To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User's Manual

16

H8/38799 Group

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer
H8 Family / H8/300H Super Low Power Series

H8/38799F
H8/38799
H8/38798

Renesas Electronics
www.renesas.com

Rev.1.00 2007.07

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

General Precautions on Handling of Product

1. Treatment of NC Pins

Note: Do not connect anything to the NC pins.
The NC (not connected) pins are either not connected to any of the internal circuitry or are used as test pins or to reduce noise. If something is connected to the NC pins, the operation of the LSI is not guaranteed.

2. Treatment of Unused Input Pins

Note: Fix all unused input pins to high or low level.
Generally, the input pins of CMOS products are high-impedance input pins. If unused pins are in their open states, intermediate levels are induced by noise in the vicinity, a pass-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 throughout the chip and a low level is input on the reset pin. During the period where the states are undefined, the register settings and the output state of each pin are also undefined. Design your system so that it does not malfunction because of processing while it is in this undefined state. For those products which have a reset function, reset the LSI immediately 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 registers may have been be allocated to these addresses. Do not access these registers; the system's operation is not guaranteed if they are accessed.

Configuration of This Manual

This manual comprises the following items:

1. General Precautions on Handling of Product
2. Configuration of This Manual
3. Preface
4. Contents
5. Overview
6. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

   The configuration of the functional description of each module differs according to the 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. Each section includes notes in relation to the descriptions given, and usage notes are given, as required, as the 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 versions. This does not include all of the revised contents. For details, see the actual locations in this manual.

11. Index

Preface

H8/38799 Group is single-chip microcontrollers made up of the high-speed H8/300H CPU employing Renesas Technology original architecture as their cores, and the peripheral functions required to configure a system. The H8/300H CPU has an instruction set that is compatible with the H8/300 CPU.

Target Users: This manual was written for users who will be using the H8/38799 Group in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logical circuits, and microcontrollers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8/38799 Group to the target users.
Refer to the H8/300H Series Software Manual for a detailed description of the 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 into parts on the CPU, system control functions, peripheral functions, and electrical characteristics.
- 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 on the register. The addresses, bits, and initial values of the registers are summarized in section 24, List of Registers.

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

Notes:

When using an on-chip emulator (E8) for H8/38799 program development and debugging, the following restrictions must be noted.

1. The $\overline{\text{NMI}}$ pin is reserved for the E8, and cannot be used.
2. Pins P16, P36, and P37 cannot be used. In order to use these pins, additional hardware must be provided on the user board.
3. Area H'020000 to H'020FFF must on no account be accessed.
4. Area H′FFA000 to H′FFA7FF must on no account be accessed.
5. When the E8 is used, address breaks can be set as either available to the user or for use by the E8. If address breaks are set as being used by the E8, the address break control registers must not be accessed.
6. When the E8 is used, $\overline{\text{NMI}}$ is an input pin, P16 and P36 are input pins, and P37 is an output pin.
7. When on-board programming/erasing is performed in boot mode, the SCI3_1 (P41/RXD and P42/TXD) is used.
8. When using the E8, set the FROMCKSTP bit in clock halt register 1 to 1.

Related Manuals: The latest versions of all related manuals are available from our web site. Please ensure you have the latest versions of all documents you require (http://www.renesas.com/)

H8/38799 Group manuals:

Document Title	Document No.
H8/38799 Group Hardware Manual	This manual
H8/300H Series Software Manual	REJ09B0213

User's manuals for development tools:

Document Title	Document No.
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	REJ10B0058
H8S, H8/300 Series Simulator/Debugger User's Manual	ADE-702-037
H8S, H8/300 Series High-performance Embedded Workshop 3 Tutorial	REJ10B0024
H8S, H8/300 Series High-performance Embedded Workshop 3 User's Manual	REJ10B0026

Application notes:

Document Title	Document No.
H8S, H8/300 Series C/C++ Compiler Package Application Note	REJ05B0464

All trademarks and registered trademarks are the property of their respective owners.

Contents

Figures

Section 6 Power-Down Modes

Section 7 ROM

Section 9 I/O Ports

Section 10 Realtime Clock (RTC)

Section 15 Asynchronous Event Counter (AEC)

Section 16 Watchdog Timer

Section 17 Serial Communications Interface 3 (SCI3, IrDA)

Section 18 Serial Communication Interface 4 (SCI4)

Section 19 14-Bit PWM

Section 20 A/D Converter

Section 21 I^2C Bus Interface 2 (IIC2)

Section 22 Power-On Reset Circuit

Section 23 Address Break

Section 25 Electrical Characteristics

Appendix

Tables

Section 7 ROM

Section 10 Realtime Clock (RTC)

Section 11 Timer C

Section 12 Timer F

Section 13 Timer F

Section 14 16-Bit Timer Pulse Unit (TPU)

Appendix

Section 1 Overview

1.1 Features

- High-speed H8/300H central processing unit with an internal 16-bit architecture
  Upward-compatible with H8/300 CPU on an object level
  Sixteen 16-bit general registers
  62 basic instructions
- Various peripheral functions
  RTC (can be used as a free-running counter)
  Asynchronous event counter (AEC)
  Timer C
  Timer F
  Timer G
  16-bit timer pulse unit (TPU)
  14-bit PWM
  Watchdog timer
  SCI (Asynchronous or clock synchronous serial communications interface)
  I^2C bus interface (conforms to the I^2C bus interface format that is advocated by Philips Electronics)
  10-bit A/D converter

- On-chip memory

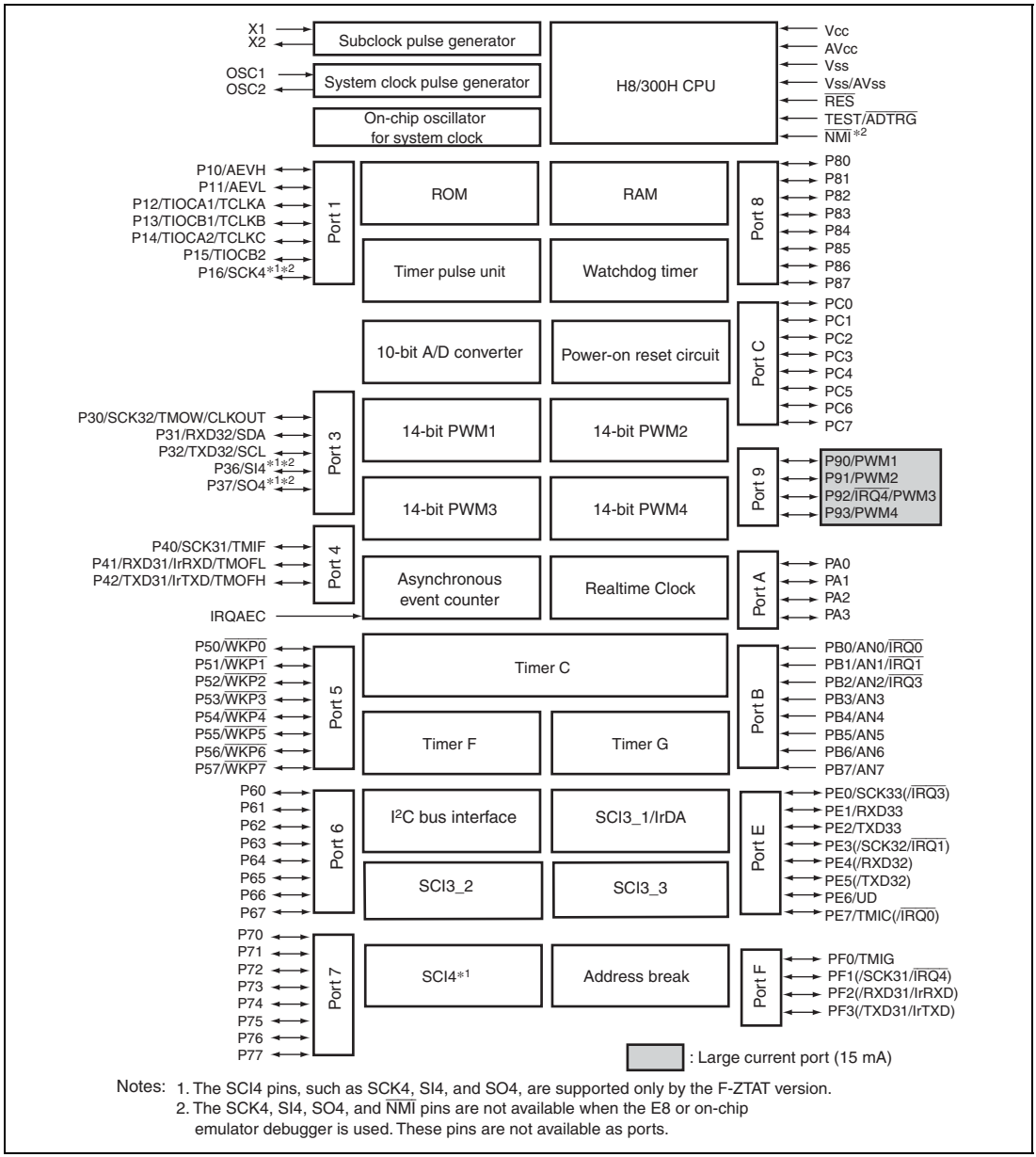
Product Classification		Model	ROM	RAM
Flash memory version (F-ZTAT™ version)	H8/38799F	HD64F38799	128 Kbytes	4 Kbytes
Masked ROM version	H8/38799	HD64338799	128 Kbytes	4 Kbytes
	H8/38798	HD64338798	96 Kbytes	2 Kbytes

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

- General I/O ports
  I/O pins: 75 I/O pins, including 4 large current ports (I_{OL} = 15 mA, @V_{OL} = 1.0 V)
  Input-only pins: 8 input pins
- Supports various power-down states
- Compact package

Package	Code	Body Size	Pin Pitch	Remarks
P-LQFP-100	PLQP0100KB-A	14 × 14 mm	0.5 mm	

1.2 Internal Block Diagram



Notes: 1. The SCI4 pins, such as SCK4, SI4, and SO4, are supported only by the F-ZTAT version.
2. The SCK4, SI4, SO4, and NMI pins are not available when the E8 or on-chip emulator debugger is used. These pins are not available as ports.

Figure 1.1 Internal Block Diagram of H8/38799 Group

1.3 Pin Assignment

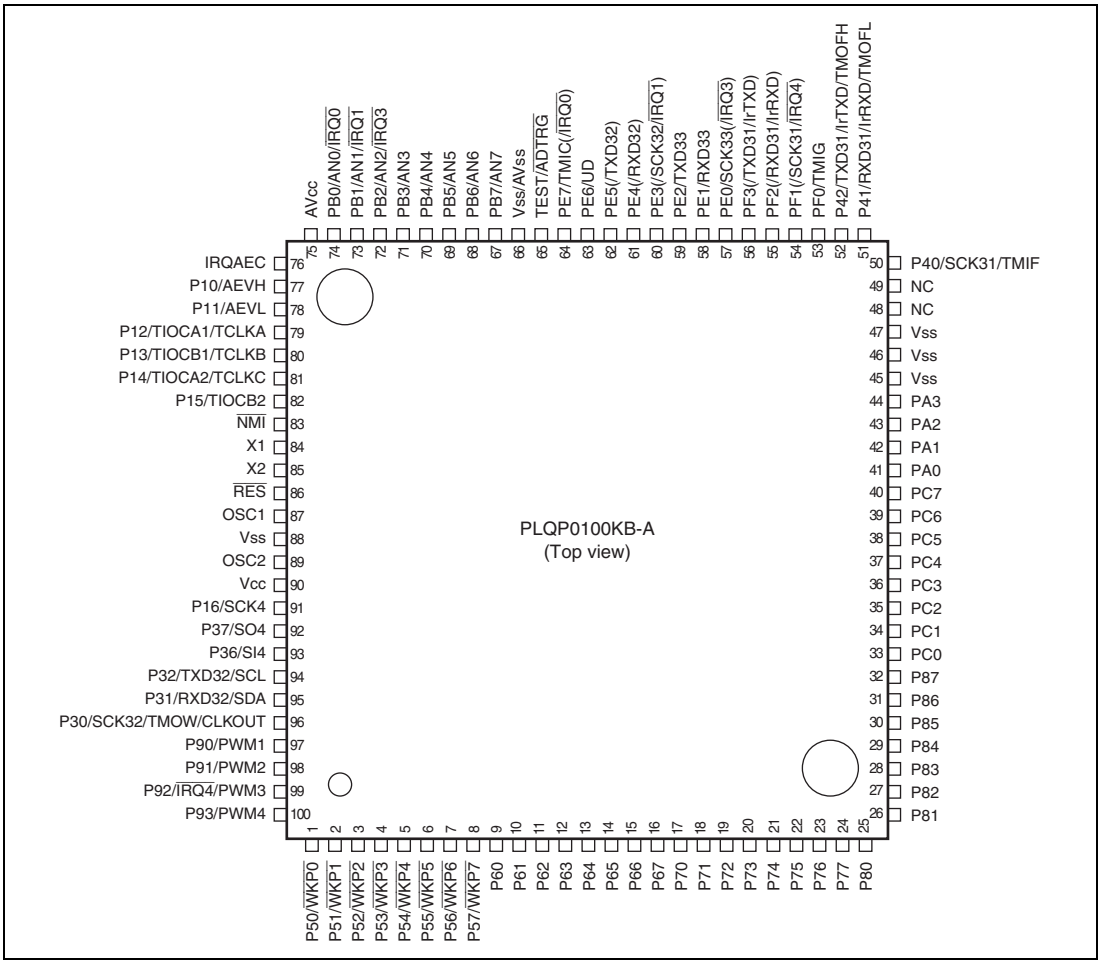


Figure 1.2 Pin Assignment of H8/38799 Group (PLQP0100KB-A)

1.4 Pin Functions

Table 1.1 Pin Functions

Type	Symbol	Pin No.	I/O	Functions
Power supply pins	Vcc	90	Input	Power supply pin. Connect this pin to the system power supply.
	Vss	88, 66, 45, 46, 47 (= AVss)	Input	Ground pins. Connect these pins to the system power supply (0 V).
	AVcc	75	Input	Analog power supply pin for the A/D converter. When the A/D converter is not used, connect this pin to the system power supply.
	AVss	66 (= Vss)	Input	Ground pin for the A/D converter. Connect this pin to the system power supply (0 V).
Clock pins	OSC1	87	Input	These pins connect crystal or ceramic resonator for the system clock, or can be used to input an external clock. See section 5, Clock Pulse Generator, for a connection.
	OSC2	89	Output	
	X1	84	Input	These pins connect a 32.768- or 38.4-kHz crystal resonator for the subclock. See section 5, Clock Pulse Generator, for a connection.
	X2	85	Output	
	CLKOUT	96	Output	Pin for supplying the system clock to external devices
System control	$\overline{\text{RES}}$	86	Input	Reset pin. The power-on reset circuit is incorporated. When externally driven low, the chip is reset.
	TEST	65	Input	Test pin. Also used as the $\overline{\text{ADTRG}}$ pin. When this pin is not used as the $\overline{\text{ADTRG}}$ pin, users cannot use this pin. Connect this pin to Vss. When this pin is used as the $\overline{\text{ADTRG}}$ pin, see section 20.4.2, External Trigger Input Timing.

Type	Symbol	Pin No.	I/O	Functions
Interrupt pins	$\overline{\text{NMI}}$	83	Input	NMI interrupt request pin. Non-maskable interrupt request input pin.
	$\overline{\text{IRQ0}}$	74	Input	External interrupt request input pins. Sensing of rising or falling edges selectable.
	($\overline{\text{IRQ0}}$)	(64)		
	$\overline{\text{IRQ1}}$	73	Input	
	($\overline{\text{IRQ1}}$)	(60)		
	$\overline{\text{IRQ3}}$	72	Input	
	($\overline{\text{IRQ3}}$)	(57)		
	$\overline{\text{IRQ4}}$	99	Input	
	($\overline{\text{IRQ4}}$)	(54)		
	IRQAEC	76	Input	Interrupt input pin for the asynchronous event counter. This pin enables the asynchronous event input. In the masked ROM version, this pin turns on/off the on-chip oscillator for the system clock during a reset.
	$\overline{\text{WKP0}}$ to $\overline{\text{WKP7}}$	1 to 8	Input	Wakeup interrupt request input pins. Sensing of rising or falling edges selectable.
16-bit timer pulse unit (TPU)	TIOCA1	79	I/O	Pin for the TGR1A input capture input or output compare output, or PWM output.
	TIOCB1	80	I/O	Pin for the TGR1B input capture input or output compare output, or PWM output.
	TIOCA2	81	I/O	Pin for the TGR2A input capture input or output compare output, or PWM output.
	TIOCB2	82	I/O	Pin for the TGR2B input capture input or output compare output, or PWM output.
	TCLKA	79	Input	External clock input pins.
	TCLKB	80	Input	
	TCLKC	81	Input	
Timer C	TMIC	64	Input	Event input pin for the timer C counter (TCC).
	UD	63	Input	Selects whether the TCC counts up or down. TCC counts up when the UD pin input is low, and counts down when the input is high.

Type	Symbol	Pin No.	I/O	Functions
Timer F	TMIF	50	Input	Event input pin for input to the timer F counter.
	TMOFL	51	Output	Output pin for waveforms generated by the timer FL output compare function.
	TMOFH	52	Output	Output pin for waveforms generated by the timer FH output compare function.
Timer G	TMIG	53	Input	Pin for the timer G input capture input
Asynch-ronous event counter (AEC)	AEVL	78	Input	Event input pins for input to the asynchronous event counter.
	AEVH	77	Input	
RTC	TMOW	96	Output	Divided clock output pin for the RTC.
14-bit PWM	PWM1	97	Output	Output pins for waveforms generated by the 14-bit PWM in PWM channels 1, 2, 3, and 4.
	PWM2	98	Output	
	PWM3	99	Output	
	PWM4	100	Output	
Serial commu-nications interface 4 (SCI4) (F-ZTAT version only)	SCK4	91	I/O	Transfer clock pin for SCI4 data transmission/reception. When the E8 or on-chip emulator debugger is used, this pin is not available.
	SI4	93	Input	SCI4 data input pin. When the E8 or on-chip emulator debugger is used, this pin is not available.
	SO4	92	Output	SCI4 data output pin. When the E8 or on-chip emulator debugger is used, this pin is not available.
Serial commu-nications interface 3 (SCI3)	SCK31	50	I/O	SCI3_1 clock I/O pins.
	(SCK31)	54		
	RXD31/IrRXD	51	Input	SCI3_1 data input pins or data input pins for the IrDA format.
	(RXD31/IrRXD)	55		
	TXD31/IrTXD	52	Output	SCI3_1 data output pins or data output pins for the IrDA format.
	(TXD31/IrTXD)	56		

Type	Symbol	Pin No.	I/O	Functions
Serial communications interface 3 (SCI3)	SCK32	96	I/O	SCI3_2 clock I/O pins.
	(SCK32)	60		
	RXD32	95	Input	SCI3_2 data input pins.
	(RXD32)	61		
	TXD32	94	Output	SCI3_2 data output pins.
	(TXD32)	62		
	SCK33	57	I/O	SCI3_3 clock I/O pin.
	RXD33	58	Input	SCI3_3 data input pin.
	TXD33	59	Output	SCI3_3 data output pin.
A/D converter	AN0 to AN7	74 to 67	Input	Analog data input pins for the A/D converter.
	$\overline{\text{ADTRG}}$	65	Input	External trigger input pin for the A/D converter.
I^2C bus interface 2 (IIC2)	SDA	95	I/O	IIC data I/O pin.
	SCL	94	I/O	IIC clock I/O pin.
I/O ports	P10 to P16	77 to 82, 91	I/O	7-bit I/O pins. Input or output can be designated for each bit by means of the port control register 1 (PCR1).
	P30 to P32, P36, P37	96 to 92	I/O	5-bit I/O pins. Input or output can be designated for each bit by means of the port control register 3 (PCR3).
	P40 to P42	50 to 52	I/O	3-bit I/O pins. Input or output can be designated for each bit by means of the port control register 4 (PCR4).
	P50 to P57	1 to 8	I/O	8-bit I/O pins. Input or output can be designated for each bit by means of the port control register 5 (PCR5).
	P60 to P67	9 to 16	I/O	8-bit I/O pins. Input or output can be designated for each bit by means of the port control register 6 (PCR6).
	P70 to P77	17 to 24	I/O	8-bit I/O pins. Input or output can be designated for each bit by means of the port control register 7 (PCR7).
	P80 to P87	25 to 32	I/O	8-bit I/O pins. Input or output can be designated for each bit by means of the port control register 8 (PCR8).
	P90 to P93	97 to 100	I/O	4-bit I/O pins. Input or output can be designated for each bit by means of the port control register 9 (PCR9).

Type	Symbol	Pin No.	I/O	Functions
I/O ports	PA0 to PA3	41 to 44	I/O	4-bit I/O pins. Input or output can be designated for each bit by means of the port control register A (PCRA).
	PB0 to PB7	74 to 67	Input	8-bit input-only pins
	PC0 to PC7	33 to 40	I/O	8-bit I/O pins. Input or output can be designated for each bit by means of the port control register C (PCRC).
	PE0 to PE7	57 to 64	I/O	8-bit I/O pins. Input or output can be designated for each bit by means of the port control register E (PCRE).
	PF0 to PF3	53 to 56	I/O	4-bit I/O pins. Input or output can be designated for each bit by means of the port control register F (PCRF).

Section 2 CPU

This LSI has an H8/300H CPU with an internal 32-bit architecture that is upward-compatible with the H8/300 CPU. The H8/300H can handle a 16-Mbyte linear address space and is ideal for realtime control.

- Upward-compatible with H8/300 CPUs
  Can execute H8/300 CPUs object programs
  Additional eight 16-bit extended registers
  32-bit transfer and arithmetic and logic instructions are added
  Signed multiply and divide instructions are added.
- General-register architecture
  Sixteen 16-bit general registers also usable as sixteen 8-bit registers and eight 16-bit 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]
- 16-Mbyte linear address space
- High-speed operation
  All frequently-used instructions execute in two or four states
  8/16/32-bit register-register add/subtract : 2 state
  8 × 8-bit register-register multiply : 14 states
  16 ÷ 8-bit register-register divide : 14 states
  16 × 16-bit register-register multiply : 22 states
  32 ÷ 16-bit register-register divide : 22 states

- Two types of CPU operating modes
  Normal mode
  Advanced mode

Note: Normal mode cannot be used in this LSI.

- Power-down state
  Transition to power-down state by SLEEP instruction

2.1 Address Space and Memory Map

The address space of this LSI is 16 Mbytes, which includes the program area and the data area.

Figure 2.1 shows the memory map.

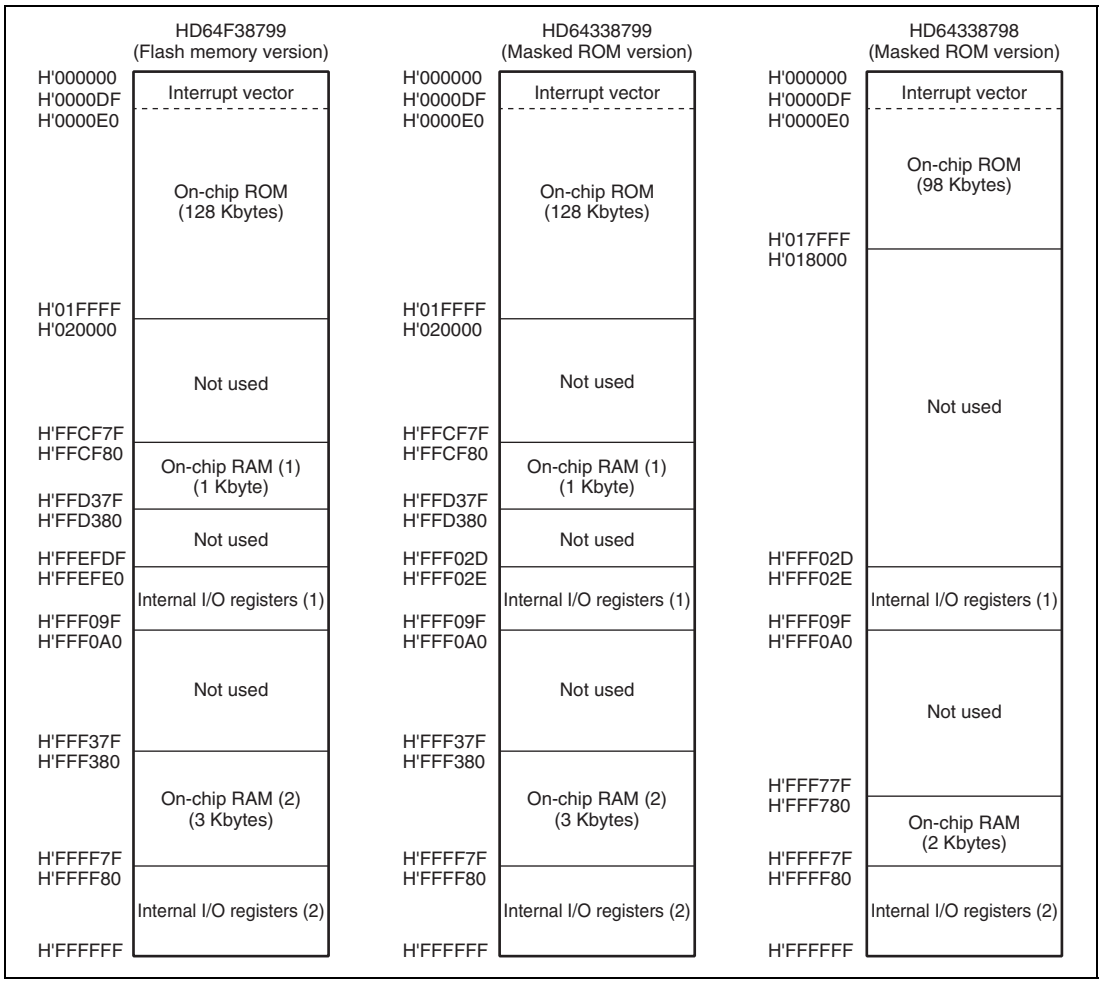


Figure 2.1 Memory Map

2.2 Register Configuration

The H8/300H CPU has the internal registers shown in figure 2.2. There are two types of registers; general registers and control registers. The control registers are a 24-bit program counter (PC), and an 8-bit condition-code register (CCR).

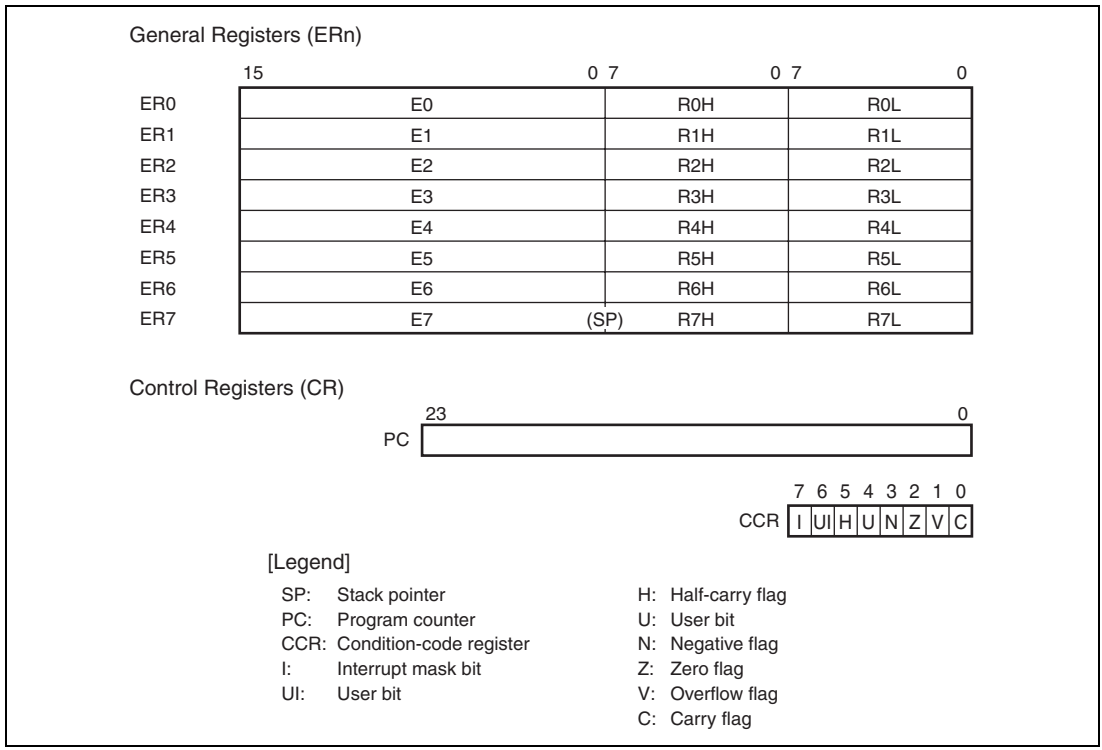


Figure 2.2 CPU Registers

2.2.1 General Registers

The H8/300H CPU has eight 32-bit general registers. These general registers are all functionally identical and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. Figure 2.3 illustrates the usage of the general registers. When the general registers are used as 32-bit registers or address registers, they are designated by the letters ER (ER0 to ER7).

The ER registers divide into 16-bit general registers designated by the letters E (E0 to E7) and R (R0 to R7). These registers are functionally equivalent, providing a maximum of sixteen 16-bit registers. The E registers (E0 to E7) are also referred to as extended registers.

The R registers divide into 8-bit registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing a maximum of sixteen 8-bit registers.

The usage of each register can be selected independently.

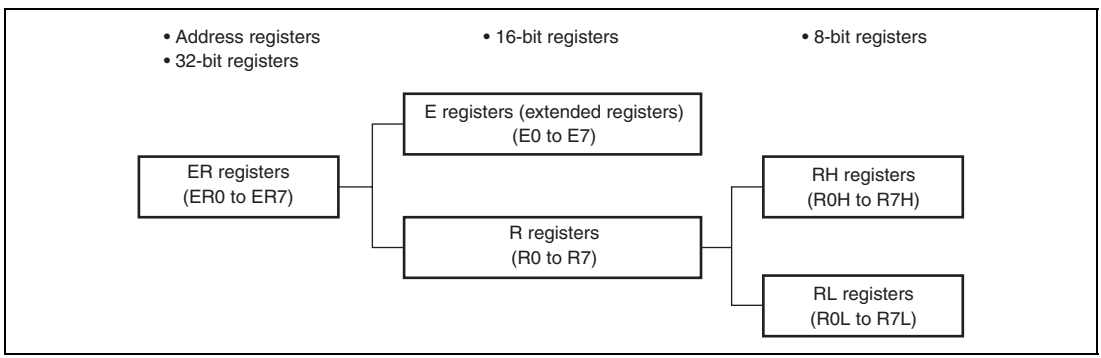


Figure 2.3 Usage of General Registers

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

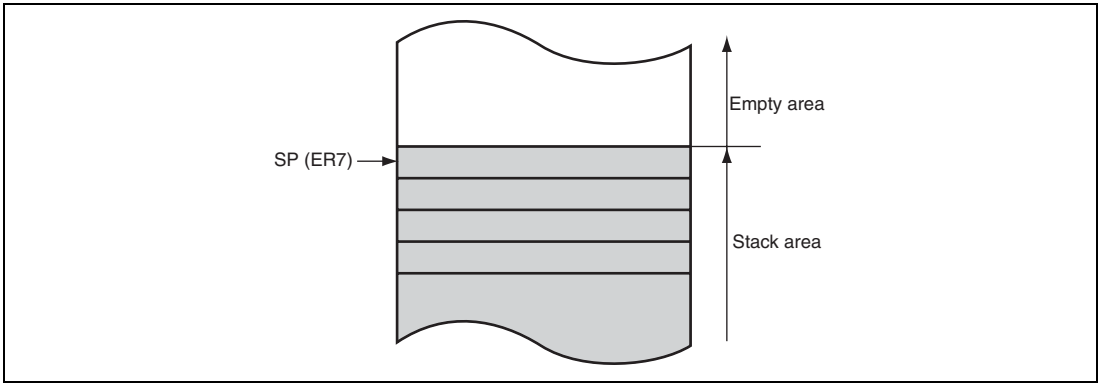


Figure 2.4 Relationship between Stack Pointer and Stack Area

2.2.2 Program Counter (PC)

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

2.2.3 Condition-Code Register (CCR)

This 8-bit register contains internal CPU status information, including an interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags. The I bit is initialized to 1 by reset exception-handling sequence, but other bits are not initialized.

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

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

Bit	Bit Name	Initial Value	R/W	Description
7	I	1	R/W	Interrupt Mask Bit Masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting. The I bit is set to 1 at the start of an exception-handling sequence.
6	UI	Undefined	R/W	User Bit Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.
5	H	Undefined	R/W	Half-Carry Flag When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.
4	U	Undefined	R/W	User Bit Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit of data as a sign bit.
2	Z	Undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.
1	V	Undefined	R/W	Overflow Flag Set to 1 when an arithmetic overflow occurs, and 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 carry The carry flag is also used as a bit accumulator by bit manipulation instructions.

2.3 Data Formats

The H8/300H CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit n (n = 0, 1, 2, …, 7) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

2.3.1 General Register Data Formats

Figure 2.5 shows the data formats in general registers.

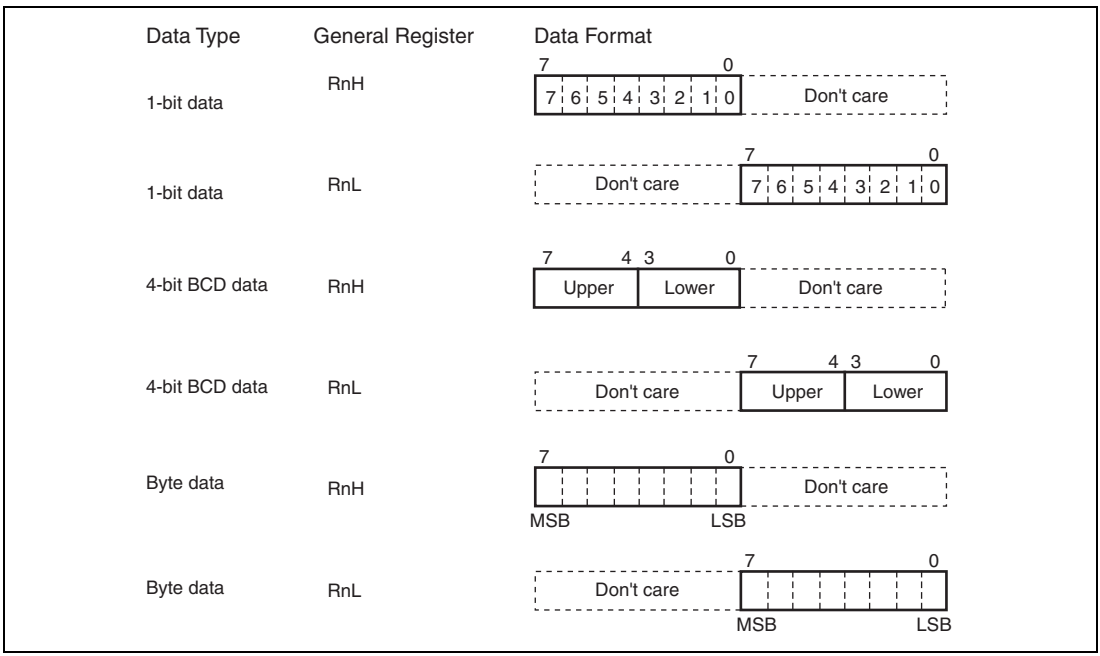


Figure 2.5 General Register Data Formats (1)

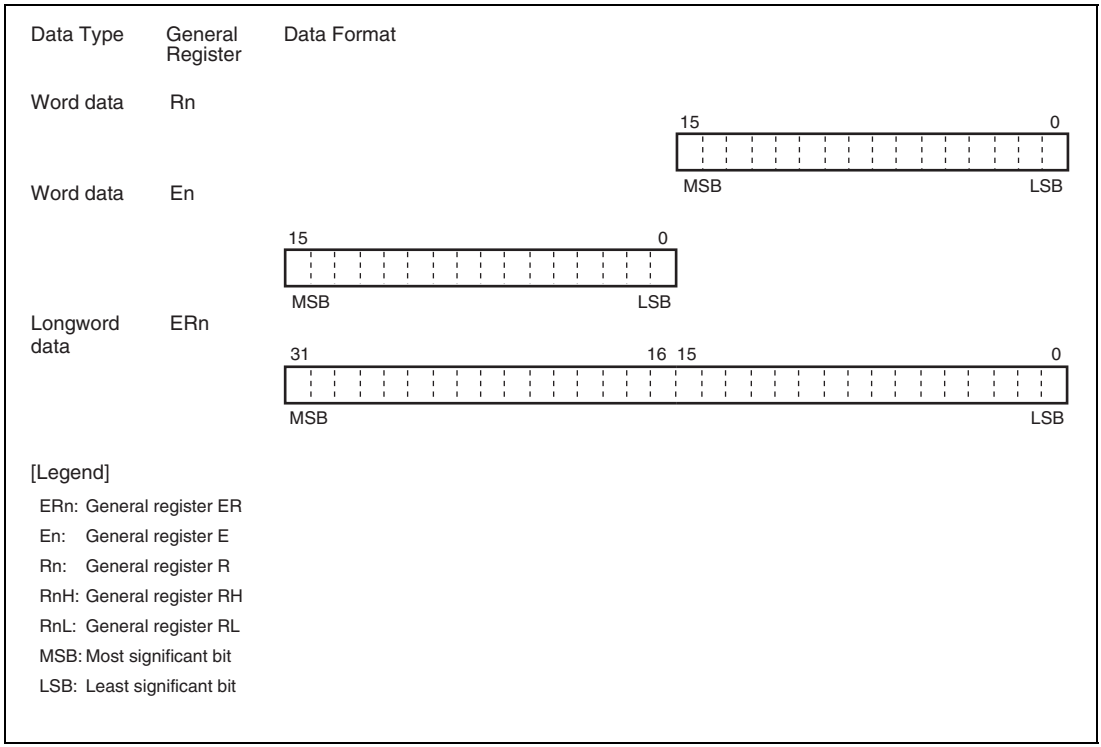


Figure 2.5 General Register Data Formats (2)

2.3.2 Memory Data Formats

Figure 2.6 shows the data formats in memory. The H8/300H CPU can access word data and longword data in memory, however word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, an address error does not occur, however the least significant bit of the address is regarded as 0, so access begins the preceding address. This also applies to instruction fetches.

When ER7 (SP) is used as an address register to access the stack area, the operand size should be word or longword.

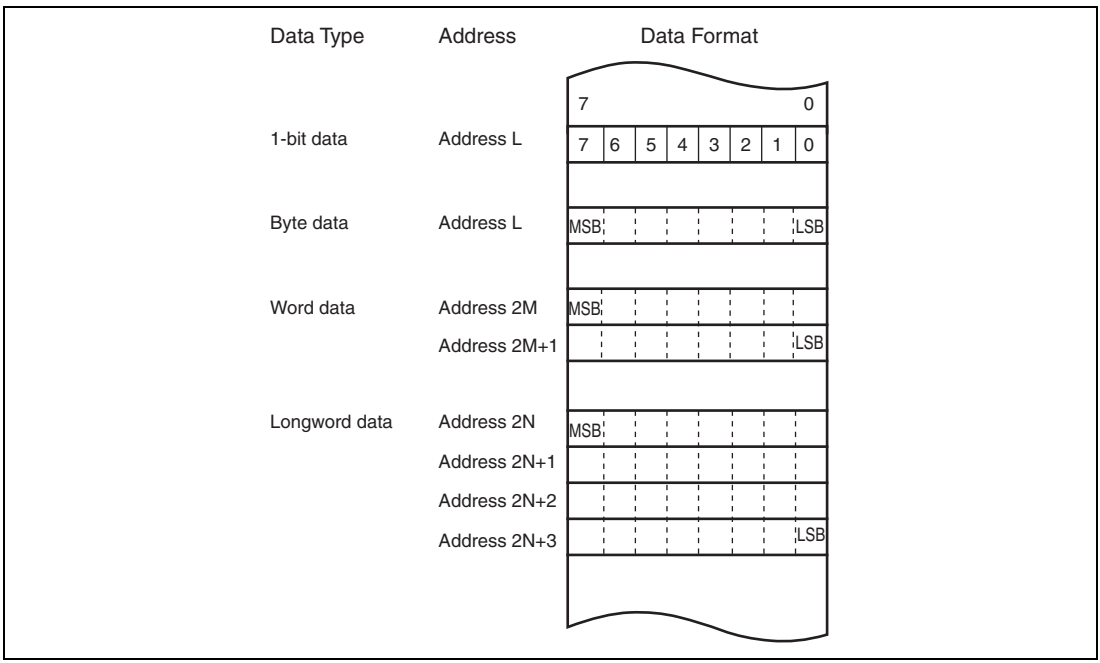


Figure 2.6 Memory Data Formats

2.4 Instruction Set

2.4.1 Table of Instructions Classified by Function

The H8/300H CPU has 62 instructions. Tables 2.2 to 2.9 summarize the instructions in each functional category. The notation used in tables 2.2 to 2.9 is defined in table 2.1.

Table 2.1 Operation Notation

Symbol	Description
Rd	General register (destination)*
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)

Symbol	Description
:3/:8/:16/:24	3-, 8-, 16-, or 24-bit length

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers/address register (ER0 to ER7).

Table 2.2 Data Transfer Instructions

Instruction	Size*	Function
MOV	B/W/L	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
MOVFPE	B	(EAs) → Rd Cannot be used in this LSI.
MOVTPE	B	Rs → (EAs) Cannot be used in this LSI.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, ERn.
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @–SP. PUSH.L ERn is identical to MOV.L ERn, @–SP.

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

Table 2.3 Arithmetic Operations Instructions (1)

Instruction	Size*	Function
ADD SUB	B/W/L	Rd ± Rs → Rd, Rd ± #IMM → Rd Performs addition or subtraction on data in two general registers, or on immediate data and data in a general register (immediate byte data cannot be subtracted from byte data in a general register. Use the SUBX or ADD instruction.)
ADDX SUBX	B	Rd ± Rs ± C → Rd, Rd ± #IMM ± C → Rd Performs addition or subtraction with carry on byte data in two general registers, or on immediate data and data in a general register.
INC DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (Byte operands can be incremented or decremented by 1 only.)
ADDS SUBS	L	Rd ± 1 → Rd, Rd ± 2 → Rd, Rd ± 4 → Rd Adds or subtracts the value 1, 2, or 4 to or from data in a 32-bit register.
DAA DAS	B	Rd (decimal adjust) → Rd Decimal-adjusts an addition or subtraction result in a general register by referring to the CCR to produce 4-bit BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general registers: either 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: either 8 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: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.

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

Table 2.3 Arithmetic Operations Instructions (2)

Instruction	Size*	Function
DIVXS	B/W	Rd ÷ Rs → Rd Performs signed division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.
CMP	B/W/L	Rd – Rs, Rd – #IMM Compares data in a general register with data in another general register or with immediate data, and sets CCR bits according to the result.
NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic complement) of data in a general register.
EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by padding with zeros on the left.
EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by extending the sign bit.

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

Table 2.4 Logic Operations Instructions

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

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

Table 2.5 Shift Instructions

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

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

Table 2.6 Bit Manipulation Instructions

Instruction	Size*	Function
BSET	B	1 → (<bit-No.> of <EAd>) Sets a specified bit in a general register or memory operand to 1. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BCLR	B	0 → (<bit-No.> of <EAd>) Clears a specified bit in a general register or memory operand to 0. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BNOT	B	¬ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operand and sets or clears the Z flag accordingly. The bit number is specified by 3-bit 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 memory 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 general register or memory operand and stores the result in the carry flag. 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 memory 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 general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C XORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIXOR	B	C ⊕ ¬ (<bit-No.> of <EAd>) → C XORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

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

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

Table 2.7 Branch Instructions

Instruction	Size	Function		
Bcc*	—	Branches to a specified address if a specified condition is true. The branching conditions are listed below.		
		Mnemonic	**Description**	**Condition**
		BRA(BT)	Always (true)	Always
		BRN(BF)	Never (false)	Never
		BHI	High	$C \vee Z = 0$
		BLS	Low or same	$C \vee Z = 1$
		BCC(BHS)	Carry clear (high or same)	$C = 0$
		BCS(BLO)	Carry set (low)	$C = 1$
		BNE	Not equal	$Z = 0$
		BEQ	Equal	$Z = 1$
		BVC	Overflow clear	$V = 0$
		BVS	Overflow set	$V = 1$
		BPL	Plus	$N = 0$
		BMI	Minus	$N = 1$
		BGE	Greater or equal	$N \oplus V = 0$
		BLT	Less than	$N \oplus V = 1$
		BGT	Greater than	$Z \vee (N \oplus V) = 0$
		BLE	Less or equal	$Z \vee (N \oplus V) = 1$
JMP	—	Branches unconditionally to a specified address.		
BSR	—	Branches to a subroutine at a specified address.		
JSR	—	Branches to a subroutine at a specified address.		
RTS	—	Returns from a subroutine		

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

Table 2.8 System Control Instructions

Instruction	Size*	Function
RTE	—	Returns from an exception-handling routine.
SLEEP	—	Causes a transition to a power-down state.
LDC	B/W	(EAs) → CCR Moves the source operand contents to the CCR. The CCR size is one byte, but in transfer from memory, data is read by word access.
STC	B/W	CCR → (EAd) Transfers the CCR contents to a destination location. The condition code register size is one byte, but in transfer to memory, data is written by word access.
ANDC	B	CCR ∧ #IMM → CCR Logically ANDs the CCR with immediate data.
ORC	B	CCR ∨ #IMM → CCR Logically ORs the CCR with immediate data.
XORC	B	CCR ⊕ #IMM → CCR Logically XORs the CCR with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

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

Table 2.9 Block Data Transfer Instructions

Instruction	Size	Function
EEPMOV.B	—	if R4L ≠ 0 then Repeat @ER5+ → @ER6+, R4L–1 → R4L Until R4L = 0 else next;
EEPMOV.W	—	if R4 ≠ 0 then Repeat @ER5+ → @ER6+, R4–1 → R4 Until R4 = 0 else next; Transfers a data block. Starting from the address set in ER5, transfers data for the number of bytes set in R4L or R4 to the address location set in ER6. Execution of the next instruction begins as soon as the transfer is completed.

2.4.2 Basic Instruction Formats

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

Figure 2.7 shows examples of instruction formats.

(1) Operation Field

Indicates the function of the instruction, the addressing mode, and the operation to be carried out on the operand. The operation field always includes the first four bits of the instruction. Some instructions have two operation fields.

(2) Register Field

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

(3) Effective Address Extension

8, 16, or 32 bits specifying immediate data, an absolute address, or a displacement. A24-bit address or displacement is treated as a 32-bit data in which the first 8 bits are 0 (H'00).

(4) Condition Field

Specifies the branching condition of Bcc instructions.

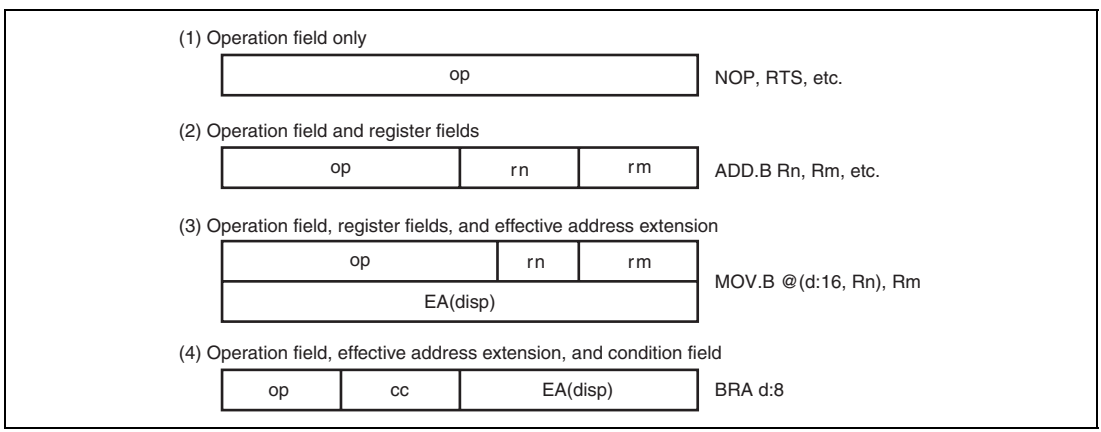


Figure 2.7 Instruction Formats

2.5 Addressing Modes and Effective Address Calculation

2.5.1 Addressing Modes

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

Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit-manipulation instructions use register direct, register indirect, or the absolute addressing mode (@aa:8) to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) 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 containing the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

(2) Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn), the lower 24 bits of which contain the address of the operand on memory.

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

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

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

- Register indirect with post-increment—@ERn+

  The register field of the instruction code specifies an address register (ERn) the lower 24 bits of which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents (32 bits) and the sum is stored in the address register. The value added is 1 for byte access, 2 for word access, or 4 for longword access. For the word 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 register field in the instruction code, and the lower 24 bits of the result is the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word access, or 4 for longword access. For the word or longword access, the register value 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 address may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24)

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

Table 2.11 shows the access ranges of absolute addresses.

Table 2.11 Absolute Address Access Ranges

Absolute Address	Access Range
8 bits (@aa:8)	H'FFFF00 to H'FFFFFF (16776960 to 16777215)
16 bits (@aa:16)	H'000000 to H'007FFF, H'FF8000 to H'FFFFFF (0 to 32767, 16744448 to 16777215)
24 bits (@aa:24)	H'000000 to H'FFFFFF (0 to 16777215)

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

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

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some bit manipulation instructions contain 3-bit immediate data in the instruction code, specifying a bit number.

(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 the instruction is sign-extended and added to the 24-bit PC contents to generate a branch address. The PC value to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) or –32766 to +32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting value should be an even number.

(8) Memory Indirect—@@aa:8

This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand. This memory operand contains a branch address. The memory operand is accessed in words, generating a 16-bit branch address. Figure 2.8 shows how to specify branch address for in memory indirect mode. The upper bits of the absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF).

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

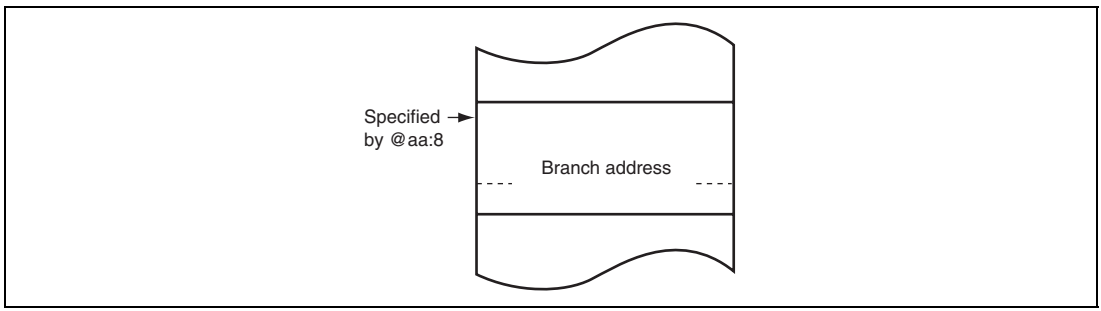


Figure 2.8 Branch Address Specification in Memory Indirect Mode

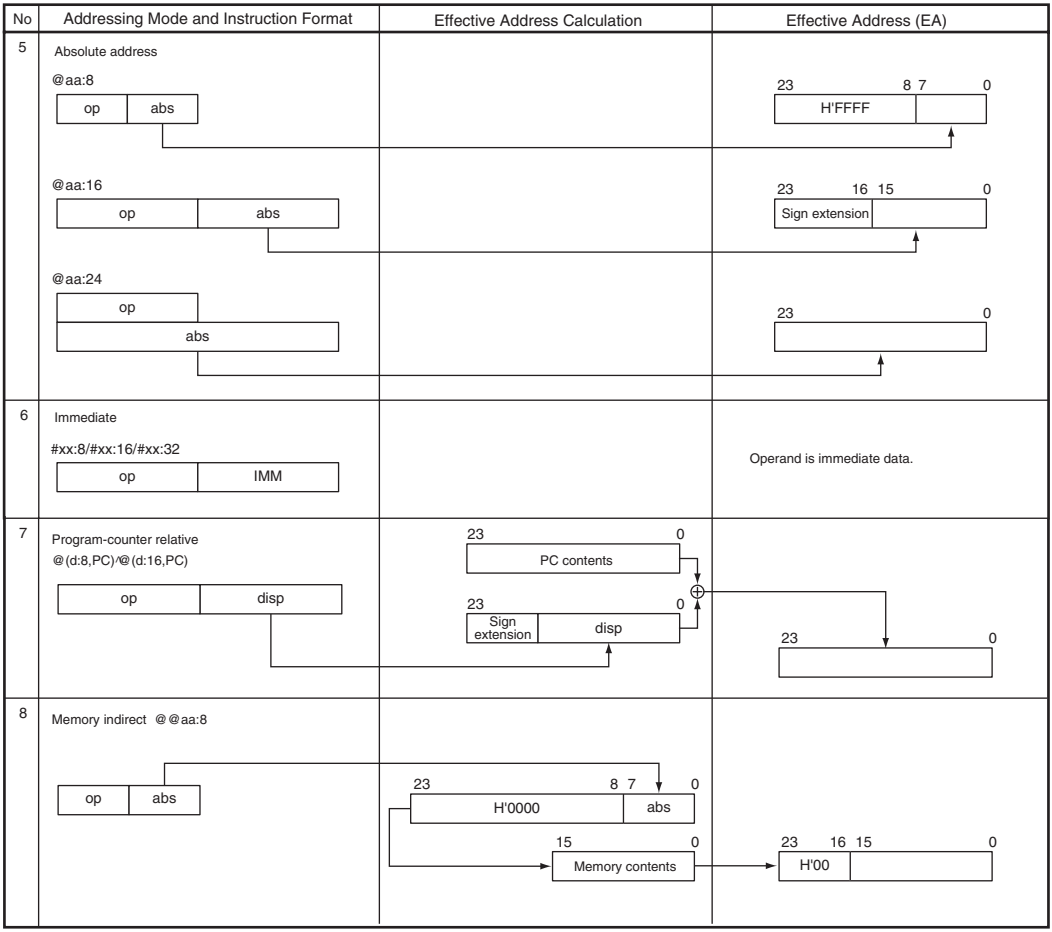
2.5.2 Effective Address Calculation

Table 2.12 indicates how effective addresses are calculated in each addressing mode.

Table 2.12 Effective Address Calculation (1)

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

Table 2.12 Effective Address Calculation (2)



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

2.6 Basic Bus Cycle

CPU operation is synchronized by a system clock (ϕ) or a subclock (ϕ_{SUB}). The period from a rising edge of ϕ or ϕ_{SUB} to the next rising edge is called one state. A bus cycle consists of two states or three states. The cycle differs depending on whether access is to on-chip memory or to on-chip peripheral modules.

2.6.1 Access to On-Chip Memory (RAM, ROM)

Access to on-chip memory takes place in two states. The data bus width is 16 bits, allowing access in byte or word size. Figure 2.9 shows the on-chip memory access cycle.

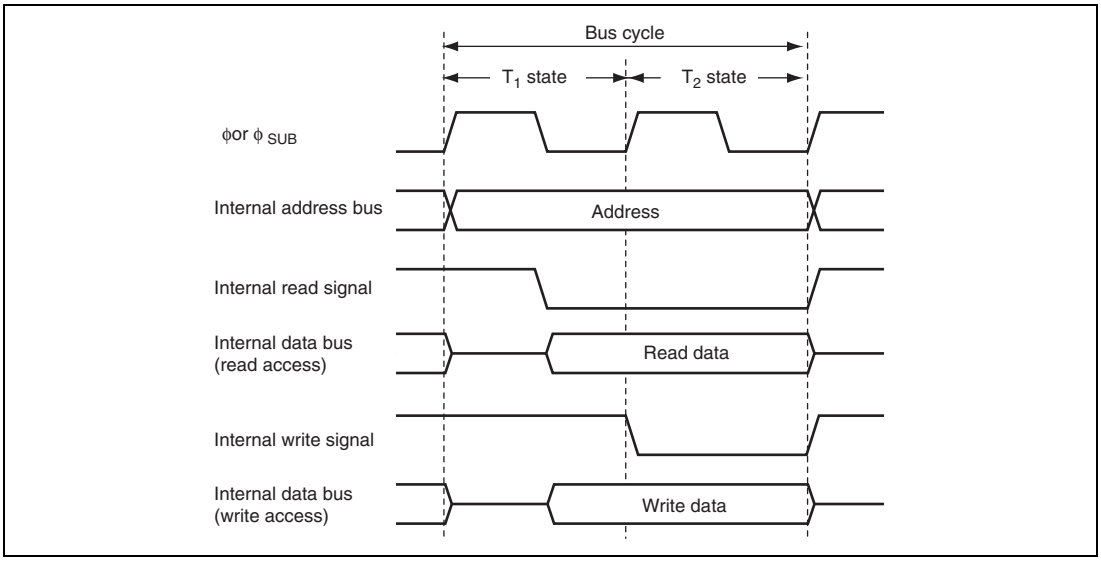


Figure 2.9 On-Chip Memory Access Cycle

2.6.2 On-Chip Peripheral Modules

On-chip peripheral modules are accessed in two states or three states. The data bus width is 8 bits or 16 bits depending on the register. For description on the data bus width and number of accessing states of each register, refer to section 24.1, Register Addresses (Address Order). Registers with 16-bit data bus width can be accessed by word size only. Registers with 8-bit data bus width can be accessed by byte or word size. When a register with 8-bit data bus width is accessed by word size, a bus cycle occurs twice. In two-state access, the operation timing is the same as that for on-chip memory.

Figure 2.10 shows the operation timing in the case of three-state access to an on-chip peripheral module.

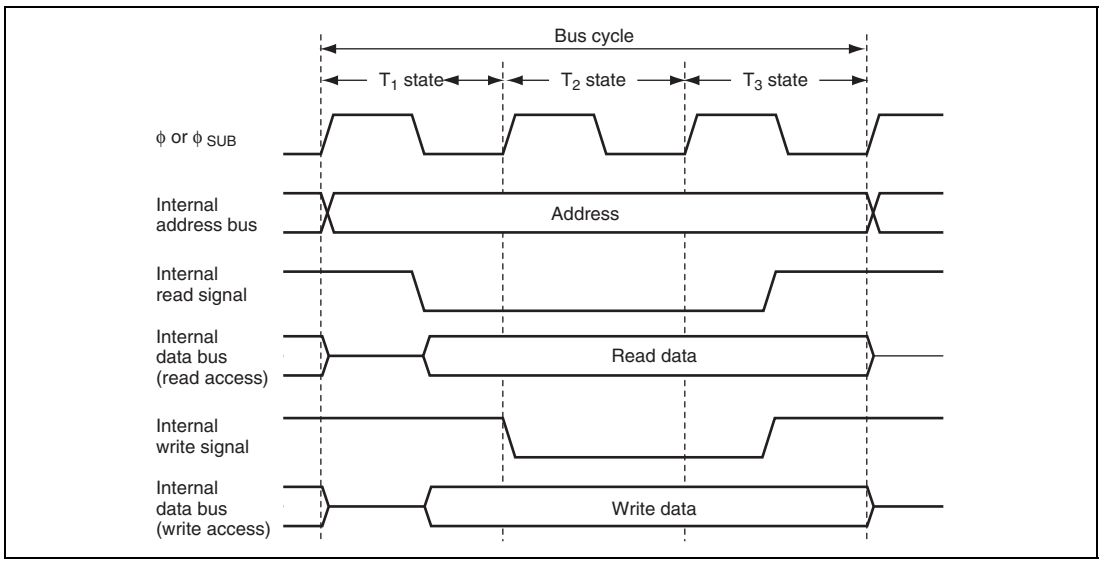


Figure 2.10 On-Chip Peripheral Module Access Cycle (3-State Access)

2.7 CPU States

There are four CPU states: the reset state, program execution state, program halt state, and exception-handling state. The program execution state includes active (high-speed or medium-speed) mode and subactive mode. For the program halt state, there are sleep (high-speed or medium-speed) mode, standby mode, watch mode, and subsleep mode. These states are shown in figure 2.11. Figure 2.12 shows the state transitions. For details on program execution state and program halt state, refer to section 6, Power-Down Modes. For details on exception handling, refer to section 3, Exception Handling.

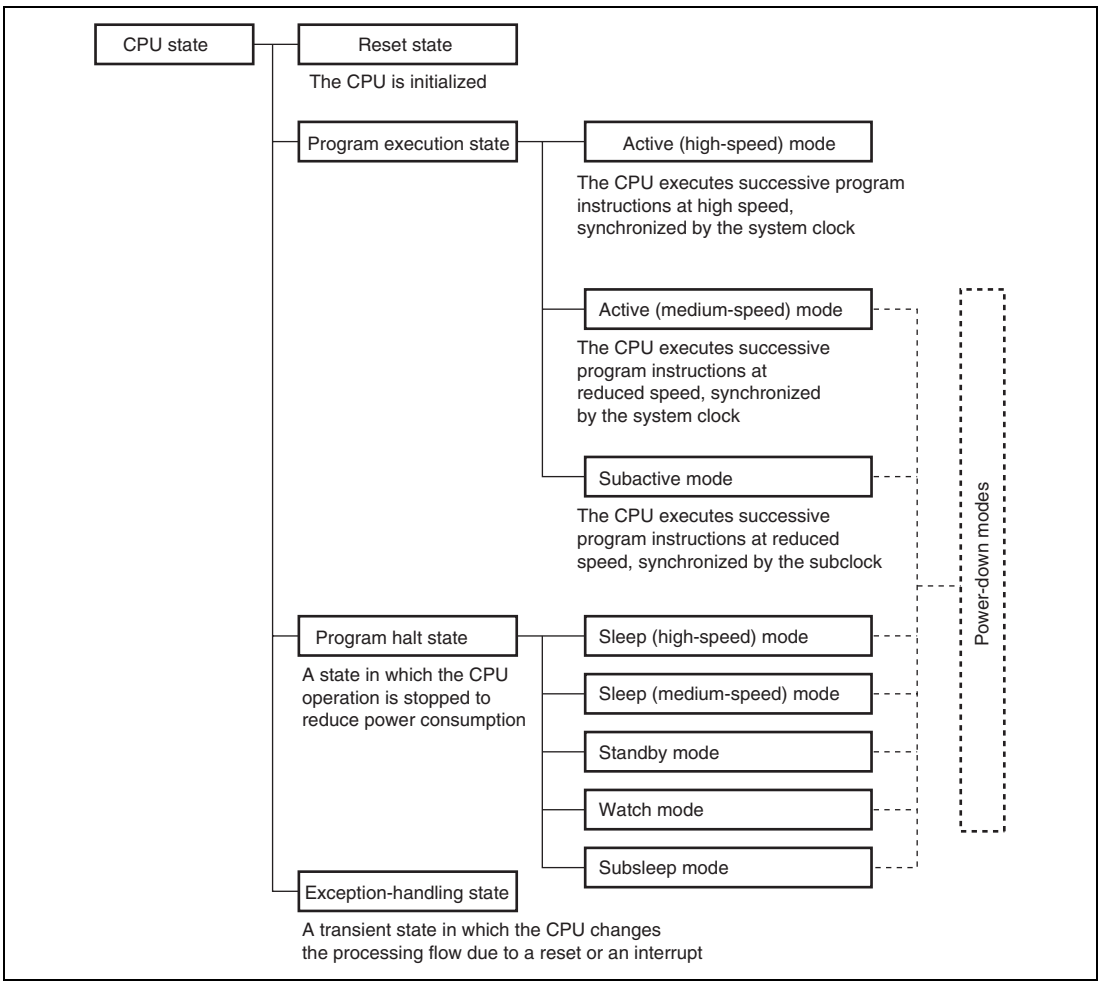


Figure 2.11 CPU Operating States

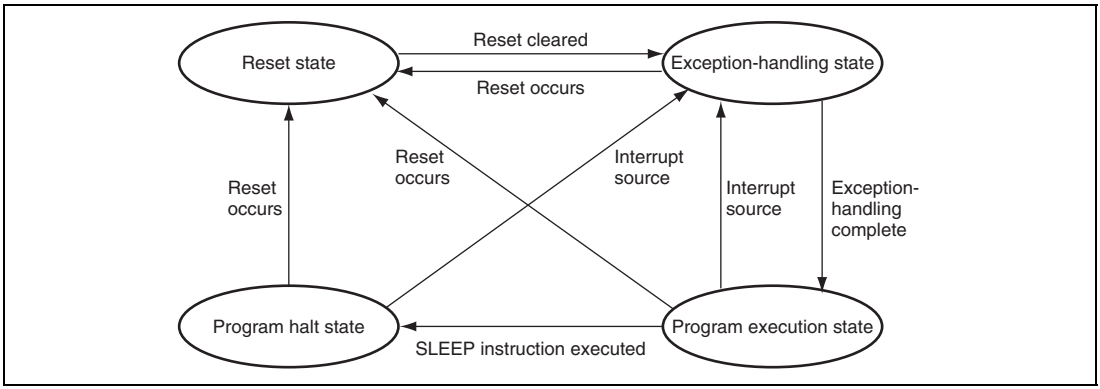


Figure 2.12 State Transitions

2.8 Usage Notes

2.8.1 Notes on Data Access to Empty Areas

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

2.8.2 EEPMOV Instruction

EEPMOV is a block-transfer instruction and transfers the byte size of data indicated by R4 or R4L, which starts from the address indicated by ER5, to the address indicated by ER6. Set R4 (or R4L) and ER6 so that the end address of the destination address (value of ER6 + R4 or ER6 + R4L) does not exceed H'00FFFFFF (the value of ER6 must not change from H'00FFFFFF to H'01000000 during execution).

2.8.3 Bit-Manipulation Instruction

The BSET, BCLR, BNOT, BST, and BIST instructions read data from the specified address in byte units, manipulate the data of the target bit, and write data to the same address again in byte units. Special care is required when using these instructions in cases where two registers are 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

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

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

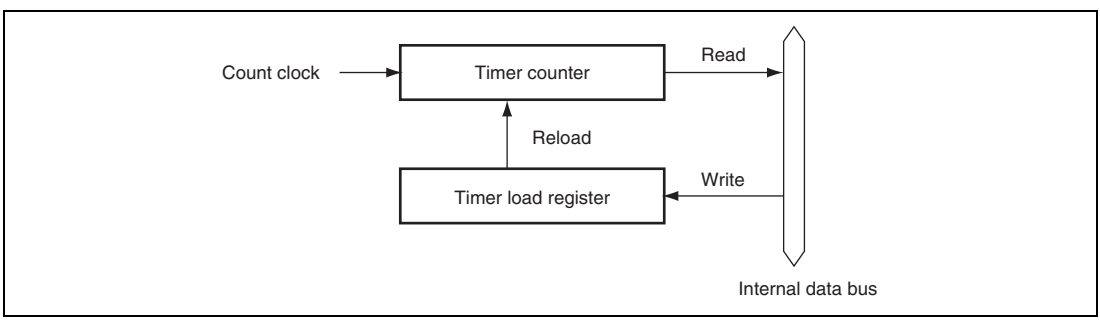


Figure 2.13 Example of Timer Configuration with Two Registers Allocated to Same Address

Example 2: When 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 input at P56. P55 to P50 are output pins and output low-level signals. An example to output a high-level signal at P50 with a BSET instruction is shown below.

- Prior to executing BSET instruction

	P57	P56	P55	P54	P53	P52	P51	P50
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	0

- BSET instruction executed

```
BSET    #0,    @PDR5:8
```

The BSET instruction is executed for port 5.

- After executing BSET instruction

	P57	P56	P55	P54	P53	P52	P51	P50
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR5	0	0	1	1	1	1	1	1
PDR5	0	1	0	0	0	0	0	1

- Description on operation

1. When the BSET instruction is executed, first the CPU reads port 5.
   Since P57 and P56 are input pins, the CPU reads the pin states (low-level and high-level input).
   P55 to P50 are output pins, so the CPU reads the value in PDR5. In this example PDR5 has a 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-level signal. However, bits 7 and 6 of PDR5 end up with different values. To prevent this problem, store a copy of the PDR5 data in a work area in memory. Perform the bit manipulation on the data in the work area, then write this data to PDR5.

- Prior to executing BSET instruction

```
MOV.B   #H'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**	**P50**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	0
RAM0	1	0	0	0	0	0	0	0

- BSET instruction executed

```
BSET    #0,   @RAM0:8
```

The BSET instruction is executed designating the PDR5 work area (RAM0).

- After executing BSET instruction

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

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

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	1
RAM0	1	0	0	0	0	0	0	1

(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 input at P56. P55 to P50 are output pins that output low-level signals. An example of setting the P50 pin as an input pin by the BCLR instruction is shown below. It is assumed that a high-level signal will be input to this input pin.

- Prior to executing BCLR instruction

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	0

- BCLR instruction executed

```
BCLR    #0,     @PCR5:8
```

The BCLR instruction is executed for PCR5.

- After executing BCLR instruction

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
Input/output	Output	Output	Output	Output	Output	Output	Output	Input
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR5	1	1	1	1	1	1	1	0
PDR5	1	0	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 write-only 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. However, bits 7 and 6 in PCR5 change to 1, so that P57 and P56 change from input pins to output pins. To prevent this problem, store a copy of the PDR5 data in a work area in memory and manipulate data of the bit in the work area, then write this data to PDR5.

- Prior to executing BCLR instruction

```
MOV.B   #H'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**	**P50**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1	1

- BCLR instruction executed

```
BCLR   #0,    @RAM0:8
```

The BCLR instructions executed for the PCR5 work area (RAM0).

- After executing BCLR instruction

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

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

	P57	P56	P55	P54	P53	P52	P51	P50
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR5	0	0	1	1	1	1	1	0
PDR5	1	0	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1	0

Section 3 Exception Handling

Exception handling may be caused by a reset or interrupts.

- Reset

  A reset has the highest exception priority. Exception handling starts as soon as the reset is cleared by the $\overline{\text{RES}}$ pin. The chip is also reset when the watchdog timer overflows, and exception handling starts. Exception handling is the same as exception handling by the $\overline{\text{RES}}$ pin.

- Interrupts

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

3.1 Exception Sources and Vector Address

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

Table 3.1 Exception Sources and Vector Address

Source Origin	Exception Sources	Vector Number	Vector Address	Priority
$\overline{\text{RES}}$ pin/Watchdog timer	Reset	0	H'000000 to H'000003	High
—	Reserved for system use	1, 2	H'000004 to H'00000B	
External interrupt	NMI	3	H'00000C to H'00000F	
—	Reserved for system use	4	H'000010 to H'000013	
Address break	Break conditions satisfied	5	H'000014 to H'000017	
External interrupts	IRQ0	6	H'000018 to H'00001B	
	IRQ1	7	H'00001C to H'00001F	
	IRQAEC	8	H'000020 to H'000023	
	IRQ3	9	H'000024 to H'000027	
	IRQ4	10	H'000028 to H'00002B	
	WKP0	11	H'00002C to H'00002F	
	WKP1	12	H'000030 to H'000033	
	WKP2	13	H'000034 to H'000037	
	WKP3	14	H'000038 to H'00003B	
	WKP4	15	H'00003C to H'00003F	
	WKP5	16	H'000040 to H'000043	
	WKP6	17	H'000044 to H'000047	
	WKP7	18	H'000048 to H'00004B	
Internal interrupts*	—	19 to 55	H'00004C to H'0000DF	Low

Note: * For details on the vector table of internal interrupts, see section 4.5, Interrupt Exception Handling Vector Table.

3.2 Reset

A reset has the highest exception priority. Table 3.2 shows the three sources that cause a reset.

Table 3.2 Interrupt Sources that Cause a Reset

Origin of Interrupt Source	Description
$\overline{\text{RES}}$ pin	Low-level input
Power-on reset circuit	Rising of the power-supply voltage (Vcc) For details, see section 22, Power-On Reset Circuit.
Watchdog timer	Counter overflow For details, see section 16, Watchdog Timer.

3.2.1 Reset Exception Handling

When a reset is generated, all processing halts and this LSI enters the reset state. A reset initializes the internal state of the CPU and the registers of the on-chip peripheral modules.

To ensure that this LSI be reset, the $\overline{\text{RES}}$ pin has to be held low for the oscillation stabilization time of the system clock oscillator either after power-on or when the system clock oscillator is halted. If the system clock oscillator is functioning, the $\overline{\text{RES}}$ pin has to be held low for the number of the t_{REL} state as is specified by the electrical characteristics.

When a reset source has been generated, this LSI starts reset exception handling as follows.

1. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized and the I bit in CCR is set to 1.
2. The reset exception handling vector address (H'00000000 to H'00000003) is read and transferred to the PC, and then program execution starts from the address indicated by the PC.

The sequence of the reset exception handling caused by the $\overline{\text{RES}}$ pin is shown in figure 3.1.

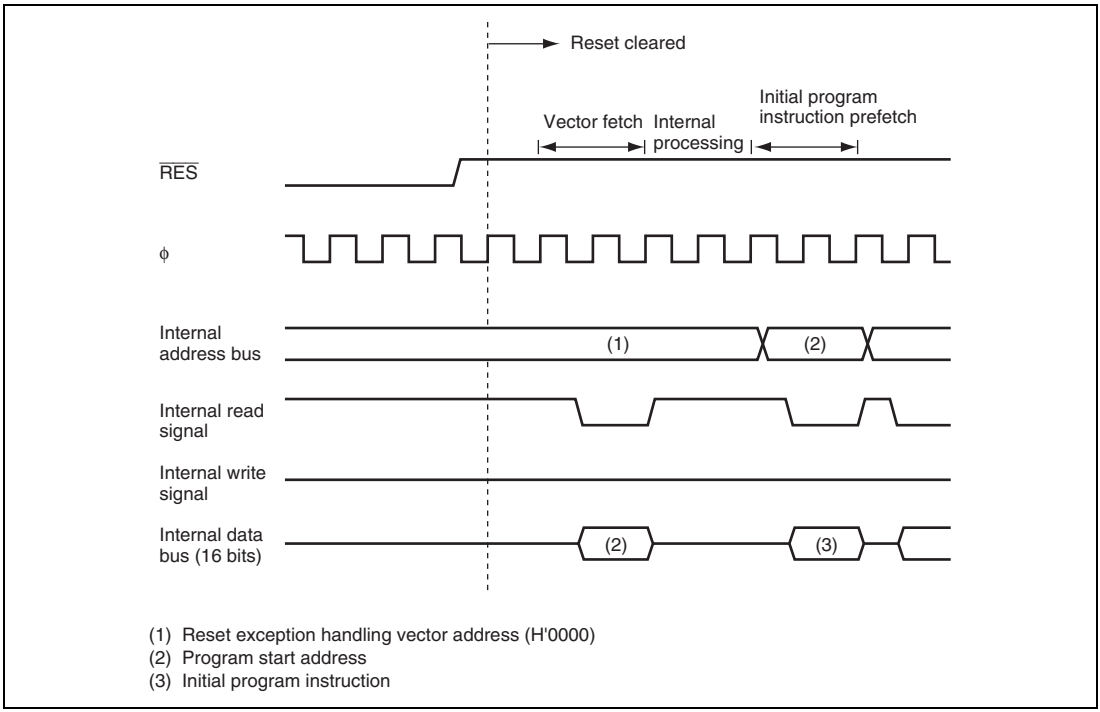


Figure 3.1 Reset Exception Handling Sequence

3.2.2 Interrupt Immediately after Reset

Immediately after a reset, if an interrupt is accepted before the stack pointer (SP) is initialized, PC and CCR will not be pushed onto the stack correctly, resulting in program runaway. To prevent this, immediately after reset exception handling all interrupts are masked. For this reason, the initial program instruction is always executed immediately after a reset. This instruction should initialize the stack pointer (e.g. MOV.L #xx: 32, SP).

3.3 Interrupts

The interrupt sources include 14 external interrupts (NMI, IRQ0, IRQ1, IRQ3, IRQ4, IRQAEC, and WKP7 to WKP0) and 28 internal interrupts (for the flash memory version) or 27 internal interrupts (for the masked ROM version) from on-chip peripheral modules. Figure 3.2 shows the interrupt sources and their numbers.

The on-chip peripheral modules which require interrupt sources are the watchdog timer (WDT), address break, realtime clock (RTC), 16-bit timer pulse unit (TPU), asynchronous event counter (AEC), timer C, timer F, timer G, serial communication interface (SCI), and A/D converter. Interrupt vector addresses are allocated to individual sources.

NMI is an interrupt with the highest priority and accepted at all times. Interrupts are controlled by the interrupt controller. The interrupt controller sets interrupts other than NMI to three mask levels in order to control multiple interrupts. The interrupt priority registers A to F (IPRA to IPRF) of the interrupt controller set the interrupt mask levels.

For details on interrupts, see section 4, Interrupt Controller.

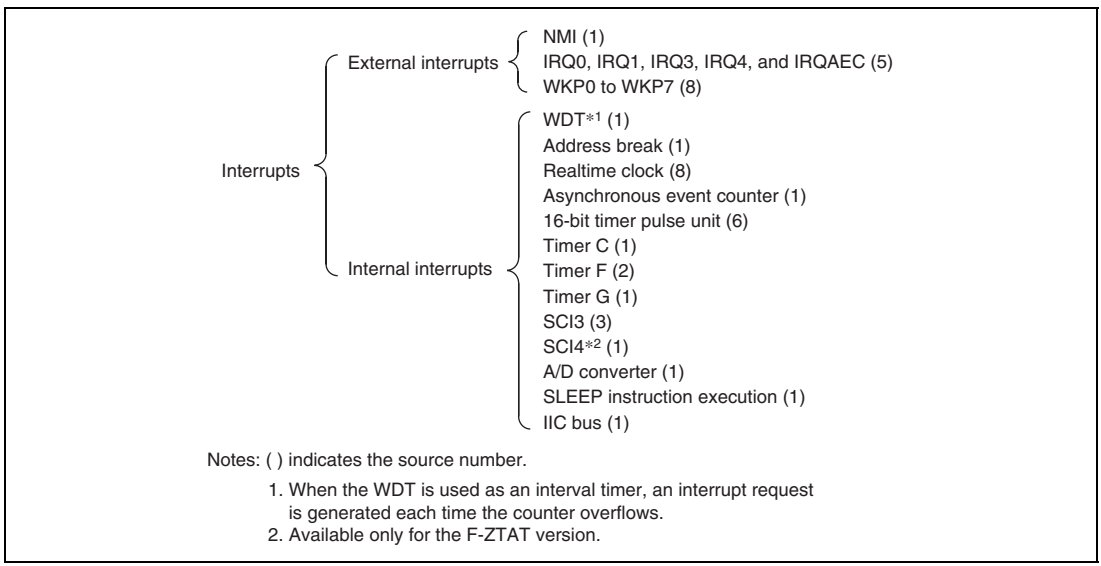
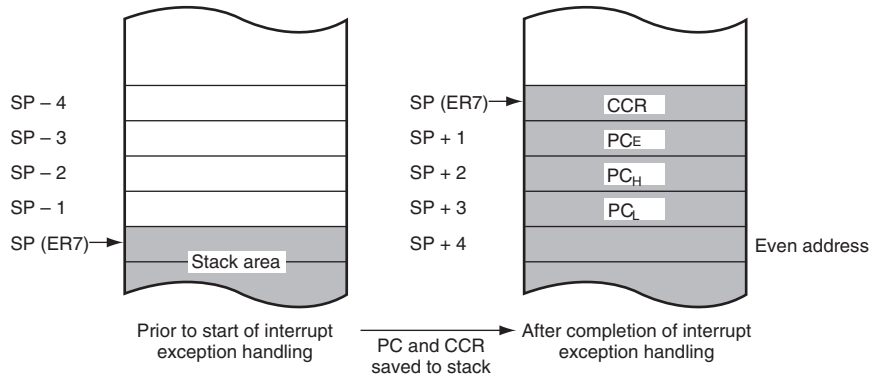


Figure 3.2 Interrupt Sources and their Numbers

3.4 Stack Status after Exception Handling

Figures 3.3 shows the stack after completion of interrupt exception handling.



[Legend]
PC_E: Bits 23 to 16 of program counter (PC)
PC_H: Bits 15 to 8 of program counter (PC)
PC_L: Bits 7 to 0 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 in word or longword units, starting from an even-numbered address.

Figure 3.3 Stack Status after Exception Handling

3.5 Usage Notes

3.5.1 Notes on Stack Area Use

When word data or longword data is accessed in this LSI, the least significant bit of the address is regarded as 0. The stack must always be accessed in word units or longword units, and the stack pointer (SP: ER7) should never indicate an odd address. Use PUSH.W Rn (MOV.W Rn, @–SP) or PUSH.L ERn (MOV.L ERn, @–SP) to save register values. To restore register values, use POP.W Rn (MOV.W @SP+, Rn) or POP.L ERn (MOV.L @SP+, ERn).

Setting an odd address in SP may cause a program to crash. An example is shown in figure 3.4.

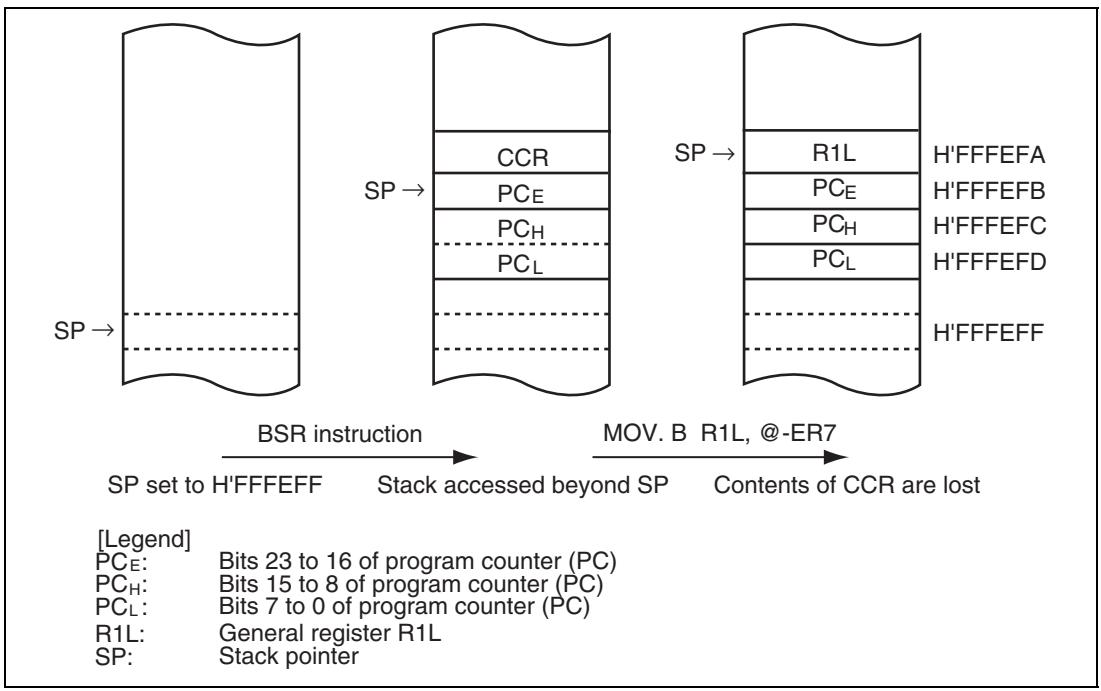


Figure 3.4 Operation when Odd Address is Set in SP

3.5.2 Notes on Rewriting Port Mode Registers

When a port mode register is rewritten to switch the functions of external interrupt pins and when the value of the ECPWME bit in AEGSR is rewritten to switch between selection and non-selection of IRQAEC, the following points should be observed.

When a pin function is switched by rewriting a port mode register that controls an external interrupt pin ($\overline{\text{IRQ4}}$, $\overline{\text{IRQ3}}$, $\overline{\text{IRQ1}}$, $\overline{\text{IRQ0}}$, or $\overline{\text{WKP7}}$ to $\overline{\text{WKP0}}$), the interrupt request flag is set to 1 at the time the pin function is switched, even if no valid interrupt is input at the pin. Be sure to clear the interrupt request flag to 0 after switching the pin function. When the value of the ECPWME bit in AEGSR that sets selection or non-selection of IRQAEC is rewritten, the interrupt request flag may be set to 1, even if a valid edge has not arrived on the selected IRQAEC or IECPWM (PWM output for the AEC). Therefore, be sure to clear the interrupt request flag to 0 after switching the pin function.

Table 3.3 shows the conditions under which interrupt request flags are set to 1.

Table 3.3 Conditions under which Interrupt Request Flag is Set to 1

Interrupt Request Flags Set to 1		Conditions
IRR1	IRRI4	When the IRQ4 bit in PMR9 or PMRF is changed from 0 to 1 while the $\overline{\text{IRQ4}}$ pin is low and the IEG4 bit in IEGR is 0. When the IRQ4 bit in PMR9 or PMRF is changed from 1 to 0 while the $\overline{\text{IRQ4}}$ pin is low and the IEG4 bit in IEGR is 1.
	IRRI3	When the IRQ3 bit in PMRB or PMRE is changed from 0 to 1 while the $\overline{\text{IRQ3}}$ pin is low and the IEG3 bit in IEGR is 0. When the IRQ3 bit in PMRB or PMRE is changed from 1 to 0 while the $\overline{\text{IRQ3}}$ pin is low and the IEG3 bit in IEGR is 1.
	IRREC2	When an edge as designated by the AIEGS1 and AIEGS0 bits in AEGSR is detected because the values of the IRQAEC pin and of IECPWM at switching are different (e.g., when the rising edge has been selected and the ECPWME bit in AEGSR is changed from 1 to 0 while the IRQAEC pin is low and IECPWM is 1).
	IRRI1	When the IRQ1 bit in PMRB or PMRE is changed from 0 to 1 while the $\overline{\text{IRQ1}}$ pin is low and the IEG1 bit in IEGR is 0. When the IRQ1 bit in PMRB or PMRE is changed from 1 to 0 while the $\overline{\text{IRQ1}}$ pin is low and the IEG1 bit in IEGR is 1.
	IRRI0	When the IRQ0 bit in PMRB or PMRE is changed from 0 to 1 while the $\overline{\text{IRQ0}}$ pin is low and the IEG0 bit in IEGR is 0. When the IRQ0 bit in PMRB or PMRE is changed from 1 to 0 while the $\overline{\text{IRQ0}}$ pin is low and the IEG0 bit in IEGR is 1.
IWPR	IWPF7	When the WKP7 bit in PMR5 is changed from 0 to 1 while the $\overline{\text{WKP7}}$ pin is low.
	IWPF6	When the WKP6 bit in PMR5 is changed from 0 to 1 while the $\overline{\text{WKP6}}$ pin is low.
	IWPF5	When the WKP5 bit in PMR5 is changed from 0 to 1 while the $\overline{\text{WKP5}}$ pin is low.
	IWPF4	When the WKP4 bit in PMR5 is changed from 0 to 1 while the $\overline{\text{WKP4}}$ pin is low.
	IWPF3	When the WKP3 bit in PMR5 is changed from 0 to 1 while the $\overline{\text{WKP3}}$ pin is low.
	IWPF2	When the WKP2 bit in PMR5 is changed from 0 to 1 while the $\overline{\text{WKP2}}$ pin is low.
	IWPF1	When the WKP1 bit in PMR5 is changed from 0 to 1 while the $\overline{\text{WKP1}}$ pin is low.
	IWPF0	When the WKP0 bit in PMR5 is changed from 0 to 1 while the $\overline{\text{WKP0}}$ pin is low.

Figure 3.5 shows the procedure for setting a bit in a port mode register and clearing the interrupt request flag. This procedure also applies to AEGSR setting.

When switching a pin function, mask the interrupt before setting the bit in the port mode register (or AEGSR). After accessing the port mode register (or AEGSR), execute at least one instruction (e.g., NOP), then clear the interrupt request flag from 1 to 0. If the instruction to clear the flag to 0 is executed immediately after the port mode register (or AEGSR) access without executing an instruction, the flag will not be cleared.

An alternative method is to avoid the setting of interrupt request flags when pin functions are switched by keeping the pins at the high level so that the conditions in table 3.3 are not satisfied. However, the procedure in figure 3.5 is recommended because IECPWM is an internal signal and determining its value is complicated.

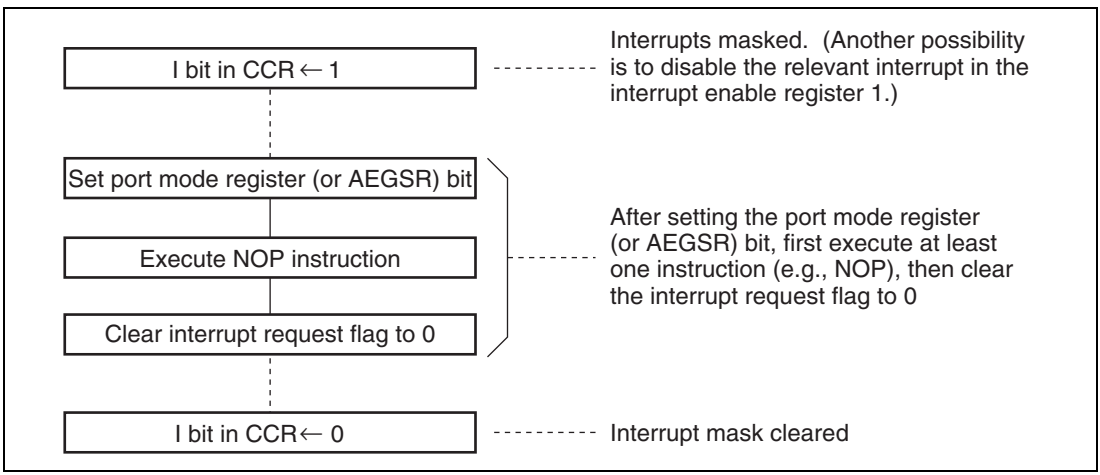


Figure 3.5 Port Mode Register (or AEGSR) Setting and Interrupt Request Flag Clearing Procedure

3.5.3 Method for Clearing Interrupt Request Flags

Use the recommended method given below when clearing the flags in interrupt request registers (IRR1, IRR2, and IWPR).

(1) Recommended method

Use a single instruction to clear flags. The bit manipulation instruction and byte-size data transfer instruction can be used. Two examples of program code for clearing IRRI1 (bit 1 in IRR1) are given below.

```
BCLR  #1,  @IRR1:8

MOV.B R1L, @IRR1:8 (set the value of R1L to B'11111101)
```

(2) Example of a malfunction

When flags are cleared with multiple instructions, other flags might be cleared during execution of the instructions, even though they are currently set, and this will cause a malfunction.

Here is an example in which IRRI0 is cleared and disabled in the process of clearing IRRI1 (bit 1 in IRR1).

```
MOV.B @IRR1:8,R1L ......... IRRI0 = 0 at this time

AND.B #B'11111101,R1L ..... Here, IRRI0 = 1

MOV.B R1L,@IRR1:8 ......... IRRI0 is cleared to 0
```

In the above example, it is assumed that an IRQ0 interrupt is generated while the AND.B instruction is executing.

The IRQ0 interrupt is disabled because, although the original objective is clearing IRRI1, IRRI0 is also cleared.

Section 4 Interrupt Controller

4.1 Features

This LSI includes an interrupt controller, which has the following features.

- Priorities settable with IPR

  An interrupt priority register (IPR) is provided for setting mask levels for interrupts. Three mask levels can be set for each module for all interrupts except an NMI and address break.
- Interrupts can be enabled or disabled in three levels by the INTM1 and INTM0 bits in the interrupt mask register (INTM).
- Fourteen external interrupts

  NMI is the highest-priority interrupt, and is accepted at all times. Rising or falling edge sensing can be selected for NMI. Rising or falling edge sensing can be selected for IRQ0, IRQ1, IRQ3, IRQ4, and WKP0 to WKP7. Rising, falling, or both edge sensing can be selected for IRQAEC.

A block diagram of the interrupt controller is shown in figure 4.1.

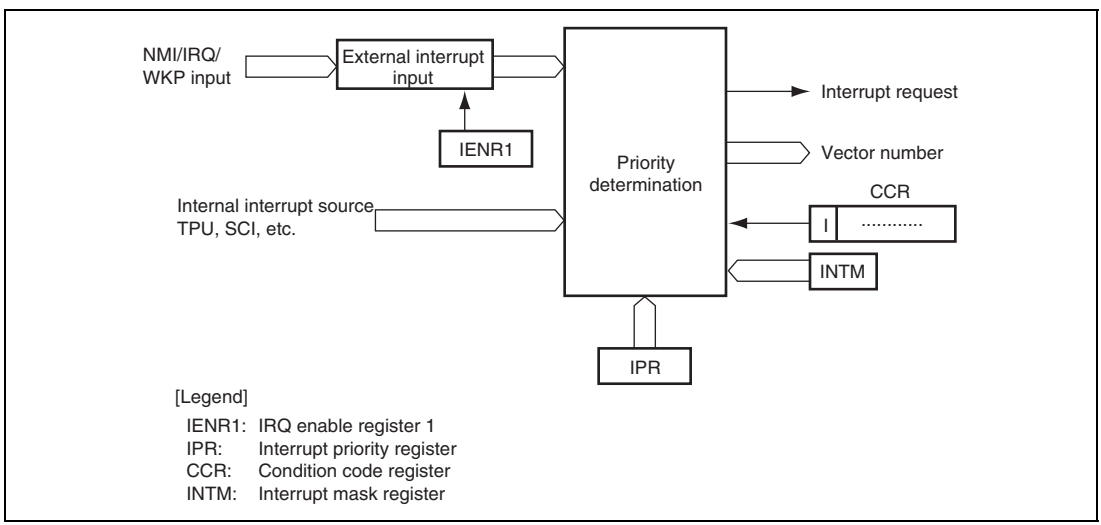


Figure 4.1 Block Diagram of Interrupt Controller

4.2 Input/Output Pins

Table 4.1 shows the pin configuration of the interrupt controller.

Table 4.1 Pin Configuration

Pin Name	I/O	Function
$\overline{\text{NMI}}$	Input	Nonmaskable external interrupt pin Rising or falling edge can be selected
IRQAEC	Input	Maskable external interrupt pin Rising, falling, or both edges can be selected
$\overline{\text{IRQ4}}$	Input	Maskable external interrupt pins Rising or falling edge can be selected
$\overline{\text{IRQ3}}$	Input	
$\overline{\text{IRQ1}}$	Input	
$\overline{\text{IRQ0}}$	Input	
$\overline{\text{WKP7}}$ to $\overline{\text{WKP0}}$	Input	Maskable external interrupt pins Accepted at a rising or falling edge

4.3 Register Descriptions

The interrupt controller has the following registers.

- Interrupt edge select register (IEGR)
- Wakeup edge select register (WEGR)
- Interrupt enable register 1 (IENR1)
- Interrupt enable register 2 (IENR2)
- Interrupt request register 1 (IRR1)
- Interrupt request register 2 (IRR2)
- Wakeup interrupt request register (IWPR)
- Interrupt priority register A (IPRA)
- Interrupt priority register B (IPRB)
- Interrupt priority register C (IPRC)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt mask register (INTM)

4.3.1 Interrupt Edge Select Register (IEGR)

IEGR selects the sense of an edge that generates interrupt requests of the $\overline{\text{NMI}}$, TMIF, $\overline{\text{ADTRG}}$, $\overline{\text{IRQ4}}$, $\overline{\text{IRQ3}}$, $\overline{\text{IRQ1}}$, and $\overline{\text{IRQ0}}$ pins.

Bit	Bit Name	Initial Value	R/W	Descriptions
7	NMIEG	0	R/W	NMI Edge Select 0: Detects a falling edge of the $\overline{\text{NMI}}$ pin input 1: Detects a rising edge of the $\overline{\text{NMI}}$ pin input
6	TMIFEG	0	R/W	TMIF Edge Select 0: Detects a falling edge of the TMIF pin input 1: Detects a rising edge of the TMIF pin input
5	ADTRGNEG	0	R/W	$\overline{\text{ADTRG}}$ Edge Select 0: Detects a falling edge of the $\overline{\text{ADTRG}}$ pin input 1: Detects a rising edge of the $\overline{\text{ADTRG}}$ pin input
4	IEG4	0	R/W	IRQ4 Edge Select 0: Detects a falling edge of the $\overline{\text{IRQ4}}$ pin input 1: Detects a rising edge of the $\overline{\text{IRQ4}}$ pin input
3	IEG3	0	R/W	IRQ3 Edge Select 0: Detects a falling edge of the $\overline{\text{IRQ3}}$ pin input 1: Detects a rising edge of the $\overline{\text{IRQ3}}$ pin input
2	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
1	IEG1	0	R/W	IRQ1 Edge Select 0: Detects a falling edge of the $\overline{\text{IRQ1}}$ pin input 1: Detects a rising edge of the $\overline{\text{IRQ1}}$ pin input
0	IEG0	0	R/W	IRQ0 Edge Select 0: Detects a falling edge of the $\overline{\text{IRQ0}}$ pin input 1: Detects a rising edge of the $\overline{\text{IRQ0}}$ pin input

4.3.2 Wakeup Edge Select Register (WEGR)

WEGR selects the sense of an edge that generates interrupt requests of the $\overline{\text{WKP7}}$ to $\overline{\text{WKP0}}$ pins.

Bit	Bit Name	Initial Value	R/W	Description
7	WKEGS7	0	R/W	WKP7 Edge Select 0: Detects a falling edge of the $\overline{\text{WKP7}}$ pin input 1: Detects a rising edge of the $\overline{\text{WKP7}}$ pin input
6	WKEGS6	0	R/W	WKP6 Edge Select 0: Detects a falling edge of the $\overline{\text{WKP6}}$ pin input 1: Detects a rising edge of the $\overline{\text{WKP6}}$ pin input
5	WKEGS5	0	R/W	WKP5 Edge Select 0: Detects a falling edge of the $\overline{\text{WKP5}}$ pin input 1: Detects a rising edge of the $\overline{\text{WKP5}}$ pin input
4	WKEGS4	0	R/W	WKP4 Edge Select 0: Detects a falling edge of the $\overline{\text{WKP4}}$ pin input 1: Detects a rising edge of the $\overline{\text{WKP4}}$ pin input
3	WKEGS3	0	R/W	WKP3 Edge Select 0: Detects a falling edge of the $\overline{\text{WKP3}}$ pin input 1: Detects a rising edge of the $\overline{\text{WKP3}}$ pin input
2	WKEGS2	0	R/W	WKP2 Edge Select 0: Detects a falling edge of the $\overline{\text{WKP2}}$ pin input 1: Detects a rising edge of the $\overline{\text{WKP2}}$ pin input
1	WKEGS1	0	R/W	WKP1 Edge Select 0: Detects a falling edge of the $\overline{\text{WKP1}}$ pin input 1: Detects a rising edge of the $\overline{\text{WKP1}}$ pin input
0	WKEGS0	0	R/W	WKP0 Edge Select 0: Detects a falling edge of the $\overline{\text{WKP0}}$ pin input 1: Detects a rising edge of the $\overline{\text{WKP0}}$ pin input

4.3.3 Interrupt Enable Register 1 (IENR1)

IENR1 enables the RTC, WKP7 to WKP0, IRQ0, IRQ1, IRQ3, IRQ4, and IRQAEC interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	IENRTC	0	R/W	RTC Interrupt Request Enable The RTC interrupt request is enabled when this bit is set to 1.
6	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
5	IENWP	0	R/W	Wakeup Interrupt Request Enable The WKP7 to WKP0 interrupt requests are enabled when this bit is set to 1.
4	IEN4	0	R/W	IRQ4 Interrupt Request Enable The IRQ4 interrupt request is enabled when this bit is set to 1.
3	IEN3	0	R/W	IRQ3 Interrupt Request Enable The IRQ3 interrupt request is enabled when this bit is set to 1.
2	IENEC2	0	R/W	IRQAEC Interrupt Request Enable The IRQAEC interrupt request is enabled when this bit is set to 1.
1	IEN1	0	R/W	IRQ1 Interrupt Request Enable The IRQ1 interrupt request is enabled when this bit is set to 1.
0	IEN0	0	R/W	IRQ0 Interrupt Request Enable The IRQ0 interrupt request is enabled when this bit is set to 1.

4.3.4 Interrupt Enable Register 2 (IENR2)

IENR2 enables the direct transition, A/D converter, timer G, timer F, timer C, and asynchronous event counter interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	IENDT	0	R/W	Direct Transition Interrupt Request Enable The direct transition interrupt request is enabled when this bit is set to 1.
6	IENAD	0	R/W	A/D Converter Interrupt Request Enable The A/D converter interrupt request is enabled when this bit is set to 1.
5	—	0	R/W	Reserved This bit can be read from or written to.
4	IENTG	0	R/W	Timer G Interrupt Request Enable The timer G interrupt request is enabled when this bit is set to 1.
3	IENTFH	0	R/W	Timer FH Interrupt Request Enable The timer FH interrupt request is enabled when this bit is set to 1.
2	IENTFL	0	R/W	Timer FL Interrupt Request Enable The timer FL interrupt request is enabled when this bit is set to 1.
1	IENTC	0	R/W	Timer C Interrupt Request Enable The timer C interrupt request is enabled when this bit is set to 1.
0	IENEC	0	R/W	Asynchronous Event Counter Interrupt Request Enable The asynchronous event counter interrupt request is enabled when this bit is set to 1.

4.3.5 Interrupt Request Register 1 (IRR1)

IRR1 indicates the $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, $\overline{\text{IRQ3}}$, $\overline{\text{IRQ4}}$, and IRQAEC interrupt request status.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
4	IRRI4	0	R/W	$\overline{\text{IRQ4}}$ Interrupt Request Flag [Setting condition] The $\overline{\text{IRQ4}}$ pin is set as the interrupt input pin and the specified edge is detected [Clearing condition] Writing of 0 to this bit
3	IRRI3	0	R/W	$\overline{\text{IRQ3}}$ Interrupt Request Flag [Setting condition] The $\overline{\text{IRQ3}}$ pin is set as the interrupt input pin and the specified edge is detected [Clearing condition] Writing of 0 to this bit
2	IRREC2	0	R/W	IRQAEC Interrupt Request Flag [Setting condition] The IRQAEC pin is set as the interrupt input pin and the specified edge is detected [Clearing condition] Writing of 0 to this bit
1	IRRI1	0	R/W	$\overline{\text{IRQ1}}$ Interrupt Request Flag [Setting condition] The $\overline{\text{IRQ1}}$ pin is set as the interrupt input pin and the specified edge is detected [Clearing condition] Writing of 0 to this bit

Bit	Bit Name	Initial Value	R/W	Description
0	IRRI0	0	R/W	$\overline{\text{IRQ0}}$ Interrupt Request Flag [Setting condition] The $\overline{\text{IRQ0}}$ pin is set as the interrupt input pin and the specified edge is detected [Clearing condition] Writing of 0 to this bit

4.3.6 Interrupt Request Register 2 (IRR2)

IRR2 indicates the state of the direct transition, A/D converter, timer G, timer F, timer C, and asynchronous event counter interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	IRRDT	0	R/W	Direct Transition Interrupt Request Flag [Setting condition] Execution of a SLEEP instruction while the DTON bit in SYSCR2 is set to 1, so that a direct transition is made to sleep mode [Clearing condition] Writing of 0 to this bit
6	IRRAD	0	R/W	A/D Converter Interrupt Request Flag [Setting condition] When A/D conversion ends [Clearing condition] Writing of 0 to this bit
5	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
4	IRRTG	0	R/W	Timer G Interrupt Request Flag [Setting condition] The timer G input capture or overflow occurs. [Clearing condition] Writing of 0 to this bit

Bit	Bit Name	Initial Value	R/W	Description
3	IRRTFH	0	R/W	Timer FH Interrupt Request Flag [Setting condition] The timer FH compare match or overflow occurs [Clearing condition] Writing of 0 to this bit
2	IRRTFL	0	R/W	Timer FL Interrupt Request Flag [Setting condition] The timer FL compare match or overflow occurs [Clearing condition] Writing of 0 to this bit
1	IRRTC	0	R/W	Timer C Interrupt Request Flag [Setting condition] The timer C overflow or underflow occurs. [Clearing condition] Writing of 0 to this bit
0	IRREC	0	R/W	Asynchronous Event Counter Interrupt Request Flag [Setting condition] The asynchronous event counter overflows [Clearing condition] Writing of 0 to this bit

4.3.7 Wakeup Interrupt Request Register (IWPR)

IWPR has the $\overline{\text{WKP7}}$ to $\overline{\text{WKP0}}$ interrupt request status flags.

Bit	Bit Name	Initial Value	R/W	Description
7	IWPF7	0	R/W	WKP7 Interrupt Request Flag [Setting condition] The $\overline{\text{WKP7}}$ pin is set as the interrupt input pin and the specified edge is detected [Clearing condition] Writing of 0 to this bit
6	IWPF6	0	R/W	WKP6 Interrupt Request Flag [Setting condition] The $\overline{\text{WKP6}}$ pin is set as the interrupt input pin and the specified edge is detected [Clearing condition] Writing of 0 to this bit
5	IWPF5	0	R/W	WKP5 Interrupt Request Flag [Setting condition] The $\overline{\text{WKP5}}$ pin is set as the interrupt input pin and the specified edge is detected [Clearing condition] Writing of 0 to this bit
4	IWPF4	0	R/W	WKP4 Interrupt Request Flag [Setting condition] The $\overline{\text{WKP4}}$ pin is set as the interrupt input pin and the specified edge is detected [Clearing condition] Writing of 0 to this bit
3	IWPF3	0	R/W	WKP3 Interrupt Request Flag [Setting condition] The $\overline{\text{WKP3}}$ pin is set as the interrupt input pin and the specified edge is detected [Clearing condition] Writing of 0 to this bit

Bit	Bit Name	Initial Value	R/W	Description
2	IWPF2	0	R/W	WKP2 Interrupt Request Flag [Setting condition] The $\overline{\text{WKP2}}$ pin is set as the interrupt input pin and the specified edge is detected [Clearing condition] Writing of 0 to this bit
1	IWPF1	0	R/W	WKP1 Interrupt Request Flag [Setting condition] The $\overline{\text{WKP1}}$ pin is set as the interrupt input pin and the specified edge is detected [Clearing condition] Writing of 0 to this bit
0	IWPF0	0	R/W	WKP0 Interrupt Request Flag [Setting condition] The $\overline{\text{WKP0}}$ pin is set as the interrupt input pin and the specified edge is detected [Clearing condition] Writing of 0 to this bit

4.3.8 Interrupt Priority Registers A to F (IPRA to IPRF)

IPR sets mask levels (levels 2 to 0) for interrupts other than the NMI and address break. The correspondence between interrupt sources and IPR settings is shown in table 4.2.

Setting a value in the range from H'0 to H'3 in the 2-bit groups of bits 7 and 6, 5 and 4, 3 and 2, and 1 and 0 sets the mask level of the corresponding interrupt. Bits 3 to 0 in IPRE and bits 1 and 0 in IPRF are reserved.

Bit	Bit Name	Initial Value	R/W	Description
7	IPRn7	0	R/W	Set the mask level of the corresponding interrupt source.
6	IPRn6	0	R/W	00: Mask level 0 (Lowest)
				01: Mask level 1
				1x: Mask level 2 (Highest)
5	IPRn5	0	R/W	Set the mask level of the corresponding interrupt source.
4	IPRn4	0	R/W	00: Mask level 0 (Lowest)
				01: Mask level 1
				1x: Mask level 2 (Highest)
3	IPRn3	0	R/W	Set the mask level of the corresponding interrupt source.
2	IPRn2	0	R/W	00: Mask level 0 (Lowest)
				01: Mask level 1
				1x: Mask level 2 (Highest)
1	IPRn1	0	R/W	Set the mask level of the corresponding interrupt source.
0	IPRn0	0	R/W	00: Mask level 0 (Lowest)
				01: Mask level 1
				1x: Mask level 2 (Highest)

[Legend]
x: Don't care
n = A to F

4.3.9 Interrupt Mask Register (INTM)

INTM is an 8-bit readable/writable register that controls 3-level interrupt masking depending on the combination of the INTM0 and INTM1 bits.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
1	INTM1	0	R/W	Set the interrupt mask level.
0	INTM0	0	R/W	1x: Mask an interrupt with mask level 1 or less 01: Mask an interrupt with mask level 0 00: Accept all interrupts

[Legend]
x: Don't care

4.4 Interrupt Sources

4.4.1 External Interrupts

There are 14 external interrupts: NMI, WKP7 to WKP0, IRQ4, IRQ3, IRQAEC, IRQ1, and IRQ0.

(1) NMI Interrupt

NMI is the highest-priority interrupt, and is always accepted by the CPU regardless of the state of the I bit in CCR. The NMIEG bit in IEGR can be used to select whether an interrupt is requested at a rising edge or a falling edge on the NMI pin.

(2) WKP7 to WKP0 Interrupts

WKP7 to WKP0 interrupts are requested by the rising or falling edge input signals at the $\overline{\text{WKP7}}$ to $\overline{\text{WKP0}}$ pins.

When the rising or falling edge is input while the $\overline{\text{WKP7}}$ to $\overline{\text{WKP0}}$ pin functions are selected by PMR5, the corresponding bit in IWPR is set to 1 and an interrupt request is generated.

Clearing the IENWP bit in IENR1 to 0 disables the wakeup interrupt request to be accepted. Setting the I bit in CCR to 1 masks all interrupts.

When exception handling for the WKP7 to WKP0 interrupts is accepted, the I bit in CCR is set to 1. The interrupt mask level can be set by IPR.

(3) IRQ4, IRQ3, IRQ1, and IRQ0 Interrupts

IRQ4, IRQ3, IRQ1, and IRQ0 interrupts are requested by input signals at $\overline{\text{IRQ4}}$, $\overline{\text{IRQ3}}$, $\overline{\text{IRQ1}}$, and $\overline{\text{IRQ0}}$ pins.

Using the IEG4, IEG3, IEG1, and IEG0 bits in IEGR, it is possible to select whether an interrupt is generated by a rising or falling edge at $\overline{\text{IRQ4}}$, $\overline{\text{IRQ3}}$, $\overline{\text{IRQ1}}$, and $\overline{\text{IRQ0}}$ pins.

When the specified edge is input while the $\overline{\text{IRQ4}}$, $\overline{\text{IRQ3}}$, $\overline{\text{IRQ1}}$, and $\overline{\text{IRQ0}}$ pin functions are selected by PFCR, PMRF, PMRE, PMRB, and PMR9, the corresponding bit in IRR1 is set to 1 and an interrupt request is generated.

Clearing the IEN4, IEN3, IEN1, and IEN0 bits in IENR1 to 0 disables the interrupt request to be accepted. Setting the I bit in CCR to 1 masks all interrupts.

The interrupt mask level can be set by IPR.

(4) IRQAEC Interrupts

An IRQAEC interrupt is requested by an input signal at the IRQAEC pin or IECPWM (PWM output for the AEC). When the IRQAEC pin is used as an external interrupt pin, clear the ECPWME bit in AEGSR to 0.

Using the AIEGS1 and AIEGS0 bits in AEGSR, it is possible to select whether an interrupt is generated by a rising edge, falling edge, or both edges.

When the IENEC2 bit in IENR1 is set to 1 and the specified edge is input, the corresponding bit in IRR1 is set to 1 and an interrupt request is generated.

When exception handling for the IRQAEC interrupt is accepted, the I bit in CCR is set to 1.

The interrupt mask level can be set by IPR.

4.4.2 Internal Interrupts

Internal interrupts generated from the on-chip peripheral modules have the following features:

- For each on-chip peripheral module, there are flags that indicate the interrupt request status, and enable bits that select enabling or disabling of these interrupts. Internal interrupts can be controlled independently. If an enable bit is set to 1, an interrupt request is sent to the interrupt controller.
- The interrupt mask level can be set by IPR.

4.5 Interrupt Exception Handling Vector Table

Table 4.2 shows interrupt exception handling sources, vector addresses, and interrupt mask levels. For mask levels, the lower the vector number, the higher the mask level. Mask levels within a module are fixed. Mask levels of interrupts other than NMI and address break can be modified by IPR.

Table 4.2 Interrupt Sources, Vector Addresses, and Interrupt Mask Levels

Origin of Interrupt Source	Name	Vector Number	Vector Address	IPR	Mask Level
$\overline{\text{RES}}$, WDT	Reset	0	H'000000	—	High
NMI	NMI	3	H'00000C		
Address break	Break conditions satisfied	5	H'000014		
External pins	IRQ0	6	H'000018	IPRA7, IPRA6	
	IRQ1	7	H'00001C	IPRA5, IPRA4	
	IRQAEC	8	H'000020	IPRA3, IPRA2	
	IRQ3	9	H'000024	IPRA1, IPRA0	
	IRQ4	10	H'000028		
	WKP0	11	H'00002C	IPRB7, IPRB6	
	WKP1	12	H'000030		
	WKP2	13	H'000034		
	WKP3	14	H'000038		
	WKP4	15	H'00003C		
	WKP5	16	H'000040		
	WKP6	17	H'000044		
	WKP7	18	H'000048		
RTC	0.25-second overflow	19	H'00004C	IPRB5, IPRB4	
	0.5-second overflow	20	H'000050		
	Second periodic overflow	21	H'000054		
	Minute periodic overflow	22	H'000058		
	Hour periodic overflow	23	H'00005C		
	Day periodic overflow	24	H'000060		
	Week periodic overflow	25	H'000064		
	Free-running overflow	26	H'000068		Low

Origin of Interrupt Source	Name	Vector Number	Vector Address	IPR	Mask Level
WDT	WDT overflow (interval timer)	27	H'00006C	IPRB3, IPRB2	High
AEC	AEC overflow	28	H'000070	IPRB1, IPRB0	
TPU_1	TG1A (TG1A input capture/compare match)	29	H'000074	IPRC7, IPRC6	
	TG1B (TG1B input capture/compare match)	30	H'000078		
	TCI1V (overflow 1)	31	H'00007C		
TPU_2	TG2A (TG2A input capture/compare match)	32	H'000080	IPRC5, IPRC4	
	TG2B (TG2B input capture/compare match)	33	H'000084		
	TCI2V (overflow 2)	34	H'000088		
Timer F	Timer FL compare match Timer FL overflow	35	H'00008C	IPRC3, IPRC2	
	Timer FH compare match Timer FH overflow	36	H'000090		
SCI4*	Receive data full/transmit data empty Transmit end/receive error	37	H'000094	IPRC1, IPRC0	
SCI3_1	Transmit completion/transmit data empty Receive data full/overrun error Framing error/parity error	38	H'000098	IPRD7, IPRD6	
SCI3_2	Transmit completion/transmit data empty Receive data full/overrun error Framing error/parity error	39	H'00009C	IPRD5, IPRD4	
IIC2	Transmit data empty/transmit end Receive data full/stop condition detection NACK detection Arbitration/overrun error	40	H'0000A0	IPRD3, IPRD2	
10-bit A/D	A/D conversion end	42	H'0000A8	IPRE7, IPRE6	
(SLEEP instruction execution)	Direct transition	43	H'0000AC	IPRE5, IPRE4	Low

Origin of Interrupt Source	Name	Vector Number	Vector Address	IPR	Mask Level
Timer C	Timer C overflow/underflow	53	H'0000D4	IPRF7, IPRF6	High
Timer G	Timer G input capture Timer G overflow	54	H'0000D8	IPRF5, IPRF4	
SCI3_3	Transmit completion/transmit data empty Receive data full/overrun error Framing error/parity error	55	H'0000DC	IPRF3, IPRF2	Low

Note: * Supported only by the flash version.

4.6 Operation

NMI and address break interrupts are accepted at all times except in the reset state. In the case of IRQ interrupts, WKP interrupts, and on-chip peripheral module interrupts, an enable bit is provided for each interrupt. Clearing an enable bit to 0 disables the corresponding interrupt request. Interrupt sources for which the enable bits are set to 1 are controlled by the interrupt controller.

Table 4.3 shows the interrupt control states. Figure 4.2 shows a flowchart of the interrupt acceptance operation.

Four-level interrupt masking is controlled according to the combination of the I bit in CCR and the INTM1 and INTM0 bits in INTM.

Table 4.3 Interrupt Control States

CCR	INTM		
I	**INTM1**	**INTM0**	**States**
1	x	x	All interrupts other than NMI and address break are masked.
0	1	x	Interrupts with mask level 1 or less are masked.
	0	1	Interrupts with mask level 0 are masked.
	0	0	All interrupts are accepted.

[Legend]
x: Don't care

1. If an interrupt source whose enable bit is set to 1 occurs, an interrupt request is sent to the interrupt controller.
2. With referring to the INTM1 and INTM0 bits in INTM and the I bit in CCR, control the following.
   - The interrupt request is held pending when the I bit is set to 1.
   - When the I bit is cleared to 0 and INTM1 bit is set to 1, interrupts with mask level 1 or less are held pending.
   - When the I bit is cleared to 0, INTM1 bit is cleared to 0, and INTM0 bit is set to 1, interrupt requests with mask level 0 are held pending.
   - When the I bit, INTM1 bit, and INTM0 bit are all cleared to 0, all interrupt requests are accepted.

3. If a conflict occurs between interrupt requests that are not held pending due to the settings of the IMTM1, IMTN0 bits in INTM or the I bit in CCR, the interrupt request with the highest mask level according to table 4.2 is selected regardless of the IPR setting.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. PC and CCR are saved to the stack area by interrupt exception handling.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except NMI and address break.
7. The CPU generates a vector address for the accepted interrupt and starts interrupt handling by reading the interrupt routine start address in the vector table.

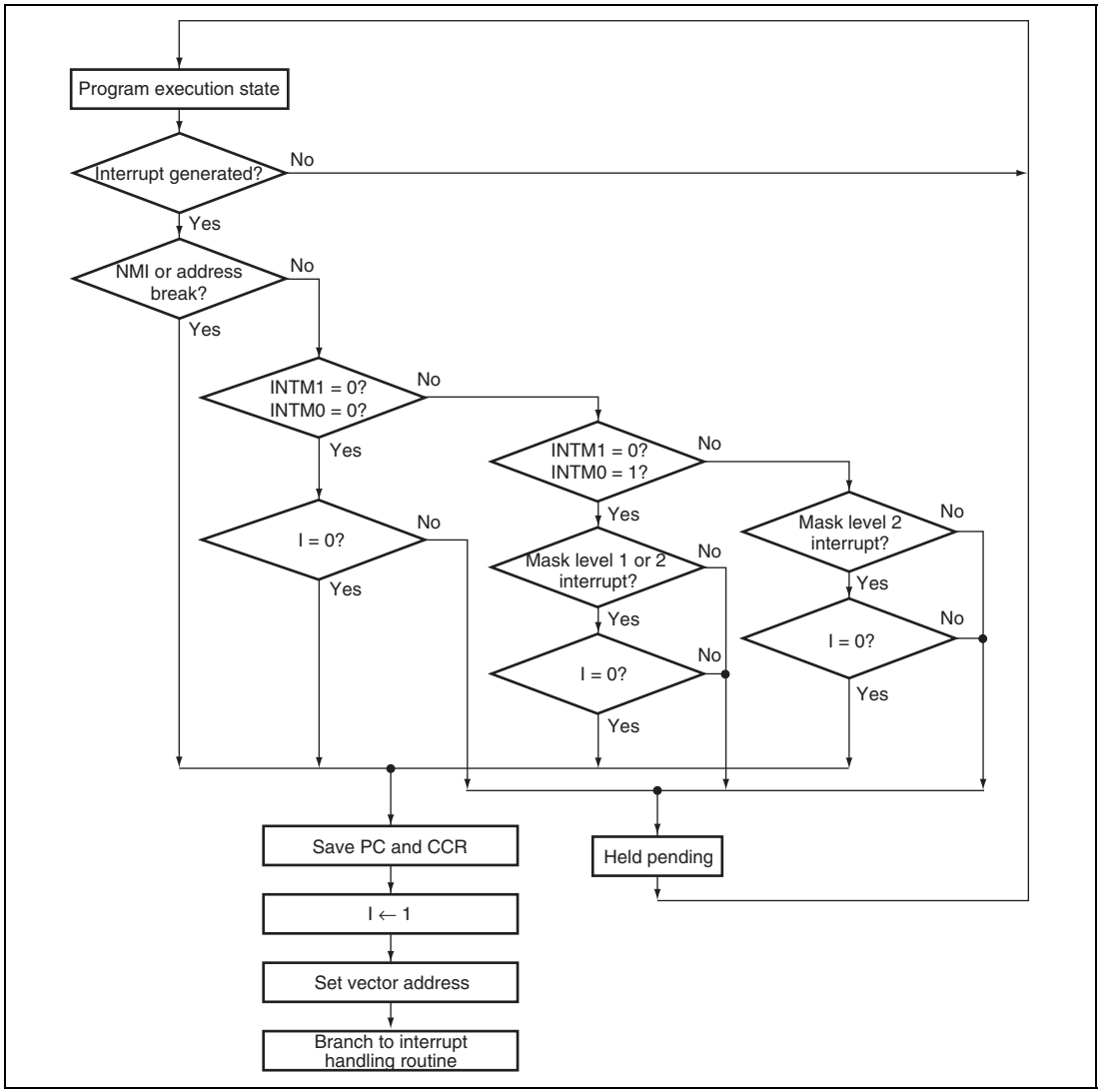


Figure 4.2 Flowchart of Procedure Up to Interrupt Acceptance

4.6.1 Interrupt Exception Handling Sequence

Figure 4.3 shows the interrupt exception handling sequence.

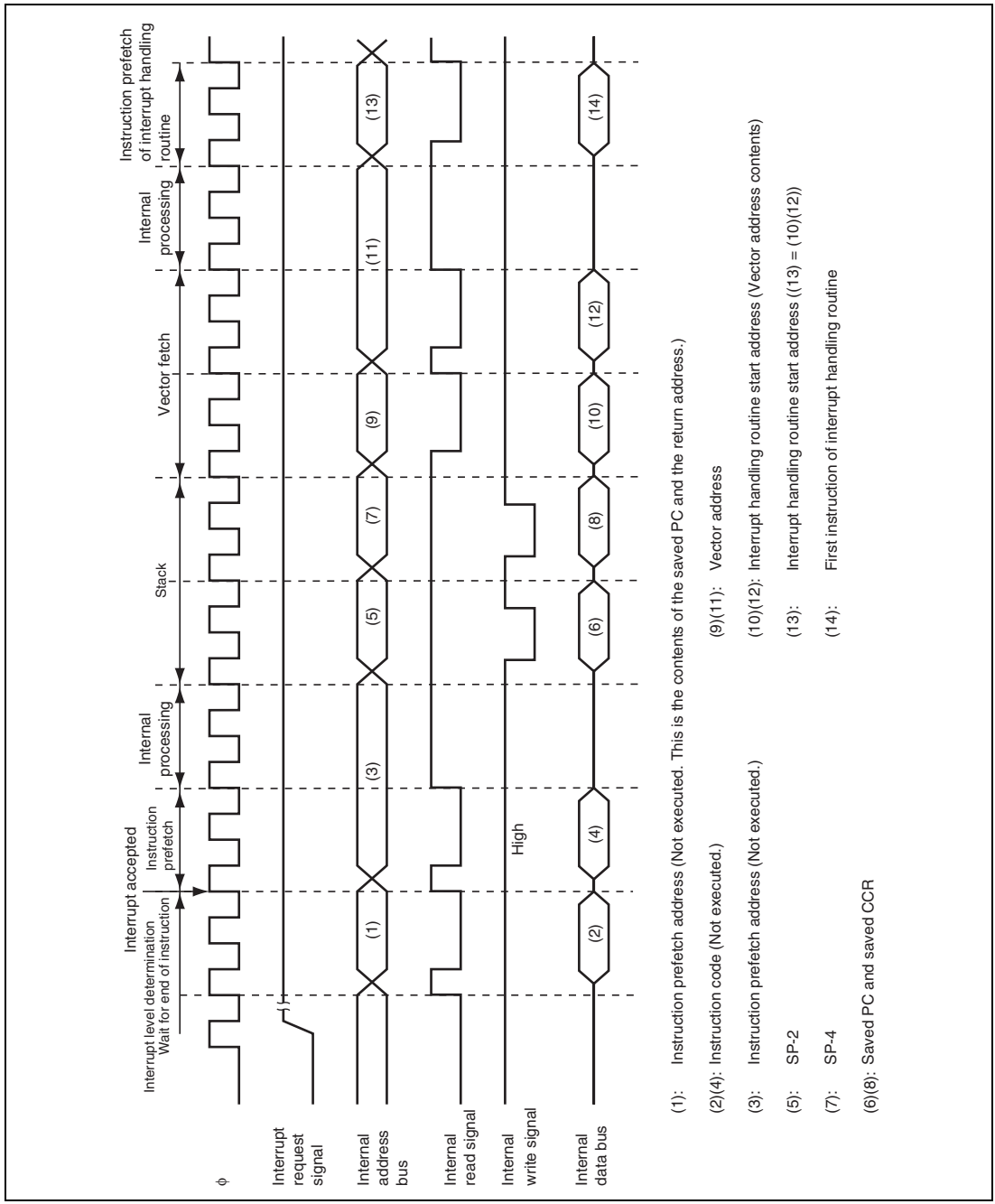


Figure 4.3 Interrupt Exception Handling Sequence

4.6.2 Interrupt Response Times

Table 4.4 shows interrupt response times – the interval between generation of an interrupt request and execution of the first instruction in the interrupt handling routine.

Table 4.4 Interrupt Response Times (States)

No.	Execution Status	Number of States
1	Interrupt mask level determination	1 or 2*[1]
2	Maximum number of wait states until executing instruction ends	1 to 23
3	PC, CCR stack	4
4	Vector fetch	4
5	Instruction fetch*[2]	4
6	Internal processing*[3]	4
	Total	18 to 41

Notes: 1. One state in case of an internal interrupt (2 states in case of an external interrupt).
2. Prefetch after interrupt acceptance and interrupt handling routine prefetch.
3. Internal processing after interrupt acceptance and internal processing after vector fetch.

4.7 Usage Notes

4.7.1 Contention between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to disable interrupts, the disabling becomes effective after execution of the instruction. When an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, and if an interrupt is generated during execution of the instruction, the interrupt concerned will still be enabled on completion of the instruction, and so interrupt exception handling for that interrupt will be executed on completion of the instruction. However, if there is an interrupt request with higher mask level than that interrupt, interrupt exception handling will be executed for the interrupt with a higher mask level interrupt, and the interrupt with a lower mask level interrupt will be ignored. The same also applies when an interrupt source flag is cleared to 0.

Figure 4.4 shows an example in which the TGIEA bit in TIER of the 16-bit timer pulse unit (TPU) is cleared to 0.

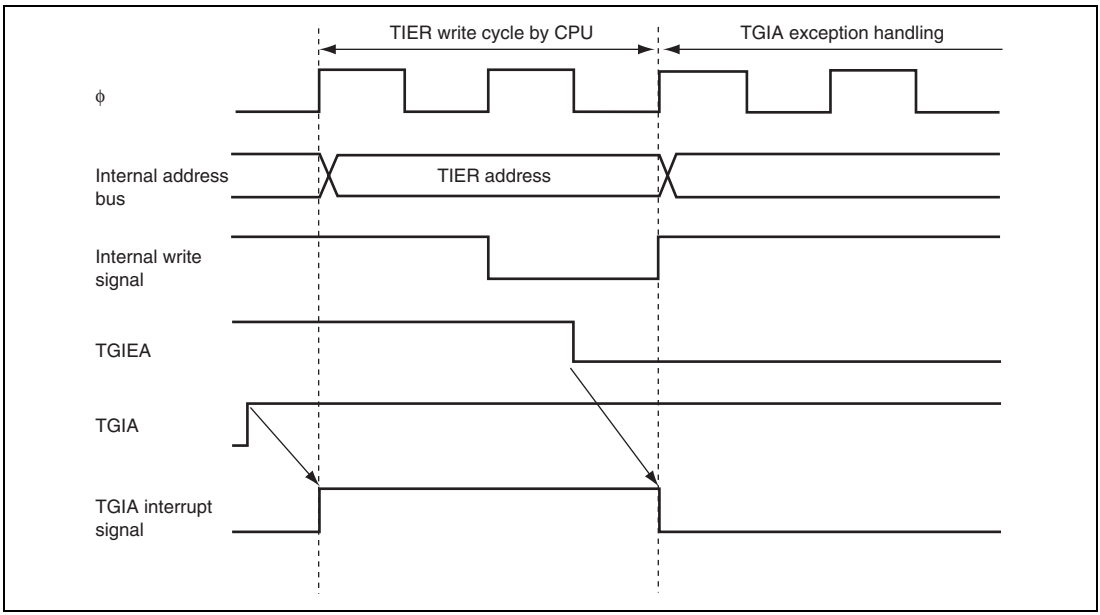


Figure 4.4 Contention between Interrupt Generation and Disabling

The above contention will not occur if an enable bit or interrupt source flag is cleared to 0 while the interrupt is masked.

4.7.2 Instructions that Disable Interrupts

The instructions that disable interrupts are LDC, ANDC, ORC, and XORC.

When an interrupt request is generated, an interrupt is requested to the CPU after the interrupt controller has determined the mask level. At that time, if the CPU is executing an instruction that disables interrupts, the CPU always executes the next instruction after the instruction execution is completed.

4.7.3 Interrupts during Execution of EEPMOV Instruction

Interrupt operation differs between the EEPMOV.B instruction and the EEPMOV.W instruction.

With the EEPMOV.B instruction, an interrupt request (including NMI) issued during transfer is not accepted until the transfer is completed.

With the EEPMOV.W instruction, even if an interrupt request other than the NMI is issued during transfer, the interrupt is not accepted until the transfer is completed. If the NMI interrupt request is issued, NMI exception handling starts at a break in the transfer cycle. The PC value saved on the stack in this case is the address of the next instruction.

Therefore, if an NMI interrupt is generated during execution of an EEPMOV.W instruction, the following coding should be used.

```
L1:  EEPMOV.W
     MOV.W      R4,R4
     BNE        L1
```

4.7.4 IENR Clearing

When an interrupt request is disabled by clearing the interrupt enable register or when the interrupt request register is cleared, the interrupt request should be masked (I bit = 1). If the above operation is executed while the I bit is 0 and contention between the instruction execution and the interrupt request generation occurs, exception handling, which corresponds to the interrupt request generated after instruction execution of the above operation is completed, is executed.

Section 5 Clock Pulse Generator

The clock pulse generator incorporated into this LSI consists of a system clock pulse generator circuit that consists of a system clock oscillator, system clock divider, and an on-chip oscillator for the system clock, and a subclock pulse generator circuit that consists of a subclock oscillator and subclock divider.

Figure 5.1 is a block diagram of the clock pulse generator.

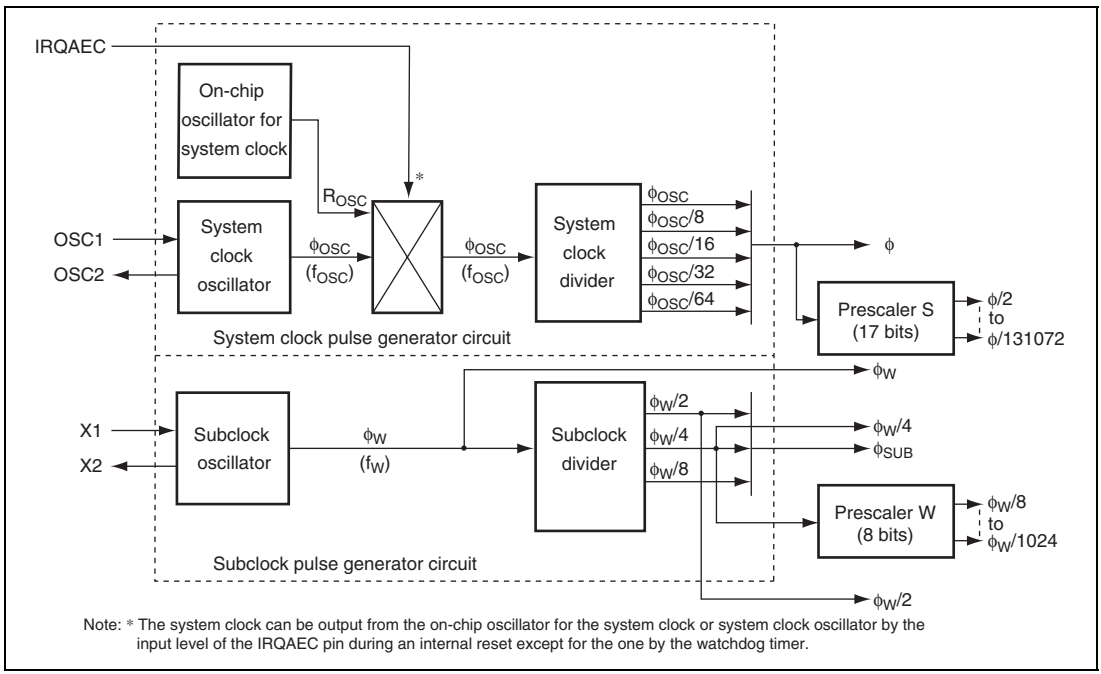


Figure 5.1 Block Diagram of Clock Pulse Generator

The basic clock signals that drive the CPU and on-chip peripheral modules are the system clock (ϕ) and subclock (ϕ_{SUB}). Prescaler S frequency-divides the clock signal S to produce clock signals at rates from ϕ/131072 to ϕ/2, and prescaler W frequency-divides the watch clock ϕ_w/4, which is the watch clock frequency-divided by four, to produce clock signals from at rates from ϕ_w/1024 to ϕ_w/8. Both the system clock and subclock signals are provided to the on-chip peripheral modules.

Since the on-chip oscillator for the system clock is available, the system clock can be selected to be output from the on-chip oscillator for the system clock or system clock oscillator by the input level of the IRQAEC pin.

5.1 Register Description

- SUB32k control register (SUB32CR)
- Oscillator Control Register (OSCCR)

5.1.1 SUB32k Control Register (SUB32CR)

SUB32CR controls whether the subclock oscillator is operating or stopped.

Bit	Bit Name	Initial Value	R/W	Description
7	32KSTOP	0	R/W	Subclock Oscillator Operation Control Controls whether the subclock oscillator is operating or stopped. When the subclock oscillator is not used, set this bit to 1. 0: Subclock oscillator operates 1: Subclock oscillator stops
6	—	0	R/W	Reserved This bit can be read from or written to.
5 to 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

5.1.2 Oscillator Control Register (OSCCR)

OSCCR is used to control the built-in feedback resistor and includes the IRQAEC and OSC flags.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved This bit can be read from or written to.
6	RFCUT	0	R/W	Built-In Feedback Resistor Control Selects whether to use the built-in feedback resistor of the system clock oscillator when the external clock is being input or the on-chip oscillator for the system clock is selected. When the external clock is being input or the on-chip oscillator for the system clock is in use, set this bit and then temporarily enter standby mode, watch mode, or subactive mode. After any of these modes is entered, the built-in feedback oscillator is used or not used according to this specification. 0: Built-in feedback resistor is used with the system clock oscillator 1: Built-in feedback resistor is not used with the system clock oscillator
5	—	0	R/W	Reserved The write value should always be 0.
4	—	0	R/W	Reserved This bit can be read from or written to.
3	—	0	R/W	Reserved The write value should always be 0.
2	IRQAECF	—*	R	IRQAEC Flag This bit indicates the level at which the IRQAEC pin is to be set during resets. 0: IRQAEC pin set to GND during resets 1: IRQAEC pin set to Vcc during resets

Bit	Bit Name	Initial Value	R/W	Description
1	OSCF	—*	R	OSC Flag This bit indicates which oscillator is acting as the system clock pulse generator. 0: The system clock oscillator is the generator (operation of the on-chip oscillator for system clock stopped) 1: The on-chip oscillator for the system clock is the generator (system clock oscillator stopped)
0	—	0	R/W	Reserved The write value should always be 0.

Note: * The initial value depends on the IRQAEC pin state. For details, see table 5.1.

