
	GRA7	GRA6	GRA5	GRA4	GRA3	GRA2	GRA1	GRA0	
GRB	GRB15	GRB14	GRB13	GRB12	GRB11	GRB10	GRB9	GRB8	
	GRB7	GRB6	GRB5	GRB4	GRB3	GRB2	GRB1	GRB0	

TCRV0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	T
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	S
BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0	
SCR3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	
SSR	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT	
RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0	
ADDRA	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A
	AD1	AD0	—	—	—	—	—	—	
ADDRB	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRC	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRD	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0	
ADCR	TRGE	—	—	—	—	—	—	—	
TCSRWD	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI	WRST	W
TCWD	TCWD7	TCWD6	TCWD5	TCWD4	TCWD3	TCWD2	TCWD1	TCWD0	
TMWD	—	—	—	—	CKS3	CKS2	CKS1	CKS0	

PDR1	P17	P16	P15	P14	—	P12	P11	P10	
PDR2	—	—	—	—	—	P22	P21	P20	
PDR5	P57	P56	P55	P54	P53	P52	P51	P50	
PDR7	—	P76	P75	P74	P73	P72	P71	P70	
PDR8	—	—	—	P84	P83	P82	P81	P80	
PDRB	—	—	—	—	PB3	PB2	PB1	PB0	
PMR1	IRQ3	—	—	IRQ0	TXD2	—	TXD	—	
PMR5	POF57	POF56	WKP5	WKP4	WKP3	WKP2	WKP1	WKP0	
PCR1	PCR17	PCR16	PCR15	PCR14	—	PCR12	PCR11	PCR10	
PCR2	—	—	—	—	—	PCR22	PCR21	PCR20	
PCR5	PCR57	PCR56	PCR55	PCR54	PCR53	PCR52	PCR51	PCR50	
PCR7	—	PCR76	PCR75	PCR74	PCR73	PCR72	PCR71	PCR70	
PCR8	—	—	—	PCR84	PCR83	PCR82	PCR81	PCR80	
SYSCR1	SSBY	STS2	STS1	STS0	—	—	—	—	Po
SYSCR2	SMSEL	—	DTON	MA2	MA1	MA0	—	—	
IEGR1	—	—	—	—	IEG3	—	—	IEG0	Int
IEGR2	—	—	WPEG5	WPEG4	WPEG3	WPEG2	WPEG1	WPEG0	
IENR1	IENDT	—	IENWP	—	IEN3	—	—	IEN0	
IRR1	IRRDT	—	—	—	IRRI3	—	—	IRRI0	
IWPR	—	—	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0	
MSTCR1	—	—	MSTS3	MSTAD	MSTWD	MSTTW	MSTTV	—	Po
MSTCR2	MSTS3_2	—	—	—	—	—	—	—	

Note: * WDT: Watchdog timer

SMCR	Initialized	—	—	Initialized	Initialized	
LVDCR	Initialized	—	—	—	—	LVDC (optional)
LVDSR	Initialized	—	—	—	—	
SMR_2	Initialized	—	—	Initialized	Initialized	SCI3_2
BRR_2	Initialized	—	—	Initialized	Initialized	
SCR3_2	Initialized	—	—	Initialized	Initialized	
TDR_2	Initialized	—	—	Initialized	Initialized	
SSR_2	Initialized	—	—	Initialized	Initialized	
RDR_2	Initialized	—	—	Initialized	Initialized	
TMRW	Initialized	—	—	—	—	Timer W
TCRW	Initialized	—	—	—	—	
TIERW	Initialized	—	—	—	—	
TSRW	Initialized	—	—	—	—	
TIOR0	Initialized	—	—	—	—	
TIOR1	Initialized	—	—	—	—	
TCNT	Initialized	—	—	—	—	
GRA	Initialized	—	—	—	—	
GRB	Initialized	—	—	—	—	
GRC	Initialized	—	—	—	—	
GRD	Initialized	—	—	—	—	
FLMCR1	Initialized	—	—	Initialized	Initialized	ROM
FLMCR2	Initialized	—	—	—	—	
EBR1	Initialized	—	—	Initialized	Initialized	
FENR	Initialized	—	—	—	—	

SCR3	Initialized	—	—	Initialized	Initialized	
TDR	Initialized	—	—	Initialized	Initialized	
SSR	Initialized	—	—	Initialized	Initialized	
RDR	Initialized	—	—	Initialized	Initialized	
ADDRA	Initialized	—	—	Initialized	Initialized	A/D converter
ADDRB	Initialized	—	—	Initialized	Initialized	
ADDRC	Initialized	—	—	Initialized	Initialized	
ADDRD	Initialized	—	—	Initialized	Initialized	
ADCSR	Initialized	—	—	Initialized	Initialized	
ADCR	Initialized	—	—	Initialized	Initialized	
TCSRWD	Initialized	—	—	—	—	WDT*
TCWD	Initialized	—	—	—	—	
TMWD	Initialized	—	—	—	—	
ABRKCR	Initialized	—	—	—	—	Address Break
ABRKSR	Initialized	—	—	—	—	
BARH	Initialized	—	—	—	—	
BARL	Initialized	—	—	—	—	
BDRH	Initialized	—	—	—	—	
BDRL	Initialized	—	—	—	—	
PUCR1	Initialized	—	—	—	—	I/O port
PUCR5	Initialized	—	—	—	—	
PDR1	Initialized	—	—	—	—	
PDR2	Initialized	—	—	—	—	
PDR5	Initialized	—	—	—	—	
PDR7	Initialized	—	—	—	—	

PCR8	Initialized	—	—	—	—	
SYSCR1	Initialized	—	—	—	—	Power-down
SYSCR2	Initialized	—	—	—	—	
IEGR1	Initialized	—	—	—	—	Interrupts
IEGR2	Initialized	—	—	—	—	
IENR1	Initialized	—	—	—	—	
IRR1	Initialized	—	—	—	—	
IWPR	Initialized	—	—	—	—	
MSTCR1	Initialized	—	—	—	—	Power-down
MSTCR2	Initialized	—	—	—	—	

Note: — is not initialized

* WDT: Watchdog timer

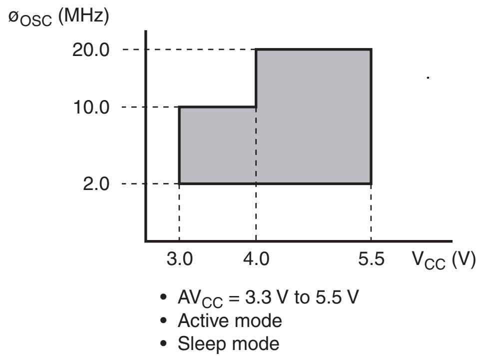
	Port B		–0.3 to AV_{cc} +0.3	V
Operating temperature		T_{opr}	–20 to +75	°C
Storage temperature		T_{stg}	–55 to +125	°C

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.

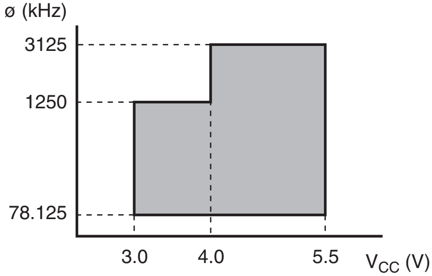
18.2 Electrical Characteristics (F-ZTAT™ Version)

18.2.1 Power Supply Voltage and Operating Ranges

(1) Power Supply Voltage and Oscillation Frequency Range

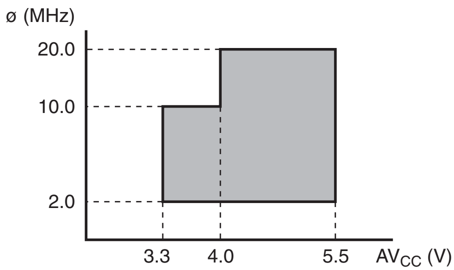


- AV_{CC} = 3.3 V to 5.5 V
- Active mode
- Sleep mode
  (When MA2 = 0 in SYSCR2)

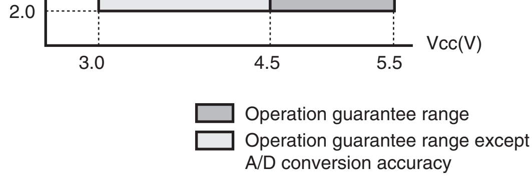


- AV_{CC} = 3.3 V to 5.5 V
- Active mode
- Sleep mode
  (When MA2 = 1 in SYSCR2)

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



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


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

		TMCIV, FTCI, FTIOA to FTIOD, SCK3, SCK3_2, SCK3_3*[1], TRGV		$V_{cc} \times 0.9$	—	$V_{cc} + 0.3$	
		RXD, RXD_2, RXD_3*[1], P12 to P10, P17 to P14, P22 to P20,	V_{cc} = 4.0 V to 5.5 V	$V_{cc} \times 0.7$	—	$V_{cc} + 0.3$	V
		P57 to P50, P76 to P70, P84 to P80		$V_{cc} \times 0.8$	—	$V_{cc} + 0.3$	
		PB3 to PB0	V_{cc} = 4.0 V 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 V 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 $\overline{WKP5}$, $\overline{IRQ0}$, $\overline{IRQ3}$, $\overline{ADTRG}$,TMRIV,	V_{cc} = 4.0 V to 5.5 V	–0.3	—	$V_{cc} \times 0.2$	V
		TMCIV, FTCI, FTIOA to FTIOD, SCK3, SCK3_2, SCK3_3*[1], TRGV		–0.3	—	$V_{cc} \times 0.1$	
		RXD, RXD_2, RXD_3*[1], P12 to P10, P17 to P14, P22 to P20,	V_{cc} = 4.0 V to 5.5 V	–0.3	—	$V_{cc} \times 0.3$	V
		P57 to P50, P76 to P70, P84 to P80 PB3 to PB0		–0.3	—	$V_{cc} \times 0.2$	
		OSC1	V_{cc} = 4.0 V to 5.5 V	–0.3	—	0.5	V
				–0.3	—	0.3	

