REJ09B0310-0100

Everywhere you imagine.

16

H8S/2125 Group
Hardware Manual

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer
H8S Family / H8S/2100 Series

H8S/2125 R4F2125
R4P2125

Rev.1.00
Revision Date: Sep. 21, 2006

Renesas Technology
www.renesas.com

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

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

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)

    Product code, Package dimensions, etc.

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

The H8S/2125 Group is a series of microcomputers (MCUs) made up of the H8S/2000 CPU with Renesas Technology's original architecture as its core, and the peripheral functions required to configure a system.

The H8S/2000 CPU has an internal 32-bit configuration, sixteen 16-bit general registers, and a simple and optimized instruction set for high-speed operation. The H8S/2000 CPU can handle a 16-Mbyte linear address space. The instruction set of the H8S/2000 CPU maintains upward compatibility at the object level with the H8/300 and H8/300H CPUs. This allows the transition from the H8/300, H8/300L, or H8/300H to the H8S/2000 CPU.

This LSI is equipped with ROM, RAM, two kinds of PWM timers (PWM and PWMX), a 16-bit free running timer (FRT), a 16-bit cycle measurement timer (TCM), a 8-bit timer (TMR), watchdog timer (WDT), serial communication interface (SCI), I^2C bus interface (IIC), an A/D converter, and I/O ports as on-chip peripheral modules required for system configuration. In addition, a data transfer controller (DTC) is included as bus master.

A flash memory (F-ZTAT™*) or PROM (OTP Version) is available for this LSI's 512-Kbyte ROM. The CPU and ROM are connected to a 16-bit bus, enabling byte data and word data to be accessed in a single state. This improves the instruction fetch and process speeds.

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

Target Users: This manual was written for users who use the H8S/2125 in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logic circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8S/2125 Group to the target users.
Refer to the H8S/2600 Series, H8S/2000 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 this manual in the order of the table of contents. This manual can be roughly categorized into the descriptions on the CPU, system control functions, peripheral functions and electrical characteristics.

- In order to understand the details of the CPU's functions

  Read the H8S/2600 Series, H8S/2000 Series Software Manual.
- In order to understand the detailed function of a register whose 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 23, List of Registers.

Rules:	Register name:	The following notation is used for cases when the same or a similar function, e.g., serial communication 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.
	Number notation:	Binary is B'xxxx, hexadecimal is H'xxxx, decimal is xxxx.
	Signal notation:	An overbar is added to a low-active signal: $\overline{\text{xxxx}}$

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/

H8S/2125 Group manuals:

Document Title	Document No.
H8S/2125 Group Hardware Manual	This manual
H8S/2600 Series, H8S/2000 Series Software Manual	REJ09B0139

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
Microcomputer Development Environment System H8S, H8/300 Series Simulator/Debugger User's Manual	ADE-702-282
H8S, H8/300 Series High-performance Embedded Workshop 3 Tutorial	REJ10B0024
H8S, H8/300 Series High-performance Embedded Workshop 3 User's Manual	REJ10B0026

Contents

Figures

Section 6 Bus Controller (BSC)

Section 7 Data Transfer Controller (DTC)

Section 8 I/O Ports

Section 9 8-Bit PWM Timer (PWM)

Section 10 14-Bit PWM Timer (PWMX)

Section 11 16-Bit Free-Running Timer (FRT)

Section 12 16-Bit Cycle Measurement Timer (TCM)

Section 13 8-Bit Timer (TMR)

Section 14 Watchdog Timer (WDT)

Section 15 Serial Communication Interface (SCI)

Section 16 I^2C Bus Interface (IIC)

Section 17 A/D Converter

Section 19 Flash Memory (0.18-µm F-ZTAT Version)

Section 20 PROM (OTP Version)

Section 21 Clock Pulse Generator

Section 22 Power-Down Modes

Section 24 Electrical Characteristics

Appendix

Tables

Appendix

Section 1 Overview

1.1 Overview

- 16-bit high-speed H8S/2000 CPU
  Upward-compatible with the H8/300 and H8/300H CPUs on an object level
  Sixteen 16-bit general registers
  65 basic instructions
- Various peripheral functions
  Data transfer controller (DTC)
  8-bit PWM timer (PWM)
  14-bit PWM timer (PWMX)
  16-bit cycle measurement timer (TCM)
  16-bit free-running timer (FRT)
  8-bit timer (TMR)
  Watchdog timer (WDT)
  Asynchronous or clocked synchronous serial communication interface (SCI)
  I^2C bus interface (IIC)
  10-bit A/D converter
  H-UDI interface (H-UDI)
  Clock pulse generator

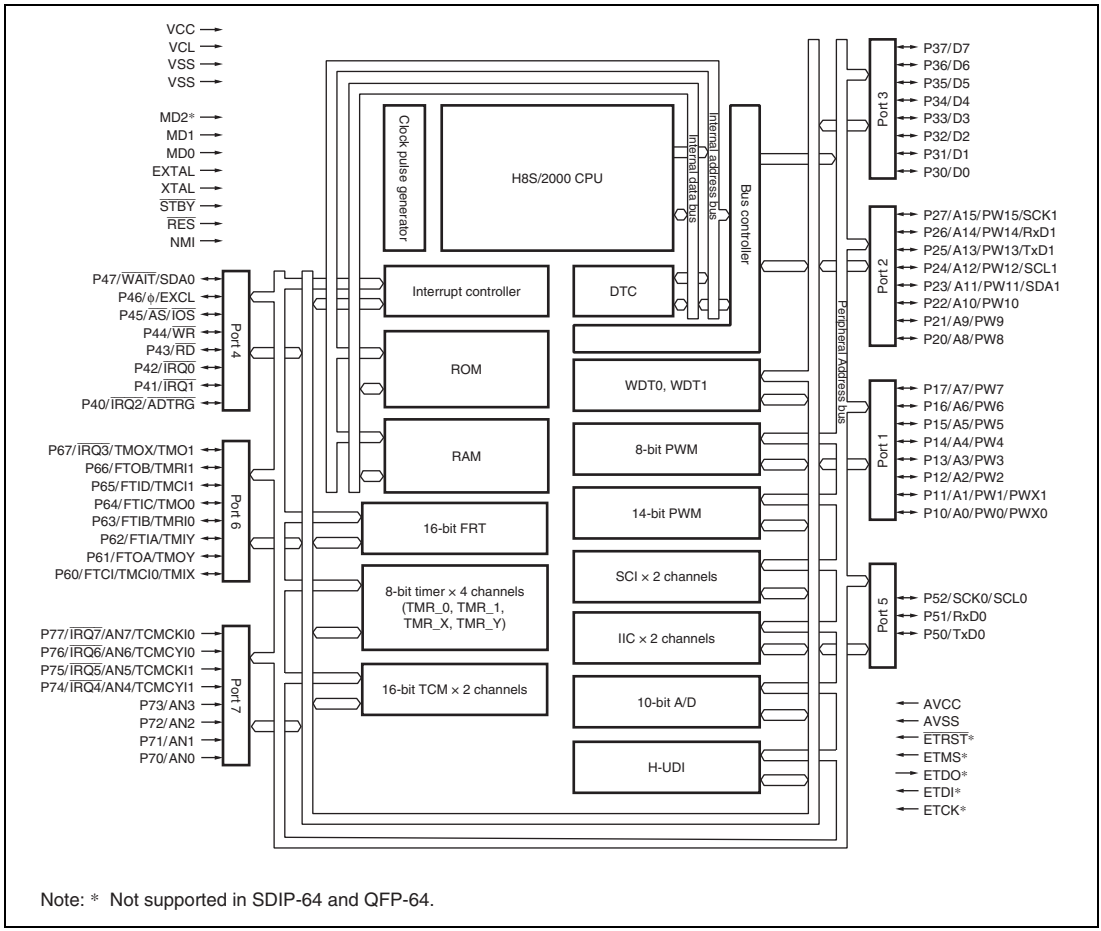
- On-chip memory

ROM Type	Model	ROM	RAM	Remarks
Flash memory version	R4F2125	512 Kbytes	8 Kbytes	Under development
PROM (OTP version)	R4P2125	512 Kbytes	8 Kbytes	Under development

- General I/O ports
  I/O pins: 43
  Input pins: 8
- Supports various power-down states
- Compact package

Package	Code	Body Size	Pin Pitch
SDIP-64	PRDP0064BB-A (DP-64S)	17.0 × 57.6 mm	1.78 mm
QFP-64	PRQP0064GB-A (FP-64A)	14.0 × 14.0 mm	0.8 mm
TQFP-80	PTQP0080KC-A (TFP-80C)	12.0 × 12.0 mm	0.5 mm

1.2 Internal Block Diagram



Note: * Not supported in SDIP-64 and QFP-64.

Figure 1.1 Block Diagram

1.3 Pin Description

1.3.1 Pin Assignments

Figure 1.2 to 1.4 show the pin assignments of the H8S/2125 Group.

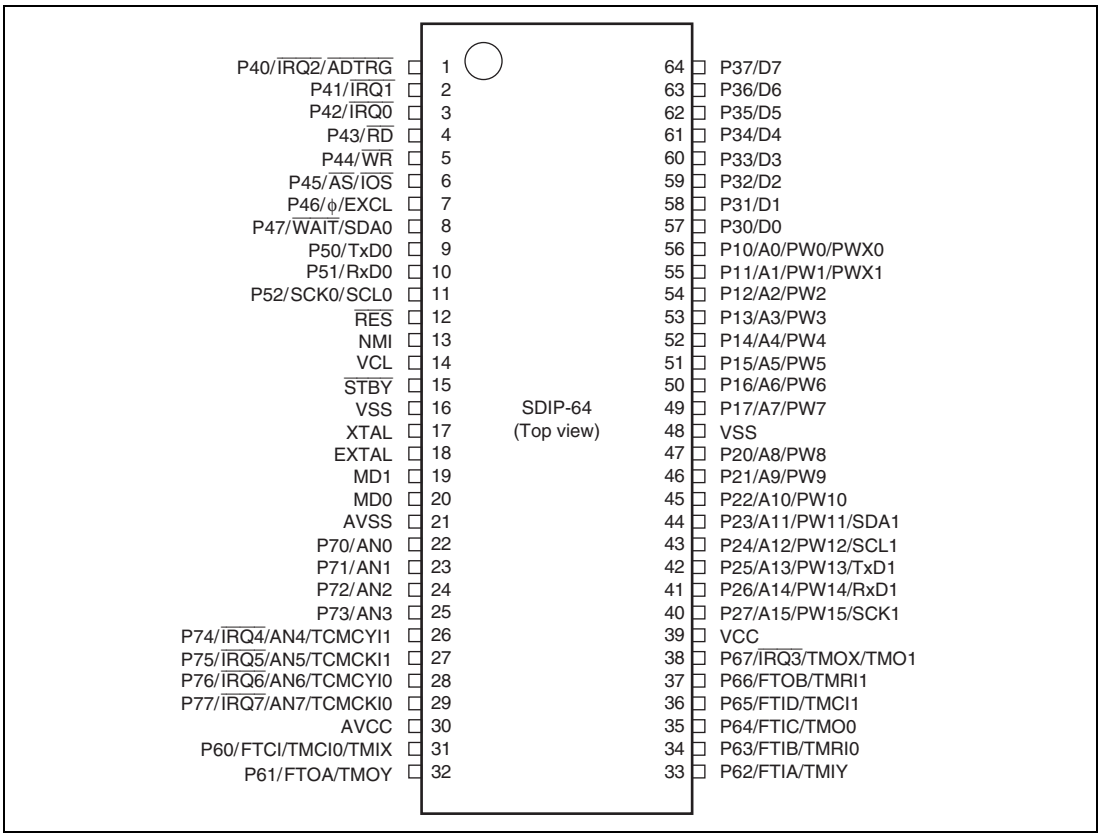


Figure 1.2 Pin Assignments (SDIP-64)

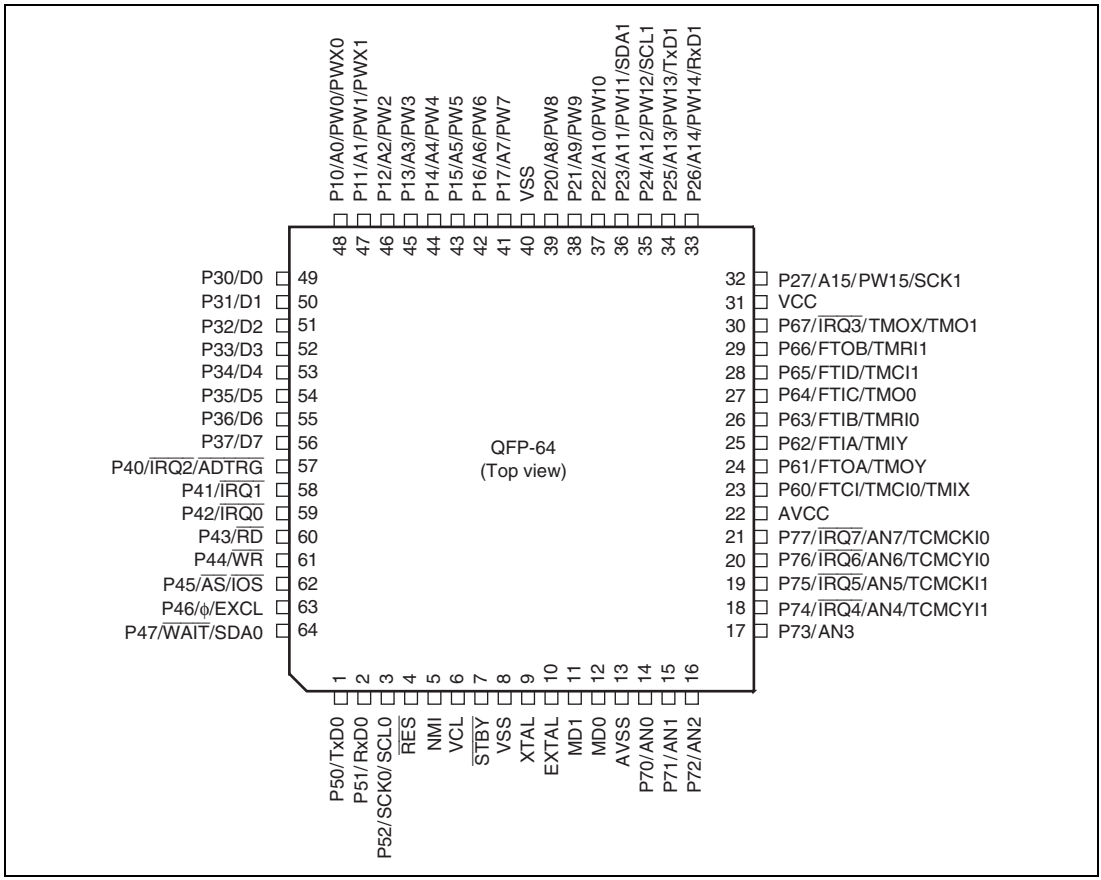


Figure 1.3 Pin Assignments (QFP-64)

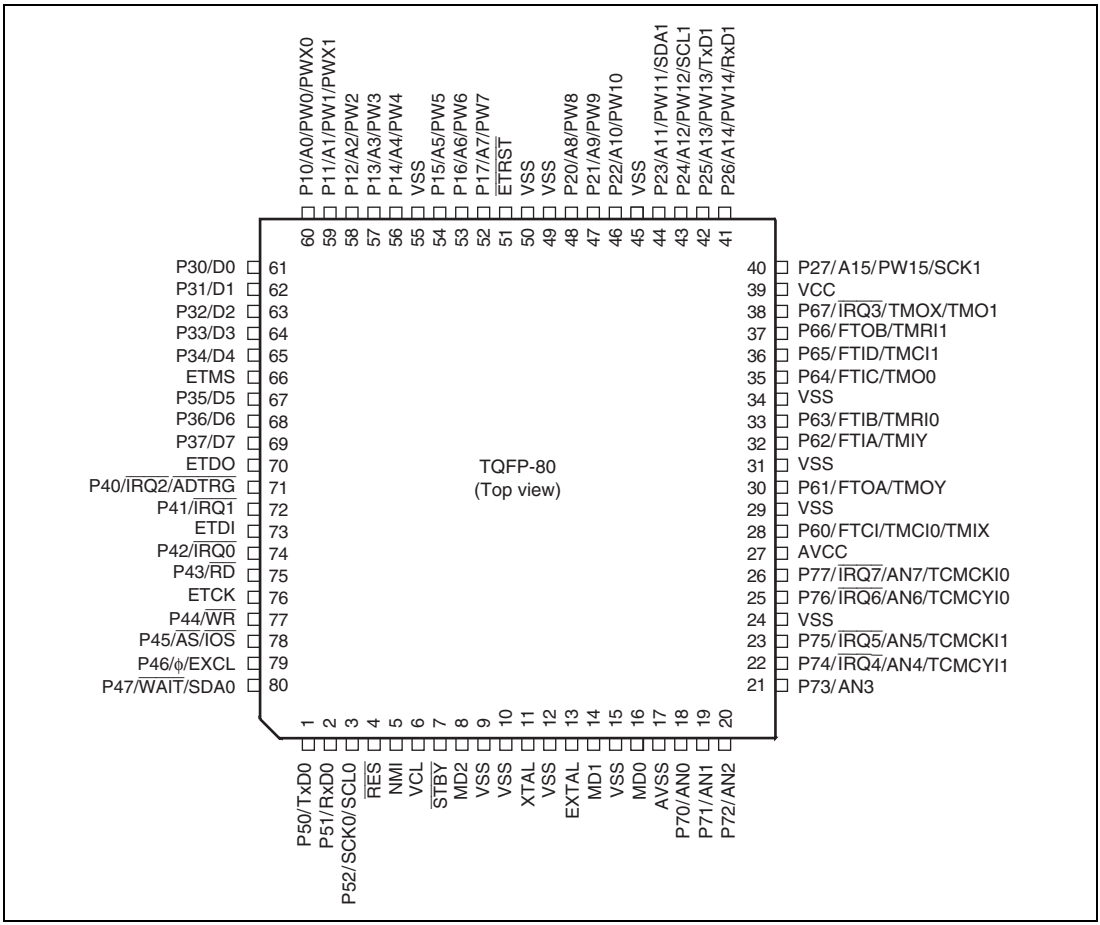


Figure 1.4 Pin Assignments (TQFP-80)

1.3.2 Pin Functions in Each Operating Mode

Table 1.1 Pin Functions in Each Operating Mode

Pin No.			Pin Name			
			Extended Mode		Single-Chip Mode	Flash Memory Programmer Mode
SDIP-64	QFP-64	TQFP-80	Mode 1	Mode 2 (EXPE = 1), Mode 3 (EXPE = 1)	Mode 2 (EXPE = 0), Mode 3 (EXPE = 0)	
1	57	71	P40/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$	P40/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$	P40/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$	VCC
2	58	72	P41/$\overline{\text{IRQ1}}$	P41/$\overline{\text{IRQ1}}$	P41/$\overline{\text{IRQ1}}$	VCC
—	—	73	ETDI	ETDI	ETDI	VCC
3	59	74	P42/$\overline{\text{IRQ0}}$	P42/$\overline{\text{IRQ0}}$	P42/$\overline{\text{IRQ0}}$	VSS
4	60	75	$\overline{\text{RD}}$	$\overline{\text{RD}}$	P43	$\overline{\text{WE}}$
—	—	76	ETCK	ETCK	ETCK	VCC
5	61	77	$\overline{\text{WR}}$	$\overline{\text{WR}}$	P44	FA15
6	62	78	$\overline{\text{AS}}$/$\overline{\text{IOS}}$	$\overline{\text{AS}}$/$\overline{\text{IOS}}$	P45	FA16
7	63	79	P46/ϕ/EXCL	P46/ϕ/EXCL	P46/ϕ/EXCL	VSS
8*	64*	80*	P47/$\overline{\text{WAIT}}$/SDA0	P47/$\overline{\text{WAIT}}$/SDA0	P47/SDA0	VCC
9	1	1	P50/TxD0	P50/TxD0	P50/TxD0	FA19
10	2	2	P51/RxD0	P51/RxD0	P51/RxD0	FA17
11*	3*	3*	P52/SCK0/SCL0	P52/SCK0/SCL0	P52/SCK0/SCL0	FA18
12	4	4	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
13	5	5	NMI	NMI	NMI	FA9
14	6	6	VCL	VCL	VCL	VCL
15	7	7	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	VCC
—	—	8	MD2	MD2	MD2	VSS
16	8	9	VSS	VSS	VSS	VSS
—	—	10	VSS	VSS	VSS	VSS
17	9	11	XTAL	XTAL	XTAL	XTAL
—	—	12	VSS	VSS	VSS	VSS
18	10	13	EXTAL	EXTAL	EXTAL	EXTAL
19	11	14	MD1	MD1	MD1	VSS
—	—	15	VSS	VSS	VSS	VSS
20	12	16	MD0	MD0	MD0	VSS

Pin No.			Pin Name			
			Extended Mode		Single-Chip Mode	Flash Memory Programmer Mode
SDIP-64	QFP-64	TQFP-80	Mode 1	Mode 2 (EXPE = 1), Mode 3 (EXPE = 1)	Mode 2 (EXPE = 0), Mode 3 (EXPE = 0)	
21	13	17	AVSS	AVSS	AVSS	VSS
22	14	18	P70/AN0	P70/AN0	P70/AN0	VSS
23	15	19	P71/AN1	P71/AN1	P71/AN1	VSS
24	16	20	P72/AN2	P72/AN2	P72/AN2	VSS
25	17	21	P73/AN3	P73/AN3	P73/AN3	VSS
26	18	22	P74/$\overline{\text{IRQ4}}$/AN4/ TCMCYI1	P74/$\overline{\text{IRQ4}}$/AN4/ TCMCYI1	P74/$\overline{\text{IRQ4}}$/AN4/ TCMCYI1	VSS
27	19	23	P75/$\overline{\text{IRQ5}}$/AN5/ TCMCKI1	P75/$\overline{\text{IRQ5}}$/AN5/ TCMCKI1	P75/$\overline{\text{IRQ5}}$/AN5/ TCMCKI1	VSS
—	—	24	VSS	VSS	VSS	VSS
28	20	25	P76/$\overline{\text{IRQ6}}$/AN6/ TCMCYI0	P76/$\overline{\text{IRQ6}}$/AN6/ TCMCYI0	P76/$\overline{\text{IRQ6}}$/AN6/ TCMCYI0	VSS
29	21	26	P77/$\overline{\text{IRQ7}}$/AN7/ TCMCKI0	P77/$\overline{\text{IRQ7}}$/AN7/ TCMCKI0	P77/$\overline{\text{IRQ7}}$/AN7/ TCMCKI0	VSS
30	22	27	AVCC	AVCC	AVCC	VCC
31	23	28	P60/FTCI/TMCI0/ TMIX	P60/FTCI/TMCI0/ TMIX	P60/FTCI/TMCI0/ TMIX	VSS
—	—	29	VSS	VSS	VSS	VSS
32	24	30	P61/FTOA/TMOY	P61/FTOA/TMOY	P61/FTOA/TMOY	VSS
—	—	31	VSS	VSS	VSS	VSS
33	25	32	P62/FTIA/TMIY	P62/FTIA/TMIY	P62/FTIA/TMIY	VSS
34	26	33	P63/FTIB/TMRI0	P63/FTIB/TMRI0	P63/FTIB/TMRI0	VSS
—	—	34	VSS	VSS	VSS	VSS
35	27	35	P64/FTIC/TMO0	P64/FTIC/TMO0	P64/FTIC/TMO0	VSS
36	28	36	P65/FTID/TMCI1	P65/FTID/TMCI1	P65/FTID/TMCI1	VSS
37	29	37	P66/FTOB/TMRI1	P66/FTOB/TMRI1	P66/FTOB/TMRI1	VSS
38	30	38	P67/$\overline{\text{IRQ3}}$/TMO1/ TMOX	P67/$\overline{\text{IRQ3}}$/TMO1/ TMOX	P67/$\overline{\text{IRQ3}}$/TMO1/ TMOX	VSS
39	31	39	VCC	VCC	VCC	VCC
40	32	40	A15	P27/A15/PW15/ SCK1	P27/PW15/SCK1	$\overline{\text{CE}}$

Pin No.			Pin Name			
			Extended Mode		Single-Chip Mode	Flash Memory Programmer Mode
SDIP-64	QFP-64	TQFP-80	Mode 1	Mode 2 (EXPE = 1), Mode 3 (EXPE = 1)	Mode 2 (EXPE = 0), Mode 3 (EXPE = 0)	
41	33	41	A14	P26/A14/PW14/ RxD1	P26/PW14/RxD1	FA14
42	34	42	A13	P25/A13/PW13/ TxD1	P25/PW13/TxD1	FA13
43	35	43	A12	P24/A12/PW12/ SCL1	P24/PW12/SCL1	FA12
44	36	44	A11	P23/A11/PW11/ SDA1	P23/PW11/SDA1	FA11
—	—	45	VSS	VSS	VSS	VSS
45	37	46	A10	P22/A10/PW10	P22/PW10	FA10
46	38	47	A9	P21/A9/PW9	P21/PW9	$\overline{OE}$
47	39	48	A8	P20/A8/PW8	P20/PW8	FA8
—	—	49	VSS	VSS	VSS	VSS
48	40	50	VSS	VSS	VSS	VSS
—	—	51	$\overline{ETRST}$	$\overline{ETRST}$	$\overline{ETRST}$	VSS
49	41	52	A7	P17/A7/PW7	P17/PW7	FA7
50	42	53	A6	P16/A6/PW6	P16/PW6	FA6
51	43	54	A5	P15/A5/PW5	P15/PW5	FA5
—	—	55	VSS	VSS	VSS	VSS
52	44	56	A4	P14/A4/PW4	P14/PW4	FA4
53	45	57	A3	P13/A3/PW3	P13/PW3	FA3
54	46	58	A2	P12/A2/PW2	P12/PW2	FA2
55	47	59	A1	P11/A1/PW1/PWX1	P11/PW1/PWX1	FA1
56	48	60	A0	P10/A0/PW0/PWX0	P10/PW0/PWX0	FA0
57	49	61	D0	D0	P30	FO0
58	50	62	D1	D1	P31	FO1
59	51	63	D2	D2	P32	FO2
60	52	64	D3	D3	P33	FO3
61	53	65	D4	D4	P34	FO4
—	—	66	ETMS	ETMS	ETMS	VCC

Pin No.			Pin Name			
			Extended Mode		Single-Chip Mode	Flash Memory Programmer Mode
SDIP-64	QFP-64	TQFP-80	Mode 1	Mode 2 (EXPE = 1), Mode 3 (EXPE = 1)	Mode 2 (EXPE = 0), Mode 3 (EXPE = 0)	
62	54	67	D5	D5	P35	FO5
63	55	68	D6	D6	P36	FO6
64	56	69	D7	D7	P37	FO7
—	—	70	ETDO	ETDO	ETDO	NC

Note: * NMOS push-pull/open-drain drive capability or 5-V tolerant input pin.

1.3.3 Pin Functions

Table 1.2 Pin Functions

Type	Symbol	Pin No. SDIP-64	Pin No. QFP-64	Pin No. TQFP-80	I/O	Name and Function
Power supply	VCC	39	31	39	Input	Power supply pin. Connect the pin to the system power supply. Connect the bypass capacitor between VCC and VSS (near VCC).
	VCL	14	6	6	Input	External capacitance pin for internal step-down power. Connect this pin to VSS through an external capacitor (that is located near this pin) to stabilize internal step-down power.
	VSS	16, 48	8, 40	9, 10, 12, 15, 24, 29, 31, 34, 45, 49, 50, 55	Input	Ground pins. Connect all these pins to the system power supply (0 V).
Clock	XTAL	17	9	11	Input	For connection to a crystal resonator. An external clock can be supplied from the EXTAL pin. For an example of crystal resonator connection, see section 21, Clock Pulse Generator.
	EXTAL	18	10	13	Input	
	ϕ	7	63	79	Output	Supplies the system clock to external devices.
	EXCL	7	63	79	Input	32.768-kHz external sub-clock should be supplied.
Operating mode control	MD2*[1] MD1 MD0	— 19 20	— 11 12	8 14 16	Input	These pins set the operating mode. Inputs at these pins should not be changed during operation.

Type	Symbol	Pin No. SDIP-64	Pin No. QFP-64	Pin No. TQFP-80	I/O	Name and Function
System control	$\overline{\text{RES}}$	12	4	4	Input	Reset pin. When this pin is low, the chip is reset.
	$\overline{\text{STBY}}$	15	7	7	Input	When this pin is low, a transition is made to hardware standby mode.
Address bus	A15 to A0	40 to 47, 49 to 56	32 to 39, 41 to 48	40 to 44, 46 to 48, 52 to 54, 56 to 60	Output	These pins output addresses.
Data bus	D7 to D0	64 to 57	56 to 49	69 to 67, 65 to 61	Input/ Output	These pins are bidirectional data bus.
Bus control	$\overline{\text{WAIT}}$	8	64	80	Input	This pin requests insertion of a wait state in the bus cycle when accessing external 3-state address space.
	$\overline{\text{RD}}$	4	60	75	Output	When this pin is low, it indicates that the external address space is being read.
	$\overline{\text{WR}}$	5	61	77	Output	When this pin is low, it indicates that the external address space is being written to.
	$\overline{\text{AS}}$/$\overline{\text{IOS}}$	6	62	78	Output	When this pin is low, it indicates that address output on the address bus is valid.
Interrupts	NMI	13	5	5	Input	Input pin for a nonmaskable interrupt request.
	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ2}}$, $\overline{\text{IRQ3}}$, $\overline{\text{IRQ4}}$ to $\overline{\text{IRQ7}}$	1 to 3 38 26 to 29	57 to 59 30 18 to 21	71, 72, 74 38 22 to 26	Input	These pins request a maskable interrupt.
PWM timer (PWM)	PW15 to PW0	40 to 47, 49 to 56	32 to 39, 41 to 48	40 to 44, 46 to 48, 52 to 54, 56 to 60	Output	PWM timer pulse output pins.
14-bit PWM timer (PWMX)	PWX0 PWX1	56 55	48 47	60 59	Output	PWMX pulse output pins

Type	Symbol	Pin No. SDIP-64	QFP-64	TQFP-80	I/O	Name and Function
16-bit free running timer (FRT)	FTCI	31	23	28	Input	External event input pin.
	FTOA	32	24	30	Output	Output compare output pins.
	FTOB	37	29	37		
	FTIA	33	25	32	Input	Input capture input pins.
	FTIB	34	26	33		
	FTIC	25	27	35		
	FTID	36	28	36		
16-bit cycle measure-ment timer (TCM_0, TCM_1)	TCMCKI0	29	21	26	Input	Input pins for the external clock input to the counter.
	TCMCKI1	27	19	23		
	TCMCYI0	28	20	25	Input	External event input pins.
	TCMCYI1	26	18	22		
8-bit timer (TMR_0, TMR_1, TMR_X, TMR_Y)	TMO0	35	27	35	Output	Waveform output pins with output compare function.
	TMO1	38	30	38		
	TMOX	38	30	38		
	TMOY	32	24	30		
	TMCI0	31	23	28	Input	Input pins for the external clock input to the counter.
	TMCI1	36	28	36		
	TMRI0	34	26	33	Input	Counter reset input pins.
	TMRI1	37	29	37		
	TMIX	31	23	28	Input	External event input pins and counter reset input pins.
	TMIY	33	25	32		
Serial communi-cation interface (SCI_0, SCI_1)	TxD0	9	1	1	Output	Transmit data output pins.
	TxD1	42	34	42		
	RxD0	10	2	2	Input	Receive data input pins.
	RxD1	41	33	41		
	SCK0	11	3	3	Input/ Output	Clock input/output pins. Output type is NMOS push-pull output.
	SCK1	40	32	40	Input/ Output	Clock input pin.

