To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User's Manual

16

H8/3672 Group

Hardware Manual

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

H8/3672F HD64F3672
H8/3670F HD64F3670

Renesas Electronics
www.renesas.com

Rev.4.00 2005.11

a third party.

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

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

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

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

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

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

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

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

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

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

11. Index

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8/3672 Group to the target users.
Refer to the H8/300H Series Software Manual for a detailed description o instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized in on the CPU, system control functions, peripheral functions and electrical characteristi
- In order to understand the details of the CPU's functions
  Read the H8/300H Series Software Manual.
- In order to understand the details of a register when its name is known
  Read the index that is the final part of the manual to find the page number of the entry register. The addresses, bits, and initial values of the registers are summarized in secti List of Registers.
  Example: Bit order: The MSB is on the left and the LSB is on the right.

Notes:

When using the on-chip emulator (E7, E8) for H8/3672 program development and debug following restrictions must be noted.

1. The $\overline{\text{NMI}}$ pin is reserved for the E7 or E8, and cannot be used.
2. Area H'4000 to H'4FFF 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.
4. When the E7 or E8 is used, address breaks can be set as either available to the user or by the E7 or E8. If address breaks are set as being used by the E7 or E8, the address b control registers must not be accessed.

User's manuals for development tools:

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

Application notes:

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

Section 3 Exception Handling

Section 4 Address Break

Appendix D Package Dimensions

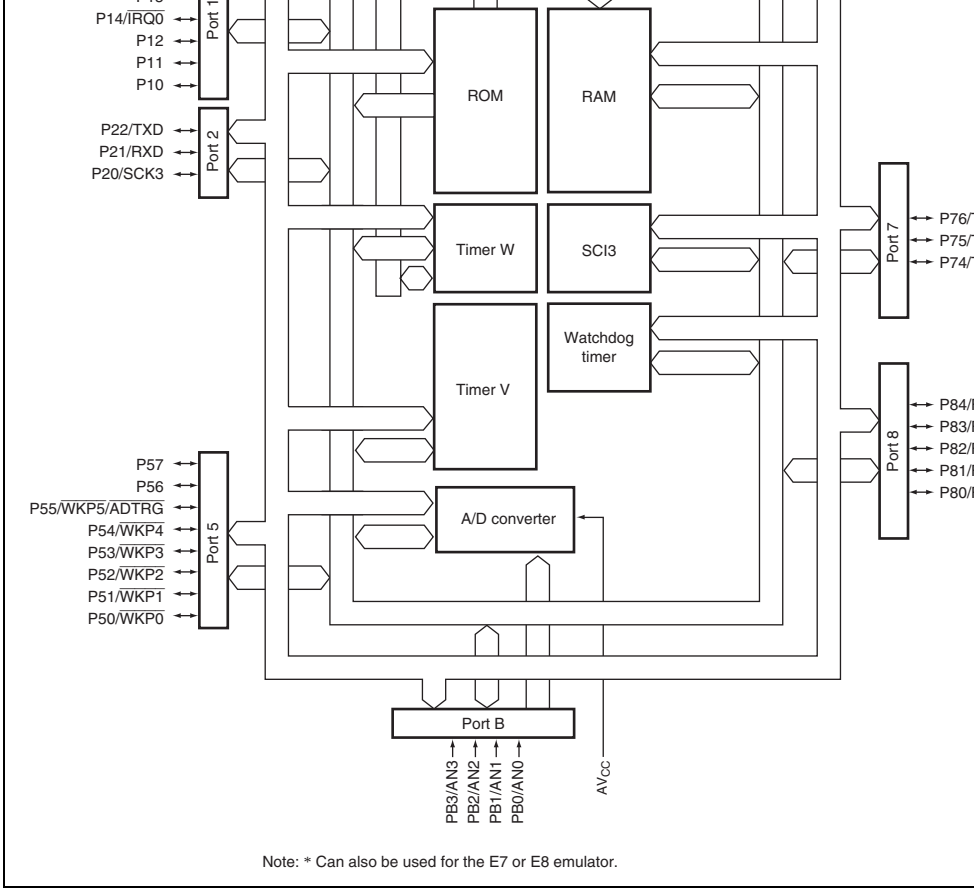
— Timer V (8-bit timer)
— Timer W (16-bit timer)
— Watchdog timer
— SCI3 (Asynchronous or clocked synchronous serial communication interface)
— 10-bit A/D converter

- On-chip memory

Product Classification		Model	ROM	RAM
Flash memory version	H8/3672	HD64F3672	16 kbytes	2,048 by
(F-ZTAT™ version)	H8/3670	HD64F3670	8 kbytes	2,048 by

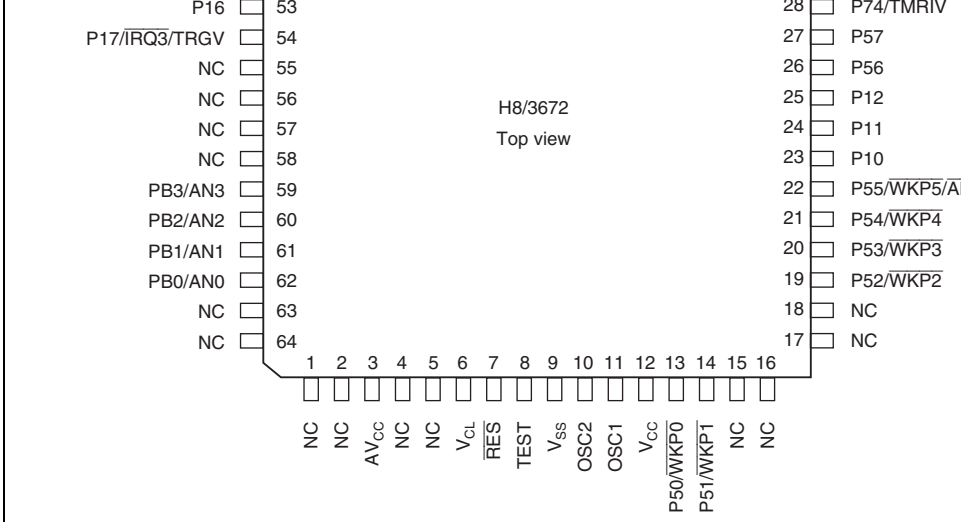
- General I/O ports
  — I/O pins: 26 I/O pins, including 5 large current ports (I_{OL} = 20 mA, @V_{OL} = 1.5 V
  — Input-only pins: 4 input pins (also used for analog input)
- Supports various power-down modes
  Note: F-ZTAT™ is a trademark of Renesas Technology Corp.
- Compact package

Package	Code	Body Size	Pin Pitch
LQFP-64	FP-64E	10.0 × 10.0 mm	0.5 mm
LQFP-48	FP-48F	10.0 × 10.0 mm	0.65 mm
LQFP-48	FP-48B	7.0 × 7.0 mm	0.5 mm



Note: * Can also be used for the E7 or E8 emulator.

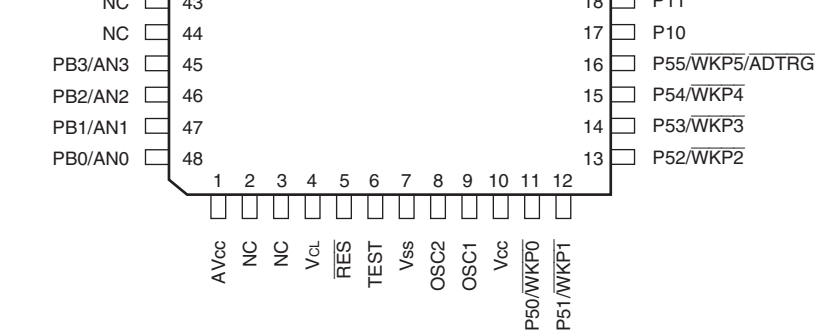
Figure 1.1 Internal Block Diagram



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

* Can also be used for the E7 or E8 emulator.

Figure 1.2 Pin Arrangement (FP-64E)



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

* Can also be used for the E7 or E8 emulator.

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

	AV_{CC}	3	1	Input	Analog power supply pin A/D converter. When the converter is not used, co pin to the system power
	V_{CL}	6	4	Input	Internal step-down powe pin. Connect a capacitor 0.1 µF between this pin a Vss pin for stabilization.
Clock pins	OSC1	11	9	Input	These pins connect to a ceramic resonator for sys clocks, or can be used to external clock. See section 5, Clock Puls Generators, for a typical connection.
	OSC2	10	8	Output	
System control	$\overline{RES}$	7	5	Input	Reset pin. When this driv the chip is reset.
	TEST	8	6	Input	Test pin. Connect this pi
Interrupt pins	$\overline{NMI}$	35	25	Input	Non-maskable interrupt input pin. Be sure to pull- pull-up resistor.
	$\overline{IRQ0}$, $\overline{IRQ3}$	51, 54	37, 40	Input	External interrupt reques pins. Can select the risin edge.
	$\overline{WKP0}$ to $\overline{WKP5}$	13, 14, 19 to 22	11 to 16	Input	External interrupt reques pins. Can select the risin edge.

	FTIOA to FTIOD	37 to 40	27 to 30	I/O	Output compare output/inp capture input/PWM output
Serial communication interface (SCI)	TXD	46	36	Output	Transmit data output pin
	RXD	45	35	Input	Receive data input pin
	SCK3	44	34	I/O	Clock I/O pin
A/D converter	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 P74	30 to 28	24 to 22	I/O	3-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, E8 emulator

- 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

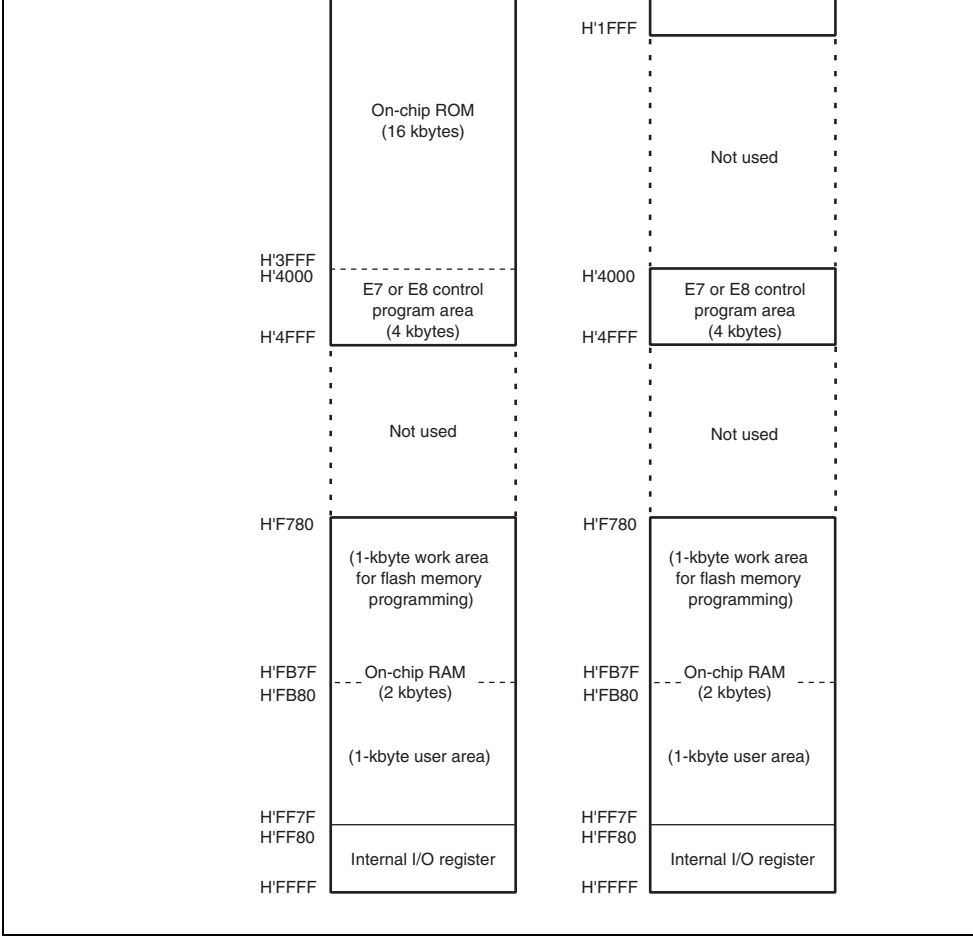
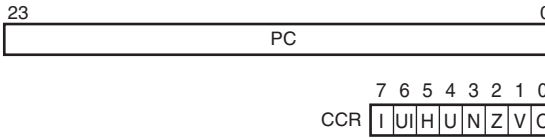


Figure 2.1 Memory Map

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

Control Registers (CR)



[Legend]

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

Figure 2.2 CPU Registers

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

The usage of each register can be selected independently.

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

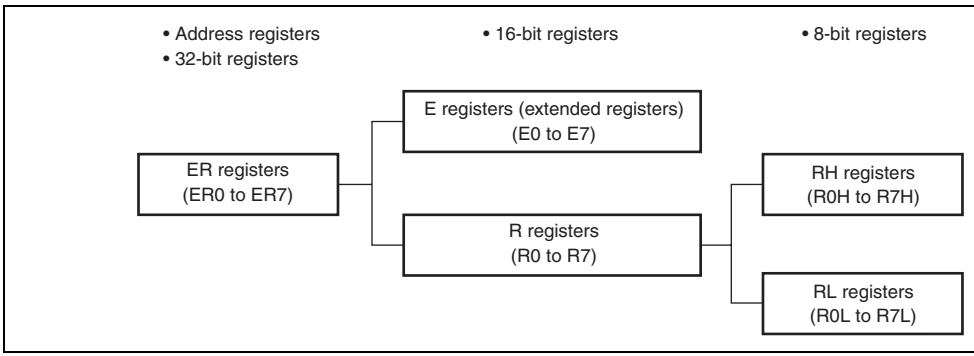


Figure 2.3 Usage of General Registers

Figure 2.4 Relationship between Stack Pointer and Stack Area

2.2.2 Program Counter (PC)

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

2.2.3 Condition-Code Register (CCR)

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

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

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

				of 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 a carry or borrow at bit 27, and cleared to 0 o
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.

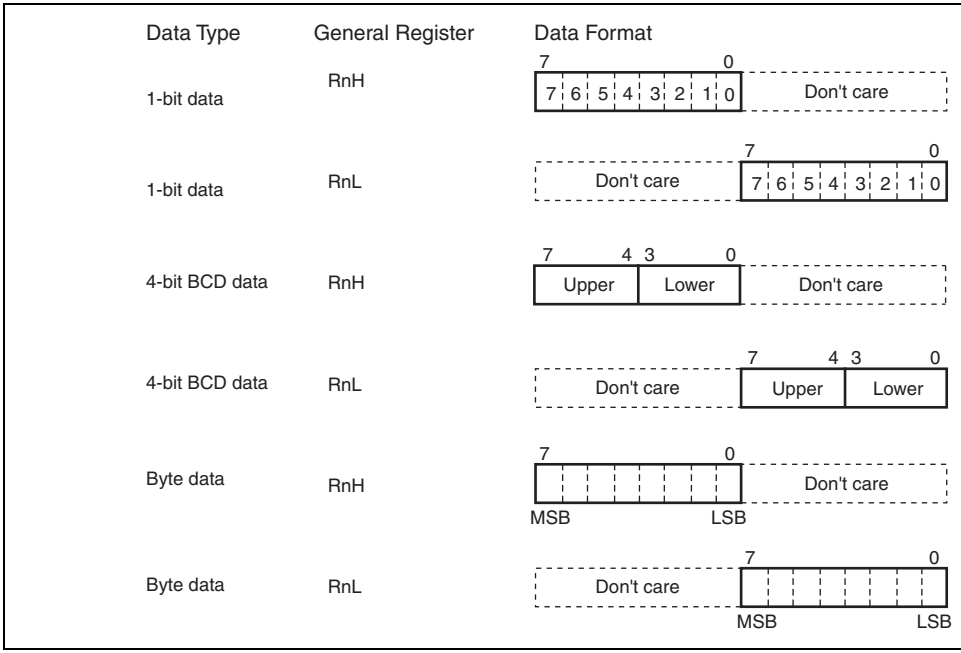
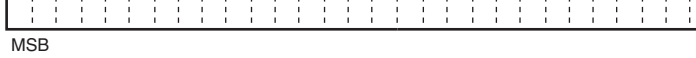


Figure 2.5 General Register Data Formats (1)



[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)

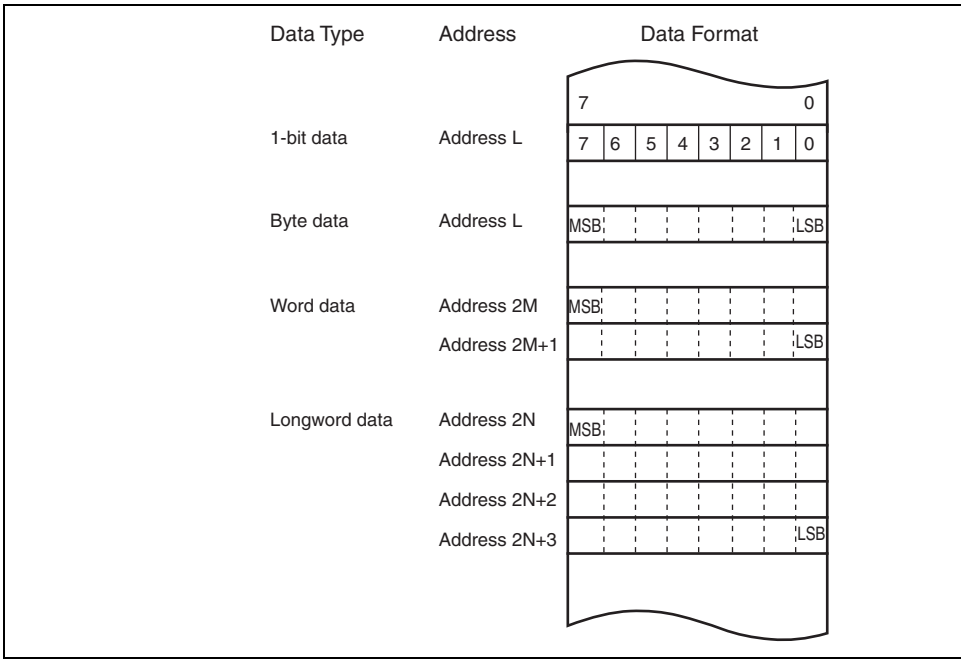


Figure 2.6 Memory Data Formats

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)
:3/:8/:16/:24	3-, 8-, 16-, or 24-bit length

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

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

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

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

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

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

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

		another general register or immediate data.
NOT	B/W/L	¬ (Rd) → (Rd) Takes the one's complement of general register contents.

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

Table 2.5 Shift Instructions

Instruction	Size*	Function
SHAL SHAR	B/W/L	Rd (shift) → Rd Performs an arithmetic shift on general register contents.
SHLL SHLR	B/W/L	Rd (shift) → Rd Performs a logical shift on general register contents.
ROTL ROTR	B/W/L	Rd (rotate) → Rd Rotates general register contents.
ROTXL ROTXR	B/W/L	Rd (rotate) → Rd Rotates general register contents through the carry flag.

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

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 general register.
BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operand a or clears the Z flag accordingly. The bit number is specified by immediate data or the lower three bits of a general register.
BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the carry flag with a specified bit in a general register or operand and stores the result in the carry flag.
BIAND	B	C ∧ ¬ (<bit-No.> of <EAd>) → C ANDs the carry flag with the inverse of a specified bit in a gen register or memory operand and stores the result in the carry f The bit number is specified by 3-bit immediate data.
BOR	B	C ∨ (<bit-No.> of <EAd>) → C ORs the carry flag with a specified bit in a general register or 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 gener or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

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

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

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

BCC(BHS)	Carry clear (high or same)	C = 0
BCS(BLO)	Carry set (low)	C = 1
BNE	Not equal	Z = 0
BEQ	Equal	Z = 1
BVC	Overflow clear	V = 0
BVS	Overflow set	V = 1
BPL	Plus	N = 0
BMI	Minus	N = 1
BGE	Greater or equal	N ⊕ V = 0
BLT	Less than	N ⊕ V = 1
BGT	Greater than	Z∨(N ⊕ V) = 0
BLE	Less or equal	Z∨(N ⊕ V) = 1

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

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

		code register size is one byte, but in transfer to memory, data is by word access.
ANDC	B	CCR ∧ #IMM → CCR, 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, t data for the number of bytes set in R4L or R4 to the address lo in ER6.
	Execution of the next instruction begins as soon as the transfe completed.

2.4.2 Basic Instruction Formats

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

Figure 2.7 shows examples of instruction formats.

- Condition Field

  Specifies the branching condition of Bcc instructions.

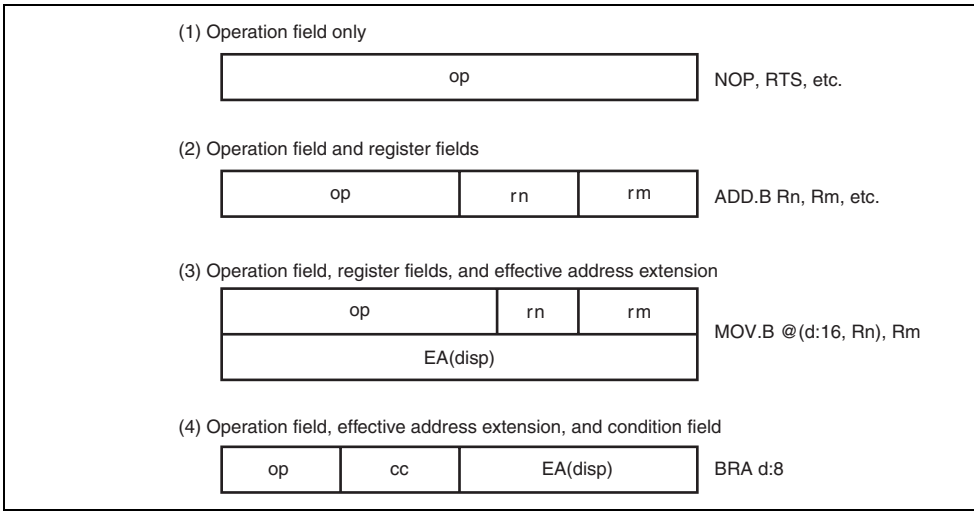


Figure 2.7 Instruction Formats

Arithmetic and logic instructions can use the register direct and immediate modes. Data
instructions can use all addressing modes except program-counter relative and memory
Bit manipulation instructions use register direct, register indirect, or the absolute addres
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

Register Direct—Rn

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

- Register indirect with post-increment—@ERn+

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

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

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

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

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

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

operand.

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

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

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

Memory Indirect—@@aa:8

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

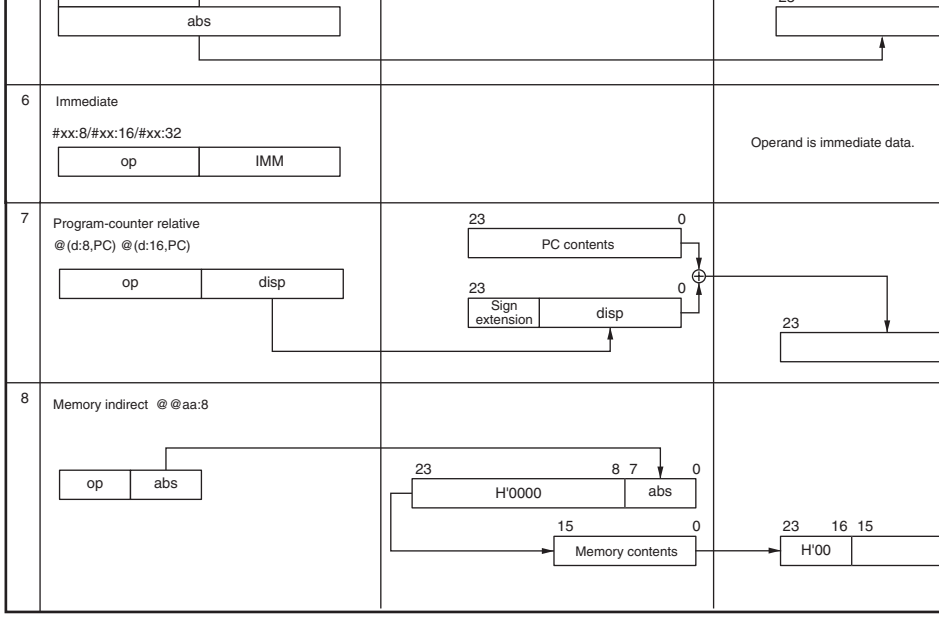
Note that the first part of the address range is also the exception vector area.

2.5.2 Effective Address Calculation

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

Table 2.12 Effective Address Calculation (1)

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



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

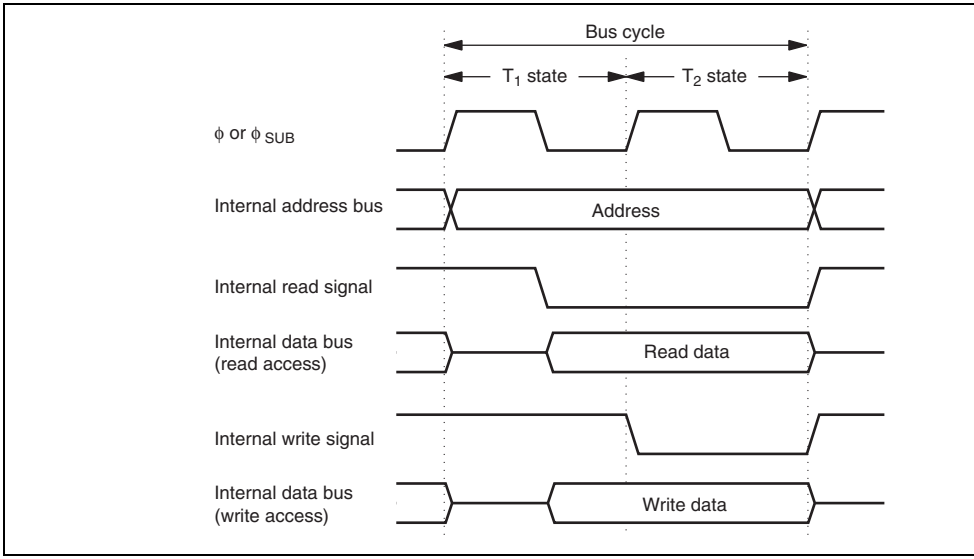


Figure 2.9 On-Chip Memory Access Cycle

module.