5.2 System Clock Generator

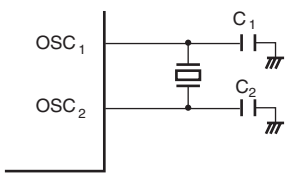
Clock pulses can be supplied to the system clock divider either by connecting a crystal or ceramic resonator, or by providing an external clock input.

The system clock oscillator or the on-chip oscillator for the system clock is selectable, as shown in figure 5.1. For details on how to select the oscillator, see section 5.2.4, Selecting On-Chip Oscillator for System Clock.

5.2.1 Connecting Crystal Resonator

Figure 5.2 shows a typical method of connecting a crystal resonator. An AT-cut parallel-resonance crystal resonator should be used.

For precautionary notes on connection of the crystal resonator, see section 5.5.2, Notes on Board Design.



R1 = 1 MΩ ±20%

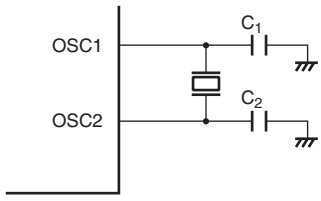
Note: Consult with the crystal resonator manufacturer to determine the circuit constants.

Frequency	Manufacturer	Product Type	C_1, C_2 Recommendation Value
4.19 MHz	KYOCERA KINSEKI CORP.	HC-49/U-S	12 pF ±20%

Figure 5.2 Typical Connection to Crystal Resonator

5.2.2 Connecting Ceramic Resonator

Figure 5.3 shows a typical method of connecting a ceramic resonator.



Frequency	Manufacturer	Product Type	C_1, C_2 Recommended Value
4.194 MHz	Murata Manufacturing Co., Ltd.	CSTLS4M19G53-B0	15 pF (on-chip)
		CSTLS4M19G56-B0	47 pF (on-chip)

Figure 5.3 Typical Connection to Ceramic Resonator

5.2.3 External Clock Input Method

Connect an external clock signal to pin OSC1, and leave pin OSC2 open. Figure 5.4 shows a typical connection. The duty cycle of the external clock signal must be 45 to 55%.

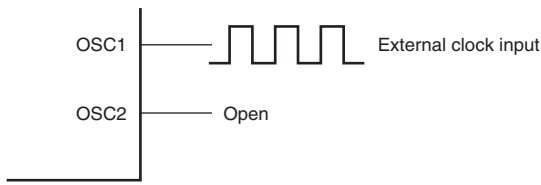


Figure 5.4 Example of External Clock Input

5.2.4 Selecting On-Chip Oscillator for System Clock

The on-chip oscillator is selected by the input level on the IRQAEC pin during a reset*. The selection of the system clock oscillator or on-chip oscillator for the system clock is as listed in table 5.1. The level being input on the IRQAEC pin during a reset should be fixed to either Vcc or GND, depending on the oscillator type to be selected. The level will be determined upon reset cancellation

When the on-chip oscillator for the system clock is selected, connection of a resonator to OSC1 or OSC2 is not necessary. In this case, the OSC1 pin should be fixed to Vcc or GND. The OSC2 pin should be left open.

Notes: When programming or erasing the flash memory, e.g. by on-board programming, the system clock oscillator should always be selected. Use of the on-chip emulator requires either a connected resonator or the supply of an external clock signal, even if the on-chip oscillator for the system clock is selected.

* This reset represents an external reset or power-on reset, but not a reset by the watchdog timer.

Table 5.1 Selection of the System Clock Oscillator or On-Chip Oscillator for the System Clock

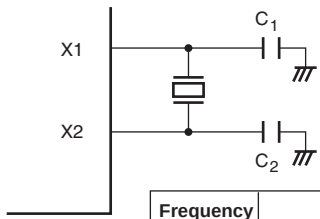
IRQAEC Input Level (During a Reset)	Oscillator of System Clock Pulse Generator Circuit	OSCF	IRQAECF
Low	System clock oscillator	0	0
High	On-chip oscillator for system clock	1	1

5.3 Subclock Generator

Subclocks can be supplied either by connecting a crystal resonator, or by providing external clock input.

5.3.1 Connecting 32.768-kHz/38.4-kHz Crystal Resonator

Figure 5.5 shows an example of connection to a 32.768-kHz or 38.4-kHz crystal resonator. Notes in section 5.5.2, Notes on Board Design, also apply to this connection.



Note: Consult with the crystal resonator manufacturer to determine the circuit constants.

Frequency	Manufacturer	Product Name	Equivalent Series Resistance
38.4 kHz	Epson Toyocom	C-4-TYPE	30 kΩ (max.)
32.768 kHz	Epson Toyocom	C-001R	35 kΩ (max.)

$C_1 = C_2 = 7$ pF (typ.)

Figure 5.5 Typical Connection to 32.768-kHz/38.4-kHz Crystal Resonator

1. When using a resonator other than the above, ensure optimal conditions by conducting sufficient evaluation of consistency in cooperation with the manufacturer of the resonator. Even if the above resonators or products equivalent to them are implemented, their oscillation characteristics are affected by the board design. Be sure to use the actual board to evaluate consistency as a system.
2. The consistency as a system has to be verified not only in a reset state (i.e., the $\overline{\text{RES}}$ pin is driven low) but also in a state where a reset state has been exited (i.e., the low-level $\overline{\text{RES}}$ signal has been driven high).

Figure 5.6 shows the equivalent circuit of the crystal resonator.

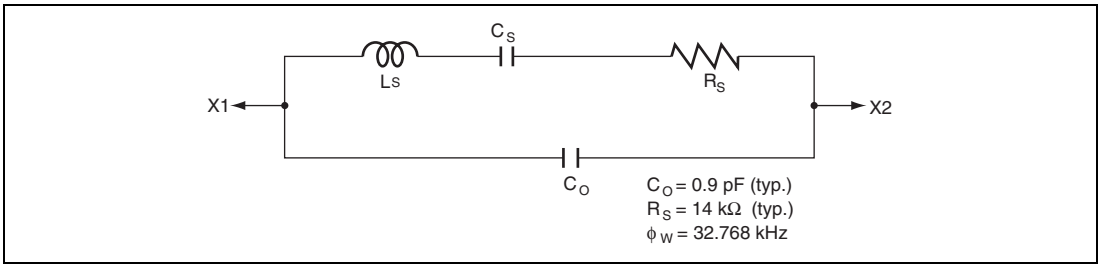


Figure 5.6 Equivalent Circuit of 32.768-kHz Crystal Resonator

5.3.2 Pin Connection when not Using Subclock

When the subclock is not used, connect the X1 pin to GND and leave the X2 pin open, as shown in figure 5.7.

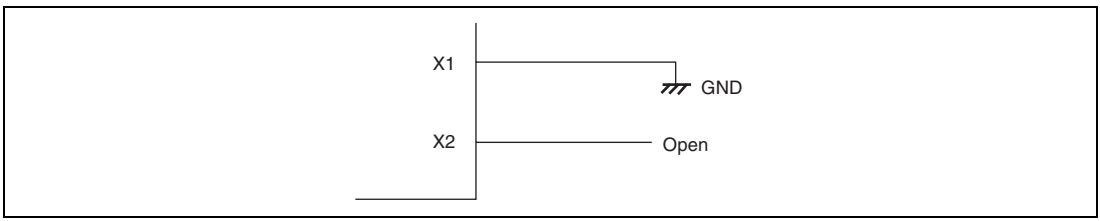


Figure 5.7 Pin Connection when not Using Subclock

5.3.3 How to Input the External Clock

Connect the external clock to the X1 pin and leave the X2 pin open, as shown in figure 5.8.

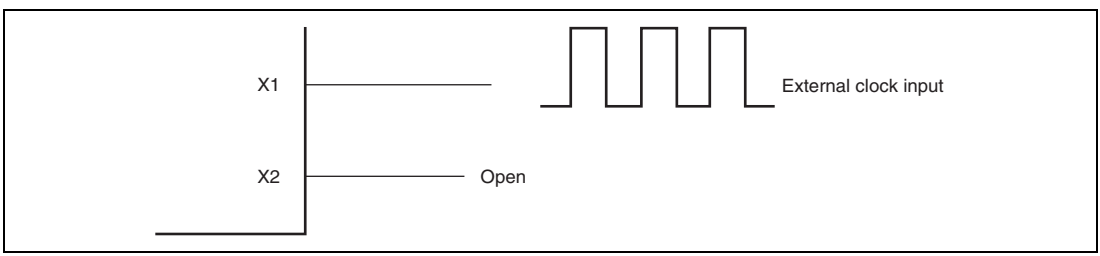


Figure 5.8 Pin Connection when Inputting External Clock

Frequency	Subclock (ϕw)
Duty	45% to 55%

5.4 Prescalers

This LSI has two prescalers (prescaler S and prescaler W), and each has its own input clock signal.

Prescaler S is a 17-bit counter that has the system clock (ϕ) as its input clock. Its prescaled outputs provide the internal clock signals that drive the on-chip peripheral modules.

Prescaler W is an 8-bit counter that has a frequency-divided signal ($\phi_w/4$) derived from the watch clock (ϕ_w) as its input clock. Its prescaled outputs provide the internal clock signals that drive the on-chip peripheral modules.

5.4.1 Prescaler S

Prescaler S is a 17-bit counter using the system clock (ϕ) as its input clock. A divided output is used as an internal clock of an on-chip peripheral module. Prescaler S is initialized to H'00000 at a reset, and starts counting up on exit from the reset state. Prescaler S stops and is initialized to H'00000 in standby mode, watch mode, subactive mode, and subsleep mode. The CPU cannot read from or write to prescaler S.

The output from prescaler S is shared by the on-chip peripheral modules. In active (medium-speed) mode and sleep mode (medium-speed), the clock input to prescaler S is determined by the division ratio designated by the MA1 and MA0 bits in SYSCR2.

5.4.2 Prescaler W

Prescaler W is an 8-bit counter that has a frequency-divided signal ($\phi_w/4$) derived from the watch clock (ϕ_w) as its input clock. This signal is further divided to produce internal clock signals for the on-chip peripheral modules. Prescaler W is initialized to H'00 by a reset, and starts counting up on exit from the reset state. Prescaler W stops in standby mode, but continues to operate in watch mode, subactive mode, and subsleep mode.

5.5 Usage Notes

5.5.1 Note on Resonators and Resonator Circuits

Resonator characteristics are closely related to board design. Therefore, resonators should be assigned after being carefully evaluated by the user in the masked ROM version and flash memory version, with referring to the examples shown in this section. Resonator circuit constants will differ depending on a resonator, stray capacitance in its mounting circuit, and other factors. Suitable constants should be determined in consultation with the resonator manufacturer. Design the circuit so that the oscillator pin is never applied voltages exceeding its maximum rating. Figure 5.9 shows an example of crystal and ceramic resonator arrangement.

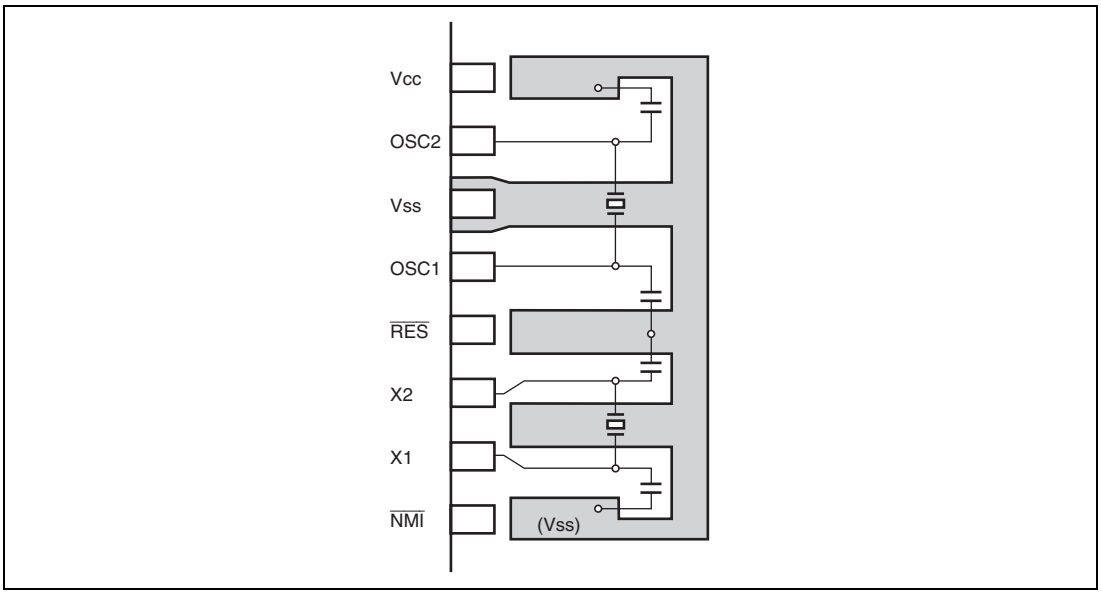


Figure 5.9 Example of Crystal and Ceramic Resonator Arrangement

Figure 5.10 (1) shows an example measuring circuit with the negative resistance recommended by the resonator manufacturer. Note that if the negative resistance of the circuit is less than that recommended by the resonator manufacturer, it may be difficult to start the main oscillator.

If it is determined that oscillation does not occur because the negative resistance is lower than the level recommended by the resonator manufacturer, the circuit must be modified as shown in figure 5.10 (2) through (4). Which of the modification suggestions to use and the capacitor capacitance should be decided based upon evaluation results such as the negative resistance and the frequency deviation.

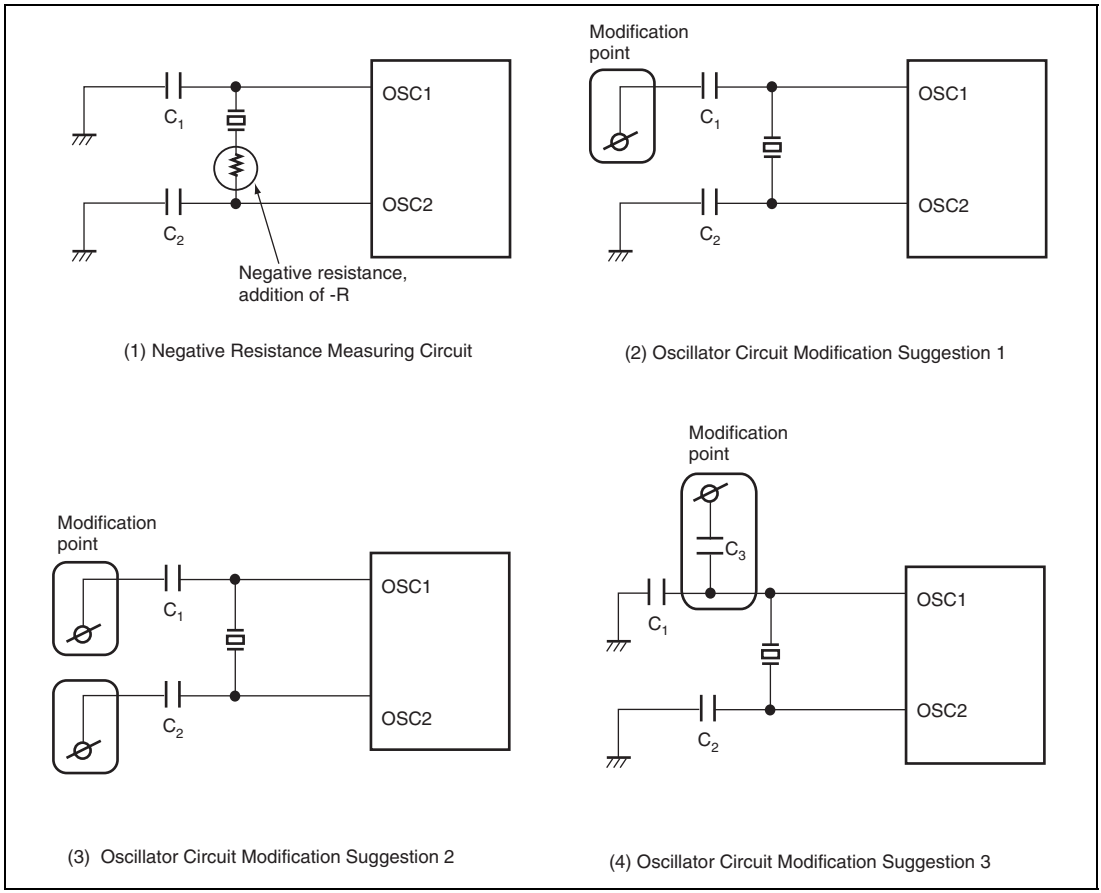


Figure 5.10 Negative Resistance Measurement and Circuit Modification Suggestions

5.5.2 Notes on Board Design

When using a crystal resonator (ceramic resonator), place the resonator and its load capacitors as close as possible to the OSC1 and OSC2 pins. Other signal lines should be routed away from the resonator circuit to prevent induction from interfering with correct oscillation (see figure 5.11).

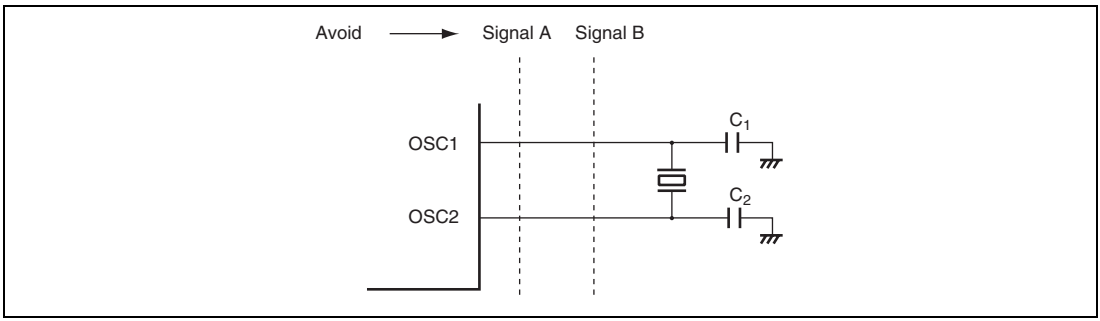


Figure 5.11 Example of Incorrect Board Design

Note: When a crystal resonator or ceramic resonator is connected, consult with the crystal resonator and ceramic resonator manufacturers to determine the circuit constants because the constants differ according to the resonator, stray capacitance of the mounting circuit, and so on.

5.5.3 Definition of Oscillation Stabilization Wait Time

Figure 5.12 shows the oscillation waveform (OSC2), system clock (ϕ), and microcontroller operating mode when a transition is made from standby mode, watch mode, or subactive mode, to active (high-speed/medium-speed) mode, with a resonator connected to the system clock oscillator.

As shown in figure 5.12, when a transition is made to active (high-speed/medium-speed) mode, from standby mode, watch mode, or subactive mode, in which the system clock oscillator is halted, the sum of the following two times (oscillation start time and wait time) is required.

(1) Oscillation Start Time

The time from the point at which the system clock oscillator oscillation waveform starts to change when an interrupt is generated, until the system clock starts to be generated.

(2) Wait Time

After the system clock is generated, the time required for the amplitude of the oscillation waveform to increase, the oscillation frequency to stabilize, and the CPU and peripheral functions to begin operating.

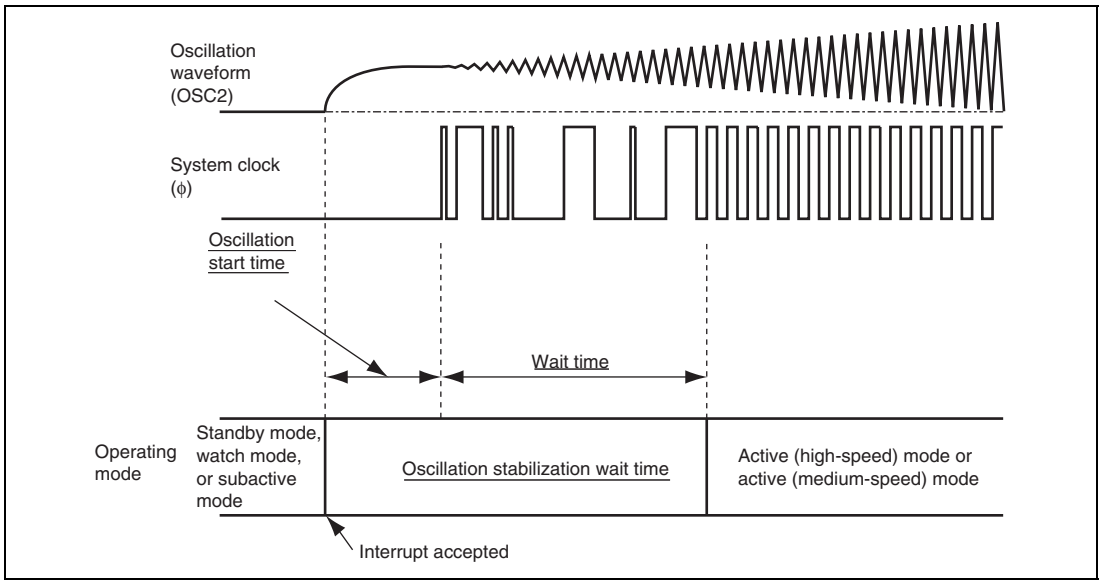


Figure 5.12 Oscillation Stabilization Wait Time

As the oscillation stabilization wait time required is the same as the oscillation stabilization time (t_{rc}) at power-on, specified in the AC characteristics, set the STS2 to STS0 bits in SYSCR1 to specify the time longer than the oscillation stabilization time (t_{rc}).

Therefore, when a transition is made from standby mode, watch mode, or subactive mode, to active (high-speed/medium-speed) mode, with an resonator connected to the system clock oscillator, careful evaluation must be carried out on the mounting circuit before deciding the oscillation stabilization wait time. For the wait time, secure the time required for the amplitude of the oscillation waveform to increase and the oscillation frequency to stabilize. In addition, since the oscillation start time differs according to mounting circuit constants, stray capacitance, and so forth, suitable constants should be determined in consultation with the resonator manufacturer.

5.5.4 Note on Subclock Stop State

To stop the subclock, a state transition should not be made except to mode in which the system clock operates. If the state transition is made to other mode, it may result in incorrect operation.

5.5.5 Note on the Oscillation Stabilization of Resonators

When a microcontroller operates, the internal power supply potential fluctuates slightly in synchronization with the system clock. Depending on the individual resonator characteristics, the oscillation waveform amplitude may not be sufficiently large immediately after the oscillation stabilization wait time, making the oscillation waveform susceptible to influence by fluctuations in the power supply potential. In this state, the oscillation waveform may be disrupted, leading to an unstable system clock and incorrect operation of the microcontroller.

If incorrect operation occurs, change the setting of the standby timer select bits 3 to 0 (STS3 to STS0) (bit 0 in the system control register 3 (SYSCR3) and bits 6 to 4 in the system control register 1 (SYSCR1)) to give a longer wait time.

For example, if incorrect operation occurs with a wait time setting of 1,024 states, check the operation with a wait time setting of 2,048 states or more. If the same kind of incorrect operation occurs after a reset as after a state transition, hold the $\overline{\text{RES}}$ pin low for a longer period.

5.5.6 Note on Using On-Chip Power-On Reset

The power-on reset circuit in this LSI adjusts the reset clear time by the capacitor capacitance, which is externally connected to the $\overline{\text{RES}}$ pin. The external capacitor capacitance should be adjusted to secure the oscillation stabilization time before reset clearing. For details, refer to section 22, Power-On Reset Circuit.

5.5.7 Note on Using the On-Chip Emulator

When using the on-chip emulator, programming and erasure of the flash memory require an accurate system clock signal. The frequency of the on-chip oscillator for the system clock varies according to voltage and temperature conditions. Thus, when the on-chip emulator is in use, a resonator must be connected between the OSC1 and OSC2 pins or an external clock signal must be supplied. In this case, the on-chip emulator is driven by the on-chip oscillator for the system clock in the execution of user programs, and by the system clock oscillator in the programming and erasure of the flash memory. This selection is achieved by fixing the level of the IRQAEC pin to the low level during the reset period of the on-chip emulator.

Section 6 Power-Down Modes

This LSI has eight modes of operation after a reset. These include a normal active (high-speed) mode and seven power-down modes, in which power consumption is significantly reduced. The module standby function reduces power consumption by selectively halting on-chip module functions.

- Active (medium-speed) mode

  The CPU and all on-chip peripheral modules are operable on the system clock. The system clock frequency can be selected from $\phi_{OSC}/8$, $\phi_{OSC}/16$, $\phi_{OSC}/32$, and $\phi_{OSC}/64$.
- Subactive mode

  The CPU and all on-chip peripheral modules are operable on the subclock. The subclock frequency can be selected from $\phi_{w}/2$, $\phi_{w}/4$, and $\phi_{w}/8$.
- Sleep (high-speed) mode

  The CPU halts. On-chip peripheral modules are operable on the system clock.
- Sleep (medium-speed) mode

  The CPU halts. On-chip peripheral modules are operable on the system clock. The system clock frequency can be selected from $\phi_{OSC}/8$, $\phi_{OSC}/16$, $\phi_{OSC}/32$, and $\phi_{OSC}/64$.
- Subsleep mode

  The CPU halts. The on-chip peripheral modules are operable on the subclock. The subclock frequency can be selected from $\phi_{w}/2$, $\phi_{w}/4$, and $\phi_{w}/8$.
- Watch mode

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

  The CPU and all on-chip peripheral modules halt.
- Module standby function

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

Note: In this manual, active (high-speed) mode and active (medium-speed) mode are collectively called active mode, and sleep (high-speed) mode and sleep (medium-speed) mode are collectively called sleep mode.

6.1 Register Descriptions

The registers related to power-down modes are as follows.

- System control register 1 (SYSCR1)
- System control register 2 (SYSCR2)
- System control register 3 (SYSCR3)
- Clock halt registers 1 to 3 (CKSTPR1 to CKSTPR3)

6.1.1 System Control Register 1 (SYSCR1)

SYSCR1 controls the power-down modes with SYSCR2 and SYSCR3.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby Selects the mode to transit after the execution of the SLEEP instruction. 0: A transition is made to sleep mode or subsleep mode. 1: A transition is made to standby mode or watch mode. For details, see table 6.2.
6	STS2	0	R/W	Standby Timer Select 2 to 0 Specify the time the CPU and peripheral modules wait for stable clock operation after exiting from standby mode, subactive mode, or watch mode to active mode or sleep mode. These bits should be specified together with the STS3 bit in SYSCR3 according to the operating frequency so that the wait time is at least equal to the oscillation stabilization time. The relationship between the specified value and the number of wait states is shown in table 6.1. When an external clock is to be used, the minimum value (STS3 = 0, STS2 = 1, STS1 = 0, STS0 = 1) is recommended. When the on-chip oscillator for the system clock is to be used, four states (STS3 = 1, STS2 = 1, STS1 = 0, STS0 = 1) are recommended. If a setting other than the recommended value is made, operation may start before the end of the wait time.
5	STS1	0	R/W	
4	STS0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
3	LSON	0	R/W	Selects the system clock (ϕ) or subclock (ϕ_{SUB}) as the CPU operating clock when watch mode is cleared. 0: The CPU operates on the system clock (ϕ) 1: The CPU operates on the subclock (ϕ_{SUB})
2	TMA3	0	R/W	Selects the mode to which the transition is made after the SLEEP instruction is executed with bits SSBY and LSON in SYSCR1 and bits DTON and MSON in SYSCR2. For details, see table 6.2.
1	MA1	1	R/W	Active Mode Clock Select 1 and 0
0	MA0	1	R/W	Select the operating clock in active (medium-speed) mode and sleep (medium-speed) mode. The MA1 and MA0 bits should be written to in active (high-speed) mode or subactive mode. 00: $\phi_{OSC}/8$ 01: $\phi_{OSC}/16$ 10: $\phi_{OSC}/32$ 11: $\phi_{OSC}/64$

6.1.2 System Control Register 2 (SYSCR2)

SYSCR2 controls the power-down modes with SYSCR1 and SYSCR3.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
4	NESEL	1	R/W	Noise Elimination Sampling Frequency Select This bit selects the sampling frequency of ϕ_{OSC} when ϕ_W is sampled. When a system clock is used, clear this bit to 0.When the on-chip oscillator is selected, set this bit to 1. 0: Sampling rate is ϕ_{OSC}/16. 1: Sampling rate is ϕ_{OSC}/4.
3	DTON	0	R/W	Direct Transfer on Flag Selects the mode to which the transition is made after the SLEEP instruction is executed with bits SSBY, TMA3, and LSON in SYSCR1 and bit MSON in SYSCR2. For details, see table 6.2.
2	MSON	0	R/W	Medium Speed on Flag After standby, watch, or sleep mode is cleared, this bit selects active (high-speed) or active (medium-speed) mode. 0: Operation in active (high-speed) mode 1: Operation in active (medium-speed) mode
1	SA1	0	R/W	Subactive Mode Clock Select 1 and 0
0	SA0	0	R/W	Select the operating clock frequency in subactive and subsleep modes. The operating clock frequency changes to the set frequency after the SLEEP instruction is executed. 00: ϕ_W/8 01: ϕ_W/4 10: ϕ_W/2 11: Setting prohibited

6.1.3 System Control Register 3 (SYSCR3)

SYSCR3 controls the power-down modes with SYSCR1 and SYSCR2.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
0	STS3	0	R/W	Standby Timer Select 3 Specifies the time the CPU and peripheral modules wait for stable clock operation after exiting from standby mode, subactive mode, or watch mode to active mode or sleep mode. This bit should be specified together with the STS2 to STS0 bits in SYSCR1 according to the operating frequency so that the wait time is at least equal to the oscillation stabilization time. The relationship between the specified value and the number of wait states is shown in table 6.1. When an external clock is to be used, the minimum value (STS3 = 0, STS2 = 1, STS1 = 0, STS0 = 1) is recommended. When the on-chip oscillator for the system clock is to be used, TBD (STS3 = TBD, STS2 = TBD, STS1 = TBD, STS0 = TBD) is recommended. If a setting other than the recommended value is made, operation may start before the end of the wait time.

Table 6.1 Operating Frequency and Wait Time

Bit				Number of	Operating Frequency and Wait Time						
STS3	STS2	STS1	STS0	Wait States	10 MHz	8 MHz	6 MHz	5 MHz	4.194 MHz	3 MHz	2 MHz
0	0	0	0	8,192 states	819.2	1,024.0	1,365.3	1,638.4	1,953.3	2,730.7	4,096.0
0	0	0	1	16,384 states	1,638.4	2,048.0	2,730.7	3,276.8	3,906.5	5,461.3	8,192.0
0	0	1	0	1,024 states	102.4	128.0	170.7	204.8	244.2	341.3	512.0
0	0	1	1	2,048 states	204.8	256.0	341.3	409.6	488.3	682.7	1,024.0
0	1	0	0	4,096 states	409.6	512.0	682.7	819.2	976.6	1,365.3	2,048.0
0	1	0	1	2 states (external clock input)	0.2	0.3	0.3	0.4	0.5	0.7	1.0
0	1	1	0	8 states	0.8	1.0	1.3	1.6	1.9	2.7	4.0
0	1	1	1	16 states	1.6	2.0	2.7	3.2	3.8	5.3	8.0
1	0	0	0	256 states	25.6	32.0	42.7	51.2	61.0	85.3	128.0
1	0	0	1	512 states	51.2	64.0	85.3	102.4	122.1	170.7	256.0
1	0	1	0	32,768 states	3,276.8	4,096.0	5,461.3	6,553.6	7,813.1	10,922.7	16,384.0
1	0	1	1	65,536 states	6,553.6	8,192.0	10,922.7	13,107.2	15,626.1	21,845.3	32,768.0
1	1	0	0	131,072 states	13,107.2	16,384.0	21,845.3	26,214.4	31,252.3	43,690.7	65,536.0
1	1	0	1	4 states	0.4	0.5	0.7	0.8	1.0	1.3	2.0
1	1	1	0	32 states	3.2	4.0	5.3	6.4	7.6	10.7	16.0
1	1	1	1	128 states	12.8	16.0	21.3	25.6	30.5	42.7	64.0

Note: Time unit is μs.
When an external clock is input, bits STS3 to STS0 should be set as external clock input mode before mode transition is executed. When an external clock is not used, these bits should not be set as external clock input mode.

6.1.4 Clock Halt Registers 1 to 3 (CKSTPR1 to CKSTPR3)

CKSTPR1, CKSTPR2, and CKSTPR3 allow the on-chip peripheral modules to enter the standby state in module units.

- CKSTPR1

Bit	Bit Name	Initial Value	R/W	Description
7	S4CKSTP*[1]*[3]	1	R/W	SCI4 Module Standby The SCI4 enters standby mode when this bit is cleared to 0.
6	S31CKSTP	1	R/W	SCI3_1 Module Standby*[2] The SCI3_1 enters standby mode when this bit is cleared to 0.
5	S32CKSTP	1	R/W	SCI3_2 Module Standby*[2] The SCI3_2 enters standby mode when this bit is cleared to 0.*[1]
4	ADCKSTP	1	R/W	A/D Converter Module Standby The A/D converter enters standby mode when this bit is cleared to 0.
3	—	1	R/W	Reserved This bit can be read from or written to.
2	TFCKSTP	1	R/W	Timer F Module Standby Timer F enters standby mode when this bit is cleared to 0.
1	FROMCKSTP*[1]*[3]	1	R/W	Flash Memory Module Standby Flash memory enters standby mode when this bit is cleared to 0. When the addresses H'000000 to H'0000FF of the flash memory space is accessed while this bit is set to 0, the RAM emulation function is enabled and the addresses H'FFFC00 to H'FFFCFF of the RAM space can be accessed. For details, see section 7.4, Using RAM to Emulate Flash Memory. The RAM emulation function is supported only by the F-ZTAT version.
0	RTCCKSTP	1	R/W	RTC Module Standby RTC enters standby mode when this bit is cleared to 0.

- CKSTPR2

Bit	Bit Name	Initial Value	R/W	Description
7	ADBCKSTP	1	R/W	Address Break Module Standby The address break enters standby mode when this bit is cleared to 0.
6	TPUCKSTP	1	R/W	TPU Module Standby The TPU enters standby mode when this bit is cleared to 0.
5	IICCKSTP	1	R/W	IIC2 Module Standby The IIC2 enters standby mode when this bit is cleared to 0.
4	PW2CKSTP	1	R/W	PWM2 Module Standby The PWM2 enters standby mode when this bit is cleared to 0.
3	AECCKSTP	1	R/W	Asynchronous Event Counter Module Standby The asynchronous event counter enters standby mode when this bit is cleared to 0.
2	WDCKSTP	1	R/W*[4]	Watchdog Timer Module Standby The watchdog timer enters standby mode when this bit is cleared to 0.
1	PW1CKSTP	1	R/W	PWM1 Module Standby The PWM1 enters standby mode when this bit is cleared to 0.
0	—	1	R/W	Reserved This bit can be read from or written to.

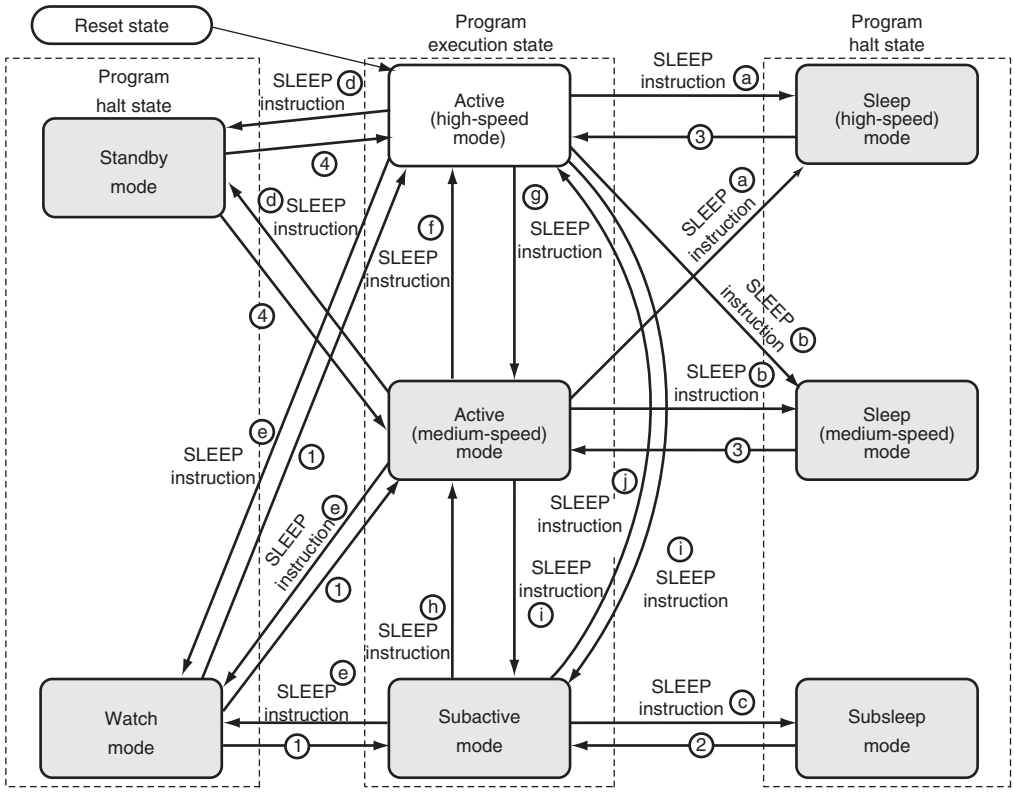
- CKSTPR3

Bit	Bit Name	Initial Value	R/W	Description
7	S33CKSTP	1	R/W	SCI3_3 Module Standby*[2] The SCI3_3 enters standby mode when this bit is cleared to 0.
6	TCCKSTP	1	R/W	Timer C Module Standby The timer C enters standby mode when this bit is cleared to 0.
5	TGCKSTP	1	R/W	Timer G Module Standby The timer G enters standby mode when this bit is cleared to 0.
4	PW4CKSTP	1	R/W	PWM4 Module Standby The PWM4 enters standby mode when this bit is cleared to 0.
3	PW3CKSTP	1	R/W	PWM3 Module Standby The PWM3 enters standby mode when this bit is cleared to 0.
2 to 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

Notes: 1. This bit is always read as 1 and cannot be modified in the masked ROM version.
2. When the SCI3 module standby is set, all registers in the SCI3 enter the reset state.
3. This bit must be set to 1 when the on-chip emulator is used.
4. This bit is valid when the WDON bit in TCSRW is 0. If this bit is cleared to 0 while the WDON bit is set to 1 (while the watchdog timer is operating), this bit is cleared to 0. However, the watchdog timer does not enter module standby mode and continues operating. When the watchdog timer stops operating and the WDON bit is cleared to 0 by software, this bit is valid and the watchdog timer enters module standby mode.

6.2 Mode Transitions and States of LSI

Figure 6.1 shows the possible transitions among these operating modes. A transition is made from the program execution state to the program halt state of the program by executing a SLEEP instruction. The program execution state is recovered from the program halt state by an interrupt. A direct transition between active mode and subactive mode, which are both program execution states, can be made without halting the program. The operating frequencies can be changed in the same mode by a direct transition from active mode to active mode, or from subactive mode to subactive mode. $\overline{\text{RES}}$ input enables transitions from a mode to the reset state. Table 6.2 shows the transition conditions of each mode after the SLEEP instruction is executed and a mode to return by an interrupt. Table 6.3 shows the internal states of the LSI in each mode.



Mode Transition Conditions (1)

	LSON	MSON	SSBY	TMA3	DTON
ⓐ	0	0	0	*	0
ⓑ	0	1	0	*	0
ⓒ	1	*	0	1	0
ⓓ	0	*	1	0	0
ⓔ	*	*	1	1	0
ⓕ	0	0	0	*	1
ⓖ	0	1	0	*	1
ⓗ	0	1	1	1	1
ⓘ	1	*	1	1	1
ⓙ	0	0	1	1	1

* Don't care

Mode Transition Conditions (2)

	Interrupt Sources
①	NMI, IRQ0, IRQ1, IRQ3, IRQ4, IRQAEC, WKP0 to WKP7, RTC, WDT, AEC, Timer C, Timer F, Timer G
②	All interrupts except for TPU, IIC2, A/D, address break interrupts
③	All interrupts
④	NMI, IRQ0, IRQ1, IRQ3, IRQ4, IRQAEC, WKP0 to WKP7, WDT, AEC

Note: A transition between different modes cannot be made to occur simply because an interrupt request is generated. Make sure that interrupt handling is performed after the interrupt is accepted.

Figure 6.1 Mode Transition Diagram

Table 6.2 Transition Mode after SLEEP Instruction Execution and Interrupt Handling

State Before Transition	LSON	MSON	SSBY	TMA3	DTON	Transition Mode after SLEEP Instruction Execution	Transition Mode due to Interrupt	Symbol in Figure 6.1
Active (high-speed) mode	0	0	0	x	0	Sleep (high-speed) mode	Active (high-speed) mode	a
	0	1	0	x	0	Sleep (medium-speed) mode	Active (medium-speed) mode	b
	0	0	1	0	0	Standby mode	Active (high-speed) mode	d
	0	1	1	0	0	Standby mode	Active (medium-speed) mode	d
	0	0	1	1	0	Watch mode	Active (high-speed) mode	e
	0	1	1	1	0	Watch mode	Active (medium-speed) mode	e
	1	x	1	1	0	Watch mode	Subactive mode	e
	0	0	0	x	1	Active (high-speed) mode (direct transition)	—	—
	0	1	0	x	1	Active (medium-speed) mode (direct transition)	—	g
	1	x	1	1	1	Subactive mode (direct transition)	—	i

State Before Transition	LSON	MSON	SSBY	TMA3	DTON	Transition Mode after SLEEP Instruction Execution	Transition Mode due to Interrupt	Symbol in Figure 6.1
Active (medium-speed) mode	0	0	0	x	0	Sleep (high-speed) mode	Active (high-speed) mode	a
	0	1	0	x	0	Sleep (medium-speed) mode	Active (medium-speed) mode	b
	0	0	1	0	0	Standby mode	Active (high-speed) mode	d
	0	1	1	0	0	Standby mode	Active (medium-speed) mode	d
	0	0	1	1	0	Watch mode	Active (high-speed) mode	e
	0	1	1	1	0	Watch mode	Active (medium-speed) mode	e
	1	1	1	1	0	Watch mode	Subactive mode	e
	0	0	0	x	1	Active (high-speed) mode (direct transition)	—	f
	0	1	0	x	1	Active (medium-speed) mode (direct transition)	—	—
	1	x	1	1	1	Subactive mode (direct transition)	—	i

State Before Transition	LSON	MSON	SSBY	TMA3	DTON	Transition Mode after SLEEP Instruction Execution	Transition Mode due to Interrupt	Symbol in Figure 6.1
Subactive mode	1	x	0	1	0	Subsleep mode	Subactive mode	c
	0	0	1	1	0	Watch mode	Active (high-speed) mode	e
	0	1	1	1	0	Watch mode	Active (medium-speed) mode	e
	1	x	1	1	0	Watch mode	Subactive mode	e
	0	0	1	1	1	Active (high-speed) mode (direct transition)	—	j
	0	1	1	1	1	Active (medium-speed) mode (direct transition)	—	h
	1	x	1	1	1	Subactive mode (direct transition)	—	—

[Legend]

x: Don't care

Table 6.3 Internal State in Each Operating Mode

Function		Active Mode: High-speed	Active Mode: Medium-speed	Sleep Mode: High-speed	Sleep Mode: Medium-speed	Watch Mode	Subactive Mode	Subsleep Mode	Stand-by Mode
System clock oscillator		Functioning	Functioning	Functioning	Functioning	Halted	Halted	Halted	Halted
Subclock oscillator		Functioning/ halted	Functioning/ halted	Functioning/ halted	Functioning/ halted	Functioning	Functioning	Functioning	Functioning/ halted
CPU	Instructions	Functioning	Functioning	Halted	Halted	Halted	Functioning	Halted	Halted
	RAM			Retained	Retained	Retained		Retained	Retained
	Registers								
	I/O								Retained*1
External interrupts	NMI	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning
	IRQ0								
	IRQ1								
	IRQ3								
	IRQ4								
	IRQAEC								
	WKP0 to WKP7								
Peripheral modules	Timer C	Functioning	Functioning	Functioning	Functioning	Functioning/ retained*2	Functioning/ retained*2	Functioning/ retained*2	Retained
	Timers F, G					Functioning/ retained*3	Functioning/ retained*3	Functioning/ retained*3	Retained
	Asynchronous event counter					Functioning	Functioning	Functioning	Functioning
	RTC					Functioning/ retained*7	Functioning/ retained*7	Functioning/ retained*7	Retained
	TPU					Retained	Retained	Retained	Retained
	WDT					Functioning/ retained*4	Functioning/ retained*4	Functioning/ retained*4	Functioning/ retained*5
	SCI3/IrDA					Reset	Functioning/ retained*6	Functioning/ retained*6	Reset
	IIC2					Retained	Retained	Retained	Retained
	PWM								
	A/D converter								
	Address break			Retained	Retained		Functioning		

Notes:
1. Register contents are retained. Output is the high-impedance state.
2. Functions if $\phi_w/4$, $\phi_w/256$, or $\phi_w/1024$ is selected as an internal clock. Halted and retained otherwise.
3. Functions if $\phi_w/4$ is selected as a clock to be used. Halted and retained otherwise.
4. Functions if the on-chip WDT oscillator is selected or if $\phi_w/16$, or $\phi_w/256$ is selected as an internal clock. Halted and retained otherwise.
5. Functions if the on-chip WDT oscillator is selected. Halted and retained otherwise.
6. Functions if $\phi_w/2$ is selected as an internal clock. Halted and retained otherwise.
7. Functions if the RTC operation is selected as a clock source. Halted and retained for the free-running counter.

6.2.1 Sleep Mode

In sleep mode, CPU operation is halted but the system clock oscillator, on-chip oscillator for the system clock, subclock oscillator, and on-chip peripheral modules continues operating. In sleep (medium-speed) mode, the on-chip peripheral modules function at the clock frequency set by the MA1 and MA0 bits in SYSCR1. CPU register contents are retained.

Sleep mode is cleared by an interrupt. When an interrupt is requested, sleep mode is cleared and interrupt exception handling starts. Sleep mode is not cleared if the I bit in CCR is set to 1 or the requested interrupt is disabled by the interrupt enable bit. After sleep mode is cleared, a transition is made from sleep (high-speed) mode to active (high-speed) mode or from sleep (medium-speed) mode to active (medium-speed) mode.

When the $\overline{\text{RES}}$ pin goes low, the CPU goes into the reset state and sleep mode is cleared. Since an interrupt request signal is synchronous with the system clock, the maximum time of $2/\phi$ (s) may be delayed from the point at which an interrupt request signal occurs until the interrupt exception handling is started.

6.2.2 Standby Mode

In standby mode, the system clock oscillator and on-chip oscillator for the system clock is halted, so the CPU and on-chip peripheral modules except WDT and asynchronous event counter stop functioning. However, as long as the rated voltage is supplied, the contents of CPU registers and some on-chip peripheral module registers are retained. On-chip RAM contents will be retained as long as the voltage set by the RAM data retention voltage is provided. The I/O ports go to the high-impedance state.

Standby mode is cleared by an interrupt. When an interrupt is requested, the system clock oscillator and on-chip oscillator for the system clock start. After the time set in the STS2 to STS0 bits in SYSCR1 and the STS3 bit in SYSCR3 has elapsed, standby mode is cleared and interrupt exception handling starts. After standby mode is cleared, a transition is made to active (high-speed) or active (medium-speed) mode according to the MSON bit in SYSCR2. Standby mode is not cleared if the I bit in CCR is set to 1 or the requested interrupt is disabled by the interrupt enable bit.

When a reset source is generated in standby mode, the system clock oscillator and the on-chip oscillator for the system clock start. The $\overline{\text{RES}}$ pin must be kept low until the system clock oscillator output stabilizes and the t_{REL} period has elapsed. The CPU starts reset exception handling when the $\overline{\text{RES}}$ pin is driven high.

6.2.3 Watch Mode

In watch mode, the system clock oscillator, on-chip oscillator for the system clock, and CPU operation stop and on-chip peripheral modules stop functioning except for the WDT, RTC, timer C, timer F, timer G, and asynchronous event counter. However, as long as the rated voltage is supplied, the contents of CPU registers, some on-chip peripheral module registers, and on-chip RAM are retained. The I/O ports retain their state before the transition.

Watch mode is cleared by an interrupt. When an interrupt is requested, watch mode is cleared and interrupt exception handling starts. When watch mode is cleared by an interrupt, a transition is made to active (high-speed) mode, active (medium-speed) mode, or subactive mode depending on the settings of the LSON bit in SYSCR1 and the MSON bit in SYSCR2. When the transition is made to active mode, after the time set in the STS2 to STS0 bits in SYSCR1 and the STS3 bit in SYSCR3 has elapsed, interrupt exception handling starts. Watch mode is not cleared if the I bit in CCR is set to 1 or the requested interrupt is disabled by the interrupt enable register.

When a reset source is generated in watch mode, the system clock oscillator starts. The $\overline{\text{RES}}$ pin must be kept low until the system clock oscillator output stabilizes and the t_{REL} period has elapsed. The CPU starts reset exception handling when the $\overline{\text{RES}}$ pin is driven high.

6.2.4 Subsleep Mode

In subsleep mode, the CPU operation stops but on-chip peripheral modules other than the TPU, A/D converter, PWM, IIC2, and address break continue running. As long as a required voltage is applied, the contents of CPU registers, the on-chip RAM, and some registers of the on-chip peripheral modules are retained. I/O ports keep the same states as before the transition.

Subsleep mode is cleared by an interrupt. When an interrupt is requested, subsleep mode is cleared and interrupt exception handling starts. After subsleep mode is cleared, a transition is made to subactive mode. Subsleep mode is not cleared if the I bit in CCR is set to 1 or the requested interrupt is disabled by the interrupt enable register.

When a reset source is generated in subsleep mode, the system clock oscillator starts. The $\overline{\text{RES}}$ pin must be kept low until the system clock oscillator output stabilizes and the t_{REL} period has elapsed. The CPU starts reset exception handling when the $\overline{\text{RES}}$ pin is driven high.

6.2.5 Subactive Mode

In subactive mode, the system clock oscillator and the on-chip oscillator for the system clock stop functioning but on-chip peripheral modules other than the A/D converter, PWM, TPU, and IIC2 continue to operate. As long as a required voltage is applied, the contents of some registers of the on-chip peripheral modules are retained.

Subactive mode is cleared by the SLEEP instruction. When subacitve mode is cleared, a transition to subsleep mode, active mode, or watch mode is made, depending on the combination of bits SSBY, LSON, and TMA3 in SYSCR1 and bits MSON and DTON in SYSCR2.

When a reset source is generated in subactive mode, the system clock oscillator starts. The $\overline{\text{RES}}$ pin must be kept low until the system clock oscillator output stabilizes and the t_{REL} period has elapsed. The CPU starts reset exception handling when the $\overline{\text{RES}}$ pin is driven high.

The operating frequency of subactive mode is selected from $\phi_w/2$, $\phi_w/4$, and $\phi_w/8$ by the SA1 and SA0 bits in SYSCR2. After the SLEEP instruction is executed, the operating frequency changes to the frequency which is set before the execution.

6.2.6 Active (Medium-Speed) Mode

In active (medium-speed) mode, the clock set by the MA1 and MA0 bits in SYSCR1 is used as the system clock, and the CPU and on-chip peripheral modules function.

Active (medium-speed) mode is cleared by the SLEEP instruction. When active (medium-speed) mode is cleared, a transition to standby mode is made depending on the combination of bits SSBY, LSON, and TMA3 in SYSCR1, a transition to watch mode is made depending on the combination of bits SSBY and TMA3 in SYSCR1, or a transition to sleep mode is made depending on the combination of bits SSBY and LSON in SYSCR1. Moreover, a transition to active (high-speed) mode or subactive mode is made by a direct transition. When a reset source is generated in active (medium-speed) mode, the CPU goes into the reset state and active (medium-sleep) mode is cleared.

6.3 Direct Transition

The CPU can execute programs in two modes: active and subactive modes. A direct transition is made between these two modes without stopping program execution. A direct transition can also be made when the operating clock is changed in active and subactive modes. The transition is made via the sleep or watch mode, by setting the DTON bit in SYSCR2 to 1 to execute a SLEEP instruction. After the mode transition, direct transition interrupt exception handling starts.

Note that if a direct transition is attempted while the I bit in CCR is 1, the transition is made to the sleep or watch mode, though not returning from the mode.

6.3.1 Direct Transition from Active (High-Speed) Mode to Active (Medium-Speed) Mode

When a SLEEP instruction is executed in active (high-speed) mode while the SSBY and LSON bits in SYSCR1 are cleared to 0 and the MSON and DTON bits in SYSCR2 are set to 1, a transition is made to active (medium-speed) mode via sleep mode.

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

Direct transition time = {(Number of SLEEP instruction execution states) + (Number of internal processing states)} × (tcyc before transition) + (Number of interrupt exception handling execution states) × (tcyc after transition) … ……(1)

Example: When ϕosc/8 is selected as the CPU operating clock after the transition

Direct transition time = (2 + 1) × 1tosc + 14 × 8tosc = 115tosc

For the legend of symbols used above, refer to section 25, Electrical Characteristics.

6.3.2 Direct Transition from Active (High-Speed) Mode to Subactive Mode

When a SLEEP instruction is executed in active (high-speed) mode while the SSBY, TMA3, and LSON bits in SYSCR1 are set to 1 and the DTON bit in SYSCR2 is set to 1, a transition is made to subactive mode via watch mode.

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

Direct transition time = {(Number of SLEEP instruction execution states) + (Number of internal processing states)} × (tcyc before transition) + (Number of interrupt exception handling execution states) × (tsubcyc after transition) ……(2)

Example: When ϕw/8 is selected as the subactive operating clock after the transition

Direct transition time = (2 + 1) × 1tosc + 14 × 8tw = 3tosc + 112tw

For the legend of symbols used above, refer to section 25, Electrical Characteristics.

6.3.3 Direct Transition from Active (Medium-Speed) Mode to Active (High-Speed) Mode

When a SLEEP instruction is executed in active (medium-speed) mode while the SSBY and LSON bits in SYSCR1 are cleared to 0, the MSON bit in SYSCR2 is cleared to 0, and the DTON bit in SYSCR2 is set to 1, a transition is made to active (high-speed) mode via sleep mode.

The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (3).

Direct transition time = {(Number of SLEEP instruction execution states) + (Number of internal processing states)} × (tcyc before transition) + (Number of interrupt exception handling execution states) × (tcyc after transition) ……..(3)

Example: When ϕosc/8 is selected as the CPU operating clock before the transition

Direct transition time = (2 + 1) × 8tosc + 14 × 1tosc = 38tosc

For the legend of symbols used above, refer to section 25, Electrical Characteristics.

6.3.4 Direct Transition from Active (Medium-Speed) Mode to Subactive Mode

When a SLEEP instruction is executed in active (medium-speed) mode while the SSBY, LSON, and TMA3 bits in SYSCR1 are set to 1 and the DTON bit in SYSCR2 is set to 1, a transition is made to subactive mode via watch mode.

The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (4).

Direct transition time = {(Number of SLEEP instruction execution states) + (Number of internal processing states)} × (tcyc before transition) + (Number of interrupt exception handling execution states) × (tsubcyc after transition) ……(4)

Example: When ϕosc/8 and ϕw/8 are selected as the CPU operating clock before and after the transition, respectively

Direct transition time = (2 + 1) × 8tosc + 14 × 8tw = 24tosc + 112tw

For the legend of symbols used above, refer to section 25, Electrical Characteristics.

6.3.5 Direct Transition from Subactive Mode to Active (High-Speed) Mode

When a SLEEP instruction is executed in subactive mode while the SSBY and TMA3 bits in SYSCR1 are set to 1, the LSON bit in SYSCR1 is cleared to 0, the MSON bit in SYSCR2 is cleared to 0, and the DTON bit in SYSCR2 is set to 1, a transition is made directly to active (high-speed) mode via watch mode after the waiting time set in bits STS2 to STS0 in SYSCR1 has elapsed.

The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (5).

Direct transition time = {(Number of SLEEP instruction execution states) + (Number of internal processing states)} × (tsubcyc before transition) + (Wait time set in bits STS2 to STS0) + (Number of interrupt exception handling execution states) × (tcyc after transition) ……..(5)

Example: When ϕw/8 is selected as the CPU operating clock after the transition and wait time = 8192 states

Direct transition time = (2 + 1) × 8tw + (8192 + 14) × 1tosc = 24tw + 8206tosc

For the legend of symbols used above, refer to section 25, Electrical Characteristics.

6.3.6 Direct Transition from Subactive Mode to Active (Medium-Speed) Mode

When a SLEEP instruction is executed in subactive mode while the SSBY and TMA3 bits in SYSCR1 are set to 1, the LSON bit in SYSCR1 is cleared to 0, and the MSON and DTON bits in SYSCR2 are set to 1, a transition is made directly to active (medium-speed) mode via watch mode after the waiting time set in bits STS2 to STS0 in SYSCR1 has elapsed.

The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (6).

Direct transition time = {(Number of SLEEP instruction execution states) + (Number of internal processing states)} × (tsubcyc before transition) + (Wait time set in bits STS2 to STS0) + (Number of interrupt exception handling execution states) × (tcyc after transition) ……..(6)

Example: When ϕw/8 and ϕosc/8 are selected as the CPU operating clock before and after the transition, respectively, and wait time = 8192 states

Direct transition time = (2 + 1) × 8tw + 8192 × 1tosc + 14 × 8tosc = 24tw + 8304tosc

For the legend of symbols used above, refer to section 25, Electrical Characteristics.

6.3.7 Notes on External Input Signal Changes before/after Direct Transition

(1) Direct transition from active (high-speed) mode to subactive mode

Since the mode transition is performed via watch mode, see section 6.5.2, Notes on External Input Signal Changes before/after Standby Mode.

(2) Direct transition from active (medium-speed) mode to subactive mode

Since the mode transition is performed via watch mode, see section 6.5.2, Notes on External Input Signal Changes before/after Standby Mode.

(3) Direct transition from subactive mode to active (high-speed) mode

Since the mode transition is performed via watch mode, see section 6.5.2, Notes on External Input Signal Changes before/after Standby Mode.

(4) Direct transition from subactive mode to active (medium-speed) mode

Since the mode transition is performed via watch mode, see section 6.5.2, Notes on External Input Signal Changes before/after Standby Mode.

6.4 Module Standby Function

The module-standby function can be set to any peripheral module. In module standby mode, the clock supply to modules stops to enter the power-down mode. Module standby mode enables each on-chip peripheral module to enter the standby state by clearing a bit that corresponds to each module in CKSTPR1 to CKSTPR3 and cancels the mode by setting the bit to 1 (see section 6.1.4, Clock Halt Registers 1 to 3 (CKSTPR1 to CKSTPR3)).

6.5 Usage Notes

6.5.1 Standby Mode Transition and Pin States

When a SLEEP instruction is executed in active (high-speed) mode or active (medium-speed) mode while the SSBY and TMA3 bits in SYSCR1 are set to 1 and the LSON bit in SYSCR1 is cleared to 0, a transition is made to standby mode. At the same time, pins go to the high-impedance state (except pins for which the pull-up MOS is designated as on). Figure 6.2 shows the timing in this case.

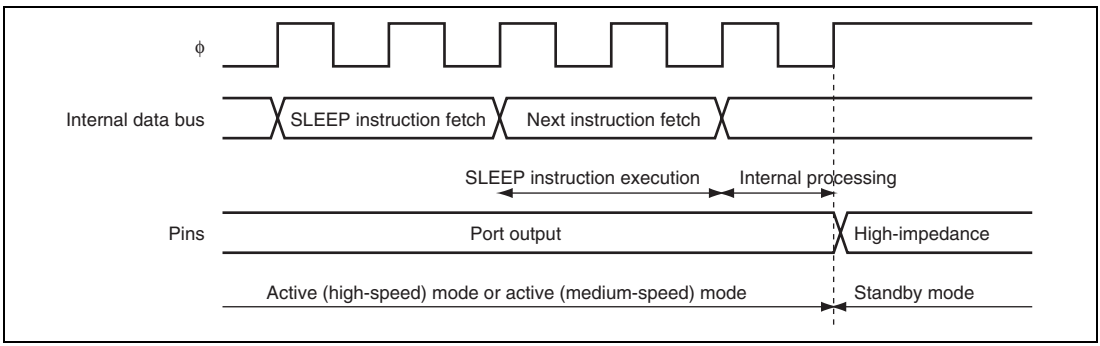


Figure 6.2 Standby Mode Transition and Pin States

6.5.2 Notes on External Input Signal Changes before/after Standby Mode

(1) When External Input Signal Changes before/after Standby Mode or Watch Mode

When an external input signal such as $\overline{\text{NMI}}$, $\overline{\text{IRQ}}$, $\overline{\text{WKP}}$, or IRQAEC is input, both the high- and low-level widths of the signal must be at least two cycles of system clock ϕ or subclock ϕ_{SUB} (referred to together in this section as the internal clock). As the internal clock stops in standby mode and watch mode, the width of external input signals requires careful attention when a transition is made via these operating modes. Ensure that external input signals conform to the conditions stated in (3), Recommended Timing of External Input Signals, below.

(2) When External Input Signals cannot be Captured because Internal Clock Stops

The case of falling edge capture is shown in figure 6.3.

As shown in the case marked "Capture not possible," when an external input signal falls immediately after a transition to active mode or subactive mode, after oscillation is started by an interrupt via a different signal, the external input signal cannot be captured if the high-level width at that point is less than 2 t_{cyc} or 2 t_{subcyc}.

(3) Recommended Timing of External Input Signals

To ensure dependable capture of an external input signal, high- and low-level signal widths of at least 2 t_{cyc} or 2 t_{subcyc} are necessary before a transition is made to standby mode or watch mode, as shown in "Capture possible: case 1."

External input signal capture is also possible with the timing shown in "Capture possible: case 2" and "Capture possible: case 3," in which a 2 t_{cyc} or 2 t_{subcyc} level width is secured.

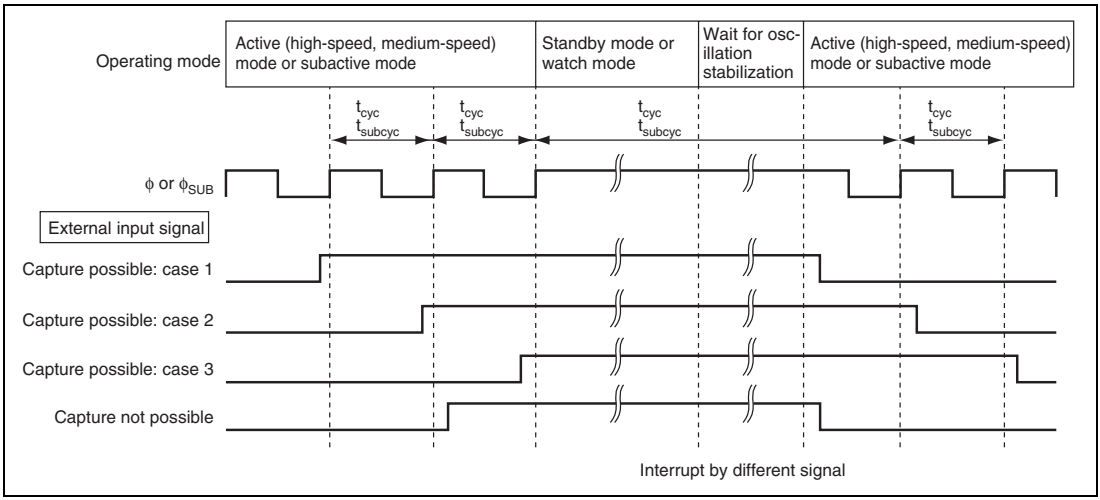


Figure 6.3 External Input Signal Capture when Signal Changes before/after Standby Mode or Watch Mode

(4) Input Pins to which these Notes Apply

$\overline{\text{NMI}}$, $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, $\overline{\text{IRQ3}}$, $\overline{\text{IRQ4}}$, IRQAEC, $\overline{\text{WKP0}}$ to $\overline{\text{WKP7}}$, TMIC, TMIF, TMIG, $\overline{\text{ADTRG}}$, TIOCA1, TIOCB1, TIOCA2, and TIOCB2.

Section 7 ROM

The features of the 128-Kbyte flash memory built into the flash memory (F-ZTAT) version are summarized below.

- Programming/erase methods

  The flash memory is programmed 128 bytes at a time. Erase is performed in single-block units. The flash memory is configured as follows: 1 Kbyte × 4 blocks, 28 Kbytes × 1 block, 16 Kbytes × 1 block, 8 Kbytes × 2 blocks, and 32 Kbytes × 2 blocks. To erase the entire flash memory, each block must be erased in turn.
- On-board programming

  On-board programming/erasing can be done in boot mode, in which the boot program built into the chip is started to erase or program of the entire flash memory. In normal user 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 match 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 a result, flash memory can be read with low power consumption.
- Module standby mode

  Use of module standby mode enables this module to be placed in standby mode independently when not used (for details, see section 6.4, Module Standby Function). When the on-chip debugger is in use, the bit 1 (FROMCKSTP) in the clock stop register 1 (CKSTPR1) must be set to 1.

7.1 Block Configuration

Figure 7.1 shows the block configuration of flash memory. The thick lines indicate erasing units, the narrow lines indicate programming units, and the values are addresses. The 128-Kbyte flash memory is divided into 1 Kbyte × 4 blocks, 28 Kbytes × 1 block, 16 Kbytes × 1 block, 8 Kbytes × 2 blocks, and 32 Kbytes × 2 blocks . Erasing is performed in these units. Programming is performed in 128-byte units starting from an address with lower eight bits H'00 or H'80.

EB0 Erase unit 1 Kbyte	H'000000	H'000001	H'000002	← Programming unit: 128 bytes →	H'00007F
	H'000380	H'000381	H'000382	– – – – – – – – – –	H'0003FF
EB1 Erase unit 1 Kbyte	H'000400	H'000401	H'000402	← Programming unit: 128 bytes →	H'00047F
	H'000780	H'000781	H'000782	– – – – – – – – – –	H'0007FF
EB2 Erase unit 1 Kbyte	H'000800	H'000801	H'000802	← Programming unit: 128 bytes →	H'00087F
	H'000B80	H'000B81	H'000B82	– – – – – – – – – –	H'000BFF
EB3 Erase unit 1 Kbyte	H'000C00	H'000C01	H'000C02	← Programming unit: 128 bytes →	H'000C7F
	H'000F80	H'000F81	H'000F82	– – – – – – – – – –	H'000FFF
EB4 Erase unit 28 Kbytes	H'001000	H'001001	H'001002	← Programming unit: 128 bytes →	H'00107F
	H'007F80	H'007F81	H'007F82	– – – – – – – – – –	H'007FFF
EB5 Erase unit 16 Kbytes	H'008000	H'008001	H'008002	← Programming unit: 128 bytes →	H'00807F
	H'00BF80	H'00BF81	H'00BF82	– – – – – – – – – –	H'00BFFF
EB6 Erase unit 8 Kbytes	H'00C000	H'00C001	H'00C002	← Programming unit: 128 bytes →	H'00C07F
	H'00DF80	H'00DF81	H'00DF82	– – – – – – – – – –	H'00DFFF
EB7 Erase unit 8 Kbytes	H'00E000	H'00E001	H'00E002	← Programming unit: 128 bytes →	H'00E07F
	H'00FF80	H'00FF81	H'00FF82	– – – – – – – – – –	H'00FFFF
EB8 Erase unit 32 Kbytes	H'010000	H'010001	H'010002	← Programming unit: 128 bytes →	H'01007F
	H'017F80	H'017F81	H'017F82	– – – – – – – – – –	H'017FFF
EB9 Erase unit 32 Kbytes	H'018000	H'018001	H'018002	← Programming unit: 128 bytes →	H'01807F
	H'01FF80	H'01FF81	H'01FF82	– – – – – – – – – –	H'01FFFF

Figure 7.1 Flash Memory Block Configuration

7.2 Register Descriptions

The flash memory has the following registers.

- Flash memory control register 1 (FLMCR1)
- Flash memory control register 2 (FLMCR2)
- Erase block register 1 (EBR1)
- Erase block register 2 (EBR2)
- Flash memory power control register (FLPWCR)
- Flash memory enable register (FENR)

7.2.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is a register that makes the flash memory change to program mode, program-verify mode, erase mode, or erase-verify mode. For details on register setting, refer to section 7.5, Flash 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 is cleared to 0, other FLMCR1 register bits and all EBR1 bits cannot be set.
5	ESU	0	R/W	Erase Setup When this bit is set to 1, the flash memory changes to the erase setup state. When it is cleared to 0, the erase setup state is cancelled. Set this bit to 1 before setting the E bit to 1 in FLMCR1.
4	PSU	0	R/W	Program Setup When this bit is set to 1, the flash memory changes to the program setup state. When it is cleared to 0, the program setup state is cancelled. Set this bit to 1 before setting the P bit in FLMCR1.

Bit	Bit Name	Initial Value	R/W	Description
3	EV	0	R/W	Erase-Verify When this bit is set to 1, the flash memory changes to erase-verify mode. When it is cleared to 0, erase-verify mode is cancelled.
2	PV	0	R/W	Program-Verify When this bit is set to 1, the flash memory changes to program-verify mode. When it is cleared to 0, program-verify mode is cancelled.
1	E	0	R/W	Erase When this bit is set to 1 while SWE=1 and ESU=1, the flash memory changes to erase mode. When it is cleared to 0, erase mode is cancelled.
0	P	0	R/W	Program When this bit is set to 1 while SWE=1 and PSU=1, the flash memory changes to program mode. When it is cleared to 0, program mode is cancelled.

7.2.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is a register that displays the state of flash memory programming/erasing. FLMCR2 is a 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 operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state. See section 7.6.3, Error Protection, for details.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

7.2.3 Erase Block Register 1 (EBR1)

EBR1 is a register that is used to specify the flash memory erase area block. EBR1 is initialized to H'00 when the SWE1 bit in FLMCR1 is 0. Do not set more than one bit in EBR1 and EBR2 to 1 at a time, or this will cause all the bits in EBR1 and EBR2 to be automatically cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7	EB7	0	R/W	When this bit is set to 1, 8 Kbytes of EB7 (H'00E000 to H'00FFFF) will be erased.
6	EB6	0	R/W	When this bit is set to 1, 8 Kbytes of EB6 (H'00C000 to H'00DFFF) will be erased.
5	EB5	0	R/W	When this bit is set to 1, 16 Kbytes of EB5 (H'008000 to H'00BFFF) will be erased.
4	EB4	0	R/W	When this bit is set to 1, 28 Kbytes of EB4 (H'001000 to H'007FFF) will be erased.
3	EB3	0	R/W	When this bit is set to 1, 1 Kbyte of EB3 (H'000C00 to H'000FFF) will be erased.
2	EB2	0	R/W	When this bit is set to 1, 1 Kbyte of EB2 (H'000800 to H'000BFF) will be erased.
1	EB1	0	R/W	When this bit is set to 1, 1 Kbyte of EB1 (H'000400 to H'0007FF) will be erased.
0	EB0	0	R/W	When this bit is set to 1, 1 Kbyte of EB0 (H'000000 to H'0003FF) will be erased.

7.2.4 Erase Block Register 2 (EBR2)

EBR2 is a register that is used to specify the flash memory erase area block. EBR2 is initialized to H'00 when the SWE1 bit in FLMCR1 is 0. Do not set more than one bit in EBR1 and EBR2 to 1 at a time, or this will cause all the bits in EBR1 and EBR2 to be automatically cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R/(W)	Reserved The initial value should not be changed.
1	EB9	0	R/W	When this bit is set to 1, 32 Kbytes of EB9 (H'018000 to H'01FFFF) will be erased.
0	EB8	0	R/W	When this bit is set to 1, 32 Kbytes of EB8 (H'010000 to H'017FFF) will be erased.

7.2.5 Flash Memory Power Control Register (FLPWCR)

FLPWCR enables or disables a transition to the flash memory power-down mode when the LSI switches to subactive mode. There are two modes: mode in which operation of the power supply circuit of flash memory is partly halted in power-down mode and flash memory can be read, and mode in which even if a transition is made to subactive mode, operation of the power supply circuit of flash memory is retained and flash memory can be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PDWND	0	R/W	Power-Down Disable When this bit is 0 and a transition is made to subactive mode, the flash memory enters the power-down mode. When this bit is 1, the flash memory remains in the normal mode even after a transition is made to subactive mode.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

7.2.6 Flash Memory Enable Register (FENR)

Bit 7 (FLSHE) in FENR enables or disables the CPU access to the flash memory control registers, FLMCR1, FLMCR2, EBR1, EBR2, 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 when this bit is set to 1. Flash memory control registers cannot be accessed when this bit is set to 0.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

7.3 On-Board Programming Modes

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

When changing to boot mode, the boot program built into this LSI is initiated. The boot program transfers the programming control program from the externally-connected host to on-chip RAM via SCI3 (channel 1). After erasing the entire flash memory, the programming control program is executed. This can be used for programming initial values in the on-board state or for a forcible return when programming/erasing can no longer be done in user program mode. In user program mode, individual blocks can be erased and programmed by branching to the user program/erase control program prepared by the user.

Table 7.1 Setting Programming Modes

TEST	$\overline{\text{NMI}}$	P36	PB0	PB1	PB2	LSI State after Reset End
0	1	x	x	x	x	User mode
0	0	1	x	x	x	Boot mode
1	x	x	0	0	0	Programmer mode

[Legend]
x: Don't care

7.3.1 Boot Mode

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

1. When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. Prepare a programming control program in accordance with the description in section 7.5, Flash Memory Programming/Erasing.
2. SCI3 should be set to asynchronous mode, and the transfer format as follows: 8-bit data, 1 stop bit, and no parity. The inversion function of TXD and RXD pins by SPCR is set to “Not to be inverted,” so do not put the circuit for inverting a value between the host and this LSI.
3. When the boot program is initiated, the chip measures the low-level period of asynchronous 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 match that of the host. The reset should end with the RXD pin high. The RXD and TXD pins should be pulled up on the board if necessary. After the reset is complete, it takes approximately 100 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 the completion of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the chip. If reception could not be performed normally, initiate boot mode again by a reset. Depending on the host's transfer bit rate and system clock frequency of this LSI, there will be a discrepancy between the bit rates of the host and the chip. To operate the SCI properly, set the host's transfer bit rate 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 H'FFF380 to H'FFFE7F is the area to which the programming control program is transferred from the host. The boot program area cannot be used until the execution state in boot mode switches to the programming control program.

6. Before branching to the programming control program, the chip terminates transfer operations by SCI3 (by clearing the RE and TE bits in SCR to 0), however the adjusted bit rate value remains set in BRR. Therefore, the programming control program can still use it for transfer of program data or verify data with the host. The TXD pin is high (PCR42 = 1, P42 = 1). The contents of the CPU general registers are undefined immediately after branching to the programming control program. These registers must be initialized at the beginning of the programming control program, as the stack pointer (SP), in particular, is used implicitly in subroutine calls, etc.
7. Boot mode can be cleared by a reset. End the reset after driving the reset pin low, waiting at least 20 states, and then setting the $\overline{\text{NMI}}$ pin. Boot mode is also cleared when a WDT overflow occurs.
8. Do not change the TEST pin and $\overline{\text{NMI}}$ pin input levels in boot mode.

Table 7.2 Boot Mode Operation

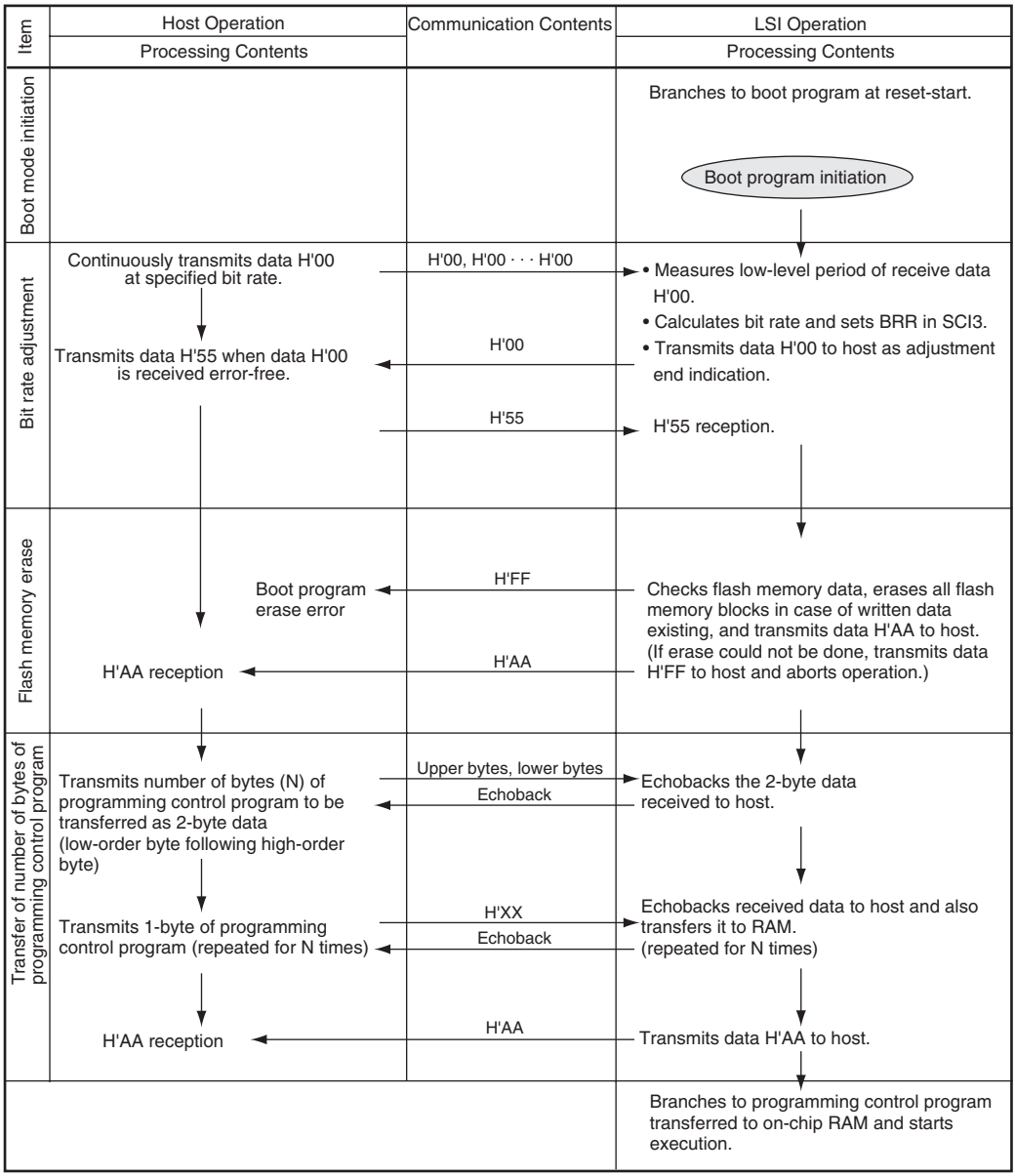


Table 7.3 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible

Host Bit Rate	System Clock Frequency Range of LSI
9,600 bps	8 to 10 MHz
4,800 bps	4 to 10 MHz
2,400 bps	2 to 10 MHz

7.3.2 Programming/Erasing in User Program Mode

On-board programming/erasing of an individual flash memory block can also be performed in user program mode by branching to a user program/erase control program. The user must set branching conditions and provide on-board means of supplying programming data. The flash memory must contain the user program/erase control program or a program that provides the user program/erase 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, as in boot mode. Figure 7.2 shows a sample procedure for programming/erasing in user program mode. Prepare a user program/erase control program in accordance with the description in section 7.5, Flash Memory Programming/Erasing.

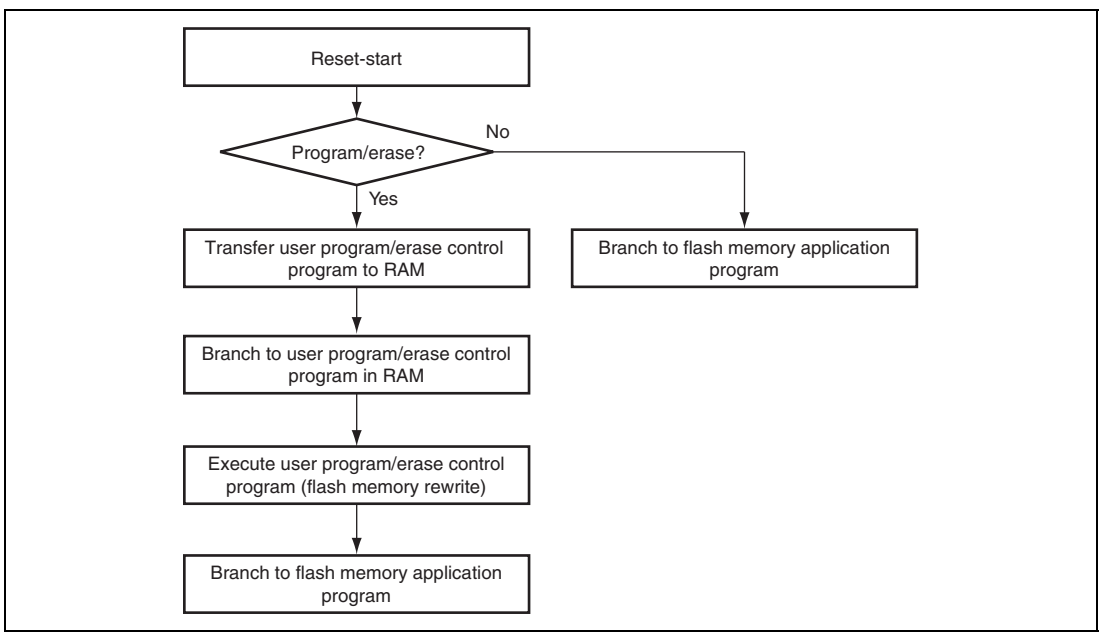


Figure 7.2 Sample Flowchart of Programming/Erasing in User Program Mode

7.4 Using RAM to Emulate Flash Memory

To use on-chip RAM in the realtime emulation of data to be written to the flash memory, the on-chip RAM area can be overlaid on several blocks of flash memory (the emulation area).

Figure 7.3 shows an example where an area of on-chip RAM is overlaid on the emulation area of the flash memory.

1. The area of on-chip RAM area to be overlaid on the emulation area (i.e. the overlay RAM area) is fixed to the 256 bytes from H'FFFC00 to H'FFFCFF.
2. The block of flash memory on which the RAM can be overlaid (i.e. the emulation area) takes up the 256 bytes from H'000000 to H'0000FF.
3. When the FROMCKSTP bit in CKSTPR1 is cleared to 0, the flash memory enters standby mode. The overlay RAM area is overlaid on the emulation area and becomes the target for access when the emulation area of the flash memory is accessed.
4. The overlay RAM area can be accessed at both the addresses within the flash memory area and the original RAM addresses. When using RAM emulation, a vector table is required for the overlaid RAM area.
5. Overlaying of the on-chip RAM is canceled when the FROMCKSTP bit in CKSTPR1 is set, taking the flash memory out of standby mode. This should only be done after execution has made the transition from the emulation area to the RAM area.

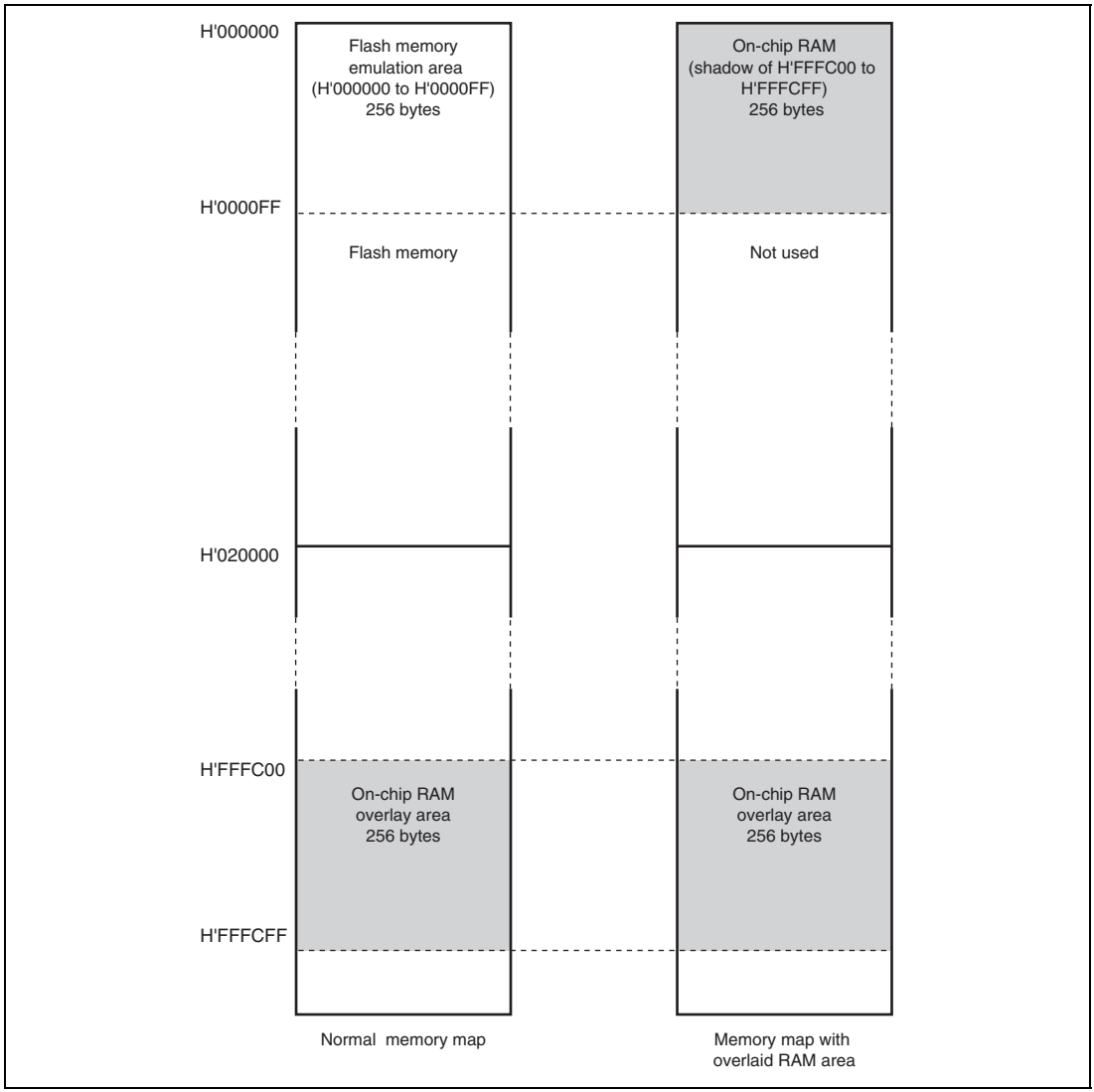


Figure 7.3 Address Map of Overlaid RAM Area

7.5 Flash Memory Programming/Erasing

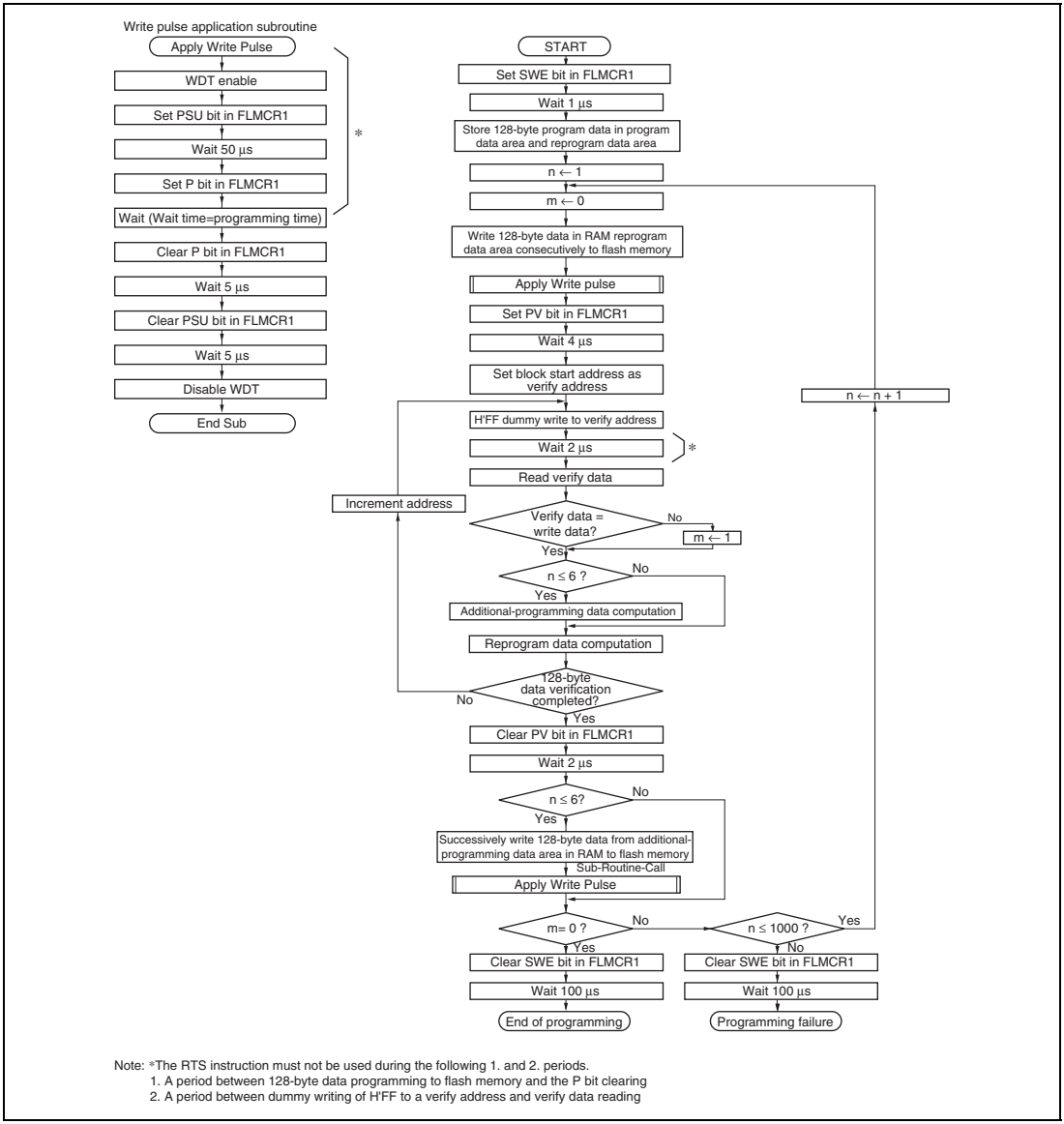
A software method using the CPU is employed to program and erase flash memory in the on-board programming modes. Depending on the FLMCR1 setting, the flash memory operates in one of the following four modes: Program mode, program-verify mode, erase mode, and erase-verify mode. The programming control program in boot mode and the user program/erase control program in user program mode use these operating modes in combination to perform programming/erasing. Flash memory programming and erasing should be performed in accordance with the descriptions in section 7.5.1, Program/Program-Verify and section 7.5.2, Erase/Erase-Verify, respectively.

7.5.1 Program/Program-Verify

When writing data or programs to the flash memory, the program/program-verify flowchart shown in figure 7.4 should be followed. Performing programming operations according to this flowchart will enable data or programs to be written to the flash memory without subjecting the chip to voltage stress or sacrificing program data reliability.

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

8. The maximum number of repetitions of the program/program-verify sequence of the same bit is 1,000.



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 7.4 Program/Program-Verify Flowchart

Table 7.4 Reprogram Data Computation Table

Program Data	Verify Data	Reprogram Data	Comments
0	0	1	Programming completed
0	1	0	Reprogram bit
1	0	1	—
1	1	1	Remains in erased state

Table 7.5 Additional-Program Data Computation Table

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

Table 7.6 Programming Time

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

Note: Time shown in μs.

7.5.2 Erase/Erase-Verify

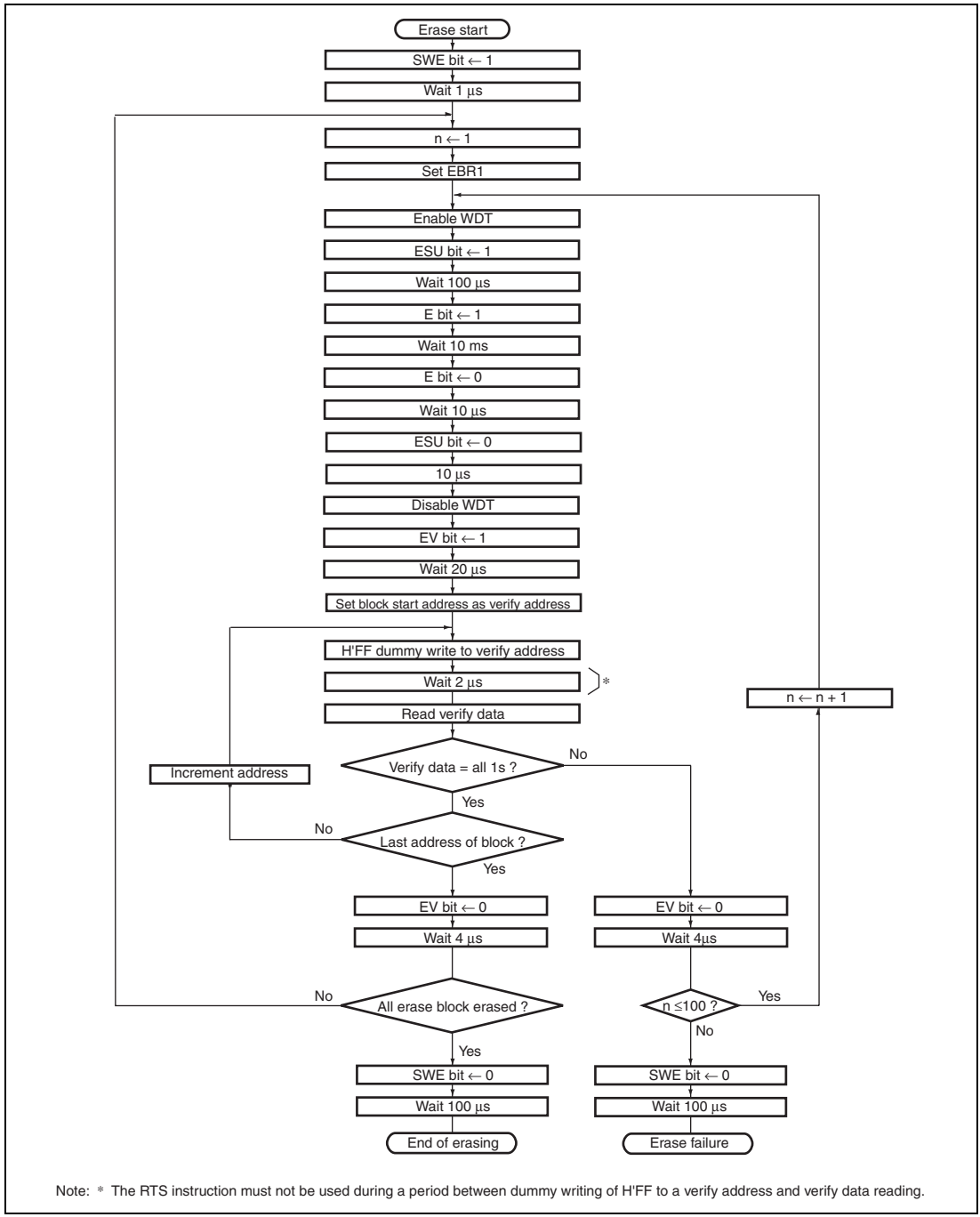
When erasing flash memory, the erase/erase-verify flowchart shown in figure 7.5 should be followed.

1. Prewriting (setting erase block data to all 0s) is not necessary.
2. Erasing is performed in block units. Make only a single-bit specification in the erase block register 1 (EBR1) or the erase block register 2 (EBR2). To erase multiple blocks, each block must be erased in turn.
3. The time during which the E bit is set to 1 is the flash memory erase time.
4. The watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc. An 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 lower two bits are B'00. Verify data can be read in longwords from the address to which a dummy write was performed.
6. If the read data is not erased successfully, set erase mode again, and repeat the erase/erase-verify sequence as before. The maximum number of repetitions of the erase/erase-verify sequence is 100.

7.5.3 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including the NMI interrupt, are disabled while flash memory is being programmed 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 erasing 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 cannot be 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 reading.

Figure 7.5 Erase/Erase-Verify Flowchart

7.6 Program/Erase Protection

There are three kinds of flash memory program/erase protection; hardware protection, software protection, and error protection.

7.6.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted because of a transition to reset, subactive mode, subsleep mode, or standby mode. Flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), erase block register 1 (EBR1), and erase block register 2 (EBR2) are initialized. In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.

7.6.2 Software Protection

Software protection can be implemented against programming/erasing of all flash memory blocks by clearing the SWE bit in FLMCR1. When software protection is in effect, setting the P or E bit in FLMCR1 does not cause a transition to program mode or erase mode. By setting the erase block register 1 (EBR1) or the erase block register 2 (EBR2), erase protection can be set for individual blocks. When EBR1 and EBR2 are set to H'00, erase protection is set for all blocks.

7.6.3 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/erase algorithm, and the program/erase operation is forcibly aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

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

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

The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained, however program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit settings are retained, and a transition can be made to verify mode. Error protection can be cleared only by a reset.

7.7 Programmer Mode

In programmer mode, a PROM programmer can be used to perform programming/erasing via a socket adapter, just as a discrete flash memory. Use a PROM programmer that supports the MCU device type with the on-chip 128-Kbyte flash memory (FZTAT128V3).

7.8 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 memory can 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 flash memory. In subactive mode, the flash memory can be set to operate in power-down mode with the PDWND bit in FLPWCR. When the flash memory returns to its normal operating state from power-down mode or standby mode, a period to stabilize operation of the power supply circuits that were stopped is needed. When the flash memory returns to its normal operating state, bits STS3 to STS0 in SYSCR1 and SYSCR3 must be set to provide a wait time of at least 20 µs, even when the external clock is in use.

Table 7.7 Flash Memory Operating States

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

Note: * When the flash memory returns to its normal operating state, a wait time of not less than 100 µs is required.

7.9 Notes on Setting Module Standby Mode

When the flash memory is set to enter module standby mode, the system clock supply is stopped to the module, the function is stopped, and the state is the same as that in standby mode. Also program operation is stopped in the flash memory. Therefore operation program should be transferred to the RAM and the program should run in the RAM. Then the flash memory should be set to enter module standby mode.

When the RAM emulation is not in use, if an interrupt is generated in module standby mode, the vector address cannot be fetched. As a result, the program may run away.

Before the flash memory is set to enter module standby mode, the corresponding bit in the interrupt enable register should be cleared to 0 and the I bit in CCR should be set to 1. Then after the flash memory enters module standby mode, NMI and address break interrupt requests should not be generated. Figure 7.6 shows a module standby mode setting when the RAM emulation is not used.

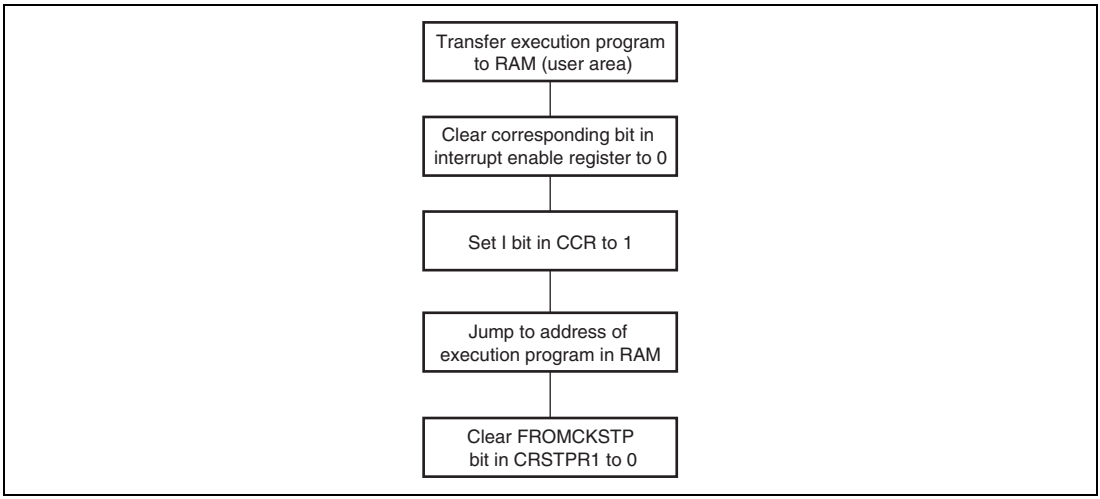


Figure 7.6 Module Standby Mode Setting when RAM Emulation is not Used

When the RAM emulation is used (an interrupt vector is provided), if an interrupt is generated in module standby mode, the vector address can be set by assigning the interrupt vector to the RAM, and this prevents the program to run away. For details, see section 7.4, Using RAM to Emulate Flash Memory.

Section 8 RAM

Microcontrollers of the H8/38799 Group include an on-chip high-speed static RAM. The RAM is connected to the CPU via a 16-bit data bus, enabling two-cycle access by the CPU to both byte and word data.

Product Classification		RAM Size	RAM Address Ranges
Flash memory version	H8/38799F	4 Kbytes	H'FFCF80 to H'FFD37F, H'FFF380 to H'FFFF7F
Masked ROM version	H8/38799	4 Kbytes	H'FFCF80 to H'FFD37F, H'FFF380 to H'FFFF7F
	H8/38798	2 Kbytes	H'FFF780 to H'FFFF7F

Section 9 I/O Ports

Microcontrollers of the H8/38799 Group incorporate 75 general I/O ports and eight general input-only ports. Port 9 is a large current port, which can drive 15 mA ($@V_{OL}$ = 1.0 V) when a low level signal is output. Any of these ports can become an input port immediately after a reset. They can also be used as I/O pins of the on-chip peripheral modules or external interrupt input pins, and these functions can be switched depending on the register settings. The registers for selecting these functions can be divided into two types: those included in I/O ports and those included in each on-chip peripheral module. General I/O ports are comprised of the port control register for controlling inputs/outputs and the port data register for storing output data and can select inputs/outputs in bit units.

For details on the execution of bit manipulation instructions to the port data register (PDR), see section 2.8.3, Bit-Manipulation Instruction. For details on block diagrams for each port, see appendix B.1, I/O Port Block Diagrams.

9.1 Port 1

Port 1 is an I/O port; its pins can also be configured to function as an SCI4 I/O pin, TPU I/O pins, and asynchronous event counter input pins. Figure 9.1 shows the pin configuration.

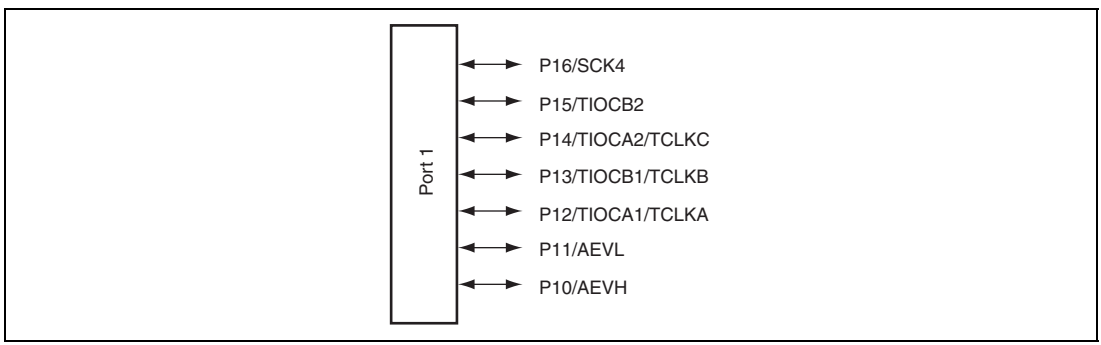


Figure 9.1 Port 1 Pin Configuration

Port 1 has the following registers.

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

9.1.1 Port Data Register 1 (PDR1)

PDR1 is a register that stores data of port 1.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
6	P16	0	R/W	If port 1 is read while PCR1 bits are set to 1, the values stored in PDR1 are read, regardless of the actual pin states. If port 1 is read while PCR1 bits are cleared to 0, the pin states are read.
5	P15	0	R/W	
4	P14	0	R/W	
3	P13	0	R/W	
2	P12	0	R/W	
1	P11	0	R/W	
0	P10	0	R/W	

9.1.2 Port Control Register 1 (PCR1)

PCR1 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 1.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
6	PCR16	0	W	Setting a PCR1 bit to 1 makes the corresponding pin (P16 to P10) an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR1 and in PDR1 are valid when the corresponding pin is designated as a general I/O pin. PCR1 is a write-only register. These bits are always read as 1.
5	PCR15	0	W	
4	PCR14	0	W	
3	PCR13	0	W	
2	PCR12	0	W	
1	PCR11	0	W	
0	PCR10	0	W	

9.1.3 Port Pull-Up Control Register 1 (PUCR1)

PUCR1 controls the pull-up MOS of the port 1 pins in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
6	PUCR16	0	R/W	When a PCR1 bit is cleared to 0, setting the corresponding PUCR1 bit to 1 turns on the pull-up MOS for the corresponding pin, while clearing the bit to 0 turns off the pull-up MOS.
5	PUCR15	0	R/W	
4	PUCR14	0	R/W	
3	PUCR13	0	R/W	
2	PUCR12	0	R/W	
1	PUCR11	0	R/W	
0	PUCR10	0	R/W	

9.1.4 Port Mode Register 1 (PMR1)

PMR1 controls the selection of functions for port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
1	AEVL	0	R/W	P11/AEVL Pin Function Switch Selects whether pin P11/AEVL is used as P11 or as AEVL. 0: P11 I/O pin 1: AEVL input pin
0	AEVH	0	R/W	P10/AEVH Pin Function Switch Selects whether pin P10/AEVH is used as P10 or as AEVH. 0: P10 I/O pin 1: AEVH input pin

9.1.5 Pin Functions

The relationship between the register settings and the port functions is shown below.

- P16/SCK4 pin

Register Name	SCSR4*[1]		PCR1	Pin Function
Bit Name	CKS3*[1]	CKS2 to CKS0*[1]	PCR16	
Setting Value	1*[1]	Other than B'111*[1]	0	P16 input pin
			1	P16 output pin
		B'111*[1]	x	SCK4 input pin*[2]
	0*[1]	x*[1]	x	SCK4 output pin*[2]

[Legend]

x: Don't care

Notes: 1. Supported only by the F-ZTAT™ version.

2. Only port function is available for the masked ROM version.

- P15/TIOCB2 pin

Register Name	TMDR_2	TIOR_2	TCR_2	PCR1	Pin Function
Bit Name	MD1, MD0	IOB3 to IOB0	CCLR1, CCLR0	PCR15	
Setting Value	B'00	B'0x00	B'xx	0	P15 input pin
				1	P15 output pin
		B'1xxx		0	P15 input/TIOCB2 input pin
				1	P15 output/TIOCB2 input pin
		B'0001 to B'0011, B'0101 to B'0111		x	TIOCB2 output pin
	B'01	B'xxxx		0	P15 input pin
				1	P15 output pin
	B'10			0	P15 input pin
				1	P15 output pin
	B'11	B'xx00		0	P15 input pin
				1	P15 output pin
		Other than B'xx00	B'10	0	P15 input pin
				1	P15 output pin
			Other than B'10	x	TIOCB2 output pin

[Legend]
x: Don't care

- P14/TIOCA2/TCLKC pin

Register Name	TMDR_2	TIOR_2	TCR_2	PCR1	Pin Function
Bit Name	MD1, MD0	IOA3 to IOA0	CCLR1, CCLR0	PCR14	
Setting Value	B'00	B'0x00	B'xx	0	P14 input pin/ TCLKC*1 input pin
				1	P14 output pin/ TCLKC*1 input pin
		B'1xxx		0	P14 input pin/TIOCA2/ TCLKC*1 input pin
				1	P14 output pin/TIOCA2/ TCLKC*1 input pin
		B'0001 to B'0011, B'0101 to B'0111		x	TIOCA2 output pin*2/ TCLKC*1 input pin
	B'01	B'xxxx		0	P14 input pin/TCLKC*1 input pin
				1	P14 output pin/TCLKC*1 input pin
	B'10	B'xx00		0	P14 input pin/TCLKC*1 input pin
				1	P14 output pin/TCLKC*1 input pin
		Other than B'xx00		x	TIOCA2 output pin*2/ TCLKC*1 input pin
	B'11	B'xx00		0	P14 input pin/TCLKC*1 input pin
				1	P14 output pin/TCLKC*1 input pin
		Other than B'xx00	B'01	0	P14 input pin/TCLKC*1 input pin
				1	P14 output pin/TCLKC*1 input pin
			Other than B'01	x	TIOCA2 output pin/ TCLKC*1 input pin

[Legend]

x: Don't care

Notes: 1. When the TPSC2 to TPSC0 bits in TCR_2 are set to B'110, the pin function becomes the TCLKC input pin.

2. The output of the TIOCB2 pin is disabled.

- P13/TIOCB1/TCLKB pin

Register Name	TMDR_1	TIOR_1	TCR_1	PCR1	Pin Function
Bit Name	MD1, MD0	IOB3 to IOB0	CCLR1, CCLR0	PCR13	
Setting Value	B'00	B'0x00	B'xx	0	P13 input pin/TCLKB* input pin
				1	P13 output pin/TCLKB* input pin
		B'1xxx		0	P13 input pin/TIOCB1/TCLKB* input pin
				1	P13 output pin/TIOCB1/TCLKB* input pin
		B'0001 to B'0011, B'0101 to B'0111		x	TIOCB1 output pin/TCLKB* input pin
	B'01	B'xxxx		0	P13 input pin/TCLKB* input pin
				1	P13 output pin/TCLKB* input pin
	B'10			0	P13 input pin/TCLKB* input pin
				1	P13 output pin/TCLKB* input pin
	B'11	B'xx00		0	P13 input pin/TCLKB* input pin
				1	P13 output pin/TCLKB* input pin
		Other than B'xx00	B'10	0	P13 input pin/TCLKB* input pin
				1	P13 output pin/TCLKB* input pin
			Other than B'10	x	TIOCB1 output pin/TCLKB* input pin

[Legend]

x: Don't care

Note: * When the TPSC2 to TPSC0 bits in either TCR_1 or TCR_2 are set to B'101, the pin function becomes the TCLKB input pin.

- P12/TIOCA1/TCLKA pin

Register Name	TMDR_1	TIOR_1	TCR_1	PCR1	Pin Function
Bit Name	MD1, MD0	IOA3 to IOA0	CCLR1, CCLR0	PCR12	
Setting Value	B'00	B'0x00	B'xx	0	P12 input pin/TCLKA*1 input pin
				1	P12 output pin/TCLKA*1 input pin
		B'1xxx		0	P12 input pin/TIOCA1/TCLKA*1 input pin
				1	P12 output pin/TIOCA1/TCLKA*1 input pin
		B'0001 to B'0011, B'0101 to B'0111		x	TIOCA1 output pin*2/TCLKA*1 input pin
	B'01	B'xxxx		0	P12 input pin/TCLKA*1 input pin
				1	P12 output pin/TCLKA*1 input pin
	B'10	B'xx00		0	P12 input pin/TCLKA*1 input pin
				1	P12 output pin/TCLKA*1 input pin
		Other than B'xx00		x	TIOCA1 output pin*2/TCLKA*1 input pin
	B'11	B'xx00		0	P12 input pin/TCLKA*1 input pin
				1	P12 output pin/TCLKA*1 input pin
		Other than B'xx00	B'01	0	P12 input pin/TCLKA*1 input pin
				1	P12 output pin/TCLKA*1 input pin
			Other than B'01	x	TIOCA1 output pin/TCLKA*1 input pin

[Legend]

x: Don't care

Notes: 1. When the TPSC2 to TPSC0 bits in either TCR_1 or TCR_2 are set to B'100, the pin function becomes the TCLKA input pin.

2. The output of the TIOCB1 pin is disabled.

- P11/AEVL pin

Register Name	PMR1	PCR1	Pin Function
Bit Name	AEVL	PCR11	
Setting Value	0	0	P11 input pin
		1	P11 output pin
	1	x	AEVL input pin

[Legend]
x: Don't care

- P10/AEVH pin

Register Name	PMR1	PCR1	Pin Function
Bit Name	AEVH	PCR10	
Setting Value	0	0	P10 input pin
		1	P10 output pin
	1	x	AEVH input pin

[Legend]
x: Don't care

9.1.6 Input Pull-Up MOS

Port 1 has an on-chip input pull-up MOS function that can be controlled by software. When a PCR1 bit is cleared to 0, setting the corresponding PUCR1 bit to 1 turns on the input pull-up MOS for that pin. The input pull-up MOS function is in the off state after a reset.

(n = 6 to 0)

PCR1n	0		1
PUCR1n	0	1	x
Input Pull-Up MOS	Off	On	Off

[Legend]
x: Don't care

9.2 Port 3

Port 3 is an I/O port; its pins can also be configured to function as an SCI4 I/O pin, SCI3_2 I/O pin, IIC2 I/O pin, and RTC output pin. Figure 9.2 shows the pin configuration.

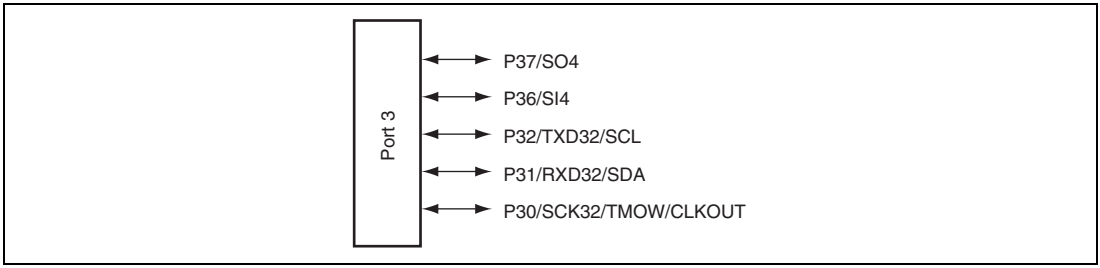


Figure 9.2 Port 3 Pin Configuration

Port 3 has the following registers.

- Port data register 3 (PDR3)
- Port control register 3 (PCR3)
- Port pull-up control register 3 (PUCR3)
- Port mode register 3 (PMR3)

9.2.1 Port Data Register 3 (PDR3)

PDR3 is a register that stores data of port 3.

Bit	Bit Name	Initial Value	R/W	Description
7	P37	0	R/W	If port 3 is read while PCR3 bits are set to 1, the values stored in PDR3 are read, regardless of the actual pin states. If port 3 is read while PCR3 bits are cleared to 0, the pin states are read.
6	P36	0	R/W	
5 to 3	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
2	P32	0	R/W	If port 3 is read while PCR3 bits are set to 1, the values stored in PDR3 are read, regardless of the actual pin states. If port 3 is read while PCR3 bits are cleared to 0, the pin states are read.
1	P31	0	R/W	
0	P30	0	R/W	

9.2.2 Port Control Register 3 (PCR3)

PCR3 selects inputs/outputs in bit units for pins of port 3.

Bit	Bit Name	Initial Value	R/W	Description
7	PCR37	0	W	Setting a PCR3 bit to 1 makes the corresponding pin (P37 or P36) an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR3 and in PDR3 are valid when the corresponding pin is designated as a general I/O pin. PCR3 is a write-only register. These bits are always read as 1.
6	PCR36	0	W	
5 to 3	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
2	PCR32	0	W	Setting a PCR3 bit to 1 makes the corresponding pins (P32 to P30) an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR3 and in PDR3 are valid when the corresponding pin is designated as a general I/O pin. PCR3 is a write-only register. These bits are always read as 1.
1	PCR31	0	W	
0	PCR30	0	W	

9.2.3 Port Pull-Up Control Register 3 (PUCR3)

PUCR3 controls the pull-up MOS of port 3 pins in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	PUCR37	0	R/W	When a PCR3 bit is cleared to 0, setting the corresponding PUCR3 bit to 1 turns on the pull-up MOS for the corresponding pin, while clearing the bit to 0 turns off the pull-up MOS.
6	PUCR36	0	R/W	
5 to 1	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
0	PUCR30	0	R/W	When a PCR3 bit is cleared to 0, setting the corresponding PUCR3 bit to 1 turns on the pull-up MOS for the corresponding pin, while clearing the bit to 0 turns off the pull-up MOS.

9.2.4 Port Mode Register 3 (PMR3)

PMR3 controls the selection of functions for port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
0	TMOW	0	R/W	P30/SCK32/TMOW/CLKOUT Pin Function Switch Selects whether pin P30/SCK32/TMOW/CLKOUT is used as P30/SCK32 or as TMOW/CLKOUT. 0: P30/SCK32 I/O pin 1: TMOW/CLKOUT output pin

9.2.5 Pin Functions

The relationship between the register settings and the port functions is shown below.

- P37/SO4 pin

Register Name	SCR4*[1]	PCR3	Pin Function
Bit Name	TE*[1]	PCR37	
Setting Value	0*[1]	0	P37 input pin
		1	P37 output pin
	1*[1]	x	SO4 output pin*[2]

[Legend]

x: Don't care

Notes: 1. Supported only by the F-ZTAT™ version.

2. Only port function is available for the masked ROM version.

- P36/SI4 pin

Register Name	SCR4*[1]	PCR3	Pin Function
Bit Name	RE*[1]	PCR36	
Setting Value	0*[1]	0	P36 input pin
		1	P36 output pin
	1*[1]	x	SI4 input pin*[2]

[Legend]

x: Don't care

Notes: 1. Supported only by the F-ZTAT™ version.

2. Only port function is available for the masked ROM version.

- P32/TXD32/SCL pin

Register Name	ICCR1	PFCR	SPCR	PCR3	Pin Function
Bit Name	ICE	SC32S	SPC32	PCR32	
Setting Value	0	0	0	0	P32 input pin
				1	P32 output pin
			1	x	TXD32 output pin
		1	x	0	P32 input pin
				1	P32 output pin
	1	x		x	SCL I/O pin

[Legend]
x: Don't care

- P31/RXD32/SDA pin

Register Name	ICCR1	PFCR	SCR3_2	PCR3	Pin Function
Bit Name	ICE	SC32S	RE	PCR31	
Setting Value	0	0	0	0	P31 input pin
				1	P31 output pin
			1	x	RXD32 input pin
		1	x	0	P31 input pin
				1	P31 output pin
	1	x		x	SDA I/O pin

[Legend]
x: Don't care

- P30/SCK32/TMOW/CLKOUT pin

Register Name	PMR3	PFCR			SMR3_2	SCR3_2		PCR3	Pin Function
Bit Name	TMOW	CLKOUT1	CLKOUT0	SC32S	COM	CKE1	CKE0	PCR30	
Setting Value	0	x	x	0	0	0	0	0	P30 input pin
								1	P30 output pin
							1	x	SCK32 output pin
						1	0		SCK32 input pin
							1		Setting prohibited
					1	0	0		SCK32 output pin
							1		Setting prohibited
						1	0		SCK32 input pin
							1		Setting prohibited
				1	x	x	x	0	P30 input pin
								1	P30 output pin
	1	0	0	x				x	CLKOUT output pin (ϕ_{osc})
			1						CLKOUT output pin (ϕ_{osc}/2)
		1	0						CLKOUT output pin (ϕ_{osc}/4)
			1						TMOW output pin

[Legend]
x: Don't care

9.2.6 Input Pull-Up MOS

Port 3 has an on-chip input pull-up MOS function that can be controlled by software. When a PCR3 bit is cleared to 0, setting the corresponding PUCR3 bit to 1 turns on the input pull-up MOS for that pin. The input pull-up MOS function is in the off state after a reset.

(n = 7, 6, 0)

PCR3n	0		1
PUCR3n	0	1	x
Input Pull-Up MOS	Off	On	Off

[Legend]
x: Don't care

9.3 Port 4

Port 4 is an I/O port; its pins can also be configured to function as SCI3_1 data I/O pins and timer F I/O pins. Figure 9.3 shows its pin configuration.

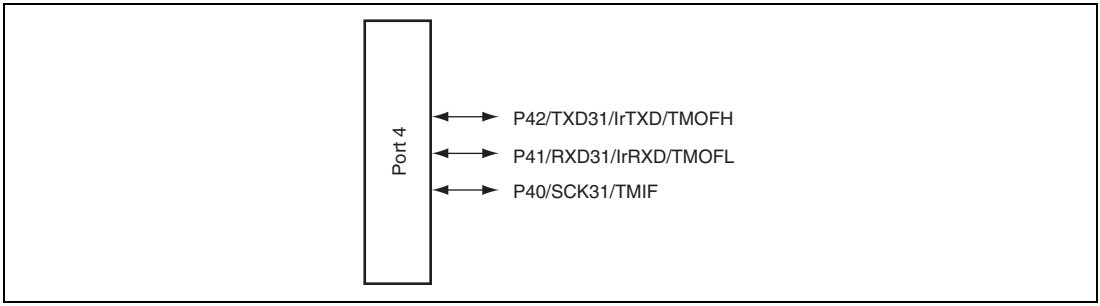


Figure 9.3 Port 4 Pin Configuration

Port 4 has the following registers.

- Port data register 4 (PDR4)
- Port control register 4 (PCR4)
- Port mode register 4 (PMR4)

9.3.1 Port Data Register 4 (PDR4)

PDR4 is a register that stores data of port 4.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
2	P42	0	R/W	If port 4 is read while PCR4 bits are set to 1, the values stored in PDR4 are read, regardless of the actual pin states. If port 4 is read while PCR4 bits are cleared to 0, the pin states are read.
1	P41	0	R/W	
0	P40	0	R/W	

9.3.2 Port Control Register 4 (PCR4)

PCR4 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 4.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
2	PCR42	0	W	Setting a PCR4 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR4 and in PDR4 are valid when the corresponding pin is designated as a general I/O pin. PCR4 is a write-only register. These bits are always read as 1.
1	PCR41	0	W	
0	PCR40	0	W	

9.3.3 Port Mode Register 4 (PMR4)

PMR4 controls the selection of functions for port 4 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
2	TMOFH	0	R/W	P42/TXD31/IrTXD/TMOFH Pin Function Switch Selects whether pin P42/TXD31/IrTXD/TMOFH is used as P42 or TXD31/IrTXD, or as TMOFH. 0: P42 I/O pin or TXD31/IrTXD output pin 1: TMOFH output pin
1	TMOFL	0	R/W	P41/RXD31/IrRXD/TMOFL Pin Function Switch Selects whether pin P41/RXD31/IrRXD/TMOFL is used as P41 or RXD31/IrRXD, or as TMOFL. 0: P41 I/O pin or RXD31/IrRXD input pin 1: TMOFL output pin
0	TMIF	0	R/W	P40/SCK31/TMIF Pin Function Switch Selects whether pin P40/SCK31/TMIF is used as P40/SCK31 or as TMIF. 0: P40/SCK31 I/O pin 1: TMIF input pin

9.3.4 Pin Functions

The relationship between the register settings and the port functions is shown below.

- P42/TXD31/IrTXD/TMOFH pin

Register Name	PMR4	PFCR	SPCR	IrCR	PCR4	Pin Function
Bit Name	TMOFH	SC31S	SPC31	IrE	PCR42	
Setting Value	0	0	0	x	0	P42 input pin
					1	P42 output pin
			1	0	x	TXD31 output pin
				1		IrTXD output pin
		1	x	x	0	P42 input pin
					1	P42 output pin
	1	x			x	TMOFH output pin

[Legend]
x: Don't care

- P41/RXD31/IrRXD/TMOFL pin

Register Name	PMR4	PFCR	SCR3_1	IrCR	PCR4	Pin Function
Bit Name	TMOFL	SC31S	RE	IrE	PCR41	
Setting Value	0	0	0	x	0	P41 input pin
					1	P41 output pin
			1	0	x	RXD31 output pin
				1		IrRXD output pin
		1	x	x	0	P41 input pin
					1	P41 output pin
	1	x			x	TMOFL output pin

[Legend]
x: Don't care

- P40/SCK31/TMIF pin

Register Name	PMR4	PFCR	SMR3_1	SCR3_1		PCR4	Pin Function
Bit Name	TMIF	SC31S	COM	CKE1	CKE0	PCR40	
Setting Value	0	0	0	0	0	0	P40 input pin
						1	P40 output pin
					1	x	SCK31 output pin
				1	0		SCK31 input pin
					1		Setting prohibited
			1	0	0		SCK31 output pin
					1		Setting prohibited
				1	0		SCK31 input pin
					1		Setting prohibited
		1	x	x	x	0	P40 input pin
						1	P40 output pin
	1	x				x	TMIF input pin

[Legend]
x: Don't care

9.4 Port 5

Port 5 is an I/O port; its pins can also be configured to function as wakeup interrupt input pins. Figure 9.4 shows the pin configuration.

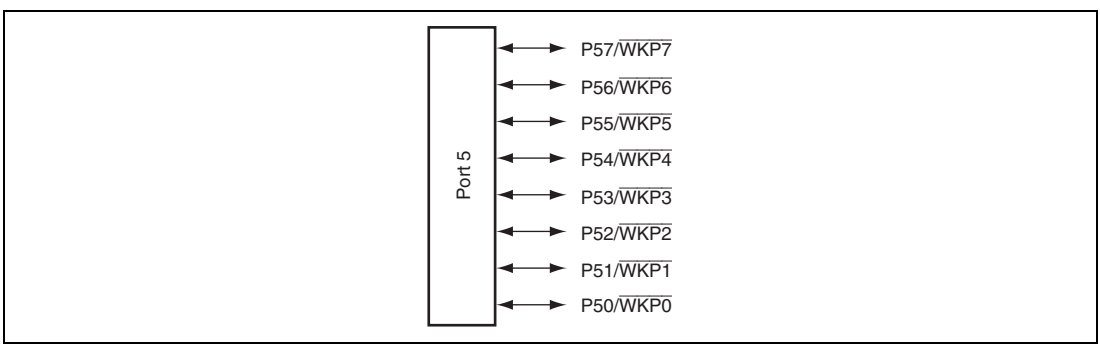


Figure 9.4 Port 5 Pin Configuration

Port 5 has the following registers.

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

9.4.1 Port Data Register 5 (PDR5)

PDR5 is a register that stores data of port 5.

Bit	Bit Name	Initial Value	R/W	Description
7	P57	0	R/W	If port 5 is read while PCR5 bits are set to 1, the values stored in PDR5 are read, regardless of the actual pin states. If port 5 is read while PCR5 bits are cleared to 0, the pin states are read.
6	P56	0	R/W	
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	

9.4.2 Port Control Register 5 (PCR5)

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

Bit	Bit Name	Initial Value	R/W	Description
7	PCR57	0	W	Setting a PCR5 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR5 and in PDR5 are valid when the corresponding pin is designated as a general I/O pin.
6	PCR56	0	W	
5	PCR55	0	W	
4	PCR54	0	W	
3	PCR53	0	W	PCR5 is a write-only register. These bits are always read as 1.
2	PCR52	0	W	
1	PCR51	0	W	
0	PCR50	0	W	

9.4.3 Port Pull-Up Control Register 5 (PUCR5)

PUCR5 controls the pull-up MOS of the port 5 pins in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	PUCR57	0	R/W	When a PCR5 bit is cleared to 0, setting the corresponding PUCR5 bit to 1 turns on the pull-up MOS for the corresponding pin, while clearing the bit to 0 turns off the pull-up MOS.
6	PUCR56	0	R/W	
5	PUCR55	0	R/W	
4	PUCR54	0	R/W	
3	PUCR53	0	R/W	
2	PUCR52	0	R/W	
1	PUCR51	0	R/W	
0	PUCR50	0	R/W	

9.4.4 Port Mode Register 5 (PMR5)

PMR5 controls the selection of functions for port 5 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	WKP7	0	R/W	P5n/$\overline{\text{WKPn}}$ Pin Function Switch
6	WKP6	0	R/W	These bits select whether the pin is used as P5n or $\overline{\text{WKPn}}$.
5	WKP5	0	R/W	0: P5n I/O pin
4	WKP4	0	R/W	1: $\overline{\text{WKPn}}$ input pin
3	WKP3	0	R/W	(n = 7 to 0)
2	WKP2	0	R/W	
1	WKP1	0	R/W	
0	WKP0	0	R/W	

9.4.5 Pin Functions

The relationship between the register settings and the port functions is shown below.

- P57/$\overline{\text{WKP7}}$ to P50/$\overline{\text{WKP0}}$ pins

(n = 7 to 0)

Register Name	PMR5	PCR5	Pin Function
Bit Name	WKPn	PCR5n	
Setting Value	0	0	P5n input pin
		1	P5n output pin
	1	0	$\overline{\text{WKPn}}$ input pin
		1	Setting prohibited

9.4.6 Input Pull-Up MOS

Port 5 has an on-chip input pull-up MOS function that can be controlled by software. When the PCR5 bit is cleared to 0, setting the corresponding PUCR5 bit to 1 turns on the input pull-up MOS for that pin. The input pull-up MOS function is in the off state after a reset.

(n = 7 to 0)

PCR5n	0		1
PUCR5n	0	1	x
Input Pull-Up MOS	Off	On	Off

[Legend]
x: Don't care

9.5 Port 6

Port 6 is a general I/O port, the pin configuration of which is shown in figure 9.5.

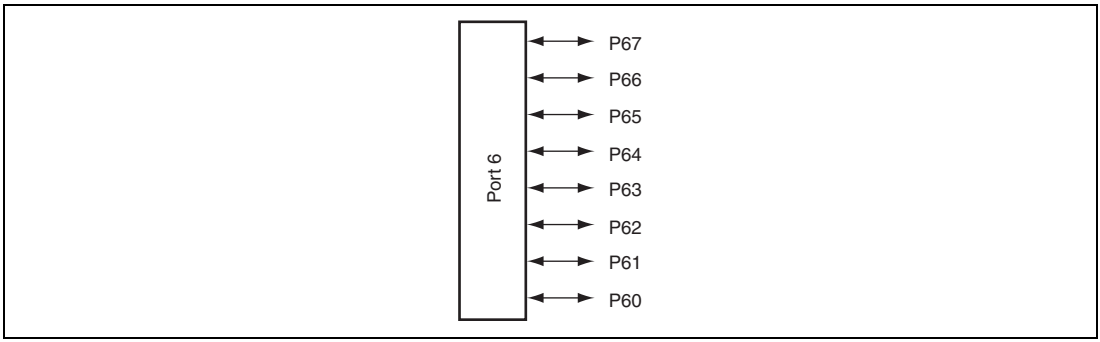


Figure 9.5 Port 6 Pin Configuration

Port 6 has the following registers.

- Port data register 6 (PDR6)
- Port control register 6 (PCR6)
- Port pull-up control register 6 (PUCR6)

9.5.1 Port Data Register 6 (PDR6)

PDR6 is a register that stores data of port 6.

Bit	Bit Name	Initial Value	R/W	Description
7	P67	0	R/W	If port 6 is read while PCR6 bits are set to 1, the values stored in PDR6 are read, regardless of the actual pin states. If port 6 is read while PCR6 bits are cleared to 0, the pin states are read.
6	P66	0	R/W	
5	P65	0	R/W	
4	P64	0	R/W	
3	P63	0	R/W	
2	P62	0	R/W	
1	P61	0	R/W	
0	P60	0	R/W	

9.5.2 Port Control Register 6 (PCR6)

PCR6 selects inputs/outputs in bit units for pins of port 6.

Bit	Bit Name	Initial Value	R/W	Description
7	PCR67	0	W	Setting a PCR6 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. PCR6 is a write-only register. These bits are always read as 1.
6	PCR66	0	W	
5	PCR65	0	W	
4	PCR64	0	W	
3	PCR63	0	W	
2	PCR62	0	W	
1	PCR61	0	W	
0	PCR60	0	W	

9.5.3 Port Pull-Up Control Register 6 (PUCR6)

PUCR6 controls the pull-up MOS of the port 6 pins in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	PUCR67	0	R/W	When a PCR6 bit is cleared to 0, setting the corresponding PUCR6 bit to 1 turns on the pull-up MOS for the corresponding pin, while clearing the bit to 0 turns off the pull-up MOS.
6	PUCR66	0	R/W	
5	PUCR65	0	R/W	
4	PUCR64	0	R/W	
3	PUCR63	0	R/W	
2	PUCR62	0	R/W	
1	PUCR61	0	R/W	
0	PUCR60	0	R/W	

9.5.4 Pin Functions

The relationship between the register settings and the port functions is shown below.

- P67 to P60 pins

(n = 7 to 0)

Register Name	PCR6	Pin Function
Bit Name	PCR6n	
Setting Value	0	P6n input pin
	1	P6n output pin

9.5.5 Input Pull-Up MOS

Port 6 has an on-chip input pull-up MOS function that can be controlled by software. When the PCR6 bit is cleared to 0, setting the corresponding PUCR6 bit to 1 turns on the input pull-up MOS for that pin. The input pull-up MOS function is in the off state after a reset.

(n = 7 to 0)

PCR6n	0		1
PUCR6n	0	1	x
Input Pull-Up MOS	Off	On	Off

[Legend]
x: Don't care

9.6 Port 7

Port 7 is a general I/O port, the pin configuration of which is shown in figure 9.6

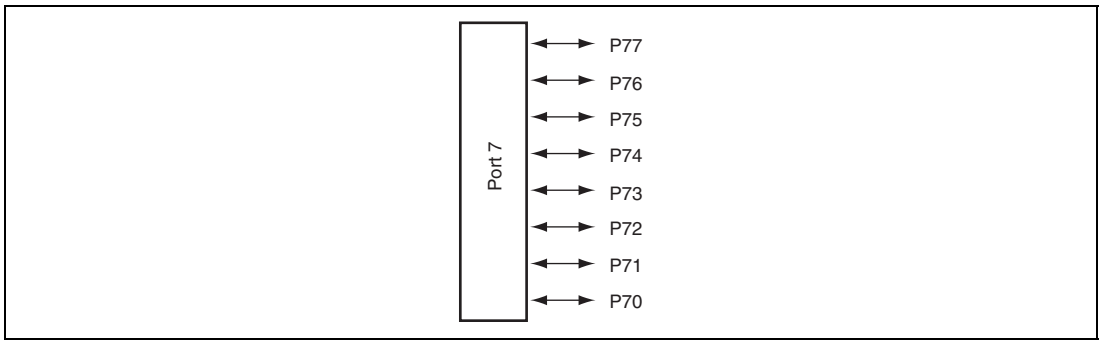


Figure 9.6 Port 7 Pin Configuration

Port 7 has the following registers.

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

9.6.1 Port Data Register 7 (PDR7)

PDR7 is a register that stores data of port 7.

Bit	Bit Name	Initial Value	R/W	Description
7	P77	0	R/W	If port 7 is read while PCR7 bits are set to 1, the values stored in PDR7 are read, regardless of the actual pin states. If port 7 is read while PCR7 bits are cleared to 0, the pin states are read.
6	P76	0	R/W	
5	P75	0	R/W	
4	P74	0	R/W	
3	P73	0	R/W	
2	P72	0	R/W	
1	P71	0	R/W	
0	P70	0	R/W	

9.6.2 Port Control Register 7 (PCR7)

PCR7 selects inputs/outputs in bit units for pins of port 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PCR77	0	W	Setting a PCR7 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. PCR7 is a write-only register. These bits are always read as 1.
6	PCR76	0	W	
5	PCR75	0	W	
4	PCR74	0	W	
3	PCR73	0	W	
2	PCR72	0	W	
1	PCR71	0	W	
0	PCR70	0	W	

9.6.3 Pin Functions

The relationship between the register settings and the port functions is shown below.

- P77 to P70 pins

(n = 7 to 0)

Register Name	PCR7	Pin Function
Bit Name	PCR7n	
Setting Value	0	P7n input pin
	1	P7n output pin

9.7 Port 8

Port 8 is a general I/O port, the pin configuration of which is shown in figure 9.7.

