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

16 H8/36057Group, H8/36037Group

Hardware Manual

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

H8/36057	HD64F36057, HD64F36057G
	HD64336057, HD64336057G
H8/36054	HD64F36054, HD64F36054G
	HD64336054, HD64336054G
H8/36037	HD64F36037, HD64F36037G
	HD64336037, HD64336037G
H8/36036	HD64336036, HD64336036G
H8/36035	HD64336035, HD64336035G
H8/36034	HD64F36034, HD64F36034G
	HD64336034, HD64336034G
H8/36033	HD64336033, HD64336033G
H8/36032	HD64336032, HD64336032G

Renesas Electronics
www.renesas.com

Rev.4.00 2006.05

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.

RENESAS

are in their open states, intermediate levels are induced by noise in the vicinity, a through current flows internally, and a malfunction may occur.

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

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

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

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

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

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

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

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

11. Index

microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8/36057 Group and H8/36037 Group to the target Refer to the H8/300H Series Software Manual for a detailed description o instruction set.

Notes on reading this manual:

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

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

by the E7 or E8. If address breaks are set as being used by the E7 or E8, the address control registers must not be accessed.

6. When the E7 or E8 is used, $\overline{\text{NMI}}$ is an input/output pin (open-drain in output mode), P87 are input pins, and P86 is an output pin.
7. In on-board programming mode by boot mode, channel 1 (P21/RXD and P22/TXD) is used.

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

H8/36057 Group and H8/36037 Group manuals:

Document Title	Docume
H8/36057 Group, H8/36037 Group Hardware Manual	This ma
H8/300H Series Software Manual	REJ09B

User's manuals for development tools:

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

Section 15 Controller Area Network for Tiny (TinyCAN)

Section 17 Subsystem Timer (Subtimer)

Section 18 A/D Converter

Section 19 Power-On Reset and Low-Voltage Detection Circuits (Optional)

Appendix B I/O Port Block Diagrams

Appendix D Package Dimensions

— Timer B1 (8-bit timer)
— Timer V (8-bit timer)
— Timer Z (16-bit timer)
— Watchdog timer
— SCI3 (asynchronous or clocked synchronous serial communication interface)
— TinyCAN (controller area network for Tiny)
— SSU (synchronous serial communication unit)
— Subsystem timer (subtimer)
— 10-bit A/D converter

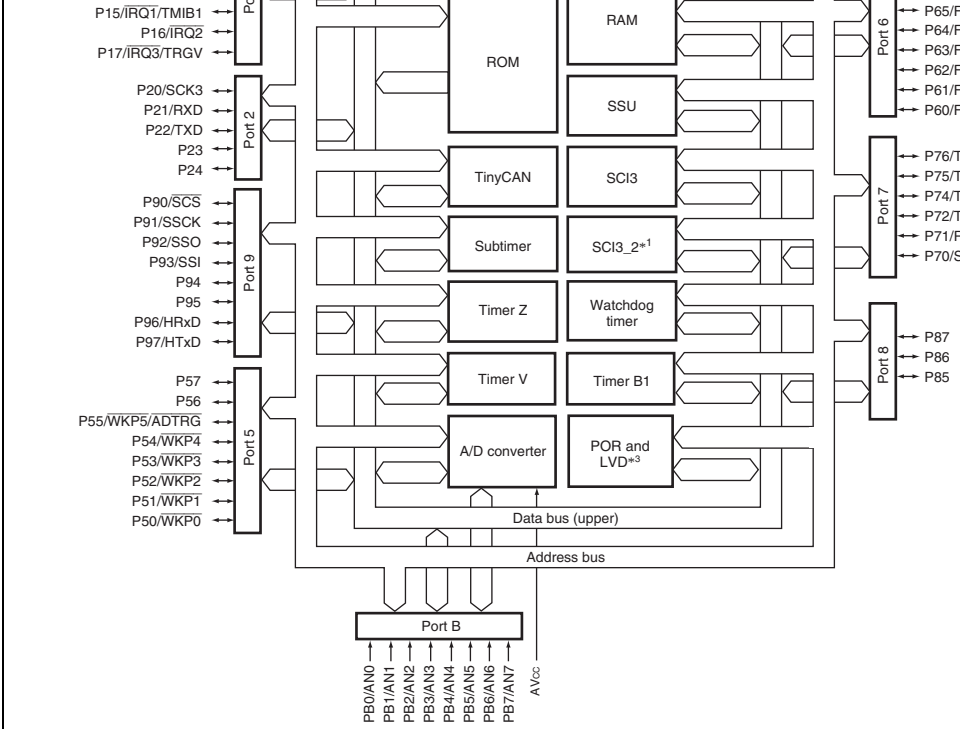
	H8/36034F	HD64F36034	HD64F36034G	32 kbytes	2
Masked ROM version	H8/36057	HD64336057	HD64336057G	56 kbytes	2
	H8/36054	HD64336054	HD64336054G	32 kbytes	2
	H8/36037	HD64336037	HD64336037G	56 kbytes	2
	H8/36036	HD64336036	HD64336036G	48 kbytes	2
	H8/36035	HD64336035	HD64336035G	40 kbytes	2
	H8/36034	HD64336034	HD64336034G	32 kbytes	2
	H8/36033	HD64336033	HD64336033G	24 kbytes	1
	H8/36032	HD64336032	HD64336032G	16 kbytes	1

- General I/O ports
  - I/O pins: 45 I/O pins, including 8 large current ports (I_{OL} = 20 mA, @V_{OL} = 1.5 V)
  - Input-only pins: 8 input pins (also used for analog input)
- Supports various power-down modes

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

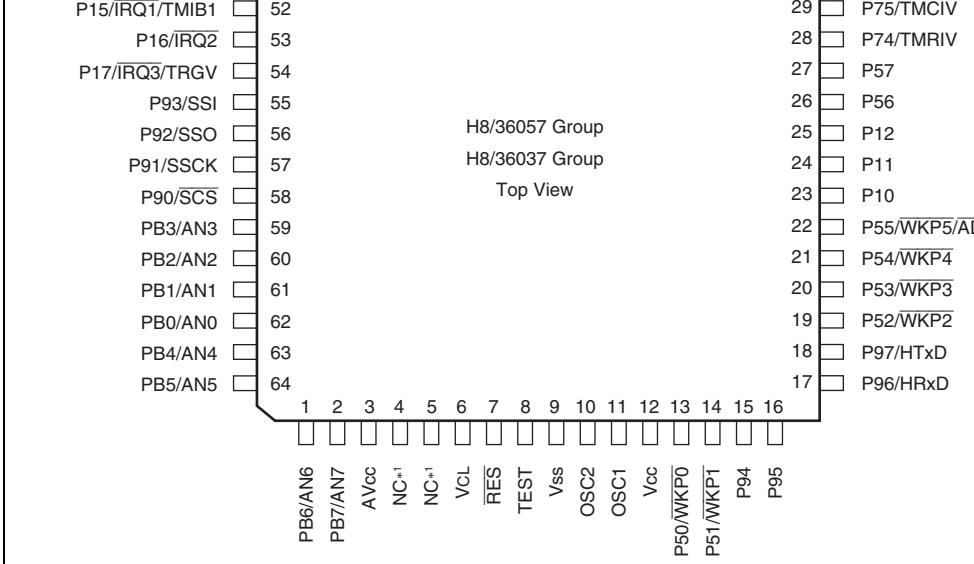
- Compact package

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



Notes: 1. The SCI3_2 is incorporated only in the H8/36057.
2. The SCK3_2, RXD_2, and TXD_2 pins are not multiplexed in the H8/36037.
3. POR and LVD function is incorporated in the H8/36057G and H8/36037G.

Figure 1.1 Internal Block Diagram of F-ZTAT™ and Masked ROM Versi



Notes: 1. Do not connect NC pins (these pins are not connected to the internal circuitry).
2. The SCK3_2, RXD_2, and TXD_2 pins are not multiplexed in the H8/36037.

Figure 1.2 Pin Arrangement of F-ZTAT™ and Masked ROM Versions (FP-64K, I

	AV_{CC}	3	Input	Analog power supply pin for the A/D When the A/D converter is not used, this pin to the system power supply.
	V_{CL}	6	Input	Internal step-down power supply pin a capacitor of around 0.1 µF betwee and the Vss pin for stabilization.
Clock pins	OSC1	11	Input	These pins connect with crystal or ce resonator for the system clock, or ca to input an external clock. See section 5, Clock Pulse Generato typical connection.
	OSC2	10	Output	
System control	$\overline{RES}$	7	Input	Reset pin. The pull-up resistor (typ. 1 incorporated. When driven low, the c reset.
	TEST	8	Input	Test pin. Connect this pin to Vss.
Interrupt pins	$\overline{NMI}$	35	Input	Non-maskable interrupt request inpu be pulled-up with a resistor.
	$\overline{IRQ0}$ to $\overline{IRQ3}$	51 to 54	Input	External interrupt request input pins. select the rising or falling edge.
	$\overline{WKP0}$ to $\overline{WKP5}$	13, 14, 19 to 22	Input	External interrupt request input pins. select the rising or falling edge.
Timer B1	TMIB1	52	Input	External event input pin.
Timer V	TMOV	30	Output	This is an output pin for waveforms g by the output compare function.
	TMCIV	29	Input	External event input pin.
	TMRIV	28	Input	Counter reset input pin.
	TRGV	54	Input	Counter start trigger input pin.

				input/PWM output pin.
	FTIOA1	37	I/O	Output compare output/input capture input/PWM output pin (at a reset, complementary PWM mode).
	FTIOB1 to FTIOD1	38 to 40	I/O	Output compare output/input capture input/PWM output pins.
Serial communication interface (SCI)	TXD, TXD_2*	46, 50	Output	Transmit data output pins.
	RXD, RXD_2*	45, 49	Input	Receive data input pins.
	SCK3, SCK3_2*	44, 48	I/O	Clock I/O pins.
Controller area network for Tiny (TinyCAN)	HRXD	17	Input	Receive data input pin.
	HTXD	18	Output	Transmit data output pin.
Synchronous serial communication unit (SSU)	$\overline{\text{SCS}}$	58	I/O	Chip select I/O pin.
	SSCK	57	I/O	Clock I/O pin.
	SSI	55	I/O	Transmit/receive data I/O pin.
	SSO	56	I/O	Transmit/receive data I/O pin.
A/D converter	AN7 to AN0	2, 1, 59 to 64	Input	Analog input pins.
	$\overline{\text{ADTRG}}$	22	Input	Conversion start trigger input pin.

P67 to P60	32 to 34, 36, 37 to 40	I/O	8-bit I/O ports.
P76 to P74, P72 to P70	28 to 30, 48 to 50	I/O	6-bit I/O ports.
P87 to P85	41 to 43	I/O	3-bit I/O ports.
P97 to P90	15 to 18, 58 to 55	I/O	8-bit I/O ports.

Note: * The SCK3_2, RXD_2, and TXD_2 pins are not multiplexed in the H8/36037.

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

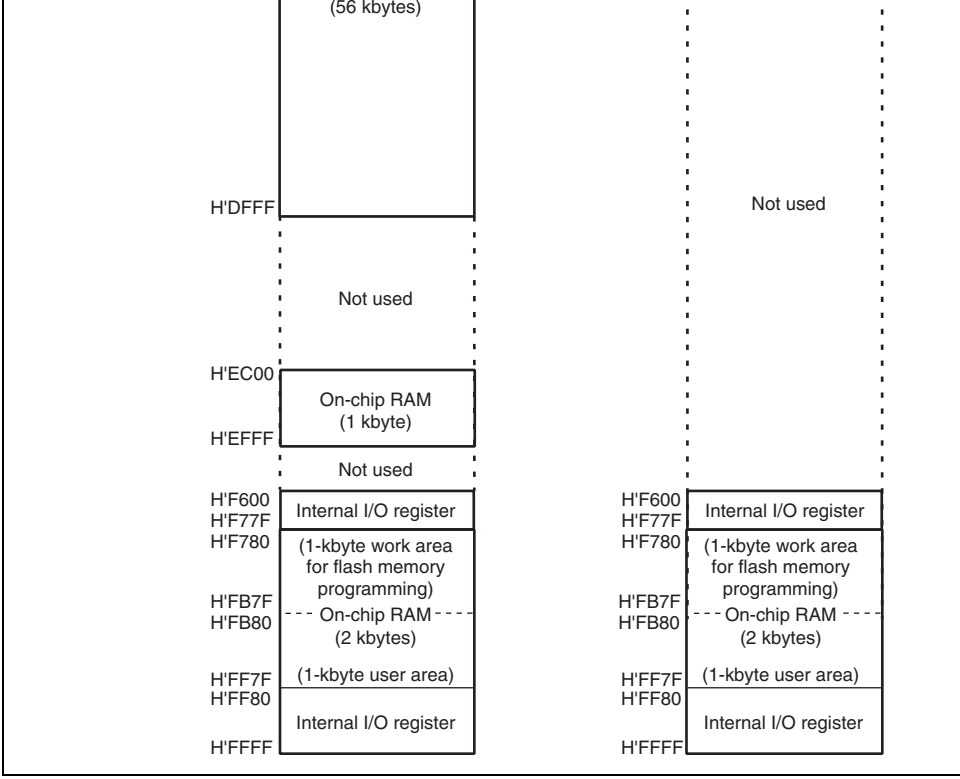


Figure 2.1 Memory Map (1)

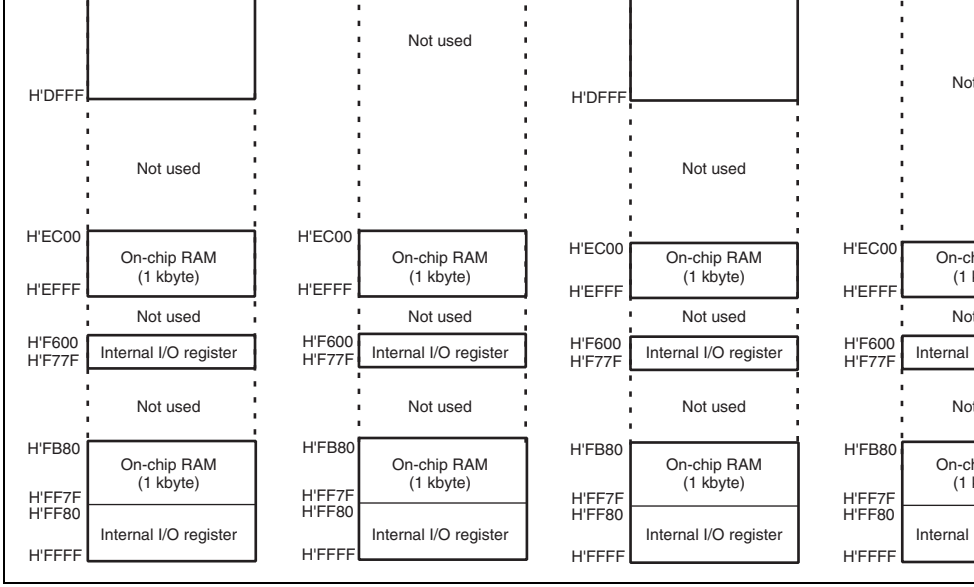


Figure 2.1 Memory Map (2)

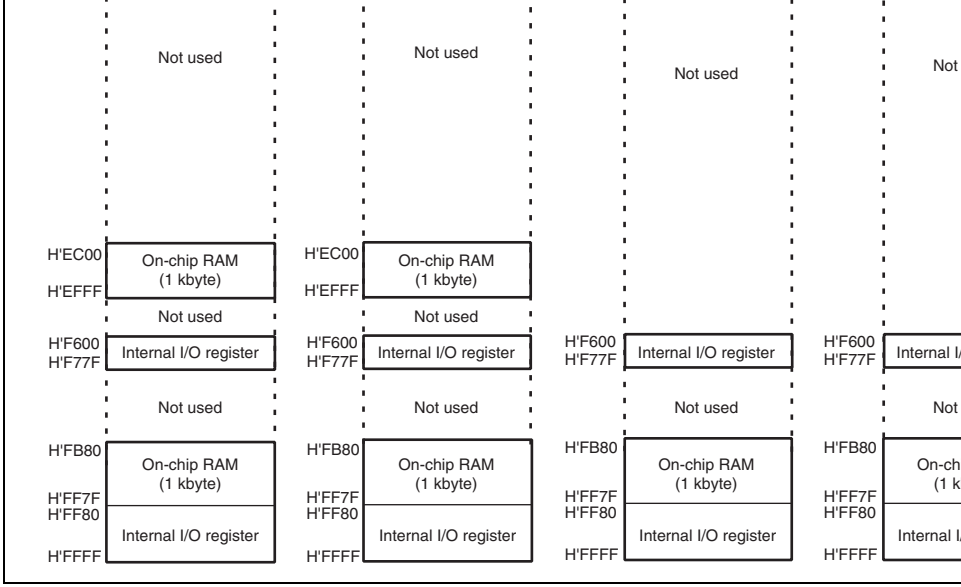
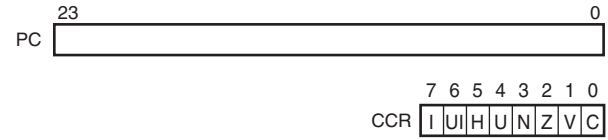


Figure 2.1 Memory Map (3)

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

Control Registers (CR)



[Legend]

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

Figure 2.2 CPU Registers

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

The usage of each register can be selected independently.

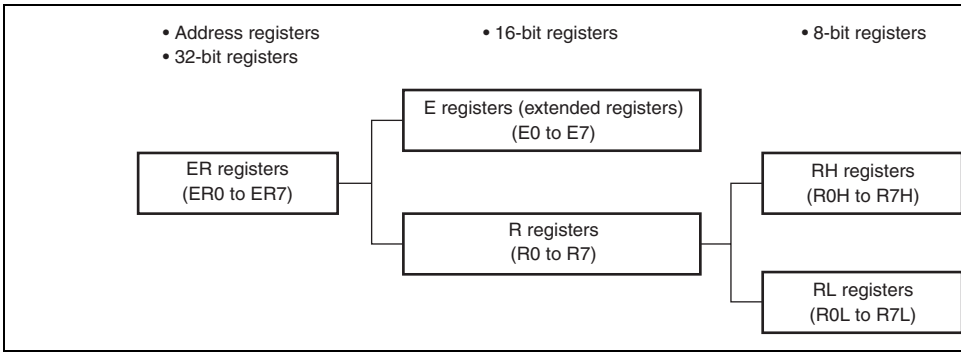


Figure 2.3 Usage of General Registers

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

Figure 2.4 Relationship between Stack Pointer and Stack Area

2.2.2 Program Counter (PC)

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

2.2.3 Condition-Code Register (CCR)

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

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

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

				When the ADD.B, ADDX.B, SUB.B, SUBX.B, NEG.B instruction is executed, this flag is set t 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, c instruction is executed, the H flag is set to 1 if carry or borrow at bit 27, and cleared to 0 othe
4	U	Undefined	R/W	User Bit Can be written and read by software using the STC, ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit of d sign bit.
2	Z	Undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to 0 indicate non-zero data.
1	V	Undefined	R/W	Overflow Flag Set to 1 when an arithmetic overflow occurs, a cleared to 0 at other times.
0	C	Undefined	R/W	Carry Flag Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by: • Add instructions, to indicate a carry • Subtract instructions, to indicate a borrow • Shift and rotate instructions, to indicate a c The carry flag is also used as a bit accumulato manipulation instructions.

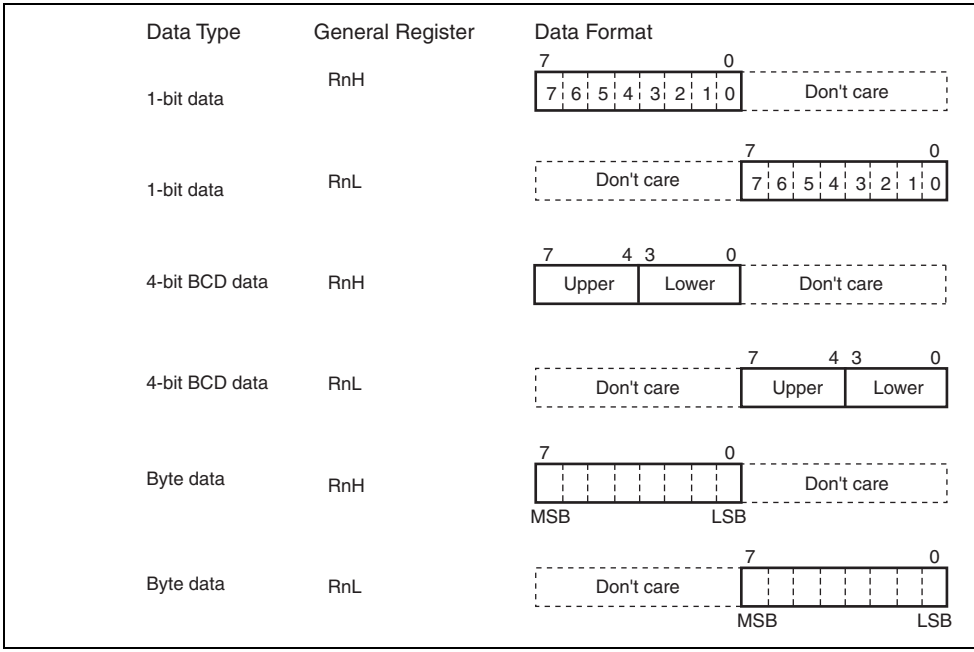
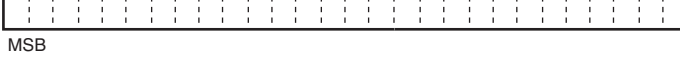


Figure 2.5 General Register Data Formats (1)



[Legend]

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

Figure 2.5 General Register Data Formats (2)

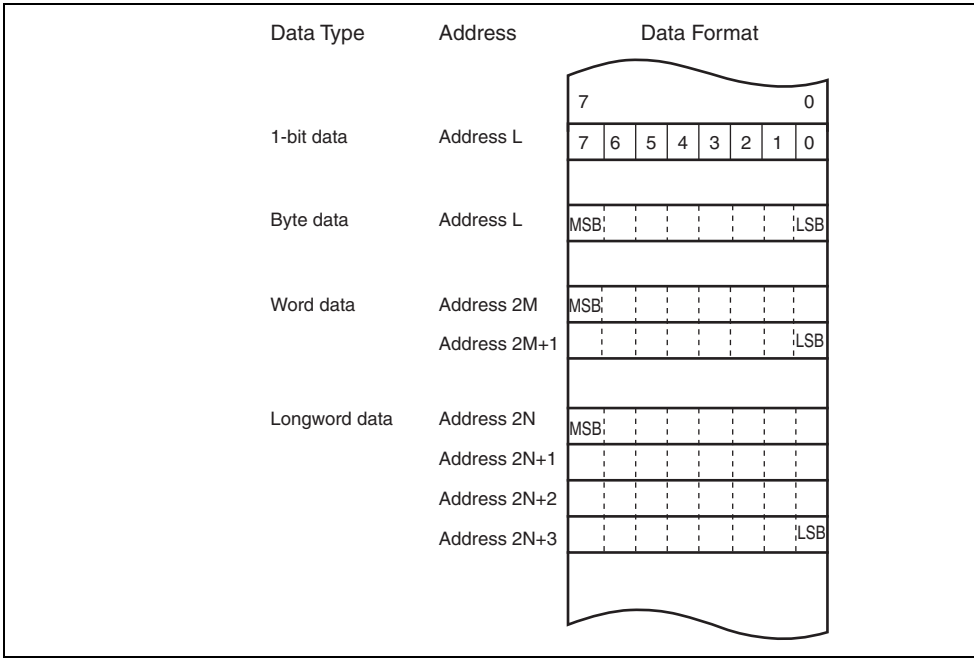


Figure 2.6 Memory Data Formats

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

MOVTPE	B	Rs → (EAs) Cannot be used in this LSI.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical t MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, E
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is identi MOV.W Rn, @–SP. PUSH.L ERn is identical to MOV.L ERn, @

[Legend]

B: Byte

W: Word

L: Longword

Note: * Refers to the operand size.

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

[Legend]

B: Byte

W: Word

L: Longword

Note: * Refers to the operand size.

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

[Legend]

B: Byte

W: Word

L: Longword

Note: * Refers to the operand size.

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

[Legend]

B: Byte

W: Word

L: Longword

Note: * Refers to the operand size.

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.

[Legend]

B: Byte

W: Word

L: Longword

Note: * Refers to the operand size.

		Inverts a specified bit in a general register or memory operand. number is specified by 3-bit immediate data or the lower three b general register.
BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operand 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 gene register or memory operand and stores the result in the carry fl The bit number is specified by 3-bit immediate data.
BOR	B	C ∨ (<bit-No.> of <EAd>) → C ORs the carry flag with a specified bit in a general register or m operand and stores the result in the carry flag.
BIOR	B	C ∨ ¬ (<bit-No.> of <EAd>) → C ORs the carry flag with the inverse of a specified bit in a genera or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

[Legend]

B: Byte

Note: * Refers to the operand size.

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

[Legend]

B: Byte

Note: * Refers to the operand size.

		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 by word access.
ANDC	B	CCR ∧ #IMM → CCR Logically ANDs the CCR with immediate data.
ORC	B	CCR ∨ #IMM → CCR Logically ORs the CCR with immediate data.
XORC	B	CCR ⊕ #IMM → CCR Logically XORs the CCR with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

[Legend]

B: Byte

W: Word

Note: * Refers to the operand size.

	else next;
	Transfers a data block. Starting from the address set in ER5, tr data for the number of bytes set in R4L or R4 to the address loc in ER6. Execution of the next instruction begins as soon as the transfer completed.

Some instructions have two operation fields.

- Register Field

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

- Effective Address Extension

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

- Condition Field

  Specifies the branching condition of Bcc instructions.

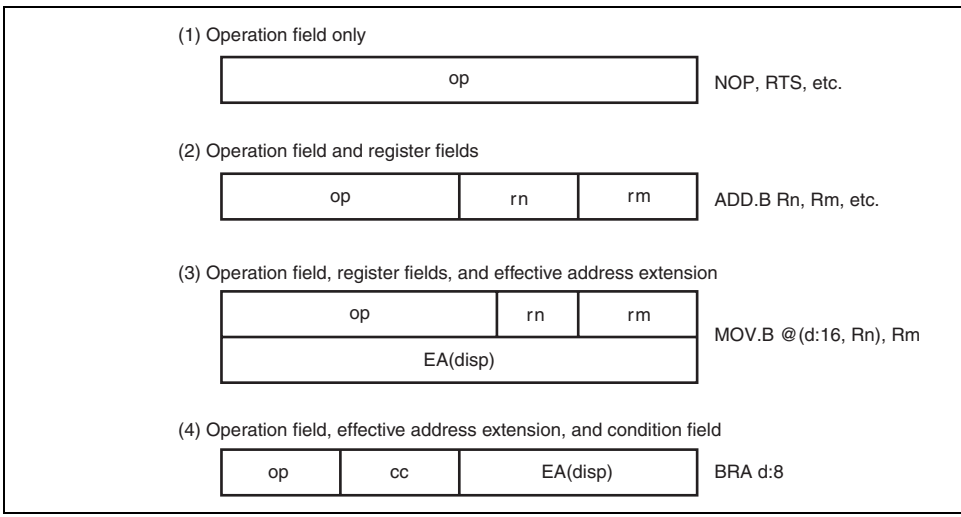


Figure 2.7 Instruction Formats

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

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

(1) Register Direct—Rn

The register field of the instruction specifies an 8-, 16-, or 32-bit general register containi
operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E
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 lowe
  of which contains the address of a memory operand. After the operand is accessed, 1
  added to the address register contents (32 bits) and the sum is stored in the address re
  The value added is 1 for byte access, 2 for word access, or 4 for longword access. Fo
  or longword access, the register value should be even.

- Register indirect with pre-decrement—@-ERn

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

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

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

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

The access ranges of absolute addresses for this LSI are those shown in table 2.11, becau
upper 8 bits are ignored.

operand.

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

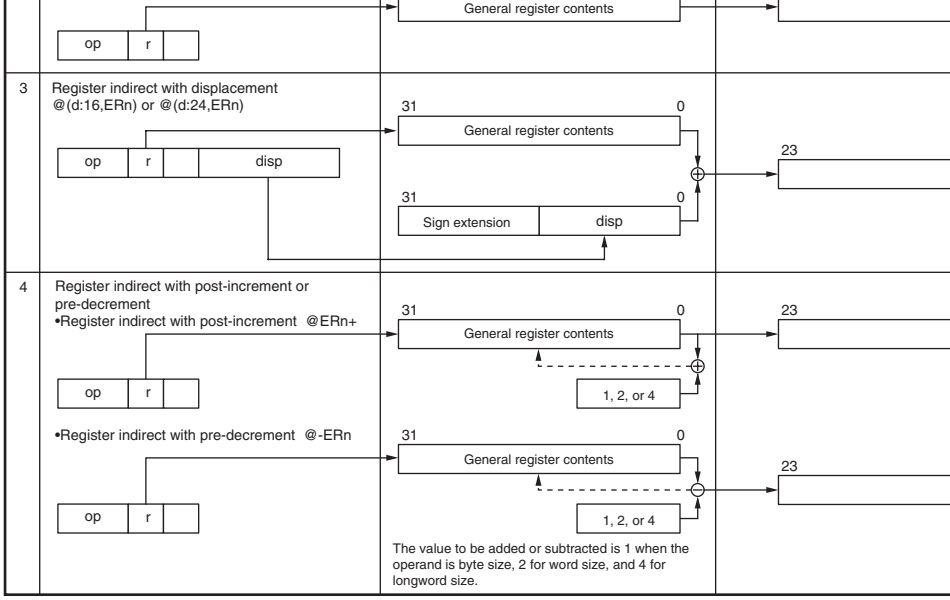
(7) Program-Counter Relative—@(d:8, PC) or @(d:16, PC)

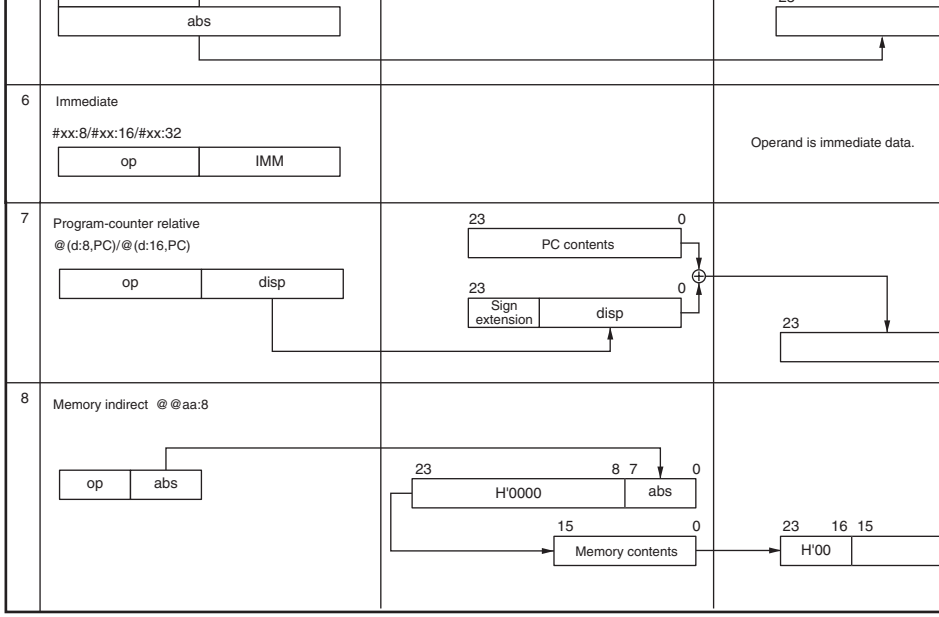
This mode is used in the BSR instruction. An 8-bit or 16-bit displacement contained in th
instruction is sign-extended and added to the 24-bit PC contents to generate a branch addr
PC value to which the displacement is added is the address of the first byte of the next ins
so the possible branching range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

(8) Memory Indirect—@@aa:8

This mode can be used by the JMP and JSR instructions. The instruction code contains an
absolute address specifying a memory operand. This memory operand contains a branch a
The memory operand is accessed by longword access. The first byte of the memory opera
ignored, generating a 24-bit branch address. Figure 2.8 shows how to specify branch addr
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.


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



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

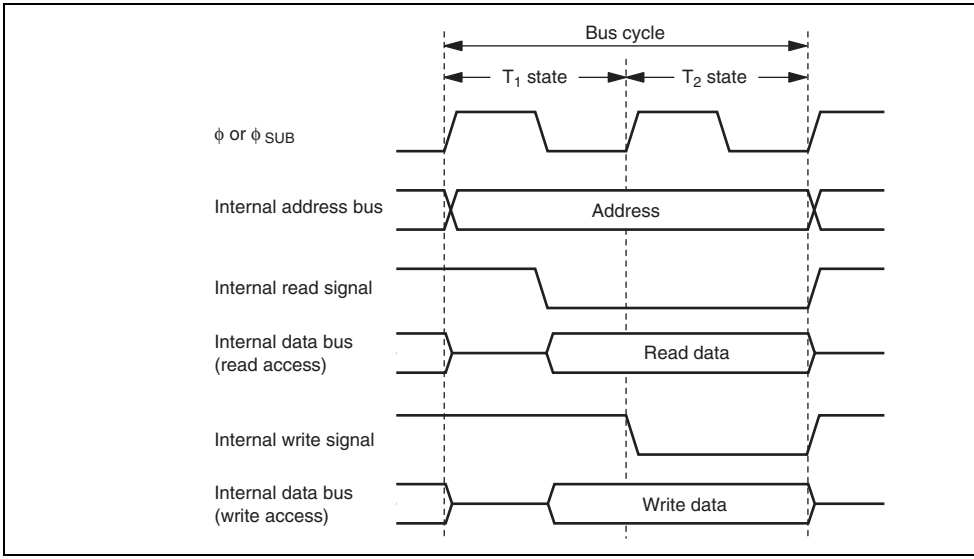


Figure 2.9 On-Chip Memory Access Cycle

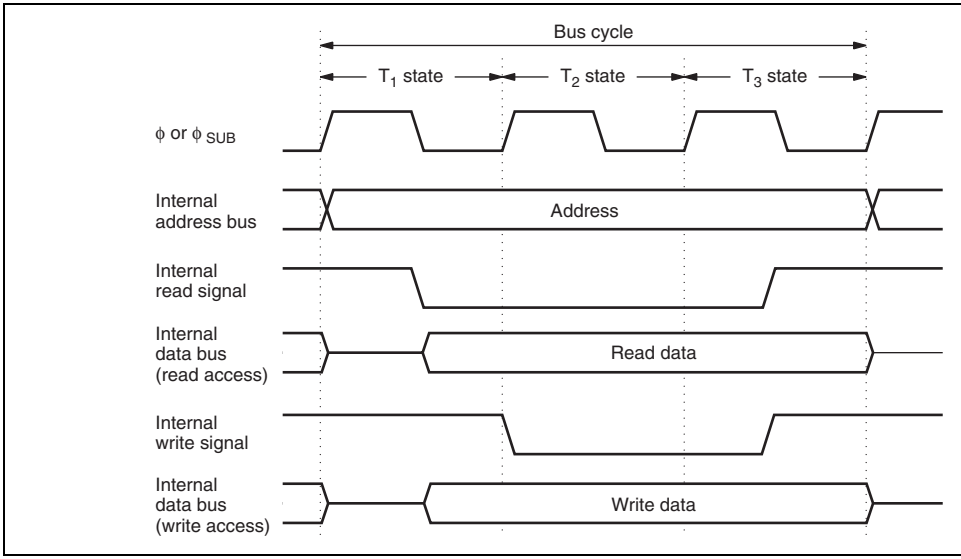


Figure 2.10 On-Chip Peripheral Module Access Cycle (3-State Access)

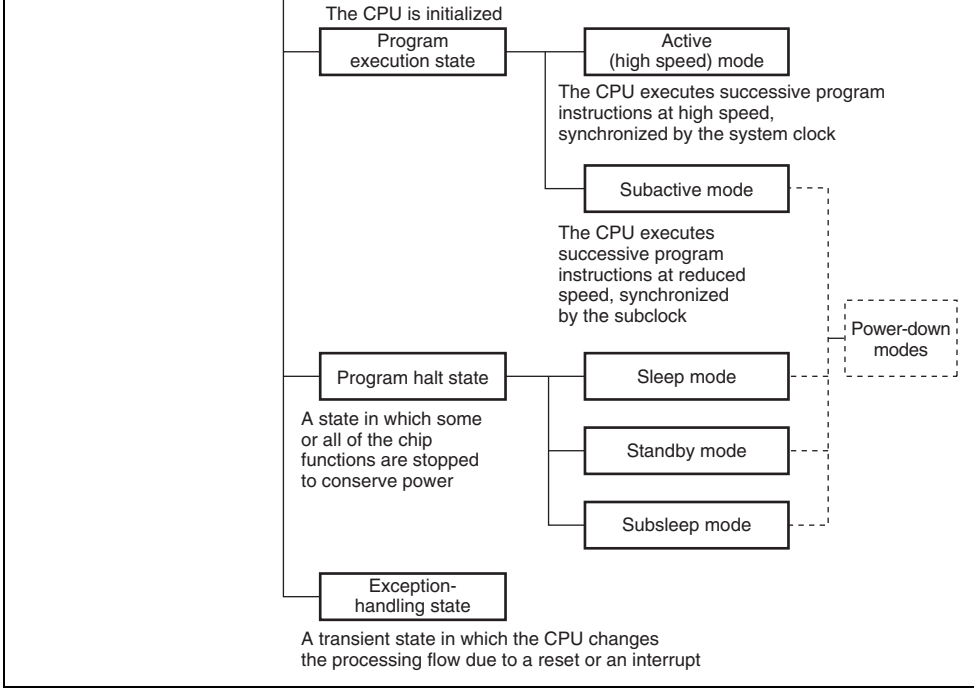


Figure 2.11 CPU Operation States

Figure 2.12 State Transitions

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

2.8.3 Bit-Manipulation Instruction

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

(1) Bit manipulation for two registers assigned to the same address

Example 1: Bit manipulation for the timer load register and timer counter

(Applicable for timer B1 in the H8/36057 Group and H8/36037 Group.)

Figure 2.13 shows an example of a timer in which two timer registers are assigned to the address. When a bit-manipulation instruction accesses the timer load register and timer co a reloadable timer, since these two registers share the same address, the following operati place.

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

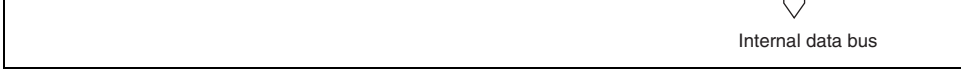


Figure 2.13 Example of Timer Configuration with Two Registers Allocated to Address

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 i P56. P55 to P50 are output pins and output low-level signals. An example to output a hi signal at P50 with a BSET instruction is shown below.

- Prior to 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	1	0	0	0	0	0	0

- BSET instruction executed instruction

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

The BSET instruction is executed for port 5.

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-lev
   input).

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

As a result of the BSET instruction, bit 0 in PDR5 becomes 1, and P50 outputs a high-lev
However, bits 7 and 6 of PDR5 end up with different values. To prevent this problem, sto
of the PDR5 data in a work area in memory. Perform the bit manipulation on the data in t
area, then write this data to PDR5.

- Prior to executing BSET instruction

```
MOV.B   #80,  R0L
MOV.B   R0L,  @RAM0
MOV.B   R0L,  @PDR5
```

The PDR5 value (H'80) is written to a work area in
memory (RAM0) as well as 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
RAM0	1	0	0	0	0	0	0

(2) Bit Manipulation in a Register Containing a Write-Only Bit

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

P57 and P56 are input pins, with a low-level signal input at P57 and a high-level signal i
P56. P55 to P50 are output pins that output low-level signals. An example of setting the
an input pin by the BCLR instruction is shown below. It is assumed that a high-level sig
input to this input pin.

- Prior to executing BCLR instruction

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

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

As a result of this operation, bit 0 in PCR5 becomes 0, making P50 an input port. Howev and 6 in PCR5 change to 1, so that P57 and P56 change from input pins to output pins. To this problem, store a copy of the PDR5 data in a work area in memory and manipulate da bit in the work area, then write this data to PDR5.

- Prior to executing BCLR instruction

```
MOV.B   #3F,   R0L
MOV.B   R0L,   @RAM0
MOV.B   R0L,   @PCR5
```

The PCR5 value (H'3F) is written to a work area in memory (RAM0) as well as 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

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 instruction generates a vector address corresponding to a vector number from 0 to 3, specified in the instruction code. Exception handling can be executed at all times in t program execution state, regardless of the setting of the I bit in CCR.

- Interrupts

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

—	Reserved for system use		1 to 6	H'0002 to H'000D
External interrupt pin	NMI		7	H'000E to H'000F
CPU	Trap instruction	(#0)	8	H'0010 to H'0011
		(#1)	9	H'0012 to H'0013
		(#2)	10	H'0014 to H'0015
		(#3)	11	H'0016 to H'0017
Address break	Break conditions satisfied		12	H'0018 to H'0019
CPU	Direct transition by executing the SLEEP instruction		13	H'001A to H'001B
External interrupt pin	IRQ0 Low-voltage detection interrupt*[1]		14	H'001C to H'001D
	IRQ1		15	H'001E to H'001F
	IRQ2		16	H'0020 to H'0021
	IRQ3		17	H'0022 to H'0023
	WKP		18	H'0024 to H'0025
—	Reserved for system use		19	H'0026 to H'0027
			20	H'0028 to H'0029
Timer V	Timer V compare match A Timer V compare match B Timer V overflow		22	H'002C to H'002D
SCI3	SCI3 receive data full SCI3 transmit data empty SCI3 transmit end SCI3 receive error		23	H'002E to H'002F
—	Reserved for system use		24	H'0030 to H'0031
A/D converter	A/D conversion end		25	H'0032 to H'0033

	Transmit data empty Transmit end Receive error		
TinyCAN	Error Reset/HALT mode processing Message reception Message transmission Wakeup	34	H'0044 to H'0045
SSU	Overrun error Transmit data empty Transmit end Receive data full Conflict error	35	H'0046 to H'0047
Subtimer	Underflow	36	H'0048 to H'0049

Notes: 1. A low-voltage detection interrupt is enabled only in the product with an on-chi
on reset and low-voltage detection circuit.
2. The H8/36037 Group does not have the SCI3_2.

- Wakeup interrupt flag register (IWPR)

3.2.1 Interrupt Edge Select Register 1 (IEGR1)

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

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

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

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

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

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

3.2.5 Interrupt Flag Register 1 (IRR1)

IRR1 is a status flag register for direct transition interrupts and IRQ3 to IRQ0 interrupt r

Bit	Bit Name	Initial Value	R/W	Description
7	IRRDT	0	R/W	Direct Transfer Interrupt Request Flag [Setting condition] When a direct transfer is made by executing a S instruction while DTON in SYSCR2 is set to 1. [Clearing condition] When IRRDT is cleared by writing 0
6	—	0	—	Reserved This bit is always read as 0.
5, 4	—	All 1	—	Reserved These bits are always read as 1.

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

				[Clearing condition] When IRRTB1 is cleared by writing 0
4 to 0	—	All 1	—	Reserved These bits are always read as 1.

3.2.7 Wakeup Interrupt Flag Register (IWPR)

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

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

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

The reset exception handling sequence is as follows:

1. Set the I bit in the condition code register (CCR) to 1.
2. The CPU generates a reset exception handling vector address (from H'0000 to H'000 data in that address is sent to the program counter (PC) as the start address, and prog execution starts from that address.

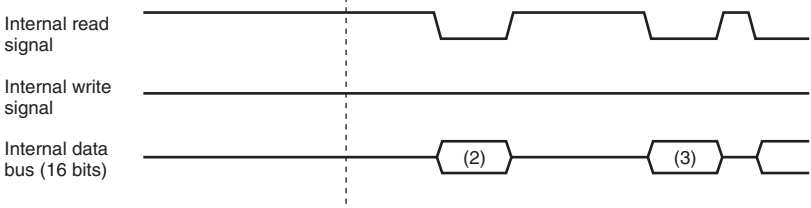
value in CCR.

IRQ3 to IRQ0 Interrupts: IRQ3 to IRQ0 interrupts are requested by input signals to pin
to $\overline{\text{IRQ0}}$. These four interrupts are given different vector addresses, and are detected indiv
by either rising edge sensing or falling edge sensing, depending on the settings of bits IE
IEG0 in IEGR1.

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

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

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



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

Figure 3.1 Reset Sequence

3.4.2 Internal Interrupts

Each on-chip peripheral module has a flag to show the interrupt request status and the en
enable or disable the interrupt. For direct transfer interrupt request generated by executio
SLEEP instruction, this function is included in IRR1, IRR2, IENR1, and IENR2.

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

3. The CPU accepts the NMI and address break without depending on the I bit value. Ot interrupt requests are accepted, if the I bit is cleared to 0 in CCR; if the I bit is set to 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.

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 shows a port mode register setting and interrupt request flag clearing procedu

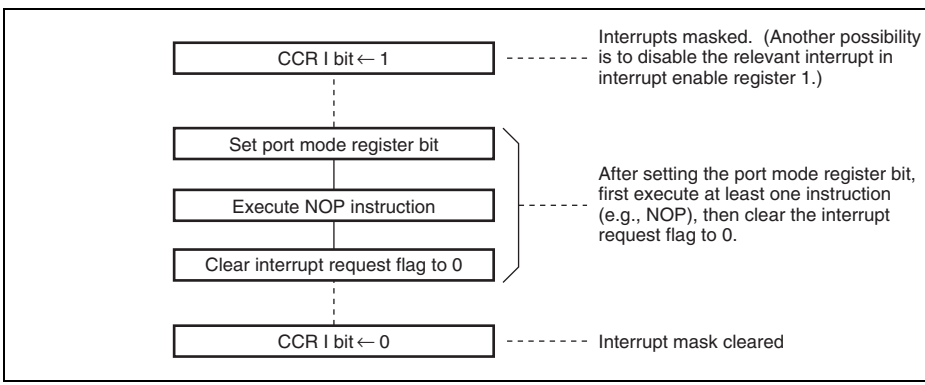
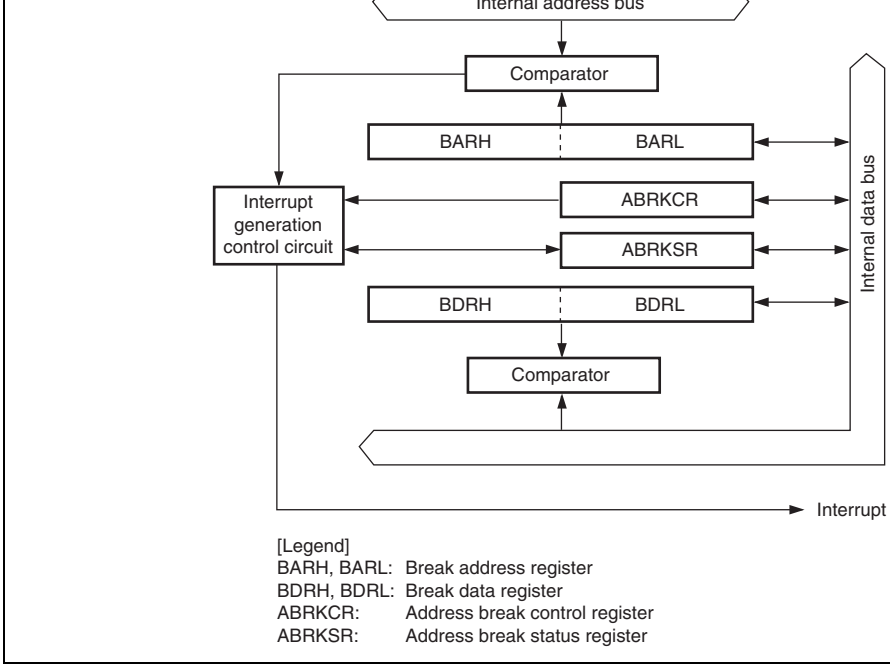


Figure 3.4 Port Mode Register Setting and Interrupt Request Flag Clearing Pr



[Legend]
BARH, BARL: Break address register
BDRH, BDRL: Break data register
ABRKCR: Address break control register
ABRKSR: Address break status register

Figure 4.1 Block Diagram of Address Break

ABRKCR sets address break conditions.

Bit	Bit Name	Initial Value	R/W	Description
7	RTINTE	1	R/W	RTE Interrupt Enable When this bit is 0, the interrupt immediately after executing RTE is masked and then one instructi be executed. When this bit is 1, the interrupt is n masked.
6	CSEL1	0	R/W	Condition Select 1 and 0 These bits set address break conditions. 00: Instruction execution cycle 01: CPU data read cycle 10: CPU data write cycle 11: CPU data read/write cycle
5	CSEL0	0	R/W	
4	ACMP2	0	R/W	Address Compare Condition Select 2 to 0 These bits set the comparison condition between address set in BAR and the internal address bus 000: Compares 16-bit addresses 001: Compares upper 12-bit addresses 010: Compares upper 8-bit addresses 011: Compares upper 4-bit addresses 1XX: Reserved (setting prohibited)
3	ACMP1	0	R/W	
2	ACMP0	0	R/W	

[Legend]
X: Don't care.

When an address break is set in the data read cycle or data write cycle, the data bus used depend on the combination of the byte/word access and address. Table 4.1 shows the ac data bus used. When an I/O register space with an 8-bit data bus width is accessed in wo byte access is generated twice. For details on data widths of each register, see section 21 Register Addresses (Address Order).

Table 4.1 Access and Data Bus Used

	Word Access		Byte Access	
	Even Address	**Odd Address**	**Even Address**	**Odd**
ROM space	Upper 8 bits	Lower 8 bits	Upper 8 bits	Uppe
RAM space	Upper 8 bits	Lower 8 bits	Upper 8 bits	Uppe
I/O register with 8-bit data bus width	Upper 8 bits	Upper 8 bits	Upper 8 bits	Uppe
I/O register with 16-bit data bus width	Upper 8 bits	Lower 8 bits	—	—

				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 rec enabled.
5 to 0	—	All 1	—	Reserved These bits are always read as 1.

4.1.3 Break Address Registers (BARH, BARL)

BARH and BARL are 16-bit readable/writable registers that set the address for generating address break interrupt. When setting the address break condition to the instruction execu cycle, set the first byte address of the instruction. The initial value of this register is H'FF

4.1.4 Break Data Registers (BDRH, BDRL)

BDRH and BDRL are 16-bit readable/writable registers that set the data for generating an break interrupt. BDRH is compared with the upper 8-bit data bus. BDRL is compared wit lower 8-bit data bus. When memory or registers are accessed by byte, the upper 8-bit data used for even and odd addresses in the data transmission. Therefore, comparison data mu in BDRH for byte access. For word access, the data bus used depends on the address. See 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.

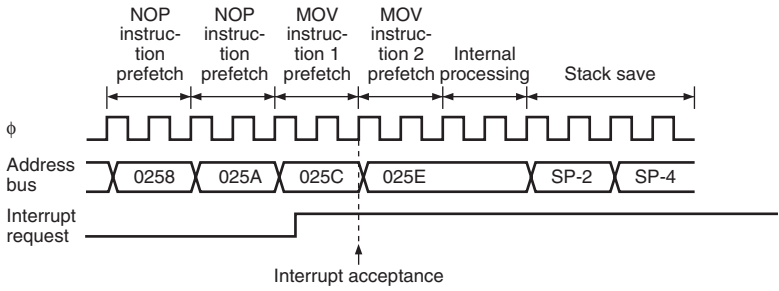


Figure 4.2 Address Break Interrupt Operation Example (1)

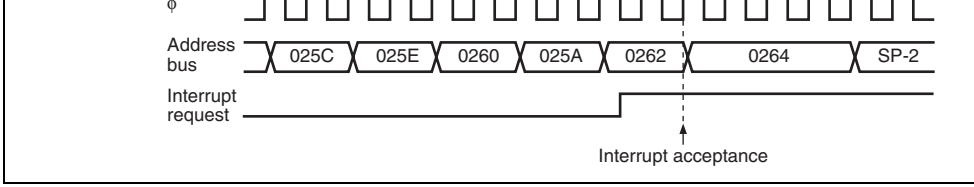


Figure 4.2 Address Break Interrupt Operation Example (2)

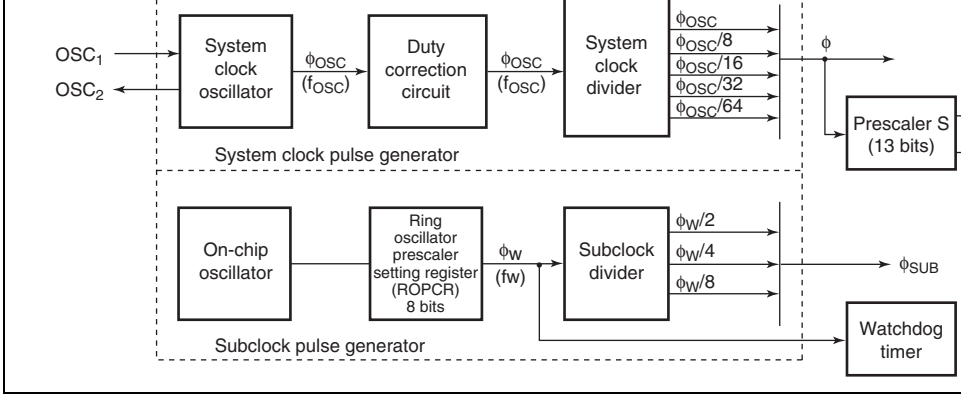


Figure 5.1 Block Diagram of Clock Pulse Generators

The basic clock signals that drive the CPU and on-chip peripheral modules are system c and subclocks (ϕ_{SUB}). The system clock is divided by the prescaler S to become a clock s $\phi/8192$ to $\phi/2$, which is provided to the on-chip peripheral modules. The output (ϕ_w) of t ratio setting register (ROPCR) for the on-chip clock pulse generator can be used as one input clocks for the watchdog timer.

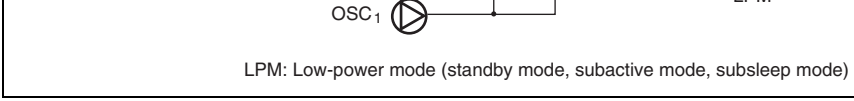


Figure 5.2 Block Diagram of System Clock Generator

5.1.1 Connecting Crystal Resonator

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

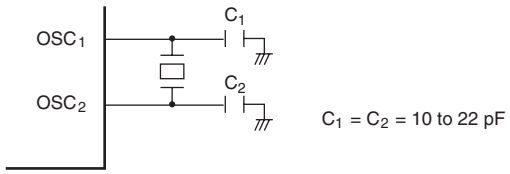


Figure 5.3 Typical Connection to Crystal Resonator

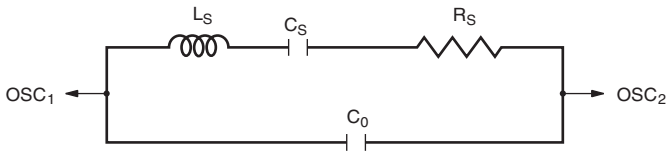


Figure 5.4 Equivalent Circuit of Crystal Resonator

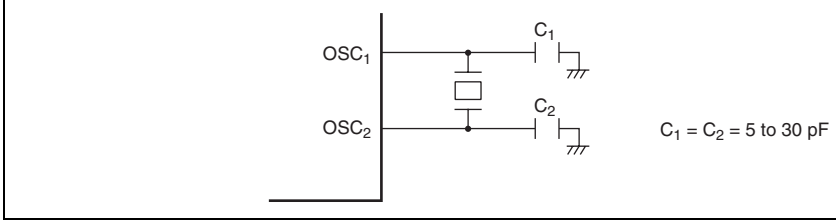


Figure 5.5 Typical Connection to Ceramic Resonator

5.1.3 External Clock Input Method

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

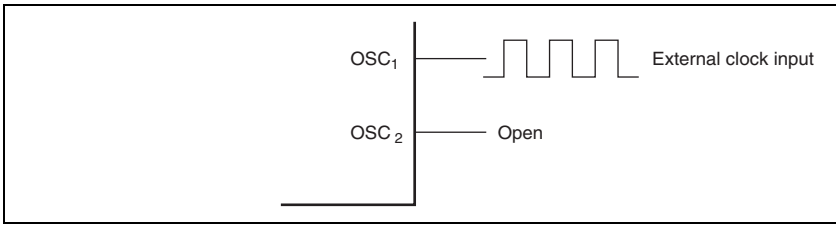


Figure 5.6 Example of External Clock Input

the clock input to prescaler S is determined by the division ratio designated by the MA2 t
bits in SYSCR2.

5.3 Usage Notes

5.3.1 Note on Resonators

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

5.3.2 Notes on Board Design

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

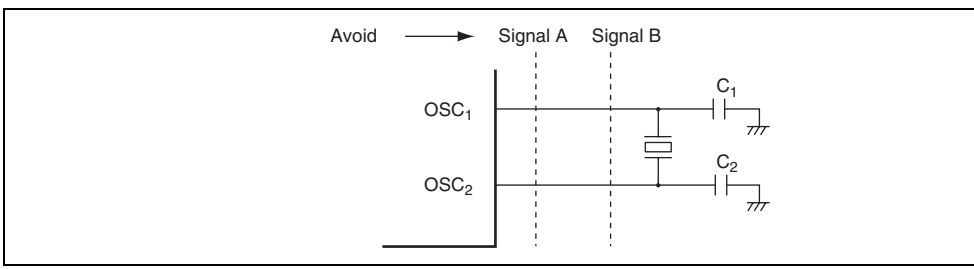


Figure 5.7 Example of Incorrect Board Design

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

- Sleep mode

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

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

  The CPU and all on-chip peripheral modules halt.
- 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: Enters sleep mode or subsleep mode. 1: Enters standby mode. For details, see table 6.2.
6	STS2	0	R/W	Standby Timer Select 2 to 0 These bits designate the time the CPU and perip modules wait for stable clock operation after exit standby mode, subactive mode, or subsleep mo active mode or sleep mode due to an interrupt. T designation should be made according to the clo frequency so that the waiting time is at least 6.5 relationship between the specified value and the of wait states is shown in table 6.1. When an ext clock is to be used, the minimum value (STS2 = STS0 =1) is recommended.
5	STS1	0	R/W	
4	STS0	0	R/W	
3 to 0	—	All 0	—	Reserved These bits are always read as 0.

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

Note: Time unit is ms.

6.1.2 System Control Register 2 (SYSCR2)

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

Bit	Bit Name	Initial Value	R/W	Description
7	SMSEL	0	R/W	Sleep Mode Selection
6	LSON	0	R/W	Low Speed on Flag
5	DTON	0	R/W	Direct Transfer on Flag These bits select the mode to enter after the ex a SLEEP instruction, as well as the SSBY bit in 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
2	MA0	0	R/W	and sleep modes. The operating clock frequenc changes to the set frequency after the SLEEP i is executed. 0XX: ϕ_{osc} 100: $\phi_{osc}/8$ 101: $\phi_{osc}/16$ 110: $\phi_{osc}/32$ 111: $\phi_{osc}/64$

[Legend]
X: Don't care.

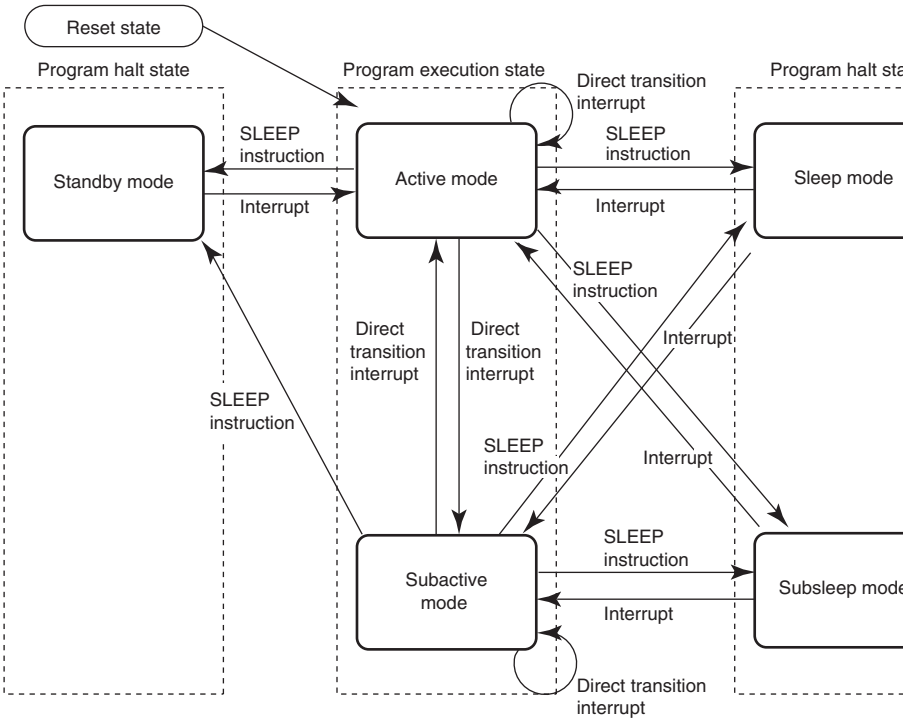
6.1.3 Module Standby Control Register 1 (MSTCR1)

MSTCR1 allows the on-chip peripheral modules to enter a standby state in module units.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0.
5	MSTS3	0	R/W	SCI3 Module Standby The SCI3 enters standby mode when this bit is s
4	MSTAD	0	R/W	A/D Converter Module Standby The A/D converter enters standby mode when th set to 1.
3	MSTWD	0	R/W	Watchdog Timer Module Standby The watchdog timer enters standby mode when set to 1. When the internal oscillator is selected f watchdog timer clock, the watchdog timer operat regardless of the setting of this bit.
2	—	0	—	Reserved This bit is always read as 0.
1	MSTTV	0	R/W	Timer V Module Standby The timer V enters standby mode when this bit is
0	—	0	—	Reserved This bit is always read as 0.

6, 5	—	All 0	—	Reserved These bits are always read as 0.
4	MSTTB1	0	R/W	Timer B1 Module Standby The timer B1 enters standby mode when this bi 1.
3, 2	—	All 0	—	Reserved These bits are always read as 0.
1	MSTTZ	0	R/W	Timer Z Module Standby The timer Z enters standby mode when this bit
0	—	0	—	Reserved This bit is always read as 0.

by an interrupt. Table 6.3 shows the internal states of the LSI in each mode.