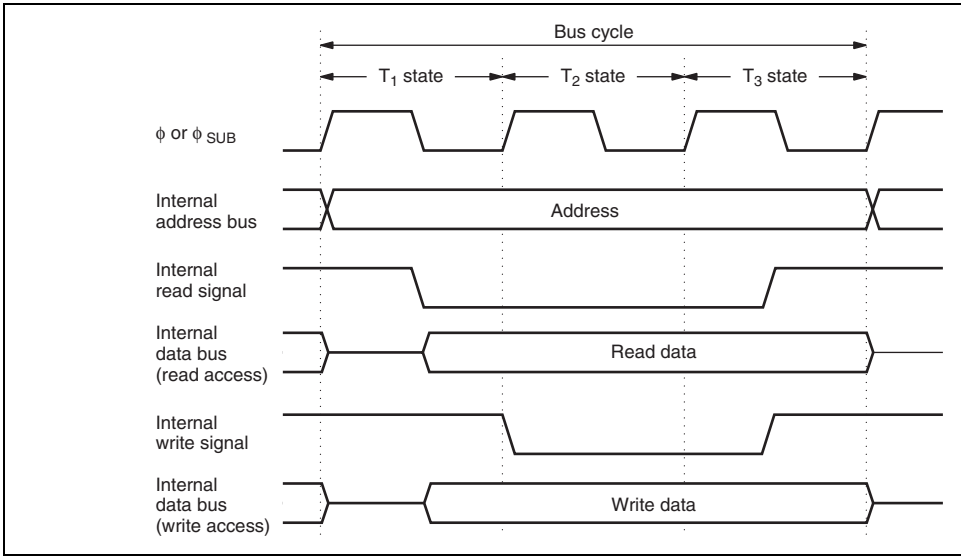


Figure 2.10 On-Chip Peripheral Module Access Cycle (3-State Access)

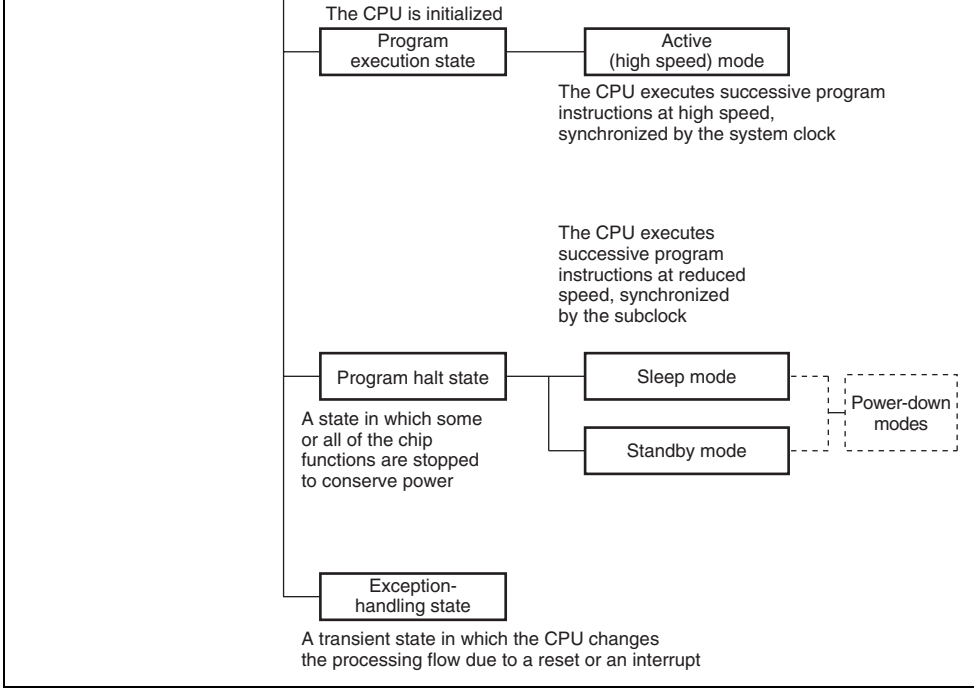


Figure 2.11 CPU Operation States

Figure 2.12 State Transitions

2.8 Usage Notes

2.8.1 Notes on Data Access to Empty Areas

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

2.8.2 EEPMOV Instruction

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

2.8.3 Bit Manipulation Instruction

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

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

The timer is counting, so the value read is not necessarily the same as the value in the tim
register. As a result, bits other than the intended bit in the timer counter may be modified
modified value may be written to the timer load register.

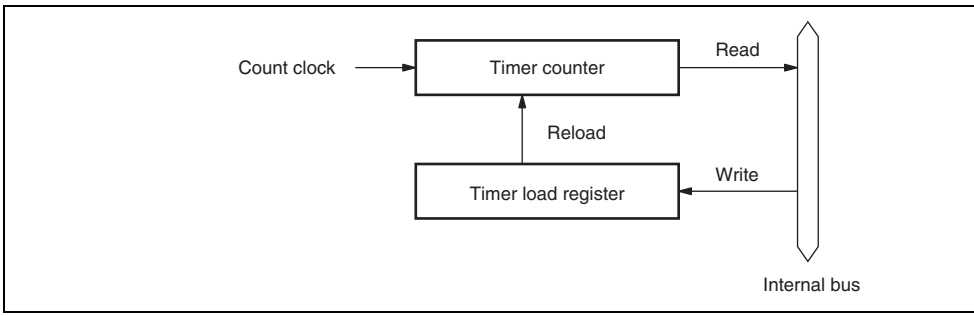


Figure 2.13 Example of Timer Configuration with Two Registers Allocated to S Address

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

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

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

The BSET instruction is executed for port 5.

- After executing BSET instruction

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

- Description on operation

1. When the BSET instruction is executed, first the CPU reads port 5.

   Since P57 and P56 are input pins, the CPU reads the pin states (low-level and high-l input).

   P55 to P50 are output pins, so the CPU reads the value in PDR5. In this example PD value of H'80, but the value read by the CPU is H'40.
2. Next, the CPU sets bit 0 of the read data to 1, changing the PDR5 data to H'41.
3. Finally, the CPU writes H'41 to PDR5, completing execution of BSET.

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

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

- BSET instruction executed

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

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

- After executing BSET instruction

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

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

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

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

- BCLR instruction executed

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

The BCLR instruction is executed for PCR5.

- After executing BCLR instruction

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

- Description on operation

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

   As a result of this operation, bit 0 in PCR5 becomes 0, making P50 an input port. Ho bits 7 and 6 in PCR5 change to 1, so that P57 and P56 change from input pins to outp To prevent this problem, store a copy of the PCR5 data in a work area in memory an 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 wo
(RAM0).

- After executing BCLR instruction

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

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

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

Exception handling starts when a trap instruction (TRAPA) is executed. The TRAPA in
generates a vector address corresponding to a vector number from 0 to 3, as specified in
instruction code. Exception handling can be executed at all times in the program executi

- Interrupts

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

—	Reserved for system use	1 to 6	H'0002 to H'000D
External interrupt pin	NMI	7	H'000E to H'000F
CPU	Trap instruction (#0)	8	H'0010 to H'0011
	(#1)	9	H'0012 to H'0013
	(#2)	10	H'0014 to H'0015
	(#3)	11	H'0016 to H'0017
Address break	Break conditions satisfied	12	H'0018 to H'0019
CPU	Direct transition by executing the SLEEP instruction	13	H'001A to H'001B
External interrupt pin	IRQ0	14	H'001C to H'001D
	IRQ3	17	H'0022 to H'0023
	WKP	18	H'0024 to H'0025
—	Reserved for system use	20	H'0028 to H'0029
Timer W	Input capture A/compare match A Input capture B/compare match B Input capture C/compare match C Input capture D/compare match D Timer W overflow	21	H'002A to H'002B
Timer V	Timer V compare match A Timer V compare match B Timer V overflow	22	H'002C to H'002D
SCI3	SCI3 receive data full SCI3 transmit data empty SCI3 transmit end SCI3 receive error	23	H'002E to H'002F
A/D converter	A/D conversion end	25	H'0032 to H'0033

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	—	1	—	Reserved
5	—	1	—	These bits are always read as 1.
4	—	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	—	0	—	Reserved
1	—	0	—	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

				1: Rising edge of $\overline{\text{WKP5}}$ ($\overline{\text{ADTRG}}$) pin input is d
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 th $\overline{WKP5}$ to $\overline{WKP0}$. When the bit is set to 1, inter 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 t pin are enabled.
2	—	0	—	Reserved
1	—	0	—	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 t pin are enabled.

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

6	—	0	—	Reserved This bit is always read as 0.
5	—	1	—	Reserved
4	—	1	—	These bits are always read as 1.
3	IRRI3	0	R/W	IRQ3 Interrupt Request Flag [Setting condition] When $\overline{\text{IRQ3}}$ pin is designated for interrupt input designated signal edge is detected. [Clearing condition] When IRRI3 is cleared by writing 0
2	—	0	—	Reserved
1	—	0	—	These bits are always read as 0.
0	IRRI0	0	R/W	IRQ0 Interrupt Request Flag [Setting condition] When $\overline{\text{IRQ0}}$ pin is designated for interrupt input designated signal edge is detected. [Clearing condition] When IRRI0 is cleared by writing 0

4	IWPF4	0	R/W	WKP4 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP4}}$ pin is designated for interrupt inp designated signal edge is detected. [Clearing condition] When IWPF4 is cleared by writing 0.
3	IWPF3	0	R/W	WKP3 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP3}}$ pin is designated for interrupt inp 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 $\overline{\text{WKP2}}$ pin is designated for interrupt inp designated signal edge is detected. [Clearing condition] When IWPF2 is cleared by writing 0.
1	IWPF1	0	R/W	WKP1 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP1}}$ pin is designated for interrupt inp designated signal edge is detected. [Clearing condition] When IWPF1 is cleared by writing 0.
0	IWPF0	0	R/W	WKP0 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP0}}$ pin is designated for interrupt inp designated signal edge is detected. [Clearing condition] When IWPF0 is cleared by writing 0.

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 progra execution starts from that address.

3.4 Interrupt Exception Handling

3.4.1 External Interrupts

There are external interrupts, NMI, IRQ3, IRQ0, and WKP.

NMI

NMI interrupt is requested by input falling edge to pin $\overline{\text{NMI}}$.

NMI is the highest interrupt, and can always be accepted without depending on the I b in CCR.

IRQ3 to IRQ0 Interrupts

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

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

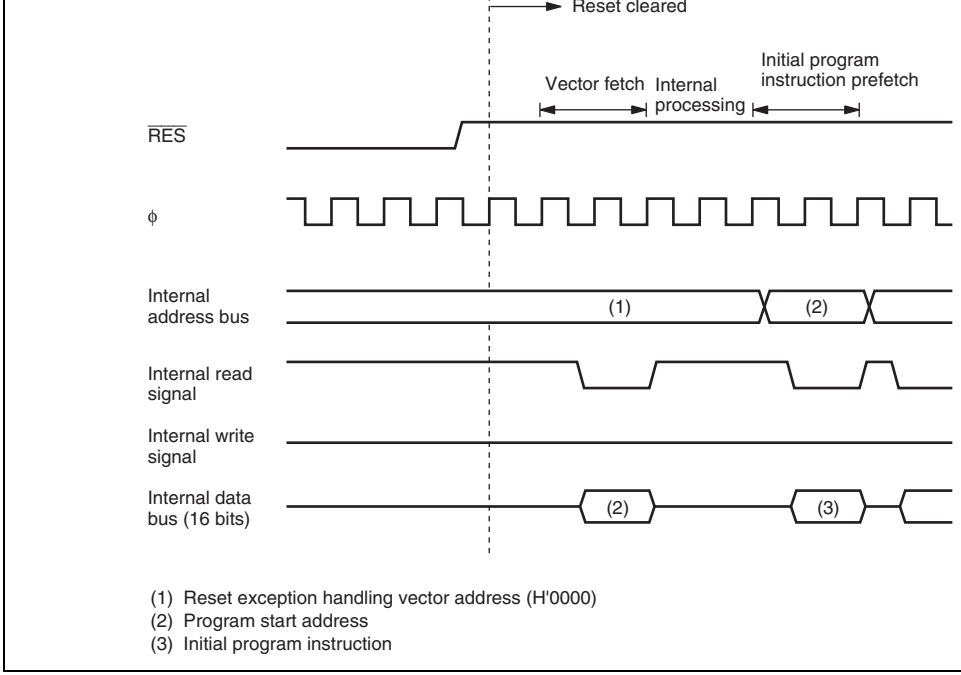


Figure 3.1 Reset Sequence

3.4.2 Internal Interrupts

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

1. If an interrupt occurs while the NMI or interrupt enable bit is set to 1, an interrupt req signal is sent to the interrupt controller.
2. When multiple interrupt requests are generated, the interrupt controller requests to the the interrupt handling with the highest priority at that time according to table 3.1. Oth interrupt requests are held pending.
3. The CPU accepts the NMI or address break without depending on the I bit value. Oth interrupt requests are accepted, if the I bit is cleared to 0 in CCR; if the I bit is set to 1 interrupt request is held pending.
4. If the CPU accepts the interrupt after processing of the current instruction is complete interrupt exception handling will begin. First, both PC and CCR are pushed onto the s state of the stack at this time is shown in figure 3.2. The PC value pushed onto the sta 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 a break. Upon return from interrupt handling, the values of I bit and other bits in CCR v 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, a transfers the address to PC as a start address of the interrupt handling-routine. Then a starts executing from the address indicated in PC.

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

[Legend]
PCH: Upper 8 bits of program counter (PC)
PCL: Lower 8 bits of program counter (PC)
CCR: Condition code register
SP: Stack pointer

Notes: 1. PC shows the address of the first instruction to be executed upon return from the interrupt handling routine.
2. Register contents must always be saved and restored by word length, starting from an even-numbered address.
3. Ignored when returning from the interrupt handling routine.

Figure 3.2 Stack Status after Exception Handling

3.4.4 Interrupt Response Time

Table 3.2 shows the number of wait states after an interrupt request flag is set until the f 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. Acc
stack always takes place in word size, so the stack pointer (SP: R7) should never indicat
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,
$\overline{\text{IRQ0}}$, 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 procedu

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
interrupt request flag from 1 to 0.

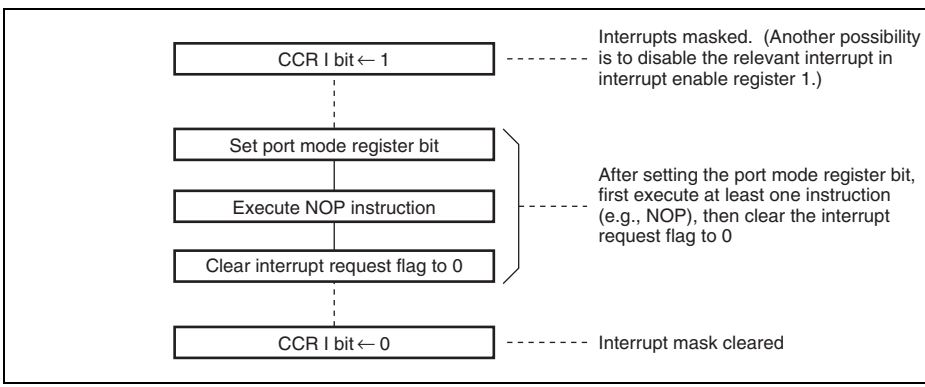


Figure 3.4 Port Mode Register Setting and Interrupt Request Flag Clearing Pr

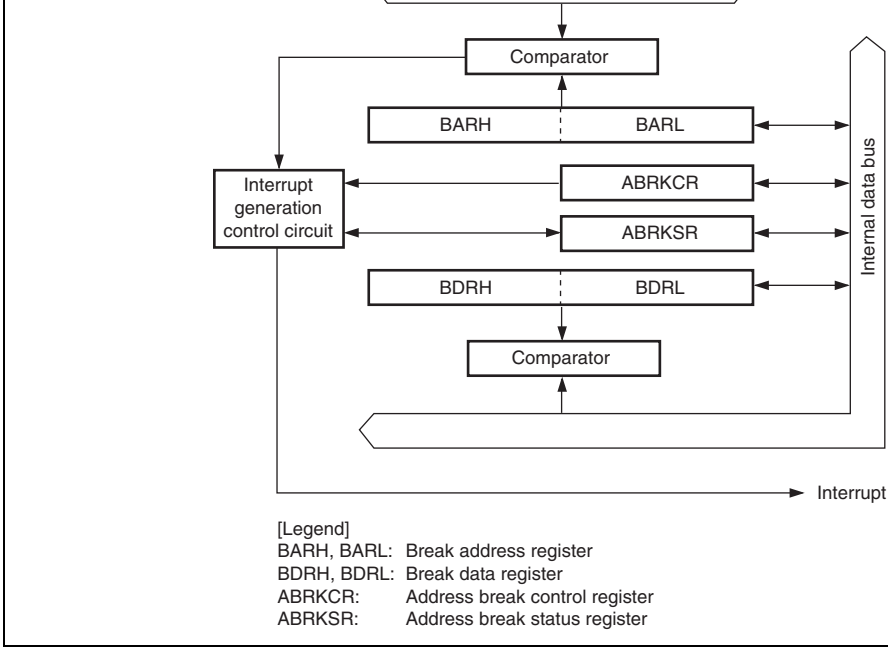


Figure 4.1 Block Diagram of Address Break

4.1 Register Descriptions

Address break has the following registers.

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

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

Legend: X: Don't care.

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

4.1.2 Address Break Status Register (ABRKSR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	ABIF	0	R/W	Address Break Interrupt Flag [Setting condition] When the condition set in ABRKCR is satisfie [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 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.

When the address break is specified in instruction execution cycle

Register setting
- ABRKCR = H'80
- BAR = H'025A

Program

	0258	NOP
*	025A	NOP
	025C	MOV.W @H'025A,R0
	0260	NOP
	0262	NOP
	:	:

Underline indicates the address to be stacked.

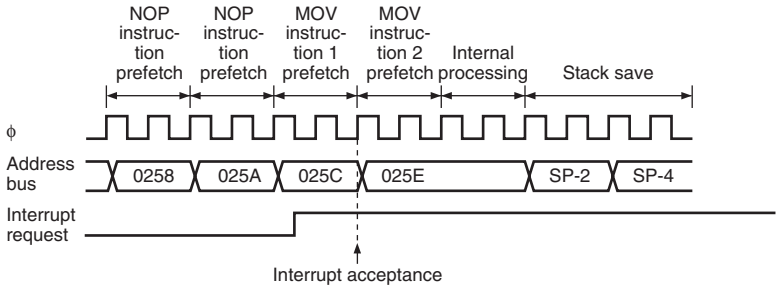


Figure 4.2 Address Break Interrupt Operation Example (1)

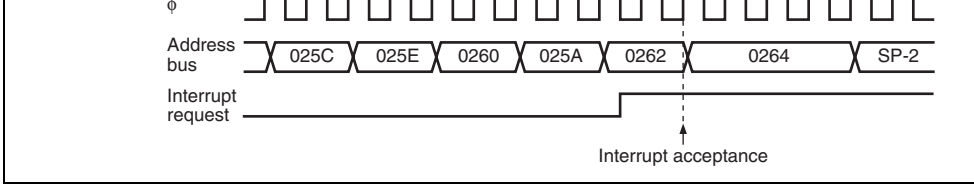


Figure 4.2 Address Break Interrupt Operation Example (2)

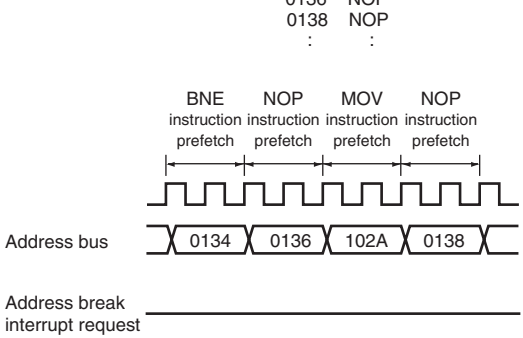


Figure 4.3 Operation when Condition is not Satisfied in Branch Instructio

When another interrupt request is accepted before an instruction to which an address bre executed, exception handling of an address break interrupt is not executed. However, th is set to 1 (see figure 4.4). Therefore the ABIF bit must be read during exception handli address break interrupt.

Figure 4.4 Operation when Another Interrupt is Accepted at Address Break Se Instruction

When an address break is set to an instruction as a branch destination of a conditional bra instruction, the instruction set when the condition of the branch instruction is not satisfied executed, and an address break is generated. Therefore an address break must not be set t instruction as a branch destination of a conditional branch instruction.

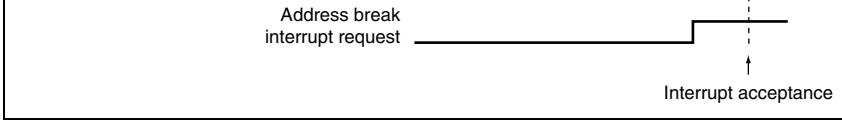


**Figure 4.5 Operation when the Instruction Set is not Executed and does not Bran
Conditions not Being Satisfied**

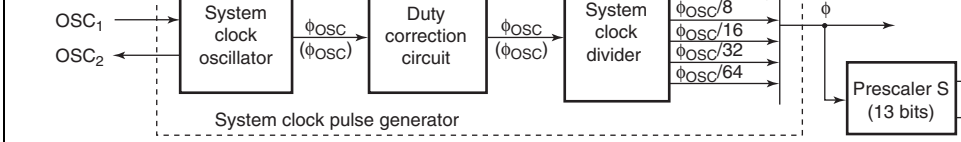


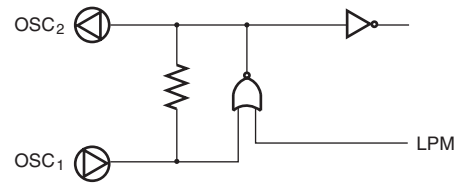
Figure 5.1 Block Diagram of Clock Pulse Generators

The basic clock signals that drive the CPU and on-chip peripheral modules are ø.

The system clock is divided into ø/8192 to ø/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.



LPM: Low-power mode (standby mode, subsleep mode)

Figure 5.2 Block Diagram of System Clock Generator

Figure 5.3 Typical Connection to Crystal Resonator

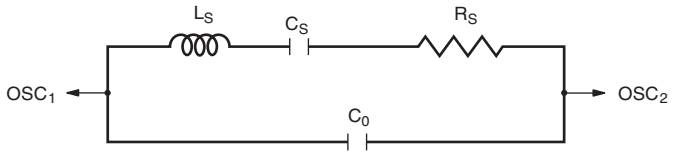


Figure 5.4 Equivalent Circuit of Crystal Resonator

Table 5.1 Crystal Resonator Parameters

Frequency (MHz)	2	4	8	10	16
R_s (max)	500 Ω	120 Ω	80 Ω	60 Ω	50 Ω
C_0 (max)	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.

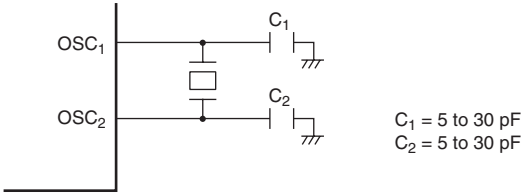


Figure 5.5 Typical Connection to Ceramic Resonator

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 (ø) as its input clock. It is increme
per clock period. Prescaler S is initialized to H'0000 by a reset, and starts counting on ex
the reset state. In standby mode 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.

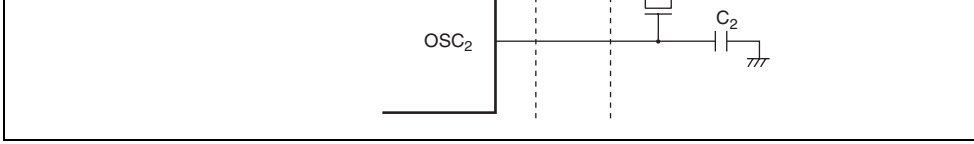


Figure 5.7 Example of Incorrect Board Design

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.

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

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby This bit selects the mode to transit after the exec the SLEEP instruction. 0: a transition is made to sleep mode 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	—	0	—	Reserved These bits are always read as 0.

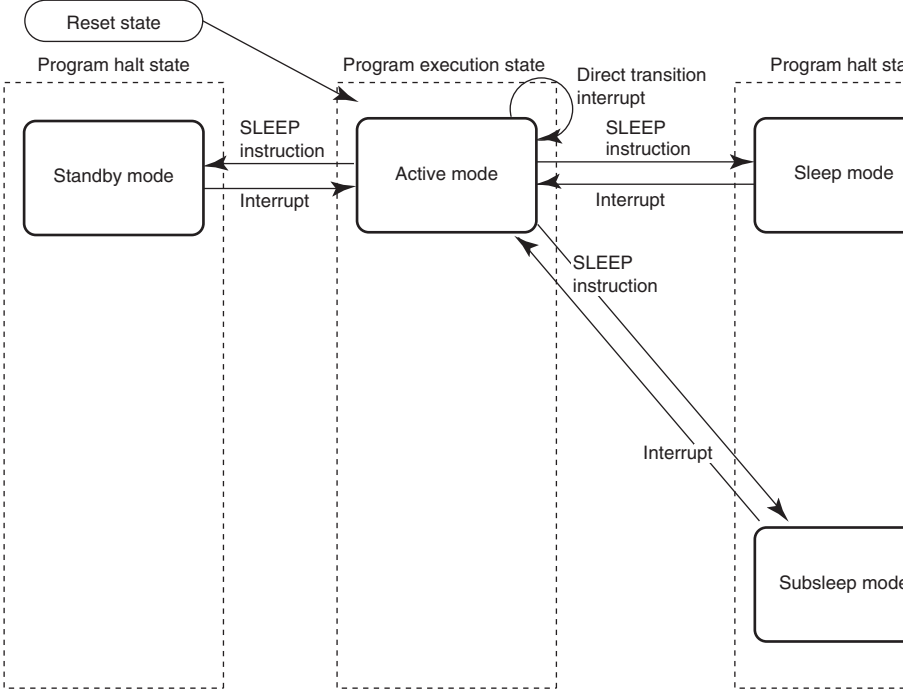
1	16 states	0.00	0.00	0.00	0.00	0.01	0.02

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

Legend: X: Don't care.

				A/D converter enters standby mode when this b 1
3	MSTWD	0	R/W	Watchdog Timer Module Standby Watchdog timer enters standby mode when this to 1.When the internal oscillator is selected for watchdog timer clock, the watchdog timer opera regardless of the setting of this bit
2	MSTTW	0	R/W	Timer W Module Standby Timer W enters standby mode when this bit is s
1	MSTTV	0	R/W	Timer V Module Standby Timer V enters standby mode when this bit is s
0	—	0	—	Reserved This bit is always read as 0.



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

Figure 6.1 Mode Transition Diagram

functions after entering active mode, reset the registers.

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 content 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 inter is selected as a the counter is in by a subclock)
	Watchdog timer	Functioning	Functioning	Retained	Retained (functio internal oscillato as a count clock
	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 mod functioning. However, as long as the rated voltage is supplied, the contents of CPU regist chip RAM, and some on-chip peripheral module registers are retained. On-chip RAM con will be retained as long as the voltage set by the RAM data retention voltage is provided. ports go to the high-impedance state.

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

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

6.2.3 Subsleep Mode

In subsleep mode, 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 CPU registers, the on-chip RAM, and some registers of the on-chip peripheral modules are reta ports keep the same states as before the transition.

Subsleep mode is cleared by an interrupt. When an interrupt is requested, the system cloc oscillator starts to oscillate. Subsleep mode is cleared and an interrupt exception handling when the time set in bits STS2 to STS0 in SYSCR1 elapses. Subsleep mode is not cleared 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 m
instead to sleep mode. Note that if a direct transition is attempted while the I bit in CCR
sleep mode will be entered, and the resulting mode cannot be cleared by means of an int

6.5 Module Standby Function

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

7.1 Block Configuration

Figure 7.1 shows the block configuration of 20-kbyte flash memory. The thick lines indi erasing units, the narrow lines indicate programming units, and the values are addresses. memory is divided into 1 kbyte × 4 blocks and 16 kbytes × 1 block. Erasing is performe units. Programming is performed in 128-byte units starting from an address with lower e H'00 or H'80.

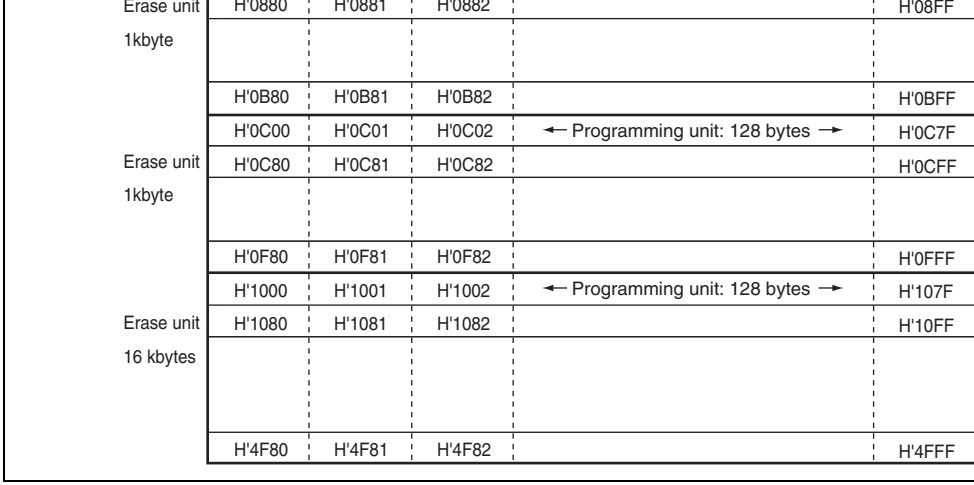


Figure 7.1 Flash Memory Block Configuration

7.2 Register Descriptions

The flash memory has the following registers.