Type	Symbol	Pin No. SDIP-64	QFP-64	TQFP-80	I/O	Name and Function
I²C bus interface (IIC)	SCL0 SCL1	11 43	3 35	3 43	Input/ Output	I²C clock input/output pins. SCL0 output type is NMOS open drain and can drive a bus directly.
	SDA0 SDA1	8 44	64 36	80 44	Input/ Output	I²C data input/output pins. SDA0 output type is NMOS open drain and can drive a bus directly.
A/D converter	AN7 to AN0	29 to 22	21 to 14	26, 25, 23 to 18	Input	Analog input pins
	$\overline{\text{ADTRG}}$	1	57	71	Input	External trigger input pin to start A/D conversion.
	AVCC	30	22	27	Input	Reference power supply for the A/D converter. When the A/D converter is not used, this pin should be connected to the system power supply (+3 V).
	AVSS	21	13	17	Input	Ground pin for the A/D converter. Connect this pin to the system power supply (0 V).
H-UDI interface (H-UDI)	$\overline{\text{ETRST}}$*1*2	—	—	51	Input	Interface pins for emulator. Reset by holding the $\overline{\text{ETRST}}$ pin to low regardless of the H-UDI activation. At this time, the $\overline{\text{ETRST}}$ pin should be held low for 20 clocks of ETCK. Then, to activate the H-UDI, the $\overline{\text{ETRST}}$ pin should be set to high and the pins ETCK, ETMS, and ETDI should be set appropriately. When in the normal operation without activating the H-UDI, pins $\overline{\text{ETRST}}$, ETCK, ETMS, and ETDI are set to high or high-impedance. As these pins are pulled up inside the chip, care is required during standby state.
	ETMS*1	—	—	66	Input	
	ETDO*1	—	—	70	Output	
	ETDI*1	—	—	73	Input	
	ETCK*1	—	—	76	Input	

Type	Symbol	Pin No. SDIP-64	QFP-64	TQFP-80	I/O	Name and Function
I/O ports	P17 to P10	49 to 56	41 to 48	52 to 54, 56 to 60	Input/ Output	8-bit input/output pins.
	P27 to P20	40 to 47	32 to 39	40 to 44, 46 to 48	Input/ Output	8-bit input/output pins.
	P37 to P30	64 to 57	56 to 49	69 to 67, 65 to 61	Input/ Output	8-bit input/output pins.
	P47 to P40	8 to 1	64 to 57	80 to 77, 75 to 74, 72 to 71	Input/ Output	8-bit input/output pins.
	P52 to P50	11 to 9	3 to 1	3 to 1	Input/ Output	3-bit input/output pins.
	P67 to P60	38 to 31	30 to 23	38 to 35, 33 to 32, 30, 28	Input/ Output	8-bit input/output pins.
	P77 to P70	29 to 22	21 to 14	26 to 25, 23 to 18	Input	8-bit input pins.

Notes: 1. Not supported in SDIP-64 and QFP-64. The input value on the MD2 or $\overline{\text{ETRST}}$ pin is fixed to 0.

2. Following precautions are required on the power-on reset signal that is applied to the $\overline{\text{ETRST}}$ pin.
The reset signal should be applied on power supply.
Set apart the power on reset circuit from this LSI to prevent the $\overline{\text{ETRST}}$ pin of the board tester from affecting the operation of this LSI.
Set apart the power on reset circuit from this LSI to prevent the system reset of this LSI from affecting the $\overline{\text{ETRST}}$ pin of the board tester.

Section 2 CPU

The H8S/2000 CPU is a high-speed central processing unit with an internal 32-bit architecture that is upward-compatible with the H8/300 and H8/300H CPUs. The H8S/2000 CPU has sixteen 16-bit general registers, can address a 16 Mbytes linear address space, and is ideal for realtime control.

This section describes the H8S/2000 CPU. The usable modes and address spaces differ depending on the product. For details on each product, see section 3, MCU Operating Modes.

2.1 Features

- Upward-compatibility with H8/300 and H8/300H CPUs
  - Can execute H8/300 CPU and H8/300H CPU object programs
- General-register architecture
  - Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-bit registers
- Sixty-five basic instructions
  - 8/16/32-bit 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:32,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 16 Mbytes address space
  - Program: 16 Mbytes
  - Data: 16 Mbytes

- High-speed operation
  - All frequently-used instructions are executed in one or two states
  - 8/16/32-bit register-register add/subtract: 1 state
  - 8 × 8-bit register-register multiply: 12 states (MULXU.B), 13 states (MULXS.B)
  - 16 ÷ 8-bit register-register divide: 12 states (DIVXU.B)
  - 16 × 16-bit register-register multiply: 20 states (MULXU.W), 21 states (MULXS.W)
  - 32 ÷ 16-bit register-register divide: 20 states (DIVXU.W)
- Two CPU operating modes
  - Normal mode
  - Advanced mode
- Power-down state
  - Transition to power-down state by SLEEP instruction
  - Selectable CPU clock speed

2.1.1 Differences between H8S/2600 CPU and H8S/2000 CPU

The differences between the H8S/2600 CPU and the H8S/2000 CPU are as shown below.

- Register configuration

  The MAC register is supported only by the H8S/2600 CPU.
- Basic instructions

  The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported only by the H8S/2600 CPU.
- The number of execution states of the MULXU and MULXS instructions

		Execution States	
Instruction	**Mnemonic**	**H8S/2600**	**H8S/2000**
MULXU	MULXU.B Rs, Rd	3	12
	MULXU.W Rs, ERd	4	20
MULXS	MULXS.B Rs, Rd	4	13
	MULXS.W Rs, ERd	5	21

In addition, there are differences in address space, CCR and EXR register functions, power-down modes, etc., depending on the model.

2.1.2 Differences from H8/300 CPU

In comparison to the H8/300 CPU, the H8S/2000 CPU has the following enhancements.

- More general registers and control registers
  - Eight 16-bit extended registers and one 8-bit control register have been added.
- Extended address space
  - Normal mode supports the same 64 Kbytes address space as the H8/300 CPU.
  - Advanced mode supports a maximum 16 Mbytes address space.
- Enhanced addressing
  - The addressing modes have been enhanced to make effective use of the 16 Mbytes address space.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Signed multiply and divide instructions have been added.
  - Two-bit shift and two-bit rotate instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.
- Higher speed
  - Basic instructions are executed twice as fast.

2.1.3 Differences from H8/300H CPU

In comparison to the H8/300H CPU, the H8S/2000 CPU has the following enhancements.

- Additional control register
  - One 8-bit control register has been added.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Two-bit shift and two-bit rotate instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.
- Higher speed
  - Basic instructions are executed twice as fast.

2.2 CPU Operating Modes

The H8S/2000 CPU has two operating modes: normal and advanced. Normal mode supports a maximum 64 Kbytes address space. Advanced mode supports a maximum 16 Mbytes address space. The mode is selected by the LSI's mode pins.

2.2.1 Normal Mode

The exception vector table and stack have the same structure as in the H8/300 CPU in normal mode.

- Address space

  Linear access to a maximum address space of 64 Kbytes is possible.
- Extended registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers.

  When extended register En is used as a 16-bit register it can contain any value, even when the corresponding general register (Rn) is used as an address register. (If general register Rn is referenced in the register indirect addressing mode with pre-decrement (@–Rn) or post-increment (@Rn+) and a carry or borrow occurs, the value in the corresponding extended register (En) will be affected.)
- Instruction set

  All instructions and addressing modes can be used. Only the lower 16 bits of effective addresses (EA) are valid.
- Exception vector table and memory indirect branch addresses

  In normal mode, the top area starting at H'0000 is allocated to the exception vector table. One branch address is stored per 16 bits. The exception vector table in normal mode is shown in figure 2.1. For details of the exception vector table, see section 4, Exception Handling.

  The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In normal mode, the operand is a 16-bit (word) operand, providing a 16-bit branch address. Branch addresses can be stored in the top area from H'0000 to H'00FF. Note that this area is also used for the exception vector table.
- Stack structure

  In normal mode, when the program counter (PC) is pushed onto the stack in a subroutine call in normal mode, and the PC and condition-code register (CCR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.2. The extended control register (EXR) is not pushed onto the stack. For details, see section 4, Exception Handling.

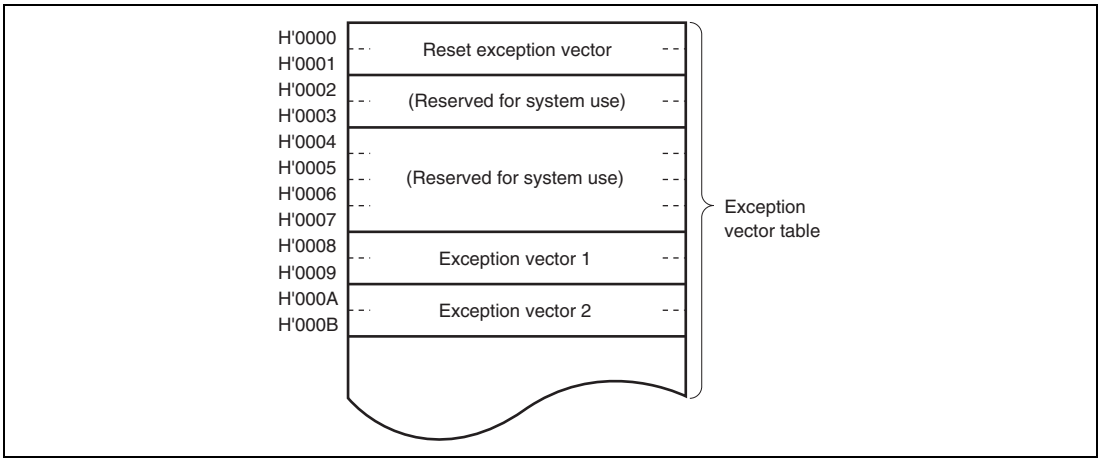


Figure 2.1 Exception Vector Table (Normal Mode)

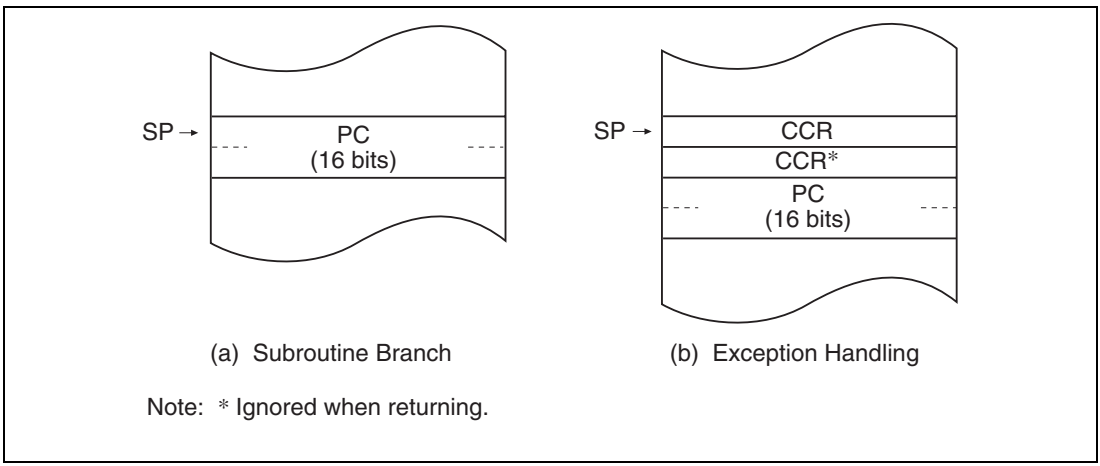


Figure 2.2 Stack Structure in Normal Mode

2.2.2 Advanced Mode

- Address space

  Linear access to a maximum address space of 16 Mbytes is possible.
- Extended registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers. They can also be used as the upper 16-bit segments of 32-bit registers or address registers.
- Instruction set

  All instructions and addressing modes can be used.
- Exception vector table and memory indirect branch addresses

  In advanced mode, the top area starting at H'00000000 is allocated to the exception vector table in 32-bit units. In each 32 bits, the upper eight bits are ignored and a branch address is stored in the lower 24 bits (see figure 2.3). For details of the exception vector table, see section 4, Exception Handling.

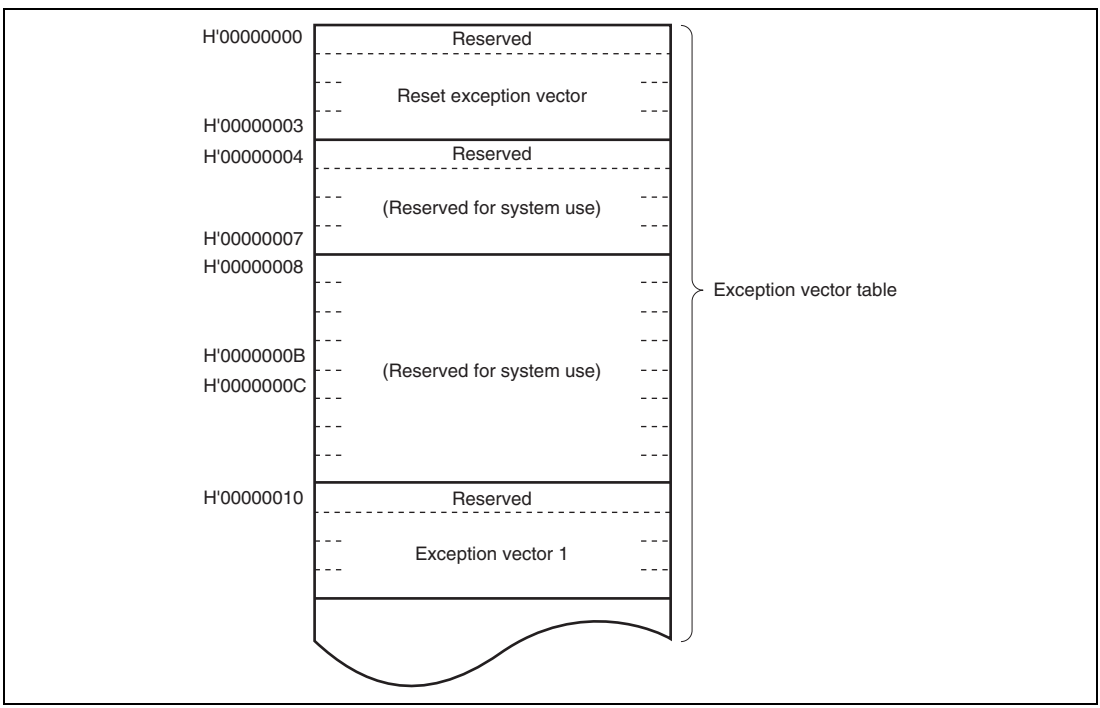


Figure 2.3 Exception Vector Table (Advanced Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In advanced mode, the operand is a 32-bit longword operand, providing a 32-bit branch address. The upper eight bits of these 32 bits are a reserved area that is regarded as H'00. Branch addresses can be stored in the area from H'00000000 to H'000000FF. Note that the top area of this range is also used for the exception vector table.

- Stack structure

  In advanced mode, when the program counter (PC) is pushed onto the stack in a subroutine call, and the PC and condition-code register (CCR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.4. The extended control register (EXR) is not pushed onto the stack. For details, see section 4, Exception Handling.

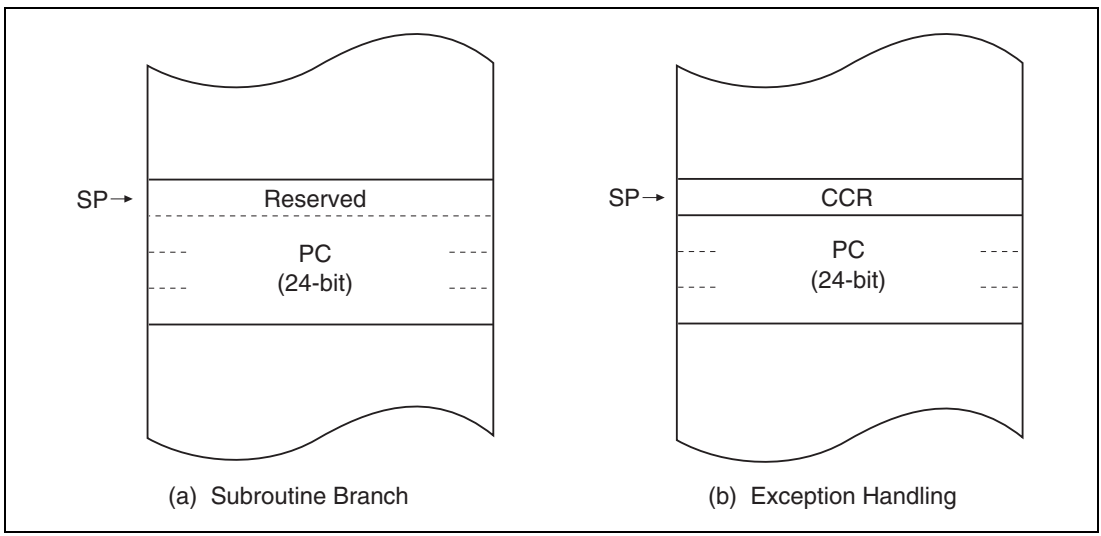


Figure 2.4 Stack Structure in Advanced Mode

2.3 Address Space

Figure 2.5 shows a memory map of the H8S/2000 CPU. The H8S/2000 CPU provides linear access to a maximum 64 Kbytes address space in normal mode, and a maximum 16 Mbytes (architecturally 4 Gbytes) address space in advanced mode. The usable modes and address spaces differ depending on the product. For details on each product, see section 3, MCU Operating Modes.

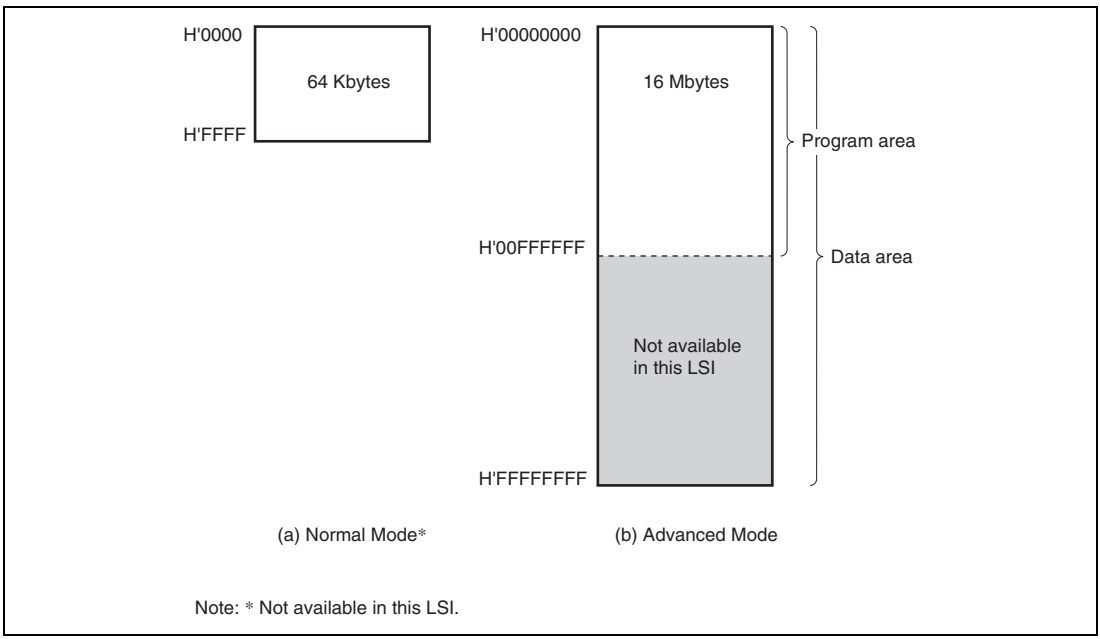


Figure 2.5 Memory Map

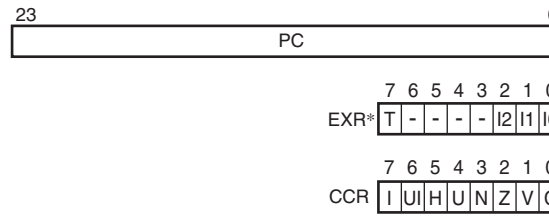
2.4 Register Configuration

The H8S/2000 CPU has the internal registers shown in figure 2.6. These are classified into two types of registers: general registers and control registers. Control registers refer to a 24-bit program counter (PC), an 8-bit extended control register (EXR), and an 8-bit condition code register (CCR).

General Registers (Rn) and Extended Registers (En)

	15 0	7 0	7 0
ER0	E0	R0H	R0L
ER1	E1	R1H	R1L
ER2	E2	R2H	R2L
ER3	E3	R3H	R3L
ER4	E4	R4H	R4L
ER5	E5	R5H	R5L
ER6	E6	R6H	R6L
ER7 (SP)	E7	R7H	R7L

Control Registers



[Legend]

SP:	Stack pointer	H:	Half-carry flag
PC:	Program counter	U:	User bit
EXR:	Extended control register	N:	Negative flag
T:	Trace bit	Z:	Zero flag
I2 to I0:	Interrupt mask bits	V:	Overflow flag
CCR:	Condition-code register	C:	Carry flag
I:	Interrupt mask bit		
UI:	User bit or interrupt mask bit		

Note: * Does not affect operation in this LSI.

Figure 2.6 CPU Internal Registers

2.4.1 General Registers

The H8S/2000 CPU has eight 32-bit general registers. These general registers are all functionally alike 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 2.7 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).

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

When the general registers are used as 8-bit registers, the R registers are divided into 8-bit general registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing sixteen 8-bit registers at the maximum.

The usage of each register can be selected independently.

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 2.8 shows the stack.

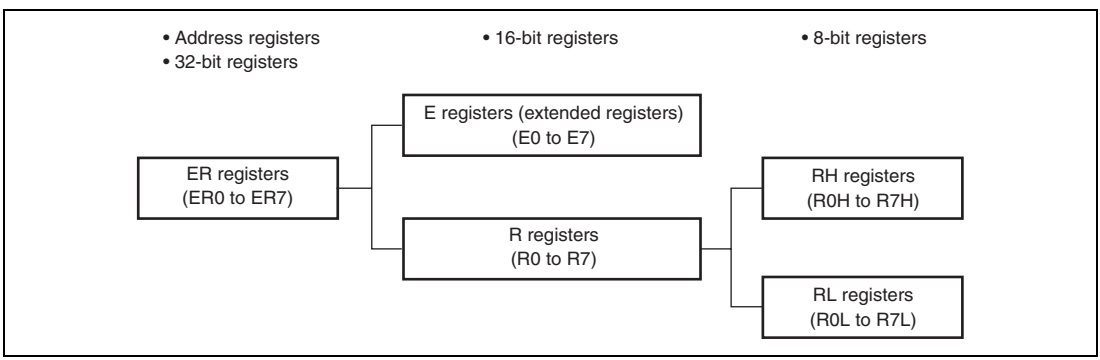


Figure 2.7 Usage of General Registers

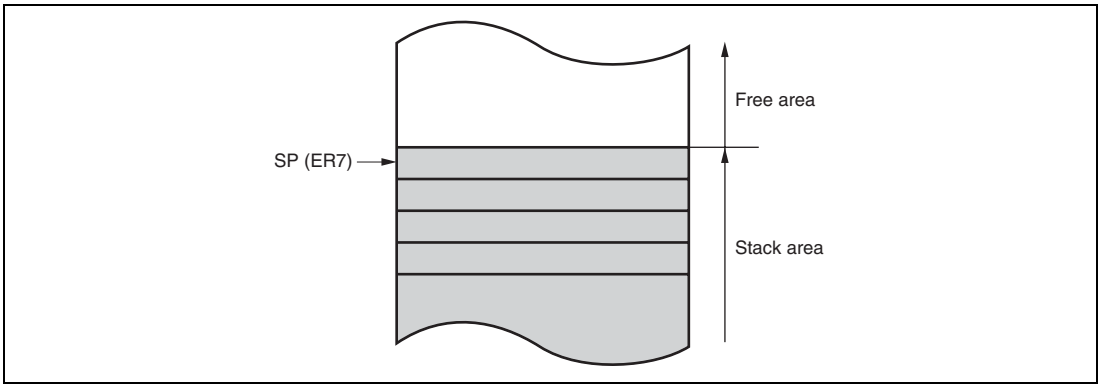


Figure 2.8 Stack

2.4.2 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 for read, the least significant PC bit is regarded as 0.)

2.4.3 Extended Control Register (EXR)

EXR does not affect operation in this LSI.

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit
				Does not affect operation in this LSI.
6 to 3	–	All 1	R	Reserved
				These bits are always read as 1.
2 to 0	I2	1	R/W	Interrupt Mask Bits 2 to 0
	I1	1	R/W	Do not affect operation in this LSI.
	I0	1	R/W	

2.4.4 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.

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.

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. For details, see section 5, Interrupt Controller.
6	UI	Undefined	R/W	User Bit or Interrupt Mask Bit Can be written to and read from 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 to and read from 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 when data is zero, and cleared to 0 when data is not zero.

Bit	Bit Name	Initial Value	R/W	Description
1	V	Undefined	R/W	Overflow Flag Set to 1 when an arithmetic overflow occurs, and cleared to 0 otherwise.
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.

2.4.5 Initial Register Values

Reset exception handling loads the CPU's program counter (PC) from the vector table, clears the trace (T) bit in EXR to 0, and sets the interrupt mask (I) bits in CCR and EXR to 1. The other CCR bits and the general registers are not initialized. Note that the stack pointer (ER7) is undefined. The stack pointer should therefore be initialized by an MOV.L instruction executed immediately after a reset.

2.5 Data Formats

The H8S/2000 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.

2.5.1 General Register Data Formats

Figure 2.9 shows the data formats of general registers.

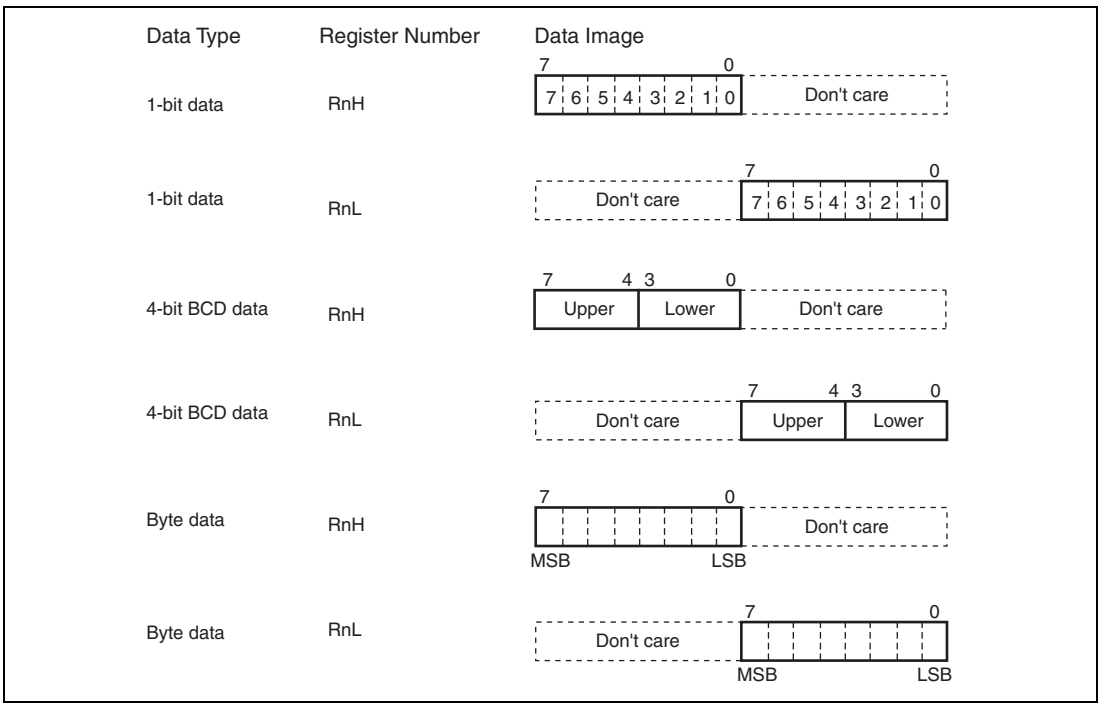


Figure 2.9 General Register Data Formats (1)

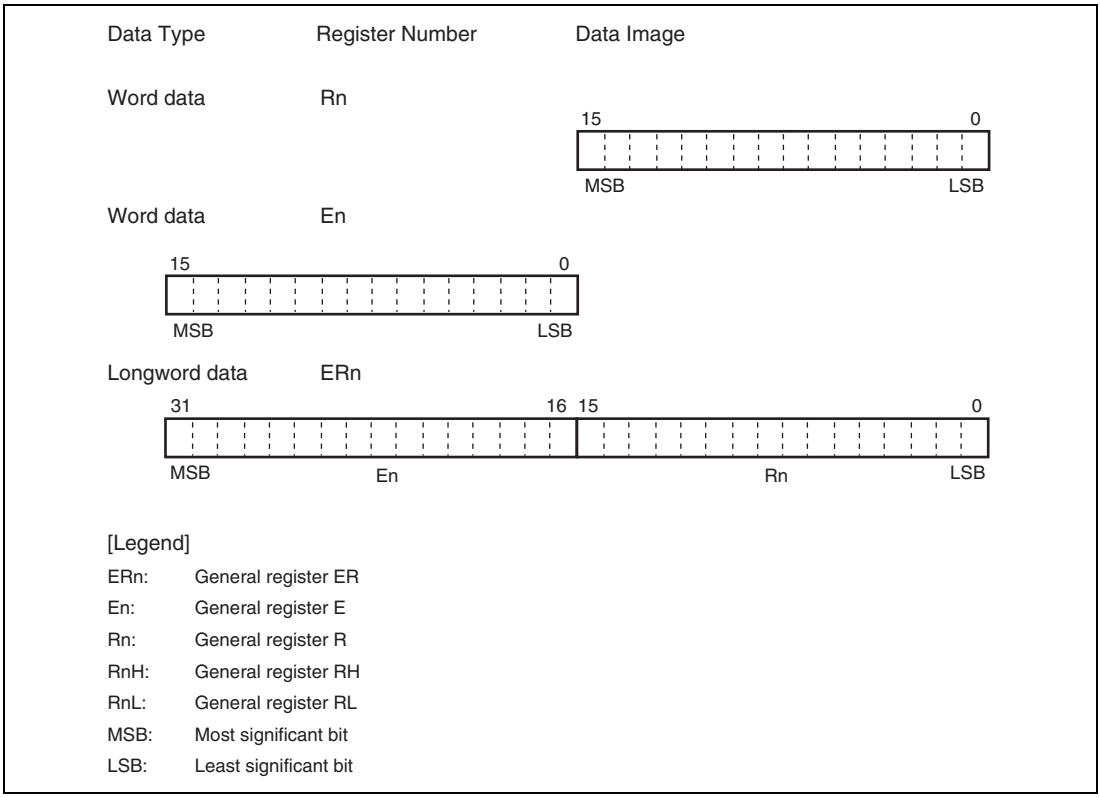


Figure 2.9 General Register Data Formats (2)

2.5.2 Memory Data Formats

Figure 2.10 shows the data formats in memory. The H8S/2000 CPU can access word data and longword data in memory, but 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, no address error occurs but the least significant bit of the address is regarded as 0, so the access starts at the preceding address. This also applies to instruction fetches.