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

Figure 6.1 Mode Transition Diagram

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

[Legend]

X: Don’t care.

Note: * When a state transition is made while SMSEL is 1, the timer V, SCI3, SCI3_2 the H8/36057 Group), and A/D converter are reset, and all registers are set to initial values. To use these functions after entering active mode, reset the reg

Table 6.3 Internal State in Each Operating Mode

Function		Active Mode	Sleep Mode	Subactive Mode	Subsleep Mode	Stand
System clock oscillator		Functioning	Functioning	Halted	Halted	Halted
CPU operations	Instructions	Functioning	Halted	Functioning	Halted	Halted
	Registers	Functioning	Retained	Functioning	Retained	Retain
RAM		Functioning	Retained	Functioning	Retained	Retain
I/O ports		Functioning	Retained	Functioning	Retained	Regist are re output imped
External interrupts	IRQ3 to IRQ0	Functioning	Functioning	Functioning	Functioning	Functi
	WKP5 to WKP0	Functioning	Functioning	Functioning	Functioning	Functi

					the on oscilla enable
Timer B1	Functioning	Functioning	Retained*	Retained	Retain
Timer Z	Functioning	Functioning	Retained (When internal clock ϕ is sel count clock, the counter counts up wit clock*.)		
A/D converter	Functioning	Functioning	Reset	Reset	Reset

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

6.2.1 Sleep Mode

In sleep mode, CPU operation is halted but the on-chip peripheral modules function at the frequency set by the MA2, MA1, and MA0 bits in SYSCR2. CPU register contents are re When an interrupt is requested, sleep mode is cleared and interrupt exception handling sta Sleep mode is not cleared if the I bit of the condition code register (CCR) is set to 1 or the requested interrupt is disabled in the interrupt enable register. After sleep mode is cleared transition is made to active mode when the LSON bit in SYSCR2 is 0, and a transition is subactive mode when the bit is 1.

When the $\overline{\text{RES}}$ pin goes low, the CPU goes into the reset state and sleep mode is cleared.

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 co will be retained as long as the voltage set by the RAM data retention voltage is provided. ports go to the high-impedance state.

In subsleep mode, operation of the CPU and on-chip peripheral modules is halted. As lo
required voltage is applied, the contents of CPU registers, the on-chip RAM, and some r
the on-chip peripheral modules are retained. I/O ports keep the same states as before the

Subsleep mode is cleared by an interrupt. When an interrupt is requested, subsleep mode
and interrupt exception handling starts. Subsleep mode is not cleared if the I bit of CCR
or the requested interrupt is disabled in the interrupt enable register. After subsleep mod
cleared, a transition is made to active mode when the LSON bit in SYSCR2 is 0, and a t
made to subactive mode when the bit is 1. After the time set in bits STS2 to STS0 in SY
elapsed, a transition is made to active mode.

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

6.2.4 Subactive Mode

The operating frequency of subactive mode is selected from $\phi_w/2$, $\phi_w/4$, and $\phi_w/8$ by the
SA0 bits in SYSCR2. After the SLEEP instruction is executed, the operating frequency
the frequency which is set before the execution. When the SLEEP instruction is execute
subactive mode, a transition to sleep mode, subsleep mode, standby mode, active mode,
subactive mode is made, depending on the combination of SYSCR1 and SYSCR2. Whe
pin goes low, the system clock pulse generator starts. Since system clock signals are sup
the entire chip as soon as the system clock pulse generator starts functioning, the $\overline{\text{RES}}$ p
kept low until the pulse generator output stabilizes. After the pulse generator output has
the CPU starts reset exception handling if the $\overline{\text{RES}}$ pin is driven high.

be made by executing a SLEEP instruction while the DTON bit in SYSCR2 is set to 1. T
transition also enables operating frequency modification in active or subactive mode. Aft
mode transition, direct transition interrupt exception handling starts.

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

6.4.1 Direct Transition from Active Mode to Subactive Mode

The time from the start of SLEEP instruction execution to the end of interrupt exception h
(the direct transition time) is calculated by equation (1).

Direct transition time = {(number of SLEEP instruction execution states) + (number of in
processing states)}× (tcyc before transition) + (number of interrupt exception handling st
(tsubcyc after transition) (1)

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

[Legend]
tosc: OSC clock cycle time
tw: Watch clock cycle time
tcyc: System clock (ϕ) cycle time
tsubcyc: Subclock (ϕ_{SUB}) cycle time

(when the CPU operating clock of $\phi_w/8 \rightarrow \phi_{osc}$ and a waiting time of 8192 states are s

[Legend]
tosc: OSC clock cycle time
tw: Watch clock cycle time
tcyc: System clock (ϕ) cycle time
tsubcyc: Subclock (ϕ_{SUB}) cycle time

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 in TCMR, SSCRL,
MSTCR1 that corresponds to each module to 1 and cancels the standby state by clearing
0.

flash memory, each block must be erased in turn.

- Reprogramming capability
  - The flash memory can be reprogrammed up to 1,000 times.
- On-board programming
  - On-board programming/erasing can be done in boot mode, in which the boot pro into the chip is started to erase or program of the entire flash memory. In normal program mode, individual blocks can be erased or programmed.
- Programmer mode
  - Flash memory can be programmed/erased in programmer mode using a PROM programmer, as well as in on-board programming mode.
- Automatic bit rate adjustment
  - For data transfer in boot mode, this LSI's bit rate can be automatically adjusted to the transfer bit rate of the host.
- Programming/erasing protection
  - Sets software protection against flash memory programming/erasing.
- Power-down mode
  - Operation of the power supply circuit can be partly halted in subactive mode. As flash memory can be read with low power consumption.

7.1 Block Configuration

Figure 7.1 shows the block configuration of flash memory. The thick lines indicate erasi the narrow lines indicate programming units, and the values are addresses. The 56-kbyte memory is divided into 1 kbyte × 4 blocks, 28 kbytes × 1 block, 16 kbytes × 1 block, an × 1 block. The 32-kbyte flash memory is divided into 1 kbyte × 4 blocks and 28 kbytes × Erasing is performed in these units. Programming is performed in 128-byte units starting address with lower eight bits H'00 or H'80.

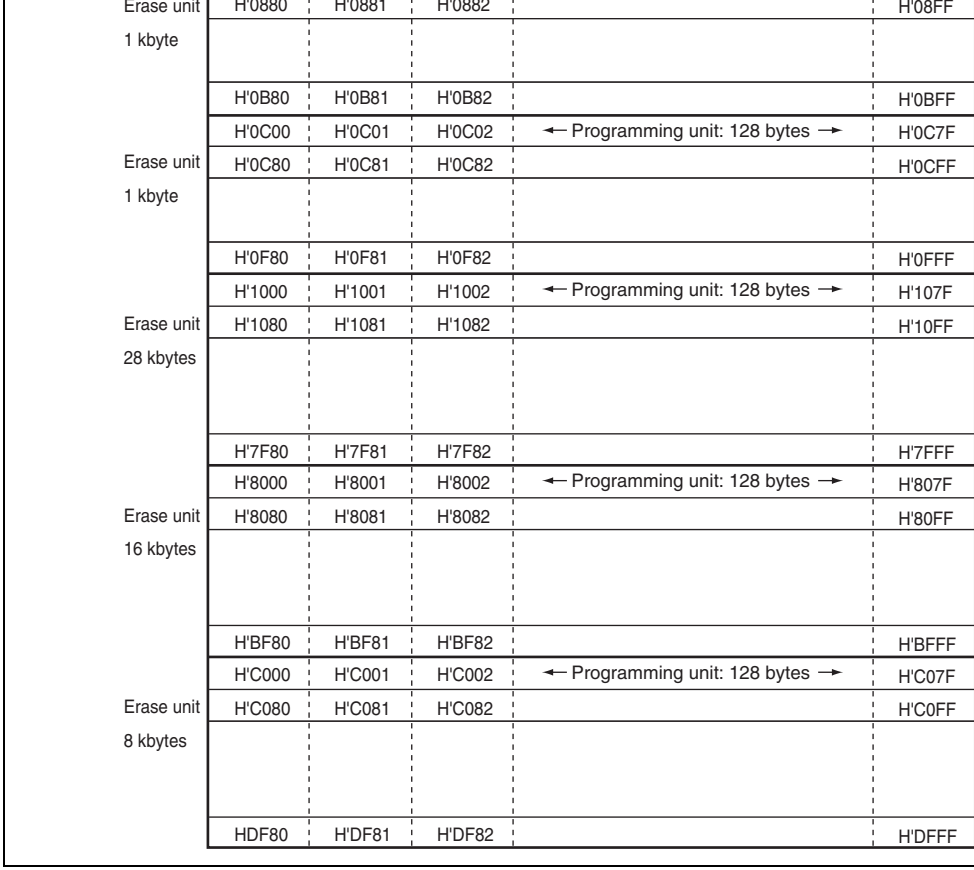


Figure 7.1 Flash Memory Block Configuration

7.2.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is a register that makes the flash memory change to program mode, program-v mode, erase mode, or erase-verify mode. For details on register setting, refer to section 7 Memory Programming/Erasing.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.
6	SWE	0	R/W	Software Write Enable When this bit is set to 1, flash memory programming/erasing is enabled. When this bit to 0, other FLMCR1 register bits and all EBR1 b be set.
5	ESU	0	R/W	Erase Setup When this bit is set to 1, the flash memory char erase setup state. When it is cleared to 0, the e state is cancelled. Set this bit to 1 before setting to 1 in FLMCR1.
4	PSU	0	R/W	Program Setup When this bit is set to 1, the flash memory char program setup state. When it is cleared to 0, th setup state is cancelled. Set this bit to 1 before P bit in FLMCR1.
3	EV	0	R/W	Erase-Verify When this bit is set to 1, the flash memory char erase-verify mode. When it is cleared to 0, eras mode is cancelled.

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

7.2.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is a register that displays the state of flash memory programming/erasing. FLM read-only register, and should not be written to.

Bit	Bit Name	Initial Value	R/W	Description
7	FLER	0	R	Flash Memory Error Indicates that an error has occurred during an op on flash memory (programming or erasing). Whe is set to 1, flash memory goes to the error-protec state. See section 7.5.3, Error Protection, for details.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

6	EB6	0	R/W	When this bit is set to 1, 8 bytes of H'C000 to H be erased.
5	EB5	0	R/W	When this bit is set to 1, 16 bytes of H'8000 to H will be erased.
4	EB4	0	R/W	When this bit is set to 1, 28 kbytes of H'1000 to will be erased.
3	EB3	0	R/W	When this bit is set to 1, 1 kbyte of H'0C00 to H be erased.
2	EB2	0	R/W	When this bit is set to 1, 1 kbyte of H'0800 to H be erased.
1	EB1	0	R/W	When this bit is set to 1, 1 kbyte of H'0400 to H be erased.
0	EB0	0	R/W	When this bit is set to 1, 1 kbyte of H'0000 to H be erased.

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

7.2.5 Flash Memory Enable Register (FENR)

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

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

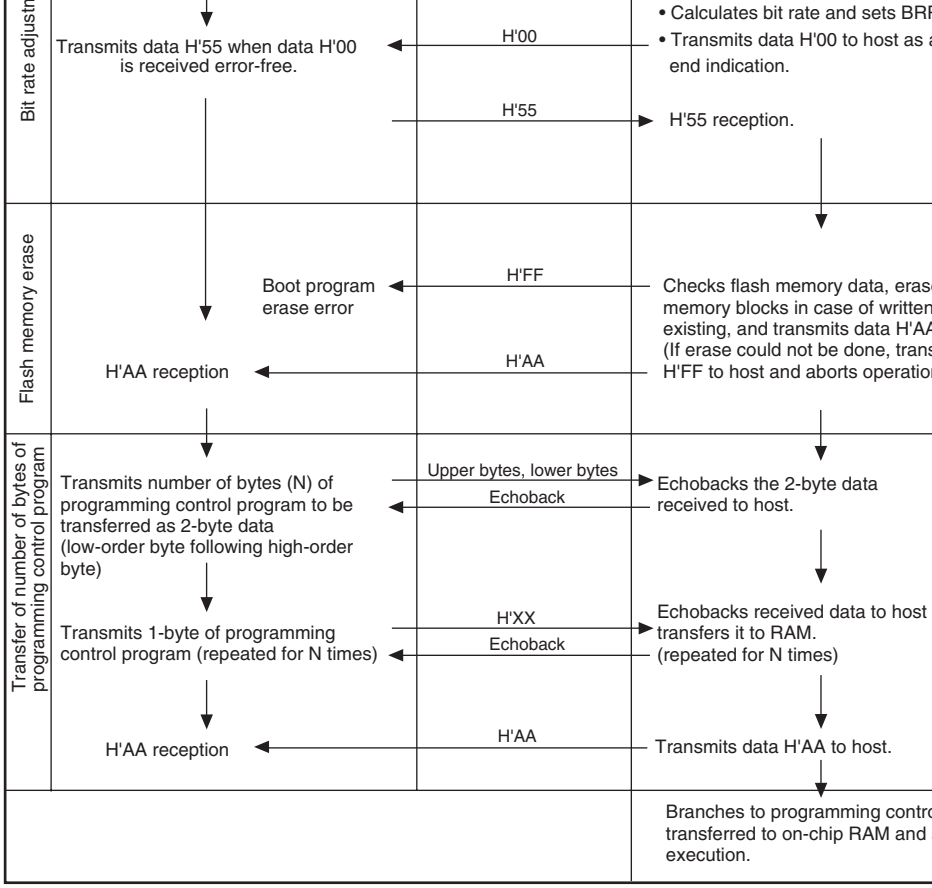
via the SCI3. After erasing the entire flash memory, the programming control program i
executed. This can be used for programming initial values in the on-board state or for a
return when programming/erasing can no longer be done in user program mode. In user
mode, individual blocks can be erased and programmed by branching to the user progra
control program prepared by the user.

Table 7.1 Setting Programming Modes

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

[Legend]
X: Don’t care.

3. When the boot program is initiated, the chip measures the low-level period of asynchr SCI communication data (H'00) transmitted continuously from the host. The chip 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 sho 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 reception 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 I 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 the SCI3 (by clearing the RE and TE bits in SCR to 0), however the adjusted bit ra remains set in BRR. Therefore, the programming control program can still use it for t program data or verify data with the host. The TXD pin is high (PCR22 = 1, P22 = 1) contents of the CPU general registers are undefined immediately after branching to th programming control program. These registers must be initialized at the beginning of programming control program, as the stack pointer (SP), in particular, is used implicit subroutine calls, etc.
7. Boot mode can be cleared by a reset. End the reset after driving the reset pin low, wai least 20 states, and then setting the $\overline{\text{NMI}}$ pin. Boot mode is also cleared when a WDT occurs.
8. Do not change the TEST pin and $\overline{\text{NMI}}$ pin input levels in boot mode.


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

7.3.2 Programming/Erasing in User Program Mode

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

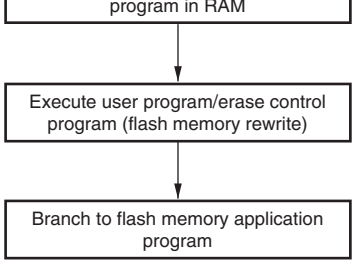
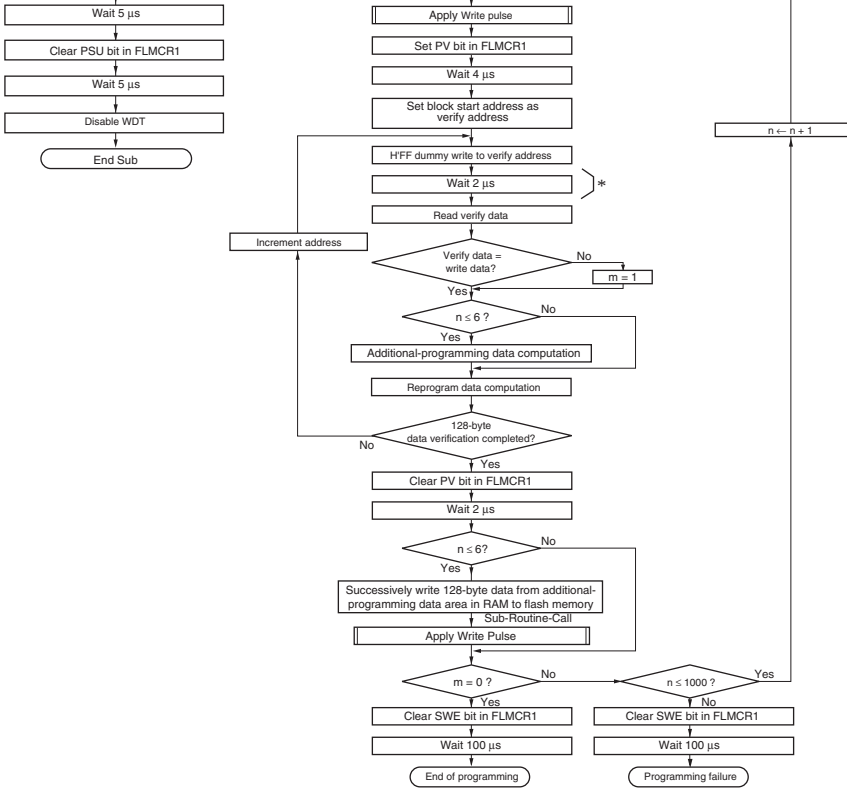


Figure 7.2 Programming/Erasing Flowchart Example in User Program Mo

7.4.1 Program/Program-Verify

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

1. Programming must be done to an empty address. Do not reprogram an address to whi programming has already been performed.
2. Programming should be carried out 128 bytes at a time. A 128-byte data transfer mus performed even if writing fewer than 128 bytes. In this case, H'FF data must be writte extra addresses.
3. Prepare the following data storage areas in RAM: A 128-byte programming data area byte reprogramming data area, and a 128-byte additional-programming data area. Per reprogramming data computation according to table 7.4, and additional programming computation according to table 7.5.
4. Consecutively transfer 128 bytes of data in byte units from the reprogramming data ar additional-programming data area to the flash memory. The program address and 128 data are latched in the flash memory. The lower 8 bits of the start address in the flash destination area must be H'00 or H'80.
5. The time during which the P bit is set to 1 is the programming time. Table 7.6 shows t allowable programming times.
6. The watchdog timer (WDT) is set to prevent overprogramming due to program runaw An overflow cycle of approximately 6.6 ms is allowed.
7. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lo are B'00. Verify data can be read in words or in longwords from the address to which write was performed.



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

Figure 7.3 Program/Program-Verify Flowchart

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

Table 7.6 Programming Time

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

Note: Time shown in μs.

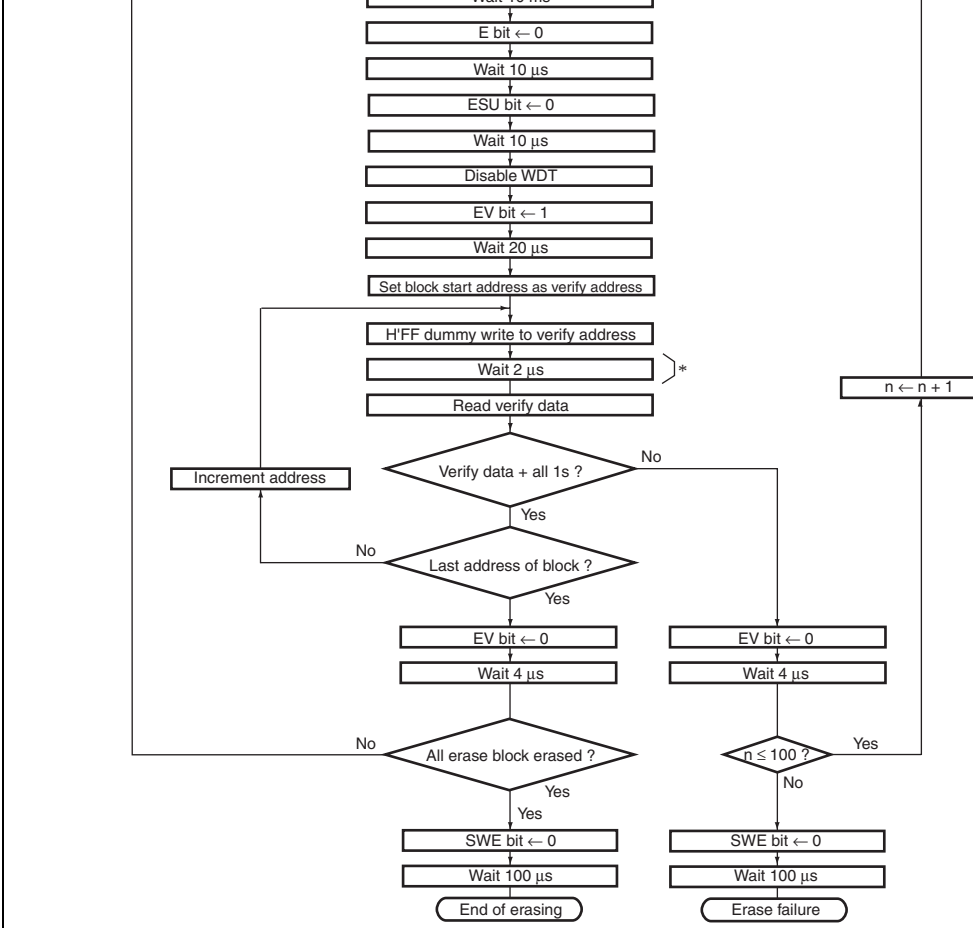
overflow cycle of approximately 19.8 ms is allowed.

5. For a dummy write to a verify address, write 1-byte data H'FF to an address whose l bits are B'00. Verify data can be read in longwords from the address to which a dum was performed.
6. If the read data is not erased successfully, set erase mode again, and repeat the erase/ verify sequence as before. The maximum number of repetitions of the erase/erase-ve sequence is 100.

7.4.3 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including the NMI interrupt, are disabled while flash memory is being pr or erased, or while the boot program is executing, for the following three reasons:

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



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

Figure 7.4 Erase/Erase-Verify Flowchart

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

7.5.2 Software Protection

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

7.5.3 Error Protection

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

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

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

7.7 Power-Down States for Flash Memory

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

- Normal operating mode
  The flash memory can be read and written to at high speed.
- Power-down operating mode
  The power supply circuit of flash memory can be partly halted. As a result, flash mem
  be read with low power consumption.
- Standby mode
  All flash memory circuits are halted.

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

Masked ROM version	H8/36057, H8/36037	2 kbytes	H'EC00 to H'EFFF, H'FB80 to H
	H8/36036	2 kbytes	H'EC00 to H'EFFF, H'FB80 to H
	H8/36035	2 kbytes	H'EC00 to H'EFFF, H'FB80 to H
	H8/36054, H8/36034	2 kbytes	H'EC00 to H'EFFF, H'FB80 to H
	H8/36033	1 kbyte	H'FB80 to H'FF7F
	H8/36032	1 kbyte	H'FB80 to H'FF7F

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

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

9.1 Port 1

Port 1 is a general I/O port also functioning as IRQ interrupt input pins, a timer B1 inpu
timer V input pin. Figure 9.1 shows its pin configuration.

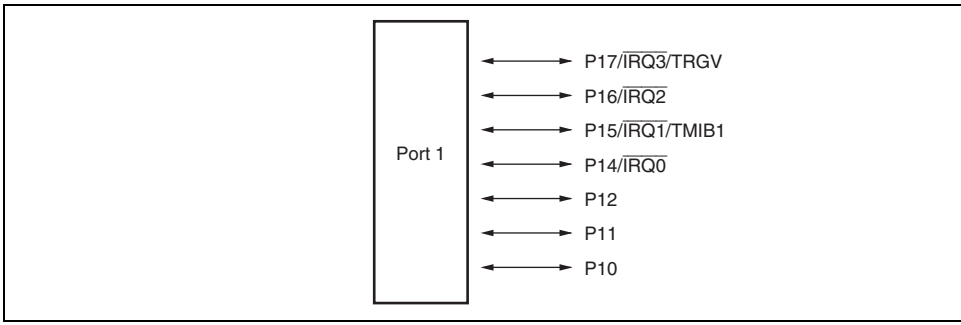


Figure 9.1 Port 1 Pin Configuration

Port 1 has the following registers.

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

				0: General I/O port 1: $\overline{\text{IRQ2}}$ input pin
5	IRQ1	0	R/W	This bit selects the function of pin P15/$\overline{\text{IRQ1}}$/TMI 0: General I/O port 1: $\overline{\text{IRQ1}}$/TMIB1 input pin
4	IRQ0	0	R/W	This bit selects the function of pin P14/$\overline{\text{IRQ0}}$. 0: General I/O port 1: $\overline{\text{IRQ0}}$ input pin
3	TXD2	0	R/W	This bit selects the function of pin P72/TXD_2. 0: General I/O port 1: TXD_2 output pin Note: This bit is reserved in the H8/36037 Group is always read as 0.
2	—	0	—	Reserved. This bit is always read as 0.
1	TXD	0	R/W	This bit selects the function of pin P22/TXD. 0: General I/O port 1: TXD output pin
0	—	0	—	Reserved. This bit is always read as 0.

3	—	—	—
2	PCR12	0	W
1	PCR11	0	W
0	PCR10	0	W

9.1.3 Port Data Register 1 (PDR1)

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

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

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

9.1.5 Pin Functions

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

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

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

[Legend]
X: Don't care.

- P15/IRQ1/TMIB1 pin

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

[Legend]
X: Don't care.

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

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

[Legend]
X: Don't care.

- P12 pin

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

Setting value	0	P10 input pin
	1	P10 output pin

9.2 Port 2

Port 2 is a general I/O port also functioning as SCI3 I/O pins. Each pin of the port 2 is sh
figure 9.2. The register settings of PMR1and SCI3 have priority for functions of the pins
uses.

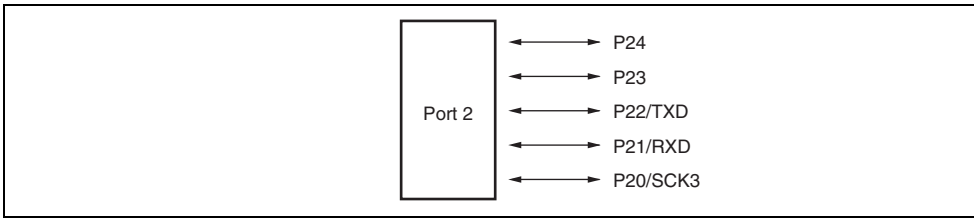


Figure 9.2 Port 2 Pin Configuration

Port 2 has the following registers.

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

1	PCR21	0	W	
0	PCR20	0	W	

9.2.2 Port Data Register 2 (PDR2)

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1.
4	P24	0	R/W	PDR2 stores output data for port 2 pins.
3	P23	0	R/W	If PDR2 is read while PCR2 bits are set to 1, th
2	P22	0	R/W	stored in PDR2 is read. If PDR2 is read while P
1	P21	0	R/W	are cleared to 0, the pin states are read regardl
0	P20	0	R/W	value stored in PDR2.

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

9.2.4 Pin Functions

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

- P24 pin

Register	PCR2	
Bit Name	**PCR24**	**Pin Function**
Setting Value	0	P24 input pin
	1	P24 output pin

- P23 pin

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

- P21/RXD pin

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

[Legend]
X: Don't care.

- P20/SCK3 pin

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

[Legend]
X: Don't care.

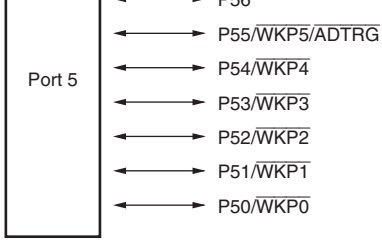


Figure 9.3 Port 5 Pin Configuration

Port 5 has the following registers.

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

				0: General I/O port 1: $\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ input pin
4	WKP4	0	R/W	This bit selects the function of pin P54/$\overline{\text{WKP4}}$. 0: General I/O port 1: $\overline{\text{WKP4}}$ input pin
3	WKP3	0	R/W	This bit selects the function of pin P53/$\overline{\text{WKP3}}$. 0: General I/O port 1: $\overline{\text{WKP3}}$ input pin
2	WKP2	0	R/W	This bit selects the function of pin P52/$\overline{\text{WKP2}}$. 0: General I/O port 1: $\overline{\text{WKP2}}$ input pin
1	WKP1	0	R/W	This bit selects the function of pin P51/$\overline{\text{WKP1}}$. 0: General I/O port 1: $\overline{\text{WKP1}}$ input pin
0	WKP0	0	R/W	This bit selects the function of pin P50/$\overline{\text{WKP0}}$. 0: General I/O port 1: $\overline{\text{WKP0}}$ input pin

3	PCR53	0	W
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 stored in PDR5 are read. If PDR5 is read while P are cleared to 0, the pin states are read regardle value stored in PDR5.
5	P55	0	R/W	
4	P54	0	R/W	
3	P53	0	R/W	
2	P52	0	R/W	
1	P51	0	R/W	
0	P50	0	R/W	

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

9.3.5 Pin Functions

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

- P57 pin

Register	PCR5	
Bit Name	**PCR57**	**Pin Function**
Setting Value	0	P57 input pin
	1	P57 output pin

- P56 pin

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

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

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

[Legend]
X: Don't care.

- 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.

- P51/$\overline{\text{WKP1}}$ pin

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

[Legend]
X: Don't care.

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

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

[Legend]
X: Don't care.

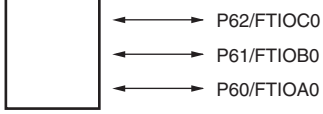


Figure 9.4 Port 6 Pin Configuration

Port 6 has the following registers.

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

9.4.1 Port Control Register 6 (PCR6)

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

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

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

9.4.3 Pin Functions

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

- P67/FTIOD1 pin

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

[Legend]
X: Don't care.

		00				

[Legend]
X: Don't care.

- P65/FTIOB1 pin

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

[Legend]
X: Don't care.

- P63/FTIOD0 pin

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

[Legend]

X: Don't care.

		00				

[Legend]
X: Don't care.

- P61/FTIOB0 pin

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

[Legend]
X: Don't care.

9.5 Port 7

Port 7 is a general I/O port also functioning as a timer V I/O pin and SCI3_2 I/O pin. Ea
the port 7 is shown in figure 9.5. The register settings of the timer V and SCI3_2* have
functions of the pins for both uses.

Note: * The H8/36037 Group does not have the SCI3_2.

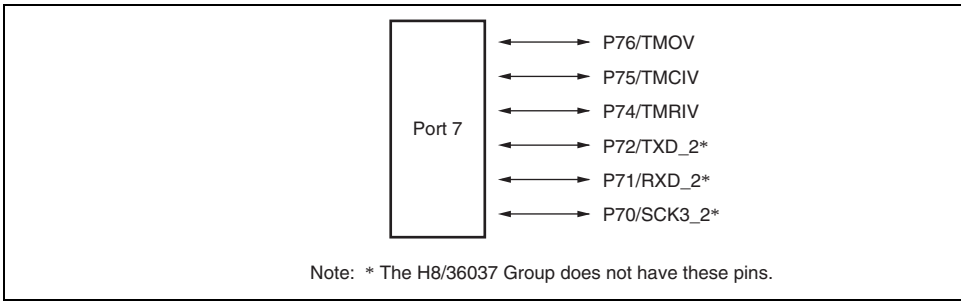


Figure 9.5 Port 7 Pin Configuration

Port 7 has the following registers.

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

3	—	—	—
2	PCR72	0	W
1	PCR71	0	W
0	PCR70	0	W

9.5.2 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	—	Stores output data for port 7 pins.
6	P76	0	R/W	If PDR7 is read while PCR7 bits are set to 1, the stored in PDR7 are read. If PDR7 is read while P are cleared to 0, the pin states are read regardle value stored in PDR7.
5	P75	0	R/W	
4	P74	0	R/W	
3	—	1	—	Bits 7 and 3 are reserved bits. These bits are alw as 1.
2	P72	0	R/W	
1	P71	0	R/W	
0	P70	0	R/W	

		the above values

[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

- P71/RXD_2* pin

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

[Legend]
X: Don't care.
Note: * The H8/36037 Group does not have this pin.

- P70/SCK3_2* pin

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

[Legend]
X: Don't care.
Note: * The H8/36037 Group does not have these pins.

Port 8 has the following registers.

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

9.6.1 Port Control Register 8 (PCR8)

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

Bit	Bit Name	Initial Value	R/W	Description
7	PCR87	0	W	When each of the port 8 pins P87 to P85 functi
6	PCR86	0	W	general I/O port, setting a PCR8 bit to 1 makes corresponding pin an output port, while clearing
5	PCR85	0	W	0 makes the pin an input port.
4 to 0	—	—	—	Reserved

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

9.6.3 Pin Functions

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

- P87 pin

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

- P86 pin

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

- P85 pin

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

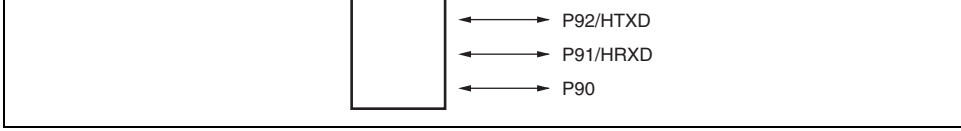


Figure 9.7 Port 9 Pin Configuration

Port 9 has the following registers.

- Port control register 9 (PCR9)
- Port data register 9 (PDR9)

9.7.1 Port Control Register 9 (PCR9)

PCR9 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 9

Bit	Bit Name	Initial Value	R/W	Description
7	PCR97	0	W	When each of the port 9 pins P97 to P90 functi
6	PCR96	0	W	general I/O port, setting a PCR9 bit to 1 makes
5	PCR95	0	W	corresponding pin an output port, while clearing
4	PCR94	0	W	0 makes the pin an input port.
3	PCR93	0	W	
2	PCR92	0	W	
1	PCR91	0	W	
0	PCR90	0	W	

3	P93	0	R/W
2	P92	0	R/W
1	P91	0	R/W
0	P90	0	R/W

9.7.3 Pin Functions

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

- P97/HTXD pin

Register	TCMR	PCR9	
Bit Name	**PMR97**	**PCR97**	**Pin Function**
Setting Value	0	0	P97 input pin
		1	P97 output pin
	1	X	HTXD output pin

[Legend]
X: Don't care.

- P95 pin

Register	PCR9	
Bit Name	**PCR95**	**Pin Function**
Setting Value	0	P95 input pin
	1	P95 output pin

- P94 pin

Register	PCR9	
Bit Name	**PCR94**	**Pin Function**
Setting Value	0	P94 input pin
	1	P94 output pin

- P93/SSI pin

Register	PCR9	
Bit Name	**PCR93**	**Pin Function**
Setting Value	0	P93 input pin
	1	P93 output pin
	X	SSI input/SSI output pin

[Legend]

X: Don't care.

Note: When this pin is used as the SSI pin, register settings of the SSU are required. Fo see section 16.4.4, Communication Modes and Pin Functions.

- P91/SSCK pin

Register	SSCRH	PCR9	
Bit Name	**SCKS**	**PCR91**	**Pin Function**
Setting Value	0	0	P91 input pin
		1	P91 output pin
	1	X	SSCK input/SSCK output pin

[Legend]

X: Don't care.

Note: When this pin is used as the SSCK pin, register settings of the SSU are required. F
details, see section 16.4.4, Communication Modes and Pin Functions.

- P90/$\overline{\text{SCS}}$ pin

Register	SSCRL	SSCRH		PCR9	
Bit Name	**SSUMS**	**CSS1**	**CSS0**	**PCR90**	**Pin Function**
Setting Value	0	X	X	0	P90 input pin
		X	X	1	P90 output pin
	1	0	0	0	P90 input pin
				1	P90 output pin
		0	1	X	$\overline{\text{SCS}}$ input pin
		1	X		$\overline{\text{SCS}}$ output pin

[Legend]

X: Don't care.

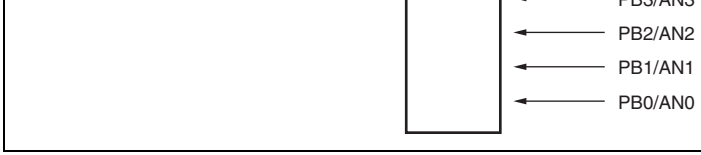


Figure 9.8 Port B Pin Configuration

Port B has the following register.

- Port data register B (PDRB)

9.8.1 Port Data Register B (PDRB)

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

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

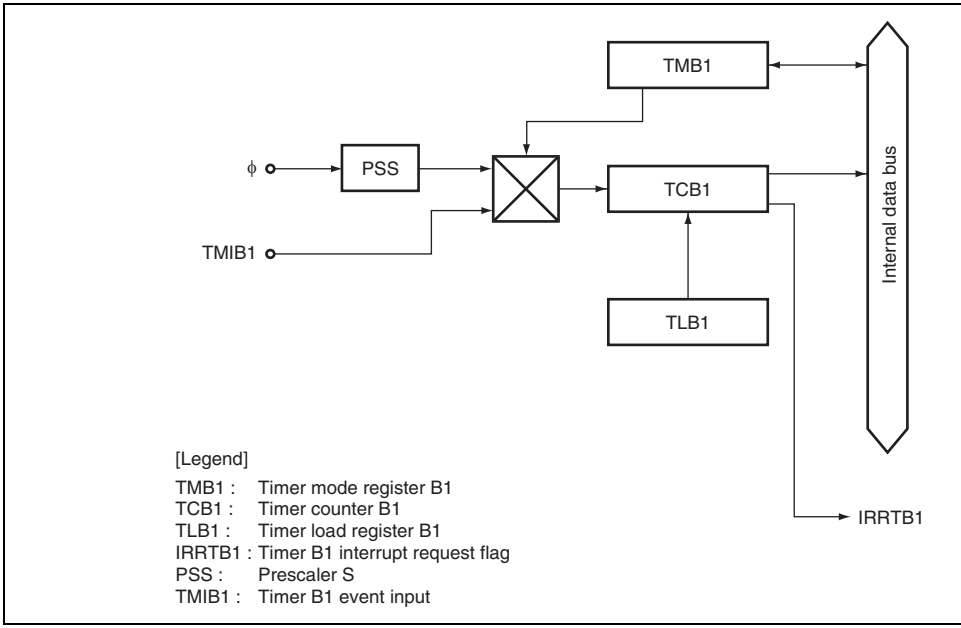


Figure 10.1 Block Diagram of Timer B1

TMB1 selects the auto-reload function and input clock.

Bit	Bit Name	Initial Value	R/W	Description
7	TMB17	0	R/W	Auto-reload function select 0: Interval timer function selected 1: Auto-reload function selected
6 to 3	—	All 1	—	Reserved These bits are always read as 1.
2	TMB12	0	R/W	Clock select
1	TMB11	0	R/W	000: Internal clock: φ/8192
0	TMB10	0	R/W	001: Internal clock: φ/2048 010: Internal clock: φ/512 011: Internal clock: φ/256 100: Internal clock: φ/64 101: Internal clock: φ/16 110: Internal clock: φ/4 111: External event (TMIB1): rising or falling ed Note: * The edge of the external event signal by bit IEG1 in the interrupt edge selec (IEGR1). See section 3.2.1, Interrupt E Select Register 1 (IEGR1), for details. setting TMB12 to TMB10 to 1, IRQ1 in mode register 1 (PMR1) should be set

TLB1 is an 8-bit write-only register for setting the reload value of TCB1. When a reload
set in TLB1, the same value is loaded into TCB1 as well, and TCB1 starts counting up fr
value. When TCB1 overflows during operation in auto-reload mode, the TLB1 value is lo
into TCB1. Accordingly, overflow periods can be set within the range of 1 to 256 input c
TLB1 is allocated to the same address as TCB1. TLB1 is initialized to H'00.

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

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

10.4.2 Auto-Reload Timer Operation

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

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

10.4.3 Event Counter Operation

Timer B1 can operate as an event counter in which TMIB1 is set to an event input pin. E event counting is selected by setting bits TMB12 to TMB10 in TMB1 to 1. TCB1 count rising or falling edge of an external event signal input at pin TMB1.

When timer B1 is used to count external event input, bit IRQ1 in PMR1 should be set to IEN1 in IENR1 should be cleared to 0 to disable IRQ1 interrupt requests.

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

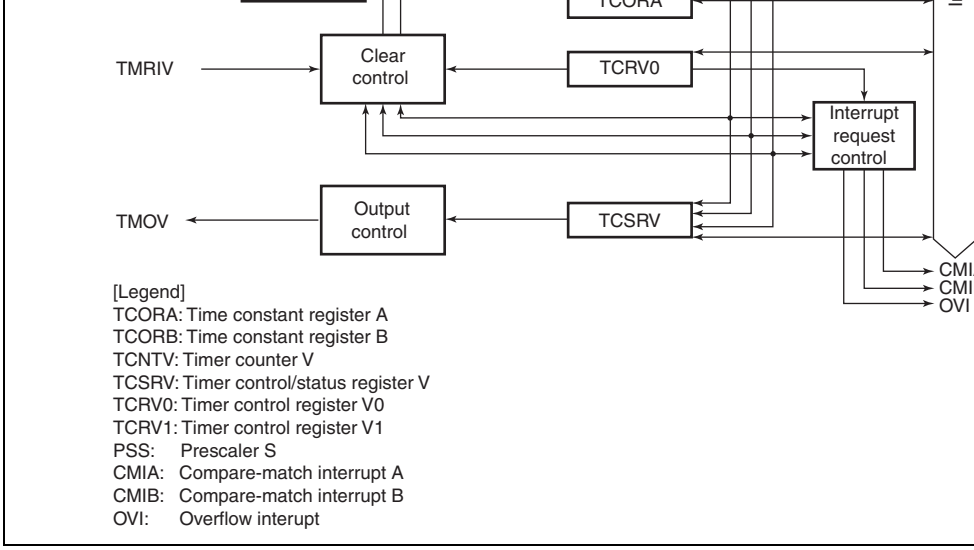


Figure 11.1 Block Diagram of Timer V

11.3 Register Descriptions

The 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)

11.3.1 Timer Counter V (TCNTV)

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

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

TCNTV is initialized to H'00.

and the settings of bits OS3 to OS0 in TCSRV.

TCORA and TCORB are initialized to H'FF.

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

Bit 2	Bit 1	Bit 0	Bit 0	
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

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

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

11.3.5 Timer Control Register V1 (TCRV1)

TCRV1 selects the edge at the TRGV pin, enables TRGV input, and selects the clock in
TCNTV.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1.
4	TVEG1	0	R/W	TRGV Input Edge Select
3	TVEG0	0	R/W	These bits select the TRGV input edge. 00: TRGV trigger input is prohibited 01: Rising edge is selected 10: Falling edge is selected 11: Rising and falling edges are both selected
2	TRGE	0	R/W	TCNT starts counting up by the input of the edg selected by TVEG1 and TVEG0. 0: Disables starting counting-up TCNTV by the the TRGV pin and halting counting-up TCNT TCNTV is cleared by a compare match. 1: Enables starting counting-up TCNTV by the i the TRGV pin and halting counting-up TCNT TCNTV is cleared by a compare match.

11.4 Operation

11.4.1 Timer V Operation

1. According to table 11.2, six internal/external clock signals output by prescaler S can b selected as the timer V operating clock signals. When the operating clock signal is sel TCNTV starts counting-up. Figure 11.2 shows the count timing with an internal clock selected, and figure 11.3 shows the count timing with both edges of an external clock selected.
2. When TCNTV overflows (changes from H'FF to H'00), the overflow flag (OVF) in T will be set. The timing at this time is shown in figure 11.4. An interrupt request is sen CPU when OVIE in TCRV0 is 1.
3. TCNTV is constantly compared with TCORA and TCORB. Compare match flag A o (CMFA or CMFB) is set to 1 when TCNTV matches TCORA or TCORB, respectivel compare-match signal is generated in the last state in which the values match. Figure shows the timing. An interrupt request is generated for the CPU when CMIEA or CM TCRV0 is 1.
4. When a compare match A or B is generated, the TMOV responds with the output valu selected by bits OS3 to OS0 in TCSRV. Figure 11.6 shows the timing when the outpu toggled by compare match A.
5. When CCLR1 or CCLR0 in TCRV0 is 01 or 10, TCNTV can be cleared by the corres compare match. Figure 11.7 shows the timing.
6. When CCLR1 or CCLR0 in TCRV0 is 11, TCNTV can be cleared by the rising edge input of TMRIV pin. A TMRIV input pulse-width of at least 1.5 system clocks is nec Figure 11.8 shows the timing.
7. When a counter-clearing source is generated with TRGE in TCRV1 set to 1, the coun halted as soon as TCNTV is cleared. TCNTV resumes counting-up when the edge sel TVEG1 or TVEG0 in TCRV1 is input from the TGRV pin.

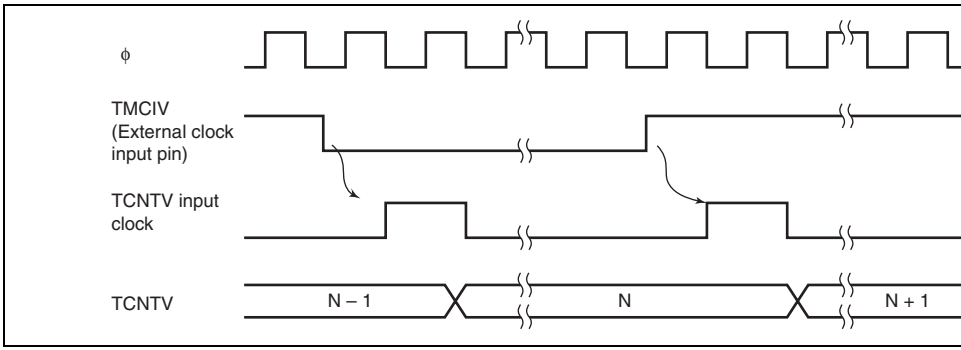


Figure 11.3 Increment Timing with External Clock

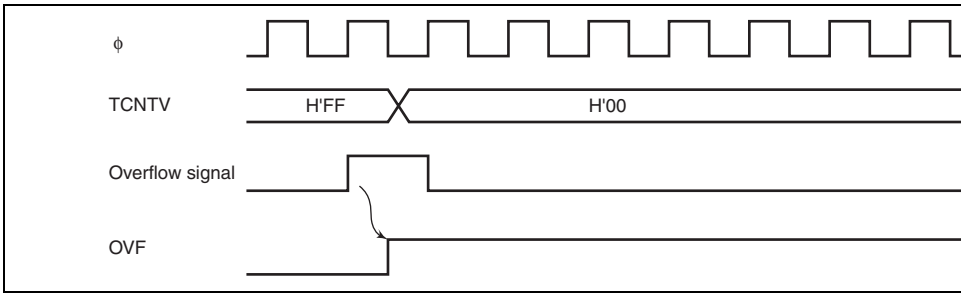


Figure 11.4 OVF Set Timing

Figure 11.5 CMFA and CMFB Set Timing

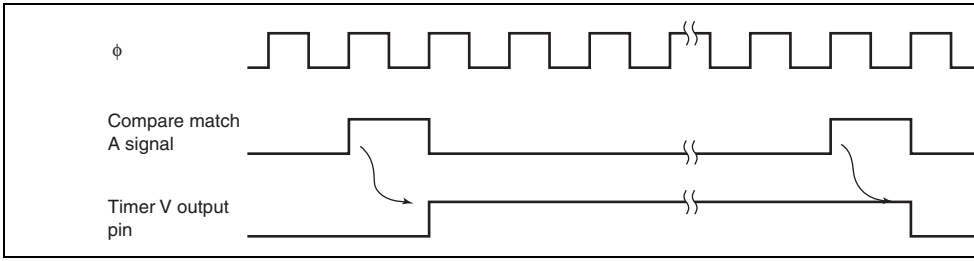


Figure 11.6 TMOV Output Timing

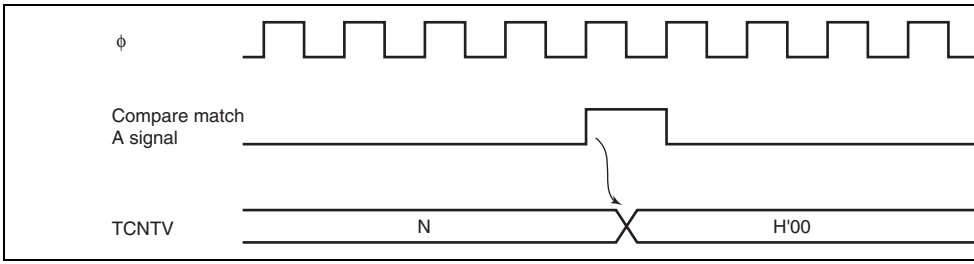


Figure 11.7 Clear Timing by Compare Match

11.5 Timer V Application Examples

11.5.1 Pulse Output with Arbitrary Duty Cycle

Figure 11.9 shows an example of output of pulses with an arbitrary duty cycle.

1. Set bits CCLR1 and CCLR0 in TCRV0 so that TCNTV will be cleared by compare m
   TCORA.
2. Set bits OS3 to OS0 in TCSRV so that the output will go to 1 at compare match with
   and to 0 at compare match with TCORB.
3. Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clo
4. With these settings, a waveform is output without further software intervention, with
   determined by TCORA and a pulse width determined by TCORB.

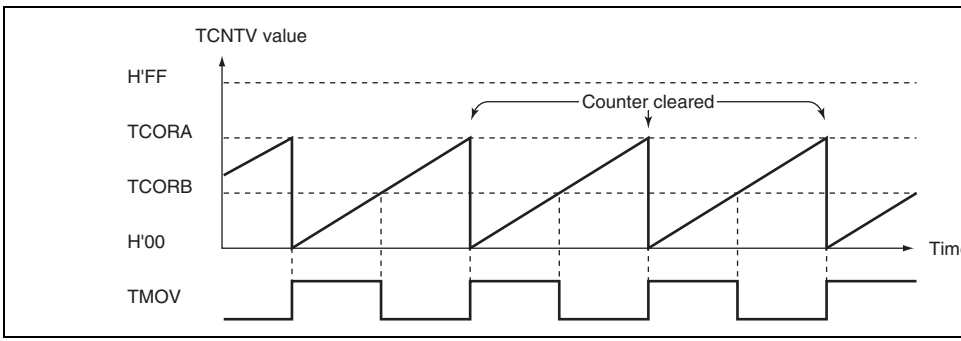


Figure 11.9 Pulse Output Example

input.

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

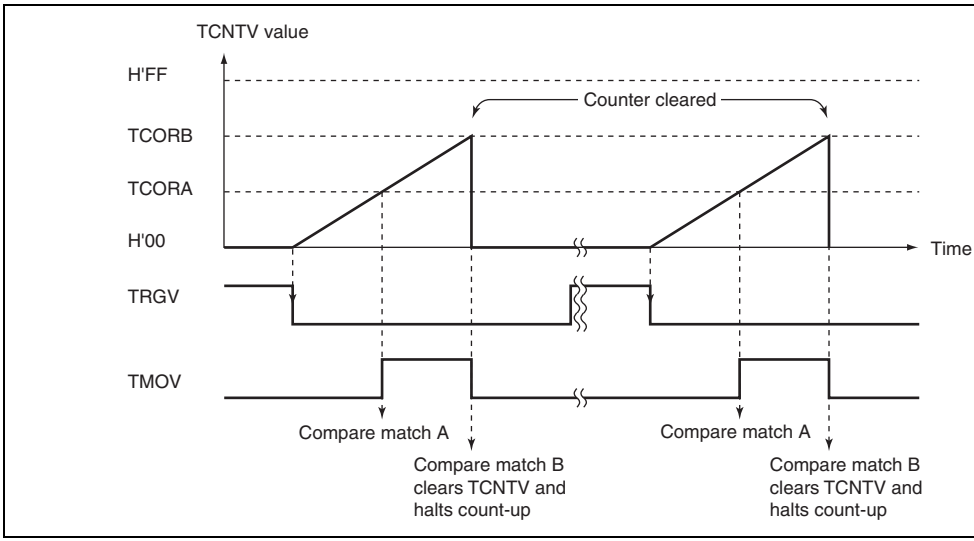


Figure 11.10 Example of Pulse Output Synchronized to TRGV Input

11.12 shows the timing.

3. If compare matches A and B occur simultaneously, any conflict between the output s for compare match A and compare match B is resolved by the following priority: tog > output 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 a divided system clock (ϕ). Therefore, in figure 11.3 the switch is from a high clock signal to a low clock signal, the switch seen as a falling edge, causing TCNTV to increment. TCNTV can also be increment switch between internal and external clocks.

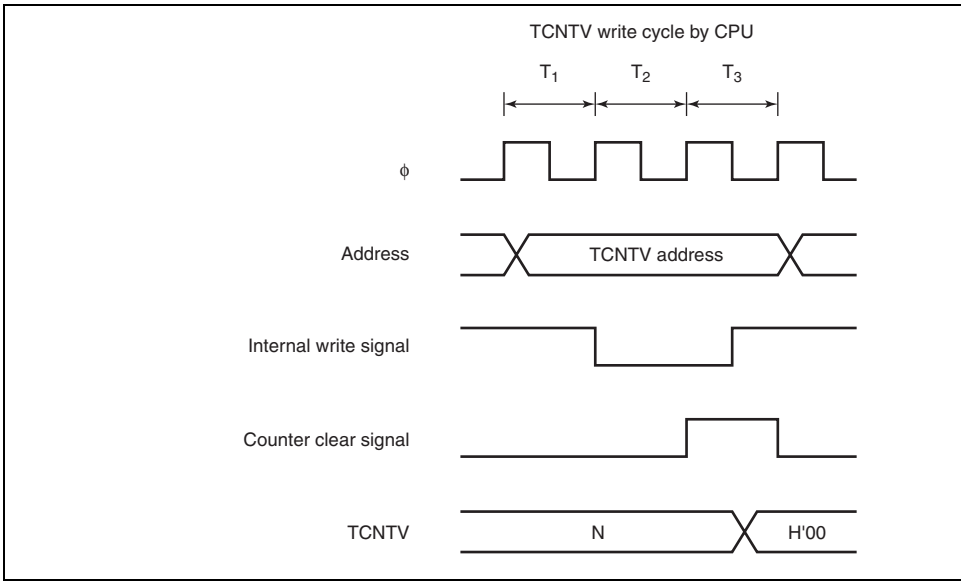


Figure 11.11 Contention between TCNTV Write and Clear

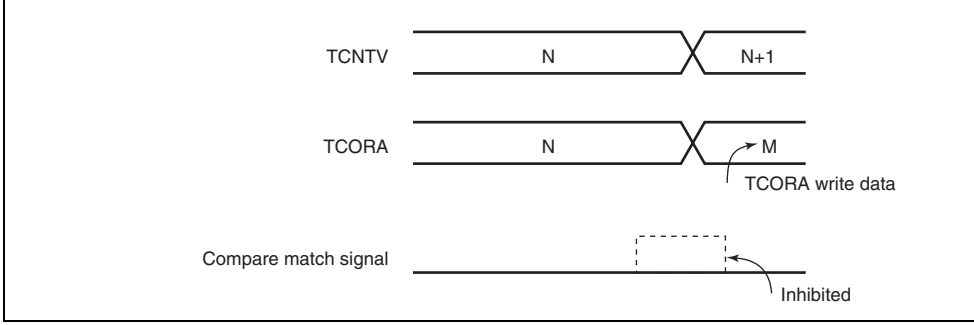


Figure 11.12 Contention between TCORA Write and Compare Match

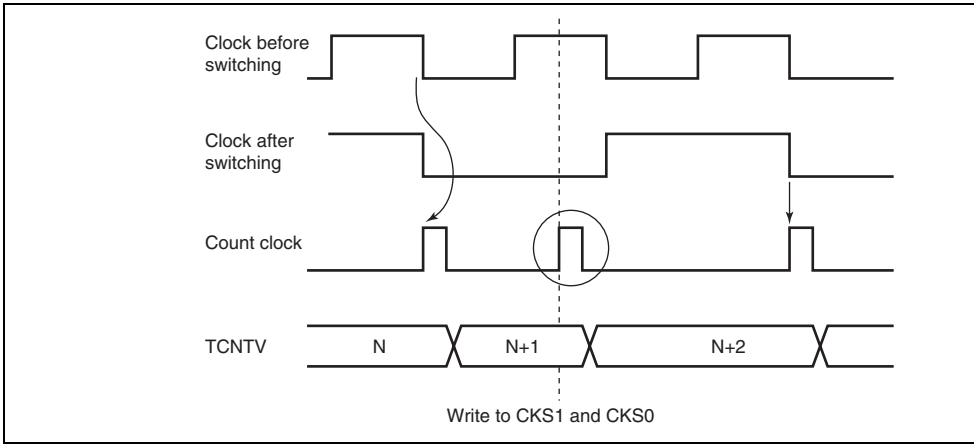


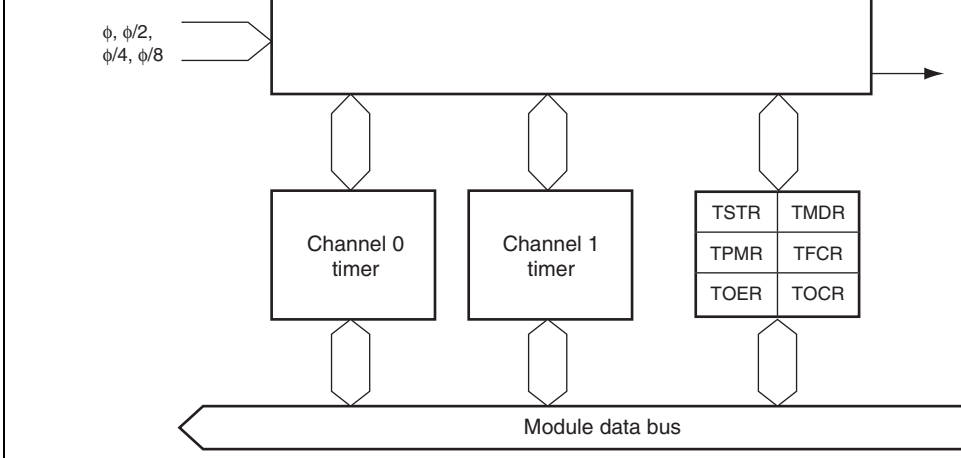
Figure 11.13 Internal Clock Switching and TCNTV Operation

— Independently assignable output compare or input capture functions

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

capture registers)

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



[Legend]

TSTR : Timer start register (8 bits)

TMDR : Timer mode register (8 bits)

TPMR : Timer PWM mode register (8 bits)

TFCR : Timer function control register (8 bits)

TOER : Timer output master enable register (8 bits)

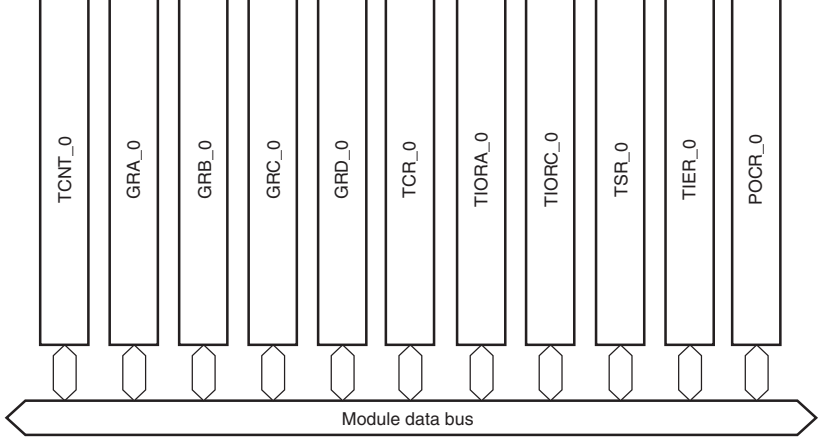
TOCR : Timer output control register (8 bits)

$\overline{\text{ADTRG}}$: A/D conversion start trigger output signal

ITMZ0 : Channel 0 interrupt

ITMZ1 : Channel 1 interrupt

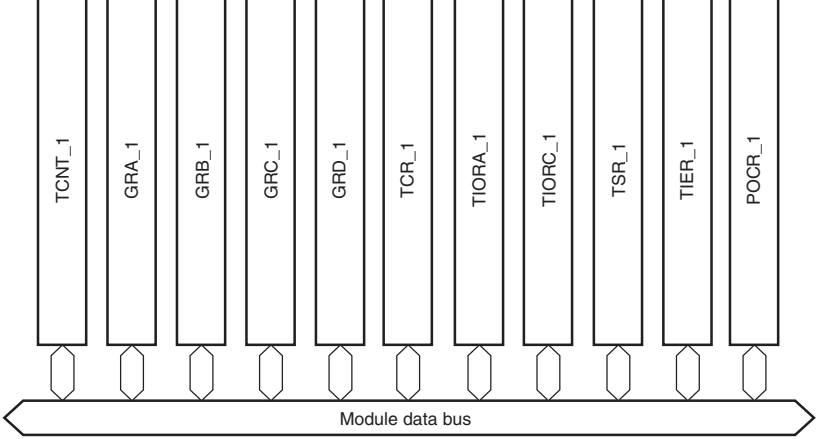
Figure 12.1 Timer Z Block Diagram



[Legend]

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

Figure 12.2 Timer Z (Channel 0) Block Diagram



[Legend]

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

Figure 12.3 Timer Z (Channel 1) Block Diagram

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

- Timer output control register (TOCR)

Channel 0

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

Channel 1

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

1	STR1	0	R/W	Channel 1 Counter Start 0: TCNT_1 halts counting 1: TCNT_1 starts counting
0	STR0	0	R/W	Channel 0 Counter Start 0: TCNT_0 halts counting 1: TCNT_0 starts counting

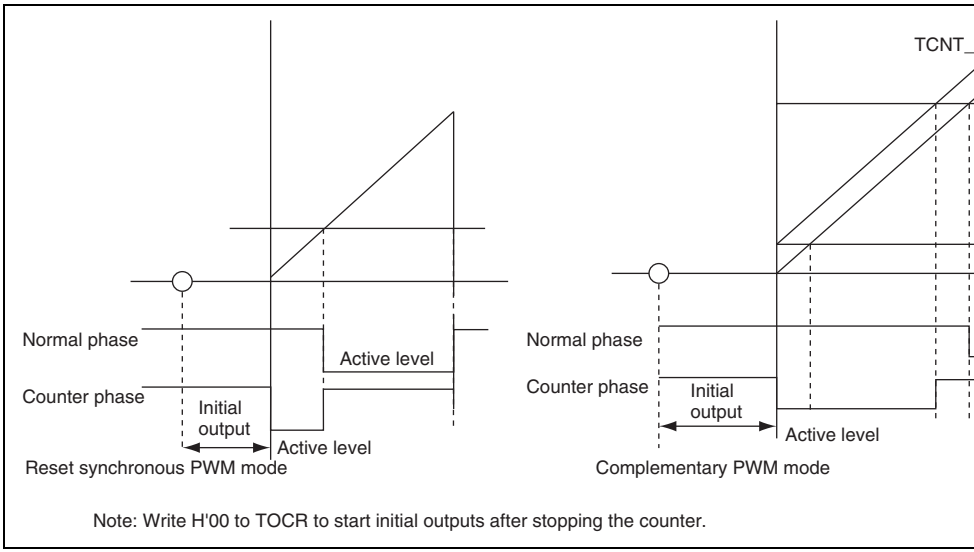
6	BFC1	0	R/W	Buffer Operation C1 0: GRC_1 operates normally 1: GRA_1 and GRD_1 are used together for bu operation
5	BFD0	0	R/W	Buffer Operation D0 0: GRD_0 operates normally 1: GRB_0 and GRD_0 are used together for bu operation
4	BFC0	0	R/W	Buffer Operation C0 0: GRC_0 operates normally 1: GRA_0 and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

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

				1: External clock input is enabled
5	ADEG	0	R/W	A/D Trigger Edge Select A/D module should be set to start an A/D convers external trigger 0: A/D trigger at the crest in complementary PWM 1: A/D trigger at the trough in complementary PW
4	ADTRG	0	R/W	External Trigger Disable 0: A/D trigger for PWM cycles is disabled in comp PWM mode 1: A/D trigger for PWM cycles is enabled in comp PWM mode
3	OLS1	0	R/W	Output Level Select 1 Selects the counter-phase output levels in reset synchronous PWM mode or complementary PWM 0: Initial output is high and the active level is low. 1: Initial output is low and the active level is high.
2	OLS0	0	R/W	Output Level Select 0 Selects the normal-phase output levels in reset synchronous PWM mode or complementary PWM 0: Initial output is high and the active level is low. 1: Initial output is low and the active level is high. Figure 12.4 shows an example of outputs in rese synchronous PWM mode and complementary PW when OLS1 = 0 and OLS0 = 0.

the crest)

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



Note: Write H'00 to TOCR to start initial outputs after stopping the counter.