		P84 to P80	V_{CC} = 4.0 V to 5.5 V I_{OL} = 20.0 mA	—	—	1.5	V
			V_{CC} = 4.0 V to 5.5 V I_{OL} = 10.0 mA	—	—	1.0	
			V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	—	—	0.4	
			I_{OL} = 0.4 mA	—	—	0.4	
Input/ output leakage current	$\| I_{IL} \|$	OSC1, $\overline{RES}$, $\overline{NMI}$ $\overline{WKP0}$, $\overline{WKP5}$, $\overline{IRQ0}$, $\overline{IRQ3}$, $\overline{ADTRG}$, TRGV, TMRIV, TMCIV, FTCI, FTIOA to FTIOD, RXD, RXD_2, RXD_3*[1], SCK3, SCK3_2, SCK3_3*[1]	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	—	—	1.0	μA
		P12 to P10, P17 to P14, P22 to P20, P57 to P50, P76 to P70, P84 to P80	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	—	—	1.0	μA
		PB3 to PB0	V_{IN} = 0.5 V to (AV_{CC} – 0.5 V)	—	—	1.0	μA
Pull-up MOS current	$–I_p$	P12 to P10, P17 to P14, P55 to P50	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	—	

	I_{OPE2}	V_{CC}	V_{CC} = 5.0 V, f_{OSC} = 20 MHz					
			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	—	11.5	22.5	mA	*
			Sleep mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	6.5	—		* 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
Standby mode current consump-tion	I_{STBY}	V_{CC}		—	—	5.0	μA	*

Active mode 2		Operates (ϕOSC/64)		ceramic or cry resonator
Sleep mode 1	V_{CC}	Only timers operate	V_{CC}	
Sleep mode 2		Only timers operate (ϕOSC/64)		
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	Main clock: ceramic or cry resonator

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

System clock (φ) cycle time	t_{cyc}			1	—	64	t_{OSC}
				—	—	12.8	μs
Instruction cycle time				2	—	—	t_{cyc}
Oscillation stabilization time (crystal resonator)	t_{rc}	OSC1, OSC2		—	—	10.0	ms
Oscillation stabilization time (ceramic resonator)	t_{rc}	OSC1, OSC2		—	—	5.0	ms
External clock high width	t_{CPH}	OSC1	V_{CC} = 4.0 V to 5.5 V	20.0	—	—	ns
				40.0	—	—	ns
External clock low width	t_{CPL}	OSC1	V_{CC} = 4.0 V to 5.5 V	20.0	—	—	ns
				40.0	—	—	ns
External clock rise time	t_{CPr}	OSC1	V_{CC} = 4.0 V to 5.5 V	—	—	10.0	ns
				—	—	15.0	ns
External clock fall time	t_{CPf}	OSC1	V_{CC} = 4.0 V to 5.5 V	—	—	10.0	ns
				—	—	15.0	ns

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

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

2. Determined by MA2 to MA0 in system control register 2 (SYSCR2).

width		SCK3_2, SCK3_3*						
Transmit data delay time (clocked synchronous)	t_{TXD}	TXD, TXD_2, TXD_3*	V_{CC} = 4.0 V to 5.5 V	—	—	1	t_{cyc}	F
				—	—	1	t_{cyc}	
Receive data setup time (clocked synchronous)	t_{RXS}	RXD, RXD_2, RXD_3*	V_{CC} = 4.0 V to 5.5 V	50.0	—	—	ns	
				100.0	—	—	ns	
Receive data hold time (clocked synchronous)	t_{RXH}	RXD, RXD_2, RXD_3*	V_{CC} = 4.0 V to 5.5 V	50.0	—	—	ns	
				100.0	—	—	ns	

Note: * The SCK3_3, RXD_3, and TXD_3 pins are not available in the H8/36014.

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}	AN3 to AN0		—	—	30.0	pF	
Allowable signal source impedance	R_{AIN}	AN3 to AN0		—	—	5.0	kΩ	
Resolution (data length)				10	10	10	bit	
Conversion time (single mode)			AV_{CC} = 3.3 V 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 V 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 and subsleep modes while the A/
converter is idle.

18.2.5 Watchdog Timer Characteristics

Table 18.6 Watchdog Timer Characteristics

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

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

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

Programming	Wait time after SWE bit setting*[1]	x		1	—	—
	Wait time after PSU bit setting*[1]	y		50	—	—
	Wait time after P bit setting *[1]*[4]	z1	$1 \leq n \leq 6$	28	30	32
		z2	$7 \leq n \leq 1000$	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 (The wait time after P bit setting (z1, z2) should be changed as follows accord value of the programming count (n).

   Programming count (n)

   $1 \leq n \leq 6$ z1 = 30 μs

   $7 \leq n \leq 1000$ z2 = 200 μs

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

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

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

Notes: 1. This voltage should be used when the falling and rising voltage detection funct used.

2. Select the low-voltage reset 2 when only the low-voltage detection reset is use

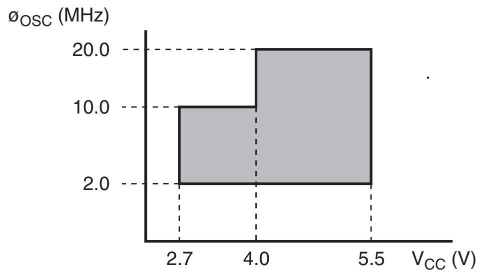
3. When the power-supply voltage (Vcc) falls below $V_{LVDRmin}$ = 1.0 V and then rises may not occur. Therefore sufficient evaluation is required.

charge of the 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

18.3 Electrical Characteristics (Masked ROM Version)

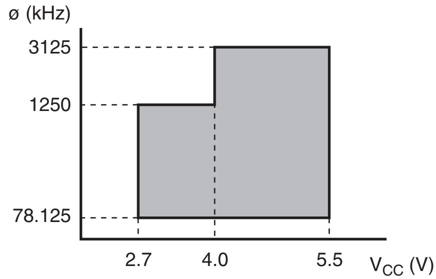
18.3.1 Power Supply Voltage and Operating Ranges

(1) Power Supply Voltage and Oscillation Frequency Range



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

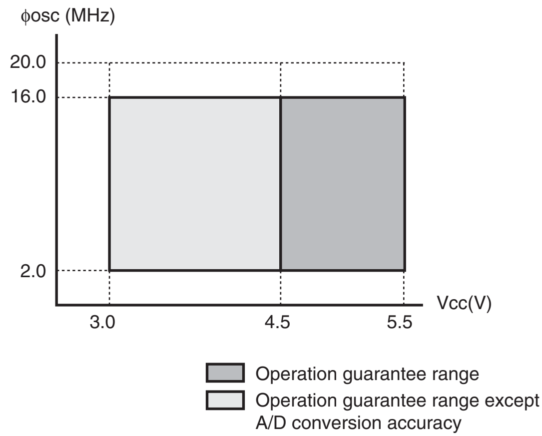
- AV_{CC} = 3.0 V to 5.5 V
- Active mode
- Sleep mode
  (When MA2 = 0 in SYSCR2)



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

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

(4) Range of Power Supply Voltage and Oscillation Frequency when Low-Voltag Detection Circuit is Used



		TMCIV, FTCI, FTIOA to FTIOD, SCK3, SCK3_2, SCK3_3*[1], TRGV		$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	
		RXD, RXD_2, RXD_3*[1], P12 to P10, P17 to P14, P22 to P20, P57 to P50, P76 to P70, P84 to P80	V_{CC} = 4.0 V to 5.5 V	$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V
				$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	
		PB3 to PB0	V_{CC} = 4.0 V 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 V 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 $\overline{WKP5}$, $\overline{IRQ0}$, $\overline{IRQ3}$, $\overline{ADTRG}$,TMRIV, TMCIV, FTCI, FTIOA to FTIOD, SCK3, SCK3_2, SCK3_3*[1], TRGV	V_{CC} = 4.0 V to 5.5 V	–0.3	—	$V_{CC} \times 0.2$	V
				–0.3	—	$V_{CC} \times 0.1$	
		RXD, RXD_2, RXD_3*[1], P12 to P10, P17 to P14, P22 to P20, P57 to P50, P76 to P70, P84 to P80 PB3 to PB0	V_{CC} = 4.0 V to 5.5 V	–0.3	—	$V_{CC} \times 0.3$	V
				–0.3	—	$V_{CC} \times 0.2$	
		OSC1	V_{CC} = 4.0 V to 5.5 V	–0.3	—	0.5	V
				–0.3	—	0.3	

		P84 to P80	V_{CC} = 4.0 V to 5.5 V I_{OL} = 20.0 mA	—	—	1.5	V
			V_{CC} = 4.0 V to 5.5 V I_{OL} = 10.0 mA	—	—	1.0	
			V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	—	—	0.4	
			I_{OL} = 0.4 mA	—	—	0.4	
Input/ output leakage current	$\| I_{IL} \|$	OSC1, $\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$, $\overline{IRQ3}$, $\overline{ADTRG}$, TRGV, TMRIV, TMCIV, FTCI, FTIOA to FTIOD, RXD, RXD_2, RXD_3*[1], SCK3, SCK3_2, SCK3_3*[1]	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	—	—	1.0	μA
		P12 to P10, P17 to P14, P22 to P20, P57 to P50, P76 to P70, P84 to P80	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	—	—	1.0	μA
		PB3 to PB0	V_{IN} = 0.5 V to (AV_{CC} – 0.5 V)	—	—	1.0	μA

consump- tion			f_{OSC}					
			Active mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	8.0	—		* 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	—	11.5	22.5	mA	*
			Sleep mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	6.5	—		* 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
Standby mode current consump- tion	I_{STBY}	V_{CC}		—	—	5.0	μA	*

Active mode 2		Operates (φOSC/64)		ceramic or cry resonator
Sleep mode 1	V_{CC}	Only timers operate	V_{CC}	
Sleep mode 2		Only timers operate (φOSC/64)		
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	Main clock: ceramic or cry resonator

		Port 8		—	—	10.0
Allowable output low current (total)	$\sum I_{OL}$	Output pins except port 8	V_{CC} = 4.0 V to 5.5 V	—	—	40.0
		Port 8		—	—	80.0
		Output pins except port 8		—	—	20.0
		Port 8		—	—	40.0
Allowable output high current (per pin)	$\lvert -I_{OH} \rvert$	All output pins	V_{CC} = 4.0 V 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 V to 5.5 V	—	—	30.0
				—	—	8.0