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

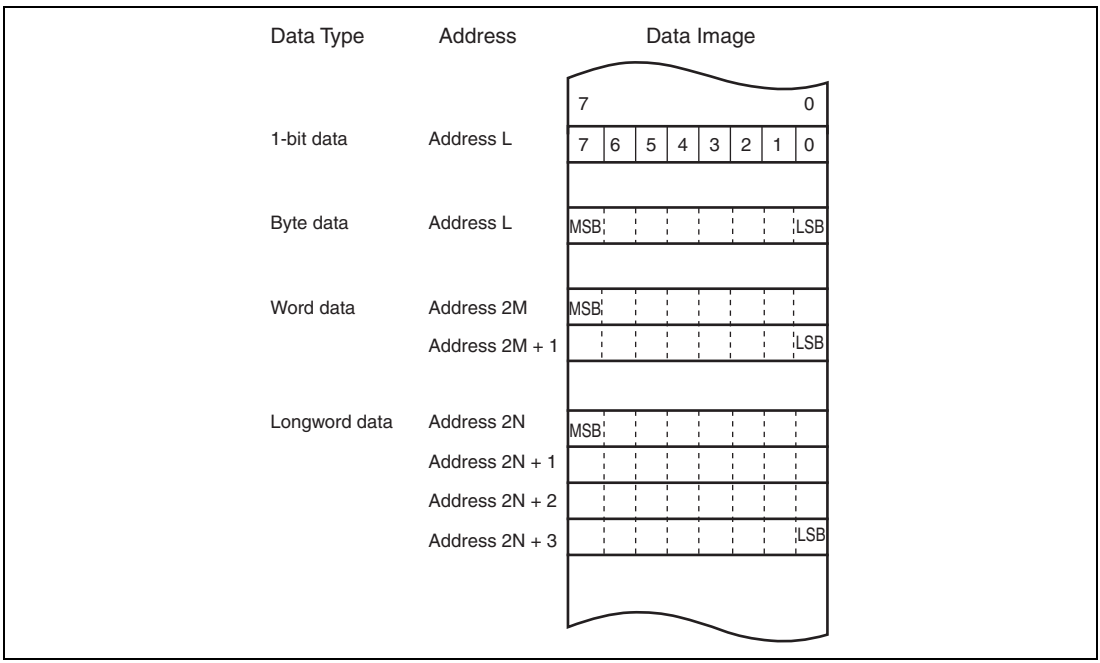


Figure 2.10 Memory Data Formats

2.6 Instruction Set

The H8S/2000 CPU has 65 types of instructions. The instructions are classified by function as shown in table 2.1.

Table 2.1 Instruction Classification

Function	Instructions	Size	Types
Data transfer	MOV	B/W/L	5
	POP*[1], PUSH*[1]	W/L	
	LDM*[5], STM*[5]	L	
	MOVFPE*[3], MOVTPE*[3]	B	
Arithmetic operations	ADD, SUB, CMP, NEG	B/W/L	19
	ADDX, SUBX, DAA, DAS	B	
	INC, DEC	B/W/L	
	ADDS, SUBS	L	
	MULXU, DIVXU, MULXS, DIVXS	B/W	
	EXTU, EXTS	W/L	
	TAS*[4]	B	
Logic operations	AND, OR, XOR, NOT	B/W/L	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	B/W/L	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B	14
Branch	B_{CC}*[2], JMP, BSR, JSR, RTS	–	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	–	9
Block data transfer	EEPMOV	–	1
			Total: 65

Notes: B: Byte size; W: Word size; L: Longword size.

1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W Rn, @-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn, @-SP.
2. B_{CC} is the generic name for conditional branch instructions.
3. Cannot be used in this LSI.
4. To use the TAS instruction, use registers ER0, ER1, ER4, and ER5.
5. Since register ER7 functions as the stack pointer in an STM/LDM instruction, it cannot be used as an STM/LDM register.

2.6.1 Table of Instructions Classified by Function

Tables 2.3 to 2.10 summarize the instructions in each functional category. The notation used in tables 2.3 to 2.10 is defined below.

Table 2.2 Operation Notation

Symbol	Description
Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
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 exclusive OR
→	Move
~	NOT (logical complement)
:8/:16/:24/:32	8-, 16-, 24-, or 32-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 (ER0 to ER7).

Table 2.3 Data Transfer Instructions

Instruction	Size*[1]	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	Cannot be used in this LSI.
MOVTPE	B	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.
LDM*[2]	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
STM*[2]	L	Rn (register list) → @-SP Pushes two or more general registers onto the stack.

Notes: 1. Size refers to the operand size.
B: Byte
W: Word
L: Longword
2. Since register ER7 functions as the stack pointer in an STM/LDM instruction, it cannot be used as an STM/LDM register.

Table 2.4 Arithmetic Operations Instructions (1)

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

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

Table 2.4 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 the 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.
TAS*[2]	B	@ERd – 0, 1 → (<bit 7> of @ERd) Tests memory contents, and sets the most significant bit (bit 7) to 1.

Notes: 1. Size refers to the operand size.
B: Byte
W: Word
L: Longword
2. To use the TAS instruction, use registers ER0, ER1, ER4, and ER5.

Table 2.5 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 data in a general register.

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

Table 2.6 Shift Instructions

Instruction	Size*	Function
SHAL	B/W/L	Rd (shift) → Rd
SHAR		Performs an arithmetic shift on data in a general register. 1-bit or 2 bit shift is possible.
SHLL	B/W/L	Rd (shift) → Rd
SHLR		Performs a logical shift on data in a general register. 1-bit or 2 bit shift is possible.
ROTL	B/W/L	Rd (rotate) → Rd
ROTR		Rotates data in a general register. 1-bit or 2 bit rotation is possible.
ROTXL	B/W/L	Rd (rotate) → Rd
ROTXR		Rotates data including the carry flag in a general register. 1-bit or 2 bit rotation is possible.

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

Table 2.7 Bit Manipulation Instructions (1)

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 Logically 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 Logically 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 Logically 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 Logically 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.

Note: * Size refers to the operand size.
B: Byte

Table 2.7 Bit Manipulation Instructions (2)

Instruction	Size*	Function
BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C Logically exclusive-ORs 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 Logically exclusive-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.
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: * Size refers to the operand size.
B: Byte

Table 2.8 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

Table 2.9 System Control Instructions

Instruction	Size*	Function
TRAPA	–	Starts trap-instruction exception handling.
RTE	–	Returns from an exception-handling routine.
SLEEP	–	Causes a transition to a power-down state.
LDC	B/W	(EAs) → CCR, (EAs) → EXR Moves the memory operand contents or immediate data to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper eight bits are valid.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a general register or memory operand. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper eight bits are valid.
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents with immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with immediate data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically exclusive-ORs the CCR or EXR contents with immediate data.
NOP	–	PC + 2 → PC Only increments the program counter.

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

Table 2.10 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.

2.6.2 Basic Instruction Formats

The H8S/2000 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 2.11 shows examples of instruction formats.

- 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.
- Register field
  Specifies a general register. Address registers are specified by 3-bit, and data registers by 3-bit or 4-bit. Some instructions have two register fields, and some have no register field.
- Effective address extension
  8-, 16-, or 32-bit specifying immediate data, an absolute address, or a displacement.
- Condition field
  Specifies the branching condition of Bcc instructions.

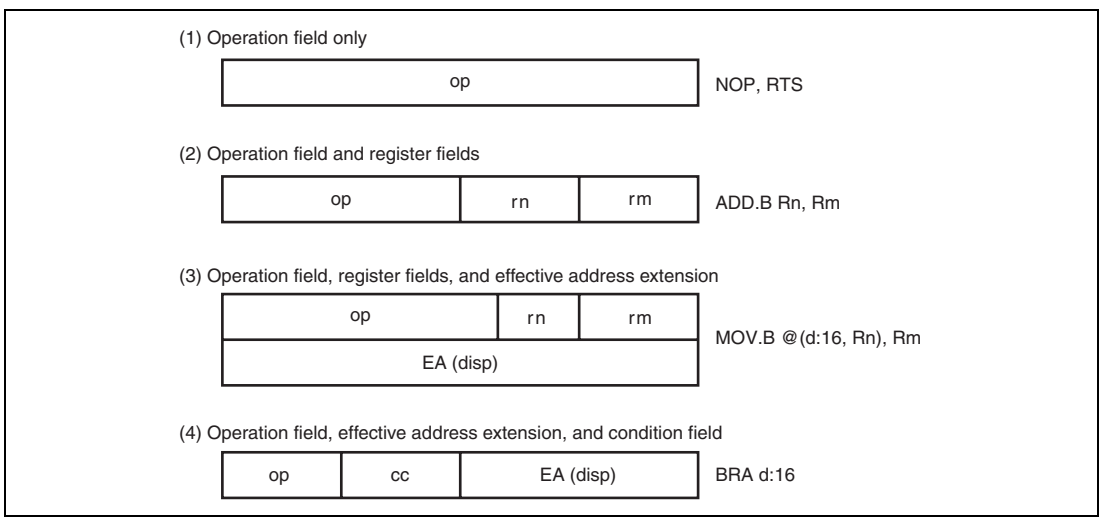


Figure 2.11 Instruction Formats (Examples)

2.7 Addressing Modes and Effective Address Calculation

The H8S/2000 CPU supports the eight addressing modes listed in table 2.11. Each instruction uses a subset of these addressing modes.

Arithmetic and logic operations instructions can use the register direct and immediate addressing modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions can use register direct, register indirect, or absolute addressing mode 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 2.11 Addressing Modes

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

2.7.1 Register Direct—Rn

The register field of the instruction code specifies an 8-, 16-, or 32-bit general register which contains 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.

2.7.2 Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn) which contains the address of a memory operand. If the address is a program instruction address, the lower 24 bits are valid and the upper eight bits are all assumed to be 0 (H'00).

2.7.3 Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn)

A 16-bit or 32-bit displacement contained in the instruction code is added to an address register (ERn) specified by the register field of the instruction, and the sum gives the address of a memory operand. A 16-bit displacement is sign-extended when added.

2.7.4 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) which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents and the sum is stored in the address register. The value added is 1 for byte access, 2 for word access, and 4 for longword access. For word or longword transfer instructions, 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 result becomes 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, and 4 for longword access. For word or longword transfer instructions, the register value should be even.

2.7.5 Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32

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), or 32 bits long (@aa:32). Table 2.12 indicates the accessible absolute address ranges.

To access data, the absolute address should be 8 bits (@aa:8), 16 bits (@aa:16), or 32 bits (@aa:32) long. For an 8-bit absolute address, the upper 24 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address, the upper 16 bits are a sign extension. For a 32-bit absolute address, the entire address space is accessed.

A 24-bit absolute address (@aa:24) indicates the address of a program instruction. The upper eight bits are all assumed to be 0 (H'00).

Table 2.12 Absolute Address Access Ranges

Absolute Address		Normal Mode	Advanced Mode
Data address	8 bits (@aa:8)	H'FF00 to H'FFFF	H'FFFF00 to H'FFFFFF
	16 bits (@aa:16)	H'0000 to H'FFFF	H'000000 to H'007FFF, H'FF8000 to H'FFFFFF
	32 bits (@aa:32)		H'000000 to H'FFFFFF
Program instruction address	24 bits (@aa:24)		

2.7.6 Immediate—#xx:8, #xx:16, or #xx:32

The 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) immediate data contained in an instruction code can be used directly as an operand.

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

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

This mode can be used by the Bcc and BSR instructions. An 8-bit or 16-bit displacement contained in the instruction code is sign-extended to 24-bit and added to the 24-bit address indicated by the PC value to generate a 24-bit branch address. Only the lower 24-bit of this branch address are valid; the upper eight bits are all assumed to be 0 (H'00). 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-byte (–63 to +64 words) or –32766 to +32768-byte (–16383 to +16384 words) from the branch instruction. The resulting value should be an even number.

2.7.8 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 which contains a branch address. The upper bits of the 8-bit absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF in normal mode, H'000000 to H'0000FF in advanced mode).

In normal mode, the memory operand is a word operand and the branch address is 16 bits long. In advanced mode, the memory operand is a longword operand, the first byte of which is assumed to be 0 (H'00).

Note that the top area of the address range in which the branch address is stored is also used for the exception vector area. For further details, see section 4, Exception Handling.

If an odd address is specified in word or longword memory access, or as a branch address, the least significant bit is regarded as 0, causing data to be accessed or the instruction code to be fetched at the address preceding the specified address. (For further information, see section 2.5.2, Memory Data Formats.)

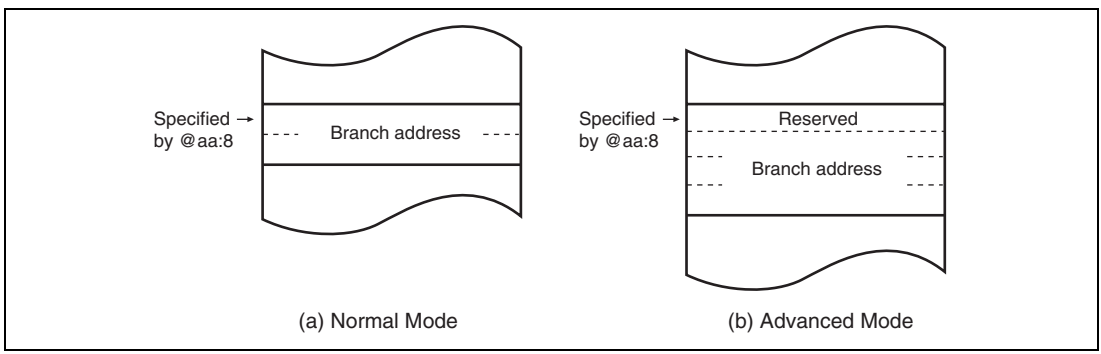


Figure 2.12 Branch Address Specification in Memory Indirect Addressing Mode

2.7.9 Effective Address Calculation

Table 2.13 indicates how effective addresses are calculated in each addressing mode. In normal mode, the upper eight bits of the effective address are ignored in order to generate a 16-bit address.

Table 2.13 Effective Address Calculation (1)

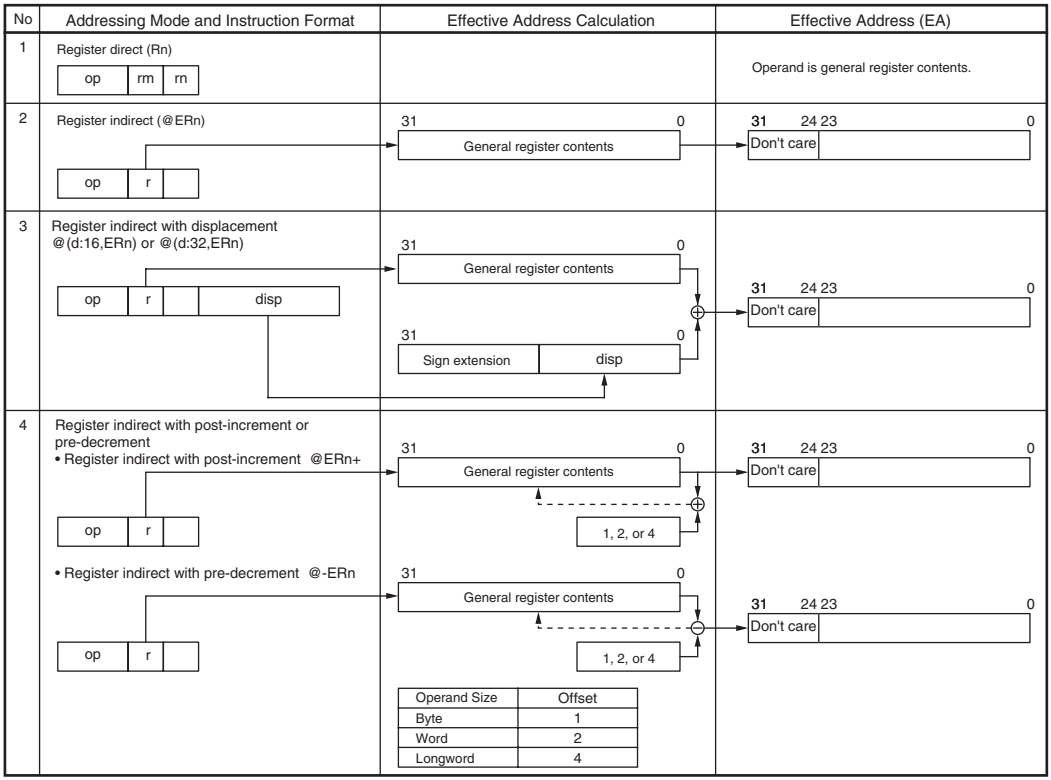
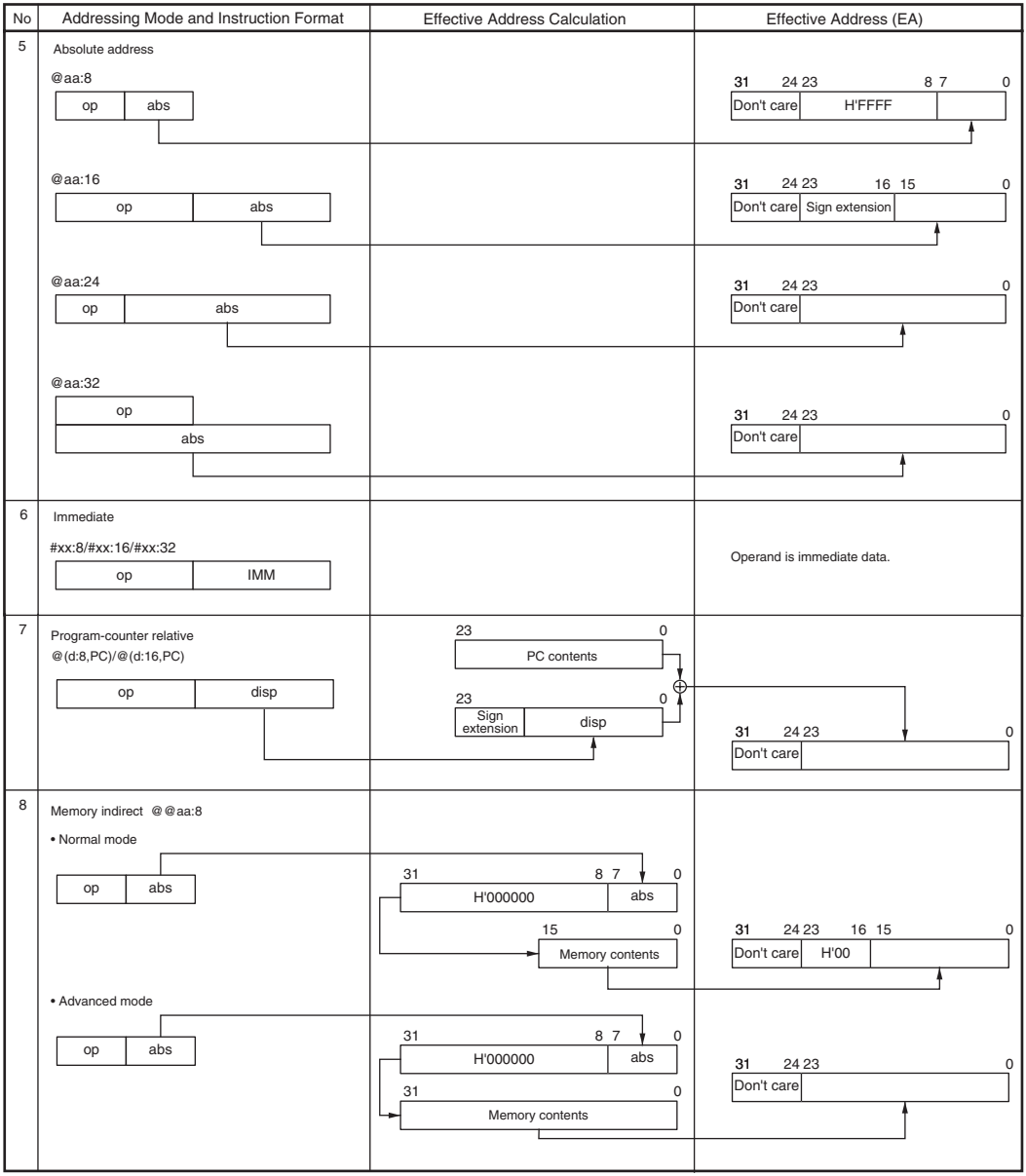


Table 2.13 Effective Address Calculation (2)



2.8 Processing States

The H8S/2000 CPU has five main processing states: the reset state, exception handling state, program execution state, bus-released state, and program stop state. Figure 2.13 indicates the state transitions.

- Reset state

  In this state the CPU and on-chip peripheral modules are all initialized and stopped. When the $\overline{\text{RES}}$ input goes low, all current processing stops and the CPU enters the reset state. All interrupts are masked in the reset state. Reset exception handling starts when the $\overline{\text{RES}}$ signal changes from low to high. For details, see section 4, Exception Handling.

  The reset state can also be entered by a watchdog timer overflow.

- Exception-handling state

  The exception-handling state is a transient state that occurs when the CPU alters the normal processing flow due to an exception source, such as, a reset, trace, interrupt, or trap instruction. The CPU fetches a start address (vector) from the exception vector table and branches to that address. For further details, see section 4, Exception Handling.

- Program execution state

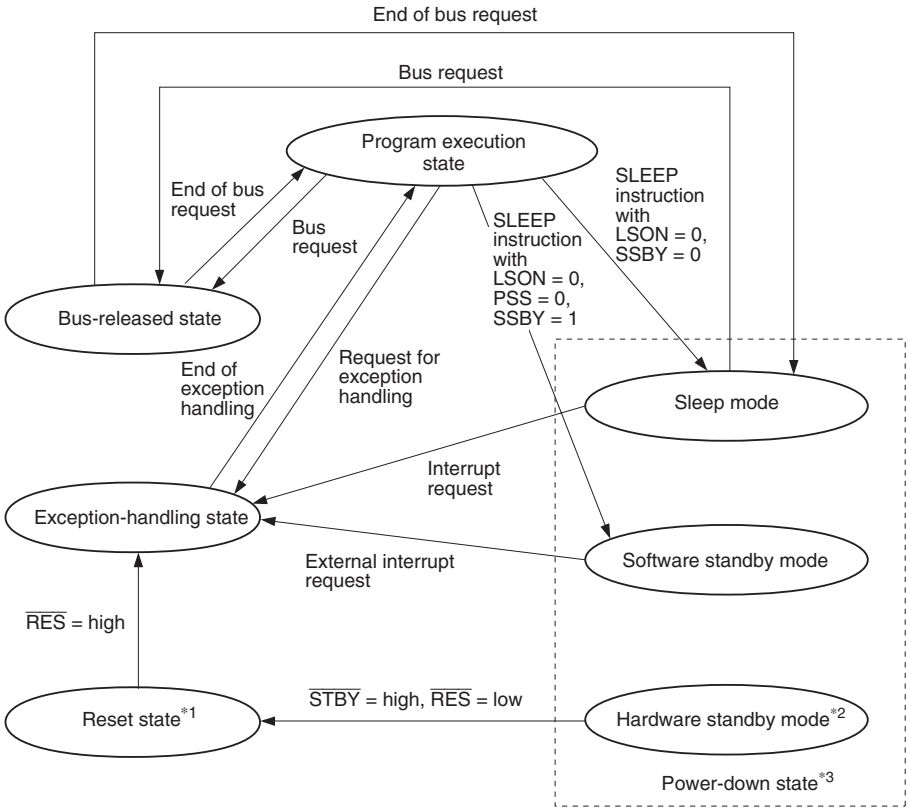
  In this state the CPU executes program instructions in sequence.

- Bus-released state

  In a product which has a bus master other than the CPU, the bus-released state occurs when the bus has been released in response to a bus request from a bus master other than the CPU. While the bus is released, the CPU halts operations. For details, see section 6, Bus Controller (BSC).

- Program stop state

  This is a power-down state in which the CPU stops operating. The program stop state occurs when a SLEEP instruction is executed or the CPU enters hardware standby mode. For details, see section 22, Power-Down Modes.



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs whenever $\overline{RES}$ goes low. A transition can also be made to the reset state when the watchdog timer overflows.
2. From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ goes low.
3. The power-down state also includes watch mode, subactive mode, subsleep mode, etc. For details, refer to section 22, Power-Down Modes.

Figure 2.13 State Transitions

2.9 Usage Notes

2.9.1 Note on TAS Instruction Usage

To use the TAS instruction, use registers ER0, ER1, ER4, and ER5.

The TAS instruction is not generated by the Renesas Technology H8S and H8/300 series C/C++ compilers. When the TAS instruction is used as a user-defined intrinsic function, registers ER0, ER1, ER4, and ER5 should be used.

2.9.2 Note on STM/LDM Instruction Usage

Since the ER7 register is used as the stack pointer in an STM/LDM instruction, it cannot be used as a register that allows save (STM) or restore (LDM) operation. Two to four registers can be saved/restored by single STM/LDM instruction. Available registers are listed below.

Two: ER0 and ER1, ER2 and ER3, ER4 and ER5
Three: ER0 to ER2, ER4 to ER6
Four: ER0 to ER3

The STM/LDM instruction with ER7 is not created by the Renesas Technology H8S or H8/300 series C/C++ compilers.

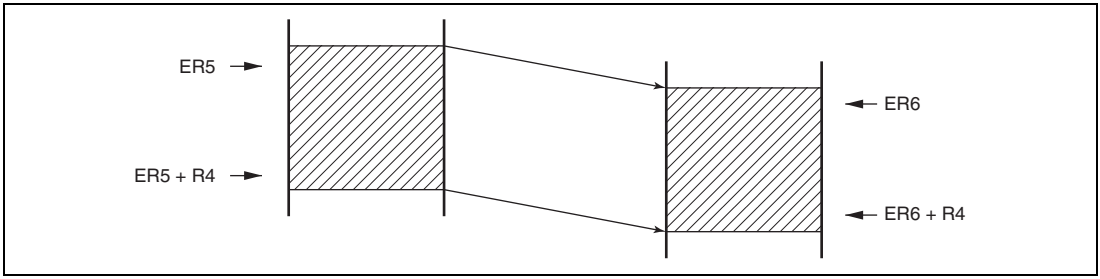
2.9.3 Note on Bit Manipulation Instructions

The BSET, BCLR, BNOT, BST, and BIST instructions read data in byte units, manipulate the data of the target bit, and write data in byte units. Special care is required when using these instructions in cases where a register containing a write-only bit is used or a bit is directly manipulated for a port.

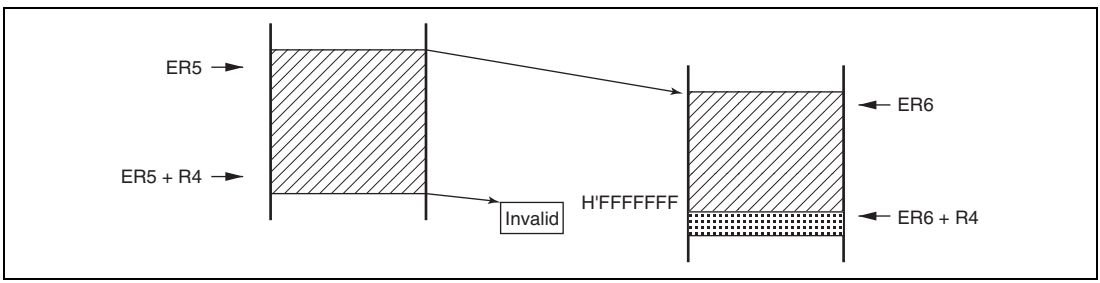
In addition, the BCLR instruction can be used to clear the flag of the internal I/O register. In this case, if the flag to be cleared has been set to 1 by an interrupt processing routine, the flag need not be read before executing the BCLR instruction.

2.9.4 EEPMOV Instruction

1. EEPMOV is a block-transfer instruction and transfers the byte size of data indicated by R4*, which starts from the address indicated by ER5, to the address indicated by ER6.



2. Set R4 and ER6 so that the end address of the destination address (value of ER6 + R4) does not exceed H'00FFFFFF (the value of ER6 must not change from H'00FFFFFF to H'01000000 during execution).



Section 3 MCU Operating Modes

3.1 Operating Mode Selection

This LSI supports five operating modes (modes 1 to 3, 6, and 7). The operating mode is determined by the setting of the mode pins (MD2, MD1, and MD0). Table 3.1 shows the MCU operating mode selection.

Table 3.1 MCU Operating Mode Selection

MCU Operating Mode	MD2*	MD1	MD0	CPU Operating Mode	Description	On-Chip ROM
1	0	0	1	Advanced	On-chip ROM disabled extended mode	Disabled
2	0	1	0	Advanced	Single-chip mode, On-chip ROM enabled extended mode	Enabled
3	0	1	1	Normal	Single-chip mode, On-chip ROM enabled extended mode	Enabled
6	1	1	0	Emulation	On-chip emulation mode	Enabled
7	1	1	1	Emulation	On-chip emulation mode	Enabled

Note: * This pin is not supported in SDIP-64 and QFP-64. The input value of MD2 is fixed to 0.

Mode 1 operates in on-chip ROM disabled extended mode. On-chip emulation is not supported.

Modes 2 and 3 operate in single-chip mode or on-chip ROM enabled extended mode.

Modes 0, 4, and 5 are not available in this LSI. Modes 6 and 7 are operating modes for a special purpose. Thus, mode pins should be set to enable mode 2 or 3 in the normal program execution state. Mode pin settings should not be changed during operation.