Figure 12.4 Example of Outputs in Reset Synchronous PWM Mode and Complementary PWM Mode

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

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

12.3.6 Timer Output Control Register (TOCR)

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

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

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

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

12.3.7 Timer Counter (TCNT)

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

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

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

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

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

Notes: 1. When GR functions as an output compare register, TCNT is cleared by comp
When GR functions as input capture, TCNT is cleared by input capture.
2. Synchronous operation is set by TMDR.
3. X: Don't care

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

				101: Input capture to GRA at the falling edge
				11X: Input capture to GRA at both rising and falli

[Legend]
X: Don't care

TIORC: TIORC selects whether GRC or GRD is used as an output compare register or capture register. When an output compare register is selected, the output setting is select an input capture register is selected, an input edge of an input capture signal is selected. also selects the function of FTIOC or FTIOD pin.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1.
6	IOD2	0	R/W	I/O Control D2 to D0
5	IOD1	0	R/W	GRD is an output compare register:
4	IOD0	0	R/W	000: Disables pin output by compare match 001: 0 output by GRD compare match 010: 1 output by GRD compare match 011: Toggle output by GRD compare match GRD is an input capture register: 100: Input capture to GRD at the rising edge 101: Input capture to GRD at the falling edge 11X: Input capture to GRD at both rising and falli
3	—	1	—	Reserved This bit is always read as 1.

				101: Input capture to GRC at the falling edge 11X: Input capture to GRC at both rising and fallin

[Legend]
X: Don't care

12.3.11 Timer Status Register (TSR)

TSR indicates generation of an overflow/underflow of TCNT and a compare match/input of GRA, GRB, GRC, and GRD. These flags are interrupt sources. If an interrupt is enable corresponding bit in TIER, TSR requests an interrupt for the CPU. Timer Z has two TSR one for each channel.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1.
5	UDF*	0	R/W	Underflow Flag [Setting condition] • When TCNT_1 underflows [Clearing condition] • When 0 is written to UDF after reading UDF =
4	OVF	0	R/W	Overflow Flag [Setting condition] • When the TCNT value underflows [Clearing condition] • When 0 is written to OVF after reading OVF =

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

- When 0 is written to IMFA after reading IMFA

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

12.3.12 Timer Interrupt Enable Register (TIER)

TIER enables or disables interrupt requests for overflow or GR compare match/input capt
Timer Z has two TIER registers, one for each channel.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1.
4	OVIE	0	R/W	Overflow Interrupt Enable 0: Interrupt requests (OVI) by OVF or UDF flag a disabled 1: Interrupt requests (OVI) by OVF or UDF flag a enabled
3	IMIED	0	R/W	Input Capture/Compare Match Interrupt Enable D 0: Interrupt requests (IMID) by IMFD flag are disa 1: Interrupt requests (IMID) by IMFD flag are ena
2	IMIEC	0	R/W	Input Capture/Compare Match Interrupt Enable C 0: Interrupt requests (IMIC) by IMFC flag are disa 1: Interrupt requests (IMIC) by IMFC flag are ena
1	IMIEB	0	R/W	Input Capture/Compare Match Interrupt Enable B 0: Interrupt requests (IMIB) by IMFB flag are disa 1: Interrupt requests (IMIB) by IMFB flag are ena

Bit	Bit Name	Initial value	R/W	Description
7 to 3	—	All 1	—	Reserved These bits are always read as 1.
2	POLD	0	R/W	PWM Mode Output Level Control D 0: The output level of FTIOD is low-active 1: The output level of FTIOD is high-active
1	POLC	0	R/W	PWM Mode Output Level Control C 0: The output level of FTIOC is low-active 1: The output level of FTIOC is high-active
0	POLB	0	R/W	PWM Mode Output Level Control B 0: The output level of FTIOB is low-active 1: The output level of FTIOB is high-active

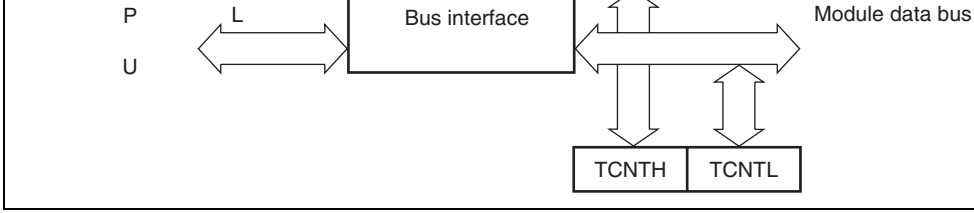


Figure 12.5 Accessing Operation of 16-Bit Register (between CPU and TCNT (16

8-Bit Register: Registers other than TCNT and GR are 8-bit registers that are connected with the CPU in an 8-bit width. Figure 12.6 shows an example of accessing the 8-bit regis

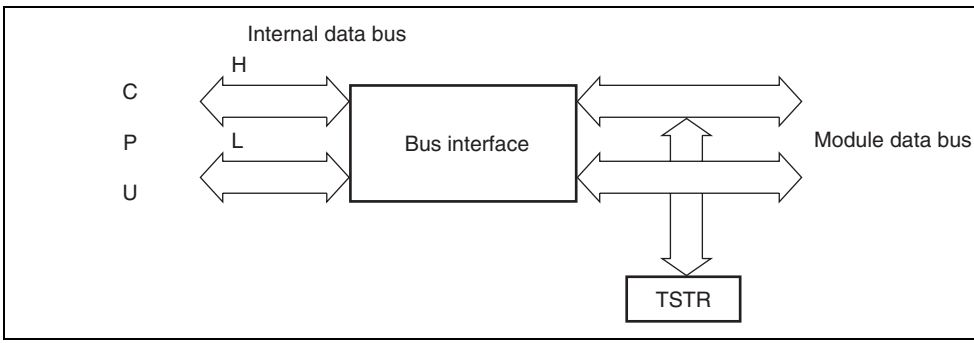
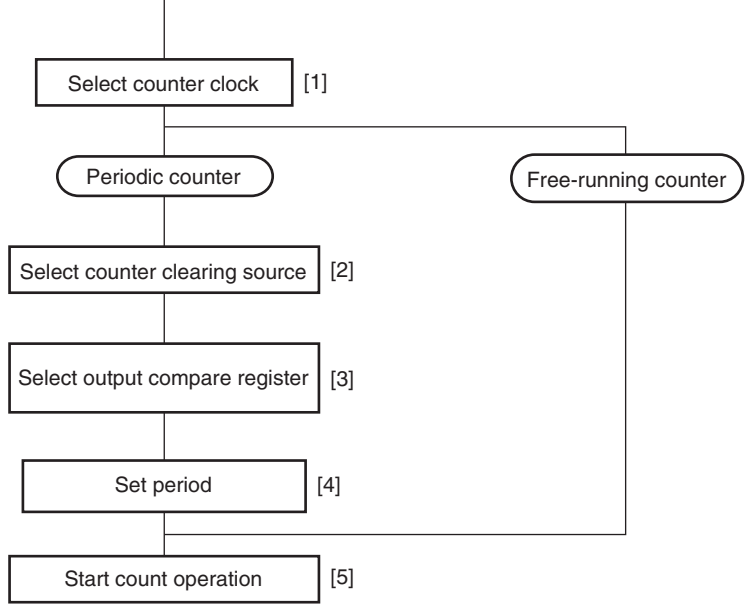


Figure 12.6 Accessing Operation of 8-Bit Register (between CPU and TSTR (8



TPSC2 to TPS
TCR. When a
clock is select
the external cl
with bits CKE
and CKEG0 in
[2] For periodic c
operation, sel
TCNT clearing
source with bi
CCLR2 to CC
TCR.
[3] Designate the
register select
as an output c
register by me
TIOR.
[4] Set the period
cycle in the ge
register select
in [2].
[5] Set the STR b
to 1 to start th
operation.

Figure 12.7 Example of Counter Operation Setting Procedure

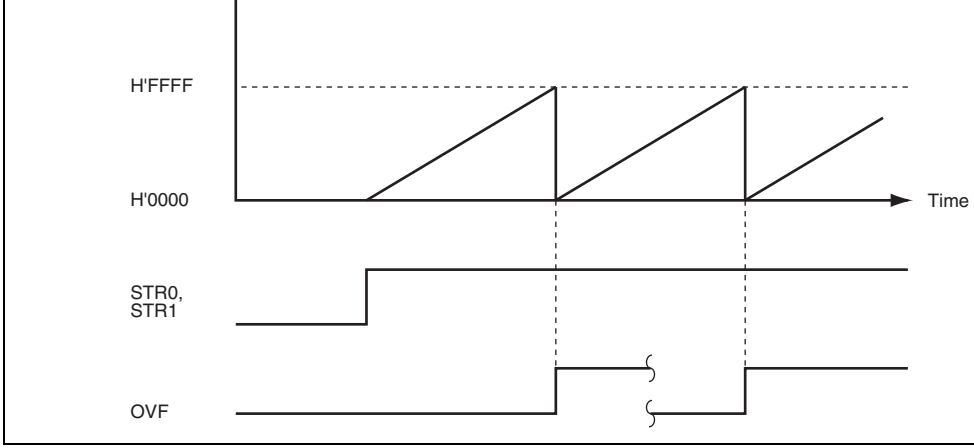


Figure 12.8 Free-Running Counter Operation

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

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

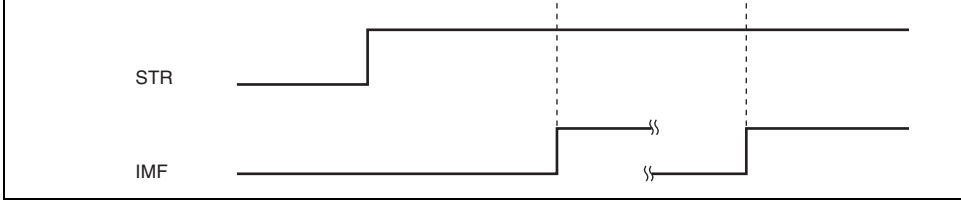


Figure 12.9 Periodic Counter Operation

TCNT Count Timing:

- Internal clock operation

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

  Figure 12.10 illustrates this timing.

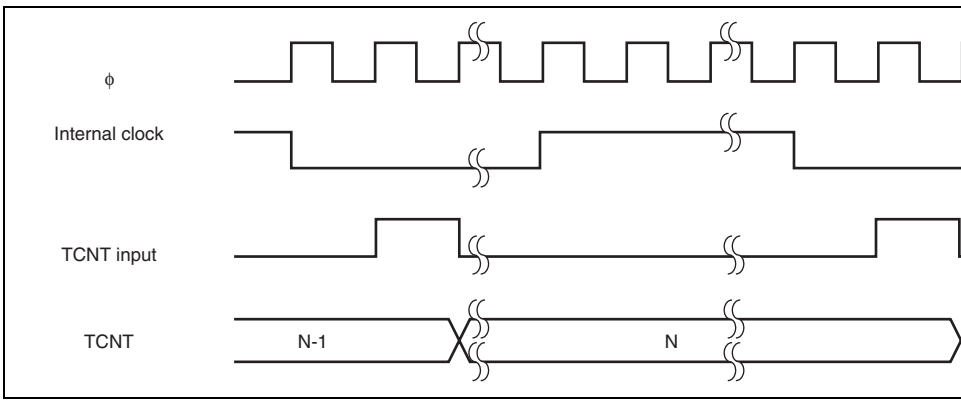


Figure 12.10 Count Timing at Internal Clock Operation

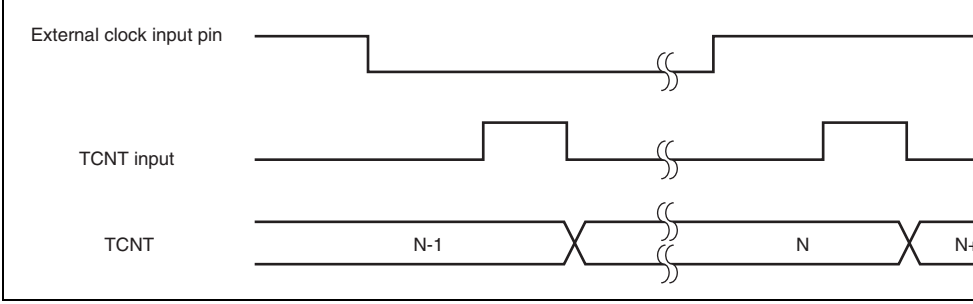
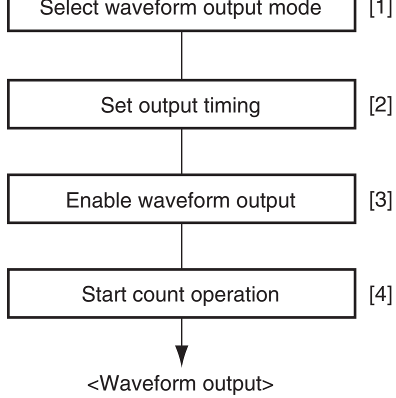


Figure 12.11 Count Timing at External Clock Operation (Both Edges Detect



match occurs.

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

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

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

Figure 12.12 Example of Setting Procedure for Waveform Output by Compare

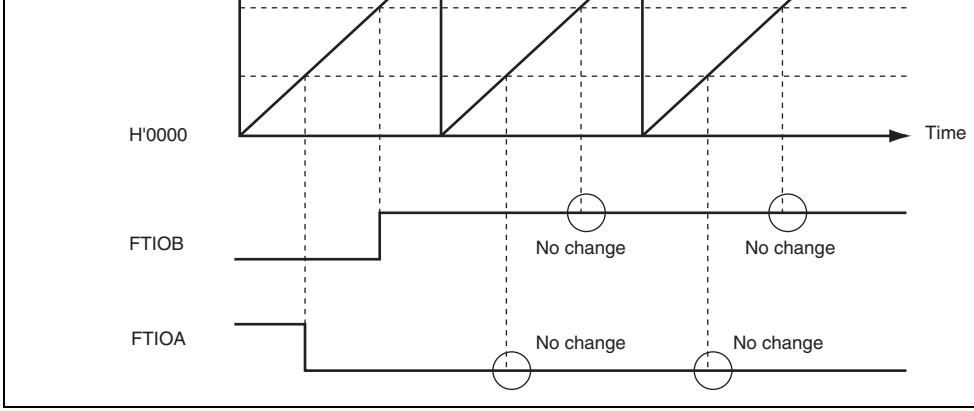


Figure 12.13 Example of 0 Output/1 Output Operation

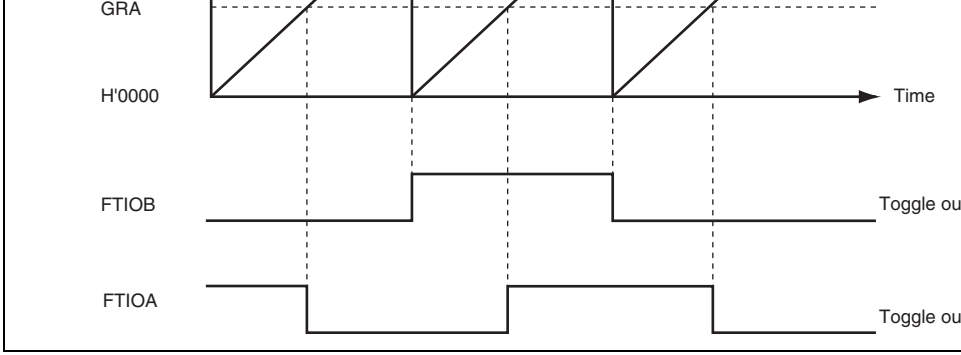


Figure 12.14 Example of Toggle Output Operation

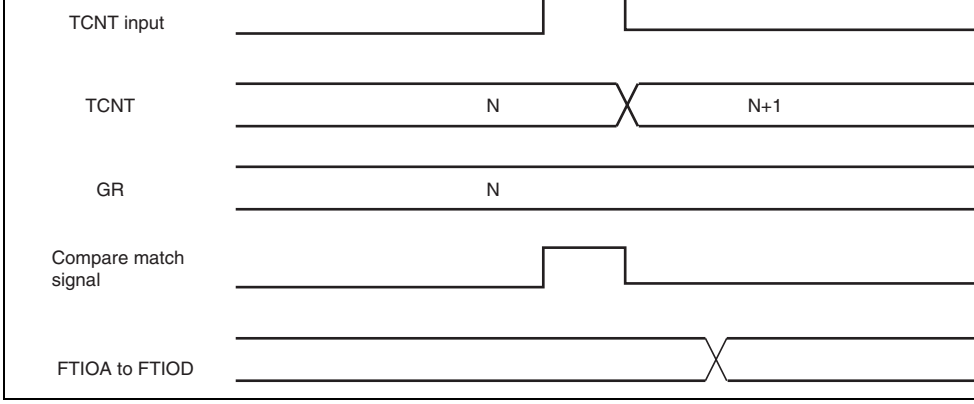
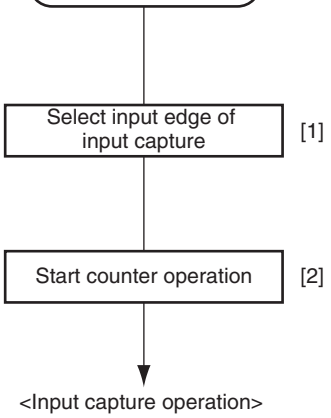


Figure 12.15 Output Compare Timing



register by means of TIOR, and select rising edge, falling edge, or both edges as the input edge of the input capture signal.

[2] Set the STR bit in TSTR to 1 to start the TCNT counter operation.

Figure 12.16 Example of Input Capture Operation Setting Procedure

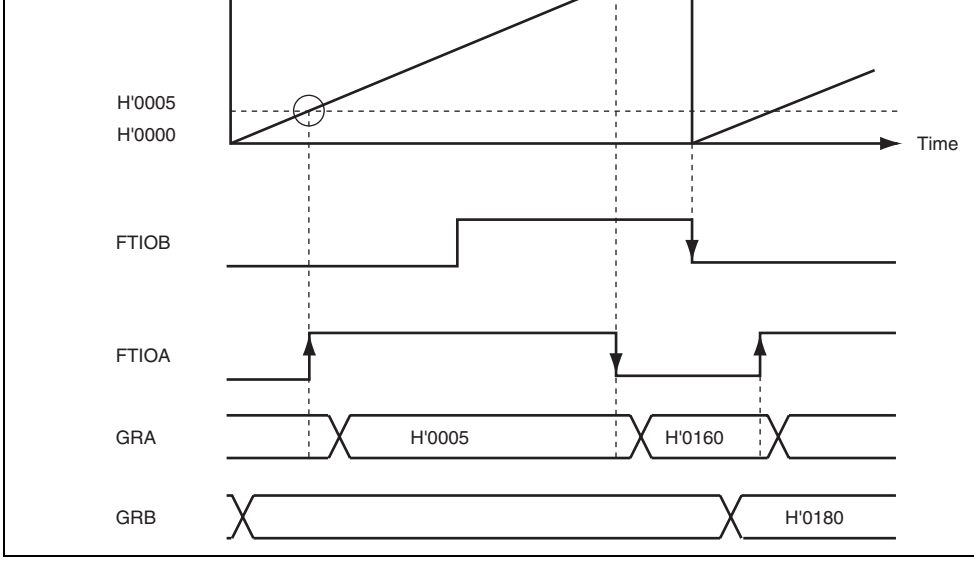


Figure 12.17 Example of Input Capture Operation

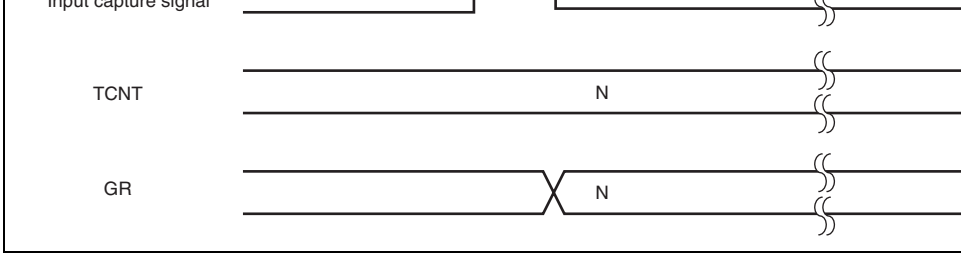
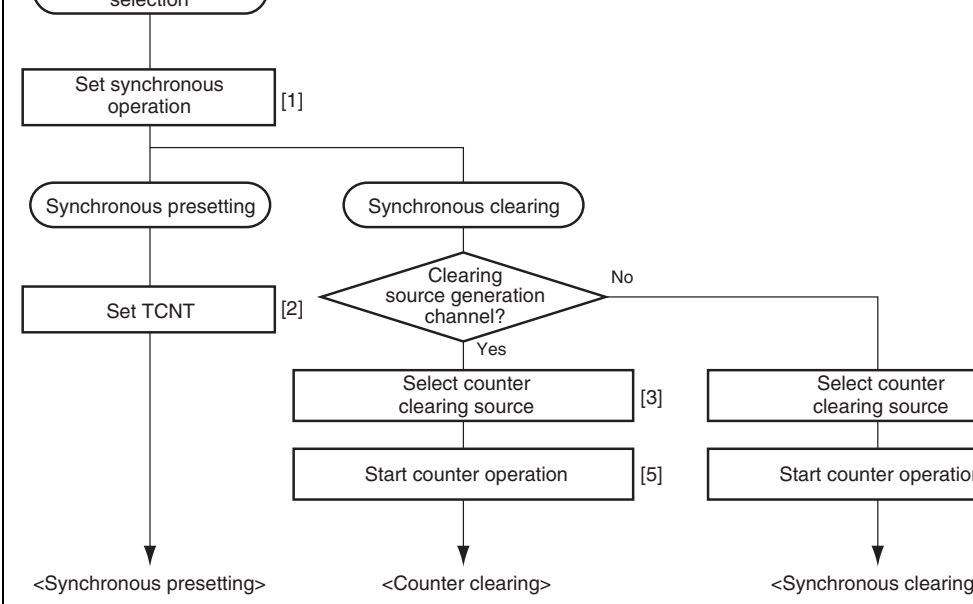


Figure 12.18 Input Capture Signal Timing



[1] Set the SYNC bits in TMDR to 1.
[2] When a value is written to either of the TCNT counters, the same value is simultaneously written to th other TCNT counter.
[3] Set bits CCLR1 and CCLR0 in TCR to specify counter clearing by compare match/input capture.
[4] Set bits CCLR1 and CCLR0 in TCR to designate synchronous clearing for the counter clearing source
[5] Set the STR bit in TSTR to 1 to start the count operation.

Figure 12.19 Example of Synchronous Operation Setting Procedure

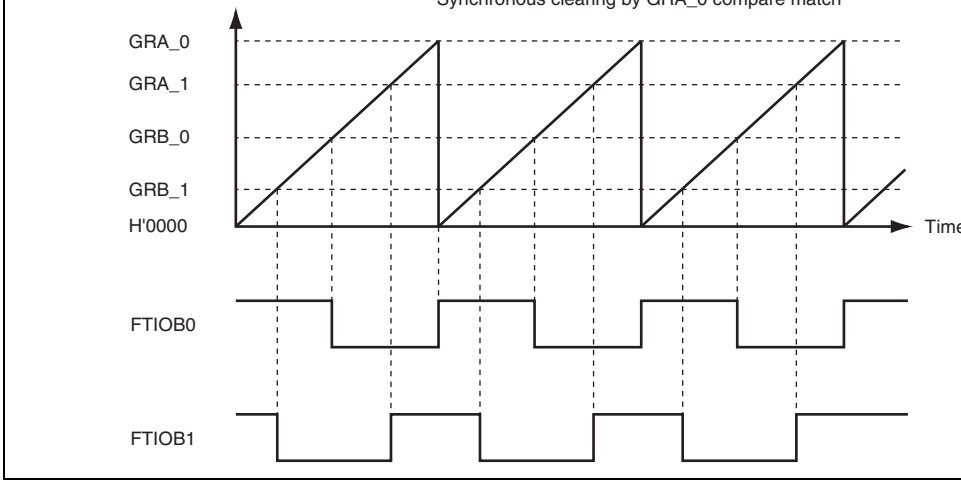
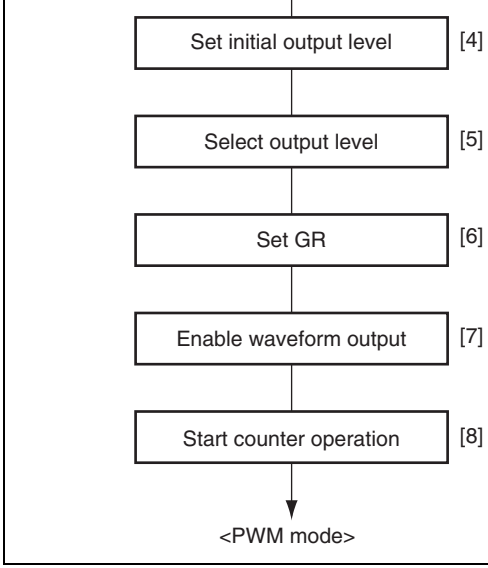


Figure 12.20 Example of Synchronous Operation

Figure 12.21 shows an example of the PWM mode setting procedure.

Table 12.3 Initial Output Level of FTIOB0 Pin

TOB0	POLB	Initial Output Level
0	0	1
0	1	0
1	0	0
1	1	1



TOB0 to TOD0 and TOB1 to TOD1 in TOCR.

[5] Set the output level with bits POLB to POLD in POCR.

[6] Set the cycle in GRA, and set the duty in the other GR.

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

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

Figure 12.21 Example of PWM Mode Setting Procedure

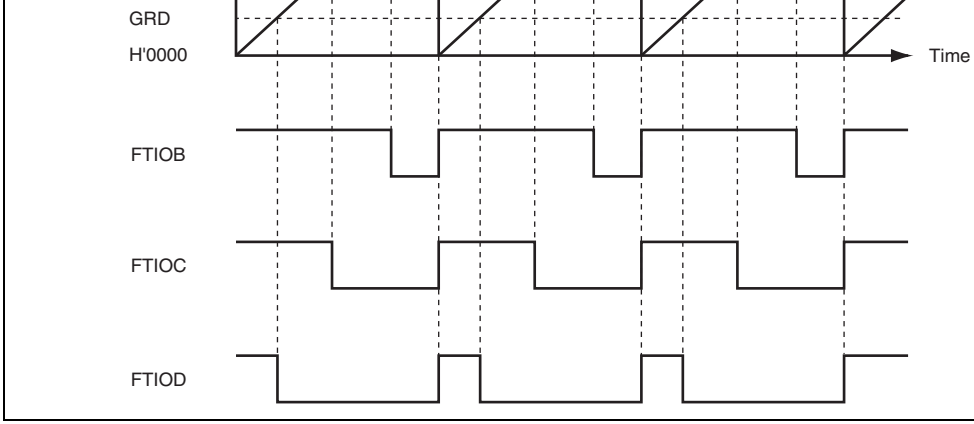


Figure 12.22 Example of PWM Mode Operation (1)

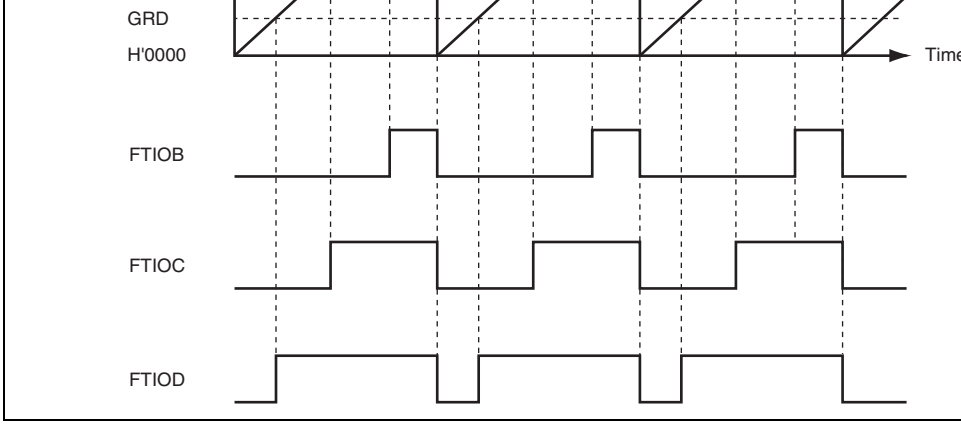


Figure 12.23 Example of PWM Mode Operation (2)

Figures 12.24 (when TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 0) and 12.2
TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 1) show examples of the output
waveforms with duty cycles of 0% and 100% in PWM mode.

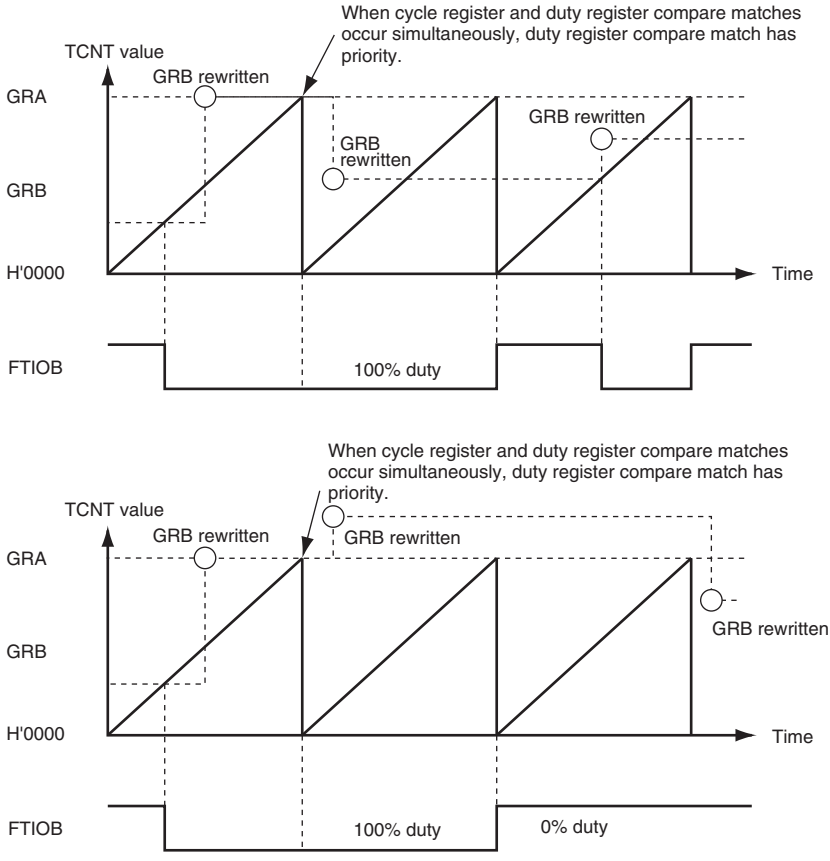


Figure 12.24 Example of PWM Mode Operation (3)

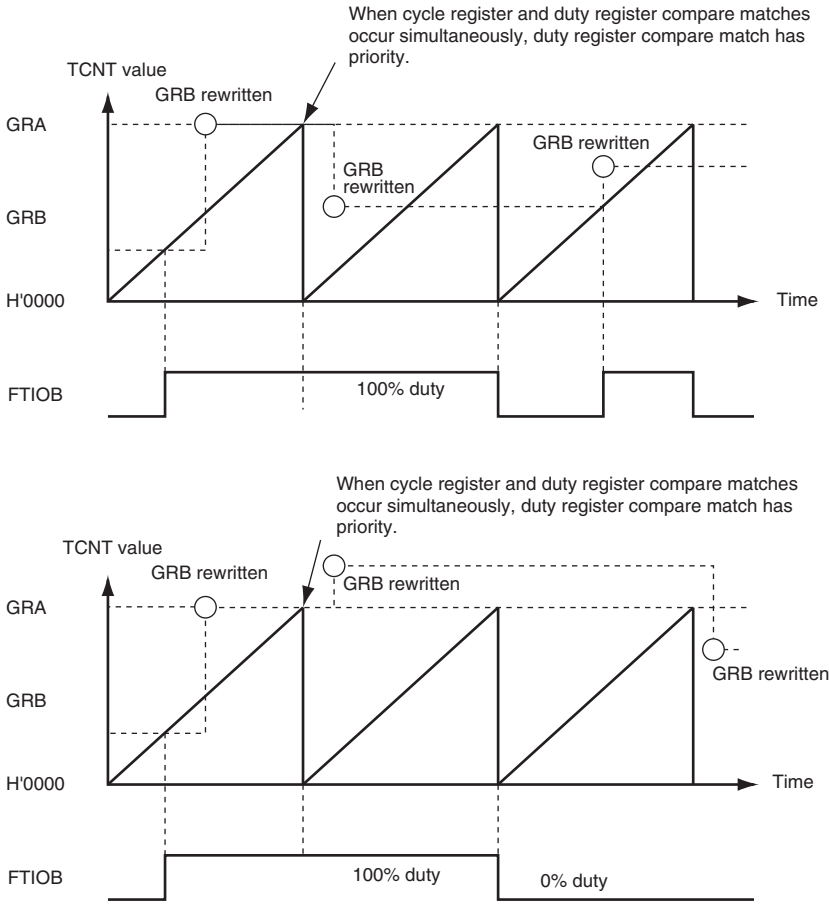
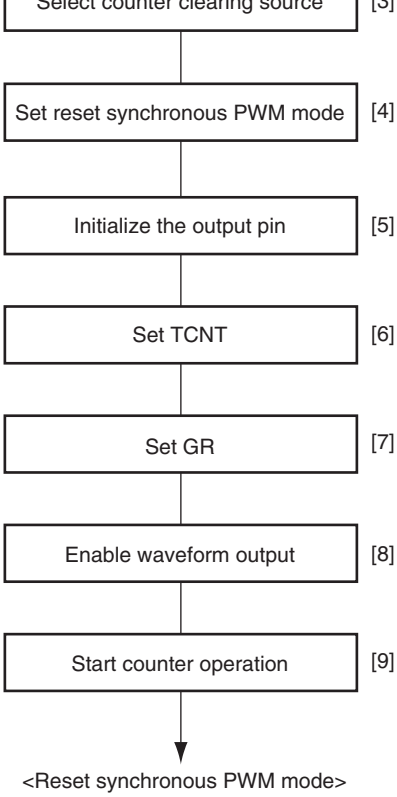


Figure 12.25 Example of PWM Mode Operation (4)

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

Table 12.5 Register Settings in Reset Synchronous PWM Mode

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



[4] Select the reset synchronous PWM m with bits CMD1 and CMD0 in TFCR. FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 become PWM output pins automatically.
[5] Set H'00 to TOCR.
[6] Set TCNT_0 as H'0000. TCNT1 does need to be set.
[7] GRA_0 is a cycle register. Set a cycl GRA_0. Set the changing point timin the PWM output waveform for GRB_0 GRA_1, and GRB_1.
[8] Enable or disable the timer output by TOER.
[9] Set the STR bit in TSTR to 1 and star counter operation.

Figure 12.26 Example of Reset Synchronous PWM Mode Setting Procedu

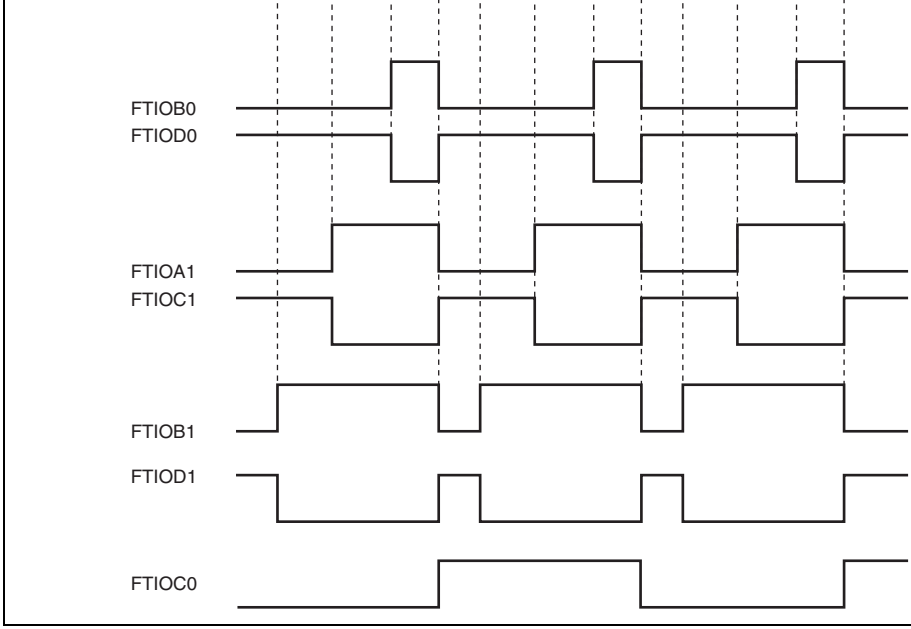


Figure 12.27 Example of Reset Synchronous PWM Mode Operation (OLS0 = OL

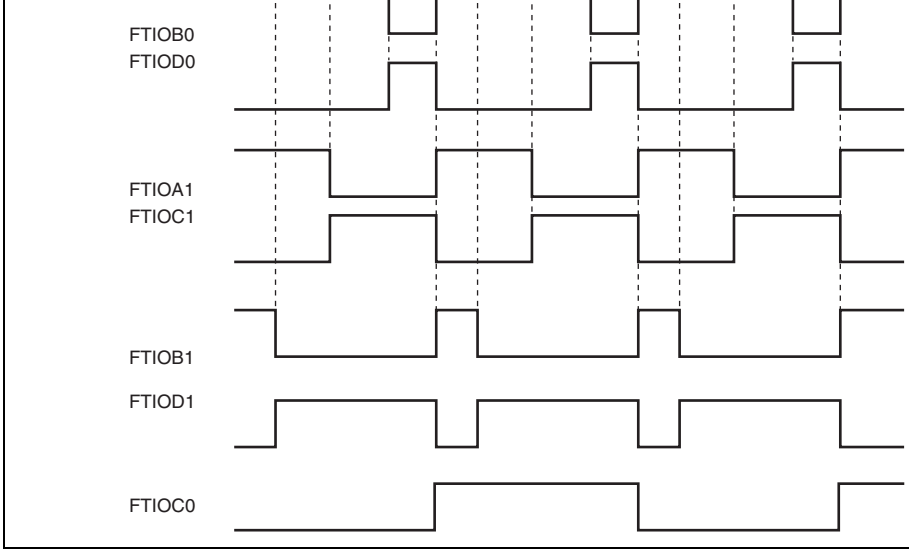


Figure 12.28 Example of Reset Synchronous PWM Mode Operation (OLS0 = O

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

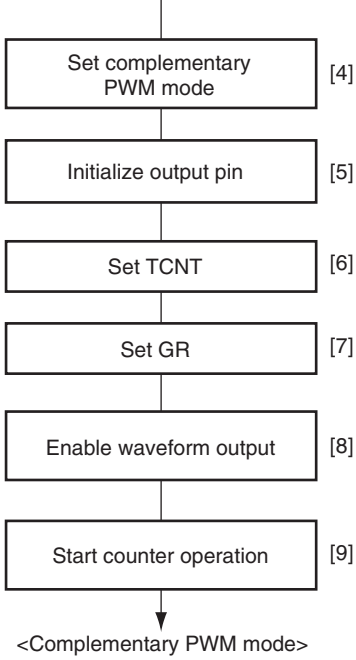
The PWM pin outputs 0 or 1 whenever a compare match between GRB_0, GRA_1, GR TCNT_0 or counter clearing occur.

For details on operations when reset synchronous PWM mode and buffer operation are simultaneously set, refer to section 12.4.8, Buffer Operation.

Table 12.6 Output Pins in Complementary PWM Mode

Channel	Pin Name	I/O	Pin Function
0	FTIOC0	Output	Toggle output in synchronous with PWM cycle
0	FTIOB0	Output	PWM output 1
0	FTIOD0	Output	PWM output 1 (counter-phase waveform non-ov with PWM output 1)
1	FTIOA1	Output	PWM output 2
1	FTIOC1	Output	PWM output 2 (counter-phase waveform non-ov with PWM output 2)
1	FTIOB1	Output	PWM output 3
1	FTIOD1	Output	PWM output 3 (counter-phase waveform non-ov with PWM output 3)

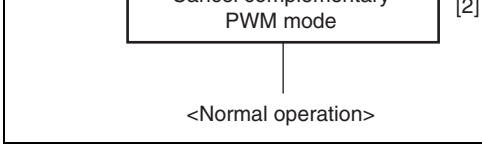
GRB_1	Set a changing point of the PWM waveform output from pins FTIOB1 a FTIOD1.



[4] Use bits CMD1 and CMD0 in TFCR to set complementary PWM mode. FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 automatically become PWM output pins.
[5] Set H'00 to TOCR.
[6] TCNT_1 must be H'0000. Set a non-overlapped period to TCNT_0.
[7] GRA_0 is a cycle register. Set the cycle to GRA_0. Set the timing to change the PWM output waveform to GRB_0, GRA_1, and GRB_1. Note that the timing must be set within the range of compare match carried out for TCNT_0 and TCNT_1. For GR settings, see Setting GR Value in Complementary PWM Mode in section 12.4.7, Complementary PWM Mode.
[8] Use TOER to enable or disable the timer output.
[9] Set the STR0 and STR1 bits in TSTR to 1 to start the count operation.

Note: To re-enter complementary PWM mode, first, enter a mode other than the complemen PWM mode. After that, repeat the setting procedures from step [1].
For settings of waveform outputs with a duty cycle of 0% and 100%, see Examples of Complementary PWM Mode Operation and Setting GR Value in Complementary PWM Mode in section 12.4.7, Complementary PWM Mode.

Figure 12.29 Example of Complementary PWM Mode Setting Procedure



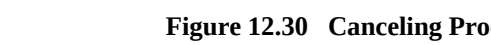


Figure 12.30 Canceling Procedure of Complementary PWM Mode

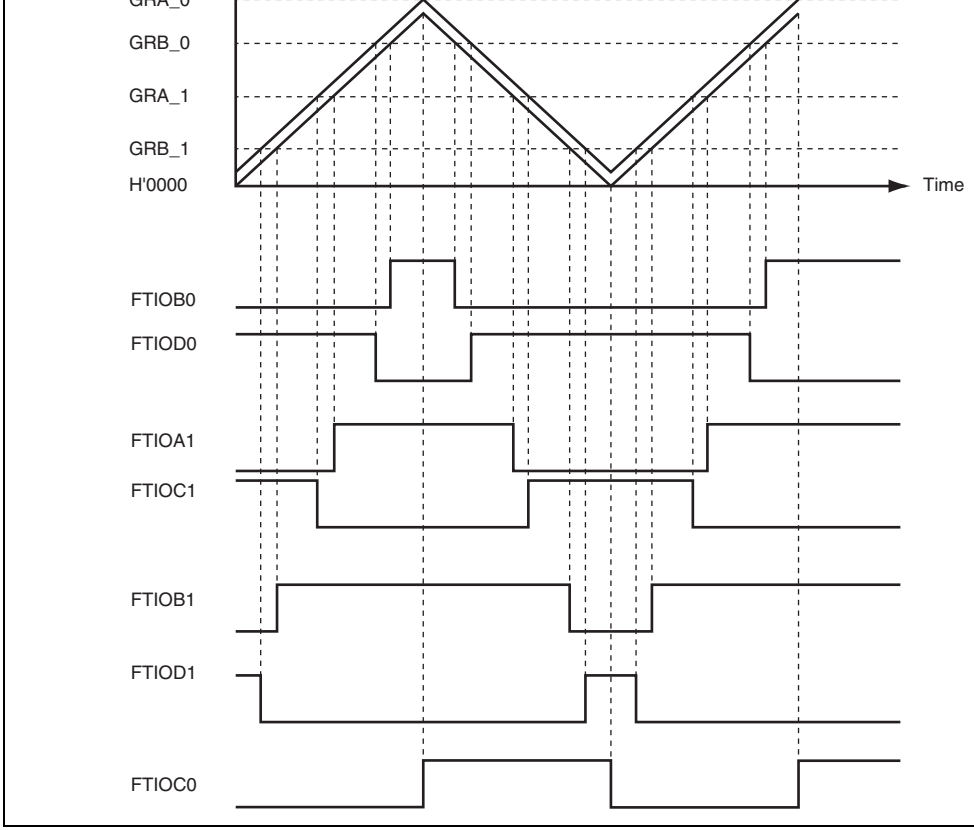


Figure 12.31 Example of Complementary PWM Mode Operation (1)

cycle waveform output, see Setting GR Value in Complementary PWM Mode: C, O
waveform with a duty cycle of 0% and 100% in section 12.4.7, Complementary PW

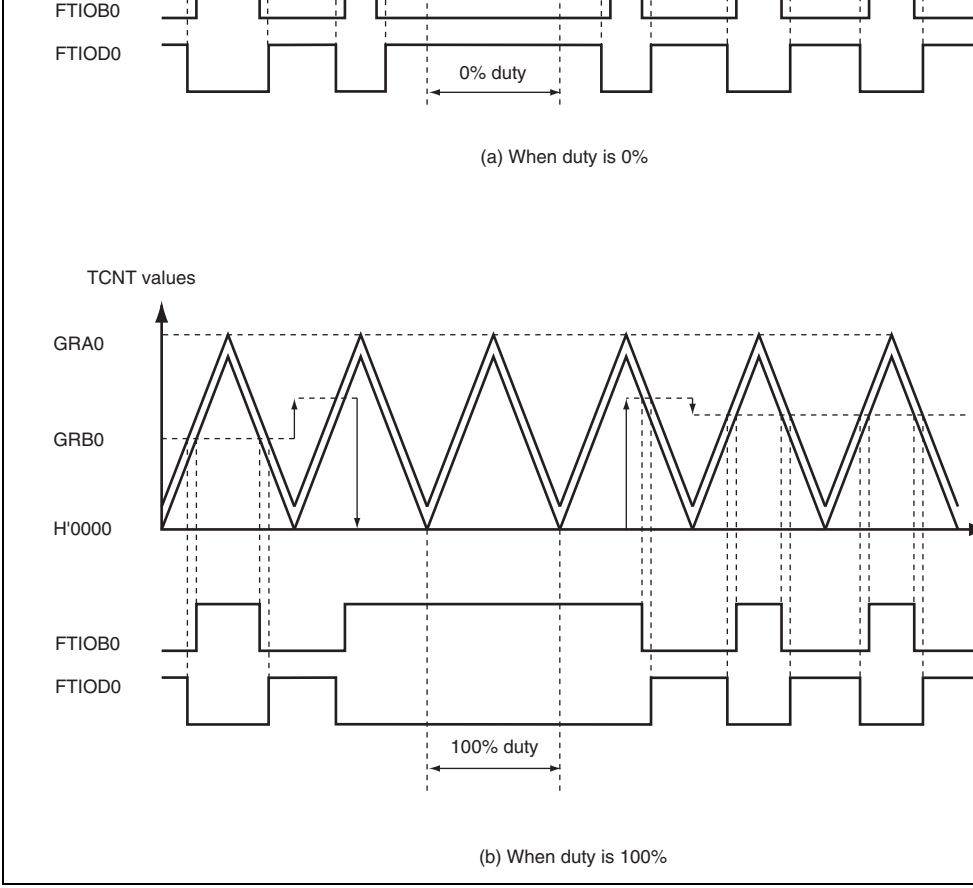


Figure 12.32 (1) Example of Complementary PWM Mode Operation (TPSC2 = TPSC1 = TPSC0 = 0) (2)

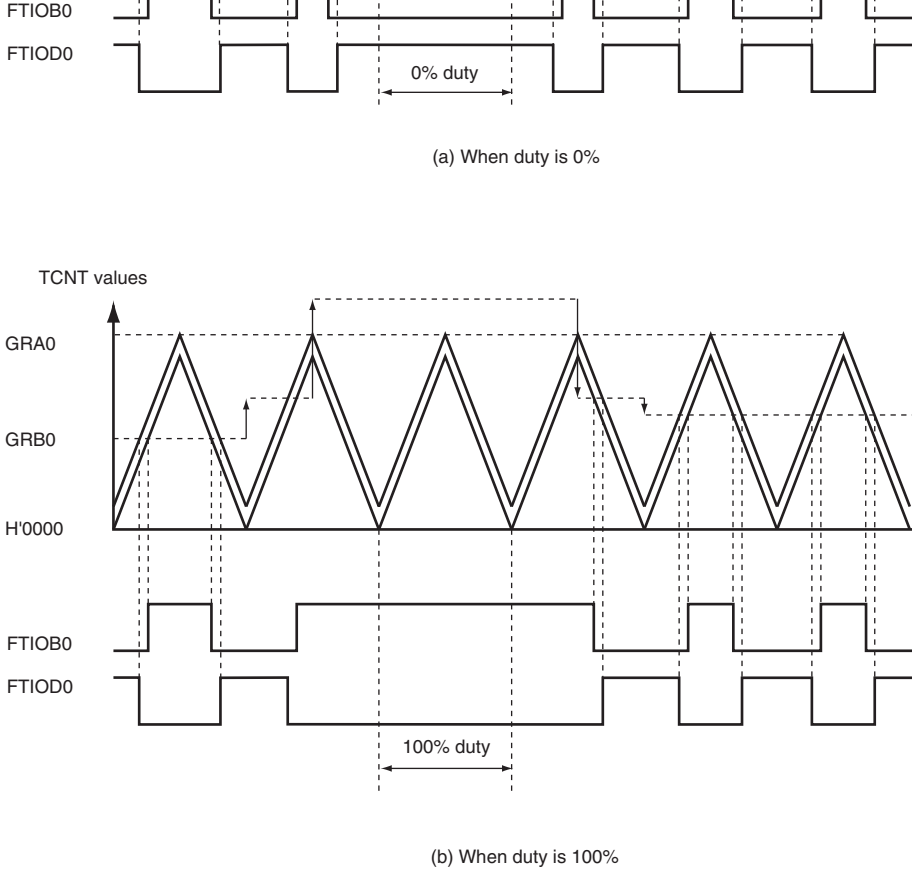


Figure 12.32 (2) Example of Complementary PWM Mode Operation (Other than TPSC2 = TPSC1 = TPSC0) (3)

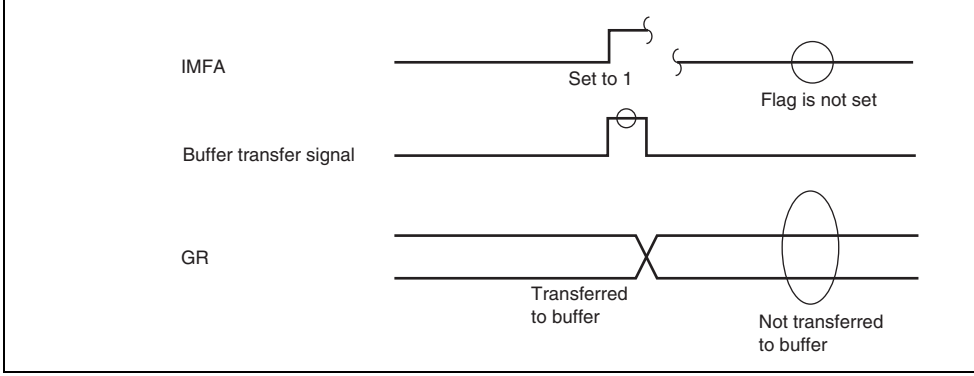


Figure 12.33 Timing of Overshooting

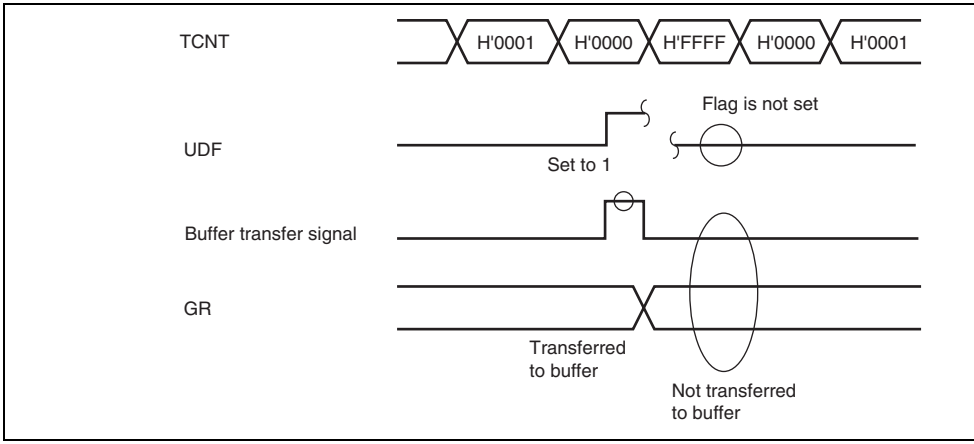


Figure 12.34 Timing of Undershooting

1. Initial value
   - When other than TPSC2 = TPSC1 = TPSC0 = 0, the GRA_0 value must be equa H'FFFC or less. When TPSC2 = TPSC1 = TPSC0 = 0, the GRA_0 value can be s H'FFFF or less.
   - H'0000 to T – 1 (T: Initial value of TCNT0) must not be set for the initial value.
   - GRA_0 – (T – 1) or more must not be set for the initial value.
   - When using buffer operation, the same values must be set in the buffer registers a corresponding general registers.
2. Modifying the setting value
   - Writing to GR directly must be performed while the TCNT_1 and TCNT_0 valu satisfy the following expression: H'0000 ≤ TCNT_1 < previous GR value, and pr value < TCNT_0 ≤ GRA_0. Otherwise, a waveform is not output correctly. For c outputting a waveform with a duty cycle of 0% and 100%, see 3., Outputting a w with a duty cycle of 0% and 100%.
   - Do not write the following values to GR directly. When writing the values, a wav not output correctly.

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

     H'0000 < GR ≤ T – 1 and GRA_0 – (T – 1) ≤ GR < GRA_0 + 1 when TPSC2 = TPSC0 = 0
   - Do not change settings of GRA_0 during operation.
3. Outputting a waveform with a duty cycle of 0% and 100%

   A. When buffer operation is not used and TPSC2 = TPSC1 = TPSC0 = 0.
      - Write H'0000 or a value equal to or more than the GRA_0 value to GR direct timing shown below.
      - To output a 0%-duty cycle waveform, write a value equal to or more than the value while H'0000 ≤ TCNT_1 < previous GR value.

Write H'0000 or a value equal to or more than the GRA_0 value to the buffer regis

- To output a 0%-duty cycle waveform, write a value equal to or more than the GRA_0value to the buffer register.
- To output a 100%-duty cycle waveform, write H'0000 to the buffer register.

For details on buffer operation, see section 12.4.8, Buffer Operation.

C. When buffer operation is not used and other than TPSC2 = TPSC1 = TPSC0 = 0.

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

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

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

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

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

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

Write a value which satisfies GRA_0 + 1 < GR < H'FFFF to the buffer register. A waveform with a duty cycle of 0% can be output. However, a waveform with a du of 100% cannot be output using the buffer operation. Also, the buffer operation ca used to change duty cycles while a waveform with a duty cycle of 100% is being o For details on buffer operation, see section 12.4.8, Buffer Operation.

GRA	GRC
GRB	GRD

When GR is Output Compare Register: When a compare match occurs, the value in t register of the corresponding channel is transferred to the general register.

This operation is illustrated in figure 12.35.

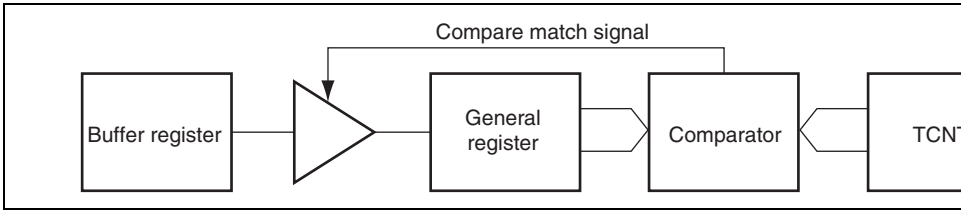


Figure 12.35 Compare Match Buffer Operation

When GR is Input Capture Register: When an input capture occurs, the value in TCN transferred to the general register and the value previously stored in the general register transferred to the buffer register.

This operation is illustrated in figure 12.36.

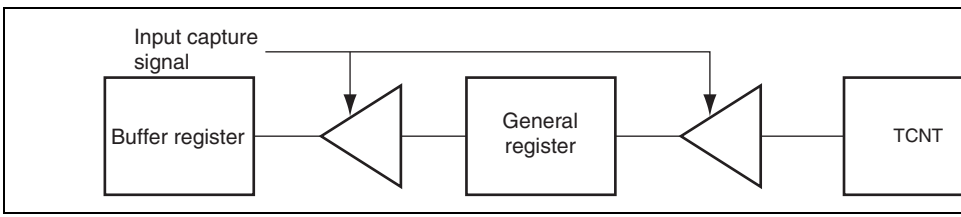


Figure 12.36 Input Capture Buffer Operation

operation setting procedure.

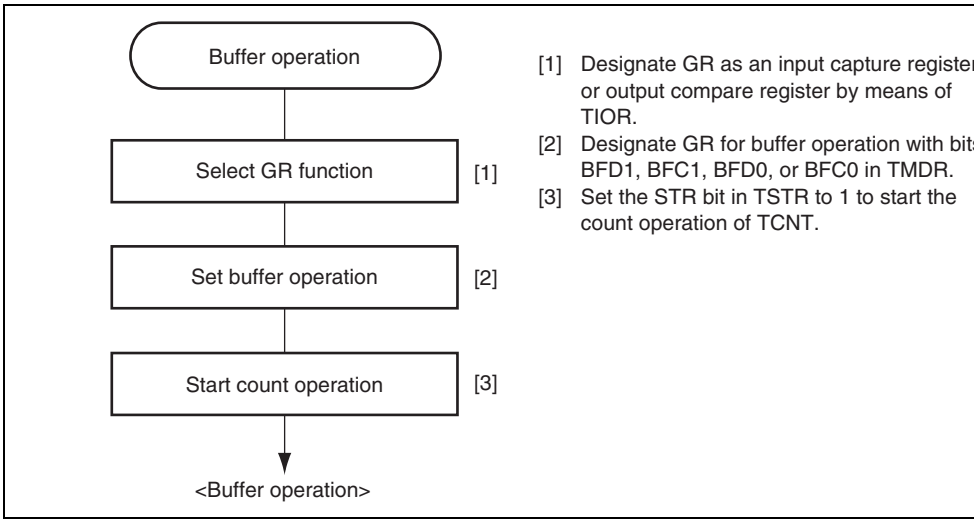


Figure 12.37 Example of Buffer Operation Setting Procedure

The timing to transfer data is shown in figure 12.39.