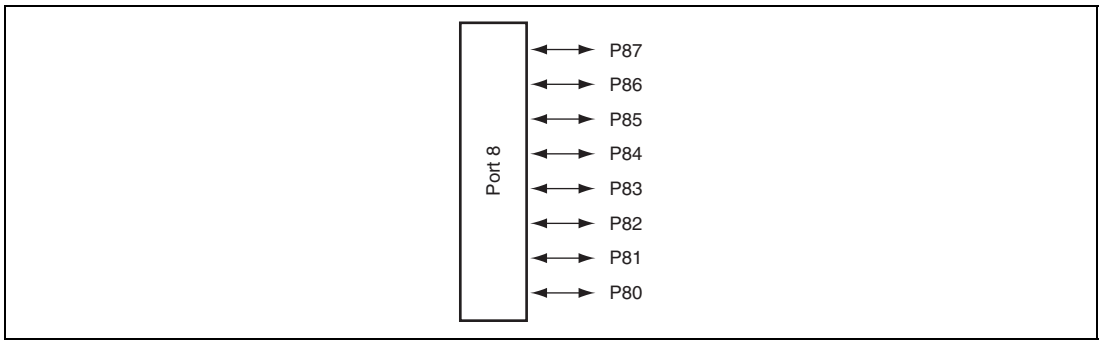


Figure 9.7 Port 8 Pin Configuration

Port 8 has the following registers.

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

9.7.1 Port Data Register 8 (PDR8)

PDR8 is a register that stores data of port 8.

Bit	Bit Name	Initial Value	R/W	Description
7	P87	0	R/W	If port 8 is read while PCR8 bits are set to 1, the values stored in PDR8 are read, regardless of the actual pin states. If port 8 is read while PCR8 bits are cleared to 0, the pin states are read.
6	P86	0	R/W	
5	P85	0	R/W	
4	P84	0	R/W	
3	P83	0	R/W	
2	P82	0	R/W	
1	P81	0	R/W	
0	P80	0	R/W	

9.7.2 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	Setting a PCR8 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. PCR8 is a write-only register. These bits are always read as 1.
6	PCR86	0	W	
5	PCR85	0	W	
4	PCR84	0	W	
3	PCR83	0	W	
2	PCR82	0	W	
1	PCR81	0	W	
0	PCR80	0	W	

9.7.3 Pin Functions

The relationship between the register settings and the port functions is shown below.

- P87 to P80 pins

(n = 7 to 0)

Register Name	PCR8	Pin Function
Bit Name	PCR8n	
Setting Value	0	P8n input pin
	1	P8n output pin

9.8 Port 9

Port 9 is a general I/O port; its pins can also be configured to function as an external interrupt input pin and PWM output pins. Figure 9.8 shows the pin configuration.

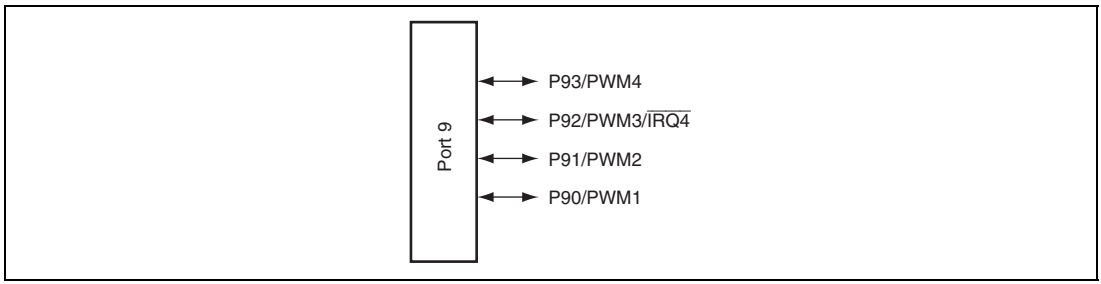


Figure 9.8 Port 9 Pin Configuration

Port 9 has the following registers.

- Port data register 9 (PDR9)
- Port control register 9 (PCR9)
- Port mode register 9 (PMR9)

9.8.1 Port Data Register 9 (PDR9)

PDR9 is a register that stores data of port 9.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
3	P93	1	R/W	If port 9 is read while PCR9 bits are set to 1, the values stored in PDR9 are read, regardless of the actual pin states. If port 9 is read while PCR9 bits are cleared to 0, the pin states are read.
2	P92	1	R/W	
1	P91	1	R/W	
0	P90	1	R/W	

9.8.2 Port Control Register 9 (PCR9)

PCR9 selects inputs/outputs in bit units for pins of port 9.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
3	PCR93	0	W	Setting a PCR9 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR9 and in PDR9 are valid when the corresponding pin is designated as a general I/O pin. PCR9 is a write-only register. These bits are always read as 1.
2	PCR92	0	W	
1	PCR91	0	W	
0	PCR90	0	W	

9.8.3 Port Mode Register 9 (PMR9)

PMR9 controls the selection of functions for port 9 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
3	—	0	R/W	Reserved Although this bit is readable/writable, 1 should not be written to this bit.
2	IRQ4	0	R/W	P92/$\overline{\text{IRQ4}}$ Pin Function Switch Selects whether pin P92/$\overline{\text{IRQ4}}$ is used as P92 or as $\overline{\text{IRQ4}}$. 0: P92 I/O pin 1: $\overline{\text{IRQ4}}$ input pin

Bit	Bit Name	Initial Value	R/W	Description
1	PWM2	0	R/W	P9n/PWMn+1 Pin Function Switch
0	PWM1	0	R/W	Select whether pin P9n/PWMn+1 is used as P9n or as PWMn+1. (n = 1, 0) 0: P9n I/O pin 1: PWMn+1 output pin

9.8.4 Pin Functions

The relationship between the register settings and the port functions is shown below.

- P93/PWM4 pin

Register Name	PFCR	PCR9	Pin Function
Bit Name	PWM4	PCR93	
Setting Value	0	0	P93 input pin
		1	P93 output pin
	1	x	PWM4 output pin

[Legend]
x: Don't care

- P92/PWM3/$\overline{\text{IRQ4}}$ pin

Register Name	PMR9	PFCR	PCR9	Pin Function
Bit Name	IRQ4	PMW3	PCR92	
Setting Value	0	0	0	P92 input pin
			1	P92 output pin
		1	x	PWM3 output pin
	1	x	0	$\overline{\text{IRQ4}}$ input pin
			1	Setting prohibited

[Legend]
x: Don't care

- P91/PWM2 and P90/PWM1 pins

(n = 1, 0)

Register Name	PMR9	PCR9	Pin Function
Bit Name	PWMn+1	PCR9n	
Setting Value	0	0	P9n input pin
		1	P9n output pin
	1	x	PWMn+1 output pin

[Legend]
x: Don't care

9.9 Port A

Port A is a general I/O port, the pin configuration of which is shown in figure 9.9.

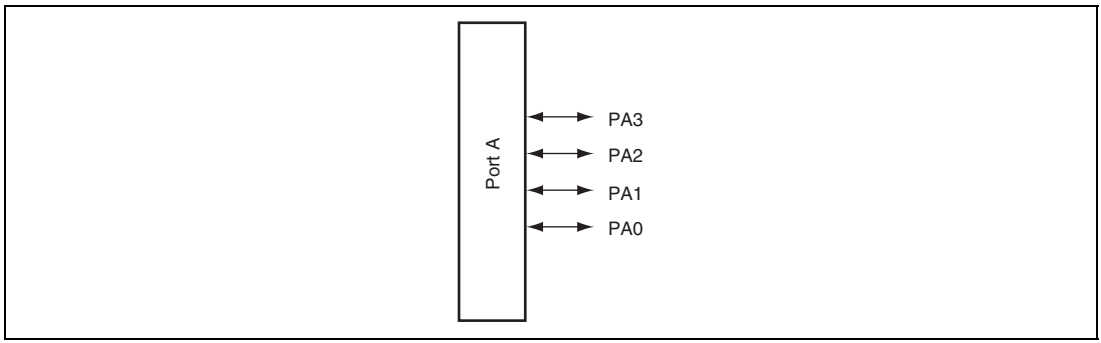


Figure 9.9 Port A Pin Configuration

Port A has the following registers.

- Port data register A (PDRA)
- Port control register A (PCRA)

9.9.1 Port Data Register A (PDRA)

PDRA is a register that stores data of port A.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
3	PA3	0	R/W	If port A is read while PCRA bits are set to 1, the values stored in PDRA are read, regardless of the actual pin states. If port A is read while PCRA bits are cleared to 0, the pin states are read.
2	PA2	0	R/W	
1	PA1	0	R/W	
0	PA0	0	R/W	

9.9.2 Port Control Register A (PCRA)

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
3	PCRA3	0	W	Setting a PCRA bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. PCRA is a write-only register. These bits are always read as 1.
2	PCRA2	0	W	
1	PCRA1	0	W	
0	PCRA0	0	W	

9.9.3 Pin Functions

The relationship between the register settings and the port functions is shown below.

- PA3 to PA0 pins

(n = 3 to 0)

Register Name	PCRA	Pin Function
Bit Name	PCRAn	
Setting Value	0	PAn input pin
	1	PAn output pin

9.10 Port B

Port B is an input-only port: its pins can also be configured to function as an interrupt input pin and analog input pin. Figure 9.10 shows the pin configuration.

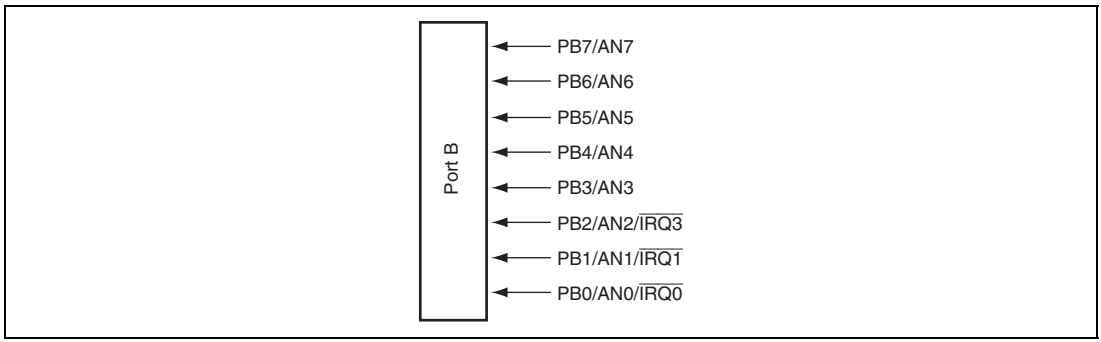


Figure 9.10 Port B Pin Configuration

Port B has the following registers.

- Port data register B (PDRB)
- Port mode register B (PMRB)

9.10.1 Port Data Register B (PDRB)

PDRB is a register that stores data of port B.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7	Undefined	R	Reading PDRB always gives the pin states. However, if a port B pin is selected as an analog input channel by the CH3 to CH0 bits in AMR of the A/D converter, that pin is read as 0 regardless of the input voltage.
6	PB6	Undefined	R	
5	PB5	Undefined	R	
4	PB4	Undefined	R	
3	PB3	Undefined	R	
2	PB2	Undefined	R	
1	PB1	Undefined	R	
0	PB0	Undefined	R	

9.10.2 Port Mode Register B (PMRB)

PMRB controls the selection of the port B pin functions and switches the TEST/$\overline{\text{ADTRG}}$ function.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
4	ADTSTCHG	0	R/W	TEST/$\overline{\text{ADTRG}}$ Pin Function Switch Selects whether pin TEST/$\overline{\text{ADTRG}}$ is used as TEST or as $\overline{\text{ADTRG}}$. 0: TEST pin 1: $\overline{\text{ADTRG}}$ input pin For details on the setting of the $\overline{\text{ADTRG}}$ input pin, refer to section 20.4.2, External Trigger Input Timing.
3	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
2	IRQ3	0	R/W	PB2/AN2/$\overline{\text{IRQ3}}$* Pin Function Switch Selects whether pin PB2/AN2/$\overline{\text{IRQ3}}$ is used as PB2/AN2 or as $\overline{\text{IRQ3}}$. 0: PB2/AN2 input pin 1: $\overline{\text{IRQ3}}$ input pin
1	IRQ1	0	R/W	PB1/AN1/$\overline{\text{IRQ1}}$* Pin Function Switch Selects whether pin PB1/AN1/$\overline{\text{IRQ1}}$ is used as PB1/AN1 or as $\overline{\text{IRQ1}}$. 0: PB1/AN1 input pin 1: $\overline{\text{IRQ1}}$ input pin
0	IRQ0	0	R/W	PB0/AN0/$\overline{\text{IRQ0}}$* Pin Function Switch Selects whether pin PB0/AN0/$\overline{\text{IRQ0}}$ is used as PB0/AN0 or as $\overline{\text{IRQ0}}$. 0: PB0/AN0 input pin 1: $\overline{\text{IRQ0}}$ input pin

Note: * For information on the power supply voltage (V_{CC}) and analog power supply voltage (AV_{CC}) when using this pin as an IRQn (n = 0, 1, 3) pin, see section 25.2.2 or 25.4.2, DC Characteristics.

9.10.3 Pin Functions

The relationship between the register settings and the port functions is shown below.

- PB7/AN7 pin

Register Name	AMR	Pin Function
Bit Name	CH3 to CH0	
Setting Value	Other than B'1011	PB7 input pin
	B'1011	AN7 input pin

- PB6/AN6 pin

Register Name	AMR	Pin Function
Bit Name	CH3 to CH0	
Setting Value	Other than B'1010	PB6 input pin
	B'1010	AN6 input pin

- PB5/AN5 pin

Register Name	AMR	Pin Function
Bit Name	CH3 to CH0	
Setting Value	Other than B'1001	PB5 input pin
	B'1001	AN5 input pin

- PB4/AN4 pin

Register Name	AMR	Pin Function
Bit Name	CH3 to CH0	
Setting Value	Other than B'1000	PB4 input pin
	B'1000	AN4 input pin

- PB3/AN3 pin

Register Name	AMR	Pin Function
Bit Name	CH3 to CH0	
Setting Value	Other than B'0111	PB3 input pin
	B'0111	AN3 input pin

- PB2/AN2/$\overline{\text{IRQ3}}$ pin

Register Name	PMRB	AMR	Pin Function
Bit Name	IRQ3	CH3 to CH0	
Setting Value	0	Other than B'0110	PB2 input pin
		B'0110	AN2 input pin
	1	Other than B'0110	$\overline{\text{IRQ3}}$ input pin
		B'0110	Setting prohibited

- PB1/AN1/$\overline{\text{IRQ1}}$ pin

Register Name	PMRB	AMR	Pin Function
Bit Name	IRQ1	CH3 to CH0	
Setting Value	0	Other than B'0101	PB1 input pin
		B'0101	AN1 input pin
	1	Other than B'0101	$\overline{\text{IRQ1}}$ input pin
		B'0101	Setting prohibited

- PB0/AN0/$\overline{\text{IRQ0}}$ pin

Register Name	PMRB	AMR	Pin Function
Bit Name	IRQ0	CH3 to CH0	
Setting Value	0	Other than B'0100	PB0 input pin
		B'0100	AN0 input pin
	1	Other than B'0100	$\overline{\text{IRQ0}}$ input pin
		B'0100	Setting prohibited

9.11 Port C

Port C is a general I/O port, the pin configuration of which is shown in figure 9.11.

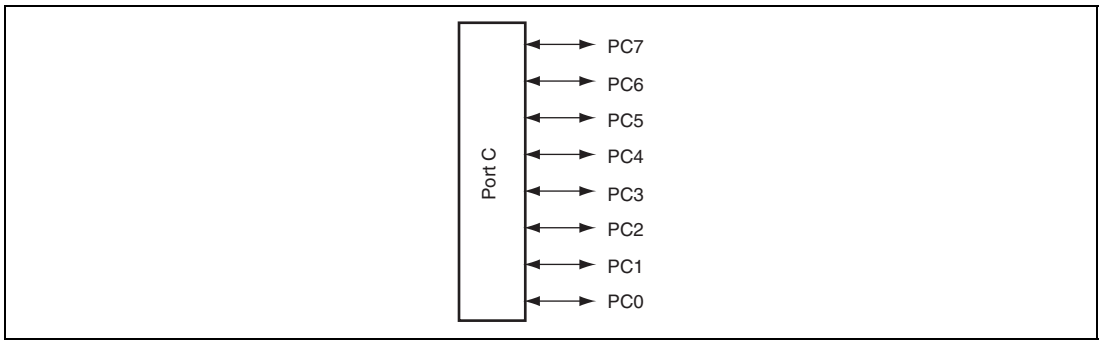


Figure 9.11 Port C Pin Configuration

Port C has the following registers.

- Port data register C (PDRC)
- Port control register C (PCRC)

9.11.1 Port Data Register C (PDRC)

PDRC is a register that stores data of port C.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7	0	R/W	If port C is read while PCRC bits are set to 1, the values stored in PDRC are read, regardless of the actual pin states. If port C is read while PCRC bits are cleared to 0, the pin states are read.
6	PC6	0	R/W	
5	PC5	0	R/W	
4	PC4	0	R/W	
3	PC3	0	R/W	
2	PC2	0	R/W	
1	PC1	0	R/W	
0	PC0	0	R/W	

9.11.2 Port Control Register C (PCRC)

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

Bit	Bit Name	Initial Value	R/W	Description
7	PCRC7	0	W	Setting a PCRC bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. PCRC is a write-only register. These bits are always read as 1.
6	PCRC6	0	W	
5	PCRC5	0	W	
4	PCRC4	0	W	
3	PCRC3	0	W	
2	PCRC2	0	W	
1	PCRC1	0	W	
0	PCRC0	0	W	

9.11.3 Pin Functions

The relationship between the register settings and the port functions is shown below.

- PC7 to PC0 pins

(n = 7 to 0)

Register Name	PCRC	Pin Function
Bit Name	PCRCn	
Setting Value	0	PCn input pin
	1	PCn output pin

9.12 Port E

Port E is an I/O port; its pins can also be configured to function as external interrupt input pins, SCI3_2 I/O pins, SCI3_3 I/O pins, and timer C input pin. Figure 9.12 shows its pin configuration.

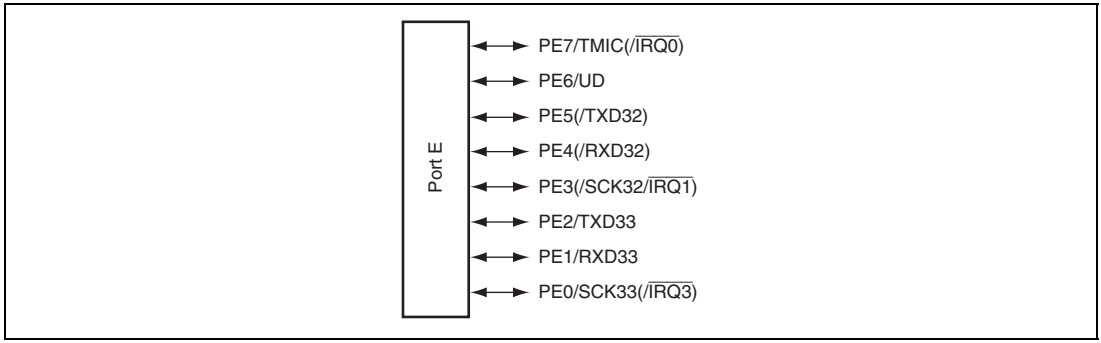


Figure 9.12 Port E Pin Configuration

Port E has the following registers.

- Port data register E (PDRE)
- Port control register E (PCRE)
- Port mode register E (PMRE)

9.12.1 Port Data Register E (PDRE)

PDRE is a register that stores data of port E.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7	0	R/W	If port E is read while PCRE bits are set to 1, the values stored in PDRE are read, regardless of the actual pin states. If port E is read while PCRE bits are cleared to 0, the pin states are read.
6	PE6	0	R/W	
5	PE5	0	R/W	
4	PE4	0	R/W	
3	PE3	0	R/W	
2	PE2	0	R/W	
1	PE1	0	R/W	
0	PE0	0	R/W	

9.12.2 Port Control Register E (PCRE)

PCRE selects inputs/outputs in bit units for pins of port E.

Bit	Bit Name	Initial Value	R/W	Description
7	PCRE7	0	W	Setting a PCRE bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCRE and in PDRE are valid when the corresponding pin is designated as a general I/O pin. PCRE is a write-only register. These bits are always read as 1.
6	PCRE6	0	W	
5	PCRE5	0	W	
4	PCRE4	0	W	
3	PCRE3	0	W	
2	PCRE2	0	W	
1	PCRE1	0	W	
0	PCRE0	0	W	

9.12.3 Port Mode Register E (PMRE)

PMRE controls the selection of the port E pin functions.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
4	TMIC	0	R/W	PE7/TMIC Pin Function Switch 0: PE7 I/O pin 1: TMIC input pin
3	IRQ0	0	R/W	PE7/$\overline{\text{IRQ0}}$ Pin Function Switch 0: PE7 I/O pin 1: $\overline{\text{IRQ0}}$ input pin
2	UD	0	R/W	PE6/UD Pin Function Switch 0: PE6 I/O pin 1: UD input pin
1	IRQ1	0	R/W	PE3/$\overline{\text{IRQ1}}$ Pin Function Switch 0: PE3 I/O pin 1: $\overline{\text{IRQ1}}$ input pin

Bit	Bit Name	Initial Value	R/W	Description
0	IRQ3	0	R/W	PE0/$\overline{\text{IRQ3}}$ Pin Function Switch 0: PE0 I/O pin 1: $\overline{\text{IRQ3}}$ input pin

9.12.4 Pin Functions

The relationship between the register settings and the port functions is shown below.

- PE7/TMIC (/$\overline{\text{IRQ0}}$) pin

Register Name	PMRB	PMRE		PCRE	Pin Function
Bit Name	IRQ0	IRQ0	TMIC	PCRE7	
Setting Value	0	1	x	x	$\overline{\text{IRQ0}}$ input pin
	Other than above		0	0	PE7 input pin
				1	PE7 output pin
			1	x	TMIC input pin

[Legend]
x: Don't care

- PE6/UD pin

Register Name	PMRE	PCRE	Pin Function
Bit Name	UD	PCRE6	
Setting Value	0	0	PE6 input pin
		1	PE6 output pin
	1	x	UD input pin

[Legend]
x: Don't care

- PE5 (/TXD32) pin

Register Name	PFCR	SPCR	PCRE	Pin Function
Bit Name	SC32S	SPC32	PCRE5	
Setting Value	0	x	0	PE5 input pin
			1	PE5 output pin
	1	0	0	PE5 input pin
			1	PE5 output pin
		1	x	TXD32 output pin

[Legend]
x: Don't care

- PE4 (/RXD32) pin

Register Name	PFCR	SCR3_2	PCRE	Pin Function
Bit Name	SC32S	RE	PCRE4	
Setting Value	0	x	0	PE4 input pin
			1	PE4 output pin
	1	0	0	PE4 input pin
			1	PE4 output pin
		1	x	RXD32 input pin

[Legend]
x: Don't care

- PE3 (/SCK32/$\overline{\text{IRQ1}}$) pin

Register Name	PMRB	PMRE	PFCR	SMR3_2	SCR3_2		PCRE	Pin Function
Bit Name	IRQ1	IRQ1	SC32S	COM	CKE1	CKE0	PCRE3	
Setting Value	0	1	x	x	x	x	x	$\overline{\text{IRQ1}}$ input pin
	Other than above		0	x	x	x	0	PE3 input pin
							1	PE3 output pin
			1	0	0	0	0	PE3 input pin
							1	PE3 output pin
						1	x	SCK32 output pin
					1	0		SCK32 input pin
						1		Setting prohibited
				1	0	0		SCK32 output pin
						1		Setting prohibited
					1	0		SCK32 input pin
						1		Setting prohibited

[Legend]
x: Don't care

- PE2/TXD33 pin

Register Name	SPCR2	PCRE	Pin Function
Bit Name	SPC33	PCRE2	
Setting Value	0	0	PE2 input pin
		1	PE2 output pin
	1	x	TXD33 output pin

[Legend]
x: Don't care

- PE1/RXD33 pin

Register Name	SCR3_3	PCRE	Pin Function
Bit Name	RE	PCRE1	
Setting Value	0	0	PE1 input pin
		1	PE1 output pin
	1	x	RXD33 input pin

[Legend]
x: Don't care

- PE0/SCK33 (/$\overline{\text{IRQ3}}$) pin

Register Name	PMRB	PMRE	SMR3_3	SCR3_3		PCRE	Pin Function
Bit Name	IRQ3	IRQ3	COM	CKE1	CKE0	PCRE0	
Setting Value	0	1	x	x	x	x	$\overline{\text{IRQ3}}$ input pin
	Other than above		0	0	0	0	PE0 input pin
						1	PE0 output pin
					1	x	SCK33 output pin
				1	0		SCK33 input pin
					1		Setting prohibited
			1	0	0		SCK33 output pin
					1		Setting prohibited
				1	0		SCK33 input pin
					1		Setting prohibited

[Legend]
x: Don't care

9.13 Port F

Port F is an I/O port: its pins can also be configured to function as an external interrupt input pin, SCI3_1 I/O pins, and timer G input pin. Figure 9.13 shows the pin configuration.

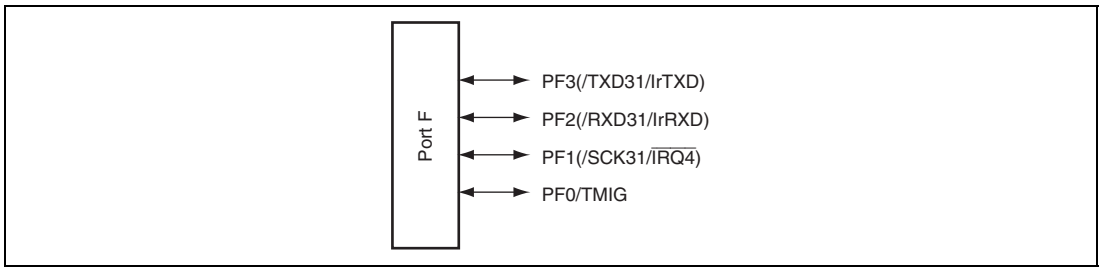


Figure 9.13 Port F Pin Configuration

Port F has the following registers.

- Port data register F (PDRF)
- Port control register F (PCRF)
- Port mode register F (PMRF)

9.13.1 Port Data Register F (PDRF)

PDRF is a register that stores data of port F.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
4	—	—	—	Reserved The write value should always be 0.
3	PF3	0	R/W	If port F is read while PCRF bits are set to 1, the values stored in PDRF are read, regardless of the actual pin states. If port F is read while PCRF bits are cleared to 0, the pin states are read.
2	PF2	0	R/W	
1	PF1	0	R/W	
0	PF0	0	R/W	

9.13.2 Port Control Register F (PCRF)

PCRF selects inputs/outputs in bit units for pins of port F.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
4	—	—	—	Reserved The write value should always be 0.
3	PCRF3	0	W	Setting a PCRF bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCRF and in PDRF are valid when the corresponding pin is designated as a general I/O pin. PCRF is a write-only register. These bits are always read as 1.
2	PCRF2	0	W	
1	PCRF1	0	W	
0	PCRF0	0	W	

9.13.3 Port Mode Register F (PMRF)

PMRF controls the selection of the pin functions on port F.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
2	IRQ4	0	R/W	PF1/$\overline{\text{IRQ4}}$ Pin Function Switch 0: PF1 I/O pin 1: $\overline{\text{IRQ4}}$ input pin
1	NCS	0	R/W	Controls usage of the TMIG noise cancellation circuit. 0: Noise cancellation is disabled 1: Noise cancellation is enabled
0	TMIG	0	R/W	PF0/TMIG Pin Function Switch 0: PF0 I/O pin 1: TMIG input pin

9.13.4 Pin Functions

The relationship between the register settings and the port functions is shown below.

- PF3 (/TXD31/IrTXD) pin

Register Name	PFCR	SPCR	IrCR	PCRF	Pin Function
Bit Name	SC31S	SPC31	IrE	PCRF3	
Setting Value	0	x	x	0	PF3 input pin
				1	PF3 output pin
	1	0		0	PF3 input pin
				1	PF3 output pin
		1	0	x	TXD31 output pin
			1		IrTXD output pin

[Legend]
x: Don't care

- PF2 (/RXD31/IrRXD) pin

Register Name	PFCR	SCR3_1	IrCR	PCRF	Pin Function
Bit Name	SC31S	RE	IrE	PCRF2	
Setting Value	0	x	x	0	PF2 input pin
				1	PF2 output pin
	1	0		0	PF2 input pin
				1	PF2 output pin
		1	0	x	RXD31 input pin
			1		IrRXD input pin

[Legend]
x: Don't care

- PF1 (/SCK31/$\overline{\text{IRQ4}}$) pin

Register Name	PMR9	PMRF	PFCR	SMR3_1	SCR3_1		PCRF	Pin Function
Bit Name	IRQ4	IRQ4	SC31S	COM	CKE1	CKE0	PCRF1	
Setting Value	0	1	x	x	x	x	x	$\overline{\text{IRQ4}}$ input pin
	Other than above		0	x	x	x	0	PF1 input pin
							1	PF1 output pin
			1	0	0	0	0	PF1 input pin
							1	PF1 output pin
						1	x	SCK31 output pin
					1	0		SCK31 input pin
						1		Setting prohibited
				1	0	0		SCK31 output pin
						1		Setting prohibited
					1	0		SCK31 input pin
						1		Setting prohibited

[Legend]
x: Don't care

- PF0/TMIG pin

Register Name	PMRF	PCRF	Pin Function
Bit Name	TMIG	PCRF0	
Setting Value	0	0	PF0 input pin
		1	PF0 output pin
	1	x	TMIG input pin

[Legend]
x: Don't care

9.14 Input/Output Data Inversion

9.14.1 Serial Port Control Register, Serial Port Control Register 2 (SPCR, SPCR2)

SPCR switches input/output data inversion of the RXD (IrRXD) and TXD (IrTXD) pins.

Figure 9.14 shows a input/output data inversion function.

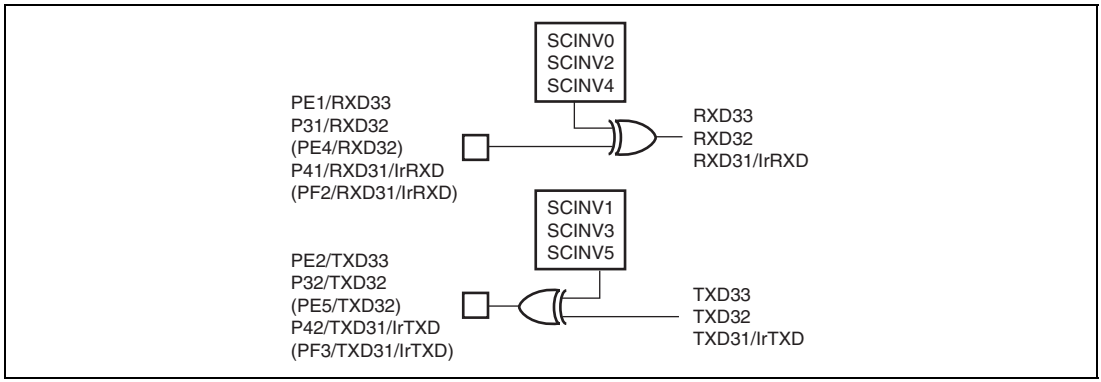


Figure 9.14 Input/Output Data Inversion Function

- SPCR

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved
6	—	1	—	These bits are always read as 1 and cannot be modified.
5	SPC32	0	R/W	P32/TXD32/SCL (PE5/TXD32) Pin Function Switch Selects whether pin P32/TXD32/SCL (PE5/TXD32) is used as P32/SCL (PE5) or as TXD32. 0: P32/ SCL (PE5) I/O pin 1: TXD32 output pin* Note: Set the TE bit in SCR3_2 after having set this bit to 1.
4	SPC31	0	R/W	P42/TXD31/IrTXD/TMOFH (PF3/TXD31/IrTXD) Pin Function Switch Selects whether pin P42/TXD31/IrTXD/TMOFH (PF3/TXD31/IrTXD) is used as P42/TMOFH (PF3) or as TXD31/IrTXD. 0: P42 (PF3) I/O pin or TMOFH output pin 1: TXD31/IrTXD output pin* Note: Set the TE bit in SCR3_1 after having set this bit to 1.

Bit	Bit Name	Initial Value	R/W	Description
3	SCINV3	0	R/W	TXD32 Pin Output Data Inversion Switch Specifies whether the logic level of output data of the TXD32 pin is to be inverted or not. 0: TXD32 output data is not inverted 1: TXD32 output data is inverted
2	SCINV2	0	R/W	RXD32 Pin Input Data Inversion Switch Specifies whether the logic level of input data of the RXD32 pin is to be inverted or not. 0: RXD32 input data is not inverted 1: RXD32 input data is inverted
1	SCINV1	0	R/W	TXD31/IrTXD Pin Output Data Inversion Switch Specifies whether the logic level of output data of the TXD31/IrTXD pin is to be inverted or not. 0: TXD31/IrTXD output data is not inverted 1: TXD31/IrTXD output data is inverted
0	SCINV0	0	R/W	RXD31/IrRXD Pin Input Data Inversion Switch Specifies whether the logic level of input data of the RXD31/IrRXD pin is to be inverted or not. 0: RXD31/IrRXD input data is not inverted 1: RXD31/IrRXD input data is inverted

Note: When the serial port control register is modified, the data being input or output up to that point is inverted immediately after the modification, and an invalid data change is input or output. When modifying the serial port control register, modification must be made in a state in which data changes are invalidated.

- SPCR2

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
4	SPC33	0	R/W	PE2/TXD33 Pin Function Switch Selects whether pin PE2/TXD33 is used as PE2 or as TXD33. 0: PE2 I/O pin 1: TXD33 output pin Set the TE bit in SCR3_3 after having set this bit to 1.
3	—	1	—	Reserved
2	—	1	—	These bits are always read as 1 and cannot be modified.
1	SCINV5	0	R/W	TXD33 Pin Output Data Inversion Switch Specifies whether the logic level of output data of the TXD33 pin is inverted. 0: TXD33 output data is not inverted 1: TXD33 output data is inverted
0	SCINV4	0	R/W	RXD33 Pin Input Data Inversion Switch Specifies whether the logic level of input data of the RXD33 pin is inverted. 0: RXD33 input data is not inverted 1: RXD33 input data is inverted

Note: When the serial port control register 2 is modified, the data being input or output up to that point is inverted immediately after the modification, and an invalid data change is input or output. When modifying the serial port control register 2, modification must be made in a state in which data changes are invalidated.

9.15 Port Function Switch

9.15.1 Port Function Control Register (PFCR)

PFCR controls the assignments of pins for SCI3_1 and SCI3_2 and the functions of other pins.

Bit	Bit Name	Initial Value	R/W	Description
7	CLKOUT1	1	R/W	TMOW/CLKOUT Pin Function Switch
6	CLKOUT0	1	R/W	00: CLKOUT output pin (ϕ_{OSC}) 01: CLKOUT output pin (ϕ_{OSC}/2) 10: CLKOUT output pin (ϕ_{OSC}/4) 11: TMOW output pin
5	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
4	PWM4	0	R/W	P93/PWM4 Pin Function Switch 0: P93 I/O pin 1: PWM4 output pin
3	PWM3	0	R/W	P92/PWM3 Pin Function Switch 0: P92 I/O pin 1: PWM3 output pin
2	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
1	SC32S	0	R/W	SCI3_2 Pin Assignment Select 0: TXD32 is assigned to P32 RXD32 is assigned to P31 SCK32 is assigned to P30 1: TXD32 is assigned to PE5 RXD32 is assigned to PE4 SCK32 is assigned to PE3

Bit	Bit Name	Initial Value	R/W	Description
0	SC31S	0	R/W	SCI3_1 Pin Assignment Select 0: TXD31 is assigned to P42 RXD31 is assigned to P41 SCK31 is assigned to P40 1: TXD31 is assigned to PF3 RXD31 is assigned to PF2 SCK31 is assigned to PF1

9.16 Usage Notes

9.16.1 How to Handle Unused Pin

If an I/O pin not used by the user system is floating, pull it up or down.

- If an unused pin is an input pin, it is recommended to handle it in one of the following ways:
  - Pull it up to Vcc with an on-chip pull-up MOS.
  - Pull it up to Vcc with an external resistor of approximately 100 kΩ.
  - Pull it down to Vss with an external resistor of approximately 100 kΩ.
  - For a pin also used by the A/D converter, pull it up to AVcc. With an external resistor of approximately 100 kΩ.
- If an unused pin is an output pin, it is recommended to handle it in one of the following ways:
  - Set the output of the unused pin to high and pull it up to Vcc with an external resistor of approximately 100 kΩ.
  - Set the output of the unused pin to low and pull it down to GND with an external resistor of approximately 100 kΩ.

9.16.2 Input Characteristics Difference due to Pin Function

When the functions of pins $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, $\overline{\text{IRQ3}}$, $\overline{\text{IRQ4}}$, IRQACE, $\overline{\text{WKP0}}$ to $\overline{\text{WKP7}}$, AEVL, AEVH, TMIC, TMIF, TMIG, SCK31 to SCK33, SDA, and SCL are selected, the corresponding pins have the schmitt-trigger input characteristics, which are different from the ones when they are used as the port input pins.

For example, the input high voltage and the input low voltage of the PB0/AN0/$\overline{\text{IRQ0}}$ pin differ when the pin is used as PB0 input or $\overline{\text{IRQ0}}$ input. For details, refer to tables 25.2 and 25.11.

Section 10 Realtime Clock (RTC)

The realtime clock (RTC) is a timer used to count periods of time ranging from a second to a week. Interrupts can be generated at intervals ranging from 0.25 seconds to a week. Figure 10.1 is a block diagram of the RTC.

10.1 Features

- Counts seconds, minutes, hours, and day-of-week
- Start/stop function
- Reset function
- Readable/writable counter of seconds, minutes, hours, and day-of-week with BCD codes
- Periodic (0.25 seconds, 0.5 seconds, one second, minute, hour, day, and week) interrupts
- 8-bit free running counter
- Selection of clock source
- Module standby mode allows this module to enter standby mode independently when it is not in use (for details, see section 6.4, Module Standby Function).

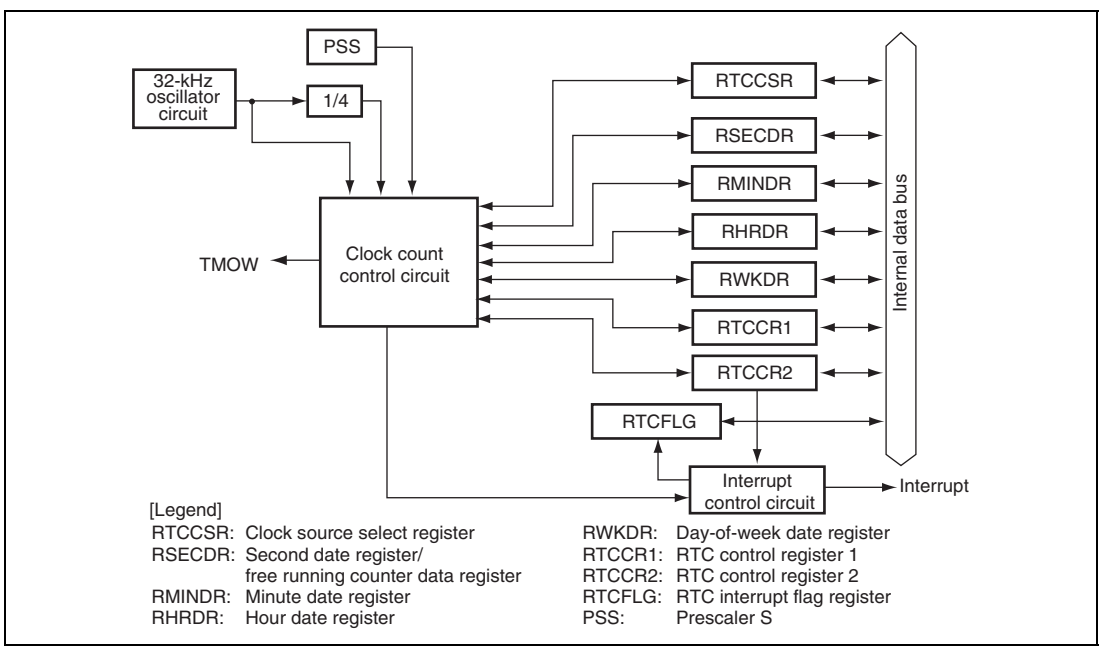


Figure 10.1 Block Diagram of RTC

10.2 Input/Output Pin

Table 10.1 shows the RTC input/output pin.

Table 10.1 Pin Configuration

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

10.3 Register Descriptions

The RTC has the following registers.

- Second data register/free running counter data register (RSECDR)
- Minute data register (RMINDR)
- Hour data register (RHRDR)
- Day-of-week data register (RWKDR)
- RTC control register 1 (RTCCR1)
- RTC control register 2 (RTCCR2)
- Clock source select register (RTCCSR)
- RTC Interrupt flag register (RTCFLG)

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

RSECDR counts the BCD-coded second value. The setting range is decimal 00 to 59. It is an 8-bit read register used as a counter, when it operates as a free running counter. For more information on reading seconds, minutes, hours, and day-of-week, see section 10.4.3, Data Reading Procedure.

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

Note: * This is the initial value after a reset by the RST bit in RTCCR1.

10.3.2 Minute Data Register (RMINDR)

RMINDR counts the BCD-coded minute value on the carry generated once per minute by the RSECDR counting. The setting range is decimal 00 to 59.

Bit	Bit Name	Initial Value	R/W	Description
7	BSY	—/(0)*	R	RTC Busy This bit is set to 1 when the RTC is updating (operating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted.
6	MN12	—/(0)*	R/W	Counting Ten's Position of Minutes
5	MN11	—/(0)*	R/W	Counts on 0 to 5 for 60-minute counting.
4	MN10	—/(0)*	R/W	
3	MN03	—/(0)*	R/W	Counting One's Position of Minutes
2	MN02	—/(0)*	R/W	Counts on 0 to 9 once per minute. When a carry is generated, 1 is added to the ten's position.
1	MN01	—/(0)*	R/W	
0	MN00	—/(0)*	R/W	

Note: * This is the initial value after a reset by the RST bit in RTCCR1.

10.3.3 Hour Data Register (RHRDR)

RHRDR counts the BCD-coded hour value on the carry generated once per hour by RMINDR. The setting range is either decimal 00 to 11 or 00 to 23 by the selection of the 12/24 bit in RTCCR1.

Bit	Bit Name	Initial Value	R/W	Description
7	BSY	—/(0)*	R	RTC Busy This bit is set to 1 when the RTC is updating (operating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted.
6	—	0	—	Reserved This bit is always read as 0.
5	HR11	—/(0)*	R/W	Counting Ten's Position of Hours
4	HR10	—/(0)*	R/W	Counts on 0 to 2 for ten's position of hours.
3	HR03	—/(0)*	R/W	Counting One's Position of Hours
2	HR02	—/(0)*	R/W	Counts on 0 to 9 once per hour. When a carry is generated, 1 is added to the ten's position.
1	HR01	—/(0)*	R/W	
0	HR00	—/(0)*	R/W	

Note: * This is the initial value after a reset by the RST bit in RTCCR1.

10.3.4 Day-of-Week Data Register (RWKDR)

RWKDR counts the BCD-coded day-of-week value on the carry generated once per day by RHRDR. The setting range is decimal 0 to 6 using bits WK2 to WK0.

Bit	Bit Name	Initial Value	R/W	Description
7	BSY	—/(0)*	R	RTC Busy This bit is set to 1 when the RTC is updating (operating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted.
6 to 3	—	All 0	—	Reserved These bits are always read as 0.
2	WK2	—/(0)*	R/W	Day-of-Week Counting
1	WK1	—/(0)*	R/W	Day-of-week is indicated with a binary code
0	WK0	—/(0)*	R/W	000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: Setting prohibited

Note: * This is the initial value after a reset by the RST bit in RTCCR1.

10.3.5 RTC Control Register 1 (RTCCR1)

RTCCR1 controls start/stop and reset of the clock timer. For the definition of time expression, see figure 10.2.

Bit	Bit Name	Initial Value	R/W	Description
7	RUN	—/(0)*	R/W	RTC Operation Start 0: Stops RTC operation 1: Starts RTC operation
6	12/24	—/(0)*	R/W	Operating Mode 0: RTC operates in 12-hour mode. RHRDR counts on 0 to 11. 1: RTC operates in 24-hour mode. RHRDR counts on 0 to 23.
5	PM	—/(0)*	R/W	A.m./P.m. 0: Indicates a.m. when RTC is in the 12-hour mode. 1: Indicates p.m. when RTC is in the 12-hour mode.
4	RST	0	R/W	Reset 0: Normal operation 1: Resets registers and control circuits except RTCCSR and this bit. Clear this bit to 0 after having been set to 1.
3	—	—	R/W	Reserved The write value should be 0.
2 to 0	—	All 0	—	Reserved These bits are always read as 0.

Note: * This is the initial value after a reset by the RST bit in RTCCR1.

Noon

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

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

Figure 10.2 Definition of Time Expression

10.3.6 RTC Control Register 2 (RTCCR2)

RTCCR2 controls RTC periodic interrupts of week, day, hour, minute, one second, 0.5 seconds, and 0.25 seconds. Enabling interrupts of week, day, hour, minute, one second, 0.5 seconds, and 0.25 seconds sets the corresponding flag to 1 in the RTC interrupt flag register (RTCFLG) when an interrupt occurs. It also controls an overflow interrupt of a free running counter when RTC operates as a free running counter.

Bit	Bit Name	Initial Value	R/W	Description
7	FOIE	—/(0)*	R/W	Free Running Counter Overflow Interrupt Enable 0: Disables an overflow interrupt 1: Enables an overflow interrupt
6	WKIE	—/(0)*	R/W	Week Periodic Interrupt Enable 0: Disables a week periodic interrupt 1: Enables a week periodic interrupt
5	DYIE	—/(0)*	R/W	Day Periodic Interrupt Enable 0: Disables a day periodic interrupt 1: Enables a day periodic interrupt
4	HRIE	—/(0)*	R/W	Hour Periodic Interrupt Enable 0: Disables an hour periodic interrupt 1: Enables an hour periodic interrupt
3	MNIE	—/(0)*	R/W	Minute Periodic Interrupt Enable 0: Disables a minute periodic interrupt 1: Enables a minute periodic interrupt
2	1SEIE	—/(0)*	R/W	One-Second Periodic Interrupt Enable 0: Disables a one-second periodic interrupt 1: Enables a one-second periodic interrupt
1	05SEIE	—/(0)*	R/W	0.5-Second Periodic Interrupt Enable 0: Disables a 0.5-second periodic interrupt 1: Enables a 0.5-second periodic interrupt
0	025SEIE	—/(0)*	R/W	0.25-Second Periodic Interrupt Enable 0: Disables a 0.25-second periodic interrupt 1: Enables a 0.25-second periodic interrupt

Note: * This is the initial value after a reset by the RST bit in RTCCR1.

10.3.7 Clock Source Select Register (RTCCSR)

RTCCSR selects clock source. A free running counter controls start/stop of counter operation by the RUN bit in RTCCR1. When a clock other than $\phi_w/4$ is selected, the RTC is disabled and operates as an 8-bit free running counter. When the RTC operates as an 8-bit free running counter, RSECDR enables counter values to be read. An interrupt can be generated by setting 1 to the FOIE bit in RTCCR2 and enabling an overflow interrupt of the free running counter. A clock in which the system clock is divided by 32, 16, 8, or 4 is output in active or sleep mode. The ϕ_w clock is output in active, sleep, subactive, subsleep, and watch modes.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—/(0)*	R	Reserved This bit cannot be modified.
6	RCS6	0	R/W	Clock Output Selection
5	RCS5	0	R/W	Select a clock output from the TMOW pin when setting the TMOW bit in PMR3 to 1.
4	SUB32K	0	R/W	000: $\phi/4$ 010: $\phi/8$ 100: $\phi/16$ 110: $\phi/32$ xx1: ϕ_w
3	RCS3	1	R/W	Clock Source Selection
2	RCS2	0	R/W	0000: $\phi/8$····· Free running counter operation
1	RCS1	0	R/W	0001: $\phi/32$····· Free running counter operation
0	RCS0	0	R/W	0010: $\phi/128$····· Free running counter operation 0011: $\phi/256$····· Free running counter operation 0100: $\phi/512$····· Free running counter operation 0101: $\phi/2048$····· Free running counter operation 0110: $\phi/4096$····· Free running counter operation 0111: $\phi/8192$····· Free running counter operation 1000: $\phi_w/4$····· RTC operation 1001 to 1111: Setting prohibited

[Legend]
x: Don't care.
Note: * This is the initial value after a reset by the RST bit in RTCCR1.

10.3.8 RTC Interrupt Flag Register (RTCFLG)

RTCFLG sets the corresponding flag when an interrupt occurs. Each flag is not cleared automatically even if the interrupt is accepted. To clear the flag, 0 should be written to the flag.

Bit	Bit Name	Initial Value	R/W	Description
7	FOIFG	—/(0)*2	R/W*1	[Setting condition] A free running counter overflows [Clearing condition] 0 is written to FOIFG when FOIFG = 1
6	WKIFG	—/(0)*2	R/W*1	[Setting condition] A week periodic interrupt occurs [Clearing condition] 0 is written to WKIFG when WKIFG = 1
5	DYIFG	—/(0)*2	R/W*1	[Setting condition] A day periodic interrupt occurs [Clearing condition] 0 is written to DYIFG when DYIFG = 1
4	HRIFG	—/(0)*2	R/W*1	[Setting condition] An hour periodic interrupt occurs [Clearing condition] 0 is written to HRIFG when HRIFG = 1
3	MNIFG	—/(0)*2	R/W*1	[Setting condition] A minute periodic interrupt occurs [Clearing condition] 0 is written to MNIFG when MNIFG = 1
2	SEIFG	—/(0)*2	R/W*1	[Setting condition] A one-second periodic interrupt occurs [Clearing condition] 0 is written to SEIFG when SEIFG = 1
1	05SEIFG	—/(0)*2	R/W*1	[Setting condition] A 0.5-second periodic interrupt occurs [Clearing condition] 0 is written to 05SEIFG when 05SEIFG = 1
0	025SEIFG	—/(0)*2	R/W*1	[Setting condition] A 0.25-second periodic interrupt occurs [Clearing condition] 0 is written to 025SEIFG when 025SEIFG = 1

Notes: 1. Only 0 can be written here, to clear the flag.
2. This is the initial value after a reset by the RST bit in RTCCR1.

10.4 Operation

10.4.1 Initial Settings of Registers after Power-On

The RTC registers that store second, minute, hour, and day-of-week data, control registers, and interrupt registers are not initialized by a $\overline{\text{RES}}$ input, or by a reset source caused by a watchdog timer. Therefore, all registers must be set to their initial values after power-on. Once the register settings are made, the RTC provides an accurate time as long as power is supplied regardless of a $\overline{\text{RES}}$ input.

10.4.2 Initial Setting Procedure

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

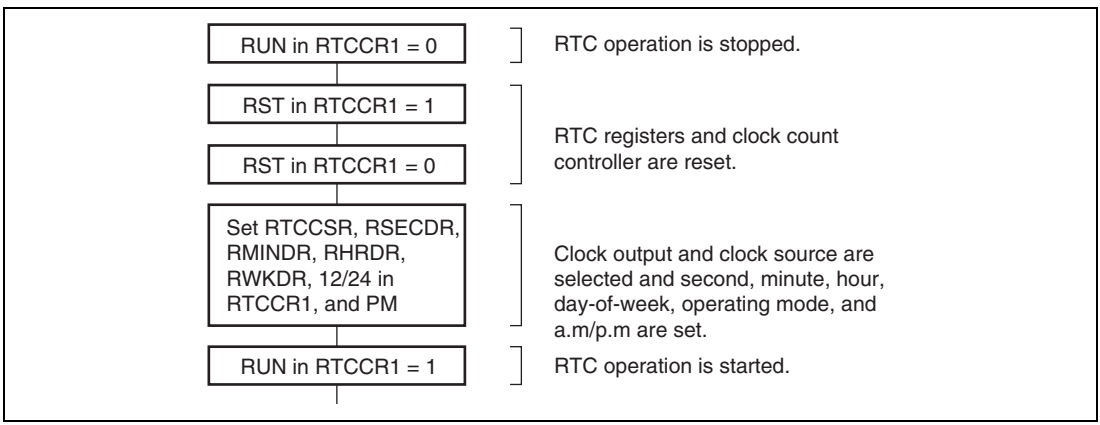


Figure 10.3 Initial Setting Procedure

10.4.3 Data Reading Procedure

When the seconds, minutes, hours, or day-of-week datum is updated while time data is being read, the data obtained may not be correct, and so the time data must be read again. Figure 10.4 shows an example in which correct data is not obtained. In this example, since only RSECDR is read after data update, about 1-minute inconsistency occurs.

To avoid reading in this timing, the following processing must be performed.

1. Check the setting of the BSY bit, and when the BSY bit changes from 1 to 0, read from the second, minute, hour, and day-of-week registers. When about 62.5 ms is passed after the BSY bit is set to 1, the registers are updated, and the BSY bit is cleared to 0.
2. Making use of interrupts, read from the second, minute, hour, and day-of week registers after the corresponding flag of RTCFLG is set to 1 and the BSY bit is confirmed to be 0.
3. Read from the second, minute, hour, and day-of week registers twice in a row, and if there is no change in the read data, the read data is used.

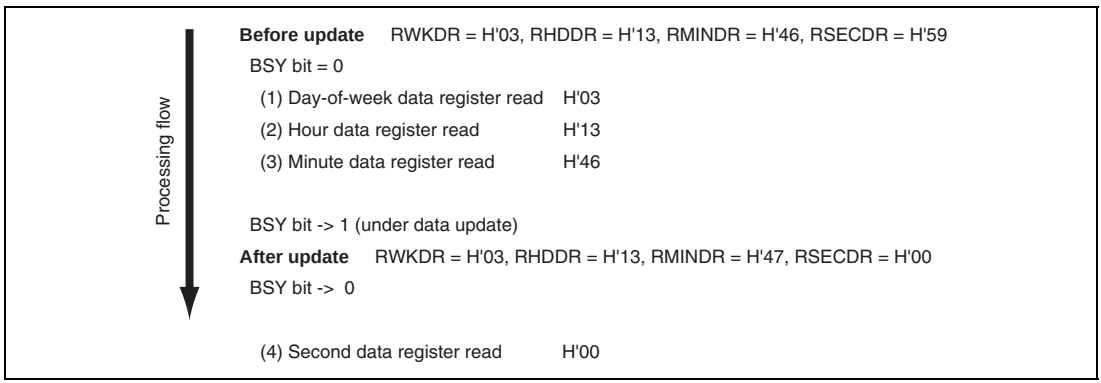


Figure 10.4 Example: Reading of Inaccurate Time Data

10.5 Interrupt Sources

There are eight kinds of RTC interrupts: a free-running counter overflow, week interrupt, day interrupt, hour interrupt, minute interrupt, one-second interrupt, 0.5-second interrupt, and 0.25-second interrupt.

When using an interrupt, set the IENRTC (RTC interrupt request enable) bit in IENR1 to 1 last after other registers are set.

When an interrupt request of the RTC occurs, the corresponding flag in RTCFLG is set to 1. When clearing the flag, write 0.

Table 10.2 shows a interrupt sources.

Table 10.2 Interrupt Sources

Interrupt Name	Interrupt Source	Interrupt Enable Bit
Overflow interrupt	Occurs when the free running counter is overflowed.	FOIE
Week periodic interrupt	Occurs every week when the day-of-week date register value becomes 0.	WKIE
Day periodic interrupt	Occurs every day when the day-of-week date register is counted.	DYIE
Hour periodic interrupt	Occurs every hour when the hour date register is counted.	HRIE
Minute periodic interrupt	Occurs every minute when the minute date register is counted.	MNIE
One-second periodic interrupt	Occurs every second when the one-second date register is counted.	1SEIE
0.5-second periodic interrupt	Occurs every 0.5 seconds.	05SEIE
0.25-second periodic interrupt	Occurs every 0.25 seconds.	025SEIE

10.6 Usage Notes

10.6.1 Note on Clock Count

The subclock must be connected to the 32.768-kHz resonator. When the 38.4-kHz resonator etc. is connected, the correct time count is not possible.

10.6.2 Note when Using RTC Interrupts

The RTC registers are not reset by a $\overline{\text{RES}}$ input, power-on, or overflow of the watchdog timer, and their values are undefined after power-on.

When using RTC interrupts, make sure to initialize the values before setting the IENRTC bit in IENR1 to 1.

Section 11 Timer C

Timer C is an 8-bit timer that increments or decrements each time a clock pulse is input. This timer has two operation modes, interval and auto reload.

11.1 Features

Features of timer C are given below.

- Choice of nine internal clock sources (ϕ/8192, ϕ/2048, ϕ/512, ϕ/64, ϕ/16, ϕ/4, ϕ_w/4, ϕ_w/256, and ϕ_w/1024) or an external clock (can be used to count external events).
- An interrupt is requested when the counter overflows.
- Up/down-counter switching is selected either by the register specification or the external input level specification.
- Subactive mode or subsleep mode operation is possible when ϕ_w/4, ϕ_w/256, or ϕ_w/1024 is selected as the internal clock, or when an external clock is selected.
- Use of module standby mode enables this module to be placed in standby mode independently when not used (for details, see section 6.4, Module Standby Function).

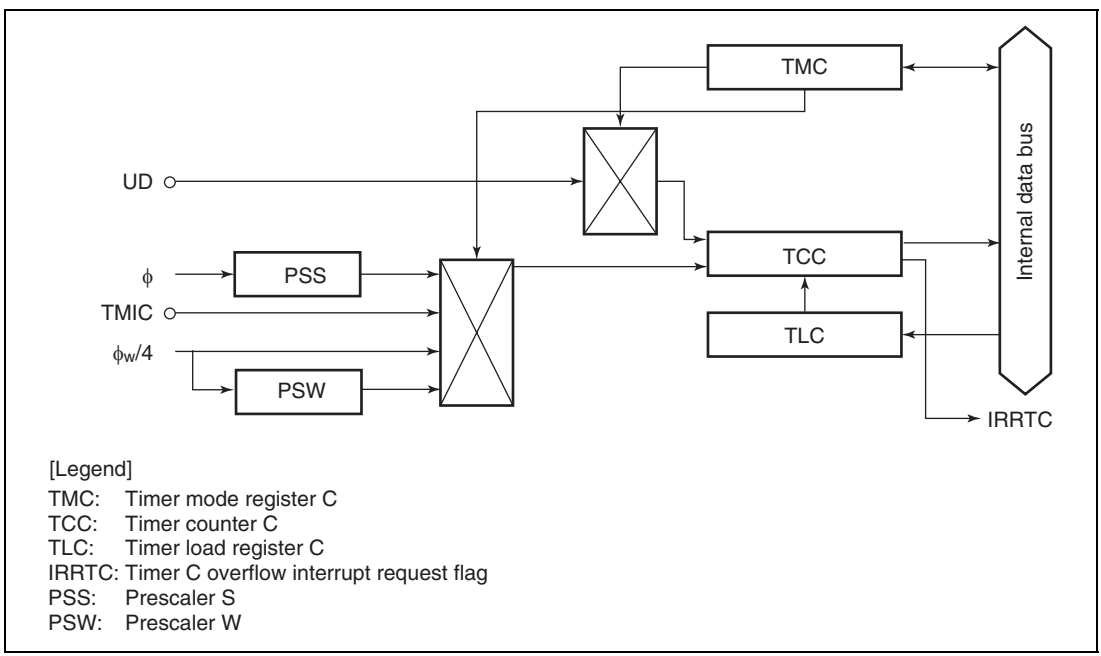


Figure 11.1 Block Diagram of Timer C

11.2 Input/Output Pins

Table 11.1 shows the input/output pins of the timer C.

Table 11.1 Pin Configuration

Name	Abbreviation	I/O	Function
Timer C event input	TMIC	Input	Input pin for event input to TCC
Timer C up/down select	UD	Input	Timer C up/down-count selection

11.3 Register Descriptions

Timer C has the following registers. For details on clock halt register 3 (CKSTPR3), see section 6.1.4, Clock Halt Registers 1 to 3 (CKSTPR1 to CKSTPR3).

- Timer mode register C (TMC)
- Timer counter C (TCC)
- Timer load register (TLC)
- Clock halt register 3 (CKSTPR3)

11.3.1 Timer Mode Register C (TMC)

TMC is an 8-bit read/write register for selecting the auto-reload function and input clock, and performing up/down-counter control.

Upon reset, TMC is initialized to H'10.

Bit	Bit Name	Initial Value	R/W	Description
7	TMC7	0	R/W	Auto-Reload Function Select Selects whether timer C is used as an interval timer or auto-reload timer. 0: Interval timer function 1: Auto-reload function
6	TMC6	0	R/W	Counter Up/Down Control
5	TMC5	0	R/W	Specifies whether TCC functions as an up-counter or down-counter, or whether selection of counting up or down is controlled by the input signal level on the UD pin. 00: TCC is an up-counter 01: TCC is a down-counter 1x: Selection through the signal level on the UD pin UD pin input high: Down-counter UD pin input low: Up-counter
4	—	1	—	Reserved This bit is always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
3	TMC3	0	R/W	Clock Select
2	TMC2	0		TMC3 to TMC0 select the clock input for TCC. For the counting of external events, either the rising or falling edge can be selected.
1	TMC1	0		
0	TMC0	0		
				x000: Internal clock counting on ϕ/8192
				x001: Internal clock counting on ϕ/2048
				x010: Internal clock counting on ϕ/512
				x011: Internal clock counting on ϕ/64
				x100: Internal clock counting on ϕ/16
				0101: Internal clock counting on ϕ/4
				0110: Internal clock counting on ϕ_W/1024
				1101: Internal clock counting on ϕ_W/256
				1110: Internal clock counting on ϕ_W/4
				0111: Counting falling edges of external events (TMIC)*
				1111: Counting rising edges of external events (TMIC)*

[Legend]

x: Don't care

Note: * The TMIC bit in the port mode register E (PMRE) must be set to 1 before the TMC3 to TMC0 bits are set to B'x111.

11.3.2 Timer Counter C (TCC)

TCC is an 8-bit read-only up/down-counter, which is incremented or decremented by internal clock or external event input. The clock source for input to this counter is selected by bits TMC3 to TMC0 in the timer mode register C (TMC). TCC values can be read by the CPU at any time.

When TCC overflows from H'FF to H'00 or to the value set in TLC, or underflows from H'00 to H'FF or to the value set in TLC, the IRRTC bit in IRR2 is set to 1.

TCC is allocated to the same address as TLC.

Upon reset, TCC is initialized to H'00.

11.3.3 Timer Load Register C (TLC)

TLC is an 8-bit write-only register for setting the reload value of timer counter C (TCC).

When a reload value is set in TLC, the same value is loaded into timer counter C as well, and TCC starts counting up/down from that value. When TCC overflows or underflows during operation in auto-reload mode, the TLC value is loaded into TCC. Accordingly, overflow/underflow period can be set within the range of 1 to 256 input clocks.

The same address is allocated to TLC as to TCC.

Upon reset, TLC is initialized to H'00.

11.3.4 Clock Halt Register 3 (CKSTPR3)

For details on placing timer C in and taking it out of standby mode (this is controlled by the TCCKSTP bit in CKSTPR3) see section 6.1.4, Clock Halt Registers 1 to 3 (CKSTPR1 to CKSTPR3).

11.4 Timer Operation

11.4.1 Interval Timer Operation

When bit TMC7 in timer mode register C (TMC) is cleared to 0, timer C functions as an 8-bit interval timer.

Upon reset, TCC is initialized to H'00 and TMC to H'10, so TCC continues up-counting as an interval up-counter without halting immediately after a reset. The timer C operating clock is selected from nine internal clock signals output by prescalers S and W, or an external clock input at pin TMIC. The selection is made by bits TMC3 to TMC0 in TMC.

TCC up/down-count control can be specified by bits TMC6 and TMC5 in TMC, or selected by the input signal level on the UD pin.

After the count value in TCC reaches H'FF (H'00), the next clock input causes timer C to overflow (underflow), setting bit IRRTC in IRR2 to 1. If IENTC = 1 in interrupt enable register 2 (IENR2), a CPU interrupt is requested.

At overflow (underflow), TCC returns to H'00 (H'FF) and starts counting up (down) again.

During interval timer operation (TMC7 = 0), when a value is set in timer load register C (TLC), the same value is set in TCC.

Note: For details on interrupts, see section 4, Interrupt Controller.

11.4.2 Auto-Reload Timer Operation

Setting bit TMC7 in TMC to 1 causes timer C to function as an 8-bit auto-reload timer. When a reload value is set in TLC, the same value is loaded into TCC, becoming the value from which TCC starts its count.

After the count value in TCC reaches H'FF (H'00), the next clock signal input causes timer C to overflow/underflow. The TLC value is then loaded into TCC, and the count continues from that value. The overflow/underflow period can be set within a range from 1 to 256 input clocks, depending on the TLC value.

The clock sources, up/down control, and interrupts in auto-reload mode are the same as in interval mode.

In auto-reload mode (TMC7 = 1), when a new value is set in TLC, the TLC value is also set in TCC.

11.4.3 Event Counter Operation

Timer C can operate as an event counter, with the TMIC pin as the event input pin. External event counting is selected by setting bits TMC3 to TMC0 in the timer mode register C (TMC) to B'0111 or B'1111, and setting the TMIC bit in PMRE to 1. TCC counts up/down at the rising/falling edge of an external event signal input at the TMIC pin.

The external event input signal is not counted correctly if it does not satisfy the high width or low width of the input pin.

11.4.4 TCC Up/Down Control by the External Input Pin

With timer C, TCC up/down control can be performed by UD pin input. When bit TMC6 in TMC is set to 1, TCC functions as an up-counter when UD pin input is low, and as a down-counter when high.

When using UD pin input, set the UD bit in PMRE to 1.

11.5 Timer C Operation States

Table 11.2 summarizes the timer C operation states.

Table 11.2 Timer C Operation States

Operation Mode		Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby	Module Standby
TCC	Interval	Reset	Functioning*1	Functioning*1	Functioning/ Halted*2	Functioning/ Halted*3	Functioning/ Halted*3	Halted	Halted
	Auto reload	Reset	Functioning*1	Functioning*1	Functioning/ Halted*2	Functioning/ Halted*3	Functioning/ Halted*3	Halted	Halted
TMC		Reset	Functioning	Retained	Retained	Functioning	Retained	Retained	Retained

Notes:
1. When $\phi_W/4$, $\phi_W/256$, or $\phi_W/1024$ is selected as the TCC internal clock in active mode or sleep mode, since the system clock and internal clock are mutually asynchronous, synchronization is maintained by a synchronization circuit. This results in a maximum count cycle error of $1/\phi$ (s).
2. When the counter is operated in watch mode, select $\phi_W/4$, $\phi_W/256$, or $\phi_W/1024$ as the clock.
3. When the counter is operated in subactive mode or subsleep mode, either select $\phi_W/4$, $\phi_W/256$, or $\phi_W/1024$ as the internal clock or select an external clock. The counter will not operate on any other internal clock. If $\phi_W/4$ is selected as the internal clock for the counter when $\phi_W/8$ has been selected as subclock ϕ_{SUB}, the lower 2 bits of the counter operate on the same cycle, and the operation of the least significant bit is unrelated to the operation of the counter.

Section 12 Timer F

The timer F is a 16-bit timer having an output compare function. The timer F also provides for external event counting, and counter resetting, interrupt request generation, toggle output, etc., using compare match signals. Thus, it can be applied to various systems. The timer F can also be used as two independent 8-bit timers (timer FH and timer FL). Figure 12.1 shows a block diagram of the timer F.

12.1 Features

- Choice of five counter input clocks
  Internal clocks (ϕ/32, ϕ/16, ϕ/4, and ϕ_w/4) or external clocks can be selected.
- Toggle output function
  Toggle output is performed to the TMOFH or TMOFL pin using a compare match signal.
  The initial value of toggle output can be set.
- Counter resetting by a compare match signal
- Two interrupt sources: One compare match, one overflow
- Choice of 16-bit or 8-bit mode by settings of bits CKSH2 to CKSH0 in TCRF
- Can operate in watch mode, subactive mode, and subsleep mode
  When ϕ_w/4 is selected as an internal clock, the timer F can operate in watch mode, subactive mode, and subsleep mode.
- Module standby mode enables this module to be placed in standby mode independently when it is not in use (for details, refer to section 6.4, Module Standby Function).

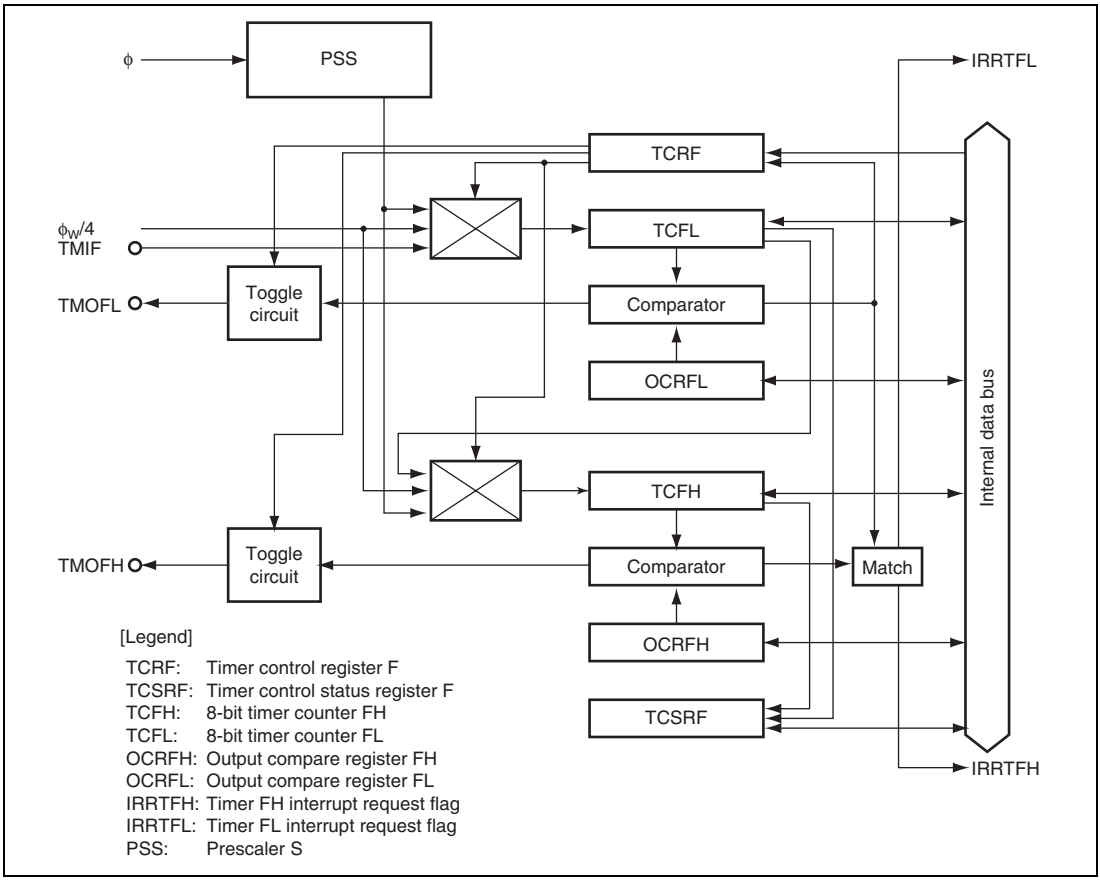


Figure 12.1 Block Diagram of Timer F

12.2 Input/Output Pins

Table 12.1 shows the input/output pins of the timer F.

Table 12.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Timer F event input	TMIF	Input	Event input pin to TCFL
Timer FH output	TMOFH	Output	Timer FH toggle output pin
Timer FL output	TMOFL	Output	Timer FL toggle output pin

12.3 Register Descriptions

The timer F has the following registers.

- Timer counters FH and FL (TCFH, TCFL)
- Output compare registers FH and FL (OCRFH, OCRFL)
- Timer control register F (TCRF)
- Timer control/status register F (TCSRF)

12.3.1 Timer Counters FH and FL (TCFH, TCFL)

TCF is a 16-bit read/write up-counter configured by cascaded connection of 8-bit timer counters TCFH and TCFL. In addition to the use of TCF as a 16-bit counter with TCFH as the upper 8 bits and TCFL as the lower 8 bits, TCFH and TCFL can also be used as independent 8-bit counters.

TCFH and TCFL are initialized to H'00 upon a reset.

(1) 16-Bit Mode (TCF)

When CKSH2 is cleared to 0 in TCRF, TCF operates as a 16-bit counter. The TCF input clock is selected by bits CKSL2 to CKSL0 in TCRF.

TCF can be cleared in the event of a compare match by means of CCLRH in TCSRF.

When TCF overflows from H'FFFF to H'0000, OVFH is set to 1 in TCSRF. If OVIEH in TCSRF is 1 at this time, IRRTFH is set to 1 in IRR2, and if IENTFH in IENR2 is 1, an interrupt request is sent to the CPU.

(2) 8-Bit Mode (TCFH/TCFL)

When CKSH2 is set to 1 in TCRF, TCFH and TCFL operate as two independent 8-bit counters. The TCFH (TCFL) input clock is selected by bits CKSH2 to CKSH0 (CKSL2 to CKSL0) in TCRF.

TCFH (TCFL) can be cleared in the event of a compare match by means of CCLRH (CCLRL) in TCSRF.

When TCFH (TCFL) overflows from H'FF to H'00, OVFH (OVFL) is set to 1 in TCSRF. If OVIEH (OVIEL) in TCSRF is 1 at this time, IRRTFH (IRRTFL) is set to 1 in IRR2, and if IENTFH (IENTFL) in IENR2 is 1, an interrupt request is sent to the CPU.

12.3.2 Output Compare Registers FH and FL (OCRFH, OCRFL)

OCRF is a 16-bit read/write register composed of the two registers OCRFH and OCRFL. In addition to the use of OCRF as a 16-bit register with OCRFH as the upper 8 bits and OCRFL as the lower 8 bits, OCRFH and OCRFL can also be used as independent 8-bit registers.

(1) 16-Bit Mode (OCRF)

When CKSH2 is cleared to 0 in TCRF, OCRF operates as a 16-bit register. OCRF contents are constantly compared with TCF, and when both values match, CMFH is set to 1 in TCSRF. At the same time, IRRTFH is set to 1 in IRR2. If IENTFH in IENR2 is 1 at this time, an interrupt request is sent to the CPU.

Toggle output can be provided from the TMOFH pin by means of compare matches, and the output level can be set by means of the TOLH bit in TCRF.

(2) 8-Bit Mode (OCRFH/OCRFL)

When CKSH2 is set to 1 in TCRF, OCRFH and OCRFL operate as two independent 8-bit registers. OCRFH contents are compared with TCFH, and OCRFL contents are with TCFL. When the OCRFH (OCRFL) and TCFH (TCFL) values match, CMFH (CMFL) is set to 1 in TCSRF. At the same time, IRRTFH (IRRTFL) is set to 1 in IRR2. If IENTFH (IENTFL) in IENR2 is 1 at this time, an interrupt request is sent to the CPU.

Toggle output can be provided from the TMOFH pin (TMOFL pin) by means of compare matches, and the output level can be set by means of the TOLH (TOLL) bit in TCRF.

12.3.3 Timer Control Register F (TCRF)

TCRF switches between 16-bit mode and 8-bit mode, selects the counter input clock, and selects the output level of the TMOFH and TMOFL pins.

Bit	Bit Name	Initial Value	R/W	Description
7	TOLH	0	W	Toggle Output Level H Sets the TMOFH pin output level. 0: Low level 1: High level
6	CKSH2	0	W	Clock Select H
5	CKSH1	0	W	Select the clock input to TCFH from among four internal clock sources or TCFL overflow.
4	CKSH0	0	W	000: 16-bit mode, counting on TCFL overflow signal 001: 16-bit mode, counting on TCFL overflow signal 010: 16-bit mode, counting on TCFL overflow signal 011: Using prohibited 100: 8-bit mode, counting on ϕ/32 101: 8-bit mode, counting on ϕ/16 110: 8-bit mode, counting on ϕ/4 111: 8-bit mode, counting on ϕ_w/4
3	TOLL	0	W	Toggle Output Level L Sets the TMOFL pin output level. 0: Low level 1: High level

Bit	Bit Name	Initial Value	R/W	Description
2	CKSL2	0	W	Clock Select L
1	CKSL1	0	W	Select the clock input to TCFL from among four internal clock sources or external event input.
0	CKSL0	0	W	000: Counting on a rising or falling edge of an external event (on the TMIF pin)*
				001: Counting on a rising or falling edge of an external event (on the TMIF pin)*
				010: Counting on a rising or falling edge of an external event (on the TMIF pin)*
				011: Using prohibited
				100: Internal clock: counting on ϕ/32
				101: Internal clock: counting on ϕ/16
				110: Internal clock: counting on ϕ/4
				111: Internal clock: counting on ϕ_w/4

Note: * The TMIFEG bit in IEGR selects which edge of an external event is used for counting.

12.3.4 Timer Control/Status Register F (TCSRF)

TCSRF performs counter clear selection, overflow flag setting, and compare match flag setting, and controls enabling of overflow interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	OVFH	0	R/(W)*	Timer Overflow Flag H
				[Setting condition]
				TCFH overflows from H'FF to H'00
				[Clearing condition]
				Writing of 0 to bit OVFH after reading OVFH = 1
6	CMFH	0	R/(W)*	Compare Match Flag H
				This is a status flag indicating that TCFH has matched OCRFH.
				[Setting condition]
				The TCFH value matches the OCRFH value
				[Clearing condition]
				Writing of 0 to bit CMFH after reading CMFH = 1

Bit	Bit Name	Initial Value	R/W	Description
5	OVIEH	0	R/W	Timer Overflow Interrupt Enable H Selects enabling or disabling of interrupt generation when TCFH overflows. 0: TCFH overflow interrupt request is disabled 1: TCFH overflow interrupt request is enabled
4	CCLRH	0	R/W	Counter Clear H In 16-bit mode, this bit selects whether TCF is cleared when TCF and OCRF match. In 8-bit mode, this bit selects whether TCFH is cleared when TCFH and OCRFH match. In 16-bit mode: 0: TCF clearing by compare match is disabled 1: TCF clearing by compare match is enabled In 8-bit mode: 0: TCFH clearing by compare match is disabled 1: TCFH clearing by compare match is enabled
3	OVFL	0	R/(W)*	Timer Overflow Flag L This is a status flag indicating that TCFL has overflowed. [Setting condition] TCFL overflows from H'FF to H'00 [Clearing condition] Writing of 0 to bit OVFL after reading OVFL = 1
2	CMFL	0	R/(W)*	Compare Match Flag L This is a status flag indicating that TCFL has matched OCRFL. [Setting condition] The TCFL value matches the OCRFL value [Clearing condition] Writing of 0 to bit CMFL after reading CMFL = 1

Bit	Bit Name	Initial Value	R/W	Description
1	OVIEL	0	R/W	Timer Overflow Interrupt Enable L Selects enabling or disabling of interrupt generation when TCFL overflows. 0: TCFL overflow interrupt request is disabled 1: TCFL overflow interrupt request is enabled
0	CCLRL	0	R/W	Counter Clear L Selects whether TCFL is cleared when TCFL and OCRFL match. 0: TCFL clearing by compare match is disabled 1: TCFL clearing by compare match is enabled

Note: * Only 0 can be written to clear the flag.

12.4 Operation

The timer F is a 16-bit counter that increments on each input clock pulse. The timer F value is constantly compared with the value set in the output compare register F, and the counter can be cleared, an interrupt requested, or port output toggled, when the two values match. The timer F can also be used as two independent 8-bit timers.

12.4.1 Timer F Operation

The timer F has two operating modes, 16-bit timer mode and 8-bit timer mode. The operation in each of these modes is described below.

(1) Operation in 16-Bit Timer Mode

When the CKSH2 bit is cleared to 0 in TCRF, the timer F operates as a 16-bit timer.

Following a reset, TCF is initialized to H'0000, OCRF to H'FFFF, and TCRF and TCSRF to H'00. The counter is incremented by an input signal from an external event (TMIF pin). The TMIFEG bit in IEGR selects which edge of an external event is used for counting.

The timer F counter input clock can be selected from internal clocks or external events according to settings of bits CKSL2 to CKSL0 in TCRF.

OCRF contents are constantly compared with TCF, and when both values satisfy the compare match condition, CMFH is set to 1 in TCSRF. If IENTFH in IENR2 is 1 at this time, an interrupt request is sent to the CPU, and at the same time, TMOFH pin output is toggled. If CCLRH in TCSRF is 1, TCF is cleared. The output level of the TMOFH pin can be set by the TOLH bit in TCRF.

When TCF overflows from H'FFFF to H'0000, OVFH is set to 1 in TCSRF. If OVIEH in TCSRF and IENTFH in IENR2 are both 1, an interrupt request is sent to the CPU.

(2) Operation in 8-Bit Timer Mode

When CKSH2 is set to 1 in TCRF, TCF operates as two independent 8-bit timers, TCFH and TCFL. The TCFH/TCFL input clock is selected by CKSH2 to CKSH0/CKSL2 to CKSL0 in TCRF.

When the OCRFH/OCRFL and TCFH/TCFL values match, CMFH/CMFL is set to 1 in TCSRF. If IENTFH/IENTFL in IENR2 is 1, an interrupt request is sent to the CPU, and at the same time, TMOFH pin/TMOFL pin output is toggled. If CCLRH/CCLRL in TCSRF is 1, TCFH/TCFL is cleared. The output level of the TMOFH pin/TMOFL pin can be set by TOLH/TOLL in TCRF.

When TCFH/TCFL overflows from H'FF to H'00, OVFH/OVFL is set to 1 in TCSRF. If OVIEH/OVIEL in TCSRF and IENTFH/IENTFL in IENR2 are both 1, an interrupt request is sent to the CPU.

12.4.2 TCF Increment Timing

(1) Internal Clock Operation

TCF is incremented by internal clock or external event input. Bits CKSH2 to CKSH0 or CKSL2 to CKSL0 in TCRF select one of internal clock sources (ϕ/32, ϕ/16, ϕ/4, or ϕ_w/4) created by dividing the system clock (ϕ or ϕ_w).

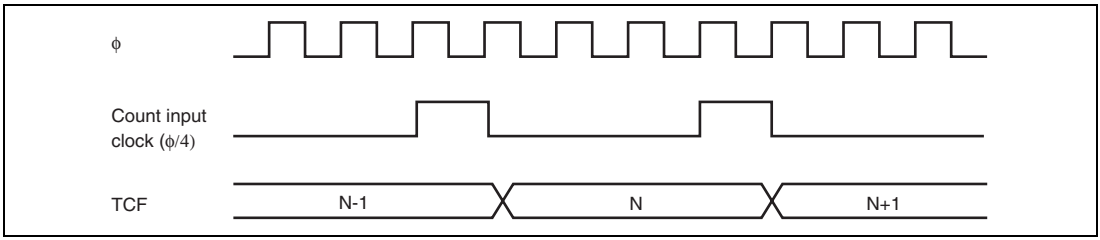


Figure 12.2 Count Timing for Internal Clock Operation

(2) External Event Operation

When the CKSL2 bit in TCRF is cleared to 0, external event input is selected. The counter is incremented at both rising and falling edges of external events. The TMIFEG bit in IEGR selects which edge of an external event is used for counting. The external event pulse width requires clock time longer than 2 system clocks (ϕ), or 2 subclocks (ϕ_{SUB}), depending on the operating mode. Note that an external event does not operate correctly with the lower pulse width.

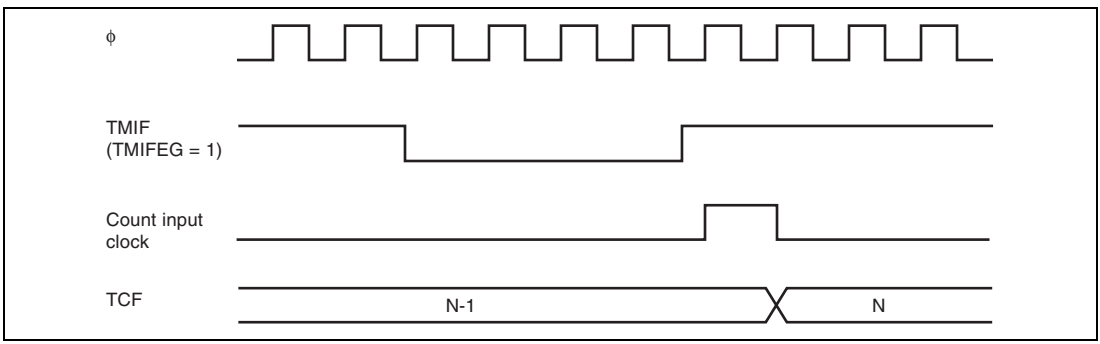


Figure 12.3 Count Timing for External Event Operation

12.4.3 TMOFH/TMOFL Output Timing

In TMOFH/TMOFL output, the value set in TOLH/TOLL in TCRF is output. The output is toggled by the occurrence of a compare match.

Figure 12.4 shows the output timing.

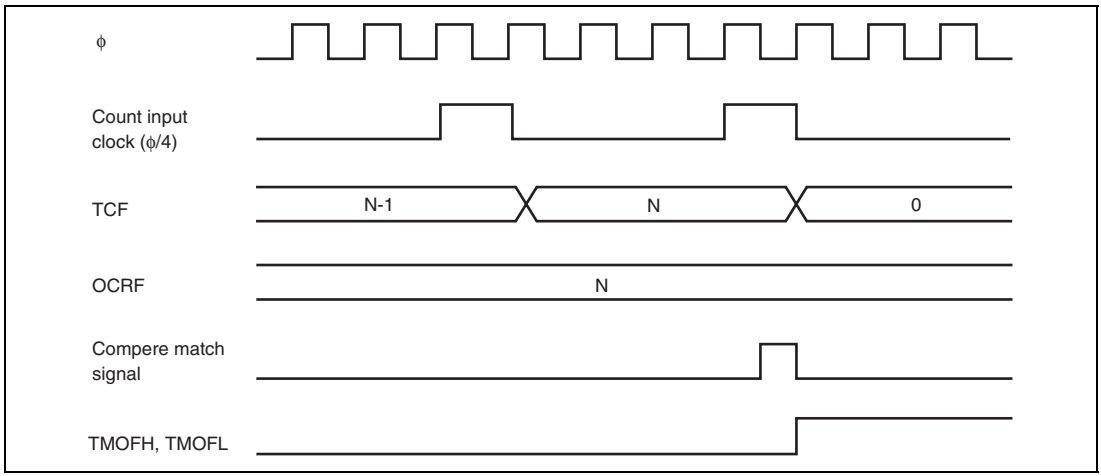


Figure 12.4 TMOFH/TMOFL Output Timing

12.4.4 TCF Clear Timing

TCF can be cleared by a compare match with OCRF.

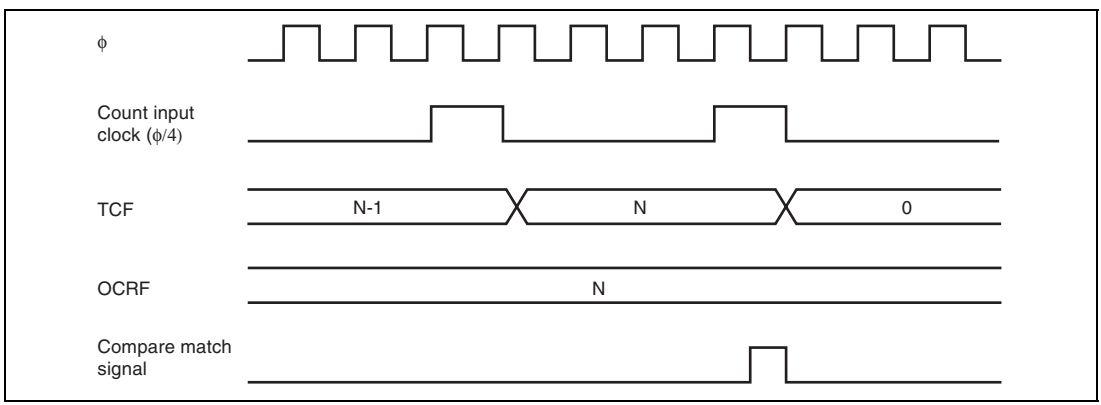


Figure 12.5 TCF Clear Timing

12.4.5 Timer Overflow Flag (OVF) Set Timing

OVF is set to 1 when TCF overflows from H'FFFF to H'0000.

12.4.6 Compare Match Flag Set Timing

The compare match flag (CMFH or CMFL) is set to 1 when the TCF and OCRF values match. The compare match signal is generated in the last state during which the values match (when TCF is updated from the matching value to a new value). When TCF matches OCRF, the compare match signal is not generated until the next counter clock.

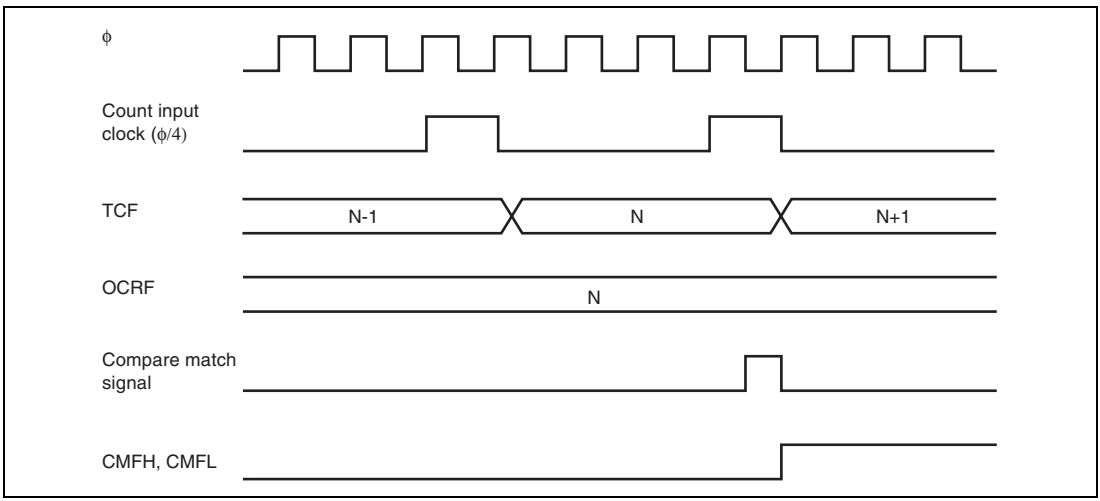


Figure 12.6 Compare Match Flag Set Timing

12.5 Timer F Operating States

The timer F operating states are shown in table 12.2.

Table 12.2 Timer F Operating States

Operating Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby	Module Standby
TCF	Reset	Functioning*1	Functioning*1	Functioning/ Halted*2	Functioning/ Halted*3	Functioning/ Halted*3	Halted	Halted
OCRF	Reset	Functioning	Retained	Retained	Functioning	Retained	Retained	Retained
TCRF	Reset	Functioning	Retained	Retained	Functioning	Retained	Retained	Retained
TCSRF	Reset	Functioning	Retained	Retained	Functioning	Retained	Retained	Retained

Note: 1. When $\phi_W/4$ is selected as the TCF input clock in active mode or sleep mode, since the system clock and internal clock are mutually asynchronous, synchronization is maintained by a synchronization circuit. This results in a maximum count cycle error of $1/\phi$ (s).

2. When the counter is operated in watch mode, $\phi_W/4$ must be selected as the internal clock. The counter will not operate if any other internal clock is selected.

3. When the counter is operated in subactive mode or subsleep mode, either an external clock or internal clock ϕ_W /4 must be selected as the input clock. The counter will not operate if any other internal clock is selected.

12.6 Usage Notes

The following types of contention and operation can occur when the timer F is used.

12.6.1 16-Bit Timer Mode

In toggle output, TMOFH pin output is toggled when all 16 bits match and a compare match signal is generated. If a TCRF write by a MOV instruction and generation of the compare match signal occur simultaneously, TOLH data is output to the TMOFH pin as a result of the TCRF write. TMOFL pin output is unstable in 16-bit mode, and should not be used; the TMOFL pin should be used as a port pin.

If an OCRFL write and compare match signal generation occur simultaneously, the compare match signal is invalid. However, if the written data and the counter value match, there is a probability of a compare match signal being generated and not being generated. As the compare match signal is output in synchronization with the TCFL clock, a compare match will not result in compare match signal generation if the clock is stopped.

Compare match flag CMFH is set when all 16 bits match and a compare match signal is generated. Compare match flag CMFL is set if the setting conditions for the lower 8 bits are satisfied.

When TCF overflows, OVFH is set. OVFL is set if the setting conditions are satisfied when the lower 8 bits overflow. If a TCFL write and overflow signal output occur simultaneously, the overflow signal is not output.

12.6.2 8-Bit Timer Mode

(1) TCFH, OCRFH

In toggle output, TMOFH pin output is toggled when a compare match occurs. If a TCRF write by a MOV instruction and generation of the compare match signal occur simultaneously, TOLH data is output to the TMOFH pin as a result of the TCRF write.

If an OCRFH write and compare match signal generation occur simultaneously, the compare match signal is invalid. However, even if the written data and the counter value match, there is a probability of a compare match signal being generated and not being generated. The compare match signal is output in synchronization with the TCFH clock.

If a TCFH write and overflow signal output occur simultaneously, the overflow signal is not output.

(2) TCFL, OCRFL

In toggle output, TMOFL pin output is toggled when a compare match occurs. If a TCRF write by a MOV instruction and generation of the compare match signal occur simultaneously, TOLL data is output to the TMOFL pin as a result of the TCRF write.

If an OCRFL write and compare match signal generation occur simultaneously, the compare match signal is invalid. However, even if the written data and the counter value match, there is a probability of a compare match signal being generated and not being generated. As the compare match signal is output in synchronization with the TCFL clock, a compare match will not result in compare match signal generation if the clock is stopped.

If a TCFL write and overflow signal output occur simultaneously, the overflow signal is not output.

12.6.3 Flag Clearing

When the internal clock $\phi_w/4$ is selected as the counter input clock, "Interrupt source generation signal" will be operated with ϕ_w and the signal will be outputted with ϕ_w width. And, "Overflow signal" and "Compare match signal" are controlled with 2 cycles of ϕ_w signals. Those signals are output with 2-cycle width of ϕ_w (figure 12.7)

In active (high-speed, medium-speed) mode, even if you cleared interrupt request flag during the term of validity of "Interrupt source generation signal", same interrupt request flag is set. (1 in figure 12.7) And, the timer overflow flag and compare match flag cannot be cleared during the term of validity of "Overflow signal" and "Compare match signal".

For interrupt request flag is set right after interrupt request is cleared, interrupt process to one time timer FH, timer FL interrupt might be repeated. (2 in figure 12.7) Therefore, to definitely clear interrupt request flag in active (high-speed, medium-speed) mode, clear should be processed after the time that calculated with below (1) formula. And, to definitely clear timer overflow flag and compare match flag, clear should be processed after read timer control status register F (TCSRF) after the time that calculated with below (1) formula.

For ST of (1) formula, please substitute the longest number of execution states in used instruction.

In subactive mode, there are not limitation for interrupt request flag, timer overflow flag, and compare match flag clear.

The term of validity of "Interrupt source generation signal"

= 1 cycle of ϕ_w + waiting time for completion of executing instruction
+ interrupt time synchronized with ϕ
= $1/\phi_w + ST \times (1/\phi) + (2/\phi)$ (second).....(1)

ST: Executing number of execution states

Method 1 is recommended to operate for time efficiency.

Method 1

1. Prohibit interrupt in interrupt handling routine (set IENFH, IENFL to 0).
2. After program process returned normal handling, clear interrupt request flags (IRRTFH, IRRTFL) after more than that calculated with (1) formula.
3. After reading the timer control status register F (TCSRF), clear the timer overflow flags (OVFH, OVFL) and compare match flags (CMFH, CMFL).
4. Enable interrupts (set IENFH, IENFL to 1).

Method 2

1. Set interrupt handling routine time to more than time that calculated with (1) formula.
2. Clear interrupt request flags (IRRTFH, IRRTFL) at the end of interrupt handling routine.
3. After read timer control status register F (TCSRF), clear timer overflow flags (OVFH, OVFL) and compare match flags (CMFH, CMFL).

All above attentions are also applied in 16-bit mode and 8-bit mode.

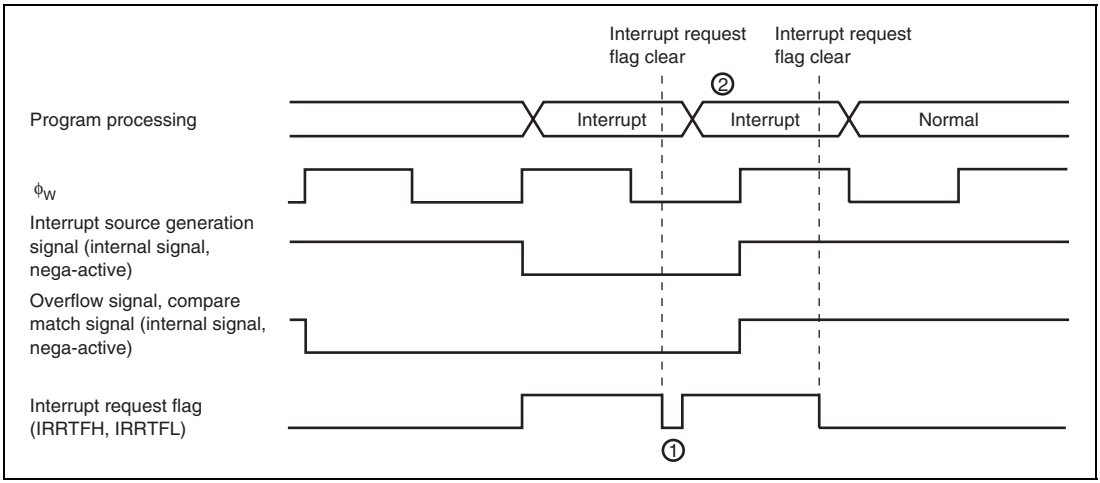


Figure 12.7 Clear Interrupt Request Flag when Interrupt Source Generation Signal is Valid

12.6.4 Timer Counter (TCF) Read/Write

When the internal clock $\phi_w/4$ is selected as the counter input clock in active (high-speed, medium-speed) mode, normal write is not performed on TCF. And when reading TCF, as the system clock and internal clock are mutually asynchronous, TCF synchronizes with synchronization circuit. This results in a maximum TCF read value error of ±1.

In subactive mode, even if ϕ_w /4 is selected as the input clock, TCF can be read from or written to normally.

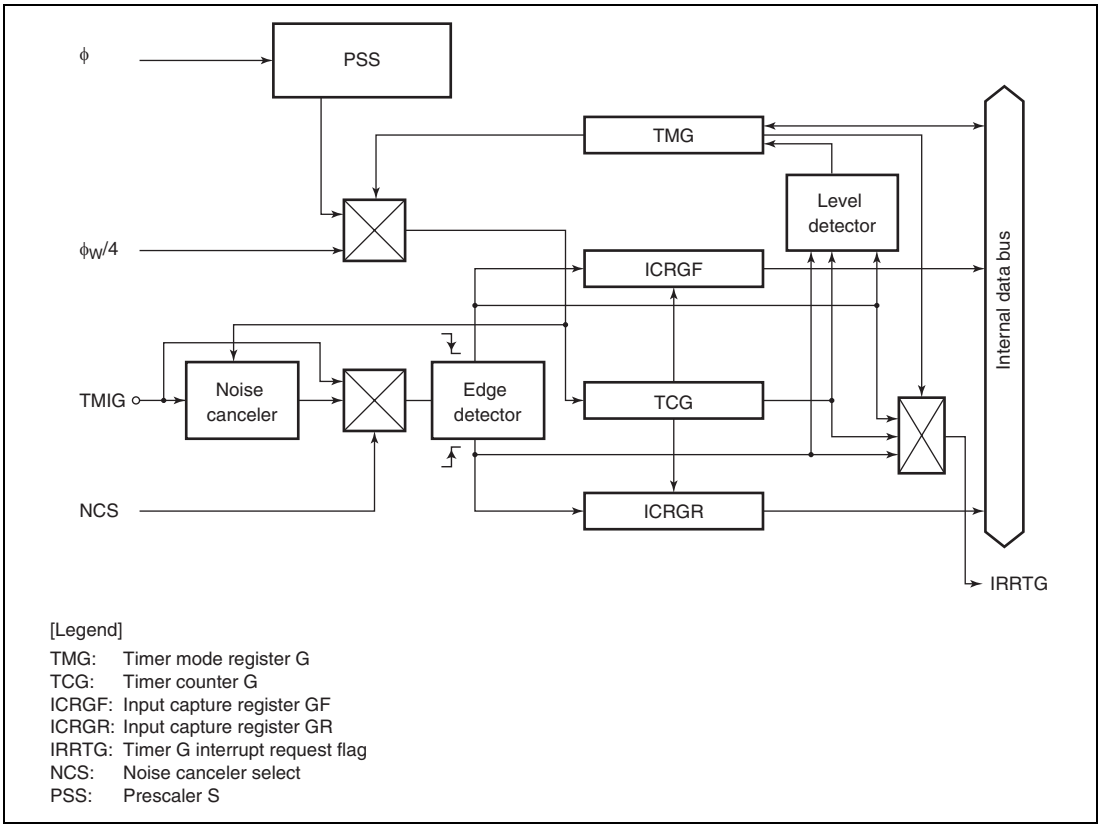
Section 13 Timer G

Timer G is an 8-bit timer with dedicated input capture functions for the rising/falling edges of pulses input from the input capture input pin (input capture input signal). High-frequency component noise in the input capture input signal can be eliminated by a noise canceller, enabling accurate measurement of the input capture input signal duty cycle. If input capture input is not set, timer G functions as an 8-bit interval timer.

13.1 Features

- Choice of four internal clock sources (ϕ/64, ϕ/32, ϕ/2, ϕ_w/4)
- Dedicated input capture functions for rising and falling edges
- Level detection at counter overflow

  It is possible to detect whether overflow occurred when the input capture input signal was high or when it was low.
- Selection of whether or not the counter value is to be cleared at the input capture input signal rising edge, falling edge, or both edges
- Two interrupt sources: one input capture, one overflow. The input capture input signal rising or falling edge can be selected as the interrupt source.
- A built-in noise canceller eliminates high-frequency component noise in the input capture input signal.
- Watch mode, subactive mode, or subsleep mode operation is possible when ϕ_w/4 is selected as the internal clock.
- Module standby mode enables this module to be placed in standby mode independently when it is not in use (for details, see section 6.4, Module Standby Function).



[Legend]
TMG: Timer mode register G
TCG: Timer counter G
ICRGF: Input capture register GF
ICRGR: Input capture register GR
IRRTG: Timer G interrupt request flag
NCS: Noise canceler select
PSS: Prescaler S

Figure 13.1 Block Diagram of Timer G

13.2 Input/Output Pins

Table 13.1 shows the timer G pin configuration.

Table 13.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Input capture input	TMIG	Input	Input capture input pin

13.3 Register Descriptions

The timer G has the following registers. For details on the clock halt register 3 (CKSTPR3), see section 6.1.4, Clock Halt Registers 1 to 3 (CKSTPR1 to CKSTPR3).

- Timer counter G (TCG)
- Input capture register GF (ICRGF)
- Input capture register GR (ICRGR)
- Timer mode register G (TMG)
- Clock halt register 3 (CKSTPR3)

13.3.1 Timer Counter G (TCG)

TCG is an 8-bit up-counter which is incremented by clock input. The input clock is selected by bits CKS1 and CKS0 in TMG.

TMIG in PMRF is set to 1 to operate TCG as an input capture timer, or cleared to 0 to operate TCG as an interval timer*. In input capture timer operation, the TCG value can be cleared by the rising edge, falling edge, or both edges of the input capture input signal, according to the setting made in TMG.

When TCG overflows from H'FF to H'00, if OVIE in TMG is 1, IRRTG in IRR2 is set to 1, and if IENTG in IENR2 is 1, an interrupt request is sent to the CPU.

For details on interrupts, see section 4, Interrupt Controller.

TCG cannot be read or written by the CPU. It is initialized to H'00 upon reset.

Note: An input capture signal may be generated when TMIG is modified.

13.3.2 Input Capture Register GF (ICRGF)

ICRGF is an 8-bit read-only register. When a falling edge of the input capture input signal is detected, the current TCG value is transferred to ICRGF. If IIEGS in TMG is 1 at this time, IRRTG in IRR2 is set to 1, and if IENTG in IENR2 is 1, an interrupt request is sent to the CPU.

For details on interrupts, see section 4, Interrupt Controller.

To ensure dependable input capture operation, the pulse width of the input capture input signal must be at least 2ϕ or 2ϕ_{SUB} (when the noise canceller is not used).

ICRGF is initialized to H'00 upon reset.

13.3.3 Input Capture Register GR (ICRGR)

ICRGR is an 8-bit read-only register. When a rising edge of the input capture input signal is detected, the current TCG value is transferred to ICRGR. If IIEGS in TMG is 0 at this time, IRRTG in IRR2 is set to 1, and if IENTG in IENR2 is 1, an interrupt request is sent to the CPU.

For details on interrupts, see section 4, Interrupt Controller.

To ensure dependable input capture operation, the pulse width of the input capture input signal must be at least 2ϕ or 2ϕ_{SUB} (when the noise canceller is not used).

ICRGR is initialized to H'00 upon reset.

13.3.4 Timer Mode Register G (TMG)

TMG is an 8-bit readable/writable register that performs TCG clock selection from four internal clock sources, counter clear selection, and edge selection for the input capture input signal interrupt request, controls enabling of overflow interrupt requests, and also contains the overflow flags.

TMG is initialized to H'00 upon reset.

Bit	Bit Name	Initial Value	R/W	Description
7	OVFH	0	R/(W)*	Timer Overflow Flag H Indicates that TCG has overflowed from H'FF to H'00 when the input capture input signal is high. This flag is set by hardware and cleared by software. It cannot be set by software. [Setting condition] Set when input capture input signal is high level and TCG overflows from H'FF to H'00 [Clearing condition] Writing 0 to OVFH after reading OVFH = 1
6	OVFL	0	R/(W)*	Timer Overflow Flag L Indicates that TCG has overflowed from H'FF to H'00 when the input capture input signal is low, or in interval operation. This flag is set by hardware and cleared by software. It cannot be set by software. [Setting condition] Set when TCG overflows from H'FF to H'00 while input capture input signal is low level or during interval operation [Clearing condition] Writing 0 to OVFL after reading OVFL = 1
5	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects enabling or disabling of interrupt generation when TCG overflows. 0: TCG overflow interrupt request is disabled 1: TCG overflow interrupt request is enabled

Bit	Bit Name	Initial Value	R/W	Description
4	IIEGS	0	R/W	Input Capture Interrupt Edge Select Selects the input capture input signal edge that generates an interrupt request. 0: Interrupt generated on rising edge of input capture input signal 1: Interrupt generated on falling edge of input capture input signal
3	CCLR1	0	R/W	Counter Clear 1 and 0
2	CCLR0	0	R/W	Specify whether or not TCG is cleared by the rising edge, falling edge, or both edges of the input capture input signal. 00: TCG clearing is disabled 01: TCG cleared by falling edge of input capture input signal 10: TCG cleared by rising edge of input capture input signal 11: TCG cleared by both edges of input capture input signal
1	CKS1	0	R/W	Clock Select
0	CKS0	0	R/W	Select the clock input to TCG from four internal clock sources. 00: Internal clock: counting on ϕ/64 01: Internal clock: counting on ϕ/32 10: Internal clock: counting on ϕ/2 11: Internal clock: counting on ϕ_w/4

Note: * Only 0 can be written to clear the flag.

13.3.5 Clock Halt Register 3 (CKSTPR3)

For details on placing timer G in and taking it out of module standby mode (this is controlled by the TGCKSTP bit in CKSTPR3) see section 6.1.4, Clock Halt Registers 1 to 3 (CKSTPR1 to CKSTPR3).

13.4 Noise Canceller

The noise canceller consists of a digital low-pass filter that eliminates high-frequency component noise from the pulses input from the input capture input pin. The noise canceller is set by NCS* in PMRF.

Figure 13.2 shows a block diagram of the noise canceller.

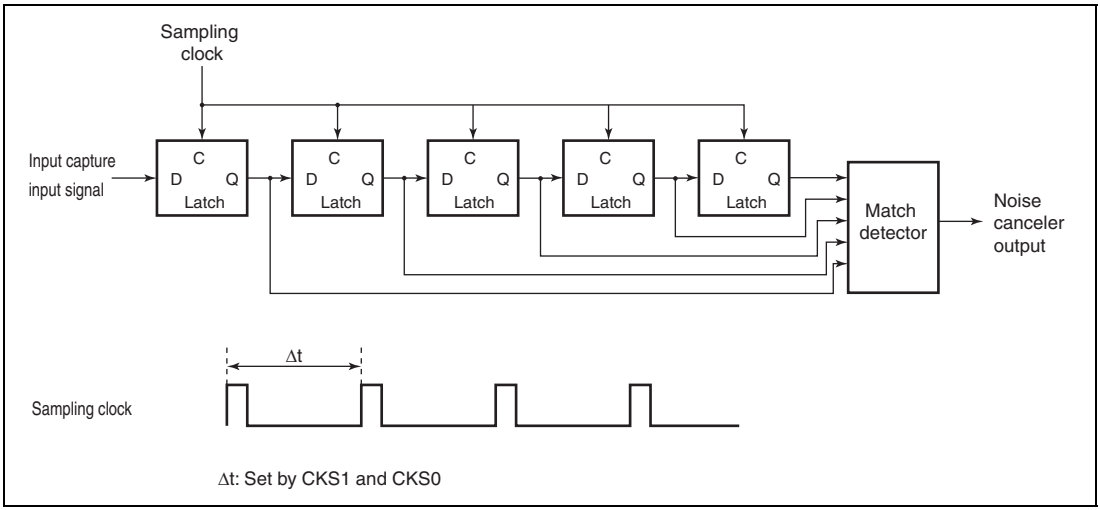


Figure 13.2 Noise Canceller Block Diagram

The noise canceller consists of five latch circuits connected in series and a match detector circuit. When the noise cancellation function is not used (NCS = 0), the system clock is selected as the sampling clock. When the noise cancellation function is used (NCS = 1), the sampling clock is the internal clock selected by CKS1 and CKS0 in TMG, the input capture input is sampled on the rising edge of this clock, and the data is judged to be correct when all the latch outputs match. If all the outputs do not match, the previous value is retained. After a reset, the noise canceller output is initialized when the falling edge of the input capture input signal has been sampled five times. Therefore, after making a setting for use of the noise cancellation function, a pulse with at least five times the width of the sampling clock is a dependable input capture signal. Even if noise cancellation is not used, an input capture input signal pulse width of at least 2ϕ or $2\phi_{SUB}$ is necessary to ensure that input capture operations are performed properly.

Note: * An input capture signal may be generated when the NCS bit is modified.

Figure 13.3 shows an example of noise canceller timing.

In this example, high-level input of not more than five times the width of the sampling clock at the input capture input pin is eliminated as noise.

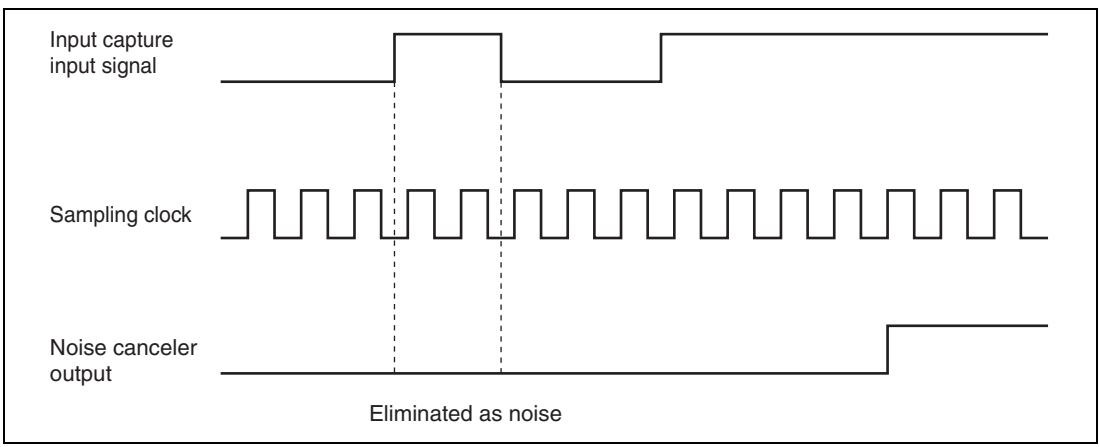


Figure 13.3 Noise Canceller Timing (Example)

13.5 Operation

Timer G is an 8-bit timer with built-in input capture and interval functions.

13.5.1 Timer G Functions

Timer G is an 8-bit up-counter with two functions, an input capture timer function and an interval timer function.

The operation of these two functions is described below.

(1) Input Capture Timer Operation

When the TMIG bit in the port mode register F (PMRF) is set to 1, timer G functions as an input capture timer*.

Upon reset, the timer mode register G (TMG), timer counter G (TCG), input capture register GF (ICRGF), and input capture register GR (ICRGR) are all initialized to H'00.

Following a reset, TCG starts counting on the ϕ/64 internal clock.

The input clock can be selected from four internal clock sources by bits CKS1 and CKS0 in TMG.

When a rising edge/falling edge is detected in the input capture signal input from the TMIG pin, the TCG value at that time is transferred to ICRGR/ICRGF. When the edge selected by IIEGS in TMG is input, IRRTG in IRR2 is set to 1, and if the IENTG bit in IENR2 is 1 at this time, an interrupt request is sent to the CPU. For details on interrupts, see section 4, Interrupt Controller.

TCG can be cleared by a rising edge, falling edge, or both edges of the input capture signal, according to the setting of bits CCLR1 and CCLR0 in TMG. If TCG overflows when the input capture signal is high, the OVFH bit in TMG is set; if TCG overflows when the input capture signal is low, the OVFL bit in TMG is set. If the OVIE bit in TMG is 1 when these bits are set, IRRTG in IRR2 is set to 1, and if the IENTG bit in IENR2 is 1, timer G sends an interrupt request to the CPU. For details on interrupts, see section 4, Interrupt Controller.

Timer G has a built-in noise canceller that enables high-frequency component noise to be eliminated from pulses input from the TMIG pin. For details, see section 13.4, Noise Canceller.

Note: * An input capture signal may be generated when TMIG is modified.

(2) Interval Timer Operation

When the TMIG bit in PMRF is cleared to 0, timer G functions as an interval timer.

Following a reset, TCG starts counting on the ϕ/64 internal clock. The input clock can be selected from four internal clock sources by bits CKS1 and CKS0 in TMG. TCG increments on the selected clock, and when it overflows from H'FF to H'00, the OVFL bit in TMG is set to 1. If the OVIE bit in TMG is 1 at this time, IRRTG in IRR2 is set to 1, and if the IENTG bit in IENR2 is 1, timer G sends an interrupt request to the CPU. For details on interrupts, see section 4, Interrupt Controller.

13.5.2 Count Timing

TCG is incremented by internal clock input. Bits CKS1 and CKS0 in TMG select one of four internal clock sources (ϕ/64, ϕ/32, ϕ/2, or ϕ_w/4) created by dividing the system clock (ϕ) or watch clock (ϕ_w).

13.5.3 Input Capture Input Timing

(1) Without Noise Cancellation Function

For input capture input, dedicated input capture functions are provided for rising and falling edges.

Figure 13.4 shows the timing for rising/falling edge input capture input.

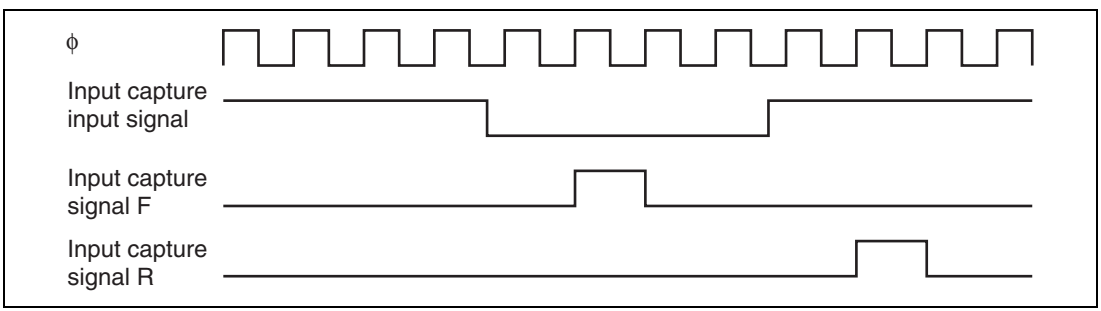


Figure 13.4 Input Capture Input Timing (without Noise Cancellation Function)

(2) With Noise Cancellation Function

When noise cancellation is performed on the input capture input, the passage of the input capture signal through the noise canceller results in a delay of five sampling clock cycles from the input capture input signal edge.

Figure 13.5 shows the timing in this case.

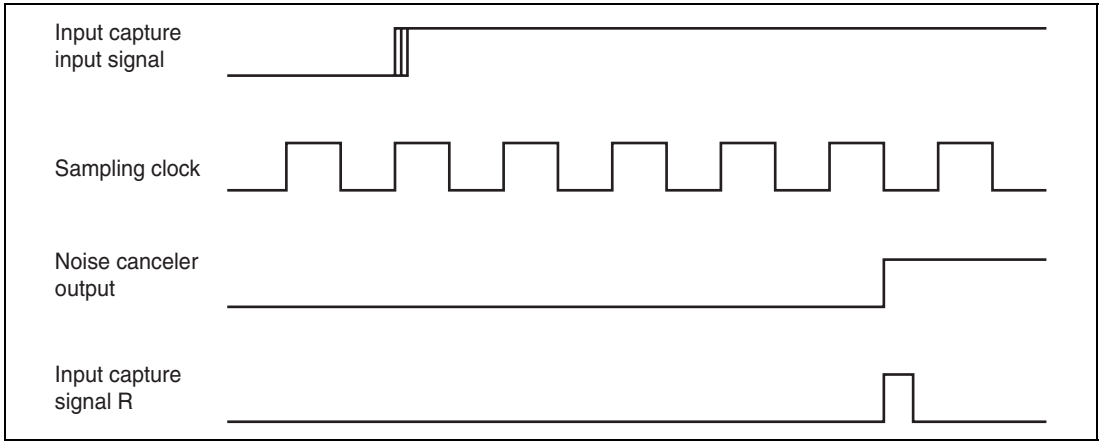


Figure 13.5 Input Capture Input Timing (with Noise Cancellation Function)

13.5.4 Timing of Input Capture by Input Capture Input

Figure 13.6 shows the timing of input capture by input capture input.

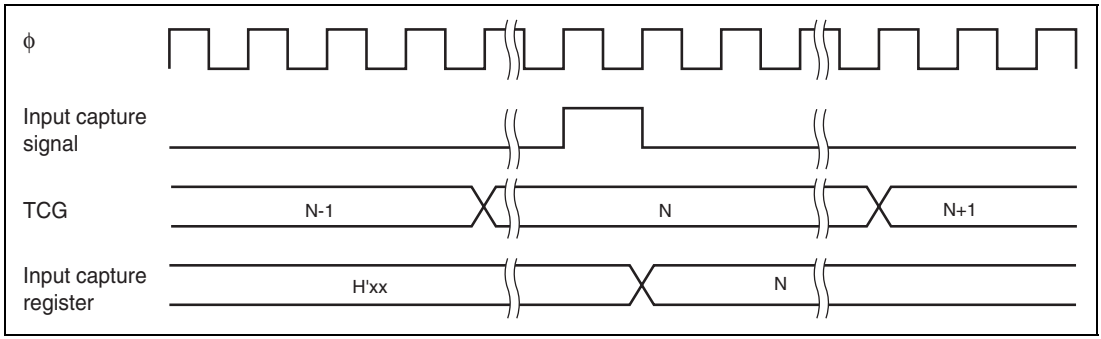


Figure 13.6 Timing of Input Capture by Input Capture Input

13.5.5 TCG Clear Timing

TCG can be cleared by the rising edge, falling edge, or both edges of the input capture input signal.

Figure 13.7 shows the timing for clearing by both edges.

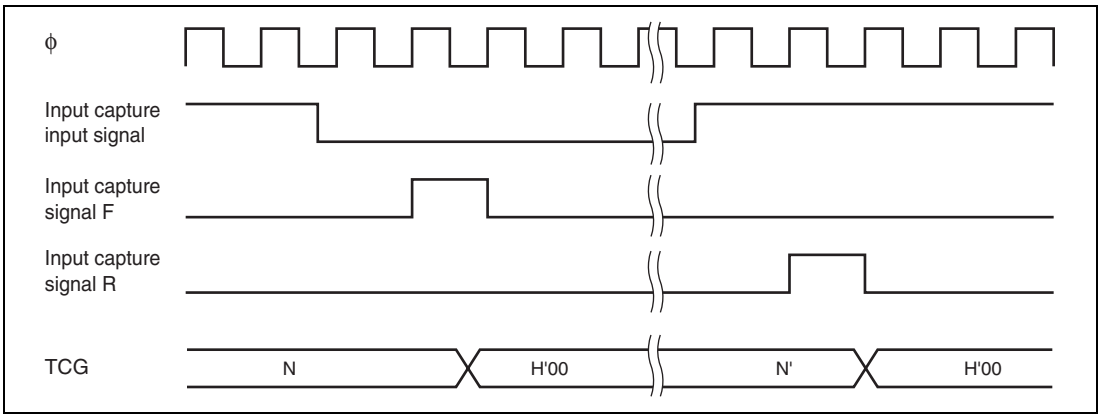


Figure 13.7 TCG Clear Timing

13.6 Timer G Operation Modes

Timer G operation modes are shown in table 13.2.

Table 13.2 Timer G Operation Modes

Operation Mode		Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module Standby
TCG	Input capture	Reset	Functioning*1	Functioning*1	Functioning/ halted*2	Functioning/ halted*3	Functioning/ halted*3	Halted	Halted
	Interval	Reset	Functioning*1	Functioning*1	Functioning/ halted*2	Functioning/ halted*3	Functioning/ halted*3	Halted	Halted
ICRGF		Reset	Functioning*1	Functioning*1	Functioning/ halted*2	Functioning/ halted*3	Functioning/ halted*3	Retained	Retained
ICRGR		Reset	Functioning*1	Functioning*1	Functioning/ halted*2	Functioning/ halted*3	Functioning/ halted*3	Retained	Retained
TMG		Reset	Functioning	Retained	Retained	Functioning	Retained	Retained	Retained

Notes: 1. When $\phi_W/4$ is selected as the TCG internal clock in active mode or sleep mode, since the system clock and internal clock are mutually asynchronous, synchronization is maintained by a synchronization circuit. This results in a maximum count cycle error of $1/\phi$(s).

2. When $\phi_W/4$ is selected as the TCG internal clock in watch mode, TCG and the noise canceller operate on the $\phi_W/4$ internal clock without regard to the ϕ_{SUB} subclock ($\phi_W/8$, $\phi_W/4$, $\phi_W/2$). Note that when another internal clock is selected, TCG and the noise canceller do not operate, and input of the input capture input signal does not result in input capture.

3. To operate the timer G in subactive mode or subsleep mode, select $\phi_W/4$ as the TCG internal clock and $\phi_W/2$ as the subclock ϕ_{SUB}. Note that when other internal clock is selected, or when $\phi_W/8$ or $\phi_W/4$ is selected as the subclock ϕ_{SUB}, TCG and the noise canceller do not operate.

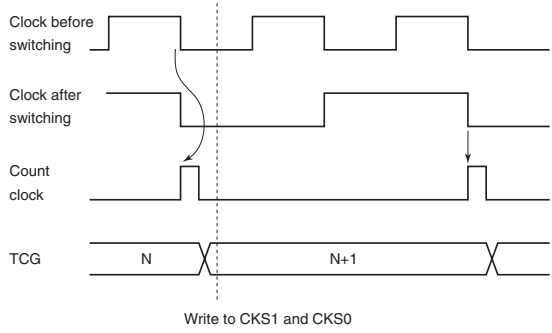
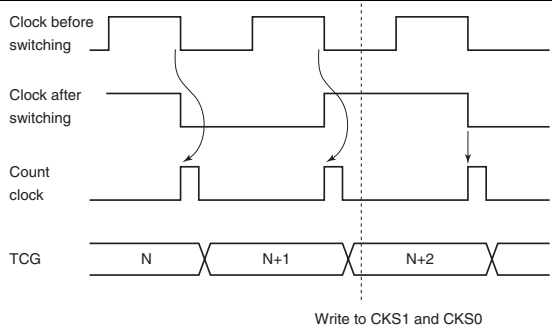
13.7 Usage Notes

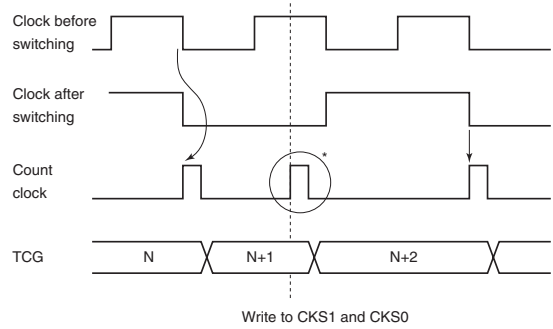
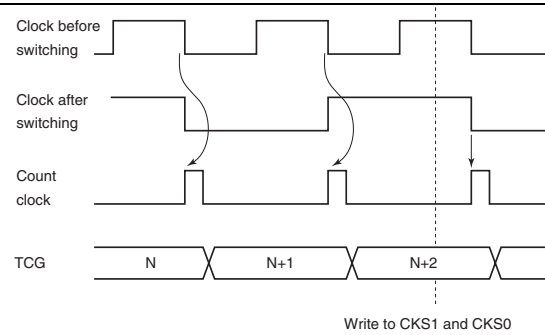
13.7.1 Internal Clock Switching and TCG Operation

Depending on the timing, TCG may be incremented by a switch between different internal clock sources. Table 13.3 shows the relation between internal clock switchover timing (by write to bits CKS1 and CKS0) and TCG operation.

When TCG is internally clocked, an increment pulse is generated on detection of the falling edge of an internal clock signal, which is divided from the system clock (ϕ) or subclock (ϕ_w). For this reason, in a case like No.3 in table 13.3 where the switch is from a high clock signal to a low clock signal, the switchover is seen as a falling edge, causing TCG to increment.

Table 13.3 Internal Clock Switching and TCG Operation

No.	Clock Levels Before and After Modifying Bits CKS1 and CKS0	TCG Operation
1	Goes from low level to low level	
2	Goes from low level to high level	

No.	Clock Levels Before and After Modifying Bits CKS1 and CKS0	TCG Operation
3	Goes from high level to low level	
4	Goes from high level to high level	

Note: * The switchover is seen as a falling edge, and TCG is incremented.

13.7.2 Notes on Port Mode Register Modification

The following points should be noted when a port mode register is modified to switch the input capture function or the input capture input noise canceller function.

- Switching input capture input pin function

  Note that when the pin function is switched by modifying TMIG in the port mode register F (PMRF), which performs input capture input pin control, an edge will be regarded as having been input at the pin even though no valid edge has actually been input. Input capture input signal input edges, and the conditions for their occurrence, are summarized in table 13.4.

Table 13.4 Input Capture Input Signal Input Edges Due to Input Capture Input Pin Switching, and Conditions for Their Occurrence

Input Capture Input Signal Input Edge	Conditions
Generation of rising edge	TMIG is modified from 0 to 1 while the TMIG pin is high
	NCS is modified from 0 to 1 while the TMIG pin is high, then TMIG is modified from 0 to 1 before the signal is sampled five times by the noise canceller
Generation of falling edge	TMIG is modified from 1 to 0 while the TMIG pin is high
	NCS is modified from 0 to 1 while the TMIG pin is low, then TMIG is modified from 0 to 1 before the signal is sampled five times by the noise canceller
	NCS is modified from 0 to 1 while the TMIG pin is high, then TMIG is modified from 1 to 0 after the signal is sampled five times by the noise canceller

Note: When the PF0 pin is not set as an input capture input pin, the timer G input capture input signal is low.

- Switching input capture input noise canceller function

  When performing noise canceller function switching by modifying NCS in the port mode register F (PMRF), which controls the input capture input noise canceller, TMIG should first be cleared to 0. Note that if NCS is modified without first clearing TMIG, an edge will be regarded as having been input at the pin even though no valid edge has actually been input. Input capture input signal input edges, and the conditions for their occurrence, are summarized in table 13.5.

Table 13.5 Input Capture Input Signal Input Edges Due to Noise Canceller Function Switching, and Conditions for Their Occurrence

Input Capture Input Signal Input Edge	Conditions
Generation of rising edge	TMIG pin is modified from 0 to 1 while TMIG is 1, then NCS is modified from 0 to 1 before the signal is sampled five times by the noise canceller
Generation of falling edge	TMIG pin is modified from 1 to 0 while TMIG is 1, then NCS is modified from 1 to 0 before the signal is sampled five times by the noise canceller

When the pin function is switched and an edge is generated in the input capture input signal, if this edge matches the edge selected by the input capture interrupt select (IIEGS) bit, the interrupt request flag will be set to 1. The interrupt request flag should therefore be cleared to 0 before use. Figure 13.8 shows the procedure for handling of the port mode register and clearing of the interrupt request flag. When switching the pin function, set the interrupt-disabled state before handling the port mode register, then, after the port mode register operation has been performed, wait for the time required to confirm the input capture input signal as an input capture signal (at least two system clocks when the noise canceller is not in use; at least five sampling clocks when the noise canceller is used), before clearing the interrupt enable flag to 0. There are two ways of preventing interrupt request flag setting when the pin function is switched: by controlling the pin level so that the conditions shown in tables 13.4 and 13.5 are not satisfied, or by setting the opposite of the generated edge in the IIEGS bit in TMG.

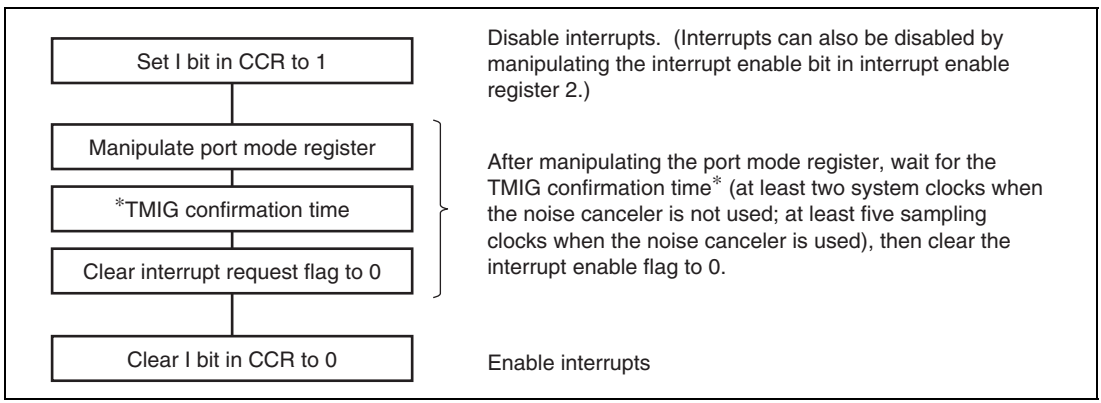


Figure 13.8 Port Mode Register Manipulation and Interrupt Enable Flag Clearing Procedure

13.8 Timer G Application Example

Using timer G, it is possible to measure the high and low widths of the input capture input signal as absolute values. For this purpose, CCLR1 and CCLR0 in TMG should both be set to 1.

Figure 13.9 shows an example of the operation in this case.

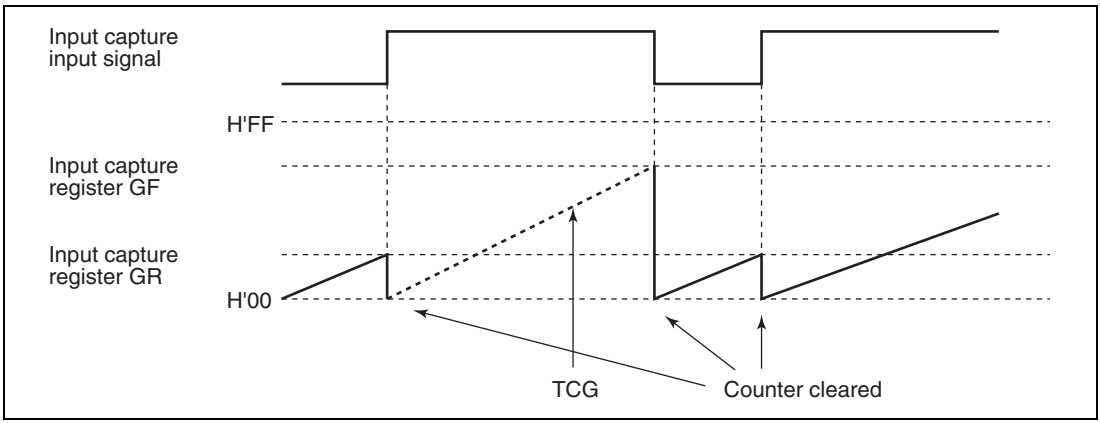


Figure 13.9 Timer G Application Example

Section 14 16-Bit Timer Pulse Unit (TPU)

Microcontrollers of the H8/38799 Group have an on-chip 16-bit timer pulse unit (TPU) that comprises two 16-bit timer channels. The function list of the TPU is shown in table 14.1. A block diagram of the TPU is shown in figure 14.1.

14.1 Features

- Maximum 4-pulse input/output
- Selection of 7 or 8 counter input clocks for each channel
- The following operations can be set for each channel:
  Waveform output at compare match
  Input capture function
  Counter clear operation
  Multiple timer counters (TCNT) can be written to simultaneously
  Simultaneous clearing by compare match and input capture is possible
  Register synchronous input/output is possible by synchronous counter operation
  PWM output with any duty level is possible
  A maximum 3-phase PWM output is possible in combination with synchronous operation
- Operation with cascaded connection
- Fast access via internal 16-bit bus
- 6 types of interrupt sources
- Module standby mode enables this module to be placed in standby mode independently when it is not in use (for details, refer to section 6.4, Module Standby Function).