Mode 4 is a boot mode for programming or erasing the flash memory. For details, see section 19, Flash Memory (0.18-µm F-ZTAT Version).

Modes 6 and 7 are on-chip emulation modes. In these modes, this LSI is controlled by an on-chip emulator (E10A) via the H-UDI, thus enabling on-chip emulation. These modes are not supported in SDIP-64 and QFP-64.

3.2 Register Descriptions

The following registers are related to the operating modes. For details on the bus control register (BCR), see section 6.3.1, Bus Control Register (BCR).

- Mode control register (MDCR)
- System control register (SYSCR)
- Serial timer control register (STCR)

3.2.1 Mode Control Register (MDCR)

MDCR is used to set an operating mode and to monitor the current operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	EXPE	0	R/W	Extended Mode Enable Mode 1 This bit is fixed to 1 and cannot be modified. Mode 2 and 3 0: Single-chip mode 1: On-chip ROM enabled extended mode
6 to 3	—	All 0	R	Reserved The initial value should not be changed.
2	MDS2	—*[1]	R	Mode Select 2 to 0 These bits indicate the input levels at mode pins (MD2, MD1, and MD0) (the current operating mode). The MDS2, MDS1, and MDS0 bits correspond to the MD2, MD1, and MD0 pins, respectively. These bits are read-only bits and cannot be written to. The input levels of the mode pins (MD2, MD1, and MD0) are latched into these bits when MDCR is read. These latches are canceled by a reset.
1	MDS1	—*[2]	R	
0	MDS0	—*[2]	R	

Notes: 1. The initial value is determined by the MD2 pin. The MD2 pin is not supported in SDIP-64 and QFP-64. This bit is always read as 0.
2. The initial value is determined by the MD1 and MD0 pins.

3.2.2 System Control Register (SYSCR)

SYSCR selects system pin function, monitors a reset source, selects the interrupt control mode and the detection edge for NMI, enables or disables access to the on-chip peripheral module registers, and enables or disables the on-chip RAM address space.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved The initial value should not be changed.
6	IOSE	0	R/W	IOS Enable Controls $\overline{AS}$/$\overline{IOS}$ pin function in extended mode. 0: AS pin Low output when accessing the external area. 1: IOS pin Low output when accessing the specified address from H'(FF)F000 to H'(FF)F7FF
5	INTM1	0	R	Interrupt Control Select Mode 1, 0
4	INTM0	0	R/W	These bits select the interrupt control mode of the interrupt controller. For details on the interrupt control modes, see section 5.6, Interrupt Control Modes and Interrupt Operation. 00: Interrupt control mode 0 01: Interrupt control mode 1 10: Setting prohibited 11: Setting prohibited
3	XRST	1	R	External Reset Indicates the reset source. A reset is caused by an external reset input, or when the watchdog timer overflows. 0: A reset is caused when the watchdog timer overflows 1: A reset is caused by an external reset
2	NMIEG	0	R/W	NMI Edge Select Selects the valid edge of the NMI interrupt input. 0: An interrupt is requested at the falling edge of NMI input 1: An interrupt is requested at the rising edge of NMI input

Bit	Bit Name	Initial Value	R/W	Description
1	—	0	R/W	Reserved The initial value should not be changed.
0	RAME	1	R/W	RAM Enable Enables or disables on-chip RAM. 0: On-chip RAM is disabled 1: On-chip RAM is enabled

3.2.3 Serial Timer Control Register (STCR)

STCR enables or disables register access, IIC operating mode, and on-chip flash memory, and selects the input clock of the timer counter.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The initial value should not be changed.
6	IICX1	0	R/W	I^2C Transfer Rate Select 1, 0
5	IICX0	0	R/W	These bits control the IIC operation. These bits select the transfer rate in master mode together with bits CKS2 to CKS0 in the I^2C bus mode register (ICMR). For details on the transfer rate, see table 16.3.

Bit	Bit Name	Initial Value	R/W	Description
4	IICE	0	R/W	I^2C Master Enable Enables or disables CPU access to IIC registers (ICCR, ICSR, ICDR/SARX, ICMR/SAR, and DDCSWR), PWMX registers (DADRAH/DACR, DADRAL, DADRBH/DACNTH, and DADRBL/DACNTL), and SCI registers (SMR, BRR, and SCMR). 0: SCI_1 registers are accessed in areas from H'(FF)FF88 to H'(FF)FF89 and from H'(FF)FF8E to H'(FF)FF8F. SCI_2 registers are accessed in areas from H'(FF)FFA0 to H'(FF)FFA1 and from H'(FF)FFA6 to H'(FF)FFA7. Access is prohibited in areas from H'(FF)FFD8 to H'(FF)FFD9 and from H'(FF)FFDE to H'(FF)FFDF. 1: IIC_1 registers are accessed in areas from H'(FF)FF88 to H'(FF)FF89 and from H'(FF)FF8E to H'(FF)FF8F. PWMX registers are accessed in areas from H'(FF)FFA0 to H'(FF)FFA1 and from H'(FF)FFA6 to H'(FF)FFA7. IIC_0 registers are accessed in areas from H'(FF)FFD8 to H'(FF)FFD9 and from H'(FF)FFDE to H'(FF)FFDF. DDCSWR is accessed in areas of H'(FF)FEE6.
3	FLSHE	0	R/W	Flash Memory Control Register Enable Enables or disables CPU access for flash memory registers (FCCS, FPCS, FECS, FKEY, FMATS, and FTDAR), power-down state control registers (SBYCR, LPWRCR, MSTPCRH, and MSTPCRL), and on-chip peripheral module control registers (BCR2, WSCR, PCSR, and SYSCR2). 0: Control registers of power-down state and peripheral modules are accessed in an area from H'(FF)FF80 to H'(FF)FF87. Area from H'(FF)FEA8 to H'(FF)FEAE is reserved. 1: Control registers of flash memory are accessed in an area from H'(FF)FEA8 to H'(FF)FEAE. Area from H'(FF)FF80 to H'(FF)FF87 is reserved.
2	—	0	R/(W)	Reserved The initial value should not be changed.
1	ICKS1	0	R/W	Internal Clock Source Select 1, 0
0	ICKS0	0	R/W	These bits select a clock to be input to the timer counter (TCNT) and a count condition together with bits CKS2 to CKS0 in the timer control register (TCR). For details, see section 13.3.4, Timer Control Register (TCR).

3.3 Operating Mode Descriptions

3.3.1 Mode 1

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled.

Ports 1 and 2 function as an address bus, port 3 functions as a data bus, and part of port 4 carries bus control signals. However, as these series have a maximum of 16 address outputs, an external address can be specified correctly only when the I/O strobe function of the $\overline{AS}$/$\overline{IOS}$ pin is used.

Mode 1 does not support on-chip emulation.

3.3.2 Mode 2

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled. After a reset, single-chip mode is set, and the EXPE bit in MDCR must be set to 1 in order to use external addresses. However, as these series have a maximum of 16 address outputs, an external address can be specified correctly only when the I/O strobe function of the $\overline{AS}$/$\overline{IOS}$ pin is used.

When the EXPE bit in MDCR is set to 1, ports 1 and 2 function as input ports after a reset. They can be set to output addresses by setting the corresponding bits in the data direction register (DDR) to 1. Port 3 functions as a data bus, and part of port 4 carries bus control signals.

3.3.3 Mode 3

The CPU can access a 64-Kbyte address space in normal mode. The on-chip ROM is enabled. After a reset, single-chip mode is set, and the EXPE bit in MDCR must be set to 1 in order to use external addresses.

When the EXPE bit in MDCR is set to 1, ports 1 and 2 function as input ports after a reset. They can be set to output addresses by setting the corresponding bits in the data direction register (DDR) to 1. Port 3 functions as a data bus, and part of port 4 carries bus control signals.

In this operating mode, the amount of on-chip ROM is limited to 56 Kbytes and the amount of on-chip RAM is limited to 4 Kbytes.

3.4 Pin Functions in Each Operating Mode

The pin functions of ports 1 to 4 vary depending on the operating mode. Table 3.2 shows their functions in each operating mode.

Table 3.2 Pin Functions in Each Mode

Port		Mode 1	Mode 2	Mode 3
Port 1		A	P*/A	P*/A
Port 2		A	P*/A	P*/A
Port 3		D	P*/D	P*/D
Port 4	P47	P*/C	P*/C	P*/C
	P46	C*/P	P*/C	P*/C
	P45 to P43	C	P*/C	P*/C
	P42 to P40	P	P	P

[Legend]

P: I/O port

A: Address bus output

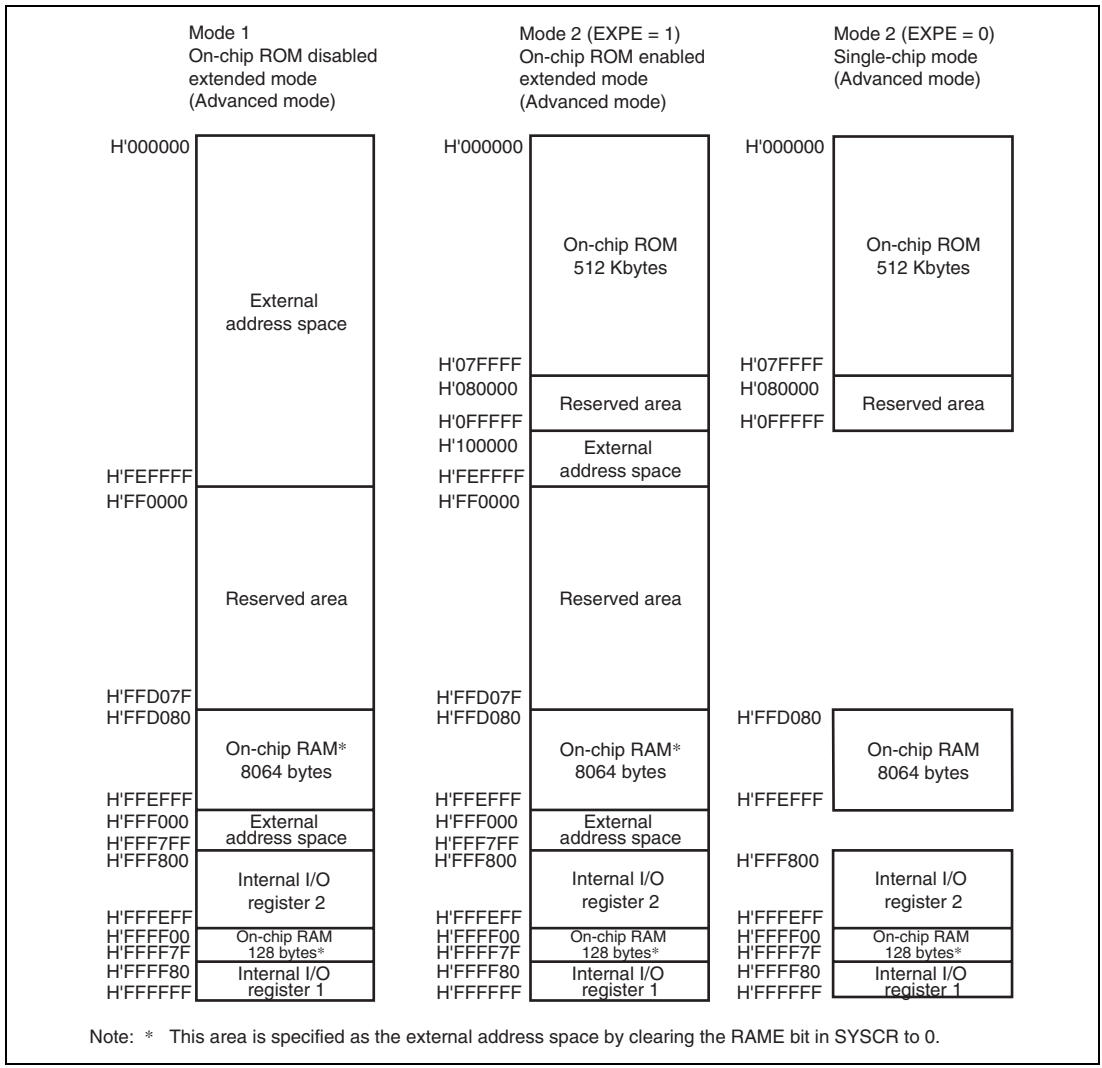
D: Data bus I/O

C: Control signals, clock I/O

*: After a reset

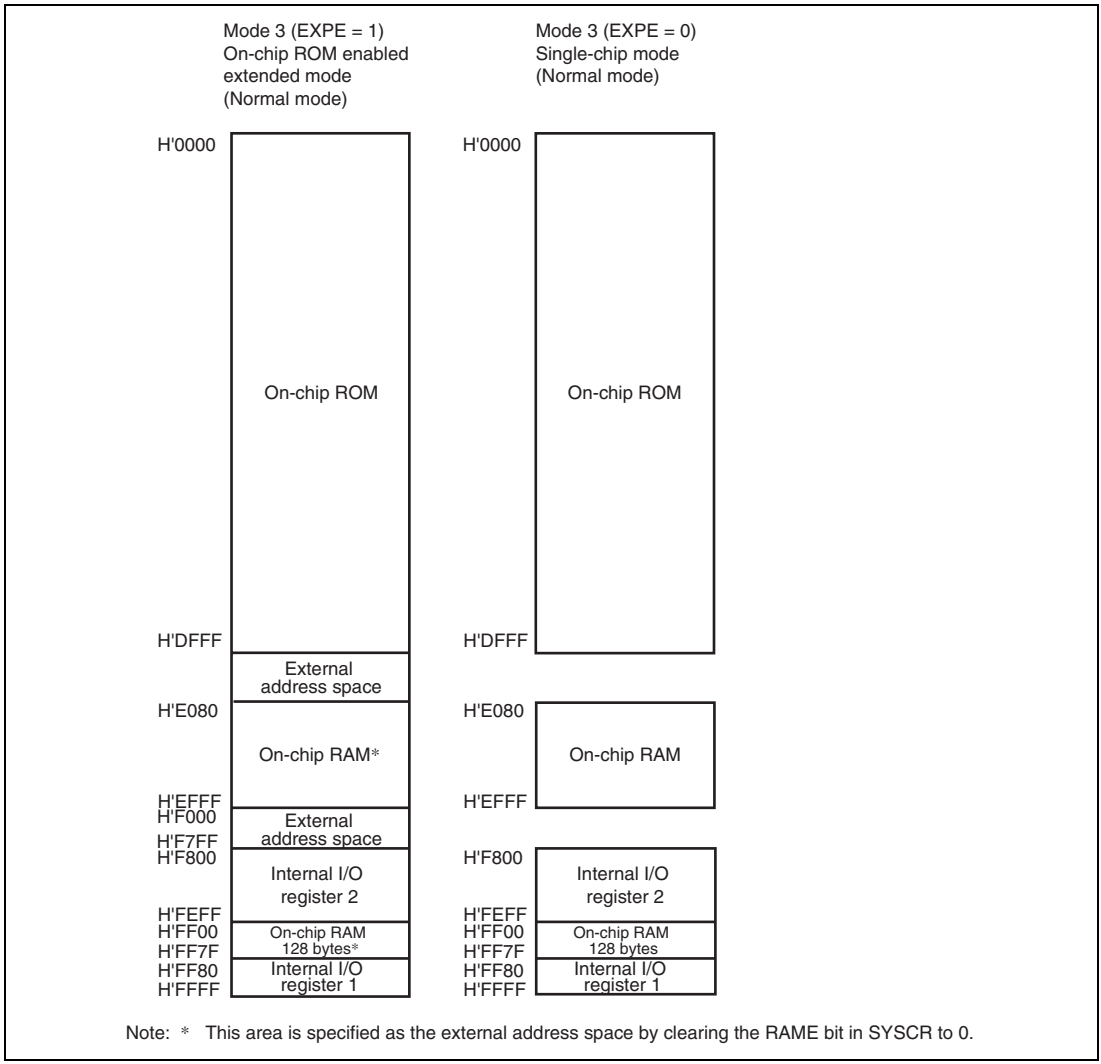
3.5 Address Map

Figures 3.1 and 3.2 show the address map in each operating mode.



Note: * This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.

Figure 3.1 Address Map (1)



Note: * This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.

Figure 3.2 Address Map (2)

Section 4 Exception Handling

4.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, interrupt, direct transition, or trap instruction. Exception handling is prioritized as shown in table 4.1. If two or more exceptions occur simultaneously, they are accepted and processed in order of priority.

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High ↑ Low	Reset	Starts immediately after a low-to-high transition of the $\overline{\text{RES}}$ pin, or when the watchdog timer overflows.
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued. Interrupt detection is not performed on completion of ANDC, ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.
	Direct transition	Starts when a direct transition occurs as the result of SLEEP instruction execution.
	Trap instruction	Started by execution of a trap (TRAPA) instruction. Trap instruction exception handling requests are accepted at all times in the program execution state.

4.2 Exception Sources and Exception Vector Table

Different vector addresses are assigned to exception sources. Table 4.2 lists the correspondence between exception sources and vector addresses.

Table 4.2 Exception Handling Vector Table

Exception Source		Vector Number	Vector Address	
			Normal Mode	Advanced Mode
Reset		0	H'0000 to H'0001	H'000000 to H'000003
Reserved for system use		1 ⏐ 5	H'0002 to H'0003 ⏐ H'000A to H'000B	H'000004 to H'000007 ⏐ H'000014 to H'000017
Direct transition		6	H'000C to H'000D	H'000018 to H'00001B
External interrupt (NMI)		7	H'000E to H'000F	H'00001C to H'00001F
Trap instruction (four sources)		8	H'0010 to H'0011	H'000020 to H'000023
		9	H'0012 to H'0013	H'000024 to H'000027
		10	H'0014 to H'0015	H'000028 to H'00002B
		11	H'0016 to H'0017	H'00002C to H'00002F
Reserved for system use		12 ⏐ 15	H'0018 to H'0019 ⏐ H'001E to H'001F	H'000030 to H'000033 ⏐ H'00003C to H'00003F
External interrupt	IRQ0	16	H'0020 to H'0021	H'000040 to H'000043
	IRQ1	17	H'0022 to H'0023	H'000044 to H'000047
	IRQ2	18	H'0024 to H'0025	H'000048 to H'00004B
	IRQ3	19	H'0026 to H'0027	H'00004C to H'00004F
	IRQ4	20	H'0028 to H'0029	H'000050 to H'000053
	IRQ5	21	H'002A to H'002B	H'000054 to H'000057
	IRQ6	22	H'002C to H'002D	H'000058 to H'00005B
	IRQ7	23	H'002E to H'002F	H'00005C to H'00005F
Internal interrupt*		24 ⏐ 127	H'0030 to H'0031 ⏐ H'00FE to H'00FF	H'000060 to H'000063 ⏐ H'0001FC to H'0001FF

Note: * For details on the internal interrupt vector table, see section 5.5, Interrupt Exception Handling Vector Table.

4.3 Reset

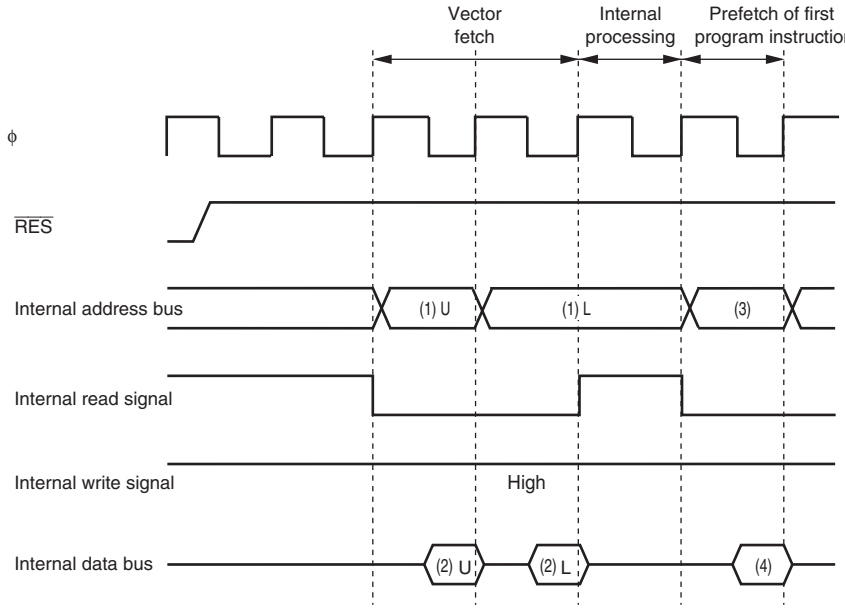
A reset has the highest exception priority. When the $\overline{\text{RES}}$ pin goes low, all processing halts and this LSI enters the reset state. To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-on. To reset the chip during operation, hold the $\overline{\text{RES}}$ pin low for at least 20 states. A reset initializes the internal state of the CPU and the registers of on-chip peripheral modules. The chip can also be reset by overflow of the watchdog timer. For details, see section 14, Watchdog Timer (WDT).

4.3.1 Reset Exception Handling

When the $\overline{\text{RES}}$ pin goes high after being held low for the necessary time, 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 is read and transferred to the PC, and then program execution starts from the address indicated by the PC.

Figure 4.1 shows an example of the reset sequence.



(1) Reset exception handling vector address (1) U = H'000000 (1) L = H'000002
(2) Start address (contents of reset exception handling vector address)
(3) Start address ((3) = (2)U + (2)L)
(4) First program instruction

Figure 4.1 Reset Sequence (Mode 2)

4.3.2 Interrupts Immediately after Reset

If an interrupt is accepted immediately after a reset and before the stack pointer (SP) is initialized, the PC and CCR will not be saved correctly, leading to a program crash. To prevent this, all interrupt requests, including NMI, are disabled immediately after a reset. Since the first instruction of a program is always executed immediately after a reset, make sure that this instruction initializes the SP (example: MOV.L #xx: 32, SP).

4.3.3 On-Chip Peripheral Modules after Reset is Cancelled

After a reset is cancelled, the module stop control registers (MSTPCRH, MSTPCRL, MSTPCRA, and MSTPCRB) are initialized, and all modules except the DTC operate in module stop mode. Therefore, the registers of on-chip peripheral modules cannot be read from or written to. To read from and write to these registers, clear module stop mode. For details on module stop mode, see section 22, Power-Down Modes.

4.4 Interrupt Exception Handling

Interrupts are controlled by the interrupt controller. The sources to start interrupt exception handling are external interrupt sources (NMI, IRQ15 to IRQ0, KIN15 to KIN0, and WUE15 to WUE0) and internal interrupt sources from the on-chip peripheral modules. NMI is an interrupt with the highest priority. For details, see section 5, Interrupt Controller.

Interrupt exception handling is conducted as follows:

1. The values in the program counter (PC) and condition code register (CCR) are saved in the stack.
2. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution starts from that address.

4.5 Trap Instruction Exception Handling

Trap instruction exception handling starts when a TRAPA instruction is executed. Trap instruction exception handling can be executed at all times in the program execution state.

Trap instruction exception handling is conducted as follows:

1. The values in the program counter (PC) and condition code register (CCR) are saved in the stack.
2. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution starts from that address.

The TRAPA instruction fetches a start address from a vector table corresponding to a vector number from 0 to 3, as specified in the instruction code.

Table 4.3 shows the status of CCR after execution of trap instruction exception handling.

Table 4.3 Status of CCR after Trap Instruction Exception Handling

	CCR	
Interrupt Control Mode	**I**	**UI**
0	Set to 1	Retains value prior to execution
1	Set to 1	Set to 1

4.6 Stack Status after Exception Handling

Figure 4.2 shows the stack after completion of trap instruction exception handling and interrupt exception handling.

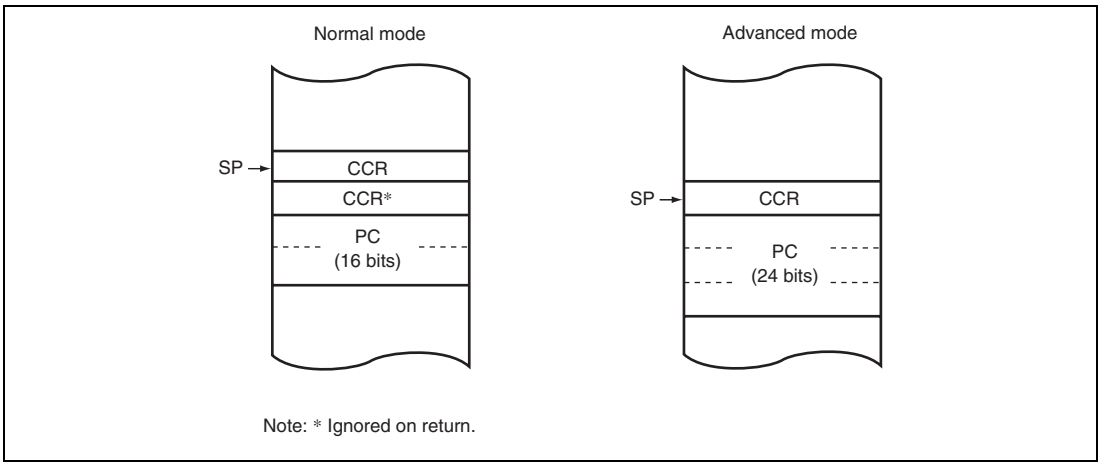


Figure 4.2 Stack Status after Exception Handling

4.7 Usage Note

When accessing word data or longword data, this LSI assumes that the lowest address bit is 0. The stack should always be accessed in words or longwords, and the value of the stack pointer (SP: ER7) should always be kept even.

Use the following instructions to save registers:

```
PUSH.W   Rn    (or MOV.W Rn, @-SP)
PUSH.L   ERn   (or MOV.L ERn, @-SP)
```

Use the following instructions to restore registers:

```
POP.W    Rn    (or MOV.W @SP+, Rn)
POP.L    ERn   (or MOV.L @SP+, ERn)
```

Setting SP to an odd value may lead to a malfunction. Figure 4.3 shows an example of what occurs when the SP value is odd.

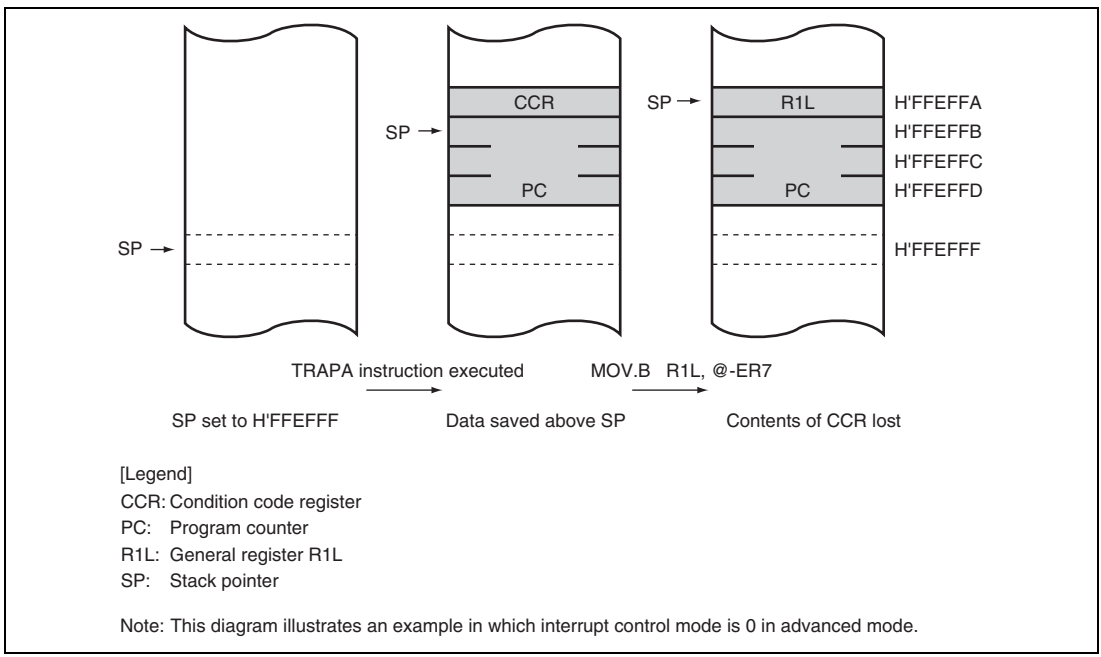


Figure 4.3 Operation when SP Value Is Odd

Section 5 Interrupt Controller

5.1 Features

- Two interrupt control modes

  Two interrupt control modes can be set by means of the INTM1 and INTM0 bits in the system control register (SYSCR).
- Priorities settable with ICR

  An interrupt control register (ICR) is provided for setting in each module interrupt priority levels for all interrupt requests excluding NMI and address breaks.
- Three-level interrupt mask control

  By means of the interrupt control mode, I and UI bits in CCR and ICR, 3-level interrupt mask control is performed.
- Independent vector addresses

  All interrupt sources are assigned independent vector addresses, making it unnecessary for the source to be identified in the interrupt handling routine.
- Nine external interrupt pins

  NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or falling edge detection can be selected for NMI. Falling-edge, rising-edge, or both-edge detection, or level sensing, can be independently selected for $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.
- DTC control

  The DTC can be activated by an interrupt request.
- General ports for $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input are selectable

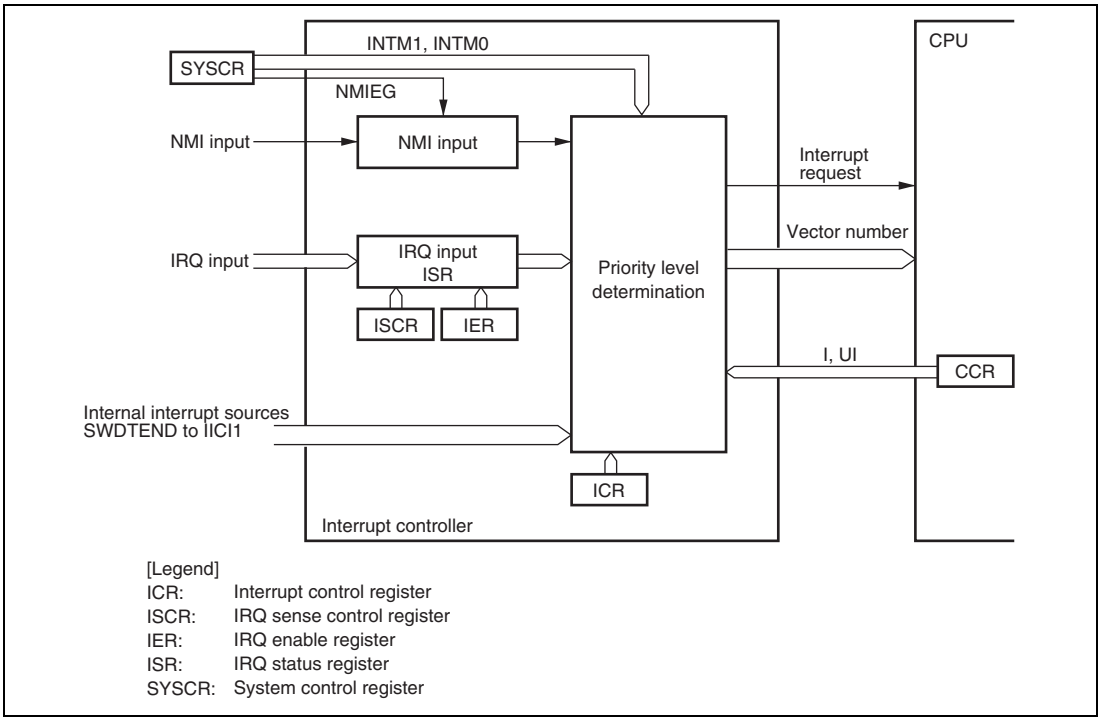


Figure 5.1 Block Diagram of Interrupt Controller

5.2 Input/Output Pins

Table 5.1 summarizes the pins of the interrupt controller.