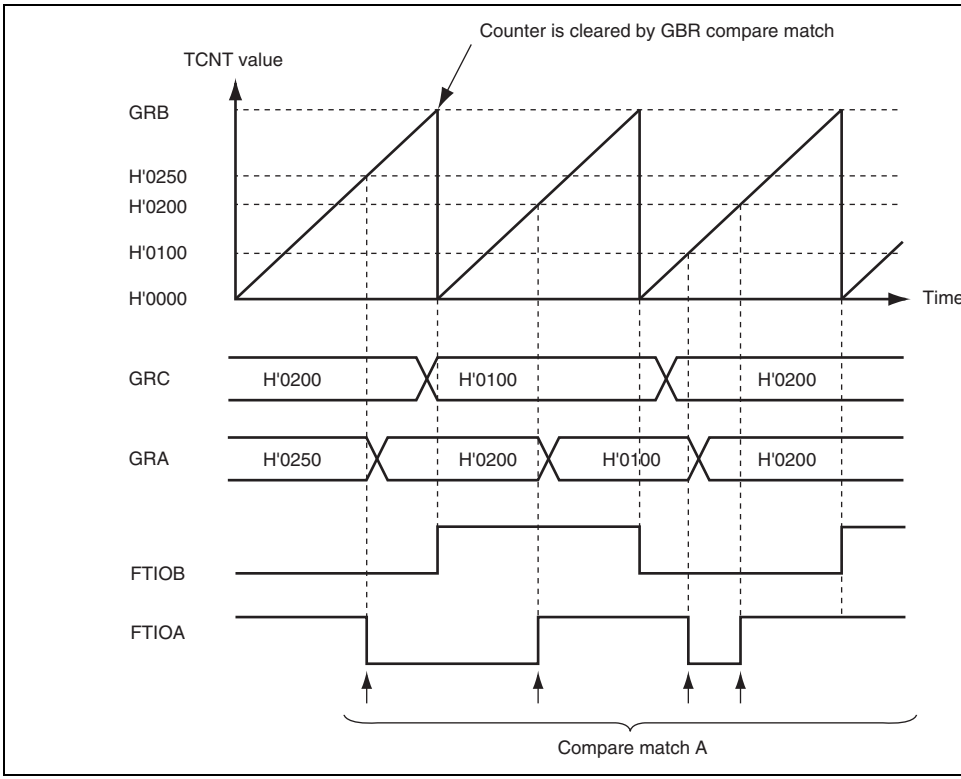


Figure 12.38 Example of Buffer Operation (1)
(Buffer Operation for Output Compare Register)

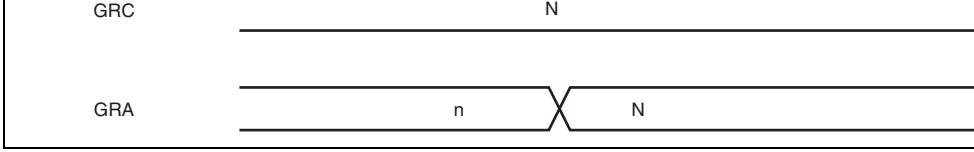


Figure 12.39 Example of Compare Match Timing for Buffer Operation

Figure 12.40 shows an operation example in which GRA has been designated as an input register, and buffer operation has been designated for GRA and GRC.

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

As buffer operation has been set, when the TCNT value is stored in GRA upon the occurr input capture A, the value previously stored in GRA is simultaneously transferred to GRC transfer timing is shown in figure 12.41.

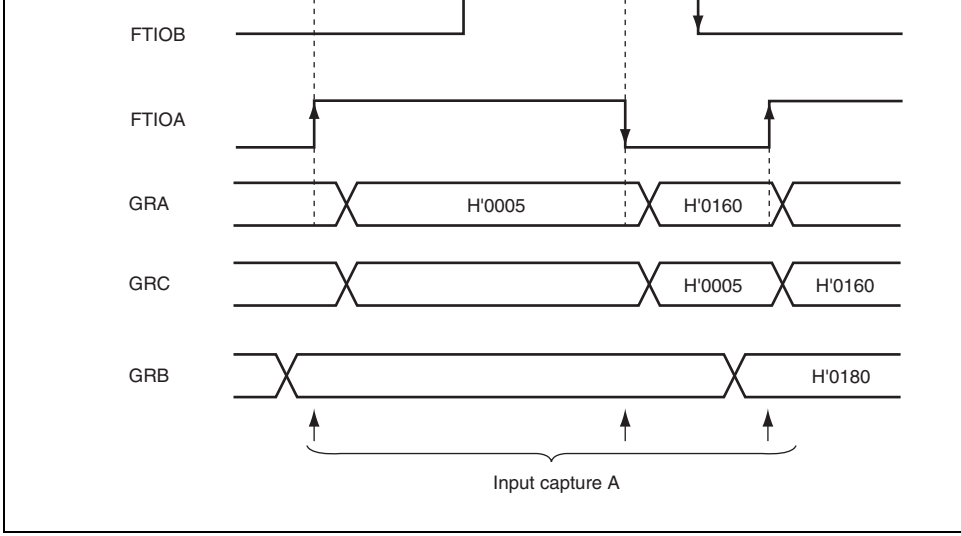


Figure 12.40 Example of Buffer Operation (2)
(Buffer Operation for Input Capture Register)

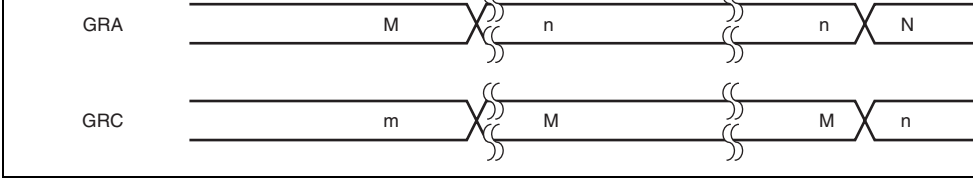


Figure 12.41 Input Capture Timing of Buffer Operation

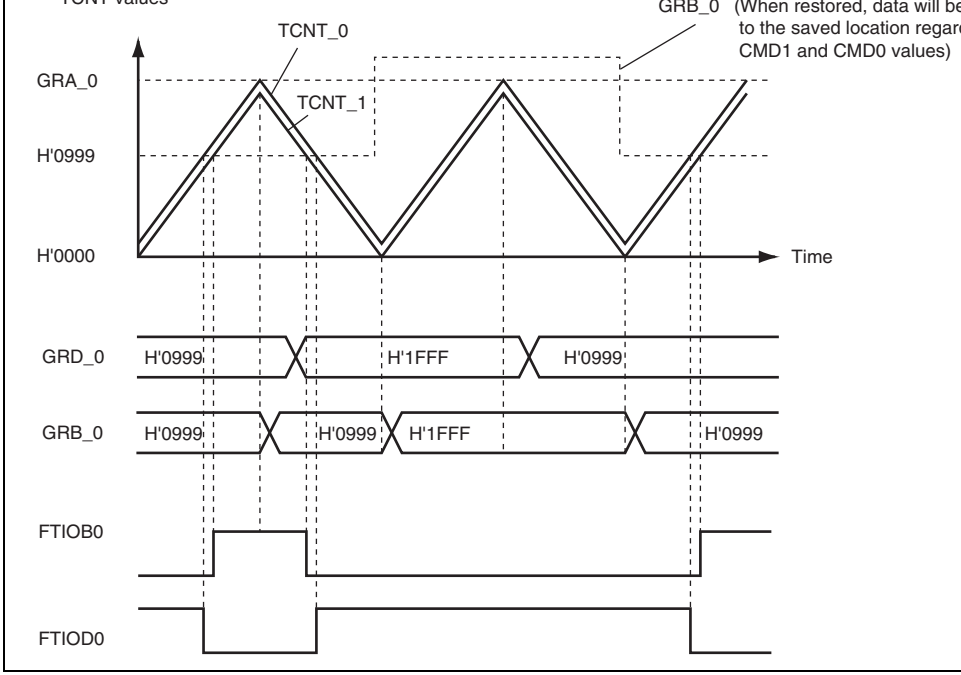


Figure 12.42 Buffer Operation (3)
(Buffer Operation in Complementary PWM Mode CMD1 = CMD0 = 1)

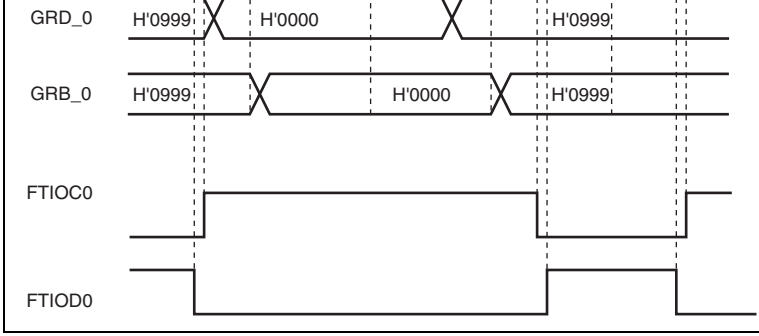


Figure 12.43 Buffer Operation (4)
(Buffer Operation in Complementary PWM Mode CMD1 = CMD0 = 1)

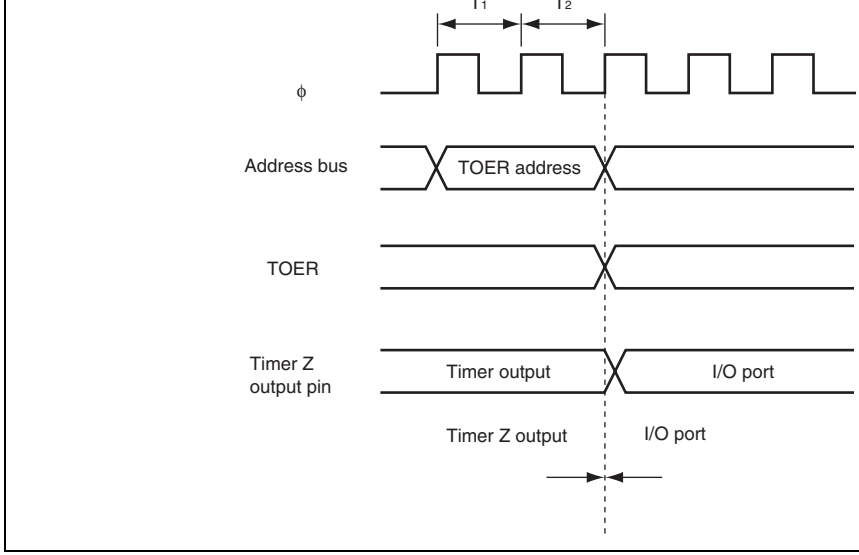


Figure 12.44 Example of Output Disable Timing of Timer Z by Writing to T

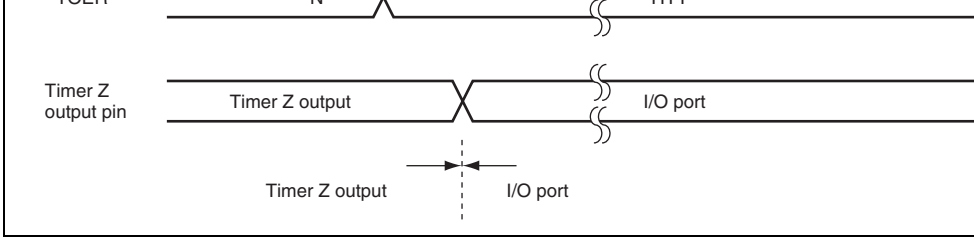


Figure 12.45 Example of Output Disable Timing of Timer Z by External Trig

Output Inverse Timing by TFCR: The output level can be inverted by inverting the OL
OLS0 bits in TFCR in reset synchronous PWM mode or complementary PWM mode. Fig
12.46 shows the timing.

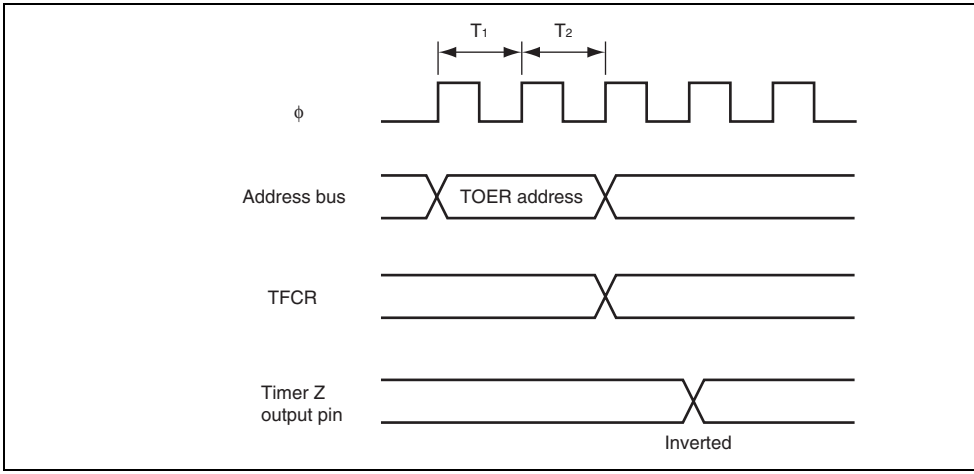


Figure 12.46 Example of Output Inverse Timing of Timer Z by Writing to TF

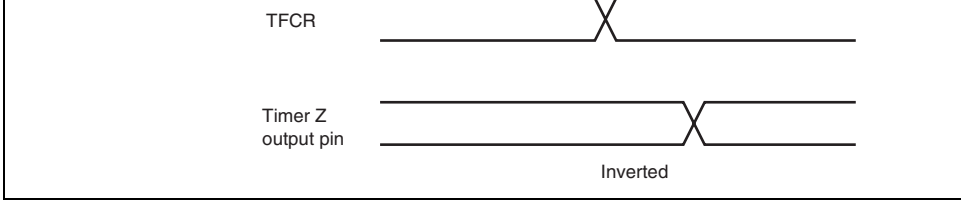


Figure 12.47 Example of Output Inverse Timing of Timer Z by Writing to P

matching (timing to update the counter value when the GR and TCNT match). Therefore, the TCNT and GR matches, the compare match signal will not be generated until the TCN clock is generated. Figure 12.48 shows the timing to set the IMF flag.

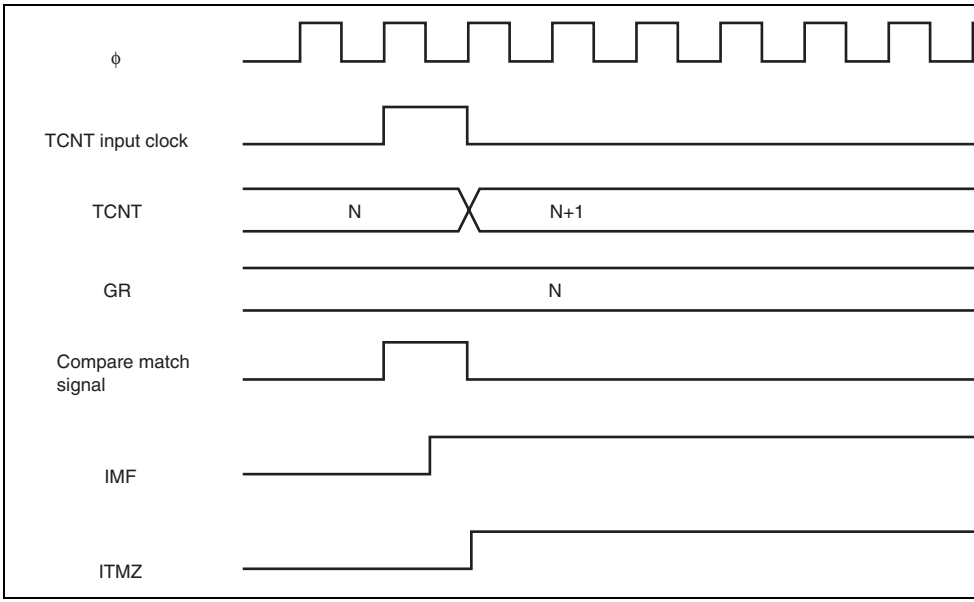


Figure 12.48 IMF Flag Set Timing when Compare Match Occurs

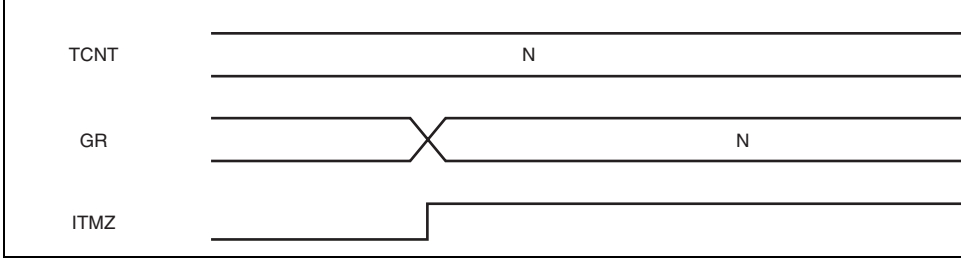


Figure 12.49 IMF Flag Set Timing at Input Capture

Overflow Flag (OVF) Set Timing: The overflow flag is set to 1 when the TCNT overfl
Figure 12.50 shows the timing.

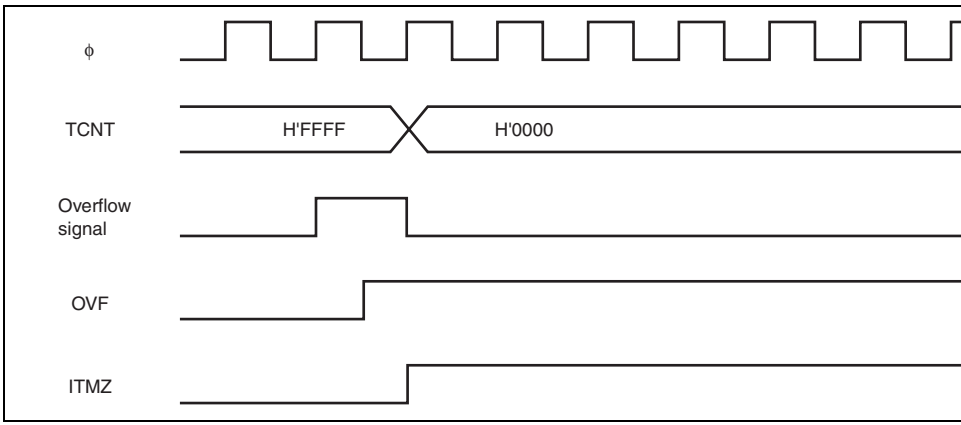


Figure 12.50 OVF Flag Set Timing

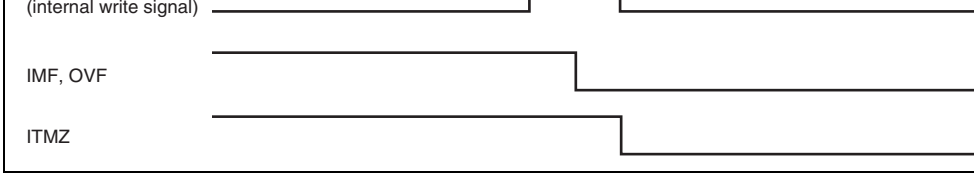


Figure 12.51 Status Flag Clearing Timing

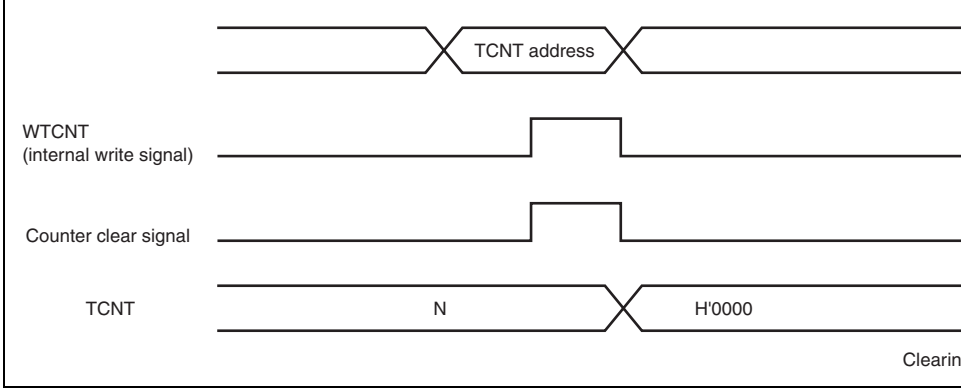


Figure 12.52 Contention between TCNT Write and Clear Operations

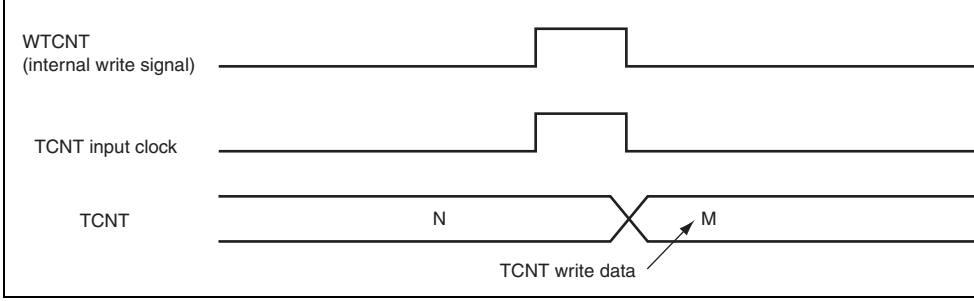


Figure 12.53 Contention between TCNT Write and Increment Operations

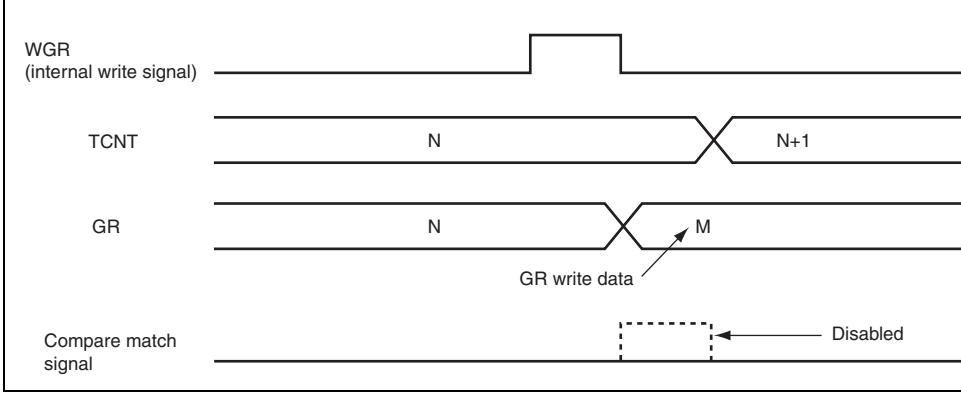


Figure 12.54 Contention between GR Write and Compare Match

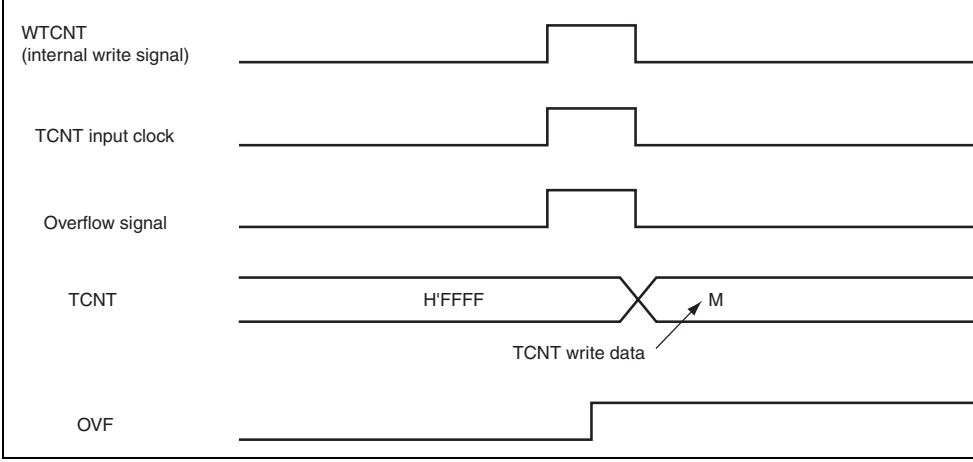


Figure 12.55 Contention between TCNT Write and Overflow

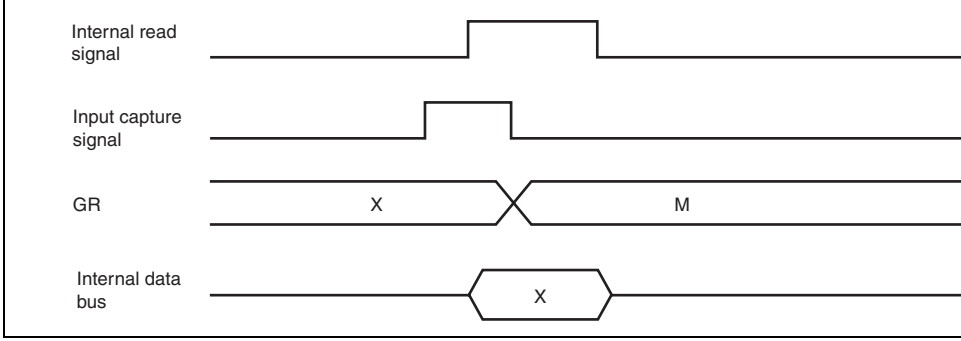


Figure 12.56 Contention between GR Read and Input Capture

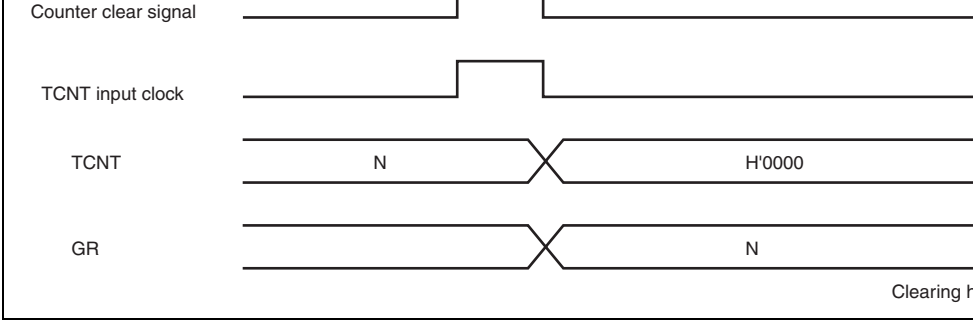


Figure 12.57 Contention between Count Clearing and Increment Operation by Input Capture

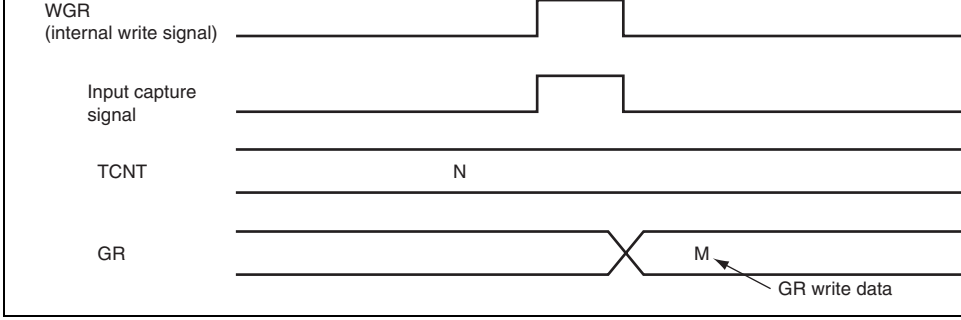


Figure 12.58 Contention between GR Write and Input Capture

Notes on Setting Reset Synchronous PWM Mode/Complementary PWM Mode: Wh
CMD1 and CMD0 in TFCR are set, note the following:

1. Write bits CMD1 and CMD0 while TCNT_1 and TCNT_0 are halted.
2. Changing the settings of reset synchronous PWM mode to complementary PWM mo
versa is disabled. Set reset synchronous PWM mode or complementary PWM mode
normal operation (bits CMD1 and CMD0 are cleared to 0) has been set.

MOV.B R0L,@TSR

Note on Writing to the TOA0 to TOD0 Bits and the TOA1 to TOD1 Bits in TOCR:
The TOA0 to TOD0 bits and the TOA1 to TOD1 bits in TOCR decide the value of the FT
which is output until the first compare match occurs. Once a compare match occurs and t
compare match changes the values of FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 outpu
values of the FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pin output and the values read
TOA0 to TOD0 and TOA1 to TOD1 bits may differ. Moreover, when the writing to TOC
the generation of the compare match A0 to D0 and A1 to D1 occur at the same timing, th
to TOCR has the priority. Thus, output change due to the compare match is not reflected
FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pins. Therefore, when bit manipulation inst
used to write to TOCR, the values of the FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pi
may result in an unexpected result. When TOCR is to be written to while compare match
operating, stop the counter once before accessing to TOCR, read the port 6 state to reflec
values of FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 output, to TOA0 to TOD0 and TO
TOD1, and then restart the counter. Figure 12.59 shows an example when the compare m
the bit manipulation instruction to TOCR occur at the same timing.

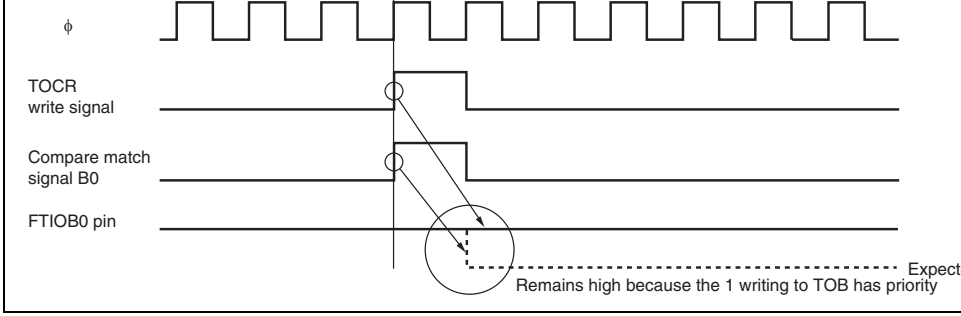


Figure 12.59 When Compare Match and Bit Manipulation Instruction to TOCR the Same Timing

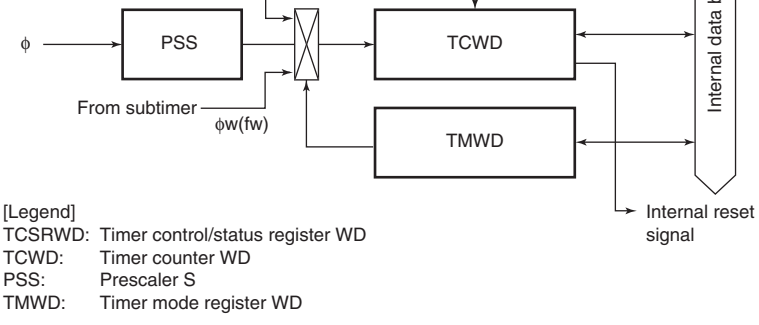


Figure 13.1 Block Diagram of Watchdog Timer

13.1 Features

- Selectable from nine counter input clocks.
  Eight internal clock sources (φ/64, φ/128, φ/256, φ/512, φ/1024, φ/2048, φ/4096, and
  or the internal oscillator (WDT and SBT) can be selected as the timer-counter clock.
  internal oscillator is selected, it can operate as the watchdog timer in any operating m
- Reset signal generated on counter overflow
  An overflow period of 1 to 256 times the selected clock can be set.

[Legend]
WDT: Watchdog timer
SBT: Subtimer

TCSRWD performs the TCSRWD and TCWD write control. TCSRWD also controls the watchdog timer operation and indicates the operating state. TCSRWD must be rewritten by the MOV instruction. The bit manipulation instruction cannot be used to change the setti

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

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

Bit	Bit Name	Initial Value	R/W	Description
7	CKS7	1	R/W	Clock Select 7 Selects the subtimer internal oscillator. CKS7 CKS3 0 1 : SBT internal oscillator X 0 : WDT internal oscillator 1 1 : WDT internal oscillator
6 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 φ/8192 0XXX: WDT internal oscillator For the internal oscillator overflow periods, see s 22, Electrical Characteristics.

[Legend]
X: Don't care.

Figure 13.2 shows an example of watchdog timer operation.

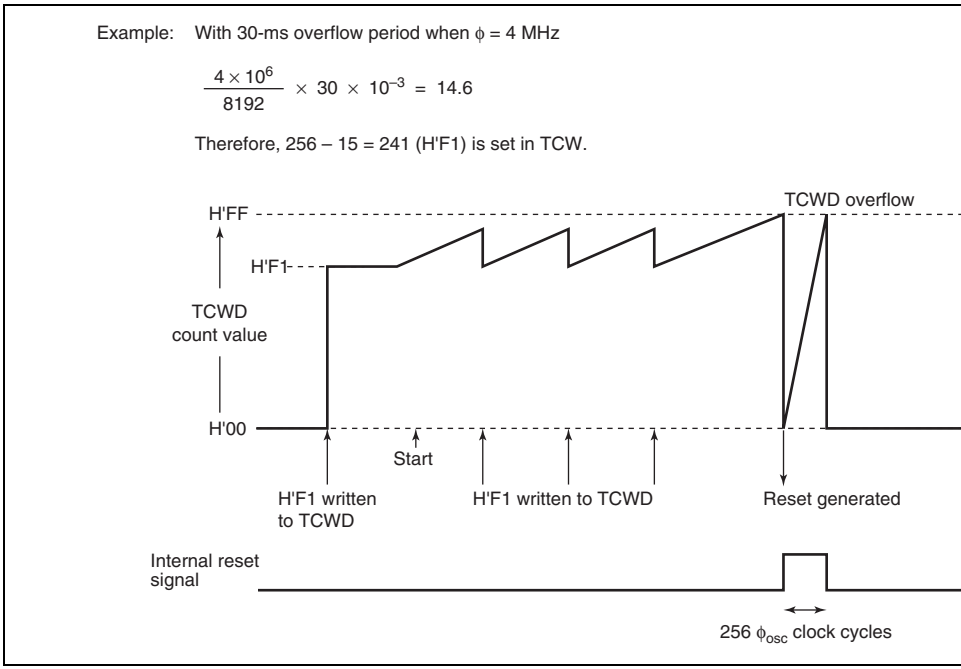


Figure 13.2 Watchdog Timer Operation Example

Table 14.1 shows the SCI3 channel configuration and figure 14.1 shows a block diagram
SCI3. Since pin functions are identical for each of the two channels (SCI3 and SCI3_2*
explanations are not given in this section.

Notes: 1. Only one channel is available in the H8/36037 Group.
2. The H8/36037 Group does not have the SCI3_2.

14.1 Features

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

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

  Double-buffering is used in both the transmitter and the receiver, enabling continuou
  transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected
- External clock or on-chip baud rate generator can be selected as a transfer clock sou
- Six interrupt sources

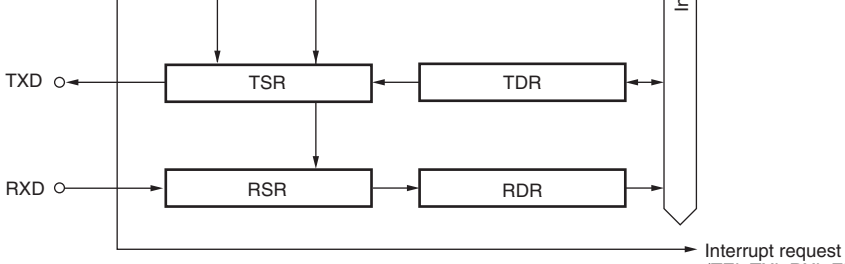
  Transmit-end, transmit-data-empty, receive-data-full, overrun error, framing error, a
  error.

Asynchronous mode

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Receive error detection: Parity, overrun, and framing errors

		RXD TXD	BRR	H'FFA9
			SCR3	H'FFAA
			TDR	H'FFAB
			SSR	H'FFAC
			RDR	H'FFAD
			RSR	—
			TSR	—
Channel 2*[2]	SCI3_2	SCK3_2 RXD_2 TXD_2	SMR_2	H'F740
			BRR_2	H'F741
			SCR3_2	H'F742
			TDR_2	H'F743
			SSR_2	H'F744
			RDR_2	H'F745
			RSR_2	—
			TSR_2	—

Notes: 1. Channel 1 is used in on-board programming mode by boot mode.
2. The H8/36037 Group does not have the channel 2.



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

Figure 14.1 Block Diagram of SCI3

14.3 Register Descriptions

The SCI3 has the following registers for each channel.

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

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

14.3.3 Transmit Shift Register (TSR)

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

14.3.4 Transmit Data Register (TDR)

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

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

and the baud rate, see section 14.3.8, Bit Rate
(BRR). n is the decimal representation of the va
BRR (see section 14.3.8, Bit Rate Register (BR

14.3.6 Serial Control Register 3 (SCR3)

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

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

2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, TEI interrupt request is
1	CKE1	0	R/W	Clock Enable 0 and 1
0	CKE0	0	R/W	Selects the clock source. • Asynchronous mode 00: On-chip baud rate generator 01: On-chip baud rate generator Outputs a clock of the same frequency as th from the SCK3 pin. 10: External clock Inputs a clock with a frequency 14 times the from the SCK3 pin. 11:Reserved • Clocked synchronous mode 00: On-chip clock (SCK3 pin functions as clock o 01: Reserved 10: External clock (SCK3 pin functions as clock i 11: Reserved

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

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

[Asynchronous Mode]

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

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

[Clocked Synchronous Mode]

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

[Legend]

B: Bit rate (bit/s)

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

ϕ: Operating frequency (MHz)

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

1200	0	51	0.14	0	54	–0.70	0	63	0.00	0	77
2400	0	25	0.14	0	26	1.14	0	31	0.00	0	38
4800	0	12	0.14	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	—	—

2400	0	47	0.00	0	51	0.14	0	63	0.00	0	64
4800	0	23	0.00	0	25	0.14	0	31	0.00	0	32
9600	0	11	0.00	0	12	0.14	0	15	0.00	0	15
19200	0	5	0.00	0	6	–6.99	0	7	0.00	0	7
31250	—	—	—	0	3	0.00	0	4	–1.70	0	4
38400	0	2	0.00	0	2	8.51	0	3	0.00	0	3

[Legend]

—: A setting is available but error occurs.

1200	0	155	0.14	0	159	0.00	0	191
2400	0	77	0.14	0	79	0.00	0	95
4800	0	38	0.14	0	39	0.00	0	47
9600	0	19	–2.34	0	19	0.00	0	23
19200	0	9	–2.34	0	9	0.00	0	11
31250	0	5	0.00	0	5	2.40	0	6
38400	0	4	–2.34	0	4	0.00	0	5

2400	0	103	0.14	0	127	0.00	0	129	0.14	0	15
4800	0	51	0.14	0	63	0.00	0	64	0.14	0	77
9600	0	25	0.14	0	31	0.00	0	32	–1.36	0	38
19200	0	12	0.14	0	15	0.00	0	15	1.73	0	19
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11
38400	0	6	-6.99	0	7	0.00	0	7	1.73	0	9

[Legend]

—: A setting is available but error occurs.

1200	1	79	0.00	1	90	0.14	1	95	0.00	1	103
2400	0	159	0.00	0	181	0.14	0	191	0.00	0	207
4800	0	79	0.00	0	90	0.14	0	95	0.00	0	103
9600	0	39	0.00	0	45	–0.93	0	47	0.00	0	51
19200	0	19	0.00	0	22	–0.93	0	23	0.00	0	25
31250	0	11	2.40	0	13	0.00	0	14	–1.70	0	15
38400	0	9	0.00	—	—	—	0	11	0.00	0	12

2400	0	233	0.14	1	64	0.14
4800	0	114	0.14	0	129	0.14
9600	0	58	–0.96	0	64	0.14
19200	0	28	1.02	0	32	–1.36
31250	0	17	0.00	0	19	0.00
38400	0	14	–2.34	0	15	1.73

[Legend]

—: A setting is available but error occurs.

Table 14.4 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	8	250000	0
2.097152	65536	0	0	9.8304	307200	0
2.4576	76800	0	0	10	312500	0
3	93750	0	0	12	375000	0
3.6864	115200	0	0	12.288	384000	0
4	125000	0	0	14	437500	0
4.9152	153600	0	0	14.7456	460800	0
5	156250	0	0	14	500000	0
6	187500	0	0	17.2032	537600	0
6.144	192000	0	0	18	562500	0
7.3728	230400	0	0	20	625000	0

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

[Legend]

Blank: No setting is available.

—: A setting is available but error occurs.

*: Continuous transfer is not possible.

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

[Legend]

Blank: No setting is available.

—: A setting is available but error occurs.

*: Continuous transfer is not possible.

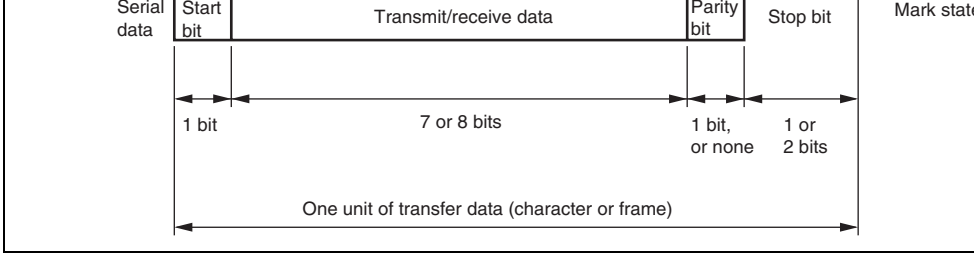


Figure 14.2 Data Format in Asynchronous Communication

14.4.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock
the SCK3 pin can be selected as the SCI3's serial clock, according to the setting of the C
SMR and the CKE0 and CKE1 bits in SCR3. When an external clock is input at the SCK
clock frequency should be 14 times the bit rate used.

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

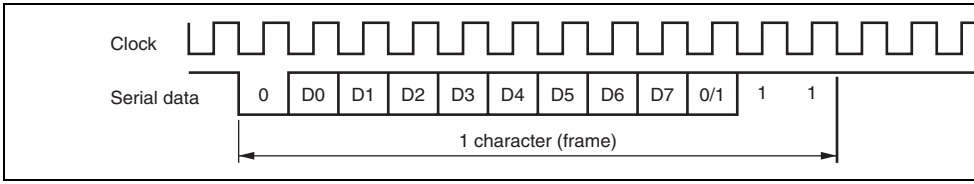
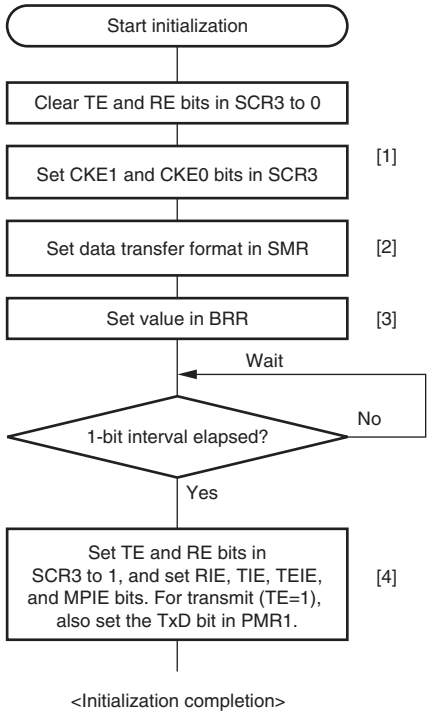


Figure 14.3 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode) (Example with 8-Bit Data, Parity, Two Stop Bits)



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

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

[2] Set the data transfer format in SMR.

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

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

Figure 14.4 Sample SCI3 Initialization Flowchart

3. The SCI3 checks the TDRE flag at the timing for sending the stop bit.
4. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, an serial transmission of the next frame is started.
5. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then t state" is entered, in which 1 is output. If the TEIE bit in SCR3 is set to 1 at this time, a interrupt request is generated.
6. Figure 14.6 shows a sample flowchart for transmission in asynchronous mode.

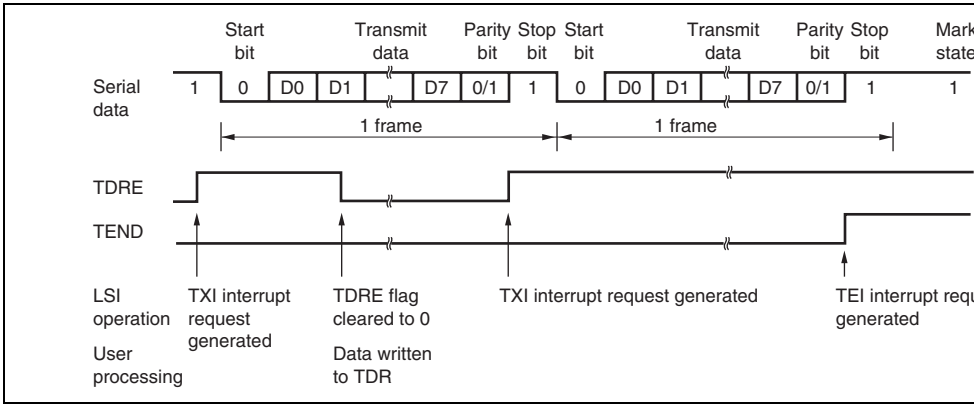
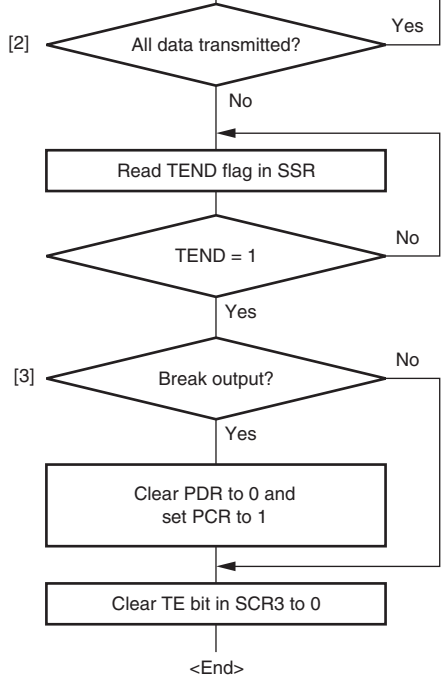


Figure 14.5 Example of SCI3 Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)



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

Figure 14.6 Sample Serial Transmission Data Flowchart (Asynchronous Mo

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

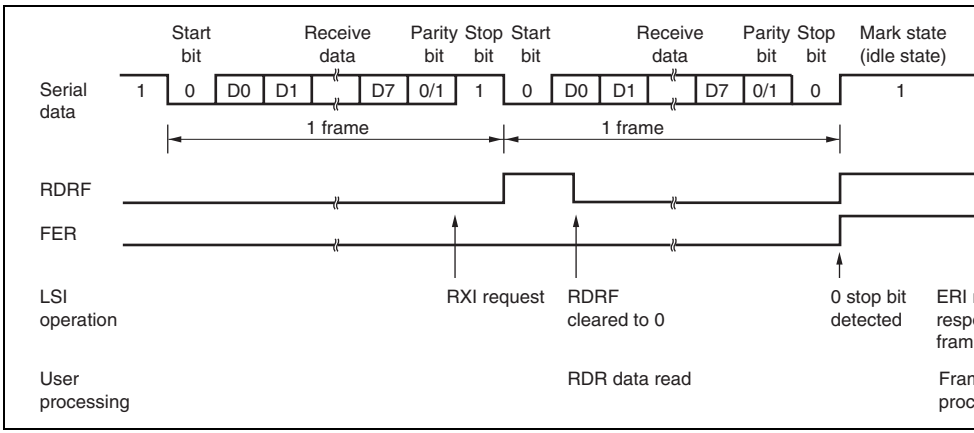
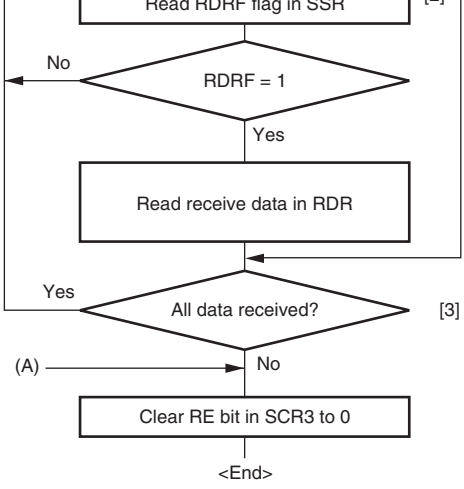


Figure 14.7 Example of SCI3 Reception in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)

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

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



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

Figure 14.8 Sample Serial Reception Data Flowchart (Asynchronous Mode)

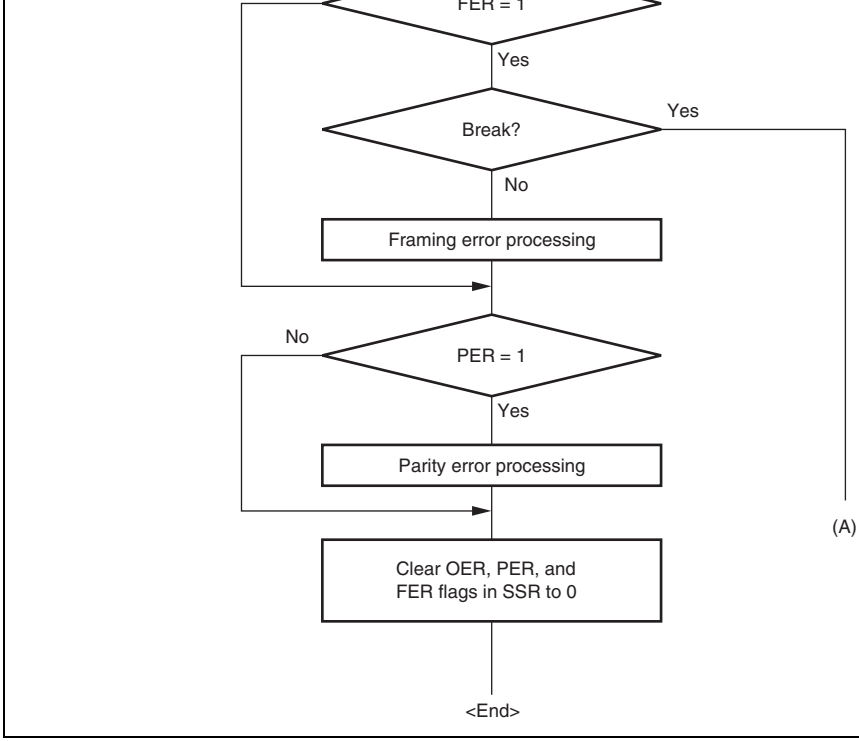


Figure 14.8 Sample Serial Reception Data Flowchart (Asynchronous Mode)

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

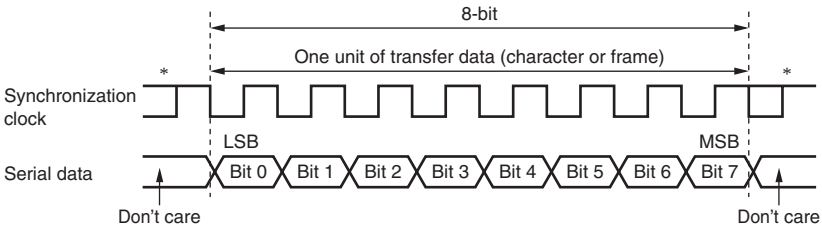


Figure 14.9 Data Format in Clocked Synchronous Communication

14.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK3 pin can be selected, according to the setting of th bit in SMR and CKE0 and CKE1 bits in SCR3. When the SCI3 is operated on an internal the synchronization clock is output from the SCK3 pin. Eight synchronization clock pulse output in the transfer of one character, and when no transfer is performed the clock is fixe

14.5.2 SCI3 Initialization

Before transmitting and receiving data, the SCI3 should be initialized as described in a sa flowchart in figure 14.4.

has been specified. Serial data is transmitted sequentially from the LSB (bit 0), from pin.

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

Figure 14.11 shows a sample flow chart for serial data transmission. Even if the TDRE f cleared to 0, transmission will not start while a receive error flag (OER, FER, or PER) is Make sure that the receive error flags are cleared to 0 before starting transmission.

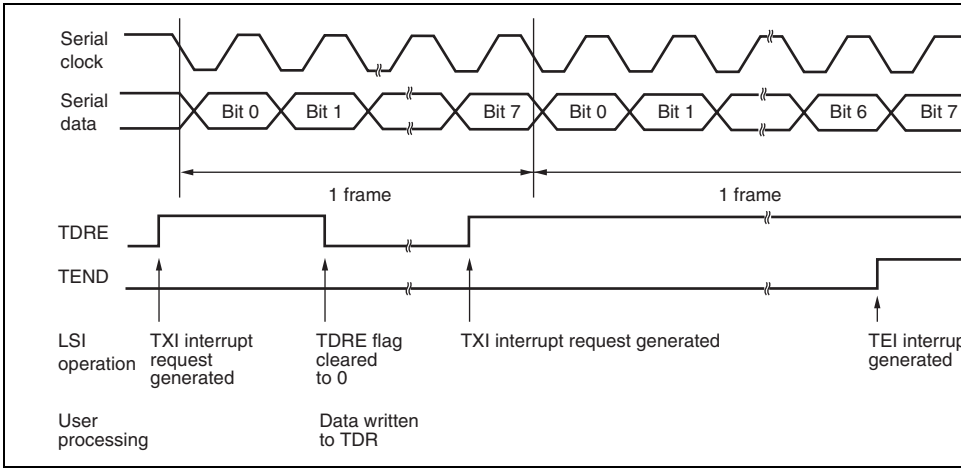


Figure 14.10 Example of SCI3 Transmission in Clocked Synchronous Mo

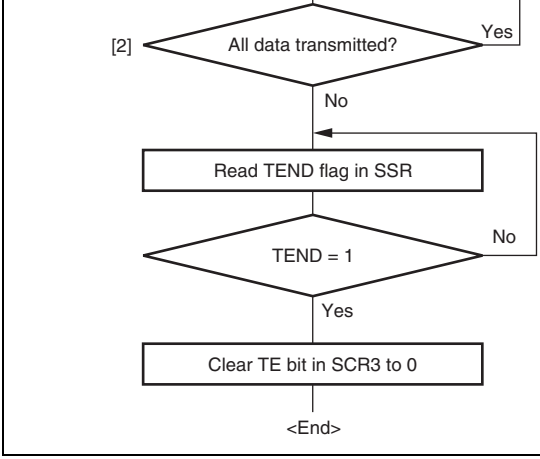


Figure 14.11 Sample Serial Transmission Flowchart (Clocked Synchronous M

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

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

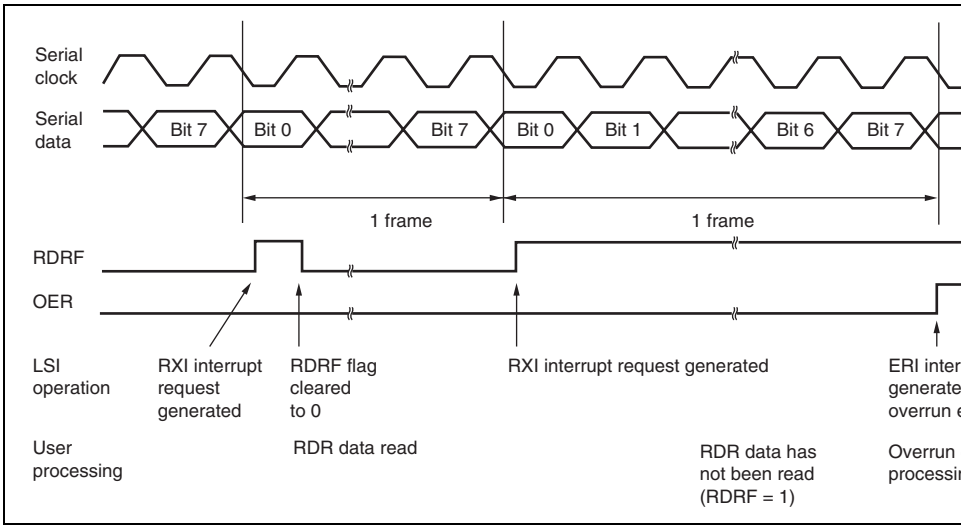
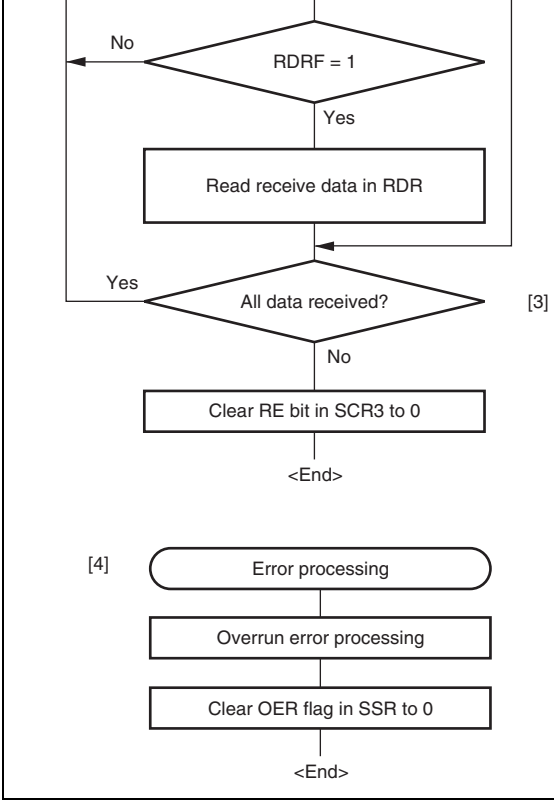


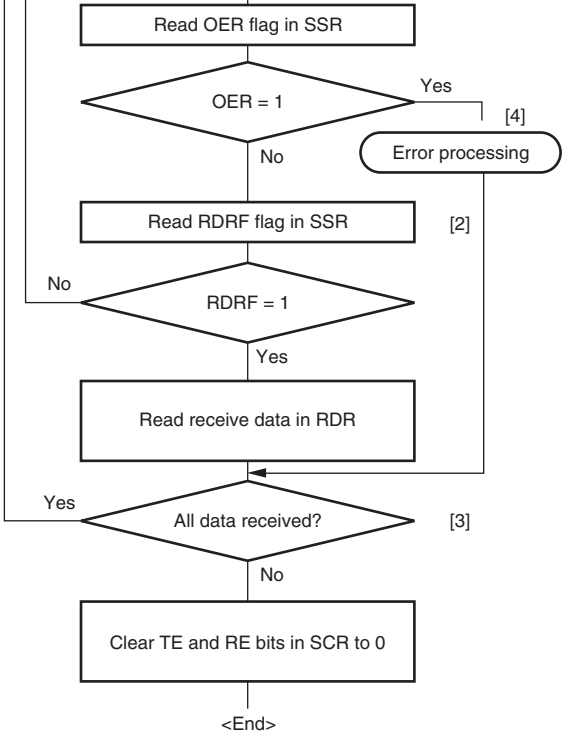
Figure 14.12 Example of SCI3 Reception in Clocked Synchronous Mode

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



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

Figure 14.13 Sample Serial Reception Flowchart (Clocked Synchronous 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.

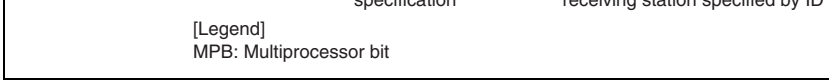
[4] If an overrun error occurs, read t
OER flag in SSR, and after
performing the appropriate error
processing, clear the OER flag t
Transmission/reception cannot b
resumed if the OER flag is set to
For overrun error processing, se
figure 14.13.

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

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

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



[Legend]
MPB: Multiprocessor bit

Figure 14.15 Example of Inter-Processor Communication Using Multiprocessor F (Transmission of Data H'AA to Receiving Station A)

14.6.1 Multiprocessor Serial Data Transmission

Figure 14.14 shows a sample flowchart for multiprocessor serial data transmission. For an transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmis cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI3 operations are as those in asynchronous mode.