Table 14.1 TPU Functions

Item		Channel 1	Channel 2
Count clock		φ/1	φ/1
		φ/4	φ/4
		φ/16	φ/16
		φ/64	φ/64
		φ/256	φ/1024
		TCLKA	TCLKA
		TCLKB	TCLKB
			TCLKC
General registers (TGR)		TGRA_1	TGRA_2
		TGRB_1	TGRB_2
I/O pins		TIOCA1	TIOCA2
		TIOCB1	TIOCB2
Counter clear function		TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	Available	Available
	1 output	Available	Available
	Toggle output	Available	Available
Input capture function		Available	Available
Synchronous operation		Available	Available
PWM mode		Available	Available
Interrupt sources		3 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow	3 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow

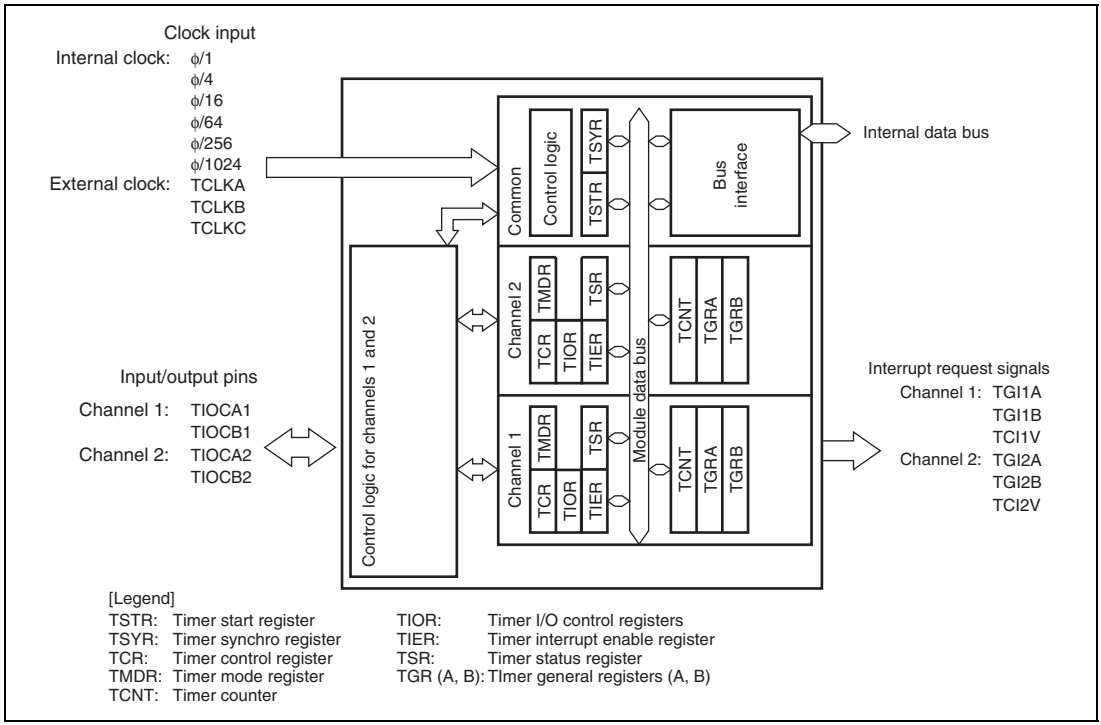


Figure 14.1 Block Diagram of TPU

14.2 Input/Output Pins

Table 14.2 Pin Configuration

Channel	Symbol	I/O	Function
Common	TCLKA	Input	External clock A input pin
	TCLKB	Input	External clock B input pin
	TCLKC	Input	External clock C input pin
1	TIOCA1	I/O	TGRA_1 input capture input/output compare output/PWM output pin
	TIOCB1	I/O	TGRB_1 input capture input/output compare output/PWM output pin
2	TIOCA2	I/O	TGRA_2 input capture input/output compare output/PWM output pin
	TIOCB2	I/O	TGRB_2 input capture input/output compare output/PWM output pin

14.3 Register Descriptions

The TPU has the following registers for each channel.

Channel 1:

- Timer control register_1 (TCR_1)
- Timer mode register_1 (TMDR_1)
- Timer I/O control register_1 (TIOR_1)
- Timer interrupt enable register_1 (TIER_1)
- Timer status register_1 (TSR_1)
- Timer counter_1 (TCNT_1)
- Timer general register A_1 (TGRA_1)
- Timer general register B_1 (TGRB_1)

Channel 2:

- Timer control register_2 (TCR_2)
- Timer mode register_2 (TMDR_2)
- Timer I/O control register_2 (TIOR_2)
- Timer interrupt enable register_2 (TIER_2)
- Timer status register_2 (TSR_2)
- Timer counter_2 (TCNT_2)
- Timer general register A_2 (TGRA_2)
- Timer general register B_2 (TGRB_2)

Common:

- Timer start register (TSTR)
- Timer synchro register (TSYR)

14.3.1 Timer Control Register (TCR)

TCR controls TCNT operation for each channel. The TPU has a total of two TCR registers, one for each channel. TCR should be set when TCNT operation is stopped.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
6	CCLR1	0	R/W	Counter Clear 1 and 0
5	CCLR0	0	R/W	These bits select the TCNT counter clearing source. See table 14.3 for details.
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the input clock edge. When the internal clock is counted using both edges, the input clock period is halved (e.g. ϕ/4 both edges = ϕ/2 rising edge). Internal clock edge selection is valid when the input clock is ϕ/4 or slower. If the input clock is ϕ/1, this setting is ignored and count at a rising edge is selected. 00: Count at rising edge 01: Count at falling edge 1x: Count at both edges [Legend] x: Don't care
2	TPSC2	0	R/W	Timer Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables14.4 and 14.5 for details.
0	TPSC0	0	R/W	

Table 14.3 CCLR1 and CCLR0 (Channels 1 and 2)

Channel	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRA compare match/input capture
	1	0	TCNT cleared by TGRB compare match/input capture
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*

Note: * Synchronous operation is selected by setting the SYNC bit in TSYR to 1.

Table 14.4 TPSC2 to TPSC0 (Channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	Internal clock: counts on ϕ/256
			1	Counts on TCNT_2 overflow

Table 14.5 TPSC2 to TPSC0 (Channel 2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	Internal clock: counts on ϕ/1024

14.3.2 Timer Mode Register (TMDR)

TMDR sets the operating mode for each channel. The TPU has a total of two TMDR registers, one for each channel. TMDR should be set when TCNT operation is stopped.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
5, 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3, 2	—	All 0	—	Reserved The write value should always be 0.
1	MD1	0	R/W	Modes 1 and 0
0	MD0	0	R/W	These bits set the timer operating mode. See table 14.6 for details.

Table 14.6 MD1 to MD0

Bit 1 MD1	Bit 0 MD0	Description
0	0	Normal operation
	1	Reserved
1	0	PWM mode 1
	1	PWM mode 2

14.3.3 Timer I/O Control Register (TIOR)

TIOR controls TGR. The TPU has a total of two TIOR registers, one for each channel. Care is required as TIOR is affected by the TMDR setting.

The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

- TIOR_1, TIOR_2

Bit	Bit Name	Initial Value	R/W	Description
7	IOB3	All 0	R/W	I/O Control B3 to B0
6	IOB2		R/W	Specify the function of TGRB.
5	IOB1		R/W	For details, refer to tables 14.7 and 14.8.
4	IOB0		R/W	
3	IOA3	All 0	R/W	I/O Control A3 to A0
2	IOA2		R/W	Specify the function of TGRA.
1	IOA1		R/W	For details, refer to tables 14.9 and 14.10.
0	IOA0		R/W	

Table 14.7 TIOR_1 (Channel 1)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_1 Function**	**TIOCB1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB1 pin Input capture at rising edge
			1		Capture input source is TIOCB1 pin Input capture at falling edge
		1	x		Capture input source is TIOCB1 pin Input capture at both edges
	1	x	x		Setting prohibited

[Legend]

x: Don't care

Table 14.8 TIOR_2 (Channel 2)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_2 Function**	**TIOCB2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCB2 pin Input capture at rising edge
			1		Capture input source is TIOCB2 pin Input capture at falling edge
		1	x		Capture input source is TIOCB2 pin Input capture at both edges

[Legend]

x: Don't care

Table 14.9 TIOR_1 (Channel 1)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_1 Function**	**TIOCA1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA1 pin Input capture at rising edge
			1		Capture input source is TIOCA1 pin Input capture at falling edge
		1	x		Capture input source is TIOCA1 pin Input capture at both edges
	1	x	x		Setting prohibited

[Legend]

x: Don't care

Table 14.10 TIOR_2 (Channel 2)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_2 Function**	**TIOCA2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCA2 pin Input capture at rising edge
			1		Capture input source is TIOCA2 pin Input capture at falling edge
		1	x		Capture input source is TIOCA2 pin Input capture at both edges

[Legend]

x: Don't care

14.3.4 Timer Interrupt Enable Register (TIER)

TIER controls enabling or disabling of interrupt requests for each channel. The TPU has a total of two TIER registers, one for each channel.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved This bit is readable/writable.
6	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
5	—	0	—	Reserved The write value should always be 0.
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3, 2	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabled 1: Interrupt requests (TGIB) by TGFB bit enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled

14.3.5 Timer Status Register (TSR)

TSR indicates the status for each channel. The TPU has a total of two TSR registers, one for each channel.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
5	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
4	TCFV	0	R/(W)*	Overflow Flag Status flag that indicates that TCNT overflow has occurred. [Setting condition] The TCNT value overflows (changes from H'FFFF to H'0000) [Clearing condition] Writing of 0 to bit TCFV after reading TCFV = 1
3, 2	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
1	TGFB	0	R/(W)*	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGRB input capture or compare match. [Setting conditions] • TCNT = TGRB and TGRB is functioning as output compare register • The TCNT value is transferred to TGRB by input capture signal and TGRB is functioning as input capture register [Clearing condition] • Writing of 0 to bit TGFB after reading TGFB = 1

Bit	Bit Name	Initial value	R/W	Description
0	TGFA	0	R/(W)*	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TGRA input capture or compare match. [Setting conditions] • TCNT = TGRA and TGRA is functioning as output compare register • The TCNT value is transferred to TGRA by input capture signal and TGRA is functioning as input capture register [Clearing condition] • Writing of 0 to bit TGFA after reading TGFA = 1

Note: * Only 0 can be written to clear the flag.

14.3.6 Timer Counter (TCNT)

TCNT is a 16-bit readable/writable counter. The TPU has a total of two TCNT counters, one for each channel.

TCNT is initialized to H'0000 by a reset or in standby mode.

TCNT cannot be accessed in 8-bit units; it must always be accessed in 16-bit units.

14.3.7 Timer General Register (TGR)

TGR is a 16-bit readable/writable register, functioning as either output compare or input capture register. The TPU has a total of four TGR registers, two for each channel. TGR is initialized to H'FFFF by a reset. TGR cannot be accessed in 8-bit units; it must always be accessed in 16-bit units.

14.3.8 Timer Start Register (TSTR)

TSTR selects TCNT operation/stoppage for channels 1 and 2. TCNT starts counting for channel in which the corresponding bit is set to 1. When setting the operating mode in TMDR or setting the TCNT count clock in TCR, first stop the TCNT operation.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R/W	Reserved The write value should always be 0.
2	CST2	0	R/W	Counter Start 2 and 1
1	CST1	0	R/W	These bits select operation or stoppage for TCNT. If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the output compare output level of the TIOC pin is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_n count operation is stopped 1: TCNT_n performs count operation [Legend] n = 2 or 1
0	—	0	R/W	Reserved The write value should always be 0.

14.3.9 Timer Synchro Register (TSYR)

TSYR selects independent operation or synchronous operation of TCNT for each channel. Synchronous operation is performed for channel in which the corresponding bit in TSYR is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R/W	Reserved The write value should always be 0.
2	SYNC2	0	R/W	Timer Synchro 2 and 1
1	SYNC1	0	R/W	These bits select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by counter clearing on another channel, are possible. To set synchronous operation, the SYNC bits must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR1 and CCLR0 in TCR. 0: TCNT_n operates independently (TCNT presetting/ clearing is unrelated to other channels) 1: TCNT_n performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible [Legend] n = 2 or 1
0	—	0	R/W	Reserved The write value should always be 0.

14.4 Interface to CPU

14.4.1 16-Bit Registers

TCNT and TGR are 16-bit registers. As the data bus to the CPU is 16 bits wide, these registers cannot be read or written to in 8-bit units; 16-bit access must always be used.

An example of 16-bit register access operation is shown in figure 14.2.

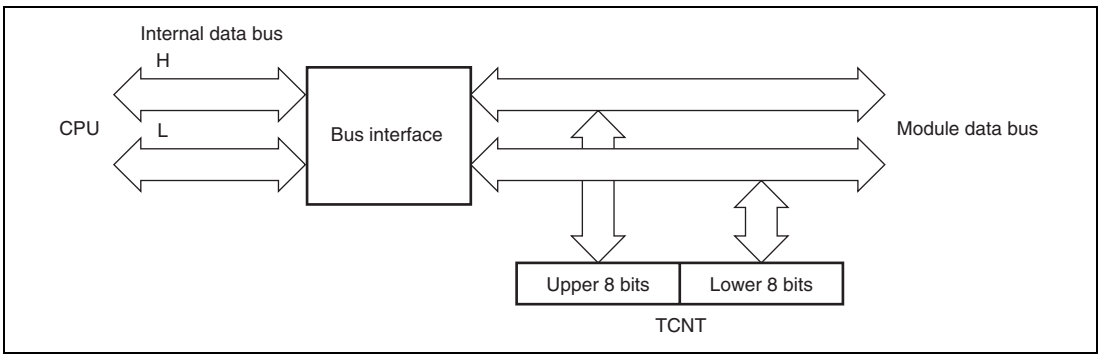


Figure 14.2 16-Bit Register Access Operation [CPU ↔ TCNT (16 Bits)]

14.4.2 8-Bit Registers

Registers other than TCNT and TGR are 8-bit registers, which can be read from and written to in 8-bit units.

Examples of 8-bit register access operation are shown in figure 14.3 and 14.4.

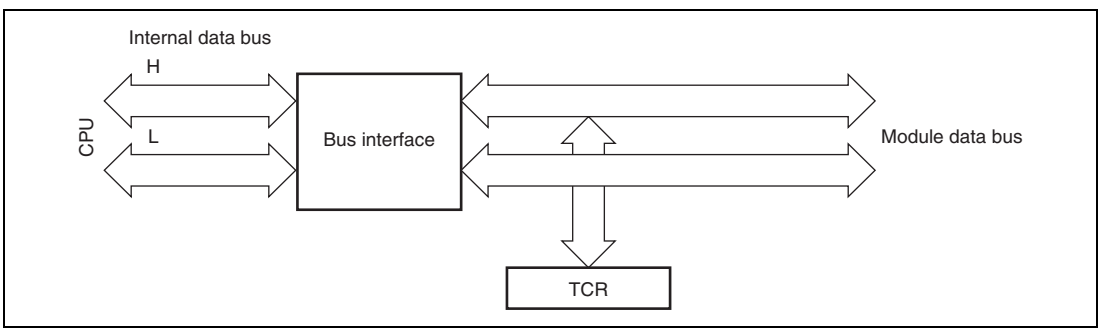


Figure 14.3 8-Bit Register Access Operation [CPU ↔ TCR (Upper 8 Bits)]

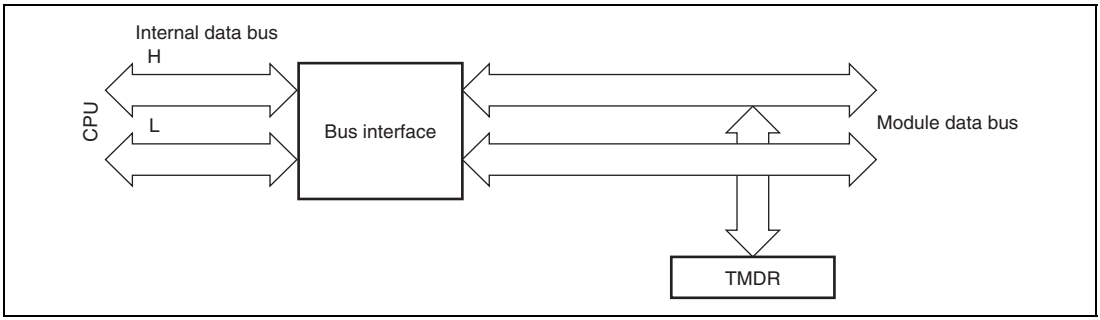


Figure 14.4 8-Bit Register Access Operation [CPU ↔ TMDR (Lower 8 Bits)]

14.5 Operation

14.5.1 Basic Functions

Each channel has TCNT and TGR. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.

TGR can be used as an input capture register or output compare register.

(1) Counter Operation

When one of bits CST1 and CST2 is set to 1 in TSTR, TCNT for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

(a) Example of Count Operation Setting Procedure

Figure 14.5 shows an example of the count operation setting procedure.

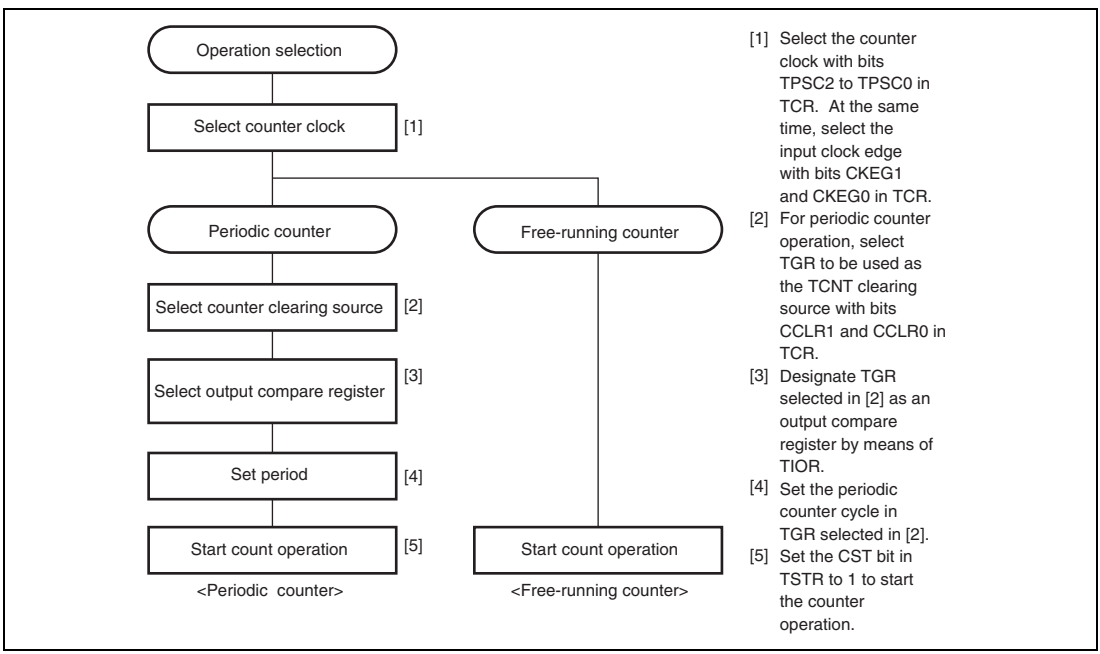


Figure 14.5 Example of Counter Operation Setting Procedure

(b) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the TPU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1, the corresponding TCNT starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the TPU requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

Figure 14.6 illustrates free-running counter operation.

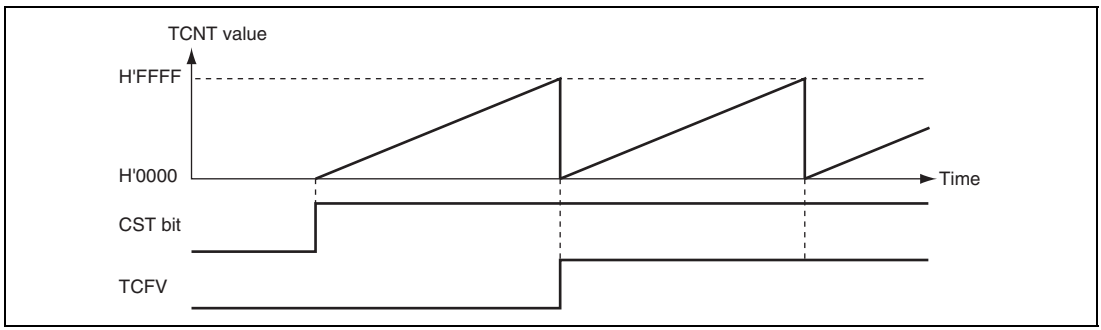


Figure 14.6 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, TCNT for the relevant channel performs periodic count operation. TGR for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR0 and CCLR1 in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 14.7 illustrates periodic counter operation.

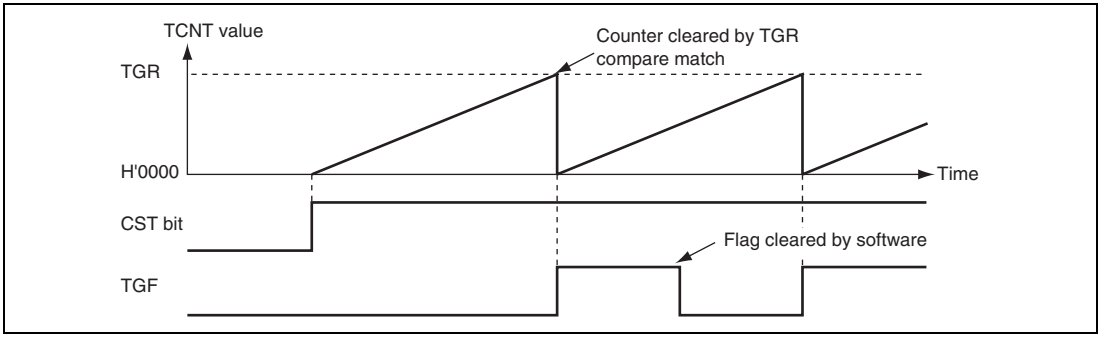


Figure 14.7 Periodic Counter Operation

(2) Waveform Output by Compare Match

The TPU can perform 0, 1, or toggle output from the corresponding output pin using compare match.

(a) Example of Setting Procedure for Waveform Output by Compare Match

Figure 14.8 shows an example of the setting procedure for waveform output by compare match.

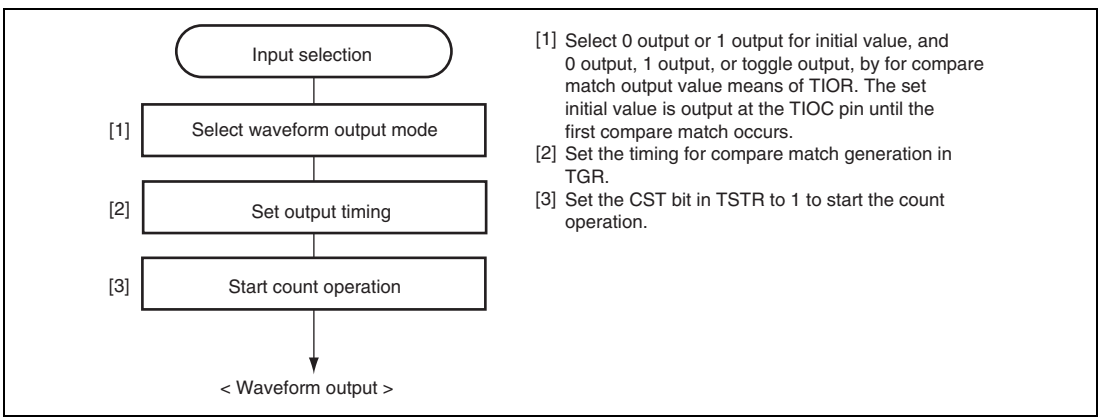


Figure 14.8 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of Waveform Output Operation

Figure 14.9 shows an example of 0 output/1 output.

In this example, TCNT has been designated as a free-running counter, and settings have been made such that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level match, the pin level does not change.

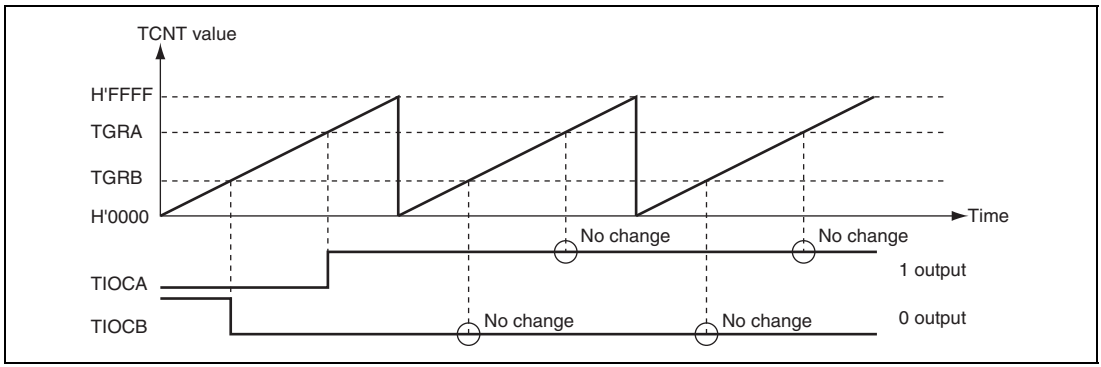


Figure 14.9 Example of 0 Output/1 Output Operation

Figure 14.10 shows an example of toggle output.

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

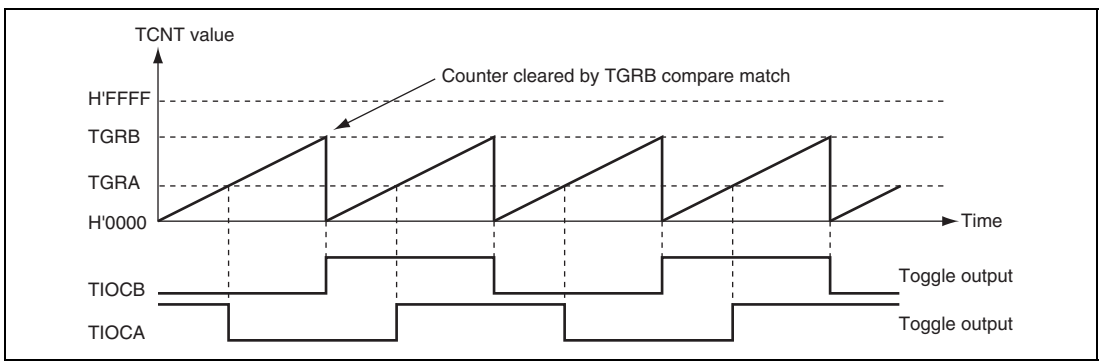


Figure 14.10 Example of Toggle Output Operation

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the detected edge.

(a) Example of Input Capture Operation Setting Procedure

Figure 14.11 shows an example of the setting procedure for input capture operation.

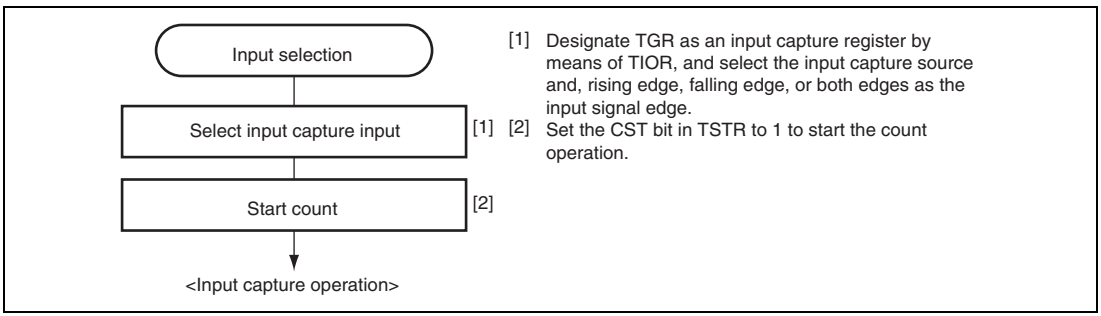


Figure 14.11 Example of Setting Procedure for Input Capture Operation

(b) Example of Input Capture Operation

Figure 14.12 shows an example of input capture operation.

In this example, both rising and falling edges have been selected as the input capture input edge of the TIOCA pin, the falling edge has been selected as the input capture input edge of the TIOCB pin, and counter clearing by TGRB input capture has been designated for TCNT.

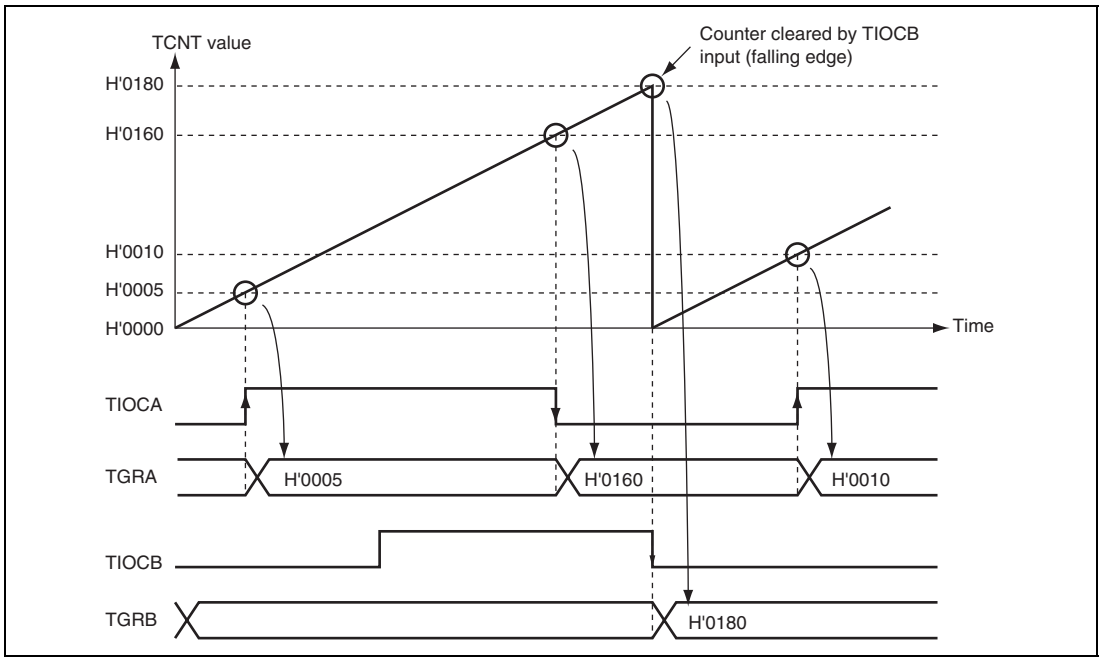


Figure 14.12 Example of Input Capture Operation

14.5.2 Synchronous Operation

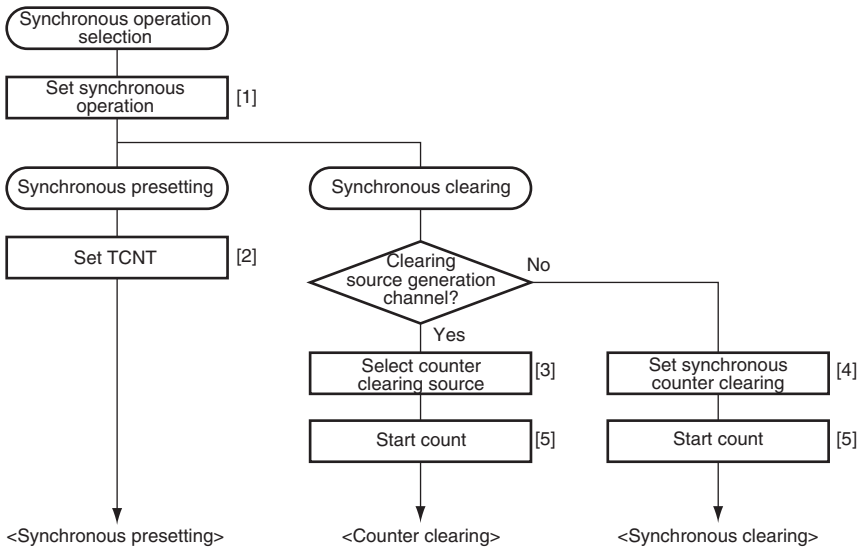
In synchronous operation, the values in multiple TCNT counters can be rewritten simultaneously (synchronous presetting). Also, multiple TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time base.

Synchronous operation can be set for each channel.

(1) Example of Synchronous Operation Setting Procedure

Figure 14.13 shows an example of the synchronous operation setting procedure.



[1] Set 1 to the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.
[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
[3] Use bits CCLR1 and CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use bits CCLR1 and CCLR0 in TCR to designate synchronous clearing for the counter clearing source.
[5] Set 1 to the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 14.13 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 14.14 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 1 and 2, TGRB_1 compare match has been set as the channel 1 counter clearing source, and synchronous clearing has been set for the channel 2 counter clearing source.

Two-phase PWM waveforms are output from pins TIOC1A and TIOC2A. At this time, synchronous presetting, and synchronous clearing by TGRB_1 compare match, are performed for channel 1 and 2 TCNT counters, and the data set in TGRB_1 is used as the PWM cycle.

For details on PWM modes, see section 14.5.4, PWM Modes.

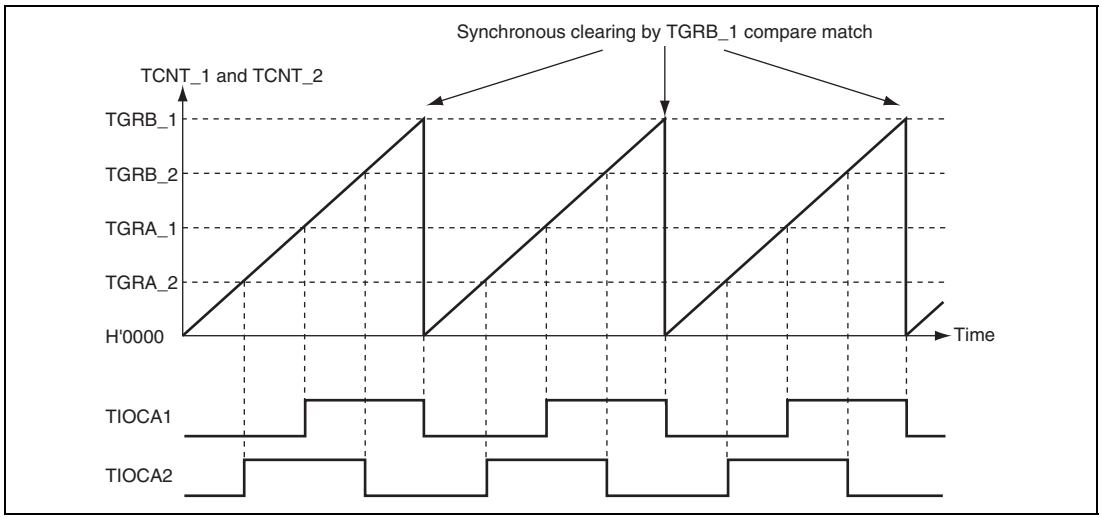


Figure 14.14 Example of Synchronous Operation

14.5.3 Operation with Cascaded Connection

Operation as a 32-bit counter can be performed by cascading two 16-bit counter channels.

This function is enabled when the TPSC2 to TPSC0 bits in TCR are set to count on TCNT2 overflow for the channel 1 counter clock.

Table 14.11 shows the counter combination used in operation with the cascaded connection.

Table 14.11 Counter Combination in Operation with Cascaded Connection

Combination	Upper 16 bits	Lower 16 bits
Channel 1 and channel 2	TCNT1	TCNT2

(1) Setting Procedure for Operation with Cascaded Connection

Figure 14.15 shows the setting procedure for cascaded connection operation.

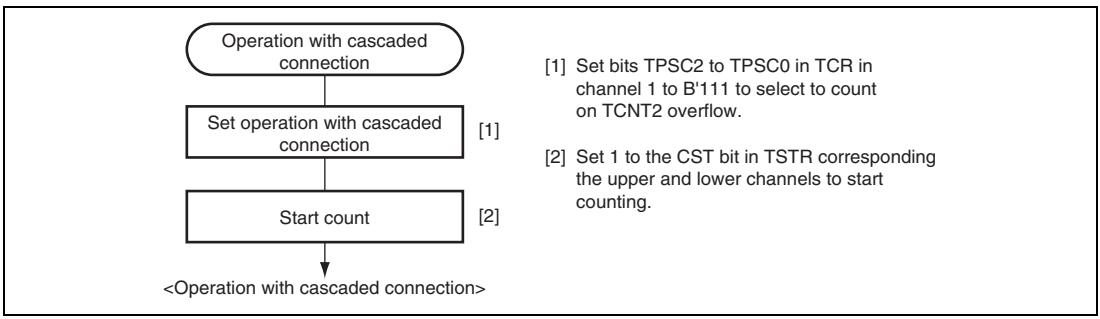


Figure 14.15 Setting Procedure for Operation with Cascaded Operation

(2) Example of Operation with Cascaded Connection

Figure 14.16 shows an example of operation with cascaded connection, where TCNT1 is set to count TCNT2 overflow, TCRA_1 and TCRA_2 are set to be input capture registers, and the TIOC pin rising edge is selected.

If rising edges are input simultaneously to the TIOCA1 and TIOCA2 pins, the upper 16 bits of 32-bit data are transferred to TGRA_1 and the lower 16 bits are transferred to TGRA_2.

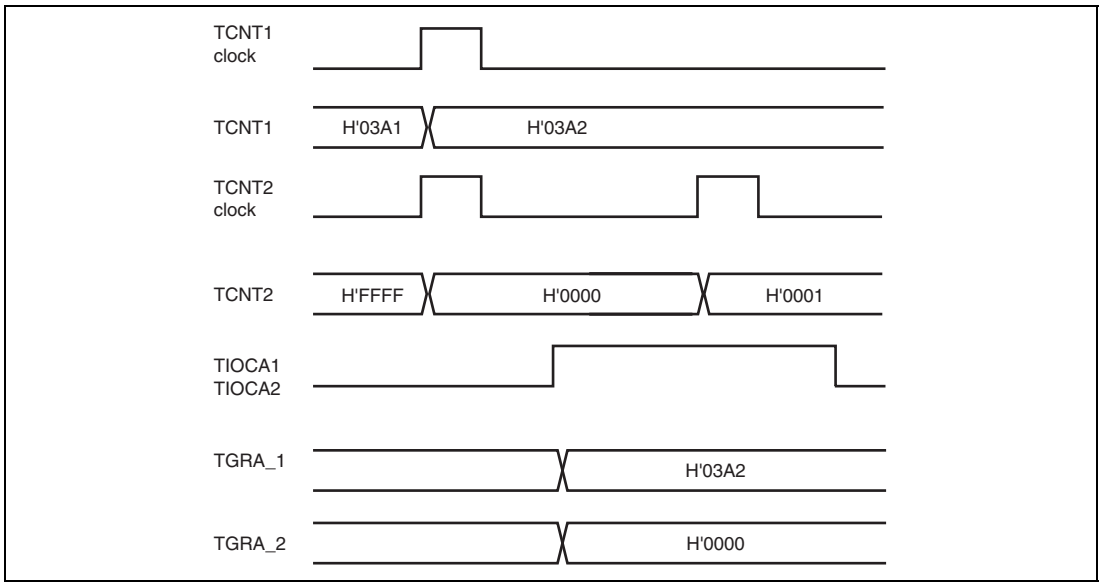


Figure 14.16 Example of Operation with Cascaded Connection

14.5.4 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. The output level can be selected as 0, 1, or toggle output in response to a compare match of each TGR.

Designating TGR compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

(1) PWM Mode 1

PWM output is generated from the TIOCA pin by pairing TGRA with TGRB. The level specified by bits IOA0 to IOA3 in TIOR is output from the TIOCA pin at compare match A, and the level specified by bits IOB0 to IOB3 in TIOR is output at compare match B. The initial output value is the value set in TGRA. If the set values of paired TGRs are identical, the output value does not change even if a compare match occurs.

In PWM mode 1, PWM output is enabled up to 2 phases.

(2) PWM Mode 2

PWM output is generated using one TGR as the cycle register and the others as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a synchronization register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the cycle and duty registers are identical, the output value does not change even if a compare match occurs.

In PWM mode 2, PWM output is enabled up to 3 phases.

The correspondence between PWM output pins and registers is shown in table 14.12.

Table 14.12 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Registers**	**PWM Mode 1**	**PWM Mode 2***
1	TGRA_1	TIOCA1	TIOCA1
	TGRB_1		TIOCB1
2	TGRA_2	TIOCA2	TIOCA2
	TGRB_2		TIOCB2

Note: * In PWM mode 2, PWM output is not possible for TGR in which the period is set.

(3) Example of PWM Mode Setting Procedure

Figure 14.17 shows an example of the PWM mode setting procedure.

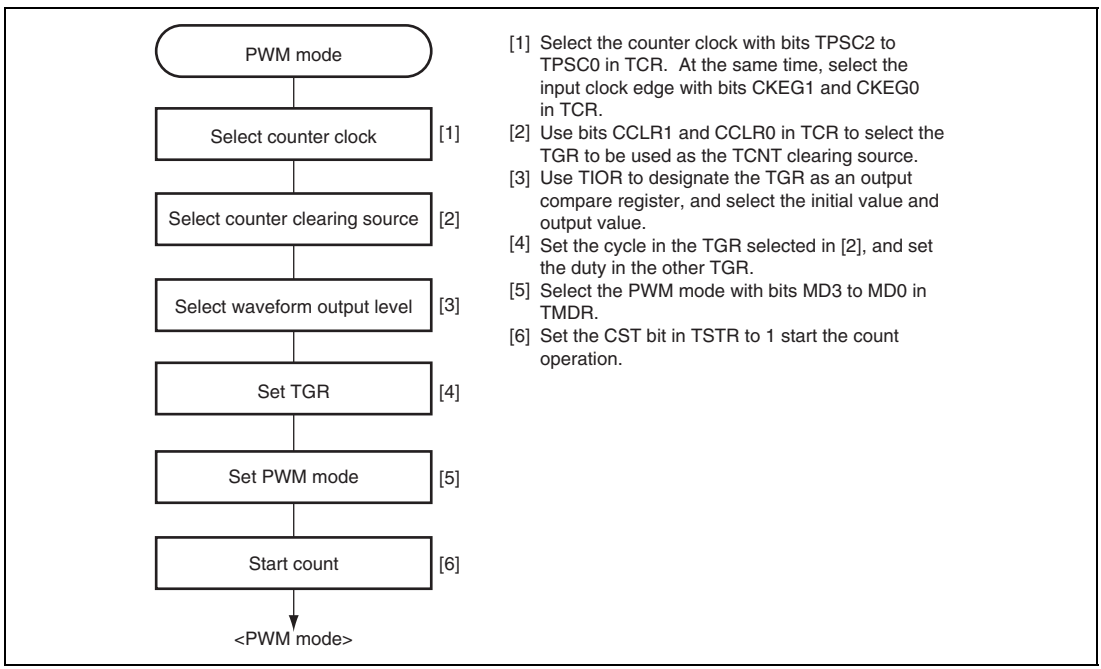


Figure 14.17 Example of PWM Mode Setting Procedure

(4) Examples of PWM Mode Operation

Figure 14.18 shows an example of PWM mode 1 operation. In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in TGRB are used as the duty levels.

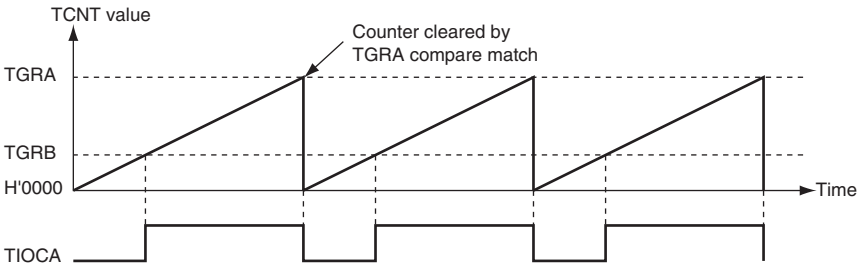


Figure 14.18 Example of PWM Mode Operation (1)

Figure 14.19 shows an example of PWM mode 2 operation. In this example, synchronous operation is designated for channels 1 and 2, TGRB_2 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA_1, TGRB_1, and TGRA_2), outputting a 3-phase PWM waveform.

In this case, the value set in TGRB_2 is used as the cycle, and the values set in the other TGRs are used as the duty levels.

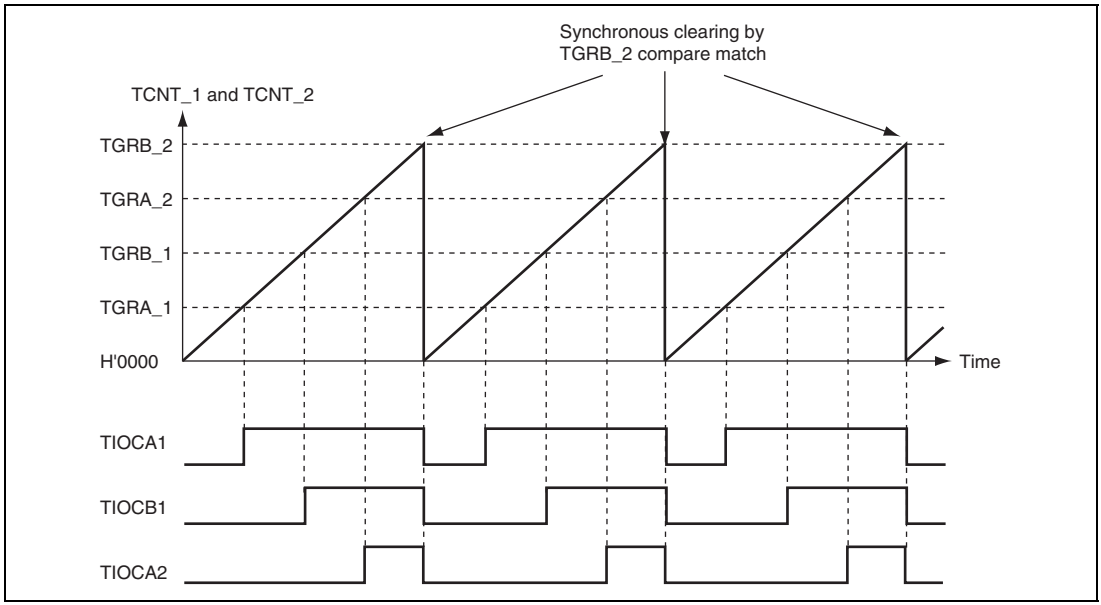
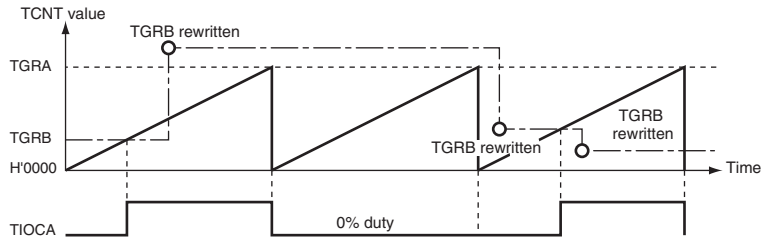


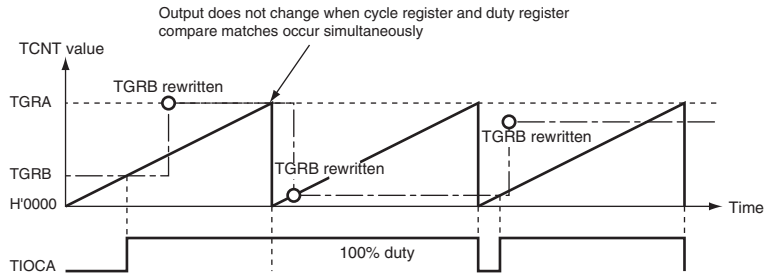
Figure 14.19 Example of PWM Mode Operation (2)

Figure 14.20 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

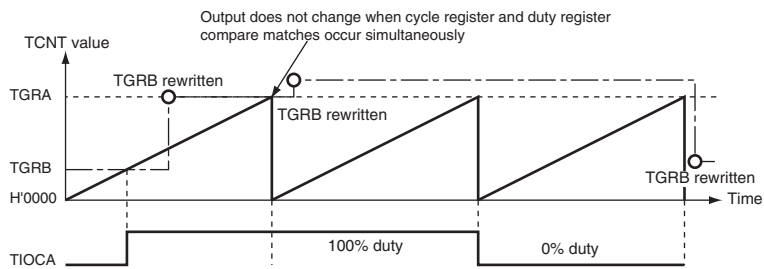
(1) When the value of the duty register is larger than that of the cycle register



(2) When the values of the duty and cycle registers are identical



(3) When the value of the duty register is made larger than that of the cycle register, after the values of the duty and cycle registers are set identical



(4) When the value of the duty register is made smaller than that of TCNT, before duty register compare match occurs

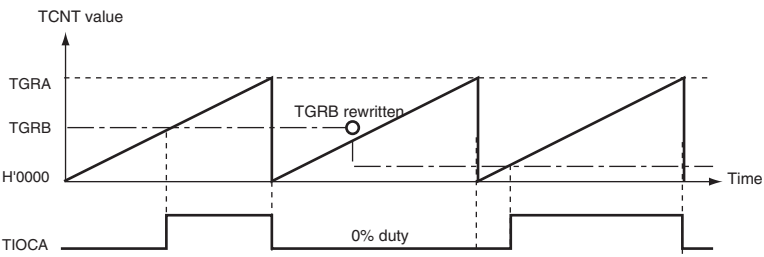


Figure 14.20 Example of PWM Mode Operation (3)

14.6 Interrupt Sources

There are two kinds of TPU interrupt source; TGR input capture/compare match and TCNT overflow. Each interrupt source has its own status flag and enable/disable bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt source is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0.

Mask levels within a channel can be changed by the interrupt controller. For details, see section 4, Interrupt Controller.

Table 14.13 lists the TPU interrupt sources.

Table 14.13 TPU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	Priority
1	TGI1A	TGRA_1 input capture/compare match	TGFA_1	High
	TGI1B	TGRB_1 input capture/compare match	TGFB_1	
	TCI1V	TCNT_1 overflow	TCFV_1	
2	TGI2A	TGRA_2 input capture/compare match	TGFA_2	
	TGI2B	TGRB_2 input capture/compare match	TGFB_2	
	TCI2V	TCNT_2 overflow	TCFV_2	Low

(1) Input Capture/Compare Match Interrupt

An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has a total of four input capture/compare match interrupts, two for each channel.

(2) Overflow Interrupt

An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The TPU has a total of two overflow interrupts, one for each channel.

14.7 Operation Timing

14.7.1 Input/Output Timing

(1) TCNT Count Timing

Figure 14.21 shows TCNT count timing in internal clock operation, and figure 14.22 shows TCNT count timing in external clock operation.

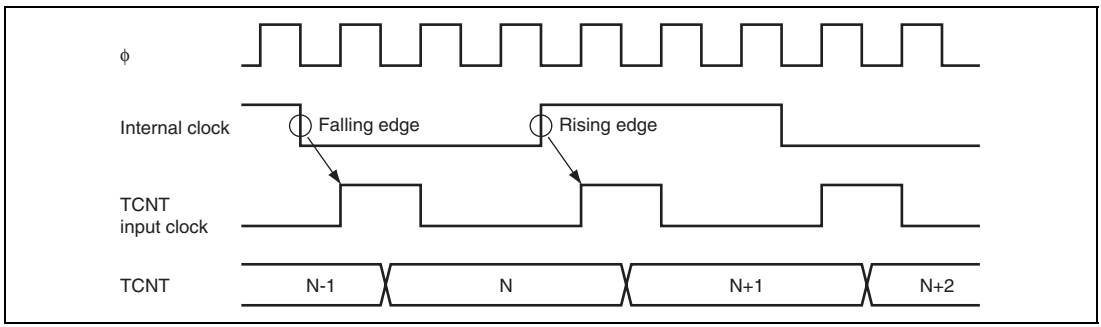


Figure 14.21 Count Timing in Internal Clock Operation

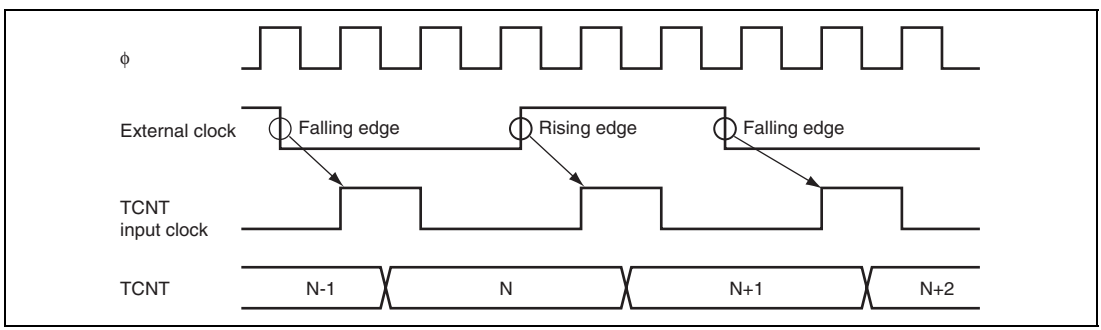


Figure 14.22 Count Timing in External Clock Operation

(2) Output Compare Output Timing

A compare match signal is generated in the last state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin (TIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 14.23 shows output compare output timing.

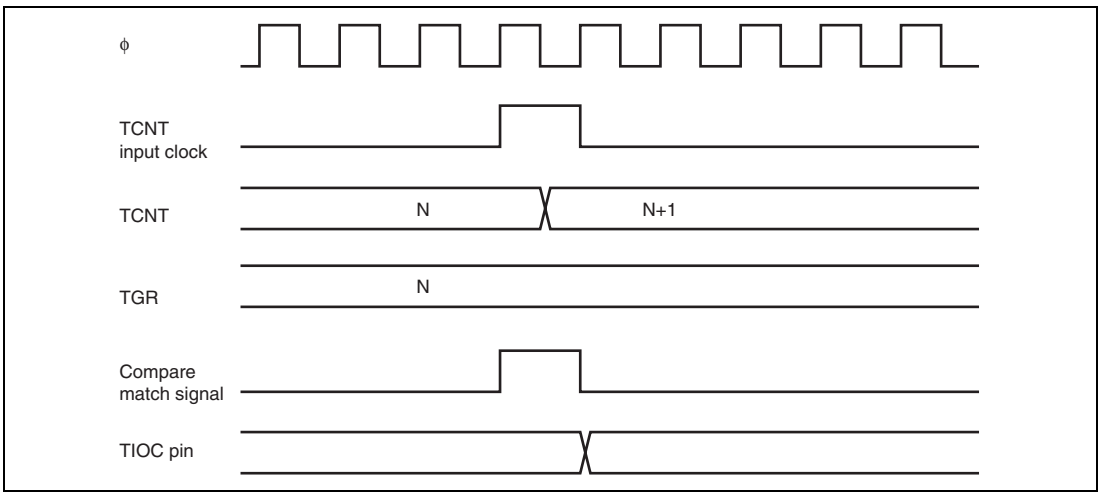


Figure 14.23 Output Compare Output Timing

(3) Input Capture Signal Timing

Figure 14.24 shows input capture signal timing.

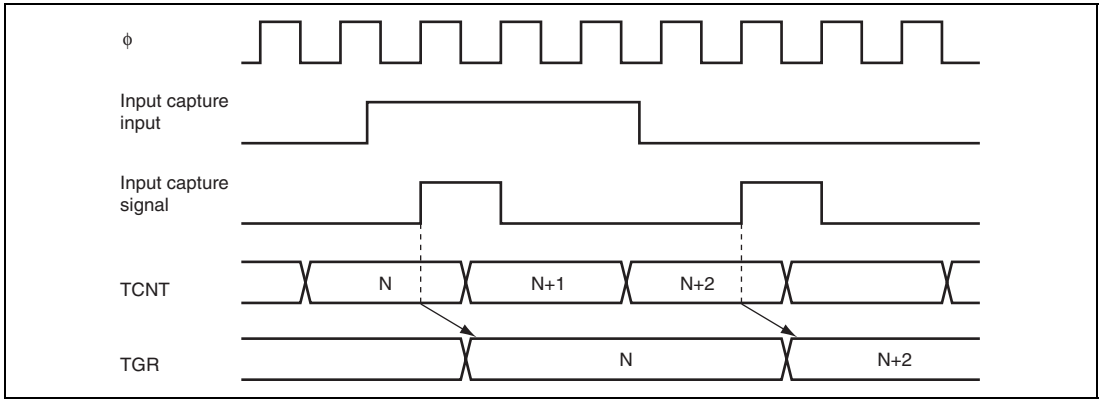


Figure 14.24 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 14.25 shows the timing when counter clearing on compare match is specified, and figure 14.26 shows the timing when counter clearing on input capture is specified.

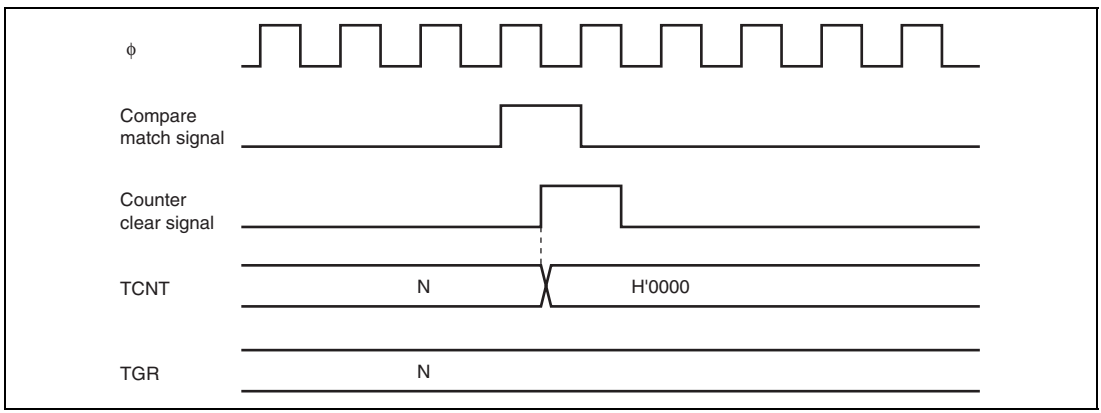


Figure 14.25 Counter Clear Timing (Compare Match)

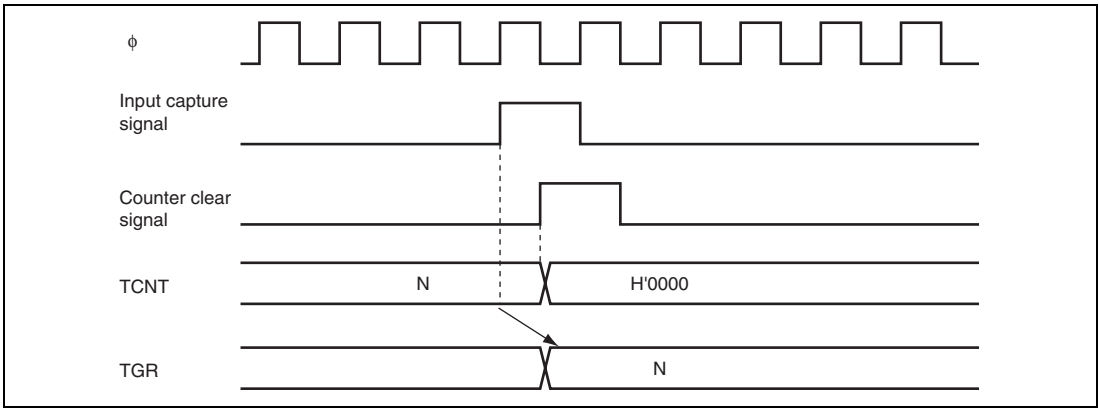


Figure 14.26 Counter Clear Timing (Input Capture)

14.7.2 Interrupt Signal Timing

(1) TGF Flag Setting Timing in Case of Compare Match

Figure 14.27 shows the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

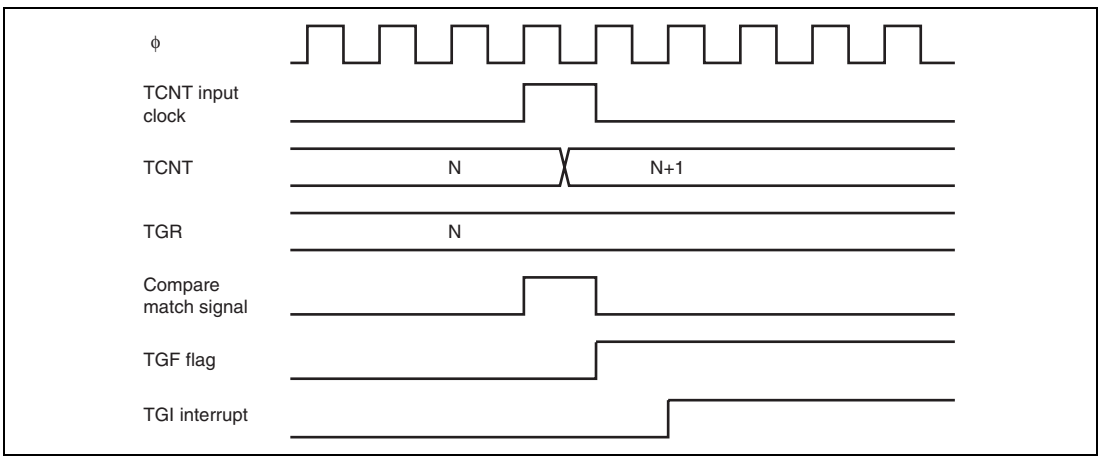


Figure 14.27 TGI Interrupt Timing (Compare Match)

(2) TGF Flag Setting Timing in Case of Input Capture

Figure 14.28 shows the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

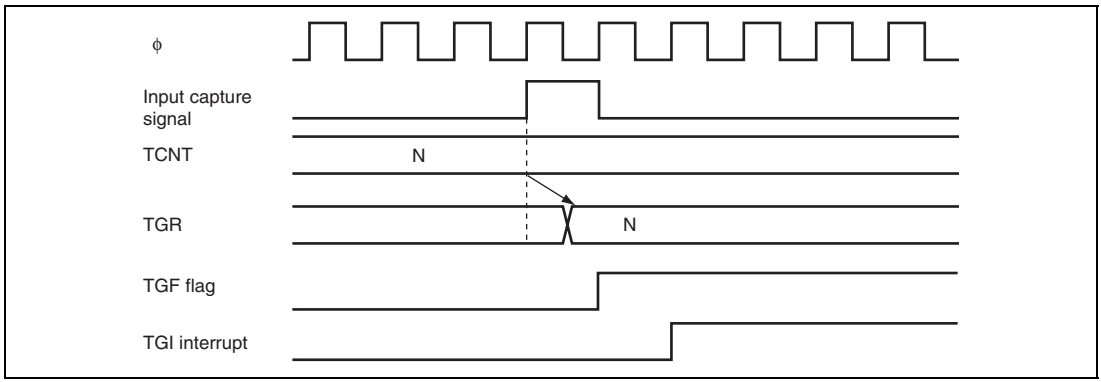


Figure 14.28 TGI Interrupt Timing (Input Capture)

(3) TCFV Flag Setting Timing

Figure 14.29 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

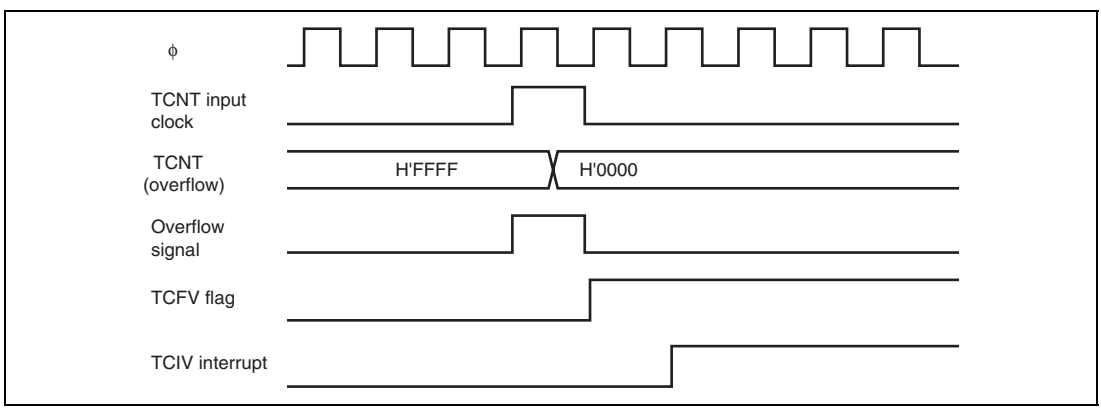


Figure 14.29 TCIV Interrupt Setting Timing

(4) Status Flag Clearing Timing

After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. Figure 14.30 shows the timing for status flag clearing by the CPU.

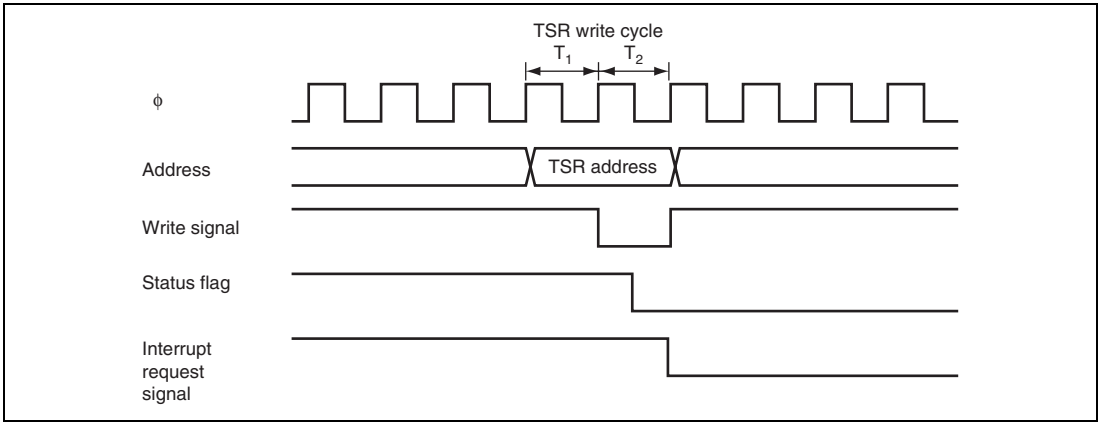


Figure 14.30 Timing for Status Flag Clearing by CPU

14.8 Usage Notes

14.8.1 Module Standby Function Setting

TPU operation can be disabled or enabled using the clock stop register. The initial setting is for the TPU to operate. Register access is enabled by clearing the module standby function. For details, refer to section 6.4, Module Standby Function.

14.8.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The TPU will not operate properly at narrower pulse widths.

14.8.3 Caution on Period Setting

When counter clearing on compare match is set, TCNT is cleared in the last state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{\phi}{(N + 1)}$$

Where f: Counter frequency
ϕ: Operating frequency
N: TGR set value

14.8.4 Contention between TCNT Write and Clear Operation

If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes priority and the TCNT write is not performed.

Figure 14.31 shows the timing in this case.

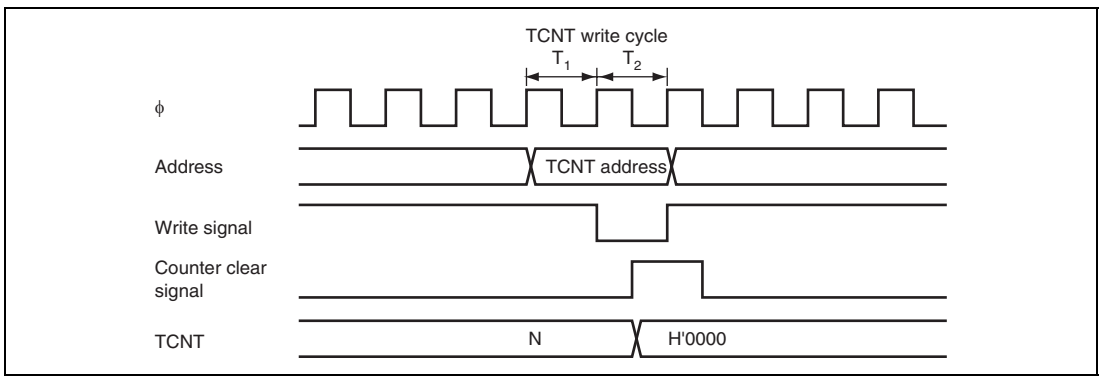


Figure 14.31 Contention between TCNT Write and Clear Operation

14.8.5 Contention between TCNT Write and Increment Operation

If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes priority and TCNT is not incremented.

Figure 14.32 shows the timing in this case.

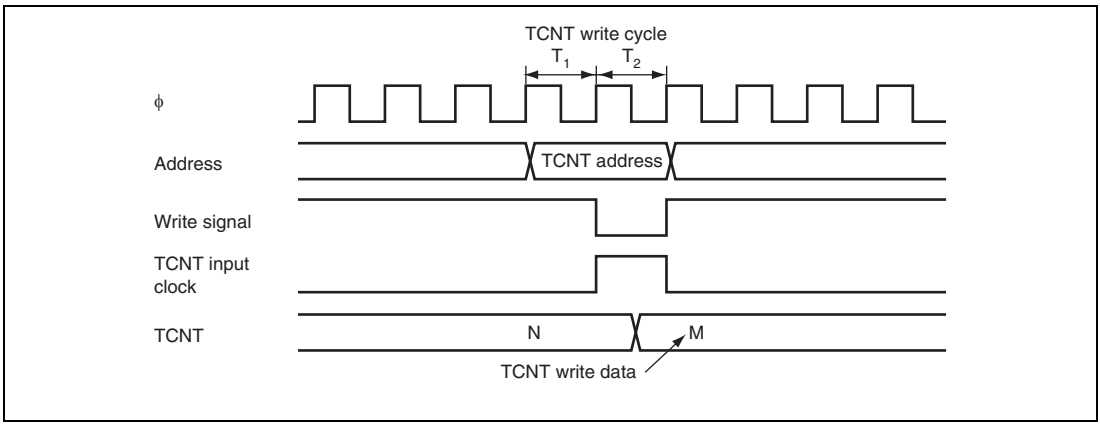


Figure 14.32 Contention between TCNT Write and Increment Operation

14.8.6 Contention between TGR Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the TGR write takes priority and the compare match signal is inhibited. A compare match does not occur even if the previous value is written.

Figure 14.33 shows the timing in this case.

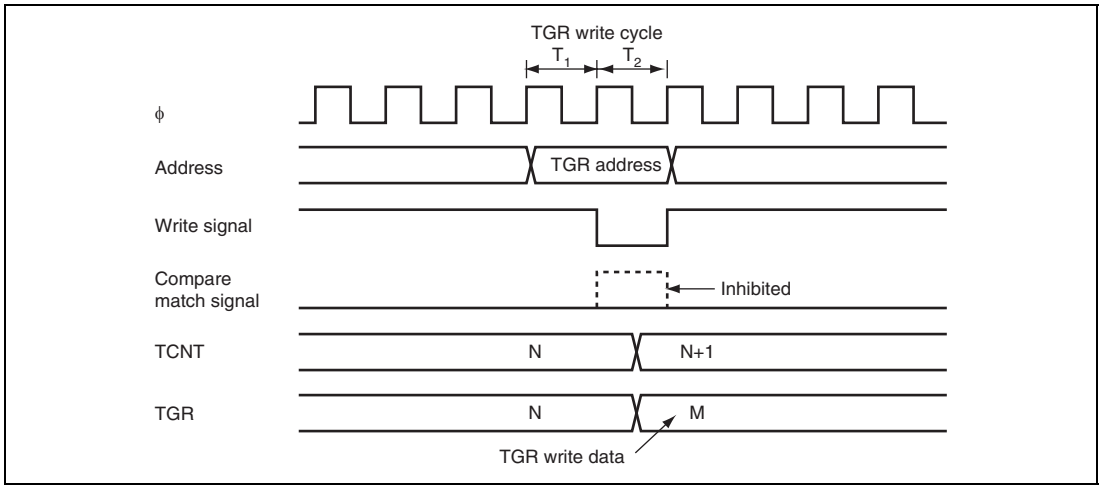


Figure 14.33 Contention between TGR Write and Compare Match

14.8.7 Contention between TGR Read and Input Capture

If an input capture signal is generated in the T1 state of a TGR read cycle, data that is read will be data after input capture transfer.

Figure 14.34 shows the timing in this case.

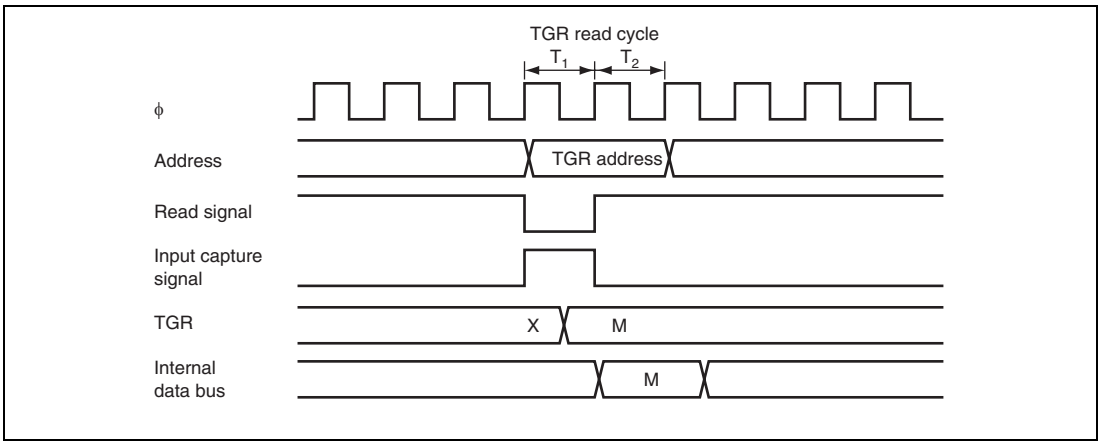


Figure 14.34 Contention between TGR Read and Input Capture

14.8.8 Contention between TGR Write and Input Capture

If an input capture signal is generated in the T2 state of a TGR write cycle, the input capture operation takes priority and the write to TGR is not performed.

Figure 14.35 shows the timing in this case.

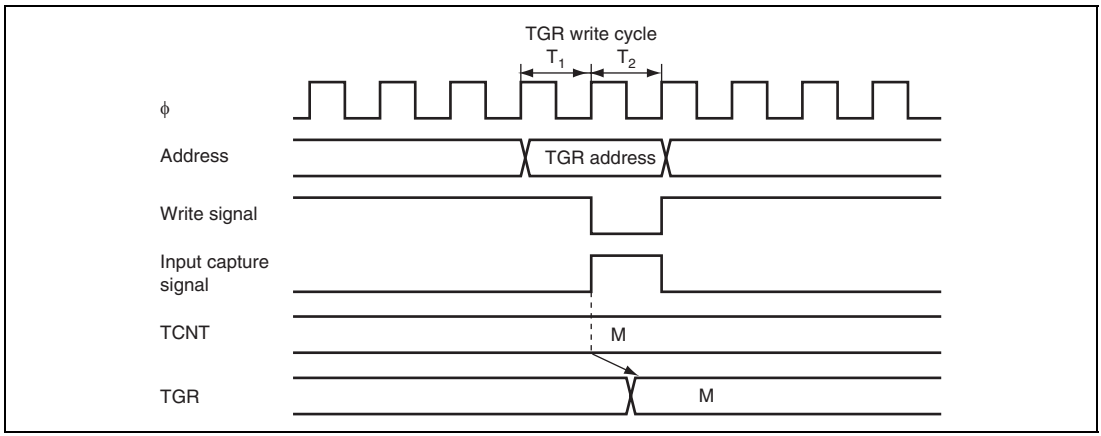


Figure 14.35 Contention between TGR Write and Input Capture

14.8.9 Contention between Overflow and Counter Clearing

If overflow and counter clearing occur simultaneously, the TCFV flag in TSR is not set and TCNT clearing takes priority.

Figure 14.36 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

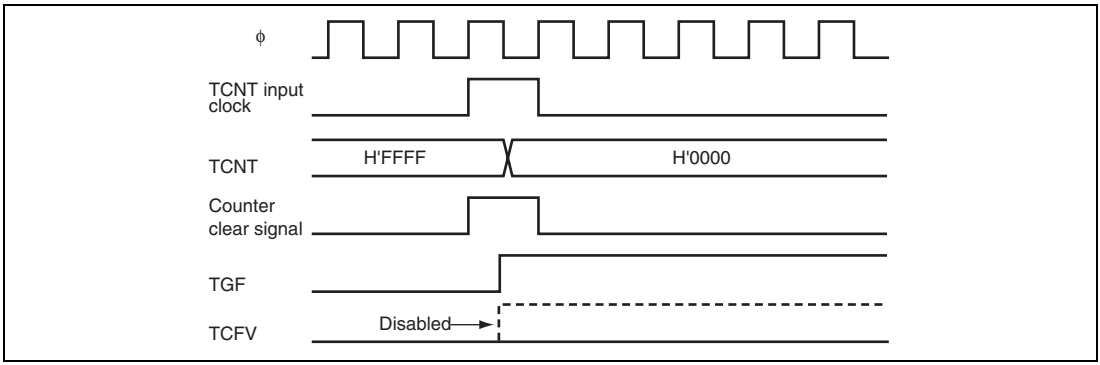


Figure 14.36 Contention between Overflow and Counter Clearing

14.8.10 Contention between TCNT Write and Overflow

If there is an up-count in the T2 state of a TCNT write cycle and overflow occurs, the TCNT write takes priority and the TCFV flag in TSR is not set.

Figure 14.37 shows the operation timing when there is contention between TCNT write and overflow.

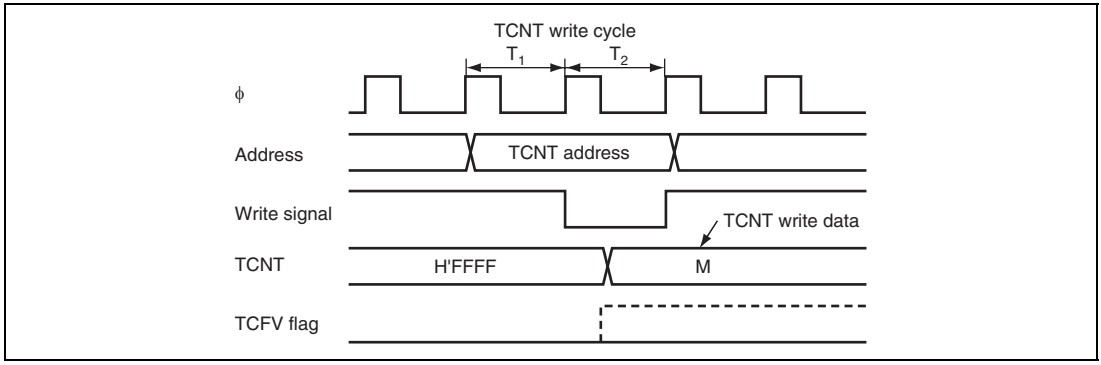


Figure 14.37 Contention between TCNT Write and Overflow

14.8.11 Multiplexing of I/O Pins

The TIOCA1 I/O pin is multiplexed with the TCLKA input pin, the TIOCB1 I/O pin with the TCLKB input pin, and the TIOCA2 I/O pin with the TCLKC input pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

14.8.12 Interrupts when Module Standby Function is Used

If the module standby function is used when an interrupt has been requested, it will not be possible to clear the CPU interrupt source with the interrupt request enabled. Interrupts should therefore be disabled before using the module standby function.

14.8.13 Output Conditions for 0% Duty and 100% Duty

When TGR is rewritten to change the duty in PWM mode, 0% duty or 100% duty is specified depending on the TCT value when rewritten, the TGR value before rewritten, and the TGR value after rewritten. For derails, refer to figure 14.20.

Section 15 Asynchronous Event Counter (AEC)

The asynchronous event counter (AEC) is an event counter that is incremented by external event clock or internal clock input. Figure 15.1 shows a block diagram of the asynchronous event counter.

15.1 Features

- Can count asynchronous events

  Can count external events input asynchronously without regard to the operation of system clocks (ϕ) or subclocks (ϕ_{SUB}).
- Can be used as two-channel independent 8-bit event counter or single-channel independent 16-bit event counter.
- Event/clock input is enabled when IRQAEC goes high or event counter PWM output (IECPWM) goes high.
- Falling edge, rising edge, or both edges can be selected as the detected edge for IRQAEC or event counter PWM output (IECPWM) interrupts. When the asynchronous counter is not used, they can be used as independent interrupts.
- When an event counter PWM is used, event clock input enabling/disabling can be controlled at a constant cycle.
- Selection of four clock sources

  Three internal clocks ($\phi/2$, $\phi/4$, or $\phi/8$) or external event can be selected.
- Falling edge, rising edge, or both edge counting is possible for the AEVL and AEVH pins.
- Counter resetting and halting of the count-up function can be controlled by software.
- Automatic interrupt generation on detection of an event counter overflow
- Module standby mode enables this module to be placed in standby mode independently when it is not in use (For details, refer to section 6.4, Module Standby Function).

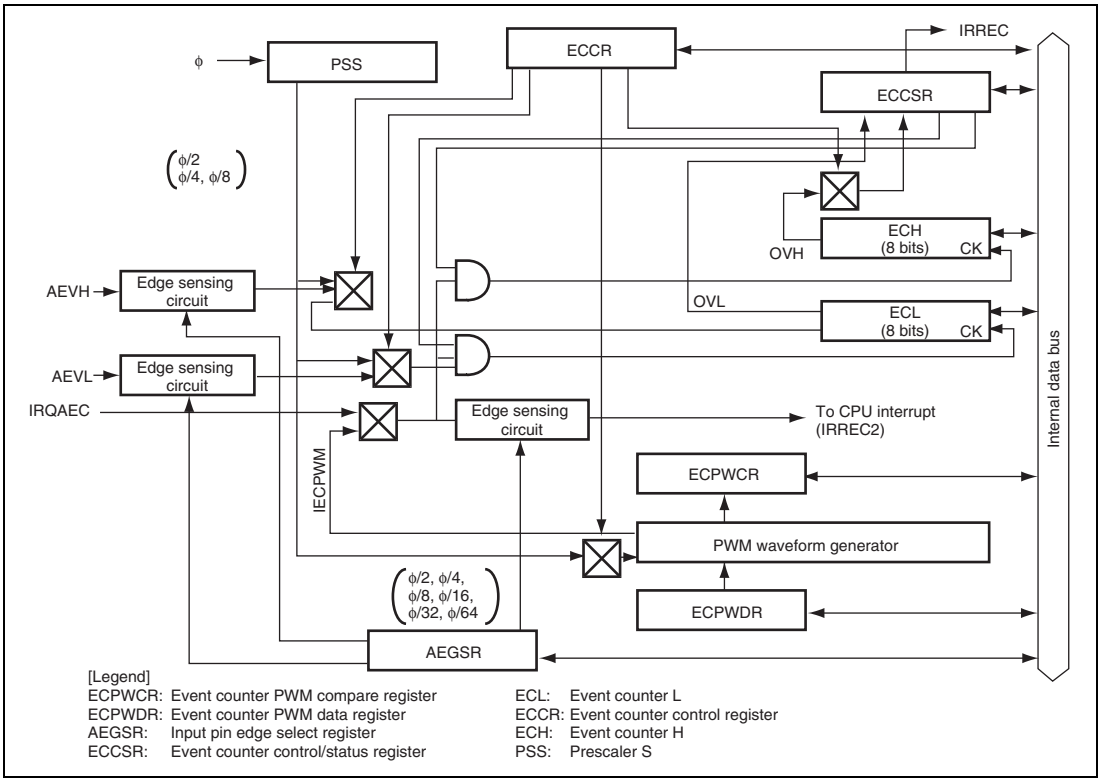


Figure 15.1 Block Diagram of Asynchronous Event Counter

15.2 Input/Output Pins

Table 15.1 shows the pin configuration of the asynchronous event counter.

Table 15.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Asynchronous event input H	AEVH	Input	Event input pin for input to event counter H
Asynchronous event input L	AEVL	Input	Event input pin for input to event counter L
Event input enable interrupt input	IRQAEC	Input	Input pin for interrupt enabling event input

15.3 Register Descriptions

The asynchronous event counter has the following registers.

- Event counter PWM compare register (ECPWCR)
- Event counter PWM data register (ECPWDR)
- Input pin edge select register (AEGSR)
- Event counter control register (ECCR)
- Event counter control/status register (ECCSR)
- Event counter H (ECH)
- Event counter L (ECL)

15.3.1 Event Counter PWM Compare Register (ECPWCR)

ECPWCR sets the one conversion period of the event counter PWM waveform. ECPWCR should be read from or written to in word units.

Bit	Bit Name	Initial Value	R/W	Description
15	ECPWCR15	1	R/W	One Conversion Period of Event Counter PWM Waveform When the ECPWME bit in AEGSR is 1, the event counter PWM is operating and therefore ECPWCR should not be modified. When changing the conversion period, the event counter PWM must be halted by clearing the ECPWME bit in AEGSR to 0 before modifying ECPWCR.
14	ECPWCR14	1	R/W	
13	ECPWCR13	1	R/W	
12	ECPWCR12	1	R/W	
11	ECPWCR11	1	R/W	
10	ECPWCR10	1	R/W	
9	ECPWCR9	1	R/W	
8	ECPWCR8	1	R/W	
7	ECPWCR7	1	R/W	
6	ECPWCR6	1	R/W	
5	ECPWCR5	1	R/W	
4	ECPWCR4	1	R/W	
3	ECPWCR3	1	R/W	
2	ECPWCR2	1	R/W	
1	ECPWCR1	1	R/W	
0	ECPWCR0	1	R/W	

15.3.2 Event Counter PWM Data Register (ECPWDR)

ECPWDR controls data of the event counter PWM waveform generator. ECPWDR should be read from or written to in word units.

Bit	Bit Name	Initial Value	R/W	Description
15	ECPWDR15	0	W	Data Control of Event Counter PWM Waveform Generator When the ECPWME bit in AEGSR is 1, the event counter PWM is operating and therefore ECPWDR should not be modified. When changing the conversion cycle, the event counter PWM must be halted by clearing the ECPWME bit in AEGSR to 0 before modifying ECPWDR. The read value is undefined.
14	ECPWDR14	0	W	
13	ECPWDR13	0	W	
12	ECPWDR12	0	W	
11	ECPWDR11	0	W	
10	ECPWDR10	0	W	
9	ECPWDR9	0	W	
8	ECPWDR8	0	W	
7	ECPWDR7	0	W	
6	ECPWDR6	0	W	
5	ECPWDR5	0	W	
4	ECPWDR4	0	W	
3	ECPWDR3	0	W	
2	ECPWDR2	0	W	
1	ECPWDR1	0	W	
0	ECPWDR0	0	W	

15.3.3 Input Pin Edge Select Register (AEGSR)

AEGSR selects rising, falling, or both edge sensing for the AEVH, AEVL, and IRQAEC pins.

Bit	Bit Name	Initial Value	R/W	Description
7	AHEGS1	0	R/W	AEC Edge Select H
6	AHEGS0	0	R/W	Select rising, falling, or both edge sensing for the AEVH pin. 00: Falling edge on AEVH pin is sensed 01: Rising edge on AEVH pin is sensed 10: Both edges on AEVH pin are sensed 11: Setting prohibited
5	ALEGS1	0	R/W	AEC Edge Select L
4	ALEGS0	0	R/W	Select rising, falling, or both edge sensing for the AEVL pin. 00: Falling edge on AEVL pin is sensed 01: Rising edge on AEVL pin is sensed 10: Both edges on AEVL pin are sensed 11: Setting prohibited
3	AIEGS1	0	R/W	IRQAEC Edge Select
2	AIEGS0	0	R/W	Select rising, falling, or both edge sensing for the IRQAEC pin. 00: Falling edge on IRQAEC pin is sensed 01: Rising edge on IRQAEC pin is sensed 10: Both edges on IRQAEC pin are sensed 11: Setting prohibited
1	ECPWME	0	R/W	Event Counter PWM Enable Controls operation of event counter PWM and selection of IRQAEC. 0: AEC PWM halted, IRQAEC selected 1: AEC PWM enabled, IRQAEC not selected
0	—	0	R/W	Reserved This bit can be read from or written to. However, this bit should not be set to 1.

15.3.4 Event Counter Control Register (ECCR)

ECCR controls the counter input clock and IRQAEC/IECPWM.

Bit	Bit Name	Initial Value	R/W	Description
7	ACKH1	0	R/W	AEC Clock Select H
6	ACKH0	0	R/W	Select the clock used by ECH. 00: AEVH pin input 01: ϕ/2 10: ϕ/4 11: ϕ/8
5	ACKL1	0	R/W	AEC Clock Select L
4	ACKL0	0	R/W	Select the clock used by ECL. 00: AEVL pin input 01: ϕ/2 10: ϕ/4 11: ϕ/8
3	PWCK2	0	R/W	Event Counter PWM Clock Select
2	PWCK1	0	R/W	Select the event counter PWM clock.
1	PWCK0	0	R/W	000: ϕ/2 001: ϕ/4 010: ϕ/8 011: ϕ/16 1x0: ϕ/32 1x1: ϕ/64 When changing the event counter PWM clock, stop the PWM by setting the ECPWME bit in AEGSR to 0, and then change the value of these bits.
0	—	0	R/W	Reserved This bit can be read from or written to. However, this bit should not be set to 1.

[Legend]
x: Don't care

15.3.5 Event Counter Control/Status Register (ECCSR)

ECCSR controls counter overflow detection, counter resetting, and count-up function.

Bit	Bit Name	Initial Value	R/W	Description
7	OVH	0	R/(W)*	Counter Overflow H This is a status flag indicating that ECH has overflowed. [Setting condition] ECH overflows from H'FF to H'00 [Clearing condition] Writing of 0 to bit OVH after reading OVH = 1
6	OVL	0	R/(W)*	Counter Overflow L This is a status flag indicating that ECL has overflowed. [Setting condition] ECL overflows from H'FF to H'00 while CH2 is set to 1 [Clearing condition] Writing of 0 to bit OVL after reading OVL = 1
5	—	0	R/W	Reserved Although this bit is readable/writable, it should not be set to 1.
4	CH2	0	R/W	Channel Select Selects how ECH and ECL event counters are used 0: ECH and ECL are used together as a single-channel 16-bit event counter 1: ECH and ECL are used as two-channel 8-bit event counter
3	CUEH	0	R/W	Count-Up Enable H Enables event clock input to ECH. 0: ECH event clock input is disabled (ECH value is retained) 1: ECH event clock input is enabled

Bit	Bit Name	Initial Value	R/W	Description
2	CUEL	0	R/W	Count-Up Enable L Enables event clock input to ECL. 0: ECL event clock input is disabled (ECL value is retained) 1: ECL event clock input is enabled
1	CRCH	0	R/W	Counter Reset Control H Controls resetting of ECH. 0: ECH is reset 1: ECH reset is cleared and count-up function is enabled
0	CRCL	0	R/W	Counter Reset Control L Controls resetting of ECL. 0: ECL is reset 1: ECL reset is cleared and count-up function is enabled

Note: * Only 0 can be written to clear the flag.

15.3.6 Event Counter H (ECH)

ECH is an 8-bit read-only up-counter that operates as an independent 8-bit event counter. ECH also operates as the upper 8-bit up-counter of a 16-bit event counter configured in combination with ECL. When word access is performed for ECH, the upper 8 bits and lower 8 bits of a 16-bit event counter can be read from in one bus cycle.

Bit	Bit Name	Initial Value	R/W	Description
7	ECH7	0	R	Either the external asynchronous event AEVH pin, ϕ/2, ϕ/4, or ϕ/8, or the overflow signal from lower 8-bit counter ECL can be selected as the input clock source. ECH can be cleared to H'00 by the CRCH bit in ECCSR.
6	ECH6	0	R	
5	ECH5	0	R	
4	ECH4	0	R	
3	ECH3	0	R	
2	ECH2	0	R	
1	ECH1	0	R	
0	ECH0	0	R	

15.3.7 Event Counter L (ECL)

ECL is an 8-bit read-only up-counter that operates as an independent 8-bit event counter. ECL also operates as the upper 8-bit up-counter of a 16-bit event counter configured in combination with ECH. When word access is performed for ECL, the upper 8 bits and lower 8 bits of a 16-bit event counter can be read from in one bus cycle.

Bit	Bit Name	Initial Value	R/W	Description
7	ECL7	0	R	Either the external asynchronous event AEVL pin, ϕ/2, ϕ/4, or ϕ/8 can be selected as the input clock source. ECL can be cleared to H'00 by the CRCL bit in ECCSR.
6	ECL6	0	R	
5	ECL5	0	R	
4	ECL4	0	R	
3	ECL3	0	R	
2	ECL2	0	R	
1	ECL1	0	R	
0	ECL0	0	R	

15.4 Operation

15.4.1 16-Bit Counter Operation

When bit CH2 is cleared to 0 in ECCSR, ECH and ECL operate as a 16-bit event counter.

Any of four input clock sources—ϕ/2, ϕ/4, ϕ/8, or AEVL pin input—can be selected by means of bits ACKL1 and ACKL0 in ECCR. When AEVL pin input is selected, input sensing is selected with bits ALEGS1 and ALEGS0.

Note that the input clock is enabled when IRQAEC is high or IECPWM is high. When IRQAEC is low or IECPWM is low, the input clock is not input to the counter, which therefore does not operate. Figure 15.2 shows the software procedure when ECH and ECL are used as a 16-bit event counter.

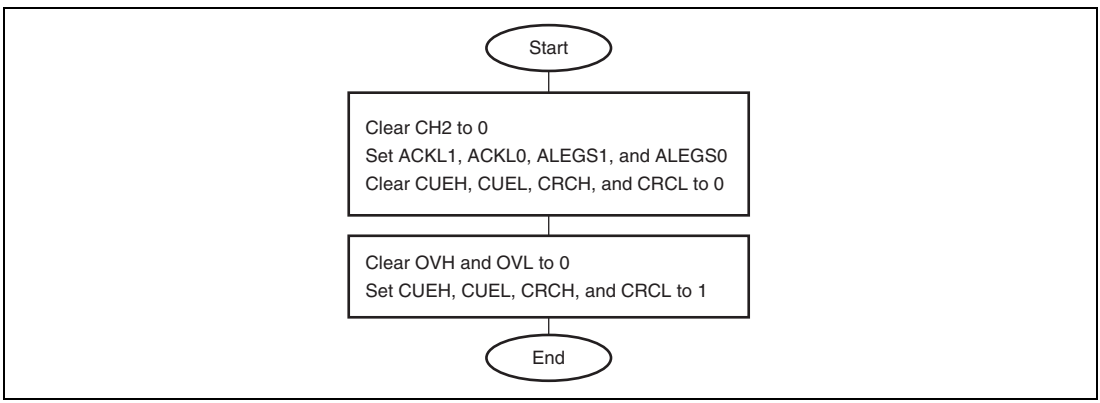


Figure 15.2 Software Procedure when Using ECH and ECL as 16-Bit Event Counter

As CH2 is cleared to 0 by a reset, ECH and ECL operate as a 16-bit event counter after a reset, and as ACKL1 and ACKL0 are cleared to B'00, the operating clock is asynchronous event input from the AEVL pin (using falling edge sensing).

When the next clock is input after the count value reaches H'FF in both ECH and ECL, ECH and ECL overflow from H'FFFF to H'0000, the OVH flag is set to 1 in ECCSR, the ECH and ECL count values each return to H'00, and counting up is restarted. When an overflow occurs, the IRREC bit is set to 1 in IRR2. If the IENEC bit in IENR2 is 1 at this time, an interrupt request is sent to the CPU.

15.4.2 8-Bit Counter Operation

When bit CH2 is set to 1 in ECCSR, ECH and ECL operate as independent 8-bit event counters.

$\phi/2$, $\phi/4$, $\phi/8$, or AEVH pin input can be selected as the input clock source for ECH by means of bits ACKH1 and ACKH0 in ECCR, and $\phi/2$, $\phi/4$, $\phi/8$, or AEVL pin input can be selected as the input clock source for ECL by means of bits ACKL1 and ACKL0 in ECCR. Input sensing is selected with bits AHEGS1 and AHEGS0 when AEVH pin input is selected, and with bits ALEGS1 and ALEGS0 when AEVL pin input is selected.

Note that the input clock is enabled when IRQAEC is high or IECPWM is high. When IRQAEC is low or IECPWM is low, the input clock is not input to the counter, which therefore does not operate. Figure 15.3 shows the software procedure when ECH and ECL are used as 8-bit event counters.

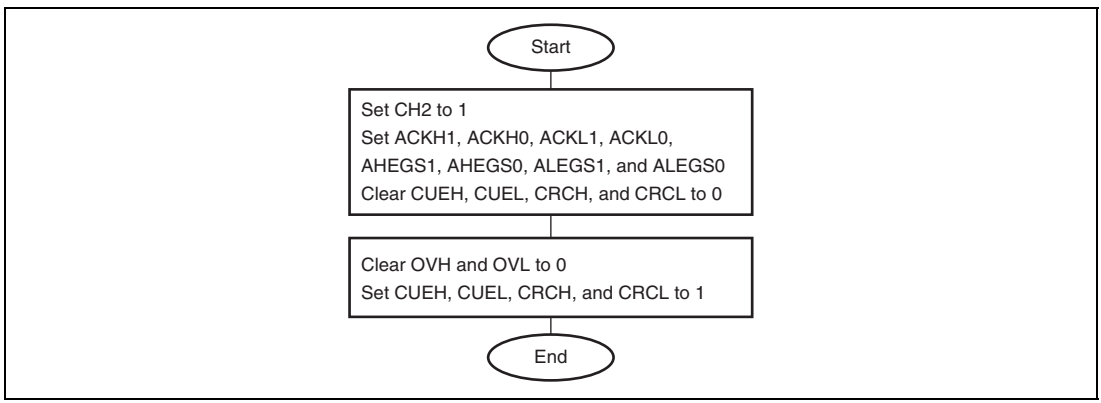


Figure 15.3 Software Procedure when Using ECH and ECL as 8-Bit Event Counters

When the next clock is input after the ECH count value reaches H'FF, ECH overflows, the OVH flag is set to 1 in ECCSR, the ECH count value returns to H'00, and counting up is restarted. Similarly, when the next clock is input after the ECL count value reaches H'FF, ECL overflows, the OVL flag is set to 1 in ECCSR, the ECL count value returns to H'00, and counting up is restarted. When an overflow occurs, the IRREC bit is set to 1 in IRR2. If the IENEC bit in IENR2 is 1 at this time, an interrupt request is sent to the CPU.

15.4.3 IRQAEC Operation

When the ECPWME bit in AEGSR is 0, the ECH and ECL input clocks are enabled when IRQAEC goes high. When IRQAEC goes low, the input clocks are not input to the counters, and so ECH and ECL do not count. ECH and ECL count operations can therefore be controlled from outside by controlling IRQAEC. In this case, ECH and ECL cannot be controlled individually.

IRQAEC can also operate as an interrupt source.

Interrupt enabling is controlled by IENEC2 in IENR1. When an IRQAEC interrupt is generated, IRR1 interrupt request flag IRREC2 is set to 1. If IENEC2 in IENR1 is set to 1 at this time, an interrupt request is sent to the CPU.

Rising, falling, or both edge sensing can be selected for the IRQAEC input pin with bits AIAGS1 and AIAGS0 in AEGSR.

15.4.4 Event Counter PWM Operation

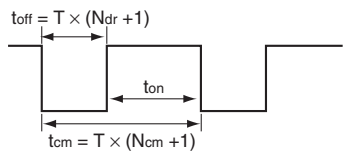
When the ECPWME bit in AEGSR is 1, the ECH and ECL input clocks are enabled when event counter PWM output (IECPWM) is high. When IECPWM is low, the input clocks are not input to the counters, and so ECH and ECL do not count. ECH and ECL count operations can therefore be controlled cyclically from outside by controlling event counter PWM. In this case, ECH and ECL cannot be controlled individually.

IECPWM can also operate as an interrupt source.

Interrupt enabling is controlled by IENEC2 in IENR1. When an IECPWM interrupt is generated, IRR1 interrupt request flag IRREC2 is set to 1. If IENEC2 in IENR1 is set to 1 at this time, an interrupt request is sent to the CPU.

Rising, falling, or both edge detection can be selected for IECPWM interrupt sensing with bits AIAGS1 and AIAGS0 in AEGSR.

Figure 15.4 and table 15.2 show examples of event counter PWM operation.



ton: Clock input enable time
toff: Clock input disable time
tcm: One conversion period
T: ECPWM input clock cycle
Ndr: Value of ECPWDR
Fixed low when Ndr =H'FFFF
Ncm: Value of ECPWCR

Figure 15.4 Event Counter Operation Waveform

Note: Ndr and Ncm above must be set so that Ndr < Ncm. If the settings do not satisfy this condition, the output of the event counter PWM is fixed low.

Table 15.2 Examples of Event Counter PWM Operation

Conditions: fosc = 4 MHz, fϕ = 4 MHz, high-speed active mode, ECPWCR value (Ncm) = H'7A11, ECPWDR value (Ndr) = H'16E3

Clock Source Selection	Clock Source Cycle (T)*	ECPWCR Value (Ncm)	ECPWDR Value (Ndr)	toff = T × (Ndr + 1)	tcm = T × (Ncm + 1)	ton = tcm – toff
ϕ/2	0.5 µs	H'7A11	H'16E3	2.93 ms	15.625 ms	12.695 ms
ϕ/4	1 µs	D'31249	D'5859	5.86 ms	31.25 ms	25.39 ms
ϕ/8	2 µs			11.72 ms	62.5 ms	50.78 ms
ϕ/16	4 µs			23.44 ms	125.0 ms	101.56 ms
ϕ/32	8 µs			46.88 ms	250.0 ms	203.12 ms
ϕ/64	16 µs			93.76 ms	500.0 ms	406.24 ms

Note: * toff minimum width

15.4.5 Operation of Clock Input Enable/Disable Function

The clock input to the event counter can be controlled by the IRQAEC pin when ECPWME in AEGSR is 0, and by the event counter PWM output, IECPWM when ECPWME in AEGSR is 1. As this function forcibly terminates the clock input by each signal, a maximum error of one count will occur depending on the IRQAEC or IECPWM timing. Figure 15.5 shows an example of the operation.

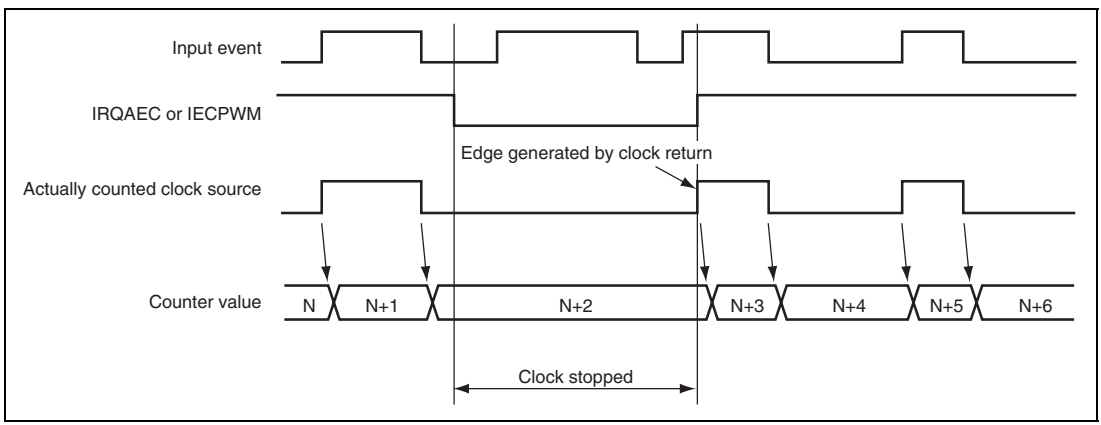


Figure 15.5 Example of Clock Control Operation

15.5 Operating States of Asynchronous Event Counter

The operating states of the asynchronous event counter are shown in table 15.3.

Table 15.3 Operating States of Asynchronous Event Counter

Operating Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby	Module Standby
AEGSR	Reset	Functioning	Functioning	Retained*1	Functioning	Functioning	Retained*1	Retained
ECCR	Reset	Functioning	Functioning	Retained*1	Functioning	Functioning	Retained*1	Retained
ECCSR	Reset	Functioning	Functioning	Retained*1	Functioning	Functioning	Retained*1	Retained
ECH	Reset	Functioning	Functioning	Functioning*1*2	Functioning*2	Functioning*2	Functioning*1*2	Halted
ECL	Reset	Functioning	Functioning	Functioning*1*2	Functioning*2	Functioning*2	Functioning*1*2	Halted
IRQAEC	Reset	Functioning	Functioning	Retained*3	Functioning	Functioning	Retained*3	Retained*4
Event counter PWM	Reset	Functioning	Functioning	Retained	Retained	Retained	Retained	Retained

Notes: 1. When an asynchronous external event is input, the counter increments. In addition, when an overflow occurs, an interrupt is requested.
2. Functions when asynchronous external events are selected; halted and retained otherwise.
3. Clock control by IRQAEC operates, but interrupts do not.
4. As the clock is stopped in module standby mode, IRQAEC has no effect.

15.6 Usage Notes

1. When reading the values in ECH and ECL, first clear bits CUEH and CUEL to 0 in ECCSR in 8-bit mode and clear bit CUEL to 0 in 16-bit mode to prevent asynchronous event input to the counter. The correct value will not be returned if the event counter increments while being read.
2. For input to the AEVH and AEVL pins, use a clock with a frequency of up to 4.2 MHz within the range from 1.8 to 3.6 V and up to 10 MHz within the range from 2.7 to 3.6 V. For the high and low widths of the clock, see section 25, Electrical Characteristics. The duty cycle is arbitrary.

Table 15.4 shows a maximum clock frequency.

Table 15.4 Maximum Clock Frequency

Mode		Maximum Clock Frequency Input to AEVH/AEVL Pin
Active (high-speed), sleep (high-speed)		10 MHz
Active (medium-speed), sleep (medium-speed)	(ϕ_{osc}/8)	2 f_{osc}
	(ϕ_{osc}/16)	f_{osc}
	(ϕ_{osc}/32)	1/2 f_{osc}
f_{osc} = 2 MHz to 4 MHz	(ϕ_{osc}/64)	1/4 f_{osc}
Watch, subactive, subsleep, standby	(ϕ_w/2)	1000 kHz
	(ϕ_w/4)	500 kHz
ϕ_w = 32.768 kHz or 38.4 kHz	(ϕ_w/8)	250 kHz

3. When AEC uses with 16-bit mode, set CUEH in ECCSR to 1 first, set CRCH in ECCSR to 1 second, or set both CUEH and CRCH to 1 at same time before clock input. When AEC is operating on 16-bit mode, do not change CUEH. Otherwise, ECH will be miscounted up.
4. When ECPWME in AEGSR is 1, the event counter PWM is operating and therefore ECPWCR and ECPWDR should not be modified.

   When changing the data, clear the ECPWME bit in AEGSR to 0 (halt the event counter PWM) before modifying these registers.
5. The event counter PWM data register and event counter PWM compare register must be set so that event counter PWM data register < event counter PWM compare register. If the settings do not satisfy this condition, do not set ECPWME to 1 in AEGSR.

6. As synchronization is established internally when an IRQAEC interrupt is generated, a maximum error of 1 t_{cyc} will occur between clock halting and interrupt acceptance.

Section 16 Watchdog Timer

This LSI incorporates the watchdog timer (WDT). The WDT is an 8-bit timer that can generate an internal reset signal if a system crash prevents the CPU from writing to the timer counter, thus allowing it to overflow.

When this watchdog timer function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is generated each time the counter overflows.

16.1 Features

The WDT features are described below.

- Selectable from eleven counter input clocks
  Ten internal clock sources (ϕ/64, ϕ/128, ϕ/256, ϕ/512, ϕ/1024, ϕ/2048, ϕ/4096, ϕ/8192, ϕ_w/16, and ϕ_w/256) or the on-chip watchdog timer oscillator can be selected as the timer-counter clock.
- Watchdog timer mode
  If the counter overflows, this LSI is internally reset.
- Interval timer mode
  If the counter overflows, an interval timer interrupt is generated.
- Use of module standby mode enables this module to be placed in standby mode independently when not used. (The WDT is operating as the initial value. For details, refer to section 6.4, Module Standby Function.)

Figure 16.1 shows a block diagram of the WDT.

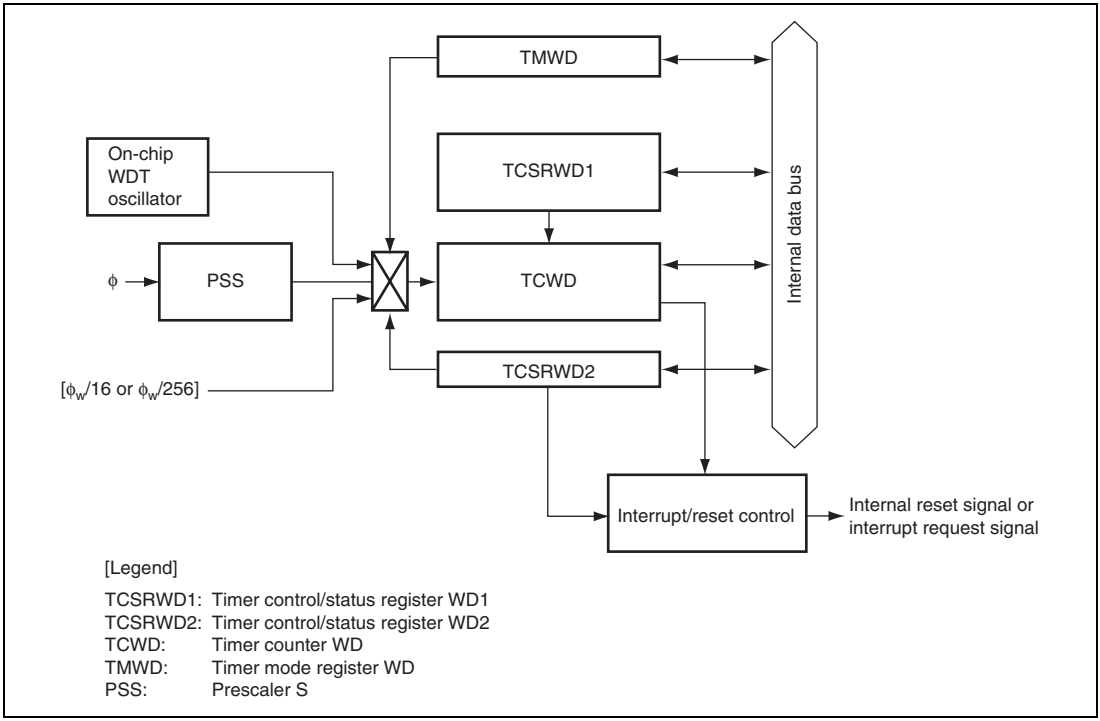


Figure 16.1 Block Diagram of Watchdog Timer

16.2 Register Descriptions

The watchdog timer has the following registers.

- Timer control/status register WD1 (TCSRWD1)
- Timer control/status register WD2 (TCSRWD2)
- Timer counter WD (TCWD)
- Timer mode register WD (TMWD)

16.2.1 Timer Control/Status Register WD1 (TCSRWD1)

TCSRWD1 performs the TCSRWD1 and TCWD write control. TCSRWD1 also controls the watchdog timer operation and indicates the operating state. TCSRWD1 must be rewritten by using the MOV instruction. The bit manipulation instruction cannot be used to change the setting value.

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 value of 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 to 1. When writing data to this bit, the write value for bit 7 must be 0.
5	B4WI	1	R/W	Bit 4 Write Inhibit The TCSRWE bit can be written only when the write value of the B4WI bit is 0. This bit is always read as 1.
4	TCSRWE	0	R/W	Timer Control/Status Register WD Write Enable The WDON and WRST bits can be written when the TCSRWE bit is set to 1. When writing data to this bit, the write value for bit 5 must be 0.
3	B2WI	1	R/W	Bit 2 Write Inhibit The WDON bit can be written only when the write value of the B2WI bit is 0. This bit is always read as 1.

Bit	Bit Name	Initial Value	R/W	Description
2	WDON	1	R/W	Watchdog Timer On TCWD starts counting up when the WDON bit is set to 1 and halts when the WDON bit is cleared to 0. [Setting condition] • When 1 is written to the WDON bit and 0 to the B2WI bit while the TCSRWE bit is 1 • Reset by $\overline{\text{RES}}$ pin [Clearing condition] • When 0 is written to the WDON bit and 0 to the B2WI bit while the TCSRWE bit is 1
1	B0WI	1	R/W	Bit 0 Write Inhibit The WRST bit can be written only when the write value of the B0WI bit is 0. This bit is always read as 1.
0	WRST	0	R/W	Watchdog Timer Reset Indicates whether a reset caused by the watchdog timer is generated. This bit is not cleared by a reset caused by the watchdog timer. [Setting condition] When TCWD overflows and an internal reset signal is generated [Clearing conditions] • Reset by $\overline{\text{RES}}$ pin • When 0 is written to the WRST bit and 0 to the B0WI bit while the TCSRWE bit is 1

16.2.2 Timer Control/Status Register WD2 (TCSRWD2)

TCSRWD2 performs the TCSRWD2 write control, mode switching, and interrupt control. TCSRWD2 must be rewritten by using the MOV instruction. The bit manipulation instruction cannot be used to change the setting value.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*[1]	Overflow Flag Indicates that TCWD has overflowed (changes from H'FF to H'00). [Setting condition] TCWD overflows (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, this bit is cleared automatically by the internal reset after it has been set. [Clearing condition] • TCSRWD2 is read when OVF = 1, then 0 is written to OVF
6	B5WI	1	R/(W)*[2]	Bit 5 Write Inhibit The WT/$\overline{IT}$ bit can be written only when the write value of the B5WI bit is 0. This bit is always read as 1.
5	WT/$\overline{IT}$	0	R/(W)*[3]	Timer Mode Select Selects whether the WDT is used as a watchdog timer or interval timer. 0: Watchdog timer mode 1: Interval timer mode
4	B3WI	1	R/(W)*[2]	Bit 3 Write Inhibit The IEOVF bit can be written only when the write value of the B3WI bit is 0. This bit is always read as 1.
3	IEOVF	0	R/(W)*[3]	Overflow Interrupt Enable Enables or disables an overflow interrupt request in interval timer mode. 0: Disables an overflow interrupt 1: Enables an overflow interrupt

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	—	All 1	—	Reserved These bits are always read as 1.

Notes: 1. Only 0 can be written to clear the flag.
2. Write operation is necessary because this bit controls data writing to other bit. This bit is always read as 1.
3. Writing is possible only when the write conditions are satisfied.

16.2.3 Timer Counter WD (TCWD)

TCWD is an 8-bit readable/writable up-counter. When TCWD overflows from H'FF to H'00, the internal reset signal is generated and the WRST bit in TCSRWD1 is set to 1. TCWD is initialized to H'00.

16.2.4 Timer Mode Register WD (TMWD)

TMWD selects the input clock.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	CKS3	0	R/W	Clock Select 3 to 0
2	CKS2	0	R/W	Select the clock to be input to TCWD.
1	CKS1	0	R/W	00xx: On-chip WDT oscillator
0	CKS0	0	R/W	0100: Internal clock: counts on $\phi_w/16$
				0101: Internal clock: counts on $\phi_w/256$
				011x: Reserved
				1000: Internal clock: counts on $\phi/64$
				1001: Internal clock: counts on $\phi/128$
				1010: Internal clock: counts on $\phi/256$
				1011: Internal clock: counts on $\phi/512$
				1100: Internal clock: counts on $\phi/1024$
				1101: Internal clock: counts on $\phi/2048$
				1110: Internal clock: counts on $\phi/4096$
				1111: Internal clock: counts on $\phi/8192$
				For the overflow periods of the on-chip WDT oscillator, see section 25, Electrical Characteristics.
				In active (medium-speed), sleep (medium-speed), subactive, and subsleep modes, the 00xx value and the interval timer mode cannot be set simultaneously.
				In subactive and subsleep modes, when the subclock frequency is $\phi_w/8$, the 010x value and the interval timer mode cannot be set simultaneously.

[Legend]

x: Don't care

16.3 Operation

16.3.1 Watchdog Timer Mode

The watchdog timer is provided with an 8-bit up-counter. To use it as the watchdog timer, clear the WT/$\overline{IT}$ bit in TCSRWD2 to 0. (To write the WT/$\overline{IT}$ bit, two write accesses are required.) If 1 is written to the WDON bit and 0 to the B2WI bit simultaneously when the TCSRWE bit in TCSRWD1 is set to 1, TCWD begins counting up. (To operate the watchdog timer, two write accesses to TCSRWD1 are required.) When a clock pulse is input after the TCWD count value has reached H'FF, the watchdog timer overflows and an internal reset signal is generated. The internal reset signal is output for a period of 512 clock cycles of the on-chip watchdog timer oscillator. TCWD is a writable counter, and when a value is set in TCWD, the count-up starts from that value. An overflow period in the range of 1 to 256 input clock cycles can therefore be set, according to the TCWD set value.

Figure 16.2 shows an example of watchdog timer operation.

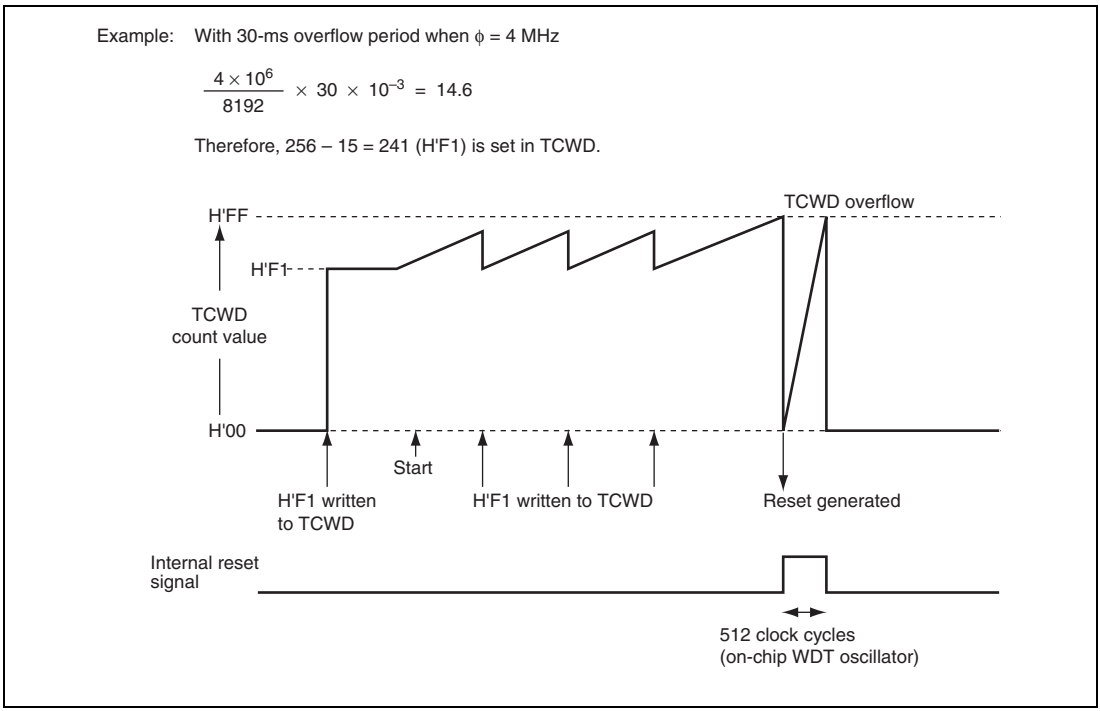


Figure 16.2 Example of Watchdog Timer Operation

16.3.2 Interval Timer Mode

Figure 16.3 shows the operation in interval timer mode. To use the WDT as an interval timer, set the WT/$\overline{IT}$ bit in TCSRWD2 to 1.

When the WDT is used as an interval timer, an interval timer interrupt request is generated each time the TCWD overflows. Therefore, an interval timer interrupt can be generated at intervals.

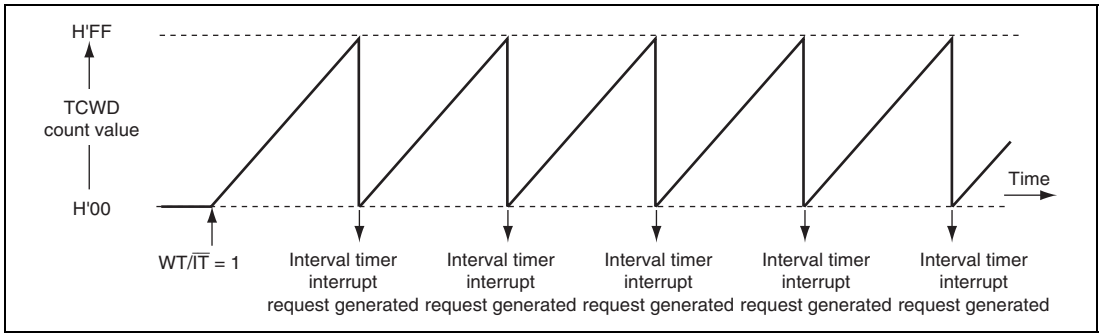


Figure 16.3 Interval Timer Mode Operation

16.3.3 Timing of Overflow Flag (OVF) Setting

Figure 16.4 shows the timing of the OVF flag setting. The OVF flag in TCSRWD2 is set to 1 if TCWD overflows. At the same time, a reset signal is output in watchdog timer mode and an interval timer interrupt is generated in interval timer mode.

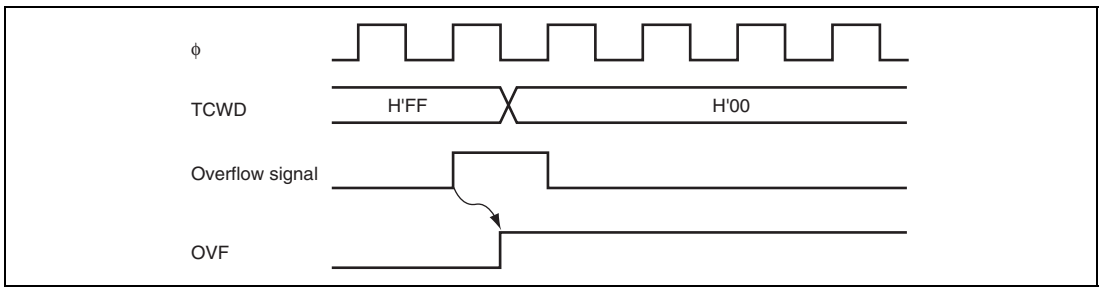


Figure 16.4 Timing of OVF Flag Setting

16.4 Interrupt

During interval timer mode operation, an overflow generates an interval timer interrupt. The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSRWD2. The OVF flag must be cleared to 0 in the interrupt handling routine.

16.5 Usage Notes

16.5.1 Switching between Watchdog Timer Mode and Interval Timer Mode

If modes are switched between watchdog timer and interval timer, while the WDT is operating, an error may occur in the count value. Software must stop the watchdog timer (by clearing the WDON bit to 0) before switching modes.

16.5.2 Module Standby Mode Control

The WDCKSTP bit in CKSTPR2 is valid when the WDON bit in the timer control/status register WD1 (TCSRWD1) is cleared to 0. The WDCKSTP bit can be cleared to 0 while the WDON bit is set to 1 (while the watchdog timer is operating). However, the watchdog timer does not enter module standby mode but continues operating. When the WDON bit is cleared to 0 by software after the watchdog timer stops operating, the WDCKSTP bit is valid at the same time and the watchdog timer enters module standby mode.

16.5.3 Writing to Timer Counter WD (TCWD) with the On-Chip Watchdog Timer Oscillator Selected

When the timer counter WD (TCWD) is written to with the on-chip watchdog timer oscillator selected as the clock to drive the counter, updating of values read from TCWD requires up to (on-chip watchdog timer oscillator overflow time)/256. The watchdog timer does not overflow between writing of the new value to the register and updating of the read values.

Section 17 Serial Communications Interface 3 (SCI3, IrDA)

The serial communications interface 3 (SCI3) can handle both asynchronous and clock synchronous serial communications. The asynchronous method allows the handling of serial data communications with standard asynchronous communications chips such as Universal Asynchronous Receiver/Transmitters (UARTs) and Asynchronous Communications Interface Adapters (ACIAs). A function is also provided for serial communications between processors (multiprocessor communication function) on three channels (SCI3_1, SCI3_2, and SCI3_3). Table 17.1 shows the configuration of the SCI3 channels.

The SCI3_1 can transmit and receive IrDA signals that conform to version 1.0 of the Infrared Data Association (IrDA) standard.

17.1 Features

- Choice of asynchronous or clock synchronous serial communications mode
- Full-duplex communications capability

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

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

  Transmit-end, transmit-data-empty, receive-data-full, overrun error, framing error, and parity error.
- Use of module standby mode enables this module to be placed in standby mode independently when it is not in use (for details, see section 6.4, Module Standby Function).

Asynchronous mode

- Data length: 7, 8, or 5 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Receive error detection: Parity, overrun, and framing errors
- Break detection: Break can be detected by reading the RXD31, RXD32, or RXD33 pin level directly in the case of a framing error

Clock synchronous mode

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

Note: When using this function, do not use the on-chip oscillator for the system clock.

Table 17.1 SCI3 Channel Configuration

Channel	Abbreviation	Pin*[1]	Register*[2]	Register Address
Channel 1	SCI3_1	SCK31	SMR3_1	H'FFFF98
		RXD31	BRR3_1	H'FFFF99
		TXD31	SCR3_1	H'FFFF9A
			TDR3_1	H'FFFF9B
			SSR3_1	H'FFFF9C
			RDR3_1	H'FFFF9D
			RSR3_1	—
			TSR3_1	—
			SEMR	H'FFFFA6
			IrCR	H'FFFFA7
Channel 2	SCI3_2	SCK32	SMR3_2	H'FFFFA8
		RXD32	BRR3_2	H'FFFFA9
		TXD32	SCR3_2	H'FFFFAA
			TDR3_2	H'FFFFAB
			SSR3_2	H'FFFFAC
			RDR3_2	H'FFFFAD
			RSR3_2	—
			TSR3_2	—
Channel 3	SCI3_3	SCK33	SMR3_3	H'FFF088
		RXD33	BRR3_3	H'FFF089
		TXD33	SCR3_3	H'FFF08A
			TDR3_3	H'FFF08B
			SSR3_3	H'FFF08C
			RDR3_3	H'FFF08D
			RSR3_3	—
			TSR3_3	—

Notes: 1. Pin names SCK3, RXD3, and TXD3 are used in the text for all channels, omitting the channel designation.

2. In the text, channel description is omitted for registers and bits.

Figures 17.1 (1), 17.1 (2), and 17.1 (3) show block diagrams of the SCI3_1, SCI3_2, and SCI3_3, respectively.

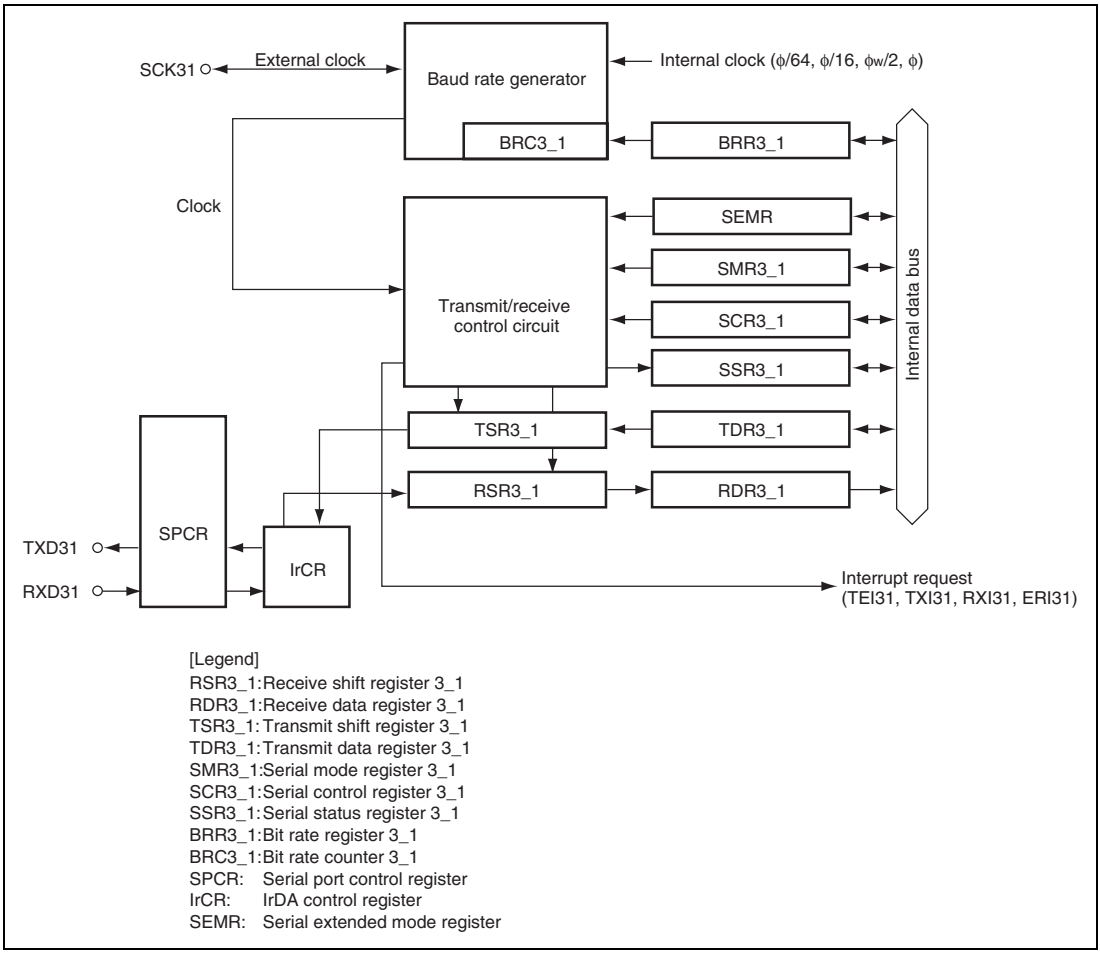
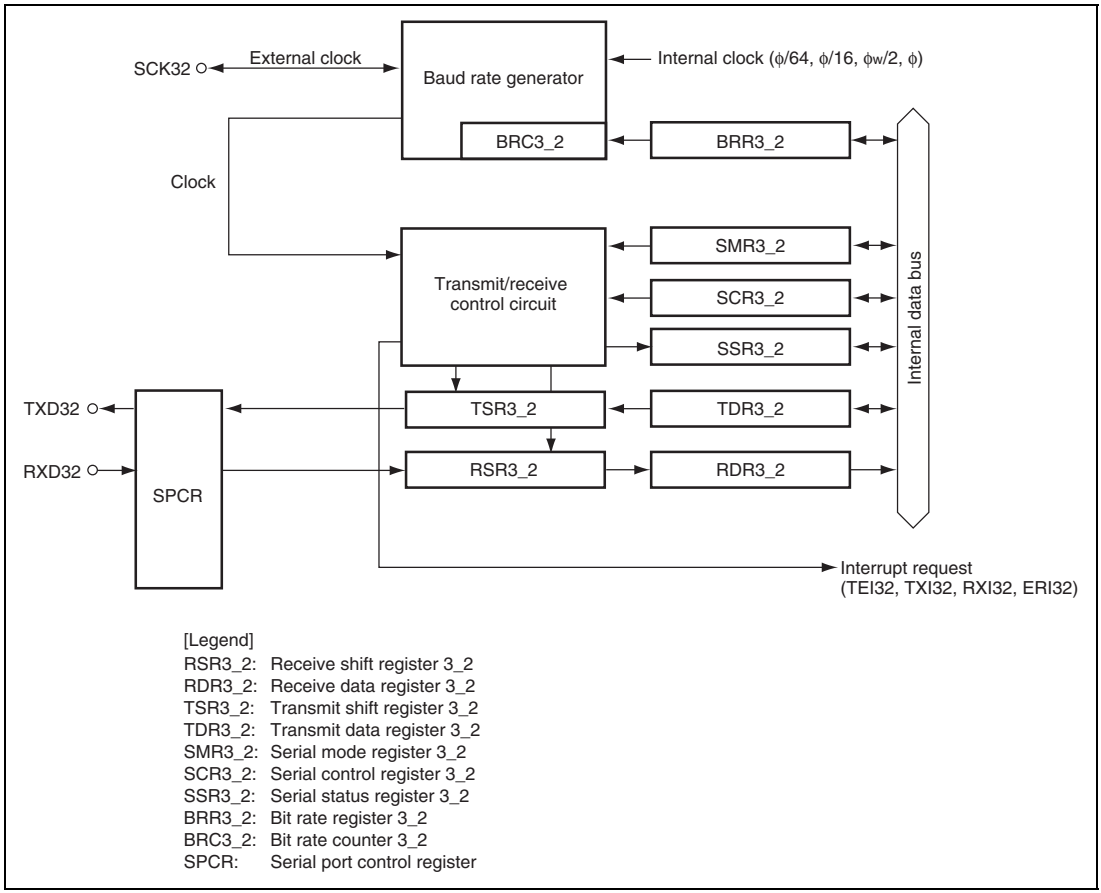


Figure 17.1 (1) Block Diagram of SCI3_1



[Legend]
RSR3_2: Receive shift register 3_2
RDR3_2: Receive data register 3_2
TSR3_2: Transmit shift register 3_2
TDR3_2: Transmit data register 3_2
SMR3_2: Serial mode register 3_2
SCR3_2: Serial control register 3_2
SSR3_2: Serial status register 3_2
BRR3_2: Bit rate register 3_2
BRC3_2: Bit rate counter 3_2
SPCR: Serial port control register

Figure 17.1 (2) Block Diagram of SCI3_2

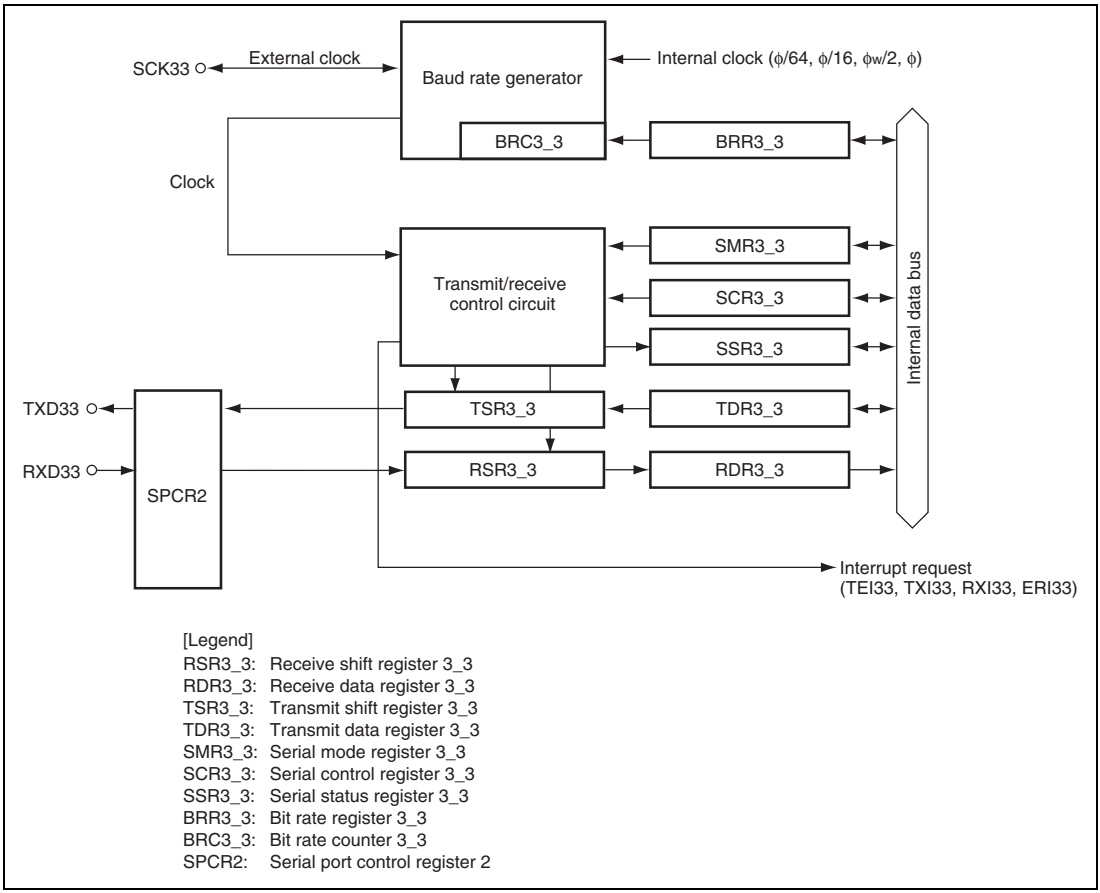


Figure 17.1 (3) Block Diagram of SCI3_3

17.2 Input/Output Pins

Table 17.2 shows the SCI3 pin configuration.

Table 17.2 Pin Configuration

Pin Name	Abbreviation	I/O	Function
SCI3 clock	SCK31, SCK32, SCK33	I/O	SCI3 clock input/output
SCI3 receive data input	RXD31, RXD32, RXD33	Input	SCI3 receive data input
SCI3 transmit data output	TXD31, TXD32, TXD33	Output	SCI3 transmit data output

17.3 Register Descriptions

The SCI3 has the following registers for each channel.

- Receive shift register 3 (RSR3)*
- Receive data register 3 (RDR3)*
- Transmit shift register 3 (TSR3)*
- Transmit data register 3 (TDR3)*
- Serial mode register 3 (SMR3)*
- Serial control register 3 (SCR3)*
- Serial status register 3 (SSR3)*
- Bit rate register 3 (BRR3)*
- Serial port control register (SPCR)
- Serial port control register 2 (SPCR2)
- IrDA control register (IrCR)
- Serial extended mode register (SEMR)

Note: * These register names are abbreviated to RSR, RDR, TSR, TDR, SMR, SCR, SSR, and BRR in the text.