System clock (φ) cycle time	t_{cyc}			1		64	t_{OSC}
				—	—	12.8	μs
Instruction cycle time				2	—	—	t_{cyc}
Oscillation stabilization time (crystal resonator)	t_{rc}	OSC1, OSC2		—	—	10.0	ms
Oscillation stabilization time (ceramic resonator)	t_{rc}	OSC1, OSC2		—	—	5.0	ms
External clock high width	t_{CPH}	OSC1	V_{CC} = 4.0 V to 5.5 V	20.0	—	—	ns
				40.0	—	—	ns
External clock low width	t_{CPL}	OSC1	V_{CC} = 4.0 V to 5.5 V	20.0	—	—	ns
				40.0	—	—	ns
External clock rise time	t_{CPr}	OSC1	V_{CC} = 4.0 V to 5.5 V	—	—	10.0	ns
				—	—	15.0	ns
External clock fall time	t_{CPf}	OSC1	V_{CC} = 4.0 V to 5.5 V	—	—	10.0	ns
				—	—	15.0	ns
$\overline{RES}$ pin low width	t_{REL}	$\overline{RES}$	At power-on and in modes other than those below	t_{rc}	—	—	ms
			In active mode and sleep mode operation	200	—	—	ns

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

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

2. Determined by the MA2 to MA0 bits in the system control register 2 (SYSCR2)

width	t_{SCKW}	SCK3_2, SCK3_3*					t_{Scyc}
Transmit data delay time (clocked synchronous)	t_{TXD}	TXD, TXD_2, TXD_3*	V_{CC} = 4.0 V to 5.5 V	—	—	1	t_{cyc}
				—	—	1	t_{cyc}
Receive data setup time (clocked synchronous)	t_{RXS}	RXD, RXD_2, RXD_3*	V_{CC} = 4.0 V to 5.5 V	50.0	—	—	ns
				100.0	—	—	ns
Receive data hold time (clocked synchronous)	t_{RXH}	RXD, RXD_2, RXD_3*	V_{CC} = 4.0 V to 5.5 V	50.0	—	—	ns
				100.0	—	—	ns

Note: * The SCK3_3, RXD_3, and TXD_3 pins are not available in the H8/36014.

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}	AN3 to AN0		—	—	30.0	pF	
Allowable signal source impedance	R_{AIN}	AN3 to AN0		—	—	5.0	kΩ	
Resolution (data length)				10	10	10	bit	
Conversion time (single mode)			AV_{CC} = 3.0 V 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 V 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 and subsleep modes while the A/
converter is idle.

18.3.5 Watchdog Timer Characteristics

Table 18.14 Watchdog Timer Characteristics

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

Item	Symbol	Applicable Pins	Test Condition	Values Min	Values Typ	Values Max	Unit
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 generate
the internal oscillator is selected.

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

Notes: 1. This voltage should be used when the falling and rising voltage detection funct used.

2. Select the low-voltage reset 2 when only the low-voltage detection reset is use

3. When the power-supply voltage (Vcc) falls below $V_{LVDRmin}$ = 1.0 V and then rises may not occur. Therefore sufficient evaluation is required.

charge of the 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

18.4 Operation Timing

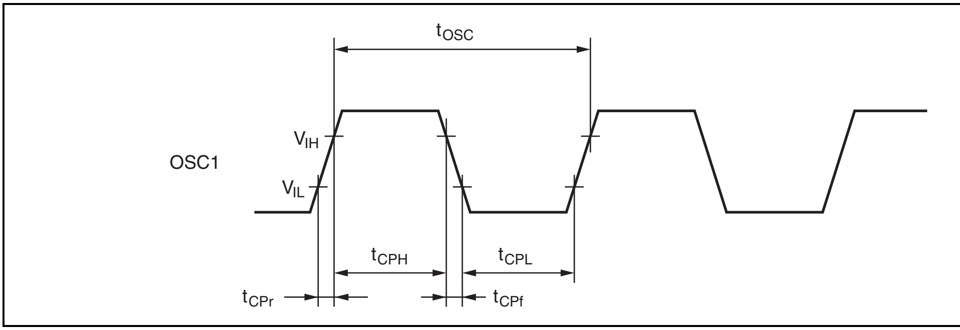


Figure 18.1 System Clock Input Timing

Figure 18.2 $\overline{RES}$ Low Width Timing

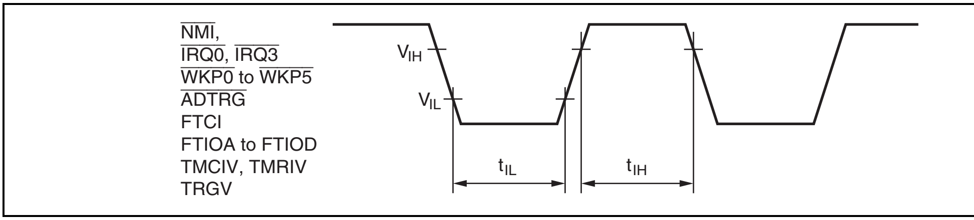


Figure 18.3 Input Timing

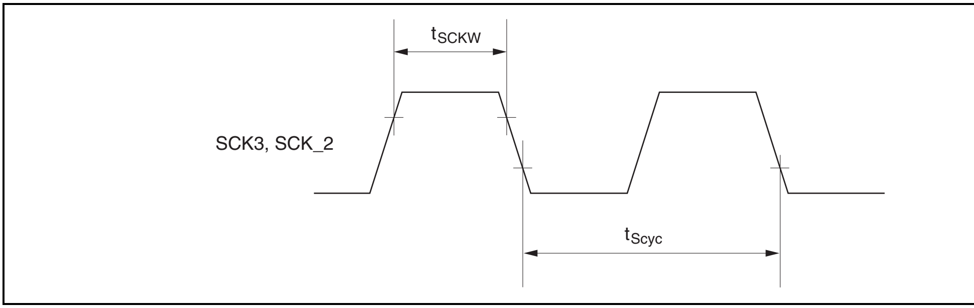
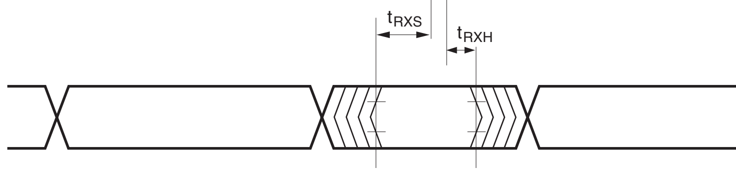


Figure 18.4 SCK3 Input Clock Timing



Notes: 1. The SCK3_3, RXD_3, and TXD_3 pins are not available in the H8/36014.
2. Output timing reference levels
Output high: V_{OH} = 2.0 V
Output low: V_{OL} = 0.8 V
Load conditions are shown in figure 18.6.

Figure 18.5 SCI3 Input/Output Timing in Clocked Synchronous Mode

18.5 Output Load Condition

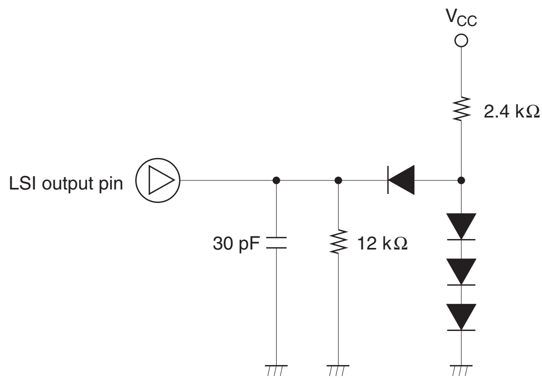


Figure 18.6 Output Load Circuit

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 transi 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)

0	Cleared to 0
1	Set to 1
—	Not affected by execution of the instruction
Δ	Varies depending on conditions, described in notes

	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

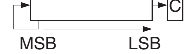
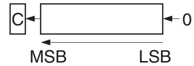
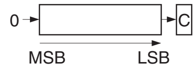
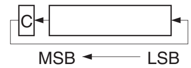
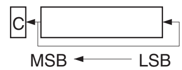
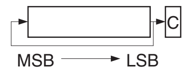
	MOV.L ERs, ERd	L		2								ERs32 → ERd32	—	—	↕	↕	0	—
	MOV.L @ERs, ERd	L			4							@ERs → ERd32	—	—	↕	↕	0	—
	MOV.L @(d:16, ERs), ERd	L				6						@(d:16, ERs) → ERd32	—	—	↕	↕	0	—
	MOV.L @(d:24, ERs), ERd	L				10						@(d:24, ERs) → ERd32	—	—	↕	↕	0	—
	MOV.L @ERs+, ERd	L					4					@ERs → ERd32 ERs32+4 → 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.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)	↕	↕	↕

	EXTS.L ERd	L	0 → (<bits 31 to 16> of ERd32)		2										0	↕	0
EXTS	EXTS.W Rd	W	(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)		2								—	—	↕	↕	0
	EXTS.L ERd	L	(<bit 15> of ERd32) → (<bits 31 to 16> of ERd32)		2								—	—	↕	↕	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.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, @ERd	B			4							(Rn8 of @ERd) ← 1	—	—	—	—	—	—
	BSET Rn, @aa:8	B						4				(Rn8 of @aa:8) ← 1	—	—	—	—	—	—
BCLR	BCLR #xx:3, Rd	B		2								(#xx:3 of Rd8) ← 0	—	—	—	—	—	—
	BCLR #xx:3, @ERd	B			4							(#xx:3 of @ERd) ← 0	—	—	—	—	—	—
	BCLR #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) ← 0	—	—	—	—	—	—
	BCLR Rn, Rd	B		2								(Rn8 of Rd8) ← 0	—	—	—	—	—	—
	BCLR Rn, @ERd	B			4							(Rn8 of @ERd) ← 0	—	—	—	—	—	—
	BCLR Rn, @aa:8	B						4				(Rn8 of @aa:8) ← 0	—	—	—	—	—	—
BNOT	BNOT #xx:3, Rd	B		2								(#xx:3 of Rd8) ← ¬ (#xx:3 of Rd8)	—	—	—	—	—	—
	BNOT #xx:3, @ERd	B			4							(#xx:3 of @ERd) ← ¬ (#xx:3 of @ERd)	—	—	—	—	—	—
	BNOT #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) ← ¬ (#xx:3 of @aa:8)	—	—	—	—	—	—
	BNOT Rn, Rd	B		2								(Rn8 of Rd8) ← ¬ (Rn8 of Rd8)	—	—	—	—	—	—
	BNOT Rn, @ERd	B			4							(Rn8 of @ERd) ← ¬ (Rn8 of @ERd)	—	—	—	—	—	—
	BNOT Rn, @aa:8	B						4				(Rn8 of @aa:8) ← ¬ (Rn8 of @aa:8)	—	—	—	—	—	—
BTST	BTST #xx:3, Rd	B		2								¬ (#xx:3 of Rd8) → Z	—	—	—	↕	—	—
	BTST #xx:3, @ERd	B			4							¬ (#xx:3 of @ERd) → Z	—	—	—	↕	—	—
	BTST #xx:3, @aa:8	B						4				¬ (#xx:3 of @aa:8) → Z	—	—	—	↕	—	—
	BTST Rn, Rd	B		2								¬ (Rn8 of @Rd8) → Z	—	—	—	↕	—	—
	BTST Rn, @ERd	B			4							¬ (Rn8 of @ERd) → Z	—	—	—	↕	—	—
	BTST Rn, @aa:8	B						4				¬ (Rn8 of @aa:8) → Z	—	—	—	↕	—	—
BLD	BLD #xx:3, Rd	B		2								(#xx:3 of Rd8) → C	—	—	—	—	—	

BST	BST #xx:3, Rd	B		2								C → (#xx:3 of Rd8)					
	BST #xx:3, @ERd	B			4							C → (#xx:3 of @ERd24)	—	—	—	—	—
BIST	BST #xx:3, @aa:8	B						4				C → (#xx:3 of @aa:8)	—	—	—	—	—
	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	—	—	—	—	—
BIAND	BAND #xx:3, @aa:8	B						4				C∧(#xx:3 of @aa:8) → C	—	—	—	—	—
	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	—	—	—	—	—

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

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

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

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

Notes: 1. The number of states in cases where the instruction code and its operands ar in on-chip memory is shown here. For other cases see Appendix A.3, Numbe Execution States.