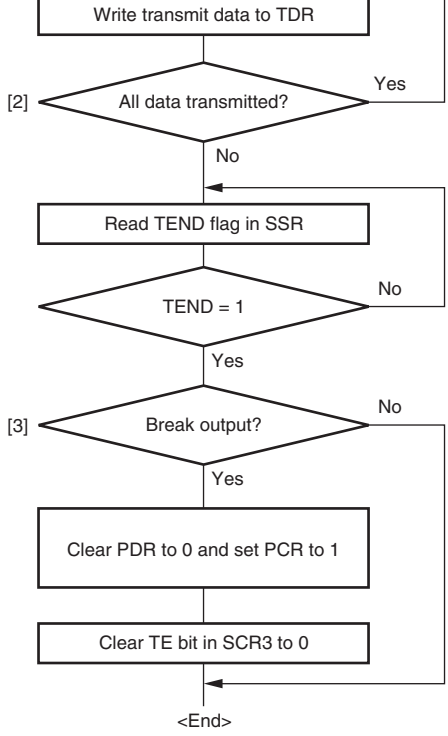


Figure 14.16 Sample Multiprocessor Serial Transmission Flowchart

14.6.2 Multiprocessor Serial Data Reception

Figure 14.17 shows a sample flowchart for multiprocessor serial data reception. If the M
SCR3 is set to 1, data is skipped until data with a 1 multiprocessor bit is sent. On receivi
with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI interrupt req

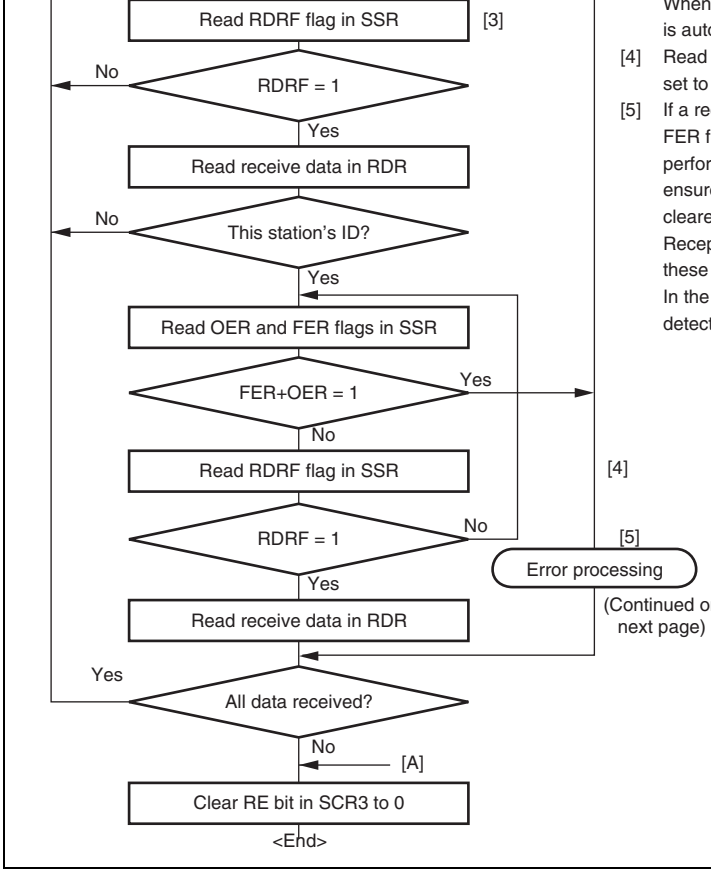


Figure 14.17 Sample Multiprocessor Serial Reception Flowchart (1)

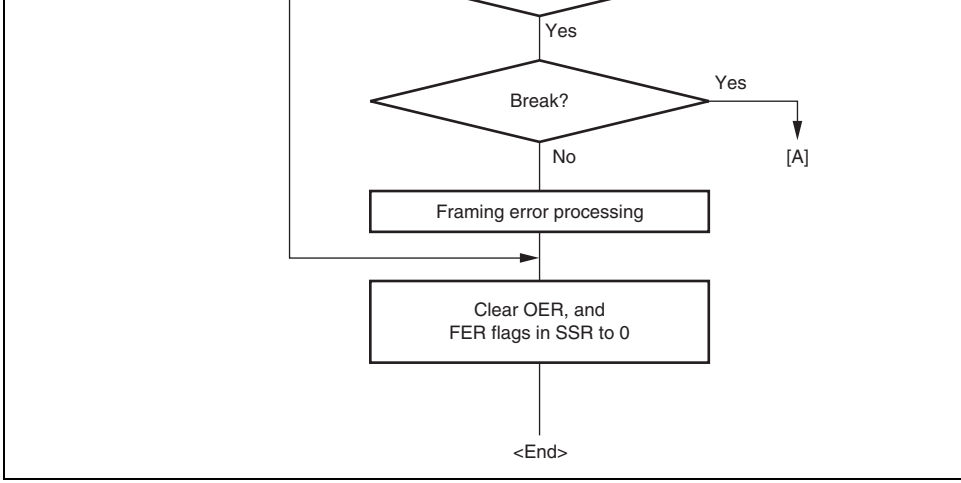


Figure 14.17 Sample Multiprocessor Serial Reception Flowchart (2)

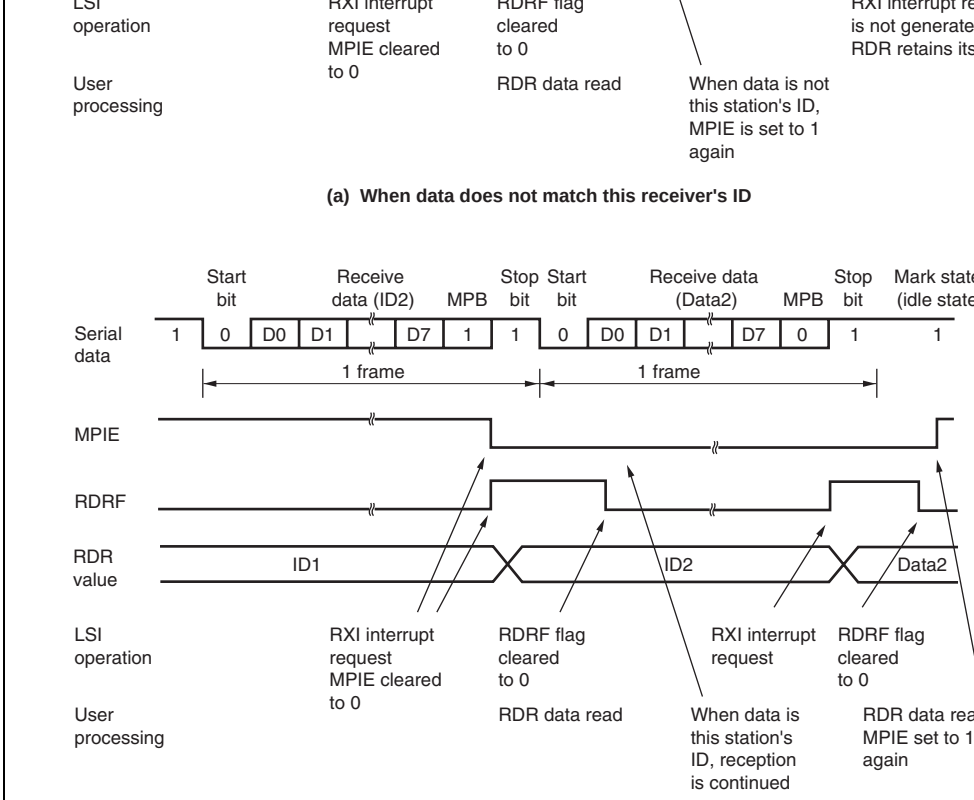


Figure 14.18 Example of SCI3 Reception Using Multiprocessor Format (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

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

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

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

14.8.3 Receive Error Flags and Transmit Operations (Clocked Synchronous Mod

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

[Legend]
N: Ratio of bit rate to clock (N = 14)
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.

$$M = \{0.5 - 1/(2 \times 14)\} \times 100\ [\%] = 46.875\%$$

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

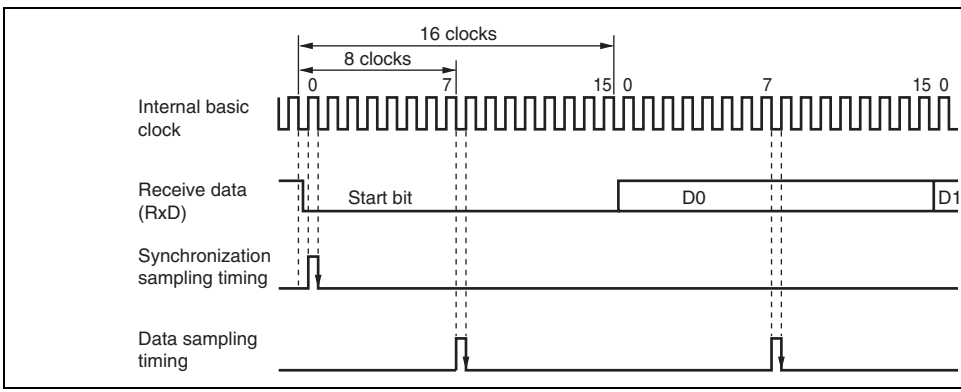


Figure 14.19 Receive Data Sampling Timing in Asynchronous Mode

Communication systems: NRZ (Non-Return to Zero) system (with bit-stuffing funct

Broadcast communication system

Transmission path: Bidirectional 2-wire serial communication

Communication speed: Max. 1 Mbps

Data length: 0 to 8 bytes

- Data buffers

  Four (one receive-only buffer and three buffers settable for transmission/reception)
- Data transmission

  Mailbox (buffer) number order (high-to-low)
- Data reception

  Message identifier match

  Reception with message identifier masked

  Supports four buffers for the filter mask
- CPU interrupt sources

  Various error interrupts

  Reset/Halt mode processing interrupt

  Message reception interrupt

  Message transmission interrupt
- TinyCAN operating modes

  Software reset

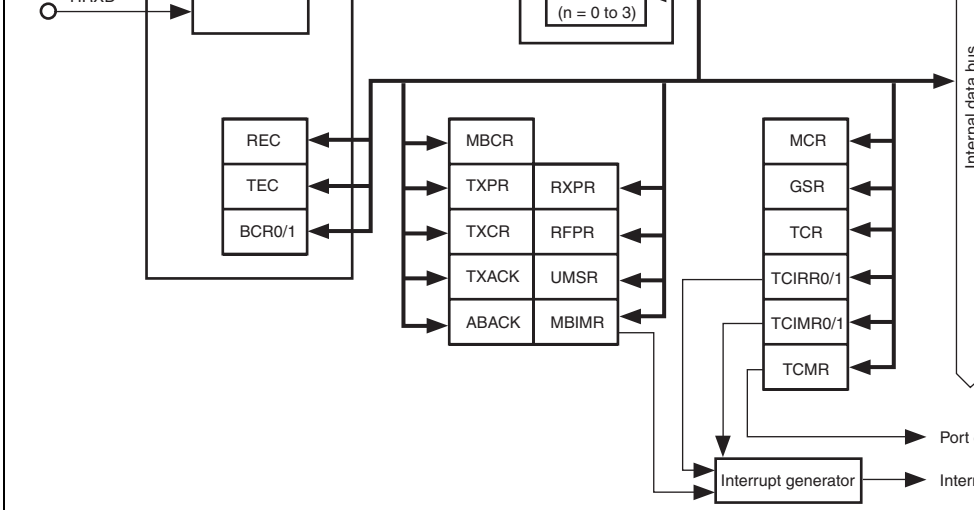
  Normal status (error-active, error-passive)

  Bus off state

  Configuration mode

  Halt mode

  Module standby mode



[Legend]

MCR:	Master control register
GSR:	General status register
TCR:	Test control register
TCIRR0/1:	TinyCAN interrupt register 0/1
TCIMR0/1:	TinyCAN interrupt mask register 0/1
MBCR:	Mailbox configuration register
TXPR:	Transmit pending register
TXCR:	Transmit pending cancel register
TXACK:	Transmit acknowledge register
ABACK:	Abort acknowlege register
RXPR:	Data frame receive complete register
RFPR:	Remote request register
UMSR:	Unread message status register
MBIMR:	Mailbox interrupt mask register
BCR0/1:	Bit configuration register 0/1
TEC:	Transmit error counter
REC:	Receive error counter
LAFMn:	Local acceptance filter mask (n = 0 to 3)
MCn0, MCn4 to MCn7:	Message control (n = 0 to 3)
MDn0 to MDn7:	Message data (n = 0 to 3)
TCMR:	TinyCAN module control register
CDLC:	CAN data link controller

The CDLC conforms to the Bosch CAN Ver.
standard, and performs transmission and re
of messages, CRC checking, bus arbitration

Figure 15.1 TinyCAN Block Diagram

Name	Abbreviation	I/O	Function
TinyCAN transmit data pin	HTXD	Output	CAN bus transmi
TinyCAN receive data pin	HRXD	Input	CAN bus recepti

15.3 Register Descriptions

The TinyCAN has the following registers.

- Test control register (TCR)
- Master control register (MCR)
- TinyCAN module control register (TCMR)
- General status register (GSR)
- Bit configuration registers 0, 1 (BCR0, BCR1)
- Mailbox configuration register (MBCR)
- Transmit pending register (TXPR)
- Transmit pending cancel register (TXCR)
- Transmit acknowledge register (TXACK)
- Abort acknowledge register (ABACK)
- Data frame receive complete register (RXPR)
- Remote request register (RFPR)
- Unread message status register (UMSR)
- TinyCAN interrupt registers 0, 1 (TCIRR0, TCIRR1)
- Mailbox interrupt mask register (MBIMR)
- TinyCAN interrupt mask registers 0, 1 (TCIMR0, TCIMR1)
- Transmit error counter (TEC)
- Receive error counter (REC)

Bit	Bit Name	Initial Value	R/W	Description
7	TSTMD	0	R/W	Test Mode Enables or disables the test mode. 0: TinyCAN in normal mode 1: TinyCAN in test mode
6	WREC	0	R/W	CAN Error Counters Write Enable Enables or disables write to TEC and REC. 0: TEC and REC can only be read 1: The same value can be written to CAN Error C (TEC and REC) simultaneously (enabled only mode)
5	FERPS	0	R/W	Force to Error Passive Mode Enables to force to the error-passive state. 0: CAN Error Counter is determined by TEC/REC 1: TinyCAN behaves as the error-passive state regardless of the TEC/REC value (enabled on mode)
4	ATACK	0	R/W	Auto-Acknowledge Enables generation of an auto-acknowledge bit i to execute the self-test. 0: Does not to generate its auto-acknowledge bit 1: Generates its auto-acknowledge bit (enabled test mode)

				0: Input from the CAN bus to the HRXD pin is e 1: Input from the CAN bus to the HRXD pin is d (enabled only in test mode) • When INTLE = 0, the HRXD pin always hol recessive data. • When INTLE = 1, data is input from the inte to the HRXD pin.
1	DTXOT	0	R/W	HTXD Pin Output Enable Enables or disables the HTXD pin to output the 0: Output from the HTXD pin to the CAN bus is 1: Output from the HTXD pin to the CAN bus is (enabled only in test mode) • When INTLE = 0, the HTXD pin always outp recessive data to the CAN bus. • When INTLE = 1, the internal HTXD outputs the internal HRXD.
0	INTLE	0	R/W	Internal Loop Enable Enables or disables connection between the int HTXD and internal HRXD. 0: Internal HRXD is supplied from the HRXD pi 1: Internal HRXD is supplied from the internal H (enabled only in test mode)

				bus. Communication with the CAN bus can be re by clearing this bit to 0 and then receiving 11 rec bits. 0: TinyCAN in normal mode 1: Halt mode is requested
0	RSTRQ	1	R/W	Reset Request Controls a software reset of the TinyCAN. After a has been requested and the initial state is entere the RESET bit in GSR and the RHI bit in TCIRR to 1. When this bit is cleared to 0, communicatio CAN bus is resumed. After powering on, this bit RESET bit are always set to 1. 0: TinyCAN in normal mode 1: Software reset of TinyCAN is requested

				the values of the TinyCAN registers are preserv 0: TinyCAN in normal mode 1: Module standby mode
6 to 2	—	All 0	—	Reserved These bits are always read as 0.
1	PMR97	0	R/W	Port Mode Register 97 Selects a function of the P97/HTXD pin. 0: P97 1: HTXD
0	PMR96	0	R/W	Port Mode Register 96 Selects a function of the P96/HRXD pin. 0: P96 1: HRXD

				Indicates whether the CDLC is in the error-passi This flag is always set to 1 when the CDLC is in passive state or bus off state. [Setting condition] When TEC ≥ 128 or REC ≥ 128 [Clearing condition] When the error-active state is entered
4	HALT	0	R	Halt Status Flag Indicates whether the TinyCAN is in halt mode. [Setting condition] When the CAN bus receives an intermission frar bus is idle with the HLTRQ bit in MCR set to 1 [Clearing condition] When the HLTRQ bit is cleared to 0 and halt mo exited
3	RESET	1	R	Reset Status Flag Indicates whether the TinyCAN is in reset mode. [Setting condition] When the TinyCAN is in the reset state [Clearing condition] When communication with the CAN bus is enabl the reset procedure completes

1	ECWRG	0	R	Error Counter Warning Flag Indicates an error warning. [Setting condition] When 96 ≤ TEC ≤ 256 or 96 ≤ REC ≤ 256 [Clearing condition] When TEC< 96, REC < 96, or TEC ≥ 256
0	BOFF	0	R	Bus Off Flag Indicates the bus off state. [Setting condition] When TEC ≥ 256 (bus off state) [Clearing condition] When the TinyCAN recovers from the bus off st

				00: 1 time quantum 01: 2 time quanta 10: 3 time quanta 11: 4 time quanta
5	BRP5	0	R/W	Baud Rate Prescaler
4	BRP4	0	R/W	These bits set the clock used for time quanta.
3	BRP3	0	R/W	000000: Setting prohibited
2	BRP2	0	R/W	000001: 2 system clocks
1	BRP1	0	R/W	: : (BRP + 1) system clocks
0	BRP0	0	R/W	111111: 64 system clocks

- BCR1

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0. The write value sho always be 0.

				101: PHSEG2 = 6 time quanta 110: PHSEG2 = 7 time quanta 111: PHSEG2 = 8 time quanta
3	TSG13	0	R/W	Time Segment 1
2	TSG12	0	R/W	This segment is used for absorbing the delay o
1	TSG11	0	R/W	output buffer, CAN bus, and input buffer. The T
0	TSG10	0	R/W	can be set within a range of 1 to 16 time quanta comprises PRSEG and PHSEG1 according to t specifications. 0000: Setting prohibited 0001: Setting prohibited 0010: Setting prohibited 0011: PRSEG + PHSEG1 = 4 time quanta : 1111: PRSEG + PHSEG1 = 16 time quanta

2	MB2	0	R/W	Mailboxes. 0: Corresponding Mailbox is configured as transm 1: Corresponding Mailbox is configured as recep
1	MB1	0	R/W	
0	—	1	—	Reserved This bit is always read as 1. This bit is relevant t receive-only Mailbox, and its value cannot be ch

3	MB3	0	R/W	[Setting condition]
2	MB2	0	R/W	When the corresponding MBCR bit for a mailbo
1	MB1	0	R/W	corresponding bit in TXPR is set to 1 (n = 3 to 1) [Clearing conditions] • When message transmission has complete successfully (TXACKn set) • When transmission cancellation for an untra message has finished (ABACKn set) • When a transmission cancellation request h occurred during message transmission, and occurs or arbitration is lost on the CAN bus set) • When a transmit error or arbitration loss oc the corresponding DART bit for a message transmitted set to 1 If the message is not transmitted successfully, t bit is not cleared to 0. If any of these MB bits in cleared to 0, the EMPI bit in TCIRR1 is set to 1 TinyCAN automatically attempts retransmission as the DART bit in the message control of the corresponding Mailbox is not set to 1 or the corresponding bit in TXCR is not set to 1. Note: When the MBn bit in MBCR is set to 1 TinyCAN does not transmit a message the MBn bit in TXPR is set to 1. To cle MBn bit in TXPR to 0, set the MBn bit to 1 beforehand.
0	—	0	—	Reserved This bit is always read as 0. This bit is relevant receive-only Mailbox, and its value cannot be c

3	MB3	0	R/W	[Setting condition]
2	MB2	0	R/W	The corresponding bit of a mailbox is set to 1
1	MB1	0	R/W	[Clearing condition]
				When the corresponding bit in TXPR is cleared (transmit message is canceled successfully)
				Note: Writing 1 to these bits is enabled only wh TXPR bit corresponding to Mailbox is set
0	—	0	—	Reserved
				This bit is always read as 0. This bit is relevant t receive-only Mailbox, and its value cannot be ch

15.3.9 Transmit Acknowledge Register (TXACK)

TXACK is a status flag that indicates the successful transmit completion of Mailbox trans messages.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved
				These bits are always read as 0.
3	MB3	0	R/(W)*	[Setting condition]
2	MB2	0	R/(W)*	When transmission of the message in the corres
1	MB1	0	R/(W)*	Mailbox has completed successfully
				[Clearing condition]
				When 1 is written to these bits

transmit request cancellation is completed, the bit in ABACK corresponding to the trans
message is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0.
3	MB3	0	R/(W)*	[Setting condition]
2	MB2	0	R/(W)*	When cancellation of the transmit message in t
1	MB1	0	R/(W)*	corresponding Mailbox has completed [Clearing condition] When 1 is written to these bits
0	—	0	—	Reserved This bit is always read as 0. This bit is relevant receive-only Mailbox, and its value cannot be c

Note: * Only 1 can be written to clear the flag.

				These bits are always read as 0.
3	MB3	0	R/(W)*	[Setting condition]
2	MB2	0	R/(W)*	When the corresponding Mailbox has completed reception of a data frame
1	MB1	0	R/(W)*	
0	MB0	0	R/(W)*	[Clearing condition]
				When 1 is written to these bits

Note: * Only 1 can be written to clear the flag.

15.3.12 Remote Request Register (RFPR)

RFPR is a status flag that indicates successful reception of remote frames in the correspo
Mailboxes. When a data frame is received, the corresponding RFPR bit is not set to 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved
				These bits are always read as 0.
3	MB3	0	R/(W)*	[Setting condition]
2	MB2	0	R/(W)*	When the corresponding Mailbox has completed reception of a remote frame
1	MB1	0	R/(W)*	
0	MB0	0	R/(W)*	[Clearing condition]
				When 1 is written to these bits

Note: * Only 1 can be written to clear the flag.

2	MB2	0	R/(W)*	overwritten/overrun an unread message.
1	MB1	0	R/(W)*	[Setting condition]
0	MB0	0	R/(W)*	When a new message is received before the corresponding bit in RXPR or RFPR is cleared
				[Clearing condition]
				When 1 is written to these bits

Note: * Only 1 can be written to clear the flag.

				[Setting condition] When an overload frame is transmitted [Clearing condition] When 1 is written to this bit
6	BOFI	0	R/(W)*	Bus Off Interrupt Flag Status flag indicating the bus off state caused by or recovery from the bus off state to the error-act [Setting condition] When TEC ≥ 256 or when 11 bits are received fo times in the bus off state [Clearing condition] When 1 is written to this bit
5	EPI	0	R/(W)*	Error Passive Interrupt Flag Status flag indicating the error-passive state cau REC or TEC. [Setting condition] When TEC ≥ 128 or REC ≥ 128 [Clearing condition] When 1 is written to this bit
4	ROWI	0	R/(W)*	Receive Overload Warning Interrupt Flag Status flag indicating the error warning state cau REC. [Setting condition] When REC ≥ 96 [Clearing condition] When 1 is written to this bit

				Status flag indicating that a remote frame has b received in a Mailbox. [Setting condition] When remote frame reception is completed, an corresponding MBIMR bit is 0 [Clearing condition] When all bits in RFPR are cleared to 0
1	DFRI	0	R	Data Frame Receive Message Interrupt Flag Status flag indicating that a data frame has bee in a Mailbox. [Setting condition] When message reception is completed, and the corresponding MBIMR bit is 0 [Clearing condition] When all bits in RXPR are cleared to 0
0	RHI	1	R/(W)*	Reset/Halt Interrupt Flag Status flag indicating that the TinyCAN has bee has entered halt mode. [Setting condition] When each processing has finished after a soft request (RSTRQ = 1) or a halt mode request (H 1) [Clearing condition] When 1 is written to this bit

Note: * Only 1 can be written to clear the flag.

				When the falling edge of HRXD is detected in sta mode [Clearing condition] When 1 is written to this bit
3, 2	—	All 0	—	Reserved These bits are always read as 0.
1	OVRI	0	R	Unread Message Interrupt Flag Status flag indicating that a new message has be received regardless of existence of an unread m The NMC bit in MCn0 (n = 0 to 3) will determine handle the newly received message: NMC = 1 s overwrite and NMC = 0 selects overrun (ignore). [Setting condition] When a new message is received with the MBIM corresponding to the receive message cleared to the corresponding bit in RXPR or RFPR set to 1 [Clearing condition] When all bits in UMSR are cleared to 0
0	EMPI	0	R	Mailbox Empty Interrupt Flag Status flag indicating that the next transmit mess be written to the Mailbox. [Setting condition] When TXPR is cleared to 0 by completion of tran or completion of transmission cancellation [Clearing condition] When TXACK and ABACK is cleared to 0

Note: * Only 1 can be written to clear the flag.

2	MB2	1	R/W	requests.
1	MB1	1	R/W	The interrupt source in a transmit Mailbox is cle the corresponding bit in TXPR caused by transı end or transmission cancellation. The interrupt a receive Mailbox is setting of the correspondin RXPR or RFPR caused by reception end. 0: An interrupt request in the corresponding Ma enabled 1: An interrupt request in the corresponding Ma disabled
0	MB0	1	R/W	

15.3.16 TinyCAN Interrupt Mask Registers 0, 1 (TCIMR0, TCIMR1)

TCIMR controls enabling or disabling of TCIRR interrupt requests. When the correspon
set to 1, the interrupt request is masked. This register corresponds to TCIRR.

- TCIMR0

Bit	Bit Name	Initial Value	R/W	Description
7	OVLIM	1	R/W	Overload Frame Transmit Interrupt Mask Enables or disables an interrupt request for ove frame transmission. 0: The interrupt request for the overload frame transmission is enabled 1: The interrupt request for the overload frame transmission is disabled

4	ROWIM	1	R/W	Receive Overload Warning Interrupt Mask Enables or disables an interrupt request for a rec overload warning. 0: The interrupt request for the receive overload is enabled 1: The interrupt request for the receive overload is disabled
3	TOWIM	1	R/W	Transmit Overload Warning Interrupt Mask Enables or disables an interrupt request for a tra overload warning. 0: The interrupt request for the transmit overload is enabled 1: The interrupt request for the transmit overload is disabled
2	RFRIM	1	R/W	Remote Frame Request Interrupt Mask Enables or disables an interrupt request for a rer frame request. 0: The interrupt request for the remote frame req enabled 1: The interrupt request for the remote frame req disabled
1	DFRIM	1	R/W	Data Frame Receive Message Interrupt Mask Enables or disables an interrupt request for a da receive message. 0: The interrupt request for the data frame receiv message is enabled 1: The interrupt request for the data frame receiv message is disabled

Bit	Bit Name	Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1.
4	WUPIM	1	R/W	Wakeup Interrupt Mask Enables or disables a wakeup interrupt request 0: The wakeup interrupt request is enabled 1: The wakeup interrupt request is disabled
3, 2	—	All 1	—	Reserved These bits are always read as 1.
1	OVRIM	1	R/W	Unread Message Interrupt Mask Enables or disables an interrupt request for an message. 0: The interrupt request for the unread message enabled 1: The interrupt request for the unread message disabled
0	EMPIM	1	R/W	Mailbox Empty Interrupt Mask Enables or disables an interrupt request for ma empty. 0: The interrupt request for mailbox empty is en 1: The interrupt request for mailbox empty is di

3	TEC3	0	R/W*	state.
2	TEC2	0	R/W*	TEC can be written to in test mode (TCR.TSTMD
1	TEC1	0	R/W*	TCR.WREC = 1). The same value can be written
0	TEC0	0	R/W*	TEC and REC. TEC should be written to in halt m other modes, a CAN bus communication error m depending on the TEC value. Note that TEC can written to only in test mode.

Note: * TEC can be written to only in test mode (TCR.TSTMD = 1 and TCR.WREC = 1 same value should be written to TEC and REC.

15.3.18 Receive Error Counter (REC)

REC counts the number of receive message errors on the CAN bus.

Bit	Bit Name	Initial Value	R/W	Description
7	REC7	0	R/W*	REC functions as a counter indicating the numbe
6	REC6	0	R/W*	receive message errors on the CAN bus. The co
5	REC5	0	R/W*	is stipulated in the CAN protocol. In normal opera REC can only be read, but can only be modified
4	REC4	0	R/W*	CDLC. REC is cleared to 0 by a reset request
3	REC3	0	R/W*	(MCR.RSTRQ = 1) or on entering the bus off sta
2	REC2	0	R/W*	REC can be written to in test mode (TCR.TSTMD
1	REC1	0	R/W*	TCR.WREC = 1). The same value can be written TEC and REC. REC should be written to in halt m
0	REC0	0	R/W*	other modes, a CAN bus communication error m depending on the REC value. Note that REC can written to only in test mode.

Note: * REC can be written to only in test mode (TCR.TSTMD = 1 and TCR.WREC = 1 same value should be written to TEC and REC.

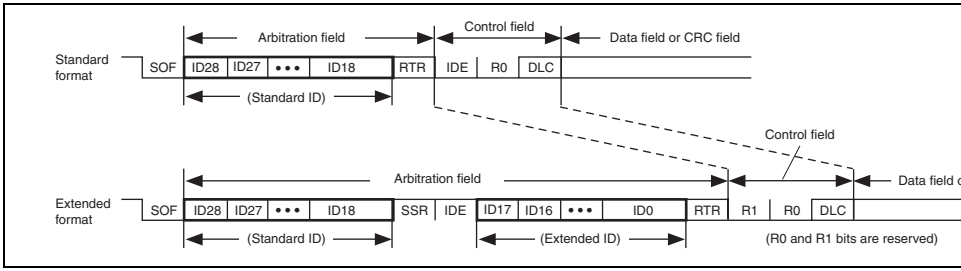


Figure 15.2 Standard Format and Extended Format

				new message, this bit selects whether to ove overwrite the unread message with the new r 0: The new receive message is ignored and t unread message is saved, and the corresp UMSR bit is set to 1 (overrun) 1: The unread message is lost by being over with the new receive message, and the corresponding UMSR bit is set to 1 (overw
	5, 4	—	—	Reserved These bits are always read as 0.
	3 to 0	DLC3 to DLC0	R/W	Data Length Code These bits set the transmit data length of dat and data length requested by remote frames bits are stipulated in Bosch 2.0B active. 0000: 0 bytes 0001: 1 byte 0010: 2 bytes 0011: 3 bytes 0100: 4 bytes 0101: 5 bytes 0110: 6 bytes 0111: 7 bytes 1xxx: 8 bytes
MCn[4] (n = 0 to 3)	7 to 5	ID20 to ID18	R/W	These bits set bits 2 to 0 in the standard iden data frames and remote frames.

				1: Extended format
	2	—	—	Reserved This bit is always read as 0.
	1, 0	ID17, ID16	R/W	These bits set bits 17 and 16 of the extende identifier and are stipulated in Bosch 2.0B a
MCn[5] (n = 0 to 3)	7 to 0	ID28 to ID21	R/W	These bits set bits 10 to 3 of the standard id and are stipulated in Bosch 2.0B active.
MCn[6] (n = 0 to 3)	7 to 0	ID7 to ID0	R/W	These bits set bits 7 to 0 of the extended ide and are stipulated in Bosch 2.0B active.
MCn[7] (n = 0 to 3)	7 to 0	ID15 to ID8	R/W	These bits set bits 15 to 8 of the extended id and are stipulated in Bosch 2.0B active.

(n = 0 to 3)		LAFMLn0		0: Receive message is stored in RXn bec RXn message identifier bits match the message identifier bits 1: Receive message is stored in RXn reg whether the RXn message identifier bit the receive message identifier bits
LAFMLn0 (n = 0 to 3)	7 to 0	LAFMLn15 to LAFMLn8	R/W	Filter mask for bits 15 to 8 of the extende identifier. 0: Receive message is stored in RXn bec RXn message identifier bits match the message identifier bits 1: Receive message is stored in RXn reg whether the RXn message identifier bit the receive message identifier bits
LAFMHn1 (n = 0 to 3)	7 to 5	LAFMHn7 to LAFMHn5	R/W	Filter mask for bits 2 to 0 of the standard 0: Receive message is stored in RXn bec RXn message identifier bits match the message identifier bits 1: Receive message is stored in RXn reg whether the RXn message identifier bit the receive message identifier bits

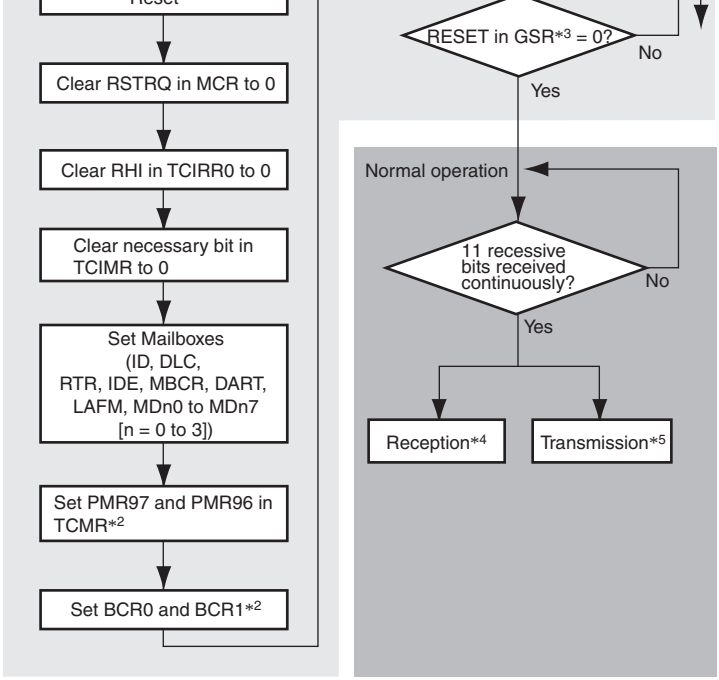
				the receive message identifier bits
LAFMHn0 (n = 0 to 3)	7 to 0	LAFMHn15 to LAFMHn8	R/W	Filter mask for bits 10 to 3 of the standa identifier. 0: Receive message is stored in RXn be RXn message identifier bits match the message identifier bits 1: Receive message is stored in RXn re whether the RXn message identifier b the receive message identifier bits

15.4.3 Message Data (MDn0 to MDn7 [n = 0 to 3])

The message data is configured as eight 8-bit registers for a single Mailbox. The transm receive data are stored from byte 0 in the low-to-high order. The TinyCAN has four sets message data. The bit order on the CAN bus is from 1 to 8 bytes. Since MDn0 to MDn7 3) are in RAM, initial values are undefined after power-on. Be sure to initialize these bit writing 0 or 1.

Mailbox 0	MD0[0]	MD0[1]	MD0[2]	MD0[3]	MD0[4]	MD0[5]	MD0[6]	MD0[7]
Mailbox 1	MD1[0]	MD1[1]	MD1[2]	MD1[3]	MD1[4]	MD1[5]	MD1[6]	MD1[7]
Mailbox 2	MD2[0]	MD2[1]	MD2[2]	MD2[3]	MD2[4]	MD2[5]	MD2[6]	MD2[7]
Mailbox 3	MD3[0]	MD3[1]	MD3[2]	MD3[3]	MD3[4]	MD3[5]	MD3[6]	MD3[7]

Figure 15.3 Message Data Configuration



Notes: 1. The TinyCAN is reset at any time when the RSTRQ bit in MCR is set to 1.
2. The PMR97 and PMR96 bits in TCMR should be set after Mailboxes and LAFM have b initialized. Then BCR1 and BVR0 should be set.
The TinyCAN starts communication with the CAN bus after BCR1 and BVR0 have bee
3. The RESET bit in GSR is a status flag that indicates CAN bus communication is possib after reset procedure. This bit is cleared to 0 when 23 clock cycles are elapsed after BC BCR1 have been set.
4. The TinyCAN receives messages when MBCR and TXPR are not set.
5. When MBCR and TXPR are set, the TinyCAN starts message transmission and carries out CAN bus arbitration. If an arbitration loss occurs, receive operation starts.

Figure 15.4 Reset Clearing Flowchart

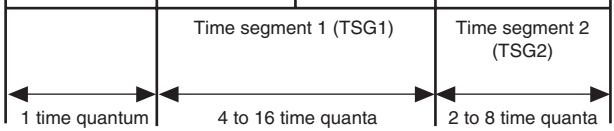


Figure 15.5 CAN Bit Configuration

The SYNC_SEG is a segment for establishing the synchronization of nodes on the CAN
Normal bit edge changes in this segment. The PRSEG is a segment for adjusting the phy
delay between networks. The PHSEG1 is a buffer segment for adjusting positive phase d
segment is extended when re-synchronization is established. The PHSEG2 is a buffer se
adjusting negative phase drift. This segment is shortened when re-synchronization is est

The range of settable values in BCR (TSG1, TSG2, BRP, and SJW) is shown in table 15

Table 15.2 Settable Values in BCR

Name	Abbreviation	Min. Value	Max. Valu
Time segment 1	TSG1*[1]	3*[3]	15
Time segment 2	TSG2*[1]	1*[4]	7
Baud rate prescaler	BRP	1	63
Re-Synchronization Jump width	SJW*[2]	0	3

Notes: 1. The time quanta values for the TSEG1 and TSEG2 are as follows: TSG value
2. In the CAN specifications, the Re-Synchronization Jump Width is stipulated a $4 \geq SJW \geq 1$. The value of SJW is given by adding 1 to the setting value of th SJW0 to SJW1 in BCR.
3. The minimum value of TSG1 is stipulated in the CAN specifications: TSG1 > TSG2
4. The minimum value of TSG2 is stipulated in the CAN specifications: $TSG2 \geq SJW$

Values that can be set for TSG1 and TSG2 in BCR1 are listed in table 15.3.

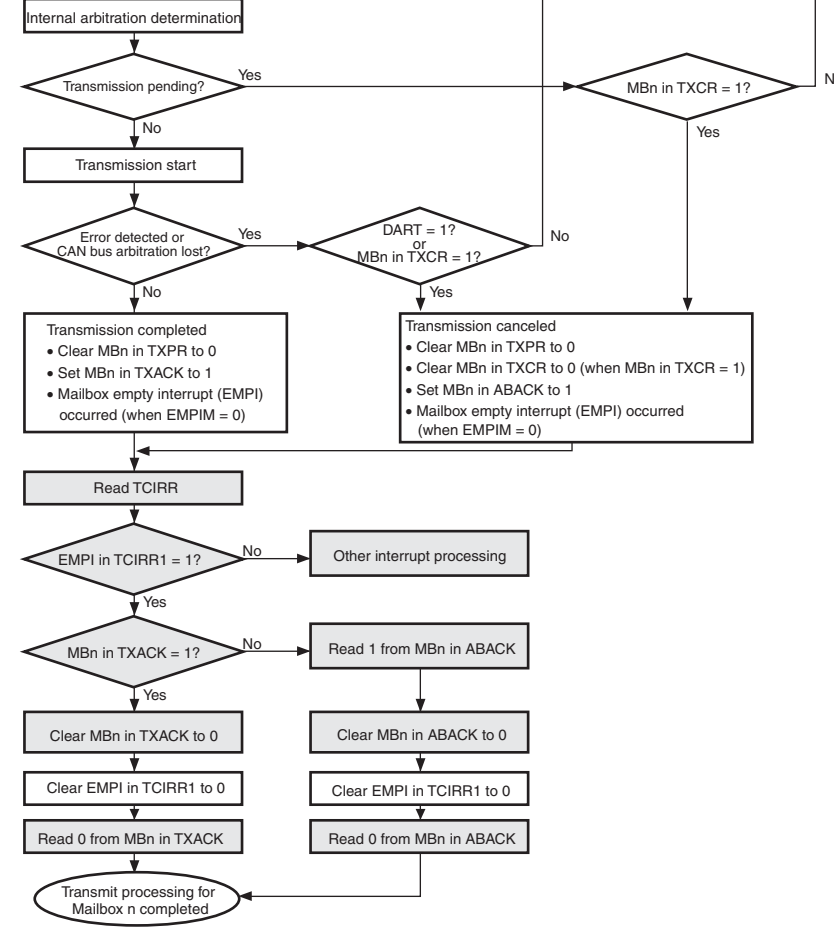
Table 15.3 Settable Values for TSG1 and TSG2 in BCR1

		TSG2					
		001	010	011	100	101	110
TSG1	0011	No	Yes	No	No	No	No
	0100	Yes	Yes	Yes	No	No	No
	0101	Yes	Yes	Yes	Yes	No	No
	0110	Yes	Yes	Yes	Yes	Yes	No
	0111	Yes	Yes	Yes	Yes	Yes	Yes
	1000	Yes	Yes	Yes	Yes	Yes	Yes
	1001	Yes	Yes	Yes	Yes	Yes	Yes
	1010	Yes	Yes	Yes	Yes	Yes	Yes
	1011	Yes	Yes	Yes	Yes	Yes	Yes
	1100	Yes	Yes	Yes	Yes	Yes	Yes
	1101	Yes	Yes	Yes	Yes	Yes	Yes
	1110	Yes	Yes	Yes	Yes	Yes	Yes
	1111	Yes	Yes	Yes	Yes	Yes	Yes

[Legend] Yes: Setting is possible
No: Setting is prohibited

[Example]

To have a baud rate of 1 Mbps with ϕ = 16 MHz, BRP = 1, and (1 + TSG1) + (1 + TSG2)
this case, the settings are BCR1 = H′23 and BCR0 = H'01.



Note: Processing in a shaded box requires setting by software.

Figure 15.6 Transmission Request Flowchart

1. Write data of a transmit message to MCn0, MCn4 to MCn7, and MDn0 to MDn7 [n =
before clearing the MBn bit in MBCR corresponding to the Mailbox of the transmit m
0 (initial setting).
2. Set the corresponding MBn bit in TXPR to 1 (start condition issuance). Then, the star
condition is generated.
3. The internal arbitration for message 1 is determined and the transmit message is trans
the temporary buffer. After that, even if a transmit request cancellation is issued to the
being transmitted by the DART or MBn bit in TXCR, message 1 is transmitted contin
unless the TinyCAN detects an arbitration loss or error on the CAN bus.
4. After the seventh bit of the EOF has been transmitted (normal message transmission e
MBn bits in TXPR and TXCR which are corresponding Mailbox are cleared to 0 and
bit in TXACK and the EMPI bit in TCIRR1 are set to 1. At this time, the MBn bit in
is always 0. Then, message transmission is completed.
5. When there is a transmit request other than for message 1, the transmit message is tra
to the temporary buffer and transmitted to the CAN bus after the arbitration for messa
been determined. When there is no transmit request other than for message 1, the Tiny
performs reception.

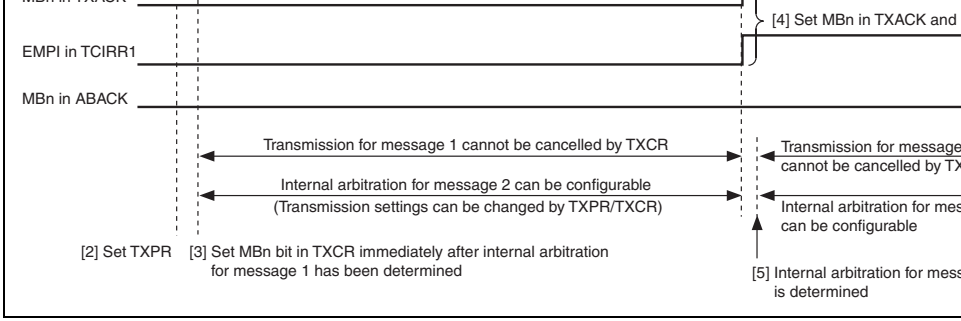


Figure 15.7 Internal Arbitration at Transmission Caused by TXCR/TXPR S

Arbitration Lost during Message Transmission: If an arbitration loss on the CAN bus
the TinyCAN halts transmission, and starts message reception. If the DART bit for the
transmission-requested message is cleared to 0, on the completion of reception, the trans
requested message is retransmitted. However, if the DART bit is set to 1, it is not transm
frame 2. Figures 15.8 to 15.10 show the timings of the arbitration loss on the CAN bus.
and operation are as follows.

1. Write data of a transmit message to MCn0, MCn4 to MCn7, and MDn0 to MDn7 [n
before clearing the MBn bit in MBCR corresponding to the Mailbox of the transmit
0 (initial setting).
2. Set the corresponding MBn bit in TXPR to 1 (start condition issuance). Then, the sta
condition is generated.
3. The internal arbitration for message 1 is determined and the transmit message is tran
the temporary buffer. After that, even if a transmit request cancellation is issued to th
being transmitted by the DART or MBn bit in TXCR, message 1 is transmitted conti
unless the TinyCAN detects an arbitration loss or error on the CAN bus.

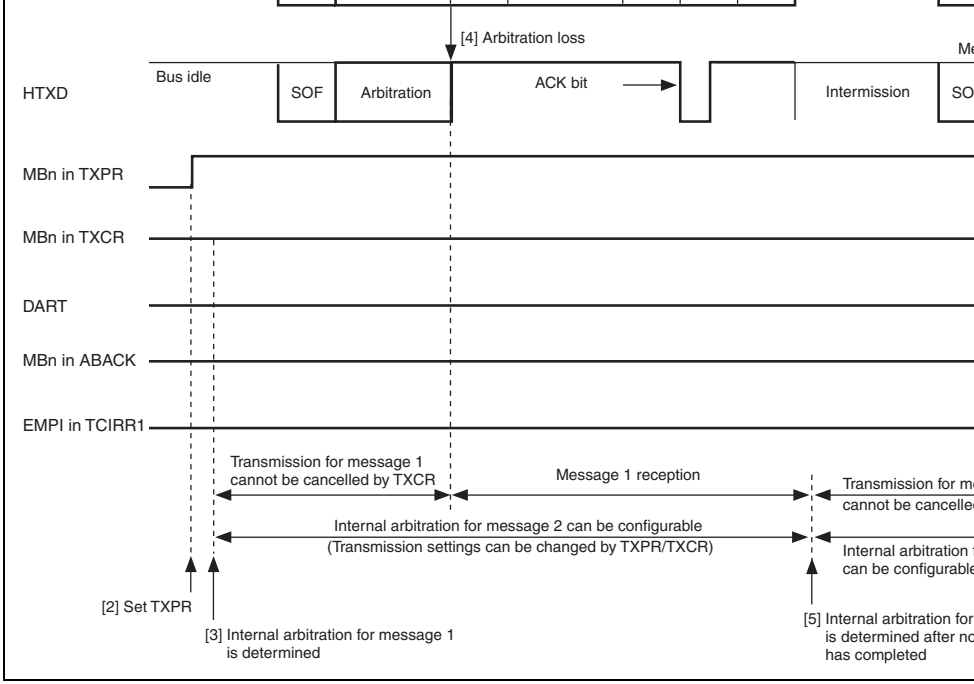


Figure 15.8 Internal Arbitration at Reception Caused by CAN Bus Arbitration (MBn in TXCR = 0 and DART = 0)

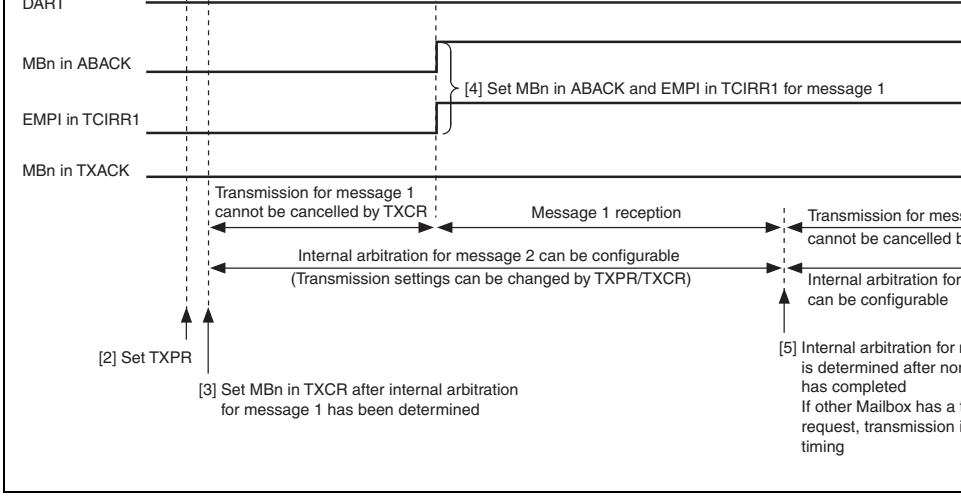


Figure 15.9 Internal Arbitration at Reception Caused by CAN Bus Arbitratio (MBn in TXCR = 1)

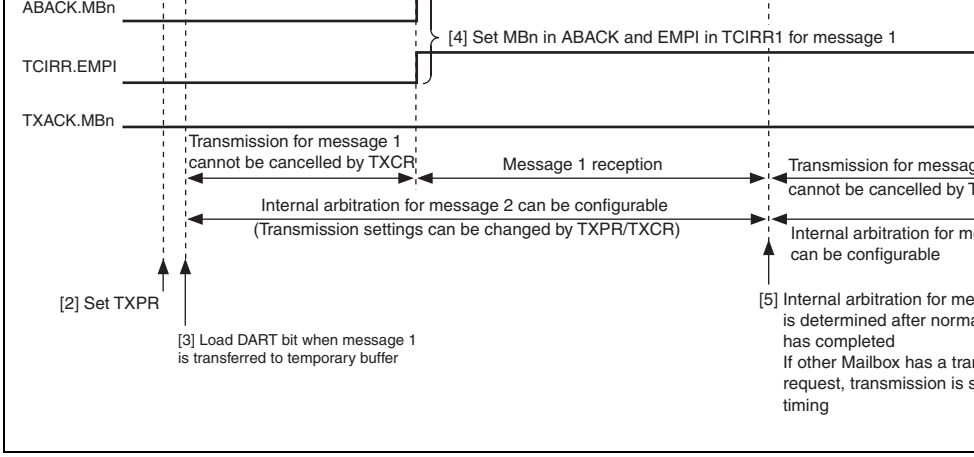


Figure 15.10 Internal Arbitration at Reception Caused by CAN Bus Arbitration (DART = 1)

CAN Bus Error: Figures 15.11 to 15.13 show timings for internal arbitration caused by on the CAN bus. Procedure and operation are as follows.

1. Write data of a transmit message to MCn0, MCn4 to MCn7, and MDn0 to MDn7 [n = before clearing the MBn bit in MBCR corresponding to the Mailbox of the transmit m 0 (initial setting).
2. Set the MBn bit in TXPR to 1 (start condition issuance). Then, the start condition is g
3. The internal arbitration for message 1 is determined and the transmit message is trans the temporary buffer. After that, even if a transmit request cancellation is issued to the being transmitted by the DART or MBn bit in TXCR, message 1 is transmitted contin unless the TinyCAN detects an arbitration loss or error on the CAN bus.

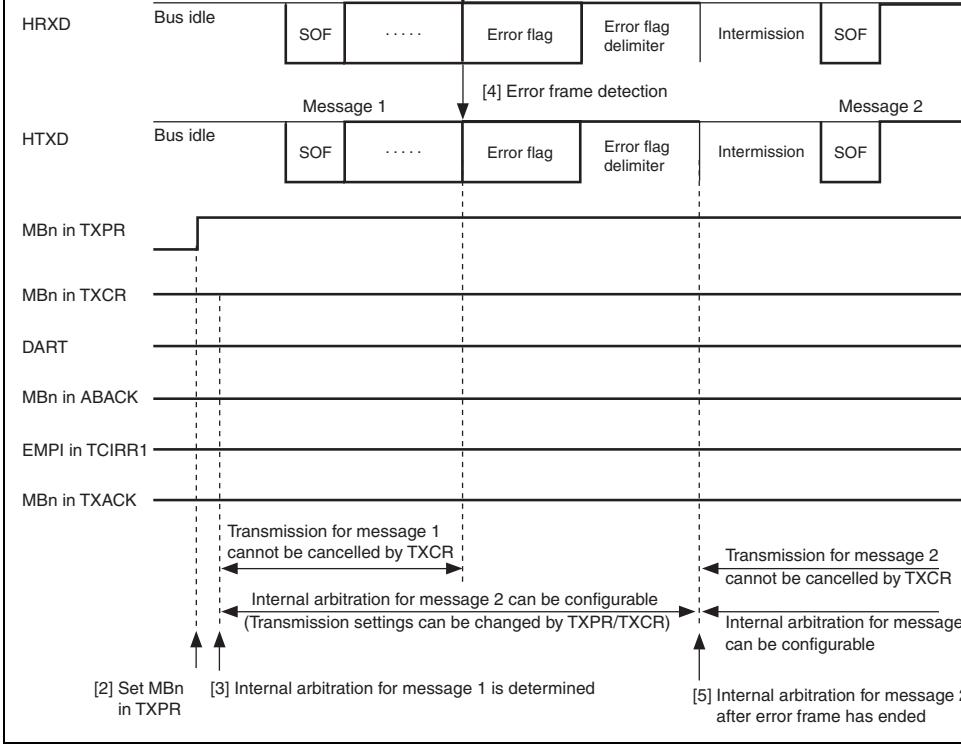


Figure 15.11 Internal Arbitration at Error Detection (MBn in TXCR = 0 and DA

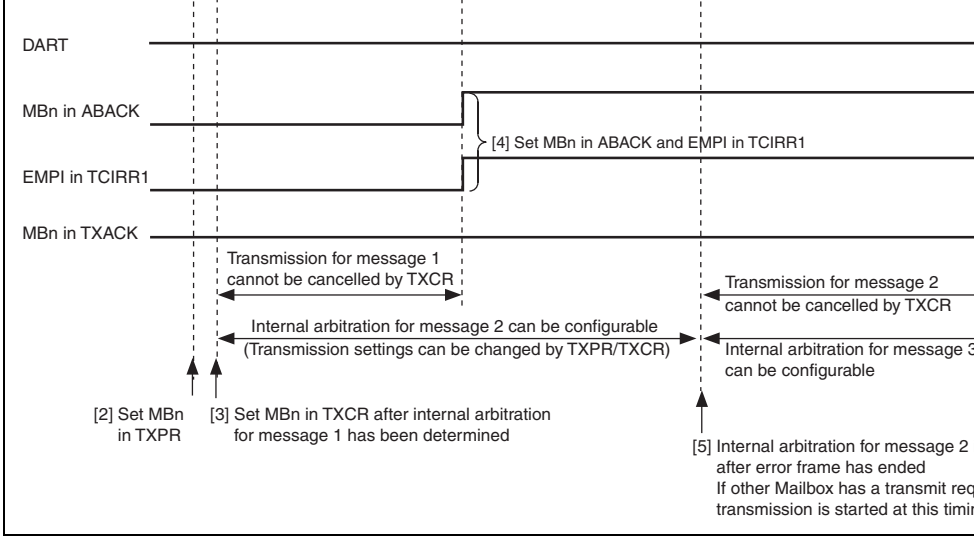


Figure 15.12 Internal Arbitration at Error Detection (MBn in TXCR = 1)

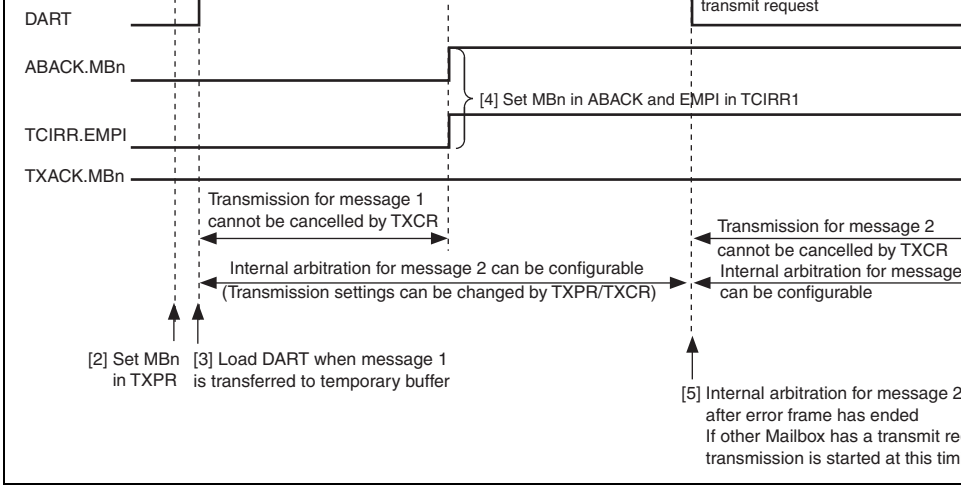


Figure 15.13 Internal Arbitration at Error Detection (DART = 1)

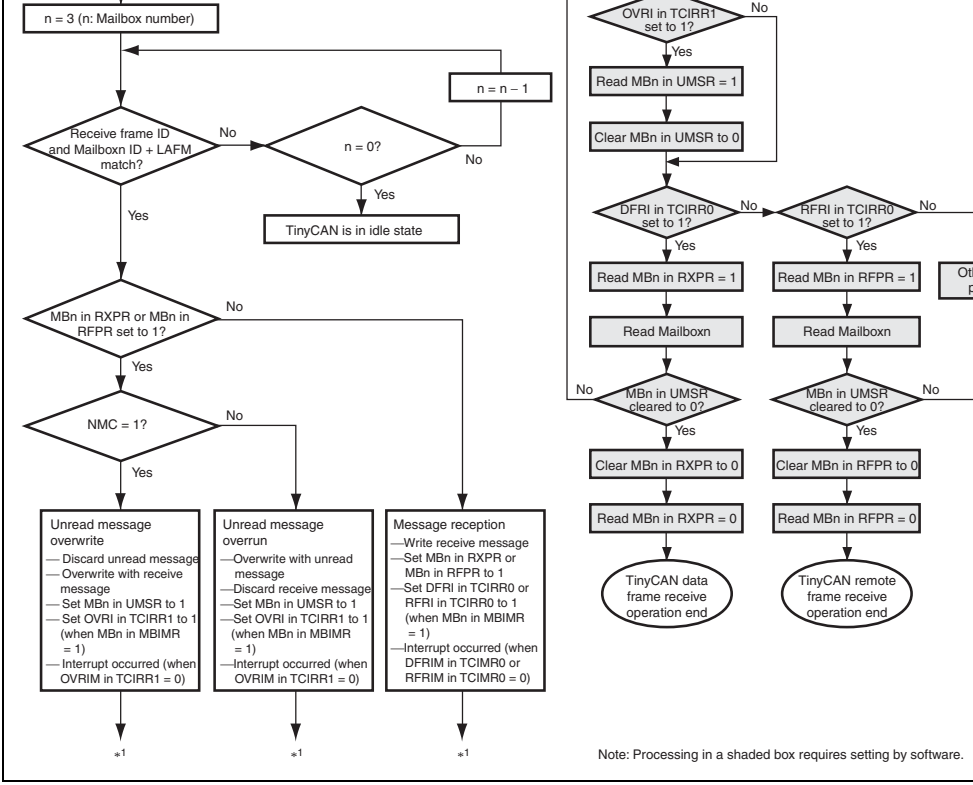


Figure 15.14 Message Reception Flowchart

the TinyCAN clears the temporary buffer and enters the idle state.

3. When the identifier has been compared at the seventh bit of the EOF or the higher bi message is written to the receive Mailbox whose identifier matches that of the receiv message. The identifier, which is masked by LAFM, may be overwritten. If there is one Mailbox whose ID and LAFM matches with those of the receive message, the M with the highest number is always to receive the relevant message. Note that Mailbo lower numbers cannot receive that message.
4. After the message has been written to the receive Mailbox, the DFRI bit in TCIRR0 MBn bit in RXPR are set to 1 when the receive message is a data frame, the RFRI bi TCIRR0 and the MBn bit in RFPR are set to 1 when the receive message is a remote and the OVRI bit in TCIRR1 and the MBn bit in UMSR are set to 1 when an overru overwrite occurs.
5. When the MBn bit in RXPR or the MBn bit in RFPR is set to 1, the temporary buffe cleared. When the TinyCAN transmits message 2, the internal arbitration is determin the transmit message is transferred to the temporary buffer and output to HTXD afte temporary buffer has been cleared.

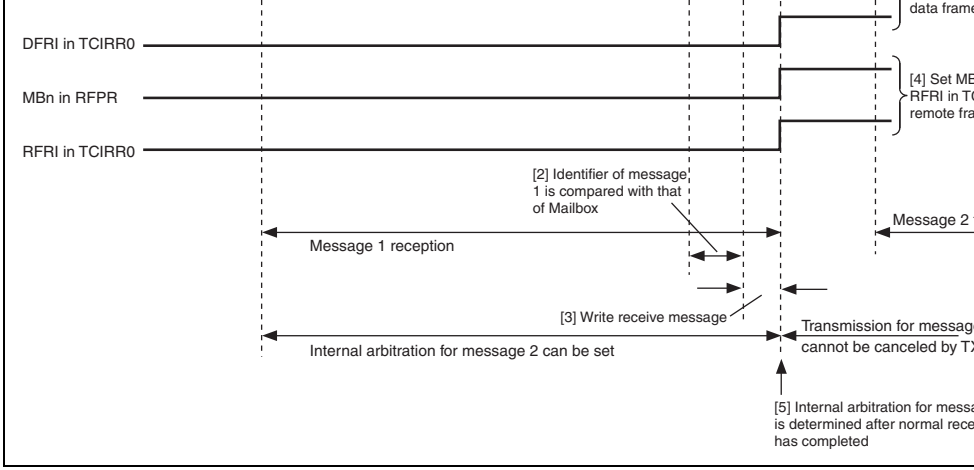


Figure 15.15 Set Timing for Message Reception

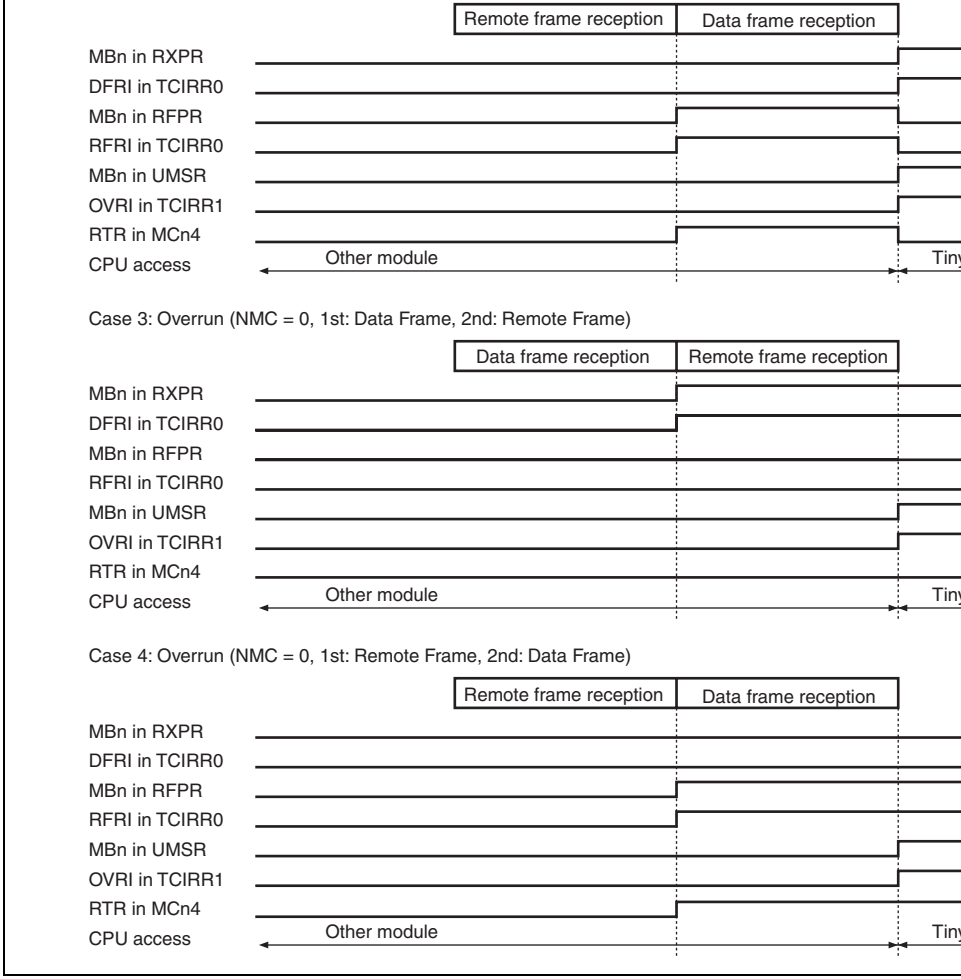
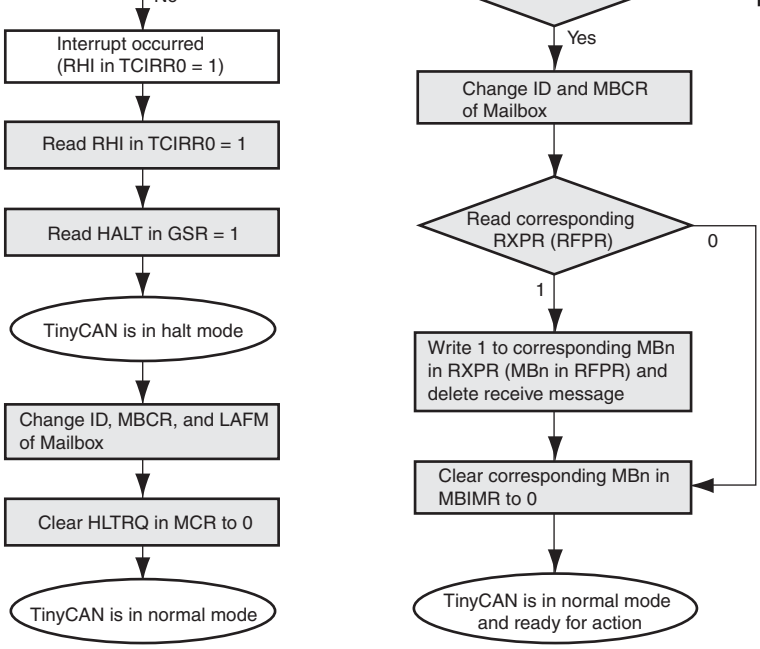


Figure 15.16 RXPR/RFPR Set/Clear Timing when Overrun/Overwrite Occ

1. Set the HLTRQ bit in MCR bit to 1.
2. Determine whether the TinyCAN is during transmission or reception, or in the bus off and wait for recovery from transmission, reception, or bus off state.
3. The TinyCAN enters halt mode at the first bit in the intermission frame of the messag sets the RHI bit in TCIRR0 and the HALT bit in GSR to 1. Note that the TinyCAN ca transmit or receive a message in halt mode.
4. Confirm that the RHI bit in TCIRR0 and the HALT bit in GSR are both set to 1 befor changing settings of the identifier, LAFM, and the MBn bit in MBCR of the Mailbox.
5. When the HLTRQ bit in MCR is cleared to 0, the TinyCAN returns to normal operati 11 recessive bits have been continuously received.