17.3.1 Receive Shift Register (RSR)

RSR is a shift register that receives serial data input from the RXD3 pin and converts it into parallel data. When one byte of data has been received, it is transferred to RDR automatically. RSR cannot be directly accessed by the CPU.

17.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores receive data. When the SCI3 has received one byte of serial data, it transfers the received serial data from RSR to RDR, where it is stored. After this, RSR is receive-enabled. As RSR and RDR function as a double buffer in this way, continuous receive operations are possible. After confirming that the RDRF bit in SSR is set to 1, read RDR only once. RDR cannot be written to by the CPU. RDR is initialized to H'00.

RDR is initialized to H'00 by a reset or in standby mode, watch mode, or module standby mode.

17.3.3 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI3 first transfers transmit data from TDR to TSR automatically, and then sends the data that starts from the LSB to the TXD3 pin. Data transfer from TDR to TSR is not performed if no data has been written to TDR (if the TDRE bit in SSR is set to 1). TSR cannot be directly accessed by the CPU.

17.3.4 Transmit Data Register (TDR)

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

TDR is initialized to H'FF by a reset or in standby mode, watch mode, or module standby mode.

17.3.5 Serial Mode Register (SMR)

SMR sets the SCI3's serial communication format and selects the clock source for the on-chip baud rate generator.

SMR is initialized to H'00 by a reset or in standby mode, watch mode, or module standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	COM	0	R/W	Communication Mode 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous mode) 0: Selects 8 or 5 bits as the data length 1: Selects 7 or 5 bits as the data length When 7-bit data is selected. the MSB (bit 7) in TDR is not transmitted. To select 5 bits as the data length, set 1 to both the PE and MP bits. The three most significant bits (bits 7, 6, and 5) in TDR are not transmitted. In clock synchronous mode, the data length is fixed to 8 bits regardless of the CHR bit setting.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. In clock synchronous mode, parity bit addition and checking is not performed regardless of the PE bit setting.

Bit	Bit Name	Initial Value	R/W	Description
4	PM	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity 1: Selects odd parity When even parity is selected, a parity bit is added in transmission so that the total number of 1 bits in the transmit data plus the parity bit is an even number, in reception, a check is carried out to confirm that the number of 1 bits in the receive data plus the parity bit is an even number. When odd parity is selected, a parity bit is added in transmission so that the total number of 1 bits in the transmit data plus the parity bit is an odd number, in reception, a check is carried out to confirm that the number of 1bits in the receive data plus the parity bit is an odd number. If parity bit addition and checking is disabled in clock synchronous mode and asynchronous mode, the PM bit setting is invalid.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous mode) Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits For reception, only the first stop bit is checked, regardless of the value in the bit. If the second stop bit is 0, it is treated as the start bit of the next transmit character.
2	MP	0	R/W	Multiprocessor Mode When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and PM bit settings are invalid. In clock synchronous mode, this bit should be cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
1	CKS1	0	R/W	Clock Select 0 and 1
0	CKS0	0	R/W	These bits select the clock source for the on-chip baud rate generator. 00: ϕ clock (n = 0) 01: ϕ_w/2 clock (n = 0) 10: ϕ/16 clock (n = 2) 11: ϕ/64 clock (n = 3) When ϕ_w/2 clock is selected in subactive mode and subsleep mode, the SCI3 is only enabled when ϕ_w/2 clock is selected for the CPU operating clock. For the relationship between the bit rate register setting and the baud rate, see section 17.3.8, Bit Rate Register (BRR). n is the decimal representation of the value of n in BRR (see section 17.3.8, Bit Rate Register (BRR)).

17.3.6 Serial Control Register (SCR)

SCR enables or disables SCI3 transfer operations and interrupt requests, and selects the transfer clock source. For details on interrupt requests, refer to section 17.8, Interrupt Requests.

SCR is initialized to H'00 by a reset or in standby mode, watch mode, or module standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, the TXI3 interrupt request is enabled. TXI3 can be released by clearing the TDRE bit or TIE bit to 0.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI3 and ERI3 interrupt requests are enabled. RXI3 and ERI3 can be released by clearing the RDRF bit or the FER, PER, or OER error flag to 0, or by clearing the RIE bit to 0.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled. When this bit is 0, the TDRE bit in SSR is fixed at 1. When transmit data is written to TDR while this bit is 1, Bit TDRE in SSR is cleared to 0 and serial data transmission is started. Be sure to carry out SMR settings, and setting of bit SPC3 in SPCR or SPCR2, to decide the transmission format before setting bit TE to 1.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled. In this state, serial data reception is started when a start bit is detected in asynchronous mode or serial clock input is detected in clock synchronous mode. Be sure to carry out the SMR settings to decide the reception format before setting bit RE to 1. Note that the RDRF, FER, PER, and OER flags in SSR are not affected when bit RE is cleared to 0, and retain their previous state.

Bit	Bit Name	Initial Value	R/W	Description
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode) When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped, and setting of the RDRF, FER, and OER status flags in SSR is prohibited. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared and normal reception is resumed. For details, refer to section 17.6, Multiprocessor Communication Function.
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, the TEI3 interrupt request is enabled. TEI3 can be released by clearing bit TDRE to 0 and clearing bit TEND to 0 in SSR, or by clearing bit TEIE to 0.
1	CKE1	0	R/W	Clock Enable 0 and 1
0	CKE0	0	R/W	Select the clock source. Asynchronous mode: 00: Internal baud rate generator (The SCK3 pin functions as an I/O port) 01: Internal baud rate generator (Outputs a clock of the same frequency as the bit rate from the SCK3 pin) 10: External clock (Inputs a clock with a frequency 16 times the bit rate from the SCK3 pin) 11: Reserved Clock synchronous mode: 00: Internal clock (The SCK3 pin functions as clock output) 01: Reserved 10: External clock (The SCK3 pin functions as clock input) 11: Reserved

17.3.7 Serial Status Register (SSR)

SSR consists of status flags of the SCI3 and multiprocessor bits for transfer. 1 cannot be written to flags TDRE, RDRF, OER, PER, and FER; they can only be cleared.

SSR is initialized to H'84 by a reset or in standby mode, watch mode, or module standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Indicates that transmit data is stored in TDR. [Setting conditions] • The TE bit in SCR is 0 • Data is transferred from TDR to TSR [Clearing conditions] • Writing of 0 to bit TDRE after reading TDRE = 1 • The transmit data is written to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that the received data is stored in RDR. [Setting condition] • Serial reception ends normally and receive data is transferred from RSR to RDR [Clearing condition] • Writing of 0 to bit RDRF after reading RDRF = 1 • When data is read from RDR If an error is detected in reception, or if the RE bit in SCR has been cleared to 0, RDR and bit RDRF are not affected and retain their previous state. Note that if data reception is completed while bit RDRF is still set to 1, an overrun error (OER) will occur and the receive data will be lost.

Bit	Bit Name	Initial Value	R/W	Description
5	OER	0	R/(W)*	Overrun Error [Setting condition] • An overrun error occurs in reception [Clearing condition] • Writing of 0 to bit OER after reading OER = 1 When bit RE in SCR is cleared to 0, bit OER is not affected and retains its previous state. When an overrun error occurs, RDR retains the receive data it held before the overrun error occurred, and data received after the error is lost. Reception cannot be continued with bit OER set to 1, and in clock synchronous mode, transmission cannot be continued either.
4	FER	0	R/(W)*	Framing Error [Setting condition] • A framing error occurs in reception [Clearing condition] • Writing of 0 to bit FER after reading FER = 1 When bit RE in SCR is cleared to 0, bit FER is not affected and retains its previous state. Note that, in 2-stop-bit mode, only the first stop bit is checked for a value of 1, and the second stop bit is not checked. When a framing error occurs, the receive data is transferred to RDR but bit RDRF is not set. Reception cannot be continued with bit FER set to 1. In clock synchronous mode, neither transmission nor reception is possible when bit FER is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/(W)*	Parity Error [Setting condition] • A parity error is generated during reception [Clearing condition] • Writing of 0 to bit PER after reading PER = 1 When bit RE in SCR is cleared to 0, bit PER is not affected and retains its previous state. Receive data in which a parity error has occurred is still transferred to RDR, but bit RDRF is not set. Reception cannot be continued with bit PER set to 1. In clock synchronous mode, neither transmission nor reception is possible when bit PER is set to 1.
2	TEND	1	R	Transmit End [Setting conditions] • The TE bit in SCR is 0 • TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character [Clearing conditions] • Writing of 0 to bit TDRE after reading TDRE = 1 • The transmit data is written to TDR
1	MPBR	0	R	Multiprocessor Bit Receive MPBR stores the multiprocessor bit in the receive character data. When the RE bit in SCR is cleared to 0, its previous state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be added to the transmit character data.

Note: * Only 0 can be written to clear the flag.

17.3.8 Bit Rate Register (BRR)

BRR is an 8-bit readable/writable register that specifies the bit rate. BRR is initialized to H'FF. Tables 17.3 and 17.4 show examples of the N setting in BRR and the n setting in bits CKS1 and CKS0 in SMR in asynchronous mode. Table 17.6 shows the maximum bit rate for each frequency in asynchronous mode. The values shown in these tables are values in active (high-speed) mode. When the ABCS bit in SEMR is set to 1 in asynchronous mode, the maximum bit rate is twice the values shown in table 17.6.

Tables 17.7 (1) and (2) show examples of the N setting in BRR and the n setting in bits CKS1 and CKS0 in SMR in clock synchronous mode. The values shown in these tables are values in active (high-speed) mode. The N setting in BRR and error for other operating frequencies and bit rates can be obtained by the following formulas:

[Asynchronous Mode, ABCS = 0]

$$N = \frac{\phi}{32 \times 2^{2n} \times B} - 1$$

$$\text{Error (\%)} = \frac{\text{B (bit rate obtained from n, N, } \phi\text{)} - \text{R (bit rate in left-hand column in table 17.3)}}{\text{R (bit rate in left-hand column in table 17.3)}} \times 100$$

[Asynchronous Mode, ABCS = 1*]

$$N = \frac{\phi}{16 \times 2^{2n} \times B} - 1$$

$$\text{Error (\%)} = \frac{\text{B (bit rate obtained from n, N, } \phi\text{)} - \text{R (bit rate in left-hand column in table 17.4)}}{\text{R (bit rate in left-hand column in table 17.4)}} \times 100$$

[Legend]

B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
ϕ: Operating frequency (Hz)
n: Baud rate generator input clock number (n = 0, 2, or 3)
(The correspondence between n and the clock is shown in table 17.5)

Note: * Only supported by the SCI3_1 interface.

Table 17.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode, ABCS = 0) (1)

Bit Rate (bit/s)	32.8 kHz			38.4 kHz			2 MHz			2.097152 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	—	—	—	—	—	—	2	35	–1.36	2	36	0.64
150	—	—	—	0	3	0.00	2	25	0.16	2	26	1.14
200	—	—	—	0	2	0.00	2	19	–2.34	2	19	2.40
250	0	1	2.50	—	—	—	0	249	0.00	2	15	2.40
300	—	—	—	0	1	0.00	0	207	0.16	0	217	0.21
600	—	—	—	0	0	0.00	0	103	0.16	0	108	0.21
1200	—	—	—	—	—	—	0	51	0.16	0	54	–0.70
2400	—	—	—	—	—	—	0	25	0.16	0	26	1.14
4800	—	—	—	—	—	—	0	12	0.16	0	13	–2.48
9600	—	—	—	—	—	—	—	—	—	0	6	–2.48
19200	—	—	—	—	—	—	—	—	—	—	—	—
31250	—	—	—	—	—	—	0	1	0.00	—	—	—
38400	—	—	—	—	—	—	—	—	—	—	—	—

Table 17.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode, ABCS = 0) (2)

Bit Rate (bit/s)	2.4576 MHz			3 MHz			3.6864 MHz			4 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	43	–0.83	2	52	0.50	2	64	0.70	2	70	0.03
150	2	31	0.00	2	38	0.16	2	47	0.00	2	51	0.16
200	2	23	0.00	2	28	1.02	2	35	0.00	2	38	0.16
250	2	18	1.05	2	22	1.90	2	28	–0.69	2	30	0.81
300	0	255	0.00	2	19	–2.34	2	23	0.00	2	25	0.16
600	0	127	0.00	0	155	0.16	0	191	0.00	0	207	0.16
1200	0	63	0.00	0	77	0.16	0	95	0.00	0	103	0.16
2400	0	31	0.00	0	38	0.16	0	47	0.00	0	51	0.16
4800	0	15	0.00	0	19	–2.34	0	23	0.00	0	25	0.16
9600	0	7	0.00	0	9	–2.34	0	11	0.00	0	12	0.16
19200	0	3	0.00	0	4	–2.34	0	5	0.00	—	—	—
31250	—	—	—	0	2	0.00	—	—	—	0	3	0.00
38400	0	1	0.00	—	—	—	0	2	0.00	—	—	—

Table 17.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode, ABCS = 0) (3)

Bit Rate (bit/s)	4.194304 MHz			4.9152MHz			5 MHz			6 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	73	0.64	2	86	0.31	2	88	–0.25	2	106	–0.44
150	2	54	–0.70	2	63	0.00	2	64	0.16	2	77	0.16
200	2	40	–0.10	2	47	0.00	2	48	–0.35	2	58	–0.69
250	2	32	–0.70	2	37	1.05	2	38	0.16	2	46	–0.27
300	2	26	1.14	2	31	0.00	2	32	–1.36	2	38	0.16
600	0	217	0.21	0	255	0.00	2	15	1.73	2	19	–2.34
1200	0	108	0.21	0	127	0.00	0	129	0.16	0	155	0.16
2400	0	54	–0.70	0	63	0.00	0	64	0.16	0	77	0.16
4800	0	26	1.14	0	31	0.00	0	32	–1.36	0	38	0.16
9600	0	13	–2.48	0	15	0.00	0	15	1.73	0	19	–2.34
19200	0	6	–2.48	0	7	0.00	0	7	1.73	0	9	–2.34
31250	—	—	—	0	4	–1.70	0	4	0.00	0	5	0.00
38400	—	—	—	0	3	0.00	0	3	1.73	0	4	–2.34

Table 17.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode, ABCS = 0) (4)

Bit Rate	6.144 MHz			7.3728 MHz			8 MHz			9.8304 MHz			10 MHz		
(bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	108	0.08	2	130	–0.07	2	141	0.03	2	174	–0.26	2	177	–0.25
150	2	79	0.00	2	95	0.00	2	103	0.16	2	127	0.00	2	129	0.16
200	2	59	0.00	2	71	0.00	2	77	0.16	2	95	0.00	2	97	–0.35
250	2	47	0.00	2	57	–0.69	2	62	–0.79	2	76	–0.26	2	77	0.16
300	2	39	0.00	2	47	0.00	2	51	0.16	2	63	0.00	2	64	0.16
600	2	19	0.00	2	23	0.00	2	25	0.16	2	31	0.00	2	32	–1.36
1200	0	159	0.00	0	191	0.00	0	207	0.16	0	255	0.00	2	15	1.73
2400	0	79	0.00	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
4800	0	39	0.00	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
9600	0	19	0.00	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
19200	0	9	0.00	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
31250	0	5	2.40	—	—	—	0	7	0.00	0	9	–1.70	0	9	0.00
38400	0	4	0.00	0	5	0.00	—	—	—	0	7	0.00	0	7	1.73

Table 17.4 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode, ABCS = 1) (1)

Bit Rate (bit/s)	32.8 kHz			38.4 kHz			2 MHz			2.097152 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	—	—	—	0	10	–0.83	2	70	0.03	2	73	0.64
150	0	6	–2.38	0	7	0.00	2	51	0.16	2	54	–0.70
200	0	4	2.50	0	5	0.00	2	38	0.16	2	40	–0.10
250	0	3	2.50	—	—	—	2	30	0.81	2	32	–0.70
300	—	—	—	0	3	0.00	2	25	0.16	2	26	1.14
600	—	—	—	0	1	0.00	0	207	0.16	0	217	0.21
1200	—	—	—	0	0	0.00	0	103	0.16	0	108	0.21
2400	—	—	—	—	—	—	0	51	0.16	0	54	–0.70
4800	—	—	—	—	—	—	0	25	0.16	0	26	1.14
9600	—	—	—	—	—	—	0	12	0.16	0	13	–2.48
19200	—	—	—	—	—	—	—	—	—	0	6	–2.48
31250	—	—	—	—	—	—	0	3	0.00	—	—	—
38400	—	—	—	—	—	—	—	—	—	—	—	—

Table 17.4 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode, ABCS = 1) (2)

Bit Rate (bit/s)	2.4576 MHz			3 MHz			3.6864 MHz			4 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	86	0.31	2	106	–0.44	2	130	–0. 07	2	141	0.03
150	2	63	0.00	2	77	0.16	2	95	0.00	2	103	0.16
200	2	47	0.00	2	58	–0.69	2	71	0.00	2	77	0.16
250	2	37	1.05	2	46	–0.27	2	57	–0.69	2	62	–0.79
300	2	31	0.00	2	38	0.16	2	47	0.00	2	51	0.16
600	0	255	0.00	2	19	–2.34	2	23	0.00	2	25	0.16
1200	0	127	0.00	0	155	0.16	0	191	0.00	0	207	0.16
2400	0	63	0.00	0	77	0.16	0	95	0.00	0	103	0.16
4800	0	31	0.00	0	38	0.16	0	47	0.00	0	51	0.16
9600	0	15	0.00	0	19	–2.34	0	23	0.00	0	25	0.16
19200	0	7	0.00	0	9	–2.34	0	11	0.00	0	12	0.16
31250	0	4	–1.70	0	5	0.00	—	—	—	0	7	0.00
38400	0	3	0.00	0	4	–2.34	0	5	0.00	—	—	—

Table 17.4 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode, ABCS = 1) (3)

Bit Rate (bit/s)	4.194304 MHz			4.9152MHz			5 MHz			6 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	148	–0.04	2	174	–0.26	2	177	–0.25	2	212	0.03
150	2	108	0.21	2	127	0.00	2	129	0.16	2	155	0.16
200	2	81	–0.10	2	95	0.00	2	97	–0.35	2	116	0.16
250	2	65	–0.70	2	76	–0.26	2	77	0.16	2	93	–0.27
300	2	54	–0.70	2	63	0.00	2	64	0.16	2	77	0.16
600	2	26	1.14	2	31	0.00	2	32	–1.36	2	38	0.16
1200	0	217	0.21	0	255	0.00	2	15	1.73	2	19	–2.34
2400	0	108	0.21	0	127	0.00	0	129	0.16	0	155	0.16
4800	0	54	–0.70	0	63	0.00	0	64	0.16	0	77	0.16
9600	0	26	1.14	0	31	0.00	0	32	–1.36	0	38	0.16
19200	0	13	–2.48	0	15	0.00	0	15	1.73	0	19	–2.34
31250	—	—	—	0	9	–1.70	0	9	0.00	0	11	0.00
38400	0	6	–2.48	0	7	0.00	0	7	1.73	0	9	–2.34

Table 17.4 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode, ABCS = 1) (4)

Bit Rate	6.144 MHz			7.3728 MHz			8 MHz			9.8304 MHz			10 MHz		
(bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	3	64	0.70	3	70	0.03	3	86	0.31	3	88	–0.25
150	2	159	0.00	2	191	0.00	2	207	0.16	2	255	0.00	3	64	0.16
200	2	119	0.00	2	143	0.00	2	155	0.16	2	191	0.00	2	194	0.16
250	2	95	0.00	2	114	0.17	2	124	0.00	2	153	–0.26	2	155	0.16
300	2	79	0.00	2	95	0.00	2	103	0.16	2	127	0.00	2	129	0.16
600	2	39	0.00	2	47	0.00	2	51	0.16	2	63	0.00	2	64	0.16
1200	2	19	0.00	2	23	0.00	2	25	0.16	2	31	0.00	2	32	–1.36
2400	0	159	0.00	0	191	0.00	0	207	0.16	0	255	0.00	2	15	1.73
4800	0	79	0.00	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
9600	0	39	0.00	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
19200	0	19	0.00	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
31250	0	11	2.40	0	14	–1.70	0	15	0.00	0	19	–1.70	0	19	0.00
38400	0	9	0.00	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73

Table 17.5 Correspondence between n and Clock

		SMR Setting	
n	Clock	CKS1	CKS0
0	ϕ	0	0
0	$\phi_w/2$*	0	1
2	$\phi/16$	1	0
3	$\phi/64$	1	1

Note: In subactive or subsleep mode, the SCI3_1, SCI3_2, and SCI3_3 interfaces can operate only when the CPU clock is $\phi_w/2$.

Table 17.6 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

	Maximum Bit Rate (bit/s)		Setting	
ϕ (MHz)	**ABCS = 0**	**ABCS = 1*2**	**n**	**N**
0.0328*1	512	1025	0	0
0.0384*1	600	1200	0	0
2	62500	125000	0	0
2.097152	65536	131072	0	0
2.4576	76800	153600	0	0
3	93750	187500	0	0
3.6864	115200	230400	0	0
4	125000	250000	0	0
4.194304	131072	262144	0	0
4.9152	153600	307200	0	0
5	156250	312500	0	0
6	187500	375000	0	0
6.144	192000	384000	0	0
7.3728	230400	460800	0	0
8	250000	500000	0	0
9.8304	307200	614400	0	0
10	312500	625000	0	0

Note: 1. When CKS1 = 0 and CKS0 = 1 in SMR
2. Only supported by the SCI3_1 interface.

Table 17.7 BRR Settings for Various Bit Rates (Clock Synchronous Mode) (1)

ϕ	32.8 kHz			38.4 kHz			2 MHz		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
200	0	20	–2.38	0	23	0.00	2	155	0.16
250	0	15	2.50	0	18	1.05	2	124	0.00
300	0	13	–2.38	0	15	0.00	2	103	0.16
500	0	7	2.50	—	—	—	2	62	–0.79
1 k	0	3	2.50	—	—	—	2	30	0.81
2.5 k	—	—	—	—	—	—	0	199	0.00
5 k	—	—	—	—	—	—	0	99	0.00
10 k	—	—	—	—	—	—	0	49	0.00
25 k	—	—	—	—	—	—	0	19	0.00
50 k	—	—	—	—	—	—	0	9	0.00
100 k	—	—	—	—	—	—	0	4	0.00
250 k	—	—	—	—	—	—	0	1	0.00
500 k	—	—	—	—	—	—	0*	0*	0.00*
1 M	—	—	—	—	—	—	—	—	—

Note: * Continuous transmission/reception is not possible.

Table 17.7 BRR Settings for Various Bit Rates (Clock Synchronous Mode) (2)

ϕ	4 MHz			8 MHz			10 MHz		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
200	3	77	0.16	3	155	0.16	3	194	0.16
250	2	249	0.00	3	124	0.00	3	155	0.16
300	2	207	0.16	3	103	0.16	3	129	0.16
500	2	124	0.00	2	249	0.00	3	77	0.16
1 k	2	62	–0.79	2	124	0.00	2	155	0.16
2.5 k	2	24	0.00	2	49	0.00	2	62	–0.79
5 k	0	199	0.00	2	24	0.00	2	30	0.81
10 k	0	99	0.00	0	199	0.00	0	249	0.00
25 k	0	39	0.00	0	79	0.00	0	99	0.00
50 k	0	19	0.00	0	39	0.00	0	49	0.00
100 k	0	9	0.00	0	19	0.00	0	24	0.00
250 k	0	3	0.00	0	7	0.00	0	9	0.00
500 k	0	1	0.00	0	3	0.00	0	4	0.00
1 M	0*	0*	0.00*	0	1	0.00	—	—	—

Note: * Continuous transmission/reception is not possible.

[Clock Synchronous Mode]

$$N = \frac{\phi}{4 \times 2^{2n} \times B} - 1$$

[Legend]

B: Bit rate (bit/s)

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

ϕ: Operating frequency (Hz)

n: Baud rate generator input clock number (n = 0, 2, or 3)
(The correspondence between n and the clock is shown in table 17.8.)

Table 17.8 Correspondence between n and Clock

		SMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	ϕ	0	0
0	$\phi_w/2$*	0	1
2	$\phi/16$	1	0
3	$\phi/64$	1	1

Note: In subactive or subsleep mode, the SCI3_1, SCI3_2, and SCI3_3 interfaces can operate only when the CPU clock is $\phi_w/2$.

17.3.9 Serial Port Control Register (SPCR)

SPCR selects the functions of the TXD32 and TXD31/IrTXD pins, selects whether input data of the RXD32 pin and RXD31/IrRXD pin is inverted or not, and selects output data of the TXD32 pin and TXD31/IrTXD pin is inverted or not.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved
6	—	1	—	These bits are always read as 1 and cannot be modified.
5	SPC32	0	R/W	P32/TXD32/SCL (PE5/TXD32) Pin Function Switch Selects whether pin P32/TXD32/SCL (PE5/TXD32) is used as P32/SCL (PE5) or as TXD32. 0: P32/SCL (PE5) I/O pin 1: TXD32 output pin Set the TE bit in SCR3_2 after having set this bit to 1.
4	SPC31	0	R/W	P42/TXD31/IrTXD/TMOFH (PF3/TXD31/IrTXD) Pin Function Switch Selects whether pin P42/TXD31/IrTXD/TMOFH (PF3/TXD31/IrTXD) is used as P42/TMOFH (PF3) or as TXD31/IrTXD. 0: P42 (PF3) I/O pin or TMOFH output pin 1: TXD31/IrTXD output pin Set the TE bit in SCR3_1 after having set this bit to 1.

Bit	Bit Name	Initial Value	R/W	Description
3	SCINV3	0	R/W	TXD32 Pin Output Data Inversion Switch Selects whether the logic level of output data of the TXD32 pin is inverted or not. 0: TXD32 output data is not inverted. 1: TXD32 output data is inverted.
2	SCINV2	0	R/W	RXD32 Pin Input Data Inversion Switch Selects whether the logic level of input data of the RXD32 pin is inverted or not. 0: RXD32 input data is not inverted. 1: RXD32 input data is inverted.
1	SCINV1	0	R/W	TXD31/IrTXD Pin Output Data Inversion Switch Selects whether the logic level of output data of the TXD31/IrTXD pin is inverted or not. 0: TXD31/IrTXD output data is not inverted. 1: TXD31/IrTXD output data is inverted.
0	SCINV0	0	R/W	RXD31/IrRXD Pin Input Data Inversion Switch Selects whether the logic level of input data of the RXD31/IrRXD pin is inverted or not. 0: RXD31/IrRXD input data is not inverted. 1: RXD31/IrRXD input data is inverted.

Note: When the serial port control register is modified, the data being input or output up to that point is inverted immediately after the modification, and an invalid data change is input or output. When modifying the serial port control register, modification must be made in a state in which data changes are invalidated.

17.3.10 Serial Port Control Register 2 (SPCR2)

SPCR2 selects the function of the TXD33 pin and selects whether or not data input on the RXD33 pin and output via the TXD33 pin are inverted.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
4	SPC33	0	R/W	PE2/TXD33 Pin Function Switch Selects whether pin PE2/TXD33 is used as PE2 or as TXD33. 0: PE2 I/O pin 1: TXD33 output pin Set the TE bit in SCR3_3 after having set this bit to 1.
3	—	1	—	Reserved
2	—	1	—	These bits are always read as 1 and cannot be modified.
1	SCINV5	0	R/W	TXD33 Pin Output Data Inversion Switch Specifies whether the logic level of output data of the TXD33 pin is inverted. 0: TXD33 output data is not inverted 1: TXD33 output data is inverted
0	SCINV4	0	R/W	RXD33 Pin Input Data Inversion Switch Specifies whether the logic level of input data of the RXD33 pin is inverted. 0: RXD33 input data is not inverted 1: RXD33 input data is inverted

Note: When the values in bits 1 and 0 of serial port control register 2 are changed, the data being input or output up to that point is inverted immediately after the change, and this may lead to the input or output of invalid data. Values in this control register must only be changed while changes to the data have no effect.

17.3.11 IrDA Control Register (IrCR)

IrCR controls the IrDA operation of the SCI3_1.

Bit	Bit Name	Initial Value	R/W	Description
7	IrE	0	R/W	IrDA Enable Selects whether the SCI3_1 I/O pins function as the SCI or IrDA. 0: TXD31/IrTXD or RXD31/IrRXD pin functions as TXD31 or RXD31 1: TXD31/IrTXD or RXD31/IrRXD pin functions as IrTXD or IrRXD
6	IrCKS2	0	R/W	IrDA Clock Select If the IrDA function is enabled, these bits set the high-pulse width when encoding the IrTXD output pulse. 000: Bit rate × 3/16 001: ϕ/2 010: ϕ/4 011: ϕ/8 100: ϕ/16 101: Setting prohibited 11x: Setting prohibited
5	IrCKS1	0	R/W	
4	IrCKS0	0	R/W	
3 to 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

[Legend]
x: Don't care

17.3.12 Serial Extended Mode Register (SEMR)

SEMR controls extended functions of the SCI3_1, i.e. specifies the basic clock in asynchronous mode.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved The write value should always be 0.
3	ABCS	0	R/W	Asynchronous Mode Basic Clock Select Selects the basic clock for one-bit interval in asynchronous mode. The ABCS setting is enabled in asynchronous mode (COM = 0 in SMR3) 0: Basic clock with a frequency 16 times the transfer rate 1: Basic clock with a frequency 8 times the transfer rate Clear this bit to 0 when the IrDA function is enabled.
2 to 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

17.4 Operation in Asynchronous Mode

Figure 17.2 shows the general format for asynchronous serial communication. Each frame consists of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low level), and finally stop bits (high level). In reception in asynchronous mode, synchronization is with falling edges of the start bits. The data is sampled on the 8th pulse of a clock signal with a frequency 16 times the bit rate, so that the transferred data is latched at the center of each bit. When the ABCS bit in SEMR is set to 1, data is sampled on the 4th pulse of a clock with a frequency 8 times the bit rate*. Internally, the SCI3 has independent transmitter and receiver units, which enables full duplex operation. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer. Table 17.9 shows the 16 data transfer formats that can be set in asynchronous mode. The format is selected by the settings in SMR as shown in table 17.10.

Note: Only supported by the SCI3_1 interface.

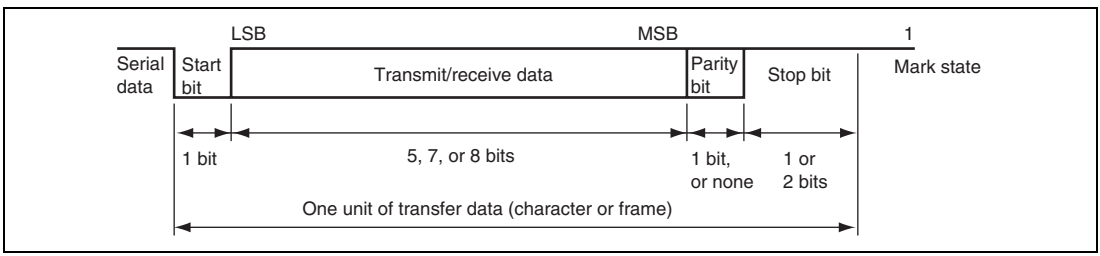


Figure 17.2 Data Format in Asynchronous Communication

17.4.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK3 pin can be selected as the SCI3's serial clock source, according to the setting of the COM bit in SMR and the CKE0 and CKE1 bits in SCR. When an external clock signal is input on the SCK3 pin, its frequency should be 16 times the bit rate (or 8 times the bit rate when the ABCS bit in SEMR is set to 1*). For details on selection of the clock source, see table 17.11. When the SCI3 is operated on an internal clock, the clock can be output from the SCK3 pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that rising edges of the clock signal are in the middle of each bit of the data to be transferred, as shown in figure 17.3.

Note: * Only supported by the SCI3_1 interface.

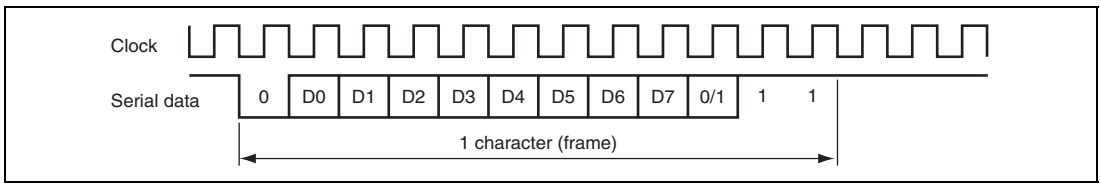


Figure 17.3 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode) (Example with 8-Bit Data, Parity, Two Stop Bits)

Table 17.9 Data Transfer Formats (Asynchronous Mode)

SMR				Serial Data Transfer Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	START 8-bit data STOP
0	0	0	1	START 8-bit data STOP STOP
0	0	1	0	START 8-bit data MPB STOP
0	0	1	1	START 8-bit data MPB STOP STOP
0	1	0	0	START 8-bit data P STOP
0	1	0	1	START 8-bit data P STOP STOP
0	1	1	0	START 5-bit data STOP
0	1	1	1	START 5-bit data STOP STOP
1	0	0	0	START 7-bit data STOP
1	0	0	1	START 7-bit data STOP STOP
1	0	1	0	START 7-bit data MPB STOP
1	0	1	1	START 7-bit data MPB STOP STOP
1	1	0	0	START 7-bit data P STOP
1	1	0	1	START 7-bit data P STOP STOP
1	1	1	0	START 5-bit data P STOP
1	1	1	1	START 5-bit data P STOP STOP

[Legend]
START: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

Table 17.10 SMR Settings and Corresponding Data Transfer Formats

SMR						Data Transfer Format			
Bit 7 COM	Bit 6 CHR	Bit 2 MP	Bit 5 PE	Bit 3 STOP	Mode	Data Length	Multiprocessor Bit	Parity Bit	Stop Bit Length
0	0	0	0	0	Asynchron-ous mode	8-bit data	No	No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	1		0	0		7-bit data		No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	0	1	0	0		8-bit data	Yes	No	1 bit
				1					2 bits
			1	0		5-bit data	No		1 bit
				1					2 bits
	1		0	0		7-bit data	Yes		1 bit
				1					2 bits
			1	0		5-bit data	No	Yes	1 bit
				1					2 bits
1	X	0	x	x	Clock synchronous mode	8-bit data	No	No	No

[Legend]

x: Don't care

Table 17.11 SMR and SCR Settings and Clock Source Selection

SMR	SCR				
Bit 7	Bit 1	Bit 0		Transmit/Receive Clock	
COM	CKE1	CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Asynchronous mode	Internal	I/O port pins (pins are not assigned to the SCK31 or SCK32 functions)
		1			Output for clock with same frequency as bit rate
	1	0		External	Input for clock with frequency 16 times bit rate*
1	0	0	Clock synchronous mode	Internal	Output for serial clock
	1	0		External	Input for serial clock
0	1	1	Reserved (Do not specify these combinations)		
1	0	1			
1	1	1			

Note: The input clock may have a frequency 8 times the bit rate when the ABCS bit in SEMR is set to 1 (only supported for SCI3_1).

17.4.2 SCI3 Initialization

Follow the flowchart as shown in figure 17.4 to initialize the SCI3. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not initialize the contents of the RDRF, PER, FER, and OER flags, or the contents of RDR. When the external clock is used in asynchronous mode, the clock must be supplied even during initialization. When the external clock is used in clock synchronous mode, the clock must not be supplied during initialization.

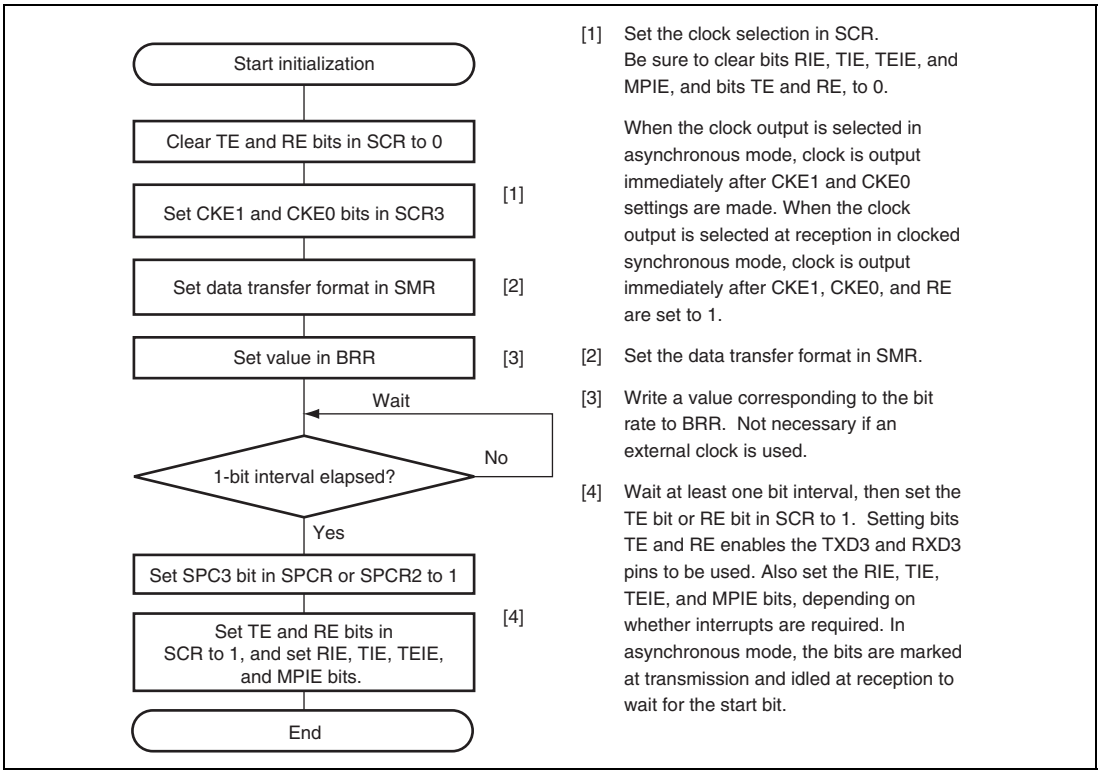


Figure 17.4 Sample SCI3 Initialization Flowchart

17.4.3 Data Transmission

Figure 17.5 shows an example of operation for transmission in asynchronous mode. In transmission, the SCI3 operates as described below.

1. The SCI3 monitors the TDRE flag in SSR. If the flag is cleared to 0, the SCI3 recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI3 sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a TXI3 interrupt request is generated. Continuous transmission is possible because the TXI3 interrupt routine writes next transmit data to TDR before transmission of the current transmit data has been completed.
3. The SCI3 checks the TDRE flag at the timing for sending the stop bit.
4. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, and then 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 the "mark state" is entered, in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a TEI3 interrupt request is generated.

Figure 17.6 shows a sample flowchart for transmission in asynchronous mode.

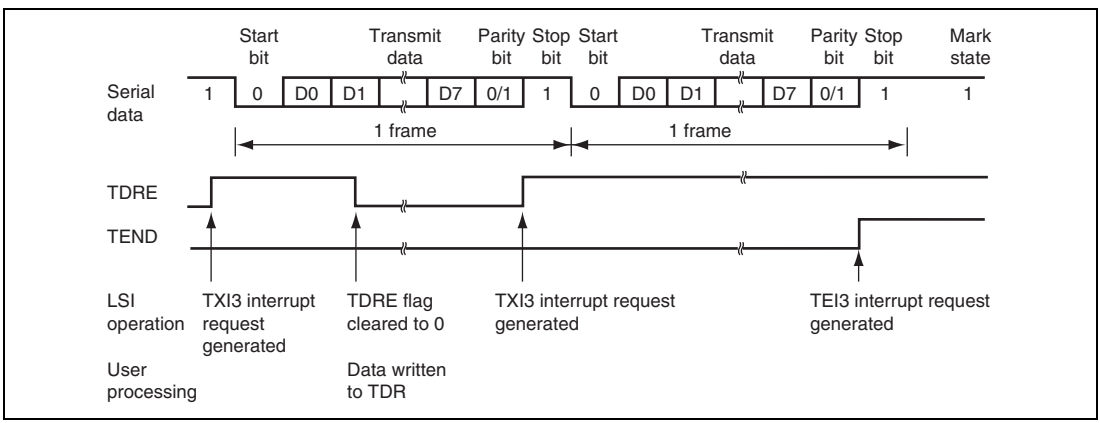
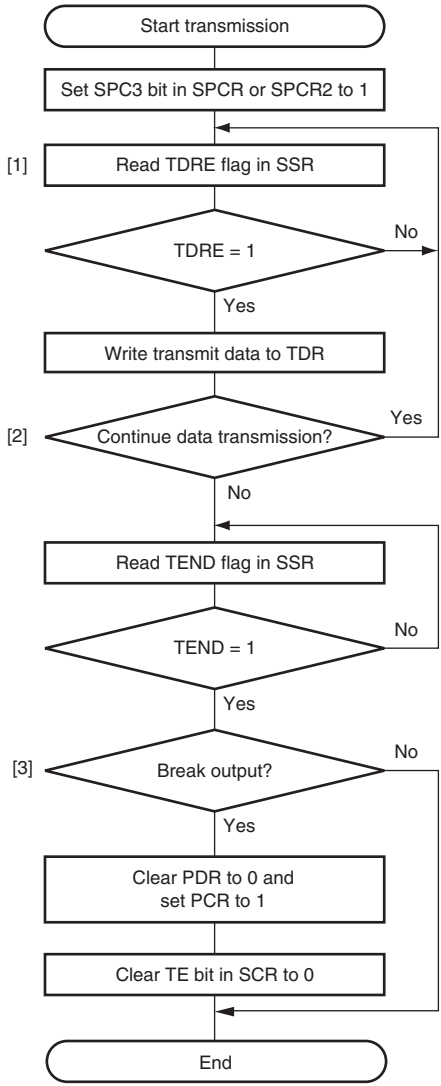


Figure 17.5 Example SCI3 Operation in Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)



[1] Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. When data is written to TDR, the TDRE flag is automaticaly cleared to 0. (After the TE bit is set to 1, one frame of 1 is output, then transmission is possible.)

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

[3] To output a break in serial transmission, after setting PCR to 1 and PDR to 0, clear the TE bit in SCR to 0.

Figure 17.6 Sample Serial Transmission Flowchart (Asynchronous Mode)

17.4.4 Serial Data Reception

Figure 17.7 shows an example of operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

1. The SCI3 monitors the communication line. If a start bit is detected, the SCI3 performs internal synchronization, receives data in RSR, and checks the parity bit and stop bit.

- Parity check

  The SCI3 checks that the number of 1 bits in the receive data conforms to the parity (odd or even) set in bit PM in the serial mode register (SMR).
- Stop bit check

  The SCI3 checks that the stop bit is 1. If two stop bits are used, only the first is checked.
- Status check

  The SCI3 checks that bit RDRF is set to 0, indicating that the receive data can be transferred from RSR to RDR.

2. If an overrun error occurs (when reception of the next data is completed while the RDRF flag is still set to 1), the OER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI3 interrupt request is generated. Receive data is not transferred to RDR.
3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI3 interrupt request is generated.
4. If a framing error is detected (when the stop bit is 0), the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE3 bit in SCR is set to 1 at this time, an ERI3 interrupt request is generated.
5. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI3 interrupt request is generated. Continuous reception is possible because the RXI3 interrupt routine reads the receive data transferred to RDR before reception of the next receive data has been completed.

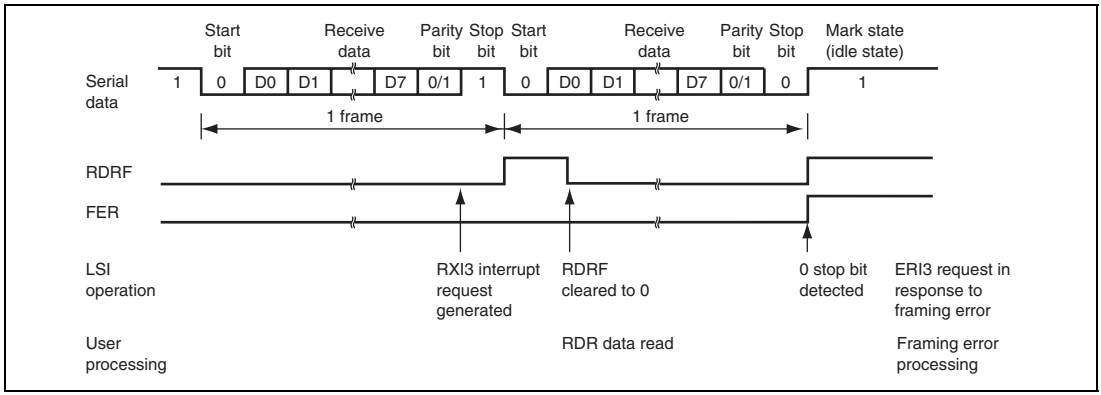


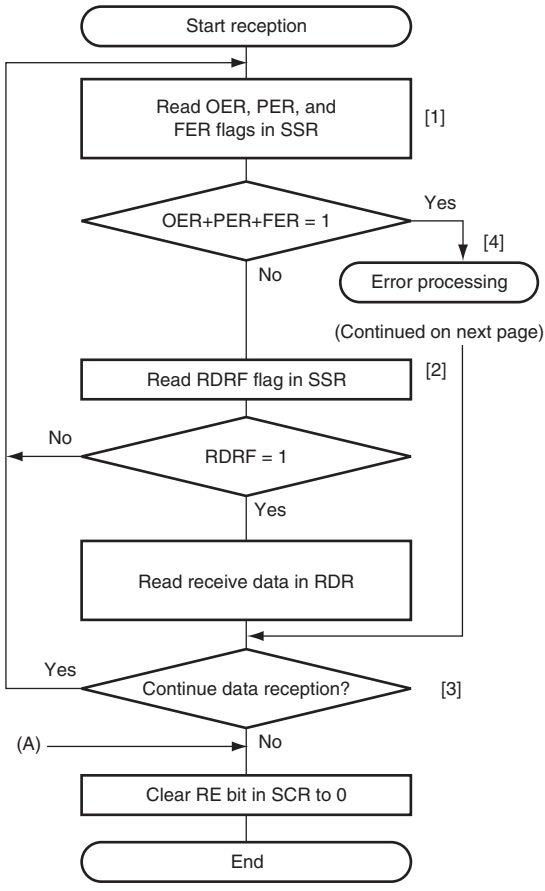
Figure 17.7 Example SCI3 Operation in Reception in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)

Table 17.12 shows the states of the SSR status flags and receive data handling when a receive error is detected. If a receive error is detected, the RDRF flag retains its state before receiving data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the OER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 17.8 shows a sample flowchart for serial data reception.

Table 17.12 SSR Status Flags and Receive Data Handling

SSR Status Flag					
RDRF*	**OER**	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	1	0	0	Lost	Overrun error
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 + framing error
1	1	0	1	Lost	Overrun error + parity error
0	0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	1	Lost	Overrun error + framing error + parity error

Note: * The RDRF flag retains the state it had before data reception. However, note that if RDR is read after an overrun error has occurred in a frame because reading of the receive data in the previous frame was delayed, the RDRF flag will be cleared to 0.



[1] Read the OER, PER, and FER flags in SSR to identify the error. If a receive error occurs, performs the appropriate error processing.

[2] Read SSR and check that RDRF = 1, then read the receive data in RDR. The RDRF flag is cleared automatically.

[3] To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag and read RDR.
The RDRF flag is cleared automatically.

[4] If a receive error occurs, read the OER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the OER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RXD3 pin.

Figure 17.8 Sample Serial Data Reception Flowchart (Asynchronous Mode) (1)

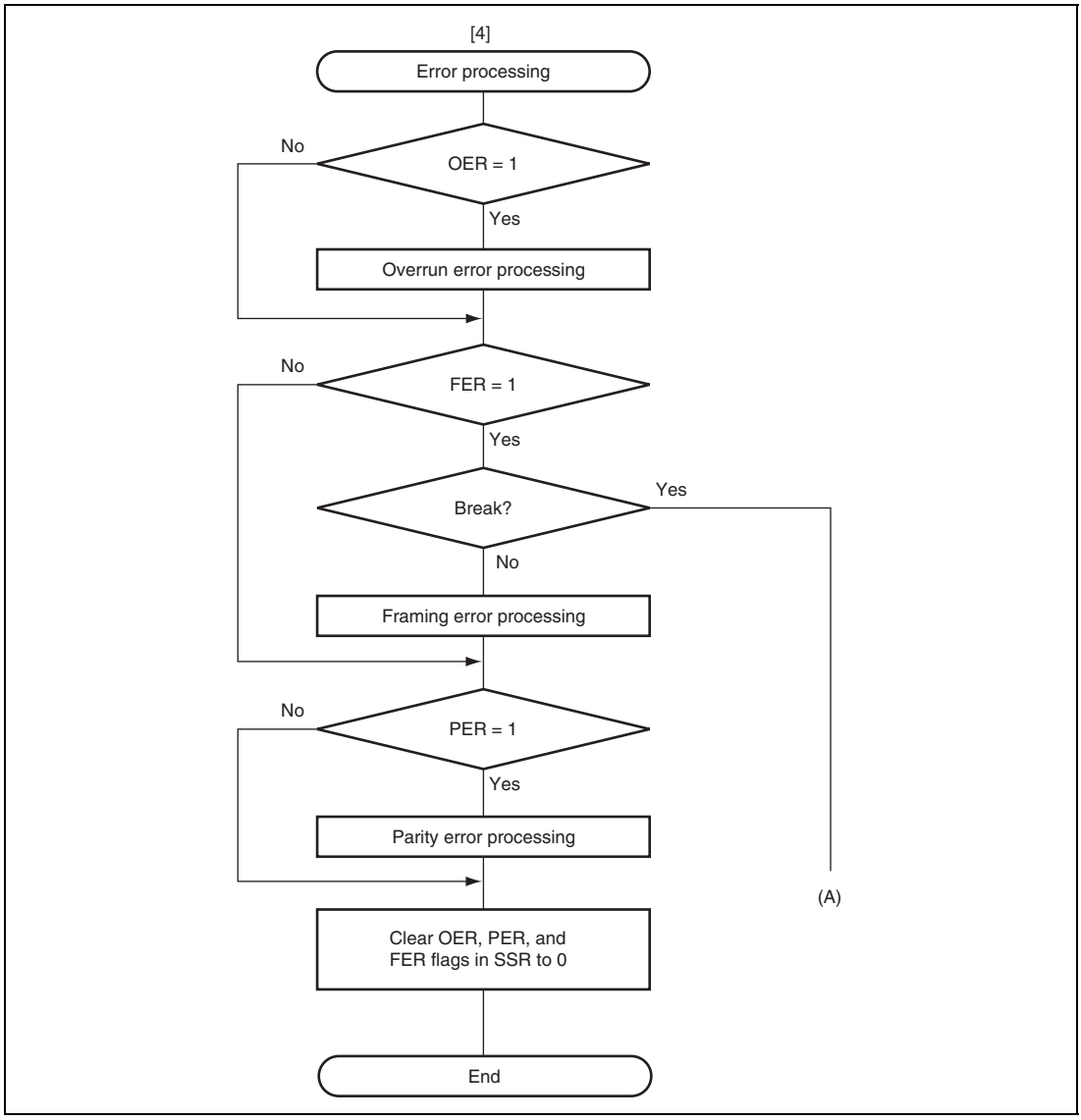


Figure 17.8 Sample Serial Data Reception Flowchart (Asynchronous Mode) (2)

17.5 Operation in Clock Synchronous Mode

Figure 17.9 shows the general format for clock synchronous communication. In clock synchronous mode, data is transmitted or received synchronous with clock pulses. A single character in the transmit data consists of the 8-bit data starting from the LSB. In clock synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. In clock synchronous mode, the SCI3 receives data in synchronous with the rising edge of the serial clock. After 8-bit data is output, the transmission line holds the MSB state. In clock synchronous mode, no parity or multiprocessor bit is added. Inside the SCI3, the transmitter and receiver are independent units, enabling full-duplex communication through the use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer.

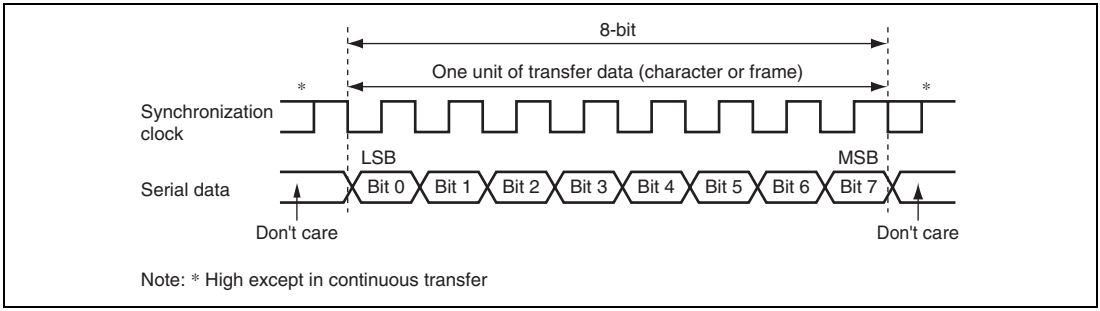


Figure 17.9 Data Format in Clock Synchronous Communications

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

17.5.2 SCI3 Initialization

Before transmitting and receiving data, the SCI3 should be initialized as described in a sample flowchart in figure 17.4.

17.5.3 Serial Data Transmission

Figure 17.10 shows an example of SCI3 operation for transmission in clock synchronous mode. In serial transmission, the SCI3 operates as described below.

1. The SCI3 monitors the TDRE flag in SSR, and if the flag is 0, the SCI recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. The SCI3 sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a TXI3 interrupt request is generated.
3. 8-bit data is sent from the TXD3 pin synchronized with the output clock when output clock mode has been specified, and synchronized with the input clock when use of an external clock has been specified. Serial data is transmitted sequentially from the LSB (bit 0), from the TXD3 pin.
4. The SCI checks the TDRE flag at the timing for sending the MSB (bit 7).
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission 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 maintains the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI3 is generated.
7. The SCK3 pin is fixed high.

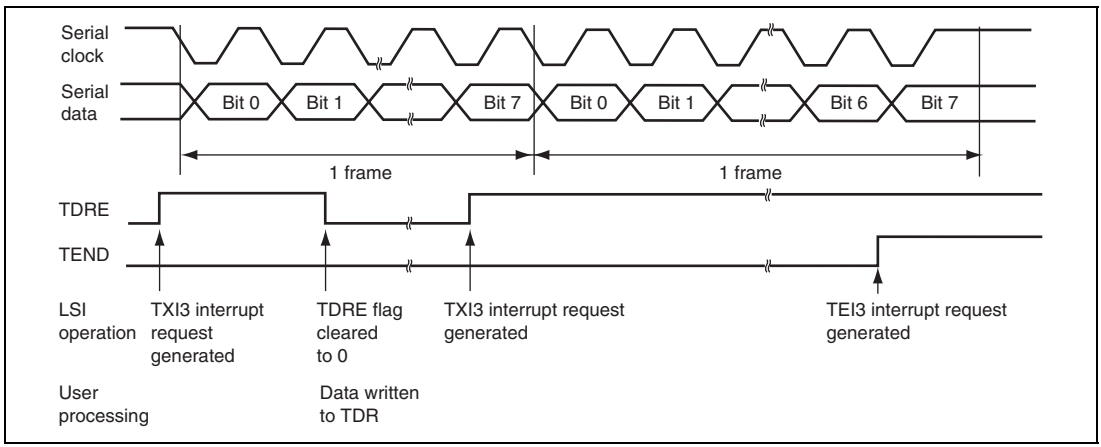


Figure 17.10 Example of SCI3 Operation in Transmission in Clock Synchronous Mode

Figure 17.11 shows a sample flowchart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (OER, FER, or PER) is set to 1. Make sure that the receive error flags are cleared to 0 before starting transmission.

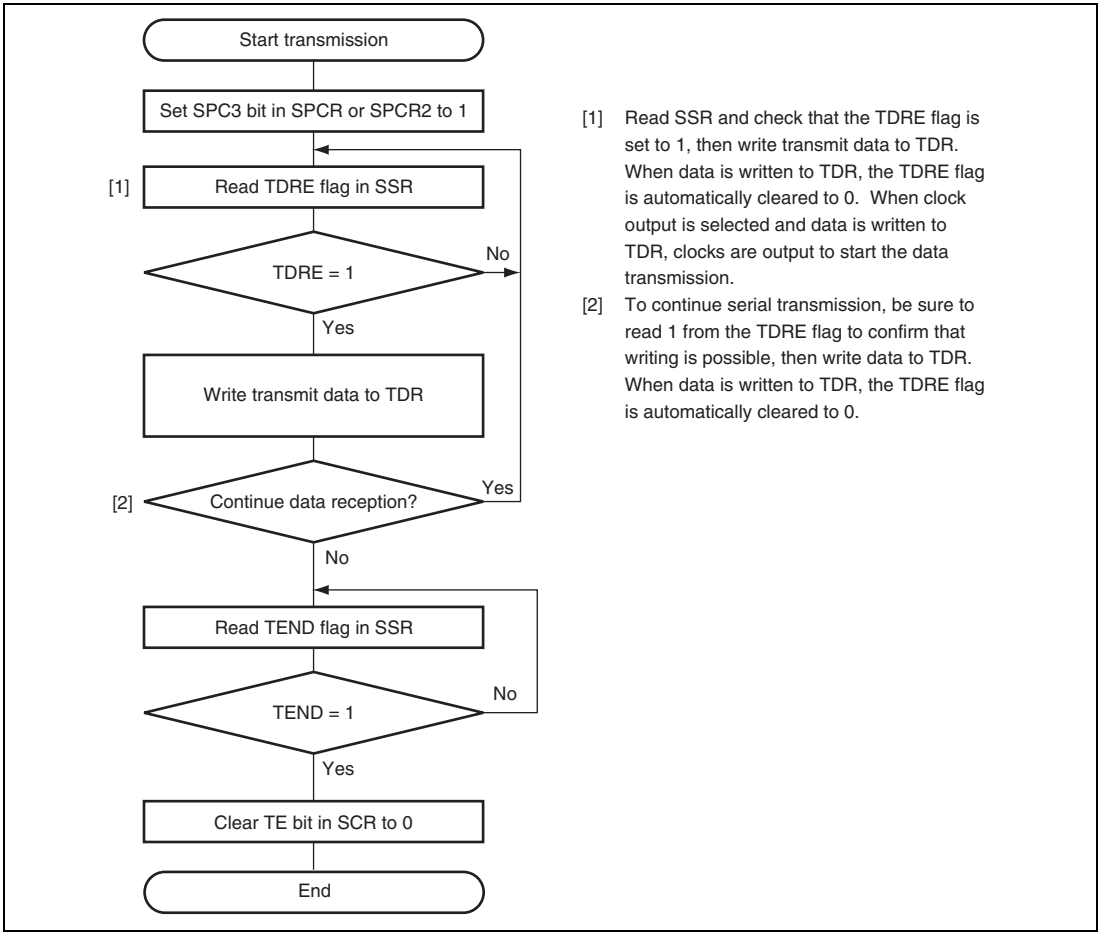


Figure 17.11 Sample Serial Transmission Flowchart (Clock Synchronous Mode)

17.5.4 Serial Data Reception (Clock Synchronous Mode)

Figure 17.12 shows an example of SCI3 operation for reception in clock synchronous mode. In serial reception, the SCI3 operates as described below.

1. The SCI3 performs internal initialization synchronous with a synchronous clock input or output, starts receiving data.
2. The SCI3 stores the received data in RSR.
3. If an overrun error occurs (when reception of the next data is completed while the RDRF flag in SSR is still set to 1), the OER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI3 interrupt request is generated, receive data is not transferred to RDR, and the 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 data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI3 interrupt request is generated.

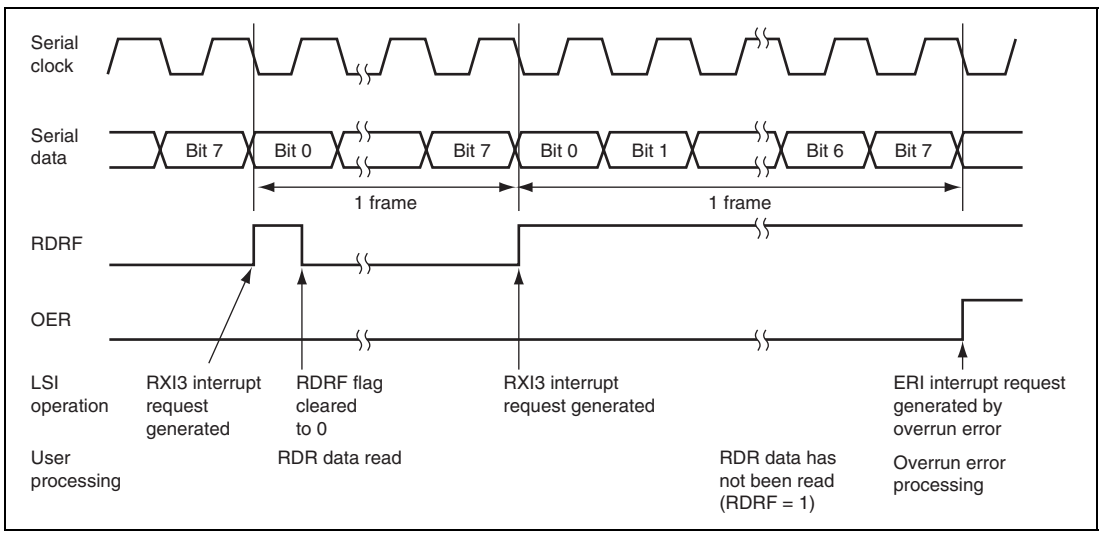


Figure 17.12 Example of SCI3 Reception Operation in Clock Synchronous Mode

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the OER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 17.13 shows a sample flowchart for serial data reception.

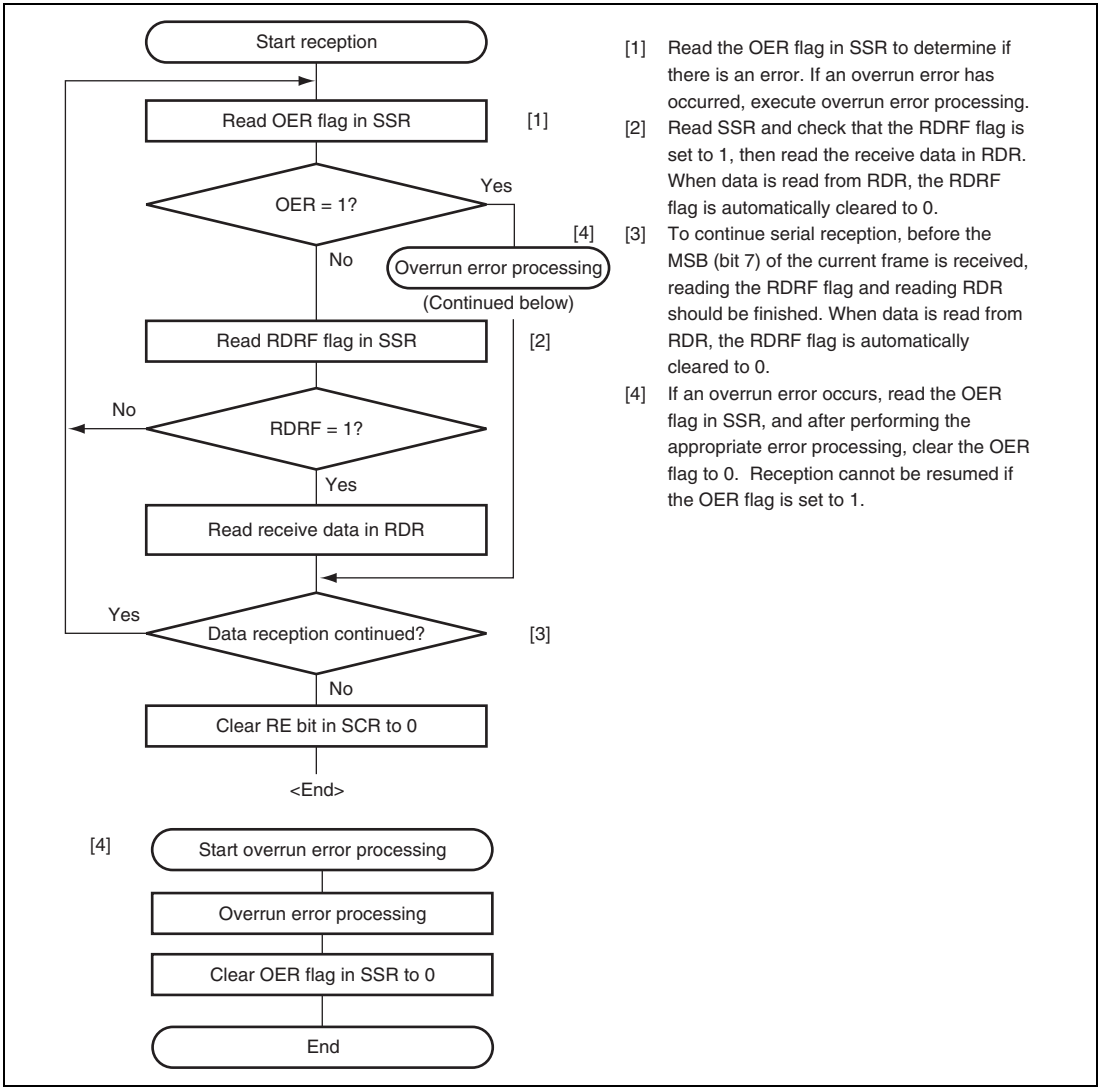


Figure 17.13 Sample Serial Reception Flowchart (Clock Synchronous Mode)

17.5.5 Simultaneous Serial Data Transmission and Reception

Figure 17.14 shows a sample flowchart for simultaneous serial transmit and receive operations. The following procedure should be used for simultaneous serial data transmit and receive operations. To switch from transmit mode to simultaneous transmit and receive mode, after checking that the SCI3 has finished transmission and the TDRE and TEND flags are set to 1, clear TE to 0. Then simultaneously set TE and RE to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI3 has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (OER, FER, and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.

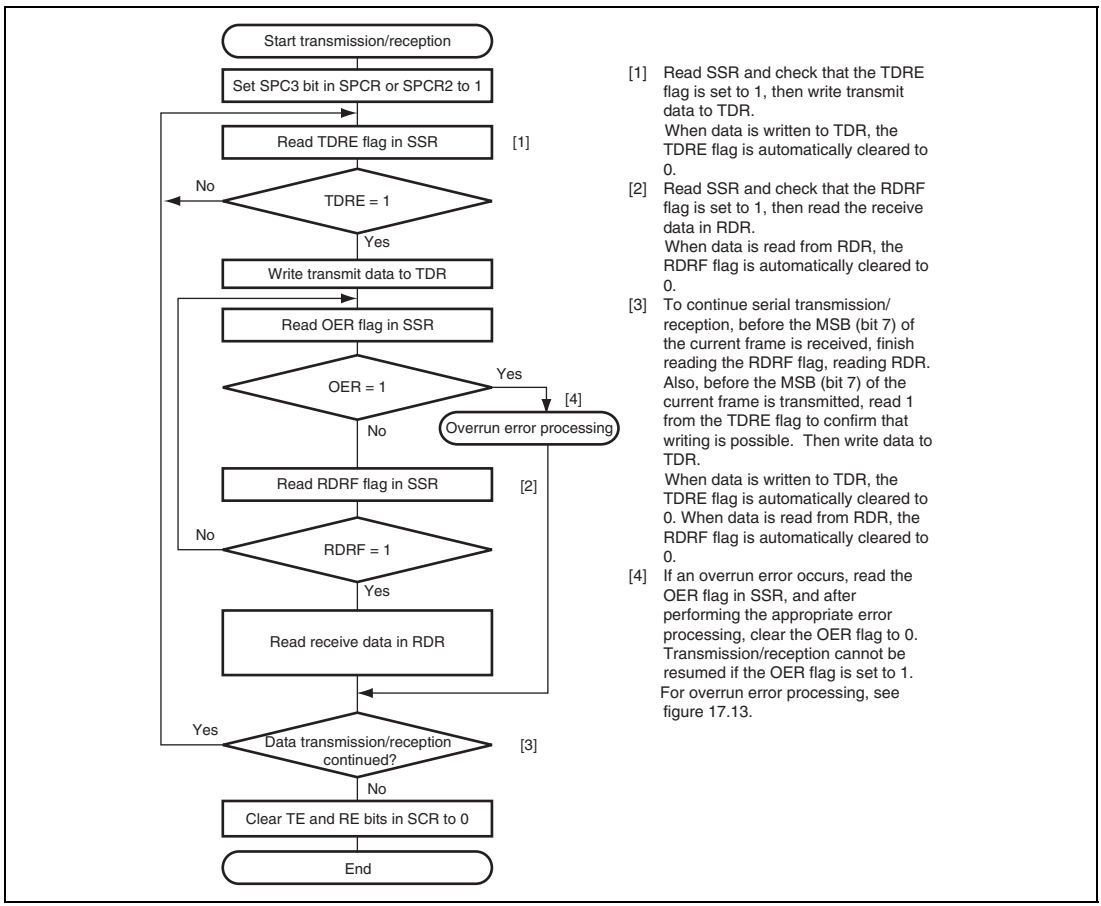


Figure 17.14 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations (Clock Synchronous Mode)

17.6 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer between a number of processors sharing communication lines by asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is performed, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles; an ID transmission cycle that specifies the receiving station, and a data transmission cycle. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle; if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 17.15 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends the ID code of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose IDs do not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI3 uses the MPIE bit in SCR to implement this function. When the MPIE bit is set to 1, transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status flags, RDRF, FER, and OER to 1, are inhibited until data with a 1 multiprocessor bit is received. On reception of a receive character with a 1 multiprocessor bit, the MPBR bit in SSR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCR is set to 1 at this time, an RXI3 interrupt is generated.

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

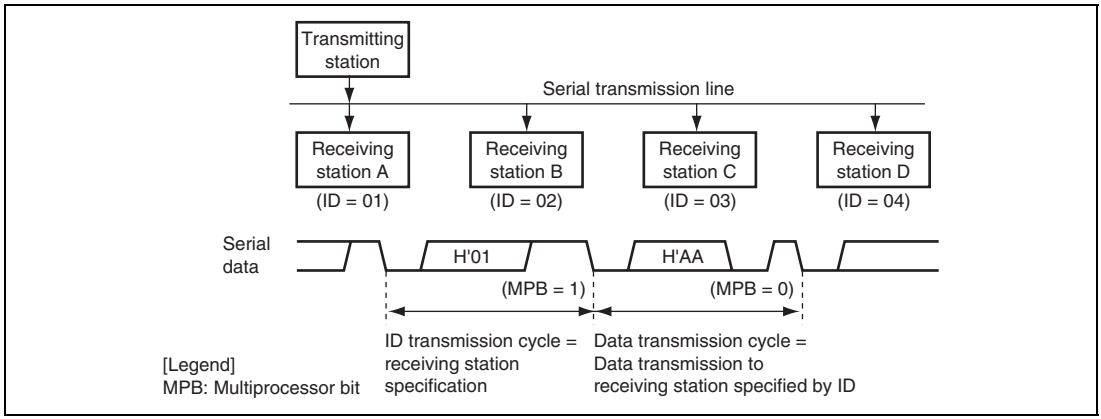


Figure 17.15 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

17.6.1 Multiprocessor Serial Data Transmission

Figure 17.16 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmission cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI3 operations are the same as those in asynchronous mode.

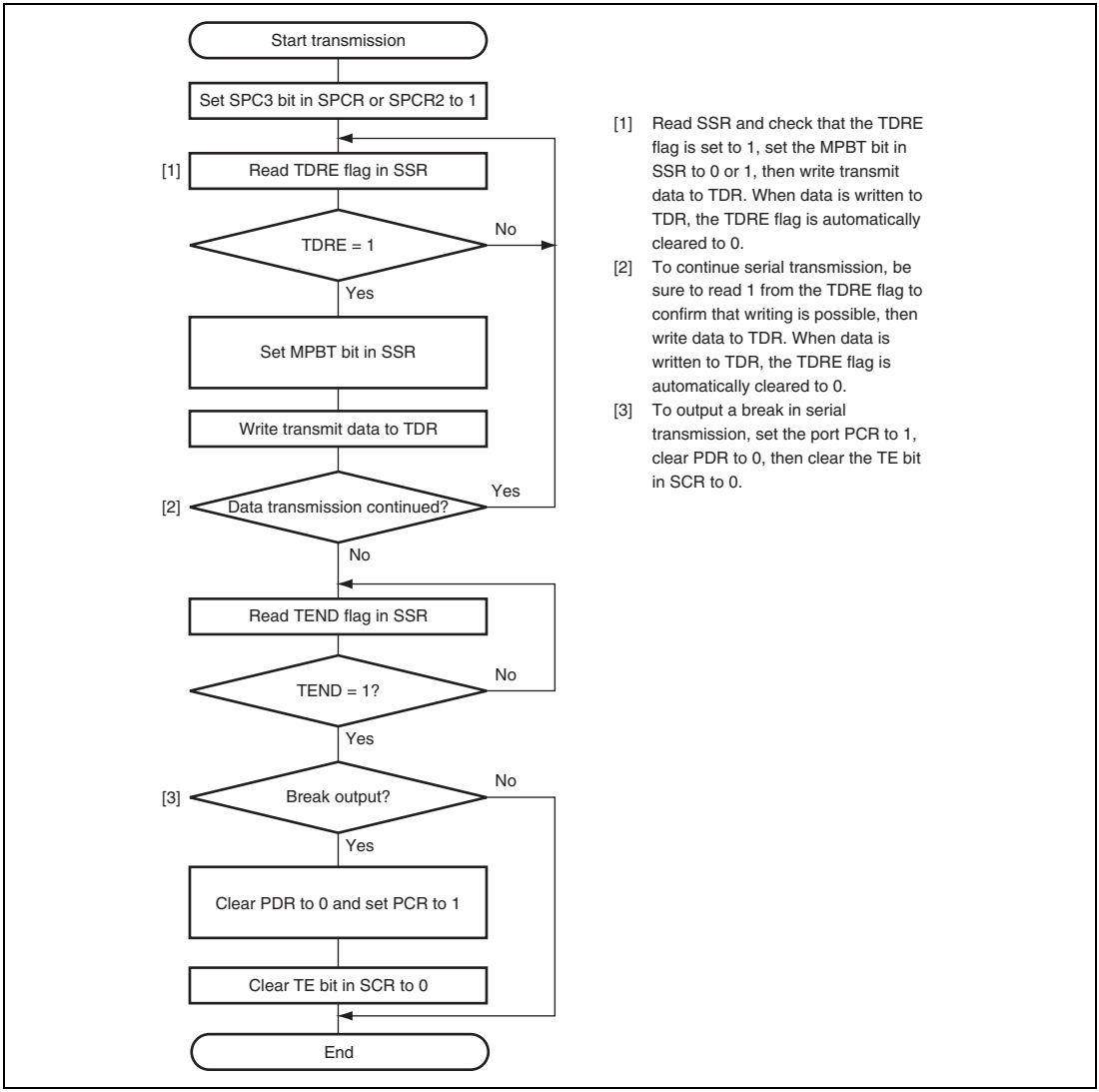


Figure 17.16 Sample Multiprocessor Serial Transmission Flowchart

17.6.2 Multiprocessor Serial Data Reception

Figure 17.17 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCR is set to 1, data is skipped until data with a 1 multiprocessor bit is received. On receiving data with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI3 interrupt request is generated at this time. All other SCI3 operations are the same as in asynchronous mode. Figure 17.18 shows an example of SCI3 operation for multiprocessor format reception.

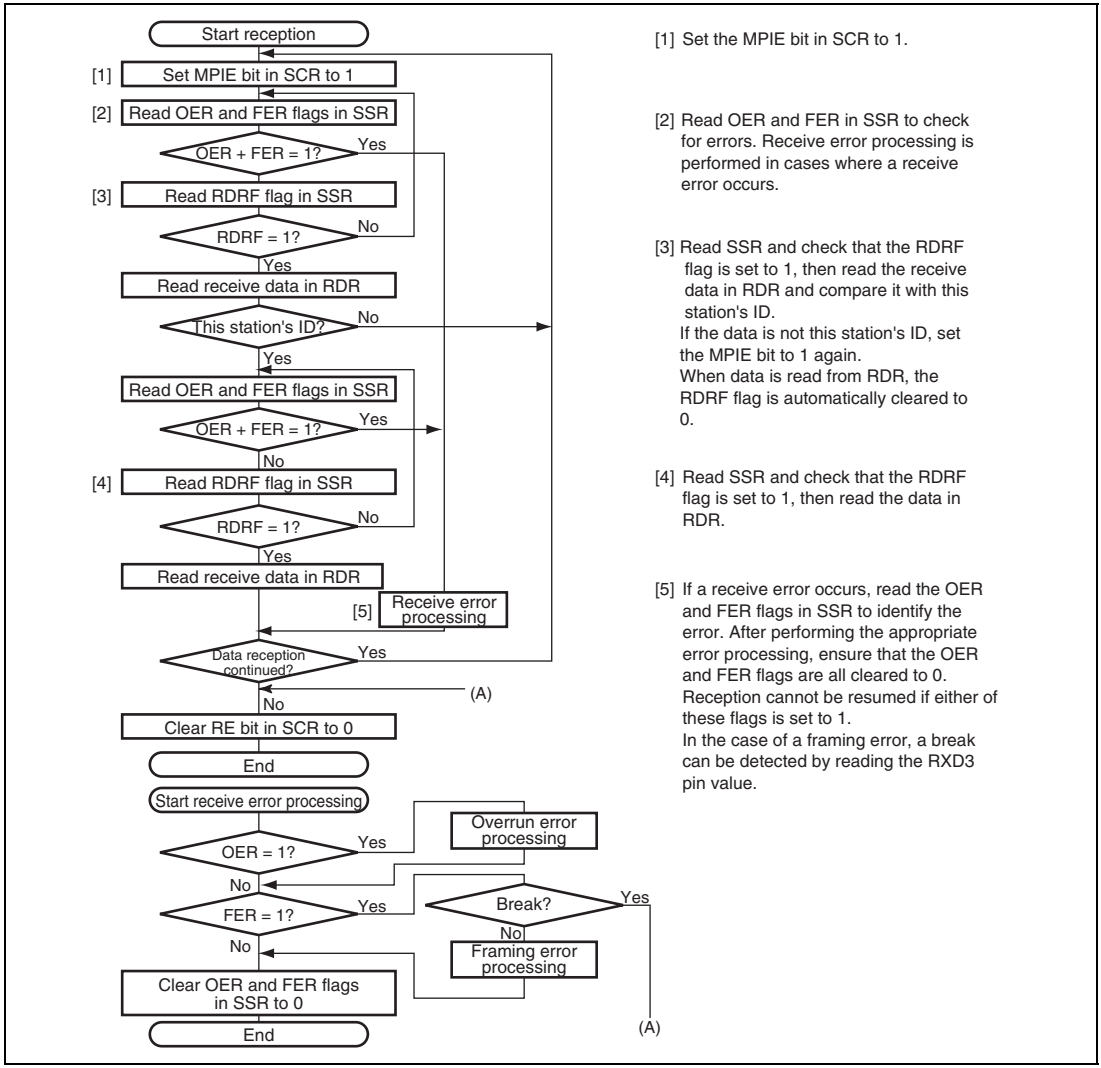


Figure 17.17 Sample Multiprocessor Serial Reception Flowchart

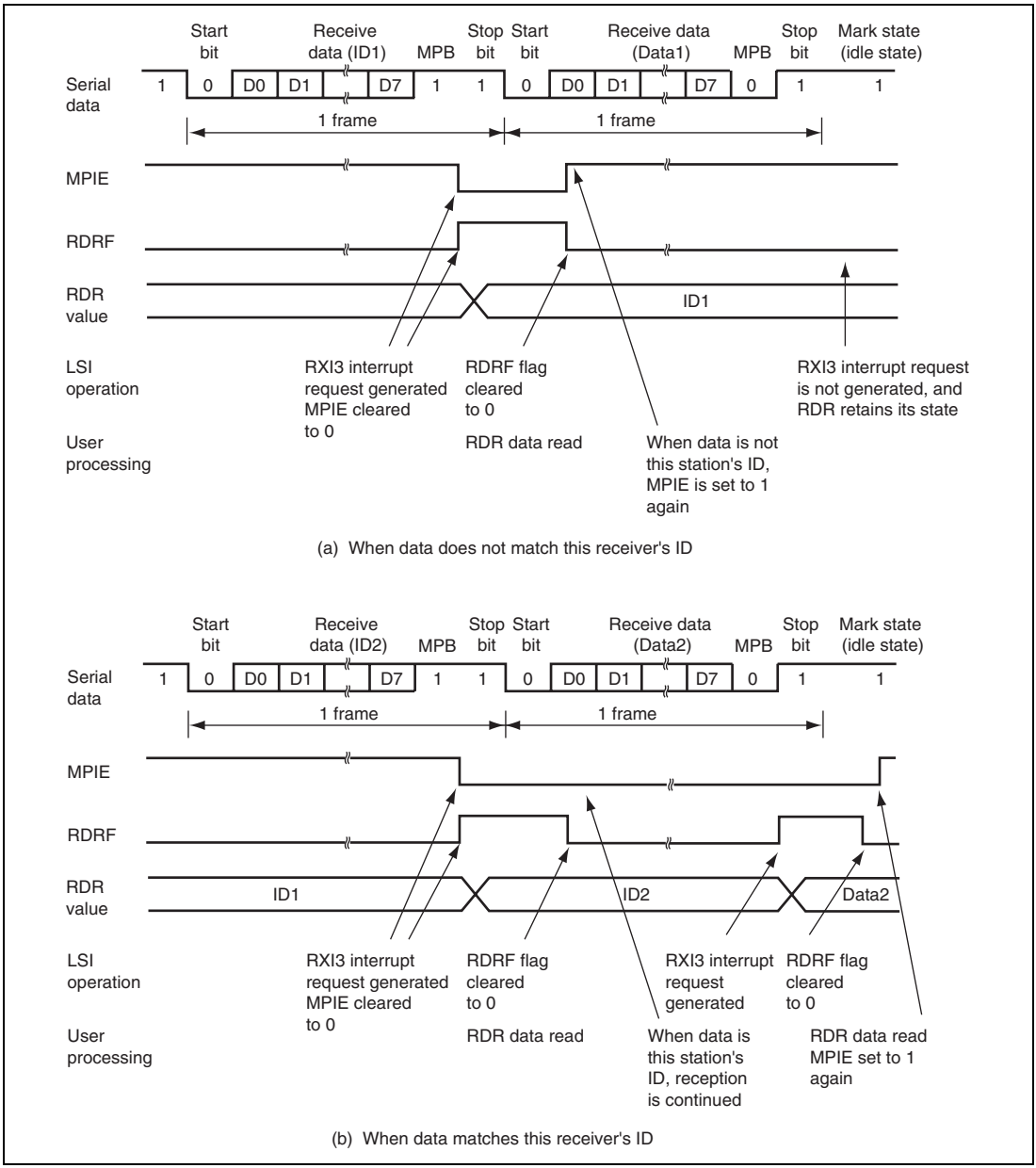


Figure 17.18 Example of SCI3 Operation in Reception Using Multiprocessor Format (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

17.7 IrDA Operation

IrDA operation can be used with the SCI3_1. Figure 17.19 shows an IrDA block diagram.

If the IrDA function is enabled using the IrE bit in IrCR, the TXD31 and RXD31 pins in the SCI3_1 are allowed to encode and decode the waveform based on the IrDA standard version 1.0 (function as the IrTXD and IrRXD pins). Connecting these pins to the infrared data transceiver/receiver achieves infrared data communication based on the system defined by the IrDA standard version 1.0.

In the system defined by the IrDA standard version 1.0, communication is started at a transfer rate of 9600 bps, which can be modified as required. The IrDA interface provided by this LSI does not incorporate the capability of automatic modification of the transfer rate; the transfer rate must be modified through programming.

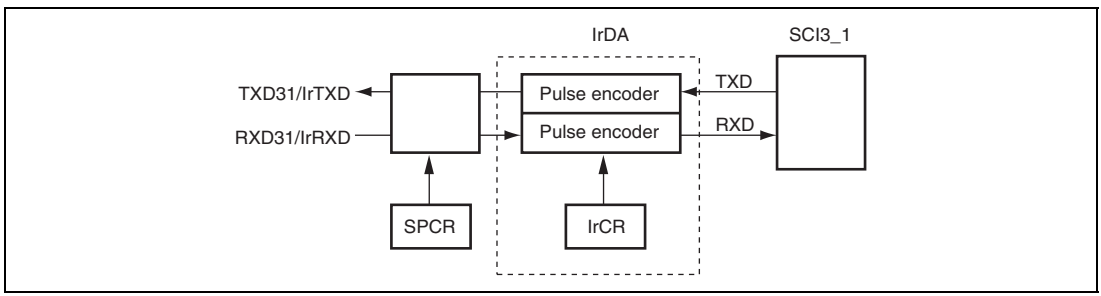


Figure 17.19 IrDA Block Diagram

17.7.1 Transmission

During transmission, the output signals from the SCI (UART frames) are converted to IR frames using the IrDA interface (see figure 17.20).

For serial data of level 0, a high-level pulse having a width of 3/16 of the bit rate (1-bit interval) is output (initial setting). The high-level pulse can be selected using the IrCKS2 to IrCKS0 bits in IrCR.

According to the standard, the high-level pulse width is defined to be 1.41 μs at minimum and (3/16 + 2.5%) x bit rate or (3/16 x bit rate) + 1.08 μs at maximum. For example, when the frequency of system clock ϕ is 10 MHz, being equal to or greater than 1.41 μs, the high-level pulse width at minimum can be specified as 1.6 μs.

For serial data of level 1, no pulses are output.

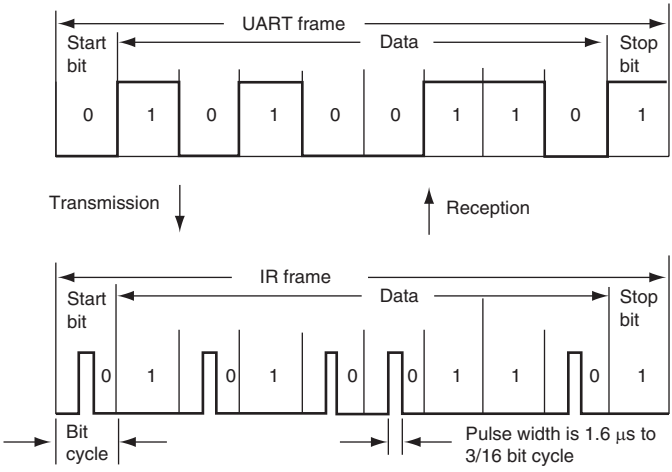


Figure 17.20 IrDA Transmission and Reception

17.7.2 Reception

During reception, IR frames are converted to UART frames using the IrDA interface before inputting to the SCI3_1.

Data of level 0 is output each time a high-level pulse is detected and data of level 1 is output when no pulse is detected in a bit cycle. If a pulse has a high-level width of less than 1.41 μs, the minimum width allowed, the pulse is not recognized.

17.7.3 High-Level Pulse Width Selection

Table 17.12 shows possible settings for bits IrCKS2 to IrCKS0 (minimum pulse width), and this LSI's operating frequencies and bit rates, for making the pulse width shorter than 3/16 times the bit rate in transmission.

Table 17.13 IrCKS2 to IrCKS0 Bit Settings

Operating Frequency	Bit Rate (bps) (Upper Row) / Bit Interval × 3/16 (μs) (Lower Row)			
	2400	9600	19200	38400
ϕ (MHz)	78.13	19.53	9.77	4.88
2	010	010	010	010
2.097152	010	010	010	010
2.4576	010	010	010	010
3	011	011	011	011
3.6864	011	011	011	011
4.9152	011	011	011	011
5	011	011	011	011
6	100	100	100	100
6.144	100	100	100	100
7.3728	100	100	100	100
8	100	100	100	100
9.8304	100	100	100	100
10	100	100	100	100

17.8 Interrupt Requests

The SCI3 creates the following six interrupt requests: transmit end, transmit data empty, receive data full, and receive errors (overrun error, framing error, and parity error). Table 17.14 shows the interrupt sources.

Table 17.14 SCI3 Interrupt Requests

Interrupt Requests	Abbreviation	Interrupt Sources
Receive data full	RXI3	Setting RDRF in SSR
Transmit data empty	TXI3	Setting TDRE in SSR
Transmission end	TEI3	Setting TEND in SSR
Receive error	ERI3	Setting OER, FER, and PER in SSR

Each interrupt request can be enabled or disabled by means of bits TIE and RIE in SCR.

When the TDRE bit in SSR is set to 1, a TXI3 interrupt is requested. When the TEND bit in SSR is set to 1, a TEI3 interrupt is requested. These two interrupts are generated during transmission.

The initial value of the TDRE flag in SSR is 1. Thus, when the TIE bit in SCR is set to 1 before transferring the transmit data to TDR, a TXI3 interrupt request is generated even if the transmit data is not ready. The initial value of the TEND flag in SSR is 1. Thus, when the TEIE bit in SCR is set to 1 before transferring the transmit data to TDR, a TEI3 interrupt request is generated even if the transmit data has not been sent. It is possible to make use of the most of these interrupt requests efficiently by transferring the transmit data to TDR in the interrupt routine. To prevent the generation of these interrupt requests (TXI3 and TEI3), set the enable bits (TIE and TEIE) that correspond to these interrupt requests to 1, after transferring the transmit data to TDR.

When the RDRF bit in SSR is set to 1, an RXI3 interrupt is requested, and if any of bits OER, PER, and FER is set to 1, an ERI3 interrupt is requested. These two interrupt requests are generated during reception.

The SCI3 can carry out continuous reception using an RXI3 and continuous transmission using a TXI3.

These interrupts are shown in table 17.15.

Table 17.15 Transmit/Receive Interrupts

Interrupt	Flags	Interrupt Request Conditions	Notes
RXI3	RDRF RIE	When serial reception is performed normally and receive data is transferred from RSR to RDR, bit RDRF is set to 1, and if bit RIE is set to 1 at this time, an RXI3 is enabled and an interrupt is requested. (See figure 17.21 (a).)	The RXI3 interrupt routine reads the receive data transferred to RDR and clears bit RDRF to 0. Continuous reception can be performed by repeating the above operations until reception of the next RSR data is completed.
TXI3	TDRE TIE	When TSR is found to be empty (on completion of the previous transmission) and the transmit data placed in TDR is transferred to TSR, bit TDRE is set to 1. If bit TIE is set to 1 at this time, a TXI3 is enabled and an interrupt is requested. (See figure 17.21 (b).)	The TXI3 interrupt routine writes the next transmit data to TDR and clears bit TDRE to 0. Continuous transmission can be performed by repeating the above operations until the data transferred to TSR has been transmitted.
TEI3	TEND TEIE	When the last bit of the character in TSR is transmitted, if bit TDRE is set to 1, bit TEND is set to 1. If bit TEIE is set to 1 at this time, a TEI3 is enabled and an interrupt is requested. (See figure 17.21 (c).)	A TEI3 indicates that the next transmit data has not been written to TDR when the last bit of the transmit character in TSR is transmitted.

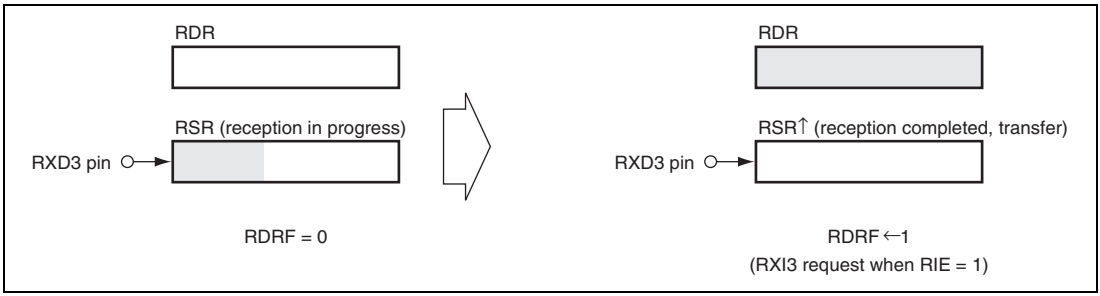


Figure 17.21 (a) RDRF Setting and RXI3 Interrupt

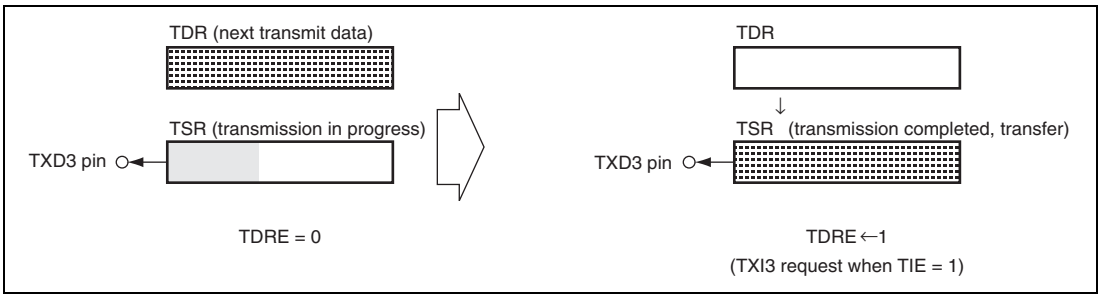


Figure 17.21 (b) TDRE Setting and TXI3 Interrupt

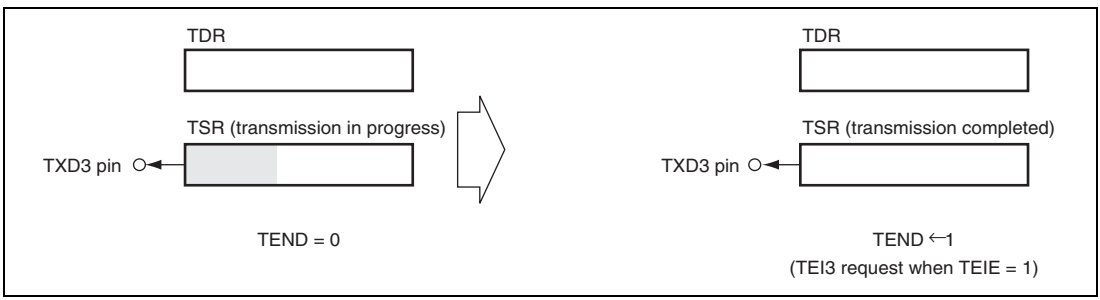


Figure 17.21 (c) TEND Setting and TEI3 Interrupt

17.9 Usage Notes

17.9.1 Break Detection and Processing

When framing error detection is performed, a break can be detected by reading the RXD3 pin value directly. In a break, the input from the RXD3 pin becomes all 0, setting the FER flag, and possibly the PER flag. Note that as the SCI3 continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

17.9.2 Mark State and Break Sending

When TE is 0, the TXD3 pin is used as an I/O port whose direction (input or output) and level are determined by PCR and PDR. This can be used to set the TXD3 pin to mark state (high level) or send a break during serial data transmission. To maintain the communication line at mark state until TE is set to 1, set both PCR and PDR to 1. As TE is cleared to 0 at this point, the TXD3 pin becomes an I/O port, and 1 is output from the TXD3 pin. To send a break during serial transmission, first set PCR to 1 and PDR to 0, and then clear TE to 0. When TE is cleared to 0, the transmitter is initialized regardless of the current transmission state, the TXD3 pin becomes an I/O port, and 0 is output from the TXD3 pin.

17.9.3 Receive Error Flags and Transmit Operations (Clock Synchronous Mode Only)

Transmission cannot be started when a receive error flag (OER, PER, or FER) is set to 1, even if 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 cleared to 0.

17.9.4 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI3 operates on a basic clock with a frequency of 16 times the transfer rate (8 times the transfer rate when the ABCS bit in SEMR is set to 1). In reception, the SCI3 samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 8th pulse of the basic clock (4th pulse of the basic clock when the ABCS bit in SEMR is set to 1*) as shown in figure 17.22.The reception margin in asynchronous mode is given by formula (1) below.

Note: Only supported by the SCI3_1 interface.

$$M = \left\{ (0.5 - \frac{1}{2N}) - \frac{D - 0.5}{N} - (L - 0.5)\,F \right\} \times 100(\%)$$

... Formula (1)

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

Assuming values of F (absolute value of clock rate deviation) = 0 and D (clock duty) = 0.5 in formula (1), the reception margin can be given by the formula.

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

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

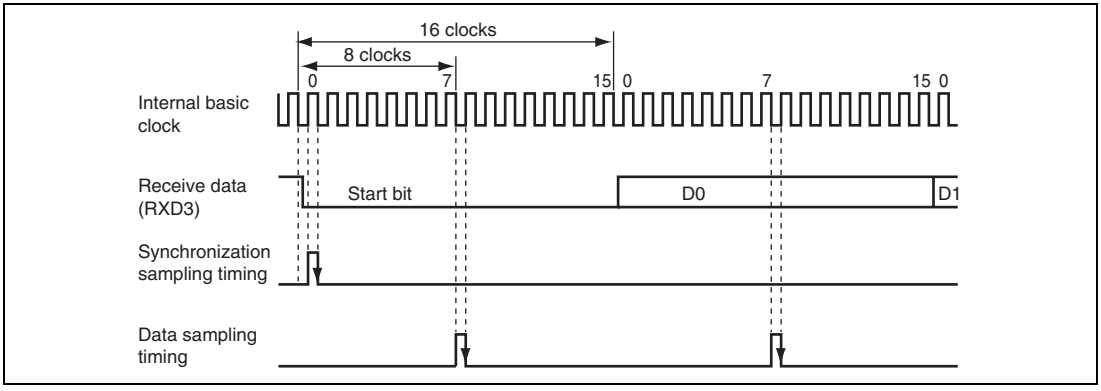


Figure 17.22 Receive Data Sampling Timing in Asynchronous Mode

17.9.5 Note on Switching SCK3 Pin Function

If pin SCK3 is used as a clock output pin by the SCI3 in clock synchronous mode and is then switched to a general input/output pin (a pin with a different function), the pin outputs a low level signal for half a system clock (ϕ) cycle immediately after it is switched.

This can be prevented by either of the following methods according to the situation.

(1) When SCK3 Function is Switched from Clock Output to Non Clock-Output

When stopping data transfer, issue one instruction to clear bits TE and RE to 0 and to set bits CKE1 and CKE0 in SCR to 1 and 0, respectively.

In this case, bit COM in SMR should be left 1. The above prevents the SCK3 pin from being used as a general input/output pin. To avoid an intermediate level of voltage from being applied to the SCK3 pin, the line connected to the SCK3 pin should be pulled up to the V_{CC} level via a resistor, or supplied with output from an external device.

(2) When SCK3 Function is Switched from Clock Output to General Input/Output

When stopping data transfer,

1. Issue one instruction to clear bits TE and RE to 0 and to set bits CKE1 and CKE0 in SCR to 1 and 0, respectively.
2. Clear bit COM in SMR to 0
3. Clear bits CKE1 and CKE0 in SCR to 0. Note that special care is also needed here to avoid an intermediate level of voltage from being applied to the SCK3 pin.

17.9.6 Relation between Writing to TDR and Bit TDRE

Bit TDRE in the serial status register (SSR) is a status flag that indicates that data for serial transmission has not been prepared in TDR. When data is written to TDR, bit TDRE is cleared to 0 automatically. When the SCI3 transfers data from TDR to TSR, bit TDRE is set to 1.

Data can be written to TDR irrespective of the state of bit TDRE, but if new data is written to TDR while bit TDRE is cleared to 0, the data previously stored in TDR will be lost if it has not yet been transferred to TSR. Accordingly, to ensure that serial transmission is performed dependably, you should first check that bit TDRE is set to 1, then write the transmit data to TDR only once (not two or more times).

17.9.7 Relation between RDR Reading and bit RDRF

In a receive operation, the SCI3 continually checks the RDRF flag. If bit RDRF is cleared to 0 when reception of one frame ends, normal data reception is completed. If bit RDRF is set to 1, this indicates that an overrun error has occurred.

When the contents of RDR are read, bit RDRF is cleared to 0 automatically. Therefore, if RDR is read more than once, the second and subsequent read operations will be performed while bit RDRF is cleared to 0. Note that, when an RDR read is performed while bit RDRF is cleared to 0, if the read operation coincides with completion of reception of a frame, the next frame of data may be read. This is shown in figure 17.23.

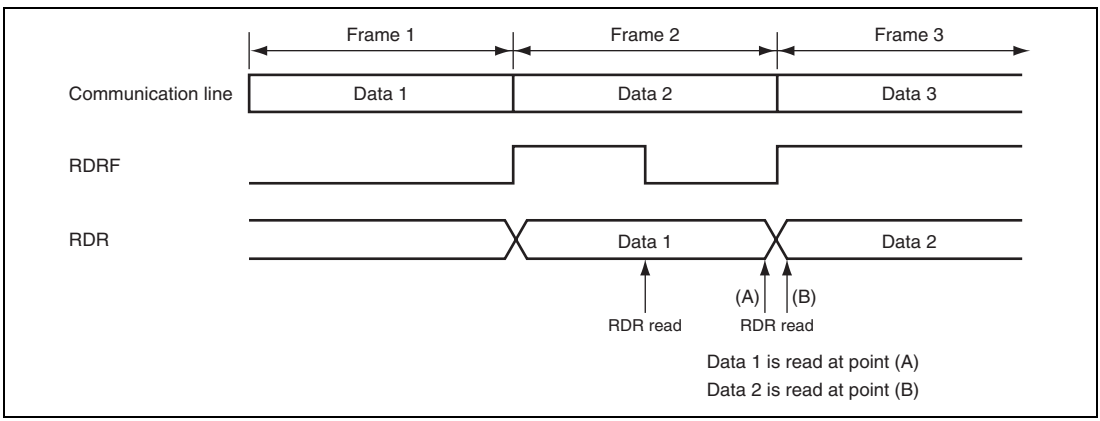


Figure 17.23 Relation between RDR Read Timing and Data

In this case, only a single RDR read operation (not two or more) should be performed after first checking that bit RDRF is set to 1. If two or more reads are performed, the data read the first time should be transferred to RAM, etc., and the RAM contents used. Also, ensure that there is sufficient margin in an RDR read operation before reception of the next frame is completed. To be precise in terms of timing, the RDR read should be completed before bit 7 is transferred in clock synchronous mode, or before the STOP bit is transferred in asynchronous mode.

17.9.8 Transmit and Receive Operations when Making State Transition

Make sure that transmit and receive operations have completely finished before carrying out state transition processing.

17.9.9 Setting in Subactive or Subsleep Mode

In subactive or subsleep mode, the SCI3 interface can operate only when the CPU clock is $\phi_w/2$. The SA1 and SA0 bits in SYSCR2 should be set to 1 and 0, respectively.

17.9.10 Oscillator when Serial Communications Interface 3 is Used

When the serial communications interface 3 is used, do not use the on-chip oscillator for the system clock. For details on selecting the system clock oscillator or on-chip oscillator for the system clock, see section 5.2.4, Selecting On-Chip Oscillator for System Clock.

Section 18 Serial Communication Interface 4 (SCI4)

The serial communication interface 4 (SCI4) can handle clock synchronous serial communication with the 8-bit buffer. The SCI4 is supported only by the F-ZTAT version. When the on-chip emulator debugger etc. is used, the SCK4, SI4, and SO4 pins of the SCI4 are used by the system, and the SCI4 is not available for the user.

18.1 Features

- Eight internal clocks (ϕ/1024, ϕ/256, ϕ/64, ϕ/32, ϕ/16, ϕ/8, ϕ/4, ϕ/2) or external clock can be selected as a clock source.
- Receive error detection: Overrun errors detected
- Four interrupt sources
  Transmit-end, transmit-data-empty, receive-data-full, and overrun error
- Full-duplex communication capability
  Buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data.
- When the on-chip emulator debugger etc. is not used, the SCI4 is available for the user.
- Use of module standby mode enables this module to be placed in standby mode independently when it is not in use. (For details, refer to section 6.4, Module Standby Function.)

Figure 18.1 shows a block diagram of the SCI4.

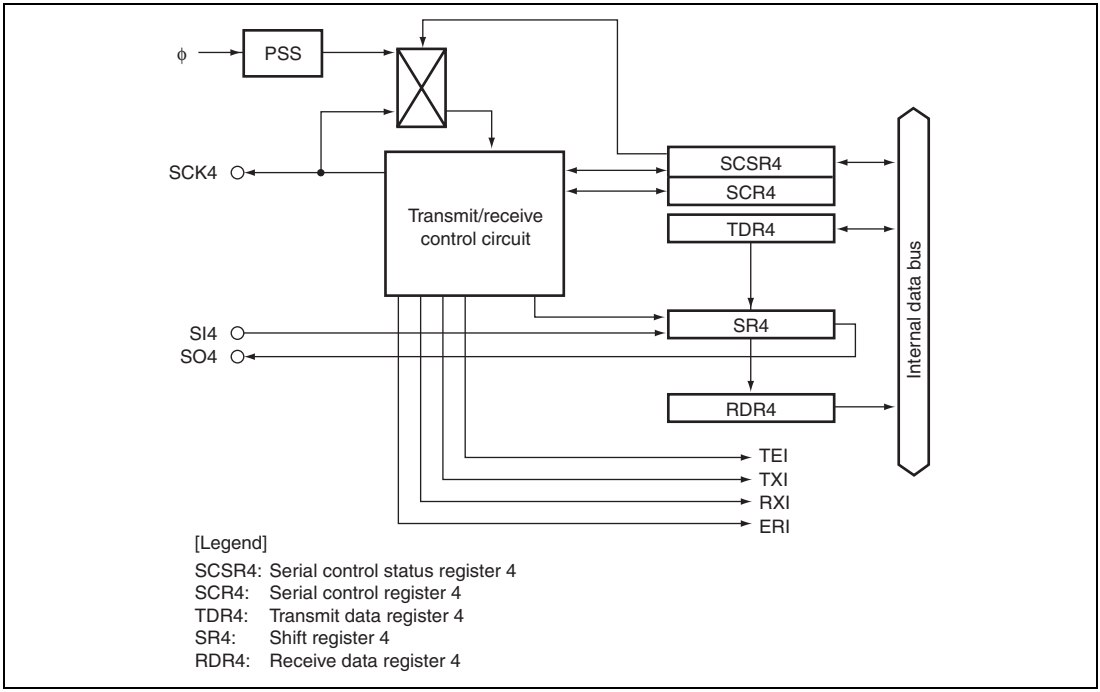


Figure 18.1 Block Diagram of SCI4

18.2 Input/Output Pins

Table 18.1 shows the SCI4 pin configuration.

Table 18.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
SCI4 clock	SCK4	I/O	SCI4 clock input/output
SCI4 data input	SI4	Input	SCI4 receive data input
SCI4 data output	SO4	Output	SCI4 transmit data output

18.3 Register Descriptions

The SCI4 has the following registers.

- Serial control register 4 (SCR4)
- Serial control/status register 4 (SCSR4)
- Transmit data register 4 (TDR4)
- Receive data register 4 (RDR4)
- Shift Register 4 (SR4)

18.3.1 Serial Control Register 4 (SCR4)

SCR4 enables or disables interrupt requests and controls SCI4 transfer operations.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables a transmit data empty interrupt (TXI) request when serial transmit data is transferred from TDR4 to SR4 and the TDRE flag in SCSR4 is set to 1. TXI can be cleared by clearing the TDRE flag in SCSR4 to 0 after the flag is read as 1 or clearing this bit to 0. 0: Transmit data empty interrupt (TXI) request disabled 1: Transmit data empty interrupt (TXI) request enabled
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables a receive data full interrupt (RXI) request and receive error interrupt (ERI) request when serial receive data is transferred from SR4 to RDR4 and the RDRF flag in SCSR4 is set to 1. RXI and ERI can be cleared by clearing the RDRF or ORER flag in SCSR4 to 0 after the flag is read as 1 or clearing this bit to 0. 0: Receive data full interrupt (RXI) request and receive error interrupt (ERI) request disabled 1: Receive data full interrupt (RXI) request and receive error interrupt (ERI) request enabled

Bit	Bit Name	Initial Value	R/W	Description
5	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables a transmit end interrupt (TEI) request when there is no valid transmit data in TDR4 during transmission of MSB data. TEI can be cleared by clearing the TEND flag in SCSR4 to 0 after the flag is read as 1 or clearing this bit to 0. 0: Transmit end interrupt (TEI) request disabled 1: Transmit end interrupt (TEI) request enabled
4	SOL	0	R/W	Extended Data Sets the output level of the SO4 pin. When this bit is read, the output level of the SO4 pin is read. The output of the SO4 pin retains the value of the last bit of transmit data after transmission is completed. However, if this bit is changed before or after transmission, the output level of the SO4 pin can be changed. When the output level of the SO4 pin is changed, the SOLP bit should be cleared to 0 and the MOV instruction should be used. Note that this bit should not be changed during transmission because incorrect operation may occur. [When reading] 0: The output level of the SO4 pin is low. 1: The output level of the SO4 pin is high. [When writing] 0: The output level of the SO4 pin is changed to low. 1: The output level of the SO4 pin is changed to high.
3	SOLP	1	R/W	SOL Write Protect Controls change of the output level of the SO4 pin due to the change of the SOL bit. When the output level of the SO4 pin is changed, the setting of SOL = 1 and SOLP = 0 or SOL = 0 and SOLP = 0 is made by the MOV instruction. This bit is always read as 1. 0: When writing, the output level is changed according to the value of the SOL pin. 1: When reading, this bit is always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
2	SRES	0	R/W	Forcible Reset When the internal sequencer is forcibly initialized, 1 should be written to this bit. When 1 is written to this flag, the internal sequencer is forcibly reset and then this flag is automatically cleared to 0. Note that the values of the internal registers are retained. (The TDRE flag in SCSR4 is set to 1 and the RDRF, ORER, and TEND flags are cleared to 0. The TE and RE bits in SCR4 are cleared to 0.) 0: Normal operation 1: Internal sequencer is forcibly reset
1	TE	0	R/W	Transmit Enable Enables or disables start of the SCI4 serial transmission. When this bit is cleared to 0, the TERE flag in SCSR4 is fixed to 1. When transmit data is written to TDR4 while this bit is set to 1, the TDRE flag in SCSR4 is automatically cleared to 0 and serial data transmission is started. 0: Transmission disabled (SO4 pin functions as I/O port) 1: Transmission enabled (SO4 pin functions as transmit data pin)
0	RE	0	R/W	Receive Enable Enables or disables start of the SCI4 serial reception. Note that the RDRF and ORER flags in SCSR4 are not affected even if this bit is cleared to 0, and retain their previous state. Serial data reception is started when the synchronous clock input is detected while this bit is set to 1 (when an external clock is selected). When an internal clock is selected, the synchronous clock is output and serial data reception is started. 0: Reception disabled (SI4 pin functions as I/O port) 1: Reception enabled (SI4 pin functions as receive data pin)

18.3.2 Serial Control/Status Register 4 (SCSR4)

SCSR4 indicates the operating state and error state, selects the clock source, and controls the prescaler division ratio.

SCSR4 can be read from or written to by the CPU at any time. 1 cannot be written to flags TDRE, RDRF, ORER, and TEND. To clear these flags to 0, 1 should be read from them in advance.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Empty Indicates that data is transferred from TDR4 to SR4 and the next serial transmit data can be written to TDR4. [Setting conditions] • The TE bit in SCR4 is 0 • Data is transferred from TDR4 to SR4 and data can be written to TDR4 [Clearing conditions] • Writing of 0 to bit TDRE after reading TDRE = 1 • Data is written to TDR4
6	RDRF	0	R/(W)*	Receive Data Full Indicates that the receive data is stored in RDR4. [Setting condition] • Serial reception ends normally and receive data is transferred from SR4 to RDR4 [Clearing conditions] • Writing of 0 to bit RDRF after reading RDRF = 1 • Data is read from RDR4

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/(W)*	Overrun Error Indicates that an overrun error occurs during reception and then abnormal termination occurs. In transfer mode, the output level of the SO4 pin is fixed to low while this flag is set to 1. When the RE bit in SCR4 is cleared to 0, the ORER flag is not affected and retains its previous state. When RDR4 retains the receive data it held before the overrun error occurred, and data received after the error is lost. Reception cannot be continued with the ORER flag set to 1, and transmission cannot be continued either. [Setting condition] • Next serial reception is completed while RDRF = 1 [Clearing condition] • Writing of 0 to bit ORER after reading ORER = 1
4	TEND	0	R/(W)*	Transmit End Indicates that the TDRE flag has been set to 1 at transmission of the last bit of transmit data. [Setting condition] • TDRE = 1 at transmission of the last bit of transmit data [Clearing conditions] • Writing of 0 to bit to TEND after reading TEND = 1 • Data is written to TDR4 with an instruction
3	CKS3	1	R/W	Clock Source Select and Pin Function Select the clock source to be supplied and set the input/output for the SCK4 pin. The prescaler division ratio and transfer clock cycle when an internal clock is selected are shown in table 18.2. When an external clock is selected, the external clock cycle should be at least 4/ϕ.
2	CKS2	0	R/W	
1	CKS1	0	R/W	
0	CKS0	0	R/W	

Note: * Only 0 can be written to clear the flag.

Table 18.2 shows a prescaler division ratio and transfer clock cycle.

Table 18.2 Prescaler Division Ratio and Transfer Clock Cycle (Internal Clock)

Bit 3	Bit 2	Bit 1	Bit 0	Prescaler Division Ratio	Transfer Clock Cycle		Function	
CKS3	CKS2	CKS1	CKS0		φ = 5 MHz	φ = 2.5 MHz	Clock Source	Pin Function
0	0	0	0	φ/1024	204.8 μs	409.6 μs	Internal clock	SCK4 output pin
0	0	0	1	φ/256	51.2 μs	102.4 μs	Internal clock	SCK4 output pin
0	0	1	0	φ/64	12.8 μs	25.6 μs	Internal clock	SCK4 output pin
0	0	1	1	φ/32	6.4 μs	12.8 μs	Internal clock	SCK4 output pin
0	1	0	0	φ/16	3.2 μs	6.4 μs	Internal clock	SCK4 output pin
0	1	0	1	φ/8	1.6 μs	3.2 μs	Internal clock	SCK4 output pin
0	1	1	0	φ/4	0.8 μs	1.6 μs	Internal clock	SCK4 output pin
0	1	1	1	φ/2	—	0.8 μs	Internal clock	SCK4 output pin
1	0	0	0	—	—	—	I/O port (initial value)	
1	0	0	1	—	—	—	I/O port	
1	0	1	0	—	—	—	I/O port	
1	0	1	1	—	—	—	I/O port	
1	1	0	0	—	—	—	I/O port	
1	1	0	1	—	—	—	I/O port	
1	1	1	0	—	—	—	I/O port	
1	1	1	1	—	—	—	External clock	SCK4 input pin

18.3.3 Transmit Data Register 4 (TDR4)

TDR4 is an 8-bit register that stores data for serial transmission. When the SCI4 detects that SR4 is empty, it transfers the transmit data written in TDR4 to SR4 and starts serial transmission. If the next transmit data is written to TDR4 while serial data in SR4 is being transmitted, continuous serial transmission is possible. TDR4 can be read from or written to by the CPU at any time. TDR4 is initialized to H'FF.

18.3.4 Receive Data Register 4 (RDR4)

RDR4 is an 8-bit register that stores receive data. When the SCI4 has received one byte of serial data, it transfers the received serial data from SR4 to RDR4, where it is stored. Then receive operation is completed. After this, SR4 is receive-enabled. RDR4 cannot be written to by the CPU. RDR4 is initialized to H'00.

18.3.5 Shift Register 4 (SR4)

SR4 is a register that receives or transmits serial data. SR4 cannot be directly read from or written to by the CPU.

18.4 Operation

The SCI4 is a serial communication interface that transmits and receives data in synchronization with a clock pulse and is suitable for high-speed serial communications. The data transfer format is fixed to 8-bit data. The internal clock or external clock can be selected as a clock source. An overrun error during reception can be detected. The transmit and receive units are configured with double buffering mechanism. Since the mechanism enables to write data during transmission and to read data during reception, data is consecutively transmitted and received.

18.4.1 Clock

The eight internal clocks or an external clock can be selected as a transfer clock. When the external clock is selected, the SCK4 pin is a clock input pin. When the internal clock is selected, the SCK4 pin is a synchronous clock output pin. The synchronous clock is output eight pulses for 1-character transmission or reception. While neither transmission nor reception is being performed, the signal is fixed high.

When the internal clock or external clock is not selected according to the combination of the CKS3 to CKS0 bits in SCSR4, the SCK4 pin functions as an I/O port.

18.4.2 Data Transfer Format

Figure 18.2 shows the SCI4 transfer format.

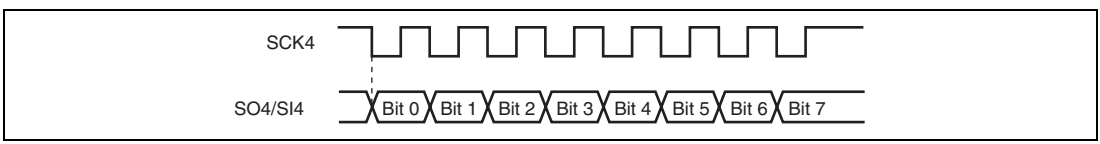


Figure 18.2 Data Transfer Format

In clock synchronous communication, data on the communication line is output from the falling edge to the next falling edge of the synchronous clock. The data is guaranteed to be settled at the rising edge of the synchronous clock. One character starts with the LSB and ends with the MSB. After transmitting the MSB, the communication line retains the MSB level.

The SCI4 latches data at the rising edge of the synchronous clock on reception. The data transfer format is fixed to 8-bit data. While transmission is stopped, the output level on the SO4 pin can be changed by the SOL setting in SCR4.

18.4.3 Data Transmission/Reception

Before data transmission and reception, clear the TE and RE bits in SCR4 to 0 and then initialize as the following procedure of figure 18.3.

Note: Before changing operating modes or communication format, the TE and RE bits must be cleared to 0. Clearing the TE bit to 0 sets the TDRE flag to 1. Note that clearing the RE bit to 0 does not affect the RDRF or ORER flag and the contents of RDR4.
When the external clock is used, the clock must not be supplied during operation including initialization.

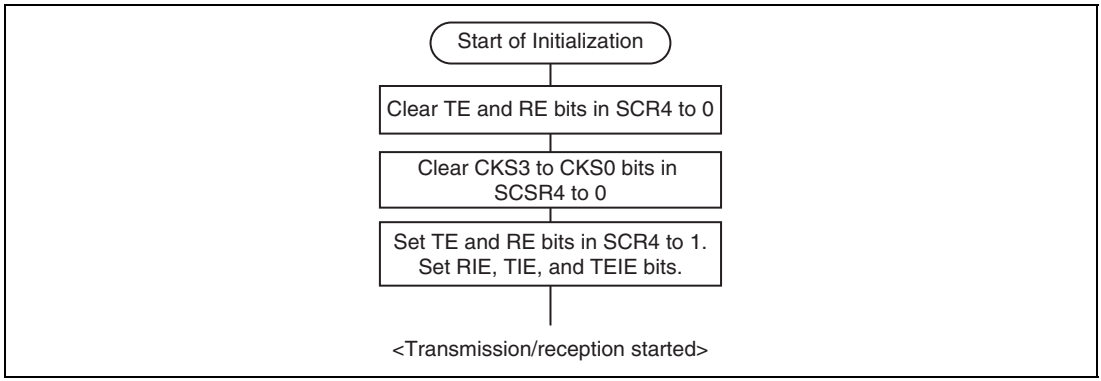


Figure 18.3 Flowchart Example of SCI4 Initialization

18.4.4 Data Transmission

Figure 18.4 shows an example flowchart of data transmission. Data transmission should be performed as the following procedure after the SCI4 initialization.

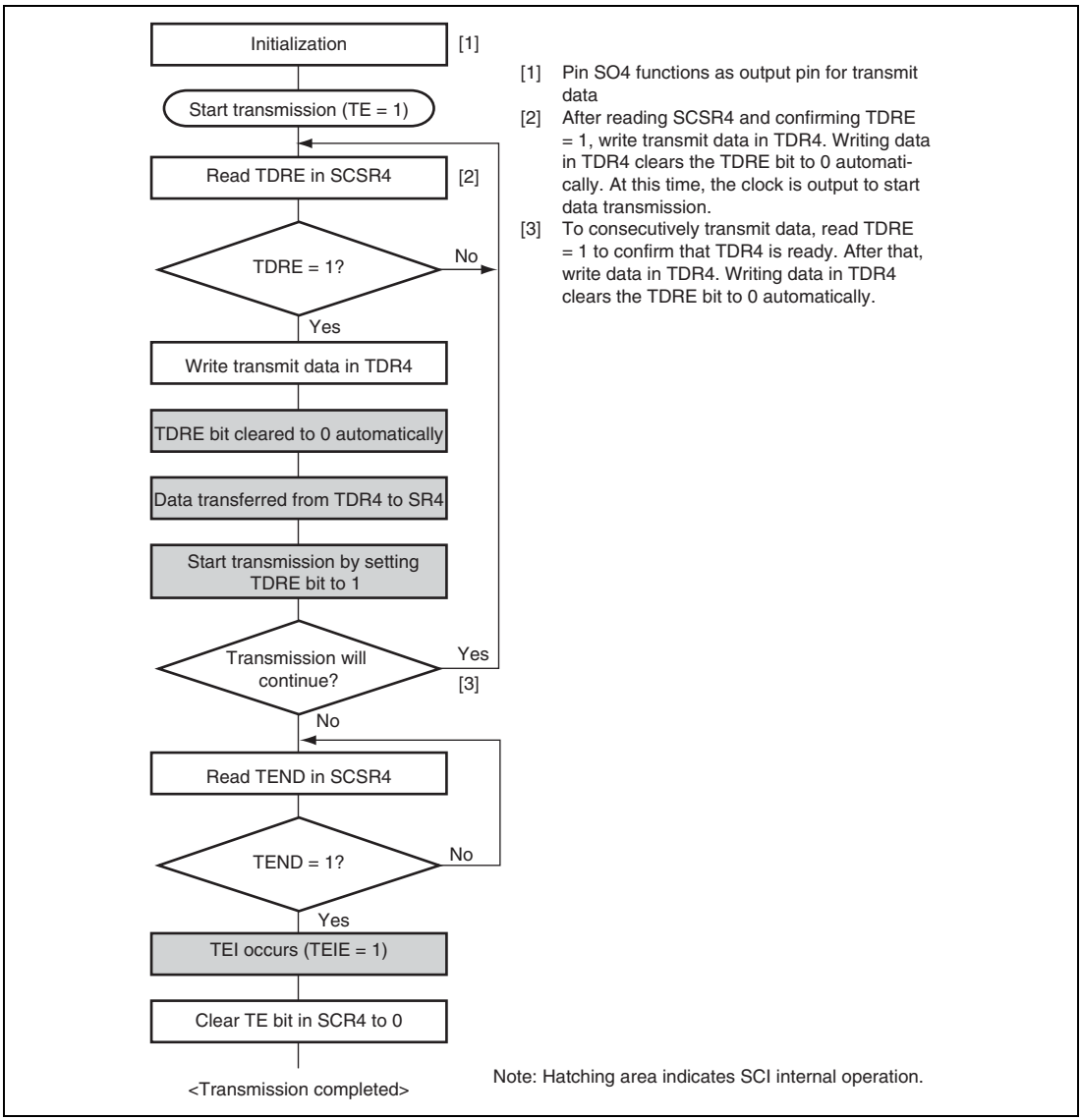


Figure 18.4 Flowchart Example of Data Transmission

During transmission, the SCI4 operates as shown below.

1. The SCI4 sets the TE bit to 1 and clears the TDRE flag to 0 when transmit data is written to in TDR4 to transmit data from TDR4 to SR4. After that, the SCI4 sets the TDRE flag to 1 to start transmission. At this time, when the TIE bit in SCR4 is set to 1, a TXI is generated.
2. In clock output mode, the SCI4 outputs eight pulses of the synchronous clock. When the external clock is selected, the SCI4 outputs data in synchronization with the input clock.
3. Serial data is output from the LSB (bit 0) to MSB (bit 7) on pin SO4. The SCI4 checks the TDRE flag at the timing of outputting the MSB (bit 7).
4. When TDRE = 0, data in TDR4 is transmitted to SR4 and then the data of the next frame starts to be transmitted. When TDRE = 1, the SCI4 sets the TEND bit to 1 and holds the output level after transmitting the MSB (bit 7). At this time, when the TEIE bit in SCR4 is set to 1, a TEI is generated.
5. After the transmission, the output level on pin SCK4 is fixed high.

Note: Transmission cannot be performed when the error flag (ORER) which indicates the data reception status is set to 1. Before transmission, confirm that the ORER flag is cleared to 0.

Figure 18.5 shows the example of transmission operation.

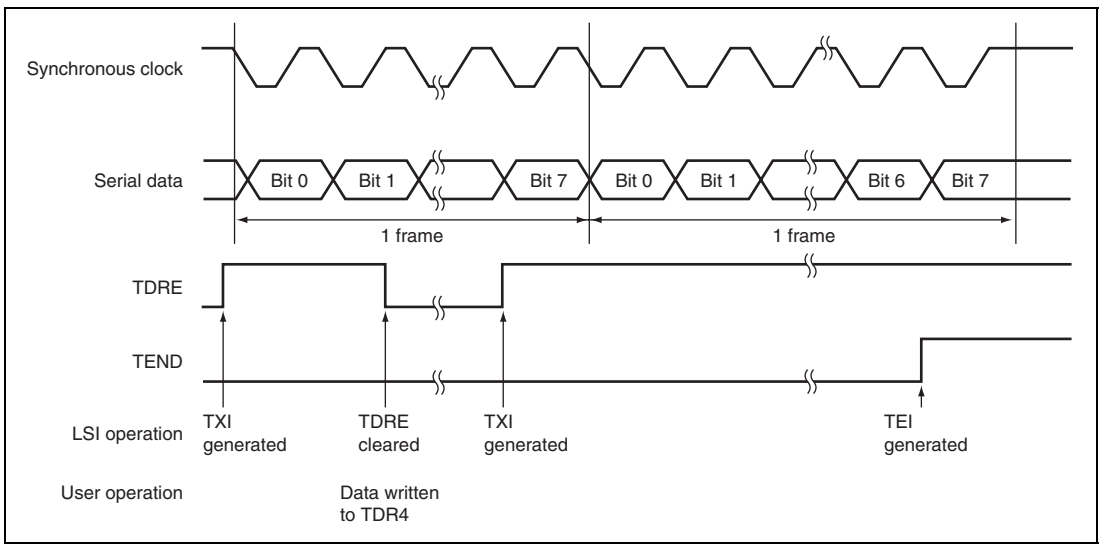


Figure 18.5 Transmit Operation Example

18.4.5 Data Reception

Figure 18.6 shows an example flowchart of data reception. Data reception should be performed as the following procedure after the SCI4 initialization.

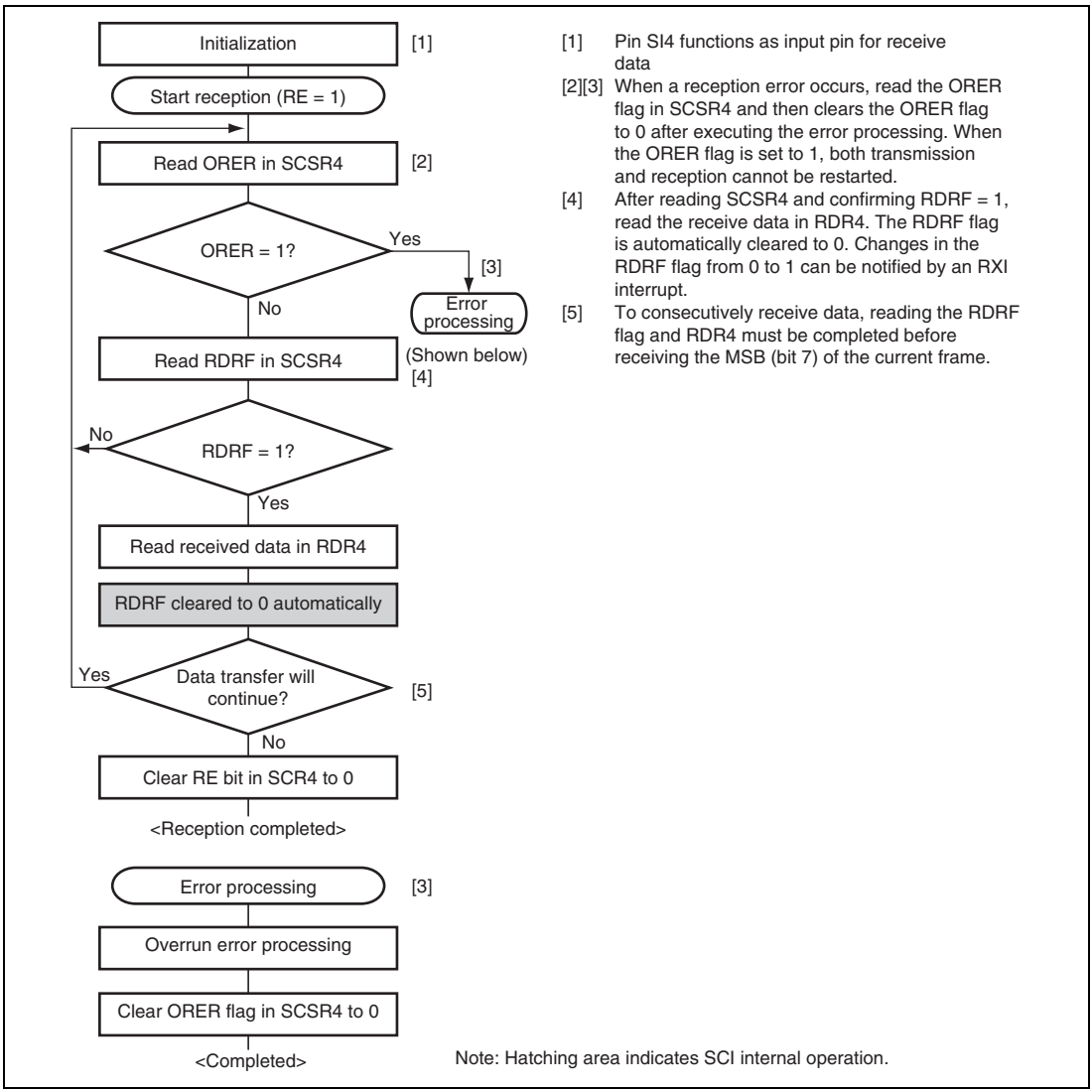


Figure 18.6 Flowchart Example of Data Reception

During reception, the SCI4 operates as shown below.

1. The SCI4 initialization is performed in synchronization with the synchronous clock input or output and starts reception.
2. The SCI4 stores received data from the LSB to MSB of SR4.
3. After reception, the SCI4 checks that RDRF = 0 and whether receive data is ready for being transferred from SR4 to RDR4.
4. When confirms that an overrun error has not occurred, the RDRF bit is set to 1 and the received data is stored in RDR4. At this time, when the RIE bit in SCR4 is set to 1, an RXI is generated. When an overrun error is detected by checking, the ORER flag is set to 1. The RDRF bit retains the previously set value. If the RIE bit in SCR4 is set to 1, an ERI is generated.
5. An overrun error is detected when the next data reception is completed with the RDRF bit in SCSR4 set to 1. The received data is not transferred from SR4 to RDR4.

Note: Reception cannot be performed when the error flag is set to 1. Before reception, confirm that the ORER and RDRF flags are cleared to 0.

Figure 18.7 shows an operation example of reception.

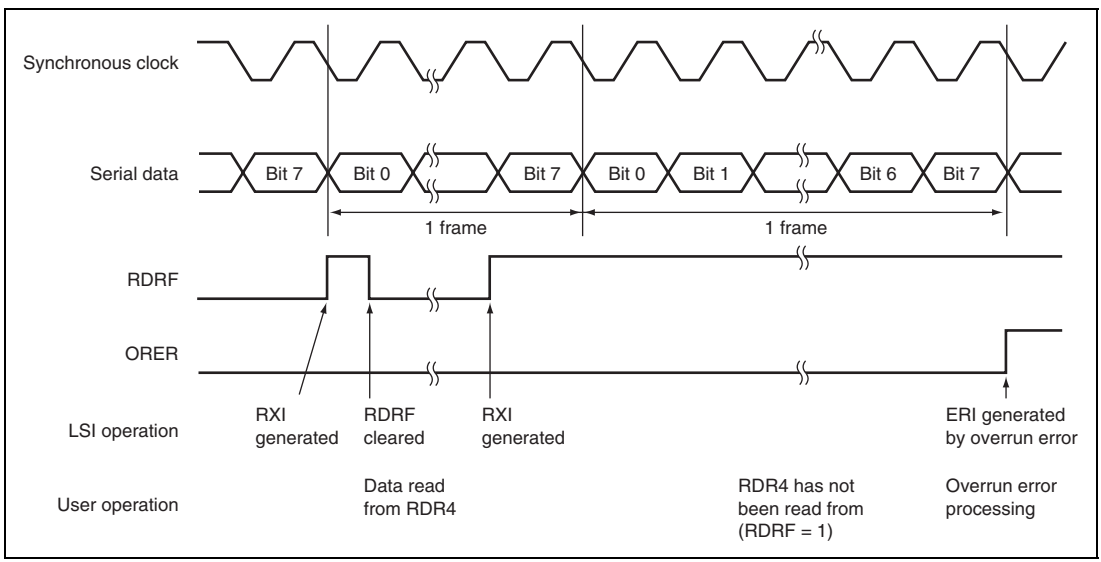
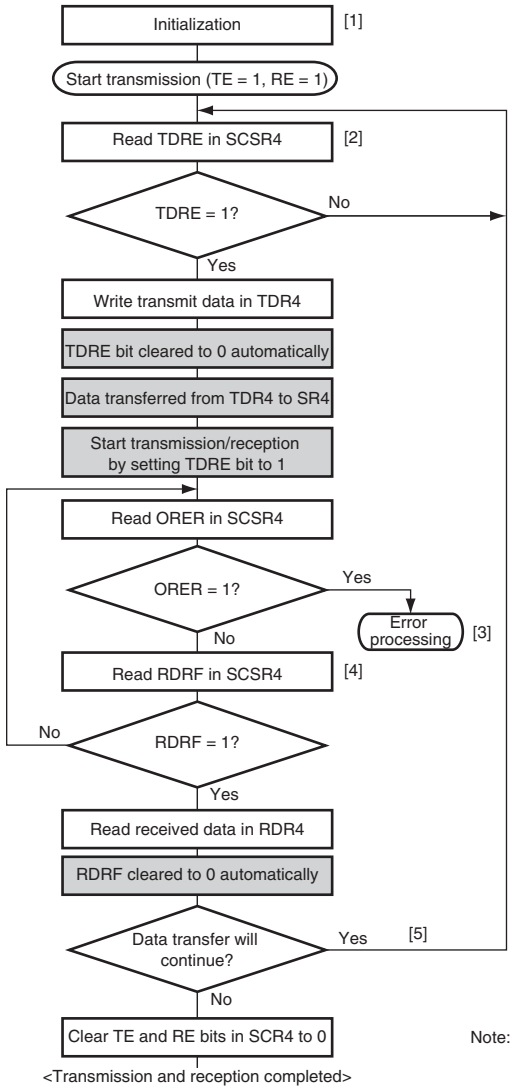


Figure 18.7 Receive Operation Example

18.4.6 Simultaneous Data Transmission and Reception

Figure 18.8 shows an example flowchart of simultaneous data transmission and reception. Simultaneous data transmission and reception should be performed as the following procedure after the SCI4 initialization.



[1] Pin SO4 functions as output pin for transmit data and pin SI4 functions as input pin for receive data. Simultaneous transmission and reception is enabled.

[2] After reading SCSR4 and confirming TDRE = 1, write transmit data in TDR4. Writing data in TDR4 clears the TDRE bit to 0 automatically. At this time, the clock is output to start data transfer.