Table 5.1 Pin Configuration

Symbol	I/O	Function
NMI	Input	Nonmaskable external interrupt pin Rising edge or falling edge can be selected
$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$	Input	Maskable external interrupt pins Rising-edge, falling-edge, or both-edge detection, or level-sensing can be selected individually for each pin.

5.3 Register Descriptions

The interrupt controller has the following registers. For details on the system control register (SYSCR), see section 3.2.2, System Control Register (SYSCR).

- Interrupt control registers A to D (ICRA to ICRD)
- Address break control register (ABRKCR)
- Break address registers A to C (BARA to BARC)
- IRQ sense control registers (ISCRH, ISCRL)
- IRQ enable registers (IER)
- IRQ status registers (ISR)

5.3.1 Interrupt Control Registers A to D (ICRA to ICRD)

The ICR registers set interrupt control levels for interrupts other than NMI. The correspondence between interrupt sources and ICRA to ICRD settings is shown in table 5.2.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ICRn7 to ICRn0	All 0	R/W	Interrupt Control Level 0: Corresponding interrupt source is interrupt control level 0 (no priority) 1: Corresponding interrupt source is interrupt control level 1 (priority)

Note: n: A to D

Table 5.2 Correspondence between Interrupt Source and ICR

		Register			
Bit	**Bit Name**	**ICRA**	**ICRB**	**ICRC**	**ICRD**
7	ICRn7	IRQ0	A/D converter	SCI_0	TCM_0
6	ICRn6	IRQ1	FRT	SCI_1	TCM_1
5	ICRn5	IRQ2, IRQ3	—	—	—
4	ICRn4	IRQ4, IRQ5	—	IIC_0	—
3	ICRn3	IRQ6, IRQ7	TMR_0	IIC_1	—
2	ICRn2	DTC	TMR_1	—	—
1	ICRn1	WDT_0	TMR_X, TMR_Y	—	—
0	ICRn0	WDT_1	—	—	—

Note: n: A to D
—: Reserved. The initial value should not be changed.

5.3.2 Address Break Control Register (ABRKCR)

ABRKCR controls the address breaks. When both the CMF flag and BIE bit are set to 1, an address break is requested.

Bit	Bit Name	Initial Value	R/W	Description
7	CMF	Undefined	R	Condition Match Flag Address break source flag. Indicates that an address specified by BARA to BARC is prefetched. [Clearing condition] When an exception handling is executed for an address break interrupt. [Setting condition] When an address specified by BARA to BARC is prefetched while the BIE bit is set to 1.
6 to 1	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
0	BIE	0	R/W	Break Interrupt Enable Enables or disables address break. 0: Disabled 1: Enabled

5.3.3 Break Address Registers A to C (BARA to BARC)

The BAR registers specify an address that is to be a break address. An address in which the first byte of an instruction exists should be set as a break address. In normal mode, addresses A23 to A16 are not compared.

- BARA

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	A23 to A16	All 0	R/W	Addresses 23 to 16 The A23 to A16 bits are compared with A23 to A16 in the internal address bus.

- BARB

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	A15 to A8	All 0	R/W	Addresses 15 to 8 The A15 to A8 bits are compared with A15 to A8 in the internal address bus.

- BARC

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	A7 to A1	All 0	R/W	Addresses 7 to 1 The A7 to A1 bits are compared with A7 to A1 in the internal address bus.
0	—	0	R	Reserved This bit is always read as 0 and cannot be modified.

5.3.4 IRQ Sense Control Registers (ISCRH, ISCRL)

The ISCR registers select the source that generates an interrupt request at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.

- ISCRH

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7SCB	0	R/W	IRQn Sense Control B
6	IRQ7SCA	0	R/W	IRQn Sense Control A
5	IRQ6SCB	0	R/W	BA
4	IRQ6SCA	0	R/W	00: Interrupt request generated at low level of $\overline{\text{IRQn}}$ input
3	IRQ5SCB	0	R/W	01: Interrupt request generated at falling edge of $\overline{\text{IRQn}}$ input
2	IRQ5SCA	0	R/W	10: Interrupt request generated at rising edge of $\overline{\text{IRQn}}$ input
1	IRQ4SCB	0	R/W	11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQn}}$ input
0	IRQ4SCA	0	R/W	(n = 7 to 4)

- ISCRL

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ3SCB	0	R/W	IRQn Sense Control B
6	IRQ3SCA	0	R/W	IRQn Sense Control A
5	IRQ2SCB	0	R/W	BA
4	IRQ2SCA	0	R/W	00: Interrupt request generated at low level of $\overline{\text{IRQn}}$ input
3	IRQ1SCB	0	R/W	01: Interrupt request generated at falling edge of $\overline{\text{IRQn}}$ input
2	IRQ1SCA	0	R/W	10: Interrupt request generated at rising edge of $\overline{\text{IRQn}}$ input
1	IRQ0SCB	0	R/W	11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQn}}$ input
0	IRQ0SCA	0	R/W	(n = 3 to 0)

5.3.5 IRQ Enable Registers (IER)

IER enables and disables interrupt requests IRQ7 to IRQ0.

- IER

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7E	0	R/W	IRQn Enable
6	IRQ6E	0	R/W	The IRQn interrupt request is enabled when this bit is 1.
5	IRQ5E	0	R/W	
4	IRQ4E	0	R/W	(n = 7 to 0)
3	IRQ3E	0	R/W	
2	IRQ2E	0	R/W	
1	IRQ1E	0	R/W	
0	IRQ0E	0	R/W	

5.3.6 IRQ Status Registers (ISR)

ISR is a flag register that indicates the status of IRQ7 to IRQ0 interrupt requests.

- ISR

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7F	0	R/(W)*	[Setting condition]
6	IRQ6F	0	R/(W)*	When the interrupt source selected by the ISCR registers occurs
5	IRQ5F	0	R/(W)*	
4	IRQ4F	0	R/(W)*	[Clearing conditions]
3	IRQ3F	0	R/(W)*	• When writing 0 to IRQnF flag after reading IRQnF = 1
2	IRQ2F	0	R/(W)*	
1	IRQ1F	0	R/(W)*	• When interrupt exception handling is executed when low-level detection is set and $\overline{\text{IRQn}}$ input is high
0	IRQ0F	0	R/(W)*	• When IRQn interrupt exception handling is executed when falling-edge, rising-edge, or both-edge detection is set
				(n = 7 to 0)

Note: * Only 0 can be written for clearing the flag.

5.4 Interrupt Sources

5.4.1 External Interrupt Sources

The interrupt sources of external interrupts are NMI and IRQ7 to IRQ0. These interrupts can be used to restore this LSI from software standby mode.

(1) NMI Interrupt

The nonmaskable external interrupt NMI is the highest-priority interrupt, and is always accepted regardless of the interrupt control mode or the status of the CPU interrupt mask bits. The NMIEG bit in SYSCR can be used to select whether an interrupt is requested at a rising edge or falling edge on the NMI pin.

(2) IRQ7 to IRQ0 Interrupts

Interrupts IRQ7 to IRQ0 are requested by an input signal at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$. Interrupts IRQ7 to IRQ0 have the following features:

- The interrupt exception handling for interrupt requests IRQ7 to IRQ0 can be started at an independent vector address.
- Using ISCR, it is possible to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges, at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.
- Enabling or disabling of interrupt requests IRQ7 to IRQ0 can be selected with IER.
- The status of interrupt requests IRQ7 to IRQ0 is indicated in ISR. ISR flags can be cleared to 0 by software.

When the interrupts are requested while IRQ7 to IRQ0 interrupt requests are generated at low level of $\overline{\text{IRQn}}$ input, hold the corresponding $\overline{\text{IRQ}}$ input at low level until the interrupt handling starts. Then put the relevant $\overline{\text{IRQ}}$ input back to high level within the interrupt handling routine and clear the IRQnF bit (n = 7 to 0) in ISR to 0. If the relevant IRQ input is put back to high level before the interrupt handling starts, the relevant interrupt may not be executed.

The detection of IRQ7 to IRQ0 interrupts does not depend on whether the relevant pin has been set for input or output. Therefore, when a pin is used as an external interrupt input pin, clear the DDR bit of the corresponding port to 0 so it is not used as an I/O pin for another function.

A block diagram of interrupts IRQ15 to IRQ0 is shown in figure 5.2.

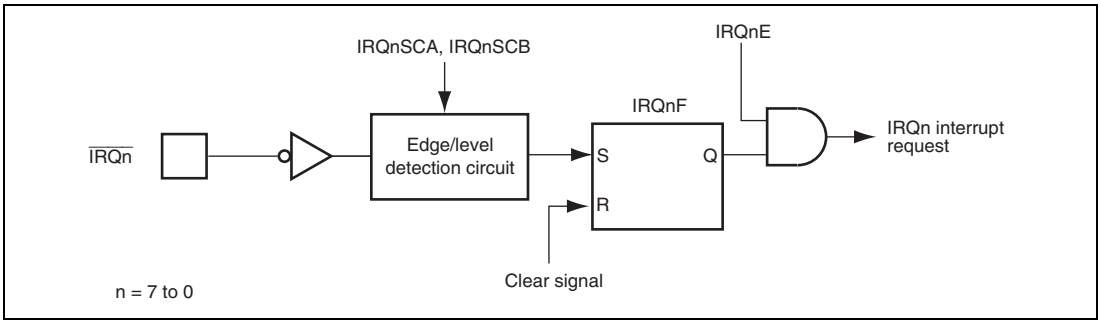


Figure 5.2 Block Diagram of Interrupts IRQ7 to IRQ0

5.4.2 Internal Interrupt Sources

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

1. For each on-chip peripheral module there are flags that indicate the interrupt request status, and enable bits that individually select enabling or disabling of these interrupts. When the enable bit for a particular interrupt source is set to 1, an interrupt request is sent to the interrupt controller.
2. The control level for each interrupt can be set by ICR.
3. The DTC can be activated by an interrupt request from an on-chip peripheral module.
4. An interrupt request that activates the DTC is not affected by the interrupt control mode or the status of the CPU interrupt mask bits.

5.5 Interrupt Exception Handling Vector Tables

Tables 5.3 lists interrupt exception handling sources, vector addresses, and interrupt priorities. For default priorities, the lower the vector number, the higher the priority. Modules set at the same priority will conform to their default priorities. Priorities within a module are fixed.

An interrupt control level can be specified for a module to which an ICR bit is assigned. Interrupt requests from modules that are set to interrupt control level 1 (priority) by the interrupt control level and the I and UI bits in CCR are given priority and processed before interrupt requests from modules that are set to interrupt control level 0 (no priority).

Table 5.3 Interrupt Sources, Vector Addresses, and Interrupt Priorities

Origin of Interrupt Source	Name	Vector Number	Vector Address: Normal Mode	Vector Address: Advanced Mode	ICR	Priority
External pin	NMI	7	H'000E	H'00001C	—	High
	IRQ0	16	H'0020	H'000040	ICRA7	
	IRQ1	17	H'0022	H'000044	ICRA6	
	IRQ2	18	H'0024	H'000048	ICRA5	
	IRQ3	19	H'0026	H'00004C		
	IRQ4	20	H'0028	H'000050	ICRA4	
	IRQ5	21	H'002A	H'000054		
	IRQ6	22	H'002C	H'000058	ICRA3	
	IRQ7	23	H'002E	H'00005C		
DTC	SWDTEND (Software activation data transfer end)	24	H'0030	H'000060	ICRA2	
WDT_0	WOVI0 (Interval timer)	25	H'0032	H'000064	ICRA1	
WDT_1	WOVI1 (Interval timer)	26	H'0034	H'000068	ICRA0	
—	Address break	27	H'0036	H'00006C	—	
A/D converter	ADI (A/D conversion end)	28	H'0038	H'000070	ICRB7	
—	Reserved for system use	29 ⎮ 47	H'003A ⎮ H'005E	H'000074 ⎮ H'0000BC	—	
FRT	ICIA (Input capture A)	48	H'0060	H'0000C0	ICRB6	
	ICIB (Input capture B)	49	H'0062	H'0000C4		
	ICIC (Input capture C)	50	H'0064	H'0000C8		
	ICID (Input capture D)	51	H'0066	H'0000CC		
	OCIA (Output compare A)	52	H'0068	H'0000D0		
	OCIB (Output compare B)	53	H'006A	H'0000D4		
	FOVI (Overflow)	54	H'006C	H'0000D8		
	Reserved for system use	55	H'006E	H'0000DC		Low

Origin of Interrupt Source	Name	Vector Number	Vector Address: Normal Mode	Vector Address: Advanced Mode	ICR	Priority
TCM_0	TICI0 (Input capture)	56	H'0070	H'0000E0	ICRD7	High
	TCMI0 (Compare match)	57	H'0072	H'0000E4		
	TOVMI0 (MAX cycle overflow)	58	H'0074	H'0000E8		
	TOVI0 (Overflow)	59	H'0076	H'0000EC		
TCM_1	TICI1 (Input capture)	60	H'0078	H'0000F0	ICRD6	
	TCMI1 (Compare match)	61	H'007A	H'0000F4		
	TOVMI1 (MAX overflow)	62	H'007C	H'0000F8		
	TOVI1 (Overflow)	63	H'007E	H'0000FC		
TMR_0	CMIA0 (Compare match A)	64	H'0080	H'000100	ICRB3	
	CMIB0 (Compare match B)	65	H'0082	H'000104		
	OVI0 (Overflow)	66	H'0084	H'000108		
	Reserved for system use	67	H'0086	H'00010C		
TMR_1	CMIA1 (Compare match A)	68	H'0088	H'000110	ICRB2	
	CMIB1 (Compare match B)	69	H'008A	H'000114		
	OVI1 (Overflow)	70	H'008C	H'000118		
	Reserved for system use	71	H'008E	H'00011C		
TMR_X TMR_Y	CMIAY (Compare match A)	72	H'0090	H'000120	ICRB1	
	CMIBY (Compare match B)	73	H'0092	H'000124		
	OVIY (Overflow)	74	H'0094	H'000128		
	ICIX (Input capture)	75	H'0096	H'00012C		
	CMIAX (Compare match A)	76	H'0098	H'000130		
	CMIBX (Compare match B)	77	H'009A	H'000134		
	OVIX (Overflow)	78	H'009C	H'000138		
—	Reserved for system use	79	H'009E	H'00013C	—	
SCI_0	ERI0 (Reception error 0)	80	H'00A0	H'000140	ICRC7	
	RXI0 (Reception completion 0)	81	H'00A2	H'000144		
	TXI0 (Transmission data empty 0)	82	H'00A4	H'000148		
	TEI0 (Transmission end 0)	83	H'00A6	H'00014C		
SCI_1	ERI1 (Reception error 1)	84	H'00A8	H'000150	ICRC6	
	ERI1 (Reception completion 1)	85	H'00AA	H'000154		
	TXI1 (Transmission data empty 1)	86	H'00AC	H'000158		
	TEI1 (Transmission end 1)	87	H'00AE	H'00015C		
—	Reserved for system use	88 ⋮ 91	H'00B0 ⋮ H'00B6	H'000160 ⋮ H'00016C	—	
IIC_0	IICI0 (1-byte transmission/reception completion)	92	H'00B8	H'000170	ICRC4	
	Reserved for system use	93	H'00BA	H'000174		
IIC_1	IICI1 (1-byte transmission/reception completion)	94	H'00BC	H'000178	ICRC3	
	Reserved for system use	95	H'00BE	H'00017C		
—	Reserved for system use	96 ⋮ 127	H'00C0 ⋮ H'00FE	H'000180 ⋮ H'0001FC	—	Low

5.6 Interrupt Control Modes and Interrupt Operation

The interrupt controller has two modes: interrupt control mode 0 and interrupt control mode 1. Interrupt operations differ depending on the interrupt control mode. NMI and address break interrupts are always accepted except for in the reset state or in hardware standby mode. The interrupt control mode is selected by SYSCR. Table 5.4 shows the interrupt control modes.

Table 5.4 Interrupt Control Modes

Interrupt Control Mode	SYSCR		Priority Setting Registers	Interrupt Mask Bits	Description
	INTM1	INTM0			
0	0	0	ICR	I	Interrupt mask control is performed by the I bit. Priority levels can be set with ICR.
1	0	1	ICR	I, UI	3-level interrupt mask control is performed by the I and UI bits. Priority levels can be set with ICR.

Figure 5.3 shows a block diagram of the priority determination circuit.

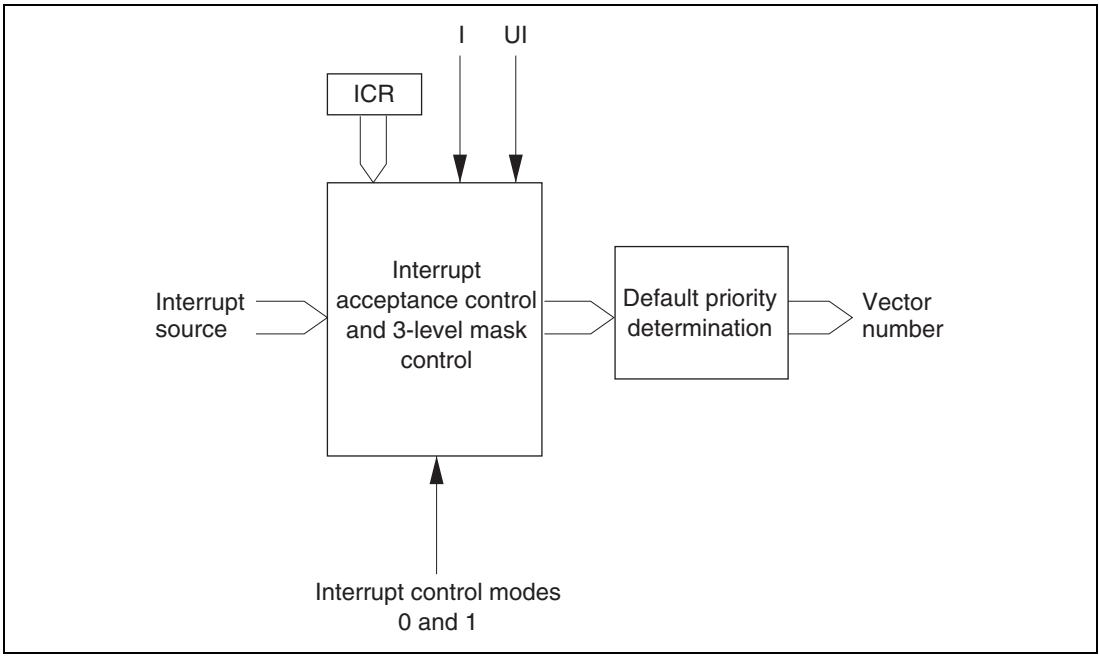


Figure 5.3 Block Diagram of Interrupt Control Operation

(1) Interrupt Acceptance Control and 3-Level Control

In interrupt control modes 0 and 1, interrupt acceptance control and 3-level mask control is performed by means of the I and UI bits in CCR and ICR (control level).

Table 5.5 shows the interrupts selected in each interrupt control mode.

Table 5.5 Interrupts Selected in Each Interrupt Control Mode

	Interrupt Mask Bits		
Interrupt Control Mode	**I**	**UI**	**Selected Interrupts**
0	0	*	All interrupts (interrupt control level 1 has priority)
	1	*	NMI and address break interrupts
1	0	*	All interrupts (interrupt control level 1 has priority)
	1	0	NMI, address break, and interrupt control level 1 interrupts
		1	NMI and address break interrupts

[Legend]

*: Don't care

(2) Default Priority Determination

The priority is determined for the selected interrupt, and a vector number is generated.

If the same value is set for ICR, acceptance of multiple interrupts is enabled, and so only the interrupt source with the highest priority according to the preset default priorities is selected and has a vector number generated.

Interrupt sources with a lower priority than the accepted interrupt source are held pending.

Table 5.6 shows operations and control signal functions in each interrupt control mode.

Table 5.6 Operations and Control Signal Functions in Each Interrupt Control Mode

Interrupt Control Mode	Setting			Interrupt Acceptance Control 3-Level Control			Default Priority Determination
	INTM1	INTM0		I	UI	ICR	
0	0	0	O	IM	—	PR	O
1		1	O	IM	IM	PR	O

[Legend]

O: Interrupt operation control is performed

IM: Used as an interrupt mask bit

PR: Priority is set

—: Not used

5.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests other than NMI and address break are masked by ICR and the I bit of CCR in the CPU. Figure 5.4 shows a flowchart of the interrupt acceptance operation.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. According to the interrupt control level specified in ICR, the interrupt controller only accepts an interrupt request with interrupt control level 1 (priority), and holds pending an interrupt request with interrupt control level 0 (no priority). If several interrupt requests are issued, an interrupt request with the highest priority is accepted according to the priority order, an interrupt handling is requested to the CPU, and other interrupt requests are held pending.
3. If the I bit in CCR is set to 1, the interrupt controller holds pending interrupt requests other than NMI and address break. If the I bit is cleared to 0, any interrupt request is accepted.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except for NMI and address break interrupts.
7. The CPU generates a vector address for the accepted interrupt request and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

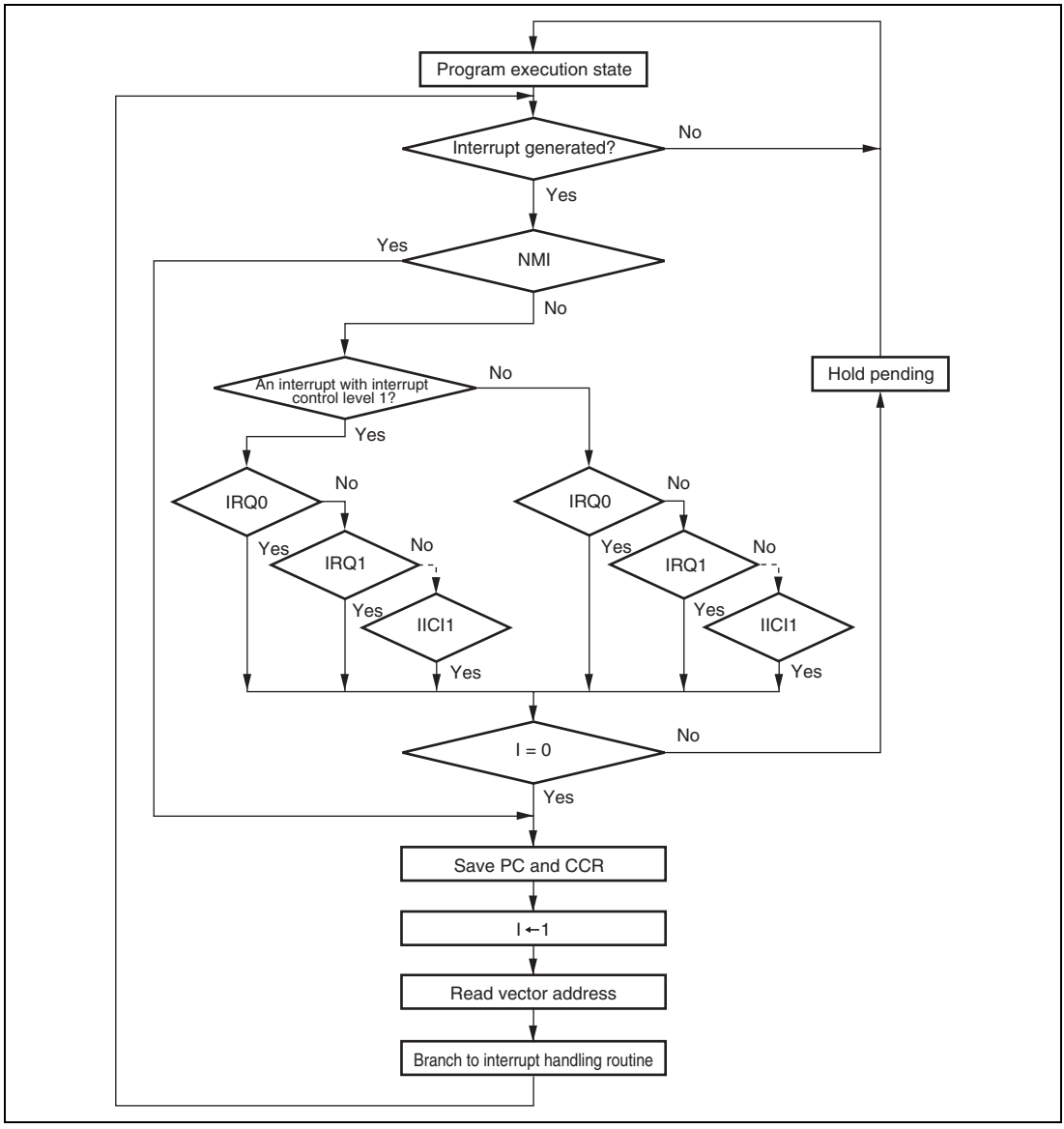


Figure 5.4 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 0

5.6.2 Interrupt Control Mode 1

In interrupt control mode 1, mask control is applied to three levels for interrupt requests other than NMI and address break by comparing the I and UI bits in CCR in the CPU, and the ICR setting.

- An interrupt request with interrupt control level 0 is accepted when the I bit in CCR is cleared to 0. When the I bit is set to 1, the interrupt request is held pending.
- An interrupt request with interrupt control level 1 is accepted when the I bit or UI bit in CCR is cleared to 0. When both the I and UI bits are set to 1, the interrupt request is held pending.

For instance, the state transition when the interrupt enable bit corresponding to each interrupt is set to 1, and ICRA to ICRD are set to H'20, H'00, H'00, and H'00, respectively (IRQ2 and IRQ3 interrupts are set to interrupt control level 1, and other interrupts are set to interrupt control level 0) is shown below. Figure 5.5 shows a state transition diagram.

- All interrupt requests are accepted when I = 0. (Priority order: NMI > IRQ2 > IRQ3 > address break > IRQ0 > IRQ1 …)
- Only NMI, IRQ2, IRQ3, and address break interrupt requests are accepted when I = 1 and UI = 0.
- Only NMI and address break interrupt requests are accepted when I = 1 and UI = 1.

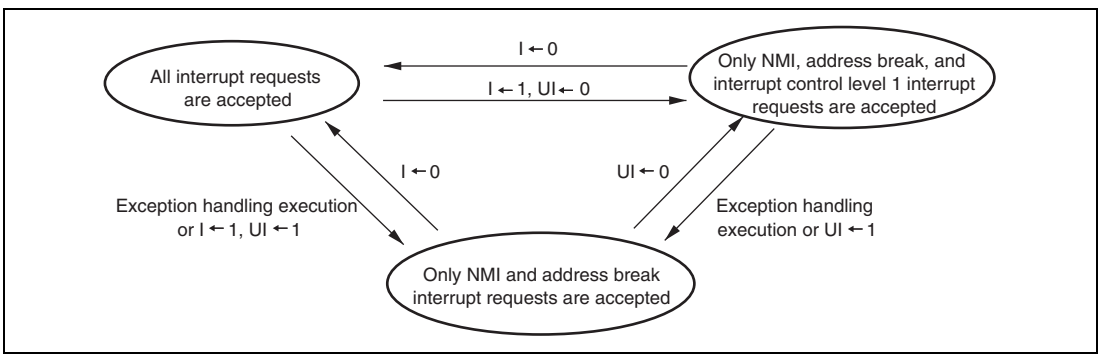


Figure 5.5 State Transition in Interrupt Control Mode 1

Figure 5.6 shows a flowchart of the interrupt acceptance operation.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. According to the interrupt control level specified in ICR, the interrupt controller only accepts an interrupt request with interrupt control level 1 (priority), and holds pending an interrupt request with interrupt control level 0 (no priority). If several interrupt requests are issued, an interrupt request with the highest priority is accepted according to the priority order, an interrupt handling is requested to the CPU, and other interrupt requests are held pending.
3. An interrupt request with interrupt control level 1 is accepted when the I bit is cleared to 0, or when the I bit is set to 1 while the UI bit is cleared to 0.

   An interrupt request with interrupt control level 0 is accepted when the I bit is cleared to 0.

   When both the I and UI bits are set to 1, only NMI and address break interrupt requests are accepted, and other interrupts are held pending.