<Method 2: Other than halt mode (see figure 15.17)>

1. Set the MBn bit in MBIMR for the corresponding Mailbox to 1 to disable interrupts. (3)
2. Determine whether the MBn bits in RXPR and RFPR are cleared to 0 to confirm that no receive messages.
3. Change the settings of the identifier, LAFM, and the MBn bit in MBCR in the Mailbo
4. Determine whether the MBn bits in RXPR and RFPR are cleared to 0 to confirm that message is received during reconfiguration. The function of MBIMR is not to prevent RFPR, or the OVRI bit in TCIRR1 from being set.
5. At this time, when the MBn bit in RXPR or RFPR is set to 1, clear the relevant bit to the receive message because it cannot be determined whether the message was addres new Mailbox ID or the old Mailbox ID.
6. Then clear the MBn bit in MBIMR for the corresponding Mailbox to 0. The TinyCAN to normal operation.



Note: Processing in a shaded box requires setting by software.

Figure 15.17 Flowchart for Changing ID, MBCR, and LAFM of Receive Ma

4. Make this LSI enter standby mode or module standby mode. In module standby mode MSTTC bit in TCMR to 1. Then the TinyCAN enters module standby mode.

Making the CAN bus enter the bus idle state using this procedure will reduce power cons of this LSI. The TinyCAN registers retain their settings in standby mode.

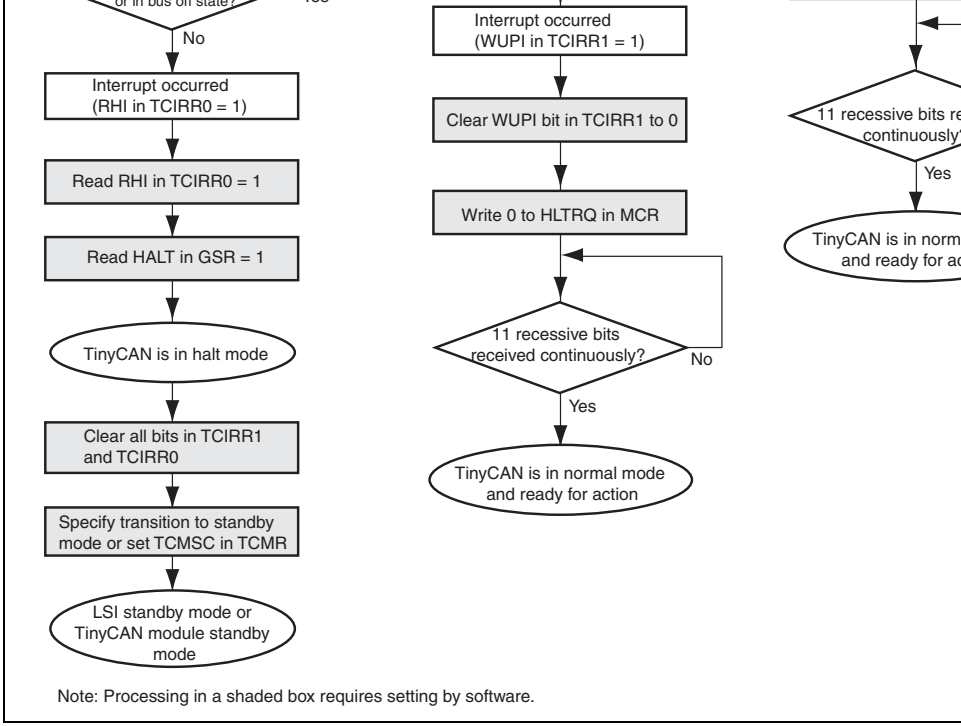
Transition from Standby Mode to Normal Operation: This LSI can make a transition standby mode to normal mode with the following procedure.

1. When data on the CAN bus changes from recessive to dominant, a falling edge is dete the HRXD pin.
2. This causes the WUPI bit in TCIRR1 to be set to 1, and generates an interrupt request
3. After an interrupt request is issued, the TinyCAN registers resume operation with the before entering standby mode. Change the settings at this timing if necessary.
4. To re-enable communication with the CAN bus, clear both the WUPI bit in TCIRR1 a HLTRQ bit in MCR to 0. After 11 recessive bits are consecutively received, commun will resume.

Note however that the first frame to be received cannot be received normally.

Transition from Module Standby Mode to Normal Operation: The TinyCAN can ma transition from module standby mode to normal mode with the following procedure.

1. When the MSTTC bit in TCMR is cleared to 0, the TinyCAN registers resume operat the settings before entering module standby mode. Change the settings at this timing i necessary.
2. To re-enable communication with the CAN bus, clear both the HLTRQ bit in MCR to 11 recessive bits are consecutively received, communication will resume.



Note: Processing in a shaded box requires setting by software.

Figure 15.18 Flowchart for Transition between Active Mode and Standby Mo
Module Standby Mode

		standby mode
Unread message	OVRI	When new message is received while M corresponding to receive message is 0 a RXPR or RFPR is 1
Mailbox empty	EMPI	When TXPR is cleared to 0 by completi transmission or completion of transmiss cancellation
Overload frame transmission	OVLI	When overload frame is transmitted
Bus off	BOFI	When TEC ≥ 256 or 11 bits are received times in bus off state
Error passive	EPI	When TEC ≥ 128 or REC ≥ 128
Receive overload warning	ROWI	When REC ≥ 96
Transmit overload warning	TOWI	When TEC ≥ 96
Remote frame request	RFRI	When remote frame is received and corresponding MBIMR is 0
Receive message	DFRI	When message reception is completed corresponding MBIMR is 0
Reset/Halt	RHI	When processing is completed after sof reset request (RSTRQ) or halt mode rec (HLTRQ) is issued

When TEC or REC becomes 128 after incrementing or decrementing, note that the error p (EPI) flag issues an interrupt request. When REC or TEC becomes 96 after incrementing decrementing, note that the receive overload warning (ROWI) flag or transmit overload w (TOWI) flag issues an interrupt request, respectively.

								(Initial
1	0	0	0	1	0	1	0	Recei mode
1	0	0	1	—	0	0	0	Extern mode
1	0	0	1	—	1	1	1	Interna mode
1	1	0	—	—	—	—	—	Count error-p
1	0	1	—	—	—	—	—	Force passiv

Normal Mode: The TinyCAN operates in normal mode.

Receive-Only Mode: This mode is used for running tests required by the ISO-11898 sta such as baud rate detection. Counting by the error counter is disabled and the TEC and R is not incremented. To prevent the TinyCAN from generating an error frame, transmissi HTXD output is disabled.

External Self-Test Mode: The TinyCAN generates its own acknowledge bit. The HRX HTXD pins must be connected to the CAN bus.

Internal Self-Test Mode: The TinyCAN generates its own acknowledge bit. Since the i is loop back to the internal Rx, the HRXD and HTXD pins need not be connected to the or another external device.

15.8 CAN Bus Interface

A bus transceiver IC and a pull-up resistor are necessary to connect this LSI to a CAN bu
Renesas Technology HA13721 transceiver IC is recommended. If any other product is us
confirm that it is compatible with the HA13721. Figure 15.19 shows a sample connection

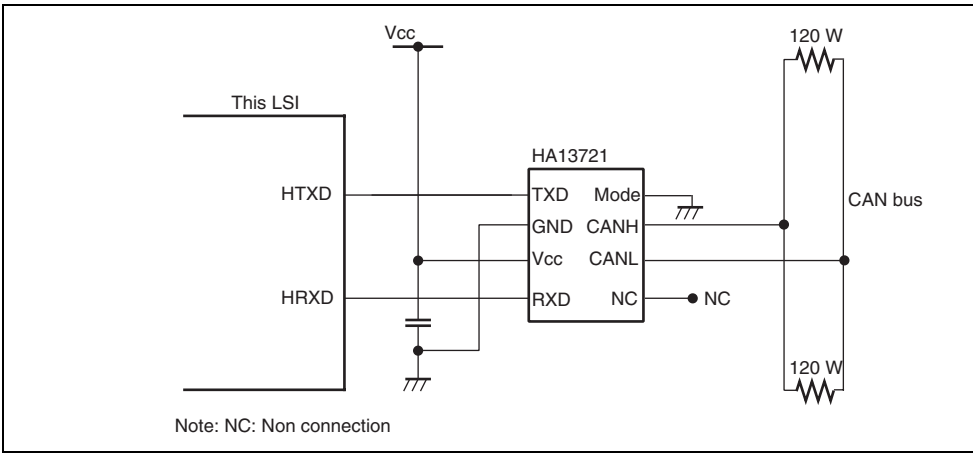


Figure 15.19 High-Speed CAN Bus Interface Using HA13721

RFPR are 0 after transition to halt mode.

5. If MCn0, MCn4 to MCn7, and MDn0 to MDn7 (n = 0 to 3) are written to by the CPU the seventh EOF bit and intermission space at message reception, note that data of th message may be overwritten.
6. If MCn0, MCn4 to MCn7, and MDn0 to MDn7 (n = 0 to 3) are written to by the CPU intermission spaces during a transmit request, note that the transmit message may be
7. RXPR and RFPR are exclusively set or cleared during overwrite.
8. A wakeup operation is enabled only in LSI standby mode. Note that it is disabled in standby mode.
9. To enter module standby mode, make a transition to halt mode beforehand. Otherwis clearing module standby mode, communication with the CAN bus resumes with the before making a transition and an error occurs.
10. When an error is detected during reception, the TinyCAN clears data in the temporar

mode (including bidirectional communication mode)
- Can be operated as a master or a slave device
- Choice of seven internal clocks (φ/256, φ/128, φ/64, φ/32, φ/16, φ/8, φ/4) and an exte as a clock source
- Clock polarity and phase of SSCK can be selected
- Choice of data transfer direction (MSB-first or LSB-first)
- Receive error detection: overrun error
- Multimaster error detection: conflict error
- Five interrupt sources: transmit-end, transmit-data-empty, receive-data-full, overrun conflict error

16.2 Continuous transmission and reception of serial data are ena since both transmitter and Input/Output Pins

Table 16.1 shows the pin configuration of the SSU.

Table 16.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
SSU clock	SSCK	I/O	SSU clock input/output
SSU data input/output	SSI	I/O	SSU data input/output
SSU data input/output	SSO	I/O	SSU data input/output
SSU chip select input/output	$\overline{\text{SCS}}$	I/O	SSU chip select input/output

- SS transmit data register (SSTDR)
- SS shift register (SSTRSR)

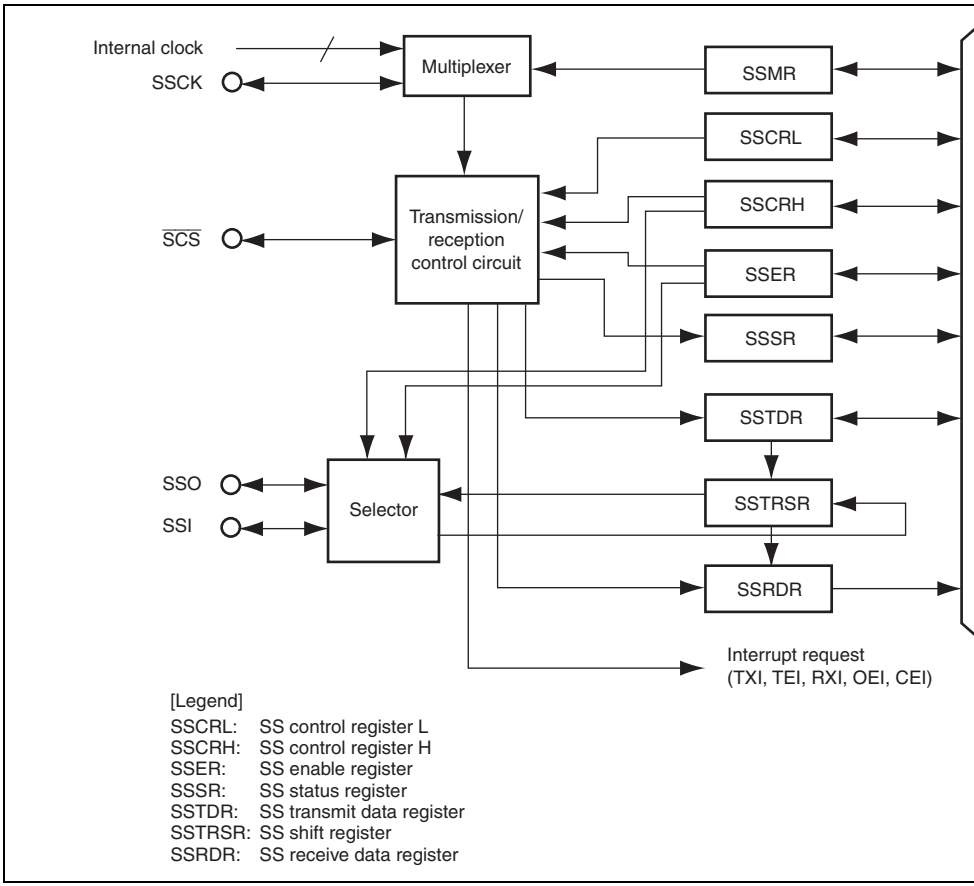


Figure 16.1 Block Diagram of SSU

				device, transfer clock is output from the SSCK p the CE bit in SSSR is set, this bit is automatical 0: Operates as a slave device 1: Operates as a master device
6	BIDE	0	R/W	Bidirectional Mode Enable Selects whether the serial data input pin and th pin are both used or only one pin is used. For d refer to section 16.4.3, Relationship between D Input/Output Pin and Shift Register. When the S in SSCRL is 0, this setting is invalid. 0: Normal mode. Communication is performed two pins. 1: Bidirectional mode. Communication is perfor using only one pin.
5	SOOS	0	R/W	Serial Data Open-Drain Output Select Selects whether the serial data output pin is CM output or NMOS open-drain output. The serial d pin is changed according to the register setting details, refer to section 16.4.3, Relationship bet Data Input/Output Pin and Shift Register. 0: CMOS output 1: NMOS open-drain output

				0: Shows serial data output level to low in readin Changes serial data output level to low in writi 1: Shows serial data output level to high in readi Changes serial data output level to high in wri
3	SOLP	1	R/W	SOL Write Protect When output level of serial data is changed, the instruction is used to set the SOL bit to 1 and cle to 0 or to clear the SOL bit and this bit to 0. 0: In writing, output level can be changed accord value of the SOL bit. 1: In reading, this bit is always read as 1. In writi cannot be modified output level.
2	SCKS	0	R/W	SSCK Pin Select Selects whether the SSCK pin functions as a po serial clock pin. 0: Functions as a port 1: Functions as a serial clock pin
1	CSS1	0	R/W	$\overline{\text{SCS}}$ Pin Select
0	CSS0	0	R/W	Selects whether the $\overline{\text{SCS}}$ pin functions as a port, input, or $\overline{\text{SCS}}$ output. When the SSUMS bit in SS 0, the $\overline{\text{SCS}}$ pin functions as a port regardless of setting of this bit. 00: Functions as a port 01: Functions as an $\overline{\text{SCS}}$ input 1X: Functions as an $\overline{\text{SCS}}$ output (however, funct an $\overline{\text{SCS}}$ input before starting transfer)

[Legend]

X: Don't care.

6	SSUMS	0	R/W	SSU Mode Select Selects which combination of the serial data inp serial data output pin is used. For details, refer to section 16.4.3, Relationship Data Input/Output Pin and Shift Register. 0: Clocked synchronous communication mode Data input: SSI pin, Data output: SSO pin 1: Four-line bus communication mode When MSS = 1 and BIDE = 0 in SSCRH: Data input: SSI pin, Data output: SSO pin When MSS = 0 and BIDE = 0 in SSCRH: Data input: SSO pin, Data output: SSI pin When BIDE = 1 in SSCRH: Data input and output: SSO pin
5	SRES	0	R/W	Software reset When this bit is set to 1, the SSU internal seque forcibly reset. Then this bit is automatically clea register value in the SSU is retained.
4	SCKOS	0	R/W	SSCK Pin Open-Drain Output Select Selects whether the SSCK pin functions as CM or NMOS open-drain output. 0: CMOS output 1: NMOS open-drain output

16.3.3 SS Mode Register (SSMR)

SSMR is a register that selects MSB-first or LSB-first, clock polarity, clock phase, and tr
clock rate.

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select Selects whether data transfer is performed in MS LSB-first. • 0: LSB-first • 1: MSB-first
6	CPOS	0	R/W	Clock Polarity Select Selects the clock polarity of SSCK. 0: Idle state = high 1: Idle state = low
5	CPHS	0	R/W	Clock Phase Select Selects the clock phase of SSCK. 0: Data change at first edge 1: Data latch at first edge
4, 3	—	All 0	—	Reserved These bits are always read as 0.

				101: ϕ/8 110: ϕ/4 111: Reserved

16.3.4 SS Enable Register (SSER)

SSER is a register that sets transmit enable, receive enable, and interrupt enable.

Bit	Bit Name	Initial Value	R/W	Description
7	TE	0	R/W	Transmit enable When this bit is 1, transmit operation is enabled
6	RE	0	R/W	Receive enable When this bit is 1, receive operation is enabled.
5	RSSTP	0	R/W	Receive single stop When this bit is 1, receive operation is complete receiving one byte.
4	—	0	—	Reserved This bit is always read as 0.
3	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, a TEI interrupt request enabled.
2	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, a TXI interrupt request enabled.

SSSR is a register that sets interrupt flags.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.
6	ORER	0	R/W	Overrun Error Flag Indicates that the RDRF bit is abnormally termina reception because an overrun error has occurred SSRDR retains received data before the overrun occurs and the received data after the overrun e occurs is lost. When this bit is set to 1, subseque reception cannot be continued. When the MSS b SSCRH is 1, this is also applied to serial transmi [Setting condition] • When the next serial reception is completed RDRF = 1 [Clearing condition] • When 0 is written to this bit after reading 1
5, 4	—	All 0	—	Reserved These bits are always read as 0.

				[Setting conditions] • When the TE bit in SSER is 0 • When data transfer is performed from SSTD SSTRSR and data can be written in SSTDR [Clearing conditions] • When 0 is written to this bit after reading 1 • When data is written in SSTDR
1	RDRF	0	R/W	Receive Data Register Full [Setting condition] • When serial reception is completed normall receive data is transferred from SSTRSR to [Clearing conditions] • When 0 is written to this bit after reading 1 • When data is read from SSRDR
0	CE	0	R/W	Conflict Error Flag [Setting conditions] • When serial communication is started while 1 and MSS =1, the $\overline{\text{SCS}}$ pin input is low • When the $\overline{\text{SCS}}$ pin level changes from low t during transfer while SSUMS = 1 and MSS [Clearing condition] • When 0 is written to this bit after reading 1

SSTDR is an 8-bit register that stores serial data for transmission. SSTDR can be read or
to by the CPU at all times. When the SSU detects that SSTRSR is empty, it transfers the t
data written in SSTDR to SSTRSR and starts serial transmission. If the next transmit data
already been written to SSTDR during serial transmission, continuous serial transmission
possible. SSTDR is initialized to H′00.

16.3.8 SS Shift Register (SSTRSR)

SSTRSR is a shift register that transmits and receives serial data. When transmit data is tr
from SSTDR to SSTRSR, bit 0 in SSTDR is transferred to bit 0 in SSTRSR while the MI
SSMR is 0 (LSB-first transfer) and bit 7 in SSTDR is transferred to bit 0 in SSTRSR whi
MLS bit in SSMR is 1 (MSB-first transfer). SSTRSR cannot be directly accessed by the C

16.4.2 Relationship between Clock Polarity and Phase, and Data

Relationship between clock polarity and phase, and transfer data changes according to a combination of the SSUMS bit in SSCRL and the CPOS and CPHS bits in SSMR. Figu shows the relationship.

MSB-first transfer or LSB first transfer can be selected by the setting of the MLS bit in S When the MLS bit is 0, transfer is started from LSB to MSB. When the MLS bit is 1, tra started from MSB to LSB.

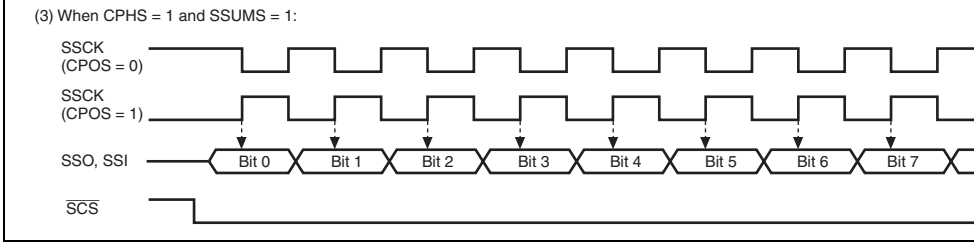


Figure 16.2 Relationship between Clock Polarity and Phase, and Data

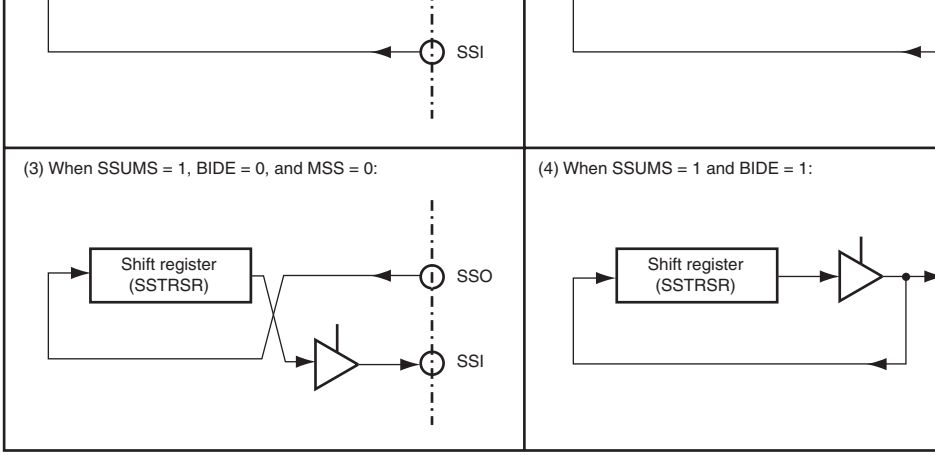


Figure 16.3 Relationship between Data Input/Output Pin and Shift Regist

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

[Legend]

—: Can be used as a general I/O port.

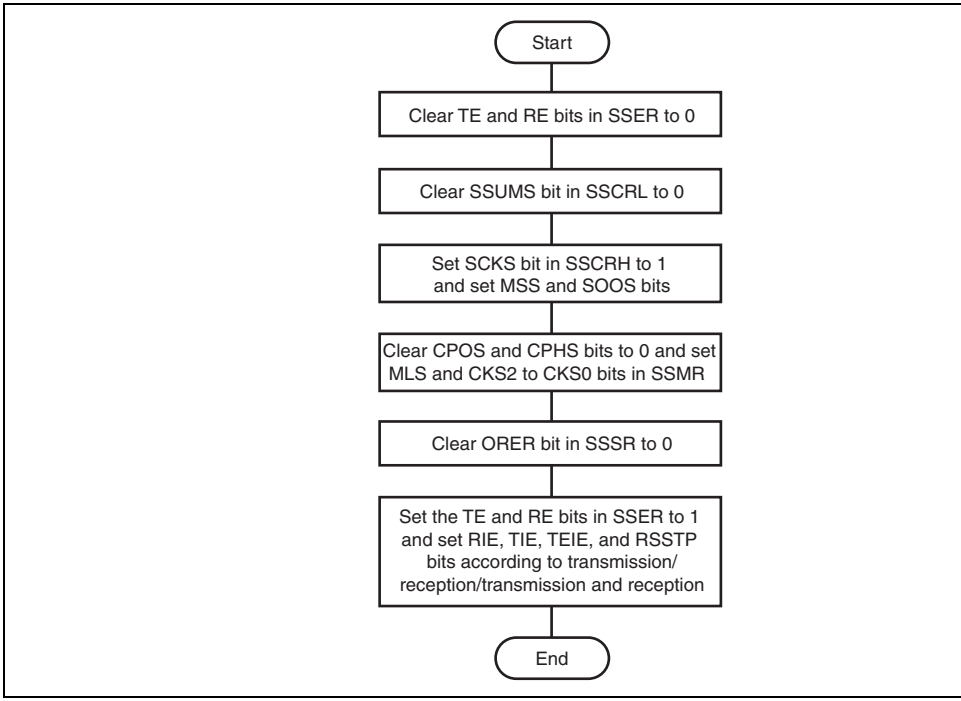


Figure 16.4 Initialization in Clocked Synchronous Communication Mod

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

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

Figure 16.6 shows a sample flowchart for serial data transmission.

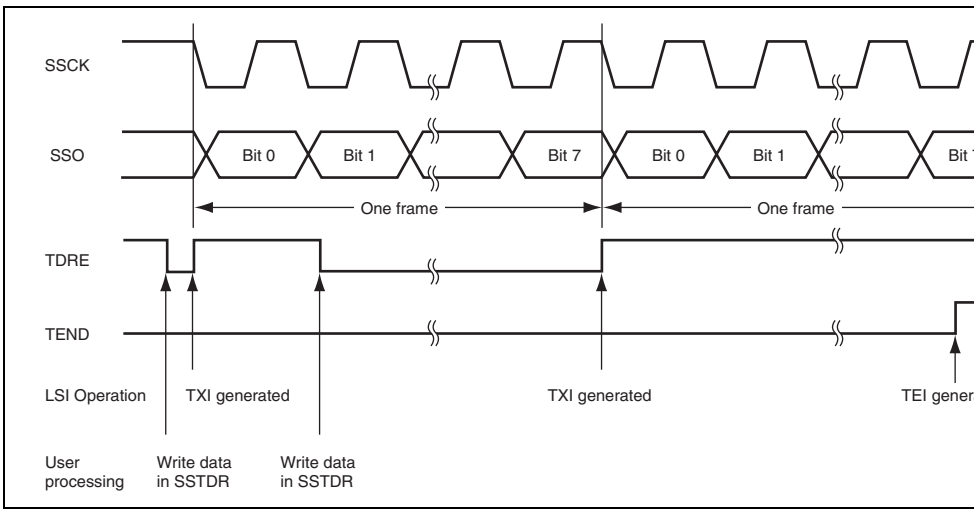


Figure 16.5 Example of Operation in Data Transmission

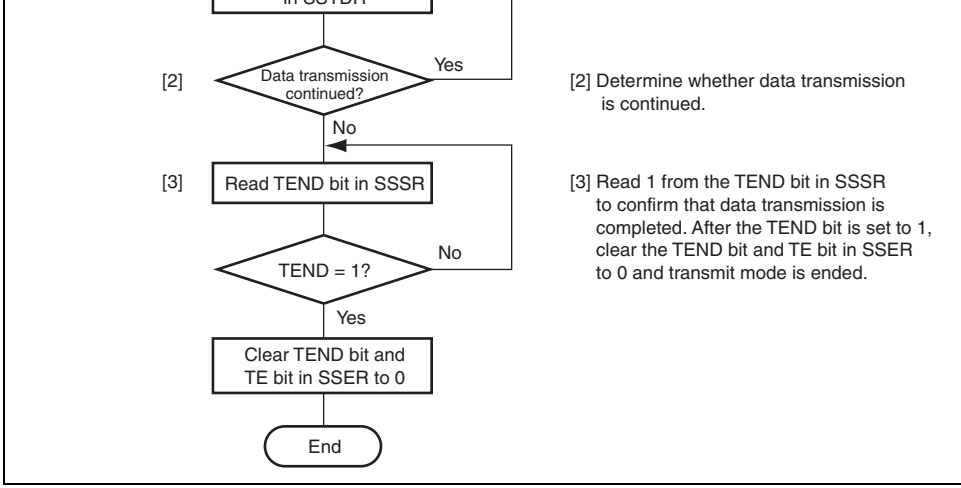


Figure 16.6 Sample Serial Transmission Flowchart

When the SSU is set as a master device and reception is ended, received data is read after the RSSTP bit in SSER to 1. Then the SSU outputs eight bits of clocks and operation is st After that, the RE and RSSTP bits are cleared to 0 and the last received data is read. Note SSRDR is read while the RE bit is set to 1, received clock is output again.

When the eighth clock rises while the RDRF bit is 1, the ORER bit in SSSR is set. Then a overrun error (OEI) is generated and operation is stopped. When the ORER bit in SSSR i reception cannot be performed. Therefore confirm that the ORER bit is cleared to 0 befor reception.

Figure 16.8 shows a sample flowchart for serial data reception.

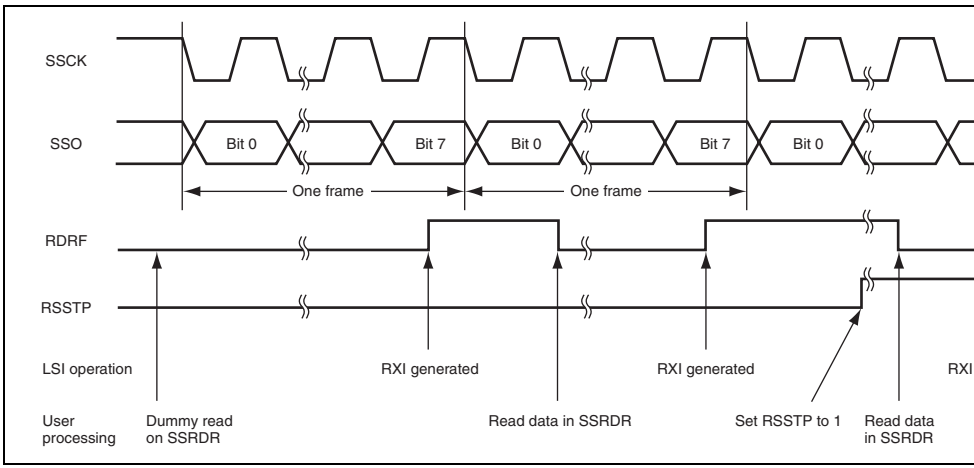
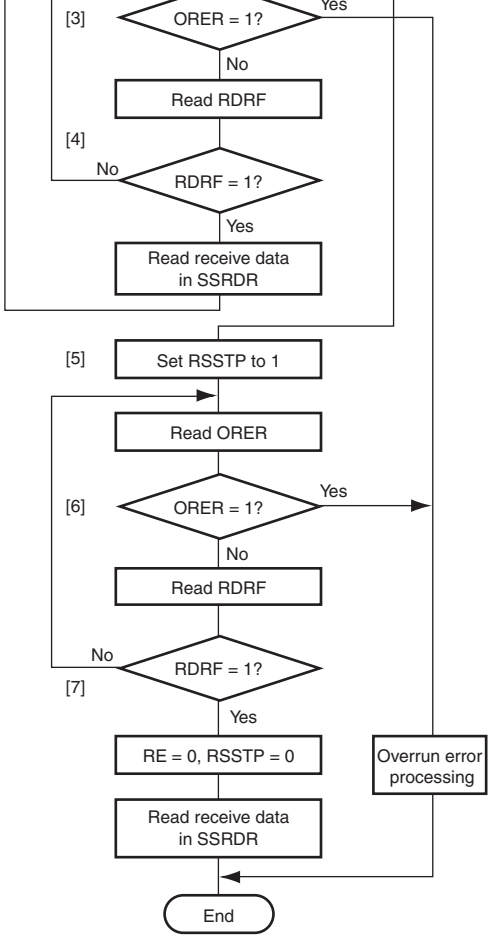


Figure 16.7 Example of Operation in Data Reception (MSS = 1)



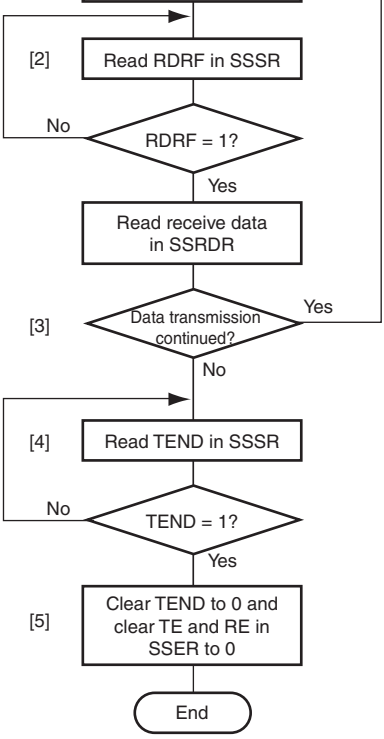
[3][6] When a receive error occurs, clear the ORER flag to 0 after the ORER flag in SSSR is read and an appropriate error processing is performed. When the ORER flag is set to 1, transmission/reception cannot be started again.

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

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

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

Figure 16.8 Sample Serial Reception Flowchart (MSS = 1)



[2] Confirm that the RDRF bit is 1. If the RDRF bit is 1, receive data in SSRDR is read. If the SSRDR bit is read, the RDRF bit is automatical cleared.

[3] Determine whether data transmission is continu

[4] Data transfer completion can be confirmed to see if TEND in SSSR becomes 1.

[5] To end transmit and receive mode, clear the TEND bit to 0 and clear the TE and RE bits in SSER to 0.

Figure 16.9 Sample Flowchart for Serial Transmit and Receive Operation

When the SSU is set as a master device, the chip select line controls output. When the SS
as a slave device, the chip select line controls input. When the SSU is set as a master devi
chip select line controls output of the $\overline{\text{SCS}}$ pin or controls output of a general port by setti
CSS1 bit in SSCRH to 1. When the SSU is set as a slave device, the chip select line sets t
pin as an input pin by setting the CSS1 and CSS0 bits in SSCRH to 01.

In four-line bus communication mode, the MLS bit in SSMR is set to 1 and transfer is pe
in MSB-first order.

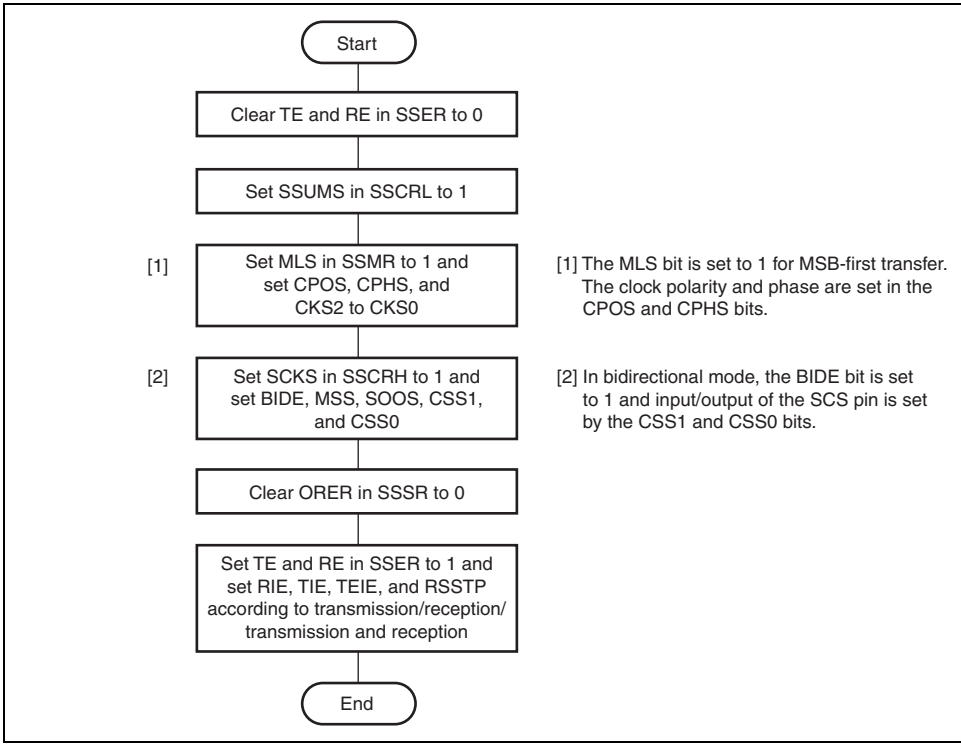


Figure 16.10 Initialization in Four-Line Bus Communication Mode

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

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

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

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

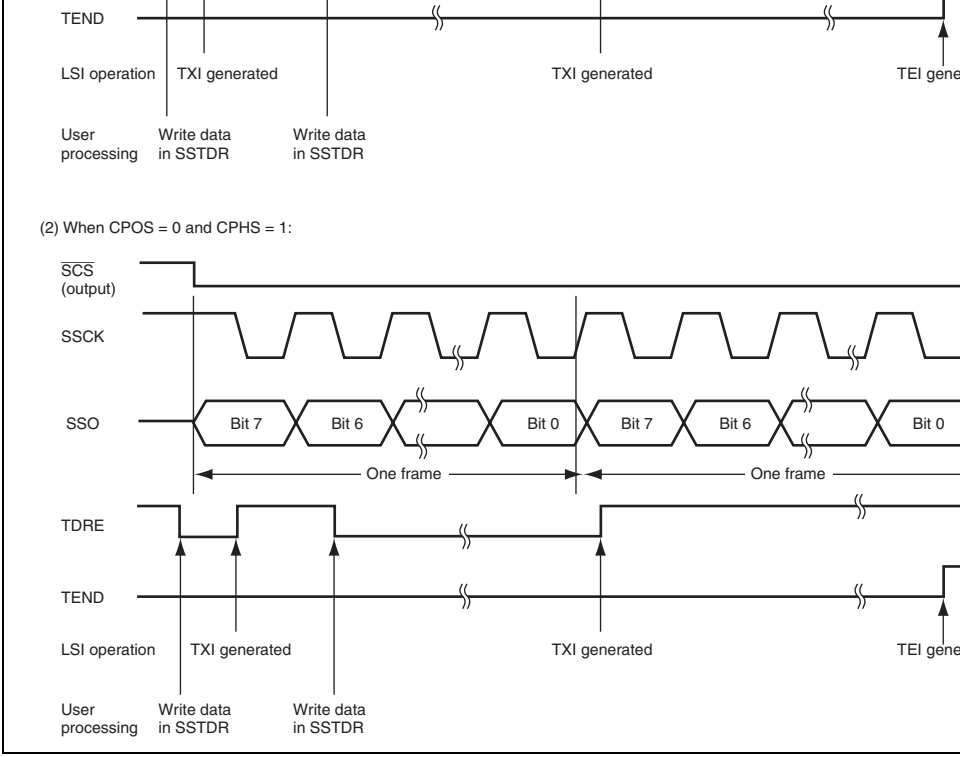


Figure 16.11 Example of Operation in Data Transmission (MSS = 1)

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

When the SSU is set as a master device and reception is ended, received data is read after
the RSSTP bit in SSER to 1. Then the SSU outputs eight bits of clocks and operation is st
After that, the RE and RSSTP bits are cleared to 0 and the last received data is read. Note
SSRDR is read while the RE bit is set to 1, received clock is output again.

When the eighth clock rises while the RDRF bit is 1, the ORER bit in SSSR is set. Then a
overrun error (OEI) is generated and operation is stopped. When the ORER bit in SSSR i
reception cannot be performed. Therefore confirm that the ORER bit is cleared to 0 befor
reception.

The set timings of the RDRF and ORER flags differ according to the CPHS setting. Thes
are shown in figure 16.2. When the CPHS bit is set to 1, the flag is set during the frame. T
care should be taken at the end of reception.

The sample flowchart for serial data reception is the same as that in clocked synchronous
communication mode.

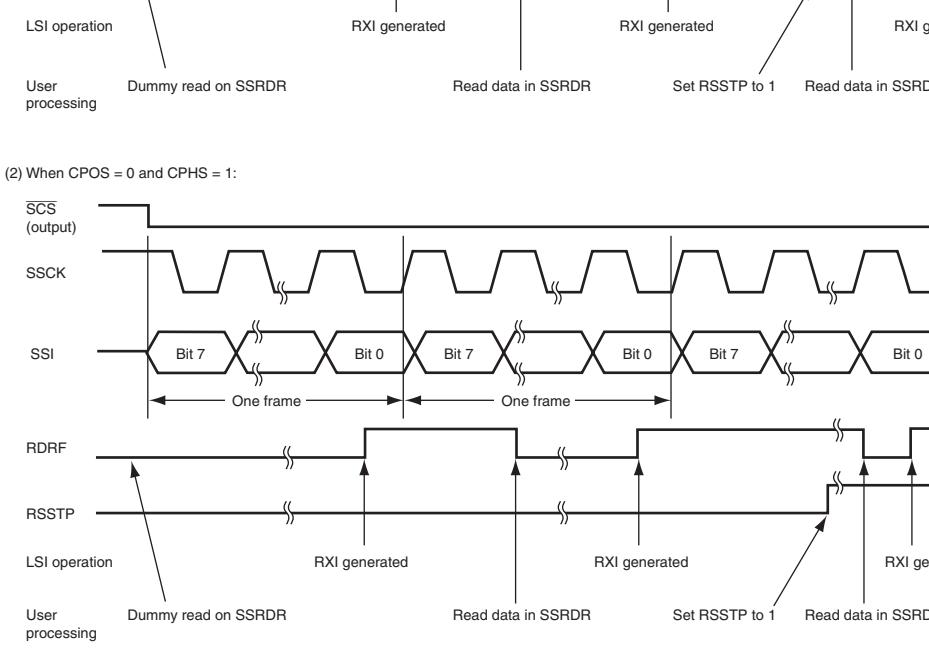


Figure 16.12 Example of Operation in Data Reception (MSS = 1)

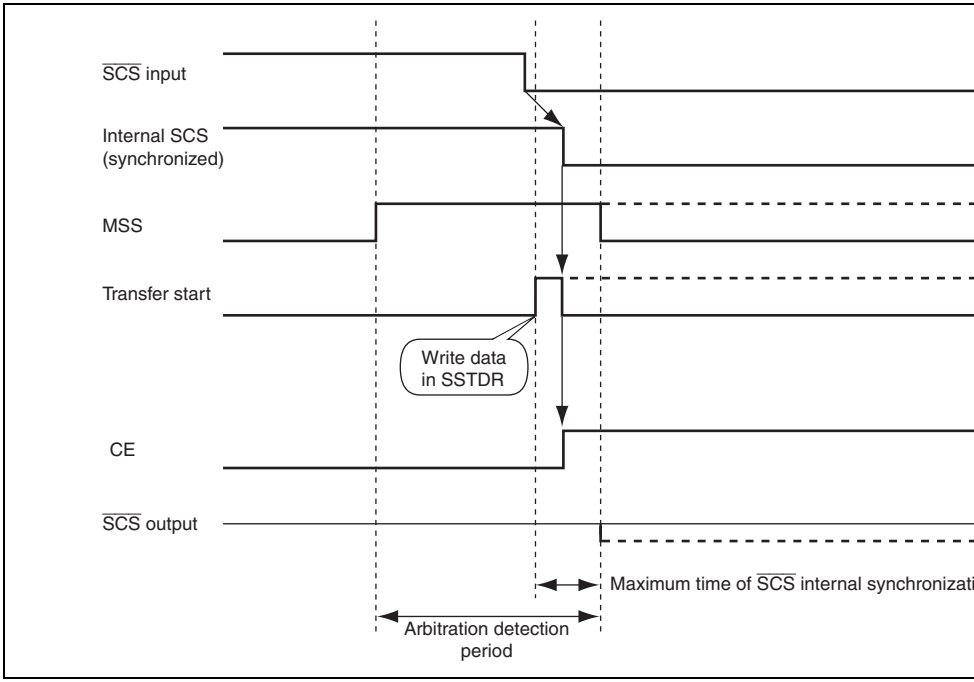


Figure 16.13 Arbitration Check Timing

Receive data full	RXI	(RIE = 1), (RDRF = 1)
Overrun error	OEI	(RIE = 1), (ORER = 1)
Conflict error	CEI	(CEIE = 1), (CE = 1)

When an interrupt condition shown in table 16.3 is 1 and the I bit in CCR is 0, the CPU
the interrupt exception handling. Each interrupt source must be cleared during the excep
handling. Note that the TDRE and TEND bits are automatically cleared by writing trans
SSTDR and the RDRF bit is automatically cleared by reading SSRDR. When transmit d
written in SSTDR, the TDRE bit is set again at the same time. Then if the TDRE bit is c
additional one byte of data may be transmitted.

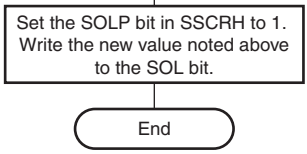


Figure 16.14 Procedures when Changing Output Level of Serial Data

Oscillation frequency: 64 kHz to 850 kHz

Temperature characteristic: Source clock ± 10% (typ.)

- Counter: two

  8-bit readable/writable down counter

  8-bit counter for measuring oscillation frequency of the on-chip oscillator
- CPU interrupt source

  Underflow (interrupt interval: 731 µsec to 67.4 msec)
- Subtimer clock supply operating modes:

  Subactive mode

  Subsleep mode
- On-chip oscillator

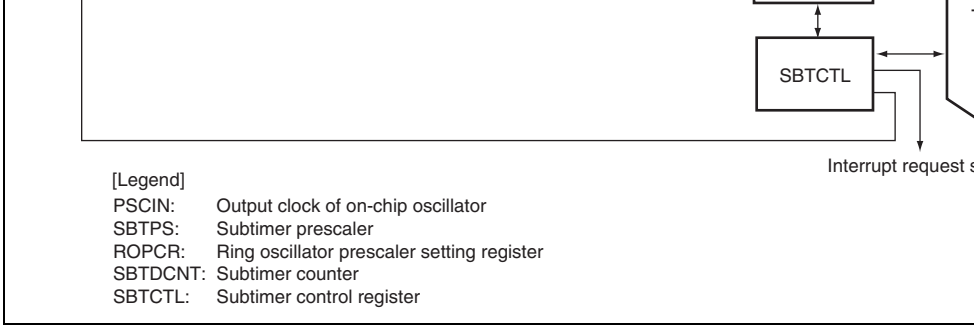
  The on-chip oscillator supplies three kinds of clocks:

  Subactive or subsleep mode (ϕ_w)

  Subtimer down counter (input clock)

  Watchdog timer (input clock)
- Subtimer prescaler (SBTPS)

  The subtimer prescaler is a divider which controls input clocks to the counter which oscillation cycle of the on-chip oscillator and the down counter for the subtimer.



[Legend]
PSCIN: Output clock of on-chip oscillator
SBTPS: Subtimer prescaler
ROPCR: Ring oscillator prescaler setting register
SBTDCNT: Subtimer counter
SBTCTL: Subtimer control register

Figure 17.1 Block Diagram of Subtimer

indicates the operating state. SBTCTL is initialized to H'60.

Bit	Bit Name	Initial Value	R/W	Description
7	PCEF	0	R/W	Division Count End Flag [Setting condition] When counting starts at the first falling edge an halts at the third falling edge after the on-chip o starts oscillation. [Clearing condition] When 0 is written to this bit after reading 1
6, 5	—	All 1	—	Reserved These bits are always read as 1.
4	START	0	R/W	Count Down Start Starts or halts the SBTDCNT operation. 0: SBTDCNT halts counting down. 1: SBTDCNT starts counting down.
3	OSCEB	0	R/W	On-Chip Oscillator Oscillation Enable Enables or disables the oscillation of the on-chi oscillator. 0: Oscillation of on-chip oscillator is disabled. 1: Oscillation of on-chip oscillator is enabled.

0	SBTUF	0	R/W	Underflow Interrupt Flag [Setting condition] When the SBTDCNT value underflows [Clearing condition] When 0 is written to this bit after reading 1

17.2.2 Subtimer Counter (SBTDCNT)

SBTDCNT is an 8-bit readable/writable down counter. When SBTDCNT underflows, an
request is issued and the SBTUF flag in SBTCTL is set to 1. SBTDCNT is initialized to H

17.2.3 Ring Oscillator Prescaler Setting Register (ROPCR)

ROPCR is an 8-bit readable/writable register. When the OSCEB bit in SBTCTL is set to
SBTPS counts two system clock cycles from the first falling edge of the on-chip oscillato
third falling edge and then transfers the count value to ROPCR. After that, ROPCR confi
division ratio of subclock. ROPCR is initialized to H'FF.

2. SBTPS halts counting at the third falling edge of PSCIN, the PCEF flag in SBTCTL
and then the SBTPS value is transferred to ROPCR.
3. By using this count value, the division ratio of the on-chip oscillator is determined a
value is set in ROPCR.
4. SBTPS starts supplying clocks and SBTDCNT starts counting down by clearing the
in SBTCTL to 0.

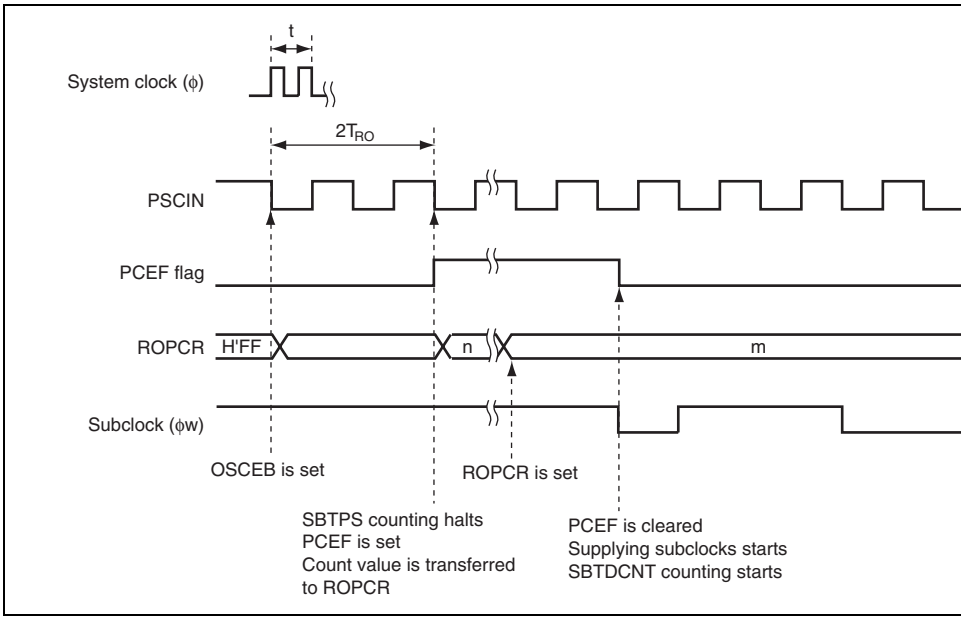


Figure 17.2 Timing for On-Chip Oscillator

Then the setting value m in ROPCR can be obtained by assigning the value k obtained by
(2) to formula (3).

$$m = \frac{T_{SUB}}{t \times n} - 2 \quad \text{(Note that } m \geq 0\text{)}$$ Formula (

The cycle to be actually used as the subclock is as follows.

$$T_{CAL} = 2\ (m + 2) \times T_{RO}$$ Formula (

Therefore, the rounding error between the expected value and the setting value of the sub
cycle can be obtained by the following formula.

$$\sigma = \frac{|\ T_{SUB} - T_{CAL}\ |}{T_{SUB}} \times 100\ (\%) = \left| 1 - \frac{K \times t \times n}{2 \times T_{SUB}} \right| \times 100\ (\%)$$ Formula (

[Legend]

t: System clock cycle
n: SBTPS count value (for two cycles)
T_{RO}: On-chip oscillator cycle (calculated value)
T_{CAL}: Subclock cycle (calculated value)
T_{SUB}: Subclock setting cycle (expected value)
k: Division ratio for oscillation cycle of on-chip oscillator and subclock setting cycle
m: ROPCR setting value

Rounding error of division ratio σ	—	+1.0 %	—
Rounding error of division ratio σ + count error	–2.0 %	—	+1.0 %

After deciding the division ratio according to formulas (1) to (3), the division ratio is configured in ROPCR. After ROPCR divides clocks of the on-chip oscillator, clocks subtimer counter, input clocks to the system, and input clocks to the watchdog timer generated.

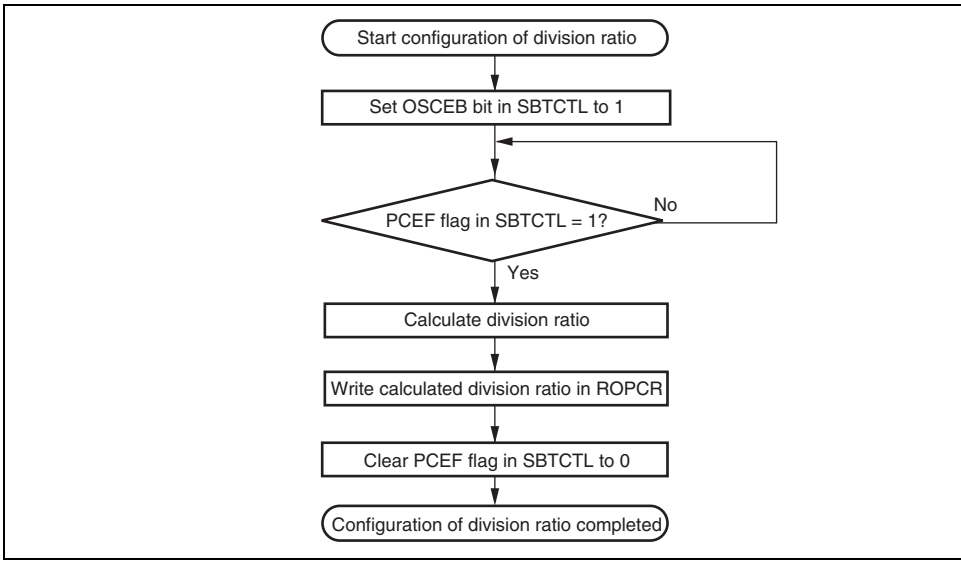


Figure 17.3 SBTPS Setting Flowchart

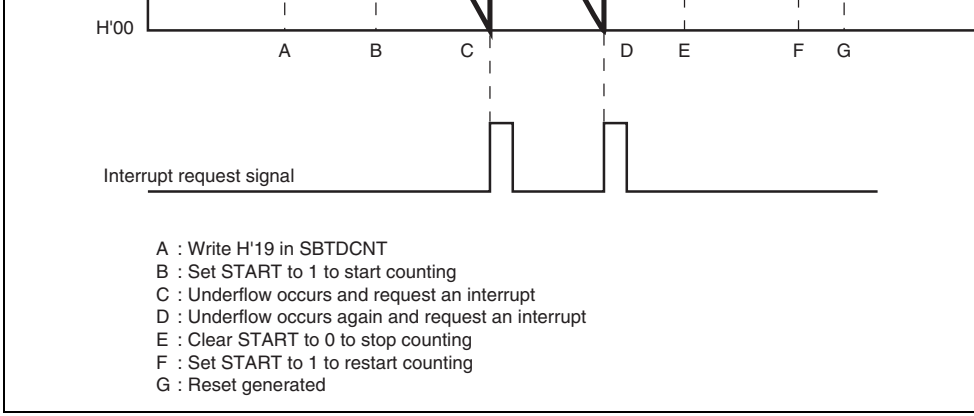
operation and figure 17.3 shows the flowchart.

Clocks are supplied to the entire chip by setting the SYSCKS bit in SBTCTL to 1. When SYSCKS bit is cleared to 0, clock supply to the entire chip is disabled and only the subti operates.

(Example) When ϕ is 32 kHz and the underflow cycle is 100 ms:

$$\frac{32 \times 10^3}{128} \times 100 \times 10^{-3} = 25$$

Therefore, set 25 (H′19) in SBTDCNT.



A : Write H'19 in SBTDCNT
B : Set START to 1 to start counting
C : Underflow occurs and request an interrupt
D : Underflow occurs again and request an interrupt
E : Clear START to 0 to stop counting
F : Set START to 1 to restart counting
G : Reset generated

Figure 17.4 Example of Subtimer Operation

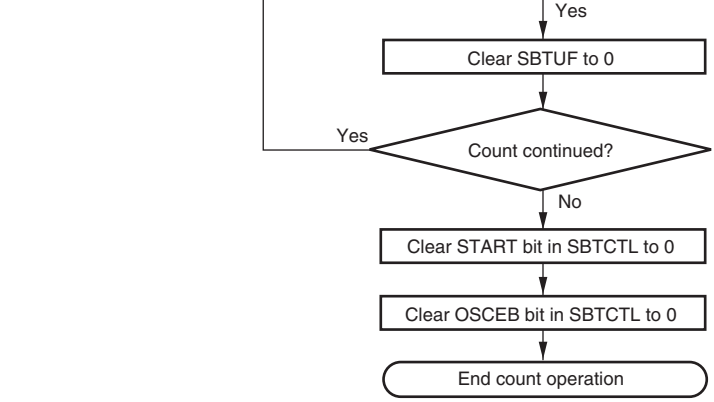
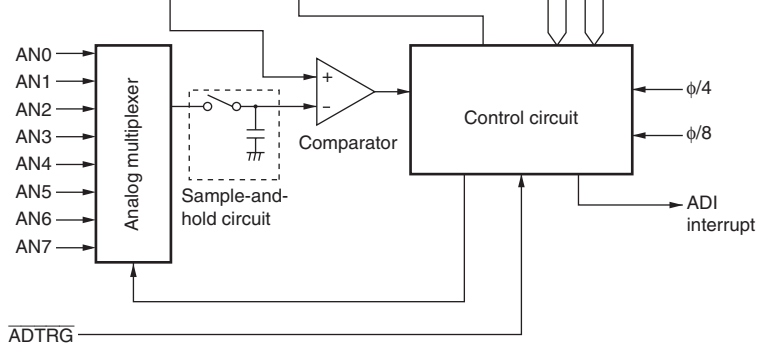


Figure 17.5 Count Operation Flowchart

ROPCR must be written to in active mode with the PCEF bit in SBTCTL set to 1. Otherv
subtimer may operate incorrectly.

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



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

Figure 18.1 Block Diagram of A/D Converter

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

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

There are four 16-bit read-only ADDR registers; ADDRA to ADDRD, used to store the r
A/D conversion. The ADDR registers, which store a conversion result for each analog inp
channel, are shown in table 18.2.

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

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

Table 18.2 Analog Input Channels and Corresponding ADDR Registers

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

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

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

18.3.3 A/D Control Register (ADCR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	TRGE	0	R/W	Trigger Enable A/D conversion is started at the falling edge and t edge of the external trigger signal ($\overline{\text{ADTRG}}$) wher is set to 1. The selection between the falling edge and rising the external trigger pin ($\overline{\text{ADTRG}}$) conforms to the bit in the interrupt edge select register 2 (IEGR2)
6 to 1	—	All 1	—	Reserved These bits are always read as 1.
0	—	0	R/W	Reserved Although this bit is readable/writable, do not set tl 1.

channel as follows:

1. A/D conversion is started when the ADST bit in ADCSR is set to 1, according to sof external trigger input.
2. When A/D conversion is completed, the result is transferred to the corresponding A/ register of the channel.
3. On completion of conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit is s this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion. When A/D conversion ends, bit is automatically cleared to 0 and the A/D converter enters the wait state.

18.4.2 Scan Mode

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

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

In scan mode, the values given in table 18.3 apply to the first conversion time. In the secc
subsequent conversions, the conversion time is 128 states (fixed) when CKS = 0 and 66 s
(fixed) when CKS = 1.

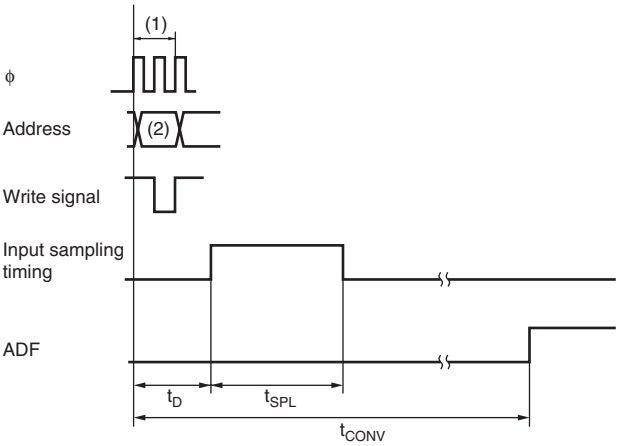


Figure 18.2 A/D Conversion Timing

18.4.4 External Trigger Input Timing

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

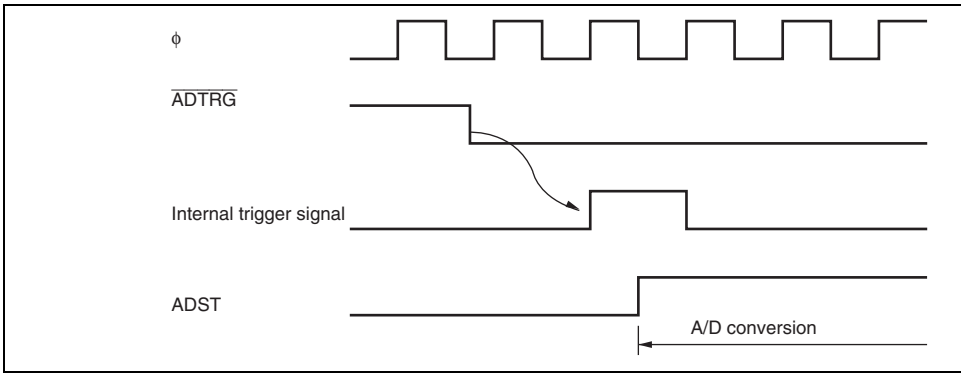


Figure 18.3 External Trigger Input Timing

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

- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion charac
  when the digital output changes from 1111111110 to 1111111111 (see figure 18.5).