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

				programming/erasing is enabled. When this bi cleared to 0, other FLMCR1 register bits and a bits cannot be set.
5	ESU	0	R/W	Erase Setup When this bit is set to 1, the flash memory cha the erase setup state. When it is cleared to 0, setup state is cancelled. Set this bit to 1 before the E bit to 1 in FLMCR1.
4	PSU	0	R/W	Program Setup When this bit is set to 1, the flash memory cha the program setup state. When it is cleared to program setup state is cancelled. Set this bit t setting the P bit in FLMCR1.
3	EV	0	R/W	Erase-Verify When this bit is set to 1, the flash memory cha erase-verify mode. When it is cleared to 0, era mode is cancelled.
2	PV	0	R/W	Program-Verify When this bit is set to 1, the flash memory cha program-verify mode. When it is cleared to 0, verify mode is cancelled.
1	E	0	R/W	Erase When this bit is set to 1, and while the SWE = ESU = 1 bits are 1, the flash memory changes mode. When it is cleared to 0, erase mode is c
0	P	0	R/W	Program When this bit is set to 1, and while the SWE = PSU = 1 bits are 1, the flash memory changes program mode. When it is cleared to 0, progra cancelled.

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

7.2.3 Erase Block Register 1 (EBR1)

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	—	Reserved These bits are always read as 0.
4	EB4	0	R/W	When this bit is set to 1, 16 kbytes of H'1000 t H'4FFF will be erased.
3	EB3	0	R/W	When this bit is set to 1, 1 kbyte of H'0C00 to will be erased.
2	EB2	0	R/W	When this bit is set to 1, 1 kbyte of H'0800 to will be erased.
1	EB1	0	R/W	When this bit is set to 1, 1 kbyte of H'0400 to will be erased.
0	EB0	0	R/W	When this bit is set to 1, 1 kbyte of H'0000 to will be erased.

6 to 0	All 0	Reserved These bits are always read as 0.

7.3 On-Board Programming Modes

There is a mode for programming/erasing of the flash memory; boot mode, which enabl
board programming/erasing. On-board programming/erasing can also be performed in u
program mode. At reset-start in reset mode, this LSI changes to a mode depending on th
pin settings, $\overline{\text{NMI}}$ pin settings, and input level of each port, as shown in table 7.1. The in
of each pin must be defined four states before the reset ends.

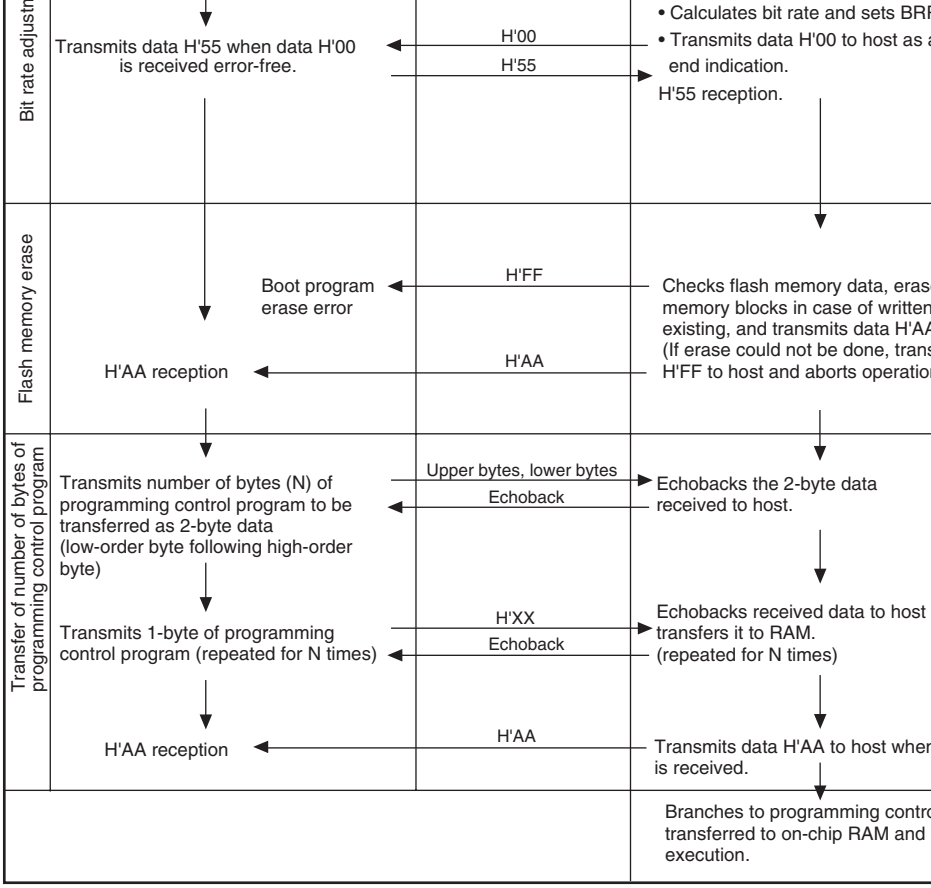
When changing to boot mode, the boot program built into this LSI is initiated. The boot
transfers the programming control program from the externally-connected host to on-chi
via SCI3. After erasing the entire flash memory, the programming control program is ex
This can be used for programming initial values in the on-board state or for a forcible re
programming/erasing can no longer be done in user program mode. In user program mo
individual blocks can be erased and programmed by branching to the user program/erase
program prepared by the user.

Table 7.1 Setting Programming Modes

TEST	$\overline{\text{NMI}}$	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.

3. When the boot program is initiated, the chip measures the low-level period of asynchron SCI communication data (H'00) transmitted continuously from the host. The chip then 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.
4. After matching the bit rates, the chip transmits one H'00 byte to the host to indicate th completion of bit rate adjustment. The host should confirm that this adjustment end in (H'00) has been received normally, and transmit one H'55 byte to the chip. If receptio not be performed normally, initiate boot mode again by a reset. Depending on the hos transfer bit rate and system clock frequency of this LSI, there will be a discrepancy be the bit rates of the host and the chip. To operate the SCI properly, set the host's transfe and system clock frequency of this LSI within the ranges listed in table 7.3.
5. In boot mode, a part of the on-chip RAM area is used by the boot program. The area f H'FEEF is the area to which the programming control program is transferred from the The boot program area cannot be used until the execution state in boot mode switches programming control program.
6. Before branching to the programming control program, the chip terminates transfer op by SCI3 (by clearing the RE and TE bits in SCR3 to 0), however the adjusted bit rate remains set in BRR. Therefore, the programming control program can still use it for tr write data or verify data with the host. The TxD pin is high (PCR22 = 1, P22 = 1). Th contents of the CPU general registers are undefined immediately after branching to th programming control program. These registers must be initialized at the beginning of programming control program, as the stack pointer (SP), in particular, is used implicit subroutine calls, etc.
7. Boot mode can be cleared by a reset. End the reset after driving the reset pin low, wai least 20 states, and then setting the TEST pin and $\overline{\text{NMI}}$ pin. Boot mode is also cleared WDT overflow occurs.
8. Do not change the TEST pin and NMI pin input levels in boot mode.


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

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

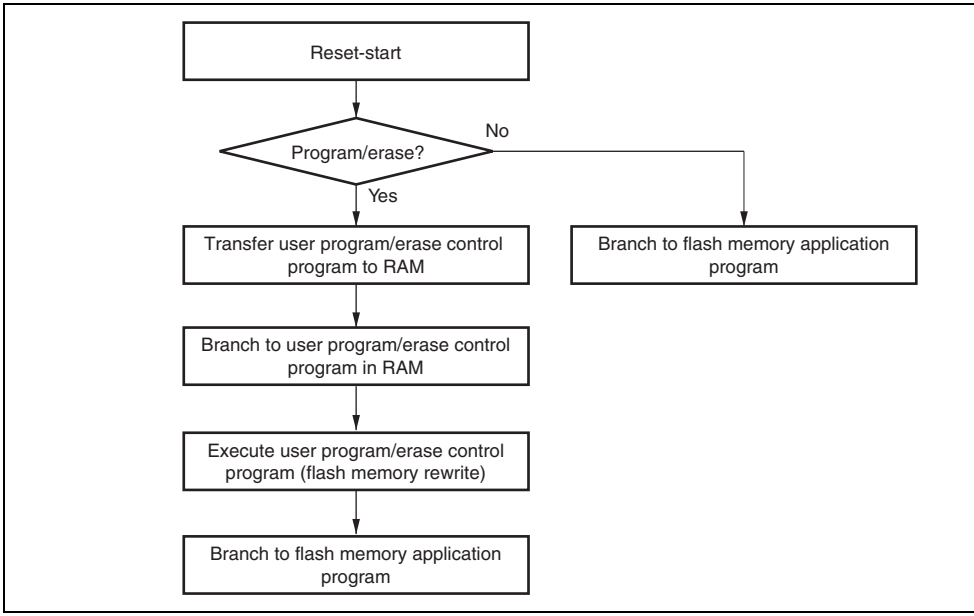
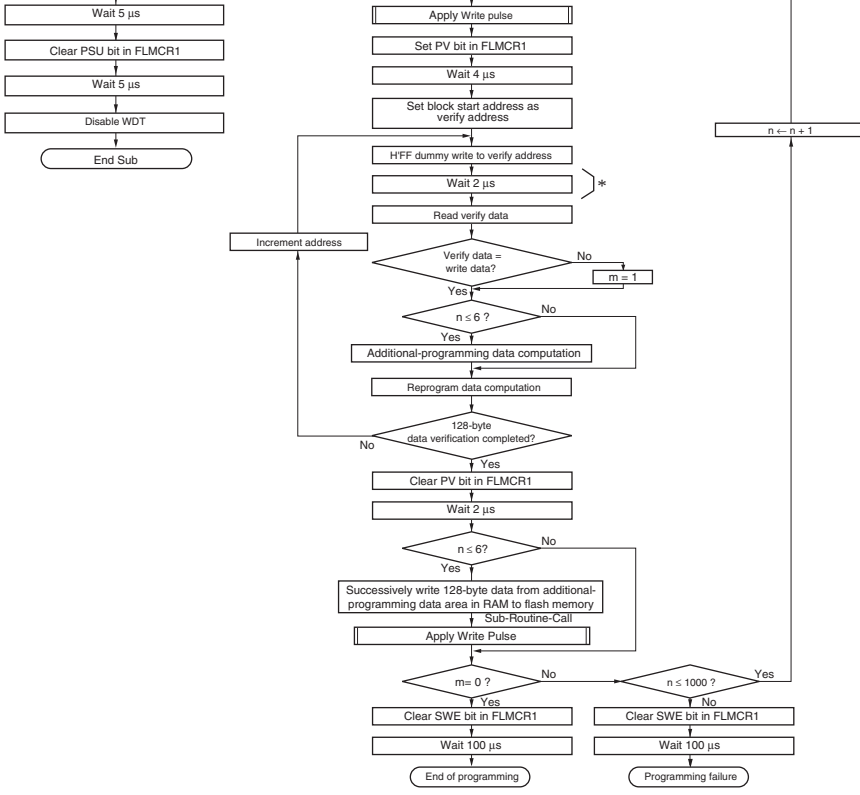


Figure 7.2 Programming/Erasing Flowchart Example in User Program Mo

7.4.1 Program/Program-Verify

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

1. Programming must be done to an empty address. Do not reprogram an address to wh
programming has already been performed.
2. Programming should be carried out 128 bytes at a time. A 128-byte data transfer mu
performed even if writing fewer than 128 bytes. In this case, H'FF data must be writt
extra addresses.
3. Prepare the following data storage areas in RAM: A 128-byte programming data are
byte reprogramming data area, and a 128-byte additional-programming data area. Pe
reprogramming data computation according to table 7.4, and additional programmin
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 12
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 l
are B'00. Verify data can be read in words or in longwords from the address to which
write was performed.



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

Figure 7.3 Program/Program-Verify Flowchart

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

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.

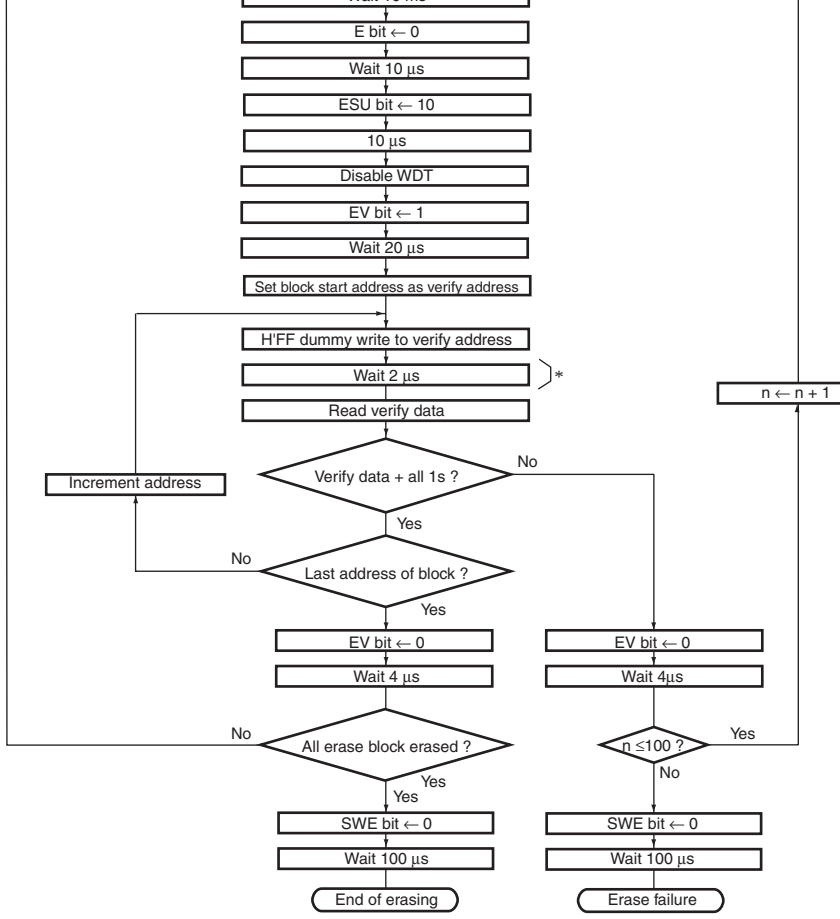
7.4.2 Erase/Erase-Verify

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, e overflow cycle of approximately 19.8 ms is allowed.

of 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 e 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 can carried out.



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

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 memor by clearing the SWE bit in FLMCR1. When software protection is in effect, setting the P in FLMCR1 does not cause a transition to program mode or erase mode. By setting the er register 1 (EBR1), erase protection can be set for individual blocks. When EBR1 is set to 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/eras algorithm, and the program/erase operation is aborted. Aborting the program/erase operat prevents damage to the flash memory due to overprogramming or overerasing.

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

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

The FLMCR1, FLMCR2, and EBR1 settings are retained, however program mode or eras is aborted at the point at which the error occurred. Program mode or erase mode cannot b entered by re-setting the P or E bit. However, PV and EV bit setting is enabled, and a tran can be made to verify mode. Error protection can be cleared only by a power-on reset.

functions in each port, see appendix B.1, I/O Port Block Diagrams. For the execution of 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 and a timer V in Figure 9.1 shows its pin configuration.

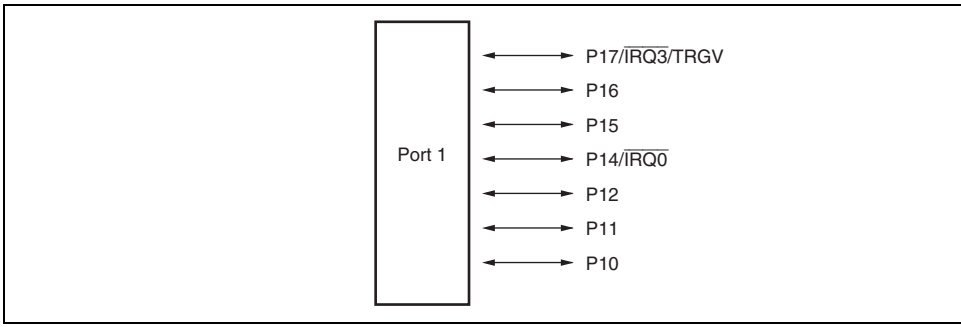


Figure 9.1 Port 1 Pin Configuration

Port 1 has the following registers.

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

6	—	0	—	Reserved
5	—	0	—	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	—	1	—	Reserved This bit is always read as 1.
2	—	0	R/W	Reserved This bit must always be cleared to 0 (setting to 1 i disabled).
1	TXD	0	R/W	P22/TXD Pin Function Switch This bit selects whether pin P22/TXD is used as P TXD. 0: General I/O port 1: TXD output pin
0	—	0	—	Reserved These bits are always read as 0.

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, the stored in PDR1 are read. If PDR1 is read while P are cleared to 0, the pin states are read regardle 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	

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 pin

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

P11 pin

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

P10 pin

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

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	—	—	—	Reserved
6	—	—	—	
5	—	—	—	
4	—	—	—	
3	—	—	—	
2	PCR22	0	W	When each of the port 2 pins P22 to P20 function
1	PCR21	0	W	general I/O port, setting a PCR2 bit to 1 makes th
0	PCR20	0	W	corresponding pin an output port, while clearing th makes the pin an input port.

2	P22	0	R/W	PDR2 stores output data for port 2 pins.
1	P21	0	R/W	If PDR2 is read while PCR2 bits are set to 1, the
0	P20	0	R/W	stored in PDR2 is read. If PDR2 is read while PC are cleared to 0, the pin states are read regardle 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 A/D trigger input pin and wakeup interr
pin. Each pin of the port 5 is shown in figure 9.3.

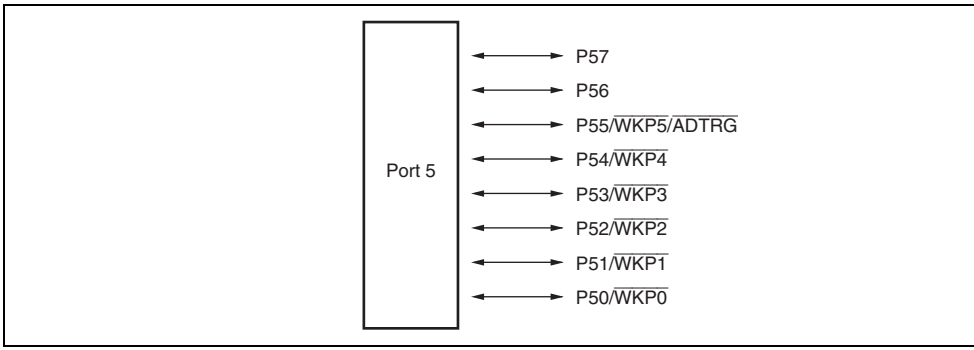


Figure 9.3 Port 5 Pin Configuration

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)

				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 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 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 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 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 0: General I/O port 1: $\overline{\text{WKP1}}$ input pin
0	WKP0	0	R/W	P50/$\overline{\text{WKP0}}$ Pin Function Switch Selects whether pin P50/$\overline{\text{WKP0}}$ is used as P50 or 0: General I/O port 1: $\overline{\text{WKP0}}$ input pin

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

9.3.3 Port Data Register 5 (PDR5)

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

Bit	Bit Name	Initial Value	R/W	Description
7	P57	0	R/W	Stores output data for port 5 pins.
6	P56	0	R/W	If PDR5 is read while PCR5 bits are set to 1, the v
5	P55	0	R/W	stored in PDR5 are read. If PDR5 is read while PC
4	P54	0	R/W	are cleared to 0, the pin states are read regardles value stored in PDR5.
3	P53	0	R/W	
2	P52	0	R/W	
1	P51	0	R/W	
0	P50	0	R/W	

2	PUCR52	0	R/W
1	PUCR51	0	R/W
0	PUCR50	0	R/W

9.3.5 Pin Functions

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

P57 pin

Register	PMR5	PCR5	
Bit Name	**POF7**	**PCR57**	**Pin Function**
Setting Value	X	0	P57 input pin
	0	1	CMOS output
	1	1	NMOS open-drain output

Legend: X: Don't care.

P56 pin

Register	PMR5	PCR5	
Bit Name	**POF6**	**PCR56**	**Pin Function**
Setting Value	X	0	P56 input pin
	0	1	CMOS output
	1	1	NMOS open-drain output

Legend: X: Don't care.

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

Legend: X: Don't care.

P53/$\overline{\text{WKP3}}$ pin

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

Legend: X: Don't care.

P52/$\overline{\text{WKP2}}$ pin

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

Legend: X: Don't care.

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

Legend: X: Don't care.

9.4 Port 7

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

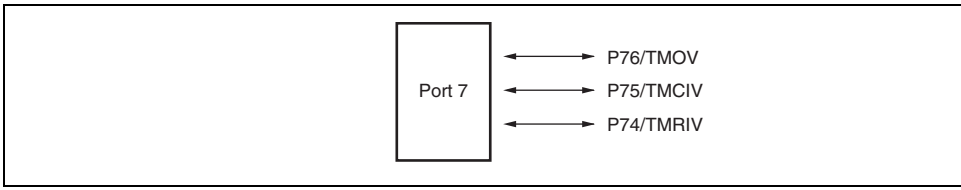


Figure 9.4 Port 7 Pin Configuration

Port 7 has the following registers.

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

3	—	—	—	Reserved
2	—	—	—	
1	—	—	—	
0	—	—	—	

9.4.2 Port Data Register 7 (PDR7)

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

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

	Other than the above values	X	TMOV output pin

Legend: X: Don't care.

P75/TMCIV pin

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

P74/TMRIV pin

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

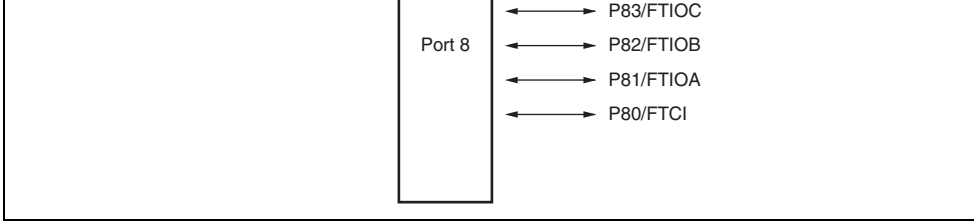


Figure 9.5 Port 8 Pin Configuration

Port 8 has the following registers.

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

9.5.1 Port Control Register 8 (PCR8)

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Reserved
6	—	—	—	
5	—	—	—	
4	PCR84	0	W	When each of the port 8 pins P84 to P80 function general I/O port, setting a PCR8 bit to 1 makes th corresponding pin an output port, while clearing th makes the pin an input port.
3	PCR83	0	W	
2	PCR82	0	W	
1	PCR81	0	W	
0	PCR80	0	W	

2	P82	0	R/W	stored in PDR8 is read. If PDR8 is read while PC
1	P81	0	R/W	are cleared to 0, the pin states are read regardle
0	P80	0	R/W	value stored in PDR8.

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 pi
	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 pi

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 p
	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 p

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 p
	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 p

Legend: X: Don't care.

B is shown in figure 9.6.

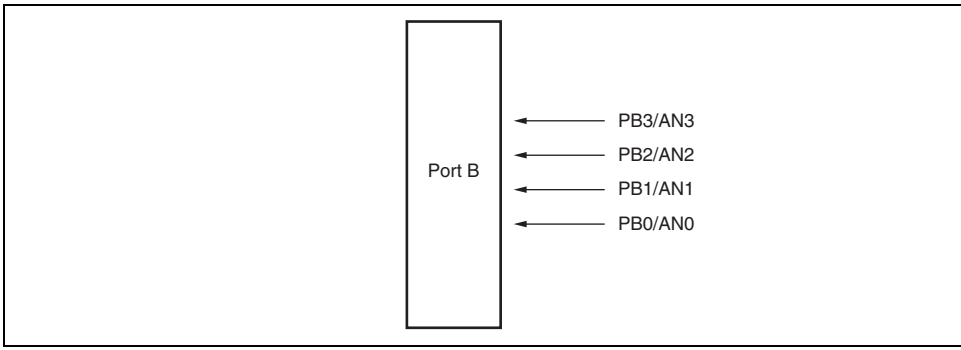


Figure 9.6 Port B Pin Configuration

Port B has the following register.

- Port data register B (PDRB)

2	PB2	—	R	However, if a port B pin is designated as an analo
1	PB1	—	R	channel by ADCSR in A/D converter, 0 is read.
0	PB0	—	R	

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

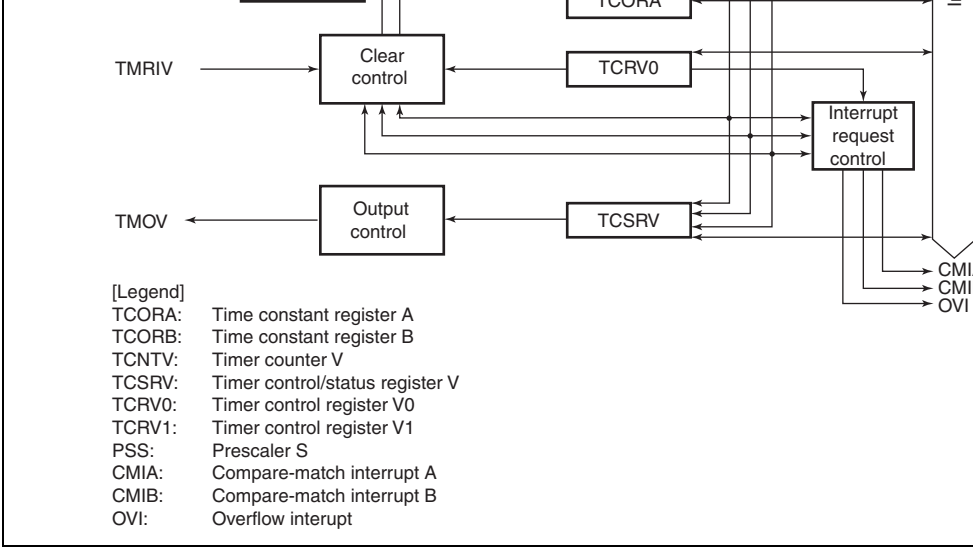


Figure 10.1 Block Diagram of Timer V

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.

TCORA and TCORB are initialized to H'FF.

10.3.3 Timer Control Register V0 (TCRV0)

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

Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	Compare Match Interrupt Enable B When this bit is set to 1, interrupt request from the bit in TCSRV is enabled.
6	CMIEA	0	R/W	Compare Match Interrupt Enable A When this bit is set to 1, interrupt request from the bit in TCSRV is enabled.
5	OVIE	0	R/W	Timer Overflow Interrupt Enable When this bit is set to 1, interrupt request from the in TCSRV is enabled.
4	CCLR1	0	R/W	Counter Clear 1 and 0
3	CCLR0	0	R/W	These bits specify the clearing conditions of TCN 00: Clearing is disabled 01: Cleared by compare match A 10: Cleared by compare match B 11: Cleared on the rising edge of the TMRIV pin. operation of TCNTV after clearing depends o in TCRV1.

CKS2	CKS1	CKS0	ICKS0	Description
0	0	0	—	Clock input prohibited
		1	0	Internal clock: counts on φ/4, falling e
			1	Internal clock: counts on φ/8, falling e
	1	0	0	Internal clock: counts on φ/16, falling
			1	Internal clock: counts on φ/32, falling
		1	0	Internal clock: counts on φ/64, falling
			1	Internal clock: counts on φ/128, fallin
1	0	0	—	Clock input prohibited
		1	—	External clock: counts on rising edge
	1	0	—	External clock: counts on falling edg
		1	—	External clock: counts on rising and edge

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

				These bits are always read as 1.
4	TVEG1	0	R/W	TRGV Input Edge Select
3	TVEG0	0	R/W	These bits select the TRGV input edge. 00: TRGV trigger input is prohibited 01: Rising edge is selected 10: Falling edge is selected 11: Rising and falling edges are both selected
2	TRGE	0	R/W	TCNTV starts counting up by the input of the e is 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 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 o
(CMFA or CMFB) is set to 1 when TCNTV matches TCORA or TCORB, respectivel
compare-match signal is generated in the last state in which the values match. Figure
shows the timing. An interrupt request is generated for the CPU when CMIEA or CM
TCRV0 is 1.
4. When a compare match A or B is generated, the TMOV responds with the output valu
selected by bits OS3 to OS0 in TCSRV. Figure 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.