2. n is the value set in register R4L or R4.

(1) Set to 1 when a carry or borrow occurs at bit 11; otherwise cleared to 0.

(2) Set to 1 when a carry or borrow occurs at bit 27; otherwise cleared to 0.

(3) Retains its previous value when the result is zero; otherwise cleared to 0.

(4) Set to 1 when the adjustment produces a carry; otherwise retains its prev

(5) The number of states required for execution of an instruction that transfer synchronization with the E clock is variable.

(6) Set to 1 when the divisor is negative; otherwise cleared to 0.

(7) Set to 1 when the divisor is zero; otherwise cleared to 0.

(8) Set to 1 when the quotient is negative; otherwise cleared to 0.

Instruction code:

1st byte		2nd byte	
AH	AL	BH	BL

(Box with diagonal line; upper left: Instruction when most significant bit of BH is; lower right: Instruction when most significant bit of BH is)

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 m
Instruction when m

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 17.1, 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.

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

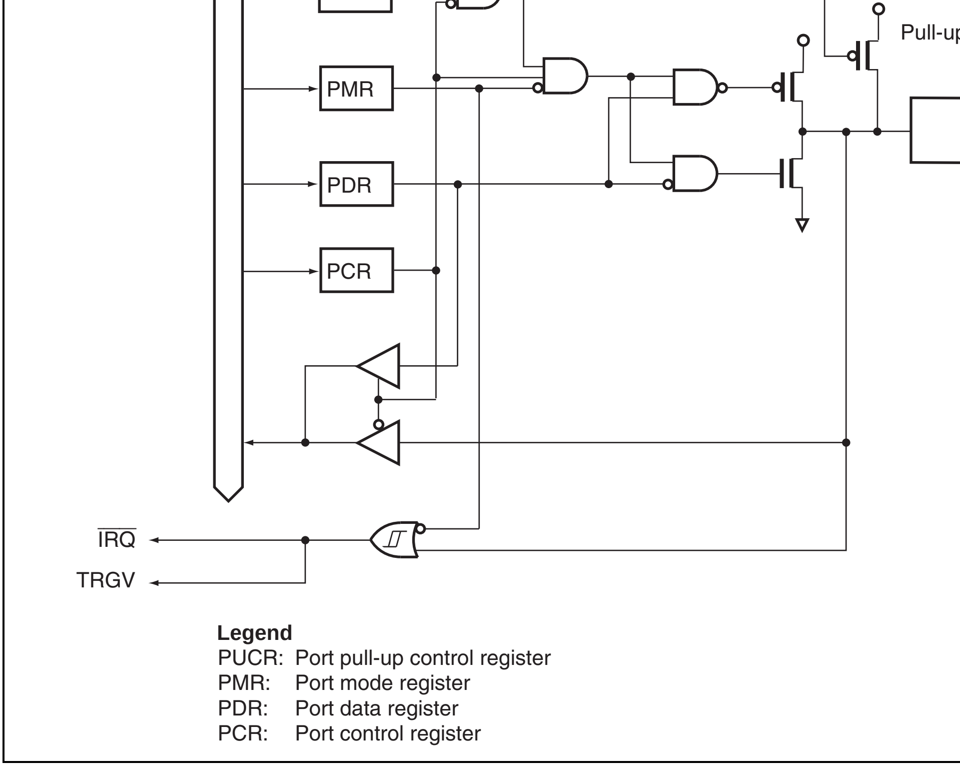
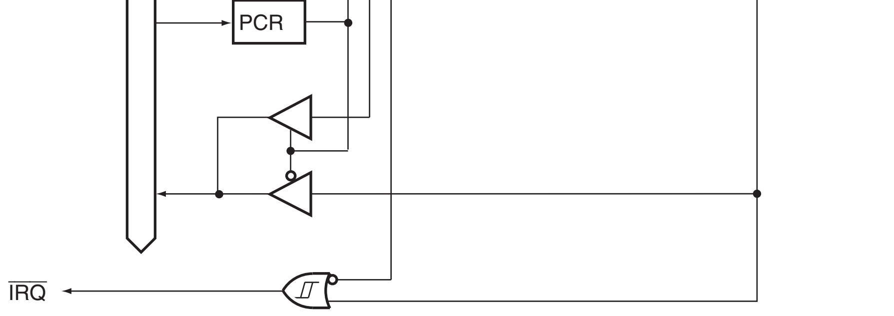


Figure B.1 Port 1 Block Diagram (P17)



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)

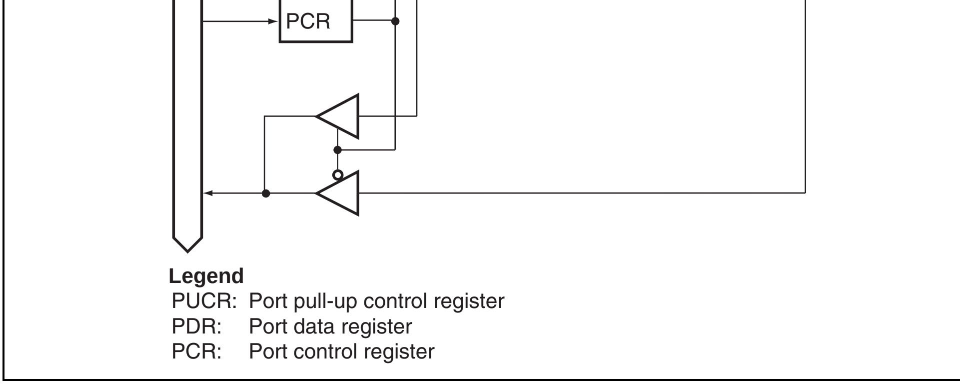


Figure B.3 Port 1 Block Diagram (P16, P15, P12*, P10)

Note: * This pin is available only in the H8/36014.

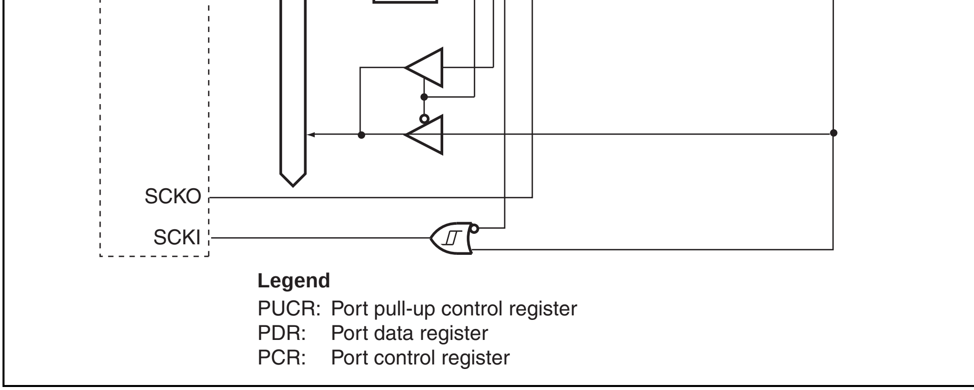
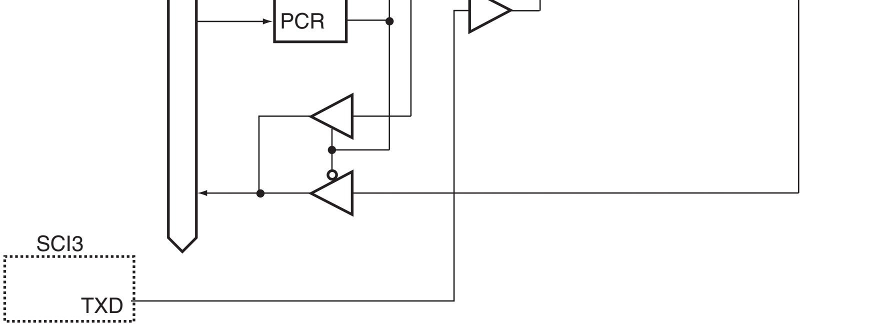


Figure B.4 Port 1 Block Diagram (P12) (H8/36024)

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

Figure B.5 Port 1 Block Diagram (P11)



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

Figure B.6 Port 2 Block Diagram (P22)

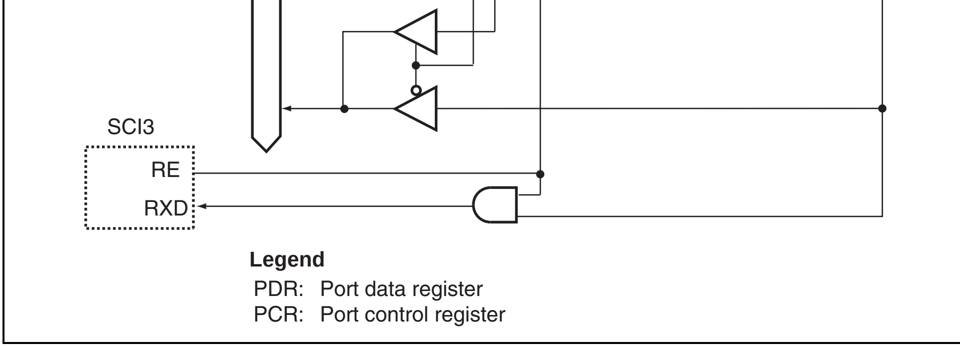


Figure B.7 Port 2 Block Diagram (P21)