   When the I bit is cleared to 0, the UI bit does not affect acceptance of interrupt requests.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. The I and UI bits in CCR are set to 1. This masks all interrupts except for NMI and address break interrupts.
7. The CPU generates a vector address for the accepted interrupt request and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

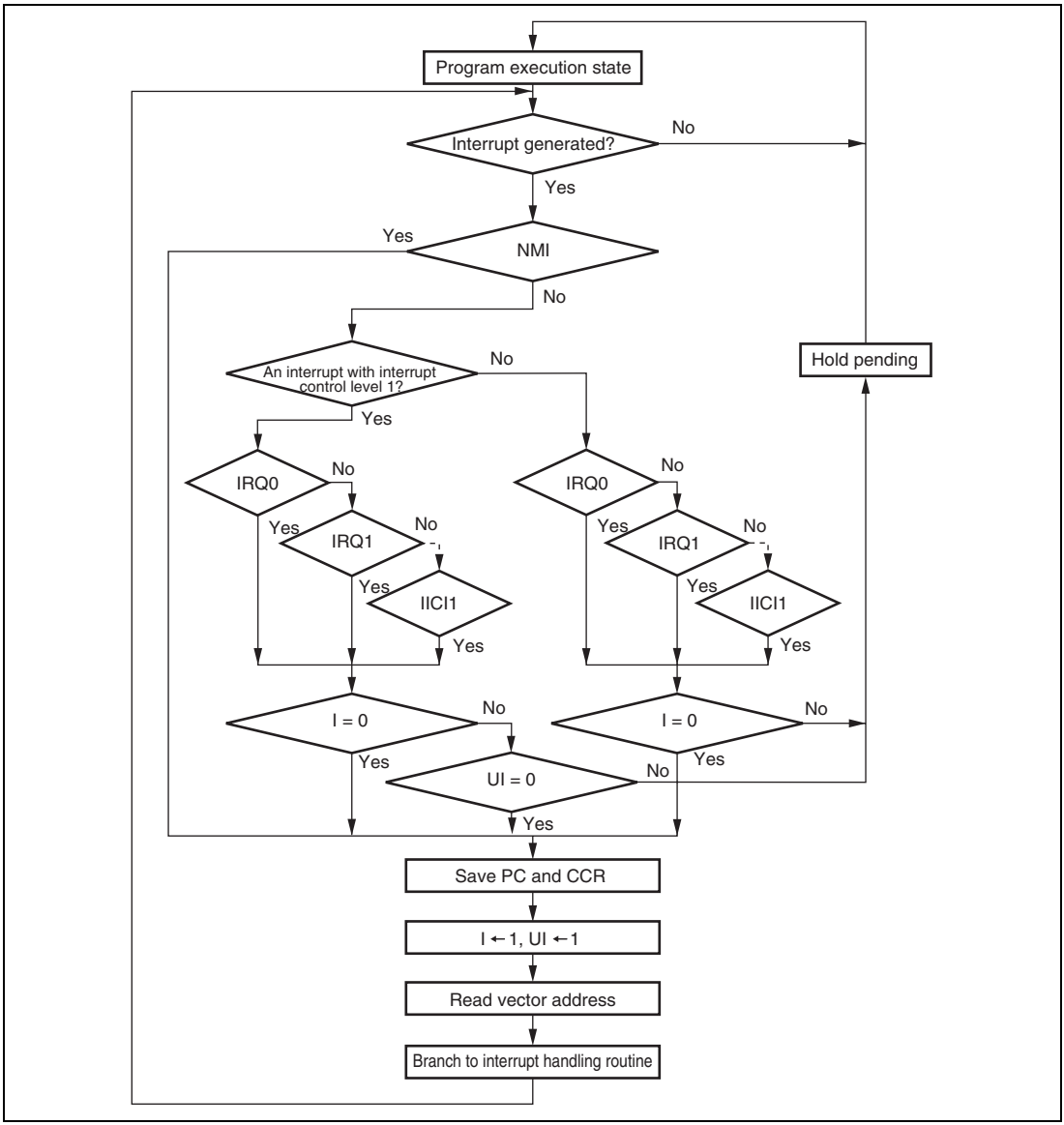


Figure 5.6 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 1

5.6.3 Interrupt Exception Handling Sequence

Figure 5.7 shows the interrupt exception handling sequence. The example shown is for the case where interrupt control mode 0 is set in advanced mode, and the program area and stack area are in on-chip memory.

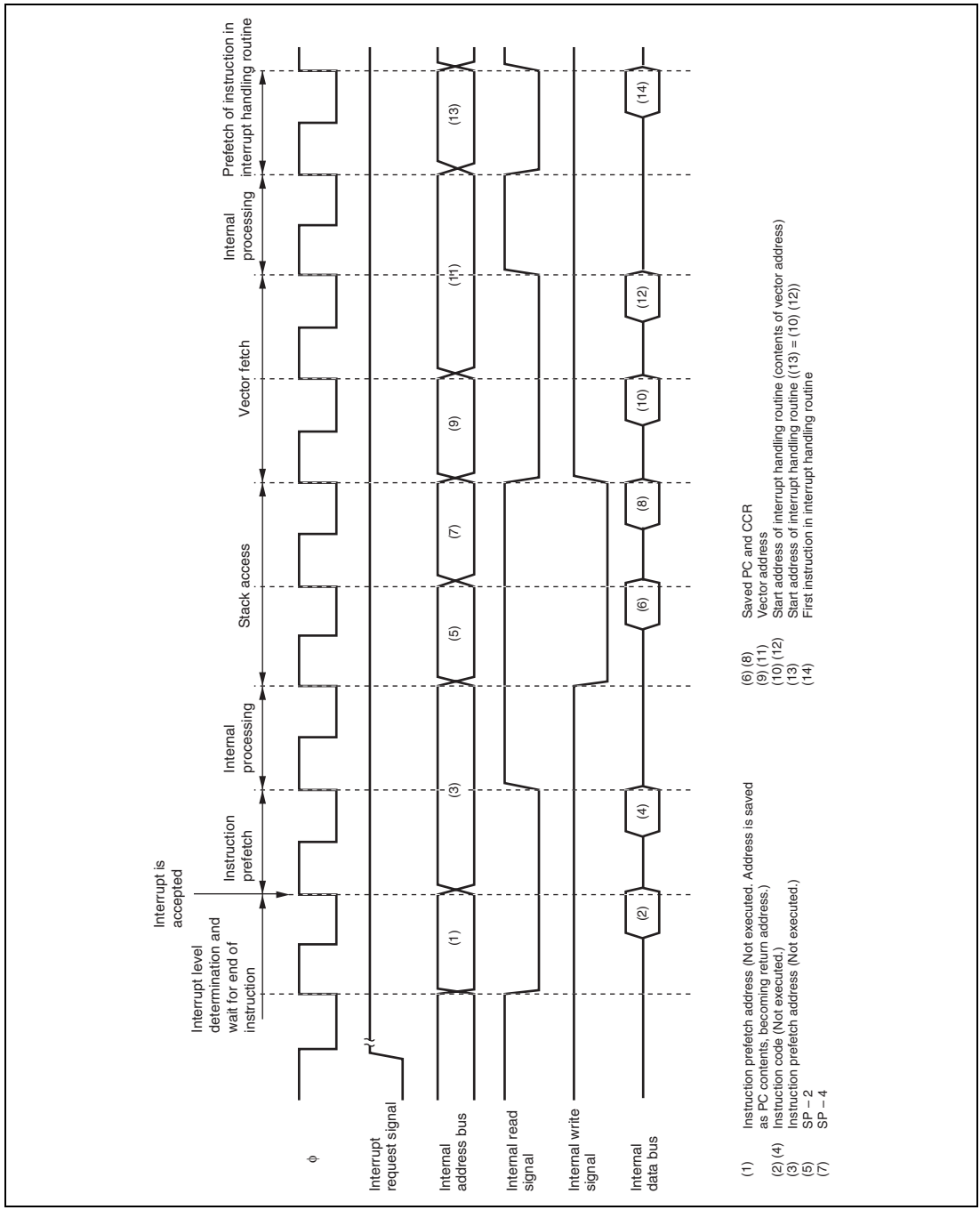


Figure 5.7 Interrupt Exception Handling

5.6.4 Interrupt Response Times

Table 5.7 shows interrupt response times – the intervals between generation of an interrupt request and execution of the first instruction in the interrupt handling routine.

Table 5.7 Interrupt Response Times

No.	Execution Status	Normal Mode	Advanced Mode
1	Interrupt priority determination*[1]	3	3
2	Number of wait states until instruction execution ends*[2]	1 to (19 + 2·S_I)	1 to (19 + 2·S_I)
3	Saving of PC and CCR in stack	2·S_K	2·S_K
4	Vector fetch	S_I	2·S_I
5	Instruction fetch*[3]	2·S_I	2·S_I
6	Internal processing*[4]	2	2
	Total (using on-chip memory)	11 to 31	12 to 32

Notes: 1. Two states in case of internal interrupt.
2. Refers to MULXS and DIVXS instructions.
3. Prefetch after interrupt acceptance and prefetch of interrupt handling routine.
4. Internal processing after interrupt acceptance and internal processing after vector fetch.

Table 5.8 Number of Execution States in Interrupt Handling Routine

	Object of Access		
		External Device	
		8-Bit Bus	
Symbol	Internal Memory	2-State Access	3-State Access
Instruction fetch S_I	1	4	6 + 2m
Brach address read S_J			
Stack manipulation S_K			

[Legend]
m: Number of wait states in external device access

5.6.5 DTC Activation by Interrupt

The DTC can be activated by an interrupt. In this case, the following options are available:

1. Interrupt request to CPU
2. Activation request to DTC
3. Both of the above

For details on interrupt requests that can be used to activate the DTC, see section 7, Data Transfer Controller (DTC). Figure 5.8 shows a block diagram of the DTC and interrupt controller.

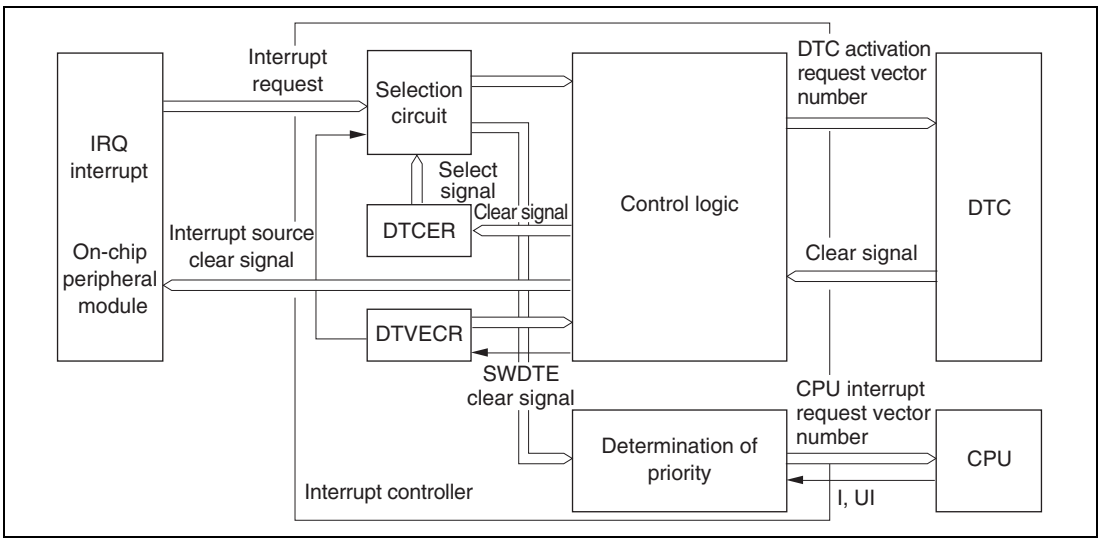


Figure 5.8 Interrupt Control for DTC

The interrupt controller has three main functions in DTC control.

(1) Selection of Interrupt Source

It is possible to select a DTC activation request or CPU interrupt request for an interrupt source with the DTCE bit in DTCERA to DTCERE of the DTC. After a DTC data transfer, the DTCE bit can be cleared to 0 and an interrupt request sent to the CPU in accordance with the specification of the DISEL bit in MRB of the DTC. When the DTC performs the specified number of data transfers and the transfer counter reaches 0, following the DTC data transfer the DTCE bit is cleared to 0 and an interrupt request is sent to the CPU.

(2) Determination of Priority

The DTC activation source is selected in accordance with the default priority order, and is not affected by mask or priority levels. See section 7.4, Location of Register Information and DTC Vector Table, for the respective priorities.

(3) Operation Order

If the same interrupt is selected as a DTC activation source and a CPU interrupt source, the DTC data transfer is performed first, followed by CPU interrupt exception handling.

Table 5.9 summarizes interrupt source selection and interrupt source clearance control according to the settings of the DTCE bit in DTCERA to DTCERE of the DTC and the DISEL bit in MRB of the DTC.

Table 5.9 Interrupt Source Selection and Clearing Control

Settings			
DTC		Interrupt Source Selection/Clearing Control	
DTCE	DISEL	DTC	CPU
0	*	×	Δ
1	0	Δ	×
	1	○	Δ

[Legend]

Δ: The relevant interrupt is used. Interrupt source clearing is performed. (The CPU should clear the source flag in the interrupt handling routine.)

○: The relevant interrupt is used. The interrupt source is not cleared.

×: The relevant interrupt cannot be used.

*: Don't care

5.7 Address Breaks

5.7.1 Features

With this LSI, it is possible to identify the prefetch of a specific address by the CPU and generate an address break interrupt, using the ABRKCR and BAR registers. When an address break interrupt is generated, address break interrupt exception handling is executed.

This function can be used to detect the beginning of execution of a bug location in the program, and branch to a correction routine.

5.7.2 Block Diagram

Figure 5.9 shows a block diagram of the address break function.

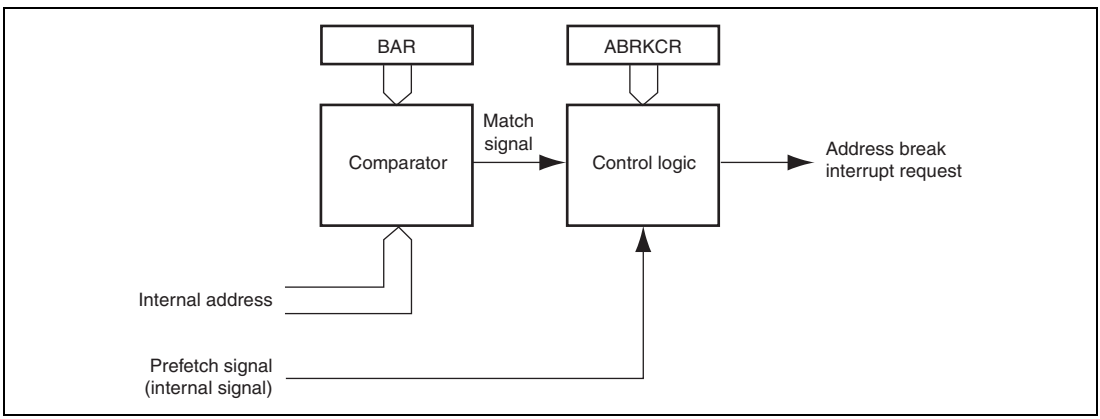


Figure 5.9 Block Diagram of Address Break Function

5.7.3 Operation

ABRKCR and BAR settings can be made so that an address break interrupt is generated when the CPU prefetches the address set in BAR. This address break function issues an interrupt request to the interrupt controller when the address is prefetched, and the interrupt controller determines the interrupt priority. When the interrupt is accepted, interrupt exception handling is started on completion of the currently executing instruction. With an address break interrupt, interrupt mask control by the I and UI bits in the CPU's CCR is ineffective.

The register settings when the address break function is used are as follows.

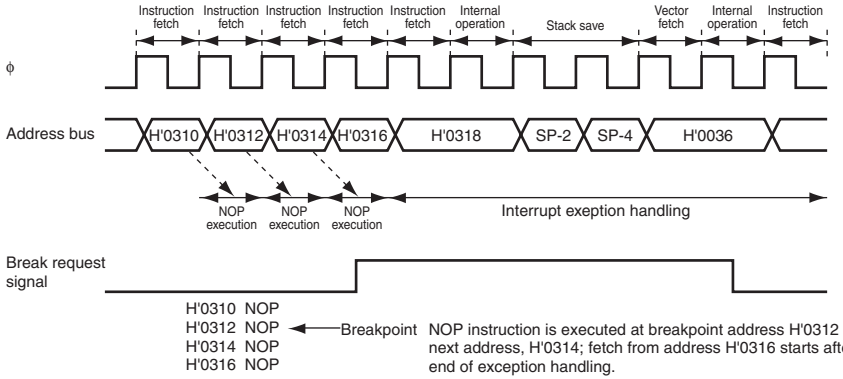
1. Set the break address in bits A23 to A1 in BAR.
2. Set the BIE bit in ABRKCR to 1 to enable address breaks. An address break will not be requested if the BIE bit is cleared to 0.

When the setting condition occurs, the CMF flag in ABRKCR is set to 1 and an interrupt is requested. If necessary, the source should be identified in the interrupt handling routine.

5.7.4 Usage Notes

1. With the address break function, the address at which the first instruction byte is located should be specified as the break address. Occurrence of the address break condition may not be recognized for other addresses.
2. In normal mode, no comparison is made with address lines A23 to A16.
3. If a branch instruction (Bcc, BSR) jump instruction (JMP, JSR), RTS instruction, or RTE instruction is located immediately before the address set in BAR, execution of this instruction will output a prefetch signal for that address, and an address break may be requested. This can be prevented by not making a break address setting for an address immediately following one of these instructions, or by determining within the interrupt handling routine whether interrupt handling was initiated by a genuine condition occurrence.
4. As an address break interrupt is generated by a combination of the internal prefetch signal and address, the timing of the start of interrupt exception handling depends on the content and execution cycle of the instruction at the set address and the preceding instruction. Figure 5.10 shows some address timing examples.

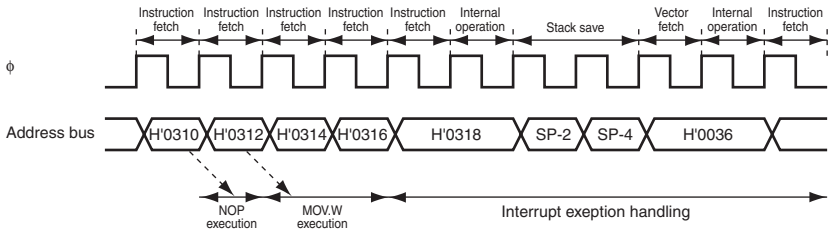
• Program area in on-chip memory, 1-state execution instruction at specified break address



H'0310 NOP
H'0312 NOP ← Breakpoint
H'0314 NOP
H'0316 NOP

NOP instruction is executed at breakpoint address H'0312 and next address, H'0314; fetch from address H'0316 starts after end of exception handling.

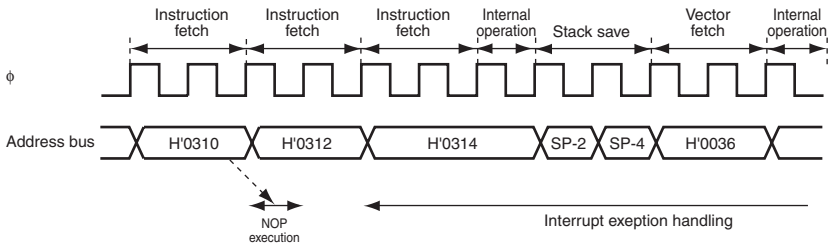
• Program area in on-chip memory, 2-state execution instruction at specified break address



H'0310 NOP
H'0312 MOV.W #xx : 16,Rd ← Breakpoint
H'0314 NOP
H'0316 NOP

MOV instruction is executed at breakpoint address H'0312, NOP instruction at next address, H'0316, is not executed; fetch from address H'0316 starts after end of exception handling.

• Program area in external memory (2-state access, 16-bit-bus access), 1-state execution instruction at specified break address (Not available in this LSI)



H'0310 NOP
H'0312 NOP ← Breakpoint
H'0314 NOP
H'0316 NOP

NOP instruction at breakpoint address H'0312 is not executed; fetch from address H'0312 starts after end of exception handling.

Figure 5.10 Examples of Address Break Timing

5.8 Usage Notes

5.8.1 Conflict between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to disable interrupt requests, 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, so interrupt exception handling for that interrupt will be executed on completion of the instruction. However, if there is an interrupt request of higher priority than that interrupt, interrupt exception handling will be executed for the higher-priority interrupt, and the lower-priority interrupt will be ignored. The same rule is also applied when an interrupt source flag is cleared to 0. Figure 5.11 shows an example where the CMIEA bit in TCR of the TMR is cleared to 0. The above conflict will not occur if an interrupt enable bit or interrupt source flag is cleared to 0 while the interrupt is disabled.

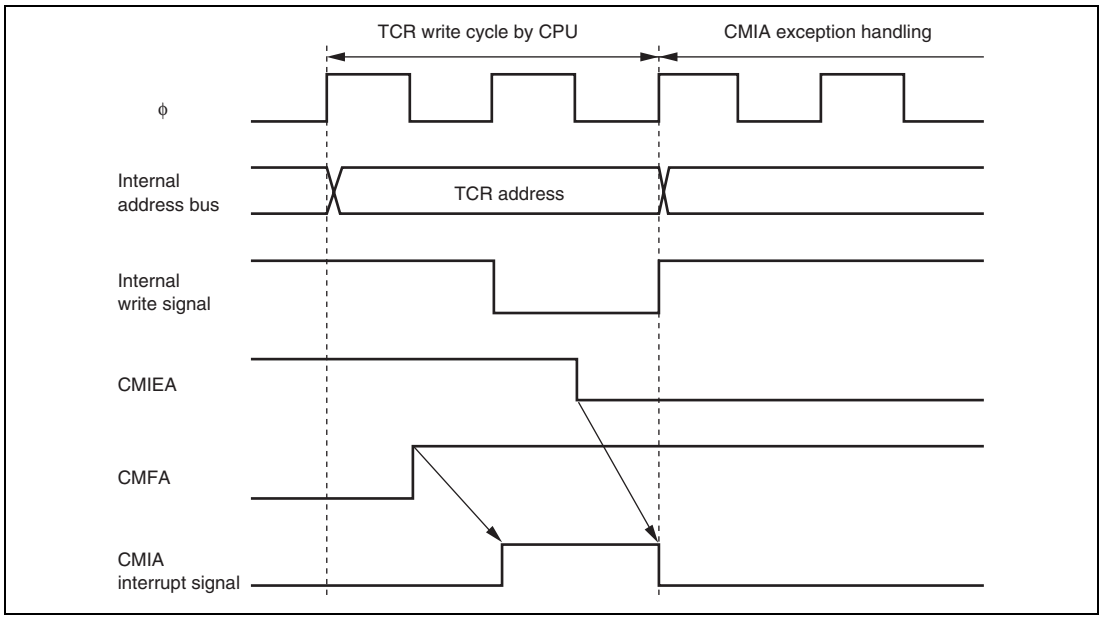


Figure 5.11 Conflict between Interrupt Generation and Disabling

5.8.2 Instructions for Disabling Interrupts

The instructions that disable interrupts are LDC, ANDC, ORC, and XORC. After any of these instructions are executed, all interrupts including NMI are disabled and the next instruction is always executed. When the I bit or UI bit is set by one of these instructions, the new value becomes valid two states after execution of the instruction ends.

5.8.3 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 data transfer is not accepted until data transfer is completed.