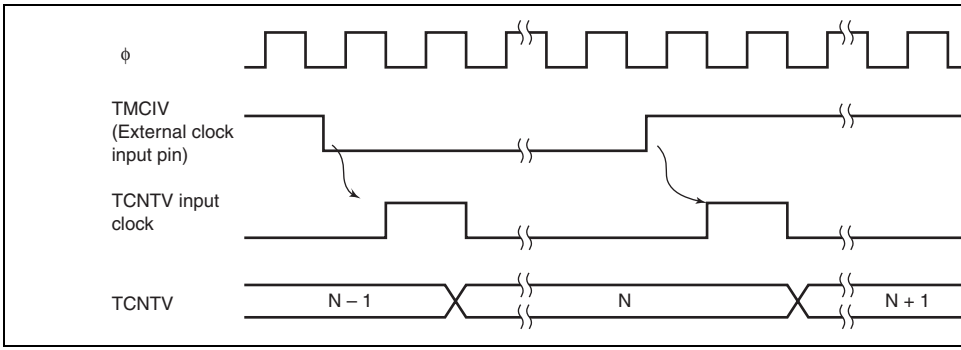


Figure 10.3 Increment Timing with External Clock

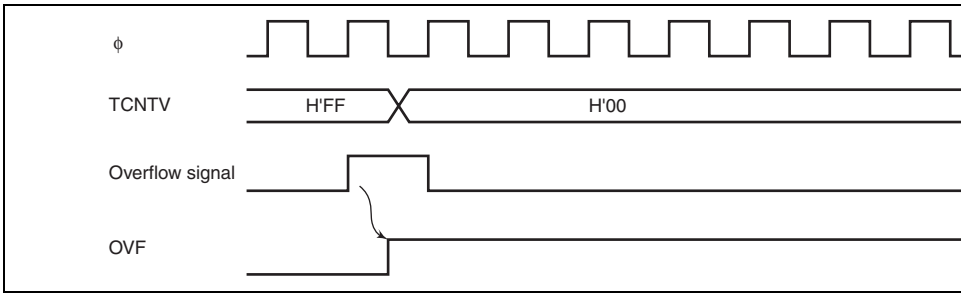


Figure 10.4 OVF Set Timing

Figure 10.5 CMFA and CMFB Set Timing

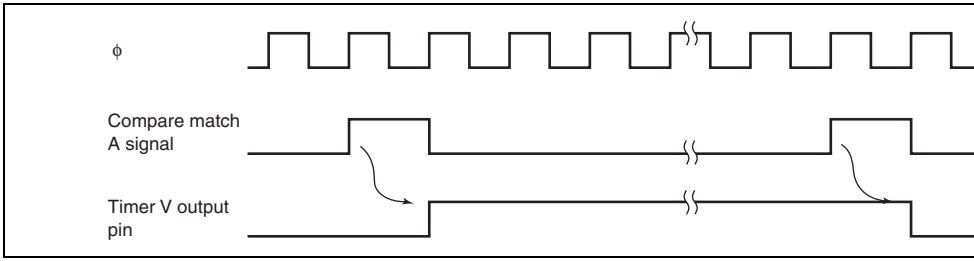


Figure 10.6 TMOV Output Timing

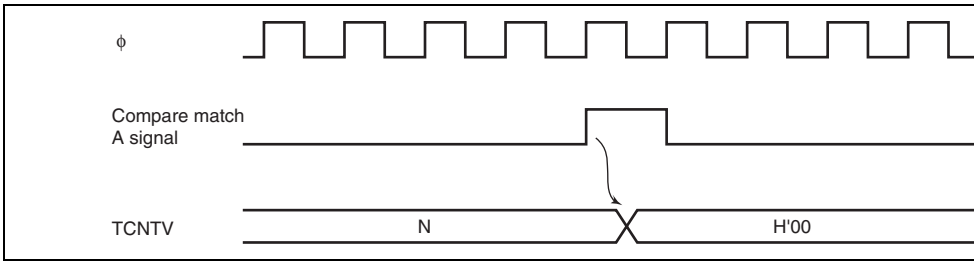


Figure 10.7 Clear Timing by Compare Match

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.

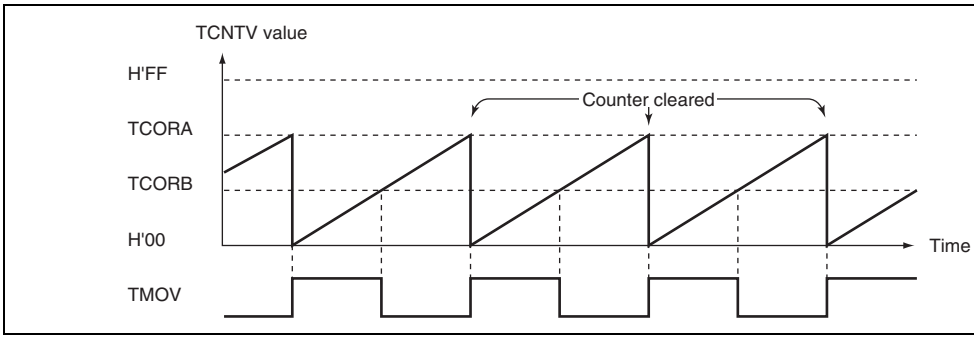


Figure 10.9 Pulse Output Example

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

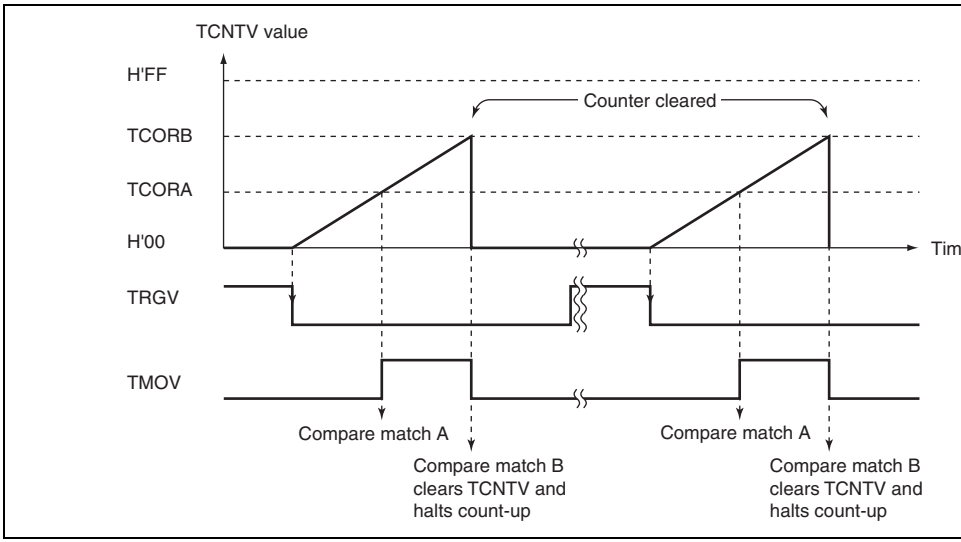


Figure 10.10 Example of Pulse Output Synchronized to TRGV Input

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 incremente switch between internal and external clocks.

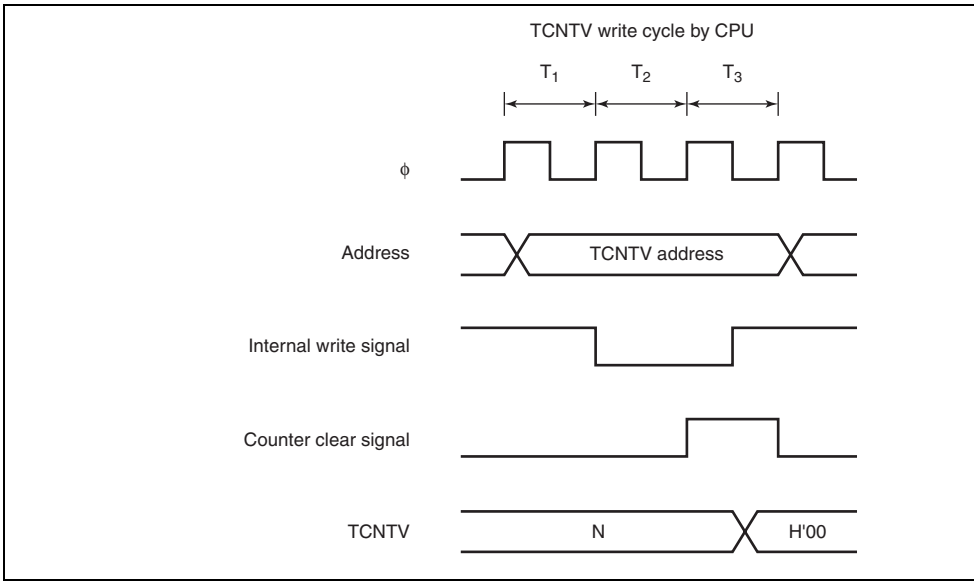


Figure 10.11 Contention between TCNTV Write and Clear

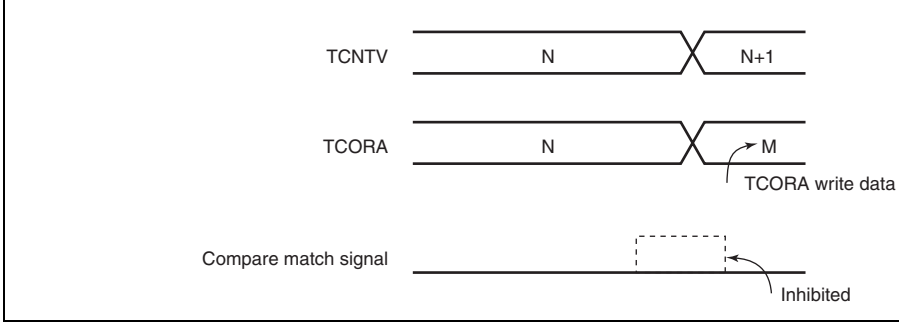


Figure 10.12 Contention between TCORA Write and Compare Match

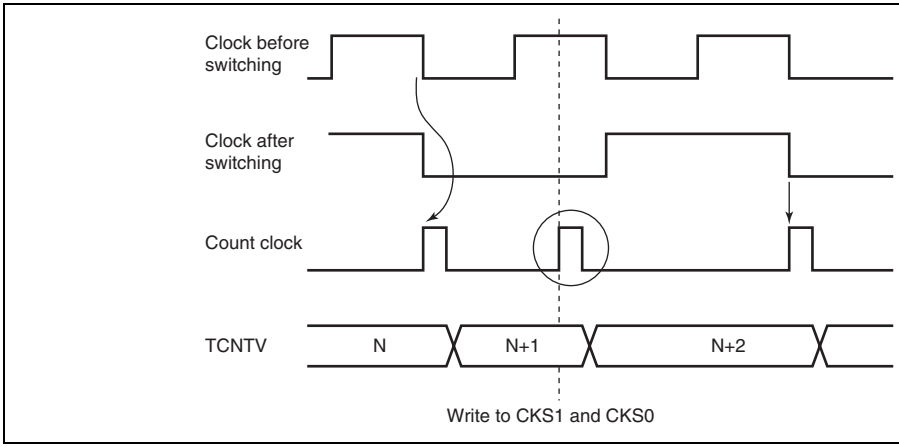
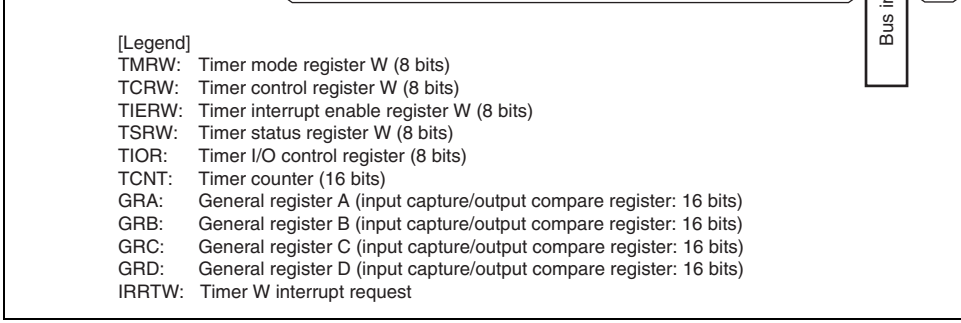


Figure 10.13 Internal Clock Switching and TCNTV Operation

- 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 of W.

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



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

Figure 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)

5	BUFEB	0	R/W	Buffer Operation B Selects the GRD function. 0: GRD operates as an input capture/output com 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 com 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 ou 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 ou 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 ou 1: PWM output

5	CKS1	0	R/W	Select the TCNT clock source.
4	CKS0	0	R/W	000: Internal clock: counts on ϕ
				001: Internal clock: counts on $\phi/2$
				010: Internal clock: counts on $\phi/4$
				011: Internal clock: counts on $\phi/8$
				1XX: Counts on rising edges of the external even
				When the internal clock source (ϕ) is selected, su sources are counted in subactive and subsleep m
3	TOD	0	R/W	Timer Output Level Setting D
				0: Output value is 0*
				1: Output value is 1*
2	TOC	0	R/W	Timer Output Level Setting C
				0: Output value is 0*
				1: Output value is 1*
1	TOB	0	R/W	Timer Output Level Setting B
				0: Output value is 0*
				1: Output value is 1*
0	TOA	0	R/W	Timer Output Level Setting A
				0: Output value is 0*
				1: Output value is 1*

Legend: X: Don't care.

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

4	—	1	—	
3	IMIED	0	R/W	Input Capture/Compare Match Interrupt Enable When this bit is set to 1, IMID interrupt requeste 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 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 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 flag in TSRW is enabled.

11.3.4 Timer Status Register W (TSRW)

TSRW shows the status of interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/W	Timer Overflow Flag [Setting condition] When TCNT overflows from H'FFFF to H'0000 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
6	—	1	—	Reserved
5	—	1	—	These bits are always read as 1.
4	—	1	—	

2	IMFC	0	R/W	Input Capture/Compare Match Flag C [Setting conditions] • TCNT = GRC when GRC functions as an outp compare register • The TCNT value is transferred to GRC by an i capture signal when GRC functions as an inpu capture register [Clearing condition] Read IMFC when IMFC = 1, then write 0 in IMFC
1	IMFB	0	R/W	Input Capture/Compare Match Flag B [Setting conditions] • TCNT = GRB when GRB functions as an outp compare register • The TCNT value is transferred to GRB by an i capture signal when GRB functions as an inpu register [Clearing condition] Read IMFB when IMFB = 1, then write 0 in IMFB
0	IMFA	0	R/W	Input Capture/Compare Match Flag A [Setting conditions] • TCNT = GRA when GRA functions as an outp compare register • The TCNT value is transferred to GRA by an i capture signal when GRA functions as an inpu register [Clearing condition] Read IMFA when IMFA = 1, then write 0 in IMFA

5	IOB1	0	R/W	I/O Control B1 and B0
4	IOB0	0	R/W	When IOB2 = 0, 00: No output at compare match 01: 0 output to the FTIOB pin at GRB compare m 10: 1 output to the FTIOB pin at GRB compare m 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
1	IOA1	0	R/W	I/O Control A1 and A0
0	IOA0	0	R/W	When IOA2 = 0, 00: No output at compare match 01: 0 output to the FTIOA pin at GRA compare m 10: 1 output to the FTIOA pin at GRA compare m 11: Output toggles to the FTIOA pin at GRA com match When IOA2 = 1, 00: Input capture at rising edge of the FTIOA pin 01: Input capture at falling edge of the FTIOA pi 1X: Input capture at rising and falling edges of th pin

Legend: X: Don't care.

5	IOD1	0	R/W	I/O Control D1 and D0
4	IOD0	0	R/W	When IOD2 = 0, 00: No output at compare match 01: 0 output to the FTIOD pin at GRD compare m 10: 1 output to the FTIOD pin at GRD compare m 11: Output toggles to the FTIOD pin at GRD comp match When IOD2 = 1, 00: Input capture at rising edge at the FTIOD pin 01: Input capture at falling edge at the FTIOD pin 1X: Input capture at rising and falling edges at the 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	IOC1	0	R/W	I/O Control C1 and C0
0	IOC0	0	R/W	When IOC2 = 0, 00: No output at compare match 01: 0 output to the FTIOC pin at GRC compare m 10: 1 output to the FTIOC pin at GRC compare m 11: Output toggles to the FTIOC pin at GRC comp match When IOC2 = 1, 00: Input capture to GRC at rising edge of the FT 01: Input capture to GRC at falling edge of the FT 1X: Input capture to GRC at rising and falling edg FTIOC pin

Legend: X: Don't care.

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

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

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

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

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

GRA to GRD must be written or read in 16-bit units; 8-bit access is not allowed. GRA to
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
in TIERW is set to 1, an interrupt request is generated. Figure 11.2 shows free-running co

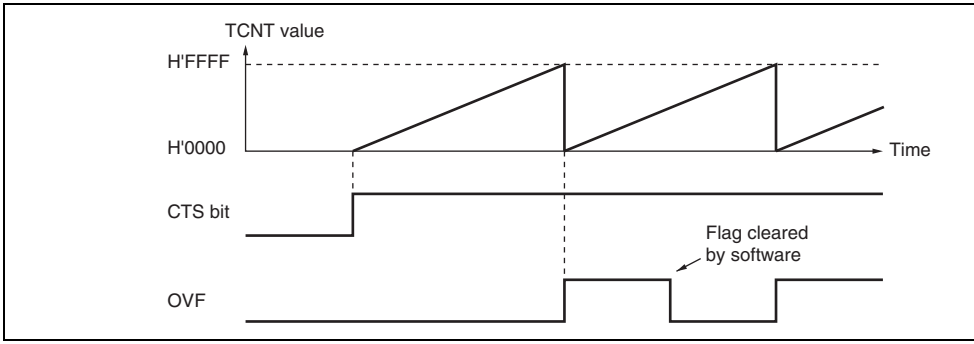


Figure 11.2 Free-Running Counter Operation

Periodic counting operation can be performed when GRA is set as an output compare reg
bit CCLR in TCRW is set to 1. When the count matches GRA, TCNT is cleared to H'000
IMFA flag in TSRW is set to 1. If the corresponding IMIEA bit in TIERW is set to 1, an
request is generated. TCNT continues counting from H'0000. Figure 11.3 shows periodic
counting.

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.

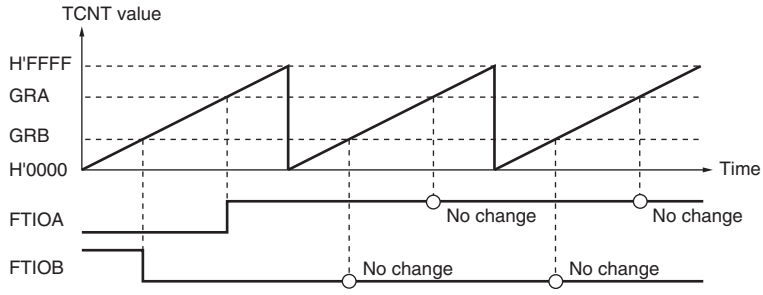


Figure 11.4 0 and 1 Output Example (TOA = 0, TOB = 1)

Figure 11.5 shows an example of toggle output when TCNT operates as a free-running c
and toggle output is selected for both compare match A and B.

Figure 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.

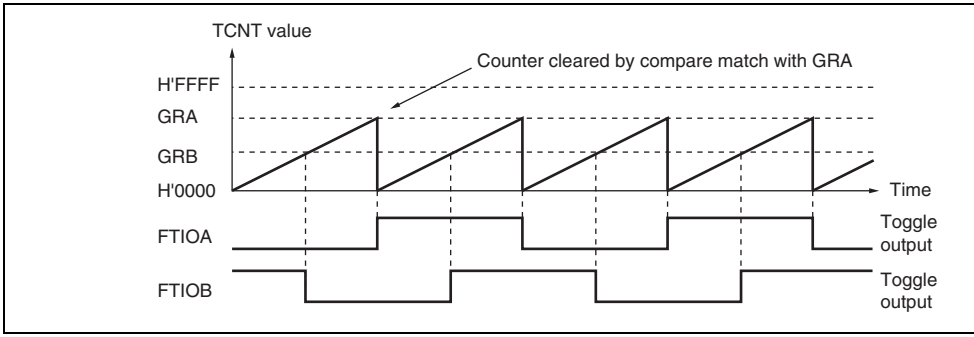


Figure 11.6 Toggle Output Example (TOA = 0, TOB = 1)

The TCNT value can be captured into a general register (GRA, GRB, GRC, or GRD) wh
signal level changes at an input-capture pin (FTIOA, FTIOB, FTIOC, or FTIOD). Captur
take place on the rising edge, falling edge, or both edges. By using the input-capture func
pulse width and periods can be measured. Figure 11.7 shows an example of input capture
both edges of FTIOA and the falling edge of FTIOB are selected as capture edges. TCNT
as a free-running counter.

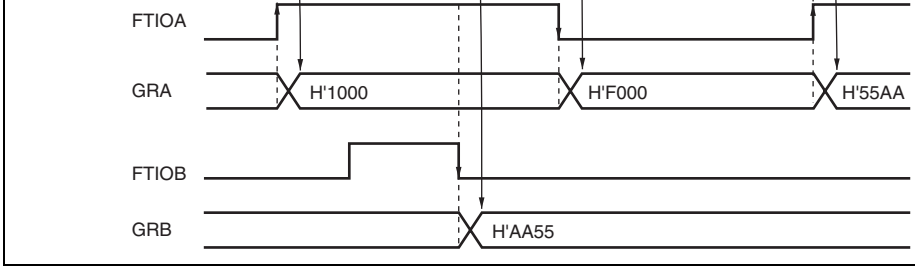


Figure 11.7 Input Capture Operating Example

Figure 11.8 shows an example of buffer operation when the GRA is set as an input-capt
register and GRC is set as the buffer register for GRA. TCNT operates as a free-running
and FTIOA captures both rising and falling edge of the input signal. Due to the buffer o
the GRA value is transferred to GRC by input-capture A and the TCNT value is stored i

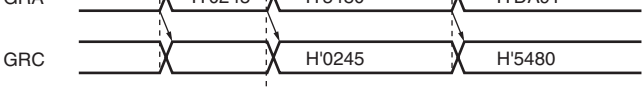


Figure 11.8 Buffer Operation Example (Input Capture)

11.4.2 PWM Operation

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

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

Figure 11.9 PWM Mode Example (1)

Figure 11.10 shows another example of operation in PWM mode. The output signals go
TCNT is cleared at compare match A, and the output signals go to 1 at compare match E
D (TOB, TOC, and TOD = 0: initial output values are set to 1).

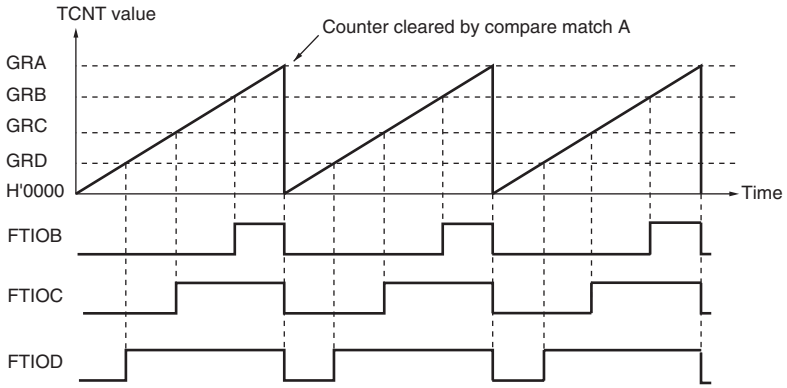


Figure 11.10 PWM Mode Example (2)

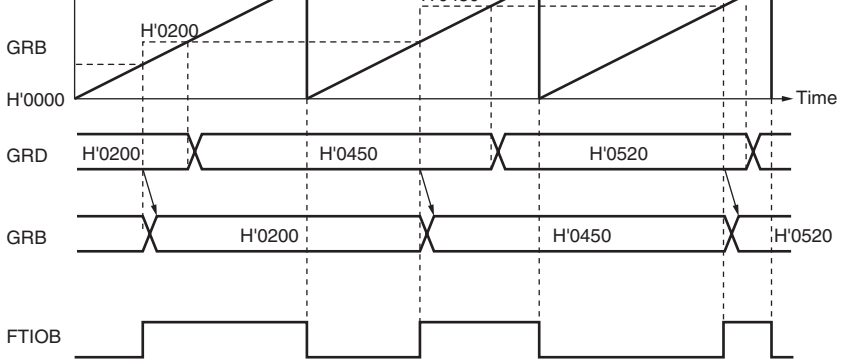


Figure 11.11 Buffer Operation Example (Output Compare)

Figures 11.12 and 11.13 show examples of the output of PWM waveforms with duty cycl
and 100%.