[3] When a reception error occurs, read the ORER flag in SCSR4 and then clear the ORER flag to 0 after executing the error processing. When the ORER flag is set to 1, both transmission and reception cannot be restarted.

[4] After reading SCSR4 and confirming RDRF = 1, read receive data in RDR4 and clear the RDRF flag to 0. An RXI interrupt can also be used to confirm that the RDRF flag value has been changed from 0 to 1.

[5] To consecutively transmit and receive data, the following operation must be completed: reading the RDRF flag and reading RDR4 before receiving the MSB (bit 7) of the current frame: confirming that TDR4 is ready for writing by reading TDRE = 1 before transmitting the MSB (bit 7) and writing data to TDR4 to clear the TDRE flag to 0.

Note: Hatching area indicates SCI internal operation.

Figure 18.8 Flowchart Example of Simultaneous Transmission and Reception

Notes: 1. When switching from transmission to simultaneous data transmission and reception, confirm that the SCI4 completes transmission and both the TDRE and TEND bits are set to 1. After that, clear the TE bit to 0 and then set both the TE and RE bits to 1.

2. When switching from reception to simultaneous data transmission and reception, confirm that the SCI4 completes reception and both the RDRF and ORER flags are cleared to 0 after clearing the RE bit to 0. After that, set both the TE and RE bits to 1.

18.5 Interrupt Sources

The SCI4 has four interrupt sources: transmit end, transmit data empty, receive data full, and receive error (overrun error).

Table 18.3 lists the descriptions of the interrupt sources.

Table 18.3 SCI4 Interrupt Sources

Abbreviation	Condition	Interrupt Source
RXI	RIE = 1	Receive data full (RDRF)
TXI	TIE = 1	Transmit data empty (TDRE)
TEI	TEIE = 1	Transmit end (TEND)
ERI	RIE = 1	Receive error (ORER)

The interrupt requests can be enabled/disabled by the TIE and RIE bits in SCR4.

When the TDRE flag in SCSR4 is set to 1, a TXI is generated. When the TEND bit in SCSR4 is set to 1, a TEI is generated. These two interrupt requests are generated during transmission.

The TDRE flag in SCSR4 is initialized to 1. Therefore, if a TXI request is enabled by setting the TIE bit in SCR4 to 1 before transmit data is transferred to TDR4, a TXI is generated even when transmit data is not ready.

If transmit data is transferred to TDR4 in the interrupt handling routine, these interrupt requests can be effectively used.

To avoid the occurrence of the interrupt requests (TXI and TEI), clear the corresponding interrupt enable bits (TIE and TEIE) to 0 after transmit data is transferred to TDR4.

When the RDRF bit in SCSR4 is set to 1, an RXI is generated. When the ORER flag is set to 1, an ERI is generated. These two interrupt requests are generated during reception.

18.6 Usage Notes

When using the SCI4, keep in mind the following.

18.6.1 Relationship between Writing to TDR4 and TDRE

The TDRE flag in SCSR4 is a status flag that indicates that data to be transmitted has not been stored in TDR4. When writing data to TDR4, the TDRE flag is automatically cleared to 0. The TDRE flag is set to 1 when the SCI4 transfers data from TDR4 to SR4.

Data is written to TDR4 regardless of the TDRE flag value. However, if data is written to TDR4 with TDRE = 0, the previous data is lost unless the previous data has been transferred to SR4. Accordingly, to ensure transmission, writing transmit data to TDR4 must be performed once after confirming that the TDRE flag has been set to 1. (Do not write more than once.)

18.6.2 Receive Error Flag and Transmission

While the receive error flag (ORER) is set to 1, transmission cannot be started even if the TDRE flag is cleared to 0. To start transmission, the ORER flag must be cleared to 0.

Note that the ORER flag cannot be cleared to 0 even if the RE bit is cleared to 0.

18.6.3 Relationship between Reading RDR4 and RDRF

The SCI4 always checks the RDRF flag status during reception. When the RDRF flag is cleared to 0 at the end of a frame, the reception is completed without error. When the RDRF flag is set to 1, it indicates that an overrun has occurred.

Since reading RDR4 clears the RDRF flag to 0 automatically, if RDR4 is read twice or more, the data is read with the RDRF flag cleared to 0. In this case, when the timing of the read operation matches that of the data reception of the next frame, the read data may be the next frame data. Figure 18.9 shows this operation.

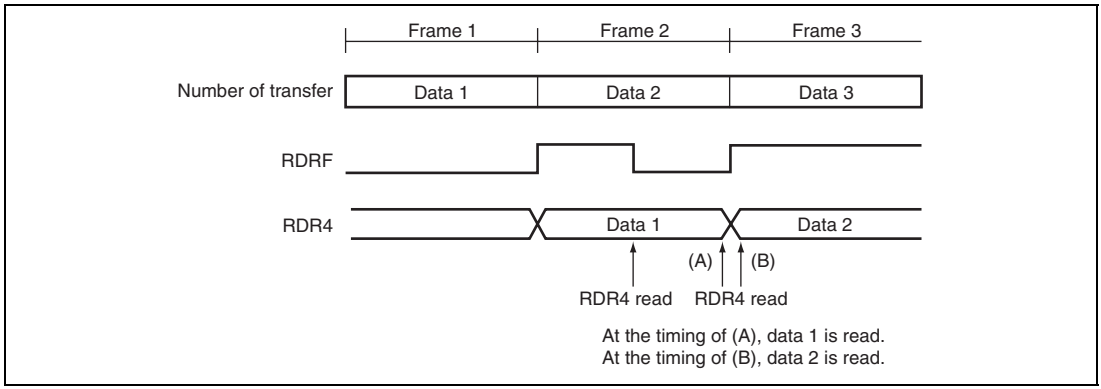


Figure 18.9 Relationship between Reading RDR4 and RDRF

In this case, RDR4 must be read only once after confirming RDRF = 1. If reading RDR4 twice or more, store the read data in the RAM, and use the stored data. In addition, there should be a margin from the timing of reading RDR4 to completion of the next frame reception (reading RDR4 should be completed before the bit 7 transfer).

18.6.4 SCK4 Output Waveform when Internal Clock of ϕ/2 is Selected

When the internal clock of ϕ/2 is selected by the CKS3 to CKS0 bits in SCSR4 and continuous transmission or reception is performed, one pulse of high period is lengthened after eight pulses of the clock has been output as shown in figure 18.10.

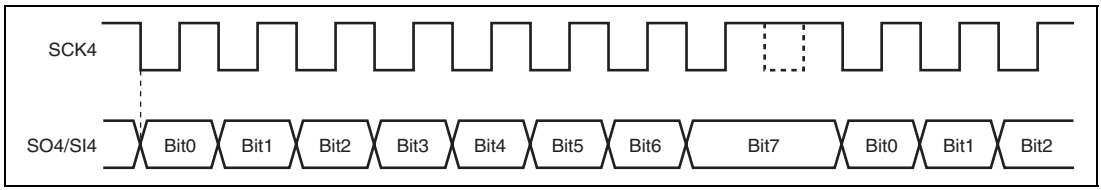


Figure 18.10 Transfer Format when Internal Clock of ϕ/2 is Selected

Section 19 14-Bit PWM

This LSI has an on-chip 14-bit pulse width modulator (PWM) with four channels. Connecting the PWM to the low-pass filter enables the PWM to be used as a D/A converter. The standard PWM or pulse-division type PWM can be selected by software. Figure 19.1 shows a block diagram of the 14-bit PWM.

19.1 Features

- Choice of four conversion periods

  A conversion period of 131,072/ϕ with a minimum modulation width of 8/ϕ, a conversion period of 65,536/ϕ with a minimum modulation width of 4/ϕ, a conversion period of 32,768/ϕ with a minimum modulation width of 2/ϕ, or a conversion period of 16,384/ϕ with a minimum modulation width of 1/ϕ, can be selected.
- Pulse division method for less ripple
- Use of module standby mode enables this module to be placed in standby mode independently when not used (for details, refer to section 6.4, Module Standby Function).
- The standard PWM or pulse-division type PWM can be selected by software.

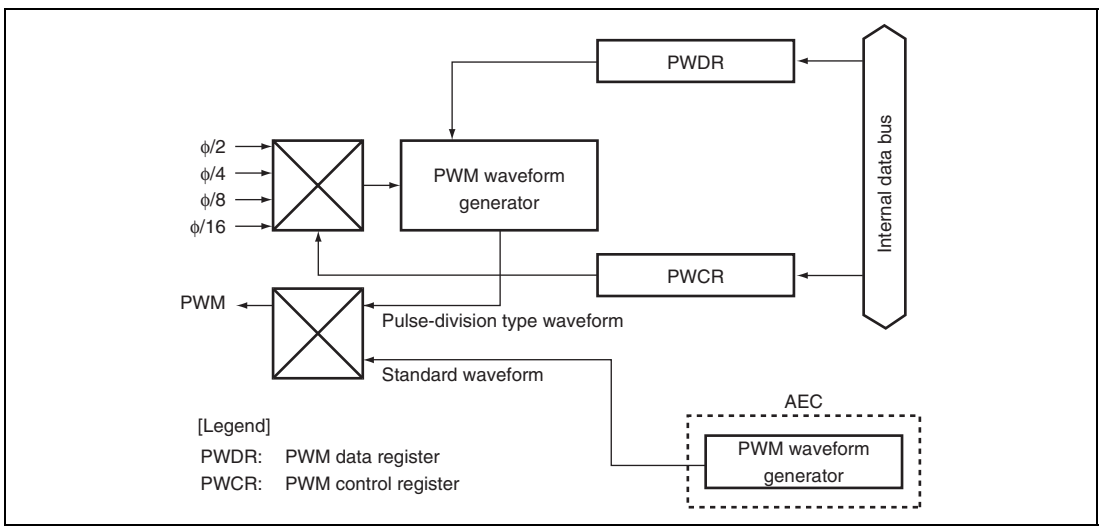


Figure 19.1 Block Diagram of 14-Bit PWM

19.2 Input/Output Pins

Table 19.1 shows the 14-bit PWM pin configuration.

Table 19.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
PWM1 output pin	PWM1	Output	Standard PWM/pulse-division type PWM waveform output (PWM1)
PWM2 output pin	PWM2	Output	Standard PWM/pulse-division type PWM waveform output (PWM2)
PWM3 output pin	PWM3	Output	Standard PWM/pulse-division type PWM waveform output (PWM3)
PWM4 output pin	PWM4	Output	Standard PWM/pulse-division type PWM waveform output (PWM4)

19.3 Register Descriptions

The 14-bit PWM has the following registers.

- PWM1 control register (PWCR1)
- PWM1 data register (PWDR1)
- PWM2 control register (PWCR2)
- PWM2 data register (PWDR2)
- PWM3 control register (PWCR3)
- PWM3 data register (PWDR3)
- PWM4 control register (PWCR4)
- PWM4 data register (PWDR4)

19.3.1 PWM Control Register (PWCR)

PWCR selects the input clocks and selects whether the standard PWM or pulse-division type PWM is used.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
2	PWCRm2	0	W	PWM Output Waveform Select Selects whether the standard PWM waveform or pulse-division type PWM waveform is output. 0: Pulse-division type PWM waveform is output 1: Standard PWM waveform is output
1	PWCRm1	0	W	Clock Select 1 and 0
0	PWCRm0	0	W	Select the clock supplied to the 14-bit PWM. These bits are write-only bits and always read as 1. 00: The input clock is ϕ/2 — A conversion period is 16,384/ϕ, with a minimum modulation width of 1/ϕ 01: The input clock is ϕ/4 — A conversion period is 32,768/ϕ, with a minimum modulation width of 2/ϕ 10: The input clock is ϕ/8 — A conversion period is 65,536/ϕ, with a minimum modulation width of 4/ϕ 11: The input clock is ϕ/16 — A conversion period is 131,072/ϕ, with a minimum modulation width of 8/ϕ

Note: m = 4 to 1

19.3.2 PWM Data Register (PWDR)

PWDR is a 14-bit write-only register. PWDR indicates the high-level width in one pulse period of the PWM waveform when the pulse-division type PWM is selected.

When data is written to the lower 14 bits of PWDR, the contents are latched in the PWM waveform generator and the PWM waveform generation data is updated.

PWDR is initialized to 0 and always read as H'FFFF. Writing to this register should be in a word unit.

19.4 Operation

19.4.1 Principle of Pulse-Division Type PWM

In pulse-division type PWM, the high-level and low-level periods of the ordinary PWM waveform are divided and output alternately. This can reduce the ripples generated when the PWM module is used as a D/A converter by connecting a low-pass filter to it.

Figure 19.2 shows an example of waveform when the pulse is divided into four. The 14-bit PWM module divides the pulse into 64.

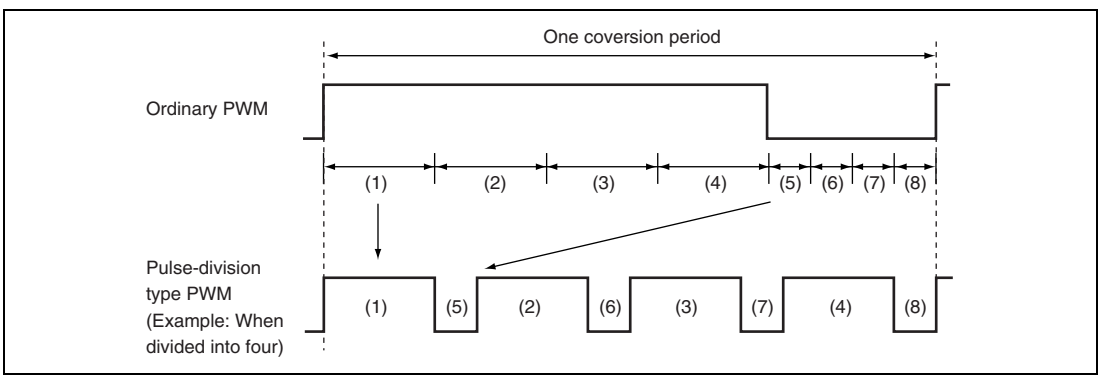


Figure 19.2 Example of Waveform Produced by Pulse-Division Type PWM (Division by 4)

19.4.2 Setting for Pulse-Division Type PWM Operation

When using the pulse-division type PWM, set the registers in this sequence:

1. In accord with the PWM channel to be used, set the PWM1, PWM2, PWM3, or PWM4 bit (the former two bits are in PMR9 and the latter two in PFCR) to 1 to specify the P90/PWM1, P91/PWM2, P92/$\overline{\text{IRQ4}}$/PWM3, or P93/PWM4 pin, respectively, to function as a PWM output pin.
2. Set PWCR to select a conversion period.
3. Set the data for output waveform in PWDR. When the data is written to PWDR, the contents are latched in the PWM waveform generator, and the PWM waveform generation data is updated.

19.4.3 Operation of Pulse-Division Type PWM

One conversion period consists of 64 pulses, as shown in figure 19.3. The total high-level width during this period (T_H) corresponds to the data in PWDR. This relation is given in table 19.2.

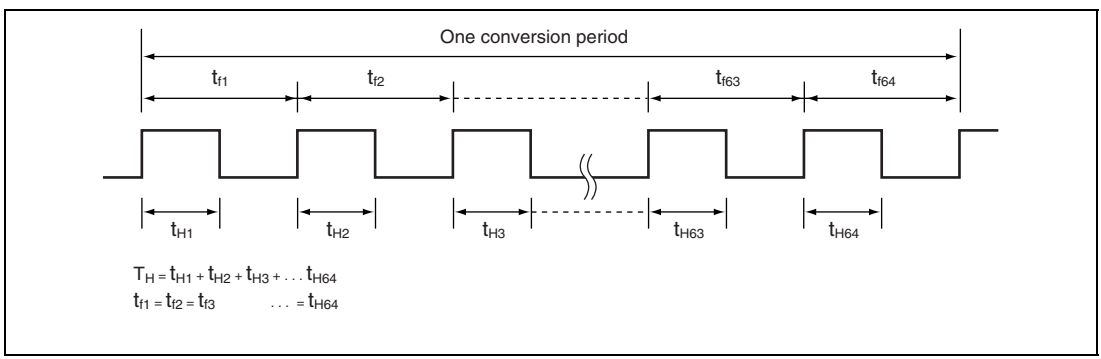


Figure 19.3 Waveform Output by PWM

Table 19.2 Relationship between PWCR, PWDR and Output Waveform

PWCRm Setting Value					
PWCRm1	PWCRm0	One Conversion Period [t_{cyc}]	T_H [t_{cyc}]	T_{fn} [t_{cyc}]	Minimum Variation Width [t_{cyc}]
0	0	16384	(PWDRm+64) × 1	256	1
0	1	32768	(PWDRm+64) × 2	512	2
1	0	65536	(PWDRm+64) × 4	1024	4
1	1	131072	(PWDRm+64) × 8	2048	8

Note: m = 1 to 4, n = 1 to 64

19.4.4 Setting for Standard PWM Operation

When using the standard PWM, set the registers in this sequence:

1. According to the PWM channel used, set the PWM1, PWM2, PWM3, or PWM4 bit (the former two bits are in PMR9 and the latter two in PFCR) to 1 to set the P90/PWM1, P91/PWM2, P92/$\overline{\text{IRQ4}}$/PWM3, or P93/PWM4 pin to function as a PWM pin.
2. Set PWCRm2 to 1 to select the standard PWM waveform. (m = 4 to 1)
3. Set the event counter PWM in the asynchronous event counter. For the setting method, see section 15.4.4, Event Counter PWM Operation.
4. The PWM pin outputs the PWM waveform set by the event counter.

Note: When the standard waveform is used, 16-bit counter operation, 8-bit counter operation, and IRQAEC operation for the asynchronous event counter are not available because the PWM for the asynchronous event counter is used.
When the IECPWM signal of the asynchronous event counter goes high, ECH and ECL increment. However, when the signal goes low, these counters stop. (For details, refer to section 15.4, Operation.)

19.4.5 PWM Operating States

The states of PWM module registers in each operating mode are shown in table 19.3.

Table 19.3 PWM Operating States

Operating Mode	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module Standby
PWCRm	Reset	Functioning	Functioning	Retained	Retained	Retained	Retained	Retained
PWDRm	Reset	Functioning	Functioning	Retained	Retained	Retained	Retained	Retained

Note: m = 4 to 1

19.5 Usage Notes

19.5.1 Timing of Writing to PWDR and Reflection in the PWM Waveform

If the PWDR register is rewritten while a PWM waveform is output, the operation will be as follows depending on in which state of the PWM waveform the writing was performed.

- During low level output: The new PWDR value is reflected from the next pulse.
- During high level output:
  - — When the new value increases the duty cycle
    The new PWDR value is reflected immediately after writing.
  - — When the new value decreases the duty cycle
    - If the high-level width of the pulse at the time of rewriting exceeds the high-level width specified by the new PWDR value, a high level is output for a single pulse period.
    - If the high-level width of the pulse at the time of rewriting does not exceed the high-level width specified by the new PWDR value, the new value is reflected immediately after writing.

Section 20 A/D Converter

This LSI includes a successive approximation type 10-bit A/D converter that allows up to eight analog input channels to be selected. The block diagram of the A/D converter is shown in figure 20.1.

20.1 Features

- 10-bit resolution
- Input channels: Eight channels
- High-speed conversion: 12.4 μs per channel (in 10-MHz operation)
- Sample and hold function
- Conversion start method

  A/D conversion can be started by software and external trigger.
- Interrupt source

  An A/D conversion end interrupt request can be generated.
- Use of module standby mode enables this module to be placed in standby mode independently when it is not in use (for details, refer to section 6.4, Module Standby Function).

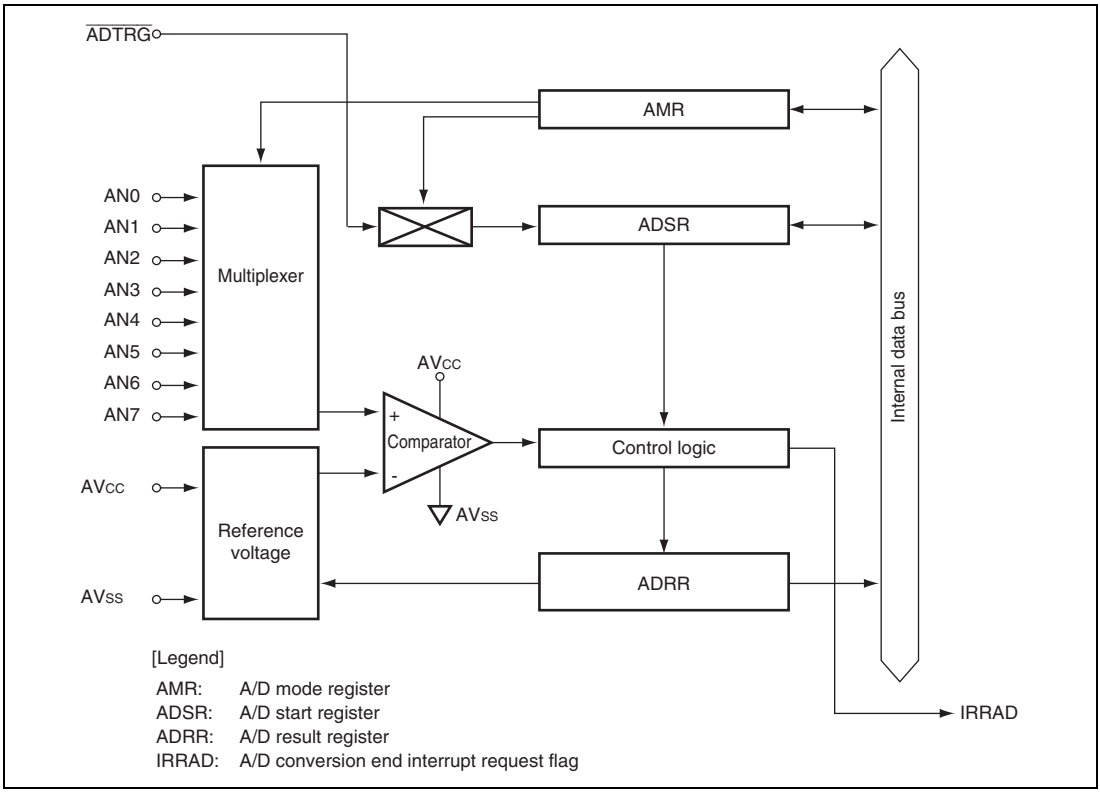


Figure 20.1 Block Diagram of A/D Converter

20.2 Input/Output Pins

Table 20.1 shows the input pins used by the A/D converter.

Table 20.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AVcc	Input	Power supply and reference voltage of analog part
Analog ground pin	AVss	Input	Ground and reference voltage of analog part
Analog input pin 0	AN0	Input	Analog input pins
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
External trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input that controls the A/D conversion start.

20.3 Register Descriptions

The A/D converter has the following registers.

- A/D result register (ADRR)
- A/D mode register (AMR)
- A/D start register (ADSR)

20.3.1 A/D Result Register (ADRR)

ADRR is a 16-bit read-only register that stores the results of A/D conversion. The data is stored in the upper 10 bits of ADRR. ADRR can be read by the CPU at any time, but the ADRR value during A/D conversion is undefined. After A/D conversion is completed, the conversion result is stored as 10-bit data, and this data is retained until the next conversion operation starts. The initial value of ADRR is undefined.

ADRR should be read in word size.

20.3.2 A/D Mode Register (AMR)

AMR sets the A/D conversion time, and selects the external trigger and analog input pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
6	TRGE	0	R/W	External Trigger Select Enables or disables the A/D conversion start by the external trigger input. 0: Disables the A/D conversion start by the external trigger input. 1: Starts A/D conversion at the rising or falling edge of the $\overline{\text{ADTRG}}$ pin. The edge of the $\overline{\text{ADTRG}}$ pin is selected by the ADTRGNEG bit in IEGR.
5	CKS1	0	R/W	Clock Select
4	CKS0	0	R/W	Selects the clock source for A/D conversion. 00: ϕ/8 (conversion time = 124 states (max.) (basic clock = ϕ)) 01: ϕ/4 (conversion time = 62 states (max.) (basic clock = ϕ)) 10: ϕ/2 (conversion time = 31 states (max.) (basic clock = ϕ)) 11: Not selectable (use prohibited)

Bit	Bit Name	Initial Value	R/W	Description
3	CH3	0	R/W	Channel Select 3 to 0
2	CH2	0	R/W	Select the analog input channel.
1	CH1	0	R/W	00xx: No channel selected
0	CH0	0	R/W	0100: AN0
				0101: AN1
				0110: AN2
				0111: AN3
				1000: AN4
				1001: AN5
				1010: AN6
				1011: AN7
				11xx: Use prohibited
				The channel selection should be made while the ADSF bit is cleared to 0.

[Legend]
x: Don't care

20.3.3 A/D Start Register (ADSR)

ADSR starts and stops the A/D conversion.

Bit	Bit Name	Initial Value	R/W	Description
7	ADSF	0	R/W	When this bit is set to 1, A/D conversion is started. When conversion is completed, the converted data is set in ADRR and at the same time this bit is cleared to 0. If this bit is written to 0, A/D conversion can be forcibly terminated.
6	LADS	0	R/W	Resistor Ladder Select 0: Resistor ladder operational while the A/D converter is in the wait state 1: Resistor ladder not operational while the A/D converter is in the wait state Resistor ladder is always halted in standby mode, watch mode, module standby mode, or on reset.
5 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

20.4 Operation

The A/D converter operates by successive approximation with 10-bit resolution. When changing the conversion time or analog input channel, in order to prevent incorrect operation, first clear the bit ADSF to 0 in ADSR.

20.4.1 A/D Conversion

1. A/D conversion is started from the selected channel when the ADSF bit in ADSR is set to 1, according to software.
2. When A/D conversion is completed, the result is transferred to the A/D result register.
3. On completion of conversion, the IRRAD flag in IRR2 is set to 1. If the IENAD bit in IENR2 is set to 1 at this time, an A/D conversion end interrupt request is generated.
4. The ADSF bit remains set to 1 during A/D conversion. When A/D conversion ends, the ADSF bit is automatically cleared to 0 and the A/D converter enters the wait state.

20.4.2 External Trigger Input Timing

The A/D converter can also start A/D conversion by input of an external trigger signal. External trigger input is enabled at the $\overline{\text{ADTRG}}$ pin when the ADTSTCHG bit in PMRB is set to 1* and TRGE bit in AMR is set to 1. Then when the input signal edge designated in the ADTRGNEG bit in IEGR is detected at the $\overline{\text{ADTRG}}$ pin, the ADSF bit in ADSR will be set to 1, starting A/D conversion.

Figure 20.2 shows the timing.

Note: * The $\overline{\text{ADTRG}}$ input pin is shared with the TEST pin. Therefore when the pin is used as the $\overline{\text{ADTRG}}$ pin, reset should be cleared while the 0-fixed or 1-fixed signal is input to the TEST pin. Then the ADTSTCHG bit should be set to 1 after the TEST signal is fixed.

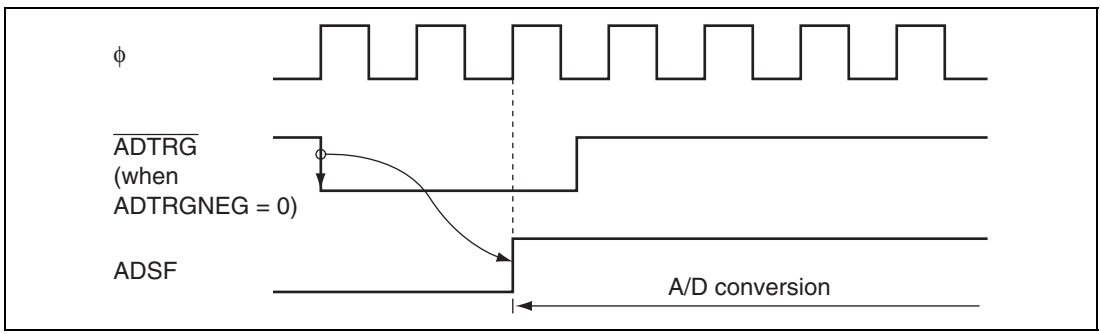


Figure 20.2 External Trigger Input Timing

20.4.3 Operating States of A/D Converter

Table 20.2 shows the operating states of the A/D converter.

Table 20.2 Operating States of A/D Converter

Operating Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby	Module Standby
AMR	Reset	Functioning	Functioning	Retained	Retained	Retained	Retained	Retained
ADSR	Reset	Functioning	Functioning	Retained	Retained	Retained	Retained	Retained
ADRR	Retained*	Functioning	Functioning	Retained	Retained	Retained	Retained	Retained

Note: * Undefined at a power-on reset.

20.5 Example of Use

An example of how the A/D converter can be used is given below, using channel 1 (pin AN1) as the analog input channel. Figure 20.3 shows the operation timing.

1. Bits CH3 to CH0 in the A/D mode register (AMR) are set to 0101, making pin AN1 the analog input channel. A/D interrupts are enabled by setting bit IENAD to 1, and A/D conversion is started by setting bit ADSF to 1.
2. When A/D conversion is completed, bit IRRAD is set to 1, and the A/D conversion result is stored in ADRR. At the same time bit ADSF is cleared to 0, and the A/D converter goes to the idle state.
3. Bit IENAD = 1, so an A/D conversion end interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The A/D conversion result is read and processed.
6. The A/D interrupt handling routine ends.

If bit ADSF is set to 1 again afterward, A/D conversion starts and steps 2 through 6 take place.

Figures 20.4 and 20.5 show flowcharts of procedures for using the A/D converter.

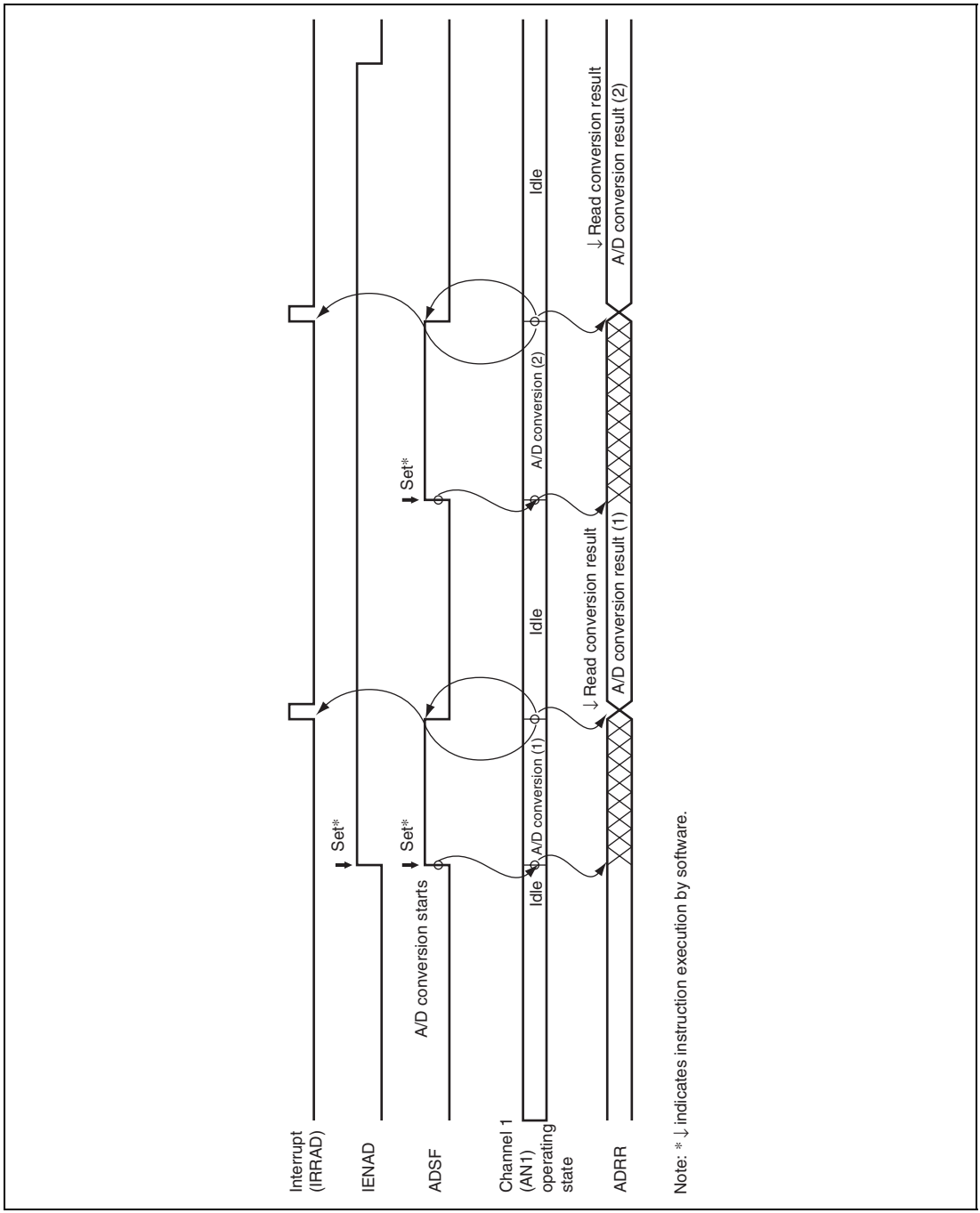


Figure 20.3 Example of A/D Conversion Operation

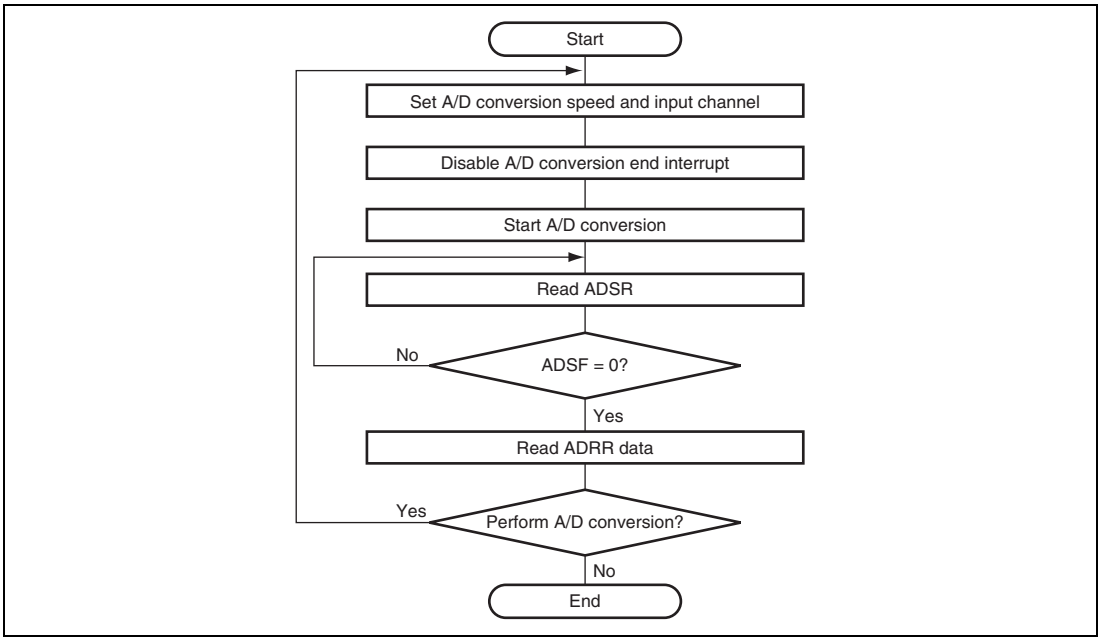


Figure 20.4 Flowchart of Procedure for Using A/D Converter (Polling by Software)

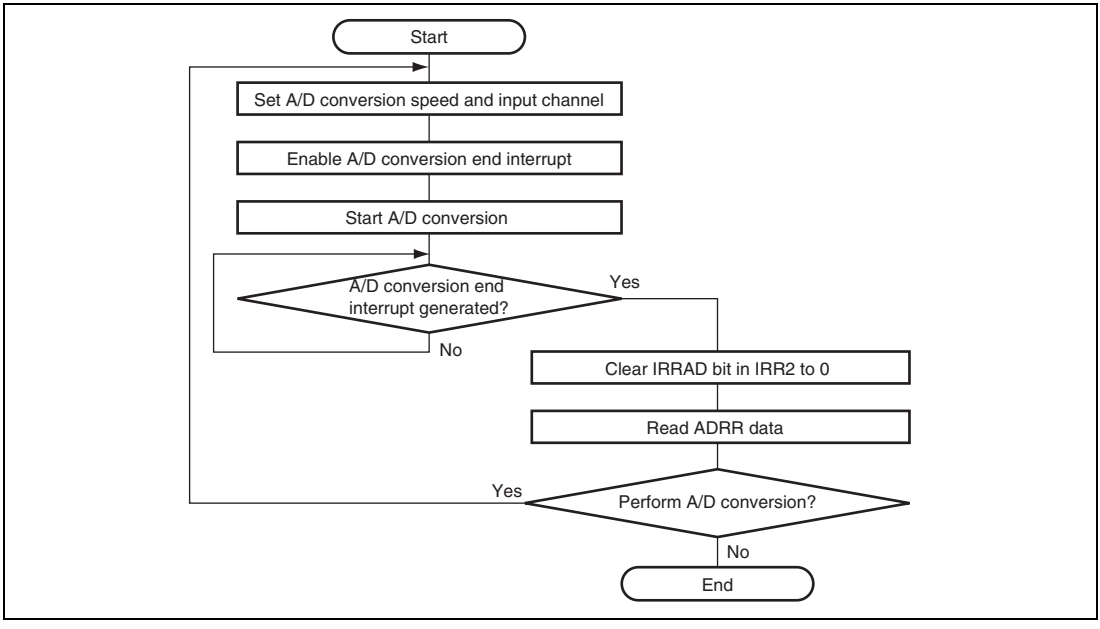


Figure 20.5 Flowchart of Procedure for Using A/D Converter (Interrupts Used)

20.6 A/D Conversion Accuracy Definitions

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital output codes
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 20.6).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value 0000000000 to 0000000001 (see figure 20.7).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from 1111111110 to 1111111111 (see figure 20.7).
- Nonlinearity error

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

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

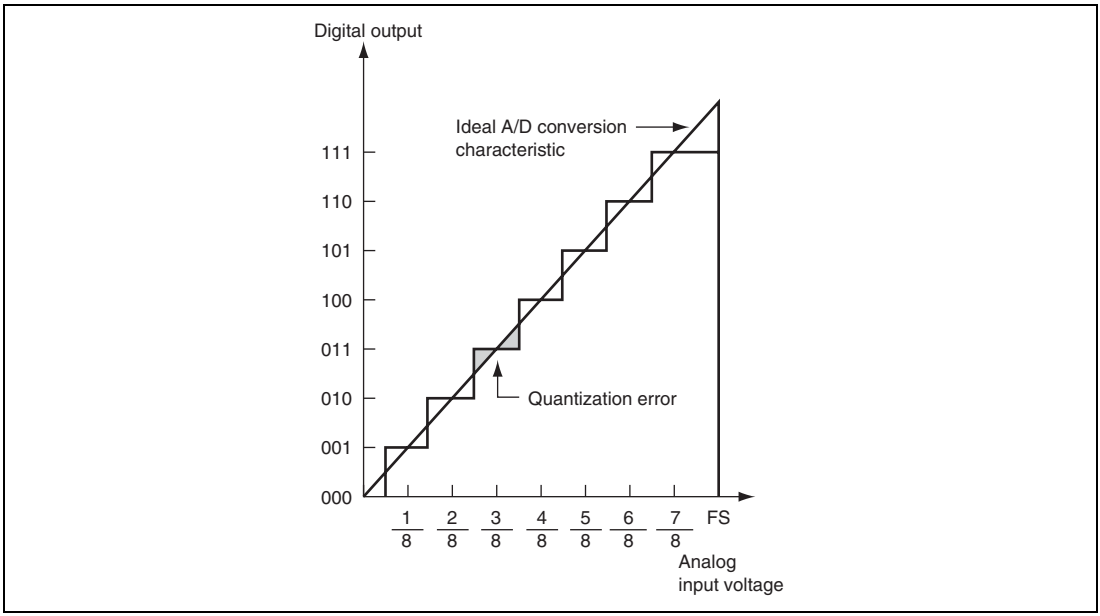


Figure 20.6 A/D Conversion Accuracy Definitions (1)

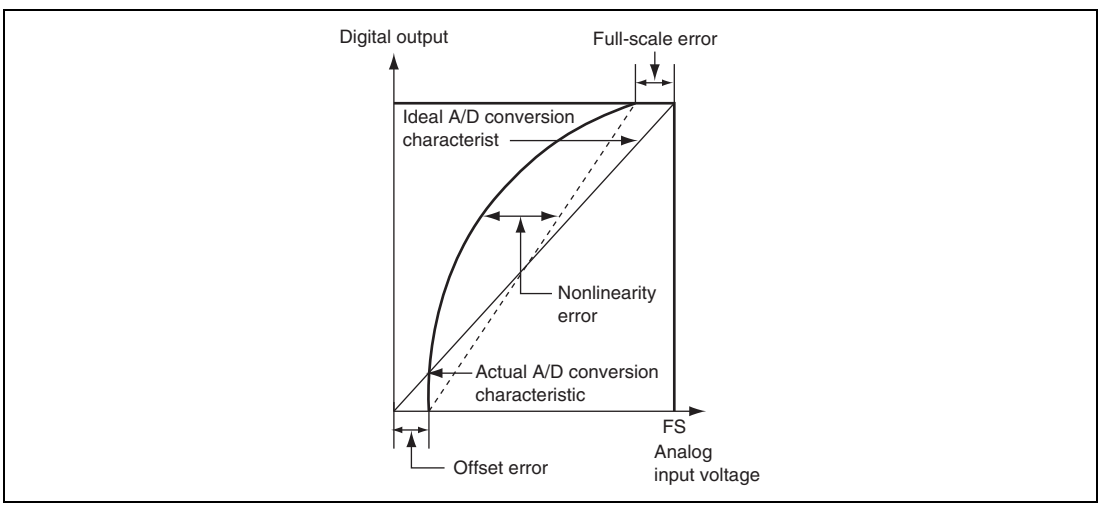


Figure 20.7 A/D Conversion Accuracy Definitions (2)

20.7 Usage Notes

20.7.1 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion accuracy is guaranteed for an input signal for which the signal source impedance is 10 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 10 kΩ, charging may be insufficient and it may not be possible to guarantee A/D conversion accuracy.

When a large capacitance is provided externally to prevent this, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored. However, as a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/μs or greater) (see figure 20.8).

When converting a high-speed analog signal, a low-impedance buffer should be inserted.

20.7.2 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely 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 act as antennas on the mounting board.

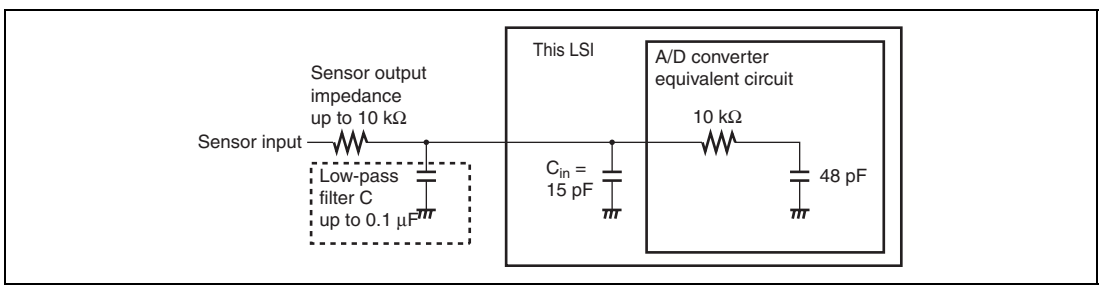


Figure 20.8 Example of Analog Input Circuit

20.7.3 Other Usage Notes

1. ADRR should be read only when the ADSF bit in ADSR is cleared to 0.
2. Changing the digital input signal at an adjacent pin during A/D conversion may adversely affect conversion accuracy.
3. When A/D conversion is started after clearing module standby mode, wait for 10ϕ clock cycles before starting A/D conversion.

Section 21 I²C Bus Interface 2 (IIC2)

The I²C bus interface 2 conforms to and provides a subset of the Philips I²C bus (inter-IC bus) interface functions. The register configuration that controls the I²C bus differs partly from the Philips configuration, however. Figure 21.1 shows a block diagram of the I²C bus interface 2. Figure 21.2 shows an example of I/O pin connections to external circuits.

21.1 Features

- Selection of I²C format or clock synchronous serial format
- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independent from each other, the continuous transmission/reception can be performed.
- Use of module standby mode enables this module to be placed in standby mode independently when not used (for details, refer to section 6.4, Module Standby Function).

I²C bus format

- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization/wait function

  In master mode, the state of SCL is monitored per bit, and the timing is synchronized automatically.

  If transmission/reception is not yet possible, set the SCL to low until preparations are completed.
- Six interrupt sources

  Transmit data empty (including slave-address match), transmit end, receive data full (including slave-address match), arbitration lost, NACK detection, and stop condition detection
- Direct bus drive

  Two pins, SCL and SDA pins, function as CMOS outputs in normal operation (when the port/serial function is selected) and NMOS outputs when the bus drive function is selected.

Clock synchronous format

- Four interrupt sources

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

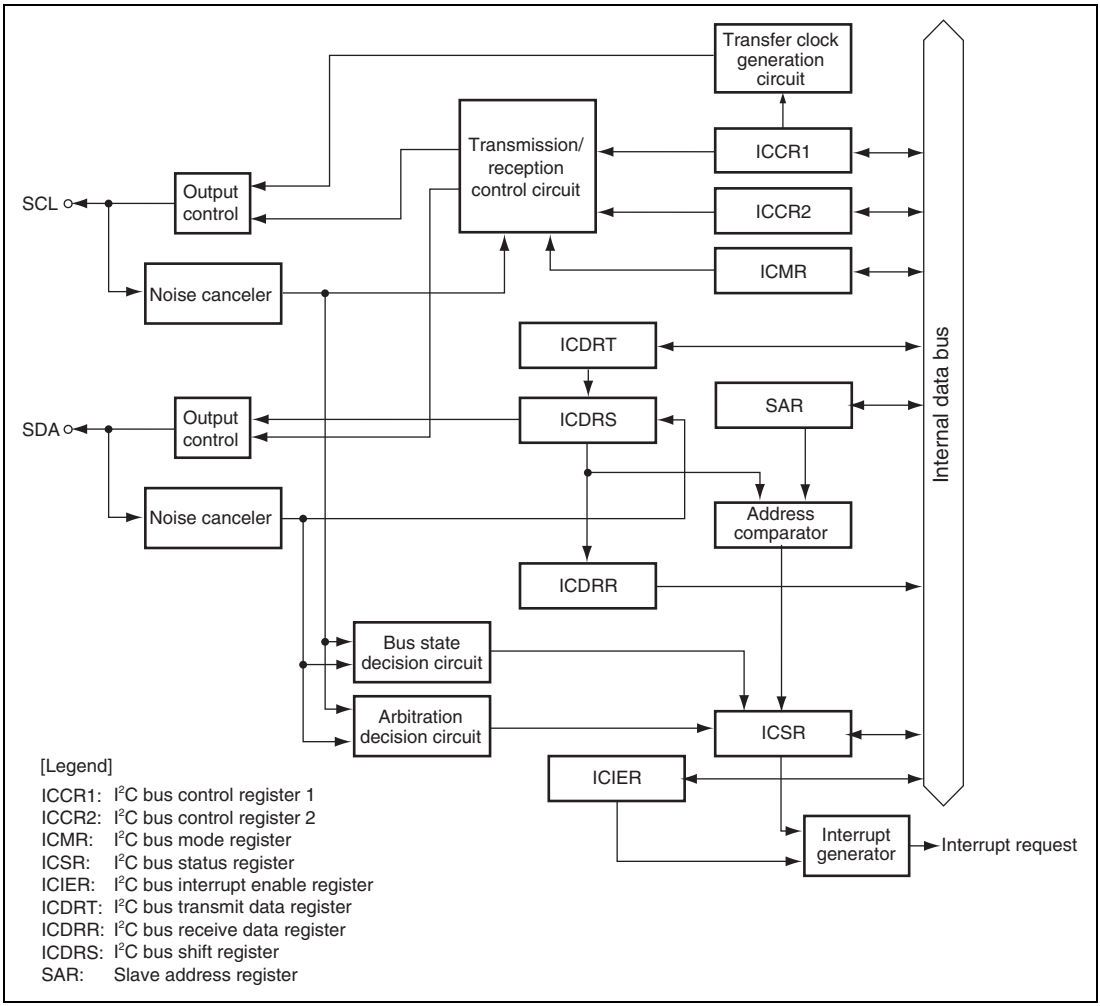


Figure 21.1 Block Diagram of I²C Bus Interface 2

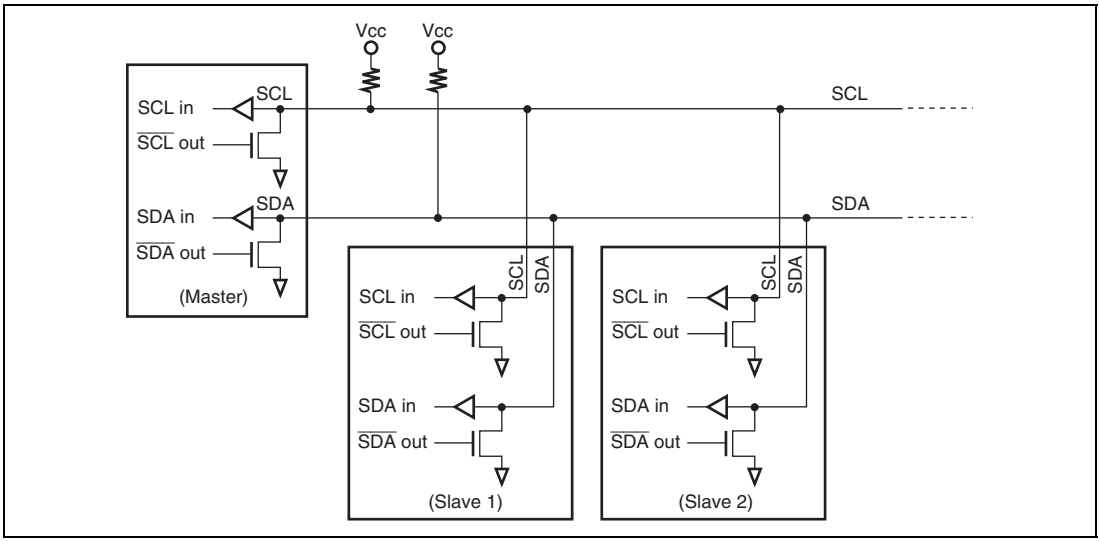


Figure 21.2 External Circuit Connections of I/O Pins

21.2 Input/Output Pins

Table 21.1 shows the input/output pins of the I²C bus interface 2.

Table 21.1 Pin Configuration

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

21.3 Register Descriptions

The I²C bus interface 2 has the following registers.

- I²C bus control register 1 (ICCR1)
- I²C bus control register 2 (ICCR2)
- I²C bus mode register (ICMR)
- I²C bus interrupt enable register (ICIER)
- I²C bus status register (ICSR)
- Slave address register (SAR)
- I²C bus transmit data register (ICDRT)
- I²C bus receive data register (ICDRR)
- I²C bus shift register (ICDRS)

21.3.1 I²C Bus Control Register 1 (ICCR1)

ICCR1 enables or disables the I²C bus interface 2, controls transmission or reception, and selects master or slave mode, transmission or reception, and transfer clock frequency in master mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface 2 Enable 0: This module is halted. (SCL and SDA pins are set to the port/serial function.) 1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable This bit enables or disables the next operation when TRS is 0 and ICDRR is read. 0: Enables next reception 1: Disables next reception

Bit	Bit Name	Initial Value	R/W	Description
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select In master mode with the I²C bus format, when arbitration is lost, MST and TRS are both reset by hardware, causing a transition to slave receive mode. Modification of the TRS bit should be made between transfer frames. After data reception has been started in slave receive mode, when the first seven bits of the receive data agree with the slave address that is set to SAR and the eighth bit is 1, TRS is automatically set to 1. If an overrun error occurs in master mode with the clock synchronous serial format, MST is cleared to 0 and slave receive mode is entered. Operating modes are described below according to MST and TRS combination. When clock synchronous serial format is selected and MST is 1, clock is output. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode
3	CKS3	0	R/W	Transfer Clock Select 3 to 0 In master mode, set these bits according to the necessary transfer rate (see table 21.2, Transfer Rate). In slave mode, these bits are used to secure the data setup time in transmission mode. When CKS3 = 0, the data setup time is 10 t_{cyc} and when CKS3 = 1, the data setup time is 20 t_{cyc}.
2	CKS2	0	R/W	
1	CKS1	0	R/W	
0	CKS0	0	R/W	

Table 21.2 Transfer Rate

Bit 3	Bit 2	Bit 1	Bit 0		Transfer Rate		
CKS3	CKS2	CKS1	CKS0	Clock	φ = 2 MHz	φ = 5 MHz	φ = 10 MHz
0	0	0	0	φ/28	71.4 kHz	179 kHz	357 kHz
			1	φ/40	50.0 kHz	125 kHz	250 kHz
		1	0	φ/48	41.7 kHz	104 kHz	208 kHz
			1	φ/64	31.3 kHz	78.1 kHz	156 kHz
	1	0	0	φ/80	25.0 kHz	62.5 kHz	125 kHz
			1	φ/100	20.0 kHz	50.0 kHz	100 kHz
		1	0	φ/112	17.9 kHz	44.6 kHz	89.3 kHz
			1	φ/128	15.6 kHz	39.1 kHz	78.1 kHz
1	0	0	0	φ/56	35.7 kHz	89.3 kHz	179 kHz
			1	φ/80	25.0 kHz	62.5 kHz	125 kHz
		1	0	φ/96	20.8 kHz	52.1 kHz	104 kHz
			1	φ/128	15.6 kHz	39.1 kHz	78.1 kHz
	1	0	0	φ/160	12.5 kHz	31.3 kHz	62.5 kHz
			1	φ/200	10.0 kHz	25.0 kHz	50.0 kHz
		1	0	φ/224	8.9 kHz	22.3 kHz	44.6 kHz
			1	φ/256	7.8 kHz	19.5 kHz	39.1 kHz

21.3.2 I²C Bus Control Register 2 (ICCR2)

ICCR1 issues start/stop conditions, handles the SDA pin, monitors the SCL pin, and controls reset in the control part of the I²C bus interface 2.

Bit	Bit Name	Initial Value	R/W	Description
7	BBSY	0	R/W	Bus Busy This bit enables to confirm whether the I²C bus is occupied or released and to issue start/stop conditions in master mode. This bit has no functional role when the clock synchronous serial format is selected. When the I²C bus format is selected, this bit is set to 1 when the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. The same procedure also applies to re-transmitting a start condition. Write 0 in BBSY and 0 in SCP to issue a stop condition. To issue start/stop conditions, use the MOV instruction.
6	SCP	1	R/W	Start/Stop Issue Condition Disable The SCP bit controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. If 1 is written, the data is not stored.
5	SDAO	1	R/W	SDA Output Value Control This bit is used with SDAOP when modifying output level of SDA. This bit should not be manipulated during transfer. 0: When reading, SDA pin outputs low. When writing, SDA pin is changed to output low. 1: When reading, SDA pin outputs high. When writing, SDA pin is changed to output Hi-Z (outputs high by external pull-up resistance).

Bit	Bit Name	Initial Value	R/W	Description
4	SDAOP	1	R/W	SDAO Write Protect This bit controls change of output level of the SDA pin by modifying the SDAO bit. To change the output level, clear SDAO and SDAOP to 0 or set SDAO to 1 and clear SDAOP to 0 by the MOV instruction. This bit is always read as 1.
3	SCLO	1	R	This bit monitors SCL output level. When SCLO is 1, SCL pin outputs high. When SCLO is 0, SCL pin outputs low.
2	—	1	—	Reserved This bit is always read as 1.
1	IICRST	0	R/W	IIC Control Part Reset This bit resets the control part except for I²C registers. If this bit is set to 1 when hang-up occurs because of communication failure during I²C operation, I²C control part can be reset without setting ports and initializing registers.
0	—	1	—	Reserved This bit is always read as 1.

21.3.3 I²C Bus Mode Register (ICMR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I²C bus format is used.
6	WAIT	0	R/W	Wait Insertion Bit In master mode with the I²C bus format, this bit selects whether to insert a wait after data transfer except the acknowledge bit. When WAIT is set to 1, after the fall of the clock for the final data bit, low period is extended for two transfer clocks. If WAIT is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted. The setting of this bit is invalid in slave mode with the I²C bus format or with the clock synchronous serial format.
5, 4	—	All 1	—	Reserved These bits are always read as 1.
3	BCWP	1	R/W	BC Write Protect This bit controls the BC2 to BC0 modifications. When modifying BC2 to BC0, this bit should be cleared to 0 and use the MOV instruction. In clock synchronous serial mode, BC should not be modified. 0: When writing, values of BC2 to BC0 are set. 1: When reading, 1 is always read. When writing, settings of BC2 to BC0 are invalid.

Bit	Bit Name	Initial Value	R/W	Description
2	BC2	0	R/W	Bit Counter 2 to 0
1	BC1	0	R/W	These bits specify the number of bits to be transferred next. When read, the remaining number of transfer bits is indicated. With the I²C bus format, the data is transferred with one addition acknowledge bit. Bit BC2 to BC0 settings should be made during an interval between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL pin is low. The value returns to 000 at the end of a data transfer, including the acknowledge bit. With the clock synchronous serial format, these bits should not be modified.
0	BC0	0	R/W	

I²C Bus Format	Clock Synchronous Serial Format
000: 9 bits	000: 8 bits
001: 2 bits	001: 1 bits
010: 3 bits	010: 2 bits
011: 4 bits	011: 3 bits
100: 5 bits	100: 4 bits
101: 6 bits	101: 5 bits
110: 7 bits	110: 6 bits
111: 8 bits	111: 7 bits

21.3.4 I²C Bus Interrupt Enable Register (ICIER)

ICIER enables or disables interrupt sources and acknowledge bits, sets acknowledge bits to be transferred, and confirms acknowledge bits to be received.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, this bit enables or disables the transmit data empty interrupt (TXI). 0: Transmit data empty interrupt request (TXI) is disabled. 1: Transmit data empty interrupt request (TXI) is enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable This bit enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by clearing the TEND bit or the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabled. 1: Transmit end interrupt request (TEI) is enabled.
5	RIE	0	R/W	Receive Interrupt Enable This bit enables or disables the receive data full interrupt request (RXI) and the overrun error interrupt request (ERI) with the clock synchronous format, when a receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) and overrun error interrupt request (ERI) with the clock synchronous format are disabled. 1: Receive data full interrupt request (RXI) and overrun error interrupt request (ERI) with the clock synchronous format are enabled.

Bit	Bit Name	Initial Value	R/W	Description
4	NAKIE	0	R/W	NACK Receive Interrupt Enable This bit enables or disables the NACK receive interrupt request (NAKI) and the overrun error (setting of the OVE bit in ICSR) interrupt request (ERI) with the clock synchronous format, when the NACKF and AL bits in ICSR are set to 1. NAKI can be canceled by clearing the NACKF, OVE, or NAKIE bit to 0. 0: NACK receive interrupt request (NAKI) is disabled. 1: NACK receive interrupt request (NAKI) is enabled.
3	STIE	0	R/W	Stop Condition Detection Interrupt Enable 0: Stop condition detection interrupt request (STPI) is disabled. 1: Stop condition detection interrupt request (STPI) is enabled.
2	ACKE	0	R/W	Acknowledge Bit Judgment Select 0: The value of the receive acknowledge bit is ignored, and continuous transfer is performed. 1: If the receive acknowledge bit is 1, continuous transfer is halted.
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowledge data that are returned by the receive device. This bit cannot be modified. 0: Receive acknowledge = 0 1: Receive acknowledge = 1
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, this bit specifies the bit to be sent at the acknowledge timing. 0: 0 is sent at the acknowledge timing. 1: 1 is sent at the acknowledge timing.

21.3.5 I²C Bus Status Register (ICSR)

ICSR performs confirmation of interrupt request flags and status.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	0	R/(W)*	Transmit Data Register Empty [Setting conditions] • Data is transferred from ICDRT to ICDRS and ICDRT becomes empty • TRS is set • A start condition (including re-transfer) has been issued • Transmit mode is entered from receive mode in slave mode [Clearing conditions] • Writing of 0 to bit TDRE after reading TDRE = 1 • Data is written to ICDRT with an instruction
6	TEND	0	R/(W)*	Transmit End [Setting conditions] • The ninth clock of SCL rises with the I²C bus format while the TDRE flag is 1 • The final bit of transmit frame is sent with the clock synchronous serial format [Clearing conditions] • Writing of 0 to bit TEND after reading TEND = 1 • Data is written to ICDRT with an instruction
5	RDRF	0	R/(W)*	Receive Data Register Full [Setting condition] • A receive data is transferred from ICDRS to ICDRR [Clearing conditions] • Writing of 0 to bit RDRF after reading RDRF = 1 • ICDRR is read with an instruction

Bit	Bit Name	Initial Value	R/W	Description
4	NACKF	0	R/(W)*	No Acknowledge Detection Flag [Setting condition] • No acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1 [Clearing condition] • Writing of 0 to bit NACKF after reading NACKF = 1
3	STOP	0	R/(W)*	Stop Condition Detection Flag [Setting conditions] • In master mode, a stop condition is detected after frame transferred • In slave mode, the slave address in the first byte after the general call and detecting start condition matches the address set in SAR, and then the stop condition is detected [Clearing condition] • Writing of 0 to bit STOP after reading STOP = 1
2	AL/OVE	0	R/(W)*	Arbitration Lost Flag/Overrun Error Flag This flag indicates that arbitration was lost in master mode with the I²C bus format and that the final bit has been received while RDRF = 1 with the clock synchronous format. When two or more master devices attempt to seize the bus at nearly the same time, if the I²C bus interface detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been taken by another master. [Setting conditions] • The internal SDA and SDA pin disagree at the rise of SCL in master transmit mode • The SDA pin outputs high in master mode while a start condition is detected • The final bit is received with the clock synchronous format while RDRF = 1 [Clearing condition] • Writing of 0 to bit AL/OVE after reading AL/OVE=1

Bit	Bit Name	Initial Value	R/W	Description
1	AAS	0	R/(W)*	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR. [Setting conditions] • The slave address is detected in slave receive mode • The general call address is detected in slave receive mode [Clearing condition] • Writing of 0 to bit AAS after reading AAS=1
0	ADZ	0	R/(W)*	General Call Address Recognition Flag This bit is valid in I²C bus format slave receive mode. [Setting condition] • The general call address is detected in slave receive mode [Clearing condition] • Writing of 0 to bit ADZ after reading ADZ=1

Note: * Only 0 can be written to clear the flag.

21.3.6 Slave Address Register (SAR)

SAR selects the communication format and sets the slave address. When the chip is in slave mode with the I²C bus format, if the upper 7 bits of SAR match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA6 to SVA0	All 0	R/W	Slave Address 6 to 0 These bits set a unique address in bits SVA6 to SVA0, differing form the addresses of other slave devices connected to the I²C bus.
0	FS	0	R/W	Format Select 0: I²C bus format is selected. 1: Clock synchronous serial format is selected.

21.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. If the next transfer data is written to ICDRT during transferring data of ICDRS, continuous transfer is possible. If the MLS bit of ICMR is set to 1 and when the data is written to ICDRT, the MSB/LSB inverted data is read. The initial value of ICDRT is H'FF.

21.3.8 I²C Bus Receive Data Register (ICDRR)

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

21.3.9 I²C Bus Shift Register (ICDRS)

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

21.4 Operation

The I²C bus interface 2 can communicate either in I²C bus mode or clock synchronous serial mode by setting the FS bit in the slave address register (SAR).

21.4.1 I²C Bus Format

Figure 21.3 shows the I²C bus formats. Figure 21.4 shows the I²C bus timing. The first frame following a start condition always consists of 8 bits.

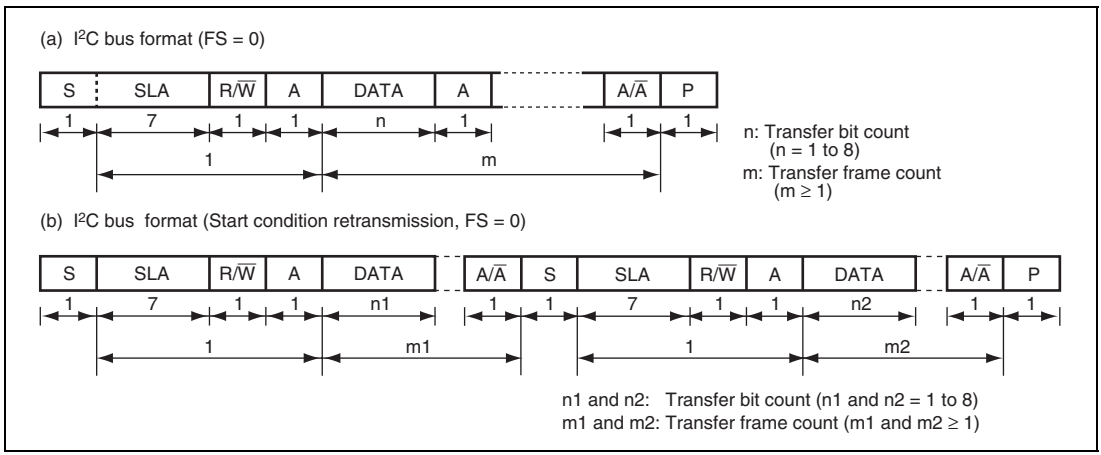


Figure 21.3 I²C Bus Formats

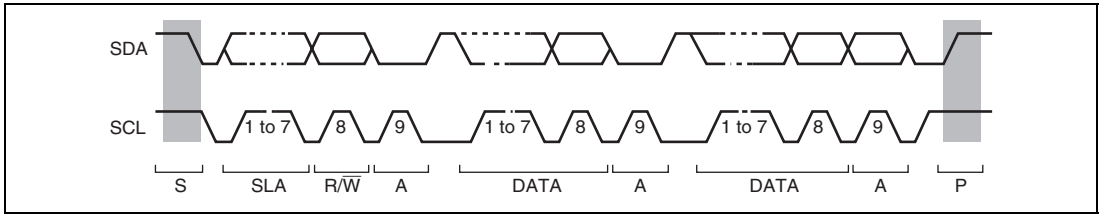


Figure 21.4 I²C Bus Timing

[Legend**]**

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

SLA: Slave address

R/$\overline{W}$: Indicates the direction of data transfer: from the slave device to the master device when R/$\overline{W}$ is 1, or from the master device to the slave device when R/$\overline{W}$ is 0.

A: Acknowledge. The receive device drives SDA to low.

DATA: Transfer data

P: Stop condition. The master device drives SDA from low to high while SCL is high.

21.4.2 Master Transmit Operation

In master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For master transmit mode operation timing, refer to figures 21.5 and 21.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is free. Set the MST and TRS bits in ICCR1 to select master transmit mode. Then, write 1 to BBSY and 0 to SCP using MOV instruction. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first byte data show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cleared to 0, and data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR is set to 1 at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confirm that the slave device has been selected. Then, write second byte data to ICDRT. When ACKBR is 1, the slave device has not been acknowledged, so issue the stop condition. To issue the stop condition, write 0 to BBSY and SCP using MOV instruction. SCL is fixed low until the transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the end of last byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) from the receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TEND or NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mode.

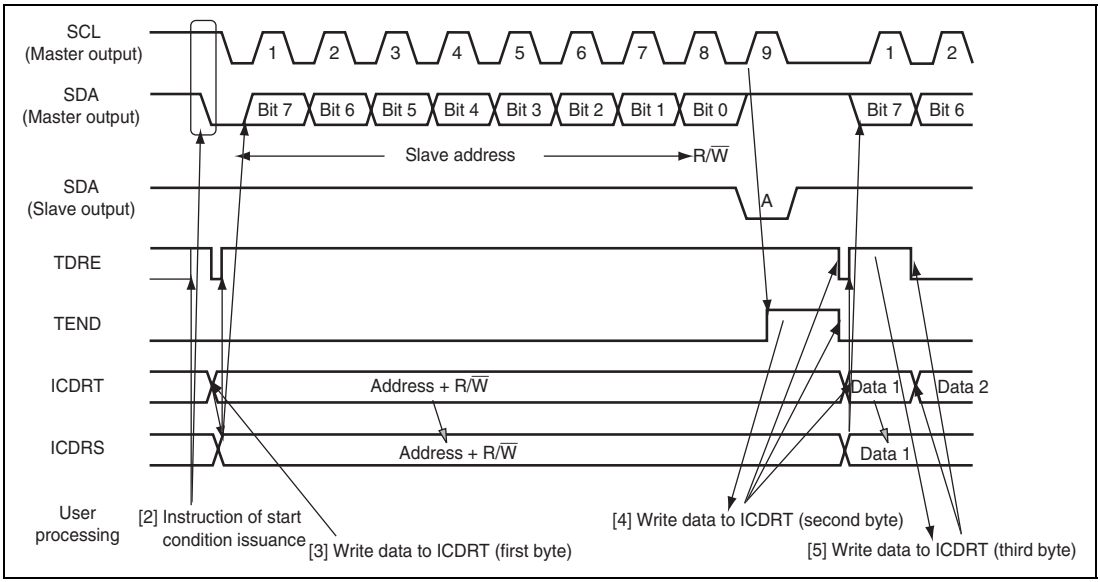


Figure 21.5 Master Transmit Mode Operation Timing (1)

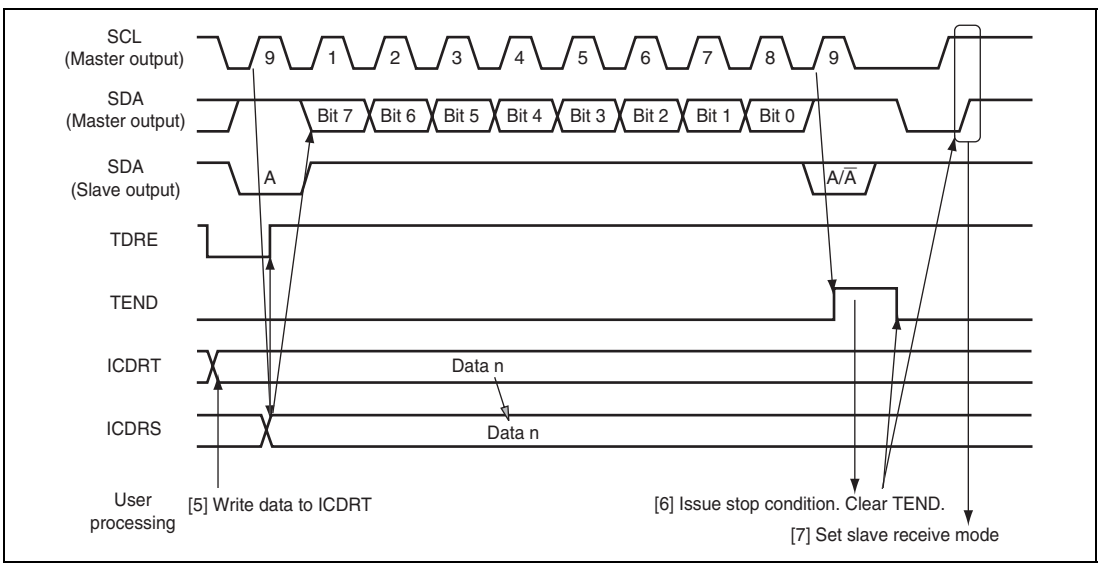


Figure 21.6 Master Transmit Mode Operation Timing (2)

21.4.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data from the slave device, and returns an acknowledge signal. For master receive mode operation timing, refer to figures 21.7 and 21.8. The reception procedure and operations in master receive mode are shown below.

1. Clear the TEND bit in ICSR to 0, then clear the TRS bit in ICCR1 to 0 to switch from master transmit mode to master receive mode. Then, clear the TDRE bit to 0 and set the ACKBT bit in ICIER.
2. When ICDRR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. The master device outputs the level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.
3. After the reception of first frame data is completed, the RDRF bit in ICST is set to 1 at the rise of 9th receive clock pulse. At this time, the receive data is read by reading ICDRR, and RDRF is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. If 8th receive clock pulse falls after reading ICDRR by the other processing while RDRF is 1, SCL is fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 and set the ACKBT bit in ICIER. to 1 before reading ICDRR. This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, and clearing the STOP bit in ICSR issue the stage condition.
7. When the STOP bit in ICSR is set to 1, read ICDRR. Then clear the RCVD bit to 0.
8. Clear the MST bit in ICCR1 and then, the operation returns to the slave receive mode.

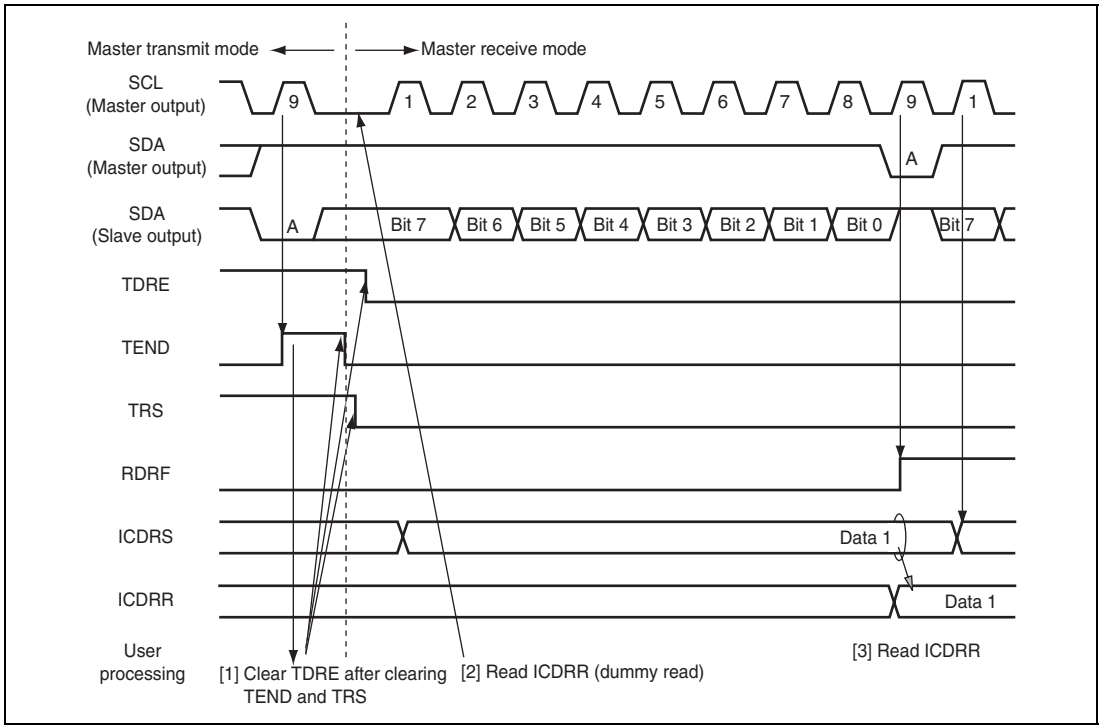


Figure 21.7 Master Receive Mode Operation Timing (1)

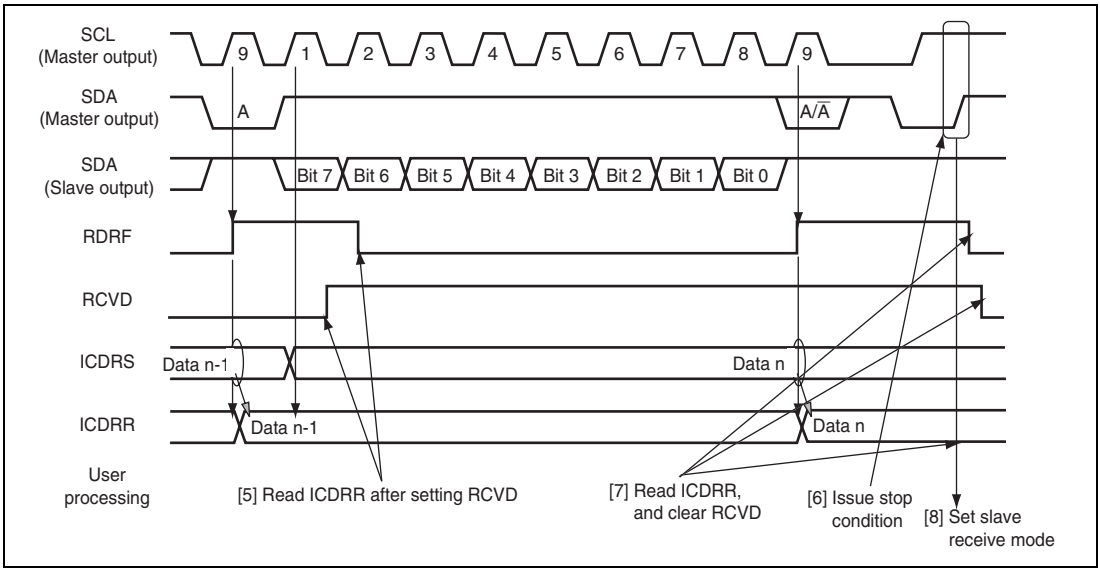


Figure 21.8 Master Receive Mode Operation Timing (2)

21.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. For slave transmit mode operation timing, refer to figures 21.9 and 21.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by the ACKBT bit in ICIER to SDA, at the rise of the 9th clock pulse. At this time, if the 8th bit data (R/$\overline{W}$) is 1, the TRS and ICSR bits in ICCR1 are set to 1, and the mode changes to slave transmit mode automatically. The continuous transmission is performed by writing transmit data to ICDRT every time TDRE is set.
3. If TDRE is set after writing last transmit data to ICDRT, wait until TEND in ICSR is set to 1, with TDRE = 1. When the TEND bit is set, clear the TEND bit.
4. Clear the TRS bit for the end processing, and read ICDRR (dummy read). SCL is free.
5. Clear the TDRE bit.

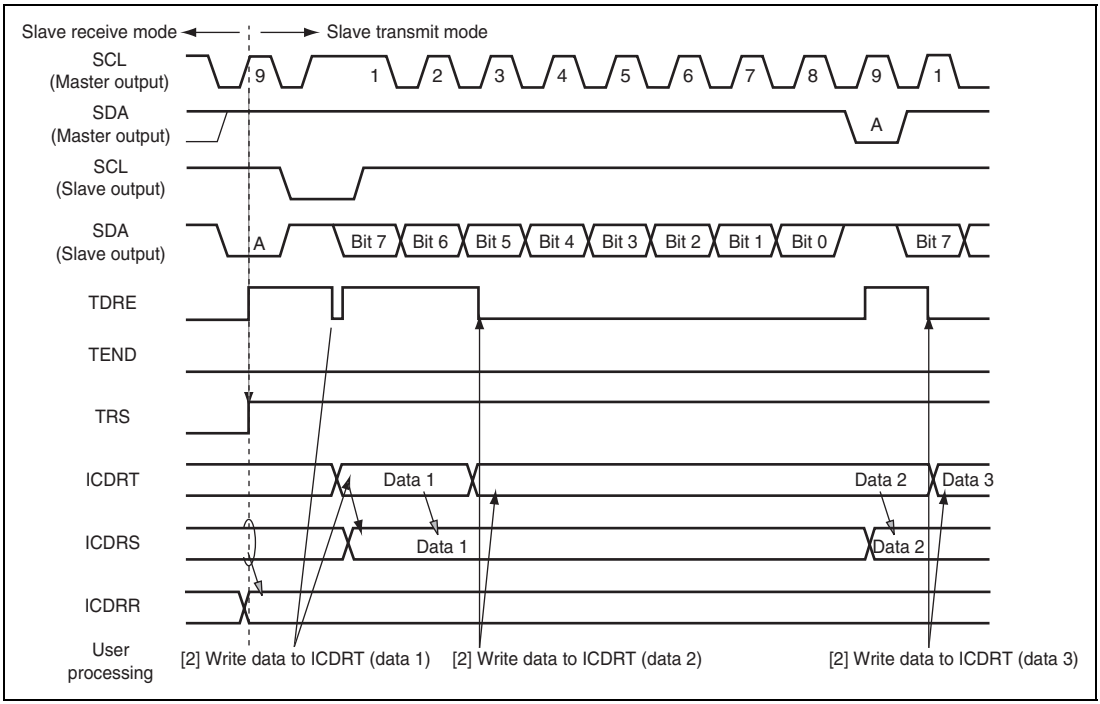


Figure 21.9 Slave Transmit Mode Operation Timing (1)

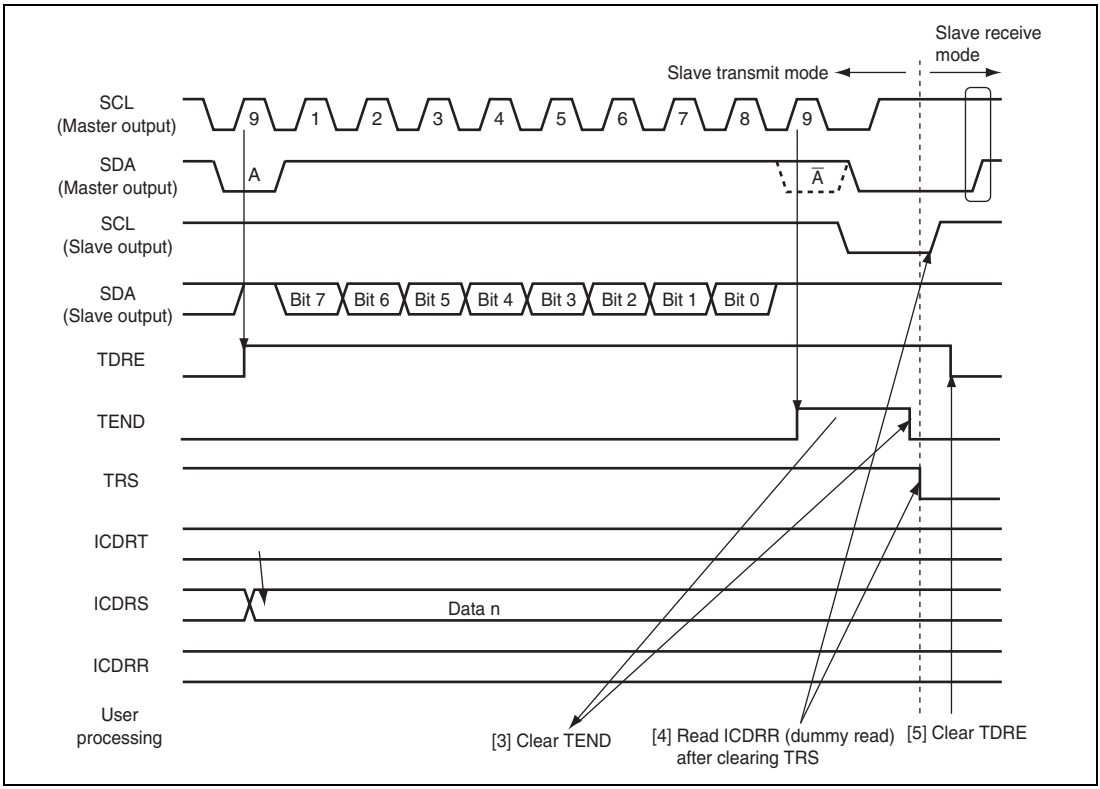


Figure 21.10 Slave Transmit Mode Operation Timing (2)

21.4.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For slave receive mode operation timing, refer to figures 21.11 and 21.12. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (Since the read data show the slave address and R/$\overline{W}$, it is not used.)
3. Read ICDRR every time RDRF is set. If 8th receive clock pulse falls while RDRF is 1, SCL is fixed low until ICDRR is read. The change of the acknowledge before reading ICDRR, to be returned to the master device, is reflected to the next transmit frame.
4. The last byte data is read by reading ICDRR.

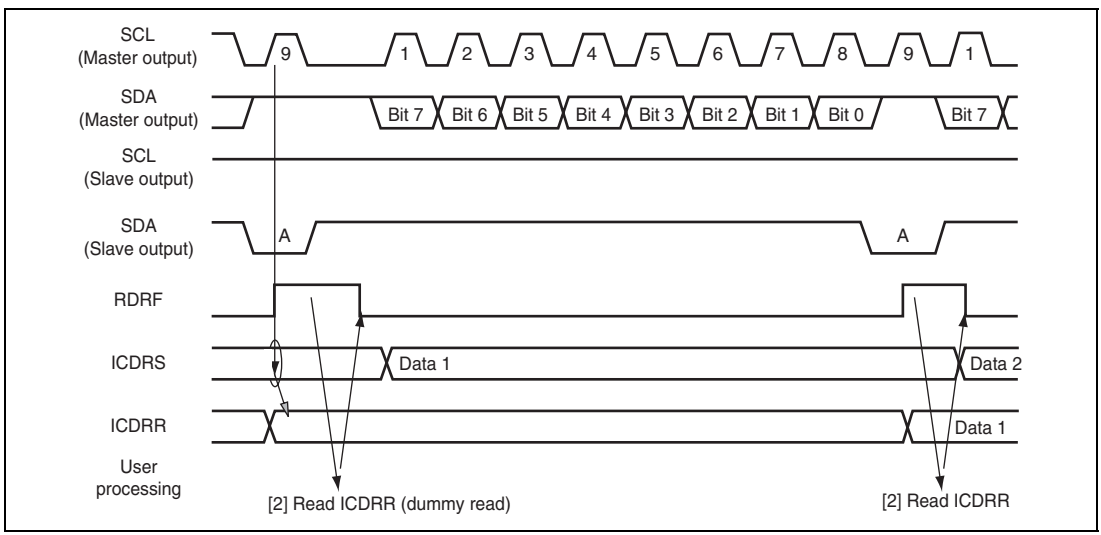


Figure 21.11 Slave Receive Mode Operation Timing (1)

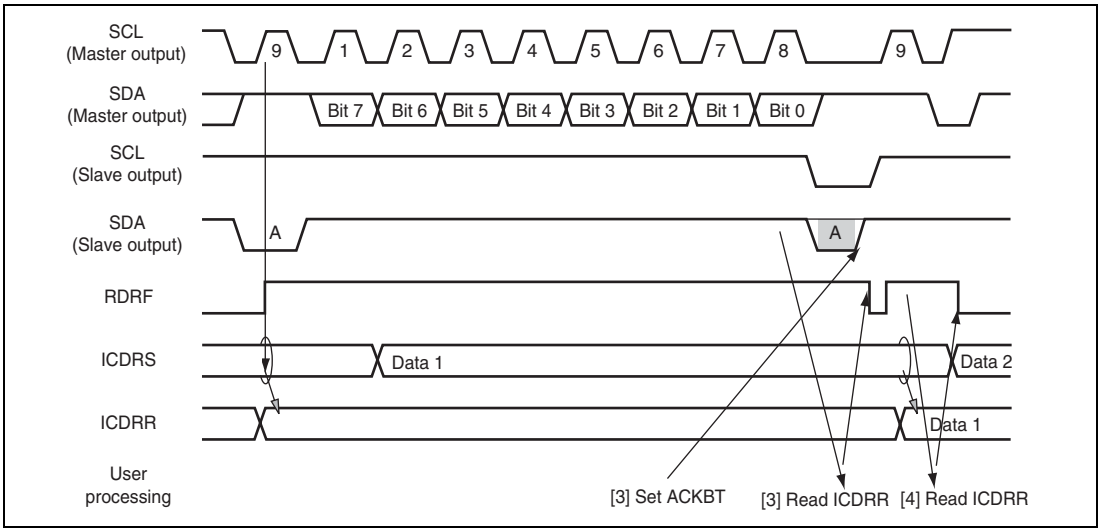


Figure 21.12 Slave Receive Mode Operation Timing (2)

21.4.6 Clock Synchronous Serial Format

This module can be operated with the clock synchronous serial format, by setting the FS bit in SAR to 1. When the MST bit in ICCR1 is 1, the transfer clock output from SCL is selected. When MST is 0, the external clock input is selected.

(1) Data Transfer Format

Figure 21.13 shows the clock synchronous serial transfer format.

The transfer data is output from the rise to the fall of the SCL clock, and the data at the rising edge of the SCL clock is guaranteed. The MLS bit in ICMR sets the order of data transfer, in either the MSB first or LSB first. The output level of SDA can be changed during the transfer wait, by the SDAO bit in ICCR2.

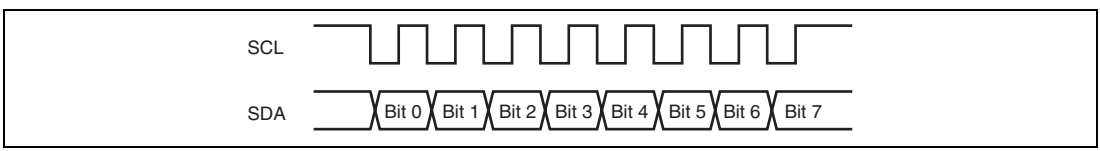


Figure 21.13 Clock Synchronous Serial Transfer Format

(2) Transmit Operation

In transmit mode, transmit data is output from SDA, in synchronization with the fall of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For transmit mode operation timing, refer to figure 21.14. The transmission procedure and operations in transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. Set the TRS bit in ICCR1 to select the transmit mode. Then, TDRE in ICSR is set.
3. Confirm that TDRE has been set. Then, write the transmit data to ICDRT. The data is transferred from ICDRT to ICDRS, and TDRE is set automatically. The continuous transmission is performed by writing data to ICDRT every time TDRE is set. When changing from transmit mode to receive mode, clear the TRS bit while the TDRE bit is 1.

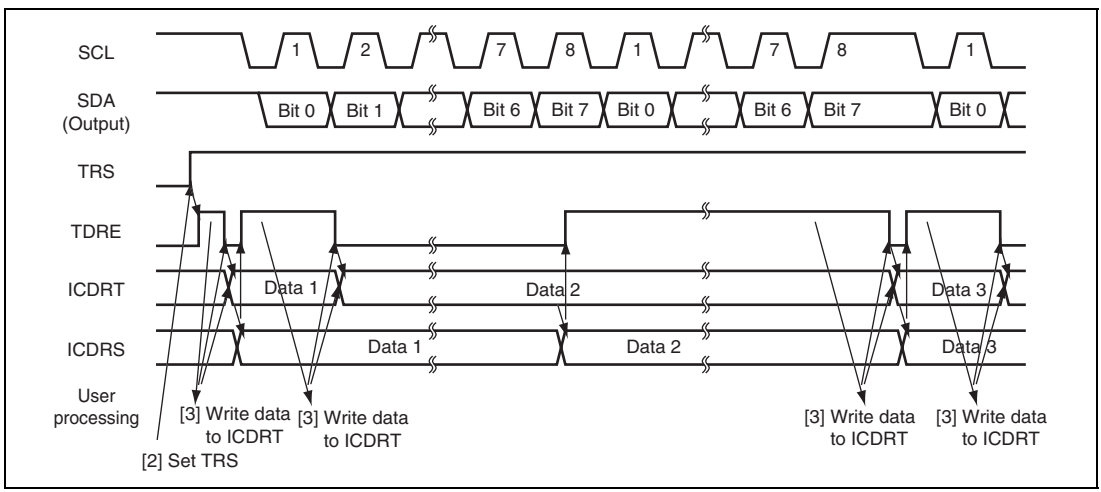


Figure 21.14 Transmit Mode Operation Timing

(3) Receive Operation

In receive mode, data is latched at the rise of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For receive mode operation timing, refer to figure 21.15. The reception procedure and operations in receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. When the transfer clock is output, set MST to 1 to start outputting the receive clock.
3. When the receive operation is completed, data is transferred from ICDRS to ICDRR and RDRF in ICSR is set. When MST = 1, the next byte can be received, so the clock is continually output. The continuous reception is performed by reading ICDRR every time RDRF is set. When the 8th clock is risen while RDRF is 1, the overrun is detected and AL/OVE in ICSR is set. At this time, the previous reception data is retained in ICDRR.
4. To stop receiving when MST = 1, set RCVD in ICCR1 to 1, then read ICDRR. Then, SCL is fixed high after receiving the next byte data.

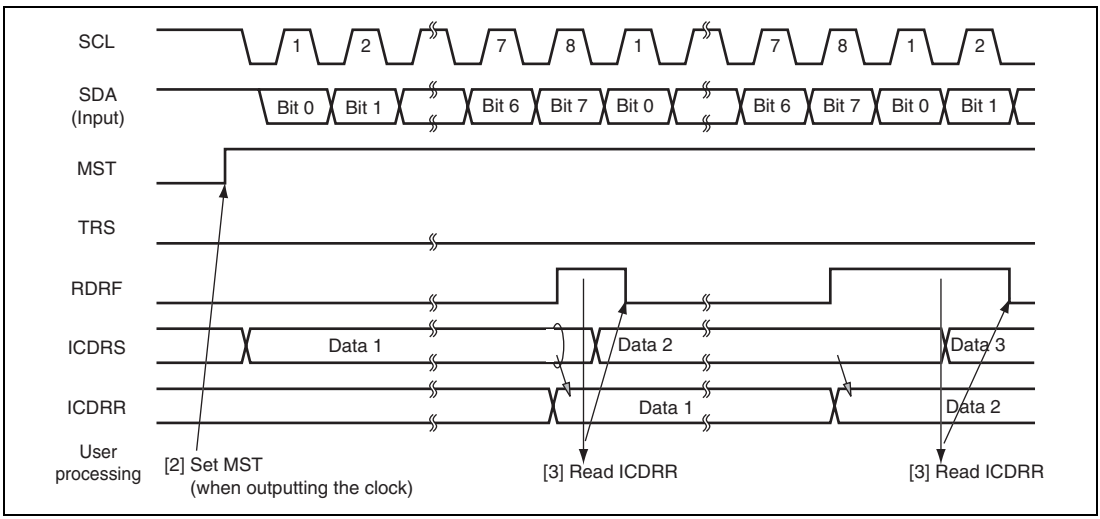


Figure 21.15 Receive Mode Operation Timing

21.4.7 Noise Canceller

The logic levels at the SCL and SDA pins are routed through noise cancellers before being latched internally. Figure 21.16 shows a block diagram of the noise canceller circuit.

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

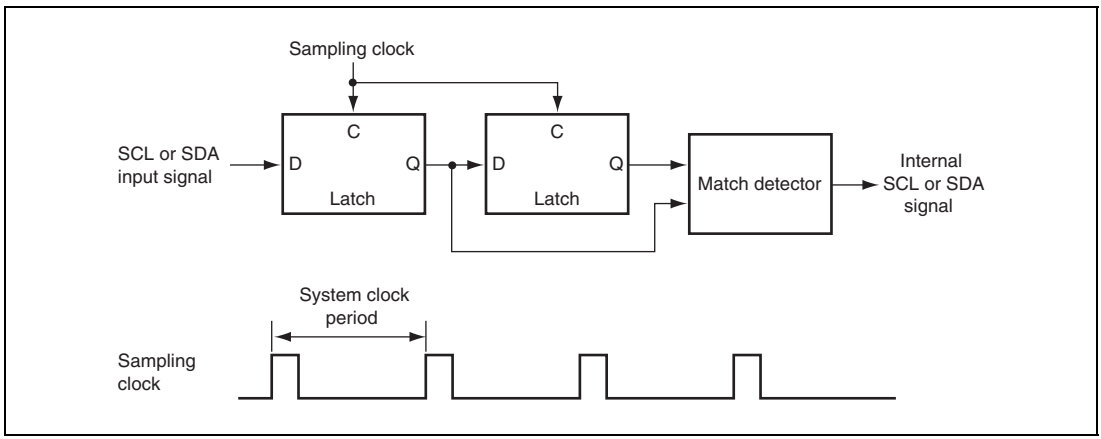


Figure 21.16 Block Diagram of Noise Canceller

21.4.8 Example of Use

Flowcharts in respective modes that use the I²C bus interface are shown in figures 21.17 to 21.20.

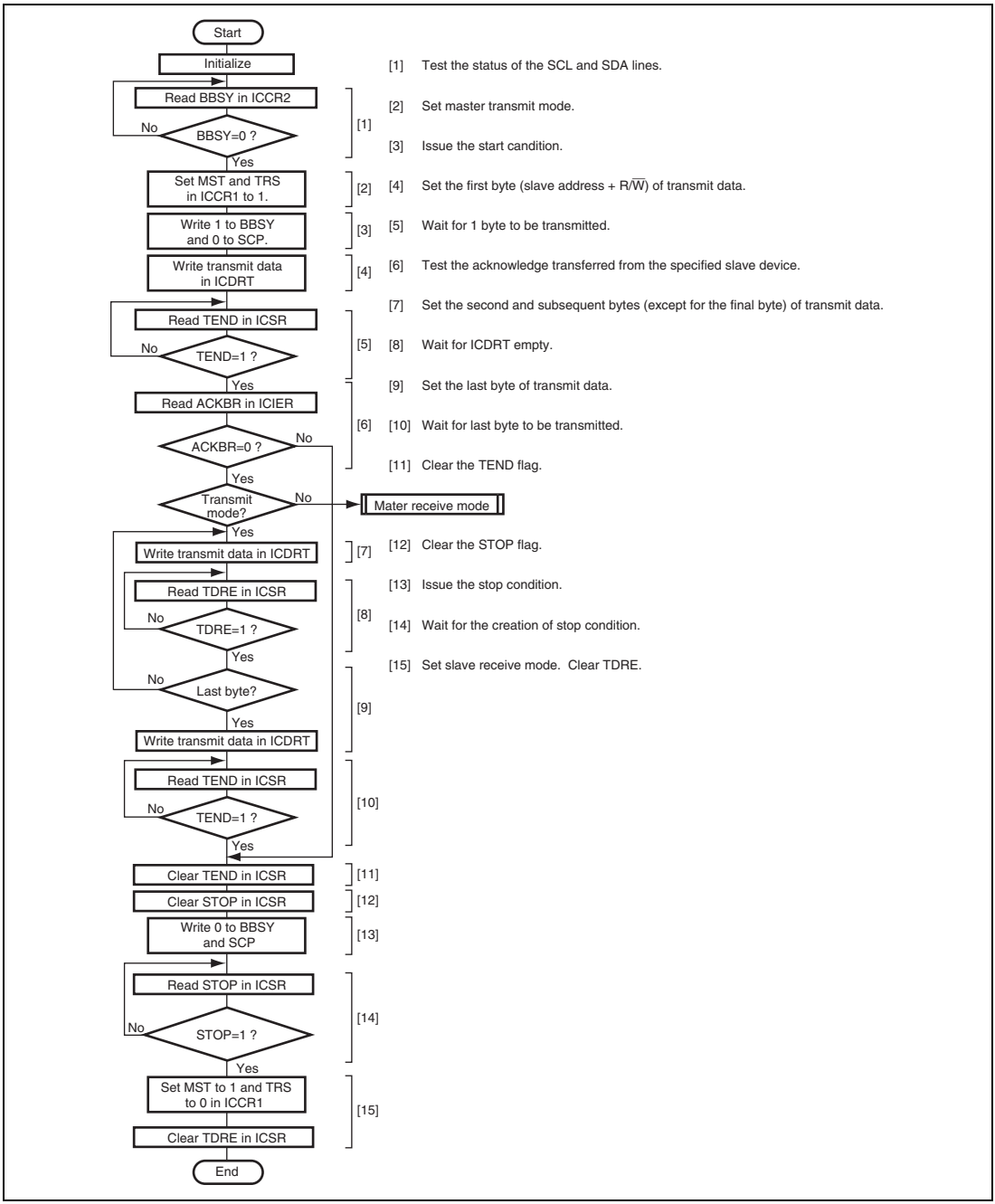
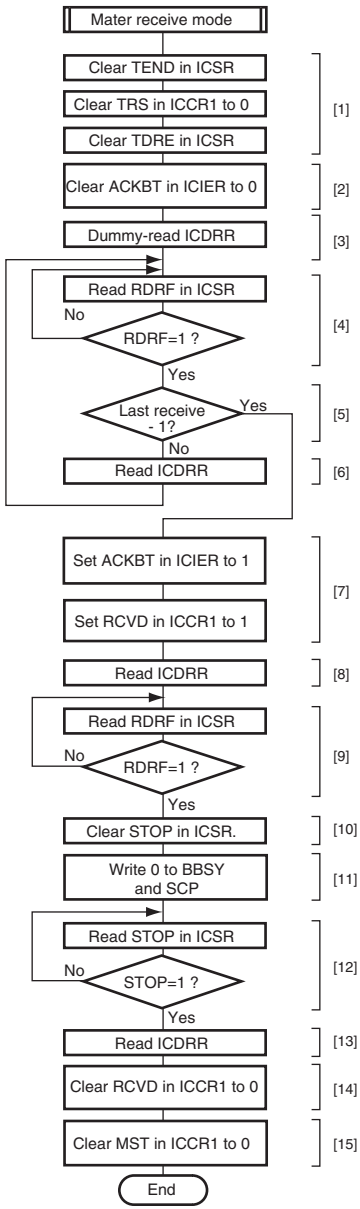


Figure 21.17 Sample Flowchart for Master Transmit Mode



[1] Clear TEND, select master receive mode, and then clear TDRE.*

[2] Set acknowledge to the transmit device.*

[3] Dummy-read ICDRR.*

[4] Wait for 1 byte to be received

[5] Check whether it is the (last receive - 1).

[6] Read the receive data.

[7] Set acknowledge of the final byte. Disable continuous reception (RCVD = 1).

[8] Read the (final byte - 1) of receive data.

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

[10] Clear the STOP flag.

[11] Issue the stop condition.

[12] Wait for the creation of stop condition.

[13] Read the last byte of receive data.

[14] Clear RCVD.

[15] Set slave receive mode.

Note: When 1 byte is received, skip steps [2] to [6] after [1] and then jump to step [7].
In step [8], dummy-read ICDRR.

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

Figure 21.18 Sample Flowchart for Master Receive Mode

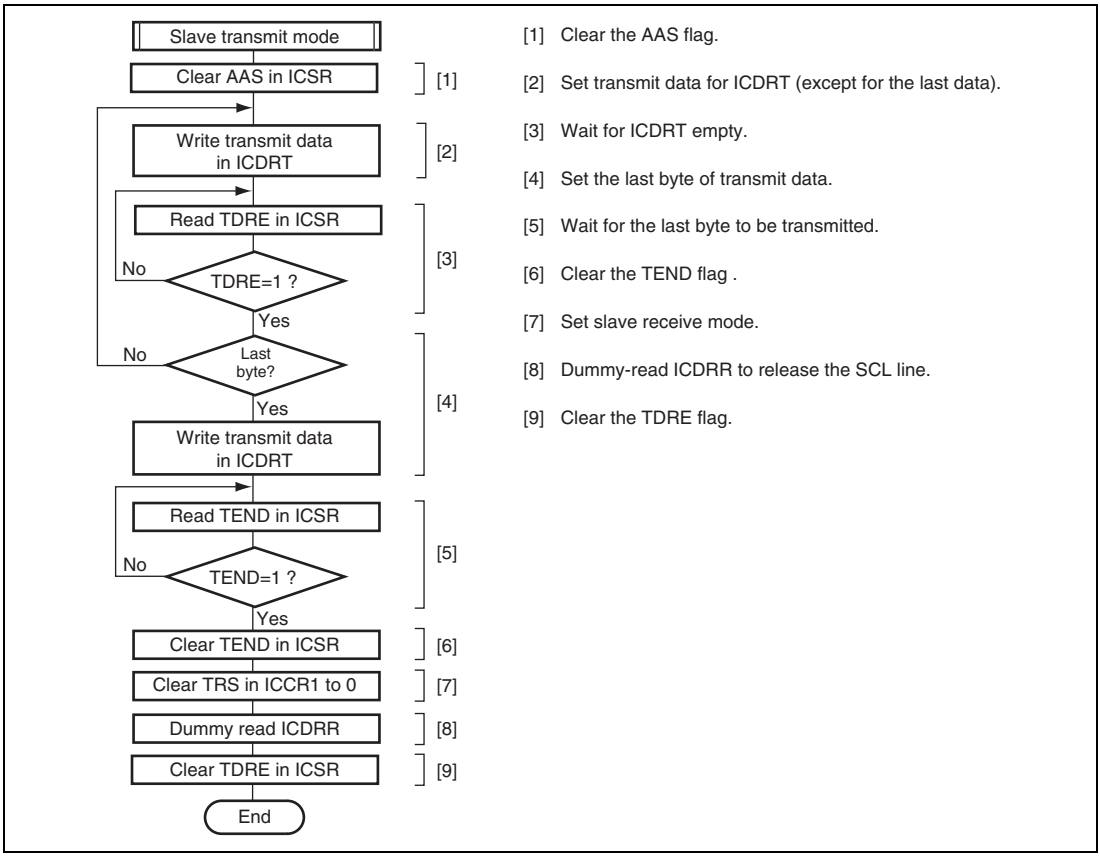


Figure 21.19 Sample Flowchart for Slave Transmit Mode

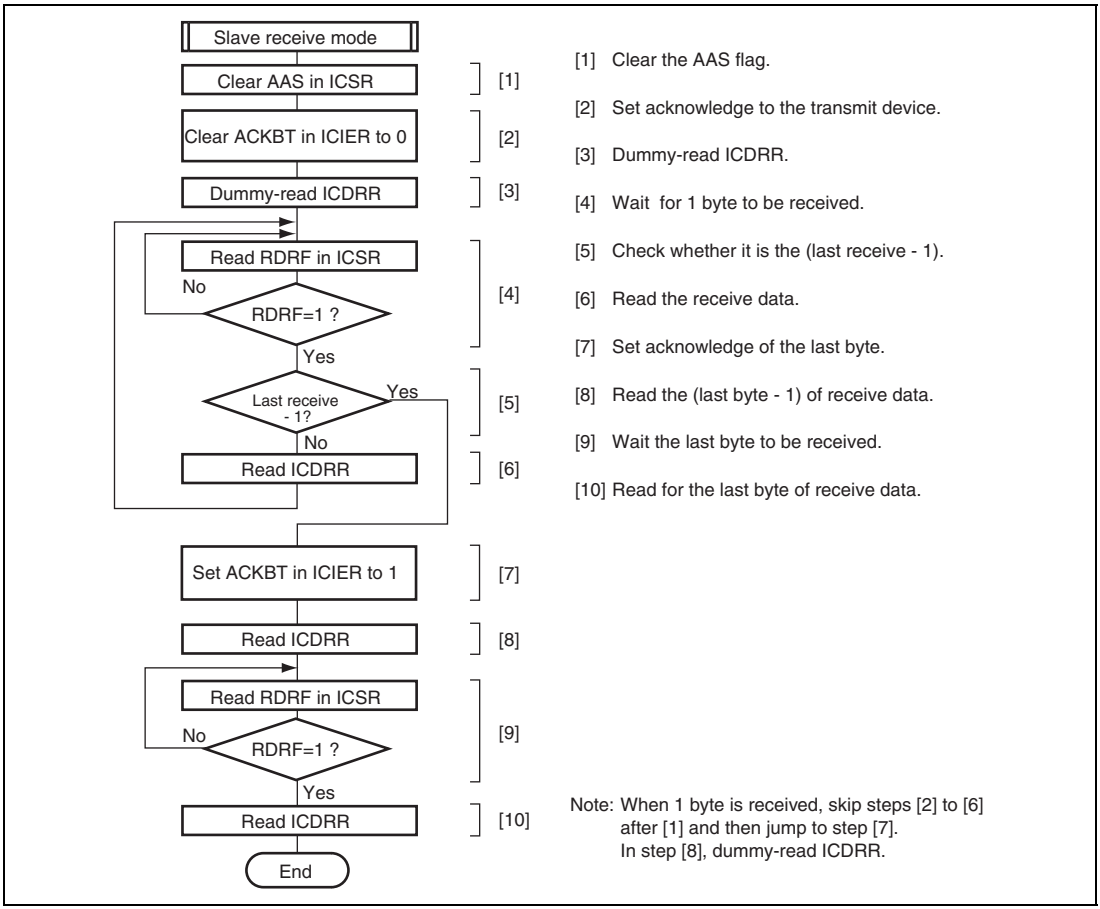


Figure 21.20 Sample Flowchart for Slave Receive Mode

21.5 Interrupt Request

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK receive, STOP recognition, and arbitration lost/overrun. Table 21.3 shows the contents of each interrupt request.

Table 21.3 Interrupt Requests

Interrupt Request	Abbreviation	Interrupt Condition	I²C Mode	Clock Synchronous Mode
Transmit data empty	TXI	(TDRE=1) • (TIE=1)	Available	Available
Transmit end	TEI	(TEND=1) • (TEIE=1)	Available	Available
Receive data full	RXI	(RDRF=1) • (RIE=1)	Available	Available
STOP recognition	STPI	(STOP=1) • (STIE=1)	Available	Not available
NACK receive	NAKI	{(NACKF=1)+(AL=1)} • (NAKIE=1)	Available	Not available
Arbitration lost/overrun			Available	Available

When interrupt conditions described in table 21.3 are 1 and the I bit in CCR is 0, the CPU executes interrupt exception processing. Interrupt sources should be cleared in the exception processing. TDRE and TEND are automatically cleared to 0 by writing the transmit data to ICDRT. RDRF are automatically cleared to 0 by reading ICDRR. TDRE is set to 1 again at the same time when transmit data is written to ICDRT. When TDRE is cleared to 0, then an excessive data of one byte may be transmitted.

21.6 Bit Synchronous Circuit

In master mode, this module has a possibility that high level period may be short in the two states described below.

- When SCL is driven to low by the slave device
- When the rising speed of SCL is lowered by the load of the SCL line (load capacitance or pull-up resistance)

Therefore, it monitors SCL and communicates by bit with synchronization.

Figure 21.21 shows the timing of the bit synchronous circuit and table 21.4 shows the time when SCL output changes from low to Hi-Z then SCL is monitored.

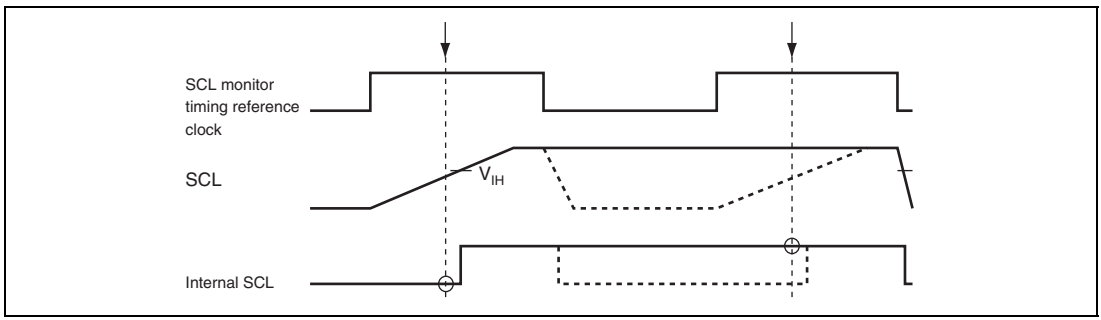


Figure 21.21 Timing of Bit Synchronous Circuit

Table 21.4 Time for Monitoring SCL

CKS3	CKS2	Time for Monitoring SCL
0	0	7.5 t_{cyc}
	1	19.5 t_{cyc}
1	0	17.5 t_{cyc}
	1	41.5 t_{cyc}

21.7 Usage Notes

21.7.1 Note on Issuing Stop Condition and Start (Re-Transmit) Condition

The stop condition or start (re-transmit) condition should be issued after recognizing the falling edge of the ninth clock. The falling edge of the ninth clock can be recognized by checking the SCLO bit in the I²C control register 2 (ICCR2). Note that if the stop condition or start (re-transmit) condition is issued in a particular timing and the situations shown below, these conditions may not correctly output.

1. The rising edge of the SCL becomes less sharp and longer due to the SCL bus load (load capacitor and pull-up resistor) than the period defined in section 21.6, Bit Synchronous Circuit.
2. When the slave device elongates the low level period between the eighth and ninth clocks and activates the bit synchronous circuit.

21.7.2 Note on Setting WAIT Bit in I²C Bus Mode Register (ICMR)

The WAIT bit in the I²C bus mode register (ICMR) should be set to 0. Note that if the WAIT bit is set to 1, when a slave device holds the SCL signal low more than one transfer clock cycle during the eighth clock, the high level period of the ninth clock may be shorter than a given period.

21.7.3 Restriction on Transfer Rate Setting in Multimaster Operation

In multimaster operation, if the IIC transfer rate setting in this LSI is slower than those of the other masters, SCL may be output with an unexpected width. To avoid this phenomenon, set the transfer rate by 1/1.8 or faster than the fastest rate of the other masters. For example, if the fastest transfer rate of the other masters is set to 400 kbps, the IIC transfer rate in this LSI should be set to 223 kbps (= 400/1.18) or more.

21.7.4 Restriction on the Use of Bit Manipulation Instructions for MST and TRS Setting in Multimaster Operation

In multimaster operation, if the master transmit is set with bit manipulation instructions in the order from the MST bit to the TRS bit, the AL bit in the ICSR register will be set to 1 but the master transmit mode (MST = 1, TRS = 1) may be set, depending on the arbitration lost timing. To avoid this phenomenon, the following actions should be performed:

- In multimaster operation, use the MOV instruction to set bits MST and TRS.
- When arbitration is lost, confirm the contents of bits MST and TRS. If the contents are other than MST = 0 and TRS = 0, set MST = 0 and TRS = 0 again.

21.7.5 Usage Note on Master Receive Mode

In master receive mode, SCL is fixed low on the falling edge of the 8th clock while the RDRF bit is set to 1. When ICDRR is read around the falling edge of the 8th clock, the clock is only fixed low in the 8th clock of the next round of data reception. The SCL is then released from its fixed state without reading ICDRR and the 9th clock is output. As a result, some receive data is lost.

To avoid this phenomenon, the following actions should be performed:

- Read ICDRR in master receive mode before the rising edge of the 8th clock.
- Set RCVD to 1 in master receive mode and perform communication in units of one byte.

Section 22 Power-On Reset Circuit

This LSI has an on-chip power-on reset circuit. A block diagram of the power-on reset circuit is shown in figure 22.1.

22.1 Feature

- Power-on reset circuit

  An internal reset signal is generated at turning the power on by externally connecting a capacitor.

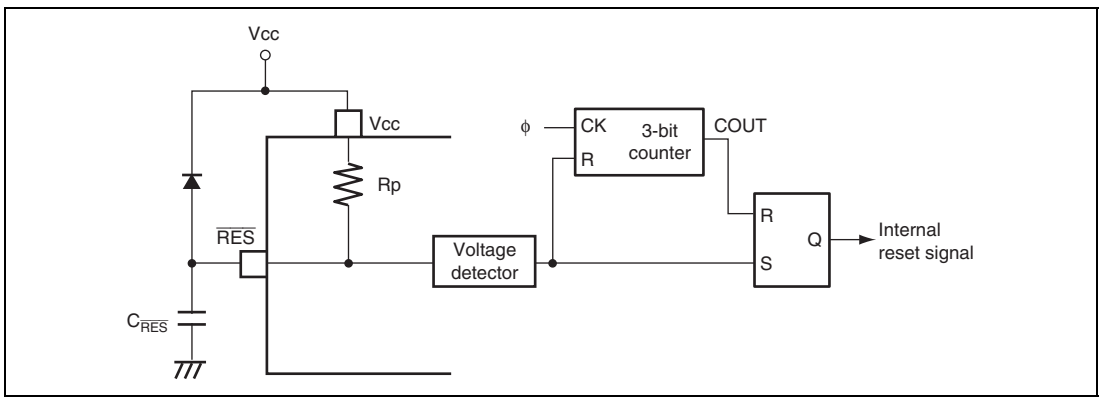


Figure 22.1 Power-On Reset Circuit

22.2 Operation

22.2.1 Power-On Reset Circuit

The operation timing of the power-on reset circuit is shown in figure 22.2. As the power supply voltage rises, the capacitor, which is externally connected to the $\overline{\text{RES}}$ pin, is gradually charged through the on-chip pull-up resistor (Rp). The low level of the $\overline{\text{RES}}$ pin is sent to the LSI and the whole LSI is reset. When the level of the $\overline{\text{RES}}$ pin reaches to the predetermined level, a voltage detection circuit detects it. Then a 3-bit counter starts counting up. When the 3-bit counter counts ϕ for 8 times, an overflow signal is generated and an internal reset signal is negated.

The capacitance ($C_{\overline{\text{RES}}}$) which is connected to the $\overline{\text{RES}}$ pin can be computed using the following formula; where the $\overline{\text{RES}}$ rising time is t. For the on-chip resistor (Rp), see section 25, Electrical Characteristics. The power supply rising time (t_vtr) should be shorter than half the $\overline{\text{RES}}$ rising time (t). The $\overline{\text{RES}}$ rising time (t) is also should be longer than the oscillation stabilization time (trc).

$$C_{\overline{\text{RES}}} = \frac{t}{Rp} \quad (t > trc,\ t > t\_vtr \times 2)$$

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

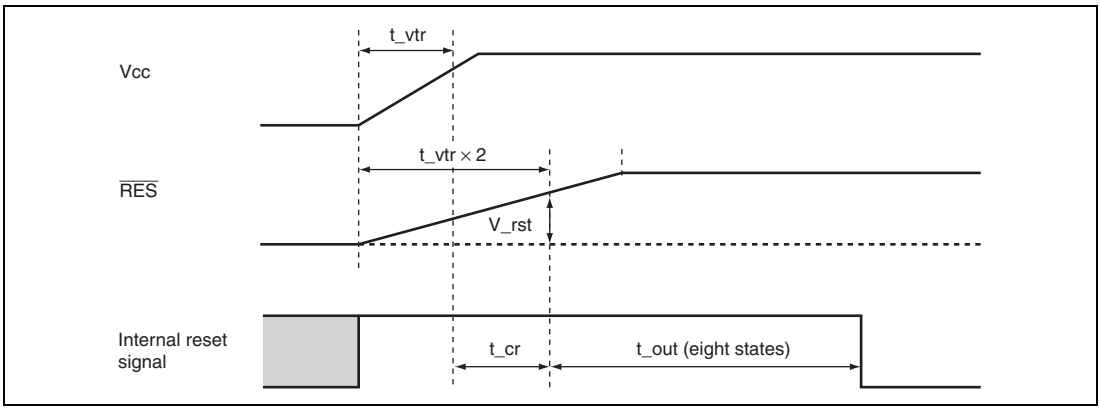


Figure 22.2 Power-On Reset Circuit Operation Timing

Section 23 Address Break

The address break simplifies on-board program debugging. It requests an address break interrupt when the set break condition is satisfied. The interrupt request is not affected by the I bit in CCR. Break conditions that can be set include instruction execution at a specific address and a combination of access and data at a specific address. With the address break function, the execution start point of a program containing a bug is detected and execution is branched to the correcting program. Use of module standby mode enables this module to be placed in standby mode independently when not used (for details, refer to section 6.4, Module Standby Function). Figure 23.1 shows a block diagram of the address break.

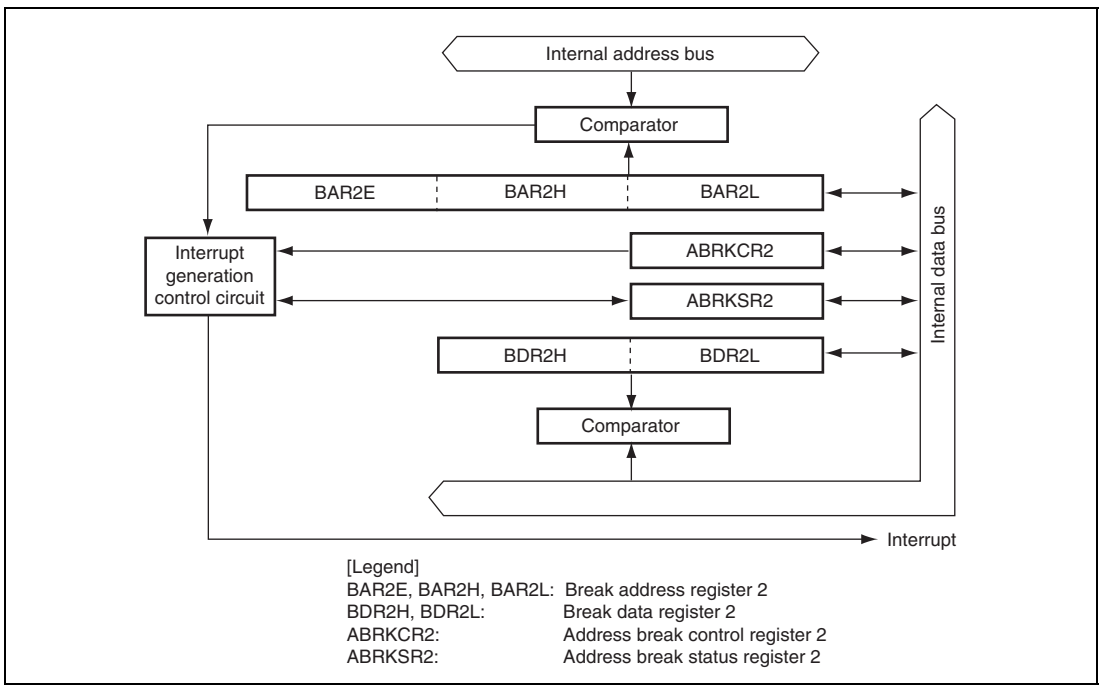


Figure 23.1 Block Diagram of Address Break

23.1 Register Descriptions

The address break has the following registers.

- Address break control register 2 (ABRKCR2)
- Address break status register 2 (ABRKSR2)
- Break address register 2 (BAR2E, BAR2H, BAR2L)
- Break data register 2 (BDR2H, BDR2L)

23.1.1 Address Break Control Register 2 (ABRKCR2)

ABRKCR2 sets address break conditions.

Bit	Bit Name	Initial Value	R/W	Description
7	RTINTE2	1	R/W	RTE Interrupt Enable When this bit is 0, the interrupt immediately after executing RTE is masked and then one instruction must be executed. When this bit is 1, the interrupt is not masked.
6	CSEL21	0	R/W	Condition Select 1 and 0
5	CSEL20	0	R/W	These bits set address break conditions. 00: Instruction execution cycle (no data comparison) 01: CPU data read cycle 10: CPU data write cycle 11: CPU data read/write cycle
4	ACMP22	0	R/W	Address Compare Condition Select 2 to 0
3	ACMP21	0	R/W	These bits set the comparison condition between the address set in BAR2 and the internal address bus.
2	ACMP20	0	R/W	000: Compares all of 24-bit addresses 001: Compares upper 20-bit addresses 010: Compares upper 16-bit addresses 011: Compares upper 12-bit addresses 100: Compares upper 8-bit addresses 101: Compares upper 4-bit addresses 11x: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
1	DCMP21	0	R/W	Data Compare Condition Select 1 and 0
0	DCMP20	0	R/W	These bits set the comparison condition between the data set in BDR2 and the internal data bus. 00: No data comparison 01: Compares lower 8-bit data between BDR2L and data bus 10: Compares upper 8-bit data between BDR2H and data bus 11: Compares 16-bit data between BDR2 and data bus

[Legend]

x: Don't care

When an address break is set in the data read cycle or data write cycle, the data bus used will depend on the combination of the byte/word access and address. Table 23.1 shows the access and data bus used. When an I/O register space with an 8-bit data bus width is accessed in word size, a byte access is generated twice. For details on data widths of each register, see section 24.1, Register Addresses (Address Order).

Table 23.1 Access and Data Bus Used

	Word Access		Byte Access	
	Even Address	Odd Address	Even Address	Odd Address
ROM space	Upper 8 bits	Lower 8 bits	Upper 8 bits	Upper 8 bits
RAM space	Upper 8 bits	Lower 8 bits	Upper 8 bits	Upper 8 bits
I/O register with 8-bit data bus width	Upper 8 bits	Upper 8 bits	Upper 8 bits	Upper 8 bits
I/O register with 16-bit data bus width*[1]	Upper 8 bits	Lower 8 bits	—	—
I/O register with 16-bit data bus width*[2]	Upper 8 bits	Lower 8 bits	Upper 8 bits	Upper 8 bits

Notes: 1. Registers whose addresses do not range from H'FFFF96 and H'FFFF97, and H'FFFFB8 to H'FFFFBB with 16-bit data bus width.

2. Registers whose addresses range from H'FFFF96 and H'FFFF97, and H'FFFFB8 to H'FFFFBB.

23.1.2 Address Break Status Register 2 (ABRKSR2)

ABRKSR2 consists of the address break interrupt flag and the address break interrupt enable bit.

Bit	Bit Name	Initial Value	R/W	Description
7	ABIF2	0	R/W	Address Break Interrupt Flag [Setting condition] When the condition set in ABRKCR2 is satisfied [Clearing condition] When 0 is written after ABIF2=1 is read
6	ABIE2	0	R/W	Address Break Interrupt Enable When this bit is 1, an address break interrupt request is enabled.
5 to 0	—	All 1	—	Reserved These bits are always read as 1.

23.1.3 Break Address Registers 2 (BAR2E, BAR2H, BAR2L)

BAR2E, BAR2H, and BAR2L are 24-bit read/write registers that set the address for generating an address break interrupt. When setting the address break condition to the instruction execution cycle, set the first byte address of the instruction. The initial value of this register is H'FFFFFF.

23.1.4 Break Data Registers 2 (BDR2H, BDR2L)

BDR2H and BDR2L are 16-bit read/write registers that set the data for generating an address break interrupt. BDR2H is compared with the upper 8-bit data bus. BDR2L is compared with the lower 8-bit data bus. When memory or registers are accessed by byte, the upper 8-bit data bus is used for even and odd addresses in the data transmission. Therefore, comparison data must be set in BDR2H for byte access. For word access, the data bus used depends on the address. See section 23.1.1, Address Break Control Register 2 (ABRKCR2), for details. The initial value of this register is undefined.

23.2 Operation

When the ABIF2 and ABIE2 bits in ABRKSR2 are set to 1, the address break function generates an interrupt request to the CPU. The ABIF2 bit in ABRKSR2 is set to 1 by the combination of the address set in BAR2, the data set in BDR2, and the conditions set in ABRKCR2. When the interrupt request is accepted, interrupt exception handling starts after the instruction being executed ends. The address break interrupt is not masked by the I bit in CCR of the CPU.

Figures 23.2 show the operation examples of the address break interrupt setting.

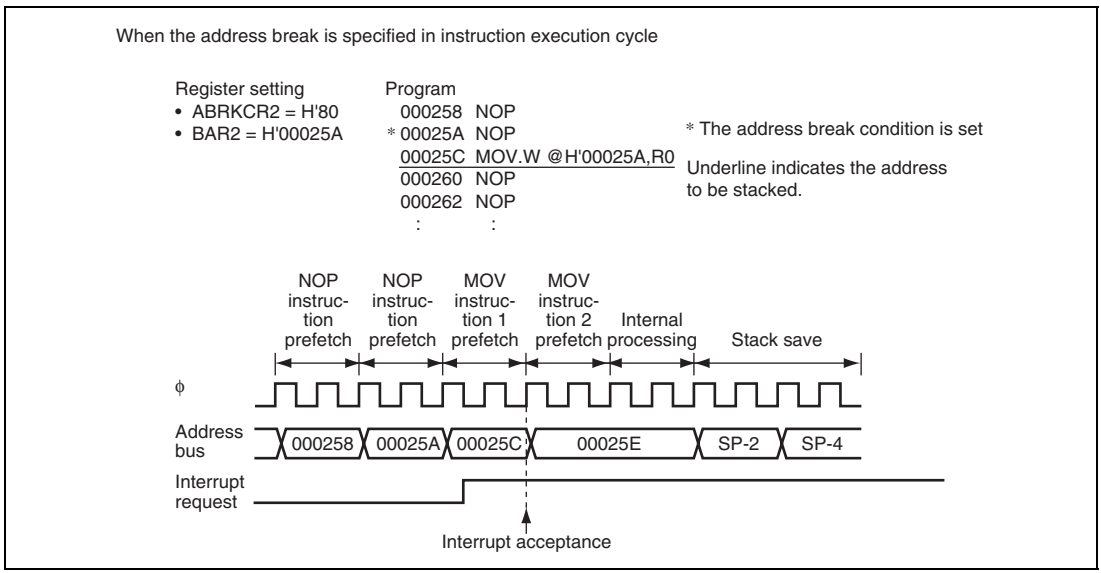


Figure 23.2 Address Break Interrupt Operation Example (1)

When the address break is specified in the data read cycle

Register setting
- ABRKCR2 = H'A0
- BAR2 = H'00025A

```
Program
  000258  NOP
  00025A  NOP
* 00025C  MOV.W @H'00025A,R0   * The address break condition is set
  000260  NOP
  000262  NOP                  Underline indicates the address
    :       :                  to be stacked.
```

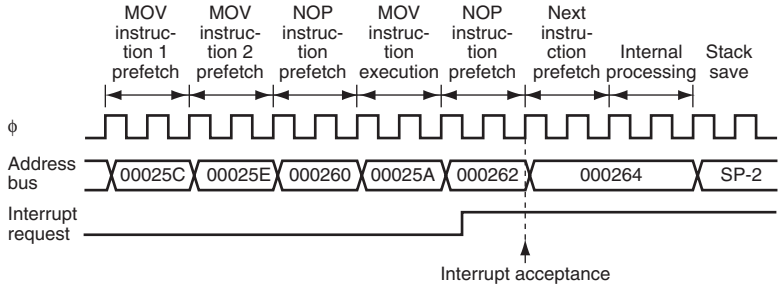


Figure 23.2 Address Break Interrupt Operation Example (2)

23.3 Operating States of Address Break

The operating states of the address break are shown in table 23.2.

Table 23.2 Operating States of Address Break

Operating Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby	Module Standby
ABRKCR2	Reset	Functioning	Retained	Retained	Functioning	Retained	Retained	Retained
ABRKSR2	Reset	Functioning	Retained	Retained	Functioning	Retained	Retained	Retained
BAR2E	Reset	Functioning	Retained	Retained	Functioning	Retained	Retained	Retained
BAR2H	Reset	Functioning	Retained	Retained	Functioning	Retained	Retained	Retained
BAR2L	Reset	Functioning	Retained	Retained	Functioning	Retained	Retained	Retained
BDR2H	Retained*	Functioning	Retained	Retained	Functioning	Retained	Retained	Retained
BDR2L	Retained*	Functioning	Retained	Retained	Functioning	Retained	Retained	Retained

Note: * Undefined at a power-on reset

Section 24 List of Registers

The register list gives information on the on-chip register addresses, how the register bits are configured, and the register states in each operating mode. The information is given as shown below.

1. Register addresses (address order)
- Registers are listed from the lower allocation addresses.
- 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 addresses.
- 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 specific reset for an on-chip peripheral module, refer to the section on that on-chip peripheral module.

24.1 Register Addresses (Address Order)

The data bus width indicates the number of bits by which the register is accessed.

The number of access states indicates the number of states based on the specified reference clock.