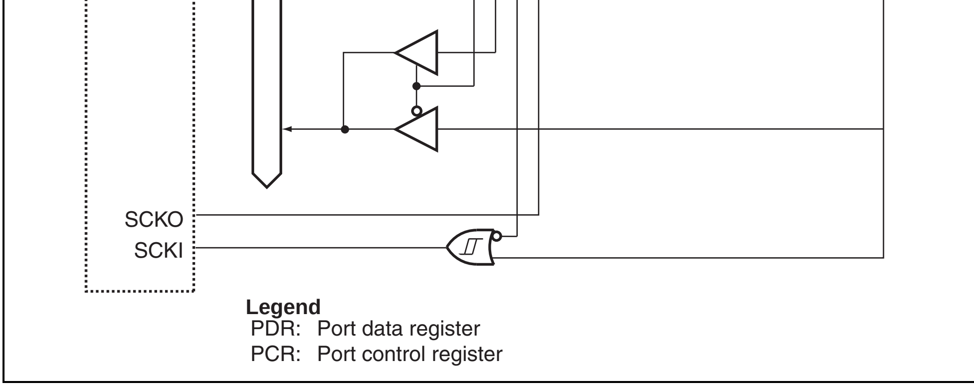


Figure B.8 Port 2 Block Diagram (P20)

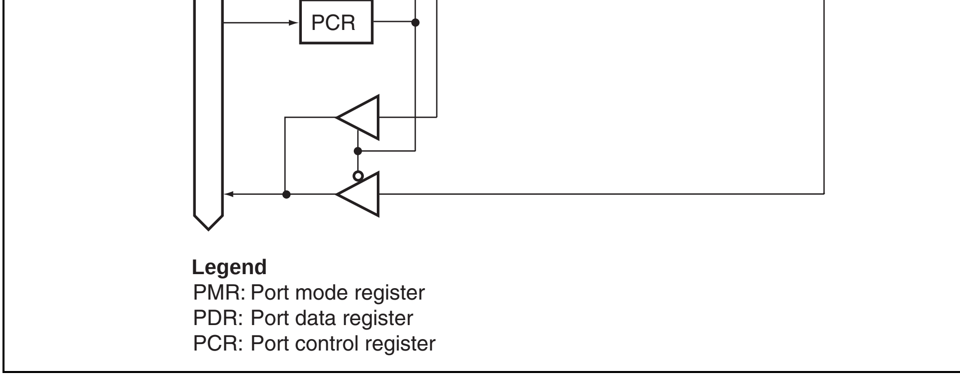


Figure B.9 Port 5 Block Diagram (P57, P56) (H8/36014)

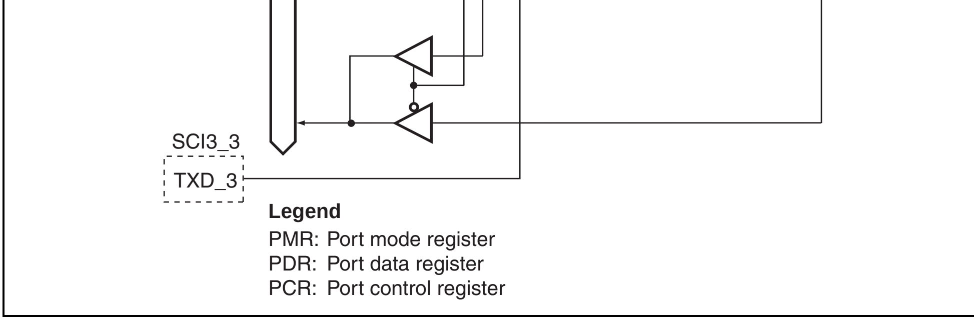


Figure B.10 Port 5 Block Diagram (P57) (H8/36024)

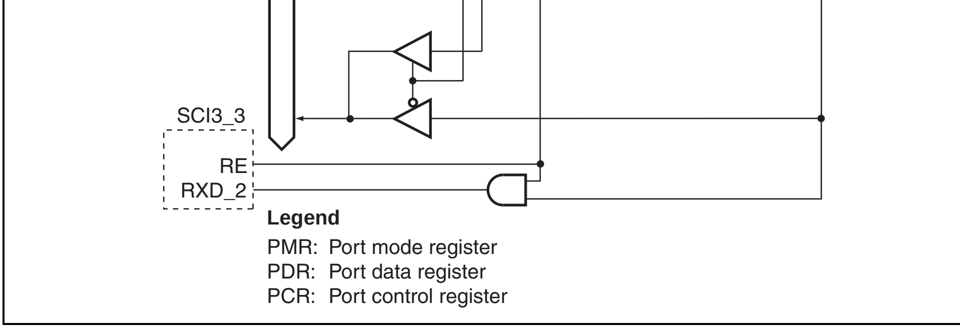


Figure B.11 Port 5 Block Diagram (P56) (H8/36024)

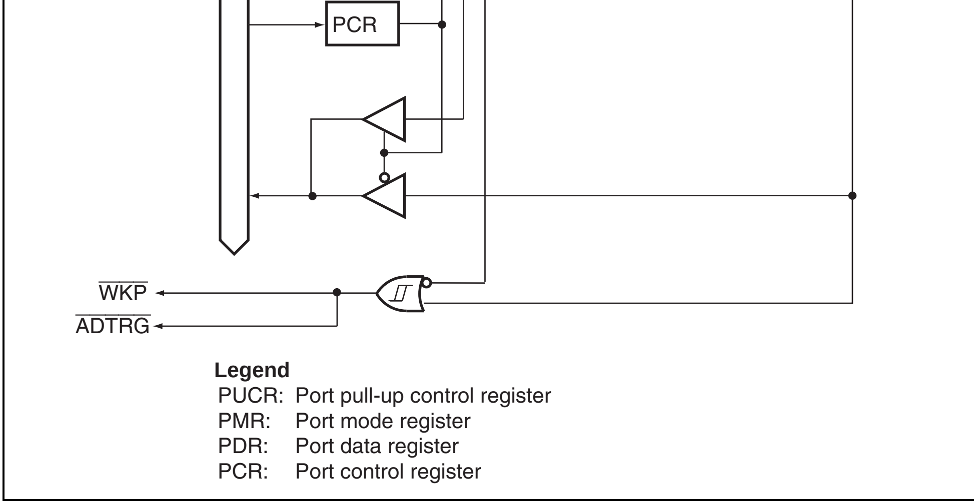


Figure B.12 Port 5 Block Diagram (P55)

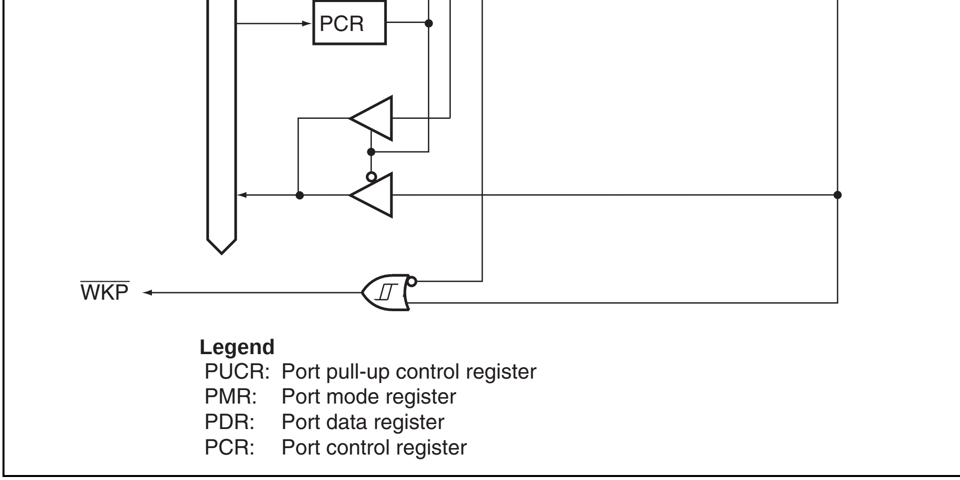


Figure B.13 Port 5 Block Diagram (P54 to P50)

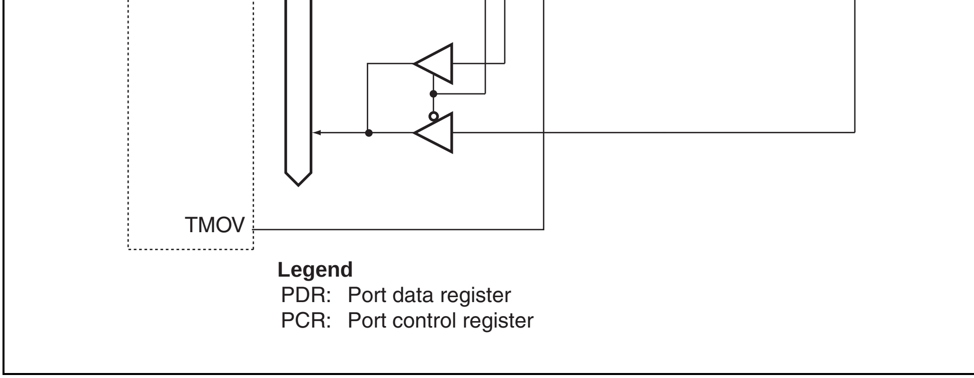


Figure B.14 Port 7 Block Diagram (P76)

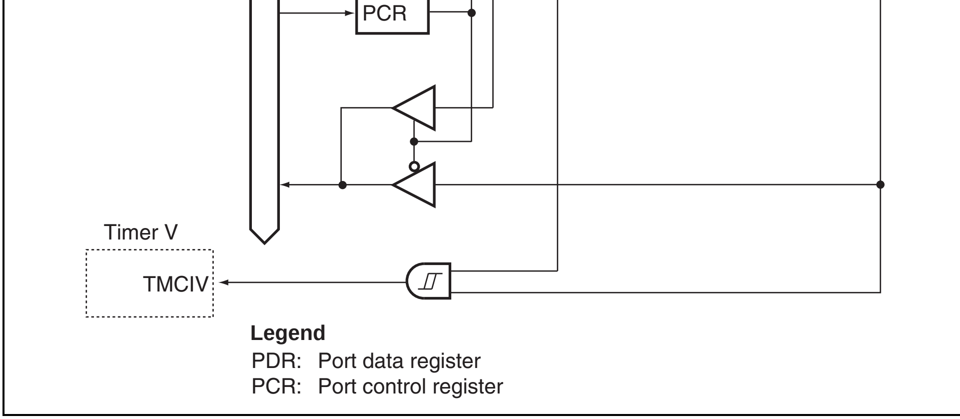


Figure B.15 Port 7 Block Diagram (P75)