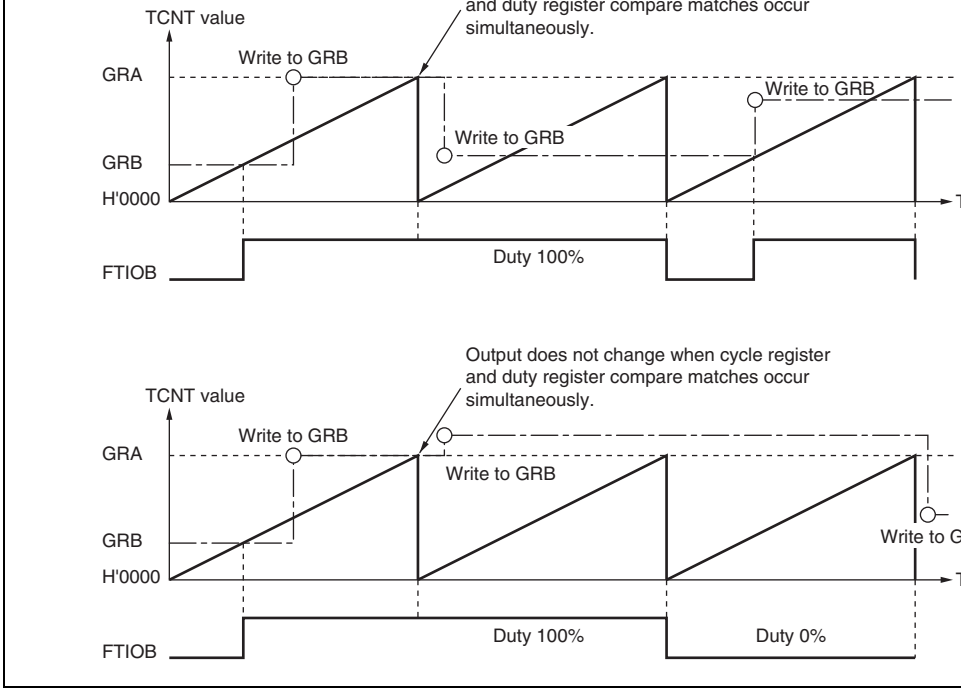


Figure 11.12 PWM Mode Example
(TOB, TOC, and TOD = 0: initial output values are set to 0)

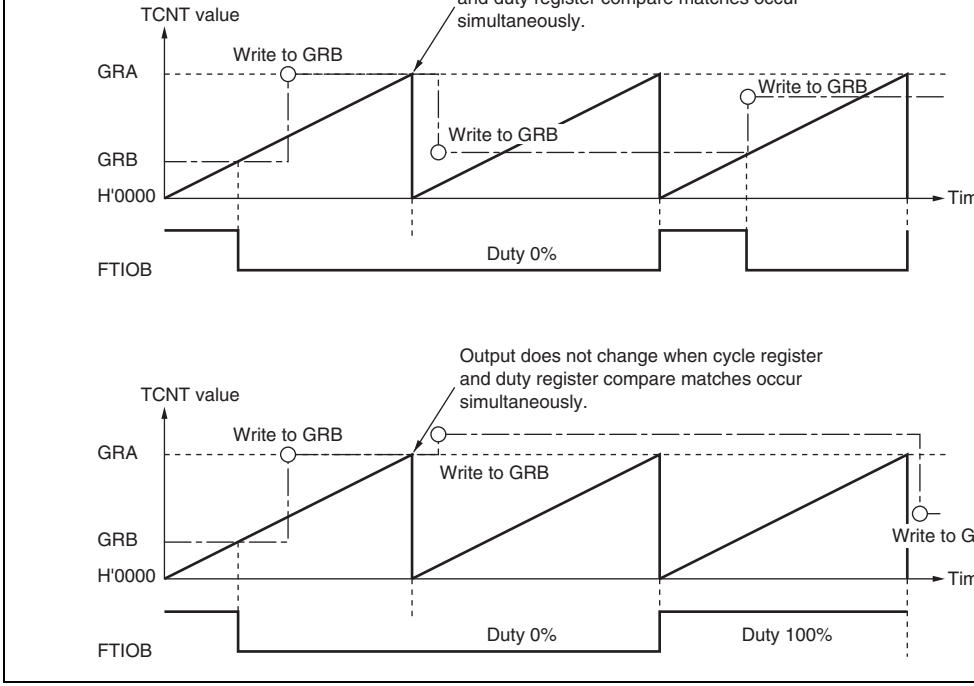


Figure 11.13 PWM Mode Example
(TOB, TOC, and TOD = 1: initial output values are set to 1)

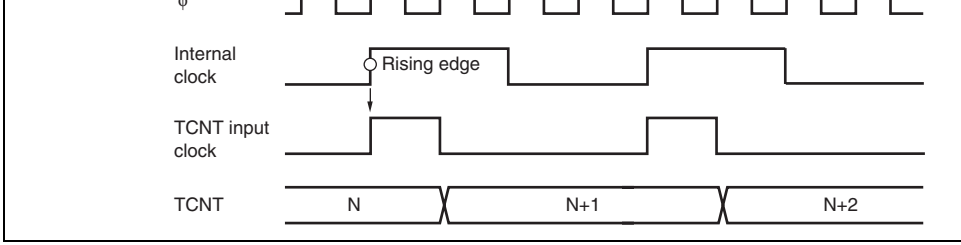


Figure 11.14 Count Timing for Internal Clock Source

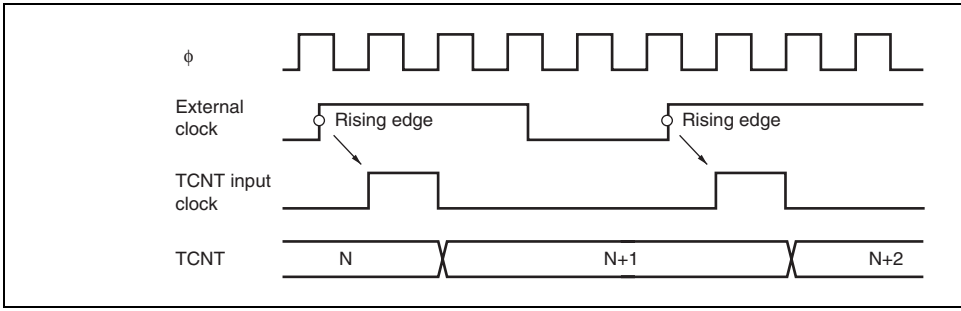


Figure 11.15 Count Timing for External Clock Source

11.5.2 Output Compare Output Timing

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

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

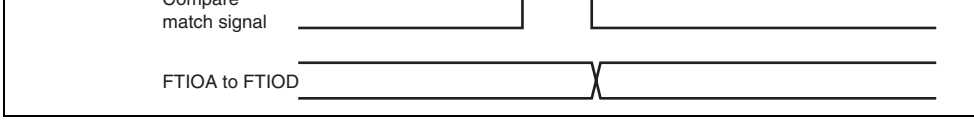


Figure 11.16 Output Compare Output Timing

11.5.3 Input Capture Timing

Input capture on the rising edge, falling edge, or both edges can be selected through settin
TIOR0 and TIOR1. Figure 11.17 shows the timing when the falling edge is selected. The
width of the input capture signal must be at least two system clock (ϕ) cycles; shorter puls
not be detected correctly.

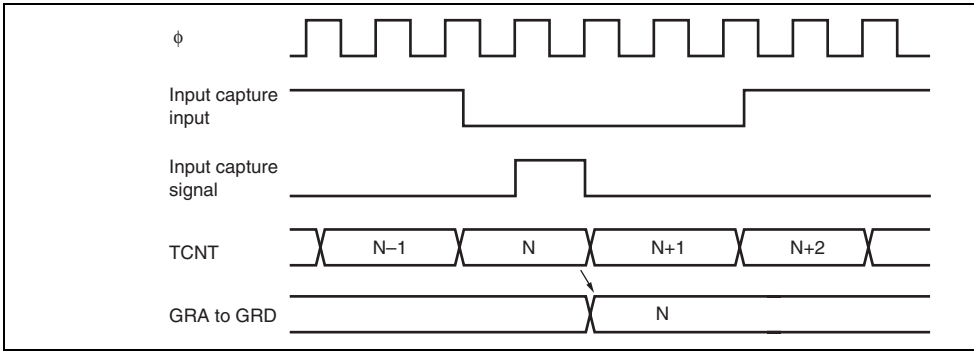


Figure 11.17 Input Capture Input Signal Timing

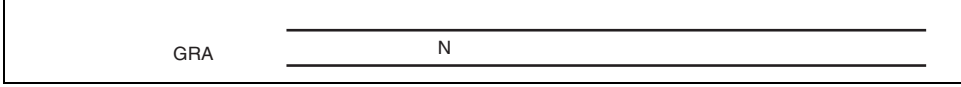


Figure 11.18 Timing of Counter Clearing by Compare Match

11.5.5 Buffer Operation Timing

Figures 11.19 and 11.20 show the buffer operation timing.

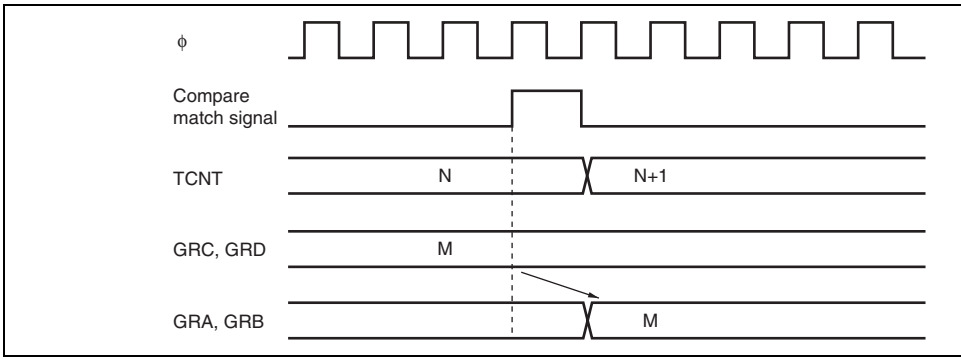


Figure 11.19 Buffer Operation Timing (Compare Match)

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, th
corresponding IMFA, IMFB, IMFC, or IMFD flag is set to 1 when TCNT matches the ge
register.

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

Figure 11.21 shows the timing of the IMFA to IMFD flag setting at compare match.

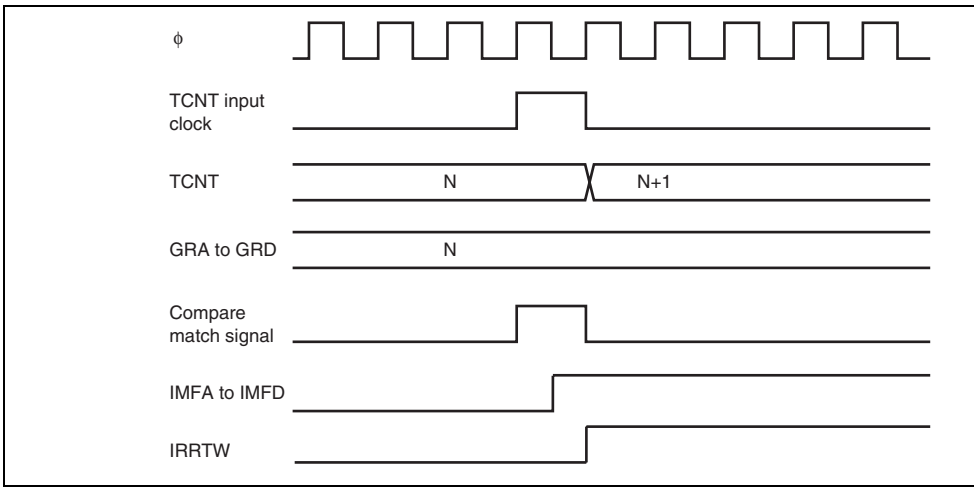


Figure 11.21 Timing of IMFA to IMFD Flag Setting at Compare Match

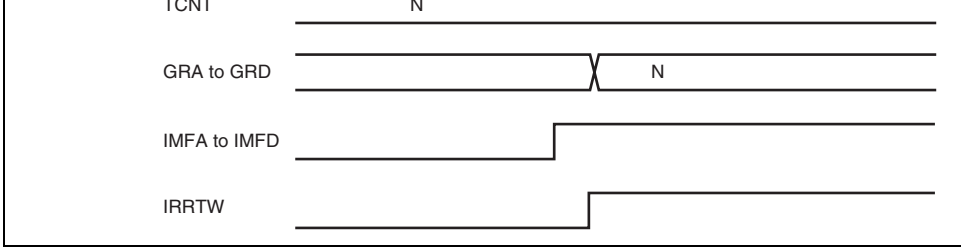


Figure 11.22 Timing of IMFA to IMFD Flag Setting at Input Capture

11.5.8 Timing of Status Flag Clearing

When the CPU reads a status flag while it is set to 1, then writes 0 in the status flag, the is cleared. Figure 11.23 shows the status flag clearing timing.

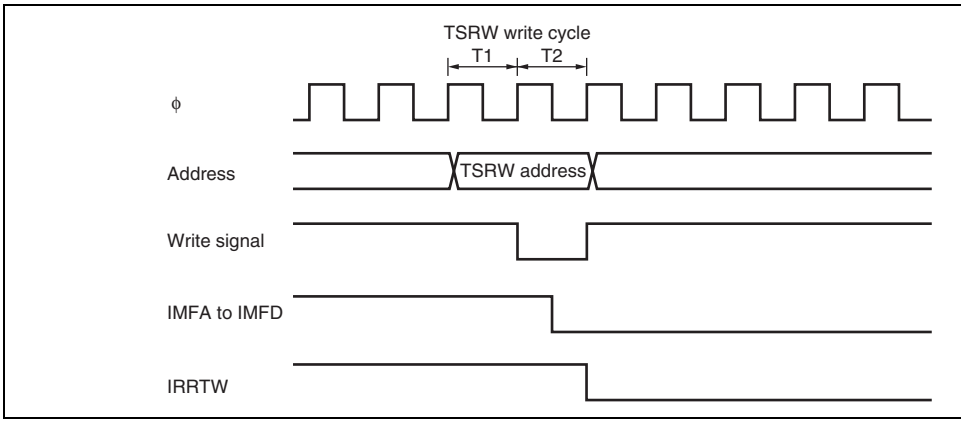


Figure 11.23 Timing of Status Flag Clearing by CPU

precedence.

3. Depending on the timing, TCNT may be incremented by a switch between different in clock sources. When TCNT is internally clocked, an increment pulse is generated fron rising edge of an internal clock signal, that is divided system clock (ϕ). Therefore, as s figure 11.25 the switch is from a low clock signal to a high clock signal, the switchov as a rising edge, causing TCNT to increment.
4. If timer W enters module standby mode while an interrupt request is generated, the in request cannot be cleared. Before entering module standby mode, disable interrupt rec

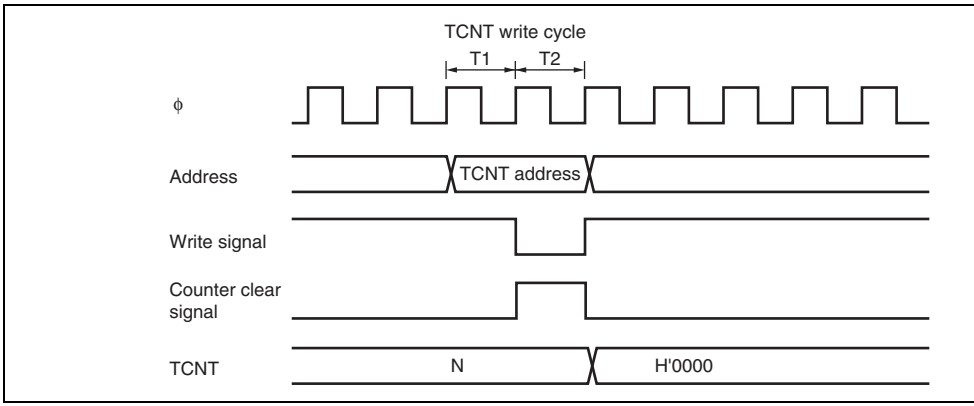


Figure 11.24 Contention between TCNT Write and Clear

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 FTIO 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

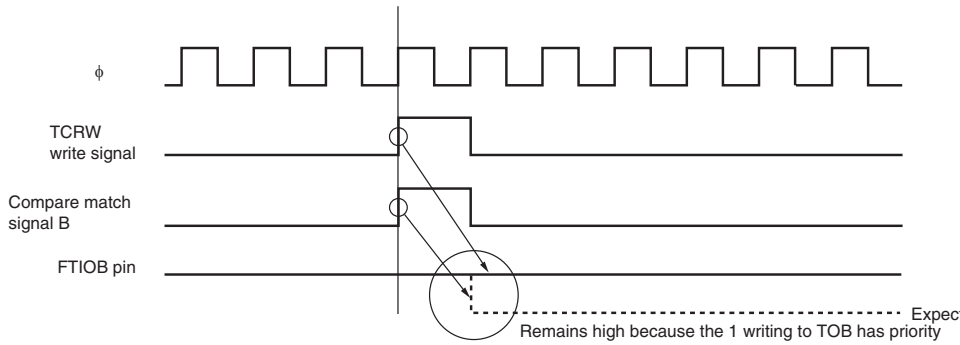


Figure 11.26 When Compare Match and Bit Manipulation Instruction to TCI Occur at the Same Timing

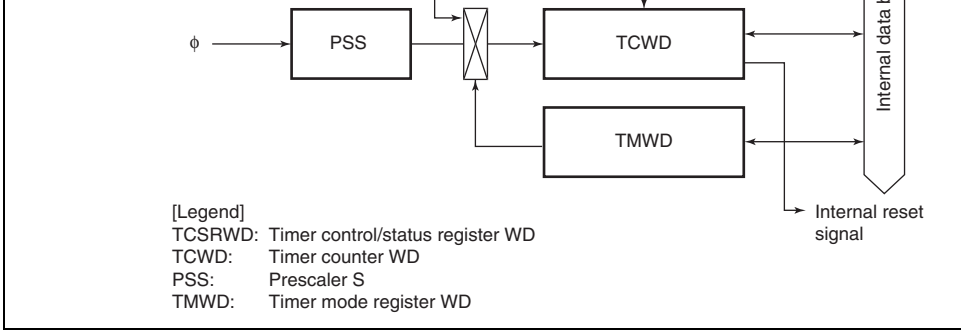


Figure 12.1 Block Diagram of Watchdog Timer

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.

12.2 Register Descriptions

The watchdog timer has the following registers.

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

6	TCWE	0	R/W	Timer Counter WD Write Enable TCWD can be written when the TCWE bit is set to When writing data to this bit, the value for bit 7 mu
5	B4WI	1	R/W	Bit 4 Write Inhibit The TCSRWE bit can be written only when the wr of the B4WI bit is 0. This bit is always read as 1.
4	TCSRWE	0	R/W	Timer Control/Status Register WD Write Enable The WDON and WRST bits can be written when t TCSRWE bit is set to 1. When writing data to this bit, the value for bit 5 mu
3	B2WI	1	R/W	Bit 2 Write Inhibit This bit can be written to the WDON bit only when write value of the B2WI bit is 0. This bit is always read as 1.
2	WDON	0	R/W	Watchdog Timer On TCWD starts counting up when WDON is set to 1 halts when WDON is cleared to 0. [Setting condition] When 1 is written to the WDON bit while writing 0 B2WI bit when the TCSRWE bit = 1 [Clearing conditions] • Reset by RES pin • When 0 is written to the WDON bit while writin 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 when value of the B0WI bit is 0. This bit is always read

12.2.2 Timer Counter WD (TCWD)

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

12.2.3 Timer Mode Register WD (TMWD)

TMWD selects the input clock.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	CKS3	1	R/W	Clock Select 3 to 0
2	CKS2	1	R/W	Select the clock to be input to TCWD.
1	CKS1	1	R/W	1000: Internal clock: counts on ϕ/64
0	CKS0	1	R/W	1001: Internal clock: counts on ϕ/128 1010: Internal clock: counts on ϕ/256 1011: Internal clock: counts on ϕ/512 1100: Internal clock: counts on ϕ/1024 1101: Internal clock: counts on ϕ/2048 1110: Internal clock: counts on ϕ/4096 1111: Internal clock: counts on ϕ8192 0XXX: Internal oscillator For the internal oscillator overflow periods, see 17, Electrical Characteristics.

Legend: X: Don't care.

Figure 12.2 shows an example of watchdog timer operation.

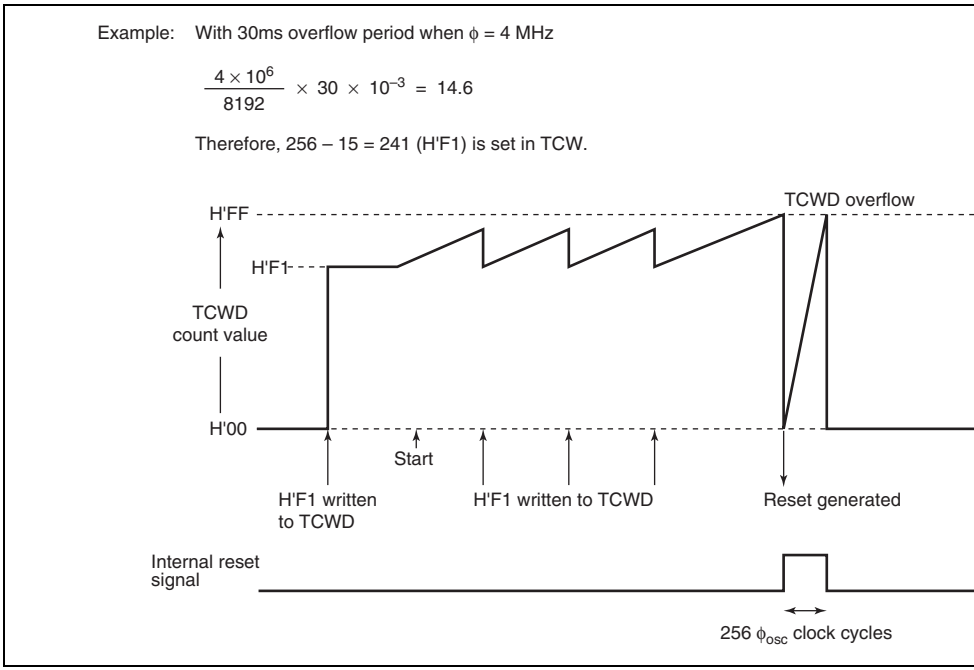


Figure 12.2 Watchdog Timer Operation Example

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 sour
- Six interrupt sources

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

Asynchronous mode

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

Clocked synchronous mode

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

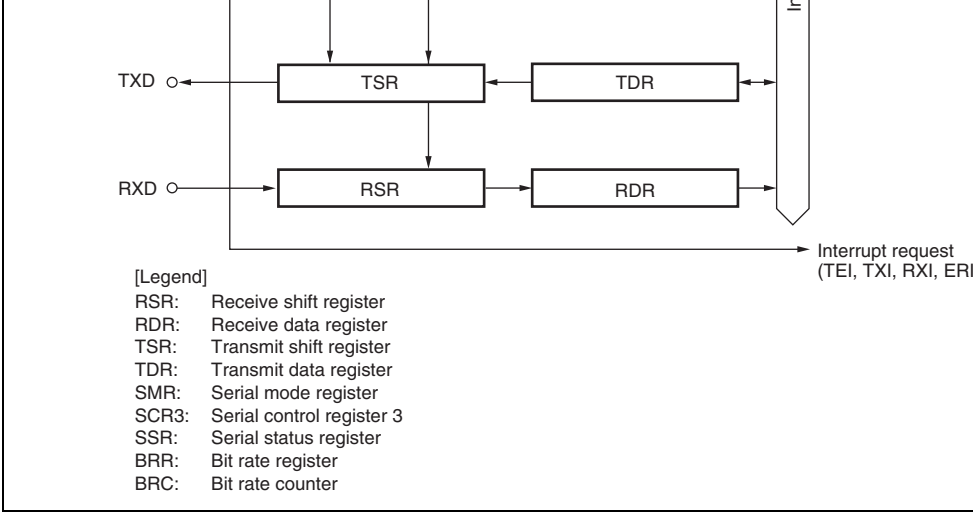


Figure 13.1 Block Diagram of SCI3

13.3 Register Descriptions

The SCI3 has the following registers.

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

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

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.

				mode)
				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 asynchron mode)
				When this bit is set to 1, the parity bit is a transmit data before transmission, and th is checked in reception.
4	PM	0	R/W	Parity Mode (enabled only when the PE b asynchronous mode)
				0: Selects even parity.
				1: Selects odd parity.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchro mode)
				Selects the stop bit length in transmissior
				0: 1 stop bit
				1: 2 stop bits
				For reception, only the first stop bit is che regardless of the value in the bit. If the se bit is 0, it is treated as the start bit of the transmit character.
2	MP	0	R/W	Multiprocessor Mode
				When this bit is set to 1, the multiprocess communication function is enabled. The P PM bit settings are invalid. In clocked syn mode, this bit should be cleared to 0.

				Rate Register (BRR). n is the decimal representation of the value of n in BRR (s 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 is enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI inte requests are enabled.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is e
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enab

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

				• When data is transferred from TDR to T [Clearing conditions] • When 0 is written to TDRE after reading = 1 • When the transmit data is written to TD
6	RDRF	0	R/W	Receive Data Register Full Indicates that the received data is stored in [Setting condition] • When serial reception ends normally ar receive data is transferred from RSR to [Clearing conditions] • When 0 is written to RDRF after reading = 1 • When data is read from RDR
5	OER	0	R/W	Overrun Error [Setting condition] • When an overrun error occurs in recept [Clearing condition] • When 0 is written to OER after reading

				Reception [Clearing condition] • When 0 is written to PER after reading
2	TEND	1	R	Transmit End [Setting conditions] • When the TE bit in SCR3 is 0 • When TDRE = 1 at transmission of the a 1-byte serial transmit character [Clearing conditions] • When 0 is written to TEND after readin = 1 • When the transmit data is written to TD
1	MPBR	0	R	Multiprocessor Bit Receive MPBR stores the multiprocessor bit in the character data. When the RE bit in SCR3 to 0, its previous state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be the transmit character data.

[Asynchronous Mode]

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

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

[Clocked Synchronous Mode]

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

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

1200	0	51	0.16	0	54	–0.70	0	65	0.00	0	77
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	19
9600	0	6	–6.99	0	6	–2.48	0	7	0.00	0	9
19200	0	2	8.51	0	2	13.78	0	3	0.00	0	4
31250	0	1	0.00	0	1	4.86	0	1	22.88	0	2
38400	0	1	–18.62	0	1	–14.67	0	1	0.00	—	—

Legend:

— : A setting is available but error occurs

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

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

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

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

Legend:

—: A setting is available but error occurs.

Table 13.3 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N	ϕ (MHz)	Maximum Bit Rate (bit/s)	n
2	62500	0	0	7.3728	230400	0
2.097152	65536	0	0	8	250000	0
2.4576	76800	0	0	9.8304	307200	0
3	93750	0	0	10	312500	0
3.6864	115200	0	0	12	375000	0
4	125000	0	0	12.288	384000	0
4.9152	153600	0	0	14	437500	0
5	156250	0	0	14.7456	460800	0
6	187500	0	0	16	500000	0
6.144	192000	0	0			

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.

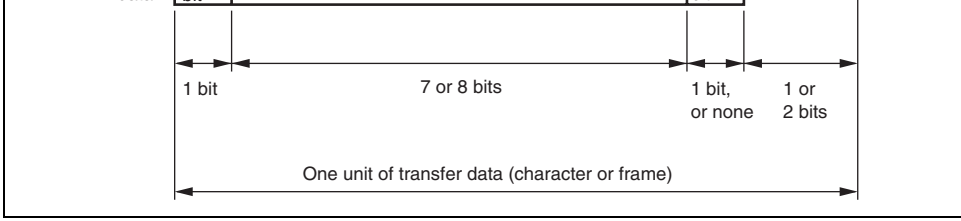


Figure 13.2 Data Format in Asynchronous Communication

13.4.1 Clock

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

When the SCI3 is operated on an internal clock, the clock can be output from the SCK3
frequency of the clock output in this case is equal to the bit rate, and the phase is such th
rising edge of the clock is in the middle of the transmit data, as shown in figure 13.3.

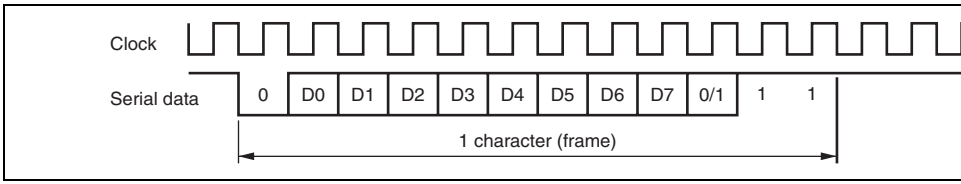
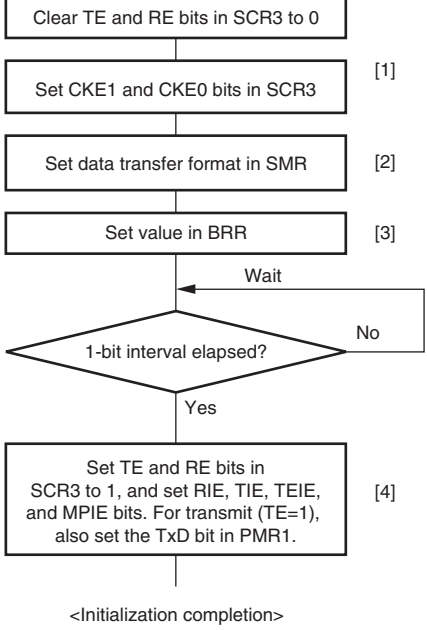


**Figure 13.3 Relationship between Output Clock and Transfer Data Phas
(Asynchronous Mode) (Example with 8-Bit Data, Parity, Two Stop Bits)**



When the clock output is selected in asynchronous mode, clock is output immediately after CKE1 and CKE0 settings are made. When the clock output is selected at reception in clocked synchronous mode, clock is output immediately after CKE1, CKE0, and RE are set to 1.

[2] Set the data transfer format in SMR.

[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR3 to 1. RE settings enable the RXD pin to be used. For transmission, set the TXD bit in PMR1 to 1 to enable the TXD output pin to be used. Also set the RIE, TIE, TEIE, and MPIE bits, depending on whether interrupts are required. In asynchronous mode, the bits are marked at transmission and idled at reception to wait for the start bit.

Figure 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, a serial transmission of the next frame is started.
5. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then state" is entered, in which 1 is output. If the TEIE bit in SCR3 is set to 1 at this time, interrupt request is generated.
6. Figure 13.6 shows a sample flowchart for transmission in asynchronous mode.