Register Name	Abbre-viation	Bit No.	Address*1	Module Name	Data Bus Width	Access State
Serial control register 4	SCR4	8	H'F00C	SCI4	8	2
Serial control/status register 4	SCSR4	8	H'F00D	SCI4	8	2
Transmit data register 4	TDR4	8	H'F00E	SCI4	8	2
Receive data register 4	RDR4	8	H'F00F	SCI4	8	2
Flash memory control register 1	FLMCR1	8	H'F020	ROM	8	2
Flash memory control register 2	FLMCR2	8	H'F021	ROM	8	2
Flash memory power control register	FLPWCR	8	H'F022	ROM	8	2
Erase block register 1	EBR1	8	H'F023	ROM	8	2
Flash memory enable register	FENR	8	H'F02B	ROM	8	2
Erase block register 2	EBR2	8	H'F02C	ROM	8	2
System control register 3	SYSCR3	8	H'F02F	SYSTEM	8	2
Timer start register	TSTR	8	H'F030	TPU	8	2
Timer synchro register	TSYR	8	H'F031	TPU	8	2
Port data register E	PDRE	8	H'F033	I/O ports	8	2
Port data register F	PDRF	8	H'F034	I/O ports	8	2
Port control register E	PCRE	8	H'F037	I/O ports	8	2
Port control register F	PCRF	8	H'F038	I/O ports	8	2
Port mode register E	PMRE	8	H'F03B	I/O ports	8	2
Port mode register F	PMRF	8	H'F03C	I/O ports	8	2
Timer control register_1	TCR_1	8	H'F040	TPU_1	8	2
Timer mode register_1	TMDR_1	8	H'F041	TPU_1	8	2
Timer I/O control register_1	TIOR_1	8	H'F042	TPU_1	8	2
Timer interrupt enable register_1	TIER_1	8	H'F044	TPU_1	8	2
Timer status register_1	TSR_1	8	H'F045	TPU_1	8	2

Register Name	Abbre-viation	Bit No.	Address*1	Module Name	Data Bus Width	Access State
Timer counter_1	TCNT_1	16	H'F046	TPU_1	16	2
Timer general register A_1	TGRA_1	16	H'F048	TPU_1	16	2
Timer general register B_1	TGRB_1	16	H'F04A	TPU_1	16	2
PWM3 control register	PWCR3	8	H'F04C	14-bit PWM_3	8	2
PWM3 data register	PWDR3	16	H'F04E	14-bit PWM_3	16	2
Timer control register_2	TCR_2	8	H'F050	TPU_2	8	2
Timer mode register_2	TMDR_2	8	H'F051	TPU_2	8	2
Timer I/O control register_2	TIOR_2	8	H'F052	TPU_2	8	2
Timer interrupt enable register_2	TIER_2	8	H'F054	TPU_2	8	2
Timer status register_2	TSR_2	8	H'F055	TPU_2	8	2
Timer counter_2	TCNT_2	16	H'F056	TPU_2	16	2
Timer general register A_2	TGRA_2	16	H'F058	TPU_2	16	2
Timer general register B_2	TGRB_2	16	H'F05A	TPU_2	16	2
PWM4 control register	PWCR4	8	H'F05D	14-bit PWM_4	8	2
PWM4 data register	PWDR4	16	H'F05E	14-bit PWM_4	16	2
RTC Interrupt flag register	RTCFLG	8	H'F067	RTC	8	2
Second data register/free running counter data register	RSECDR	8	H'F068	RTC	8	2
Minute data register	RMINDR	8	H'F069	RTC	8	2
Hour data register	RHRDR	8	H'F06A	RTC	8	2
Day-of-week data register	RWKDR	8	H'F06B	RTC	8	2
RTC control register 1	RTCCR1	8	H'F06C	RTC	8	2
RTC control register 2	RTCCR2	8	H'F06D	RTC	8	2
SUB32k control register	SUB32CR	8	H'F06E	Clock pulse generator	8	2
Clock source select register	RTCCSR	8	H'F06F	RTC	8	2
I^2C bus control register 1	ICCR1	8	H'F078	IIC2	8	2
I^2C bus control register 2	ICCR2	8	H'F079	IIC2	8	2
I^2C bus mode register	ICMR	8	H'F07A	IIC2	8	2
I^2C bus interrupt enable register	ICIER	8	H'F07B	IIC2	8	2

Register Name	Abbre-viation	Bit No.	Address*1	Module Name	Data Bus Width	Access State
I^2C bus status register	ICSR	8	H'F07C	IIC2	8	2
Slave address register	SAR	8	H'F07D	IIC2	8	2
I^2C bus transmit data register	ICDRT	8	H'F07E	IIC2	8	2
I^2C bus receive data register	ICDRR	8	H'F07F	IIC2	8	2
Interrupt priority register A	IPRA	8	H'F080	Interrupts	8	2
Interrupt priority register B	IPRB	8	H'F081	Interrupts	8	2
Interrupt priority register C	IPRC	8	H'F082	Interrupts	8	2
Interrupt priority register D	IPRD	8	H'F083	Interrupts	8	2
Interrupt priority register E	IPRE	8	H'F084	Interrupts	8	2
Interrupt priority register F	IPRF	8	H'F085	Interrupts	8	2
Serial mode register 3_3	SMR3_3	8	H'F088	SCI3_3	8	3
Bit rate register 3_3	BRR3_3	8	H'F089	SCI3_3	8	3
Serial control register 3_3	SCR3_3	8	H'F08A	SCI3_3	8	3
Transmit data register 3_3	TDR3_3	8	H'F08B	SCI3_3	8	3
Serial status register 3_3	SSR3_3	8	H'F08C	SCI3_3	8	3
Receive data register 3_3	RDR3_3	8	H'F08D	SCI3_3	8	3
Address break control register 2	ABRKCR2	8	H'F096	Address break	8	2
Address break status register 2	ABRKSR2	8	H'F097	Address break	8	2
Break address register 2H	BAR2H	8	H'F098	Address break	8	2
Break address register 2L	BAR2L	8	H'F099	Address break	8	2
Break data register 2H	BDR2H	8	H'F09A	Address break	8	2
Break data register 2L	BDR2L	8	H'F09B	Address break	8	2
Break address register 2E	BAR2E	8	H'F09D	Address break	8	2
Timer mode register G	TMG	8	H'FF84	Timer G	8	2
Input capture register GF	ICRGF	8	H'FF85	Timer G	8	2
Input capture register GR	ICRGR	8	H'FF86	Timer G	8	2
Event counter PWM compare register	ECPWCR	16	H'FF8C	AEC*2	16	2
Event counter PWM data register	ECPWDR	16	H'FF8E	AEC*2	16	2
Wakeup edge select register	WEGR	8	H'FF90	Interrupts	8	2

Register Name	Abbre-viation	Bit No.	Address*1	Module Name	Data Bus Width	Access State
Serial port control register	SPCR	8	H'FF91	SCI3	8	2
Input pin edge select register	AEGSR	8	H'FF92	AEC*2	8	2
Event counter control register	ECCR	8	H'FF94	AEC*2	8	2
Event counter control/status register	ECCSR	8	H'FF95	AEC*2	8	2
Event counter H	ECH	8	H'FF96	AEC*2	8/16	2
Event counter L	ECL	8	H'FF97	AEC*2	8/16	2
Serial mode register 3_1	SMR3_1	8	H'FF98	SCI3_1	8	3
Bit rate register 3_1	BRR3_1	8	H'FF99	SCI3_1	8	3
Serial control register 3_1	SCR3_1	8	H'FF9A	SCI3_1	8	3
Transmit data register 3_1	TDR3_1	8	H'FF9B	SCI3_1	8	3
Serial status register 3_1	SSR3_1	8	H'FF9C	SCI3_1	8	3
Receive data register 3_1	RDR3_1	8	H'FF9D	SCI3_1	8	3
Serial extended mode register	SEMR	8	H'FFA6	SCI3_1	8	3
IrDA control register	IrCR	8	H'FFA7	IrDA	8	3
Serial mode register 3_2	SMR3_2	8	H'FFA8	SCI3_2	8	3
Bit rate register 3_2	BRR3_2	8	H'FFA9	SCI3_2	8	3
Serial control register 3_2	SCR3_2	8	H'FFAA	SCI3_2	8	3
Transmit data register 3_2	TDR3_2	8	H'FFAB	SCI3_2	8	3
Serial status register 3_2	SSR3_2	8	H'FFAC	SCI3_2	8	3
Receive data register 3_2	RDR3_2	8	H'FFAD	SCI3_2	8	3
Timer mode register WD	TMWD	8	H'FFB0	WDT*3	8	2
Timer control/status register WD1	TCSRWD1	8	H'FFB1	WDT*3	8	2
Timer control/status register WD2	TCSRWD2	8	H'FFB2	WDT*3	8	2
Timer counter WD	TCWD	8	H'FFB3	WDT*3	8	2
Timer mode register C	TMC	8	H'FFB4	Timer C	8	2
Timer counter C/Timer load register C	TCC/TLC	8	H'FFB5	Timer C	8	2
Timer control register F	TCRF	8	H'FFB6	Timer F	8	2

Register Name	Abbre-viation	Bit No.	Address*[1]	Module Name	Data Bus Width	Access State
Timer control/status register F	TCSRF	8	H'FFB7	Timer F	8	2
Timer counter FH	TCFH	8	H'FFB8	Timer F	8/16	2
Timer counter FL	TCFL	8	H'FFB9	Timer F	8/16	2
Output compare register FH	OCRFH	8	H'FFBA	Timer F	8/16	2
Output compare register FL	OCRFL	8	H'FFBB	Timer F	8/16	2
A/D result register	ADRR	16	H'FFBC	A/D converter	16	2
A/D mode register	AMR	8	H'FFBE	A/D converter	8	2
A/D start register	ADSR	8	H'FFBF	A/D converter	8	2
Port mode register 1	PMR1	8	H'FFC0	I/O ports	8	2
Oscillator control register	OSCCR	8	H'FFC1	Clock pulse generator	8	2
Port mode register 3	PMR3	8	H'FFC2	I/O ports	8	2
Port mode register 4	PMR4	8	H'FFC3	I/O ports	8	2
Port mode register 5	PMR5	8	H'FFC4	I/O ports	8	2
Port mode register 9	PMR9	8	H'FFC8	I/O ports	8	2
Port mode register B	PMRB	8	H'FFCA	I/O ports	8	2
Port function control register	PFCR	8	H'FFCB	I/O ports	8	2
Serial port control register 2	SPCR2	8	H'FFCC	SCI3	8	2
PWM2 control register	PWCR2	8	H'FFCD	14-bit PWM_2	8	2
PWM2 data register	PWDR2	16	H'FFCE	14-bit PWM_2	16	2
PWM1 control register	PWCR1	8	H'FFD0	14-bit PWM_1	8	2
PWM1 data register	PWDR1	16	H'FFD2	14-bit PWM_1	16	2
Port data register 1	PDR1	8	H'FFD4	I/O ports	8	2
Port data register 3	PDR3	8	H'FFD6	I/O ports	8	2
Port data register 4	PDR4	8	H'FFD7	I/O ports	8	2
Port data register 5	PDR5	8	H'FFD8	I/O ports	8	2
Port data register 6	PDR6	8	H'FFD9	I/O ports	8	2
Port data register 7	PDR7	8	H'FFDA	I/O ports	8	2
Port data register 8	PDR8	8	H'FFDB	I/O ports	8	2

Register Name	Abbre-viation	Bit No.	Address*1	Module Name	Data Bus Width	Access State
Port data register 9	PDR9	8	H'FFDC	I/O ports	8	2
Port data register A	PDRA	8	H'FFDD	I/O ports	8	2
Port data register B	PDRB	8	H'FFDE	I/O ports	8	2
Port data register C	PDRC	8	H'FFDF	I/O ports	8	2
Port pull-up control register 1	PUCR1	8	H'FFE0	I/O ports	8	2
Port pull-up control register 3	PUCR3	8	H'FFE1	I/O ports	8	2
Port pull-up control register 5	PUCR5	8	H'FFE2	I/O ports	8	2
Port pull-up control register 6	PUCR6	8	H'FFE3	I/O ports	8	2
Port control register 1	PCR1	8	H'FFE4	I/O ports	8	2
Port control register 3	PCR3	8	H'FFE6	I/O ports	8	2
Port control register 4	PCR4	8	H'FFE7	I/O ports	8	2
Port control register 5	PCR5	8	H'FFE8	I/O ports	8	2
Port control register 6	PCR6	8	H'FFE9	I/O ports	8	2
Port control register 7	PCR7	8	H'FFEA	I/O ports	8	2
Port control register 8	PCR8	8	H'FFEB	I/O ports	8	2
Port control register 9	PCR9	8	H'FFEC	I/O ports	8	2
Port control register A	PCRA	8	H'FFED	I/O ports	8	2
Port control register C	PCRC	8	H'FFEE	I/O ports	8	2
System control register 1	SYSCR1	8	H'FFF0	SYSTEM	8	2
System control register 2	SYSCR2	8	H'FFF1	SYSTEM	8	2
Interrupt edge select register	IEGR	8	H'FFF2	Interrupts	8	2
Interrupt enable register 1	IENR1	8	H'FFF3	Interrupts	8	2
Interrupt enable register 2	IENR2	8	H'FFF4	Interrupts	8	2
Interrupt mask register	INTM	8	H'FFF5	Interrupts	8	2
Interrupt request register 1	IRR1	8	H'FFF6	Interrupts	8	2
Interrupt request register 2	IRR2	8	H'FFF7	Interrupts	8	2
Wakeup interrupt request register	IWPR	8	H'FFF9	Interrupts	8	2
Clock halt register 1	CKSTPR1	8	H'FFFA	SYSTEM	8	2

Register Name	Abbre-viation	Bit No.	Address*[1]	Module Name	Data Bus Width	Access State
Clock halt register 2	CKSTPR2	8	H'FFFB	SYSTEM	8	2
Clock halt register 3	CKSTPR3	8	H'FFFC	SYSTEM	8	2

Notes: 1. Indicates the lower 16-bit address.
2. AEC: Asynchronous event counter
3. WDT: Watchdog timer

24.2 Register Bits

Register bit names of the on-chip peripheral modules are described below.

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
SCR4	TIE	RIE	TEIE	SOL	SOLP	SRES	TE	RE	SCI4
SCSR4	TDRE	RDRF	ORER	TEND	CKS3	CKS2	CKS1	CKS0	
TDR4	TDR47	TDR46	TDR45	TDR44	TDR43	TDR42	TDR41	TDR40	
RDR4	RDR47	RDR46	RDR45	RDR44	RDR43	RDR42	RDR41	RDR40	
FLMCR1	—	SWE	ESU	PSU	EV	PV	E	P	ROM
FLMCR2	FLER	—	—	—	—	—	—	—	
FLPWCR	PDWND	—	—	—	—	—	—	—	
EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
FENR	FLSHE	—	—	—	—	—	—	—	
EBR2	—	—	—	—	—	—	EB9	EB8	
SYSCR3	—	—	—	—	—	—	—	STS3	SYSTEM
TSTR	—	—	—	—	—	CST2	CST1	—	TPU
TSYR	—	—	—	—	—	SYNC2	SYNC1	—	
PDRE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	I/O ports
PDRF	—	—	—	—	PF3	PF2	PF1	PF0	
PCRE	PCRE7	PCRE6	PCRE5	PCRE4	PCRE3	PCRE2	PCRE1	PCRE0	
PCRF	—	—	—	—	PCRF3	PCRF2	PCRF1	PCRF0	
PMRE	—	—	—	TMIC	IRQ0	UD	IRQ1	IRQ3	
PMRF	—	—	—	—	—	IRQ4	NCS	TMIG	
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_1
TMDR_1	—	—	—	—	—	—	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	—	—	—	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	—	—	—	TCFV	—	—	TGFB	TGFA	
TCNT_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
TGRB_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	TPU_1
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
PWCR3	—	—	—	—	—	PWCR32	PWCR31	PWCR30	14-bit PWM_3
PWDR3	—	—	PWDR313	PWDR312	PWDR311	PWDR310	PWDR39	PWDR38	
	PWDR37	PWDR36	PWDR35	PWDR34	PWDR33	PWDR32	PWDR31	PWDR30	
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_2
TMDR_2	—	—	—	—	—	—	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	—	—	—	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	—	—	—	TCFV	—	—	TGFB	TGFA	
TCNT_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
PWCR4	—	—	—	—	—	PWCR42	PWCR41	PWCR40	14-bit PWM_4
PWDR4	—	—	PWDR413	PWDR412	PWDR411	PWDR410	PWDR49	PWDR48	
	PWDR47	PWDR46	PWDR45	PWDR44	PWDR43	PWDR42	PWDR41	PWDR40	
RTCFLG	FOIFG	WKIFG	DYIFG	HRIFG	MNIFG	SEIFG	05SEIFG	025SEIFG	RTC
RSECDR	BSY	SC12	SC11	SC10	SC03	SC02	SC01	SC00	
RMINDR	BSY	MN12	MN11	MN10	MN03	MN02	MN01	MN00	
RHRDR	BSY	—	HR11	HR10	HR03	HR02	HR01	HR00	
RWKDR	BSY	—	—	—	—	WK2	WK1	WK0	
RTCCR1	RUN	12/24	PM	RST	—	—	—	—	
RTCCR2	FOIE	WKIE	DYIE	HRIE	MNIE	1SEIE	05SEIE	025SEIE	
SUB32CR	32KSTOP	—	—	—	—	—	—	—	Clock pulse generator
RTCCSR	—	RCS6	RCS5	SUB32K	RCS3	RCS2	RCS1	RCS0	RTC
ICCR1	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC2
ICCR2	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
ICMR	MLS	WAIT	—	—	BCWP	BC2	BC1	BC0	IIC2
ICIER	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ	
SAR	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ICDRT	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	
IPRA	IPRA7	IPRA6	IPRA5	IPRA4	IPRA3	IPRA2	IPRA1	IPRA0	Interrupts
IPRB	IPRB7	IPRB6	IPRB5	IPRB4	IPRB3	IPRB2	IPRB1	IPRB0	
IPRC	IPRC7	IPRC6	IPRC5	IPRC4	IPRC3	IPRC2	IPRC1	IPRC0	
IPRD	IPRD7	IPRD6	IPRD5	IPRD4	IPRD3	IPRD2	IPRD1	IPRD0	
IPRE	IPRE7	IPRE6	IPRE5	IPRE4	—	—	—	—	
IPRF	IPRF7	IPRF6	IPRF5	IPRF4	IPRF3	IPRF2	—	—	
SMR3_3	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0	SCI3_3
BRR3_3	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0	
SCR3_3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR3_3	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	
SSR3_3	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT	
RDR3_3	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0	
ABRKCR2	RTINTE2	CSEL21	CSEL20	ACMP22	ACMP21	ACMP20	DCMP21	DCMP20	Address break
ABRKSR2	ABIF2	ABIE2	—	—	—	—	—	—	
BAR2H	BARH27	BARH26	BARH25	BARH24	BARH23	BARH22	BARH21	BARH20	
BAR2L	BARL27	BARL26	BARL25	BARL24	BARL23	BARL22	BARL21	BARL20	
BDR2H	BDRH27	BDRH26	BDRH25	BDRH24	BDRH23	BDRH22	BDRH21	BDRH20	
BDR2L	BDRL27	BDRL26	BDRL25	BDRL24	BDRL23	BDRL22	BDRL21	BDRL20	
BAR2E	BARE27	BARE26	BARE25	BARE24	BARE23	BARE22	BARE21	BARE20	
TMG	OVFH	OVFL	OVIE	IIEGS	CCLR1	CCLR0	CKS1	CKS0	Timer G
ICRGF	ICRGF7	ICRGF6	ICRGF5	ICRGF4	ICRGF3	ICRGF2	ICRGF1	ICRGF0	
ICRGR	ICRGR7	ICRGR6	ICRGR5	ICRGR4	ICRGR3	ICRGR2	ICRGR1	ICRGR0	
ECPWCR	ECPWCR15	ECPWCR14	ECPWCR13	ECPWCR12	ECPWCR11	ECPWCR10	ECPWCR9	ECPWCR8	AEC*1
	ECPWCR7	ECPWCR6	ECPWCR5	ECPWCR4	ECPWCR3	ECPWCR2	ECPWCR1	ECPWCR0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
ECPWDR	ECPWDR15	ECPWDR14	ECPWDR13	ECPWDR12	ECPWDR11	ECPWDR10	ECPWDR9	ECPWDR8	AEC*[1]
	ECPWDR7	ECPWDR6	ECPWDR5	ECPWDR4	ECPWDR3	ECPWDR2	ECPWDR1	ECPWDR0	
WEGR	WKEGS7	WKEGS6	WKEGS5	WKEGS4	WKEGS3	WKEGS2	WKEGS1	WKEGS0	Interrupts
SPCR	—	—	SPC32	SPC31	SCINV3	SCINV2	SCINV1	SCINV0	SCI3
AEGSR	AHEGS1	AHEGS0	ALEGS1	ALEGS0	AIEGS1	AIEGS0	ECPWME	—	AEC*[1]
ECCR	ACKH1	ACKH0	ACKL1	ACKL0	PWCK2	PWCK1	PWCK0	—	
ECCSR	OVH	OVL	—	CH2	CUEH	CUEL	CRCH	CRCL	
ECH	ECH7	ECH6	ECH5	ECH4	ECH3	ECH2	ECH1	ECH0	
ECL	ECL7	ECL6	ECL5	ECL4	ECL3	ECL2	ECL1	ECL0	
SMR3_1	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0	SCI3_1
BRR3_1	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0	
SCR3_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR3_1	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	
SSR3_1	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT	
RDR3_1	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0	
SEMR	—	—	—	—	ABCS	—	—	—	
IrCR	IrE	IrCKS2	IrCKS1	IrCKS0	—	—	—	—	IrDA
SMR3_2	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0	SCI3_2
BRR3_2	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0	
SCR3_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR3_2	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	
SSR3_2	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT	
RDR3_2	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0	
TMWD	—	—	—	—	CKS3	CKS2	CKS1	CKS0	WDT*[2]
TCSRWD1	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	BOWI	WRST	
TCSRWD2	OVF	B5WI	WT/$\overline{IT}$	B3WI	IEOVF	—	—	—	
TCWD	TCW7	TCW6	TCW5	TCW4	TCW3	TCW2	TCW1	TCW0	
TMC	TMC7	TMC6	TMC5	—	—	TMC2	TMC1	TMC0	Timer C
TCC/TLC	TCC7/ TLC7	TCC6/ TLC6	TCC5/ TLC5	TCC4/ TLC4	TCC3/ TLC3	TCC2/ TLC2	TCC1/ TLC1	TCC0/ TLC0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
TCRF	TOLH	CKSH2	CKSH1	CKSH0	TOLL	CKSL2	CKSL1	CKSL0	Timer F
TCSRF	OVFH	CMFH	OVIEH	CCLRH	OVFL	CMFL	OVIEL	CCLRL	
TCFH	TCFH7	TCFH6	TCFH5	TCFH4	TCFH3	TCFH2	TCFH1	TCFH0	
TCFL	TCFL7	TCFL6	TCFL5	TCFL4	TCFL3	TCFL2	TCFL1	TCFL0	
OCRFH	OCRFH7	OCRFH6	OCRFH5	OCRFH4	OCRFH3	OCRFH2	OCRFH1	OCRFH0	
OCRFL	OCRFL7	OCRFL6	OCRFL5	OCRFL4	OCRFL3	OCRFL2	OCRFL1	OCRFL0	
ADRR	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	A/D converter
	ADR1	ADR0	—	—	—	—	—	—	
AMR	—	TRGE	CKS1	CKS0	CH3	CH2	CH1	CH0	
ADSR	ADSF	LADS	—	—	—	—	—	—	
PMR1	—	—	—	—	—	—	AEVL	AEVH	I/O ports
OSCCR	—	RFCUT	—	—	—	IRQAECF	OSCF	—	Clock pulse generator
PMR3	—	—	—	—	—	—	—	TMOW	I/O ports
PMR4	—	—	—	—	—	TMOFH	TMOFL	TMIF	
PMR5	WKP7	WKP6	WKP5	WKP4	WKP3	WKP2	WKP1	WKP0	
PMR9	—	—	—	—	—	IRQ4	PWM2	PWM1	
PMRB	—	—	—	ADTSTCHG	—	IRQ3	IRQ1	IRQ0	
PFCR	CLKOUT1	CLKOUT0	—	PWM4	PWM3	—	SC32S	SC31S	
SPCR2	—	—	—	SPC33	—	—	SCINV5	SCINV4	SCI3
PWCR2	—	—	—	—	—	PWCR22	PWCR21	PWCR20	14-bit PWM_2
PWDR2	—	—	PWDR213	PWDR212	PWDR211	PWDR210	PWDR29	PWDR28	
	PWDR27	PWDR26	PWDR25	PWDR24	PWDR23	PWDR22	PWDR21	PWDR20	
PWCR1	—	—	—	—	—	PWCR12	PWCR11	PWCR10	14-bit PWM_1
PWDR1	—	—	PWDR113	PWDR112	PWDR111	PWDR110	PWDR19	PWDR18	
	PWDR17	PWDR16	PWDR15	PWDR14	PWDR13	PWDR12	PWDR11	PWDR10	
PDR1	—	P16	P15	P14	P13	P12	P11	P10	I/O ports
PDR3	P37	P36	—	—	—	P32	P31	P30	
PDR4	—	—	—	—	—	P42	P41	P40	
PDR5	P57	P56	P55	P54	P53	P52	P51	P50	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
PDR6	P67	P66	P65	P64	P63	P62	P61	P60	I/O ports
PDR7	P77	P76	P75	P74	P73	P72	P71	P70	
PDR8	P87	P86	P85	P84	P83	P82	P81	P80	
PDR9	—	—	—	—	P93	P92	P91	P90	
PDRA	—	—	—	—	PA3	PA2	PA1	PA0	
PDRB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
PDRC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	
PUCR1	—	PUCR16	PUCR15	PUCR14	PUCR13	PUCR12	PUCR11	PUCR10	
PUCR3	PUCR37	PUCR36	—	—	—	—	—	PUCR30	
PUCR5	PUCR57	PUCR56	PUCR55	PUCR54	PUCR53	PUCR52	PUCR51	PUCR50	
PUCR6	PUCR67	PUCR66	PUCR65	PUCR64	PUCR63	PUCR62	PUCR61	PUCR60	
PCR1	—	PCR16	PCR15	PCR14	PCR13	PCR12	PCR11	PCR10	
PCR3	PCR37	PCR36	—	—	—	PCR32	PCR31	PCR30	
PCR4	—	—	—	—	—	PCR42	PCR41	PCR40	
PCR5	PCR57	PCR56	PCR55	PCR54	PCR53	PCR52	PCR51	PCR50	
PCR6	PCR67	PCR66	PCR65	PCR64	PCR63	PCR62	PCR61	PCR60	
PCR7	PCR77	PCR76	PCR75	PCR74	PCR73	PCR72	PCR71	PCR70	
PCR8	PCR87	PCR86	PCR85	PCR84	PCR83	PCR82	PCR81	PCR80	
PCR9	—	—	—	—	PCR93	PCR92	PCR91	PCR90	
PCRA	—	—	—	—	PCRA3	PCRA2	PCRA1	PCRA0	
PCRC	PCRC7	PCRC6	PCRC5	PCRC4	PCRC3	PCRC2	PCRC1	PCRC0	
SYSCR1	SSBY	STS2	STS1	STS0	LSON	TMA3	MA1	MA0	SYSTEM
SYSCR2	—	—	—	NESEL	DTON	MSON	SA1	SA0	
IEGR	NMIEG	TMIFEG	ADTRGNEG	IEG4	IEG3	—	IEG1	IEG0	Interrupts
IENR1	IENRTC	—	IENWP	IEN4	IEN3	IENEC2	IEN1	IEN0	
IENR2	IENDT	IENAD	—	IENTG	IENTFH	IENTFL	IENTC	IENEC	
INTM	—	—	—	—	—	—	INTM1	INTM0	
IRR1	—	—	—	IRR4	IRR3	IRREC2	IRRI1	IRRI0	
IRR2	IRRDT	IRRAD	—	IRRTG	IRRTFH	IRRTFL	IRRTC	IRREC	
IWPR	IWPF7	IWPF6	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
CKSTPR1	S4CK STP*3	S31CK STP	S32CK STP	ADCKSTP	—	TFCKSTP	FROMCK STP*3	RTCCK STP	SYSTEM
CKSTPR2	ADBCK STP	TPUCK STP	IICCKSTP	PW2CK STP	AECCK STP	WDCK STP	PW1CK STP	LDCKSTP	
CKSTPR3	S33CK STP	TCCKSTP	TGCKSTP	PW4CK STP	PW3CK STP	—	—	—	

Notes: 1. AEC: Asynchronous event counter
2. WDT: Watchdog timer
3. This bit is available only for the flash memory version. In the masked ROM version, this bit is reserved.

24.3 Register States in Each Operating Mode

Register Abbreviation	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module
SCR4	Initialized	—	—	—	—	—	—	SCR4
SCSR4	Initialized	—	—	—	—	—	—	
TDR4	Initialized	—	—	—	—	—	—	
RDR4	Initialized	—	—	—	—	—	—	
FLMCR1	Initialized	—	—	—	—	—	Initialized	ROM
FLMCR2	Initialized	—	—	—	—	—	—	
FLPWCR	Initialized	—	—	—	—	—	—	
EBR1	Initialized	—	—	—	—	—	Initialized	
FENR	Initialized	—	—	—	—	—	—	
EBR2	Initialized	—	—	—	—	—	Initialized	
SYSCR3	Initialized	—	—	—	—	—	—	SYSTEM
TSTR	Initialized	—	—	—	—	—	—	TPU
TSYR	Initialized	—	—	—	—	—	—	
PDRE	Initialized	—	—	—	—	—	—	I/O ports
PDRF	Initialized	—	—	—	—	—	—	
PCRE	Initialized	—	—	—	—	—	—	
PCRF	Initialized	—	—	—	—	—	—	
PMRE	Initialized	—	—	—	—	—	—	
PMRF	Initialized	—	—	—	—	—	—	
TCR_1	Initialized	—	—	—	—	—	—	TPU_1
TMDR_1	Initialized	—	—	—	—	—	—	
TIOR_1	Initialized	—	—	—	—	—	—	
TIER_1	Initialized	—	—	—	—	—	—	
TSR_1	Initialized	—	—	—	—	—	—	
TCNT_1	Initialized	—	—	—	—	—	—	
TGRA_1	Initialized	—	—	—	—	—	—	
TGRB_1	Initialized	—	—	—	—	—	—	
PWCR3	Initialized	—	—	—	—	—	—	14-bit PWM_3
PWDR3	Initialized	—	—	—	—	—	—	

Register Abbreviation	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module
TCR_2	Initialized	—	—	—	—	—	—	TPU_2
TMDR_2	Initialized	—	—	—	—	—	—	
TIOR_2	Initialized	—	—	—	—	—	—	
TIER_2	Initialized	—	—	—	—	—	—	
TSR_2	Initialized	—	—	—	—	—	—	
TCNT_2	Initialized	—	—	—	—	—	—	
TGRA_2	Initialized	—	—	—	—	—	—	
TGRB_2	Initialized	—	—	—	—	—	—	
PWCR4	Initialized	—	—	—	—	—	—	14-bit PWM_4
PWDR4	Initialized	—	—	—	—	—	—	
RTCFLG	—	—	—	—	—	—	—	RTC
RSECDR	—	—	—	—	—	—	—	
RMINDR	—	—	—	—	—	—	—	
RHRDR	—	—	—	—	—	—	—	
RWKDR	—	—	—	—	—	—	—	
RTCCR1	—	—	—	—	—	—	—	
RTCCR2	—	—	—	—	—	—	—	
SUB32CR	Initialized	—	—	—	—	—	—	Clock pulse generator
RTCCSR	Initialized	—	—	—	—	—	—	RTC
ICCR1	Initialized	—	—	—	—	—	—	IIC2
ICCR2	Initialized	—	—	—	—	—	—	
ICMR	Initialized	—	—	—	—	—	—	
ICIER	Initialized	—	—	—	—	—	—	
ICSR	Initialized	—	—	—	—	—	—	
SAR	Initialized	—	—	—	—	—	—	
ICDRT	Initialized	—	—	—	—	—	—	
ICDRR	Initialized	—	—	—	—	—	—	
IPRA	Initialized	—	—	—	—	—	—	Interrupts
IPRB	Initialized	—	—	—	—	—	—	
IPRC	Initialized	—	—	—	—	—	—	

Register Abbreviation	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module
IPRD	Initialized	—	—	—	—	—	—	Interrupts
IPRE	Initialized	—	—	—	—	—	—	
IPRF	Initialized	—	—	—	—	—	—	
SMR3_3	Initialized	—	—	Initialized	—	—	Initialized	SCI3_3
BRR3_3	Initialized	—	—	Initialized	—	—	Initialized	
SCR3_3	Initialized	—	—	Initialized	—	—	Initialized	
TDR3_3	Initialized	—	—	Initialized	—	—	Initialized	
SSR3_3	Initialized	—	—	Initialized	—	—	Initialized	
RDR3_3	Initialized	—	—	Initialized	—	—	Initialized	
ABRKCR2	Initialized	—	—	—	—	—	—	Address break
ABRKSR2	Initialized	—	—	—	—	—	—	
BAR2H	Initialized	—	—	—	—	—	—	
BAR2L	Initialized	—	—	—	—	—	—	
BDR2H	—	—	—	—	—	—	—	
BDR2L	—	—	—	—	—	—	—	
BAR2E	Initialized	—	—	—	—	—	—	
TMG	Initialized	—	—	—	—	—	—	Timer G
ICRGF	Initialized	—	—	—	—	—	—	
ICRGR	Initialized	—	—	—	—	—	—	
ECPWCR	Initialized	—	—	—	—	—	—	AEC*[1]
ECPWDR	Initialized	—	—	—	—	—	—	
WEGR	Initialized	—	—	—	—	—	—	Interrupts
SPCR	Initialized	—	—	—	—	—	—	SCI3
AEGSR	Initialized	—	—	—	—	—	—	AEC*[1]
ECCR	Initialized	—	—	—	—	—	—	
ECCSR	Initialized	—	—	—	—	—	—	
ECH	Initialized	—	—	—	—	—	—	
ECL	Initialized	—	—	—	—	—	—	
SMR3_1	Initialized	—	—	Initialized	—	—	Initialized	SCI3_1
BRR3_1	Initialized	—	—	Initialized	—	—	Initialized	
SCR3_1	Initialized	—	—	Initialized	—	—	Initialized	

Register Abbreviation	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module
TDR3_1	Initialized	—	—	Initialized	—	—	Initialized	SCI3_1
SSR3_1	Initialized	—	—	Initialized	—	—	Initialized	
RDR3_1	Initialized	—	—	Initialized	—	—	Initialized	
SEMR	Initialized	—	—	Initialized	—	—	Initialized	
IrCR	Initialized	—	—	Initialized	—	—	Initialized	IrDA
SMR3_2	Initialized	—	—	Initialized	—	—	Initialized	SCI3_2
BRR3_2	Initialized	—	—	Initialized	—	—	Initialized	
SCR3_2	Initialized	—	—	Initialized	—	—	Initialized	
TDR3_2	Initialized	—	—	Initialized	—	—	Initialized	
SSR3_2	Initialized	—	—	Initialized	—	—	Initialized	
RDR3_2	Initialized	—	—	Initialized	—	—	Initialized	
TMWD	Initialized	—	—	—	—	—	—	WDT*[2]
TCSRWD1	Initialized	—	—	—	—	—	—	
TCSRWD2	Initialized	—	—	—	—	—	—	
TCWD	Initialized	—	—	—	—	—	—	
TMC	Initialized	—	—	—	—	—	—	Timer C
TCC/TLC	Initialized	—	—	—	—	—	—	
TCRF	Initialized	—	—	—	—	—	—	Timer F
TCSRF	Initialized	—	—	—	—	—	—	
TCFH	Initialized	—	—	—	—	—	—	
TCFL	Initialized	—	—	—	—	—	—	
OCRFH	Initialized	—	—	—	—	—	—	
OCRFL	Initialized	—	—	—	—	—	—	
ADRR	—	—	—	—	—	—	—	A/D converter
AMR	Initialized	—	—	—	—	—	—	
ADSR	Initialized	—	—	—	—	—	—	
PMR1	Initialized	—	—	—	—	—	—	I/O ports
OSCCR	Initialized	—	—	—	—	—	—	Clock pulse generator

Register Abbreviation	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module
PMR3	Initialized	—	—	—	—	—	—	I/O ports
PMR4	Initialized	—	—	—	—	—	—	
PMR5	Initialized	—	—	—	—	—	—	
PMR9	Initialized	—	—	—	—	—	—	
PMRB	Initialized	—	—	—	—	—	—	
PFCR	Initialized	—	—	—	—	—	—	
SPCR2	Initialized	—	—	—	—	—	—	SCI3
PWCR2	Initialized	—	—	—	—	—	—	14-bit PWM_2
PWDR2	Initialized	—	—	—	—	—	—	
PWCR1	Initialized	—	—	—	—	—	—	14-bit PWM_1
PWDR1	Initialized	—	—	—	—	—	—	
PDR1	Initialized	—	—	—	—	—	—	I/O ports
PDR3	Initialized	—	—	—	—	—	—	
PDR4	Initialized	—	—	—	—	—	—	
PDR5	Initialized	—	—	—	—	—	—	
PDR6	Initialized	—	—	—	—	—	—	
PDR7	Initialized	—	—	—	—	—	—	
PDR8	Initialized	—	—	—	—	—	—	
PDR9	Initialized	—	—	—	—	—	—	
PDRA	Initialized	—	—	—	—	—	—	
PDRB	Initialized	—	—	—	—	—	—	
PDRC	Initialized	—	—	—	—	—	—	
PUCR1	Initialized	—	—	—	—	—	—	
PUCR3	Initialized	—	—	—	—	—	—	
PUCR5	Initialized	—	—	—	—	—	—	
PUCR6	Initialized	—	—	—	—	—	—	
PCR1	Initialized	—	—	—	—	—	—	
PCR3	Initialized	—	—	—	—	—	—	
PCR4	Initialized	—	—	—	—	—	—	

Register Abbreviation	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module
PCR5	Initialized	—	—	—	—	—	—	I/O ports
PCR6	Initialized	—	—	—	—	—	—	
PCR7	Initialized	—	—	—	—	—	—	
PCR8	Initialized	—	—	—	—	—	—	
PCR9	Initialized	—	—	—	—	—	—	
PCRA	Initialized	—	—	—	—	—	—	
PCRC	Initialized	—	—	—	—	—	—	
SYSCR1	Initialized	—	—	—	—	—	—	SYSTEM
SYSCR2	Initialized	—	—	—	—	—	—	
IEGR	Initialized	—	—	—	—	—	—	Interrupts
IENR1	Initialized	—	—	—	—	—	—	
IENR2	Initialized	—	—	—	—	—	—	
INTM	Initialized	—	—	—	—	—	—	
IRR1	Initialized	—	—	—	—	—	—	
IRR2	Initialized	—	—	—	—	—	—	
IWPR	Initialized	—	—	—	—	—	—	
CKSTPR1	Initialized	—	—	—	—	—	—	SYSTEM
CKSTPR2	Initialized	—	—	—	—	—	—	
CKSTPR3	Initialized	—	—	—	—	—	—	

Notes: — is not initialized.

1. AEC: Asynchronous event counter
2. WDT: Watchdog timer

Section 25 Electrical Characteristics

25.1 Absolute Maximum Ratings for F-ZTAT Version

Table 25.1 lists the absolute maximum ratings.

Table 25.1 Absolute Maximum Ratings

Item		Symbol	Value	Unit	Note
Power supply voltage		V_{CC}	–0.3 to +4.3	V	*1
Analog power supply voltage		AV_{CC}	–0.3 to +4.3	V	
Input voltage	Other than port B	V_{in}	–0.3 to V_{CC} +0.3	V	
	Port B	AV_{in}	–0.3 to AV_{CC} +0.3	V	
Operating temperature		T_{opr}	–20 to +75 (general specifications)*2	°C	
			–40 to +85 (wide temperature range specifications)*2		
Storage temperature		T_{stg}	–55 to +125	°C	

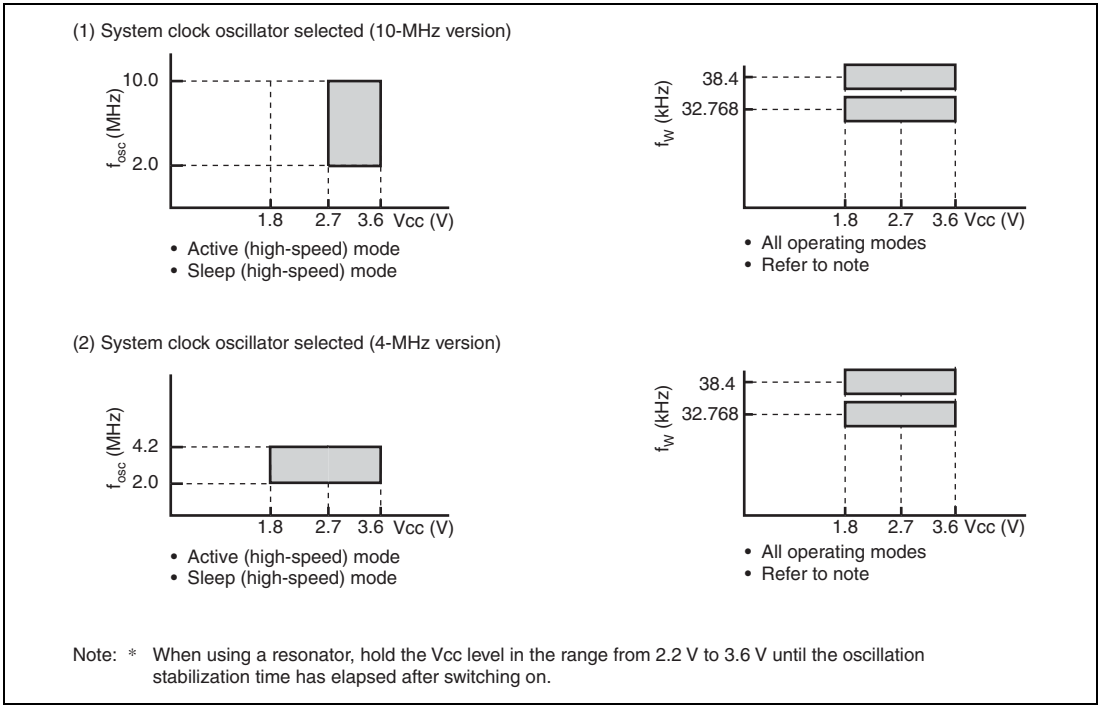
Notes: 1. Permanent damage may occur to the chip if absolute maximum ratings are exceeded. Normal operation should be under the conditions specified in Electrical Characteristics. Exceeding these values can result in incorrect operation and reduced reliability.

2. When the operating voltage (Vcc) for reading the flash memory is from 2.7 V to 3.6 V, the operating temperature (Ta) for programming/erasing ranges from –20 to +75°C. When the operating voltage (Vcc) for reading the flash memory is from 1.8 V to 3.6 V, the operating temperature (Ta) for programming/erasing ranges from –20 to +50°C.

25.2 Electrical Characteristics for F-ZTAT Version

25.2.1 Power Supply Voltage and Operating Range

The power supply voltage and operating range are indicated by the shaded region in the figures.

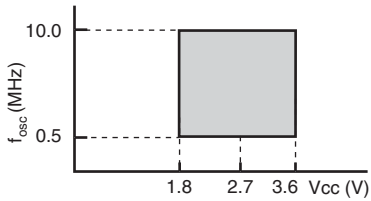


Note: * When using a resonator, hold the Vcc level in the range from 2.2 V to 3.6 V until the oscillation stabilization time has elapsed after switching on.

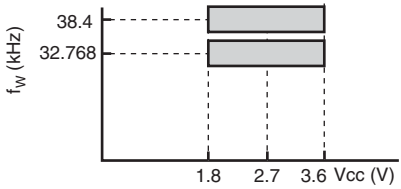
Figure 25.1 Power Supply Voltage and Oscillation Frequency Range (1)

(3) On-chip oscillator for system clock selected

R_{osc} used (reference value)



- Active (high-speed) mode
- Sleep (high-speed) mode



- All operating modes
- Refer to note

Note: * When using a resonator, hold the Vcc level in the range from 2.2 V to 3.6 V until the oscillation stabilization time has elapsed after switching on.

Figure 25.2 Power Supply Voltage and Oscillation Frequency Range (2)

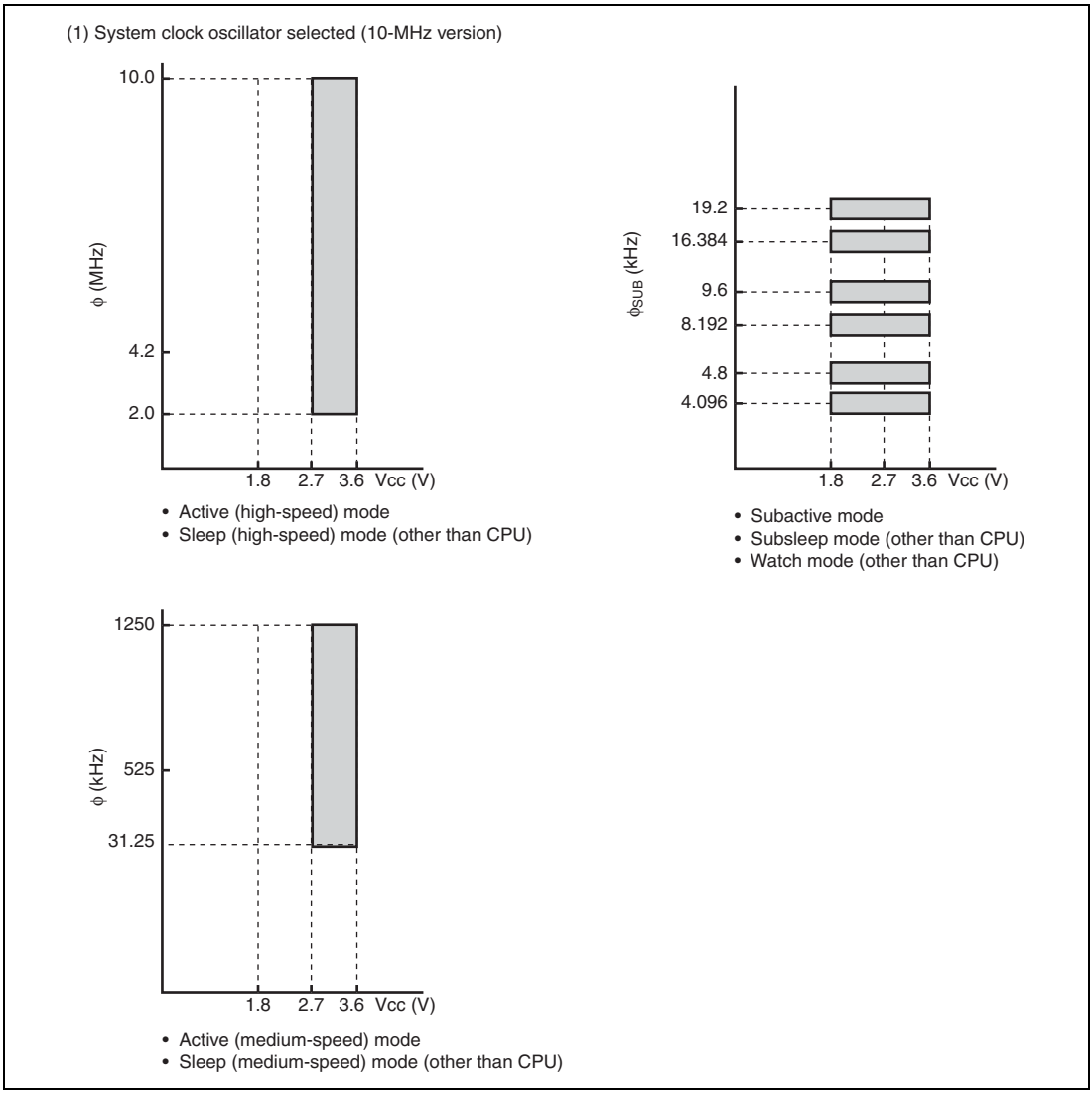
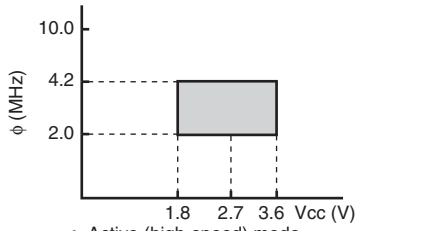
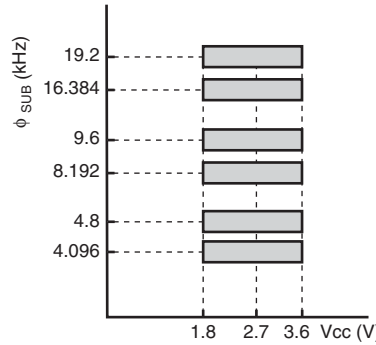


Figure 25.3 Power Supply Voltage and Operating Frequency Range (1)

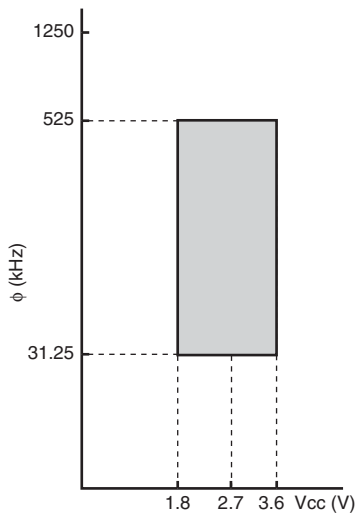
(2) System clock oscillator selected (4-MHz version)



- Active (high-speed) mode
- Sleep (high-speed) mode (other than CPU)



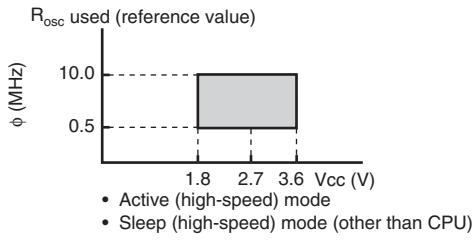
- Subactive mode
- Subsleep mode (other than CPU)
- Watch mode (other than CPU)



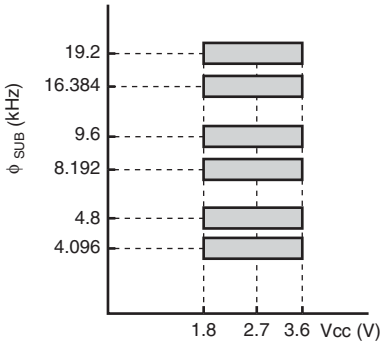
- Active (medium-speed) mode
- Sleep (medium-speed) mode (other than CPU)

Figure 25.4 Power Supply Voltage and Operating Frequency Range (2)

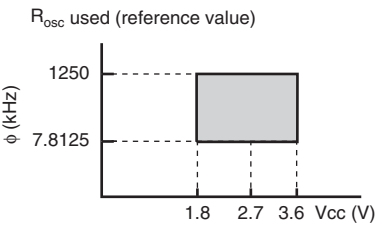
(3) On-chip oscillator for system clock selected



- Active (high-speed) mode
- Sleep (high-speed) mode (other than CPU)



- Subactive mode
- Subsleep mode (other than CPU)
- Watch mode (other than CPU)



- Active (medium-speed) mode
- Sleep (medium-speed) mode (other than CPU)

Figure 25.5 Power Supply Voltage and Operating Frequency Range (3)

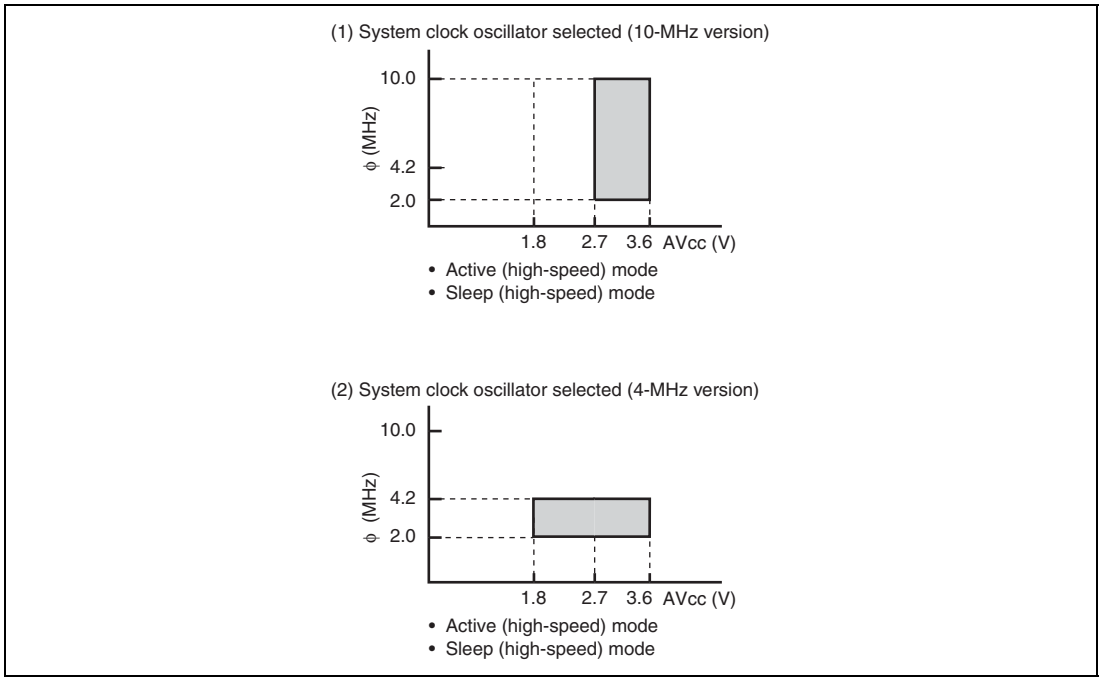


Figure 25.6 Analog Power Supply Voltage and Operating Frequency Range of A/D Converter (1)

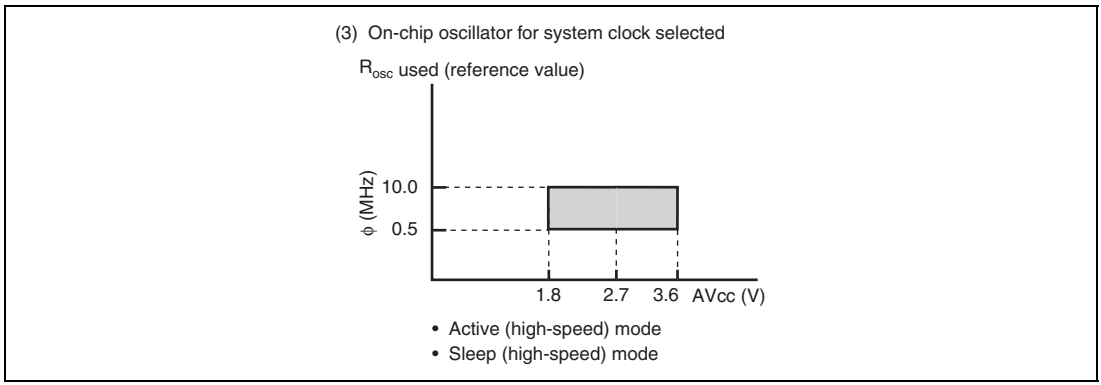


Figure 25.7 Analog Power Supply Voltage and Operating Frequency Range of A/D Converter (2)

25.2.2 DC Characteristics

Table 25.2 lists DC characteristics.

Table 25.2 DC Characteristics

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = AV_{SS} = 0.0 V, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Values Typ.	Values Max.	Unit	Note
Input high voltage	V_{IH}	$\overline{RES}$, TEST, $\overline{NMI}$*3, $\overline{WKP0}$ to $\overline{WKP7}$, $\overline{IRQ4}$, AEVL, AEVH, TMIC, TMIF, TMIG, $\overline{ADTRG}$, SCK33, SCK32, SCK31, SCK4		0.9V_{cc}	—	V_{cc} + 0.3	V	
		$\overline{IRQ0}$*5, $\overline{IRQ1}$*5, $\overline{IRQ3}$*5		0.9V_{cc}	—	AV_{cc} + 0.3		
		RXD33, RXD32, RXD31, IrRxD, UD		0.8V_{cc}	—	V_{cc} + 0.3	V	
		OSC1		0.9V_{cc}	—	V_{cc} + 0.3	V	
		X1		0.9V_{cc}	—	V_{cc} + 0.3	V	
		P10 to P16, P30 to P32, P36, P37, P40 to P42, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P93, PA0 to PA3, PC0 to PC7, PE0 to PE7, PF0 to PF3, TCLKA, TCLKB, TCLKC, TIOCA1, TIOCA2, TIOCB1, TIOCB2, SCL, SDA		0.8V_{cc}	—	V_{cc} + 0.3	V	
		PB0 to PB7		0.8V_{cc}	—	AV_{cc} + 0.3	V	
		IRQAEC		0.9V_{cc}	—	V_{cc} + 0.3	V	

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Note
Input low voltage	V_{IL}	$\overline{RES}$, TEST, $\overline{NMI}$*[3], $\overline{WKP0}$ to $\overline{WKP7}$, $\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{IRQ3}$, $\overline{IRQ4}$, IRQAEC, AEVL, AEVH, TMIC, TMIF, TMIG, $\overline{ADTRG}$, SCK33, SCK32, SCK31, SCK4		–0.3	—	$0.1V_{CC}$	V	
		RXD33, RXD32, RXD31, IrRXD, UD		–0.3	—	$0.2V_{CC}$	V	
		OSC1		–0.3	—	$0.1V_{CC}$	V	
		X1		–0.3	—	$0.1V_{CC}$	V	
		P10 to P16, P30 to P32, P36, P37, P40 to P42, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P93, PA0 to PA3, PB0 to PB7, PC0 to PC7, PE0 to PE7, PF0 to PF3, TCLKA, TCLKB, TCLKC, TIOCA1, TIOCB1, TIOCA2, TIOCB2, SCL, SDA		–0.3	—	$0.2V_{CC}$	V	

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Note
Output high voltage	V_{OH}	P10 to P16, P30 to P32, P36, P37, P40 to P42, P50 to P57, P60 to P67, P70 to P77, P80 to P87, PA0 to PA3, PC0 to PC7, PE0 to PE7, PF0 to PF3	$-I_{OH}$ = 1.0 mA V_{CC} = 2.7 to 3.6V	V_{CC} – 1.0	—	—	V	
			$-I_{OH}$ = 0.1 mA	V_{CC} – 0.3	—	—		
		P90 to P93	I_{OH} = 1.0mA V_{CC} = 2.7 to 3.6 V	V_{CC} – 1.0	—	—		
			I_{OH} = 0.1 mA	V_{CC} – 0.3	—	—		
Output low voltage	V_{OL}	P10 to P16, P30 to P32, P36, P37, P40 to P42, P50 to P57, P60 to P67, P70 to P77, P80 to P87, PA0 to PA3, PC0 to PC7, PE0 to PE7, PF0 to PF3	I_{OL} = 0.4 mA	—	—	0.5	V	
		P90 to P93	I_{OL} = 15 mA V_{CC}=2.7 to 3.6 V	—	—	1.0	V	
			I_{OL} = 10 mA V_{CC} = 2.2 to 3.6 V	—	—	0.5		
			I_{OL} = 8 mA V_{CC} = 1.8 to 3.6 V	—	—	0.5		
		SCL, SDA	I_{OL} = 3.0 mA V_{CC} = 2.0 to 3.6 V	—	—	0.4	V	
			I_{OL} = 3.0 mA V_{CC} = 1.8 to 2.0 V	—	—	$0.2V_{CC}$		

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Note
Input/output leakage current	$\lvert I_{IL} \rvert$	TEST, $\overline{NMI}$*3, OSC1, X1, P10 to P16, P30 to P32, P36, P37, P40 to P42, P50 to P57, P60 to P67, P70 to P77, P80 to P87, IRQAEC, PA0 to PA3, PC0 to PC7, PE0 to PE7, PF0 to PF3, P90 to P93	V_{IN} = 0.5 V to V_{CC} – 0.5 V	—	—	1.0	µA	
		PB0 to PB7	V_{IN} = 0.5 V to AV_{CC} – 0.5 V	—	—	1.0		
Pull-up MOS current	$-I_p$	P10 to P16, P30, P36, P37, P50 to P57, P60 to P67	V_{CC} = 3.0 V, V_{IN} = 0 V	30	—	180	µA	
Input capacitance	C_{IN}	All input pins except power supply pin	f = 1 MHz, V_{IN} = 0 V, Ta = 25°C	—	—	15.0	pF	

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Note
Active mode supply current	I_{OPE1}	V_{CC}	Active (high-speed) mode, V_{CC} = 1.8 V, f_{OSC} = 2 MHz	—	1.1	—	mA	Max. guideline = 1.1 × typ *1 *2 *4
			Active (high-speed) mode, V_{CC} = 3.0 V, f_{OSC} = Rosc	—	3.7	—		Max. guideline = 1.1 × typ *1 *2
			Active (high-speed) mode, V_{CC} = 3.0 V, f_{OSC} = 4 MHz	—	3.4	—		Max. guideline = 1.1 × typ *1 *2
			Active (high-speed) mode, V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	7.4	11.0		*1 *2
	I_{OPE2}	V_{CC}	Active (medium-speed) mode, V_{CC} = 1.8 V, f_{OSC} = 2 MHz, ϕ_{osc}/64	—	0.4	—	mA	Max. guideline = 1.1 × typ *1 *2 *5
			Active (medium-speed) mode, V_{CC} = 3.0 V, f_{OSC} = 4 MHz, ϕ_{osc}/64	—	0.7	—		Max. guideline = 1.1 × typ *1 *2
			Active (medium-speed) mode, V_{CC} = 3.0 V, f_{OSC} = 10 MHz, ϕ_{osc}/64	—	1.1	1.5		*1 *2
Sleep mode supply current	I_{SLEEP}	V_{CC}	V_{CC} = 1.8 V, f_{OSC} = 2 MHz	—	1.0	—	mA	Max. guideline = 1.1 × typ *1 *2 *4
			V_{CC} = 3.0 V, f_{OSC} = 4 MHz	—	2.5	—		Max. guideline = 1.1 × typ *1 *2
			V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	5.2	7.5		*1 *2
Subactive mode supply current	I_{SUB}	V_{CC}	V_{CC} = 2.7 V, 32-kHz crystal resonator used (ϕ_{SUB} = ϕ_w/8)	—	9.0	—	µA	*1 *2 Reference value
			V_{CC} = 2.7 V, 32-kHz crystal resonator used (ϕ_{SUB} = ϕ_w/2)	—	27.0	50.0		*1 *2

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Note
Subsleep mode supply current	I_{SUBSP}	V_{CC}	V_{CC} = 2.7 V, 32-kHz crystal resonator used ($\phi_{SUB} = \phi_W/2$)	—	5.5	9.0	µA	*1 *2
Watch mode supply current	I_{WATCH}	V_{CC}	V_{CC} = 1.8 V, Ta = 25°C 32-kHz crystal resonator used	—	0.5	—	µA	*1 *2 Reference value
			V_{CC} = 2.7 V, 32-kHz crystal resonator used	—	1.5	5.0		*1 *2
Standby mode supply current	I_{STBY}	V_{CC}	V_{CC} = 3.0 V, Ta = 25°C 32-kHz crystal resonator not used	—	0.1	—	µA	*1 *2 Reference value
			32-kHz crystal resonator not used	—	1.0	5.0		*1 *2
RAM data retaining voltage	V_{RAM}	V_{CC}		1.5	—	—	V	
Allowable output low current (per pin)	I_{OL}	Output pins except port 9		—	—	0.5	mA	
		P90 to P93		—	—	15.0		
Allowable output low current (total)	ΣI_{OL}	Output pins except port 9		—	—	20.0	mA	
		Port 9		—	—	60.0		
Allowable output high current (per pin)	$-I_{OH}$	All output pins	Vcc = 2.7 to 3.6 V	—	—	2.0	mA	
			Vcc = 1.8 to 3.6 V	—	—	0.2		
Allowable output high current (total)	$\Sigma - I_{OH}$	All output pins		—	—	10.0	mA	

Notes: 1. Pin states during current measurement.

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pins
Active (high-speed) mode (I_{OPE1})	V_{CC}	Only CPU operates On-chip WDT oscillator is off	V_{CC}	System clock oscillator: crystal resonator Subclock oscillator: Pin X1 = GND
Active (medium-speed) mode (I_{OPE2})				
Sleep mode	V_{CC}	Only on-chip timers operate On-chip WDT oscillator is off	V_{CC}	
Subactive mode	V_{CC}	Only CPU operates On-chip WDT oscillator is off	V_{CC}	System clock oscillator: crystal resonator Subclock oscillator: crystal resonator
Subsleep mode	V_{CC}	Only on-chip timers operate, CPU stops On-chip WDT oscillator is off	V_{CC}	
Watch mode	V_{CC}	Only time base operates, CPU stops On-chip WDT oscillator is off	V_{CC}	
Standby mode	V_{CC}	CPU and timers both stop On-chip WDT oscillator is off 32KSTOP = 1	V_{CC}	System clock oscillator: crystal resonator Subclock oscillator: crystal resonator

2. Excludes current in pull-up MOS transistors and output buffers.
3. Used for the determination of user mode or boot mode when the reset is released.
4. Only for 4-MHz version.
5. When IRQ0, IRQ1, and IRQ3 in PMRB are set to 0, and IRQ0, IRQ1, and IRQ3 in PMRE are set to 1, the maximum value is V_{CC} + 0.3 (V).

25.2.3 AC Characteristics

Table 25.3 lists the control signal timing, table 25.4 lists the serial interface timing, and table 25.5 lists the I^2C bus interface timing.

Table 25.3 Control Signal Timing

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = AV_{SS} = 0.0 V, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
System clock oscillation frequency	f_{OSC}	OSC1, OSC2	V_{CC} = 2.7 to 3.6 V (10-MHz version)	2.0	—	10.0	MHz	
			V_{CC} = 1.8 to 3.6 V (4-MHz version)	2.0	—	4.2		
System clock on-chip oscillation frequency	f_{Rosc}		On-chip oscillator for system clock selected V_{CC} = 1.8 to 3.6 V	0.5	—	10.0		*3
OSC clock (ϕ_{OSC}) cycle time	t_{OSC}	OSC1, OSC2	V_{CC} = 2.7 to 3.6 V (10-MHz version)	100	—	500	ns	Figure 25.14
			V_{CC} = 1.8 to 3.6 V (4-MHz version)	238	—	500		
System clock on-chip oscillation clock (ϕ_{ROSC}) cycle time	t_{ROSC}		On-chip oscillator for system clock selected V_{CC} = 1.8 to 3.6 V	100	—	2000		*3
System clock (ϕ) cycle time	t_{cyc}			1	—	64	t_{OSC}	
				—	—	32	µs	

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Reference Figure
Subclock oscillation frequency	f_w	X1, X2		—	32.768 or 38.4	—	kHz	
Watch clock (ϕ_w) cycle time	t_w	X1, X2		—	30.5 or 26.0	—	µs	Figure 25.14
Subclock (ϕ_{SUB}) cycle time	t_{subcyc}			2	—	8	t_w	*1
Instruction cycle time				2	—	—	t_{cyc} t_{subcyc}	
Oscillation stabilization time	t_{rc}	OSC1, OSC2	Ceramic resonator (Vcc = 2.2 to 3.6 V)	—	20	45	µs	Figure 25.23
			Ceramic resonator (other than above)	—	80	—		
			Crystal resonator (V_{CC} = 2.7 to 3.6 V)	—	0.8	2.0	ms	
			Crystal resonator (V_{CC} = 2.2 to 3.6 V)	—	1.2	3.0		
			Other than above	—	—	50	ms	
		On-chip oscillator for system clock	At switching on	—	—	25	µs	*3
		X1, X2	V_{CC} = 2.2 to 3.6 V	—	—	2	s	Figure 5.5
			Other than above	—	4	—		
External clock high width	t_{CPH}	OSC1	V_{CC} = 2.7 to 3.6 V (10-MHz version)	40	—	—	ns	Figure 25.14
			V_{CC} = 1.8 to 3.6 V (4-MHz version)	95	—	—		
		X1		—	15.26 or 13.02	—	µs	

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Reference Figure
External clock low width	t_{CPL}	OSC1	V_{CC} = 2.7 to 3.6 V (10-MHz version)	40	—	—	ns	Figure 25.14
			V_{CC} = 1.8 to 3.6 V (4-MHz version)	95	—	—		
		X1		—	15.26 or 13.02	—	µs	
External clock rising time	t_{CPr}	OSC1	V_{CC} = 2.7 to 3.6 V (10-MHz version)	—	—	10	ns	Figure 25.14
			V_{CC} = 1.8 to 3.6 V (4-MHz version)	—	—	24		
		X1		—	—	55.0		
External clock falling time	t_{CPf}	OSC1	V_{CC} = 2.7 to 3.6 V (10-MHz version)	—	—	10	ns	Figure 25.14
			V_{CC} = 1.8 to 3.6 V (4-MHz version)	—	—	24		
		X1		—	—	55.0		
$\overline{RES}$ pin low width	t_{REL}	$\overline{RES}$		10	—	—	t_{cyc}	Figure 25.15 *2

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Reference Figure
Input pin high width	t_{IH}	$\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{NMI}$, $\overline{IRQ3}$, $\overline{IRQ4}$, IRQAEC, $\overline{WKP0}$ to $\overline{WKP7}$, TMIC, TMIF, TMIG, $\overline{ADTRG}$		2	—	—	t_{cyc} t_{subcyc}	Figure 25.16
		AEVL, AEVH	V_{CC} = 2.7 to 3.6 V (10-MHz version)	50	—	—	ns	
			V_{CC} = 1.8 to 3.6 V (4-MHz version)	110	—	—		
	t_{TCKWH}	TCLKA, TCLKB, TCLKC, TIOCA1, TIOCB1, TIOCA2, TIOCB2	Single edge specified	1.5	—	—	t_{cyc}	Figure 25.19
			Both edges specified	2.5	—	—		
Input pin low width	t_{IL}	$\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{NMI}$, $\overline{IRQ3}$, $\overline{IRQ4}$, IRQAEC, $\overline{WKP0}$ to $\overline{WKP7}$, TMIC, TMIF, TMIG, $\overline{ADTRG}$		2	—	—	t_{cyc} t_{subcyc}	Figure 25.16
		AEVL, AEVH	V_{CC} = 2.7 to 3.6 V (10-MHz version)	50	—	—	ns	
			V_{CC} = 1.8 to 3.6 V (4-MHz version)	110	—	—		
	t_{TCKWL}	TCLKA, TCLKB, TCLKC, TIOCA1, TIOCB1, TIOCA2, TIOCB2	Single edge specified	1.5	—	—	t_{cyc}	Figure 25.19
			Both edges specified	2.5	—	—		

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Reference Figure
UD pin minimum transition width	UD			4	—	—	t_{cyc} t_{subcyc}	Figure 25.21

Notes: 1. Selected with the SA1 and SA0 bits in the system control register 2 (SYSCR2).
2. For details on the power-on reset characteristics, refer to table 25.7 and figure 25.15.
3. The specifications may vary due to the effects of temperature, power-supply voltage, and dispersion of product lots. Thorough evaluation under the actual conditions of use is essential in the design of systems. As for actual specification, please confirm with our sales representatives.

Table 25.4 Serial Interface Timing

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = AV_{SS} = 0.0 V, unless otherwise specified.

Item		Symbol	Test Condition	Values Min.	Typ.	Max.	Unit	Reference Figure
Input clock cycle	Asynchronous	t_{scyc}		4	—	—	t_{cyc} or t_{subcyc}	Figure 25.17
	Clock synchronous			6	—	—		
Input clock pulse width		t_{SCKW}		0.4	—	0.6	t_{scyc}	Figure 25.17
Transmit data delay time (clock synchronous)		t_{TXD}		—	—	1	t_{cyc} or t_{subcyc}	Figure 25.18
Receive data setup time (clock synchronous)		t_{RXS}	V_{CC} = 2.7 to 3.6 V	100	—	—	ns	Figure 25.18
			Other than above	238				
Receive data hold time (clock synchronous)		t_{RXH}	V_{CC} = 2.7 to 3.6 V	100	—	—	ns	Figure 25.18
			Other than above	238				

Table 25.5 I^2C Bus Interface Timing

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, Ta = –20 to +75°C, unless otherwise specified.

Item	Symbol	Test Condition	Values			Unit	Reference Figure
			Min.	Typ.	Max.		
SCL input cycle time	t_{SCL}		$12t_{cyc}$ + 600	—	—	ns	Figure 25.20
SCL input high width	t_{SCLH}		$3t_{cyc}$ + 300	—	—	ns	
SCL input low width	t_{SCLL}		$5t_{cyc}$ + 300	—	—	ns	
Falling time for SCL and SDA inputs	t_{Sf}		—	—	300	ns	
Pulse width of spike on SCL and SDA to be suppressed	t_{SP}		—	—	$1t_{cyc}$	ns	
SDA input bus-free time	t_{BUF}		$5t_{cyc}$	—	—	ns	
Start condition input hold time	t_{STAH}		$3t_{cyc}$	—	—	ns	
Repeated start condition input setup time	t_{STAS}		$3t_{cyc}$	—	—	ns	
Stop condition input setup time	t_{STOS}		$3t_{cyc}$	—	—	ns	
Data-input setup time	t_{SDAS}		$1t_{cyc}$ + 20	—	—	ns	
Data-input hold time	t_{SDAH}		0	—	—	ns	
Capacitive load of SCL and SDA	Cb		0	—	400	pF	
Falling time of SCL and SDA output	t_{Sf}		—	—	300	ns	

25.2.4 A/D Converter Characteristics

Table 25.6 lists the A/D converter characteristics.

Table 25.6 A/D Converter Characteristics

V_{CC} = 1.8 V to 3.6 V, V_{SS} = AV_{SS} = 0.0 V, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Notes
				Min.	Typ.	Max.		
Analog power supply voltage	AV_{CC}	AV_{CC}		1.8	—	3.6	V	*1
Analog input voltage	AV_{IN}	AN0 to AN7		–0.3	—	AV_{CC} + 0.3	V	
Analog power supply current	AI_{OPE}	AV_{CC}	AV_{CC} = 3.0 V	—	—	1.0	mA	
	AI_{STOP1}	AV_{CC}		—	600	—	µA	*2 Reference value
	AI_{STOP2}	AV_{CC}		—	—	5	µA	*3
Analog input capacitance	C_{AIN}	AN0 to AN7		—	—	15.0	pF	
Allowable signal source impedance	R_{AIN}			—	—	10.0	kΩ	
Resolution (data length)				—	—	10	bits	
Nonlinearity error			AV_{CC} = 2.7 to 3.6 V V_{CC} = 2.7 to 3.6 V	—	—	±3.5	LSB	
			AV_{CC} = 2.0 to 3.6 V V_{CC} = 2.0 to 3.6 V	—	—	±5.5		
			Other than above	—	—	±7.5		*4
Quantization error				—	—	±0.5	LSB	
Absolute accuracy			AV_{CC} = 2.7 to 3.6 V V_{CC} = 2.7 to 3.6 V	—	—	±4.0	LSB	
			AV_{CC} = 2.0 to 3.6 V V_{CC} = 2.0 to 3.6 V	—	—	±6.0		
			Other than above	—	—	±8.0		*4

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Notes
Conversion time			AV_{CC} = 2.7 to 3.6 V V_{CC} = 2.7 to 3.6 V	12.4	—	124	µs	System clock oscillator selected
				7.8	15.5	31	µs	On-chip oscillator for system clock selected Reference value (f_{Rosc} = 4 MHz)
			Other than AV_{CC} = 2.7 to 3.6 V V_{CC} = 2.7 to 3.6 V	29.5	—	124	µs	System clock oscillator selected
				31	62	124	µs	On-chip oscillator for system clock selected Reference value (f_{Rosc} = 1 MHz)

Notes: 1. Connect AV_{CC} to V_{CC} when the A/D converter is not used.
2. AI_{STOP1} is the current flowing through the ladder resistor while the A/D converter is idle.
3. AI_{STOP2} is the current at a reset, in standby or watch mode the A/D converter is idle, or in the module standby state.
4. Conversion time is 29.5 µs.

25.2.5 Power-On Reset Circuit Characteristics

Table 25.7 Power-On Reset Circuit Characteristics

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = AV_{SS} = 0.0 V,
Ta = −20 to +75°C (regular specifications), Ta = −40 to +85°C (wide-range specifications), unless otherwise specified.

Item	Symbol	Test Condition	Values			Unit	Reference Figure
			Min.	Typ.	Max.		
Reset voltage	V_rst		$0.7V_{CC}$	$0.8V_{CC}$	$0.9V_{CC}$	V	Figure 22.2
Power supply rise time	t_vtr		The Vcc rise time should be shorter than half the $\overline{RES}$ rise time.				
Reset count time	t_out		0.8	—	4.0	μs	
			0.8	—	16.0	μs	On-chip oscillator for system clock selected (Reference value)
Reset count time	t_cr		Adjustable by the value of the external capacitor of the $\overline{RES}$ pin.				
On-chip pull-up resistance	R_P	Vcc = 3.0 V	60	100	—	kΩ	Figure 22.1

25.2.6 Watchdog Timer Characteristics

Table 25.8 Watchdog Timer Characteristics

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = AV_{SS} = 0.0 V,
Ta = –20 to +75°C (regular specifications), Ta = –40 to +85°C (wide-range specifications),
unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Notes
				Min.	Typ.	Max.		
On-chip watchdog timer oscillator overflow time	t_{ovf}			0.2	0.4	—	s	*

Note: * Indicates the period from the start of counting at 0 to the generation of an internal reset when the counter reaches 255 in the case where the on-chip watchdog timer oscillator is selected.

25.2.7 Flash Memory Characteristics

Table 25.9 lists the flash memory characteristics.

Table 25.9 Flash Memory Characteristics

Condition A:
AV_{CC} = 2.7 V to 3.6 V, V_{SS} = AV_{SS} = 0.0 V, V_{CC} = 2.7 V to 3.6 V (operating voltage range in reading), V_{CC} = 3.0 V to 3.6 V (operating voltage range in programming/erasing),
Ta = –20 to +75°C (operating temperature range in programming/erasing: regular specifications, wide-range specifications)
Condition B:
AV_{CC} = 1.8 V to 3.6 V, V_{SS} = AV_{SS} = 0.0 V, V_{CC} = 1.8 V to 3.6 V (operating voltage range in reading), V_{CC} = 3.0 V to 3.6 V (operating voltage range in programming/erasing),
Ta = –20 to +50°C (operating temperature range in programming/erasing: regular specifications)

				Values			
Item		**Symbol**	**Test Condition**	**Min.**	**Typ.**	**Max.**	**Unit**
Programming time (per 128 bytes)*1*2*4		t_P		—	7	200	ms
Erasing time (per block)*1*3*6		t_E		—	100	1200	ms
Maximum programming count		N_{WEC}		1000*8*11	10000*9	—	Times
				100*8*12	10000*9	—	
Data retention time		t_{DRP}		10*10	—	—	Years
Programming	Wait time after setting SWE bit*1	x		1	—	—	µs
	Wait time after setting PSU bit*1	y		50	—	—	µs
	Wait time after setting P bit*1*4	z1	1 ≤ n ≤ 6	28	30	32	µs
		z2	7 ≤ n ≤ 1000	198	200	202	µs
		z3	Additional-programming	8	10	12	µs
	Wait time after clearing P bit*1	α		5	—	—	µs
	Wait time after clearing PSU bit*1	β		5	—	—	µs
	Wait time after setting PV bit*1	γ		4	—	—	µs

Item		Symbol	Test Condition	Values Min.	Typ.	Max.	Unit
Programming	Wait time after dummy write*[1]	ε		2	—	—	µs
	Wait time after clearing PV bit*[1]	η		2	—	—	µs
	Wait time after clearing SWE bit*[1]	θ		100	—	—	µs
	Maximum programming count*[1]*[4]*[5]	N		—	—	1000	Times
Erase	Wait time after setting SWE bit*[1]	x		1	—	—	µs
	Wait time after setting ESU bit*[1]	y		100	—	—	µs
	Wait time after setting E bit*[1]*[6]	z		10	—	100	ms
	Wait time after clearing E bit*[1]	α		10	—	—	µs
	Wait time after clearing ESU bit*[1]	β		10	—	—	µs
	Wait time after setting EV bit*[1]	γ		20	—	—	µs
	Wait time after dummy write*[1]	ε		2	—	—	µs
	Wait time after clearing EV bit*[1]	η		4	—	—	µs
	Wait time after clearing SWE bit*[1]	θ		100	—	—	µs
	Maximum erasing count*[1]*[6]*[7]	N		—	—	120	Times

Notes: 1. Make the time settings in accordance with the programming/erasing algorithms.
2. The programming time for 128 bytes. (Indicates the total time for which the P bit in the flash memory control register 1 (FLMCR1) is set. The programming-verifying time is not included.)
3. The time required to erase one block. (Indicates the total time for which the E bit in the flash memory control register 1 (FLMCR1) is set. The erasing-verifying time is not included.)
4. Programming time maximum value (t_p (max.)) = wait time after setting P bit (z) × maximum programming count (N)

5. Set the maximum programming count (N) according to the actual set values of z1, z2, and z3, so that it does not exceed the programming time maximum value (t_P (max.)). The wait time after setting P bit (z1, z2) should be changed as follows according to the 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. Erasing time maximum value (t_E (max.)) = wait time after setting E bit (z) × maximum erasing count (N)
7. Set the maximum erasing count (N) according to the actual set value of (z), so that it does not exceed the erasing time maximum value (t_E (max.)).
8. The minimum number of times in which all characteristics are guaranteed following reprogramming. (The guarantee covers the range from 1 to the minimum value.)
9. Reference value at 25°C. (Guideline showing programming count over which functioning will be retained under normal circumstances.)
10. Data retention characteristics within the range indicated in the specifications, including the minimum programming count.
11. Applies to an operating voltage range when reading data of 2.7 to 3.6 V.
12. Applies to an operating voltage range when reading data of 1.8 to 3.6 V.

25.3 Absolute Maximum Ratings for Masked ROM Version

Table 25.10 lists the absolute maximum ratings.

Table 25.10 Absolute Maximum Ratings

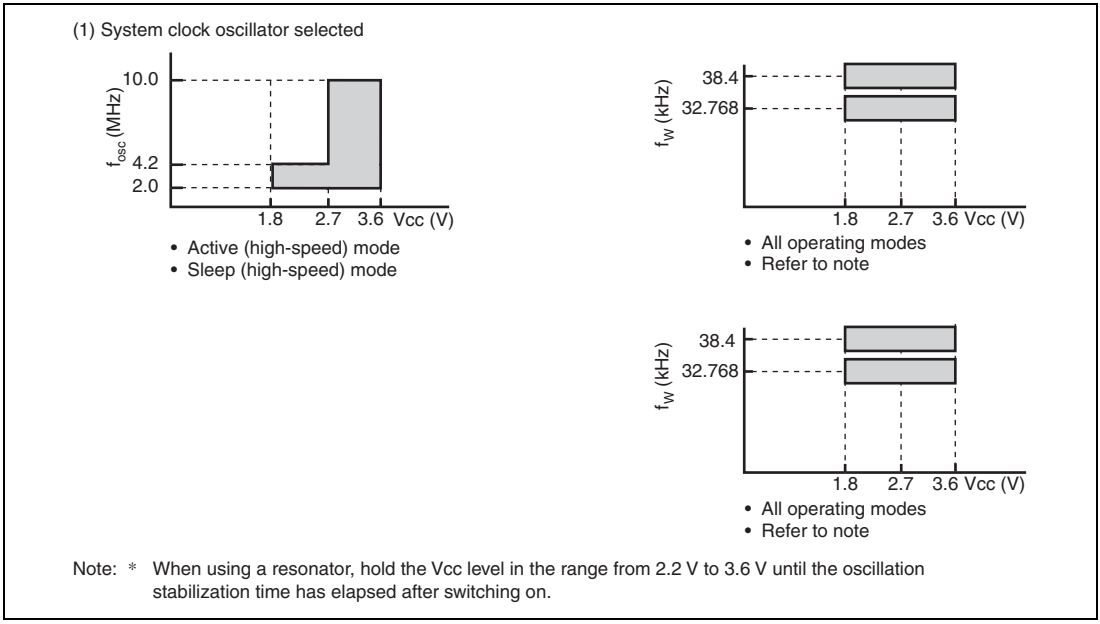
Item		Symbol	Value	Unit	Note
Power supply voltage		V_{CC}	–0.3 to +4.3	V	*
Analog power supply voltage		AV_{CC}	–0.3 to +4.3	V	
Input voltage	Other than port B	V_{in}	–0.3 to V_{CC} +0.3	V	
	Port B	AV_{in}	–0.3 to AV_{CC} +0.3	V	
Operating temperature		T_{opr}	–20 to +75 (regular specifications)	°C	
			–40 to +85 (wide-range specifications)		
Storage temperature		T_{stg}	–55 to +125	°C	

Note: * Permanent damage may occur to the chip if absolute maximum ratings are exceeded. Normal operation should be under the conditions specified in Electrical Characteristics. Exceeding these values can result in incorrect operation and reduced reliability.

25.4 Electrical Characteristics for Masked ROM Version

25.4.1 Power Supply Voltage and Operating Range

The power supply voltage and operating range are indicated by the shaded region in the figures.

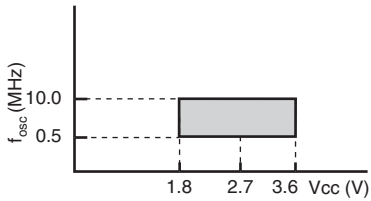


Note: * When using a resonator, hold the Vcc level in the range from 2.2 V to 3.6 V until the oscillation stabilization time has elapsed after switching on.

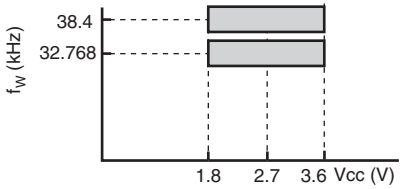
Figure 25.8 Power Supply Voltage and Oscillation Frequency Range (1)

(2) On-chip oscillator for system clock selected

R_{osc} used (reference value)



- Active (high-speed) mode
- Sleep (high-speed) mode



- All operating modes
- Refer to note

Note: * When using a resonator, hold the Vcc level in the range from 2.2 V to 3.6 V until the oscillation stabilization time has elapsed after switching on.

Figure 25.9 Power Supply Voltage and Oscillation Frequency Range (2)

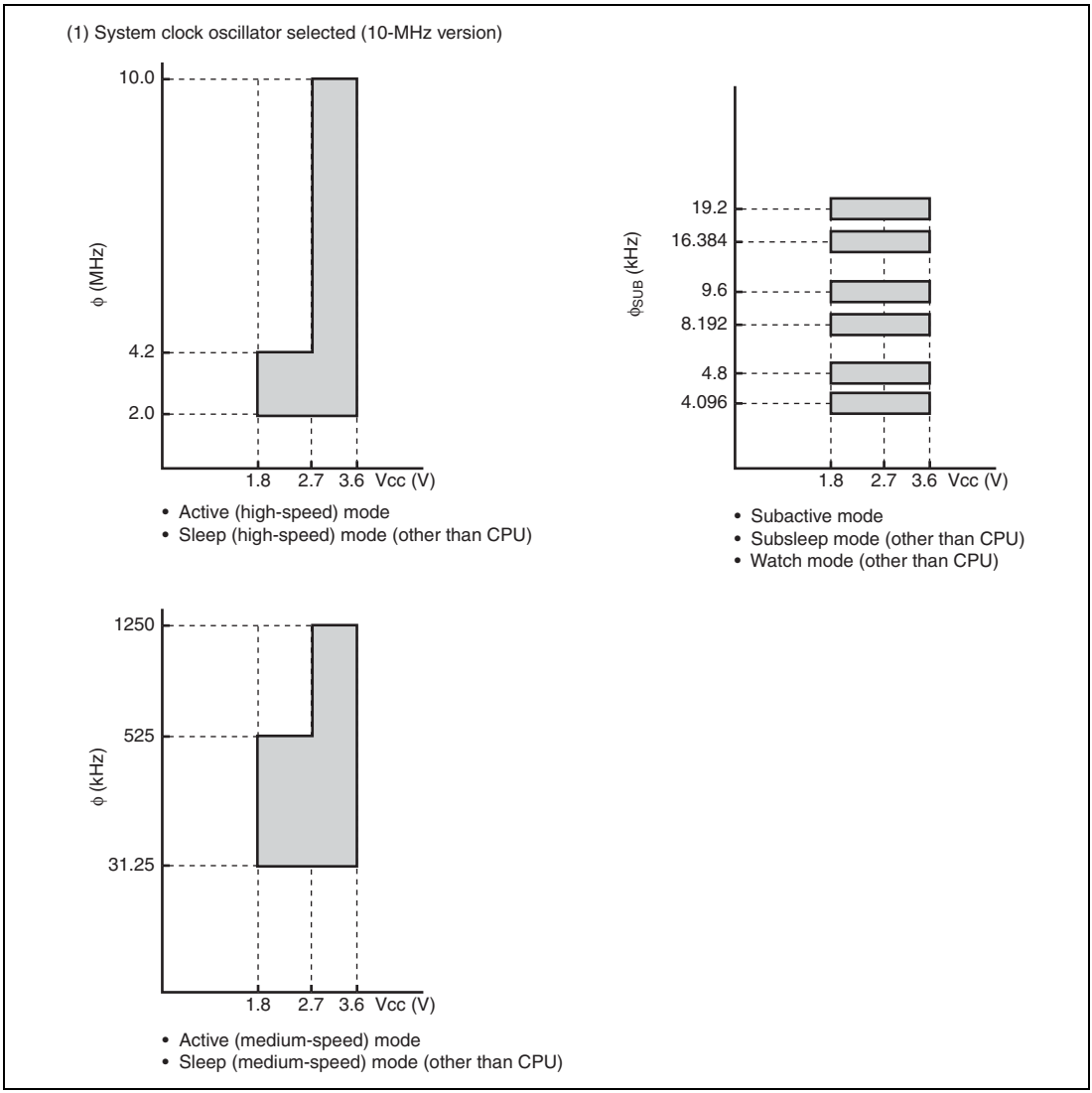


Figure 25.10 Power Supply Voltage and Operating Frequency Range (1)

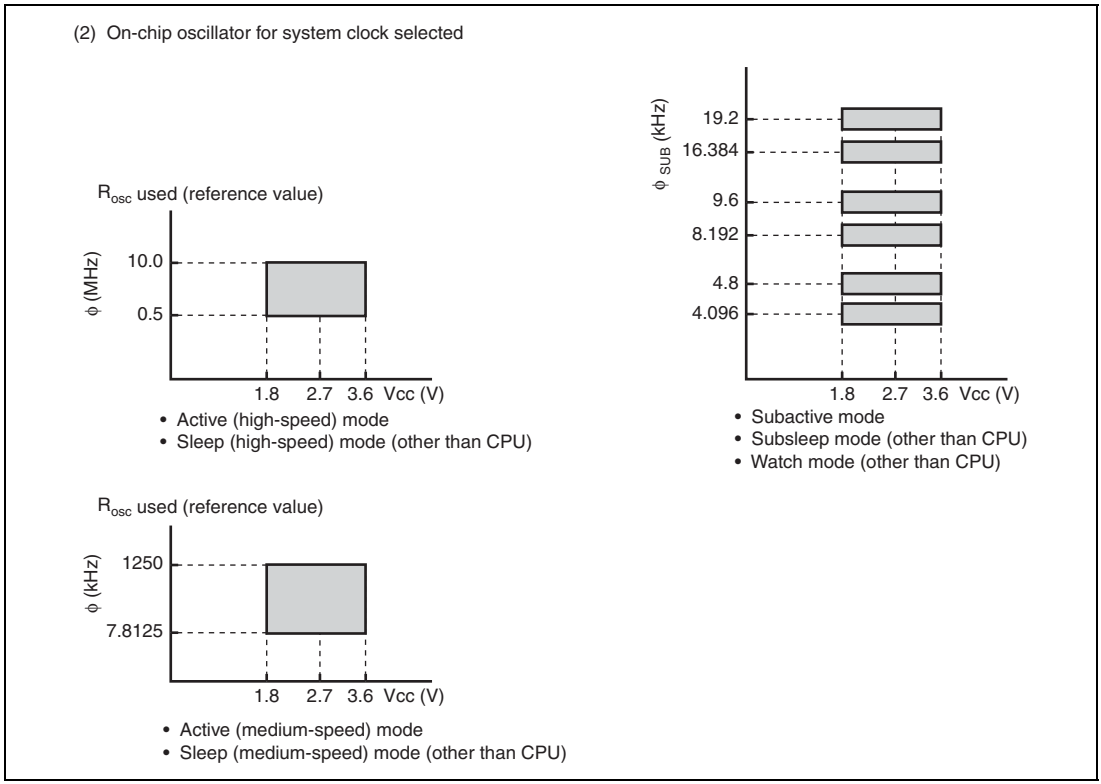


Figure 25.11 Power Supply Voltage and Operating Frequency Range (2)

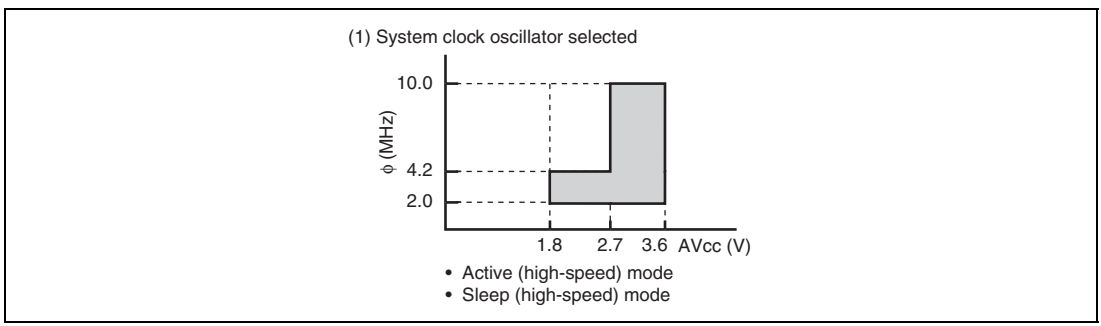


Figure 25.12 Analog Power Supply Voltage and Operating Frequency Range of A/D Converter (1)

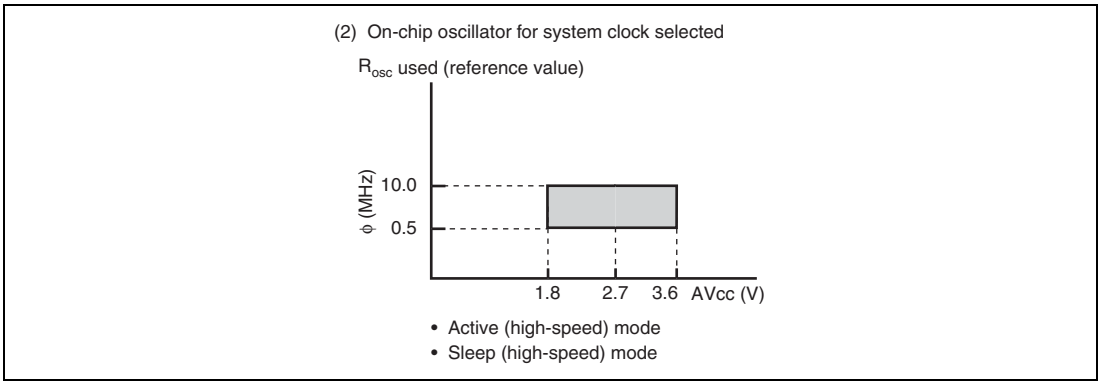


Figure 25.13 Analog Power Supply Voltage and Operating Frequency Range of A/D Converter (2)

25.4.2 DC Characteristics

Table 25.11 lists the DC characteristics.

Table 25.11 DC Characteristics

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = AV_{SS} = 0.0 V, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Values Typ.	Values Max.	Unit	Note
Input high voltage	V_{IH}	$\overline{RES}$, TEST, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP7}$, $\overline{IRQ4}$, AEVL, AEVH, TMIC, TMIF, TMIG, $\overline{ADTRG}$, SCK33, SCK32, SCK31		0.9V_{cc}	—	V_{cc} + 0.3	V	
		$\overline{IRQ0}$*3, $\overline{IRQ1}$*3, $\overline{IRQ3}$*3		0.9V_{cc}	—	AV_{cc} + 0.3		
		RXD33, RXD32, RXD31, IrRXD, UD		0.8V_{cc}	—	V_{cc} + 0.3	V	
		OSC1		0.9V_{cc}	—	V_{cc} + 0.3	V	
		X1		0.9V_{cc}	—	V_{cc} + 0.3	V	
		P10 to P16, P30 to P32, P36, P37, P40 to P42, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P93, PA0 to PA3, PC0 to PC7, PE0 to PE7, PF0 to PF3, TCLKA, TCLKB, TCLKC, TIOCA1, TIOCB1, TIOCA2, TIOCB2, SCL, SDA		0.8V_{cc}	—	V_{cc} + 0.3	V	
		PB0 to PB7		0.8V_{cc}	—	AV_{cc} + 0.3	V	
		IRQAEC		0.9V_{cc}	—	V_{cc} + 0.3	V	

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Note
Input low voltage	V_{IL}	$\overline{RES}$, TEST, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP7}$, $\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{IRQ3}$, $\overline{IRQ4}$, IRQAEC, AEVL, AEVH, TMIC, TMIF, TMIG, $\overline{ADTRG}$, SCK33, SCK32, SCK31		–0.3	—	$0.1V_{CC}$	V	
		RXD33, RXD32, RXD31, IrRXD, UD		–0.3	—	$0.2V_{CC}$	V	
		OSC1		–0.3	—	$0.1V_{CC}$	V	
		X1		–0.3	—	$0.1V_{CC}$	V	
		P10 to P16, P30 to P32 P36, P37, P40 to P42, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P93, PA0 to PA3, PB0 to PB7, PC0 to PC7, PE0 to PE7, PF0 to PF3, TCLKA, TCLKB, TCLKC, TIOCA1, TIOCB1, TIOCA2, TIOCB2, SCL, SDA		–0.3	—	$0.2V_{CC}$	V	

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Note
Output high voltage	V_{OH}	P10 to P16, P30 to P32, P36, P37, P40 to P42, P50 to P57, P60 to P67, P70 to P77, P80 to P87, PA0 to PA3, PC0 to PC7, PE0 to PE7, PF0 to PF3	$-I_{OH}$ = 1.0 mA V_{CC} = 2.7 to 3.6V	V_{CC} – 1.0	—	—	V	
			$-I_{OH}$ = 0.1 mA	V_{CC} – 0.3	—	—		
		P90 to P93	$-I_{OH}$ = 1.0 mA V_{CC} = 2.7 to 3.6V	V_{CC} – 1.0	—	—		
			$-I_{OH}$ = 0.1 mA	V_{CC} – 0.3	—	—		
Output low voltage	V_{OL}	P10 to P16, P30 to P32, P36, P37, P40 to P42, P50 to P57, P60 to P67, P70 to P77, P80 to P87, PA0 to PA3, PC0 to PC7, PE0 to PE7, PF0 to PF3	I_{OL} = 0.4 mA	—	—	0.5	V	
		P90 to P93	I_{OL} = 15 mA V_{CC} = 2.7 to 3.6V	—	—	1.0	V	
			I_{OL} = 10 mA V_{CC} = 2.2 to 3.6V	—	—	0.5		
			I_{OL} = 8 mA V_{CC} = 1.8 to 3.6V	—	—	0.5		
		SCL, SDA	I_{OL} = 3.0 mA V_{CC} = 2.0 to 3.6V	—	—	0.4	V	
			I_{OL} = 3.0 mA V_{CC} = 1.8 to 2.0V	—	—	$0.2V_{CC}$		

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Note
Input/output leakage current	$\lvert I_{IL} \rvert$	TEST, $\overline{NMI}$, OSC1, X1, P10 to P16, P30 to P32, P36, P37, P40 to P42, P50 to P57, P60 to P67, P70 to P77, P80 to P87, IRQAEC, PA0 to PA3, PC0 to PC7, PE0 to PE7, PF0 to PF3, P90 to P93	V_{IN} = 0.5 V to V_{CC} – 0.5 V	—	—	1.0	µA	
		PB0 to PB7	V_{IN} = 0.5 V to AV_{CC} – 0.5 V	—	—	1.0		
Pull-up MOS current	$-I_p$	P10 to P16, P30, P36, P37, P50 to P57, P60 to P67	V_{CC} = 3 V, V_{IN} = 0 V	30	—	180	µA	
Input capacitance	C_{IN}	All input pins except power supply pin	f = 1 MHz, V_{IN} = 0 V, Ta = 25°C	—	—	15.0	pF	

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Note
Active mode supply current	I_{OPE1}	V_{CC}	Active (high-speed) mode, V_{CC} = 1.8 V, f_{OSC} = 2 MHz	—	0.7	—	mA	Max. guideline = 1.1 × typ.*1 *2
			Active (high-speed) mode, V_{CC} = 3.0 V, f_{OSC} = R_{OSC}	—	2.2	—		Max. guideline = 1.1 × typ.*1 *2
			Active (high-speed) mode, V_{CC} = 3.0 V, f_{OSC} = 4 MHz	—	2.6	—		Max. guideline = 1.1 × typ.*1 *2
			Active (high-speed) mode, V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	6.0	9.0		*1 *2
	I_{OPE2}	V_{CC}	Active (medium-speed) mode, V_{CC} = 1.8 V, f_{OSC} = 2 MHz, ϕ_{osc}/64	—	0.1	—	mA	Max. guideline = 1.1 × typ.*1 *2
			Active (medium-speed) mode, V_{CC} = 3.0 V, f_{OSC} = 4 MHz, ϕ_{osc}/64	—	0.4	—		Max. guideline = 1.1 × typ.*1 *2
			Active (medium-speed) mode, V_{CC} = 3.0 V, f_{OSC} = 10 MHz, ϕ_{osc}/64	—	0.6	0.8		*1 *2
Sleep mode supply current	I_{SLEEP}	V_{CC}	V_{CC} = 1.8 V, f_{OSC} = 2 MHz	—	0.3	—	mA	Max. guideline = 1.1 × typ.*1 *2
			V_{CC} = 3.0 V, f_{OSC} = 4 MHz	—	1.2	—		Max. guideline = 1.1 × typ.*1 *2
			V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	2.5	4.0		*1 *2
Subactive mode supply current	I_{SUB}	V_{CC}	V_{CC} = 1.8 V, 32-kHz crystal resonator (ϕ_{SUB} = ϕ_W/2)	—	6.5	—	µA	Reference value *1 *2
			V_{CC} = 2.7 V, 32-kHz crystal resonator (ϕ_{SUB} = ϕ_W/8)	—	5.5	—		Reference value *1 *2
			V_{CC} = 2.7 V, 32-kHz crystal resonator (ϕ_{SUB} = ϕ_W/2)	—	11	17		*1 *2

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Note
Subsleep mode supply current	I_{SUBSP}	V_{CC}	V_{CC} = 2.7 V, 32-kHz crystal resonator used ($\phi_{SUB} = \phi_W/2$)	—	5.0	8.5	µA	*1 *2
Watch mode supply current	I_{WATCH}	V_{CC}	V_{CC} = 1.8 V, Ta = 25°C 32-kHz crystal resonator used	—	0.5	—	µA	Reference value *1 *2
			V_{CC} = 2.7 V, 32-kHz crystal resonator used	—	1.5	5.0		*1 *2
Standby mode supply current	I_{STBY}	V_{CC}	V_{CC} = 3.0 V, Ta = 25°C, 32-kHz crystal resonator not used	—	0.1	—	µA	Reference value *1 *2
			32-kHz crystal resonator not used	—	1.0	5.0		*1 *2
RAM data retaining voltage	V_{RAM}	V_{CC}		1.5	—	—	V	
Permissible output low current (per pin)	I_{OL}	Output pins except port 9		—	—	0.5	mA	
		P90 to P93		—	—	15.0		
Permissible output low current (total)	ΣI_{OL}	Output pins except port 9		—	—	20.0	mA	
		Port 9		—	—	TBD		
Permissible output high current (per pin)	$-I_{OH}$	All output pins	Vcc = 2.7 to 3.6 V	—	—	2.0	mA	
			Vcc = 1.8 to 3.6 V	—	—	0.2		
Permissible output high current (total)	$\Sigma - I_{OH}$	All output pins		—	—	10.0	mA	

Notes: 1. Pin states during current measurement.

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pins
Active (high-speed) mode (I_{OPE1})	V_{CC}	Only CPU operates On-chip WDT oscillator is off	V_{CC}	System clock oscillator: Crystal resonator Subclock oscillator: Pin X1 = GND
Active (medium-speed) mode (I_{OPE2})				
Sleep mode	V_{CC}	Only on-chip timers operate On-chip WDT oscillator is off	V_{CC}	
Subactive mode	V_{CC}	Only CPU operates On-chip WDT oscillator is off	V_{CC}	System clock oscillator: Crystal resonator Subclock oscillator: Crystal resonator
Subsleep mode	V_{CC}	Only on-chip timers operate, CPU stops On-chip WDT oscillator is off	V_{CC}	
Watch mode	V_{CC}	Only time base operates, CPU stops On-chip WDT oscillator is off	V_{CC}	
Standby mode	V_{CC}	CPU and timers both stop On-chip WDT oscillator is off 32KSTOP = 1	V_{CC}	System clock oscillator: Crystal resonator Subclock oscillator: Crystal resonator

2. Excludes current in pull-up MOS transistors and output buffers.
3. When IRQ0, IRQ1, and IRQ3 in PMRB are set to 0, and IRQ0, IRQ1, and IRQ3 in PMRE are set to 1, the maximum value is V_{CC} + 0.3 (V).

25.4.3 AC Characteristics

Table 25.12 lists the control signal timing, table 25.13 lists the serial interface timing, and table 25.14 lists the I^2C bus interface timing.

Table 25.12 Control Signal Timing

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = AV_{SS} = 0.0 V, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
System clock oscillation frequency	f_{OSC}	OSC1, OSC2	V_{CC} = 2.7 to 3.6 V	2.0	—	10.0	MHz	
			V_{CC} = 1.8 to 3.6 V	2.0	—	4.2		
System clock on-chip oscillation frequency	f_{Rosc}		On-chip oscillator for system clock selected V_{CC} = 1.8 to 3.6 V	0.5	—	10.0		*3
OSC clock (ϕ_{OSC}) cycle time	t_{OSC}	OSC1, OSC2	V_{CC} = 2.7 to 3.6 V	100	—	500	ns	Figure 25.14
			V_{CC} = 1.8 to 3.6 V	238	—	500		
System clock on-chip oscillation clock (ϕ_{ROSC}) cycle time	t_{ROSC}		On-chip oscillator for system clock selected V_{CC} = 1.8 to 3.6 V	100	—	2000		*3
System clock (ϕ) cycle time	t_{cyc}			1	—	64	t_{OSC}	
				—	—	32	µs	

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Reference Figure
Subclock oscillator oscillation frequency	f_W	X1, X2		—	32.768 or 38.4	—	kHz	
Watch clock (ϕ_W) cycle time	t_W	X1, X2		—	30.5 or 26.0	—	µs	Figure 25.14
Subclock (ϕ_{SUB}) cycle time	t_{subcyc}			2	—	8	t_W	*1
Instruction cycle time				2	—	—	t_{cyc} t_{subcyc}	
Oscillation stabilization time	t_{rc}	OSC1, OSC2	Ceramic resonator (Vcc = 2.2 to 3.6 V)	—	20	45	µs	Figure 25.23
			Ceramic resonator (Other than above)	—	80	—		
			Crystal resonator (Vcc = 2.7 to 3.6 V)	—	0.8	2.0	ms	
			Crystal resonator (Vcc = 2.2 to 3.6 V)	—	1.2	3.0		
			Other than above	—	—	50	ms	
		On-chip oscillator for system clock	At switching on	—	—	25	µs	*3
		X1, X2	V_{CC} = 2.2 to 3.6V	—	—	2	s	Figure 5.5
			Other than above	—	4	—		
External clock high width	t_{CPH}	OSC1	V_{CC} = 2.7 to 3.6 V	40	—	—	ns	Figure 25.14
			V_{CC} = 1.8 to 3.6 V	95	—	—		
		X1		—	15.26 or 13.02	—	µs	
External clock low width	t_{CPL}	OSC1	V_{CC} = 2.7 to 3.6 V	40	—	—	ns	Figure 25.14
			V_{CC} = 1.8 to 3.6 V	95	—	—		
		X1		—	15.26 or 13.02	—	µs	

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Reference Figure
External clock rising time	t_{CPr}	OSC1	V_{CC} = 2.7 to 3.6 V	—	—	10	ns	Figure 25.14
			V_{CC} = 1.8 to 3.6 V	—	—	24		
		X1		—	—	55.0	ns	
External clock falling time	t_{CPf}	OSC1	V_{CC} = 2.7 to 3.6 V	—	—	10	ns	Figure 25.14
			V_{CC} = 1.8 to 3.6 V	—	—	24		
		X1		—	—	55.0	ns	
$\overline{RES}$ pin low width	t_{REL}	$\overline{RES}$		10	—	—	t_{cyc}	Figure 25.15*2
Input pin high width	t_{IH}	$\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{NMI}$, $\overline{IRQ3}$, $\overline{IRQ4}$, IRQAEC, $\overline{WKP0}$ to $\overline{WKP7}$, TMIC, TMIF, TMIG, $\overline{ADTRG}$		2	—	—	t_{cyc} t_{subcyc}	Figure 25.16
		AEVL, AEVH	V_{CC} = 2.7 to 3.6 V	50	—	—	ns	
			V_{CC} = 1.8 to 3.6 V	110	—	—		
	t_{TCKWH}	TCLKA, TCLKB, TCLKC, TIOCA1, TIOCB1, TIOCA2, TIOCB2	Single edge specified	1.5	—	—	t_{cyc}	Figure 25.19
			Both edges specified	2.5	—	—		

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Reference Figure
Input pin low width	t_{IL}	$\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{NMI}$, $\overline{IRQ3}$, $\overline{IRQ4}$, IRQAEC, $\overline{WKP0}$ to $\overline{WKP7}$, TMIC, TMIF, TMIG, $\overline{ADTRG}$		2	—	—	t_{cyc} t_{subcyc}	Figure 25.16
		AEVL, AEVH	V_{CC} = 2.7 to 3.6 V	50	—	—	ns	
			V_{CC} = 1.8 to 3.6 V	110	—	—		
	t_{TCKWL}	TCLKA, TCLKB, TCLKC, TIOCA1, TIOCB1, TIOCA2, TIOCB2	Single edge specified	1.5	—	—	t_{cyc}	Figure 25.19
			Both edges specified	2.5	—	—		
UD pin minimum transition width	UD			4	—	—	t_{cyc} t_{subcyc}	Figure 24.21

Notes: 1. Selected with the SA1 and SA0 bits in the system control register 2 (SYSCR2).
2. For details on the power-on reset characteristics, refer to table 25.16 and figure 25.15.
3. The specifications may vary due to the effects of temperature, power-supply voltage, and dispersion of product lots. Thorough evaluation under the actual conditions of use is essential in the design of systems. As for actual specification, please confirm with our sales representatives.

Table 25.13 Serial Interface Timing

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, unless otherwise specified.

Item		Symbol	Test Condition	Values Min.	Typ.	Max.	Unit	Reference Figure
Input clock cycl	Asynchronous	t_{scyc}		4	—	—	t_{cyc} or t_{subcyc}	Figure 25.17
	Clocked synchronous			6	—	—		
Input clock pulse width		t_{SCKW}		0.4	—	0.6	t_{scyc}	Figure 25.17
Transmit data delay time (clocked synchronous)		t_{TXD}		—	—	1	t_{cyc} or t_{subcyc}	Figure 25.18
Receive data setup time (clocked synchronous)		t_{RXS}	V_{CC} = 2.7 to 3.6 V	100	—	—	ns	Figure 25.18
			Other than above	238				
Receive data hold time (clocked synchronous)		t_{RXH}	V_{CC} = 2.7 to 3.6 V	100	—	—	ns	Figure 25.18
			Other than above	238				

Table 25.14 I²C Bus Interface Timing

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = 0.0 V, Ta = –20 to +75°C, unless otherwise specified.

Item	Symbol	Test Condition	Values			Unit	Reference Figure
			Min.	Typ.	Max.		
SCL input cycle time	t_{SCL}		$12t_{cyc}$ + 600	—	—	ns	Figure 25.20
SCL input high width	t_{SCLH}		$3t_{cyc}$ + 300	—	—	ns	
SCL input low width	t_{SCLL}		$5t_{cyc}$ + 300	—	—	ns	
SCL and SDA input fall time	t_{Sf}		—	—	300	ns	
SCL and SDA input spike pulse removal time	t_{SP}		—	—	$1t_{cyc}$	ns	
SDA input bus-free time	t_{BUF}		$5t_{cyc}$	—	—	ns	
Start condition input hold time	t_{STAH}		$3t_{cyc}$	—	—	ns	
Retransmission start condition input setup time	t_{STAS}		$3t_{cyc}$	—	—	ns	
Setup time for stop condition input	t_{STOS}		$3t_{cyc}$	—	—	ns	
Data-input setup time	t_{SDAS}		$1t_{cyc}$ + 20	—	—	ns	
Data-input hold time	t_{SDAH}		0	—	—	ns	
Capacitive load of SCL and SDA	Cb		0	—	400	pF	
SCL and SDA output fall time	t_{Sf}		—	—	300	ns	

25.4.4 A/D Converter Characteristics

Table 25.15 lists the A/D converter characteristics.

Table 25.15 A/D Converter Characteristics

V_{CC} = 1.8 V to 3.6 V, V_{SS} = AV_{SS} = 0.0 V, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Values Typ.	Values Max.	Unit	Reference Figure
Analog power supply voltage	AV_{CC}	AV_{CC}		1.8	—	3.6	V	*1
Analog input voltage	AV_{IN}	AN0 to AN7		–0.3	—	AV_{CC} + 0.3	V	
Analog power supply current	AI_{OPE}	AV_{CC}	AV_{CC} = 3.0 V	—	—	1.0	Ma	
	AI_{STOP1}	AV_{CC}		—	600	—	µA	*2 Reference value
	AI_{STOP2}	AV_{CC}		—	—	5	µA	*3
Analog input capacitance	C_{AIN}	AN0 to AN7		—	—	15.0	pF	
Allowable signal source impedance	R_{AIN}			—	—	10.0	kΩ	
Resolution (data length)				—	—	10	bits	
Nonlinearity error			AV_{CC} = 2.7 to 3.6 V V_{CC} = 2.7 to 3.6 V	—	—	±3.5	LSB	
			AV_{CC} = 2.0 to 3.6 V V_{CC} = 2.0 to 3.6 V	—	—	±5.5		
			Other than above	—	—	±7.5		*4
Quantization error				—	—	±0.5	LSB	
Absolute accuracy			AV_{CC} = 2.7 to 3.6 V V_{CC} = 2.7 to 3.6 V	—	—	±4.0	LSB	
			AV_{CC} = 2.0 to 3.6 V V_{CC} = 2.0 to 3.6 V	—	—	±6.0		
			Other than above	—	—	±8.0		*4

Item	Symbol	Applicable Pins	Test Condition	Values Min.	Typ.	Max.	Unit	Note
Conversion time			AV_{CC} = 2.7 to 3.6 V V_{CC} = 2.7 to 3.6 V	12.4	—	124	µs	System clock oscillator selected
				7.8	15.5	31	µs	On-chip oscillator for system clock selected Reference value (f_{Rosc} = 4 MHz)
			Other than AV_{CC} = 2.7 to 3.6 V V_{CC} = 2.7 to 3.6 V	29.5	—	124	µs	System clock oscillator selected
				31	62	124	µs	On-chip oscillator for system clock selected Reference value (f_{Rosc} = 1MHz)

Notes: 1. Set AV_{CC} = V_{CC} when the A/D converter is not used.
2. AI_{STOP1} is the current in active and sleep modes while the A/D converter is idle.
3. AISTOP2 is the current at a reset, in standby or watch mode while the A/D converter is idle, or in the module standby state.
4. Conversion time is 29.5 µs. Ta = –20 to +75°C

25.4.5 Power-On Reset Circuit Characteristics

Table 25.16 Power-On Reset Circuit Characteristics

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = AV_{SS} = 0.0 V,
Ta = –20 to +75°C (general specifications), Ta = –40 to +85°C (wide temperature range specifications), unless otherwise specified.

Item	Symbol	Test Condition	Values			Unit	Reference Figure
			Min.	Typ.	Max.		
Reset voltage	V_rst		$0.7V_{CC}$	$0.8V_{CC}$	$0.9V_{CC}$	V	Figure 22.2
Power supply rise time	t_vtr		The Vcc rise time should be shorter than half the $\overline{RES}$ rise time.				
Reset count time	t_out		0.8	—	4.0	µs	
			0.8	—	16.0	µs	On-chip oscillator for system clock selected (Reference value)
Count start time	t_cr		Adjustable by the value of the external capacitor of the $\overline{RES}$ pin.				
On-chip pull-up resistance	R_P	Vcc = 3.0 V	60	100	—	kΩ	Figure 22.1

25.4.6 Watchdog Timer Characteristics

Table 25.17 Watchdog Timer Characteristics

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = AV_{SS} = 0.0 V,
Ta = –20 to +75°C (regular specifications), Ta = –40 to +85°C (wide-range specifications), unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Note
				Min.	Typ.	Max.		
On-chip watchdog timer oscillator overflow time	t_{ovf}			0.2	0.4	—	s	*

Note: * Indicates the period from the start of counting at 0 to the generation of an internal reset when the counter reaches 255 in the case where the on-chip watchdog timer oscillator is selected.

25.5 Operation Timing

Figures 25.14 to 25.21 show operation timings.

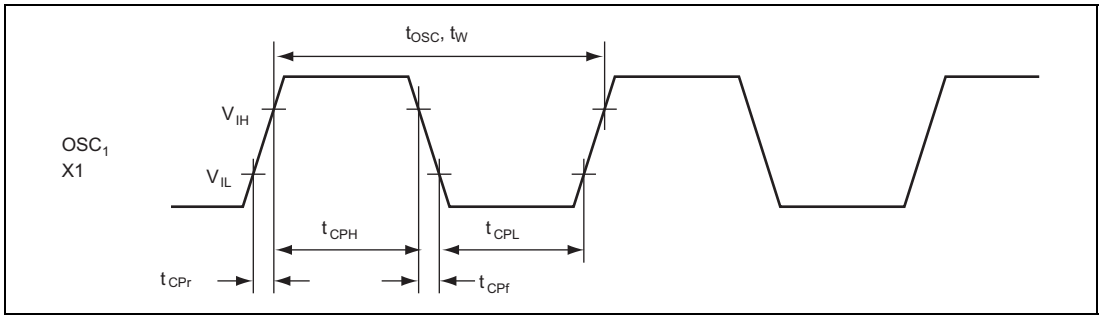


Figure 25.14 Clock Input Timing

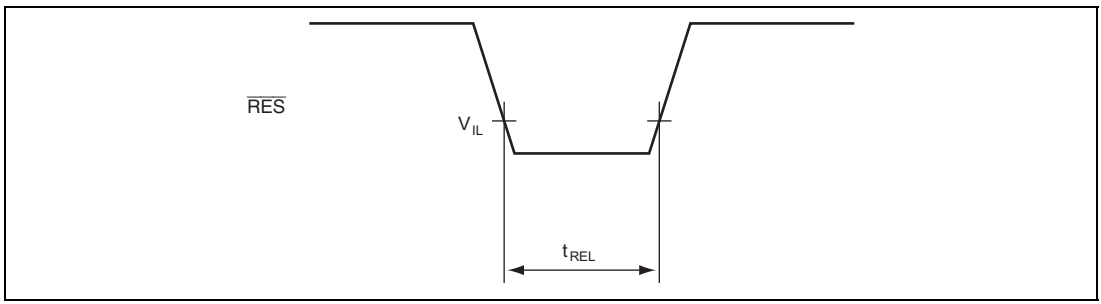


Figure 25.15 $\overline{RES}$ Low Width Timing

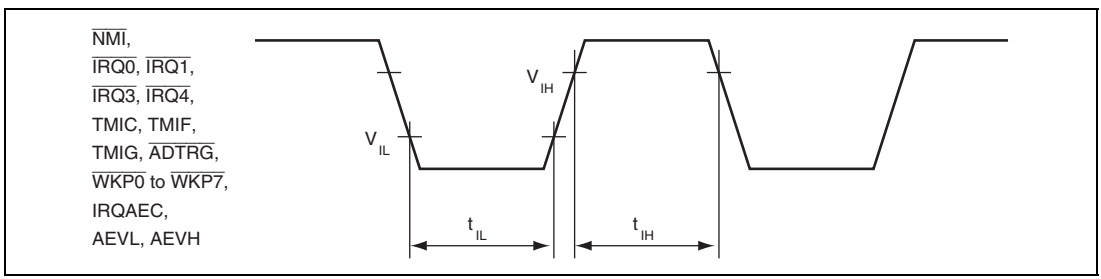


Figure 25.16 Input Timing

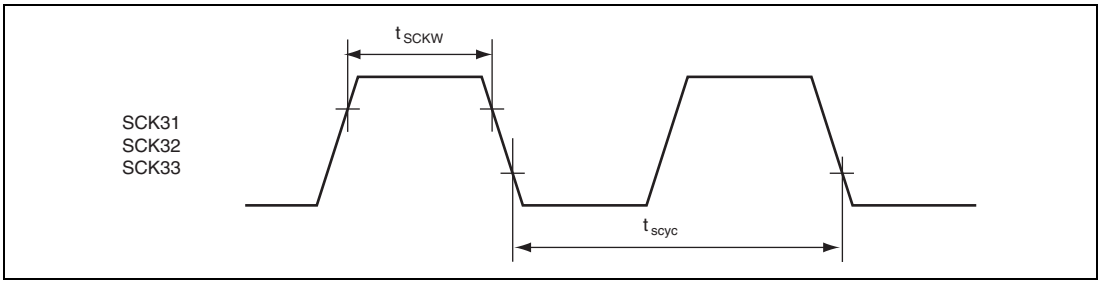
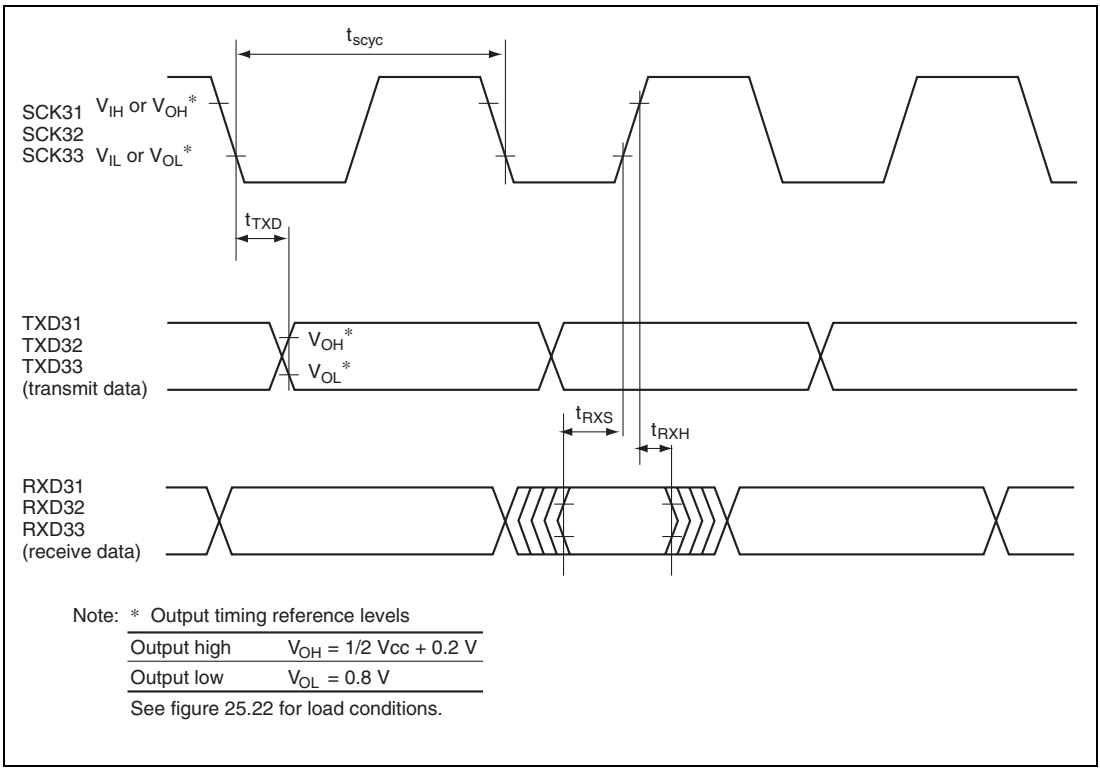


Figure 25.17 SCK3 Input Clock Timing



Note: * Output timing reference levels

Output high	V_{OH} = 1/2 Vcc + 0.2 V
Output low	V_{OL} = 0.8 V

See figure 25.22 for load conditions.

Figure 25.18 SCI3 Input/Output Timing in Clocked Synchronous Mode

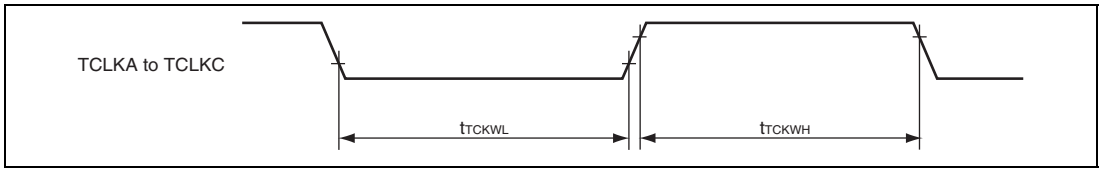


Figure 25.19 Clock Input Timing for TCLKA to TCLKC Pins

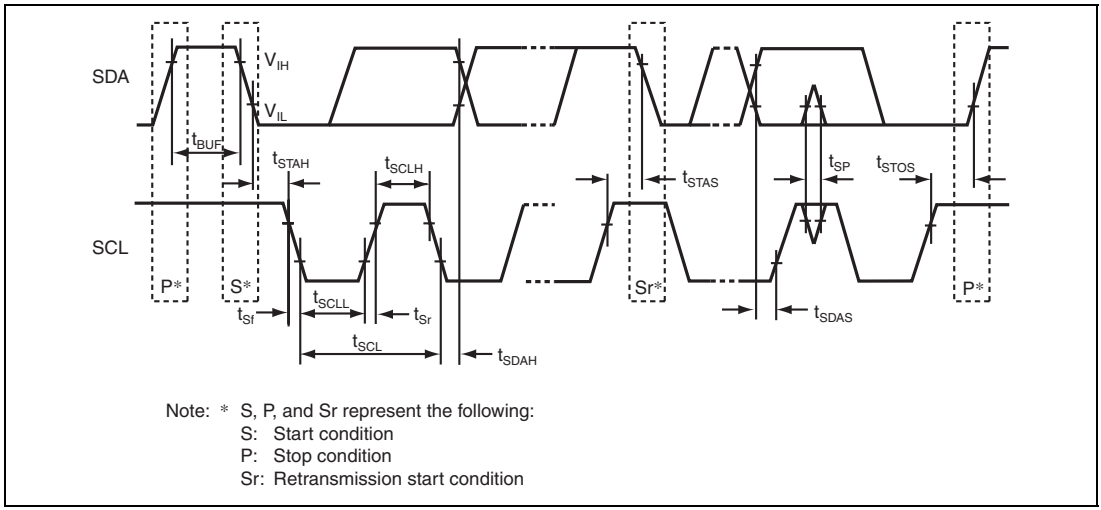


Figure 25.20 I²C Bus Interface Input/Output Timing

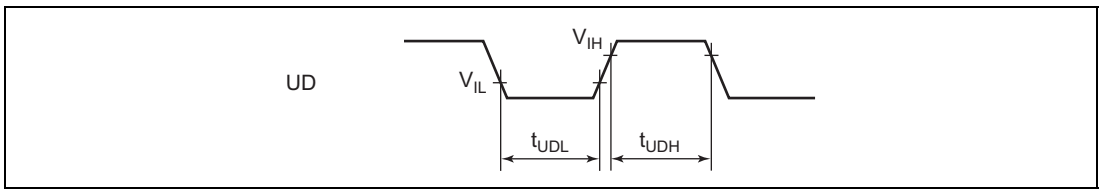


Figure 25.21 UD Pin Minimum Transition Width Timing

25.6 Output Load Circuit

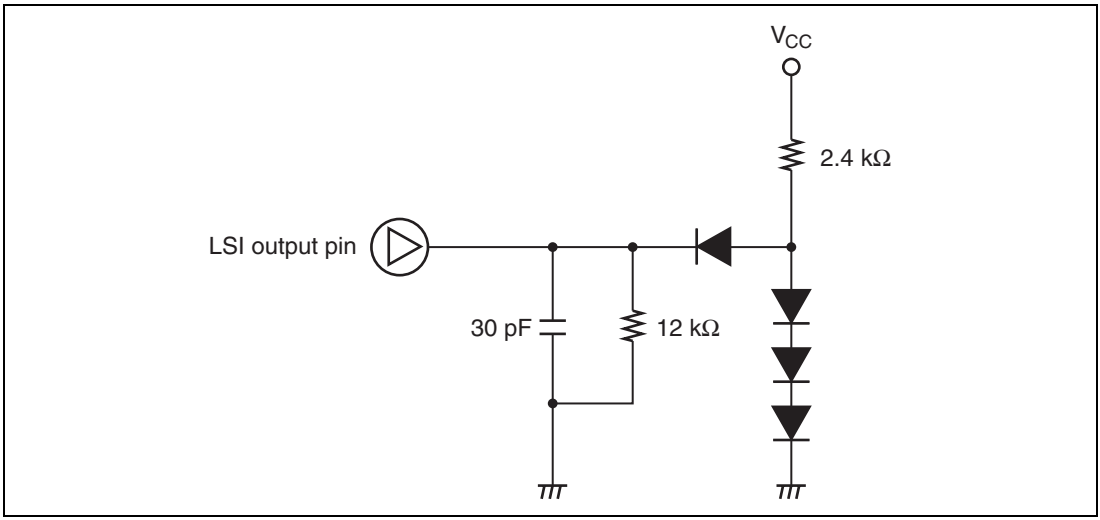


Figure 25.22 Output Load Condition

25.7 Recommended Resonators

(1) Recommended Crystal Resonators

Frequency (MHz)	Manufacturer	Product Type
4.194	NIHON DEMPA KOGYO CO.,LTD.	NR-18
10	NIHON DEMPA KOGYO CO.,LTD.	NR-18

(2) Recommended Ceramic Resonators

Frequency (MHz)	Manufacturer	Product Type
2	Murata Manufacturing Co., Ltd.	CSTCC2M00G53-B0
	Murata Manufacturing Co., Ltd.	CSTCC2M00G56-B0
4.194	Murata Manufacturing Co., Ltd.	CSTLS4M19G53-B0
	Murata Manufacturing Co., Ltd.	CSTLS4M19G56-B0
10	Murata Manufacturing Co., Ltd.	CSTLS10M0G53-B0
	Murata Manufacturing Co., Ltd.	CSTLS10M0G56-B0

Figure 25.23 Recommended Resonators

25.8 Usage Note

The F-ZTAT and masked ROM versions satisfy the electrical characteristics shown in this manual, but actual electrical characteristic values, operating margins, noise margins, and other properties may vary due to differences in manufacturing process, on-chip ROM, layout patterns, and so on.

When system evaluation testing is carried out using the F-ZTAT version, the same evaluation testing should also be conducted for the masked ROM version when changing over to that version.

Connecting a bypass capacitor is recommended for noise suppression.