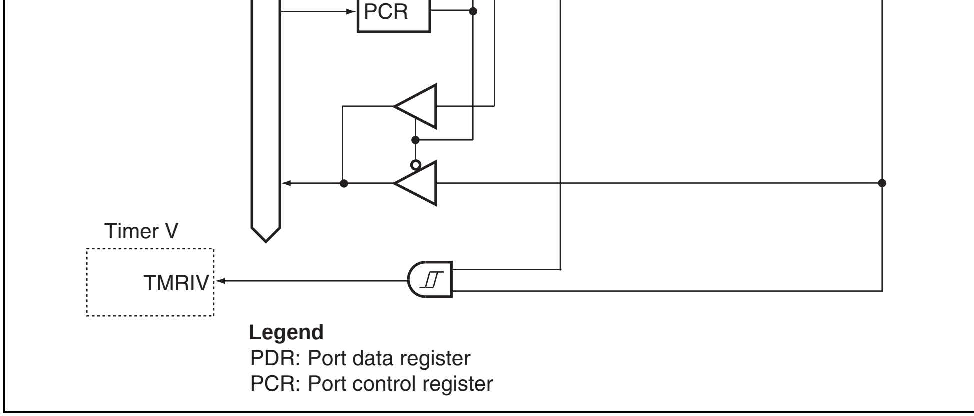
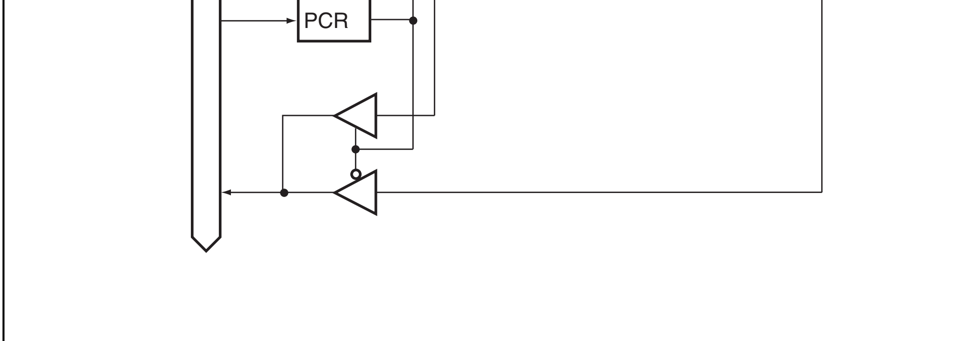


Figure B.16 Port 7 Block Diagram (P74)



Legend
PDR: Port data register
PCR: Port control register

Figure B.17 Port 7 Block Diagram (P73)

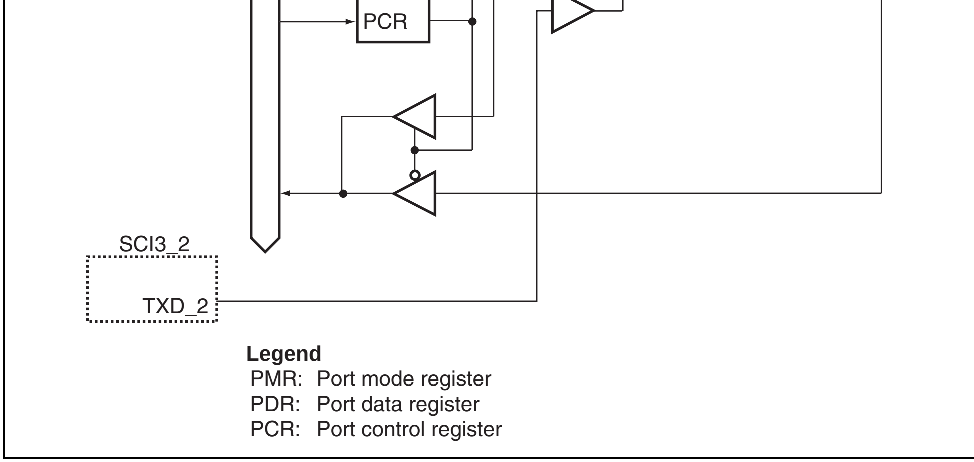


Figure B.18 Port 7 Block Diagram (P72)

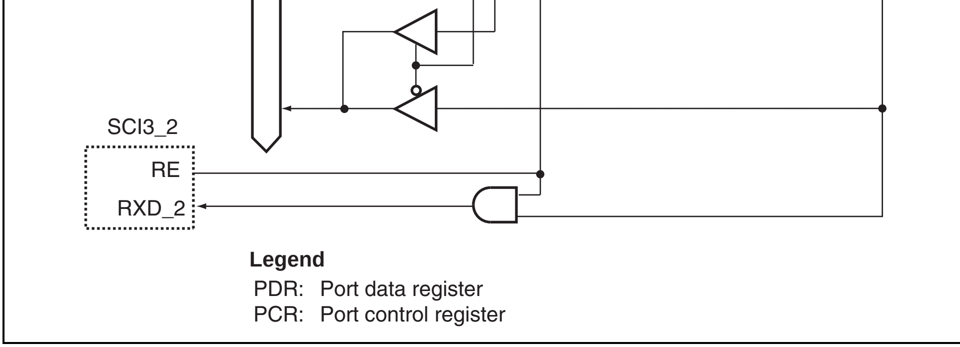


Figure B.19 Port 7 Block Diagram (P71)

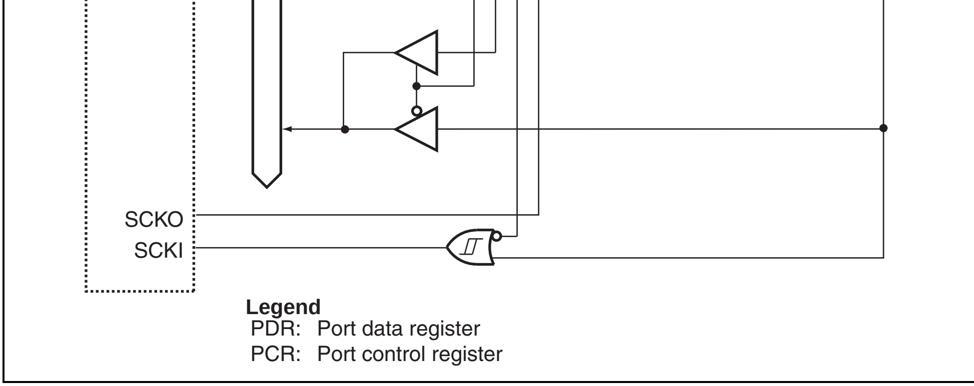
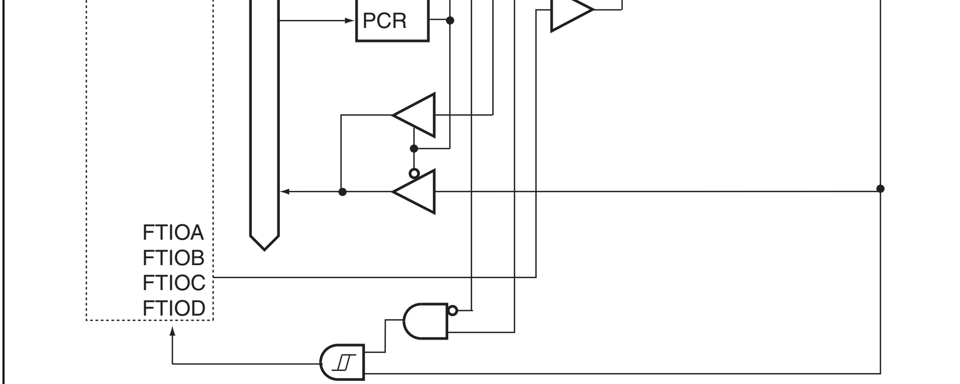


Figure B.20 Port 7 Block Diagram (P70)



Legend
PDR: Port data register
PCR: Port control register

Figure B.21 Port 8 Block Diagram (P84 to P81)

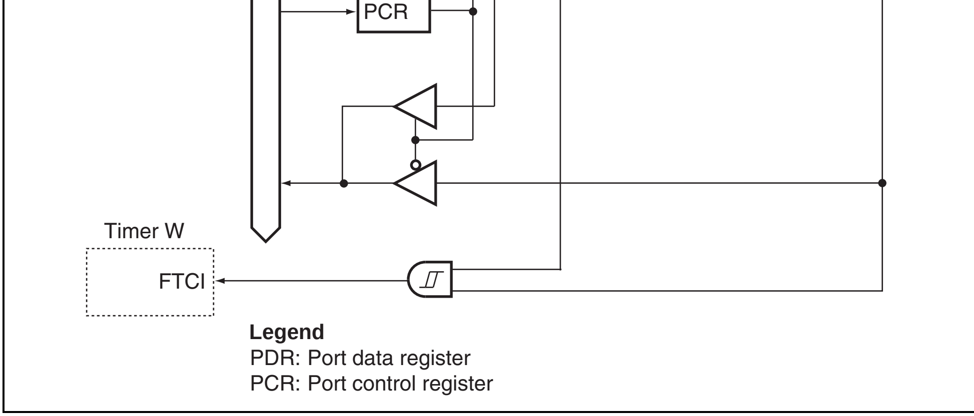


Figure B.22 Port 8 Block Diagram (P80)

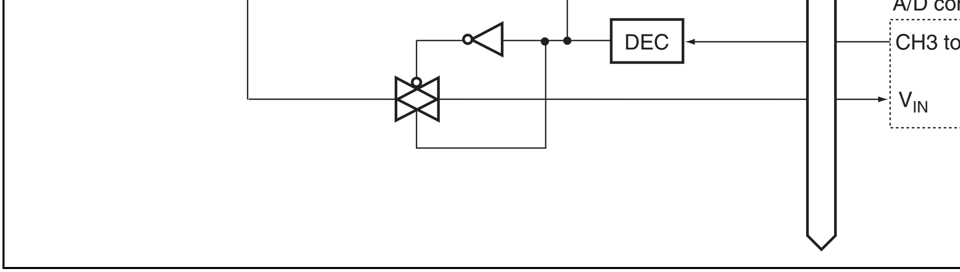


Figure B.23 Port B Block Diagram (PB3 to PB0)

B.2 Port States in Each Operating State

Port	Reset	Active	Sleep	Subsleep	Standby
P17 to P14, P12 to P10	High impedance	Functioning	Retained	Retained	High impe
P22 to P20	High impedance	Functioning	Retained	Retained	High impe
P57 to P50	High impedance	Functioning	Retained	Retained	High impe
P76 to P70	High impedance	Functioning	Retained	Retained	High impe
P84 to P80	High impedance	Functioning	Retained	Retained	High impe
PB3 to PB0	High impedance	High impedance	High impedance	Retained	High impe

Note: * High level output when the pull-up MOS is in on state.