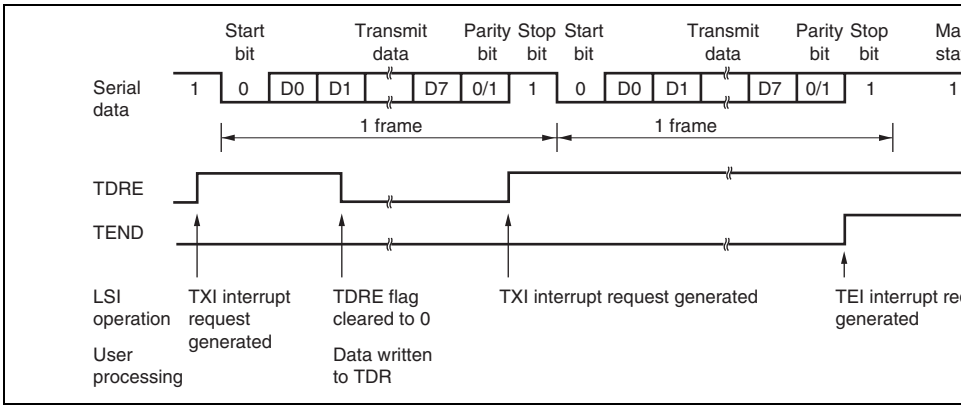
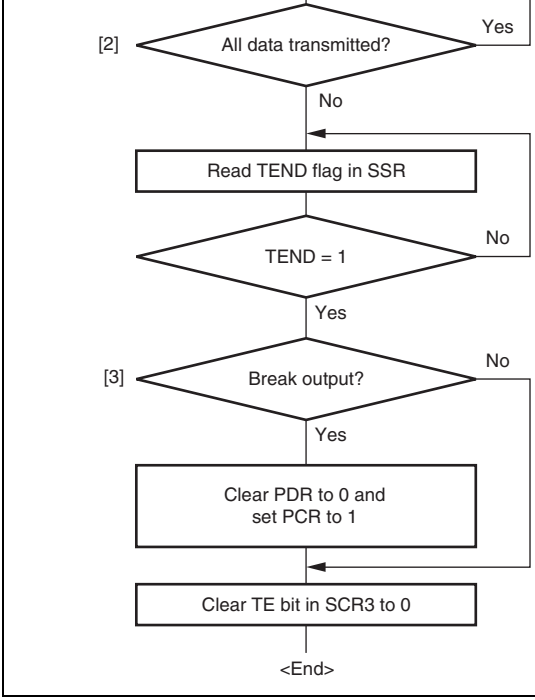


Figure 13.5 Example SCI3 Operation in Transmission in Asynchronous M (8-Bit Data, Parity, One Stop Bit)



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

Figure 13.6 Sample Serial Transmission Flowchart (Asynchronous Mode)

3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transfer RDR. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is gener
4. If a framing error is detected (when the stop bit is 0), the FER bit in SSR is set to 1 a data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an ERI inte request is generated.
5. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive d transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt re generated. Continuous reception is possible because the RXI interrupt routine reads t data transferred to RDR before reception of the next receive data has been completed

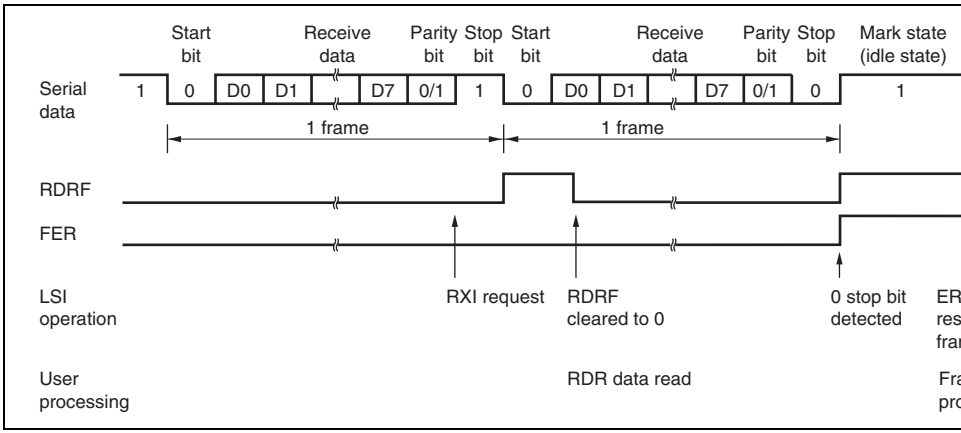
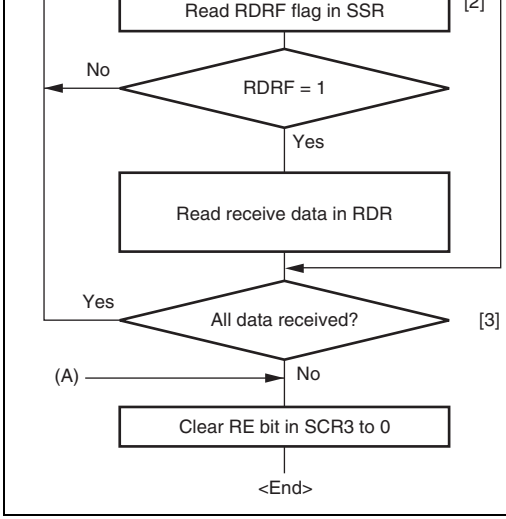


Figure 13.7 Example SCI3 Operation in Reception in Asynchronous Mod (8-Bit Data, Parity, One Stop Bit)

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

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



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

Figure 13.8 Sample Serial Data Reception Flowchart (Asynchronous mode)

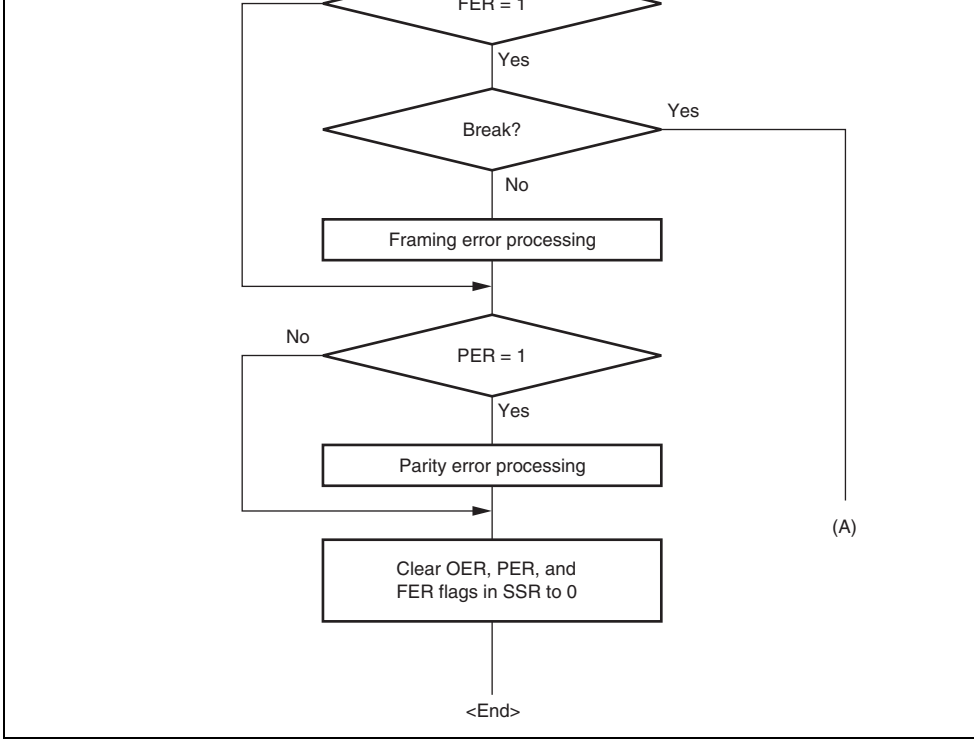


Figure 13.8 Sample Serial Reception Data Flowchart (2)

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

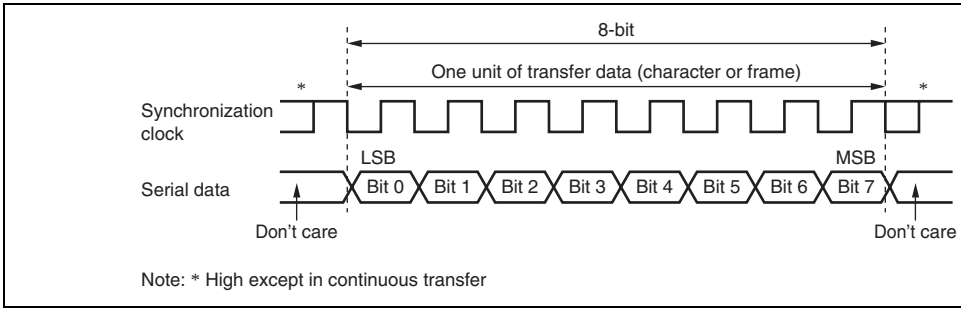


Figure 13.9 Data Format in Clocked Synchronous Communication

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 bit in SMR and CKE0 and CKE1 bits in SCR3. When the SCI3 is operated on an interna the serial clock is output from the SCK3 pin. Eight serial clock pulses are output in the t one character, and when no transfer is performed the clock is fixed high.

13.5.2 SCI3 Initialization

Before transmitting and receiving data, the SCI3 should be initialized as described in a s flowchart in figure 13.4.

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

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

Figure 13.11 shows a sample flowchart for serial data transmission. Even if the TDRE fla
cleared to 0, transmission will not start while a receive error flag (OER, FER, or PER) is
Make sure that the receive error flags are cleared to 0 before starting transmission.

Figure 13.10 Example of SCI3 Operation in Transmission in Clocked Synchrono

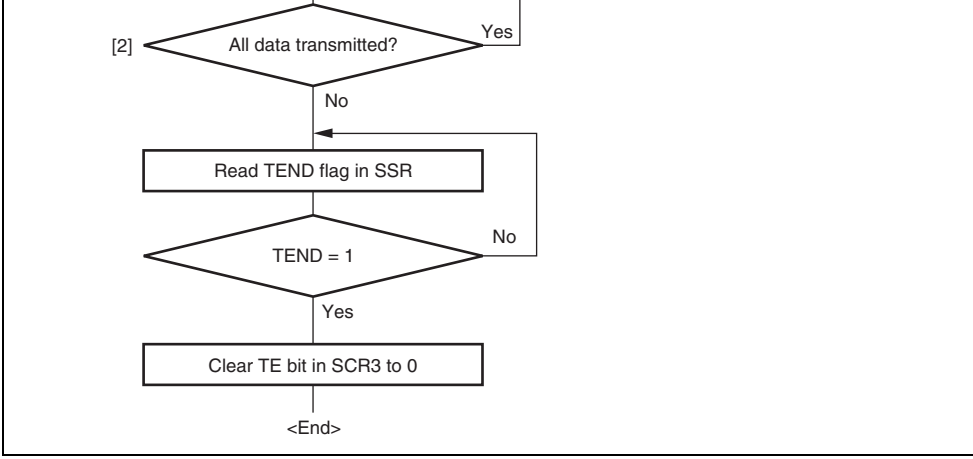


Figure 13.11 Sample Serial Transmission Flowchart (Clocked Synchronous M

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
generated.

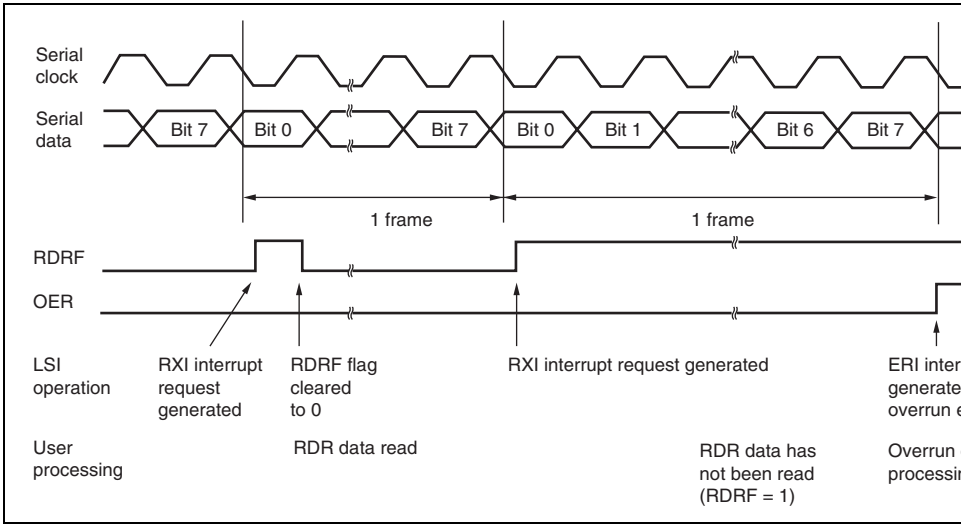
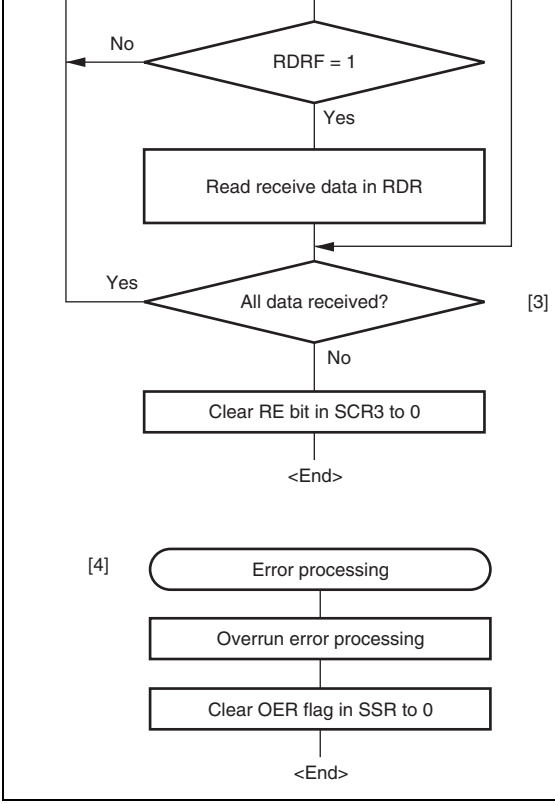


Figure 13.12 Example of SCI3 Reception Operation in Clocked Synchronous

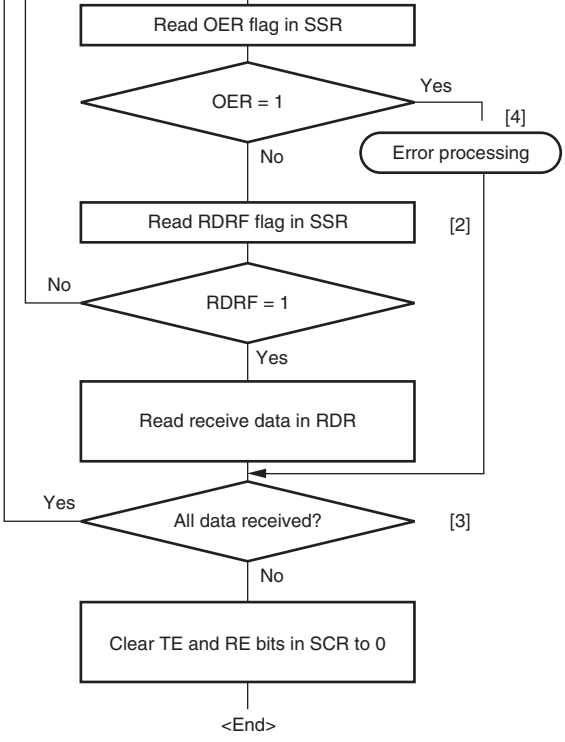
Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear th
FER, PER, and RDRF bits to 0 before resuming reception. Figure 13.13 shows a sample
for serial data reception.



cleared to 0.

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

Figure 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 15.13 shows an example of inter-processor
communication using the multiprocessor format. The transmitting station first sends the
of the receiving station with which it wants to perform serial communication as data wit
multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit
When data with a 1 multiprocessor bit is received, the receiving station compares that da
own ID. The station whose ID matches then receives the data sent next. Stations whose
match continue to skip data until data with a 1 multiprocessor bit is again received.

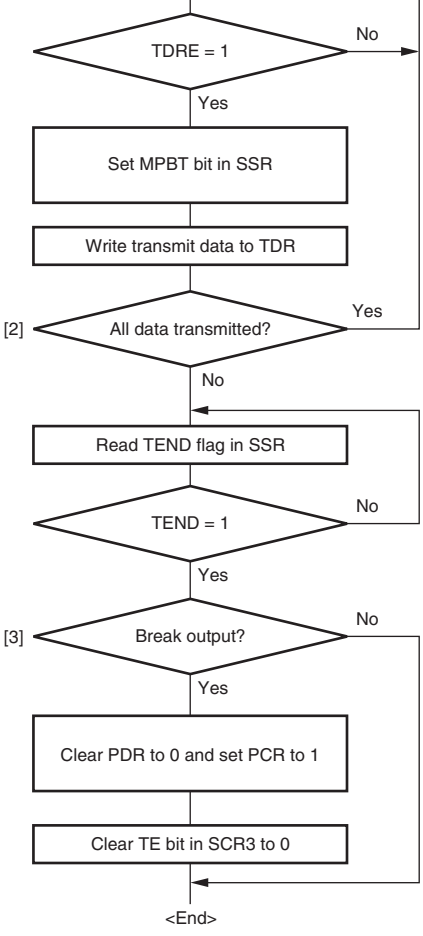
The SCI3 uses the MPIE bit in SCR3 to implement this function. When the MPIE bit is
transfer of receive data from RSR to RDR, error flag detection, and setting the SSR statu
RDRF, FER, and OER to 1, are inhibited until data with a 1 multiprocessor bit is receive
reception of a receive character with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

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

specification receiving station specified by ID

[Legend]
MPB: Multiprocessor bit

Figure 13.15 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)



TDR, the TDRE flag is automatically cleared to 0.

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

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

Figure 13.16 Sample Multiprocessor Serial Transmission Flowchart

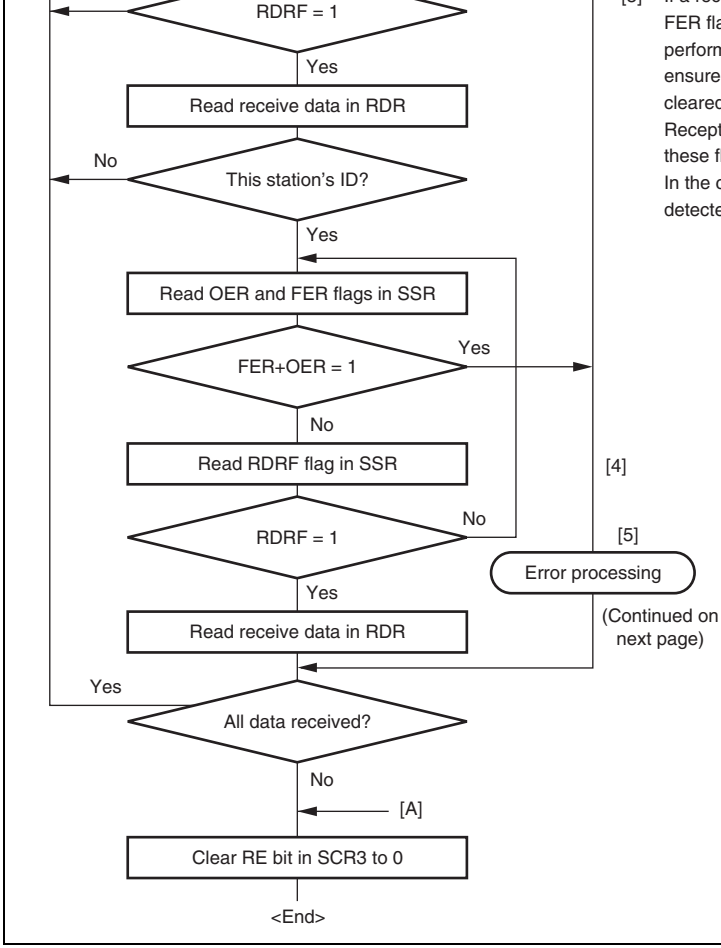


Figure 13.17 Sample Multiprocessor Serial Reception Flowchart (1)

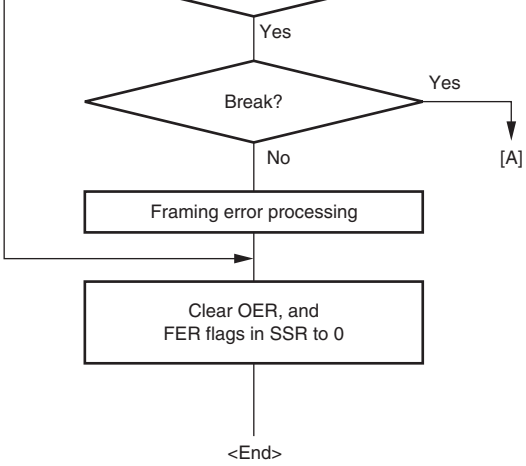
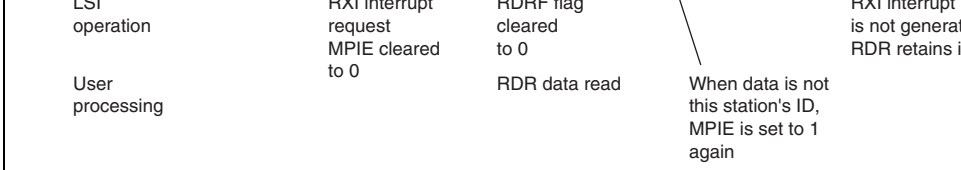
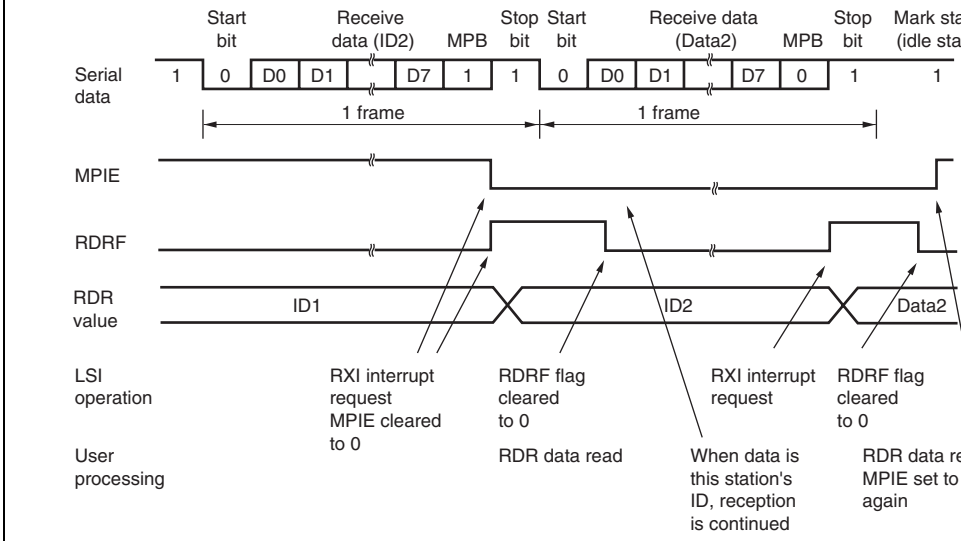


Figure 13.17 Sample Multiprocessor Serial Reception Flowchart (2)



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



(b) When data matches this receiver's ID

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

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

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

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

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.

[2N N]

... Formula (1)

Where N : Ratio of bit rate to clock (N = 16)
D : Clock duty (D = 0.5 to 1.0)
L : Frame length (L = 9 to 12)
F : Absolute value of clock rate deviation

Assuming values of F (absolute value of clock rate deviation) = 0 and D (clock duty) = 0.
formula (1), the reception margin can be given by the formula.

$M = \{0.5 - 1/(2 \times 16)\} \times 100\ [\%] = 46.875\%$

However, this is only the computed value, and a margin of 20% to 30% should be allowe
system design.

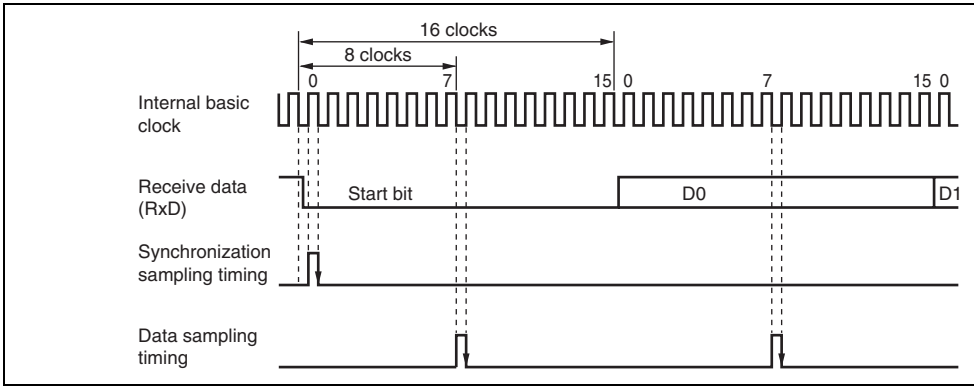
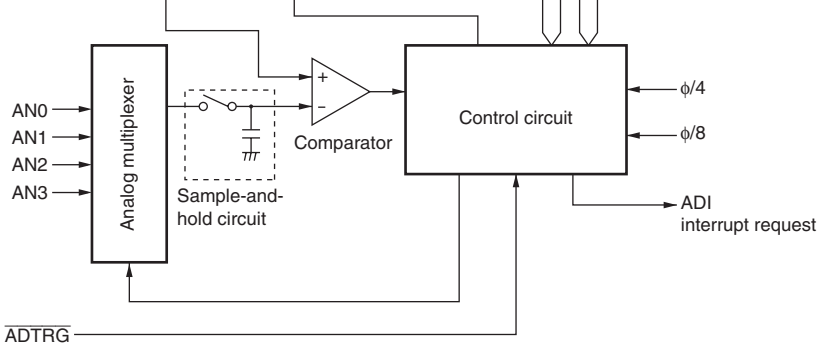


Figure 13.19 Receive Data Sampling Timing in Asynchronous Mode

- Conversion time: at least 4.4 µs per channel (at 16 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



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

Figure 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	$\overline{\text{ADTRG}}$	Input	External trigger input pin for star 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 l
directly from the CPU, however the lower byte should be read via a temporary register. T
temporary register contents are transferred from the ADDR when the upper byte data is r
Therefore byte access to ADDR should be done by reading the upper byte first then the lo
Word access is also possible. ADDR is initialized to H'0000.

Table 14.2 Analog Input Channels and Corresponding ADDR Registers

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

				[Clearing condition] • When 0 is written after reading ADF =
6	ADIE	0	R/W	A/D Interrupt Enable A/D conversion end interrupt (ADI) reques by ADF when 1 is set
5	ADST	0	R/W	A/D Start Setting this bit to 1 starts A/D conversion. mode, this bit is cleared to 0 automatically conversion on the specified channel is co scan mode, conversion continues sequen the specified channels until this bit is clea software, a reset, or a transition to standb
4	SCAN	0	R/W	Scan Mode Selects single mode or scan mode as the conversion operating mode. 0: Single mode 1: Scan mode
3	CKS	0	R/W	Clock Select Selects the A/D conversions time 0: Conversion time = 134 states (max.) 1: Conversion time = 70 states (max.) Clear the ADST bit to 0 before switching t conversion 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 edg rising edge of the external trigger signal (A when this bit is set to 1. The selection between the falling edge and edge of the external trigger pin (ADTRG) c to the WPEG5 bit in the interrupt edge sele 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/writable.

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 s this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion. When A/D conversion ends, bit is automatically cleared to 0 and the A/D converter enters the wait state.

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 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 secc
subsequent conversions, the conversion time is 128 states (fixed) when CKS = 0 and 66 s
(fixed) when CKS = 1.