With the EEPMOV.W instruction, if an interrupt request is issued during data transfer, interrupt exception handling starts at a break in the transfer cycles. The PC value saved on the stack in this case is the address of the next instruction. Therefore, if an 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
```

5.8.4 External Interrupt Pin in Software Standby Mode and Watch Mode

- When the pins ($\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$) are used as external input pins in software standby mode or watch mode, the pins should not be left floating.
- When the external interrupt pins ($\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$) are used in software standby and watch modes, the noise canceller should be disabled.

5.8.5 Noise Canceller Switching

The noise canceller should be switched when the external input pins ($\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$) are high.

5.8.6 IRQ Status Register (ISR)

Since IRQnF may be set to 1 according to the pin state after reset, the ISR should be read after reset, and then write 0 in IRQnF (n = 7 to 0).

Section 6 Bus Controller (BSC)

This LSI has an on-chip bus controller (BSC) that manages the bus width and the number of access states of the external address space. The BSC also has a bus arbitration function, and controls the operation of the internal bus masters – CPU, and data transfer controller (DTC).

6.1 Features

- Basic bus interface
  2-state access or 3-state access can be selected for each area
  Program wait states can be inserted for each area
- Burst ROM interface
  A burst ROM interface can be set for basic expansion areas
  1-state access or 2-state access can be selected for burst access
- Idle cycle insertion
  An idle cycle can be inserted for external write cycles immediately after external read cycles
- Bus arbitration function
  Includes a bus arbiter that arbitrates bus mastership between the CPU and DTC

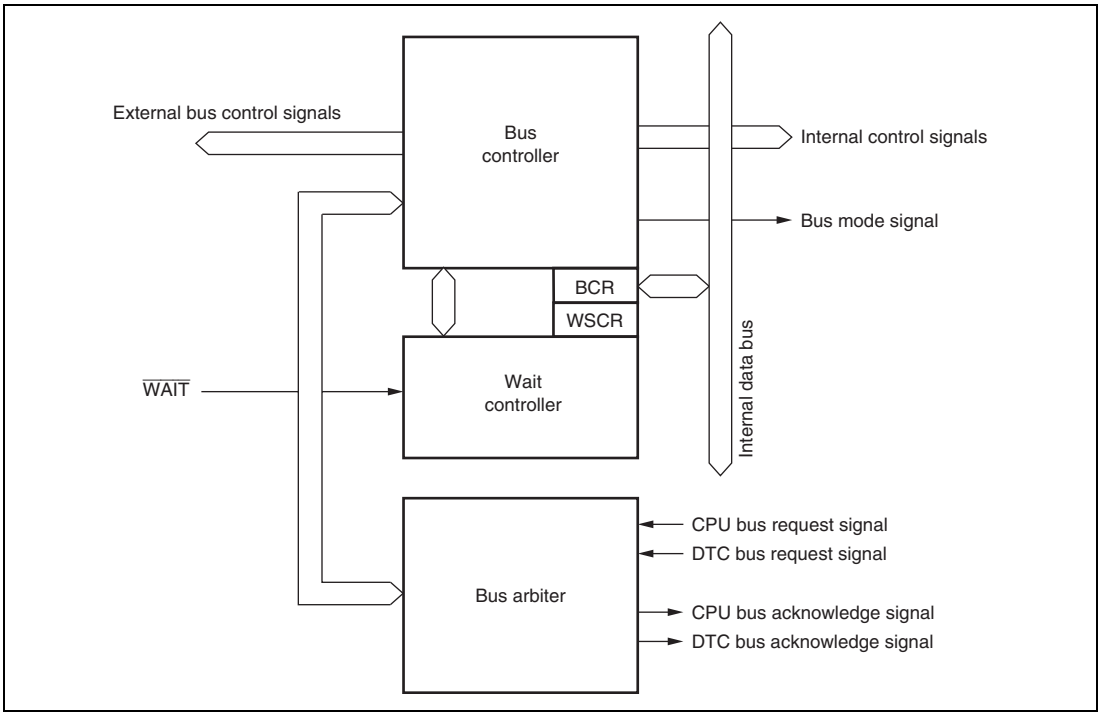


Figure 6.1 Block Diagram of Bus Controller

6.2 Input/Output Pins

Table 6.1 summarizes the pins of the bus controller.

Table 6.1 Pin Configuration

Symbol	I/O	Function
$\overline{AS}$	Output	Strobe signal indicating that address output on the address bus is enabled (when the IOSE bit in SYSCR is cleared to 0).
$\overline{IOS}$	Output	I/O select signal (when the IOSE bit in SYSCR is set to 1).
$\overline{RD}$	Output	Strobe signal indicating that the external address space is being read.
$\overline{WR}$	Output	Strobe signal indicating that the external address space is being written and the data bus is enabled.
$\overline{WAIT}$	Input	Wait request signal when accessing the external 3-state access space.

6.3 Register Descriptions

The bus controller has the following registers. For details on the system control register, refer to section 3.2.2, System Control Register (SYSCR).

- Bus control register (BCR)
- Wait state control register (WSCR)

6.3.1 Bus Control Register (BCR)

BCR is used to specify the access mode for the external address space or the I/O area range when the $\overline{AS}/\overline{IOS}$ pin is specified as an I/O strobe pin.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	R/W	Reserved The initial value should not be changed.
6	ICIS0	1	R/W	Idle Cycle Insertion Selects whether or not to insert 1-state of the idle cycle between bus cycles when the external write cycle follows the external read cycle. 0: Idle cycle not inserted when the external write cycle follows the external read cycle 1: 1-state idle cycle inserted when the external write cycle follows the external read cycle
5	BRSTRM	0	R/W	Burst ROM Enable Selects the bus interface for the external address space. 0: Basic bus interface 1: Burst ROM interface
4	BRSTS1	1	R/W	Burst Cycle Select 1 Selects the number of states in the burst cycle of the burst ROM interface. 0: 1 state 1: 2 states

Bit	Bit Name	Initial Value	R/W	Description
3	BRSTS0	0	R/W	Burst Cycle Select 0 Selects the number of words that can be accessed by burst access via the burst ROM interface. 0: Max, 4 words 1: Max, 8 words
2	—	0	R/W	Reserved The initial value should not be changed.
1	IOS1	1	R/W	IOS Select 1, 0
0	IOS0	1	R/W	Select the address range where the $\overline{\text{IOS}}$ signal is output. For details, refer to table 6.3.

6.3.2 Wait State Control Register (WSCR)

WSCR is used to specify the data bus width for external address space access, the number of access states, the wait mode, and the number of wait states for access to external address spaces. The bus width and the number of access states for internal memory and internal I/O registers are fixed regardless of the WSCR settings.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	R/W	Reserved
6	—	1	R/W	These bits should not be written by 1.
5	ABW	1	R/W	Bus Width Control The initial value should not be changed.
4	AST	1	R/W	Access State Control Selects 2 or 3 access states for access to the external address space. This bit also enables or disables wait-state insertion. 0: 2-state access space. Wait state insertion disabled in external address space access 1: 3-state access space. Wait state insertion enabled in external address space access

Bit	Bit Name	Initial Value	R/W	Description
3	WMS1	0	R/W	Wait Mode Select 1, 0
2	WMS0	0	R/W	Select the wait mode for access to the external address space when the AST bit is set to 1. 00: Program wait mode 01: Wait disabled mode 10: Pin wait mode 11: Pin auto-wait mode
1	WC1	1	R/W	Wait Count 1, 0
0	WC0	1	R/W	Select the number of program wait states to be inserted when the external address space is accessed while the AST bit is set to 1. 00: Program wait state is not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted

6.4 Bus Control

6.4.1 Bus Specifications

The external address space bus specifications consist of three elements: Bus width, the number of access states, and the wait mode and the number of program wait states. The bus width and the number of access states for on-chip memory and internal I/O registers are fixed, and are not affected by the bus controller settings.

(1) Bus Width

A bus width of 8 or 16 bits can be selected via the ABW bit in WSCR. However, 16-bit access space cannot be selected in this LSI.

(2) Number of Access States

Two or three access states can be selected via the AST bit in WSCR. When the 2-state access space is designated, wait-state insertion is disabled.

In the burst ROM interface, the number of access states is determined regardless of the AST bit setting.

(3) Wait Mode and Number of Program Wait States

When a 3-state access space is designated by the AST bit in WSCR, the wait mode and the number of program wait states to be inserted automatically is selected by the WMS1, WMS0, WC1, and WC0 bits in WSCR. From 0 to 3 program wait states can be selected.

Table 6.2 shows the bus specifications for the basic bus interface of each area.

Table 6.2 Bus Specifications for Basic Bus Interface

						Bus Specifications		
ABW	**AST**	**WMS1**	**WMS0**	**WC1**	**WC0**	**Bus Width**	**Number of Access States**	**Number of Program Wait States**
0	—	—	—	—	—	Setting prohibited in this LSI		
1	0	—	—	—	—	8	2	0
	1	0	1	—	—	8	3	0
		—*	—*	0	0		3	0
					1			1
				1	0			2
					1			3

Note: * Other than WMS1 = 0 and WMS0 = 1

6.4.2 Advanced Mode

The external address space is initialized as the basic bus interface and a 3-state access space. In on-chip ROM enable extended mode, the address space other than on-chip ROM, on-chip RAM, internal I/O registers, and their reserved areas is specified as the external address space. The on-chip RAM and its reserved area are enabled when the RAME bit in SYSCR is set to 1. The on-chip RAM and its reserved area are disabled and corresponding addresses are the external address space when the RAME bit is cleared to 0.

6.4.3 Normal Mode

The external address space is initialized as the basic bus interface and a 3-state access space. In on-chip ROM disable extended mode, the address space other than on-chip RAM and internal I/O registers is specified as the external address space. In on-chip ROM enable extended mode, the address space other than on-chip ROM, on-chip RAM, internal I/O registers, and their reserved areas is specified as the external address space. The on-chip RAM area is enabled when the RAME bit in SYSCR is set to 1, and disabled and specified as the external address space when the RAME bit is cleared to 0.

6.4.4 I/O Select Signals

The LSI can output I/O select signals ($\overline{IOS}$); the signal is driven low when the corresponding external address space is accessed. Figure 6.2 shows an example of $\overline{IOS}$ signal output timing.

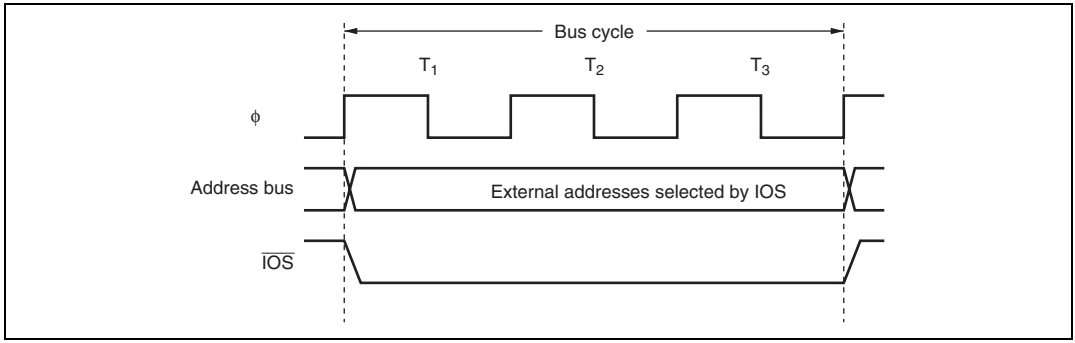


Figure 6.2 $\overline{IOS}$ Signal Output Timing

Enabling or disabling $\overline{IOS}$ signal output is performed by the IOSE bit in SYSCR. In extended mode, the $\overline{IOS}$ pin functions as an $\overline{AS}$ pin by a reset. To use this pin as an $\overline{IOS}$ pin, set the IOSE bit to 1. For details, refer to section 8, I/O Ports.

The address ranges of the $\overline{IOS}$ signal output can be specified by the IOS1 and IOS0 bits in BCR, as shown in table 6.3.

Table 6.3 Address Range for $\overline{IOS}$ Signal Output

IOS1	IOS0	$\overline{IOS}$ Signal Output Range	
0	0	H'(FF)F000 to H'(FF)F03F	
	1	H'(FF)F000 to H'(FF)F0FF	
1	0	H'(FF)F000 to H'(FF)F3FF	
	1	H'(FF)F000 to H'(FF)F7FF	(Initial value)

6.5 Basic Bus Interface

The basic bus interface enables direct connection to ROM and SRAM. For details on selection of the bus specifications when using the basic bus interface, see table 6.2.

6.5.1 Data Size and Data Alignment

Data sizes for the CPU and other internal bus masters are byte, word, and longword. The BSC has a data alignment function, and controls whether the upper data bus (D15 to D8) or lower data bus (D7 to D0) is used when the external address space is accessed, according to the bus specifications for the area being accessed (8-bit access space or 16-bit access space) and the data size. This LSI has only the upper data bus and only data alignment for the 8-bit access space is applied. The pins for the upper data bus in this LSI are D7 to D0.

(1) 8-Bit Access Space

Figure 6.3 illustrates data alignment control for the 8-bit access space. With the 8-bit access space, the upper data bus (D15 to D8) is always used for accesses. The amount of data that can be accessed at one time is one byte: a word access is performed as two byte accesses, and a longword access, as four byte accesses.

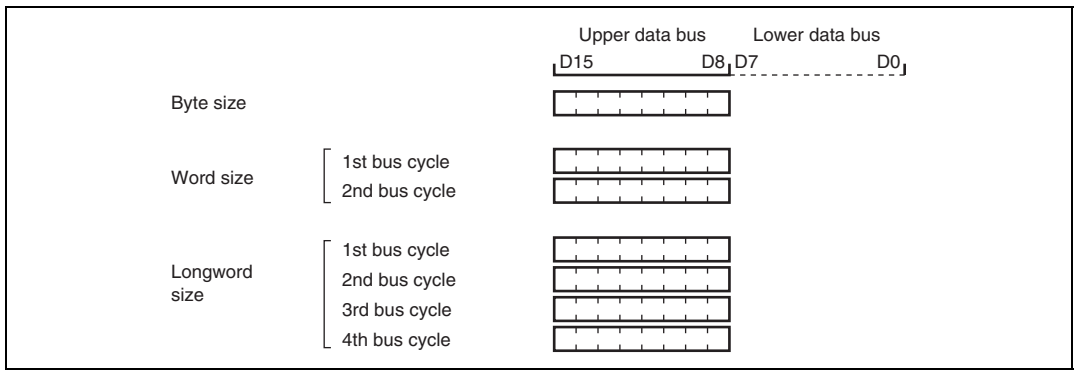


Figure 6.3 Access Sizes and Data Alignment Control (8-Bit Access Space)

(2) 16-Bit Access Space (Not available in this LSI)

Figure 6.4 illustrates data alignment control for the 16-bit access space. With the 16-bit access space, the upper data bus (D15 to D8) and lower data bus (D7 to D0) are used for accesses. The amount of data that can be accessed at one time is one byte or one word, and a longword access is executed as two word accesses.

In byte access, whether the upper or lower data bus is used is determined by whether the address is even or odd. The upper data bus is used for an even address, and the lower data bus for an odd address.

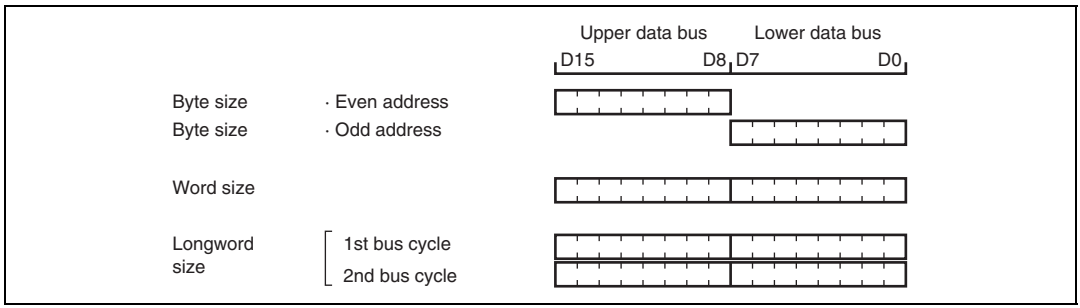


Figure 6.4 Access Sizes and Data Alignment Control (16-bit Access Space)

6.5.2 Valid Strobes

Table 6.4 shows the data buses used and valid strobes for each access space.

In a read, the $\overline{RD}$ signal is valid for both the upper and lower halves of the data bus. In a write, the $\overline{HWR}$ signal is valid for the upper half of the data bus, and the $\overline{LWR}$ signal for the lower half. This LSI has only the upper data bus and only the $\overline{RD}$ and $\overline{HWR}$ signals are enabled. The pin for $\overline{HWR}$ signal in this LSI is $\overline{WR}$.

Table 6.4 Data Buses Used and Valid Strobes

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D15 to D8)*1	Lower Data Bus (D7 to D0)*3
8-bit access space	Byte	Read	—	$\overline{RD}$	Valid	Ports or others
		Write	—	$\overline{HWR}$*2		Ports or others
16-bit access space (not available in this LSI)	Byte	Read	Even	$\overline{RD}$	Valid	Invalid
			Odd		Invalid	Valid
		Write	Even	$\overline{HWR}$	Valid	Undefined
			Odd	$\overline{LWR}$	Undefined	Valid
	Word	Read	—	$\overline{RD}$	Valid	Valid
		Write	—	$\overline{HWR}$, $\overline{LWR}$	Valid	Valid

Notes: Undefined: Undefined data is output.

Invalid: Input state with the input value ignored.

Ports or others: Used as ports or I/O pins for on-chip peripheral modules, and are not used as the data bus.

1. The pins in this LSI are D7 to D0.
2. The pin in this LSI is $\overline{WR}$.
3. No pins are available in this LSI.

6.5.3 Basic Operation Timing

(1) 8-Bit, 2-State Access Space

Figure 6.5 shows the bus timing for an 8-bit, 2-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used. Wait states cannot be inserted. This LSI does not have the lower data bus (D7 to D0) and $\overline{\text{LWR}}$ pin. The pins for the upper data bus (D15 to D8) in this LSI are D7 to D0 and the pin for the $\overline{\text{HWR}}$ signal is $\overline{\text{WR}}$.

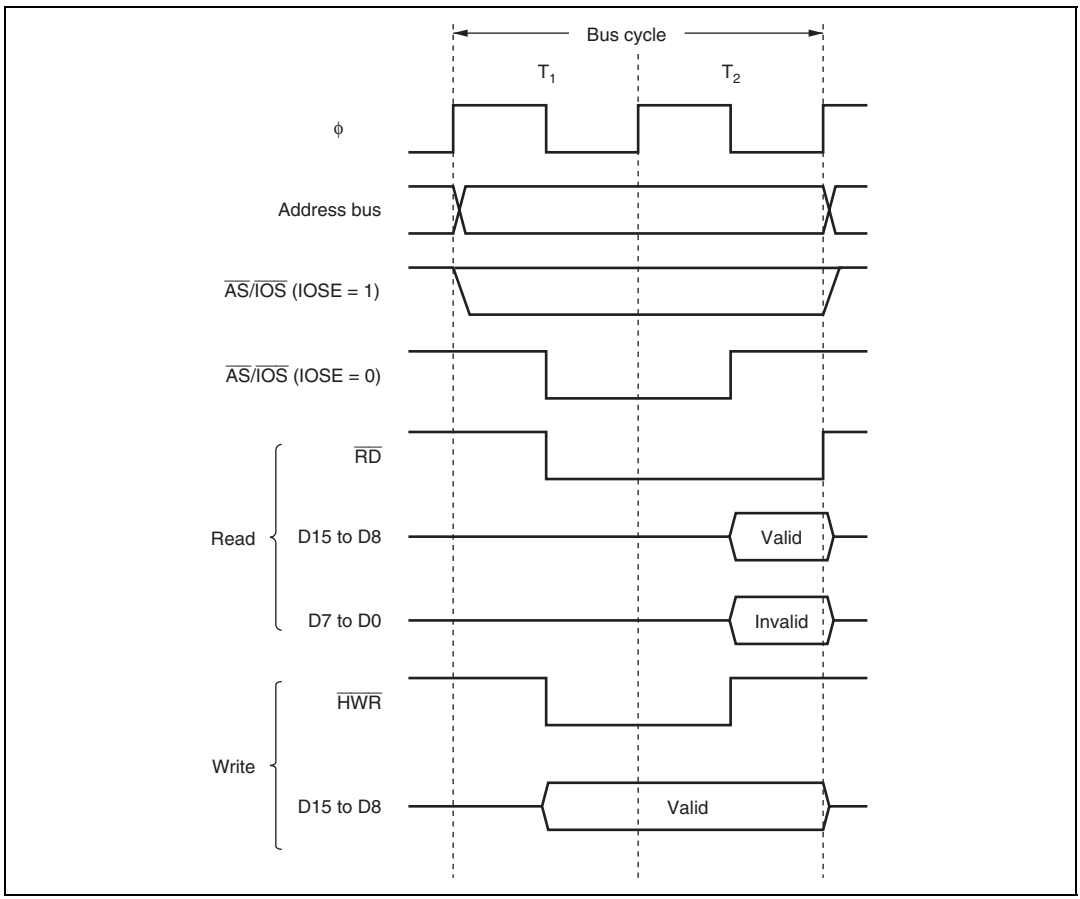


Figure 6.5 Bus Timing for 8-Bit, 2-State Access Space

(2) 8-Bit, 3-State Access Space

Figure 6.6 shows the bus timing for an 8-bit, 3-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used. Wait states can be inserted. This LSI does not have the lower data bus (D7 to D0) and $\overline{\text{LWR}}$ pin. The pins for the upper data bus (D15 to D8) in this LSI are D7 to D0 and the pin for the $\overline{\text{HWR}}$ signal is $\overline{\text{WR}}$.

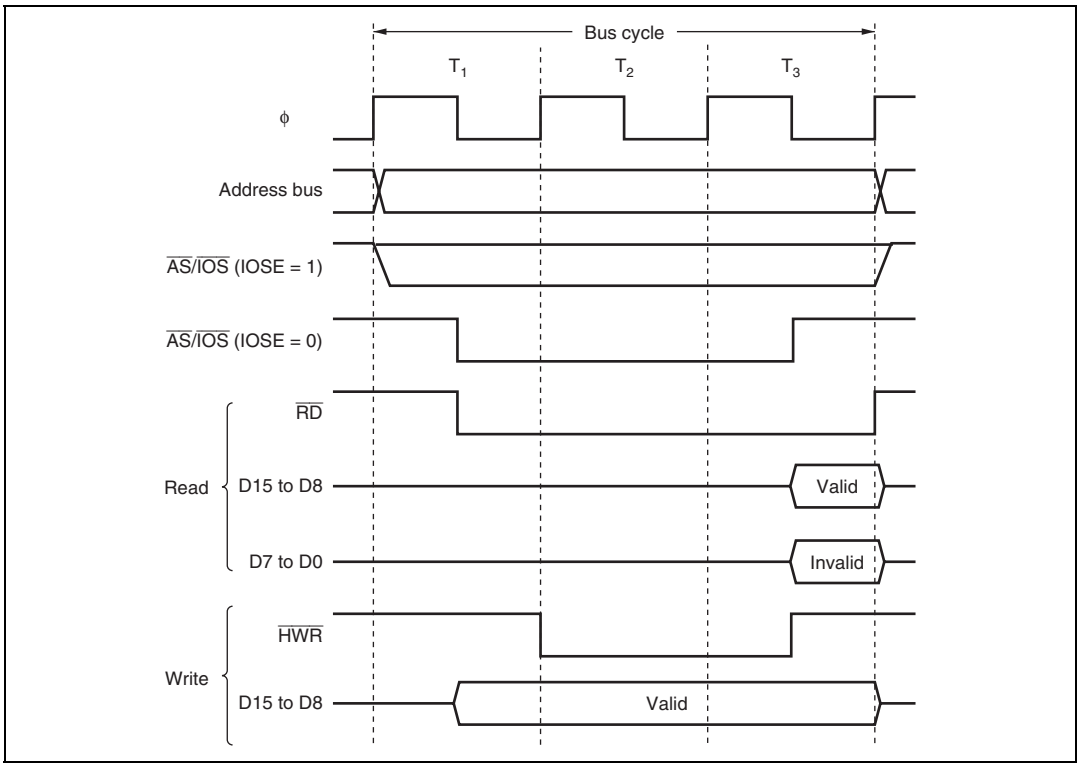


Figure 6.6 Bus Timing for 8-Bit, 3-State Access Space

6.5.4 Wait Control

When accessing the external address space, this LSI can extend the bus cycle by inserting one or more wait states (T_w). There are three ways of inserting wait states: Program wait insertion, pin wait insertion using the $\overline{\text{WAIT}}$ pin, and the combination of program wait and the $\overline{\text{WAIT}}$ pin.

(1) Program Wait Mode

A specified number of wait states T_w can be inserted automatically between the T_2 state and T_3 state when accessing the external address space always according to the settings of the WC1 and WC0 bits in WSCR.

(2) Pin Wait Mode

A specified number of wait states T_w can be inserted automatically between the T_2 state and T_3 state when accessing the external address space always according to the settings of the WC1 and WC0 bits. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_2 or T_w state, another T_w state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_w states are inserted until it goes high.

This is useful when inserting four or more T_w states, or when changing the number of T_w states to be inserted for each external device.

(3) Pin Auto-Wait Mode

A specified number of wait states T_w can be inserted automatically between the T_2 state and T_3 state when accessing the external address space according to the settings of the WC1 and WC0 bits if the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_2 state. Even if the $\overline{\text{WAIT}}$ pin is held low, T_w states can be inserted only up to the specified number of states.

This function enables the low-speed memory interface only by inputting the chip select signal to the $\overline{\text{WAIT}}$ pin.

Figure 6.7 shows an example of wait state insertion timing in pin wait mode.

The settings after a reset are: 3-state access, 3 program wait insertion, and $\overline{\text{WAIT}}$ pin input disabled.

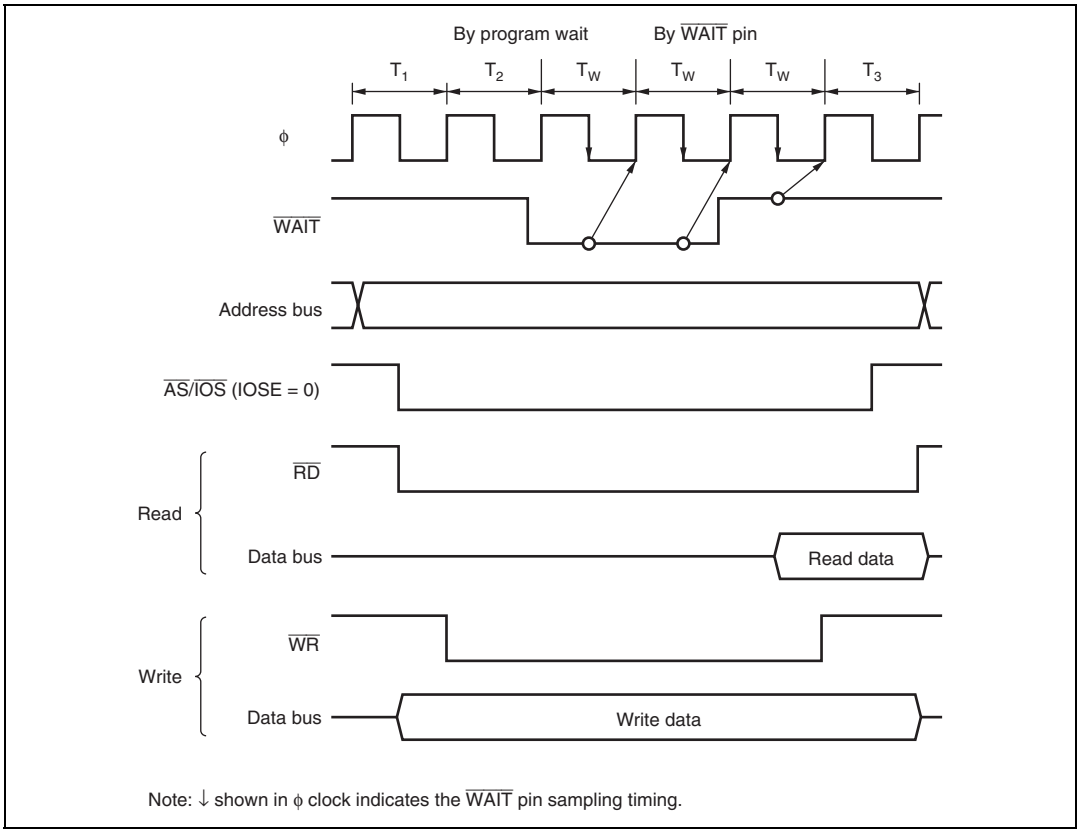


Figure 6.7 Example of Wait State Insertion Timing (Pin Wait Mode)

6.6 Burst ROM Interface

In this LSI, the external address space can be designated as the burst ROM space by setting the BRSTRM bit in BCR to 1, and the burst ROM interface enabled. Consecutive burst accesses of a maximum four or eight words can be performed only during CPU instruction fetch. 1 or 2 states can be selected for burst ROM access.

6.6.1 Basic Operation Timing

The number of access states in the initial cycle (full access) of the burst ROM interface is determined by the AST bit in WSCR. When the AST bit is set to 1, wait states can be inserted. 1 or 2 states can be selected for burst access according to the setting of the BRSTS1 bit in BCR. Wait states cannot be inserted in a burst cycle. Burst accesses of a maximum four words is performed when the BRSTS0 bit in BCR is cleared to 0, and burst accesses of a maximum eight words is performed when the BRSTS0 bit in BCR is set to 1.

The basic access timing for the burst ROM space is shown in figures 6.8 and 6.9.

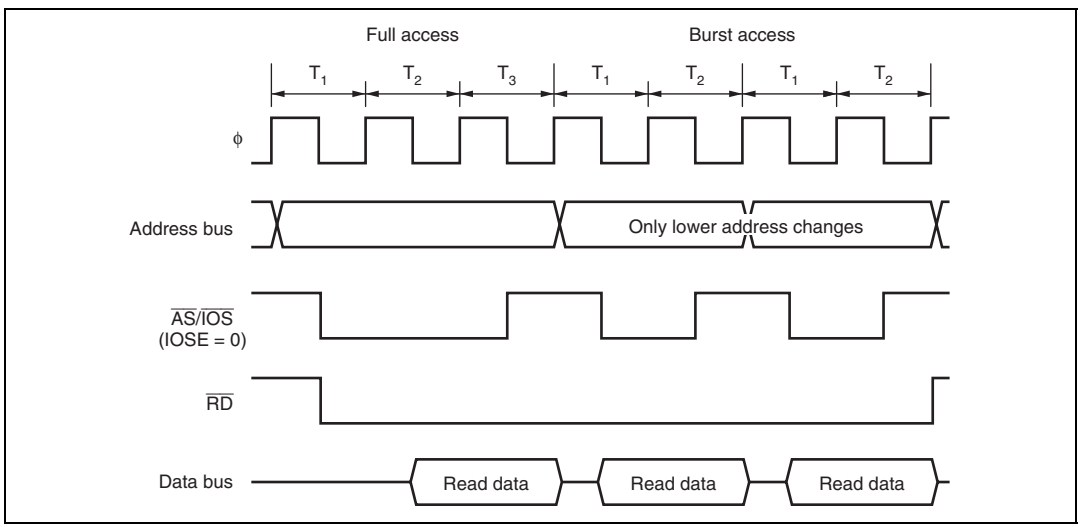


Figure 6.8 Access Timing Example in Burst ROM Space (AST = BRSTS1 = 1)

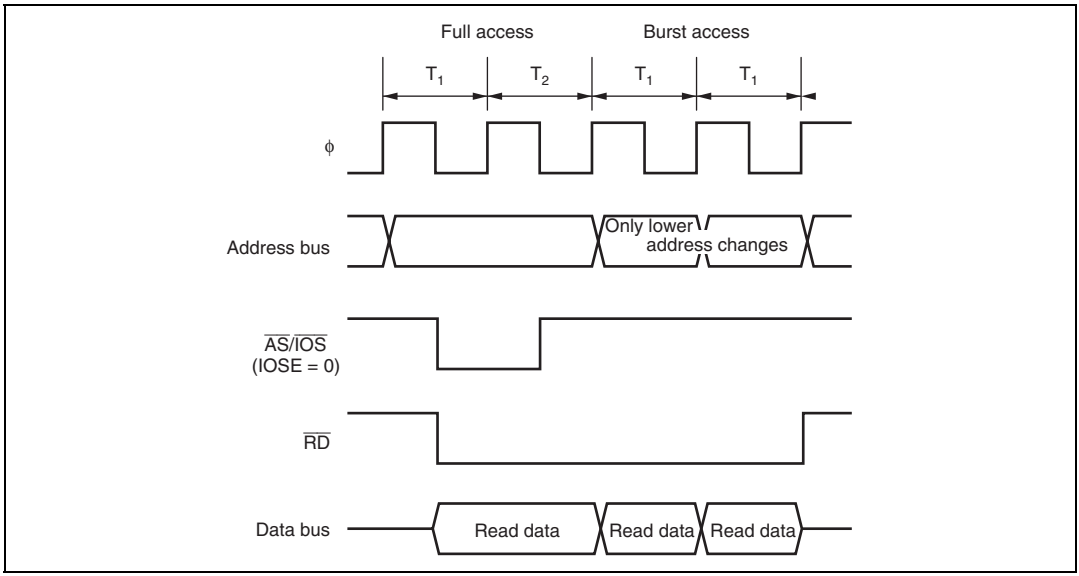


Figure 6.9 Access Timing Example in Burst ROM Space (AST = BRSTS1 = 0)

6.6.2 Wait Control

As with the basic bus interface, program wait insertion or pin wait insertion using the $\overline{\text{WAIT}}$ pin can be used in the initial cycle (full access) of the burst ROM interface. For details, see section 6.5.4, Wait Control. Wait states cannot be inserted in a burst cycle.

6.7 Idle Cycle

When this LSI accesses the external address space, it can insert a 1-state idle cycle (T_I) between bus cycles when a write cycle occurs immediately after a read cycle. By inserting an idle cycle it is possible, for example, to avoid data collisions between ROM with a long output floating time, and high-speed memory and I/O interfaces.

If an external write occurs after an external read while the ICIS0 bit is set to 1 in BCR, an idle cycle is inserted at the start of the write cycle.

Figure 6.10 shows examples of idle cycle operation. In these examples, bus cycle A is a read cycle for ROM with a long output floating time, and bus cycle B is a CPU write cycle. In figure 6.10 (a), with no idle cycle inserted, a collision occurs in bus cycle B between the read data from ROM and the CPU write data. In figure 6.10 (b), an idle cycle is inserted, thus preventing data collision.

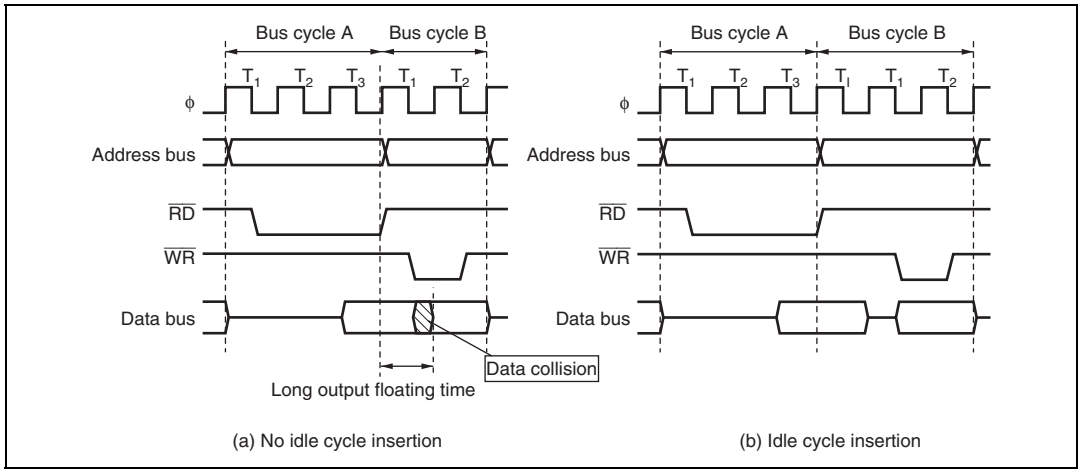


Figure 6.10 Examples of Idle Cycle Operation

Table 6.5 shows the pin states in an idle cycle.

Table 6.5 Pin States in Idle Cycle

Pins	Pin State
A15 to A0, $\overline{IOS}$	Contents of immediately following bus cycle
D7 to D0	High impedance
$\overline{AS}$	High
$\overline{RD}$	High
$\overline{WR}$	High

6.8 Bus Arbitration