- Nonlinearity error

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

- Absolute accuracy

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

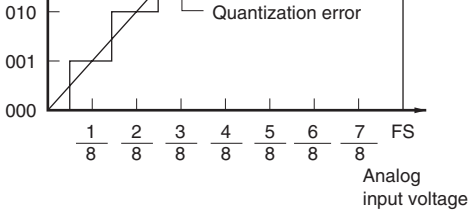


Figure 18.4 A/D Conversion Accuracy Definitions (1)

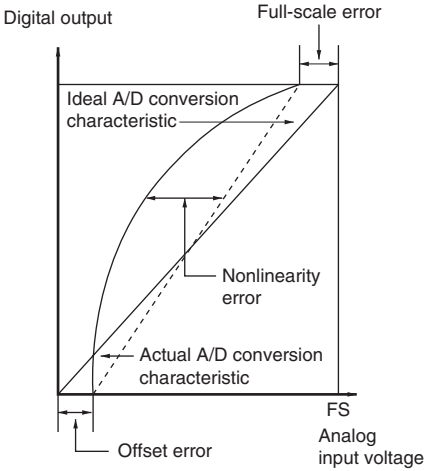


Figure 18.5 A/D Conversion Accuracy Definitions (2)

input resistance of 10 kΩ, and the signal source impedance is ignored. However, as a low filter effect is obtained in this case, it may not be possible to follow an analog signal with differential coefficient (e.g., 5 mV/µs or greater) (see figure 18.6). When converting a hig analog signal or converting in scan mode, a low-impedance buffer should be inserted.

18.6.2 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adve affect absolute accuracy. Be sure to make the connection to an electrically stable GND.

Care is also required to ensure that filter circuits do not interfere with digital signals or ac antennas on the mounting board.

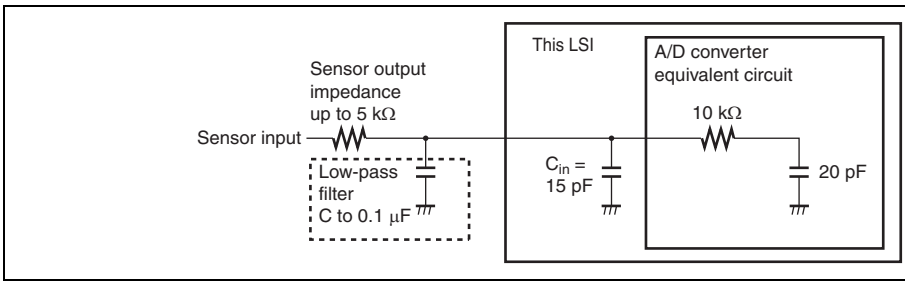


Figure 18.6 Analog Input Circuit Example

power supply voltage rises again.

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

Figure 19.1 is a block diagram of the power-on reset circuit and the low-voltage detectio

circuit is used, or when the LVDI and LVDR circuits are both used.

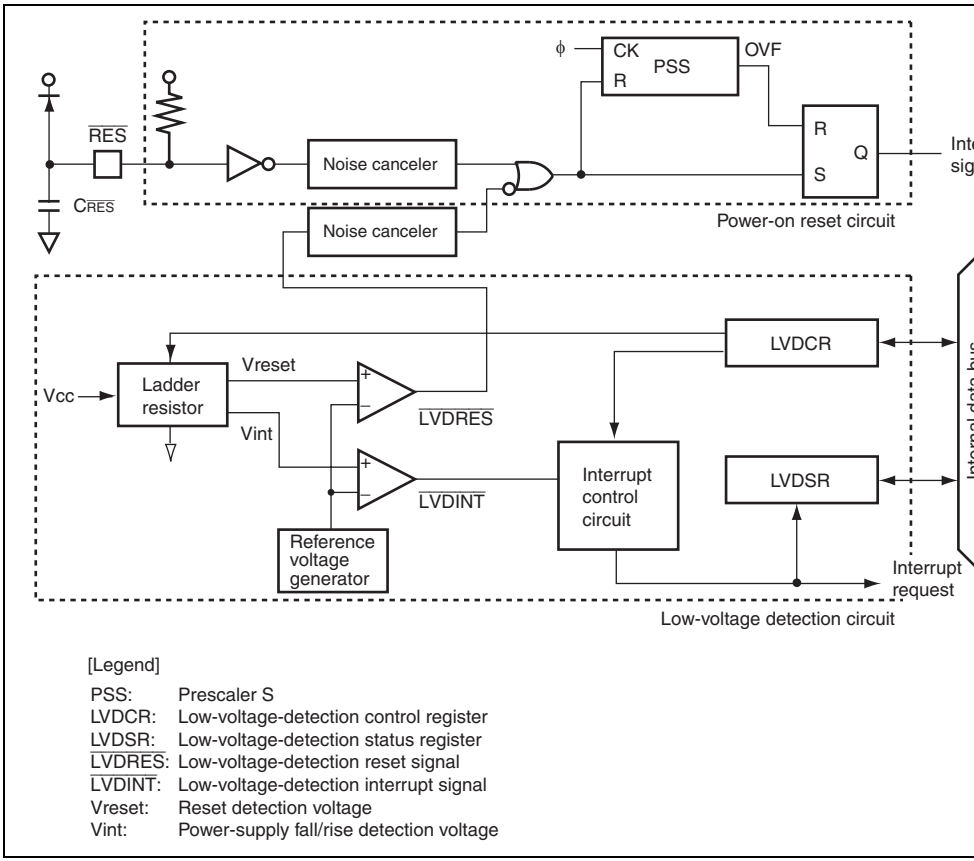


Figure 19.1 Block Diagram of Power-On Reset Circuit and Low-Voltage Detection

interrupt when the power-supply voltage rises above or falls below the respective levels.

Table 19.1 shows the relationship between the LVDCR settings and select functions. LV
should be set according to table 19.1.

Bit	Bit Name	Initial Value	R/W	Description
7	LVDE	0*	R/W	LVD Enable 0: The low-voltage detection circuit is not used standby mode) 1: The low-voltage detection circuit is used
6 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be
3	LVDSEL	0*	R/W	LVDR Detection Level Select 0: Reset detection voltage is 2.3 V (typ.) 1: Reset detection voltage is 3.6 V (typ.) When the falling or rising voltage detection inte used, reset detection voltage of 2.3 V (typ.) sho used. When only a reset detection interrupt is u detection voltage of 3.6 V (typ.) should be used
2	LVDRE	0*	R/W	LVDR Enable 0: Disables the LVDR function 1: Enables the LVDR function
1	LVDDE	0	R/W	Voltage-Fall-Interrupt Enable 0: Interrupt on the power-supply voltage falling selected detection level disabled 1: Interrupt on the power-supply voltage falling selected detection level enabled

LVDCR Settings					Select Functions			
LVDE	LVDSEL	LVDRE	LVDDE	LVDUE	Power-On Reset	LVDR	Low-Voltage-Detection Falling Interrupt	Low Dete Risi Inte
0	*	*	*	*	O	—	—	—
1	1	1	0	0	O	O	—	—
1	0	0	1	0	O	—	O	—
1	0	0	1	1	O	—	O	O
1	0	1	1	1	O	O	O	O

Note: * means invalid.

				[Setting condition] When the power-supply voltage falls below Vint = 3.7 V) [Clearing condition] Writing 0 to this bit after reading it as 1
0	LVDUF	0*	R/W	LVD Power-Supply Voltage Rise Flag [Setting condition] When the power supply voltage falls below Vint the LVDUE bit in LVDCR is set to 1, then rises (U) (typ. = 4.0 V) before falling below Vreset1 (t V) [Clearing condition] Writing 0 to this bit after reading it as 1

Note: * Initialized by LVDR.

131,072 clock (ϕ) cycles. The noise canceler of approximately 500 ns is incorporated to p
the incorrect operation of the chip by noise on the $\overline{\text{RES}}$ pin.

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

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

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

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

Figure 19.2 Operational Timing of Power-On Reset Circuit

19.3.2 Low-Voltage Detection Circuit

LVDR (Reset by Low Voltage Detect) Circuit:

Figure 19.3 shows the timing of the LVDR function. The LVDR enters the module-stan after a power-on reset is canceled. To operate the LVDR, set the LVDE bit in LVDCR t for 50 µs (t_{LVDON}) until the reference voltage and the low-voltage-detection power supply stabilized by a software timer, etc., then set the LVDRE bit in LVDCR to 1. After that, t settings of ports must be made. To cancel the low-voltage detection circuit, first the LVI should be cleared to 0 and then the LVDE bit should be cleared to 0. The LVDE and LV must not be cleared to 0 simultaneously because incorrect operation may occur.

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

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

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

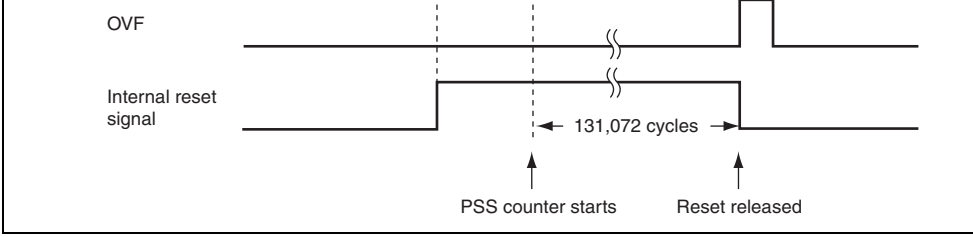


Figure 19.3 Operational Timing of LVDR Circuit

LVDI (Interrupt by Low Voltage Detect) Circuit:

Figure 19.4 shows the timing of LVDI functions. The LVDI enters the module-standby st a power-on reset is canceled. To operate the LVDI, set the LVDE bit in LVDCR to 1, wa µs (t_{LVDON}) until the reference voltage and the low-voltage-detection power supply have sta by a software timer, etc., then set the LVDDE and LVDUE bits in LVDCR to 1. After tha output settings of ports must be made. To cancel the low-voltage detection circuit, first th LVDDE and LVDUE bits should all be cleared to 0 and then the LVDE bit should be clea The LVDE bit must not be cleared to 0 at the same timing as the LVDDE and LVDUE bi because incorrect operation may occur.

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

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

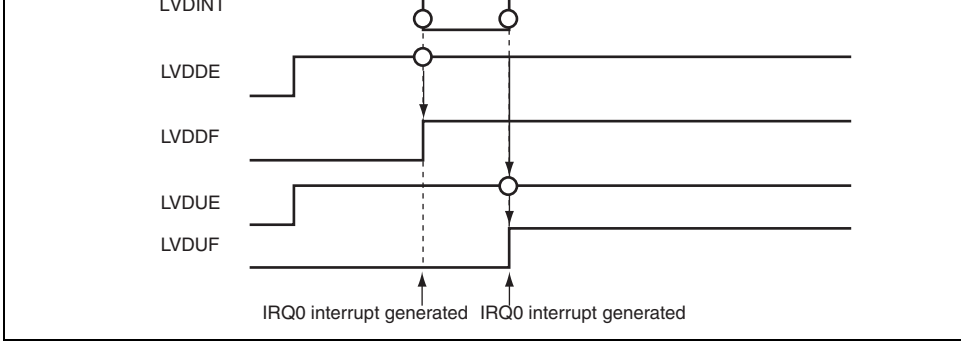


Figure 19.4 Operational Timing of LVDI Circuit

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

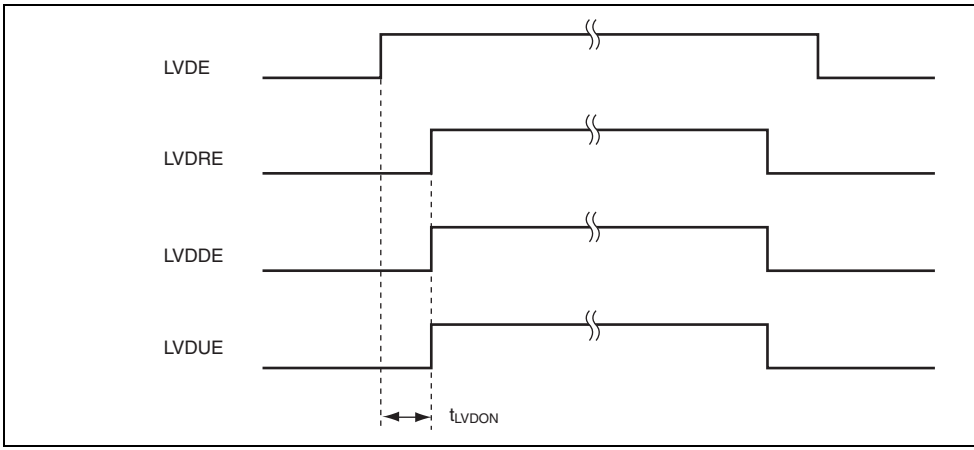


Figure 19.5 Timing for Operation/Release of Low-Voltage Detection Circui

20.1 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 20.1. 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.

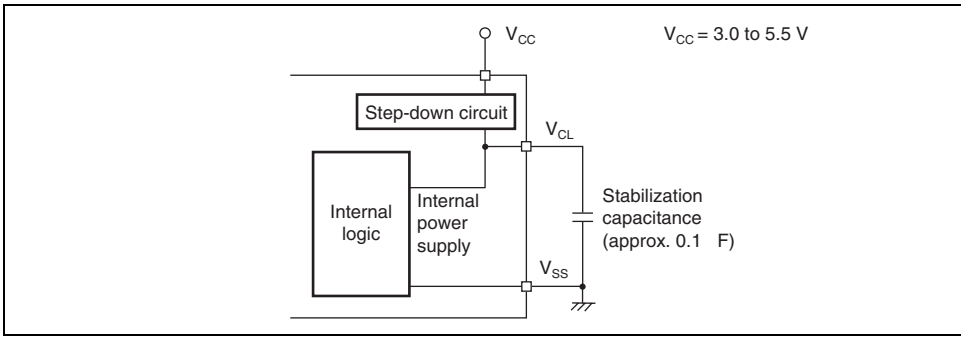


Figure 20.1 Power Supply Connection when Internal Step-Down Circuit is U

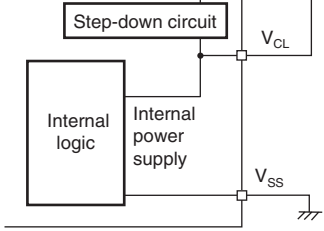


Figure 20.2 Power Supply Connection when Internal Step-Down Circuit is Not

Do not attempt to access reserved addresses.

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

2. Register bits

- Bit configurations of the registers are described in the same order as the register addr
- Reserved bits are indicated by — in the bit name column.
- When registers consist of 16 bits, bits are described from the MSB side.

3. Register states in each operating mode

- Register states are described in the same order as the register addresses.
- The register states described here are for the basic operating modes. If there is a spec for an on-chip peripheral module, refer to the section on that on-chip peripheral mod

—	—	—	H'F000 to H'F5FF	—	—
Master control register	MCR	8	H'F600	TinyCAN	8
General status register	GSR	8	H'F601	TinyCAN	8
Bit configuration register 1	BCR1	8	H'F602	TinyCAN	8
Bit configuration register 0	BCR0	8	H'F603	TinyCAN	8
Mailbox configuration register	MBCR	8	H'F604	TinyCAN	8
TinyCAN module control register	TCMR	8	H'F605	TinyCAN	8
Transmit pending register	TXPR	8	H'F606	TinyCAN	8
Transmit pending cancel register	TXCR	8	H'F608	TinyCAN	8
Transmit acknowledge register	TXACK	8	H'F60A	TinyCAN	8
Abort acknowledge register	ABACK	8	H'F60C	TinyCAN	8
Receive complete register	RXPR	8	H'F60E	TinyCAN	8
Remote request register	RFPR	8	H'F610	TinyCAN	8
TinyCAN interrupt register 1	TCIRR1	8	H'F612	TinyCAN	8
TinyCAN interrupt register 0	TCIRR0	8	H'F613	TinyCAN	8
Mailbox interrupt mask register	MBIMR	8	H'F614	TinyCAN	8
TinyCAN interrupt mask register 1	TCIMR1	8	H'F616	TinyCAN	8
TinyCAN interrupt mask register 0	TCIMR0	8	H'F617	TinyCAN	8
Receive error counter	REC	8	H'F618	TinyCAN	8
Transmit error counter	TEC	8	H'F619	TinyCAN	8
Test control register	TCR	8	H'F61A	TinyCAN	8
Unread message status register	UMSR	8	H'F61B	TinyCAN	8
Message control 0 [0]	MC0[0]	8	H'F620	TinyCAN	8

Message control 1 [6]	MC1[6]	8	H'F62E	TinyCAN	8
Message control 1 [7]	MC1[7]	8	H'F62F	TinyCAN	8
Message control 2 [0]	MC2[0]	8	H'F630	TinyCAN	8
Message control 2 [4]	MC2[4]	8	H'F634	TinyCAN	8
Message control 2 [5]	MC2[5]	8	H'F635	TinyCAN	8
Message control 2 [6]	MC2[6]	8	H'F636	TinyCAN	8
Message control 2 [7]	MC2[7]	8	H'F637	TinyCAN	8
Message control 3 [0]	MC3[0]	8	H'F638	TinyCAN	8
Message control 3 [4]	MC3[4]	8	H'F63C	TinyCAN	8
Message control 3 [5]	MC3[5]	8	H'F63D	TinyCAN	8
Message control 3 [6]	MC3[6]	8	H'F63E	TinyCAN	8
Message control 3 [7]	MC3[7]	8	H'F63F	TinyCAN	8
Message data 0 [0]	MD0[0]	8	H'F640	TinyCAN	8
Message data 0 [1]	MD0[1]	8	H'F641	TinyCAN	8
Message data 0 [2]	MD0[2]	8	H'F642	TinyCAN	8
Message data 0 [3]	MD0[3]	8	H'F643	TinyCAN	8
Message data 0 [4]	MD0[4]	8	H'F644	TinyCAN	8
Message data 0 [5]	MD0[5]	8	H'F645	TinyCAN	8
Message data 0 [6]	MD0[6]	8	H'F646	TinyCAN	8
Message data 0 [7]	MD0[7]	8	H'F647	TinyCAN	8
Message data 1 [0]	MD1[0]	8	H'F648	TinyCAN	8
Message data 1 [1]	MD1[1]	8	H'F649	TinyCAN	8
Message data 1 [2]	MD1[2]	8	H'F64A	TinyCAN	8
Message data 1 [3]	MD1[3]	8	H'F64B	TinyCAN	8

Message data 2 [3]	MD2[3]	8	H'F653	TinyCAN	8	4
Message data 2 [4]	MD2[4]	8	H'F654	TinyCAN	8	4
Message data 2 [5]	MD2[5]	8	H'F655	TinyCAN	8	4
Message data 2 [6]	MD2[6]	8	H'F656	TinyCAN	8	4
Message data 2 [7]	MD2[7]	8	H'F657	TinyCAN	8	4
Message data 3 [0]	MD3[0]	8	H'F658	TinyCAN	8	4
Message data 3 [1]	MD3[1]	8	H'F659	TinyCAN	8	4
Message data 3 [2]	MD3[2]	8	H'F65A	TinyCAN	8	4
Message data 3 [3]	MD3[3]	8	H'F65B	TinyCAN	8	4
Message data 3 [4]	MD3[4]	8	H'F65C	TinyCAN	8	4
Message data 3 [5]	MD3[5]	8	H'F65D	TinyCAN	8	4
Message data 3 [6]	MD3[6]	8	H'F65E	TinyCAN	8	4
Message data 3 [7]	MD3[7]	8	H'F65F	TinyCAN	8	4
Local acceptance filter mask L01	LAFML01	8	H'F660	TinyCAN	8	4
Local acceptance filter mask L00	LAFML00	8	H'F661	TinyCAN	8	4
Local acceptance filter mask H01	LAFMH01	8	H'F662	TinyCAN	8	4
Local acceptance filter mask H00	LAFMH00	8	H'F663	TinyCAN	8	4
Local acceptance filter mask L11	LAFML11	8	H'F664	TinyCAN	8	4
Local acceptance filter mask L10	LAFML10	8	H'F665	TinyCAN	8	4
Local acceptance filter mask H11	LAFMH11	8	H'F666	TinyCAN	8	4
Local acceptance filter mask H10	LAFMH10	8	H'F667	TinyCAN	8	4
Local acceptance filter mask L21	LAFML21	8	H'F668	TinyCAN	8	4
Local acceptance filter mask L20	LAFML20	8	H'F669	TinyCAN	8	4

			H'F69F		
SS control register H	SSCRH	8	H'F6A0	SSU	8
SS control register L	SSCRL	8	H'F6A1	SSU	8
SS mode register	SSMR	8	H'F6A2	SSU	8
SS enable register	SSER	8	H'F6A3	SSU	8
SS status register	SSSR	8	H'F6A4	SSU	8
SS receive data register	SSRDR	8	H'F6A9	SSU	8
SS transmit data register	SSTDR	8	H'F6AB	SSU	8
Subtimer control register	SBTCTL	8	H'F6B0	Subtimer	8
Subtimer counter	SBTDCNT	8	H'F6B1	Subtimer	8
Ring oscillator prescaler setting register	ROPCR	8	H'F6B2	Subtimer	8
—	—	—	H'F6B3 to H'F6FF	—	—
Timer control register_0	TCR_0	8	H'F700	Timer Z	8
Timer I/O control register A_0	TIORA_0	8	H'F701	Timer Z	8
Timer I/O control register C_0	TIORC_0	8	H'F702	Timer Z	8
Timer status register_0	TSR_0	8	H'F703	Timer Z	8
Timer interrupt enable register_0	TIER_0	8	H'F704	Timer Z	8
PWM mode output level control register_0	POCR_0	8	H'F705	Timer Z	8
Timer counter_0	TCNT_0	16	H'F706	Timer Z	16
General register A_0	GRA_0	16	H'F708	Timer Z	16
General register B_0	GRB_0	16	H'F70A	Timer Z	16

PWM mode output level control register_1	POCR_1	8	H'F715	Timer Z	8	2
Timer counter_1	TCNT_1	16	H'F716	Timer Z	16	2
General register A_1	GRA_1	16	H'F718	Timer Z	16	2
General register B_1	GRB_1	16	H'F71A	Timer Z	16	2
General register C_1	GRC_1	16	H'F71C	Timer Z	16	2
General register D_1	GRD_1	16	H'F71E	Timer Z	16	2
Timer start register	TSTR	8	H'F720	Timer Z	8	2
Timer mode register	TMDR	8	H'F721	Timer Z	8	2
Timer PWM mode register	TPMR	8	H'F722	Timer Z	8	2
Timer function control register	TFCR	8	H'F723	Timer Z	8	2
Timer output master enable register	TOER	8	H'F724	Timer Z	8	2
Timer output control register	TOCR	8	H'F725	Timer Z	8	2
—	—	—	H'F726 to H'F72F	—	—	-
Low-voltage-detection control register	LVDCR	8	H'F730	LVDC*[1]	8	2
Low-voltage-detection status register	LVDSR	8	H'F731	LVDC*[1]	8	2
—	—	—	H'F732 to H'F73F	—	—	-
Serial mode register_2	SMR_2	8	H'F740	SCI3_2*[3]	8	3
Bit rate register_2	BRR_2	8	H'F741	SCI3_2*[3]	8	3
Serial control register 3_2	SCR3_2	8	H'F742	SCI3_2*[3]	8	3
Transmit data register_2	TDR_2	8	H'F743	SCI3_2*[3]	8	3

—	—	—	H'F762 to H'FF8F	—	—
Flash memory control register 1	FLMCR1	8	H'FF90	ROM	8
Flash memory control register 2	FLMCR2	8	H'FF91	ROM	8
Flash memory power control register	FLPWCR	8	H'FF92	ROM	8
Erase block register 1	EBR1	8	H'FF93	ROM	8
—	—	—	H'FF94 to H'FF9A	—	—
Flash memory enable register	FENR	8	H'FF9B	ROM	8
—	—	—	H'FF9C to H'FF9F	—	—
Timer control register V0	TCRV0	8	H'FFA0	Timer V	8
Timer control/status register V	TCSRV	8	H'FFA1	Timer V	8
Time constant register A	TCORA	8	H'FFA2	Timer V	8
Time 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
—	—	—	H'FFA6, H'FFA7	—	—
Serial mode register	SMR	8	H'FFA8	SCI3	8
Bit rate register	BRR	8	H'FFA9	SCI3	8
Serial control register 3	SCR3	8	H'FFAA	SCI3	8
Transmit data register	TDR	8	H'FFAB	SCI3	8
Serial status register	SSR	8	H'FFAC	SCI3	8

				converter	
A/D data register D	ADDRD	16	H'FFB6	A/D converter	8
A/D control/status register	ADCSR	8	H'FFB8	A/D converter	8
A/D control register	ADCR	8	H'FFB9	A/D converter	8
—	—	—	H'FFBA to H'FFBF	—	—
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
—	—	—	H'FFC3 to H'FFC7	—	—
Address break control register	ABRKCR	8	H'FFC8	Address break	8
Address break status register	ABRKSR	8	H'FFC9	Address break	8
Break address register H	BARH	8	H'FFCA	Address break	8
Break address register L	BARL	8	H'FFCB	Address break	8
Break data register H	BDRH	8	H'FFCC	Address break	8
Break data register L	BDRL	8	H'FFCD	Address break	8

Port data register 2	PDR2	8	H'FFD5	I/O port	8
—	—	—	H'FFD6, H'FFD7	—	—
Port data register 5	PDR5	8	H'FFD8	I/O port	8
Port data register 6	PDR6	8	H'FFD9	I/O port	8
Port data register 7	PDR7	8	H'FFDA	I/O port	8
Port data register 8	PDR8	8	H'FFDB	I/O port	8
Port data register 9	PDR9	8	H'FFDC	I/O port	8
Port data register B	PDRB	8	H'FFDD	I/O port	8
—	—	—	H'FFDE, H'FFDF	—	—
Port mode register 1	PMR1	8	H'FFE0	I/O port	8
Port mode register 5	PMR5	8	H'FFE1	I/O port	8
Port mode register 3	PMR3	8	H'FFE2	I/O port	8
—	—	—	H'FFE3	—	—
Port control register 1	PCR1	8	H'FFE4	I/O port	8
Port control register 2	PCR2	8	H'FFE5	I/O port	8
—	—	—	H'FFE6, H'FFE7	—	—
Port control register 5	PCR5	8	H'FFE8	I/O port	8
Port control register 6	PCR6	8	H'FFE9	I/O port	8
Port control register 7	PCR7	8	H'FFEA	I/O port	8
Port control register 8	PCR8	8	H'FFEB	I/O port	8
Port control register 9	PCR9	8	H'FFEC	I/O port	8

Interrupt enable register 1	IENR1	8	H'FFF4	Interrupt	8
Interrupt enable register 2	IENR2	8	H'FFF5	Interrupt	8
Interrupt flag register 1	IRR1	8	H'FFF6	Interrupt	8
Interrupt flag register 2	IRR2	8	H'FFF7	Interrupt	8
Wakeup interrupt flag register	IWPR	8	H'FFF8	Interrupt	8
Module standby control register 1	MSTCR1	8	H'FFF9	Power-down	8
Module standby control register 2	MSTCR2	8	H'FFFA	Power-down	8
—	—	—	H'FFFB to H'FFFF	—	—

Notes: 1. LVDC: Low-voltage detection circuits (optional)
2. WDT: Watchdog timer
3. The H8/36037 Group does not have the SCI3_2.

BCR0	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BPR0
MBCR	—	—	—	—	MB3	MB2	MB1	—
TCMR	MSTTC	—	—	—	—	—	PMR97	PMR96
TXPR	—	—	—	—	MB3	MB2	MB1	—
TXCR	—	—	—	—	MB3	MB2	MB1	—
TXACK	—	—	—	—	MB3	MB2	MB1	—
ABACK	—	—	—	—	MB3	MB2	MB1	—
RXPR	—	—	—	—	MB3	MB2	MB1	MB0
RFPR	—	—	—	—	MB3	MB2	MB1	MB0
TCIRR1	—	—	—	WUPI	—	—	OVRI	EMPI
TCIRR0	OVLI	BOFI	EPI	ROWI	TOWI	RFRI	DFRI	RHI
MBIMR	—	—	—	—	MB3	MB2	MB1	MB0
TCIMR1	—	—	—	WUPIM	—	—	OVRIM	EMPIM
TCIMR0	OVLIM	BOFIM	EPIM	ROWIM	TOWIM	RFRIM	DFRIM	RHIM
REC	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
TEC	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
TCR	TSTMD	WREC	FERPS	ATACK	DEC	DRXIN	DTXOT	INTILE
UMSR	—	—	—	—	MB3	MB2	MB1	MB0
MC 0 [0]	DART	NMC	—	—	DLC3	DLC2	DLC1	DLC0
MC 0 [4]	ID20	ID19	ID18	RTR	IDE	—	ID17	ID16
MC 0 [5]	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
MC 0 [6]	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
MC 0 [7]	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
MC 1 [0]	DART	NMC	—	—	DLC3	DLC2	DLC1	DLC0

MC 2 [6]	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
MC 2 [7]	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
MC 3 [0]	DART	NMC	—	—	DLC3	DLC2	DLC1	DLC0
MC 3 [4]	ID20	ID19	ID18	RTR	IDE	—	ID17	ID16
MC 3 [5]	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
MC 3 [6]	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
MC 3 [7]	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
MD 0 [0]	MD07	MD06	MD05	MD04	MD03	MD02	MD01	MD00
MD 0 [1]	MD17	MD16	MD15	MD14	MD13	MD12	MD11	MD10
MD 0 [2]	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20
MD 0 [3]	MD37	MD36	MD35	MD34	MD33	MD32	MD31	MD30
MD 0 [4]	MD47	MD46	MD45	MD44	MD43	MD42	MD41	MD40
MD 0 [5]	MD57	MD56	MD55	MD54	MD53	MD52	MD51	MD50
MD 0 [6]	MD67	MD66	MD65	MD64	MD63	MD62	MD61	MD60
MD 0 [7]	MD77	MD76	MD75	MD74	MD73	MD72	MD71	MD70
MD 1 [0]	MD07	MD06	MD05	MD04	MD03	MD02	MD01	MD00
MD 1 [1]	MD17	MD16	MD15	MD14	MD13	MD12	MD11	MD10
MD 1 [2]	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20
MD 1 [3]	MD37	MD36	MD35	MD34	MD33	MD32	MD31	MD30
MD 1 [4]	MD47	MD46	MD45	MD44	MD43	MD42	MD41	MD40
MD 1 [5]	MD57	MD56	MD55	MD54	MD53	MD52	MD51	MD50
MD 1 [6]	MD67	MD66	MD65	MD64	MD63	MD62	MD61	MD60
MD 1 [7]	MD77	MD76	MD75	MD74	MD73	MD72	MD71	MD70

MD 2 [7]	MD77	MD76	MD75	MD74	MD73	MD72	MD71	MD70
MD 3 [0]	MD07	MD06	MD05	MD04	MD03	MD02	MD01	MD00
MD 3 [1]	MD17	MD16	MD15	MD14	MD13	MD12	MD11	MD10
MD 3 [2]	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20
MD 3 [3]	MD37	MD36	MD35	MD34	MD33	MD32	MD31	MD30
MD 3 [4]	MD47	MD46	MD45	MD44	MD43	MD42	MD41	MD40
MD 3 [5]	MD57	MD56	MD55	MD54	MD53	MD52	MD51	MD50
MD 3 [6]	MD67	MD66	MD65	MD64	MD63	MD62	MD61	MD60
MD 3 [7]	MD77	MD76	MD75	MD74	MD73	MD72	MD71	MD70
LAFML01	LAFML07	LAFML06	LAFML05	LAFML04	LAFML03	LAFML02	LAFML01	LAFML0
LAFML00	LAFML015	LAFML014	LAFML013	LAFML012	LAFML011	LAFML010	LAFML09	LAFML0
LAFMH01	LAFMH07	LAFMH06	LAFMH05	—	—	—	LAFMH01	LAFMH
LAFMH00	LAFMH015	LAFMH014	LAFMH013	LAFMH012	LAFMH011	LAFMH010	LAFMH09	LAFMH
LAFML11	LAFML17	LAFML16	LAFML15	LAFML14	LAFML13	LAFML12	LAFML11	LAFML1
LAFML10	LAFML115	LAFML114	LAFML113	LAFML112	LAFML111	LAFML110	LAFML19	LAFML1
LAFMH11	LAFMH17	LAFMH16	LAFMH15	—	—	—	LAFMH11	LAFMH
LAFMH10	LAFMH115	LAFMH114	LAFMH113	LAFMH112	LAFMH111	LAFMH110	LAFMH19	LAFMH
LAFML21	LAFML27	LAFML26	LAFML25	LAFML24	LAFML23	LAFML22	LAFML21	LAFML2
LAFML20	LAFML215	LAFML214	LAFML213	LAFML212	LAFML211	LAFML210	LAFML29	LAFML2
LAFMH21	LAFMH27	LAFMH26	LAFMH25	—	—	—	LAFMH21	LAFMH
LAFMH20	LAFMH215	LAFMH214	LAFMH213	LAFMH212	LAFMH211	LAFMH210	LAFMH29	LAFMH
LAFML31	LAFML37	LAFML36	LAFML35	LAFML34	LAFML33	LAFML32	LAFML31	LAFML3
LAFML30	LAFML315	LAFML314	LAFML313	LAFML312	LAFML311	LAFML310	LAFML39	LAFML3
LAFMH31	LAFMH37	LAFMH36	LAFMH35	—	—	—	LAFMH31	LAFMH
LAFMH30	LAFMH315	LAFMH314	LAFMH313	LAFMH312	LAFMH311	MAFMH310	LAFMH39	LAFMH

SBTCTL	FCLT	—	—	START	CSCLR	STSCRS	SBTIE	SBTCF
SBTDCNT	SBTDCNT7	SBTDCNT6	SBTDCNT5	SBTDCNT4	SBTDCNT3	SBTDCNT2	SBTDCNT1	SBTDCN
ROPCR	ROPCR7	ROPCR6	ROPCR5	ROPCR4	ROPCR3	ROPCR2	ROPCR1	ROPCR0
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TIORA_0	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
TIORC_0	—	IOD2	IOD1	IOD0	—	IOC2	IOC1	IOC0
TSR_0	—	—	—	OVF	IMFD	IMFC	IMFB	IMFA
TIER_0	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
POCR_0	—	—	—	—	—	POLD	POLC	POLB
TCNT_0	TCNT0H7	TCNT0H6	TCNT0H5	TCNT0H4	TCNT0H3	TCNT0H2	TCNT0H1	TCNT0H
	TCNT0L7	TCNT0L6	TCNT0L5	TCNT0L4	TCNT0L3	TCNT0L2	TCNT0L1	TCNT0L
GRA_0	GRA0H7	GRA0H6	GRA0H5	GRA0H4	GRA0H3	GRA0H2	GRA0H1	GRA0H0
	GRA0L7	GRA0L6	GRA0L5	GRA0L4	GRA0L3	GRA0L2	GRA0L1	GRA0L0
GRB_0	GRB0H7	GRB0H6	GRB0H5	GRB0H4	GRB0H3	GRB0H2	GRB0H1	GRB0H0
	GRB0L7	GRB0L6	GRB0L5	GRB0L4	GRB0L3	GRB0L2	GRB0L1	GRB0L0
GRC_0	GRC0H7	GRC0H6	GRC0H5	GRC0H4	GRC0H3	GRC0H2	GRC0H1	GRC0H0
	GRC0L7	GRC0L6	GRC0L5	GRC0L4	GRC0L3	GRC0L2	GRC0L1	GRC0L0
GRD_0	GRD0H7	GRD0H6	GRD0H5	GRD0H4	GRD0H3	GRD0H2	GRD0H1	GRD0H0
	GRD0L7	GRD0L6	GRD0L5	GRD0L4	GRD0L3	GRD0L2	GRD0L1	GRD0L0
TCR_1	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TIORA_1	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
TIORC_1	—	IOD2	IOD1	IOD0	—	IOC2	IOC1	IOC0
TSR_1	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA
TIER_1	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
POCR_1	—	—	—	—	—	POLD	POLC	POLB

	GRD1L7	GRD1L6	GRD1L5	GRD1L4	GRD1L3	GRD1L2	GRD1L1	GRD1L0
GRD_1	GRD1H7	GRD1H6	GRD1H5	GRD1H4	GRD1H3	GRD1H2	GRD1H1	GRD1H0
TSTR	—	—	—	—	—	—	STR1	STR0
TMDR	BFD1	BFC1	BFD0	BFC0	—	—	—	SYNC
TPMR	—	PWMD1	PWMC1	PWMB1	—	PWMD0	PWMC0	PWMB0
TFCR	—	STCLK	ADEG	ADTRG	OLS1	OLS0	CMD1	CMD0
TOER	ED1	EC1	EB1	EA1	ED0	EC0	EB0	EA0
TOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0
LVDCR	LVDE	—	—	—	LVDSEL	LVDRE	LVDDE	LVDUE
LVDSR	—	—	—	—	—	—	LVDDF	LVDUF
SMR_2	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0
BRR_2	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
SCR3_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_2	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
SSR_2	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT
RDR_2	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
TMB1	TMB17	—	—	—	—	TMB12	TMB11	TMB10
TCB1	TCB17	TCB16	TCB15	TCB14	TCB13	TCB12	TCB11	TCB10
TLB1	TLB17	TLB16	TCLB15	TLB14	TLB13	TLB12	TLB11	TLB10
FLMCR1	—	SWE	ESU	PSU	EV	PV	E	P
FLMCR2	FLER	—	—	—	—	—	—	—
FLPWCR	PDWND	—	—	—	—	—	—	—
EBR1	—	EB6	EB5	EB4	EB3	EB2	EB1	EB0
FENR	FLSHE	—	—	—	—	—	—	—

BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
SCR3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
SSR	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT
RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
ADDRA	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—
ADDRB	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—
ADDRC	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—
ADDRD	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—
ADCSR	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0
ADCR	TRGE	—	—	—	—	—	—	—
TCSRWD	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI	WRST
TCWD	TCWD7	TCWD6	TCWD5	TCWD4	TCWD3	TCWD2	TCWD1	TCWD0
TMWD	CKS7	—	—	—	CKS3	CKS2	CKS1	CKS0
ABRKCR	RTINTE	CSEL1	CSEL0	ACMP2	ACMP1	ACMP0	DCMP1	DCMP0
ABRKSR	ABIF	ABIE	—	—	—	—	—	—
BARH	BARH7	BARH6	BARH5	BARH4	BARH3	BARH2	BARH1	BARH0
BARL	BARL7	BARL6	BARL5	BARL4	BARL3	BARL2	BARL1	BARL0
BDRH	BDRH7	BDRH6	BDRH5	BDRH4	BDRH3	BDRH2	BDRH1	BDRH0
BDRL	BDRL7	BDRL6	BDRL5	BDRL4	BDRL3	BDRL2	BDRL1	BDRL0

PDR8	P87	P86	P85	—	—	—	—	—
PDR9	P97	P96	P95	P94	P93	P92	P91	P90
PDRB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
PMR1	IRQ3	IRQ2	IRQ1	IRQ0	TXD2*[4]	—	TXD	—
PMR5	POF57	POF56	WKP5	WKP4	WKP3	WKP2	WKP1	WKP0
PMR3	—	—	—	POF24	POF23	—	—	—
PCR1	PCR17	PCR16	PCR15	PCR14	—	PCR12	PCR11	PCR10
PCR2	—	—	—	PCR24	PCR23	PCR22	PCR21	PCR20
PCR5	PCR57	PCR56	PCR55	PCR54	PCR53	PCR52	PCR51	PCR50
PCR6	PCR67	PCR66	PCR65	PCR64	PCR63	PCR62	PCR61	PCR60
PCR7	—	PCR76	PCR75	PCR74	—	PCR72	PCR71	PCR70
PCR8	PCR87	PCR86	PCR85	—	—	—	—	—
PCR9	PCR97	PCR96	PCR95	PCR94	PCR93	PCR92	PCR91	PCR90
SYSCR1	SSBY	STS2	STS1	STS0	—	—	—	—
SYSCR2	SMSEL	LSON	DTON	MA2	MA1	MA0	SA1	SA0
IEGR1	NMIEG	—	—	—	IEG3	IEG2	IEG1	IEG0
IEGR2	—	—	WPEG5	WPEG4	WPEG3	WPEG2	WPEG1	WPEG0
IENR1	IENDT	—	IENWP	—	IEN3	IEN2	IEN1	IEN0
IENR2	—	—	IENTB1	—	—	—	—	—
IRR1	IRRDT	—	—	—	IRRI3	IRRI2	IRRI1	IRRI0
IRR2	—	—	IRRTB1	—	—	—	—	—
IWPR	—	—	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0
MSTCR1	—	—	MSTS3	MSTAD	MSTWD	—	MSTTV	—
MSTCR2	MSTS3_2*[4]	—	—	MSTTB1	—	—	MSTTZ	—

TXPR	Initialized	—	—	—	—	—
TXCR	Initialized	—	—	—	—	—
TXACK	Initialized	—	—	—	—	—
ABACK	Initialized	—	—	—	—	—
RXPR	Initialized	—	—	—	—	—
RFPR	Initialized	—	—	—	—	—
TCIRR1	Initialized	—	—	—	—	—
TCIRR0	Initialized	—	—	—	—	—
MBIMR	Initialized	—	—	—	—	—
TCIMR1	Initialized	—	—	—	—	—
TCIMR0	Initialized	—	—	—	—	—
REC	Initialized	—	—	—	—	—
TEC	Initialized	—	—	—	—	—
TCR	Initialized	—	—	—	—	—
UMSR	Initialized	—	—	—	—	—
MC0[1]	—	—	—	—	—	—
MC0[2]	—	—	—	—	—	—
MC0[3]	—	—	—	—	—	—
MC0[4]	—	—	—	—	—	—
MC0[5]	—	—	—	—	—	—
MC1[1]	—	—	—	—	—	—
MC1[2]	—	—	—	—	—	—
MC1[3]	—	—	—	—	—	—

MC3[1]	—	—	—	—	—	—
MC3[2]	—	—	—	—	—	—
MC3[3]	—	—	—	—	—	—
MC3[4]	—	—	—	—	—	—
MC3[5]	—	—	—	—	—	—
MD0[1]	—	—	—	—	—	—
MD0[2]	—	—	—	—	—	—
MD0[3]	—	—	—	—	—	—
MD0[4]	—	—	—	—	—	—
MD0[5]	—	—	—	—	—	—
MD0[6]	—	—	—	—	—	—
MD0[7]	—	—	—	—	—	—
MD0[8]	—	—	—	—	—	—
MD1[1]	—	—	—	—	—	—
MD1[2]	—	—	—	—	—	—
MD1[3]	—	—	—	—	—	—
MD1[4]	—	—	—	—	—	—
MD1[5]	—	—	—	—	—	—
MD1[6]	—	—	—	—	—	—
MD1[7]	—	—	—	—	—	—
MD1[8]	—	—	—	—	—	—
MD2[1]	—	—	—	—	—	—
MD2[2]	—	—	—	—	—	—
MD2[3]	—	—	—	—	—	—

MD3[3]	—	—	—	—	—	—	
MD3[4]	—	—	—	—	—	—	
MD3[5]	—	—	—	—	—	—	
MD3[6]	—	—	—	—	—	—	
MD3[7]	—	—	—	—	—	—	
MD3[8]	—	—	—	—	—	—	
LAFML0[1]	—	—	—	—	—	—	
LAFML0[0]	—	—	—	—	—	—	
LAFMH0[1]	—	—	—	—	—	—	
LAFMH0[0]	—	—	—	—	—	—	
LAFML1[1]	—	—	—	—	—	—	
LAFML1[0]	—	—	—	—	—	—	
LAFMH1[1]	—	—	—	—	—	—	
LAFMH1[0]	—	—	—	—	—	—	
LAFML2[1]	—	—	—	—	—	—	
LAFML2[0]	—	—	—	—	—	—	
LAFMH2[1]	—	—	—	—	—	—	
LAFMH2[0]	—	—	—	—	—	—	
LAFML3[1]	—	—	—	—	—	—	
LAFML3[0]	—	—	—	—	—	—	
LAFMH3[1]	—	—	—	—	—	—	
LAFMH3[0]	—	—	—	—	—	—	
SSCRH	Initialized	—	—	—	—	—	S
SSCRL	Initialized	—	—	—	—	—	

TCR_0	Initialized	—	—	—	—	—	Ti
TIORA_0	Initialized	—	—	—	—	—	
TIORC_0	Initialized	—	—	—	—	—	
TSR_0	Initialized	—	—	—	—	—	
TIER_0	Initialized	—	—	—	—	—	
POCR_0	Initialized	—	—	—	—	—	
TCNT_0	Initialized	—	—	—	—	—	
GRA_0	Initialized	—	—	—	—	—	
GRB_0	Initialized	—	—	—	—	—	
GRC_0	Initialized	—	—	—	—	—	
GRD_0	Initialized	—	—	—	—	—	
TCR_1	Initialized	—	—	—	—	—	
TIORA_1	Initialized	—	—	—	—	—	
TIORC_1	Initialized	—	—	—	—	—	
TSR_1	Initialized	—	—	—	—	—	
TIER_1	Initialized	—	—	—	—	—	
POCR_1	Initialized	—	—	—	—	—	
TCNT_1	Initialized	—	—	—	—	—	
GRA_1	Initialized	—	—	—	—	—	
GRB_1	Initialized	—	—	—	—	—	
GRC_1	Initialized	—	—	—	—	—	
GRD_1	Initialized	—	—	—	—	—	
TSTR	Initialized	—	—	—	—	—	

SMR_2	Initialized	—	—	Initialized	Initialized	Initialized
BRR_2	Initialized	—	—	Initialized	Initialized	Initialized
SCR3_2	Initialized	—	—	Initialized	Initialized	Initialized
TDR_2	Initialized	—	—	Initialized	Initialized	Initialized
SSR_2	Initialized	—	—	Initialized	Initialized	Initialized
RDR_2	Initialized	—	—	Initialized	Initialized	Initialized
TMB1	Initialized	—	—	—	—	—
TCB1	Initialized	—	—	—	—	—
Tlb1	Initialized	—	—	—	—	—
FLMCR1	Initialized	—	—	Initialized	Initialized	Initialized
FLMCR2	Initialized	—	—	—	—	—
FLPWCR	Initialized	—	—	—	—	—
EBR1	Initialized	—	—	Initialized	Initialized	Initialized
FENR	Initialized	—	—	—	—	—
TCRV0	Initialized	—	—	Initialized	Initialized	Initialized
TCSRV	Initialized	—	—	Initialized	Initialized	Initialized
TCORA	Initialized	—	—	Initialized	Initialized	Initialized
TCORB	Initialized	—	—	Initialized	Initialized	Initialized
TCNTV	Initialized	—	—	Initialized	Initialized	Initialized
TCRV1	Initialized	—	—	Initialized	Initialized	Initialized
SMR	Initialized	—	—	Initialized	Initialized	Initialized
BRR	Initialized	—	—	Initialized	Initialized	Initialized
SCR3	Initialized	—	—	Initialized	Initialized	Initialized
TDR	Initialized	—	—	Initialized	Initialized	Initialized

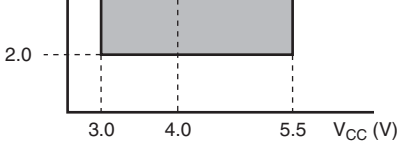
ADDR	Initialized	—	—	Initialized	Initialized	Initialized	
TCSRWD	Initialized	—	—	—	—	—	W
TCWD	Initialized	—	—	—	—	—	
TMWD	Initialized	—	—	—	—	—	
ABRKCR	Initialized	—	—	—	—	—	Ac
ABRKSR	Initialized	—	—	—	—	—	br
BARH	Initialized	—	—	—	—	—	
BARL	Initialized	—	—	—	—	—	
BDRH	Initialized	—	—	—	—	—	
BDRL	Initialized	—	—	—	—	—	
PUCR1	Initialized	—	—	—	—	—	I/O
PUCR5	Initialized	—	—	—	—	—	
PDR1	Initialized	—	—	—	—	—	
PDR2	Initialized	—	—	—	—	—	
PDR5	Initialized	—	—	—	—	—	
PDR6	Initialized	—	—	—	—	—	
PDR7	Initialized	—	—	—	—	—	
PDR8	Initialized	—	—	—	—	—	
PDR9	Initialized	—	—	—	—	—	
PDRB	Initialized	—	—	—	—	—	
PMR1	Initialized	—	—	—	—	—	
PMR5	Initialized	—	—	—	—	—	
PMR3	Initialized	—	—	—	—	—	
PCR1	Initialized	—	—	—	—	—	

STSCR2	Initialized	—	—	—	—	—	
IEGR1	Initialized	—	—	—	—	—	
IEGR2	Initialized	—	—	—	—	—	
IENR1	Initialized	—	—	—	—	—	
IENR2	Initialized	—	—	—	—	—	
IRR1	Initialized	—	—	—	—	—	
IRR2	Initialized	—	—	—	—	—	
IWPR	Initialized	—	—	—	—	—	
MSTCR1	Initialized	—	—	—	—	—	
MSTCR2	Initialized	—	—	—	—	—	

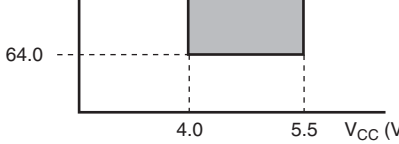
Notes: 1. LVDC: Low-voltage detection circuits (optional)
2. WDT: Watchdog timer
3. The H8/36037 Group does not have the SCI3_2.

	Port B		–0.3 to AV_{CC} +0.3	V
Operating temperature		T_{opr}	Regular specifications: –20 to +75	°C
			Wide-range specifications: –40 to +85	
Storage temperature		T_{stg}	–55 to +125	°C

Note: * Permanent damage may result if absolute maximum ratings are exceeded. N
operation should be under the conditions specified in Electrical Characteristic
Exceeding these values can result in incorrect operation and reduced reliabili



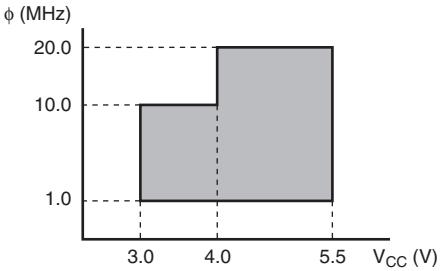
AV_{CC} = 3.3 to 5.5 V
- Active mode
- Sleep mode



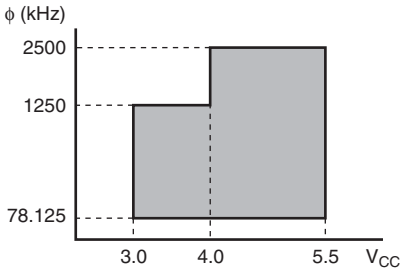
AV_{CC} = 3.3 to 5.5 V
- Active mode
- Sleep mode
- Subactive mode
- Subsleep mode

Note: This frequency range is supplied by the o
oscillator for the subtimer and is guarante

Power Supply Voltage and Operating Frequency Range:



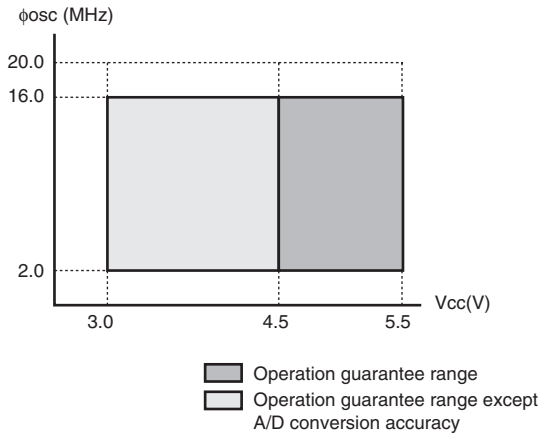
AV_{CC} = 3.3 to 5.5 V
- Active mode
- Sleep mode
(When MA2 in SYSCR2 = 0)



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

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

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



AVcc = 4.0 to 5.5 V • Active mode • Sleep mode	AVcc = 4.0 to 5.5 V • Subactive mode • Subsleep mode

Note: * Reference value

		IRQ0 to IRQ3, ADTRG, TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, SCK3, SCK3_2*[1], SCS, SSCK, TRGV, TMIB1		$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	
		RXD, RXD_2*[1], SSI, SSO, HRXD, P10 to P12, P14 to P17, P20 to P24, P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 to P87, P90 to P97	V_{CC} = 4.0 to 5.5 V	$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V
				$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	
		PB0 to PB7	V_{CC} = 4.0 to 5.5 V	$V_{CC} \times 0.7$	—	$AV_{CC} + 0.3$	V
				$V_{CC} \times 0.8$	—	$AV_{CC} + 0.3$	
		OSC1	V_{CC} = 4.0 to 5.5 V	$V_{CC} - 0.5$	—	$V_{CC} + 0.3$	V
				$V_{CC} - 0.3$	—	$V_{CC} + 0.3$	
Input low voltage	V_{IL}	RES, NMI, WKP0 to WKP5, IRQ0 to IRQ3, ADTRG, TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, SCK3, SCK3_2*[1], SCS, SSCK, TRGV, TMIB1	V_{CC} = 4.0 to 5.5 V	–0.3	—	$V_{CC} \times 0.2$	V
				–0.3	—	$V_{CC} \times 0.1$	

		PB0 to PB7	V_{CC} = 4.0 to 5.5 V	–0.3	—	V_{CC} × 0.3	V
				–0.3	—	V_{CC} × 0.2	
		OSC1	V_{CC} = 4.0 to 5.5 V	–0.3	—	0.5	V
				–0.3	—	0.3	
Output high voltage	V_{OH}	P10 to P12, P14 to P17, P20 to P24, P50 to P55, P60 to P67, P70 to P72, P74 to P76, P85 to P87, P90 to P97	V_{CC} = 4.0 to 5.5 V –I_{OH} = 1.5 mA	V_{CC} – 1.0	—	—	V
			–I_{OH} = 0.1 mA	V_{CC} – 0.5	—	—	
		P56, P57	V_{CC} = 4.0 to 5.5 V –I_{OH} = 0.1 mA	V_{CC} – 2.5	—	—	V
			V_{CC} = 3.0 to 4.0 V –I_{OH} = 0.1 mA	V_{CC} – 2.0	—	—	

			V_{CC} = 4.0 to 5.5 V I_{OL} = 10.0 mA	—	—	1.0	
			V_{CC} = 4.0 to 5.5 V I_{OL} = 1.6 mA	—	—	0.4	
			I_{OL} = 0.4 mA	—	—	0.4	
Input/ output leakage current	$\lvert I_{IL} \rvert$	OSC1, $\overline{\text{RES}}$, $\overline{\text{NMI}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{ADTRG}}$, TRGV, TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, RXD, SCK3, RXD_2*[1], SCK3_2*[1], SSCK, $\overline{\text{SCS}}$, SSI, SSO, HRXD	V_{IN} = 0.5 V or higher (V_{CC} – 0.5 V)	—	—	1.0	µA
		P10 to P12, P14 to P17, P20 to P24, P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 to P87, P90 to P97	V_{IN} = 0.5 V or higher (V_{CC} – 0.5 V)	—	—	1.0	µA
		PB0 to PB7	V_{IN} = 0.5 V or higher (AV_{CC} – 0.5 V)	—	—	1.0	µA

supply current								
			Active mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	10.0	—		*
	I_{OPE2}	V_{CC}	Active mode 2 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	1.2	3.0	mA	*
			Active mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	0.8	—		*
Sleep mode supply current	I_{SLEEP1}	V_{CC}	Sleep mode 1 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	14.0	22.5	mA	*
			Sleep mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	6.3	—		*
	I_{SLEEP2}	V_{CC}	Sleep mode 2 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	1.0	2.7	mA	*
			Sleep mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	0.7	—		*
Subactive mode supply current	I_{SUB}	V_{CC}	V_{CC} = 5.0 V (ϕ_{SUB} = ϕ_W/2)	—	60.0	100.0	µA	*
			V_{CC} = 5.0 V (ϕ_{SUB} = ϕ_W/8)	—	46.0	—		*
Subsleep mode supply current	I_{SUBSP}	V_{CC}	V_{CC} = 5.0 V (ϕ_{SUB} = ϕ_W/2)	—	50.0	80.0	µA	*

2. The **LVD** is optional.
3. Pin states during supply current measurement are given below (excluding cu pull-up MOS transistors and output buffers).

Mode	$\overline{\text{RES}}$ Pin	Internal State	Other Pins	Oscillator Pins
Active mode 1	V_{CC}	Operates	V_{CC}	Main clock: ceramic or cryst resonator
Active mode 2		Operates (ϕ_{OSC}/64)		
Sleep mode 1	V_{CC}	Only timers operate	V_{CC}	
Sleep mode 2		Only timers operate (ϕ_{OSC}/64)		
Subactive mode	V_{CC}	Operates	V_{CC}	Main clock: on-chip oscillato
Subsleep mode	V_{CC}	Only timers operate	V_{CC}	
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	Main clock: ceramic or cryst resonator

		except port 6				
		Port 6		—	—	10.0
Allowable output low current (total)	ΣI_{OL}	Output pins except port 6	V_{CC} = 4.0 to 5.5 V	—	—	40.0
		Port 6		—	—	80.0
		Output pins except port 6		—	—	20.0
		Port 6		—	—	40.0
Allowable output high current (per pin)	$\lvert -I_{OH} \rvert$	All output pins	V_{CC} = 4.0 to 5.5 V	—	—	2.0
				—	—	0.2
Allowable output high current (total)	$\Sigma \lvert -I_{OH} \rvert$	All output pins	V_{CC} = 4.0 to 5.5 V	—	—	30.0
				—	—	8.0

System clock (•) cycle time	t_{cyc}			1	—	64	t_{OSC}
				—	—	12.8	µs
Subclock oscillator oscillation frequency	f_{RO}		V_{CC} = 4.0 to 5.5 V	64.0	—	850.0	kHz
Subclock oscillator (ϕw) cycle time	t_{RO}		V_{CC} = 4.0 to 5.5 V	1.18	—	15.6	µs
Subclock (ϕsub) cycle time	t_{subcyc}		V_{CC} = 4.0 to 5.5 V	2	—	8	ϕw
Instruction cycle time				2	—	—	t_{cyc} t_{subcyc}
Oscillation stabilization time (crystal resonator)	t_{rc}	OSC1, OSC2		—	—	10.0	ms
Oscillation stabilization time (ceramic resonator)	t_{rc}	OSC1, OSC2		—	—	5.0	ms
External clock high width	t_{CPH}	OSC1	V_{CC} = 4.0 to 5.5 V	20.0	—	—	ns
				40.0	—	—	
External clock low width	t_{CPL}	OSC1	V_{CC} = 4.0 to 5.5 V	20.0	—	—	ns
				40.0	—	—	
External clock rise time	t_{CPr}	OSC1	V_{CC} = 4.0 to 5.5 V	—	—	10.0	ns
				—	—	15.0	
External clock fall time	t_{CPf}	OSC1	V_{CC} = 4.0 to 5.5 V	—	—	10.0	ns
				—	—	15.0	

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

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

2. Determined by the MA2, MA1, MA0, SA1, and SA0 bits in the system control r (SYSCR2).

Input clock pulse width	t_{SCKW}	SCK3, SCK3_2*		0.4	—	0.6	t_{scyc}
Transmit data delay time (clocked synchronous)	t_{TXD}	TXD, TXD_2*	V_{CC} = 4.0 to 5.5 V	—	—	1	t_{cyc}
				—	—	1	
Receive data setup time (clocked synchronous)	t_{RXS}	RXD, RXD_2*	V_{CC} = 4.0 to 5.5 V	50.0	—	—	ns
				100.0	—	—	
Receive data hold time (clocked synchronous)	t_{RXH}	RXD, RXD_2*	V_{CC} = 4.0 to 5.5 V	50.0	—	—	ns
				100.0	—	—	

Note: * The H8/36037 Group does not have these pins.

data setup time*						
Receive data hold time*	t_{HRXH}	HRXD	50	—	—	ns

Note: * Although the TinyCAN input/output signal is asynchronous, its state is determi
have changed at the rising-edge (two clock cycles) of the CK clock shown in fi
22.6.

Clock rise time	Master	t_{RISE}	SSCK	—	—	1	t_{CYC}
	Slave			—	—	1.0	µs
Clock fall time	Master	t_{FALL}	SSCK	—	—	1	t_{CYC}
	Slave			—	—	1.0	µs
Data input setup time		t_{SU}	SSO, SSI	1	—	—	t_{CYC}
Data input hold time		t_{H}	SSO, SSI	1	—	—	t_{CYC}
$\overline{SCS}$ setup time	Slave	t_{LEAD}	$\overline{SCS}$	1 t_{CYC} + 100	—	—	ns
$\overline{SCS}$ hold time	Slave	t_{LAG}	$\overline{SCS}$	1 t_{CYC} + 100	—	—	ns
Data output delay time		t_{OD}	SSO, SSI	—	—	1	t_{CYC}
Slave access time		t_{SA}	SSI	—	—	1 t_{CYC} + 100	ns
Slave out release time		t_{OR}	SSI	—	—	1 t_{CYC} + 100	ns

Analog input voltage	AV_{IN}	AN0 to AN7		V_{SS} – 0.3	—	AV_{CC} + 0.3	V	
Analog power supply current	AI_{OPE}	AV_{CC}	AV_{CC} = 5.0 V f_{OSC} = 20 MHz	—	—	2.0	mA	
	AI_{STOP1}	AV_{CC}		—	50	—	µA	*
	AI_{STOP2}	AV_{CC}		—	—	5.0	µA	*
Analog input capacitance	C_{AIN}	AN0 to AN7		—	—	30.0	pF	
Allowable signal source impedance	R_{AIN}	AN0 to AN7		—	—	5.0	kΩ	
Resolution (data length)				10	10	10	bit	
Conversion time (single mode)			AV_{CC} = 3.3 to 5.5 V	134	—	—	t_{cyc}	
Nonlinearity error				—	—	±7.5	LSB	
Offset error				—	—	±7.5	LSB	
Full-scale error				—	—	±7.5	LSB	
Quantization error				—	—	±0.5	LSB	
Absolute accuracy				—	—	±8.0	LSB	
Conversion time (single mode)			AV_{CC} = 4.0 to 5.5 V	70	—	—	t_{cyc}	
Nonlinearity error				—	—	±7.5	LSB	
Offset error				—	—	±7.5	LSB	
Full-scale error				—	—	±7.5	LSB	
Quantization error				—	—	±0.5	LSB	
Absolute accuracy				—	—	±8.0	LSB	

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

22.2.5 Watchdog Timer Characteristics

Table 22.8 Watchdog Timer Characteristics

V_{CC} = 3.0 to 5.5 V, V_{SS} = 0.0 V, T_a = –20 to +75°C (regular specifications) or T_a = –40 to
(wide-range specifications), unless otherwise indicated.