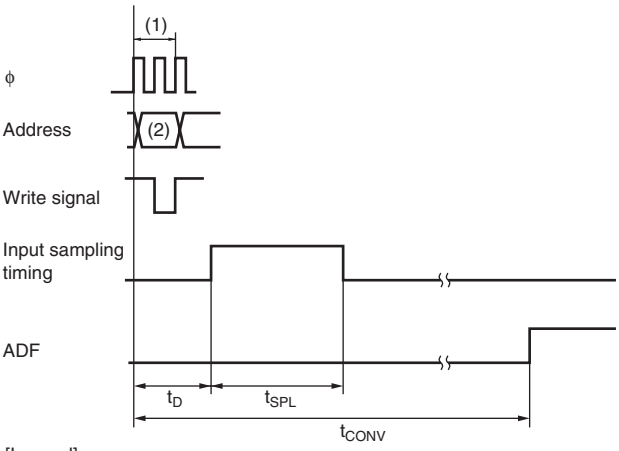


Figure 14.2 A/D Conversion Timing

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.

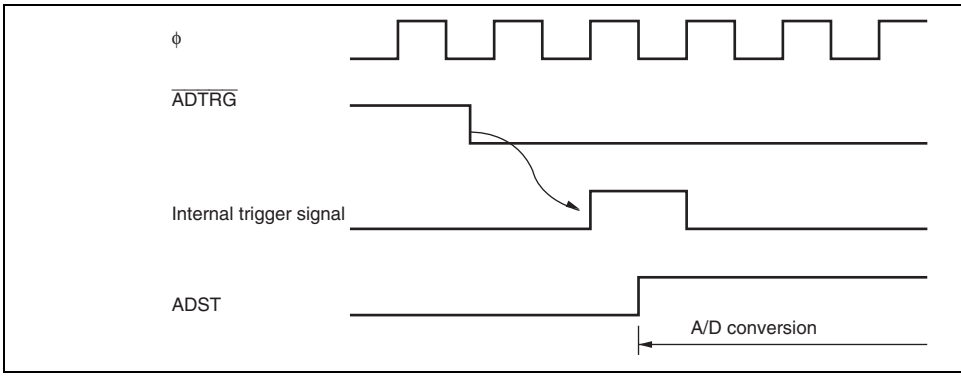


Figure 14.3 External Trigger Input Timing

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.

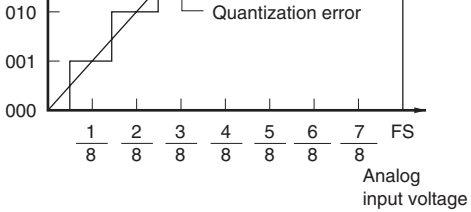


Figure 14.4 A/D Conversion Accuracy Definitions (1)

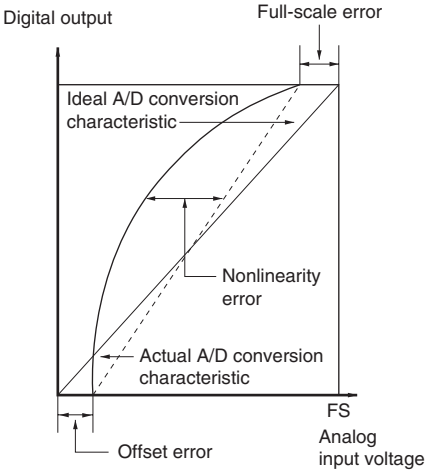


Figure 14.5 A/D Conversion Accuracy Definitions (2)

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.

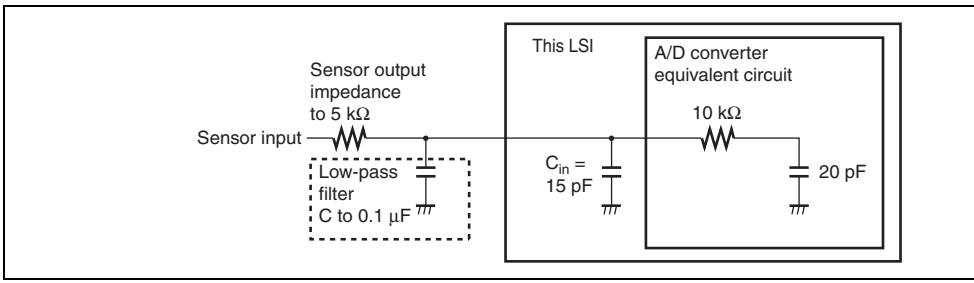


Figure 14.6 Analog Input Circuit Example

15.1 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 15.1. 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.

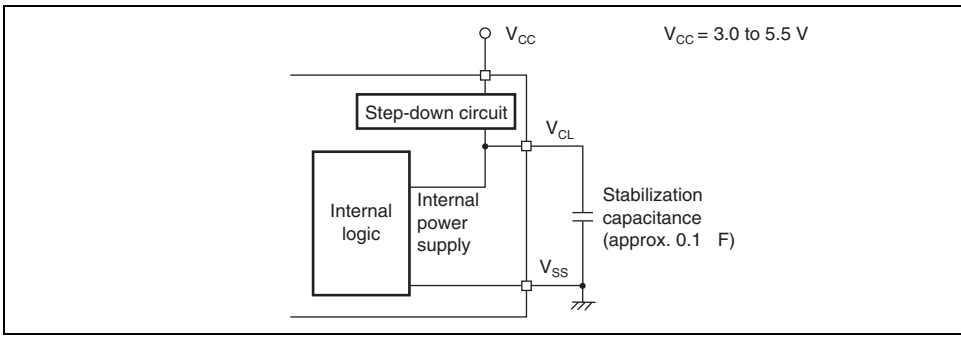


Figure 15.1 Power Supply Connection when Internal Step-Down Circuit is U

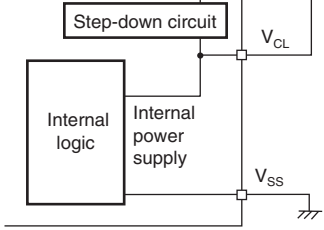


Figure 15.2 Power Supply Connection when Internal Step-Down Circuit is Not

- 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

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*[1]
General register A	GRA	16	H'FF88	Timer W	16*[1]
General register B	GRB	16	H'FF8A	Timer W	16*[1]
General register C	GRC	16	H'FF8C	Timer W	16*[1]
General register D	GRD	16	H'FF8E	Timer W	16*[1]
Flash memory control register 1	FLMCR1	8	H'FF90	ROM	8
Flash memory control register 2	FLMCR2	8	H'FF91	ROM	8
Erase block register 1	EBR1	8	H'FF93	ROM	8
Flash memory enable register	FENR	8	H'FF9B	ROM	8
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

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*[2]	8
Timer counter WD	TCWD	8	H'FFC1	WDT*[2]	8
Timer mode register WD	TMWD	8	H'FFC2	WDT*[2]	8
Address break control register	ABRKCR	8	H'FFC8	Address break	8
Address break status register	ABRKSR	8	H'FFC9	Address break	8
Break address register H	BARH	8	H'FFCA	Address break	8
Break address register L	BARL	8	H'FFCB	Address break	8
Break data register H	BDRH	8	H'FFCC	Address break	8
Break data register L	BDRL	8	H'FFCD	Address break	8
Port pull-up control register 1	PUCR1	8	H'FFD0	I/O port	8
Port pull-up control register 5	PUCR5	8	H'FFD1	I/O port	8
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 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
Interrupt edge select register 2	IEGR2	8	H'FFF3	Interrupts	8
Interrupt enable register 1	IENR1	8	H'FFF4	Interrupts	8
Interrupt flag register 1	IRR1	8	H'FFF6	Interrupts	8
Wake-up interrupt flag register	IWPR	8	H'FFF8	Interrupts	8
Module standby control register 1	MSTCR1	8	H'FFF9	Power-down	8

Notes: 1. Only word access can be used.
2. WDT: Watchdog timer

TSRW	OVF	—	—	—	IMFD	IMFC	IMFB	IMFA	
TIOR0	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0	
TIOR1	—	IOD2	IOD1	IOD0	—	IOC2	IOC1	IOC0	
TCNT	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8	
	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0	
GRA	GRA15	GRA14	GRA13	GRA12	GRA11	GRA10	GRA9	GRA8	
	GRA7	GRA6	GRA5	GRA4	GRA3	GRA2	GRA1	GRA0	
GRB	GRB15	GRB14	GRB13	GRB12	GRB11	GRB10	GRB9	GRB8	
	GRB7	GRB6	GRB5	GRB4	GRB3	GRB2	GRB1	GRB0	
GRC	GRC15	GRC14	GRC13	GRC12	GRC11	GRC10	GRC9	GRC8	
	GRC7	GRC6	GRC5	GRC4	GRC3	GRC2	GRC1	GRC0	
GRD	GRD15	GRD14	GRD13	GRD12	GRD11	GRD10	GRD9	GRD8	
	GRD7	GRD6	GRD5	GRD4	GRD3	GRD2	GRD1	GRD0	
FLMCR1	—	SWE	ESU	PSU	EV	PV	E	P	F
FLMCR2	FLER	—	—	—	—	—	—	—	
EBR1	—	—	—	EB4	EB3	EB2	EB1	EB0	
FENR	FLSHE	—	—	—	—	—	—	—	
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	

ADDRC	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRD	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0	
ADCR	TRGE	—	—	—	—	—	—	—	
TCSRWD	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI	WRST	WI
TCWD	TCWD7	TCWD6	TCWD5	TCWD4	TCWD3	TCWD2	TCWD1	TCWD0	
TMWD	—	—	—	—	CKS3	CKS2	CKS1	CKS0	
ABRKCR	RTINTE	CSEL1	CSEL0	ACMP2	ACMP1	ACMP0	DCMP1	DCMP0	Ad bre
ABRKSR	ABIF	ABIE	—	—	—	—	—	—	
BARH	BARH7	BARH6	BARH5	BARH4	BARH3	BARH2	BARH1	BARH0	
BARL	BARL7	BARL6	BARL5	BARL4	BARL3	BARL2	BARL1	BARL0	
BDRH	BDRH7	BDRH6	BDRH5	BDRH4	BDRH3	BDRH2	BDRH1	BDRH0	
BDRL	BDRL7	BDRL6	BDRL5	BDRL4	BDRL3	BDRL2	BDRL1	BDRL0	
PUCR1	PUCR17	PUCR16	PUCR15	PUCR14	—	PUCR12	PUCR11	PUCR10	I/O
PUCR5	—	—	PUCR55	PUCR54	PUCR53	PUCR52	PUCR51	PUCR50	
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	—	—	—	—	
PDR8	—	—	—	P84	P83	P82	P81	P80	

SYSCR1	SSBY	STS2	STS1	STS0	—	—	—	—	P
SYSCR2	SMSEL	—	DTON	MA2	MA1	MA0	—	—	
IEGR1	—	—	—	—	IEG3	—	—	IEG0	In
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	—	P

Note: * WDT: Watchdog timer

TCNT	Initialized	—	—	—	—	
GRA	Initialized	—	—	—	—	
GRB	Initialized	—	—	—	—	
GRC	Initialized	—	—	—	—	
GRD	Initialized	—	—	—	—	
FLMCR1	Initialized	—	—	Initialized	Initialized	ROM
FLMCR2	Initialized	—	—	—	—	
EBR1	Initialized	—	—	Initialized	Initialized	
FENR	Initialized	—	—	—	—	
TCRV0	Initialized	—	—	Initialized	Initialized	Timer V
TCSRV	Initialized	—	—	Initialized	Initialized	
TCORA	Initialized	—	—	Initialized	Initialized	
TCORB	Initialized	—	—	Initialized	Initialized	
TCNTV	Initialized	—	—	Initialized	Initialized	
TCRV1	Initialized	—	—	Initialized	Initialized	
SMR	Initialized	—	—	Initialized	Initialized	SCI3
BRR	Initialized	—	—	Initialized	Initialized	
SCR3	Initialized	—	—	Initialized	Initialized	
TDR	Initialized	—	—	Initialized	Initialized	
SSR	Initialized	—	—	Initialized	Initialized	
RDR	Initialized	—	—	Initialized	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	—	—	—	—	
PDR8	Initialized	—	—	—	—	
PDRB	Initialized	—	—	—	—	
PMR1	Initialized	—	—	—	—	
PMR5	Initialized	—	—	—	—	
PCR1	Initialized	—	—	—	—	
PCR2	Initialized	—	—	—	—	
PCR5	Initialized	—	—	—	—	
PCR7	Initialized	—	—	—	—	
PCR8	Initialized	—	—	—	—	

Notes: — is not initialized

* WDT: Watchdog timer

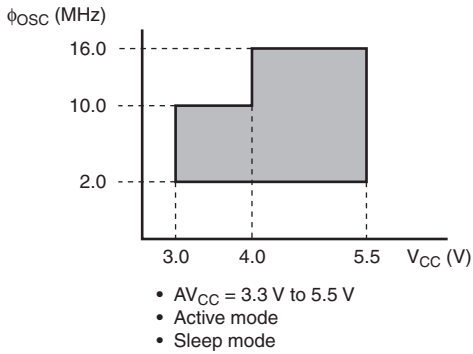
	Port B		-0.3 to AV_{CC} $+0.3$	V
Operating temperature		T_{opr}	-20 to $+75$	°C
Storage temperature		T_{stg}	-55 to $+125$	°C

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.

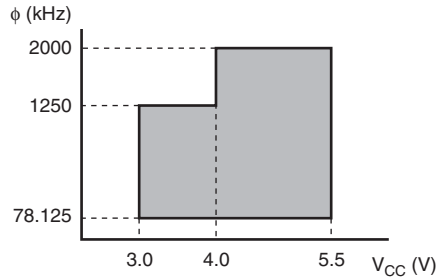
17.2 Electrical Characteristics

17.2.1 Power Supply Voltage and Operating Ranges

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)

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

		TMCIV, FTCI, FTIOA to FTIOD, SCK3, TRGV		$V_{cc} \times 0.9$	—	$V_{cc} + 0.3$	
		RXD, P12 to P10, P17 to P14, P22 to P20, P57 to P50, P76 to P74, 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, 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, P12 to P10, P17 to P14, P22 to P20, P57 to P50, P76 to P74, 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	

		P57 to P50, P76 to P74, P84 to P80					
		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	—	
Input capacitance	C_{in}	All input pins except power supply pins	f = 1 MHz, V_{IN} = 0.0 V, T_a = 25°C	—	—	15.0	pF
Active mode current consumption	I_{OPE1}	V_{CC}	Active mode 1 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	—	15.0	22.5	mA
			Active mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	8.0	—	
	I_{OPE2}	V_{CC}	Active mode 2 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	—	1.8	2.7	mA
			Active mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	1.2	—	

			V_{CC} = 3.0 V, f_{OSC} = 10 MHz				
Standby mode current consumption	I_{STBY}	V_{CC}	32-kHz crystal resonator not used	—	—	5.0	µA
RAM data retaining voltage	V_{RAM}	V_{CC}		2.0	—	—	V

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

Table 17.2 DC Characteristics (2)

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

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max
Allowable output low current (per pin)	I_{OL}	Output pins except port 8	V_{CC} = 4.0 V to 5.5 V	—	—	2.0
		Port 8		—	—	20.0
		Port 8		—	—	10.0
		Output pins except port 8		—	—	0.5
Allowable output low current (total)	ΣI_{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)	$\lvert -\Sigma -I_{OH} \rvert$	All output pins	V_{CC} = 4.0 V to 5.5 V	—	—	30.0
				—	—	8.0

System clock (3) cycle time	t_{cyc}						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	25.0	—	—	ns
				40.0	—	—	ns
External clock low width	t_{CPL}	OSC1	V_{CC} = 4.0 V to 5.5 V	25.0	—	—	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, $\overline{\text{ADTRG}}$, FTCI, FTIOA to FTIOD				
Input pin low width	t_{IL}	$\overline{\text{NMI}}$, $\overline{\text{IRQ0}}$, $\overline{\text{IRQ3}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, 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	t_{SCKW}						t_{cyc}
Transmit data delay time (clocked synchronous)	t_{TXD}	TXD	V_{CC} = 4.0 V to 5.5 V	—	—	1	t_{cyc}
				—	—	1	t_{cyc}
Receive data setup time (clocked synchronous)	t_{RXS}	RXD	V_{CC} = 4.0 V to 5.5 V	62.5	—	—	ns
				100.0	—	—	ns
Receive data hold time (clocked synchronous)	t_{RXH}	RXD	V_{CC} = 4.0 V to 5.5 V	62.5	—	—	ns
				100.0	—	—	ns

Analog power supply current	AI_{OPE}	AV_{CC}	AV_{CC} = 5.0 V f_{OSC} = 16 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} = 3.3 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.

17.2.5 Watchdog Timer

Table 17.6 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	I
				Min	Typ	Max		
On-chip oscillator overflow time	t_{OVF}			0.2	0.4	—	s	*

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

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

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

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

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

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

4. Programming time maximum value (t_P(MAX)) = wait time after P bit setting (z) maximum programming count (N)

5. Set the maximum programming count (N) according to the actual set values o and z3, so that it does not exceed the programming time maximum value (t_P(I The wait time after P bit setting (z1, z2) should be changed as follows accord value of the programming count (n).

   Programming count (n)

   $1 \leq n \leq 6$ z1 = 30 µs

   $7 \leq n \leq 1000$ z2 = 200 µs

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

7. Set the maximum maximum erase count (N) according to the actual set value that it does not exceed the erase time maximum value (t_E(max)).

Figure 17.1 System Clock Input Timing

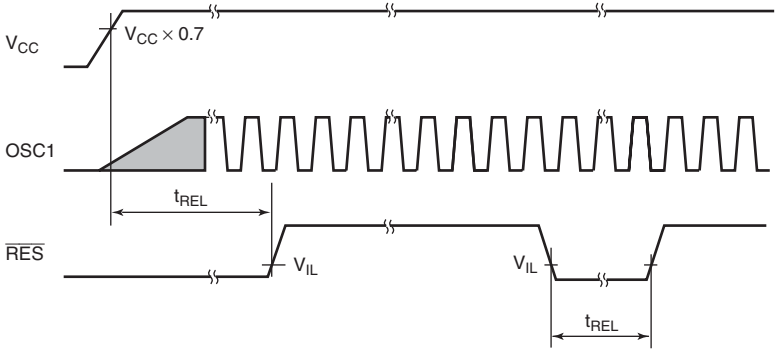


Figure 17.2 $\overline{\text{RES}}$ Low Width Timing

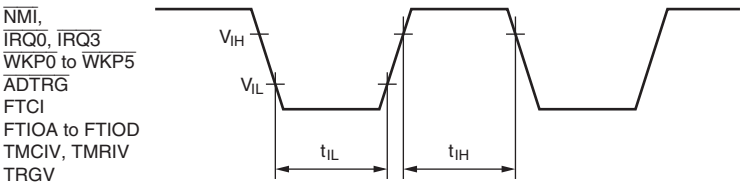
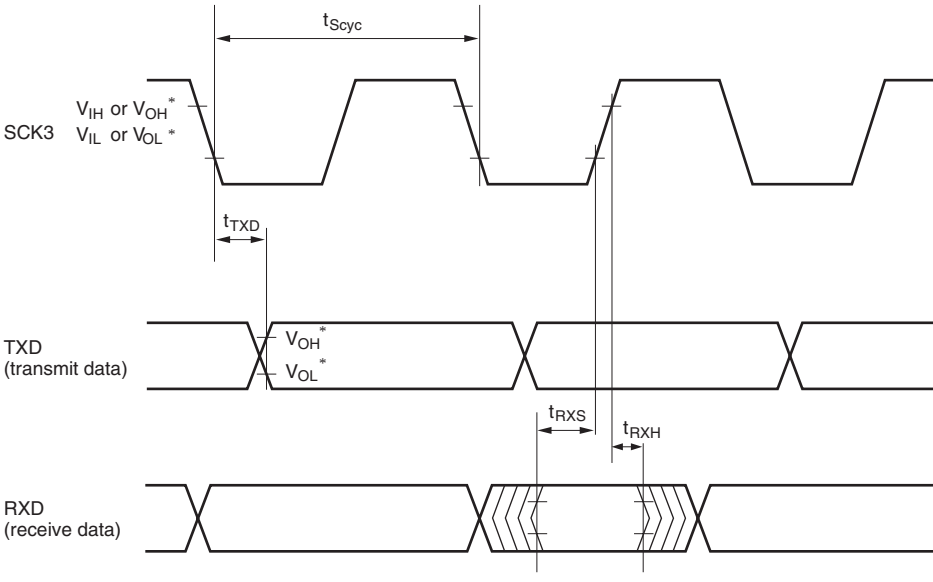


Figure 17.3 Input Timing



Note: * Output timing reference levels
Output high: V_{OH}= 2.0 V
Output low: V_{OL}= 0.8 V
Load conditions are shown in figure 17.6.

Figure 17.5 SCI3 Input/Output Timing in Clocked Synchronous Mode

Figure 17.6 Output Load Circuit

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

Note: General registers include 8-bit registers (R0H to R7H and R0L to R7L) and 16-bit (R0 to R7 and E0 to E7).

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

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

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

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

			of ERd32)															
EXTS	EXTS.W Rd	W	(<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				—	—	—	—	—				

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

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

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

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

2. n is the value set in register R4L or R4.
   (1) Set to 1 when a carry or borrow occurs at bit 11; otherwise cleared to 0.
   (2) Set to 1 when a carry or borrow occurs at bit 27; otherwise cleared to 0.
   (3) Retains its previous value when the result is zero; otherwise cleared to 0.
   (4) Set to 1 when the adjustment produces a carry; otherwise retains its prev
   (5) The number of states required for execution of an instruction that transfer
   synchronization with the E clock is variable.
   (6) Set to 1 when the divisor is negative; otherwise cleared to 0.
   (7) Set to 1 when the divisor is zero; otherwise cleared to 0.
   (8) Set to 1 when the quotient is negative; otherwise cleared to 0.

Instruction code:

1st byte	2nd byte
AH	AL	BH	BL

Instruction when most significant bit of BH is 0

Instruction when most significant bit of BH is 1

AH \ AL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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	JMP		BGE	BLT	BGT	BLE
5	MULXU	DIVXU	MULXU	DIVXU	RTS	BSR	RTE	TRAPA (2)					BSR	JSR		
6	BSET		BCLR	BTST	OR	XOR	AND	BST / BIST	MOV							
7		BNOT			BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD		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

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

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

BSET #0, @FF00

From table A.4:

I = L = 2, J = K = M = N= 0

From table A.3:

$S_I = 2$, $S_L = 2$

Number of states required for execution = $2 \times 2 + 2 \times 2 = 8$

When instruction is fetched from on-chip ROM, branch address is read from on-chip RO
on-chip RAM is used for stack area.

JSR @@ 30

From table A.4:

I = 2, J = K = 1, L = M = N = 0

From table A.3:

$S_I = S_J = S_K = 2$

Number of states required for execution = $2 \times 2 + 1 \times 2 + 1 \times 2 = 8$

Note: * Depends on which on-chip peripheral module is accessed. See section 16.1, R
Addresses (Address Order).

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

	BCS d:16(BLO d:16)	2			2
	BNE d:16	2			2
	BEQ d:16	2			2
	BVC d:16	2			2
	BVS d:16	2			2
	BPL d:16	2			2
	BMI d:16	2			2
	BGE d:16	2			2
	BLT d:16	2			2
	BGT d:16	2			2
	BLE d:16	2			2
BCLR	BCLR #xx:3, Rd	1			
	BCLR #xx:3, @ERd	2		2	
	BCLR #xx:3, @aa:8	2		2	
	BCLR Rn, Rd	1			
	BCLR Rn, @ERd	2		2	
	BCLR Rn, @aa:8	2		2	
BIAND	BIAND #xx:3, Rd	1			
	BIAND #xx:3, @ERd	2		1	
	BIAND #xx:3, @aa:8	2		1	
BILD	BILD #xx:3, Rd	1			
	BILD #xx:3, @ERd	2		1	
	BILD #xx:3, @aa:8	2		1	

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

	BXOR #xx:3, @aa:8	2		1	
CMP	CMP.B #xx:8, Rd	1			
	CMP.B Rs, Rd	1			
	CMP.W #xx:16, Rd	2			
	CMP.W Rs, Rd	1			
	CMP.L #xx:32, ERd	3			
	CMP.L ERs, ERd	1			
DAA	DAA Rd	1			
DAS	DAS Rd	1			
DEC	DEC.B Rd	1			
	DEC.W #1/2, Rd	1			
	DEC.L #1/2, ERd	1			
DUVXS	DIVXS.B Rs, Rd	2			1
	DIVXS.W Rs, ERd	2			2
DIVXU	DIVXU.B Rs, Rd	1			1
	DIVXU.W Rs, ERd	1			2
EEPMOV	EEPMOV.B	2		2n+2*[1]	
	EEPMOV.W	2		2n+2*[1]	
EXTS	EXTS.W Rd	1			
	EXTS.L ERd	1			
EXTU	EXTU.W Rd	1			
	EXTU.L ERd	1			

	JSR @@aa:8	2	1	1			
LDC	LDC #xx:8, CCR	1					
	LDC Rs, CCR	1					
	LDC@ERs, CCR	2					1
	LDC@(d:16, ERs), CCR	3					1
	LDC@(d:24,ERs), CCR	5					1
	LDC@ERs+, CCR	2					1
	LDC@aa:16, CCR	3					1
	LDC@aa:24, CCR	4					1
MOV	MOV.B #xx:8, Rd	1					
	MOV.B Rs, Rd	1					
	MOV.B @ERs, Rd	1			1		
	MOV.B @(d:16, ERs), Rd	2			1		
	MOV.B @(d:24, ERs), Rd	4			1		
	MOV.B @ERs+, Rd	1			1		
	MOV.B @aa:8, Rd	1			1		
	MOV.B @aa:16, Rd	2			1		
	MOV.B @aa:24, Rd	3			1		
	MOV.B Rs, @Erd	1			1		
	MOV.B Rs, @(d:16, ERd)	2			1		
	MOV.B Rs, @(d:24, ERd)	4			1		
	MOV.B Rs, @-ERd	1			1		
	MOV.B Rs, @aa:8	1			1		

	MOV.W @aa:16, Rd	2			1	
	MOV.W @aa:24, Rd	3			1	
	MOV.W Rs, @ERd	1			1	
	MOV.W Rs, @(d:16,ERd)	2			1	
	MOV.W Rs, @(d:24,ERd)	4			1	
MOV	MOV.W Rs, @-ERd	1			1	2
	MOV.W Rs, @aa:16	2			1	
	MOV.W Rs, @aa:24	3			1	
	MOV.L #xx:32, ERd	3				
	MOV.L ERs, ERd	1				
	MOV.L @ERs, ERd	2			2	
	MOV.L @(d:16,ERs), ERd	3			2	
	MOV.L @(d:24,ERs), ERd	5			2	
	MOV.L @ERs+, ERd	2			2	2
	MOV.L @aa:16, ERd	3			2	
	MOV.L @aa:24, ERd	4			2	
	MOV.L ERs,@ERd	2			2	
	MOV.L ERs, @(d:16,ERd)	3			2	
	MOV.L ERs, @(d:24,ERd)	5			2	
	MOV.L ERs, @-ERd	2			2	2
	MOV.L ERs, @aa:16	3			2	
	MOV.L ERs, @aa:24	4			2	
MOVFPE	MOVFPE @aa:16, Rd*[2]	2		1		
MOVTPE	MOVTPE Rs,@aa:16*[2]	2		1		

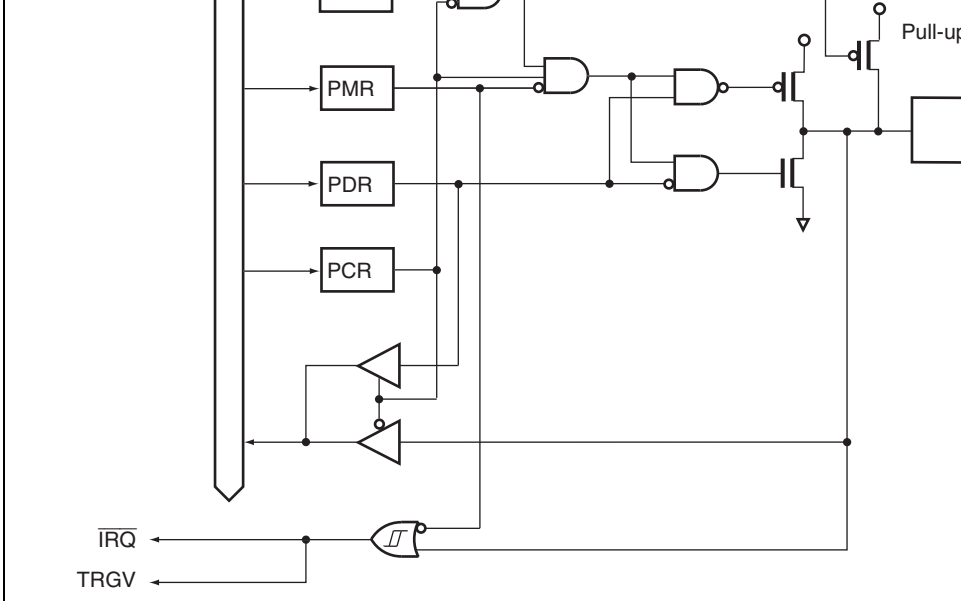
NOT	NOT.B Rd	1		
	NOT.W Rd	1		
	NOT.L ERd	1		
OR	OR.B #xx:8, Rd	1		
	OR.B Rs, Rd	1		
	OR.W #xx:16, Rd	2		
	OR.W Rs, Rd	1		
	OR.L #xx:32, ERd	3		
	OR.L ERs, ERd	2		
ORC	ORC #xx:8, CCR	1		
POP	POP.W Rn	1		1
	POP.L ERn	2		2
PUSH	PUSH.W Rn	1		1
	PUSH.L ERn	2		2
ROTL	ROTL.B Rd	1		
	ROTL.W Rd	1		
	ROTL.L ERd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

SHAR	SHAR.B Rd	1			
	SHAR.W Rd	1			
	SHAR.L ERd	1			
SHLL	SHLL.B Rd	1			
	SHLL.W Rd	1			
	SHLL.L ERd	1			
SHLR	SHLR.B Rd	1			
	SHLR.W Rd	1			
	SHLR.L ERd	1			
SLEEP	SLEEP	1			
STC	STC CCR, Rd	1			
	STC CCR, @ERd	2		1	
	STC CCR, @(d:16,ERd)	3		1	
		5		1	
	STC CCR, @(d:24,ERd)	2		1	2
		3		1	
	STC CCR,@-ERd	4		1	
	STC CCR, @aa:16				
	STC CCR, @aa:24				
SUB	SUB.B Rs, Rd	1			
	SUB.W #xx:16, Rd	2			
	SUB.W Rs, Rd	1			
	SUB.L #xx:32, ERd	3			
	SUB.L ERs, ERd	1			
SUBS	SUBS #1/2/4, ERd	1			

	XOR.L ERs, ERd	2
XORC	XORC #xx:8, CCR	1

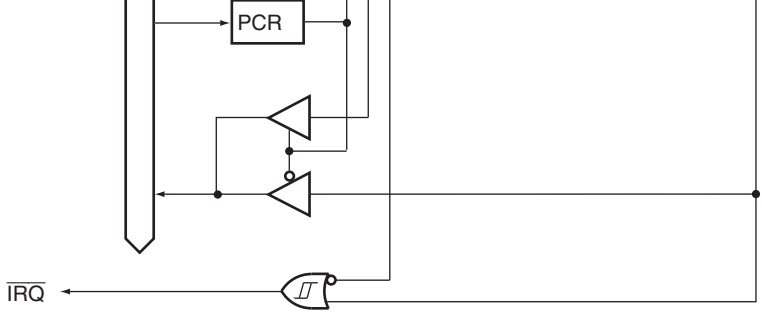
Notes: 1. n:specified value in R4L and R4. The source and destination operands are ac
n+1 times respectively.
2. Cannot be used in this LSI.