		& LVDC	HD64F36024GFY	HD64F36024GFY	LQFP-48
			HD64F36024GFT	HD64F36024GFT	QFN-48(T
	Masked ROM version	Standard product	HD64336024FP	HD64336024(***)FP	LQFP-64
			HD64336024FX	HD64336024(***)FX	LQFP-48
			HD64336024FY	HD64336024(***)FY	LQFP-48
			HD64336024FT	HD64336024(***)FT	QFN-48(T
		Product with POR & LVDC	HD64336024GFP	HD64336024G(***)FP	LQFP-64
			HD64336024GFX	HD64336024G(***)FX	LQFP-48
			HD64336024GFY	HD64336024G(***)FY	LQFP-48
			HD64336024GFT	HD64336024G(***)FT	QFN-48(T
H8/36023	Masked ROM version	Standard product	HD64336023FP	HD64336023(***)FP	LQFP-64
			HD64336023FX	HD64336023(***)FX	LQFP-48
			HD64336023FY	HD64336023(***)FY	LQFP-48
			HD64336023FT	HD64336023(***)FT	QFN-48(T
		Product with POR & LVDC	HD64336023GFP	HD64336023G(***)FP	LQFP-64
			HD64336023GFX	HD64336023G(***)FX	LQFP-48
			HD64336023GFY	HD64336023G(***)FY	LQFP-48
			HD64336023GFT	HD64336023G(***)FT	QFN-48(T

	Masked ROM version	Standard product	HD64336022FP	HD64336022(***)FP	LQFP-64
			HD64336022FX	HD64336022(***)FX	LQFP-48
			HD64336022FY	HD64336022(***)FY	LQFP-48
			HD64336022FT	HD64336022(***)FT	QFN-48(
		Product with POR & LVDC	HD64336022GFP	HD64336022G(***)FP	LQFP-64
			HD64336022GFX	HD64336022G(***)FX	LQFP-48
			HD64336022GFY	HD64336022G(***)FY	LQFP-48
			HD64336022GFT	HD64336022G(***)FT	QFN-48(
H8/36014	Flash memory version	Standard product	HD64F36014FP	HD64F36014FP	LQFP-64
			HD64F36014FX	HD64F36014FX	LQFP-48
			HD64F36014FY	HD64F36014FY	LQFP-48
			HD64F36014FT	HD64F36014FT	QFN-48(
		Product with POR & LVDC	HD64F36014GFP	HD64F36014GFP	LQFP-64
			HD64F36014GFX	HD64F36014GFX	LQFP-48
			HD64F36014GFY	HD64F36014GFY	LQFP-48
			HD64F36014GFT	HD64F36014GFT	QFN-48(
	Masked ROM version	Standard product	HD64336014FP	HD64336014(***)FP	LQFP-64
			HD64336014FX	HD64336014(***)FX	LQFP-48
			HD64336014FY	HD64336014(***)FY	LQFP-48
			HD64336014FT	HD64336014(***)FT	QFN-48(
		Product with POR & LVDC	HD64336014GFP	HD64336014G(***)FP	LQFP-64
			HD64336014GFX	HD64336014G(***)FX	LQFP-48
			HD64336014GFY	HD64336014G(***)FY	LQFP-48
			HD64336014GFT	HD64336014G(***)FT	QFN-48(

H8/36012	Flash memory version	Standard product	HD64F36012FP	HD64F36012FP	LQFP-64
			HD64F36012FX	HD64F36012FX	LQFP-48
			HD64F36012FY	HD64F36012FY	LQFP-48
			HD64F36012FT	HD64F36012FT	QFN-48(T
		Product with POR & LVDC	HD64F36012GFP	HD64F36012GFP	LQFP-64
			HD64F36012GFX	HD64F36012GFX	LQFP-48
			HD64F36012GFY	HD64F36012GFY	LQFP-48
			HD64F36012GFT	HD64F36012GFT	QFN-48(T
	Masked ROM version	Standard product	HD64336012FP	HD64336012(***)FP	LQFP-64
			HD64336012FX	HD64336012(***)FX	LQFP-48
			HD64336012FY	HD64336012(***)FY	LQFP-48
			HD64336012FT	HD64336012(***)FT	QFN-48(T
		Product with POR & LVDC	HD64336012GFP	HD64336012G(***)FP	LQFP-64
			HD64336012GFX	HD64336012G(***)FX	LQFP-48
			HD64336012GFY	HD64336012G(***)FY	LQFP-48
			HD64336012GFT	HD64336012G(***)FT	QFN-48(T
H8/36011	Masked ROM version	Standard product	HD64336011FP	HD64336011(***)FP	LQFP-64
			HD64336011FX	HD64336011(***)FX	LQFP-48
			HD64336011FY	HD64336011(***)FY	LQFP-48
			HD64336011FT	HD64336011(***)FT	QFN-48(T
		Product with POR & LVDC	HD64336011GFP	HD64336011G(***)FP	LQFP-64
			HD64336011GFX	HD64336011G(***)FX	LQFP-48
			HD64336011GFY	HD64336011G(***)FY	LQFP-48
			HD64336011GFT	HD64336011G(***)FT	QFN-48(T