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

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

Reprogramming count		N_{WEC}		1000	10000	—
Programming	Wait time after SWE bit setting*[1]	x		1	—	—
	Wait time after PSU bit setting*[1]	y		50	—	—
	Wait time after P bit setting *[1]*[4]	z1	1 ≤ n ≤ 6	28	30	32
		z2	7 ≤ n ≤ 1000	198	200	202
		z3	Additional-programming	8	10	12
	Wait time after P bit clear*[1]	α		5	—	—
	Wait time after PSU bit clear*[1]	β		5	—	—
	Wait time after PV bit setting*[1]	γ		4	—	—
	Wait time after dummy write*[1]	ε		2	—	—
	Wait time after PV bit clear*[1]	η		2	—	—
	Wait time after SWE bit clear*[1]	θ		100	—	—
	Maximum programming count *[1]*[4]*[5]	N		—	—	1000

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

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

2. The programming time for 128 bytes. (Indicates the total time for which the P flash memory control register 1 (FLMCR1) is set. The program-verify time is n included.)

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

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

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

   Programming count (n)

   $1 \leq n \leq 6$ z1 = 30 µs

   $7 \leq n \leq 1000$ z2 = 200 µs

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

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

Reset detection voltage 1*[1]	Vreset1	LVDSEL = 0	—	2.3	2.7
Reset detection voltage 2*[2]	Vreset2	LVDSEL = 1	3.0	3.6	4.2
Lower-limit voltage of LVDR operation*[3]	$V_{LVDRmin}$		1.0	—	—
LVD stabilization time	t_{LVDON}		50	—	—
Supply current in standby mode	I_{STBY}	LVDE = 1, Vcc = 5.0 V, subtimer and WDT not used	—	—	350

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

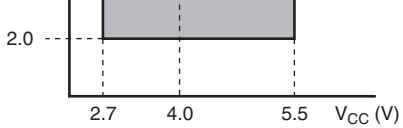
22.2.8 Power-On Reset Circuit Characteristics (Optional)

Table 22.11 Power-On Reset Circuit Characteristics

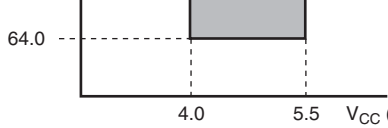
V_{SS} = 0.0 V, T_a = –20 to +75°C (regular specifications) or T_a = –40 to +85°C (wide-range specifications), unless otherwise indicated.

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

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



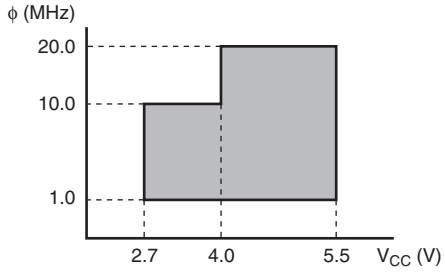
AV_{CC} = 3.3 to 5.5 V
- Active mode
- Sleep mode



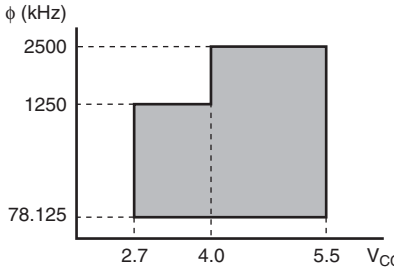
AV_{CC} = 3.3 to 5.5 V
- Active mode
- Sleep mode
- Subactive mode
- Subsleep mode

Note: This frequency range is supplied by the
oscillator for the subtimer and is guarant

Power Supply Voltage and Operating Frequency Range:



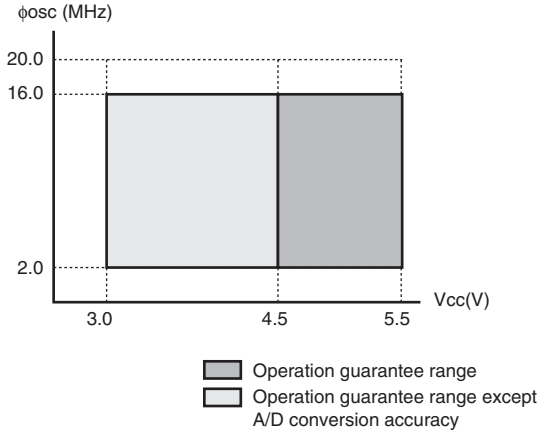
AV_{CC} = 3.3 to 5.5 V
- Active mode
- Sleep mode
(When MA2 in SYSCR2 = 0)



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

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

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



AVcc = 4.0 to 5.5 V • Active mode • Sleep mode	AVcc = 4.0 to 5.5 V • Subactive mode • Subsleep mode

Note: * Reference value

	IRQ0 to IRQ3, ADTRG,TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, SCK3, SCK3_2*[1], SCS, SSCK, TRGV, TMIB1		$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	
	RXD, RXD_2*[1], SSI, SSO, HRXD, P10 to P12, P14 to P17, P20 to P24, P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 to P87 P90 to P97	V_{CC} = 4.0 to 5.5 V	$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V
			$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	
V_{IH}	PB0 to PB7	V_{CC} = 4.0 to 5.5 V	$V_{CC} \times 0.7$	—	$AV_{CC} + 0.3$	V
			$V_{CC} \times 0.8$	—	$AV_{CC} + 0.3$	
	OSC1	V_{CC} = 4.0 to 5.5 V	$V_{CC} - 0.5$	—	$V_{CC} + 0.3$	V
			$V_{CC} - 0.3$	—	$V_{CC} + 0.3$	

		RXD, RXD_2*[1], SSI, SSO, HRXD, P10 to P12, P14 to P17, P20 to P24, P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 to P87, P90 to P97	V_{CC} = 4.0 to 5.5 V	–0.3	—	$V_{CC} \times 0.3$	V
				–0.3	—	$V_{CC} \times 0.2$	
		PB0 to PB7	V_{CC} = 4.0 to 5.5 V	–0.3	—	$V_{CC} \times 0.3$	V
				–0.3	—	$V_{CC} \times 0.2$	
		OSC1	V_{CC} = 4.0 to 5.5 V	–0.3	—	0.5	V
				–0.3	—	0.3	
Output high voltage	V_{OH}	P10 to P12, P14 to P17, P20 to P24, P50 to P55, P60 to P67, P70 to P72, P74 to P76, P85 to P87, P90 to P97	V_{CC} = 4.0 to 5.5 V $-I_{OH}$ = 1.5 mA	V_{CC} – 1.0	—	—	V
			$-I_{OH}$ = 0.1 mA	V_{CC} – 0.5	—	—	
		P56, P57	V_{CC} = 4.0 to 5.5 V $-I_{OH}$ = 0.1 mA	V_{CC} – 2.5	—	—	V
			V_{CC} = 3.0 to 4.0 V $-I_{OH}$ = 0.1 mA	V_{CC} – 2.0	—	—	

			V_{CC} = 4.0 to 5.5 V I_{OL} = 10.0 mA	—	—	1.0	
			V_{CC} = 4.0 to 5.5 V I_{OL} = 1.6 mA	—	—	0.4	
			I_{OL} = 0.4 mA	—	—	0.4	
Input/ output leakage current	$\lvert I_{IL} \rvert$	OSC1, $\overline{\text{RES}}$, $\overline{\text{NMI}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{ADTRG}}$, TRGV, TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, RXD, SCK3, RXD_2*[1], SCK3_2*[1], SSCK, $\overline{\text{SCS}}$, SSI, SSO, HRXD	V_{IN} = 0.5 V or higher (V_{CC} – 0.5 V)	—	—	1.0	µA
		P10 to P12, P14 to P17, P20 to P24, P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 to P87, P90 to P97	V_{IN} = 0.5 V or higher (V_{CC} – 0.5 V)	—	—	1.0	µA
		PB0 to PB7	V_{IN} = 0.5 V or higher (AV_{CC} – 0.5 V)	—	—	1.0	µA
Pull-up MOS current	$-I_p$	P10 to P12, P14 to P17, P50 to P55	V_{CC} = 5.0 V, V_{IN} = 0.0 V	50.0	—	300.0	µA
			V_{CC} = 3.0 V, V_{IN} = 0.0 V	—	60.0	—	

	I_{OPE2}	V_{CC}	Active mode 2 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	1.2	3.0	mA
			Active mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	0.8	—	
Sleep mode supply current	I_{SLEEP1}	V_{CC}	Sleep mode 1 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	14.0	22.5	mA
			Sleep mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	6.3	—	
	I_{SLEEP2}	V_{CC}	Sleep mode 2 V_{CC} = 5.0 V, f_{OSC} = 20 MHz	—	1.0	2.7	mA
			Sleep mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	0.7	—	
Subactive mode supply current	I_{SUB}	V_{CC}	V_{CC} = 5.0 V (ϕ_{SUB} = ϕ_W/2)	—	60.0	100.0	µA
			V_{CC} = 5.0 V (ϕ_{SUB} = ϕ_W/8)	—	46.0	—	
Subsleep mode supply current	I_{SUBSP}	V_{CC}	V_{CC} = 5.0 V (ϕ_{SUB} = ϕ_W/2)	—	50.0	80.0	µA
Standby mode supply current	I_{STBY}	V_{CC}	Subtimer, WDT, and LVD*[2] not used	—	—	5.0	µA

Mode	$\overline{\text{RES}}$ Pin	Internal State	Other Pins	Oscillator Pins
Active mode 1	V_{CC}	Operates	V_{CC}	Main clock: ceramic or crys resonator
Active mode 2		Operates (ϕ_{OSC}/64)		
Sleep mode 1	V_{CC}	Only timers operate	V_{CC}	
Sleep mode 2		Only timers operate (ϕ_{OSC}/64)		
Subactive mode	V_{CC}	Operates	V_{CC}	Main clock: on-chip oscillat
Subsleep mode	V_{CC}	Only timers operate	V_{CC}	
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	Main clock: ceramic or crys resonator

		except port 6				
		Port 6		—	—	10.0
Allowable output low current (total)	ΣI_{OL}	Output pins except port 6	V_{cc} = 4.0 to 5.5 V	—	—	40.0
		Port 6		—	—	80.0
		Output pins except port 6		—	—	20.0
		Port 6		—	—	40.0
Allowable output high current (per pin)	$\lvert -I_{OH} \rvert$	All output pins	V_{cc} = 4.0 to 5.5 V	—	—	2.0
				—	—	0.2
Allowable output high current (total)	$\Sigma \lvert -I_{OH} \rvert$	All output pins	V_{cc} = 4.0 to 5.5 V	—	—	30.0
				—	—	8.0

frequency				2.0		16.0		
System clock (ϕ) cycle time	t_{cyc}			1	—	64	t_{OSC}	*
				—	—	12.8	µs	
Subclock oscillator oscillation frequency	f_{RO}		V_{CC} = 4.0 to 5.5 V	64.0	—	850.0	kHz	
Subclock oscillator (ϕw) cycle time	t_{RO}		V_{CC} = 4.0 to 5.5 V	1.18	—	15.6	µs	
Subclock (ϕsub) cycle time	t_{subcyc}		V_{CC} = 4.0 to 5.5 V	2	—	8	ϕw	
Instruction cycle time				2	—	—	t_{cyc} t_{subcyc}	
Oscillation stabilization time (crystal resonator)	t_{rc}	OSC1, OSC2		—	—	10.0	ms	
Oscillation stabilization time (ceramic resonator)	t_{rc}	OSC1, OSC2		—	—	5.0	ms	
External clock high width	t_{CPH}	OSC1	V_{CC} = 4.0 to 5.5 V	20.0	—	—	ns	F
				40.0	—	—		
External clock low width	t_{CPL}	OSC1	V_{CC} = 4.0 to 5.5 V	20.0	—	—	ns	
				40.0	—	—		
External clock rise time	t_{CPr}	OSC1	V_{CC} = 4.0 to 5.5 V	—	—	10.0	ns	
				—	—	15.0		
External clock fall time	t_{CPf}	OSC1	V_{CC} = 4.0 to 5.5 V	—	—	10.0	ns	
				—	—	15.0		

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

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

2. Determined by the MA2, MA1, MA0, SA1, and SA0 bits in the system control
(SYSCR2).

Input clock pulse width	t_{SCKW}	SCK3, SCK3_2*		0.4	—	0.6	t_{scyc}	
Transmit data delay time (clocked synchronous)	t_{TXD}	TXD, TXD_2*	V_{CC} = 4.0 to 5.5 V	—	—	1	t_{cyc}	Fig
				—	—	1		
Receive data setup time (clocked synchronous)	t_{RXS}	RXD, RXD_2*	V_{CC} = 4.0 to 5.5 V	50.0	—	—	ns	
				100.0	—	—		
Receive data hold time (clocked synchronous)	t_{RXH}	RXD, RXD_2*	V_{CC} = 4.0 to 5.5 V	50.0	—	—	ns	
				100.0	—	—		

Note: * The H8/36037 Group does not have these pins.

data setup time*						
Receive data hold time*	t_{HRXH}	HRXD	50	—	—	ns

Note: * Although the TinyCAN input/output signal is asynchronous, its state is determ
have changed at the rising-edge (two clock cycles) of the CK clock shown in
22.6.

Clock rise time	Master	t_{RISE}	SSCK	—	—	1	t_{CYC}
	Slave			—	—	1.0	µs
Clock fall time	Master	t_{FALL}	SSCK	—	—	1	t_{CYC}
	Slave			—	—	1.0	µs
Data input setup time		t_{SU}	SSO, SSI	1	—	—	t_{CYC}
Data input hold time		t_{H}	SSO, SSI	1	—	—	t_{CYC}
$\overline{SCS}$ setup time	Slave	t_{LEAD}	$\overline{SCS}$	1 t_{CYC} + 100	—	—	ns
$\overline{SCS}$ hold time	Slave	t_{LAG}	$\overline{SCS}$	1 t_{CYC} + 100	—	—	ns
Data output delay time		t_{OD}	SSO, SSI	—	—	1	t_{CYC}
Slave access time		t_{SA}	SSI	—	—	1 t_{CYC} + 100	ns
Slave out release time		t_{OR}	SSI	—	—	1 t_{CYC} + 100	ns

Analog input voltage	AV_{IN}	AN0 to AN7		V_{SS} – 0.3	—	AV_{CC} + 0.3	V
Analog power supply current	AI_{OPE}	AV_{CC}	AV_{CC} = 5.0 V f_{OSC} = 20 MHz	—	—	2.0	mA
	AI_{STOP1}	AV_{CC}		—	50	—	µA
	AI_{STOP2}	AV_{CC}		—	—	5.0	µA
Analog input capacitance	C_{AIN}	AN0 to AN7		—	—	30.0	pF
Allowable signal source impedance	R_{AIN}	AN0 to AN7		—	—	5.0	kΩ
Resolution (data length)				10	10	10	bit
Conversion time (single mode)			AV_{CC} = 3.3 to 5.5 V	134	—	—	t_{cyc}
Nonlinearity error				—	—	±7.5	LSB
Offset error				—	—	±7.5	LSB
Full-scale error				—	—	±7.5	LSB
Quantization error				—	—	±0.5	LSB
Absolute accuracy				—	—	±8.0	LSB
Conversion time (single mode)			AV_{CC} = 4.0 to 5.5 V	70	—	—	t_{cyc}
Nonlinearity error				—	—	±7.5	LSB
Offset error				—	—	±7.5	LSB
Full-scale error				—	—	±7.5	LSB
Quantization error				—	—	±0.5	LSB
Absolute accuracy				—	—	±8.0	LSB

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

22.3.5 Watchdog Timer Characteristics

Table 22.19 Watchdog Timer Characteristics

V_{CC} = 2.7 to 5.5 V, V_{SS} = 0.0 V, T_a = –20 to +75°C (regular specifications) or T_a = –40 to (wide-range specifications), unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Re Fi
				Min.	Typ.	Max.		
Internal oscillator overflow time	t_{OVF}			0.2	0.4	—	s	*

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

Reset detection voltage 1*[1]	Vreset1	LVDSEL = 0	—	2.3	2.7
Reset detection voltage 2*[2]	Vreset2	LVDSEL = 1	3.0	3.6	4.2
Lower-limit voltage of LVDR operation*[3]	$V_{LVDRmin}$		1.0	—	—
LVD stabilization time	t_{LVDON}		50	—	—
Supply current in standby mode	I_{STBY}	LVDE = 1, Vcc = 5.0 V, subtimer and WDT not used	—	—	350

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

22.3.7 Power-On Reset Circuit Characteristics (Optional)

Table 22.21 Power-On Reset Circuit Characteristics

V_{SS} = 0.0 V, T_a = –20 to +75°C (regular specifications) or T_a = –40 to +85°C (wide-range specifications), unless otherwise indicated.

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

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

Figure 22.1 System Clock Input Timing

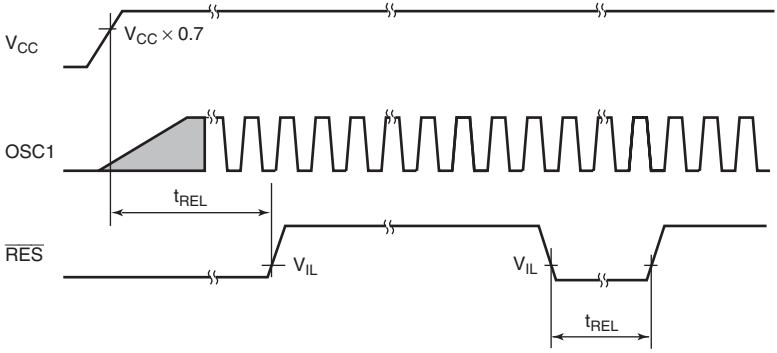


Figure 22.2 $\overline{\text{RES}}$ Low Width Timing

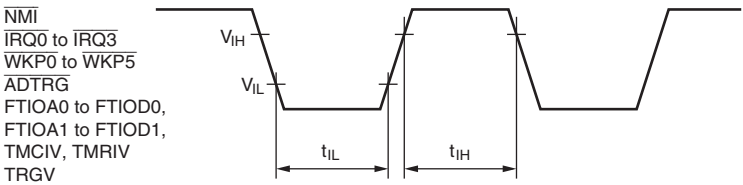
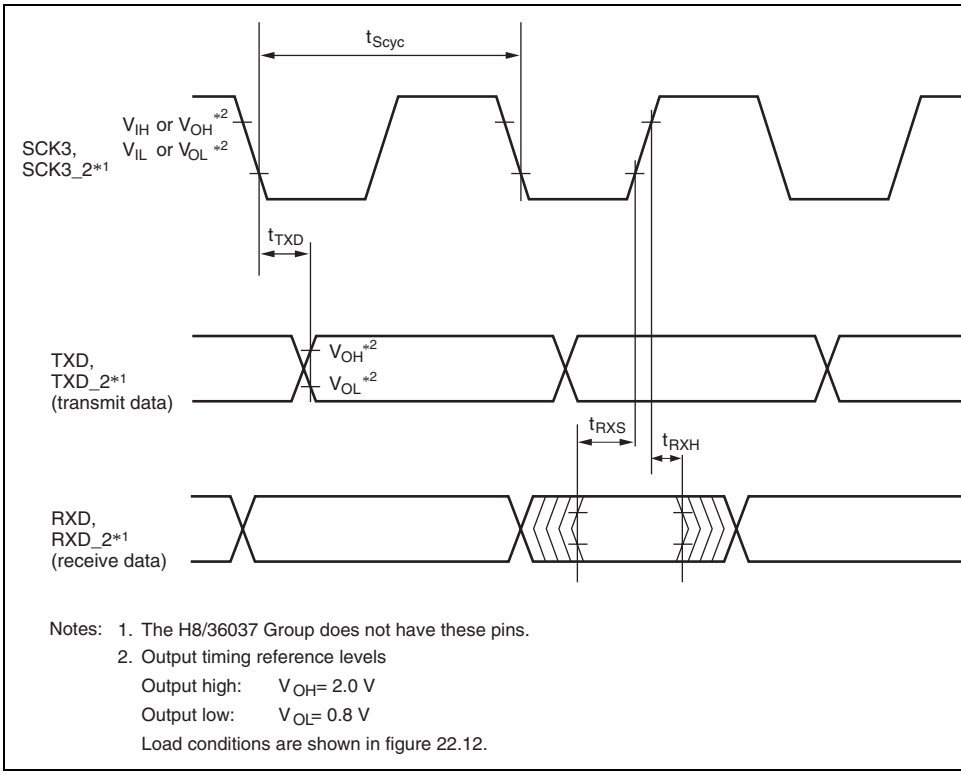


Figure 22.3 Input Timing



Notes: 1. The H8/36037 Group does not have these pins.
2. Output timing reference levels
Output high: V_{OH}= 2.0 V
Output low: V_{OL}= 0.8 V
Load conditions are shown in figure 22.12.

Figure 22.5 SCI Input/Output Timing in Clocked Synchronous Mode

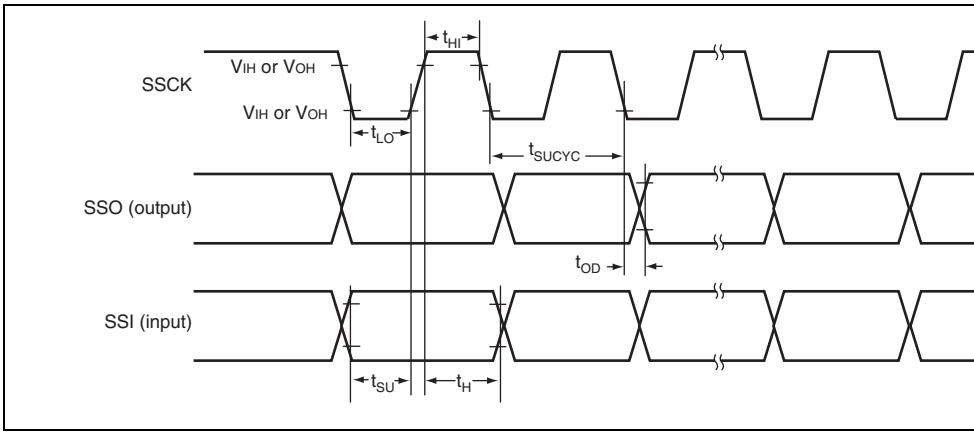


Figure 22.7 SSU Input/Output Timing in Clocked Synchronous Mode

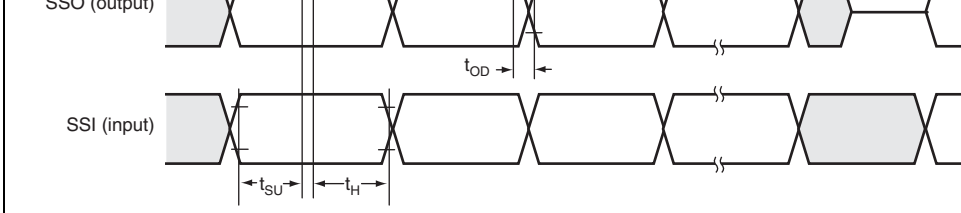


Figure 22.8 SSU Input/Output Timing
(Four-Line Bus Communication Mode, Master, CPHS = 1)

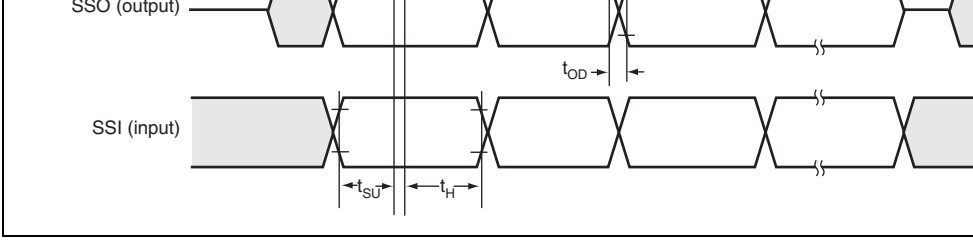


Figure 22.9 SSU Input/Output Timing
(Four-Line Bus Communication Mode, Master, CPHS = 0)

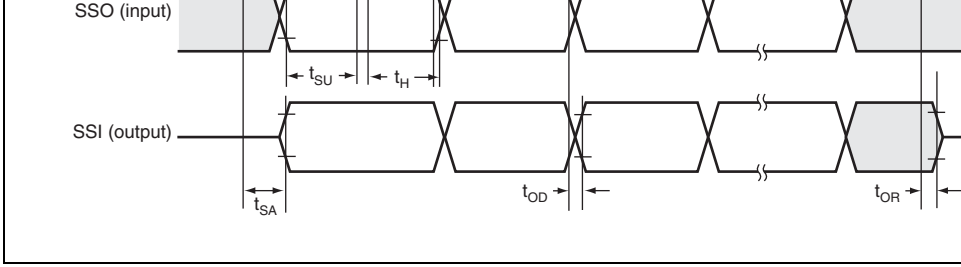


Figure 22.10 SSU Input/Output Timing
(Four-Line Bus Communication Mode, Slave, CPHS = 1)

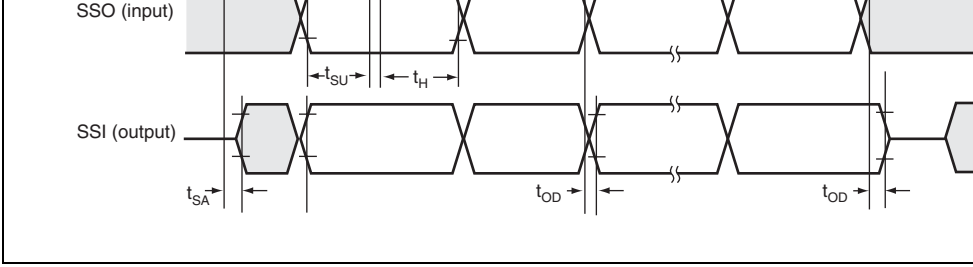


Figure 22.11 SSU Input/Output Timing
(Four-Line Bus Communication Mode, Slave, CPHS = 0)

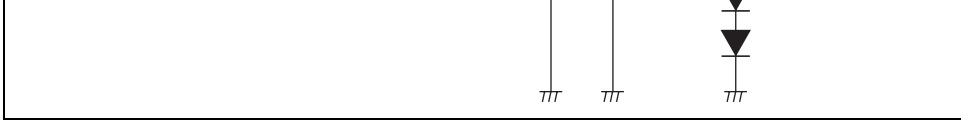


Figure 22.12 Output Load Circuit

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

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

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

	MOV.L @ERs, ERd	L			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		2							(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)	—	—	↕	↕	0	
	EXTS.L ERd	L		2							(<bit 15> of ERd32) → (<bits 31 to 16> of ERd32)	—	—	↕	↕	0	

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

	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				—	—	—	—	—			

									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 B

Instruction when most significant bit of B

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

Instruction code:

1st byte		2nd byte	
AH	AL	BH	BL

AH AL \ BH	0	1	2	3	4	5	6	7	8	9	A	B
01	MOV				LDC/STC				SLEEP			
0A	INC											A
0B	ADDS					INC		INC	ADDS			
0F	DAA											M
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											S
1B	SUBS					DEC		DEC	SUB			
1F	DAS											C
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 / STC
01C05	MULXS		MULXS									
01D05		DIVXS		DIVXS								
01F06					OR	XOR	AND					
7Cr06 *1				BTST								
7Cr07 *1				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD				
7Dr06 *1	BSET	BNOT	BCLR					BST / BIST				
7Dr07 *1	BSET	BNOT	BCLR									
7Eaa6 *2				BTST								
7Eaa7 *2				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD				
7Faa6 *2	BSET	BNOT	BCLR					BST / BIST				
7Faa7 *2	BSET	BNOT	BCLR									

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

BSET #0, @FF00

From table A.4:

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

From table A.3:

$S_I = 2$, $S_L = 2$

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

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

JSR @@ 30

From table A.4:

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

From table A.3:

$S_I = S_J = S_K = 2$

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

Note: * Depends on which on-chip peripheral module is accessed. See section 21.1, F
Addresses (Address Order).

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

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

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

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

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

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

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

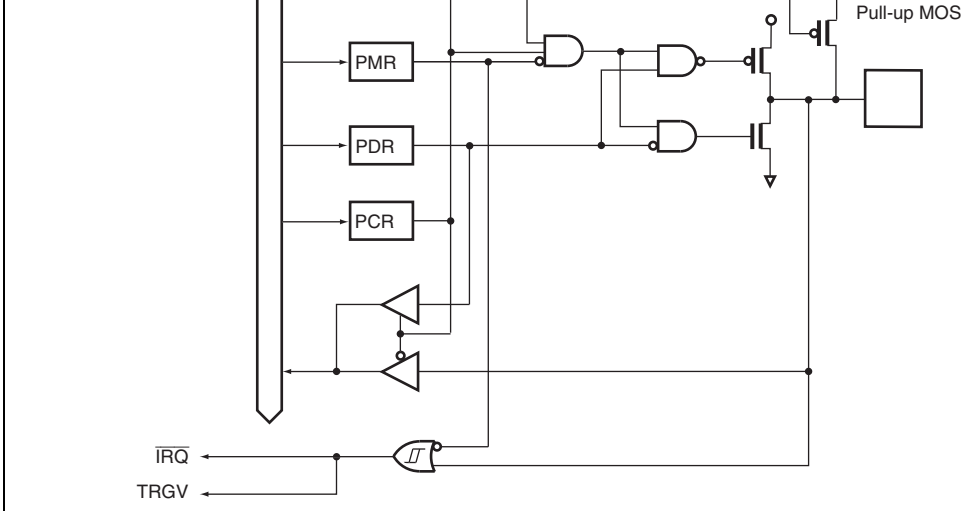
	SHAL.L ERd	1		
SHAR	SHAR.B Rd	1		
	SHAR.W Rd	1		
	SHAR.L ERd	1		
SHLL	SHLL.B Rd	1		
	SHLL.W Rd	1		
	SHLL.L ERd	1		
SHLR	SHLR.B Rd	1		
	SHLR.W Rd	1		
	SHLR.L ERd	1		
SLEEP	SLEEP	1		
STC	STC CCR, Rd	1		
	STC CCR, @ERd	2		1
	STC CCR, @(d:16,ERd)	3		1
	STC CCR, @(d:24,ERd)	5		1
	STC CCR,@-ERd	2		1
	STC CCR,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

	XOR.L #xx:32, ERd	3
	XOR.L ERs, ERd	2
XORC	XORC #xx:8, CCR	1

Notes: 1. n: Specified value in R4L and R4. The source and destination operands are a
n+1 times respectively.

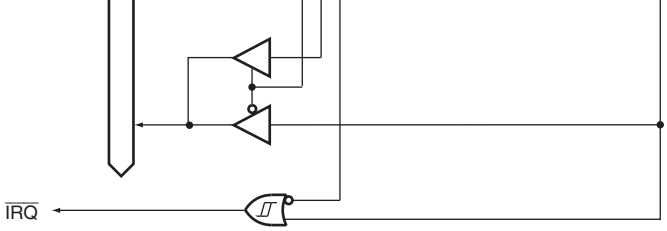
2. Cannot be used in this LSI.

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



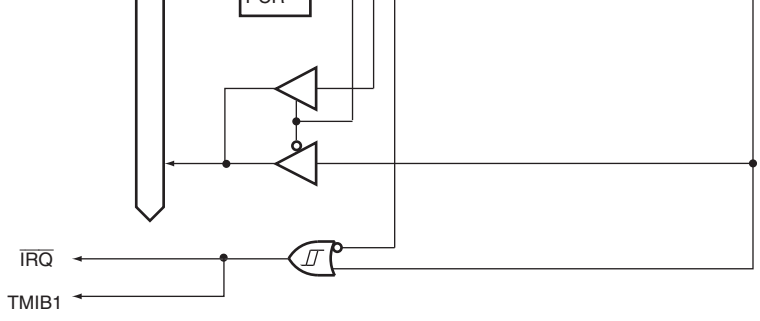
[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.1 Port 1 Block Diagram (P17)



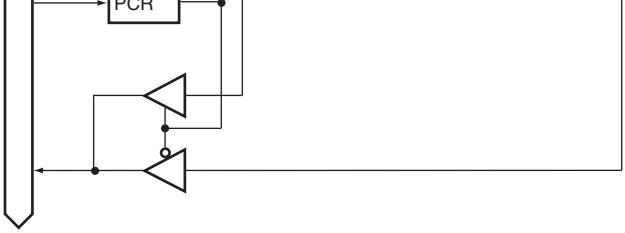
[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

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



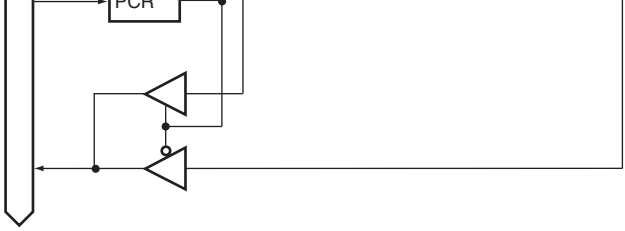
[Legend]
PUCR : Port pull-up control register
PMR : Port mode register
PDR : Port data register
PCR : Port control register

Figure B.3 Port 1 Block Diagram (P15)



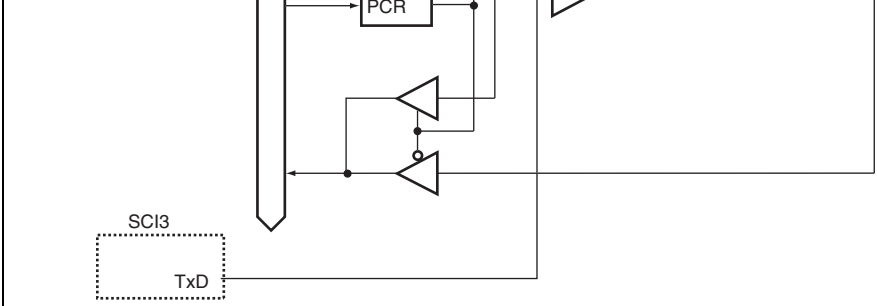
[Legend]
PUCR : Port pull-up control register
PDR : Port data register
PCR : Port control register

Figure B.4 Port 1 Block Diagram (P12, P11, P10)



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

Figure B.5 Port 2 Block Diagram (P24, P23)



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

Figure B.6 Port 2 Block Diagram (P22)

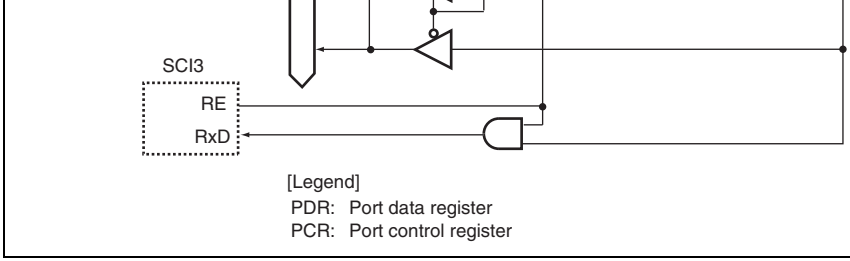
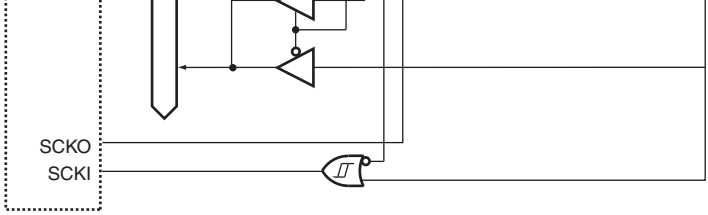
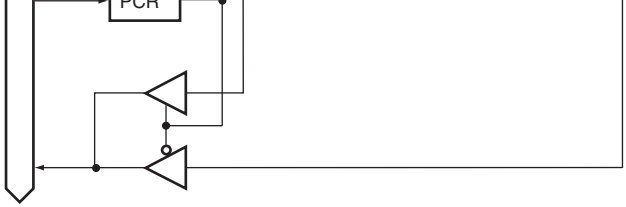


Figure B.7 Port 2 Block Diagram (P21)



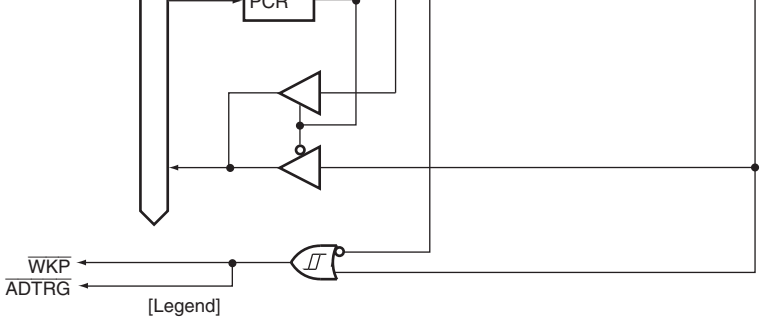
[Legend]
PDR: Port data register
PCR: Port control register

Figure B.8 Port 2 Block Diagram (P20)



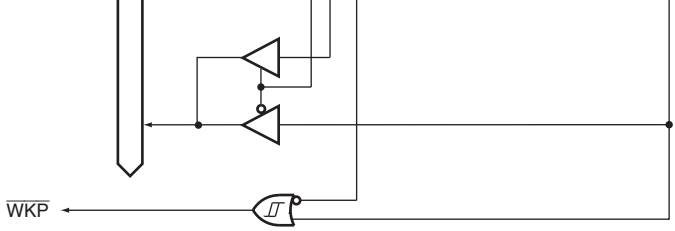
[Legend]
PMR : Port mode register
PDR : Port data register
PCR : Port control register

Figure B.9 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.10 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.11 Port 5 Block Diagram (P54 to P55)

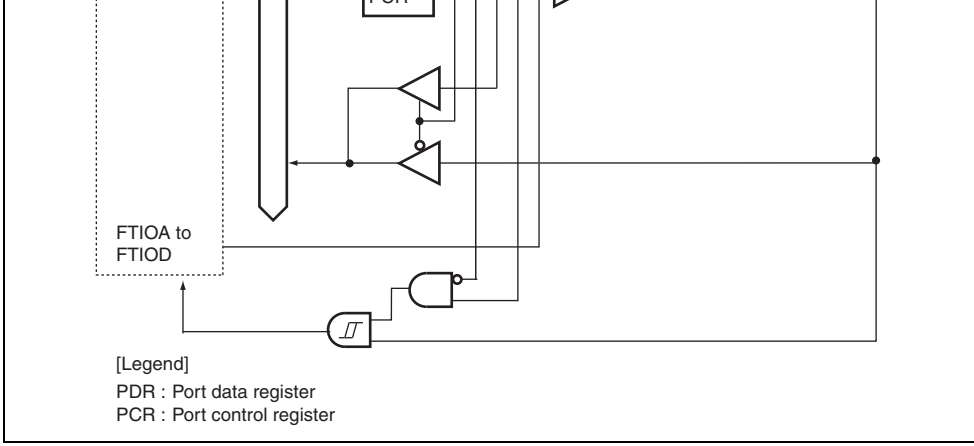
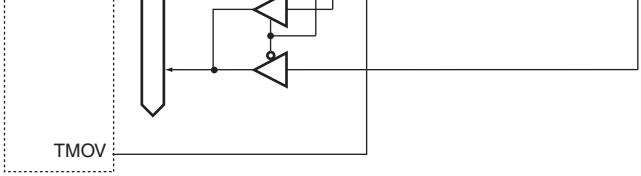


Figure B.12 Port 6 Block Diagram (P67 to P60)



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

Figure B.13 Port 7 Block Diagram (P76)

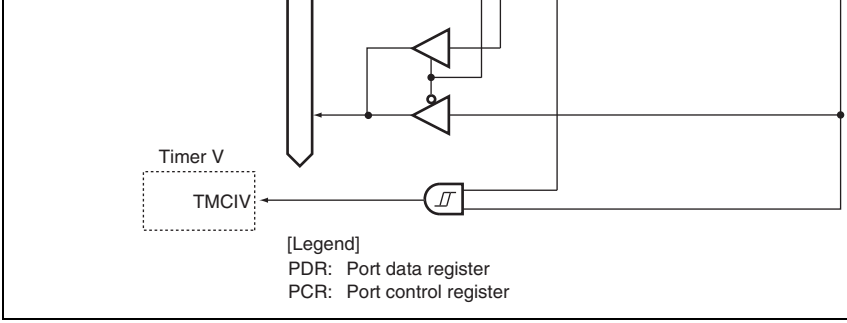


Figure B.14 Port 7 Block Diagram (P75)

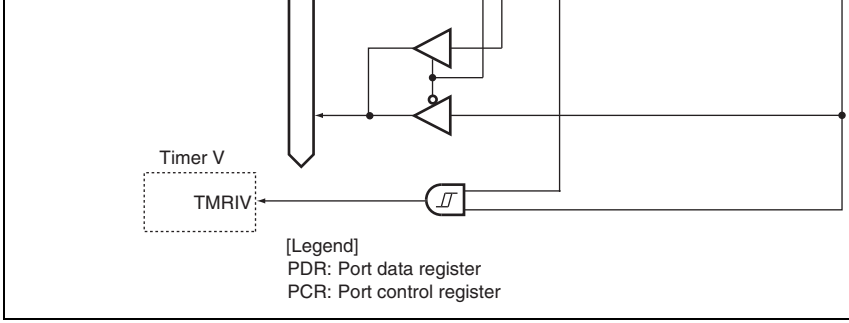
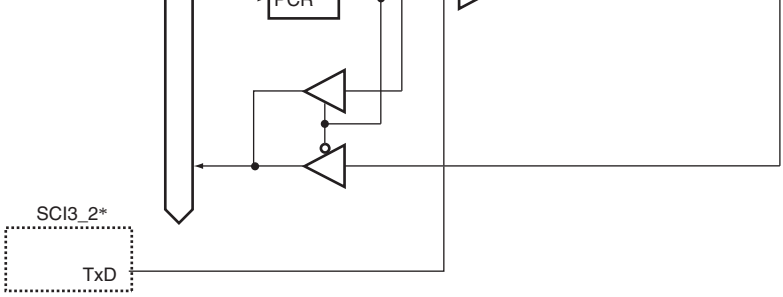
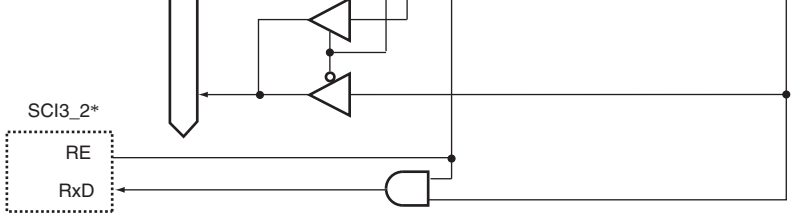


Figure B.15 Port 7 Block Diagram (P74)



[Legend]
PMR : Port mode register
PDR : Port data register
PCR : Port control register
Note: The H8/36037 Group does not have the SCI3_2.

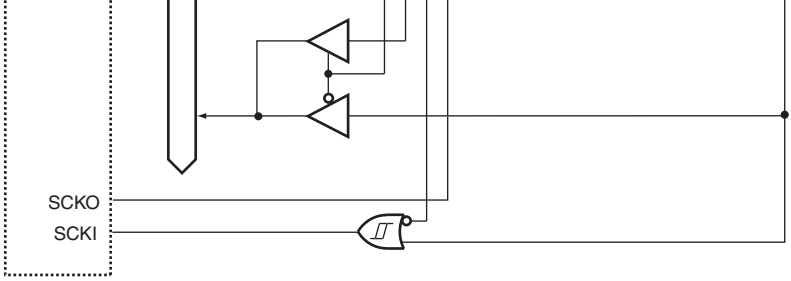
Figure B.16 Port 7 Block Diagram (P72)



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

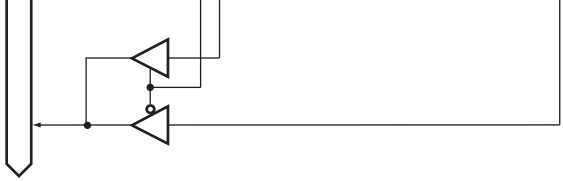
Note: The H8/36037 Group does not have the SCI3_2.

Figure B.17 Port 7 Block Diagram (P71)



[Legend]
PDR : Port data register
PCR : Port control register
Note: The H8/36037 Group does not have the SCI3_2.

Figure B.18 Port 7 Block Diagram (P70)



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

Figure B.19 Port 8 Block Diagram (P87 to P85)

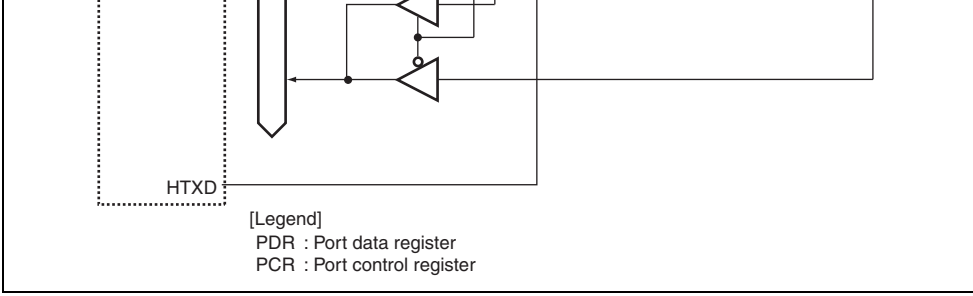


Figure B.20 Port 9 Block Diagram (P97)

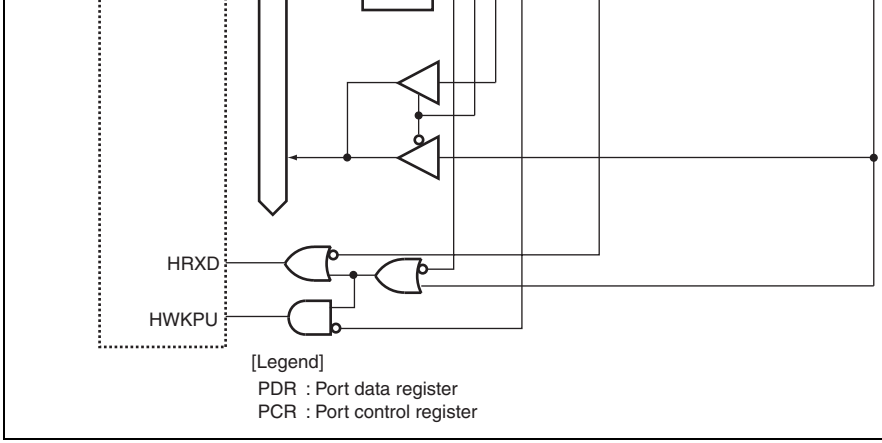
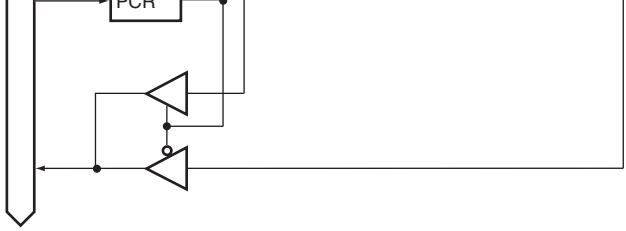


Figure B.21 Port 9 Block Diagram (P96)



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

Figure B.22 Port 9 Block Diagram (P94, P95)

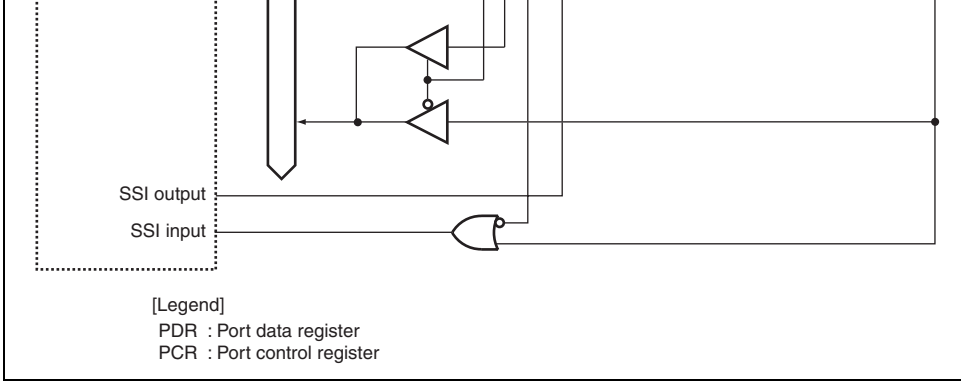


Figure B.23 Port 9 Block Diagram (P93)

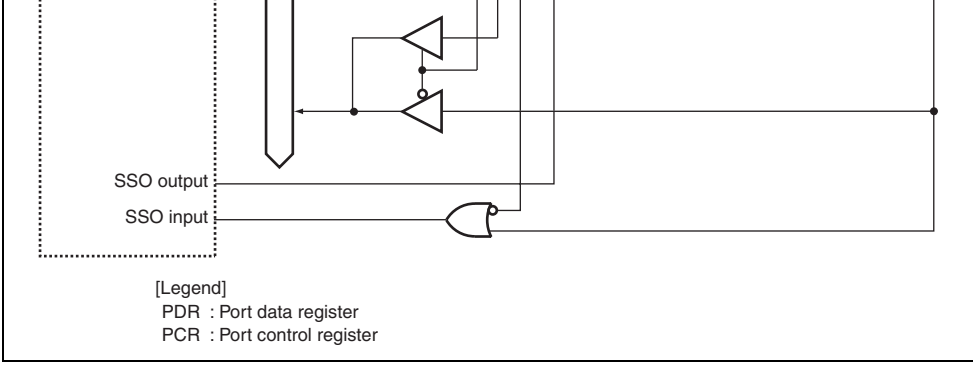


Figure B.24 Port 9 Block Diagram (P92)

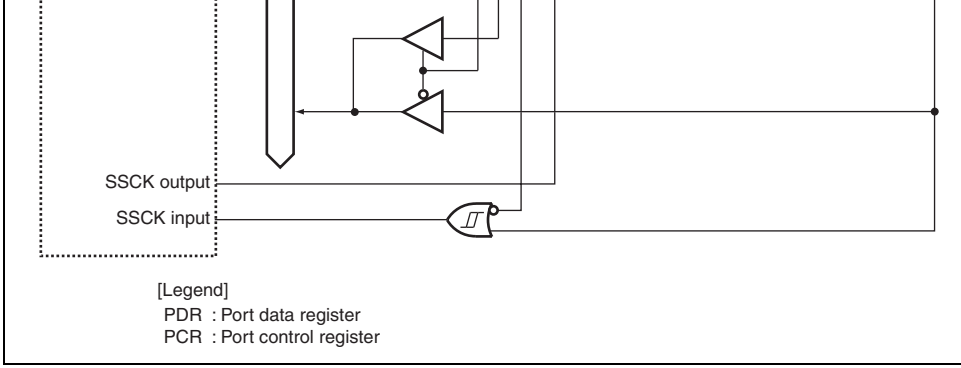


Figure B.25 Port 9 Block Diagram (P91)

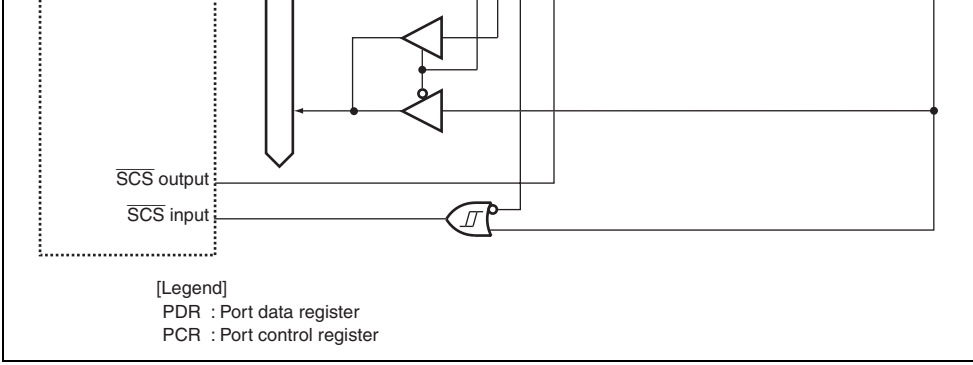


Figure B.26 Port 9 Block Diagram (P90)

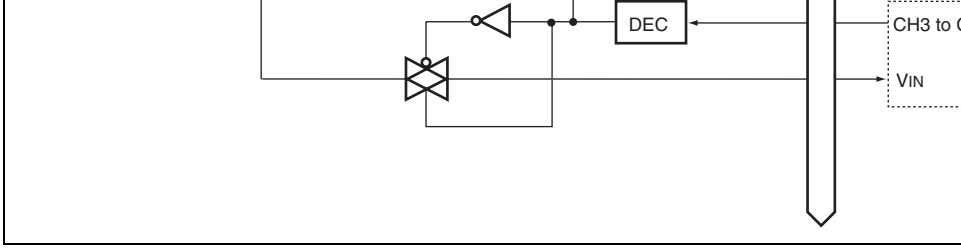


Figure B.27 Port B Block Diagram (PB7 to PB0)

	impedance			impedance		
P76 to P74, P72 to P70	High impedance	Retained	Retained	High impedance	Functioning	Fun
P87 to P85	High impedance	Retained	Retained	High impedance	Functioning	Fun
P97 to P90	High impedance	Retained	Retained	High impedance	Functioning	Fun
PB7 to PB0	High impedance	High impedance	High impedance	High impedance	High impedance	Higl imp

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

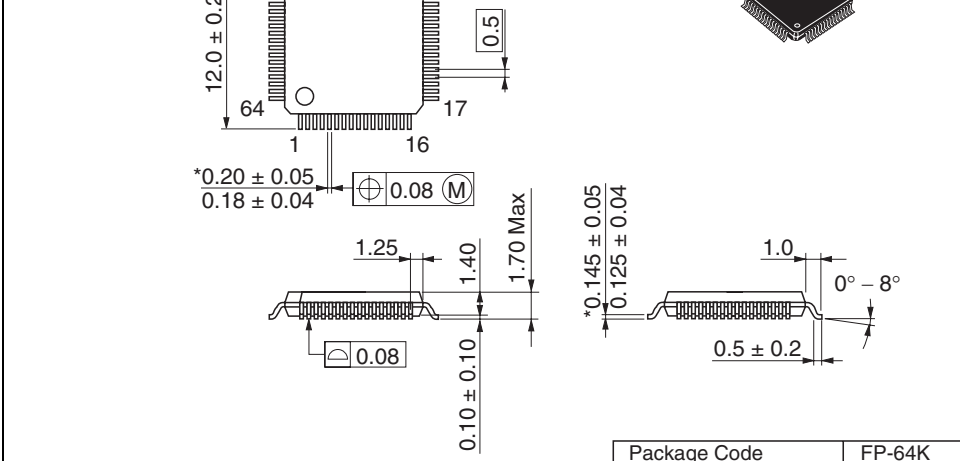
		Product with POR & LVDC	HD64336057G(***)H	HD6433605
H8/36054	Flash memory version	Standard product	HD64F36054H	HD64F3605
		Product with POR & LVDC	HD64F36054GH	HD64F3605
	Masked ROM version	Standard product	HD64336054(***)H	HD6433605
		Product with POR & LVDC	HD64336054G(***)H	HD6433605
H8/36037	Flash memory version	Standard product	HD64F36037H	HD64F3603
		Product with POR & LVDC	HD64F36037GH	HD64F3603
	Masked ROM version	Standard product	HD64336037(***)H	HD6433603
		Product with POR & LVDC	HD64336037G(***)H	HD6433603
H8/36036	Masked ROM version	Standard product	HD64336036(***)H	HD6433603
		Product with POR & LVDC	HD64336036G(***)H	HD6433603
H8/36035	Masked ROM version	Standard product	HD64336035(***)H	HD6433603
		Product with POR & LVDC	HD64336035G(***)H	HD6433603

H8/36033	Masked ROM version	Standard product	HD64336033(***)H	HD64336033(
		Product with POR & LVDC	HD64336033G(***)H	HD64336033(
H8/36032	Masked ROM version	Standard product	HD64336032(***)H	HD64336032(
		Product with POR & LVDC	HD64336032G(***)H	HD64336032(

[Legend]

(***): ROM code

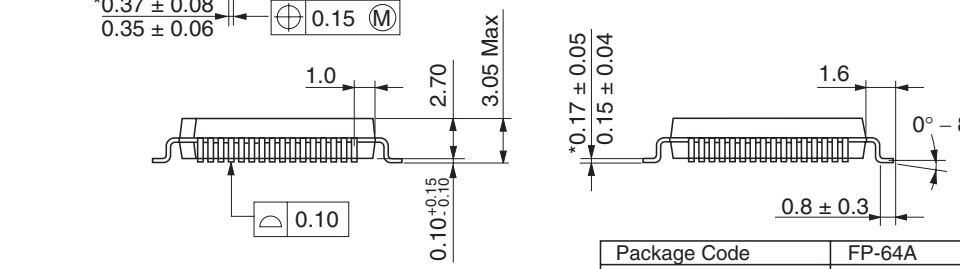
POR & LVDC: Power-on reset and low-voltage detection circuits



Package Code	FP-64K
JEDEC	–
EIAJ	Conforms
Mass (reference value)	0.3 g

*Dimension including the plating thickness
Base material dimension

Figure D.1 FP-64K Package Dimensions



Package Code	FP-64A
JEDEC	–
EIAJ	Conforms
Mass (reference value)	1.2 g

Figure D.2 FP-64A Package Dimensions

use these pins, additional hardware must b
provided on the user board.

3. Area H'D000 to H'DFFF is used by the E7
is not available to the user.
4. Area H'F780 to H'FB7F must on no accoun
accessed.
5. When the E7 or E8 is used, address break
set as either available to the user or for use
E7 or E8. If address breaks are set as bein
the E7 or E8, the address break control reg
must not be accessed.
6. When the E7 or E8 is used, $\overline{\text{NMI}}$ is an input
pin (open-drain in output mode), P85 and P
input pins, and P86 is an output pin.

Section 8 RAM — 109

Added

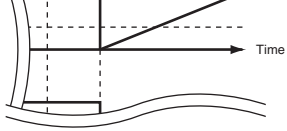
Note: * When the E7 or E8 is used, area H'F7
H'FB7F must not be accessed.

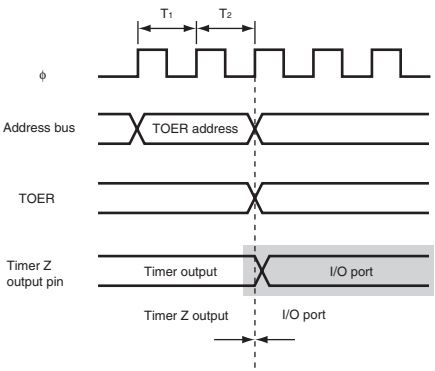
12.3.2 Timer Mode Register (TMDR) — 171

Amended

Bit	Bit Name	Description
0	SYNC	Timer Synchronization 0: TCNT_1 and TCNT_0 o as a different timer 1: TCNT_1 and TCNT_0 a synchronized TCNT_1 and TCNT_0 c pre-set or cleared synch

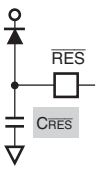
---



12.4.4 Synchronous Operation	199	Added Figure 12.20 shows an example of synchronou operation. In this example, …. set for the chan counter clearing source. In addition, the same clock has been set as the counter input clock f channel 0 and channel 1. Two-phase PWM wa are….
Figure 12.44 Example of Output Disable Timing of Timer Z by Writing to TOER	229	Amended 

13.2.1 Timer Control/Status Register WD (TCSRWD)	246	Amended

Bit	Bit Name	Description
4	TCSRWE	Timer Control/Status Reg Write Enable The WDON and WRST b be written when the TCSI is set to 1. When writing data to this value for bit 5 must be 0.

16.5 Usage Note	378	Added
18.3.1 A/D Data Registers A to D (ADDRA to ADDRD)	394	Amended …. The temporary register contents are transf from the ADDR when the upper byte data is r Therefore, byte access to ADDR should be do reading the upper byte first then the lower one access is also possible. ADDR is initialized to
Figure 19.1 Block Diagram of Power-On Reset Circuit and Low-Voltage Detection Circuit	404	Amended 

		Sleep mode 2 Only timers operate(

Figure 22.8 SSU Input/Output Timing (Four-Line Bus Communication Mode, Master, CPHS = 1) to Figure 22.11 SSU Input/Output Timing (Four-Line Bus Communication Mode, Slave, CPHS = 0)	481 to 484	t_{OH} deleted

Table A.1 Instruction Set

2. Arithmetic Instructions

491

Amended

Mnemonic		Operand Size	Condition Code						No. of States*1	
			I	H	N	Z	V	C	Normal	Advanced
DAA	DAA Rd	B	—	*	↕	↕	*	↕	2	

C

E

I

L

V

W

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8/36057 Group, H8/36037 Group

Publication Date: Rev.1.00, Jun. 27, 2003
Rev.4.00, Mar. 15, 2006
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

RENESAS SALES OFFICES

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

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

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

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

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

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

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

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

16 H8/36057Group, H8/36037Group Hardware Manual

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

REJ09B0026-0400

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

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

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

Other Similar products are found below :

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