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



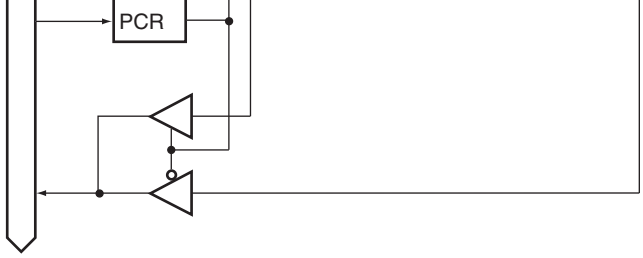
[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.1 Port 1 Block Diagram (P17)



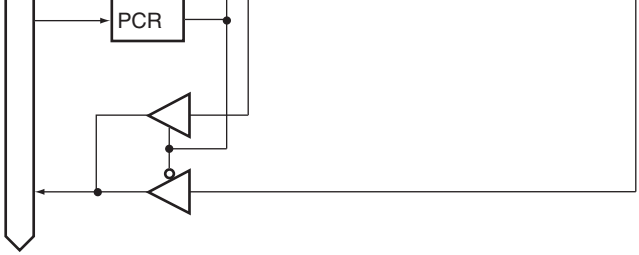
[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.2 Port 1 Block Diagram (P14)



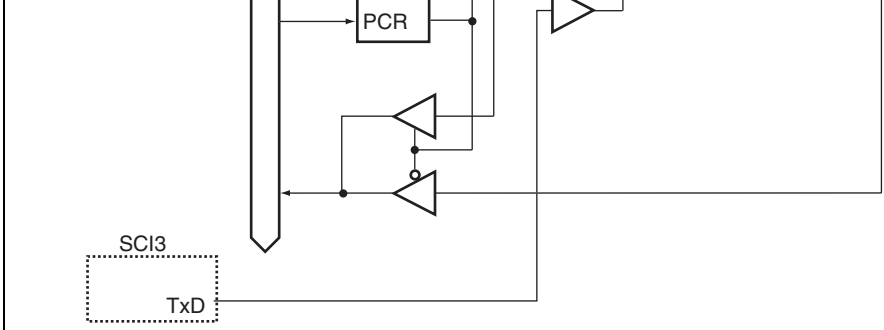
[Legend]
PUCR: Port pull-up control register
PDR: Port data register
PCR: Port control register

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



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

Figure B.4 Port 1 Block Diagram (P11)



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

Figure B.5 Port 2 Block Diagram (P22)

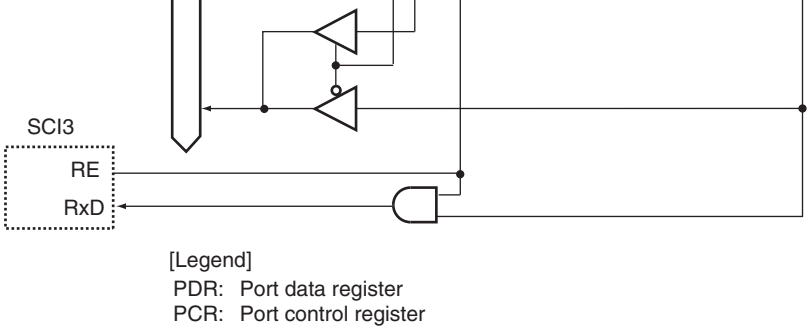
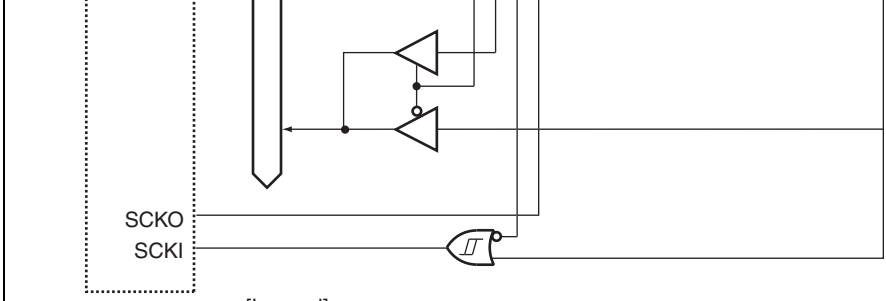
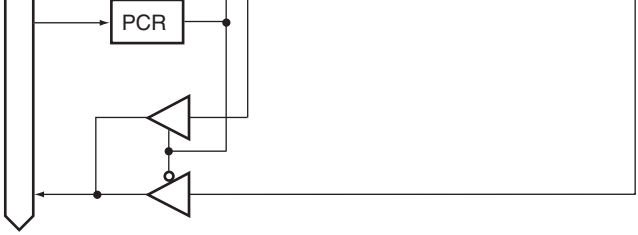


Figure B.6 Port 2 Block Diagram (P21)



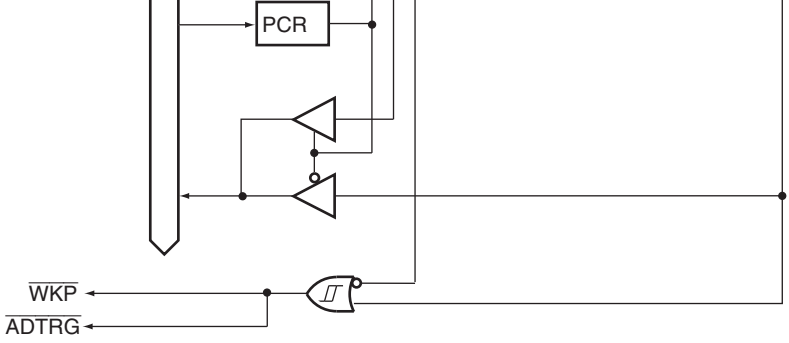
[Legend]
PDR: Port data register
PCR: Port control register

Figure B.7 Port 2 Block Diagram (P20)



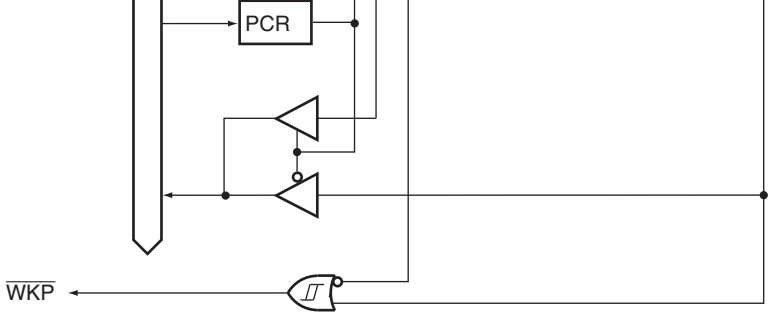
[Legend]
PMR: Port mode register
PDR: Port data register
PCR: Port control register

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



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

Figure B.9 Port 5 Block Diagram (P55)



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

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

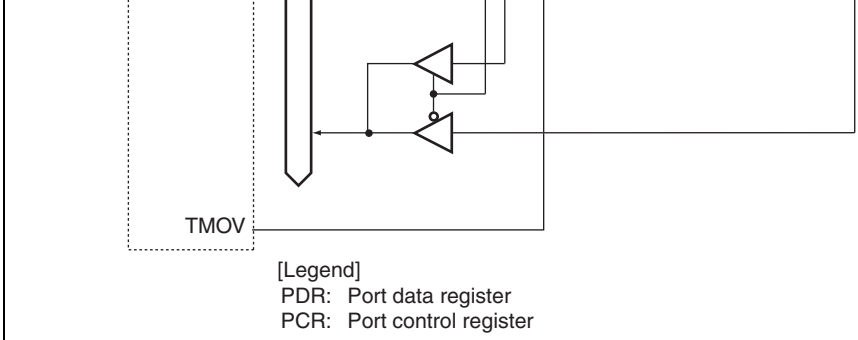


Figure B.11 Port 7 Block Diagram (P76)

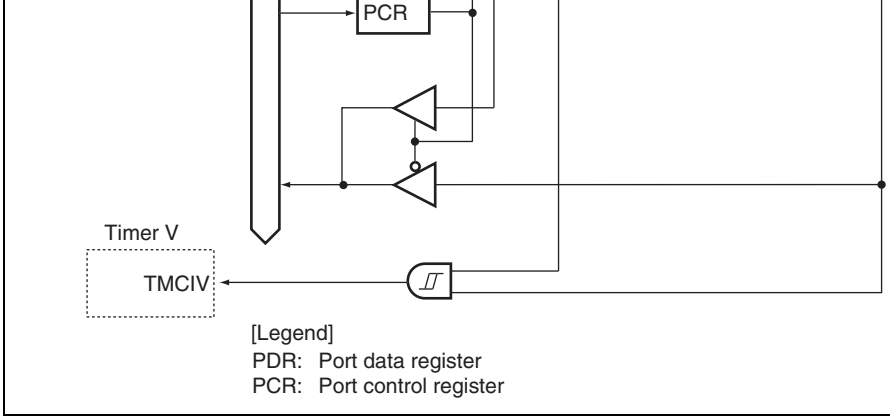


Figure B.12 Port 7 Block Diagram (P75)

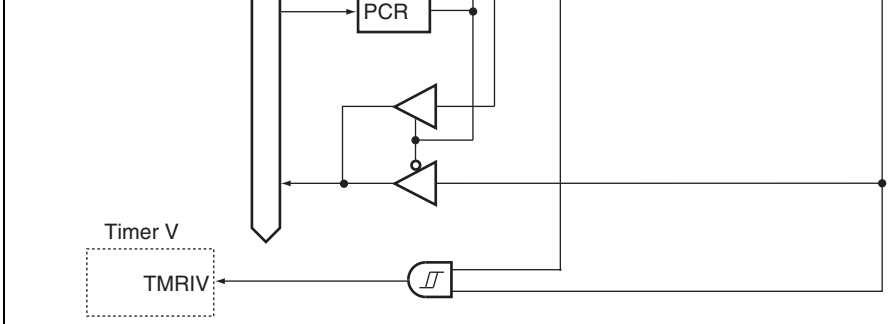


Figure B.13 Port 7 Block Diagram (P74)

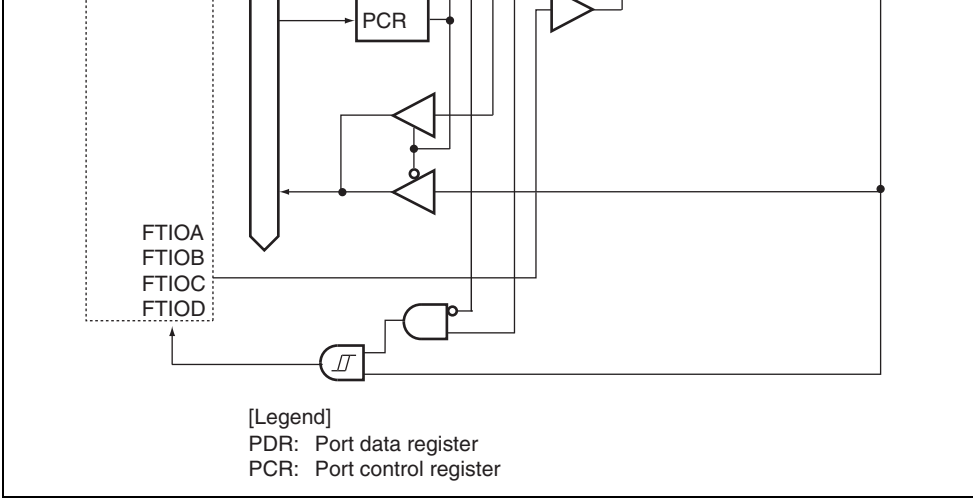
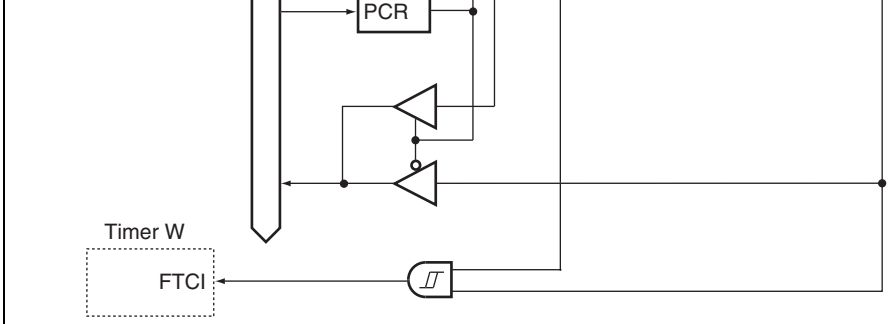


Figure B.14 Port 8 Block Diagram (P84 to P81)



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

Figure B.15 Port 8 Block Diagram (P80)

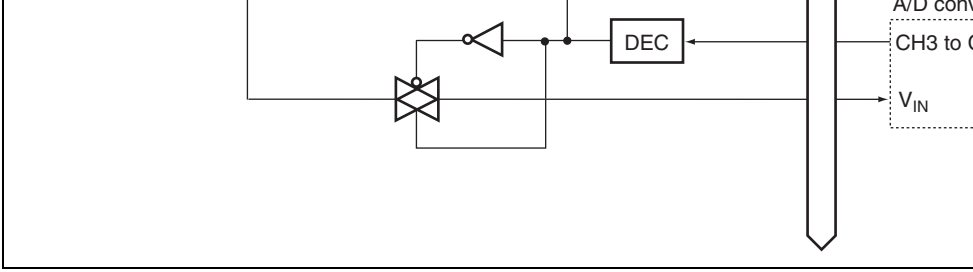


Figure B.16 Port B Block Diagram (PB3 to PB0)

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.

HD64F3670FX	HD64F3670FX	LQFP-48 (FI
HD64F3670FY	HD64F3670FY	LQFP-48 (FI

Figure D.1 FP-64E Package Dimensions

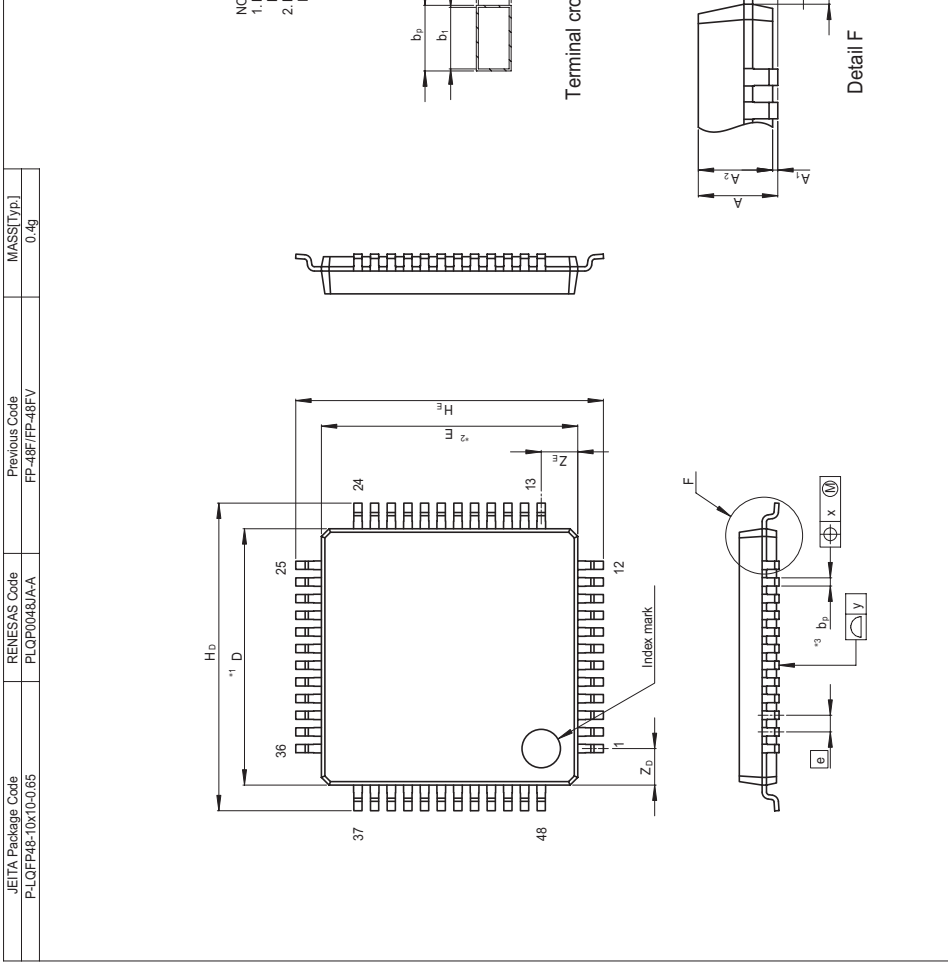


Figure D.2 FP-48F Package Dimensions

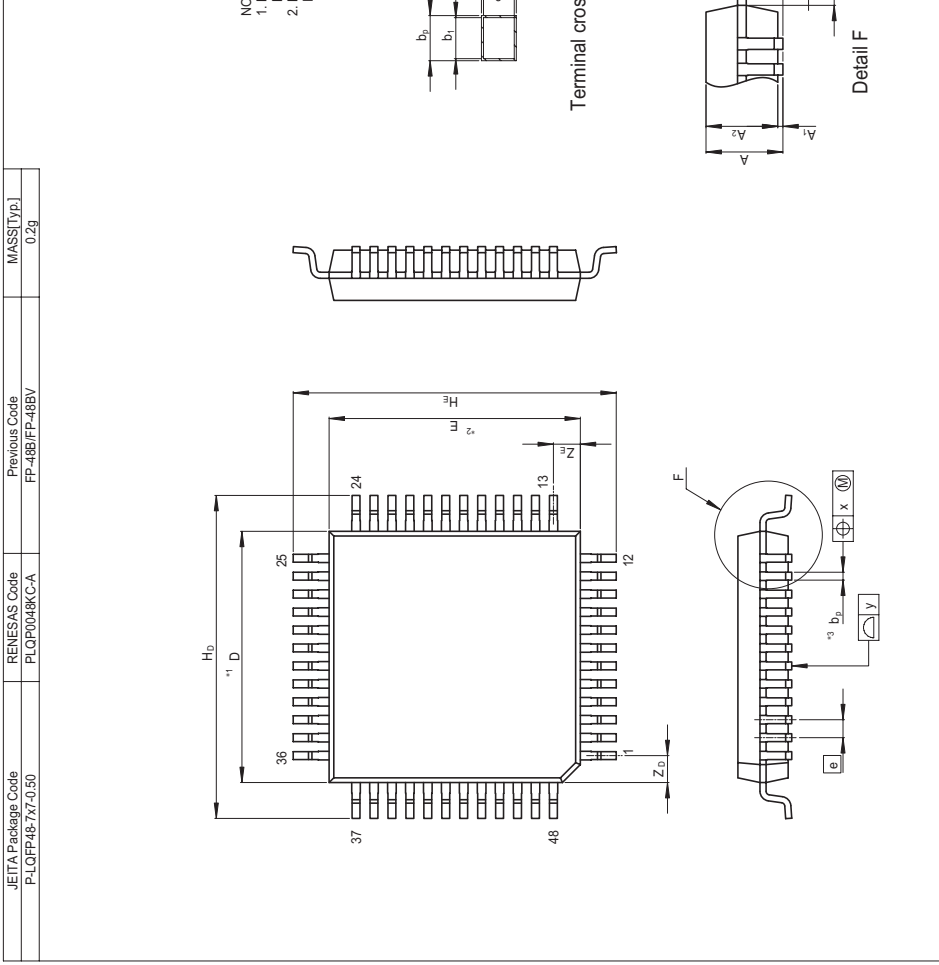
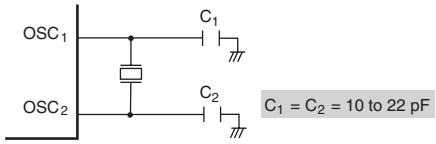
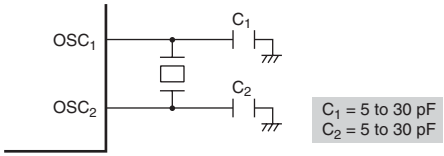
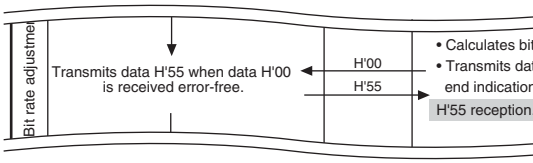


Figure D.3 FP-48B Package Dimensions

		4. When the E7 or E8 is used, address breaks can either available to the user or for use by the E7 address breaks are set as being used by the E7 address break control registers must not be acc 5. When the E7 or E8 is used, $\overline{\text{NMI}}$ is an input/outp (open-drain in output mode). Note has been deleted.
Section 1 Overview Figure 1.1 Internal Block Diagram Figure 1.2 Pin Arrangement (FP-64E) Figure 1.3 Pin Arrangement (FP-48F, FP-48B)	2 to 4	Note: * Can also be used for the E7 or E8 emulator.
1.4 Pin Functions Table 1.1 Pin Functions	6	**Type** / **Functions** E10T / Interface pin for the E10T, E7, or E emulator

Note has been deleted.

Section 5 Clock Pulse Generators Figure 5.3 Typical Connection to Crystal Resonator	66	
Figure 5.5 Typical Connection to Ceramic Resonator	66	
Section 7 ROM	79	The features of the 20-kbyte (4 kbytes of them are th E8 control program area) flash memory built into HD6 are summarized below.
Table 7.2 Boot Mode Operation	85	

			10: Output 11: Output toggles

Section 12 Watchdog Timer 12.2.1 Timer Control/Status Register WD (TCSRWD)	160	**Bit** 4	**Bit Name** TCSRWE	**Description** Timer Control/Status Regis Write Enable :

Section 14 A/D Converter 14.3.1 A/D Data Registers A to D (ADDRA to ADDRD)	206	Therefore byte access to ADDR should be done by the upper byte first then the lower one. Word access possible. ADDR is initialized to H'0000.	

Section 17 Electrical Characteristics
Table 17.2 DC Characteristics (1) — 230

Item	Symbol	Applicable Pins	Test Condition
Input high voltage	V_{IH}	PB3 to PB0	V_{CC} = 4.0 V to 5.5 V
Input low voltage	V_{IL}	RXD, P12 to P10, P17 to P14, P22 to P20, P57 to P50, P76 to P74, P84 to P80 PB3 to PB0	V_{CC} = 4.0 V to 5.5 V

C

E

G

I

M

O

P

T

V

W

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8/3672 Group

Publication Date: 1st Edition, Mar, 2001
Rev.4.00, Nov. 02, 2005
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

RENESAS SALES OFFICES

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, United Kingdom
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

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

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001
Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> 2-796-3115, Fax: <82> 2-796-2145

Renesas Technology Malaysia Sdn. Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan,
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8/3672 Group
Hardware Manual

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

REJ09B0143-0400

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Microprocessors - MPU ***category:***

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

Other Similar products are found below :

MC68302EH20C MC7457RX1000LC MC7457RX1267LC MC7457VG1267LC A2C00010998 A A2C52004004 R5F117BCGNA#20 R5F52106BDLA#U0 R5S72690W266BG#U0 ADJ3400IAA5DOE MPC8245TVV266D MPC8245TZU300D MPC8260ACVVMHBB MPC8323ECVRAFDCA MPC8323VRADDCA MPC8536ECVJAVLA BOXNUC5PGYH0AJ 20-668-0024 P1010NSN5DFB P2010NSE2KHC P2010NSN2MHC P2020NXE2HHC P5020NSE7QMB P5020NSE7TNB P5020NSE7VNB LS1020ASN7KQB LS1020AXN7HNB LS1020AXN7KQB A2C00010729 A A2C00039344 T1022NSE7MQB T1022NXN7PQB T1023NSE7MQA T1024NXE7PQA T1042NSE7MQB T1042NSN7MQB T1042NXN7WQB T2080NSE8TTB T2080NSN8PTB T2080NXE8TTB T2081NXN8TTB R5F101AFASP#V0 MC68302CEH20C TS68040MF33A MPC8260ACVVMIBB MPC8280CZUUPEA MPC8313ECVRAFFC MPC8313ECVRAGDC MPC8313EVRADDC MPC8313EVRAFFC