Legend

POR & LVDC: Power-on reset and low-voltage detection circuits

(***): ROM code

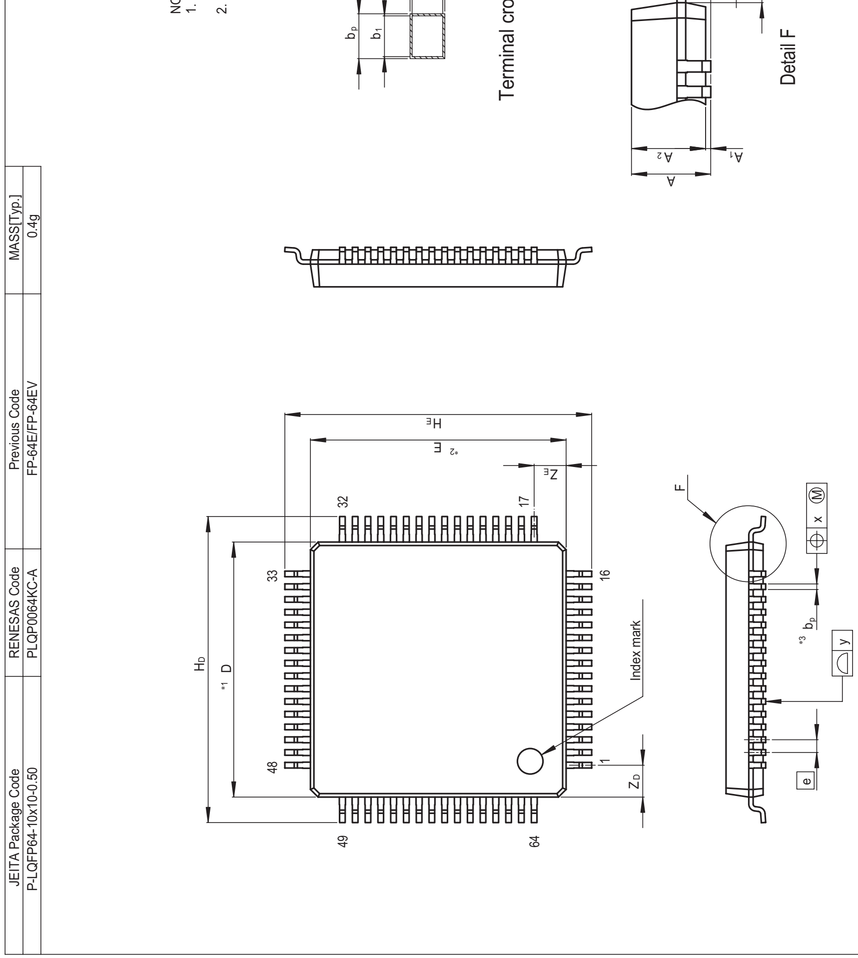


Figure D.1 FP-64E Package Dimensions

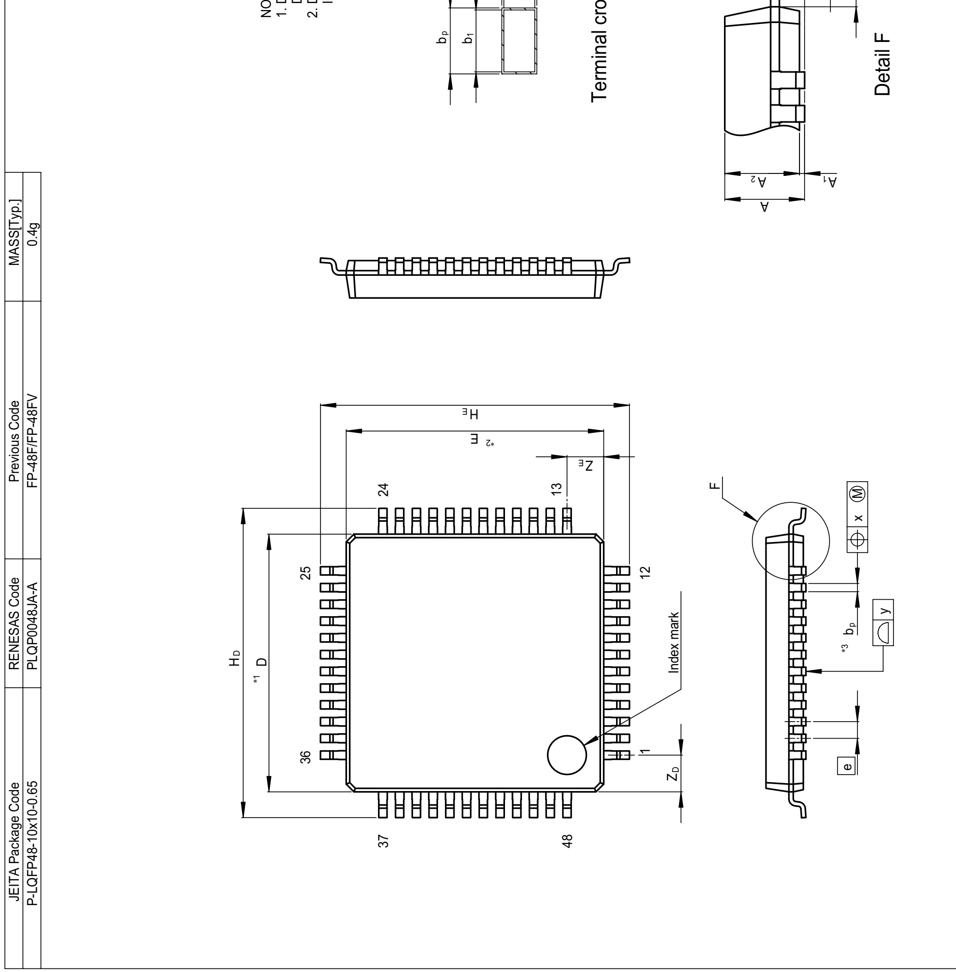


Figure D.2 FP-48F Package Dimensions

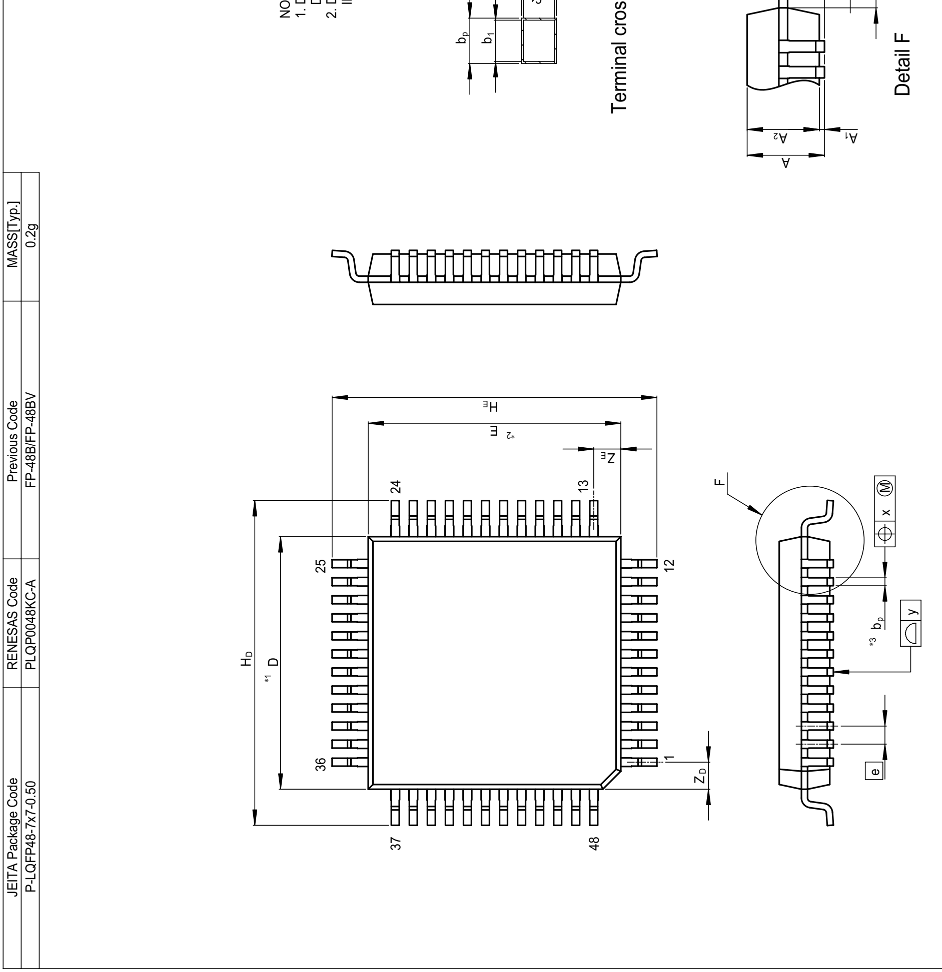


Figure D.3 FP-48B Package Dimensions

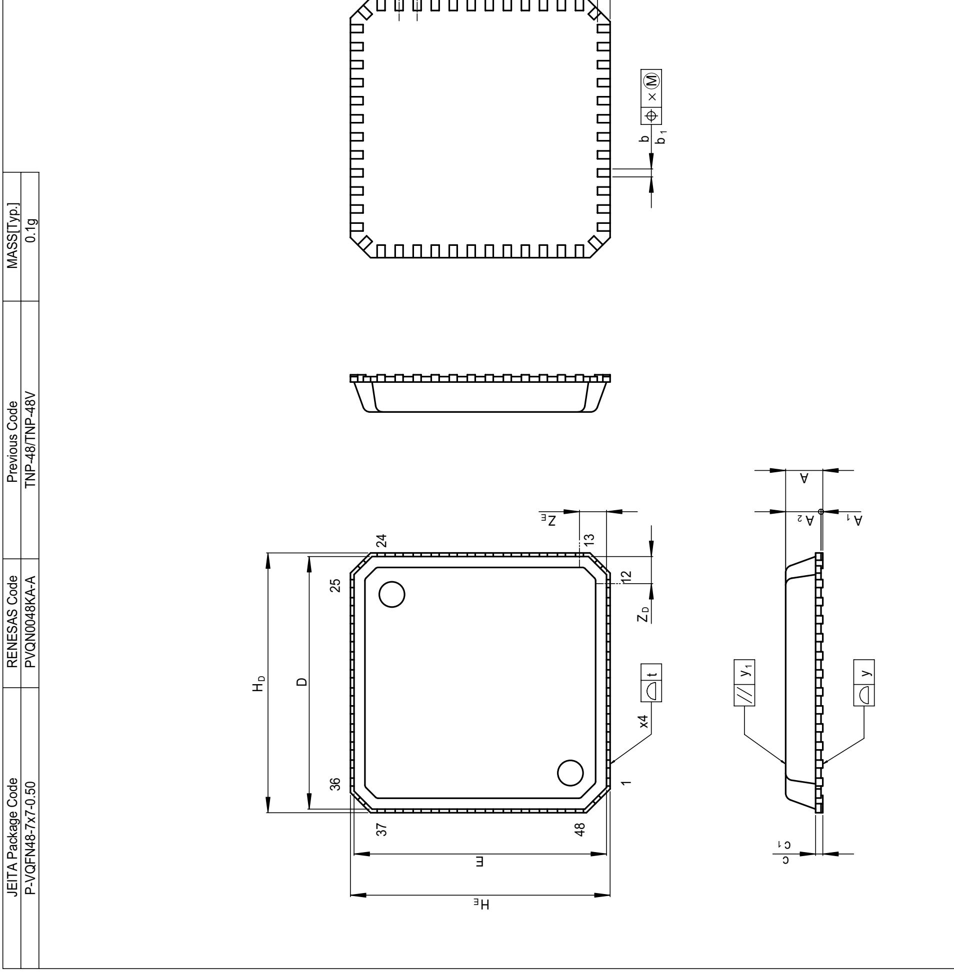


Figure D.4   TNP-48 Package Dimensions

<table>
<tr><td></td><td></td><td>4. When the E7 or E8 is used, address breaks can
either available to the user or for use by the E7 c
address breaks are set as being used by the E7
address break control registers must not be acce
5. When the E7 or E8 is used, $\overline{\text{NMI}}$ is an input/outp
(open-drain in output mode).
6. Use channel 1 of the SCI3 (P21/RXD, P22/TXD)
board programming mode by boot mode.
Note has been deleted.</td></tr>
<tr><td>Section 1 Overview
1.2 Internal Block Diagram
Figure 1.1 Internal Block Diagram</td><td>3</td><td>3 Can also be used for the E7 or E8 emulator.</td></tr>
<tr><td>Figure 1.2 Pin Arrangement (FP-64E)</td><td>4</td><td>2 Can also be used for the E7 or E8 emulator.</td></tr>
<tr><td>Figure 1.3 Pin Arrangement (FP-48F, FP-48B, TNP-48)</td><td>5</td><td>2 Can also be used for the E7 or E8 emulator.</td></tr>
<tr><td>Table 1.1 Pin Functions</td><td>7</td><td><table><tr><th>Type</th><th>Functions</th></tr><tr><td>E10T</td><td>Interface pin for the E10T, E8, or E7 emul</td></tr></table></td></tr>
<tr><td>Section 7 ROM</td><td>77</td><td>The features of the 32-kbyte (4 kbytes of them are t
program area for E7 or E8) flash memory built into t
HD64F36024 and HD64F36014 are summarized be</td></tr>
<tr><td>Section 8 RAM</td><td>93</td><td>Note: When the E7 or E8 is used, area H'F780 to H'
must not be accessed.</td></tr>
</table>

Characteristics

Table 18.2 DC Characteristics (1)

Item	Symbol	Applicable Pins	Test Condition	V
				M
Input high voltage	V_{IH}	PB3 to PB0	V_{CC} = 4.0 V to 5.5 V	V
				V
Input low voltage	V_{IL}	RXD, RXD_2, RXD_3*[1], P12 to P10, P17 to P14, : PB3 to PB0	V_{CC} = 4.0 V to 5.5 V	–(

Table 18.2 DC Characteristics (1) 257

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

PB3 to PB0

Table 18.10 DC Characteristics (1)	273	(see nested table below)
Appendix D Package Dimensions Figure D.1 FP-64E Package Dimensions	343	Swapped with new one.
Figure D.2 FP-48F Package Dimensions	344	Swapped with new one.
Figure D.3 FP-48B Package Dimensions	345	Swapped with new one.
Figure D.4 TNP-48 Package Dimensions	346	Swapped with new one.

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

C

E

G

I

M

O

P

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8/36024 Group, H8/36014 Group

Publication Date:	1st Edition, Mar., 2001
	Rev.4.00, Sep. 23, 2005
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Customer Support Department
	Global Strategic Communication Div.
	Renesas Solutions Corp.

RENESAS SALES OFFICES

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 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 (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 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

H8/36024Group, H8/36014Group
Hardware Manual

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

REJ09B0025-0400