Appendix

A. Instruction Set

A.1 Instruction List

Condition Code

Symbol	Description
Rd	General destination register
Rs	General source register
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 transition from 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

Symbol	Description
¬	NOT (logical complement)
( ), < >	Contents of operand

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

Condition Code Notation (cont)

Symbol	Description
↕	Changed according to execution result
*	Undetermined (no guaranteed value)
0	Cleared to 0
1	Set to 1
—	Not affected by execution of the instruction
Δ	Varies depending on conditions, described in notes

Table A.1 Instruction Set

1. Data Transfer Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
MOV	MOV.B #xx:8, Rd	B	2									#xx:8 → Rd8	—	—	↕	↕	0	—	2	
	MOV.B Rs, Rd	B		2								Rs8 → Rd8	—	—	↕	↕	0	—	2	
	MOV.B @ERs, Rd	B			2							@ERs → Rd8	—	—	↕	↕	0	—	4	
	MOV.B @(d:16, ERs), Rd	B				4						@(d:16, ERs) → Rd8	—	—	↕	↕	0	—	6	
	MOV.B @(d:24, ERs), Rd	B				8						@(d:24, ERs) → Rd8	—	—	↕	↕	0	—	10	
	MOV.B @ERs+, Rd	B					2					@ERs → Rd8 ERs32+1 → ERs32	—	—	↕	↕	0	—	6	
	MOV.B @aa:8, Rd	B						2				@aa:8 → Rd8	—	—	↕	↕	0	—	4	
	MOV.B @aa:16, Rd	B						4				@aa:16 → Rd8	—	—	↕	↕	0	—	6	
	MOV.B @aa:24, Rd	B						6				@aa:24 → Rd8	—	—	↕	↕	0	—	8	
	MOV.B Rs, @ERd	B			2							Rs8 → @ERd	—	—	↕	↕	0	—	4	
	MOV.B Rs, @(d:16, ERd)	B				4						Rs8 → @(d:16, ERd)	—	—	↕	↕	0	—	6	
	MOV.B Rs, @(d:24, ERd)	B				8						Rs8 → @(d:24, ERd)	—	—	↕	↕	0	—	10	
	MOV.B Rs, @–ERd	B					2					ERd32–1 → ERd32 Rs8 → @ERd	—	—	↕	↕	0	—	6	
	MOV.B Rs, @aa:8	B						2				Rs8 → @aa:8	—	—	↕	↕	0	—	4	
	MOV.B Rs, @aa:16	B						4				Rs8 → @aa:16	—	—	↕	↕	0	—	6	
	MOV.B Rs, @aa:24	B						6				Rs8 → @aa:24	—	—	↕	↕	0	—	8	
	MOV.W #xx:16, Rd	W	4									#xx:16 → Rd16	—	—	↕	↕	0	—	4	
	MOV.W Rs, Rd	W		2								Rs16 → Rd16	—	—	↕	↕	0	—	2	
	MOV.W @ERs, Rd	W			2							@ERs → Rd16	—	—	↕	↕	0	—	4	
	MOV.W @(d:16, ERs), Rd	W				4						@(d:16, ERs) → Rd16	—	—	↕	↕	0	—	6	
	MOV.W @(d:24, ERs), Rd	W				8						@(d:24, ERs) → Rd16	—	—	↕	↕	0	—	10	
	MOV.W @ERs+, Rd	W					2					@ERs → Rd16 ERs32+2 → @ERd32	—	—	↕	↕	0	—	6	
	MOV.W @aa:16, Rd	W						4				@aa:16 → Rd16	—	—	↕	↕	0	—	6	
	MOV.W @aa:24, Rd	W						6				@aa:24 → Rd16	—	—	↕	↕	0	—	8	
	MOV.W Rs, @ERd	W			2							Rs16 → @ERd	—	—	↕	↕	0	—	4	
	MOV.W Rs, @(d:16, ERd)	W				4						Rs16 → @(d:16, ERd)	—	—	↕	↕	0	—	6	
	MOV.W Rs, @(d:24, ERd)	W				8						Rs16 → @(d:24, ERd)	—	—	↕	↕	0	—	10	

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
MOV	MOV.W Rs, @–ERd	W					2					ERd32–2 → ERd32 Rs16 → @ERd	—	—	↕	↕	0	—	6	
	MOV.W Rs, @aa:16	W						4				Rs16 → @aa:16	—	—	↕	↕	0	—	6	
	MOV.W Rs, @aa:24	W						6				Rs16 → @aa:24	—	—	↕	↕	0	—	8	
	MOV.L #xx:32, Rd	L	6									#xx:32 → Rd32	—	—	↕	↕	0	—	6	
	MOV.L ERs, ERd	L		2								ERs32 → ERd32	—	—	↕	↕	0	—	2	
	MOV.L @ERs, ERd	L			4							@ERs → ERd32	—	—	↕	↕	0	—	8	
	MOV.L @(d:16, ERs), ERd	L				6						@(d:16, ERs) → ERd32	—	—	↕	↕	0	—	10	
	MOV.L @(d:24, ERs), ERd	L				10						@(d:24, ERs) → ERd32	—	—	↕	↕	0	—	14	
	MOV.L @ERs+, ERd	L					4					@ERs → ERd32 ERs32+4 → ERs32	—	—	↕	↕	0	—	10	
	MOV.L @aa:16, ERd	L						6				@aa:16 → ERd32	—	—	↕	↕	0	—	10	
	MOV.L @aa:24, ERd	L						8				@aa:24 → ERd32	—	—	↕	↕	0	—	12	
	MOV.L ERs, @ERd	L			4							ERs32 → @ERd	—	—	↕	↕	0	—	8	
	MOV.L ERs, @(d:16, ERd)	L				6						ERs32 → @(d:16, ERd)	—	—	↕	↕	0	—	10	
	MOV.L ERs, @(d:24, ERd)	L				10						ERs32 → @(d:24, ERd)	—	—	↕	↕	0	—	14	
	MOV.L ERs, @–ERd	L					4					ERd32–4 → ERd32 ERs32 → @ERd	—	—	↕	↕	0	—	10	
	MOV.L ERs, @aa:16	L						6				ERs32 → @aa:16	—	—	↕	↕	0	—	10	
	MOV.L ERs, @aa:24	L						8				ERs32 → @aa:24	—	—	↕	↕	0	—	12	
POP	POP.W Rn	W									2	@SP → Rn16 SP+2 → SP	—	—	↕	↕	0	—	6	
	POP.L ERn	L									4	@SP → ERn32 SP+4 → SP	—	—	↕	↕	0	—	10	
PUSH	PUSH.W Rn	W									2	SP–2 → SP Rn16 → @SP	—	—	↕	↕	0	—	6	
	PUSH.L ERn	L									4	SP–4 → SP ERn32 → @SP	—	—	↕	↕	0	—	10	
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							

2. Arithmetic Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
ADD	ADD.B #xx:8, Rd	B	2									Rd8+#xx:8 → Rd8	—	↕	↕	↕	↕	↕	2	
	ADD.B Rs, Rd	B		2								Rd8+Rs8 → Rd8	—	↕	↕	↕	↕	↕	2	
	ADD.W #xx:16, Rd	W	4									Rd16+#xx:16 → Rd16	—	(1)	↕	↕	↕	↕	4	
	ADD.W Rs, Rd	W		2								Rd16+Rs16 → Rd16	—	(1)	↕	↕	↕	↕	2	
	ADD.L #xx:32, ERd	L	6									ERd32+#xx:32 → ERd32	—	(2)	↕	↕	↕	↕	6	
	ADD.L ERs, ERd	L		2								ERd32+ERs32 → ERd32	—	(2)	↕	↕	↕	↕	2	
ADDX	ADDX.B #xx:8, Rd	B	2									Rd8+#xx:8 +C → Rd8	—	↕	↕	(3)	↕	↕	2	
	ADDX.B Rs, Rd	B		2								Rd8+Rs8 +C → Rd8	—	↕	↕	(3)	↕	↕	2	
ADDS	ADDS.L #1, ERd	L		2								ERd32+1 → ERd32	—	—	—	—	—	—	2	
	ADDS.L #2, ERd	L		2								ERd32+2 → ERd32	—	—	—	—	—	—	2	
	ADDS.L #4, ERd	L		2								ERd32+4 → ERd32	—	—	—	—	—	—	2	
INC	INC.B Rd	B		2								Rd8+1 → Rd8	—	—	↕	↕	↕	—	2	
	INC.W #1, Rd	W		2								Rd16+1 → Rd16	—	—	↕	↕	↕	—	2	
	INC.W #2, Rd	W		2								Rd16+2 → Rd16	—	—	↕	↕	↕	—	2	
	INC.L #1, ERd	L		2								ERd32+1 → ERd32	—	—	↕	↕	↕	—	2	
	INC.L #2, ERd	L		2								ERd32+2 → ERd32	—	—	↕	↕	↕	—	2	
DAA	DAA Rd	B		2								Rd8 decimal adjust → Rd8	—	*	↕	↕	*	↕	2	
SUB	SUB.B Rs, Rd	B		2								Rd8–Rs8 → Rd8	—	↕	↕	↕	↕	↕	2	
	SUB.W #xx:16, Rd	W	4									Rd16–#xx:16 → Rd16	—	(1)	↕	↕	↕	↕	4	
	SUB.W Rs, Rd	W		2								Rd16–Rs16 → Rd16	—	(1)	↕	↕	↕	↕	2	
	SUB.L #xx:32, ERd	L	6									ERd32–#xx:32 → ERd32	—	(2)	↕	↕	↕	↕	6	
	SUB.L ERs, ERd	L		2								ERd32–ERs32 → ERd32	—	(2)	↕	↕	↕	↕	2	
SUBX	SUBX.B #xx:8, Rd	B	2									Rd8–#xx:8–C → Rd8	—	↕	↕	(3)	↕	↕	2	
	SUBX.B Rs, Rd	B		2								Rd8–Rs8–C → Rd8	—	↕	↕	(3)	↕	↕	2	
SUBS	SUBS.L #1, ERd	L		2								ERd32–1 → ERd32	—	—	—	—	—	—	2	
	SUBS.L #2, ERd	L		2								ERd32–2 → ERd32	—	—	—	—	—	—	2	
	SUBS.L #4, ERd	L		2								ERd32–4 → ERd32	—	—	—	—	—	—	2	
DEC	DEC.B Rd	B		2								Rd8–1 → Rd8	—	—	↕	↕	↕	—	2	
	DEC.W #1, Rd	W		2								Rd16–1 → Rd16	—	—	↕	↕	↕	—	2	
	DEC.W #2, Rd	W		2								Rd16–2 → Rd16	—	—	↕	↕	↕	—	2	

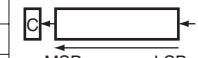
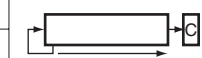
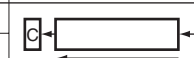
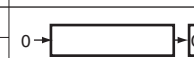
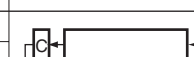
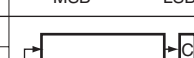
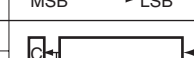
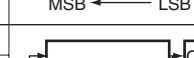
Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
DEC	DEC.L #1, ERd	L		2								ERd32–1 → ERd32	—	—	↕	↕	↕	—	2	
	DEC.L #2, ERd	L		2								ERd32–2 → ERd32	—	—	↕	↕	↕	—	2	
DAS	DAS.Rd	B		2								Rd8 decimal adjust → Rd8	—	*	↕	↕	*	—	2	
MULXU	MULXU. B Rs, Rd	B		2								Rd8 × Rs8 → Rd16 (unsigned multiplication)	—	—	—	—	—	—	14	
	MULXU. W Rs, ERd	W		2								Rd16 × Rs16 → ERd32 (unsigned multiplication)	—	—	—	—	—	—	22	
MULXS	MULXS. B Rs, Rd	B		4								Rd8 × Rs8 → Rd16 (signed multiplication)	—	—	↕	↕	—	—	16	
	MULXS. W Rs, ERd	W		4								Rd16 × Rs16 → ERd32 (signed multiplication)	—	—	↕	↕	—	—	24	
DIVXU	DIVXU. B Rs, Rd	B		2								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	(6)	(7)	—	—	14	
	DIVXU. W Rs, ERd	W		2								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	(6)	(7)	—	—	22	
DIVXS	DIVXS. B Rs, Rd	B		4								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	(8)	(7)	—	—	16	
	DIVXS. W Rs, ERd	W		4								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	(8)	(7)	—	—	24	
CMP	CMP.B #xx:8, Rd	B	2									Rd8–#xx:8	—	↕	↕	↕	↕	↕	2	
	CMP.B Rs, Rd	B		2								Rd8–Rs8	—	↕	↕	↕	↕	↕	2	
	CMP.W #xx:16, Rd	W	4									Rd16–#xx:16	—	(1)	↕	↕	↕	↕	4	
	CMP.W Rs, Rd	W		2								Rd16–Rs16	—	(1)	↕	↕	↕	↕	2	
	CMP.L #xx:32, ERd	L	6									ERd32–#xx:32	—	(2)	↕	↕	↕	↕	4	
	CMP.L ERs, ERd	L		2								ERd32–ERs32	—	(2)	↕	↕	↕	↕	2	

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
NEG	NEG.B Rd	B		2								0–Rd8 → Rd8	—	↕	↕	↕	↕	↕	2	
	NEG.W Rd	W		2								0–Rd16 → Rd16	—	↕	↕	↕	↕	↕	2	
	NEG.L ERd	L		2								0–ERd32 → ERd32	—	↕	↕	↕	↕	↕	2	
EXTU	EXTU.W Rd	W		2								0 → (<bits 15 to 8> of Rd16)	—	—	0	↕	0	—	2	
	EXTU.L ERd	L		2								0 → (<bits 31 to 16> of ERd32)	—	—	0	↕	0	—	2	
EXTS	EXTS.W Rd	W		2								(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)	—	—	↕	↕	0	—	2	
	EXTS.L ERd	L		2								(<bit 15> of ERd32) → (<bits 31 to 16> of ERd32)	—	—	↕	↕	0	—	2	

3. Logic Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
AND	AND.B #xx:8, Rd	B	2									Rd8∧#xx:8 → Rd8	—	—	↕	↕	0	—	2	
	AND.B Rs, Rd	B		2								Rd8∧Rs8 → Rd8	—	—	↕	↕	0	—	2	
	AND.W #xx:16, Rd	W	4									Rd16∧#xx:16 → Rd16	—	—	↕	↕	0	—	4	
	AND.W Rs, Rd	W		2								Rd16∧Rs16 → Rd16	—	—	↕	↕	0	—	2	
	AND.L #xx:32, ERd	L	6									ERd32∧#xx:32 → ERd32	—	—	↕	↕	0	—	6	
	AND.L ERs, ERd	L		4								ERd32∧ERs32 → ERd32	—	—	↕	↕	0	—	4	
OR	OR.B #xx:8, Rd	B	2									Rd8∨#xx:8 → Rd8	—	—	↕	↕	0	—	2	
	OR.B Rs, Rd	B		2								Rd8∨Rs8 → Rd8	—	—	↕	↕	0	—	2	
	OR.W #xx:16, Rd	W	4									Rd16∨#xx:16 → Rd16	—	—	↕	↕	0	—	4	
	OR.W Rs, Rd	W		2								Rd16∨Rs16 → Rd16	—	—	↕	↕	0	—	2	
	OR.L #xx:32, ERd	L	6									ERd32∨#xx:32 → ERd32	—	—	↕	↕	0	—	6	
	OR.L ERs, ERd	L		4								ERd32∨ERs32 → ERd32	—	—	↕	↕	0	—	4	
XOR	XOR.B #xx:8, Rd	B	2									Rd8⊕#xx:8 → Rd8	—	—	↕	↕	0	—	2	
	XOR.B Rs, Rd	B		2								Rd8⊕Rs8 → Rd8	—	—	↕	↕	0	—	2	
	XOR.W #xx:16, Rd	W	4									Rd16⊕#xx:16 → Rd16	—	—	↕	↕	0	—	4	
	XOR.W Rs, Rd	W		2								Rd16⊕Rs16 → Rd16	—	—	↕	↕	0	—	2	
	XOR.L #xx:32, ERd	L	6									ERd32⊕#xx:32 → ERd32	—	—	↕	↕	0	—	6	
	XOR.L ERs, ERd	L		4								ERd32⊕ERs32 → ERd32	—	—	↕	↕	0	—	4	
NOT	NOT.B Rd	B		2								¬ Rd8 → Rd8	—	—	↕	↕	0	—	2	
	NOT.W Rd	W		2								¬ Rd16 → Rd16	—	—	↕	↕	0	—	2	
	NOT.L ERd	L		2								¬ Rd32 → Rd32	—	—	↕	↕	0	—	2	

4. Shift Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
SHAL	SHAL.B Rd	B		2									—	—	↕	↕	↕	↕	2	
	SHAL.W Rd	W		2									—	—	↕	↕	↕	↕	2	
	SHAL.L ERd	L		2									—	—	↕	↕	↕	↕	2	
SHAR	SHAR.B Rd	B		2									—	—	↕	↕	0	↕	2	
	SHAR.W Rd	W		2									—	—	↕	↕	0	↕	2	
	SHAR.L ERd	L		2									—	—	↕	↕	0	↕	2	
SHLL	SHLL.B Rd	B		2									—	—	↕	↕	0	↕	2	
	SHLL.W Rd	W		2									—	—	↕	↕	0	↕	2	
	SHLL.L ERd	L		2									—	—	↕	↕	0	↕	2	
SHLR	SHLR.B Rd	B		2									—	—	↕	↕	0	↕	2	
	SHLR.W Rd	W		2									—	—	↕	↕	0	↕	2	
	SHLR.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTXL	ROTXL.B Rd	B		2									—	—	↕	↕	0	↕	2	
	ROTXL.W Rd	W		2									—	—	↕	↕	0	↕	2	
	ROTXL.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTXR	ROTXR.B Rd	B		2									—	—	↕	↕	0	↕	2	
	ROTXR.W Rd	W		2									—	—	↕	↕	0	↕	2	
	ROTXR.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTL	ROTL.B Rd	B		2									—	—	↕	↕	0	↕	2	
	ROTL.W Rd	W		2									—	—	↕	↕	0	↕	2	
	ROTL.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTR	ROTR.B Rd	B		2									—	—	↕	↕	0	↕	2	
	ROTR.W Rd	W		2									—	—	↕	↕	0	↕	2	
	ROTR.L ERd	L		2									—	—	↕	↕	0	↕	2	

5. Bit-Manipulation Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
BSET	BSET #xx:3, Rd	B		2								(#xx:3 of Rd8) ← 1	—	—	—	—	—	—	2	
	BSET #xx:3, @ERd	B			4							(#xx:3 of @ERd) ← 1	—	—	—	—	—	—	8	
	BSET #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) ← 1	—	—	—	—	—	—	8	
	BSET Rn, Rd	B		2								(Rn8 of Rd8) ← 1	—	—	—	—	—	—	2	
	BSET Rn, @ERd	B			4							(Rn8 of @ERd) ← 1	—	—	—	—	—	—	8	
	BSET Rn, @aa:8	B						4				(Rn8 of @aa:8) ← 1	—	—	—	—	—	—	8	
BCLR	BCLR #xx:3, Rd	B		2								(#xx:3 of Rd8) ← 0	—	—	—	—	—	—	2	
	BCLR #xx:3, @ERd	B			4							(#xx:3 of @ERd) ← 0	—	—	—	—	—	—	8	
	BCLR #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) ← 0	—	—	—	—	—	—	8	
	BCLR Rn, Rd	B		2								(Rn8 of Rd8) ← 0	—	—	—	—	—	—	2	
	BCLR Rn, @ERd	B			4							(Rn8 of @ERd) ← 0	—	—	—	—	—	—	8	
	BCLR Rn, @aa:8	B						4				(Rn8 of @aa:8) ← 0	—	—	—	—	—	—	8	
BNOT	BNOT #xx:3, Rd	B		2								(#xx:3 of Rd8) ← ¬ (#xx:3 of Rd8)	—	—	—	—	—	—	2	
	BNOT #xx:3, @ERd	B			4							(#xx:3 of @ERd) ← ¬ (#xx:3 of @ERd)	—	—	—	—	—	—	8	
	BNOT #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) ← ¬ (#xx:3 of @aa:8)	—	—	—	—	—	—	8	
	BNOT Rn, Rd	B		2								(Rn8 of Rd8) ← ¬ (Rn8 of Rd8)	—	—	—	—	—	—	2	
	BNOT Rn, @ERd	B			4							(Rn8 of @ERd) ← ¬ (Rn8 of @ERd)	—	—	—	—	—	—	8	
	BNOT Rn, @aa:8	B						4				(Rn8 of @aa:8) ← ¬ (Rn8 of @aa:8)	—	—	—	—	—	—	8	
BTST	BTST #xx:3, Rd	B		2								¬ (#xx:3 of Rd8) → Z	—	—	—	↕	—	—	2	
	BTST #xx:3, @ERd	B			4							¬ (#xx:3 of @ERd) → Z	—	—	—	↕	—	—	6	
	BTST #xx:3, @aa:8	B						4				¬ (#xx:3 of @aa:8) → Z	—	—	—	↕	—	—	6	
	BTST Rn, Rd	B		2								¬ (Rn8 of @Rd8) → Z	—	—	—	↕	—	—	2	
	BTST Rn, @ERd	B			4							¬ (Rn8 of @ERd) → Z	—	—	—	↕	—	—	6	
	BTST Rn, @aa:8	B						4				¬ (Rn8 of @aa:8) → Z	—	—	—	↕	—	—	6	
BLD	BLD #xx:3, Rd	B		2								(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
BLD	BLD #xx:3, @ERd	B			4							(#xx:3 of @ERd) → C	—	—	—	—	—	↕	6	
	BLD #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BILD	BILD #xx:3, Rd	B		2								¬ (#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BILD #xx:3, @ERd	B			4							¬ (#xx:3 of @ERd) → C	—	—	—	—	—	↕	6	
	BILD #xx:3, @aa:8	B						4				¬ (#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BST	BST #xx:3, Rd	B		2								C → (#xx:3 of Rd8)	—	—	—	—	—	—	2	
	BST #xx:3, @ERd	B			4							C → (#xx:3 of @ERd24)	—	—	—	—	—	—	8	
	BST #xx:3, @aa:8	B						4				C → (#xx:3 of @aa:8)	—	—	—	—	—	—	8	
BIST	BIST #xx:3, Rd	B		2								¬ C → (#xx:3 of Rd8)	—	—	—	—	—	—	2	
	BIST #xx:3, @ERd	B			4							¬ C → (#xx:3 of @ERd24)	—	—	—	—	—	—	8	
	BIST #xx:3, @aa:8	B						4				¬ C → (#xx:3 of @aa:8)	—	—	—	—	—	—	8	
BAND	BAND #xx:3, Rd	B		2								C∧(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BAND #xx:3, @ERd	B			4							C∧(#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BAND #xx:3, @aa:8	B						4				C∧(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BIAND	BIAND #xx:3, Rd	B		2								C∧ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BIAND #xx:3, @ERd	B			4							C∧ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BIAND #xx:3, @aa:8	B						4				C∧ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BOR	BOR #xx:3, Rd	B		2								C∨(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BOR #xx:3, @ERd	B			4							C∨(#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BOR #xx:3, @aa:8	B						4				C∨(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BIOR	BIOR #xx:3, Rd	B		2								C∨ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BIOR #xx:3, @ERd	B			4							C∨ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BIOR #xx:3, @aa:8	B						4				C∨ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BXOR	BXOR #xx:3, Rd	B		2								C⊕(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BXOR #xx:3, @ERd	B			4							C⊕(#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BXOR #xx:3, @aa:8	B						4				C⊕(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BIXOR	BIXOR #xx:3, Rd	B		2								C⊕ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BIXOR #xx:3, @ERd	B			4							C⊕ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BIXOR #xx:3, @aa:8	B						4				C⊕ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	

6. Branching Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		Branch Condition	I	H	N	Z	V	C	Normal	Advanced
Bcc	BRA d:8 (BT d:8)	—							2			If condition is true then PC ← PC+d else next;	Always	—	—	—	—	—	—	4	
	BRA d:16 (BT d:16)	—							4					—	—	—	—	—	—	6	
	BRN d:8 (BF d:8)	—							2				Never	—	—	—	—	—	—	4	
	BRN d:16 (BF d:16)	—							4					—	—	—	—	—	—	6	
	BHI d:8	—							2				C∨ Z = 0	—	—	—	—	—	—	4	
	BHI d:16	—							4					—	—	—	—	—	—	6	
	BLS d:8	—							2				C∨ Z = 1	—	—	—	—	—	—	4	
	BLS d:16	—							4					—	—	—	—	—	—	6	
	BCC d:8 (BHS d:8)	—							2				C = 0	—	—	—	—	—	—	4	
	BCC d:16 (BHS d:16)	—							4					—	—	—	—	—	—	6	
	BCS d:8 (BLO d:8)	—							2				C = 1	—	—	—	—	—	—	4	
	BCS d:16 (BLO d:16)	—							4					—	—	—	—	—	—	6	
	BNE d:8	—							2				Z = 0	—	—	—	—	—	—	4	
	BNE d:16	—							4					—	—	—	—	—	—	6	
	BEQ d:8	—							2				Z = 1	—	—	—	—	—	—	4	
	BEQ d:16	—							4					—	—	—	—	—	—	6	
	BVC d:8	—							2				V = 0	—	—	—	—	—	—	4	
	BVC d:16	—							4					—	—	—	—	—	—	6	
	BVS d:8	—							2				V = 1	—	—	—	—	—	—	4	
	BVS d:16	—							4					—	—	—	—	—	—	6	
	BPL d:8	—							2				N = 0	—	—	—	—	—	—	4	
	BPL d:16	—							4					—	—	—	—	—	—	6	
	BMI d:8	—							2				N = 1	—	—	—	—	—	—	4	
	BMI d:16	—							4					—	—	—	—	—	—	6	
	BGE d:8	—							2				N⊕V = 0	—	—	—	—	—	—	4	
	BGE d:16	—							4					—	—	—	—	—	—	6	
	BLT d:8	—							2				N⊕V = 1	—	—	—	—	—	—	4	
	BLT d:16	—							4					—	—	—	—	—	—	6	
	BGT d:8	—							2				Z∨ (N⊕V) = 0	—	—	—	—	—	—	4	
	BGT d:16	—							4					—	—	—	—	—	—	6	
	BLE d:8	—							2				Z∨ (N⊕V) = 1	—	—	—	—	—	—	4	
	BLE d:16	—							4					—	—	—	—	—	—	6	

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
JMP	JMP @ERn	—			2							PC ← ERn	—	—	—	—	—	—	4	
	JMP @aa:24	—						4				PC ← aa:24	—	—	—	—	—	—	6	
	JMP @@aa:8	—								2		PC ← @aa:8	—	—	—	—	—	—	8	10
BSR	BSR d:8	—							2			PC → @–SP PC ← PC+d:8	—	—	—	—	—	—	6	8
	BSR d:16	—							4			PC → @–SP PC ← PC+d:16	—	—	—	—	—	—	8	10
JSR	JSR @ERn	—			2							PC → @–SP PC ← ERn	—	—	—	—	—	—	6	8
	JSR @aa:24	—						4				PC → @–SP PC ← aa:24	—	—	—	—	—	—	8	10
	JSR @@aa:8	—								2		PC → @–SP PC ← @aa:8	—	—	—	—	—	—	8	12
RTS	RTS	—									2	PC ← @SP+	—	—	—	—	—	—	8	10

7. System Control Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
RTE	RTE	—										CCR ← @SP+ PC ← @SP+	↕	↕	↕	↕	↕	↕	10	
SLEEP	SLEEP	—										Transition to power-down state							2	
LDC	LDC #xx:8, CCR	B	2									#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
	LDC Rs, CCR	B		2								Rs8 → CCR	↕	↕	↕	↕	↕	↕	2	
	LDC @ERs, CCR	W			4							@ERs → CCR	↕	↕	↕	↕	↕	↕	6	
	LDC @(d:16, ERs), CCR	W				6						@(d:16, ERs) → CCR	↕	↕	↕	↕	↕	↕	8	
	LDC @(d:24, ERs), CCR	W				10						@(d:24, ERs) → CCR	↕	↕	↕	↕	↕	↕	12	
	LDC @ERs+, CCR	W					4					@ERs → CCR ERs32+2 → ERs32	↕	↕	↕	↕	↕	↕	8	
	LDC @aa:16, CCR	W						6				@aa:16 → CCR	↕	↕	↕	↕	↕	↕	8	
	LDC @aa:24, CCR	W						8				@aa:24 → CCR	↕	↕	↕	↕	↕	↕	10	
STC	STC CCR, Rd	B		2								CCR → Rd8							2	
	STC CCR, @ERd	W			4							CCR → @ERd							6	
	STC CCR, @(d:16, ERd)	W				6						CCR → @(d:16, ERd)							8	
	STC CCR, @(d:24, ERd)	W				10						CCR → @(d:24, ERd)							12	
	STC CCR, @–ERd	W					4					ERd32–2 → ERd32 CCR → @ERd							8	
	STC CCR, @aa:16	W						6				CCR → @aa:16							8	
	STC CCR, @aa:24	W						8				CCR → @aa:24							10	
ANDC	ANDC #xx:8, CCR	B	2									CCR∧#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
ORC	ORC #xx:8, CCR	B	2									CCR∨#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
XORC	XORC #xx:8, CCR	B	2									CCR⊕#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
NOP	NOP	—									2	PC ← PC+2							2	

8. Block Transfer Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
EEPMOV	EEPMOV. B	—									4	if R4L ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4L–1 → R4L until R4L=0 else next	—	—	—	—	—	—	8+ 4n*2	
	EEPMOV. W	—									4	if R4 ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4–1 → R4 until R4=0 else next	—	—	—	—	—	—	8+ 4n*2	

Notes: 1. The number of states in cases where the instruction code and its operands are located in on-chip memory is shown here. For other cases, see appendix A.3, Number of 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 previous value.

(5) The number of states required for execution of an instruction that transfers data in 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.

A.2 Operation Code Map

Table A.2 Operation Code Map (1)

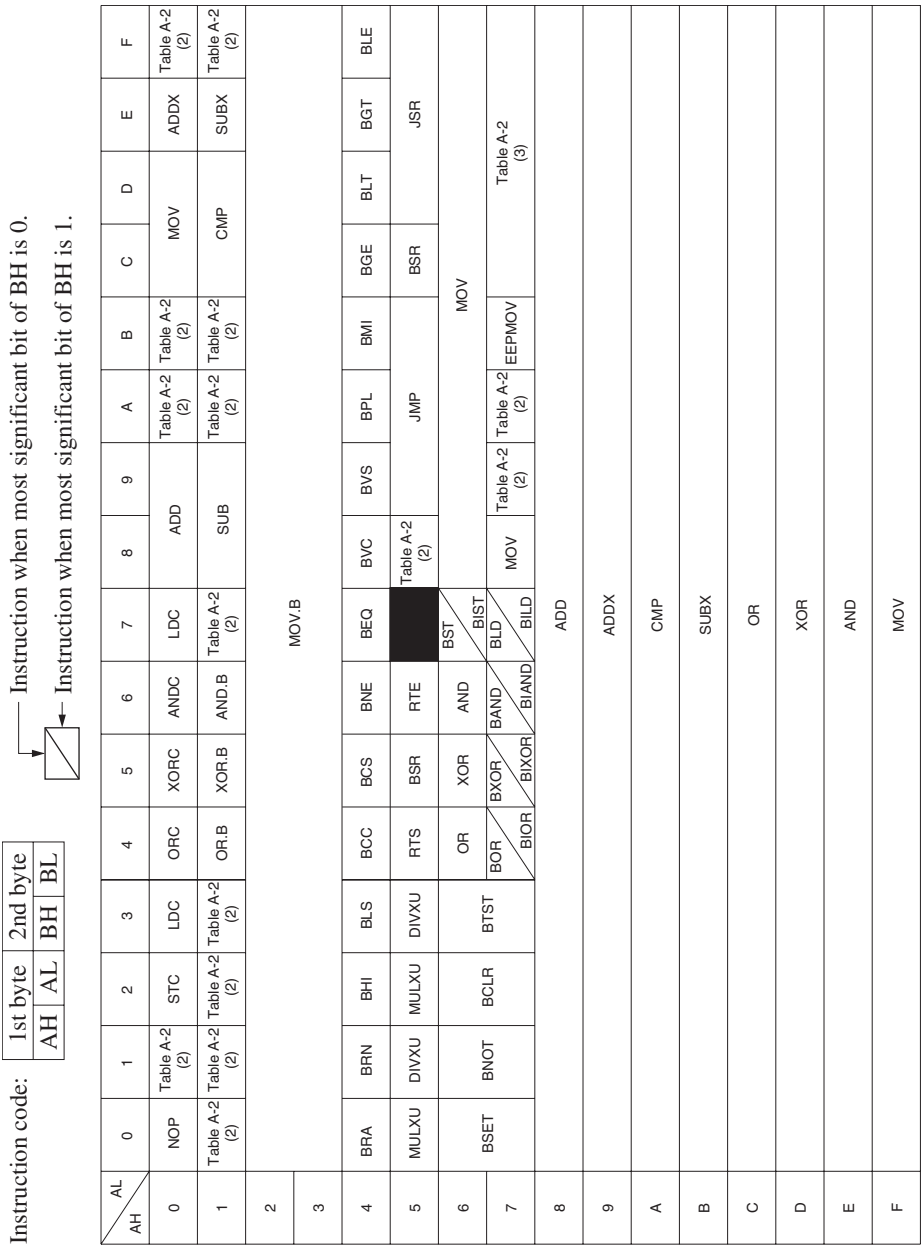


Table A.2 Operation Code Map (2)

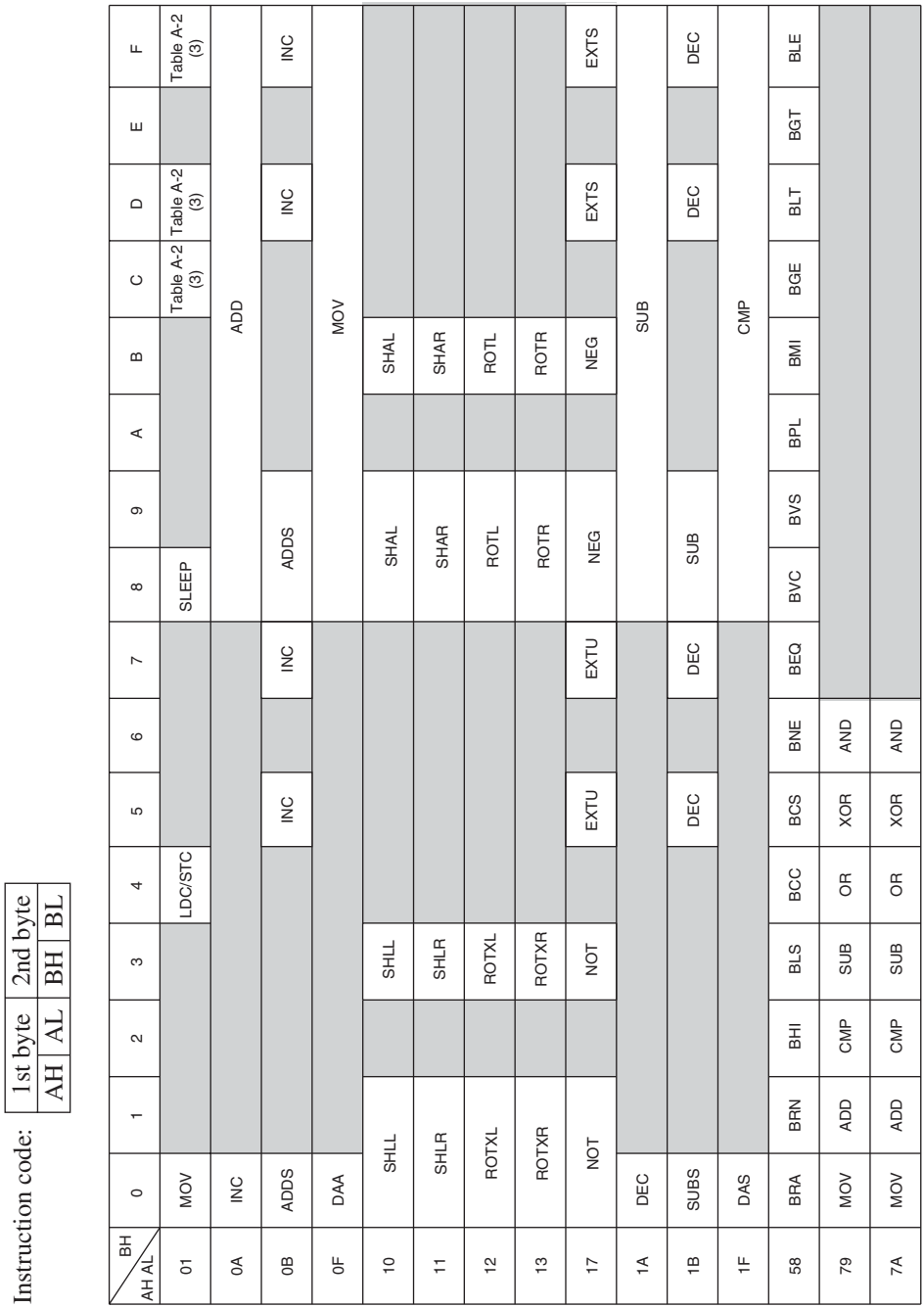
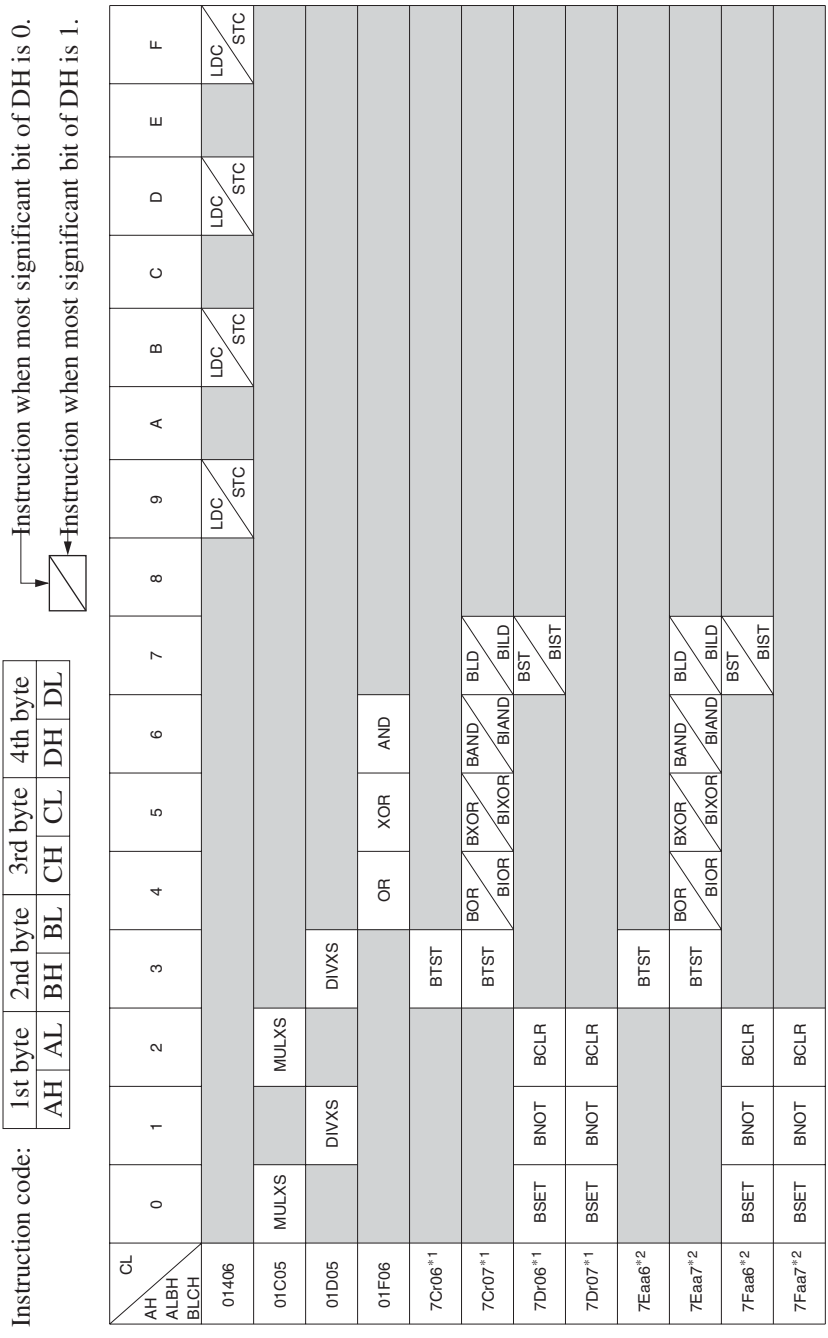


Table A.2 Operation Code Map (3)



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

A.3 Number of Execution States

The status of execution for each instruction of the H8/300H CPU and the method of calculating the number of states required for instruction execution are shown below. Table A.4 shows the number of cycles of each type occurring in each instruction, such as instruction fetch and data read/write. Table A.3 shows the number of states required for each cycle. The total number of states required for execution of an instruction can be calculated by the following expression:

$$\text{Execution states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

Examples: When instruction is fetched from on-chip ROM, and an on-chip RAM is accessed.

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 ROM, and 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 = $2 \times 2 + 1 \times 2 + 1 \times 2 = 8$

Table A.3 Number of Cycles in Each Instruction

Execution Status (Instruction Cycle)		Access Location	
		On-Chip Memory	On-Chip Peripheral Module
Instruction fetch	S_I	2	—
Branch address read	S_J		
Stack operation	S_K		
Byte data access	S_L		2 or 3*
Word data access	S_M		—
Internal operation	S_N	1	

Note: * Depends on which on-chip peripheral module is accessed. See section 24.1, Register Addresses (Address Order).

Table A.4 Number of Cycles in Each Instruction

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ADD	ADD.B #xx:8, Rd	1					
	ADD.B Rs, Rd	1					
	ADD.W #xx:16, Rd	2					
	ADD.W Rs, Rd	1					
	ADD.L #xx:32, ERd	3					
	ADD.L ERs, ERd	1					
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					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
Bcc	BLT d:8	2					
	BGT d:8	2					
	BLE d:8	2					
	BRA d:16(BT d:16)	2					2
	BRN d:16(BF d:16)	2					2
	BHI d:16	2					2
	BLS d:16	2					2
	BCC d:16(BHS d:16)	2					2
	BCS d:16(BLO d:16)	2					2
	BNE d:16	2					2
	BEQ d:16	2					2
	BVC d:16	2					2
	BVS d:16	2					2
	BPL d:16	2					2
	BMI d:16	2					2
	BGE d:16	2					2
	BLT d:16	2					2
	BGT d:16	2					2
	BLE d:16	2					2
BCLR	BCLR #xx:3, Rd	1					
	BCLR #xx:3, @ERd	2			2		
	BCLR #xx:3, @aa:8	2			2		
	BCLR Rn, Rd	1					
	BCLR Rn, @ERd	2			2		
	BCLR Rn, @aa:8	2			2		
BIAND	BIAND #xx:3, Rd	1					
	BIAND #xx:3, @ERd	2			1		
	BIAND #xx:3, @aa:8	2			1		
BILD	BILD #xx:3, Rd	1					
	BILD #xx:3, @ERd	2			1		
	BILD #xx:3, @aa:8	2			1		

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BIOR	BIOR #xx:3, Rd	1					
	BIOR #xx:3, @ERd	2			1		
	BIOR #xx:3, @aa:8	2			1		
BIST	BIST #xx:3, Rd	1					
	BIST #xx:3, @ERd	2			2		
	BIST #xx:3, @aa:8	2			2		
BIXOR	BIXOR #xx:3, Rd	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			2
BST	BST #xx:3, Rd	1					
	BST #xx:3, @ERd	2			2		
	BST #xx:3, @aa:8	2			2		

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BTST	BTST #xx:3, Rd	1					
	BTST #xx:3, @ERd	2			1		
	BTST #xx:3, @aa:8	2			1		
	BTST Rn, Rd	1					
	BTST Rn, @ERd	2			1		
	BTST Rn, @aa:8	2			1		
BXOR	BXOR #xx:3, Rd	1					
	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					12
	DIVXS.W Rs, ERd	2					20
DIVXU	DIVXU.B Rs, Rd	1					12
	DIVXU.W Rs, ERd	1					20
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					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
INC	INC.B Rd	1					
	INC.W #1/2, Rd	1					
	INC.L #1/2, ERd	1					
JMP	JMP @ERn	2					
	JMP @aa:24	2					2
	JMP @@aa:8	2	1				2
JSR	JSR @ERn	2		1			
	JSR @aa:24	2		1			2
	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	2
	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		2
	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		2
	MOV.B Rs, @aa:8	1			1		

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MOV	MOV.B Rs, @aa:16	2			1		
	MOV.B Rs, @aa:24	3			1		
	MOV.W #xx:16, Rd	2					
	MOV.W Rs, Rd	1					
	MOV.W @ERs, Rd	1				1	
	MOV.W @(d:16,ERs), Rd	2				1	
	MOV.W @(d:24,ERs), Rd	4				1	
	MOV.W @ERs+, Rd	1				1	2
	MOV.W @aa:16, Rd	2				1	
	MOV.W @aa:24, Rd	3				1	
	MOV.W Rs, @ERd	1				1	
	MOV.W Rs, @(d:16,ERd)	2				1	
	MOV.W Rs, @(d:24,ERd)	4				1	
MOV	MOV.W Rs, @-ERd	1				1	2
	MOV.W Rs, @aa:16	2				1	
	MOV.W Rs, @aa:24	3				1	
	MOV.L #xx:32, ERd	3					
	MOV.L ERs, ERd	1					
	MOV.L @ERs, ERd	2				2	
	MOV.L @(d:16,ERs), ERd	3				2	
	MOV.L @(d:24,ERs), ERd	5				2	
	MOV.L @ERs+, ERd	2				2	2
	MOV.L @aa:16, ERd	3				2	
	MOV.L @aa:24, ERd	4				2	
	MOV.L ERs,@ERd	2				2	
	MOV.L ERs, @(d:16,ERd)	3				2	
	MOV.L ERs, @(d:24,ERd)	5				2	
	MOV.L ERs, @-ERd	2				2	2
	MOV.L ERs, @aa:16	3				2	
	MOV.L ERs, @aa:24	4				2	
MOVFPE	MOVFPE @aa:16, Rd*[2]	2			1		
MOVTPE	MOVTPE Rs,@aa:16*[2]	2			1		

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MULXS	MULXS.B Rs, Rd	2					12
	MULXS.W Rs, ERd	2					20
MULXU	MULXU.B Rs, Rd	1					12
	MULXU.W Rs, ERd	1					20
NEG	NEG.B Rd	1					
	NEG.W Rd	1					
	NEG.L ERd	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	2
	POP.L ERn	2				2	2
PUSH	PUSH.W Rn	1				1	2
	PUSH.L ERn	2				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					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ROTXR	ROTXR.B Rd	1					
	ROTXR.W Rd	1					
	ROTXR.L ERd	1					
RTE	RTE	2		2			2
RTS	RTS	2		1			2
SHAL	SHAL.B Rd	1					
	SHAL.W Rd	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	2
	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					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
SUBX	SUBX #xx:8, Rd	1					
	SUBX. Rs, Rd	1					
XOR	XOR.B #xx:8, Rd	1					
	XOR.B Rs, Rd	1					
	XOR.W #xx:16, Rd	2					
	XOR.W Rs, Rd	1					
	XOR.L #xx:32, ERd	3					
	XOR.L ERs, ERd	2					
XORC	XORC #xx:8, CCR	1					

Notes: 1. n: Specified value in R4L. The source and destination operands are accessed n+1 times respectively.

2. It cannot be used in this LSI.

A.4 Combinations of Instructions and Addressing Modes

Table A.5 Combinations of Instructions and Addressing Modes

Functions	Instructions	Addressing Mode												
		#xx	Rn	@ERn	@(d:16.ERn)	@(d:24.ERn)	@ERn+/@ERn	@aa:8	@aa:16	@aa:24	@(d:8.PC)	@(d:16.PC)	@@aa:8	|
Data transfer instructions	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL	—	—	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—	—	WL
	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	RTE	—	—	—	—	—	—	—	—	—	—	—	—	◯
	SLEEP	—	—	—	—	—	—	—	—	—	—	—	—	◯
	LDC	B	B	W	W	W	W	—	W	W	—	—	—	◯
	STC	—	B	W	W	W	W	—	W	W	—	—	—	—
	ANDC, ORC, XORC	B	—	—	—	—	—	—	—	—	—	—	—	—
	NOP	—	—	—	—	—	—	—	—	—	—	—	—	◯
Block data transfer instructions		—	—	—	—	—	—	—	—	—	—	—	—	BW

B. I/O Ports

B.1 I/O Port Block Diagrams

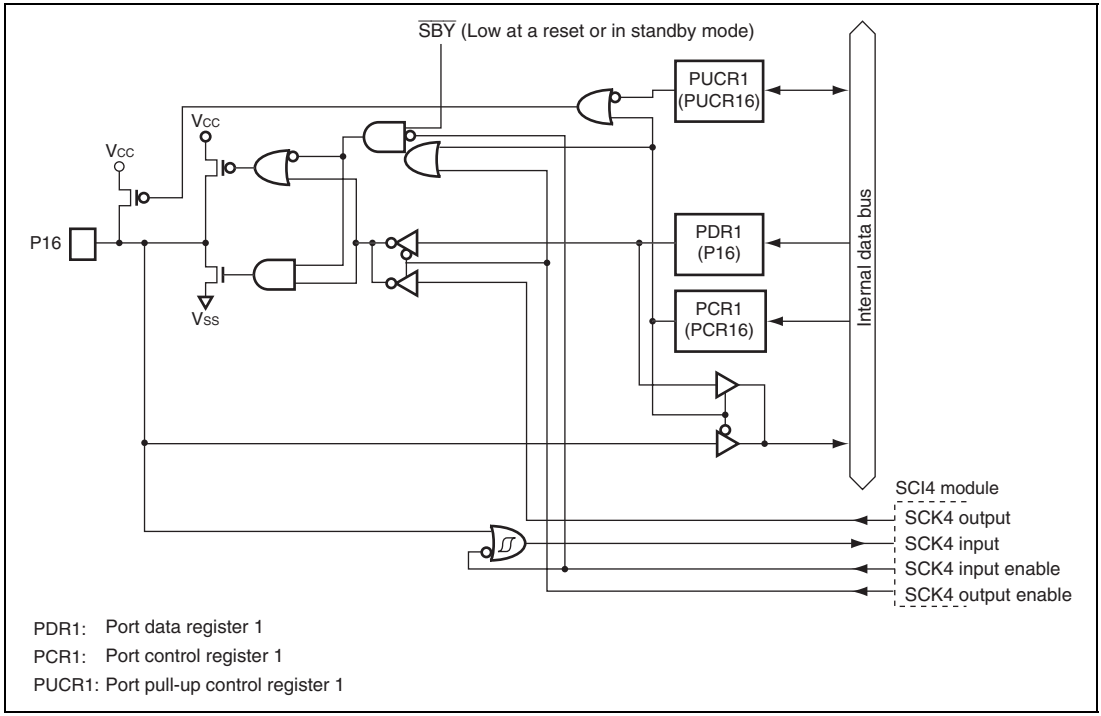


Figure B.1 (a) Port 1 Block Diagram (P16) (F-ZTAT™ Version)

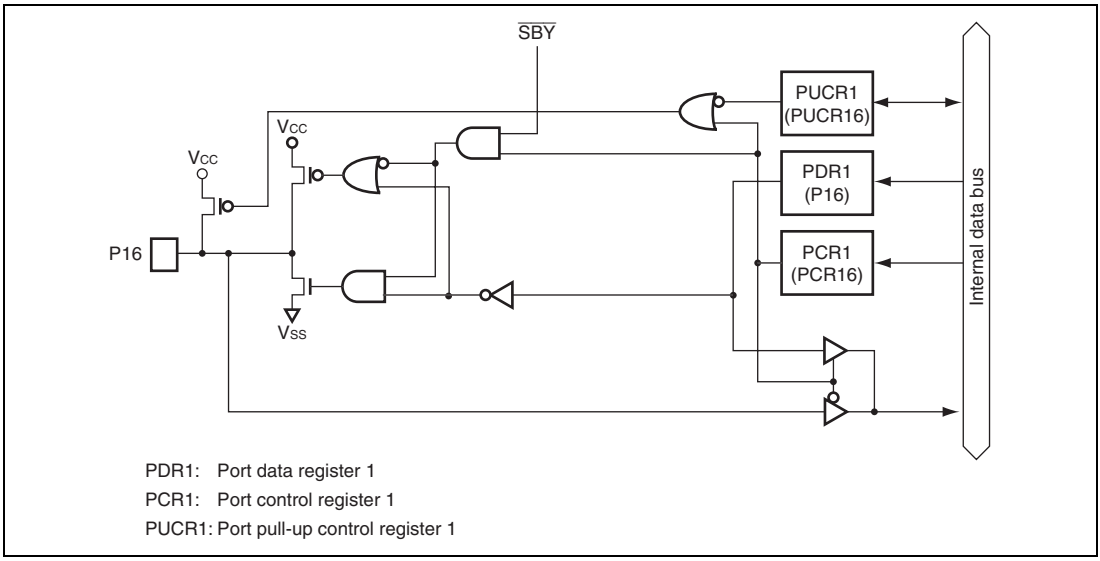


Figure B.1 (b) Port 1 Block Diagram (P16) (Masked ROM Version)

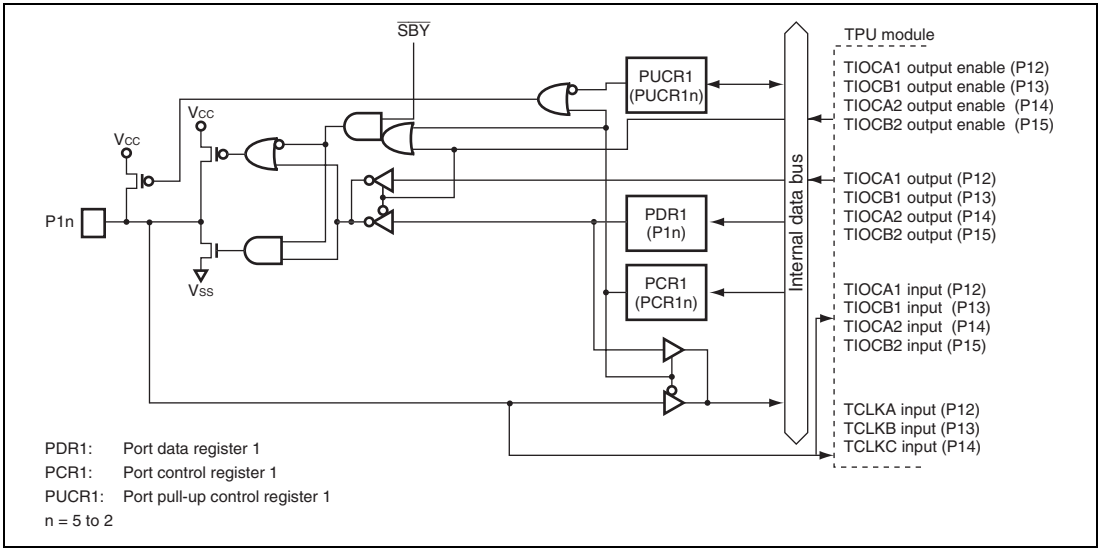
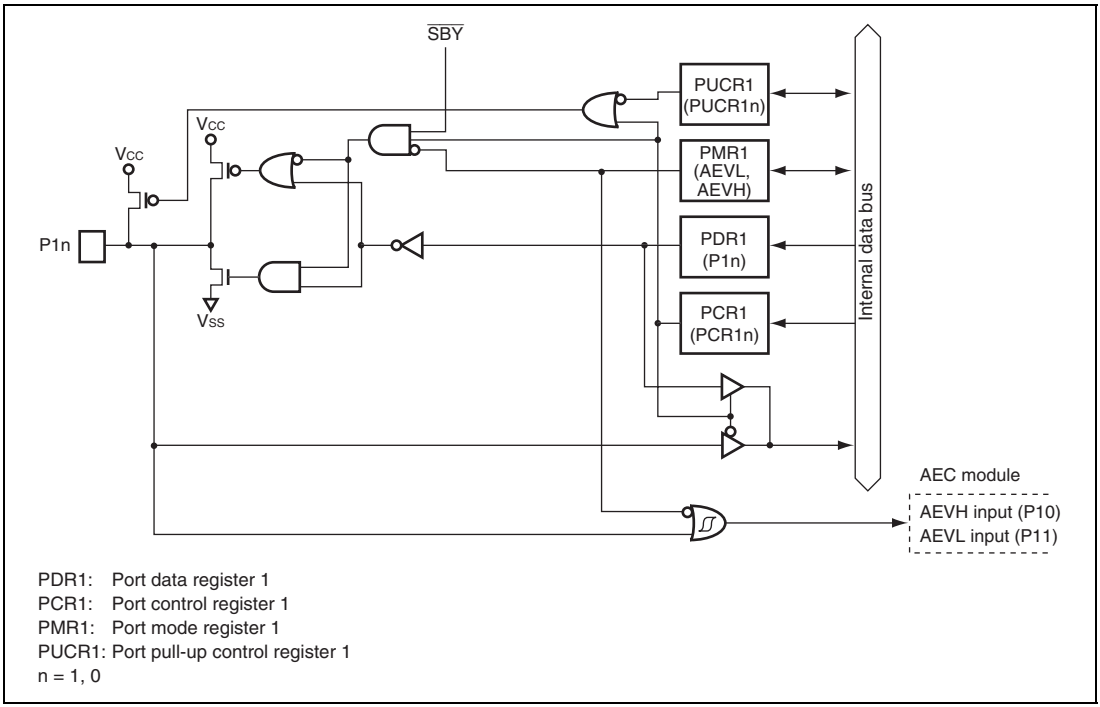


Figure B.1 (c) Port 1 Block Diagram (P15 to P12)



PDR1: Port data register 1
PCR1: Port control register 1
PMR1: Port mode register 1
PUCR1: Port pull-up control register 1
n = 1, 0

Figure B.1 (d) Port 1 Block Diagram (P11, P10)

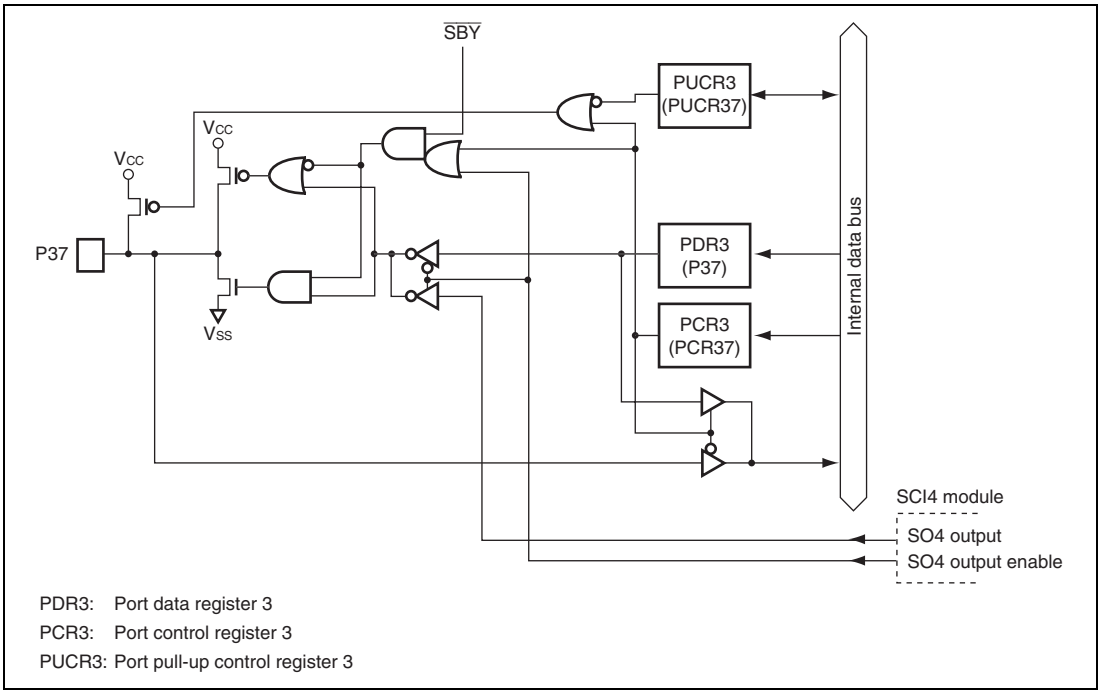


Figure B.2 (a) Port 3 Block Diagram (P37) (F-ZTAT™ Version)

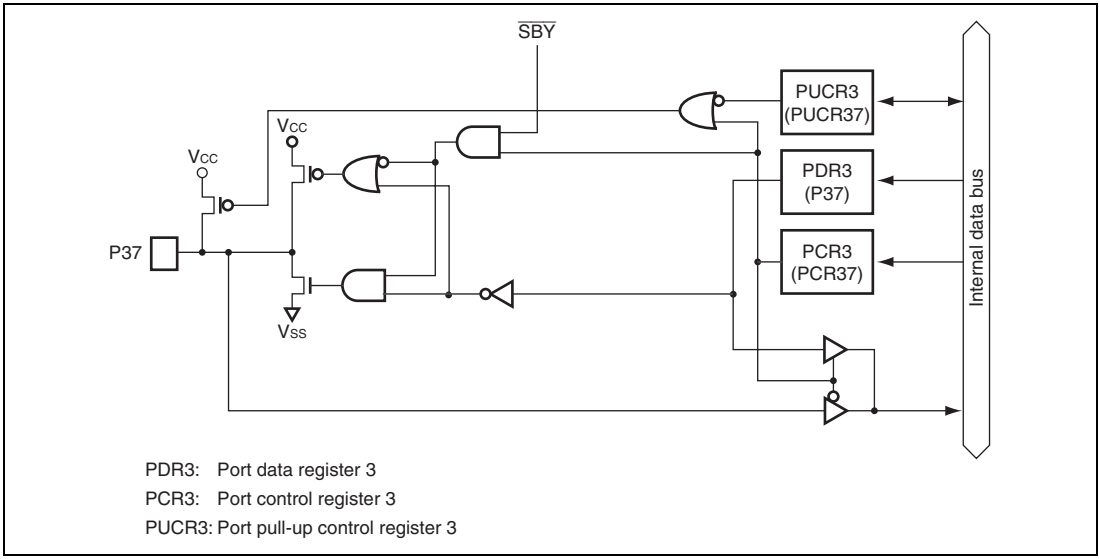


Figure B.2 (b) Port 3 Block Diagram (P37) (Masked ROM Version)

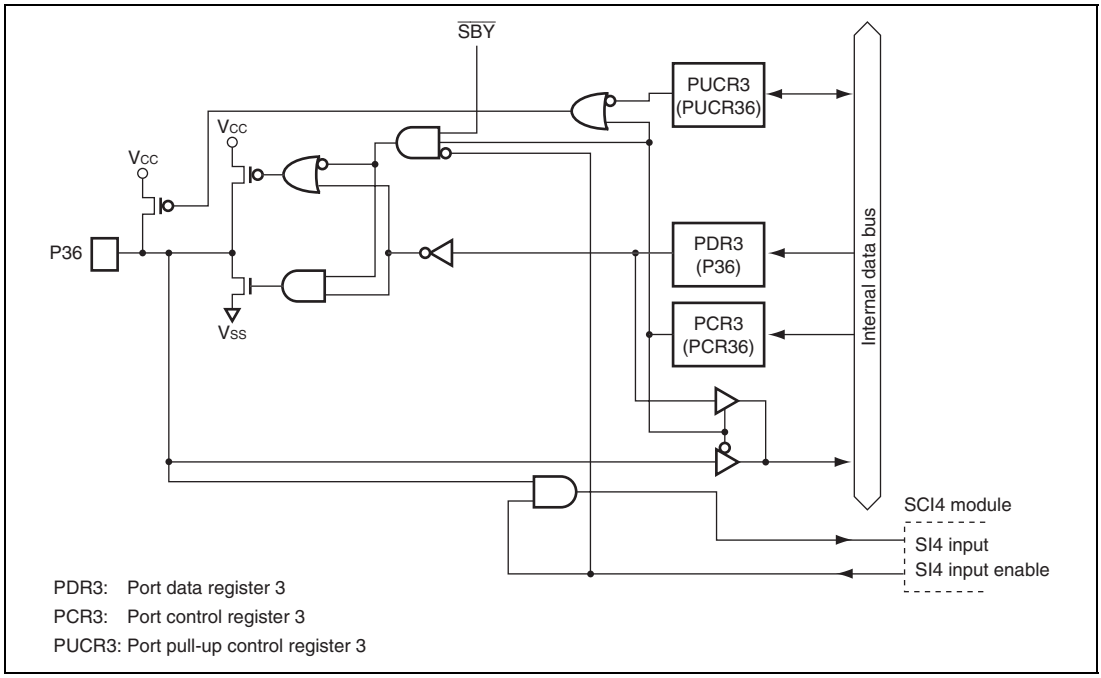


Figure B.2 (c) Port 3 Block Diagram (P36) (F-ZTAT™ Version)

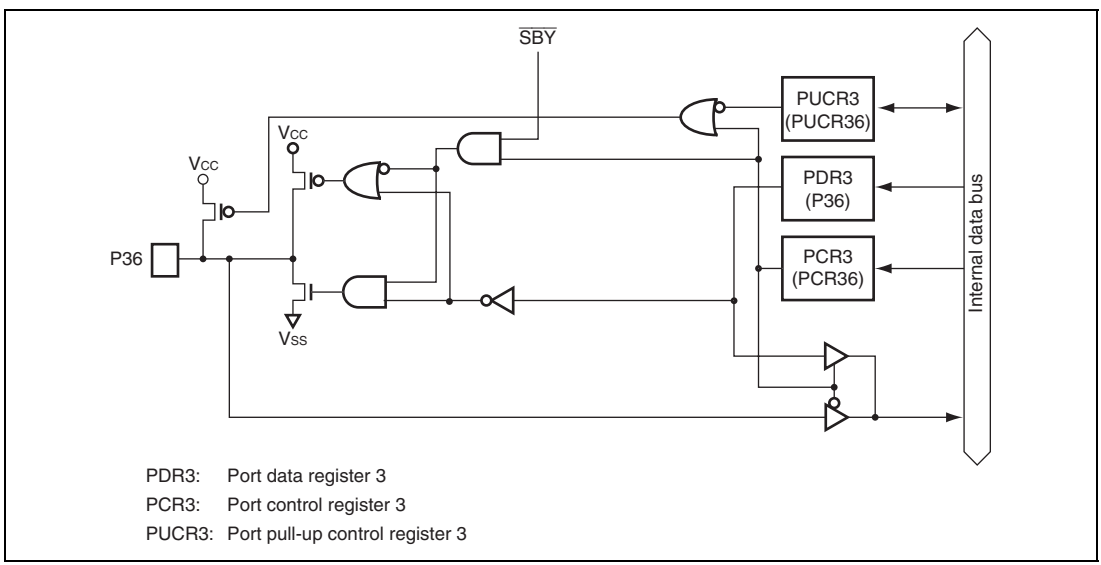


Figure B.2 (d) Port 3 Block Diagram (P36) (Masked ROM Version)

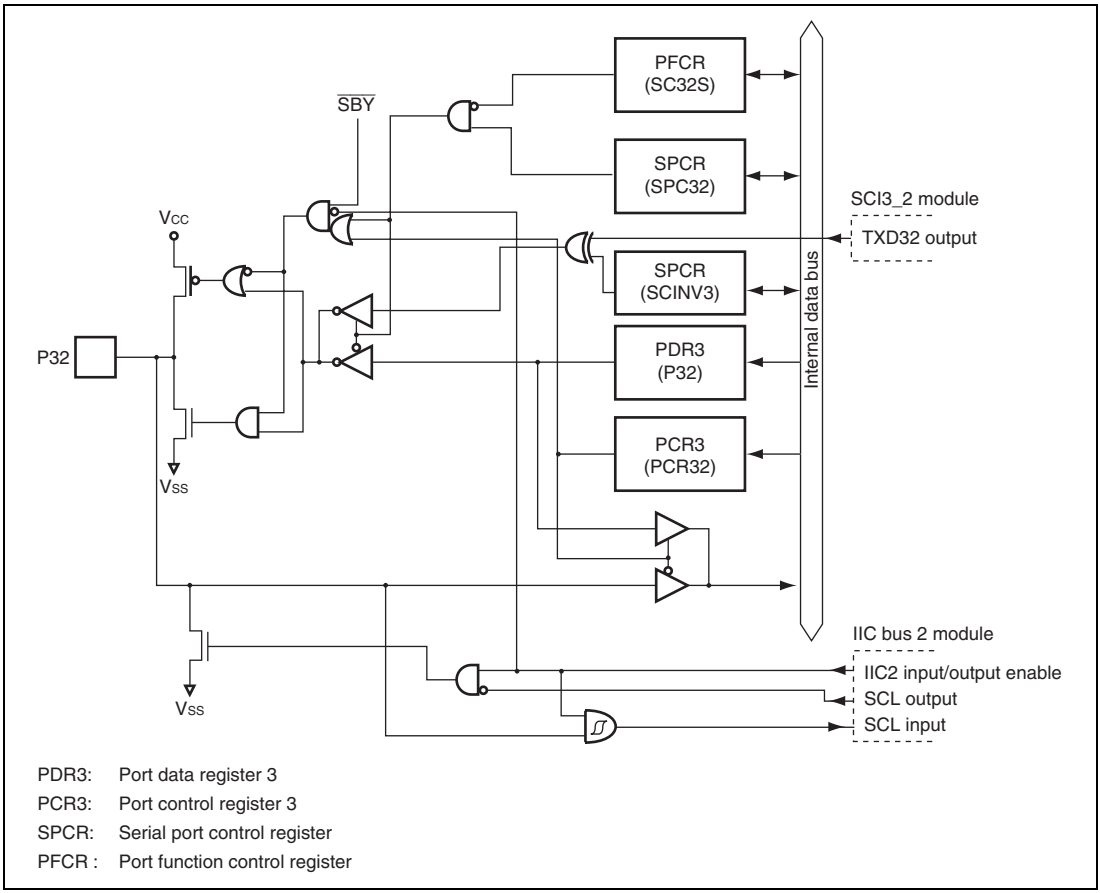


Figure B.2 (e) Port 3 Block Diagram (P32)

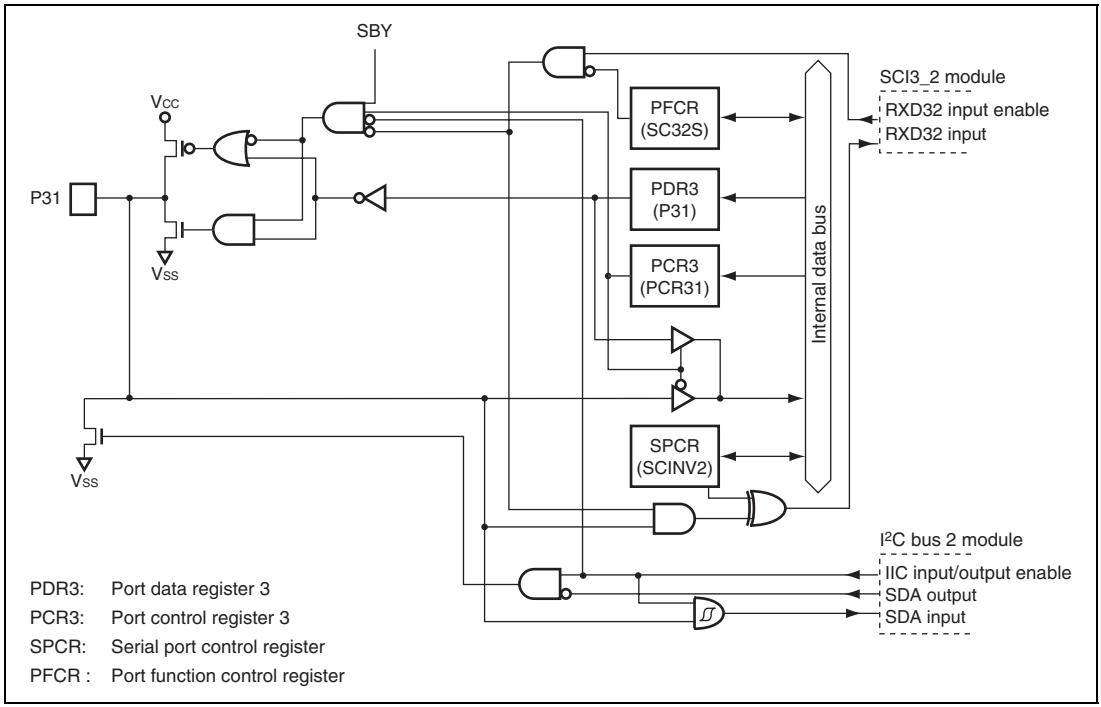


Figure B.2 (f) Port 3 Block Diagram (P31)

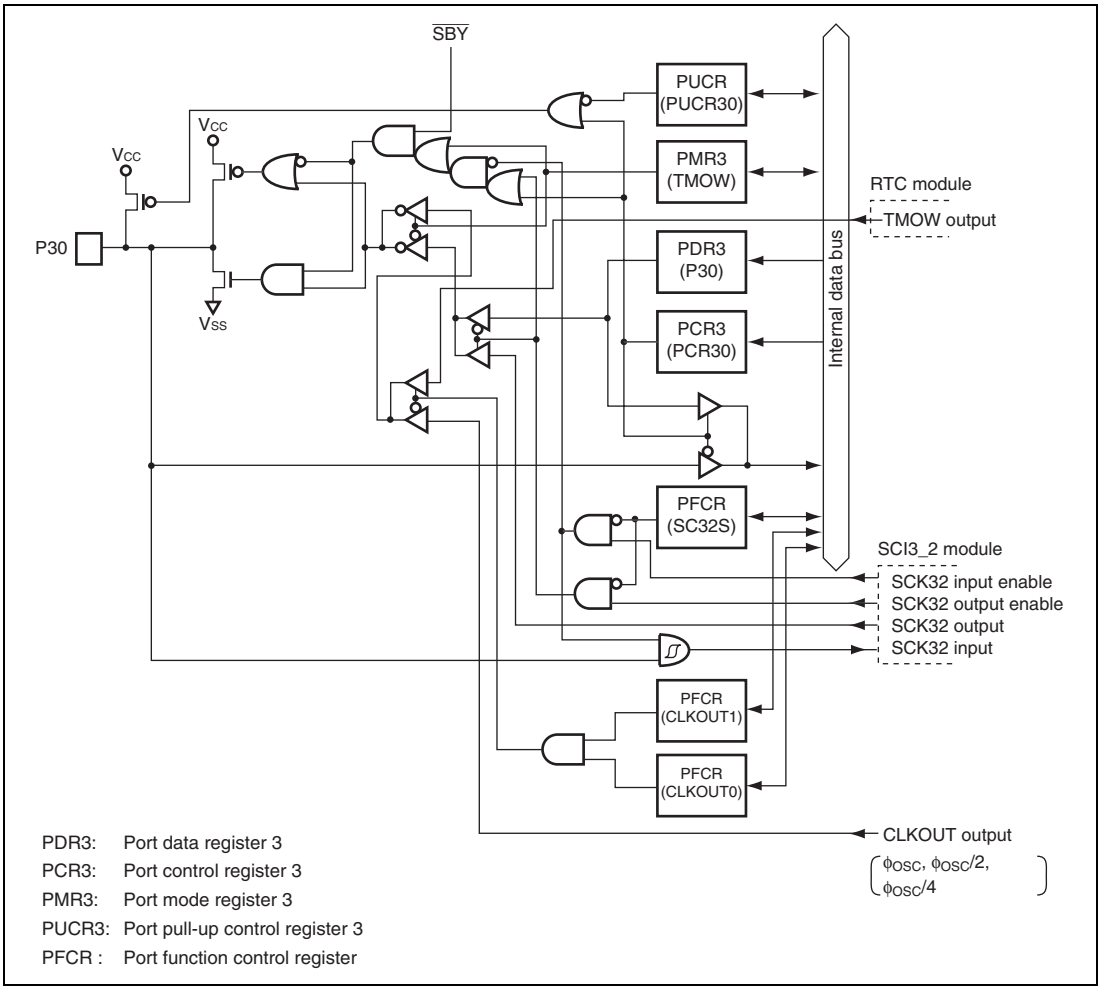


Figure B.2 (g) Port 3 Block Diagram (P30)

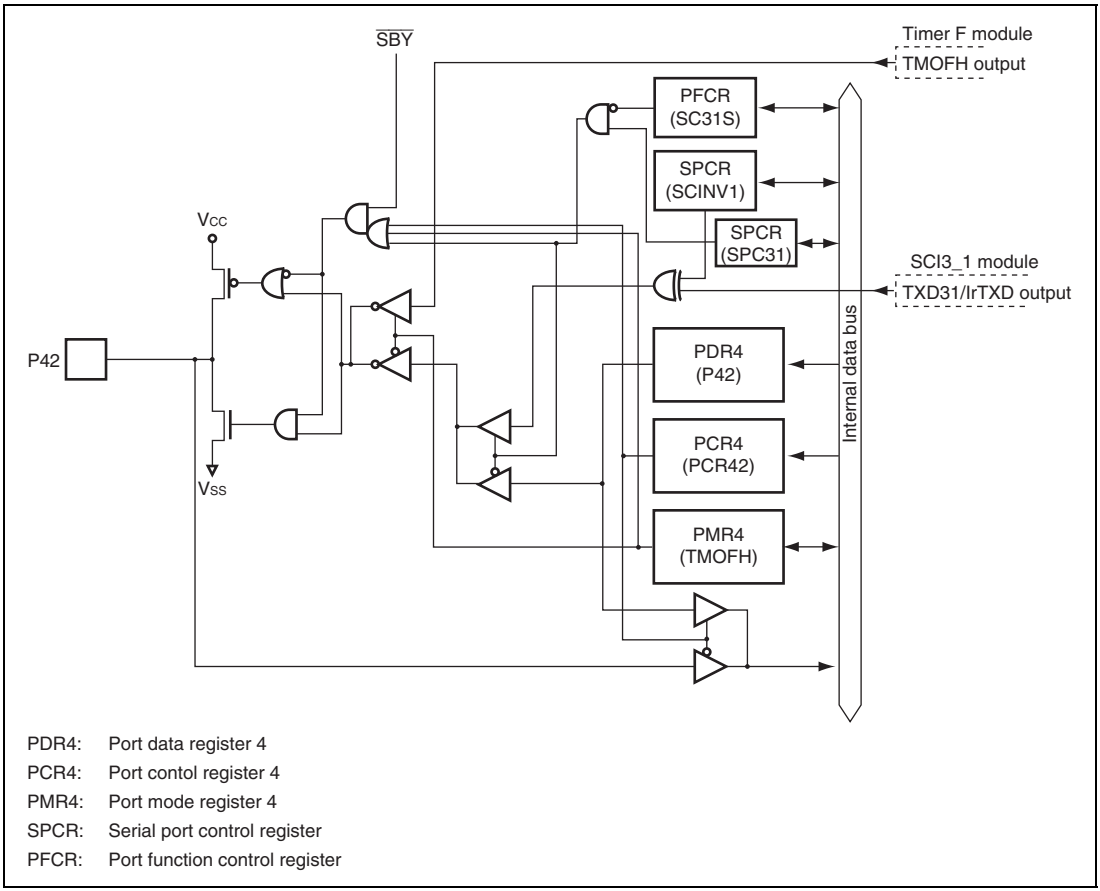


Figure B.3 (a) Port 4 Block Diagram (P42)

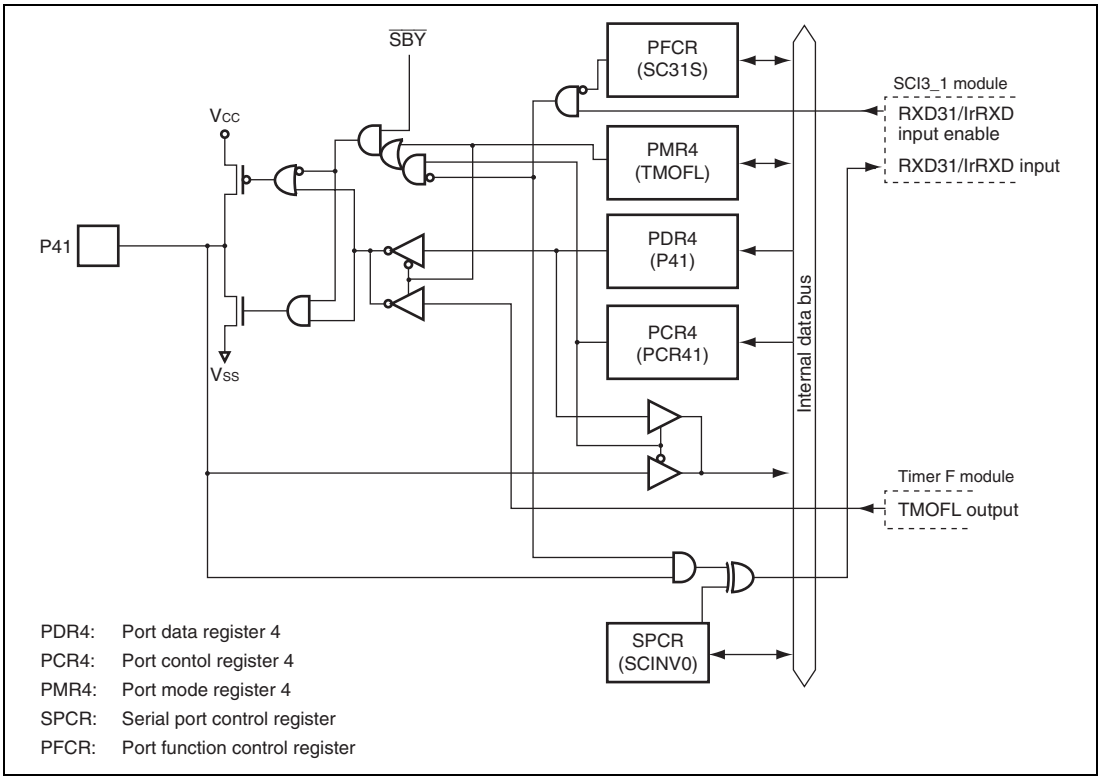


Figure B.3 (b) Port 4 Block Diagram (P41)

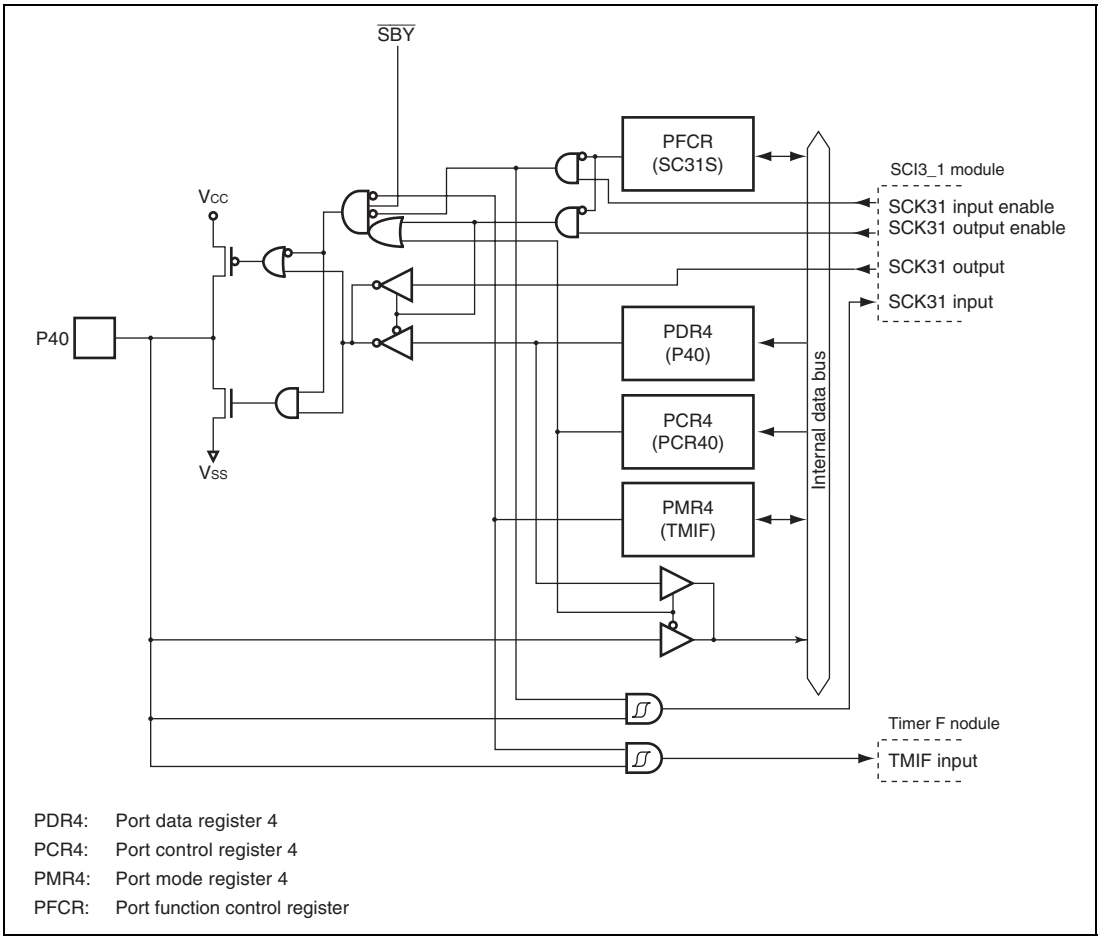


Figure B.3 (c) Port 4 Block Diagram (P40)

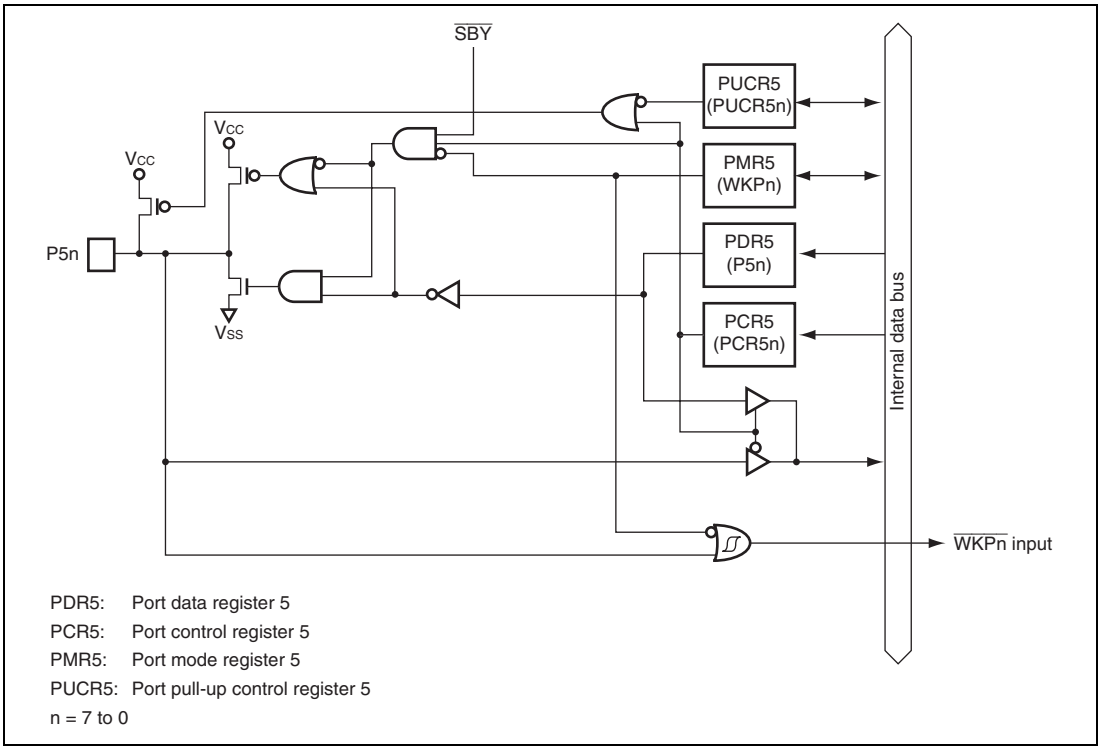
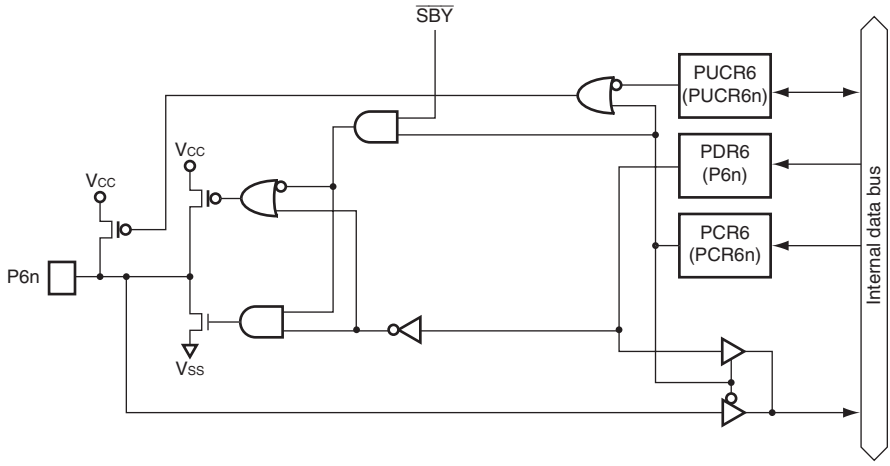
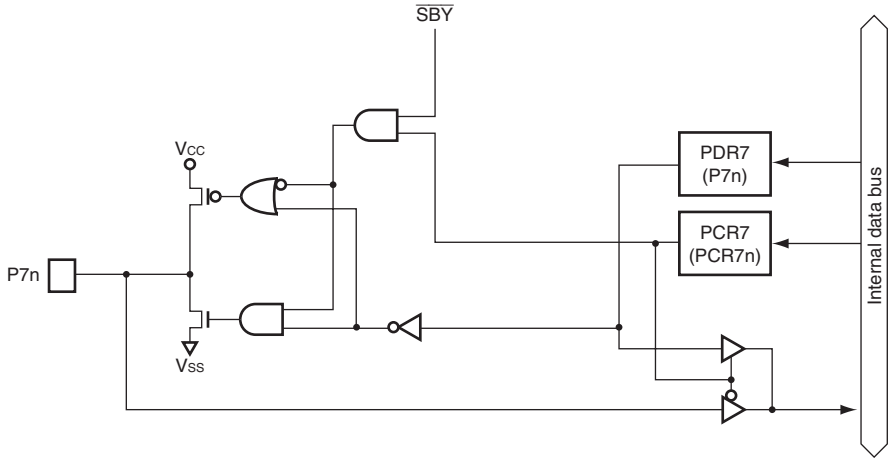


Figure B.4 Port 5 Block Diagram



PDR6: Port data register 6
PCR6: Port control register6
PUCR6: Port pull-up control register 6
n = 7 to 0

Figure B.5 Port 6 Block Diagram



PDR7: Port data register 7
PCR7: Port control register 7
n = 7 to 0

Figure B.6 Port 7 Block Diagram

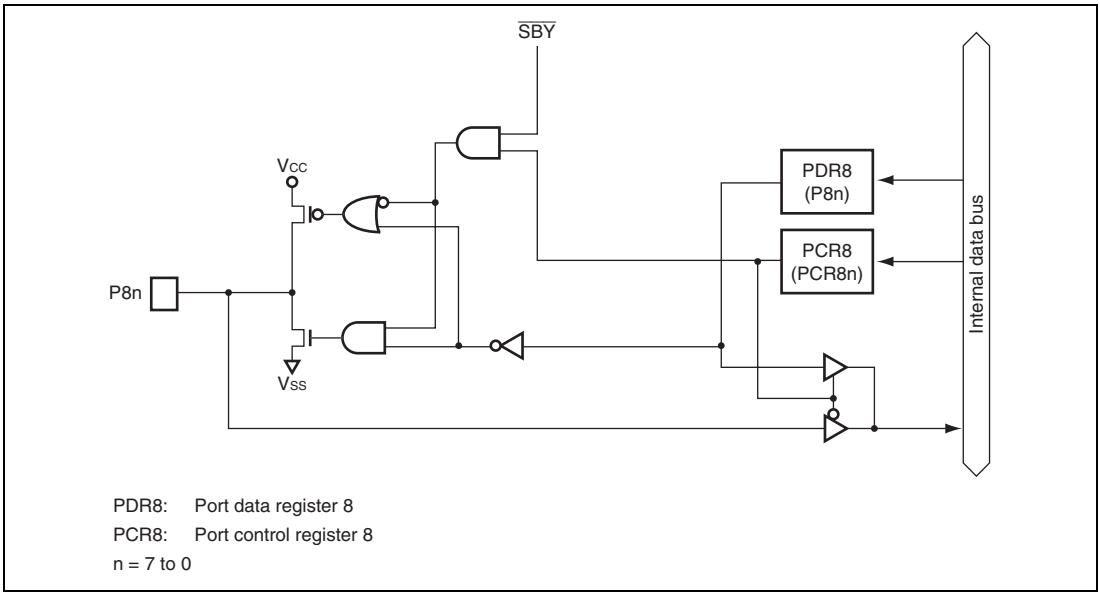


Figure B.7 Port 8 Block Diagram

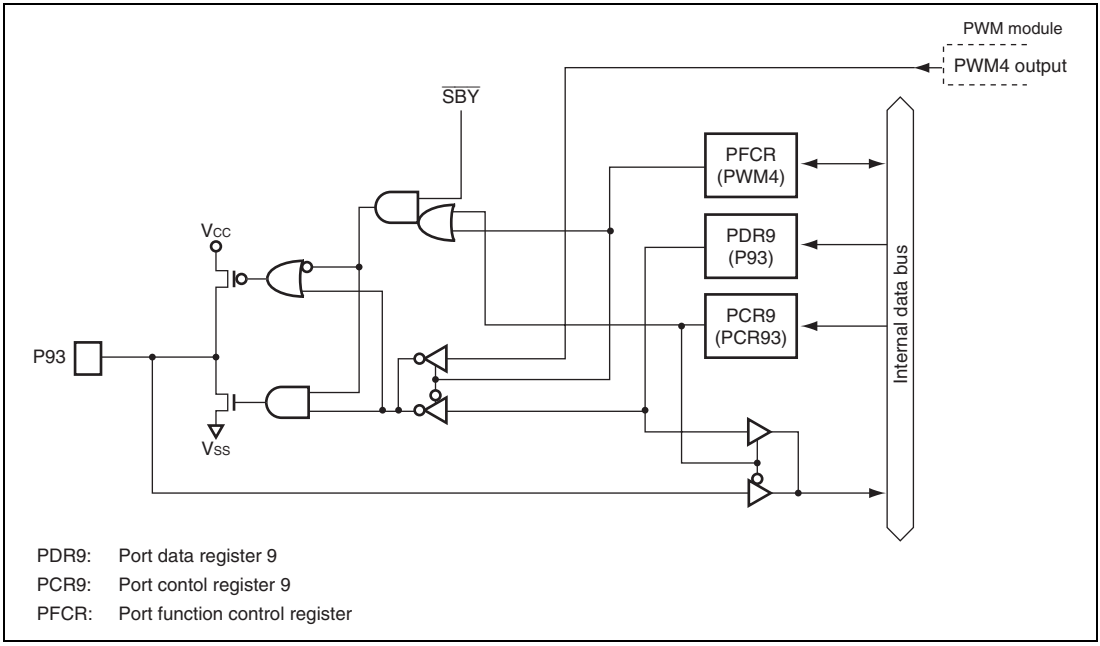


Figure B.8 (a) Port 9 Block Diagram (P93)

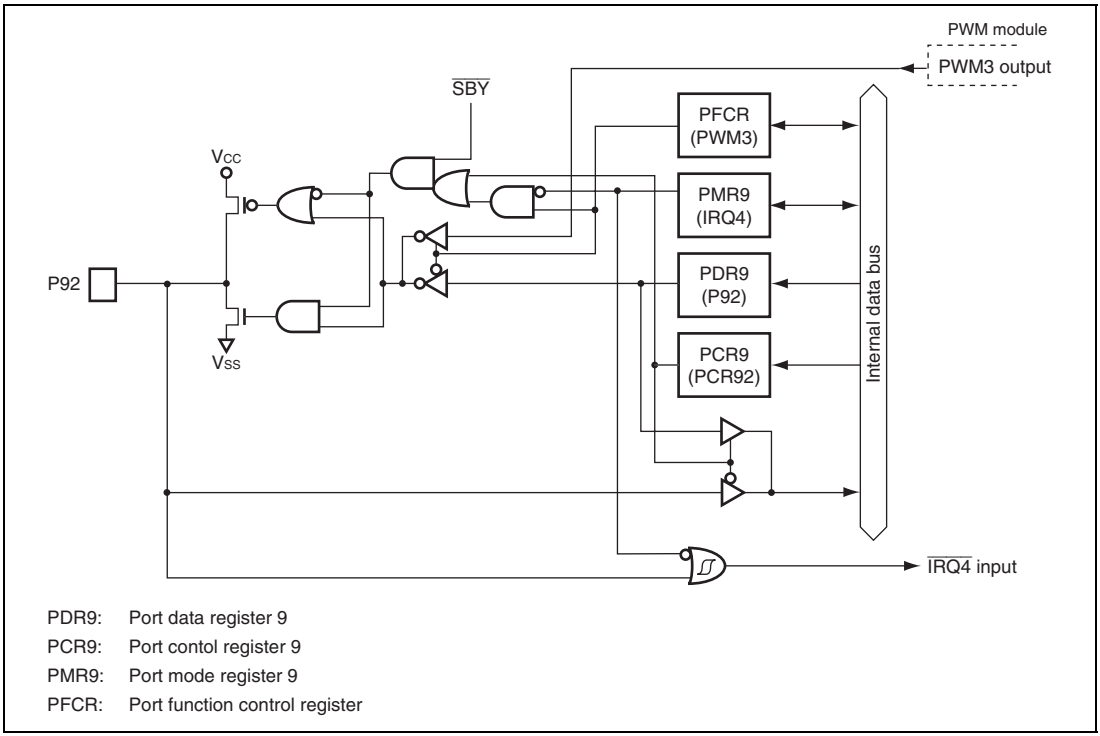


Figure B.8 (b) Port 9 Block Diagram (P92)

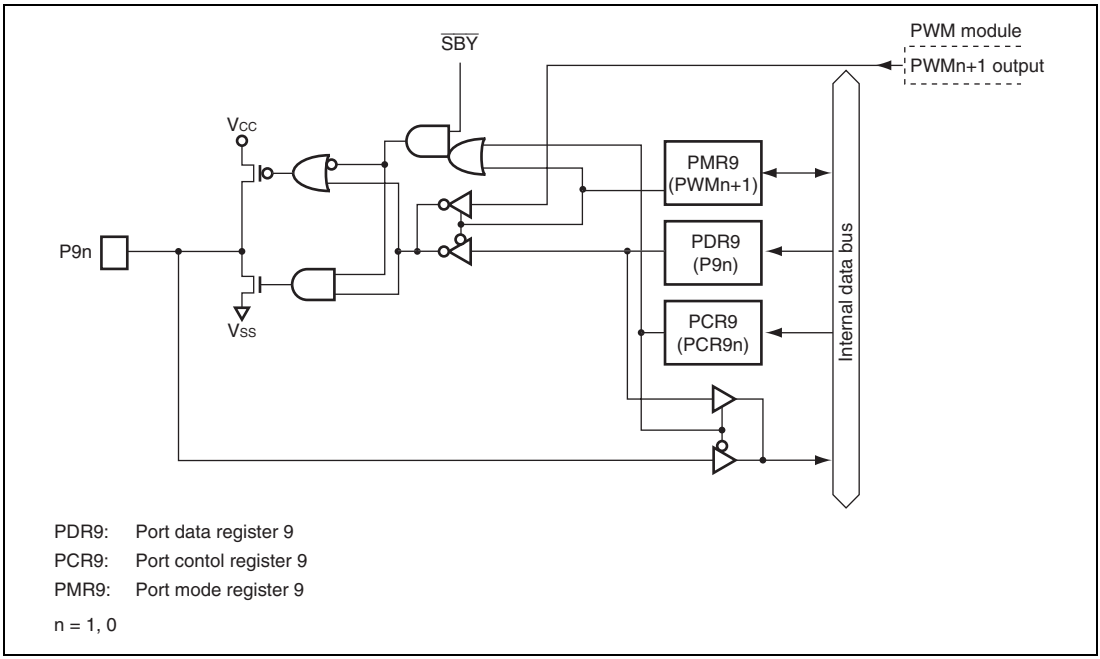


Figure B.8 (c) Port 9 Block Diagram (P91, P90)

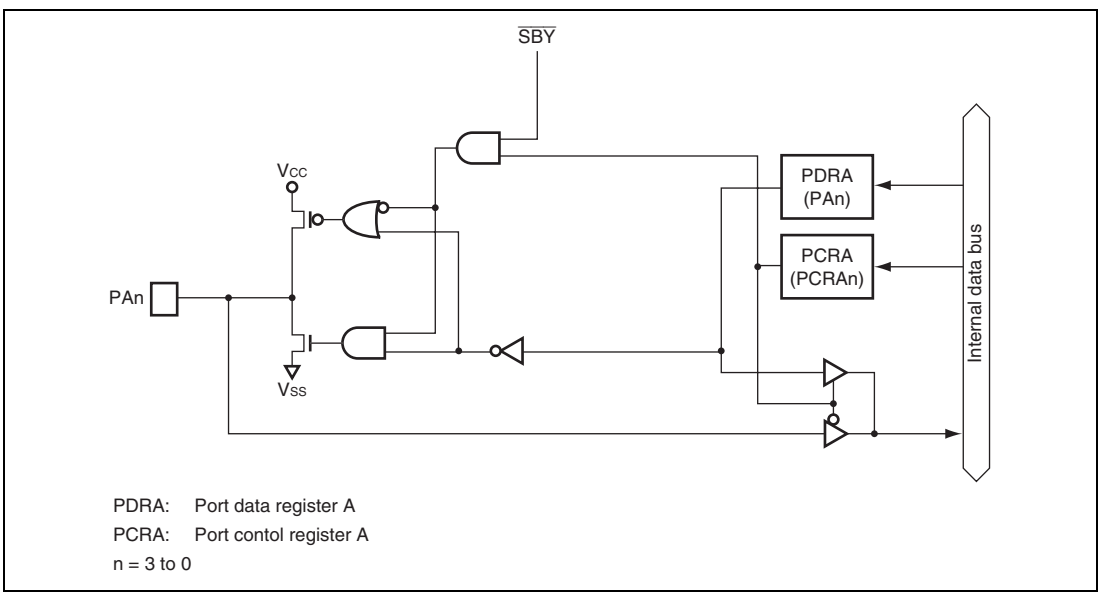


Figure B.9 Port A Block Diagram

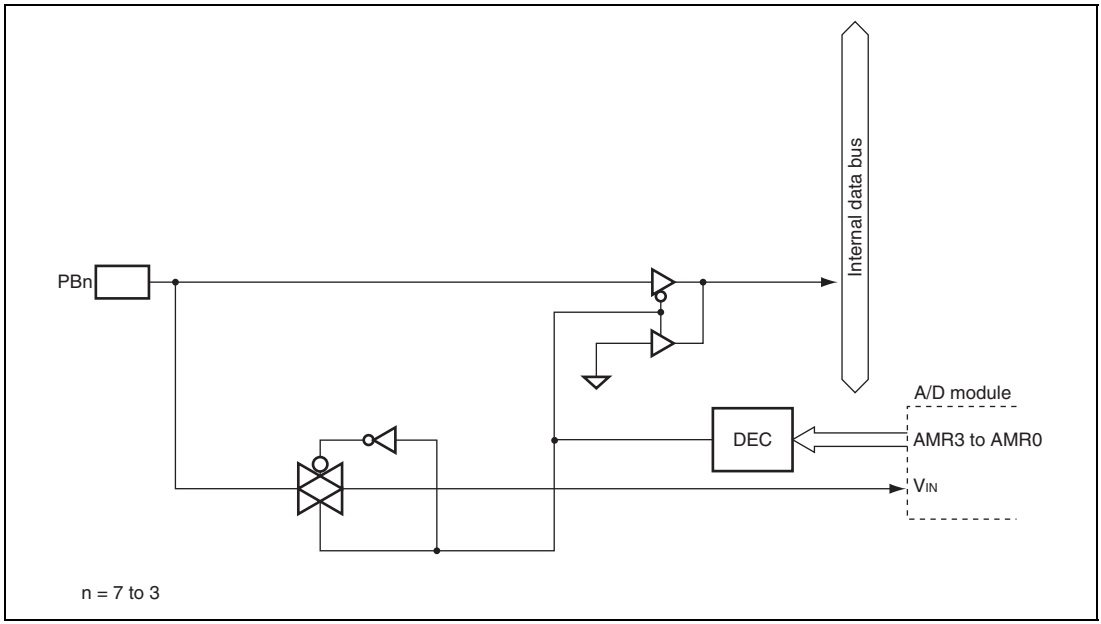


Figure B.10 (a) Port B Block Diagram (PB7 to PB3)

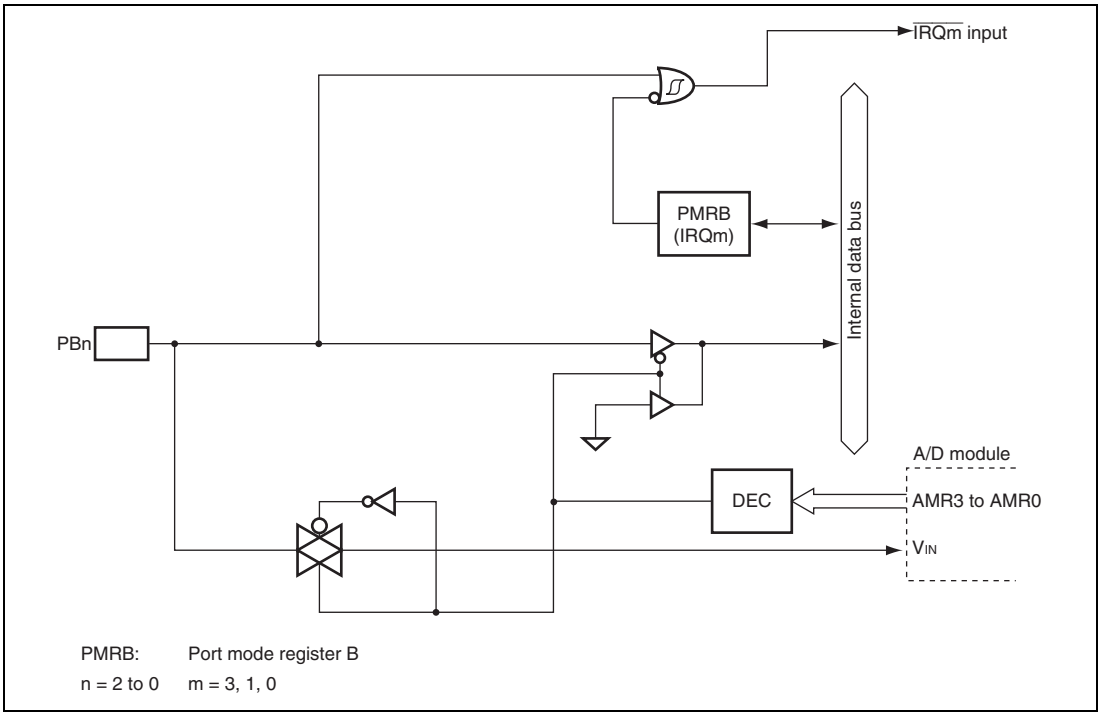


Figure B.10 (b) Port B Block Diagram (PB2 to PB0)

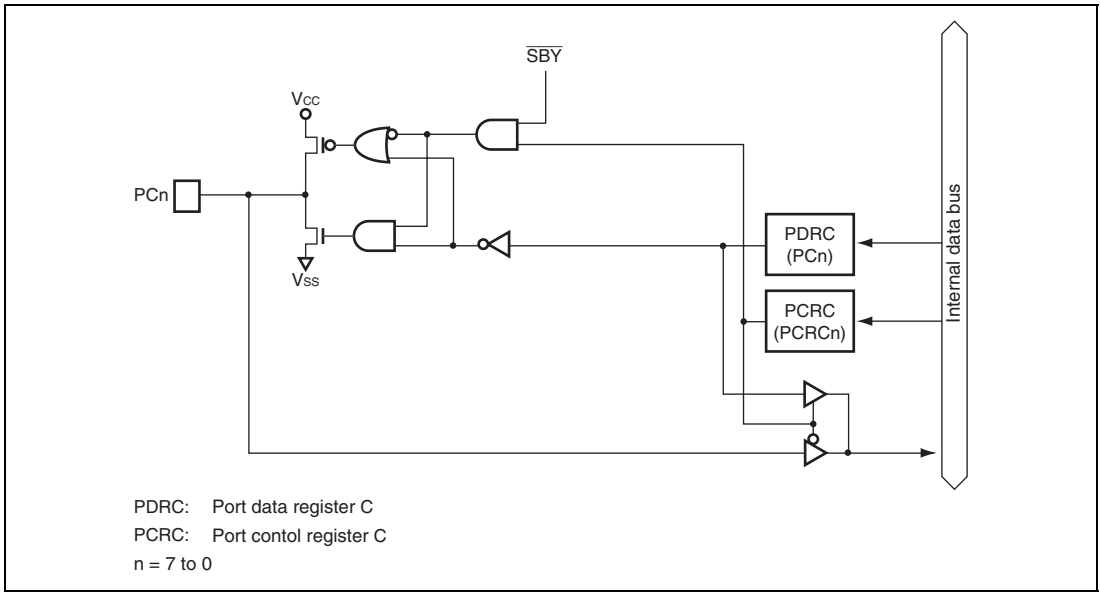


Figure B.11 Port C Block Diagram (PC7 to PC0)

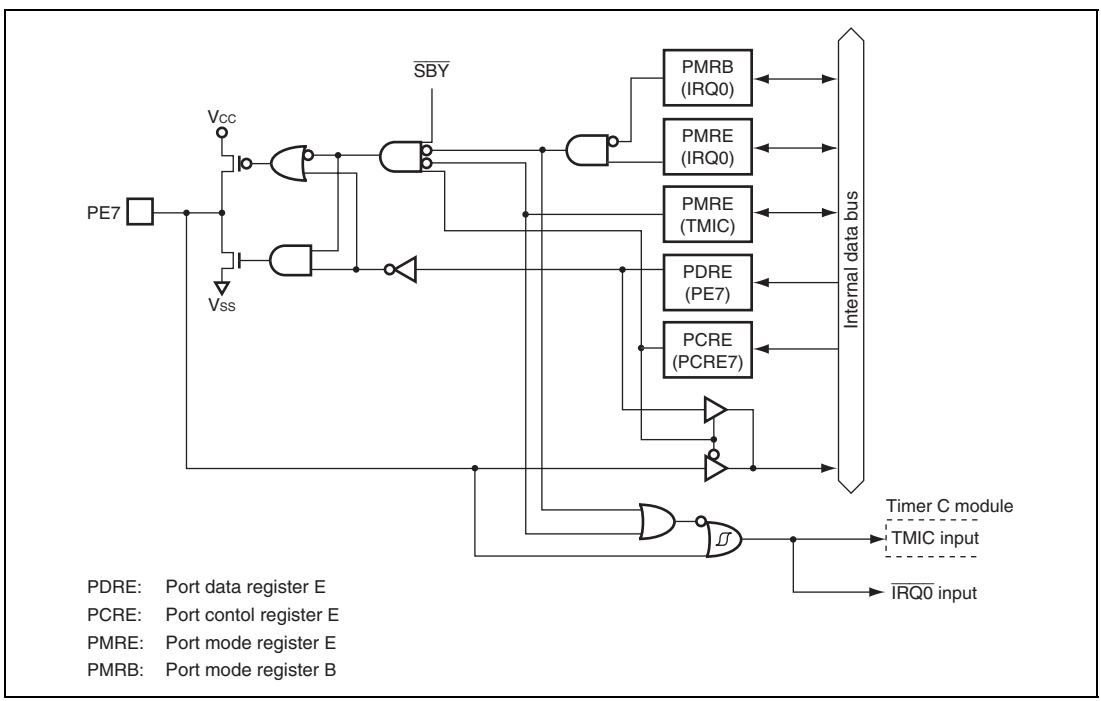


Figure B.12 (a) Port E Block Diagram (PE7)

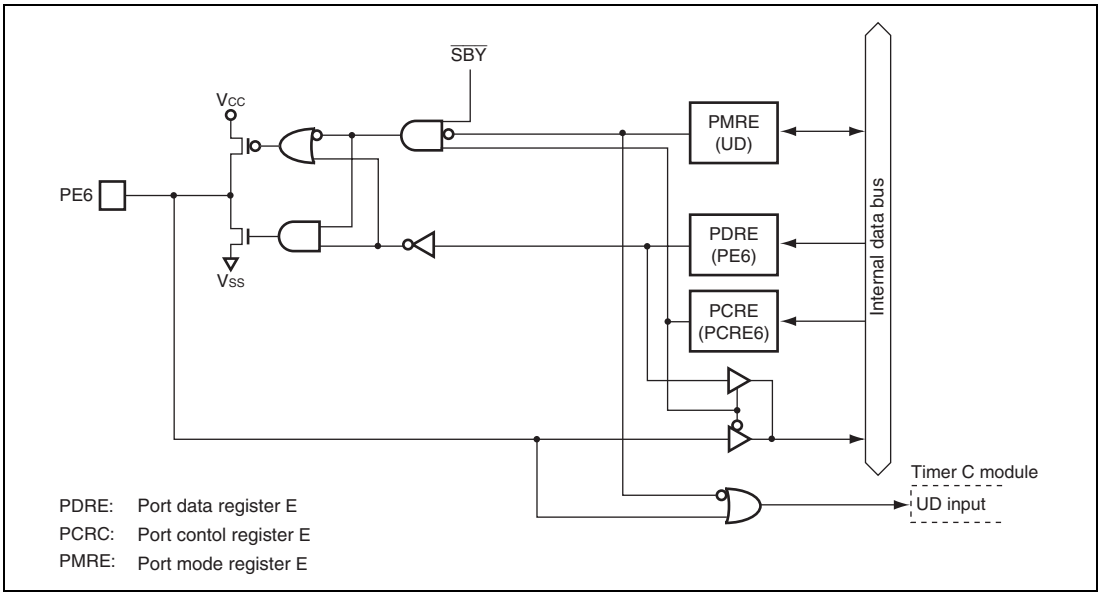


Figure B.12 (b) Port E Block Diagram (PE6)

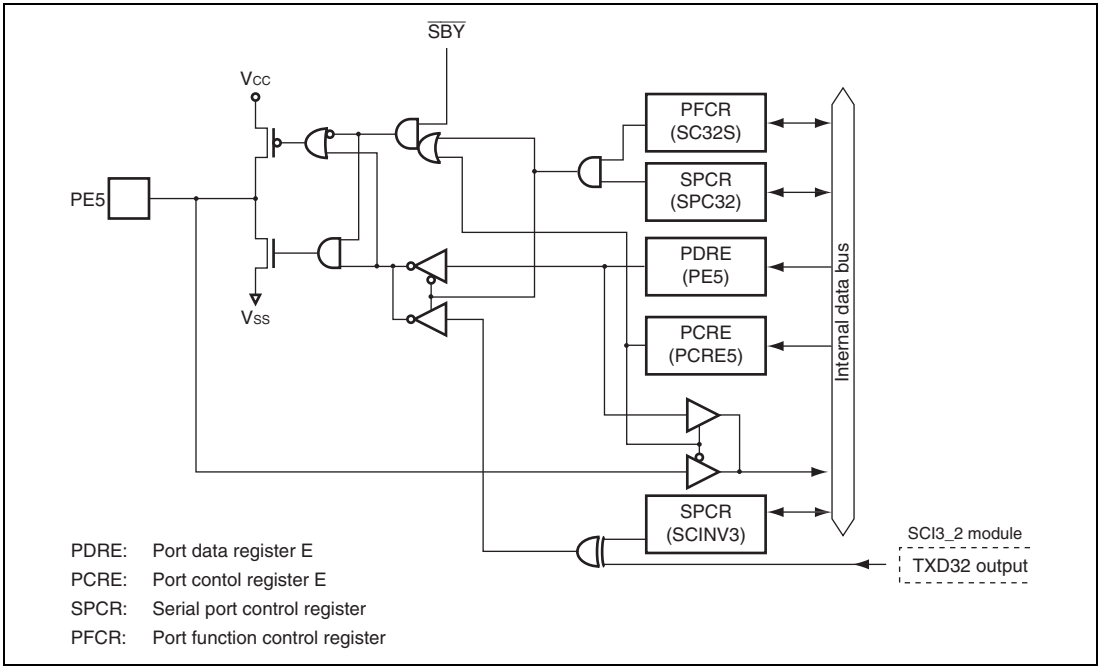


Figure B.12 (c) Port E Block Diagram (PE5)

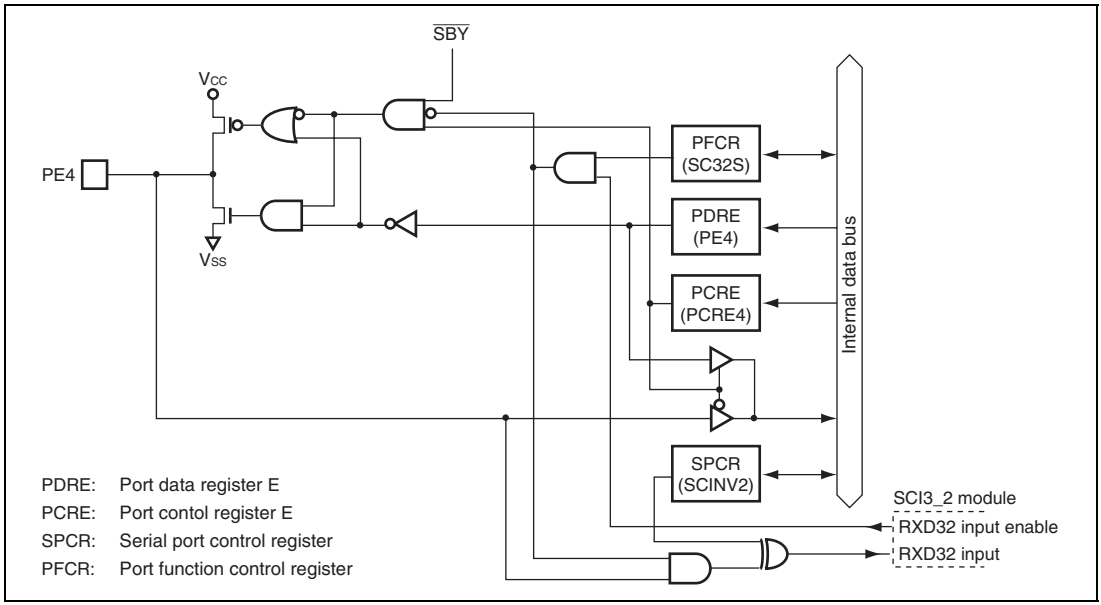


Figure B.12 (d) Port E Block Diagram (PE4)

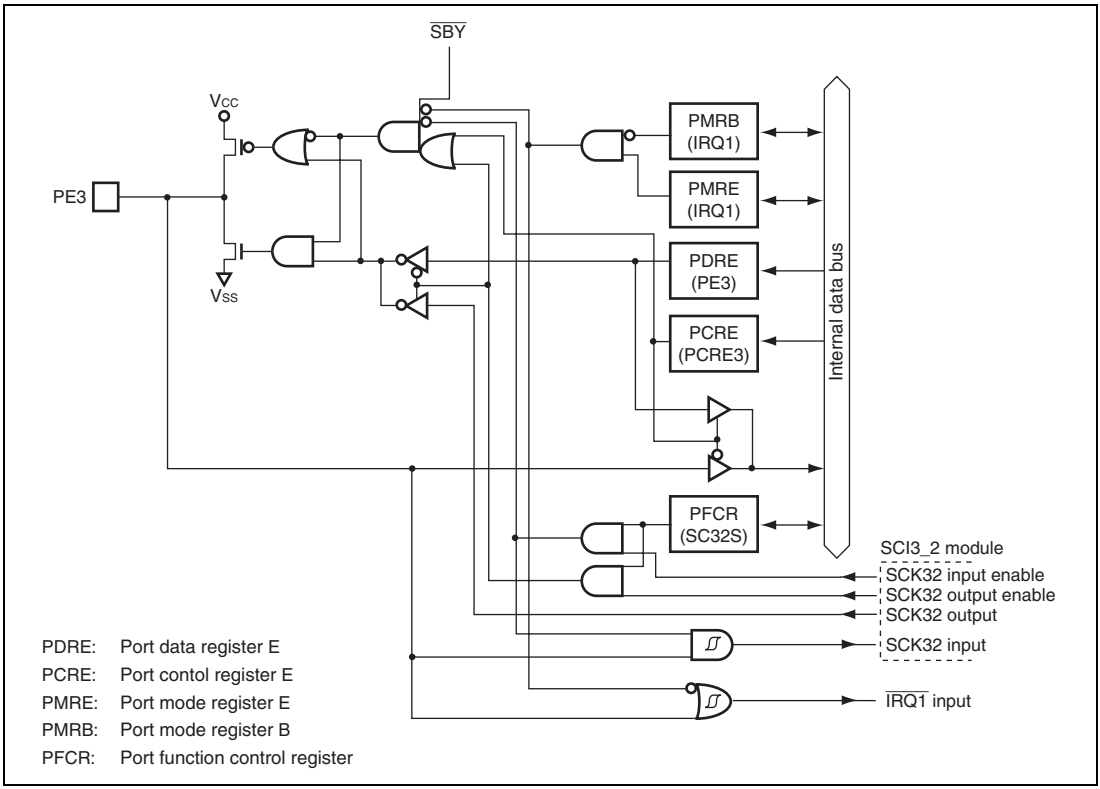


Figure B.12 (e) Port E Block Diagram (PE3)

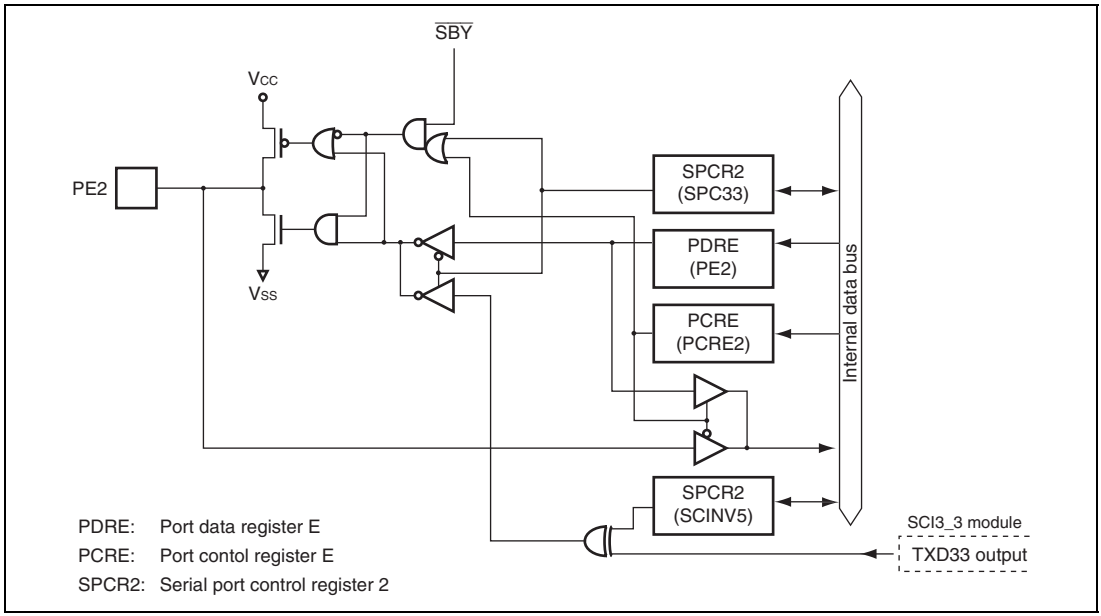


Figure B.12 (f) Port E Block Diagram (PE2)

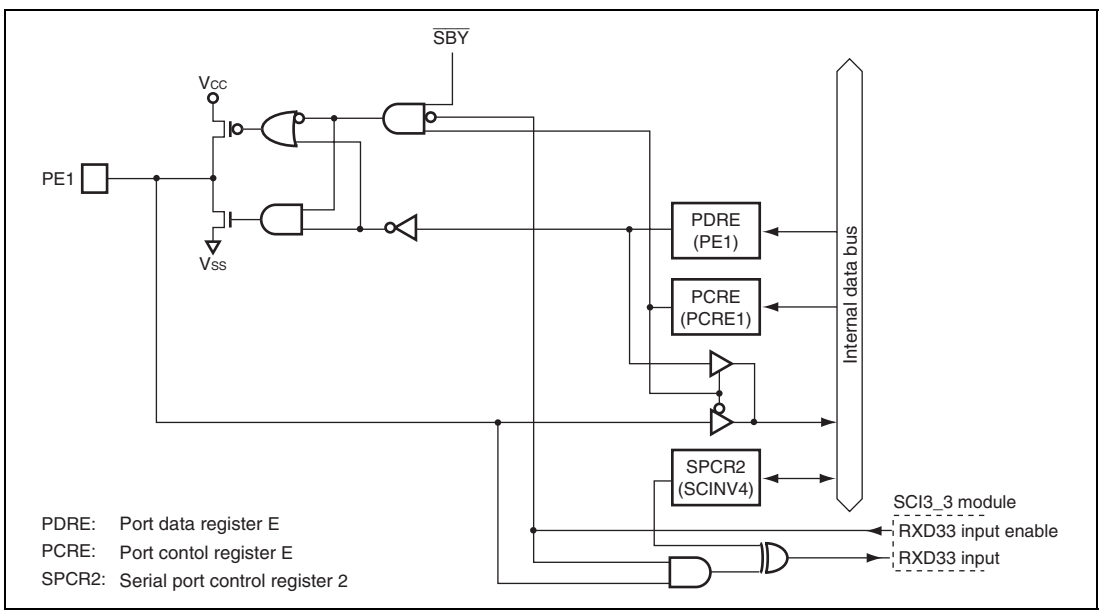


Figure B.12 (g) Port E Block Diagram (PE1)

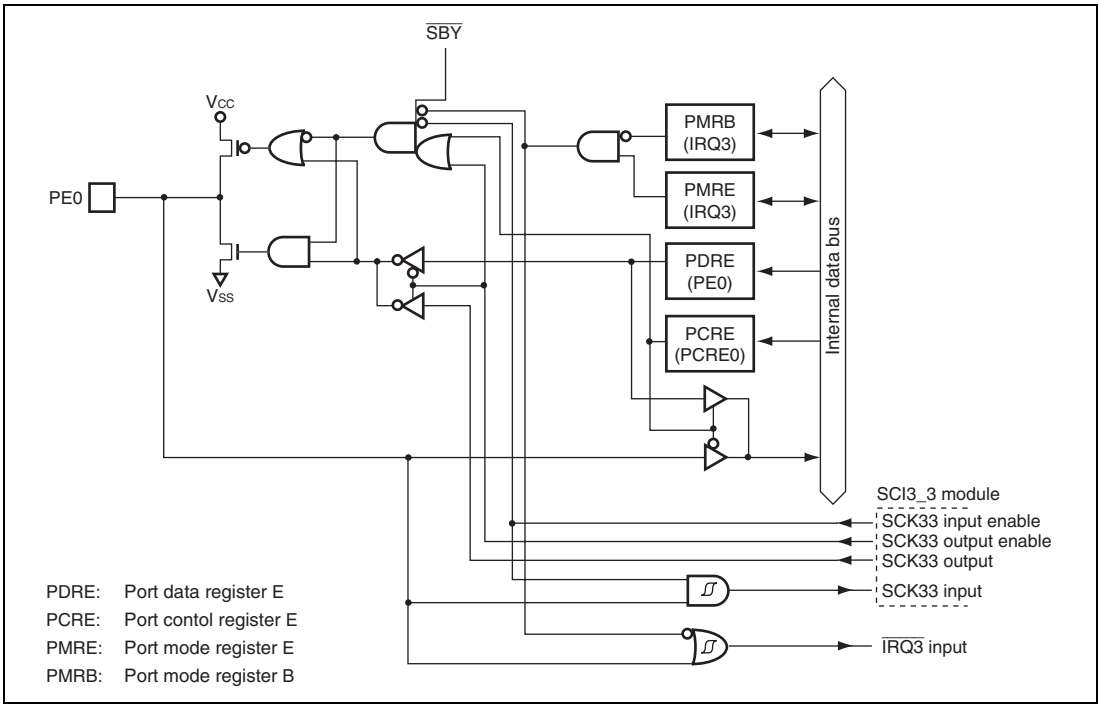


Figure B.12 (h) Port E Block Diagram (PE0)

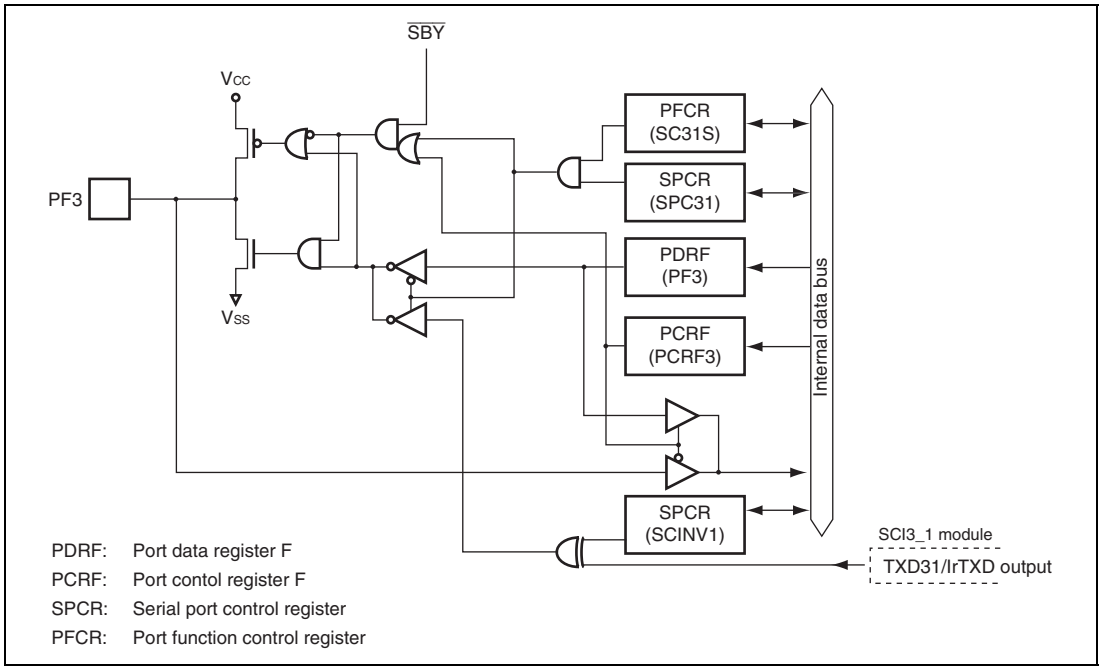


Figure B.13 (a) Port F Block Diagram (PF3)

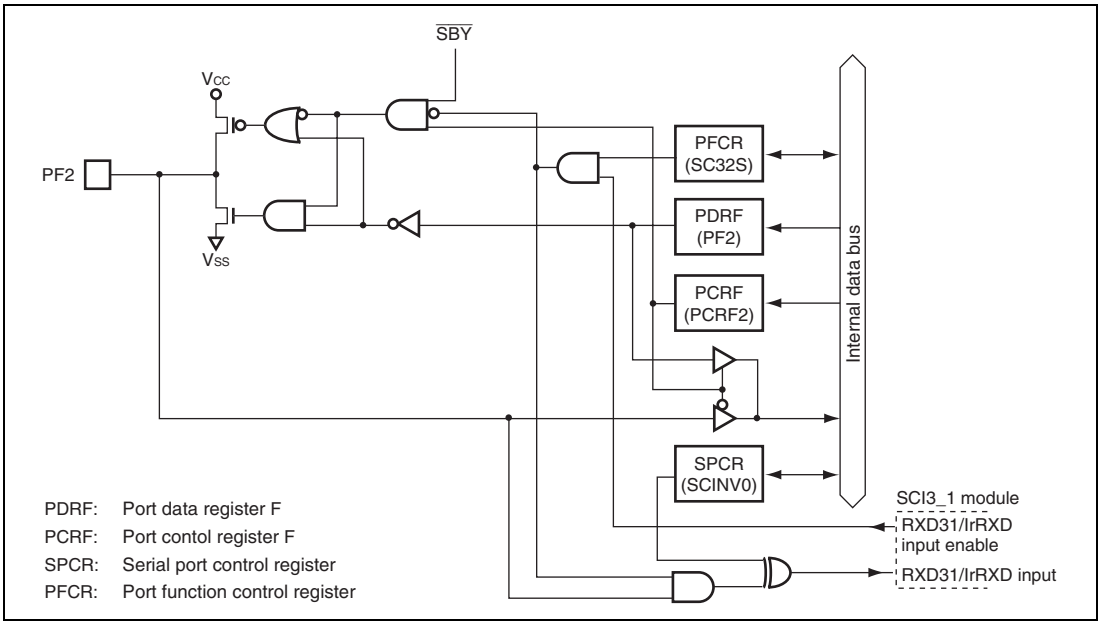


Figure B.13 (b) Port F Block Diagram (PF2)

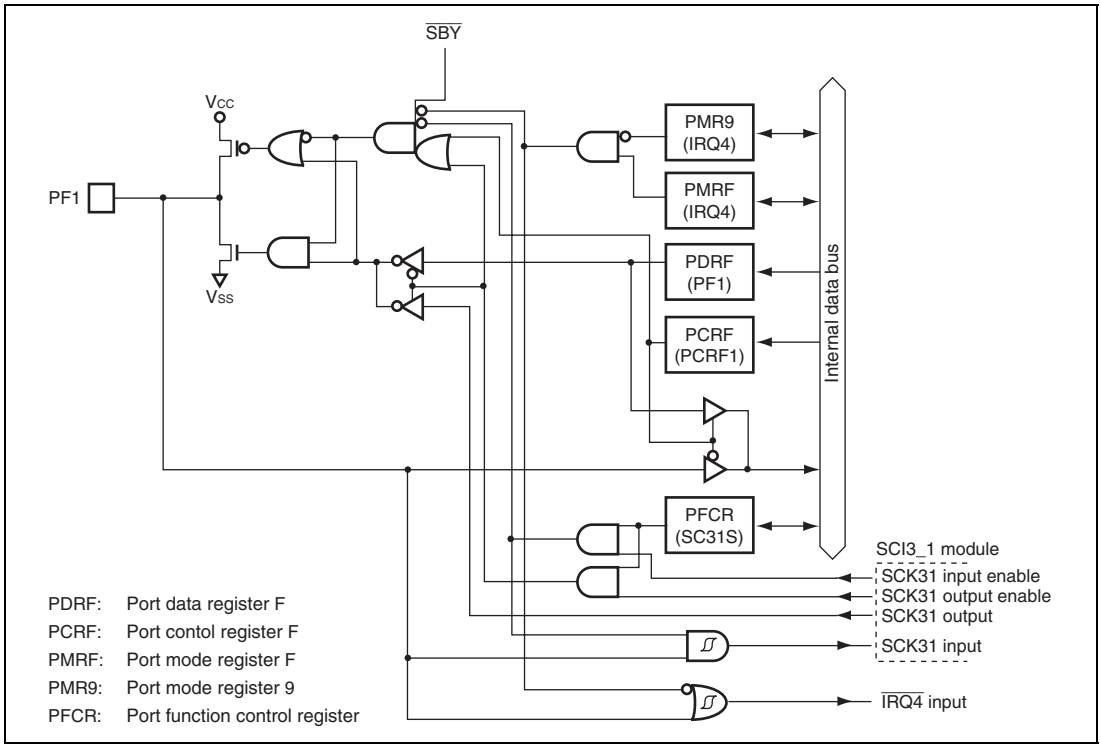


Figure B.13 (c) Port F Block Diagram (PF1)

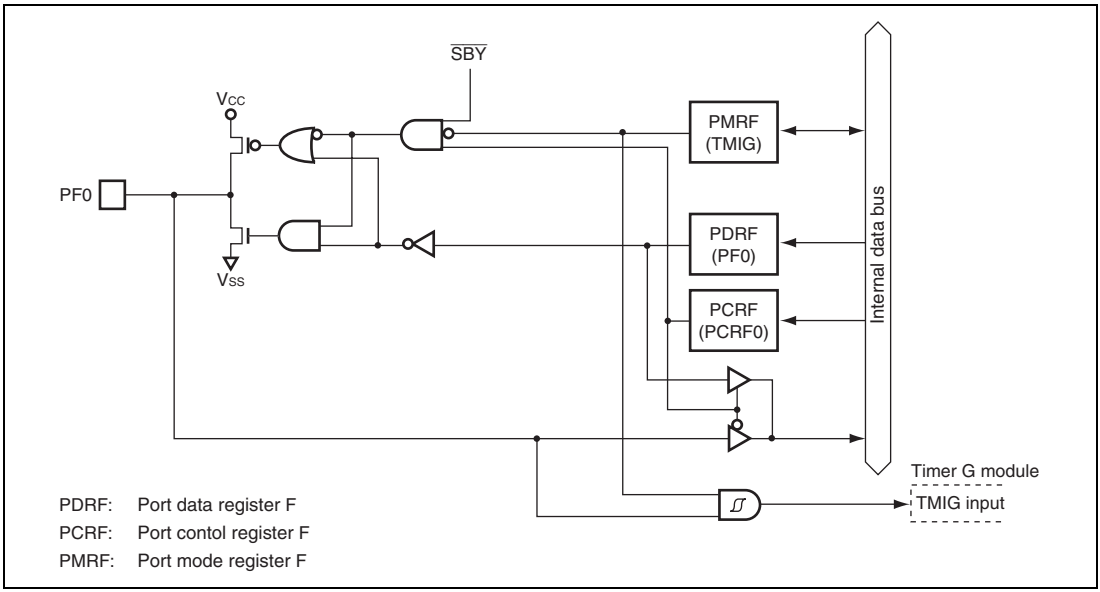


Figure B.13 (d) Port F Block Diagram (PF0)

B.2 Port States in Each Operating State

Operating Mode	Reset	Active (High-Speed/ Medium-Speed)	Sleep (High-Speed/ Medium-Speed)	Watch	Subactive	Subsleep	Standby
P16 to P10	High impedance	Functioning	Retained	Retained	Functioning	Retained	High impedance *1*2
P37, P36, P30	High impedance	Functioning	Retained	Retained	Functioning	Retained	High impedance *1
P32, P31	High impedance	Functioning	Retained	Retained	Functioning	Retained	High impedance *1*2
P42 to P40	High impedance	Functioning	Retained	Retained	Functioning	Retained	High impedance *1
P57 to P50	High impedance	Functioning	Retained	Retained	Functioning	Retained	High impedance *1*2
P67 to P60	High impedance	Functioning	Retained	Retained	Functioning	Retained	High impedance *1*2
P77 to P70	High impedance	Functioning	Retained	Retained	Functioning	Retained	High impedance *1
P87 to P80	High impedance	Functioning	Retained	Retained	Functioning	Retained	High impedance *1
P93 to P90	High impedance	Functioning	Retained	Retained	Functioning	Retained	High impedance *1
PA3 to PA0	High impedance	Functioning	Retained	Retained	Functioning	Retained	High impedance *1

Operating Mode	Reset	Active (High-Speed/ Medium-Speed)	Sleep (High-Speed/ Medium-Speed)	Watch	Subactive	Subsleep	Standby
PB7 to PB0	High impedance	High impedance*3	High impedance*3	High impedance	High impedance *3	High impedance *3	High impedance *1
PC7 to PC0	High impedance	Functioning	Retained	Retained	Functioning	Retained	High impedance *1
PE7 to PE0	High impedance	Functioning	Retained	Retained	Functioning	Retained	High impedance *1
PF3 to PF0	High impedance	Functioning	Retained	Retained	Functioning	Retained	High impedance *1

Notes: 1. Registers are retained and output level is high impedance.
2. High-level output when the pull-up MOS is turned on.
3. The pin selected by the CH3 to CH0 bits in AMR is connected to the A/D converter.

C. Product Part No. Lineup

Product Classification			Product Part No.	Model Marking	Package (Package Code)
H8/38799	Flash memory version	Regular specifications	HD64F38799FP4	F38799FP4	100 pin LQFP
			HD64F38799FP10	F38799FP10	(PLQP0100KB-A)
		Wide-range specifications	HD64F38799FP10W	F38799FP10	100 pin LQFP (PLQP0100KB-A)
	Masked ROM version	Regular specifications	HD64338799FP	38799(***)FP	100 pin LQFP (PLQP0100KB-A)
		Wide-range specifications	HD64338799FPW	38799(***)FP	100 pin LQFP (PLQP0100KB-A)
H8/38798	Masked ROM version	Regular specifications	HD64338798FP	38798(***)FP	100 pin LQFP (PLQP0100KB-A)
		Wide-range specifications	HD64338798FPW	38798(***)FP	100 pin LQFP (PLQP0100KB-A)

[Legend]
(***): ROM code

D. Package Dimensions

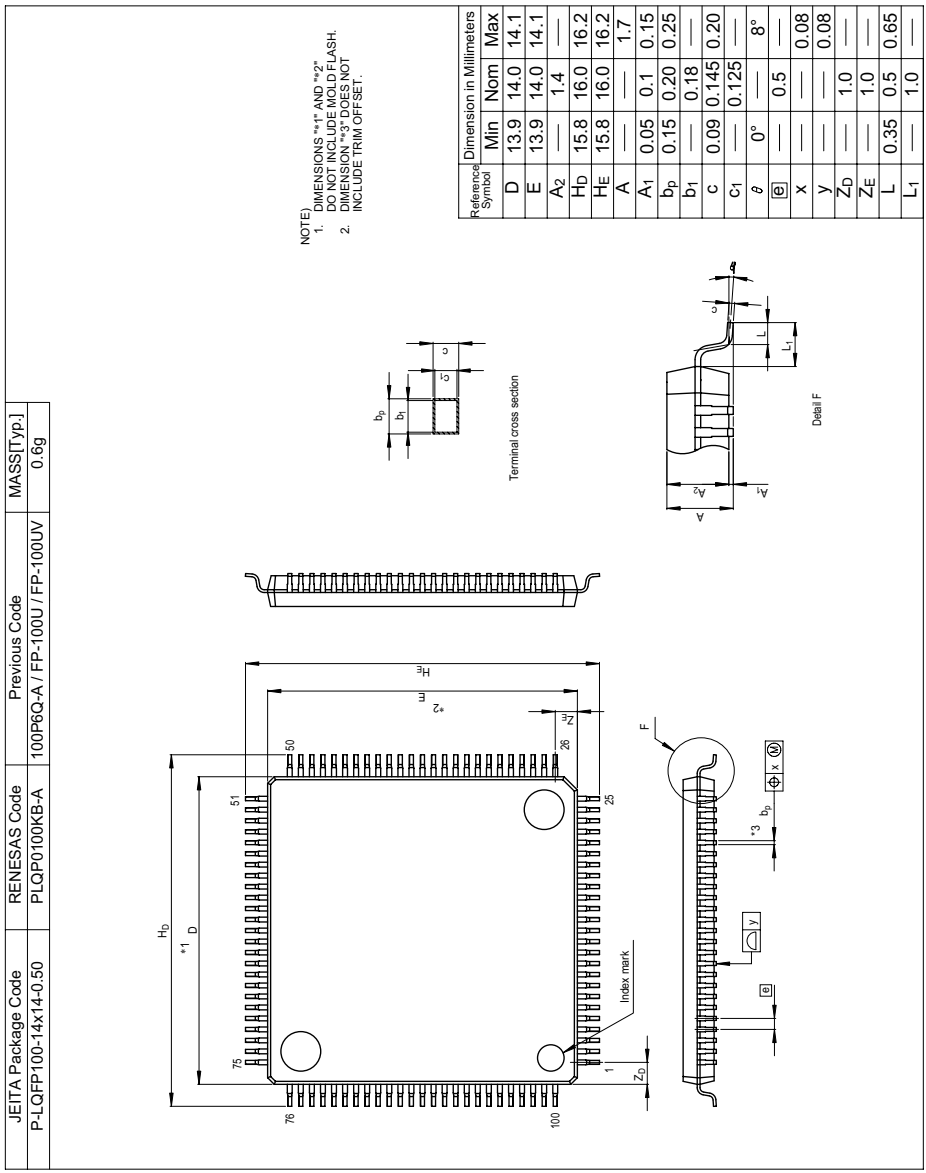


Figure D.1 Package Dimensions (PLQP0100KB-A)

Index

S

T

V

W

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8/38799 Group

Publication Date:	Rev.1.00, Jul. 11, 2007
Published by:	Sales Strategic Planning Div. Renesas Technology Corp.
Edited by:	Customer Support Department Global Strategic Communication Div. Renesas Solutions Corp.

© 2007. Renesas Technology Corp., All rights reserved. Printed in Japan.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

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, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8/38799 Group
Hardware Manual

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

REJ09B0380-0100