The bus controller has a bus arbiter that arbitrates bus master operations. There are two bus masters – the CPU and DTC – that perform read/write operations when they have possession of the bus.

6.8.1 Priority of Bus Masters

Each bus master requests the bus by means of a bus request signal. The bus arbiter detects the bus masters' bus request signals, and if a bus request occurs, it sends a bus request acknowledge signal to the bus master making the request at the designated timing. If there are bus requests from more than one bus master, the bus request acknowledge signal is sent to the one with the highest priority. When a bus master receives the bus request acknowledge signal, it takes possession of the bus until that signal is canceled. The order of priority of the bus masters is as follows:

(High) DTC > CPU (Low)

6.8.2 Bus Transfer Timing

When a bus request is received from a bus master with a higher priority than that of the bus master that has acquired the bus and is currently operating, the bus is not necessarily transferred immediately. Each bus master can relinquish the bus at the timings given below.

(1) CPU

The CPU is the lowest-priority bus master, and if a bus request is received from the DTC, the bus arbiter transfers the bus to the DTC.

- DTC bus transfer timing
  - The bus is transferred at a break between bus cycles. However, if a bus cycle is executed in discrete operations, as in the case of a longword-size access, the bus is not transferred between the component operations. For details, refer to the H8S/2600 Series, H8S/2000 Series Software Manual.
  - If the CPU is in sleep mode, the bus is transferred immediately.

(2) DTC

The DTC has the highest bus master priority. The DTC sends the bus arbiter a request for the bus when an activation request is generated. The DTC does not release the bus until it completes its operation.

Section 7 Data Transfer Controller (DTC)

This LSI includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software, to transfer data.

Figure 7.1 shows a block diagram of the DTC. The DTC's register information is stored in the on-chip RAM. When the DTC is used, the RAME bit in SYSCR must be set to 1. A 32-bit bus connects the DTC to addresses H'(FF)EC00 to H'(FF)EFFF in on-chip RAM (1 Kbyte), enabling 32-bit/1-state reading and writing of the DTC register information.

7.1 Features

- Transfer is possible over any number of channels
- Three transfer modes
  Normal, repeat, and block transfer modes are available
- One activation source can trigger a number of data transfers (chain transfer)
- Direct specification of 16 Mbytes address space is possible
- Activation by software is possible
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC

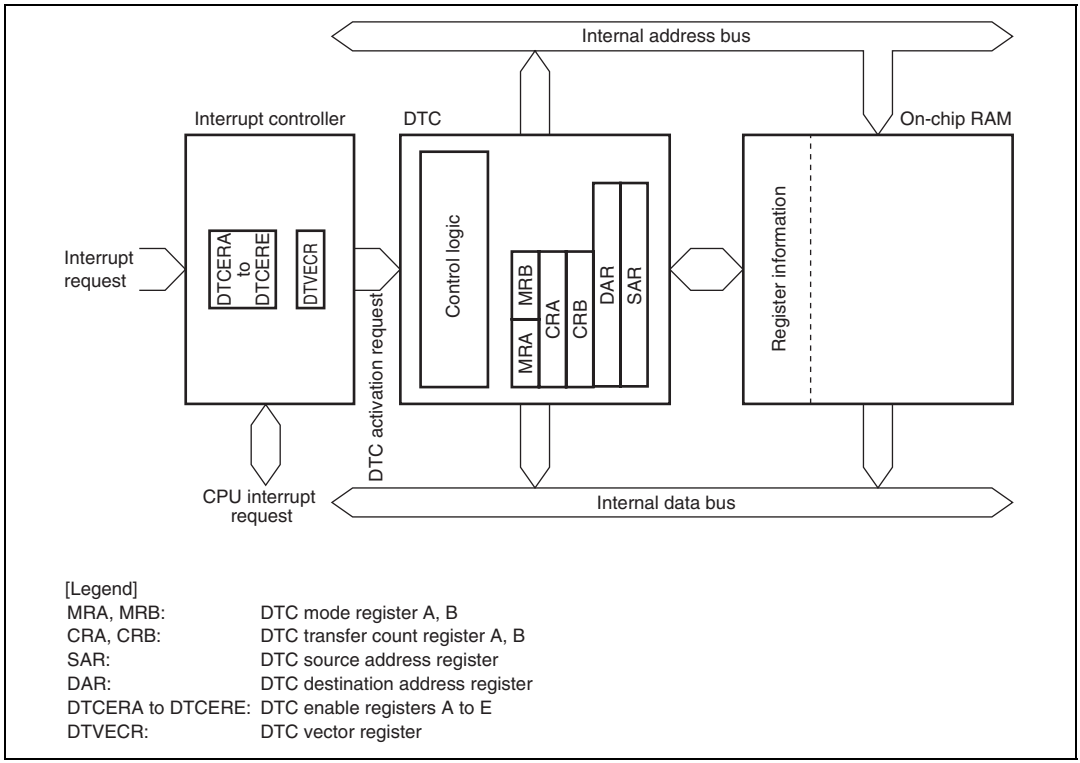


Figure 7.1 Block Diagram of DTC

7.2 Register Descriptions

The DTC has the following registers.

- DTC mode register A (MRA)
- DTC mode register B (MRB)
- DTC source address register (SAR)
- DTC destination address register (DAR)
- DTC transfer count register A (CRA)
- DTC transfer count register B (CRB)

These six registers cannot be directly accessed from the CPU. When a DTC activation interrupt source occurs, the DTC reads a set of register information that is stored in on-chip RAM to the corresponding DTC registers and transfers data. After the data transfer, it writes a set of updated register information back to on-chip RAM.

- DTC enable register (DTCER)
- DTC vector register (DTVECR)

7.2.1 DTC Mode Register A (MRA)

MRA selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	SM1	Undefined	—	Source Address Mode 1, 0
6	SM0	Undefined	—	These bits specify an SAR operation after a data transfer. 0*: SAR is fixed 10: SAR is incremented after a transfer (by +1 when Sz = 0, by +2 when Sz = 1) 11: SAR is decremented after a transfer (by –1 when Sz = 0, by –2 when Sz = 1)
5	DM1	Undefined	—	Destination Address Mode 1, 0
4	DM0	Undefined	—	These bits specify a DAR operation after a data transfer. 0*: DAR is fixed 10: DAR is incremented after a transfer (by +1 when Sz = 0, by +2 when Sz = 1) 11: DAR is decremented after a transfer (by –1 when Sz = 0, by –2 when Sz = 1)
3	MD1	Undefined	—	DTC Mode
2	MD0	Undefined	—	These bits specify the DTC transfer mode. 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: Setting prohibited
1	DTS	Undefined	—	DTC Transfer Mode Select Specifies whether the source side or the destination side is set to be a repeat area or block area in repeat mode or block transfer mode. 0: Destination side is repeat area or block area 1: Source side is repeat area or block area
0	Sz	Undefined	—	DTC Data Transfer Size Specifies the size of data to be transferred. 0: Byte-size transfer 1: Word-size transfer

[Legend]

*: Don't care

7.2.2 DTC Mode Register B (MRB)

MRB selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable When this bit is set to 1, a chain transfer will be performed. For details, see section 7.5.4, Chain Transfer. In data transfer with CHNE set to 1, determination of the end of the specified number of data transfers, clearing of the interrupt source flag, and clearing of DTCER are not performed.
6	DISEL	Undefined	—	DTC Interrupt Select When this bit is set to 1, a CPU interrupt request is generated every time data transfer ends. (DTC does not clear the interrupt source flag which is as an activation source, to 0.) When this bit is cleared to 0, a CPU interrupt request is generated only when the specified number of data transfers ends. (DTC does not clear the interrupt source flag which is as an activation source, to 0.)
5 to 0	—	All undefined	—	Reserved These bits have no effect on DTC operation. The write value should always be 0.

7.2.3 DTC Source Address Register (SAR)

SAR is a 24-bit register that designates the source address of data to be transferred by the DTC. For word-size transfer, specify an even source address.

7.2.4 DTC Destination Address Register (DAR)

DAR is a 24-bit register that designates the destination address of data to be transferred by the DTC. For word-size transfer, specify an even destination address.

7.2.5 DTC Transfer Count Register A (CRA)

CRA is a 16-bit register that designates the number of times data is to be transferred by the DTC.

In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 65536). It is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

In repeat mode or block transfer mode, the CRA is divided into two parts; the upper eight bits (CRAH) and the lower 8 bits (CRAL). CRAH holds the number of transfers while CRAL functions as an 8-bit transfer counter (1 to 256). CRAL is decremented by 1 every time data is transferred, and the contents of CRAH are sent when the count reaches H'00.

7.2.6 DTC Transfer Count Register B (CRB)

CRB is a 16-bit register that designates the number of times data is to be transferred by the DTC in block transfer mode. It functions as a 16-bit transfer counter (1 to 65536) that is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

7.2.7 DTC Enable Registers (DTCER)

DTCER specifies DTC activation interrupt sources. DTCER is comprised of five registers: DTCERA to DTCERE. The correspondence between interrupt sources and DTCE bits is shown in tables 7.1 and 7.2. For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. Multiple DTC activation sources can be set at one time (only at the initial setting) by masking all interrupts and writing data after executing a dummy read on the relevant register.

Bit	Bit Name	Initial Value	R/W	Description
7	DTCEn7	0	R/W	DTC Activation Enable
6	DTCEn6	0	R/W	Setting this bit to 1 specifies a relevant interrupt source as a DTC activation source.
5	DTCEn5	0	R/W	[Clearing conditions]
4	DTCEn4	0	R/W	• When data transfer has ended with the DISEL bit in MRB set to 1
3	DTCEn3	0	R/W	• When the specified number of transfers have ended
2	DTCEn2	0	R/W	These bits are not cleared when the DISEL bit is 0 and the specified number of transfers have not been completed
1	DTCEn1	0	R/W	
0	DTCEn0	0	R/W	

Note: n: A to E

Table 7.1 Correspondence between Interrupt Sources and DTCER

		Register				
Bit	Bit Name	DTCERA	DTCERB	DTCERC	DTCERD	DTCERE
7	DTCEn7	(16)IRQ0	(53)OCIB	(69)CMIB1	(86)TXI1	—
6	DTCEn6	(17)IRQ1	(56)TICI0	(72)CMIAY	—	—
5	DTCEn5	(18)IRQ2	(57)TCMI0	(73)CMIBY	—	—
4	DTCEn4	—	(60)TICI1	—	(92)IICI0	—
3	DTCEn3	(28)ADI	(61)TCMI1	—	(94)IICI1	—
2	DTCEn2	(48)ICIA	(64)CMIA0	(81)RXI0	—	—
1	DTCEn1	(49)ICIB	(65)CMIB0	(82)TXI0	—	—
0	DTCEn0	(52)OCIA	(68)CMIA1	(85)RXI1	—	—

Note: n: A to E
( ): Vector number

7.2.8 DTC Vector Register (DTVECR)

DTVECR enables or disables DTC activation by software, and sets a vector number for the software activation interrupt.

Bit	Bit Name	Initial Value	R/W	Description
7	SWDTE	0	R/W	DTC Software Activation Enable Setting this bit to 1 activates DTC. Only 1 can always be written to this bit. Writing 0 is enabled only after 1 has been read. [Clearing conditions] • When the DISEL bit is 0 and the specified number of transfers have not ended • When 0 is written to the DISEL bit after a software-activated data transfer end interrupt (SWDTEND) request has been sent to the CPU. [Holding conditions] • When the DISEL bit is set to 1 and data transfer has ended • The specified number of transfers have ended • On data transfer by software activation
6	DTVEC6	0	R/W	DTC Software Activation Vectors 6 to 0 These bits specify a vector number for DTC software activation. The vector address is expressed as H'0400 + (vector number × 2). For example, when DTVEC6 to DTVEC0 = H'10, the vector address is H'0420. When the SWDTE bit is 0, these bits can be written to.
5	DTVEC5	0	R/W	
4	DTVEC4	0	R/W	
3	DTVEC3	0	R/W	
2	DTVEC2	0	R/W	
1	DTVEC1	0	R/W	
0	DTVEC0	0	R/W	

7.3 Activation Sources

The DTC is activated by an interrupt request or by a write to DTVECR by software. The interrupt request source to activate the DTC is selected by DTCER. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the interrupt flag that became the activation source or the corresponding DTCER bit is cleared. The activation source flag, in the case of RXI1, for example, is the RDRF flag in SCI_1.

When an interrupt has been designated as a DTC activation source, the existing CPU mask level and interrupt controller priorities have no effect. If there is more than one activation source at the same time, the DTC operates in accordance with the default priorities. Figure 7.2 shows a block diagram of DTC activation source control. For details on the interrupt controller, see section 5, Interrupt Controller.

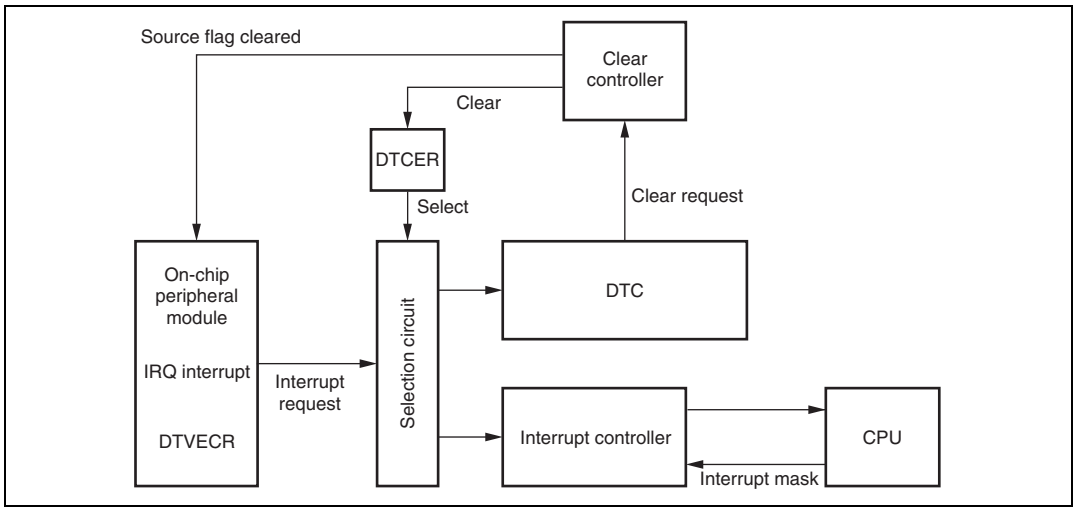


Figure 7.2 Block Diagram of DTC Activation Source Control

7.4 Location of Register Information and DTC Vector Table

Locate the register information in the on-chip RAM (addresses: H'(FF)EC00 to H'(FF)EFFF). Register information should be located at an address that is a multiple of four within the range. The method for locating the register information in address space is shown in figure 7.3. Locate MRA, SAR, MRB, DAR, CRA, and CRB, in that order, from the start address of the register information. In the case of chain transfer, register information should be located in consecutive areas as shown in figure 7.3, and the register information start address should be located at the vector address corresponding to the interrupt source in the DTC vector table. The DTC reads the start address of the register information from the vector table set for each activation source, and then reads the register information from that start address.

When the DTC is activated by software, the vector address is obtained from: H'0400 + (DTVECR[6:0] × 2). For example, if DTVECR is H'10, the vector address is H'0420.

The configuration of the vector address is a 2-byte unit. Specify the lower two bytes of the register information start address.

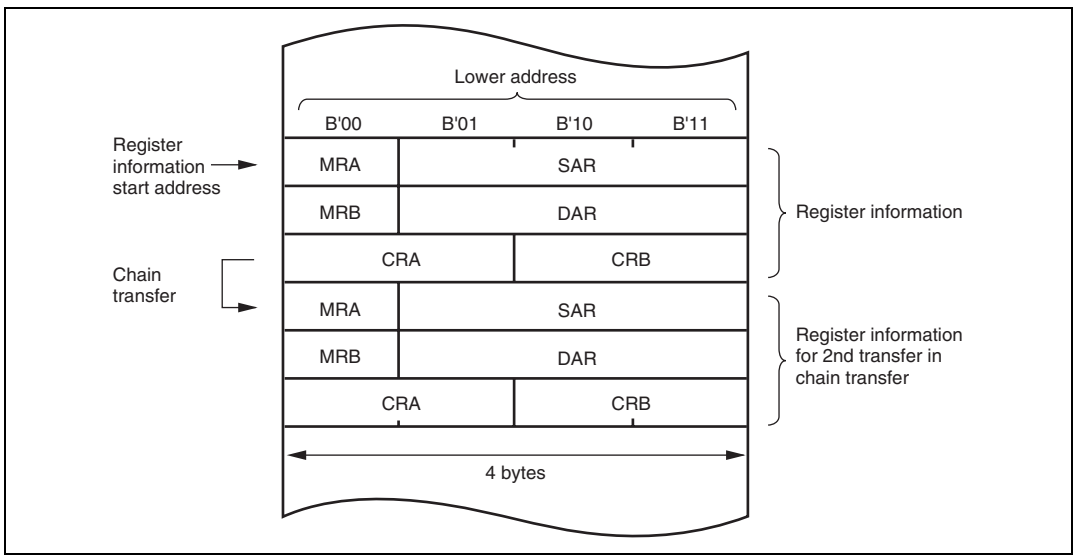


Figure 7.3 DTC Register Information Location in Address Space

Table 7.2 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Activation Source Origin	Activation Source	Vector Number	DTC Vector Address	DTCE*	Priority
Software	Write to DTVECR	DTVECR	H'0400 + (vector number x 2)	—	High
External pins	IRQ0	16	H'0420	DTCEA7	
	IRQ1	17	H'0422	DTCEA6	
	IRQ2	18	H'0424	DTCEA5	
ADC	ADI	28	H'0438	DTCEA3	
FRT	ICIA	48	H'0460	DTCEA2	
	ICIB	49	H'0462	DTCEA1	
	OCIA	52	H'0468	DTCEA0	
	OCIB	53	H'046A	DTCEB7	
TCM_0	TICI0	56	H'0470	DTCEB6	
	TCMI0	57	H'0472	DTCEB5	
TCM_1	TICI1	60	H'0478	DTCEB4	
	TCMI1	61	H'047A	DTCEB3	
TMR_0	CMIA0	64	H'0480	DTCEB2	
	CMIB0	65	H'0482	DTCEB1	
TMR_1	CMIA1	68	H'0488	DTCEB0	
	CMIB1	69	H'048A	DTCEC7	
TMR_Y	CMIAY	72	H'0490	DTCEC6	
	CMIBY	73	H'0492	DTCEC5	
SCI_0	RXI0	81	H'04A2	DTCEC2	
	TXI0	82	H'04A4	DTCED1	
SCI_1	RXI1	85	H'04AA	DTCED0	
	TXI1	86	H'04AC	DTCED7	
IIC_0	IICI0	92	H'04B8	DTCED4	
IIC_1	IICI1	94	H'04BC	DTCED3	Low

Note: * DTCE bits with no corresponding interrupt are reserved, and the write value should always be 0.

7.5 Operation

The DTC stores register information in on-chip RAM. When activated, the DTC reads register information in on-chip RAM and transfers data. After the data transfer, the DTC writes updated register information back to on-chip RAM. The pre-storage of register information in memory makes it possible to transfer data over any required number of channels. The transfer mode can be specified as normal, repeat, or block transfer mode. Setting the CHNE bit in MRB to 1 makes it possible to perform a number of transfers with a single activation source (chain transfer).

The 24-bit SAR designates the DTC transfer source address, and the 24-bit DAR designates the transfer destination address. After each transfer, SAR and DAR are independently incremented, decremented, or left fixed depending on its register information.

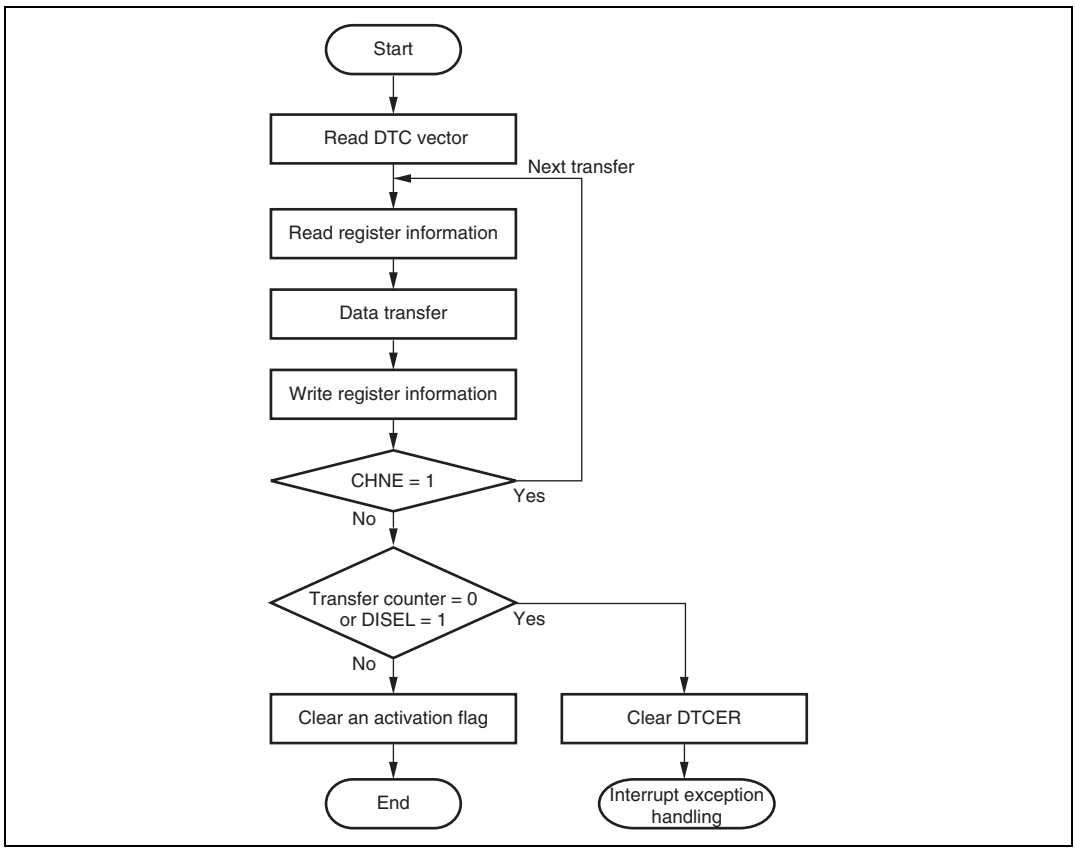


Figure 7.4 DTC Operation Flowchart

7.5.1 Normal Mode

In normal mode, one activation source transfers one byte or one word of data. Table 7.3 lists the register functions in normal mode. From 1 to 65,536 transfers can be specified. Once the specified number of transfers has been completed, a CPU interrupt can be requested.

Table 7.3 Register Functions in Normal Mode

Name	Abbreviation	Function
DTC source address register	SAR	Transfer source address
DTC destination address register	DAR	Transfer destination address
DTC transfer count register A	CRA	Transfer counter
DTC transfer count register B	CRB	Not used

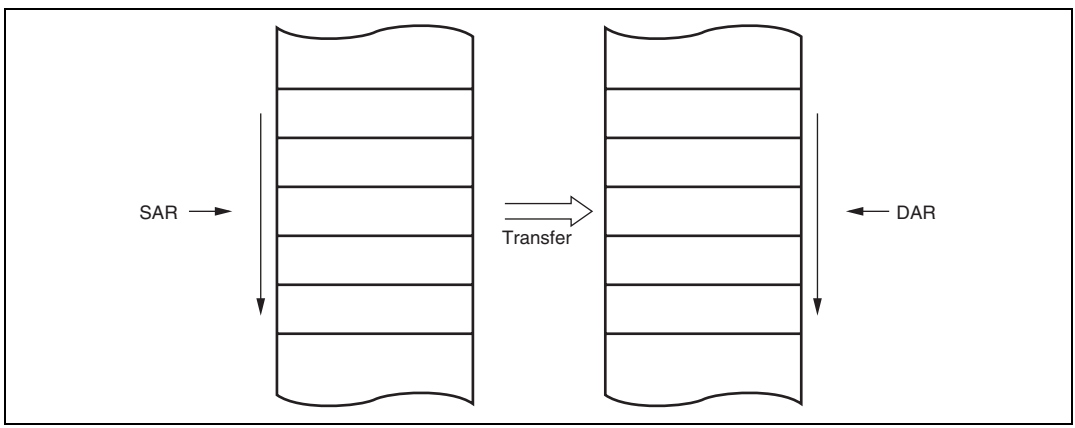


Figure 7.5 Memory Mapping in Normal Mode

7.5.2 Repeat Mode

In repeat mode, one activation source transfers one byte or one word of data. Table 7.4 lists the register functions in repeat mode. From 1 to 256 transfers can be specified. Once the specified number of transfers has been completed, the initial states of the transfer counter and the address register that is specified as the repeat area is restored, and transfer is repeated. In repeat mode, the transfer counter value does not reach H'00, and therefore CPU interrupts cannot be requested when the DISEL bit in MRB is cleared to 0.

Table 7.4 Register Functions in Repeat Mode

Name	Abbreviation	Function
DTC source address register	SAR	Transfer source address
DTC destination address register	DAR	Transfer destination address
DTC transfer count register AH	CRAH	Holds number of transfers
DTC transfer count register AL	CRAL	Transfer Count
DTC transfer count register B	CRB	Not used

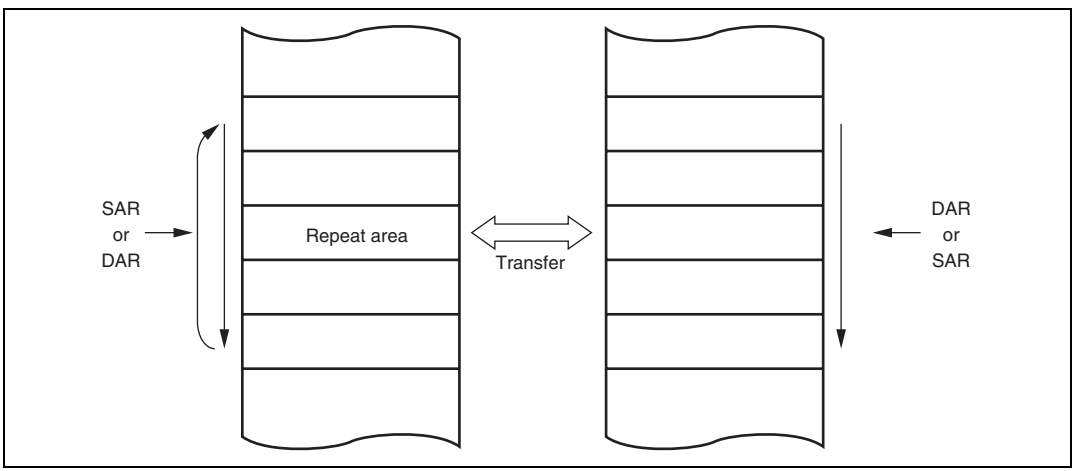


Figure 7.6 Memory Mapping in Repeat Mode

7.5.3 Block Transfer Mode

In block transfer mode, one activation source transfers one block of data. Either the transfer source or the transfer destination is designated as a block area. Table 7.5 lists the register functions in block transfer mode. The block size can be between 1 and 256. When the transfer of one block ends, the initial state of the block size counter and the address register that is specified as the block area is restored. The other address register is then incremented, decremented, or left fixed according to the register information. From 1 to 65,536 transfers can be specified. Once the specified number of transfers has been completed, a CPU interrupt is requested.

Table 7.5 Register Functions in Block Transfer Mode

Name	Abbreviation	Function
DTC source address register	SAR	Transfer source address
DTC destination address register	DAR	Transfer destination address
DTC transfer count register AH	CRAH	Holds block size
DTC transfer count register AL	CRAL	Block size counter
DTC transfer count register B	CRB	Transfer counter

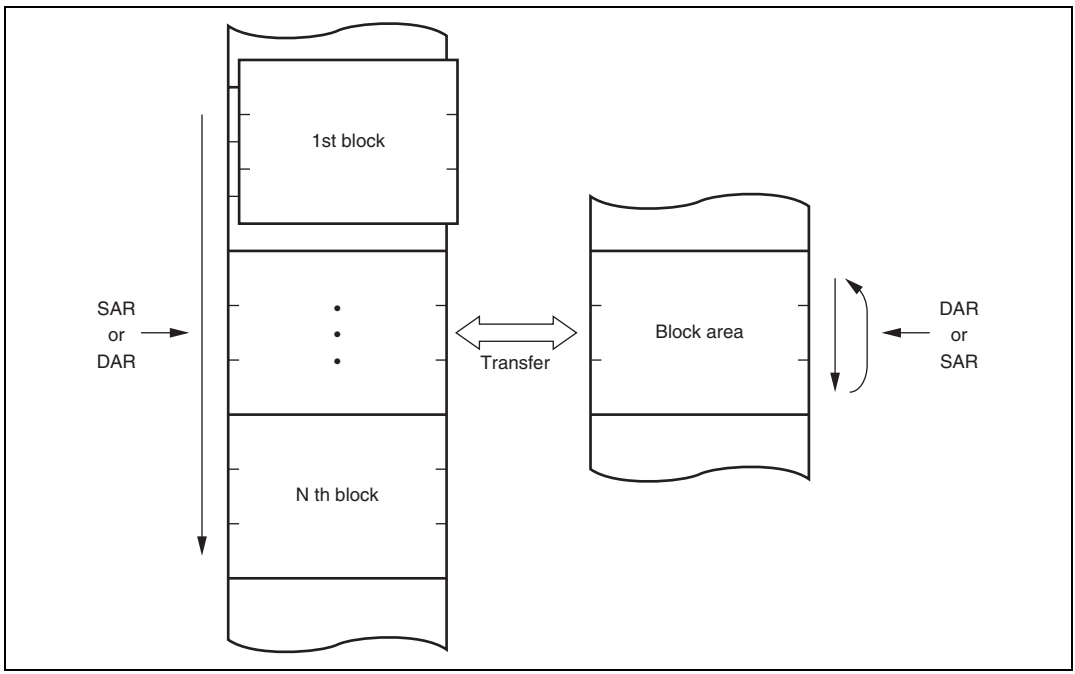


Figure 7.7 Memory Mapping in Block Transfer Mode

7.5.4 Chain Transfer

Setting the CHNE bit in MRB to 1 enables a number of data transfers to be performed consecutively in response to a single transfer request. SAR, DAR, CRA, CRB, MRA, and MRB, which define data transfers, can be set independently.

Figure 7.8 shows the overview of chain transfer operation. When activated, the DTC reads the register information start address stored at the DTC vector address, and then reads the first register information at that start address. After the data transfer, the CHNE bit will be tested. When it has been set to 1, DTC reads the next register information located in a consecutive area and performs the data transfer. These sequences are repeated until the CHNE bit is cleared to 0.

In the case of transfer with the CHNE bit set to 1, an interrupt request to the CPU is not generated at the end of the specified number of transfers or by setting of the DISEL bit to 1, and the interrupt source flag for the activation source is not affected.

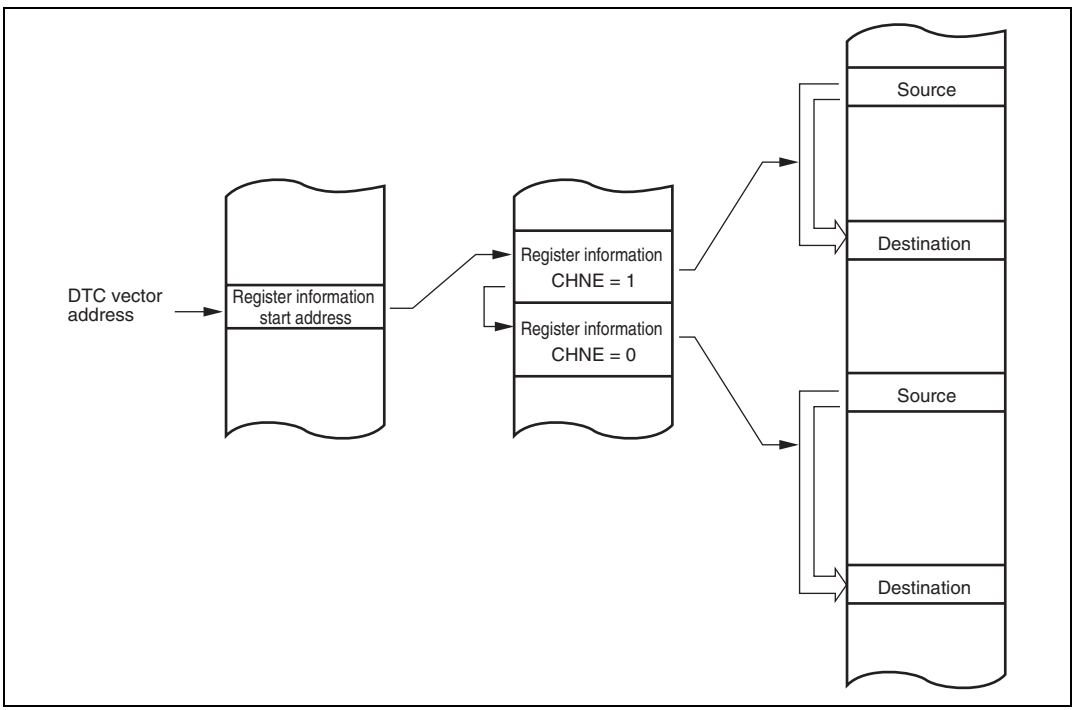


Figure 7.8 Chain Transfer Operation

7.5.5 Interrupt Sources

An interrupt request is issued to the CPU when the DTC has completed the specified number of data transfers, or a data transfer for which the DISEL bit was set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and priority level control by the interrupt controller.

In the case of software activation, a software-activated data transfer end interrupt (SWDTEND) is generated.

When the DISEL bit is 1 and one data transfer has been completed, or the specified number of transfers have been completed, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND interrupt is generated. The interrupt handling routine will then clear the SWDTE bit to 0.

When the DTC is activated by software, an SWDTEND interrupt is not generated during a data transfer wait or during data transfer even if the SWDTE bit is set to 1.

7.5.6 Operation Timing

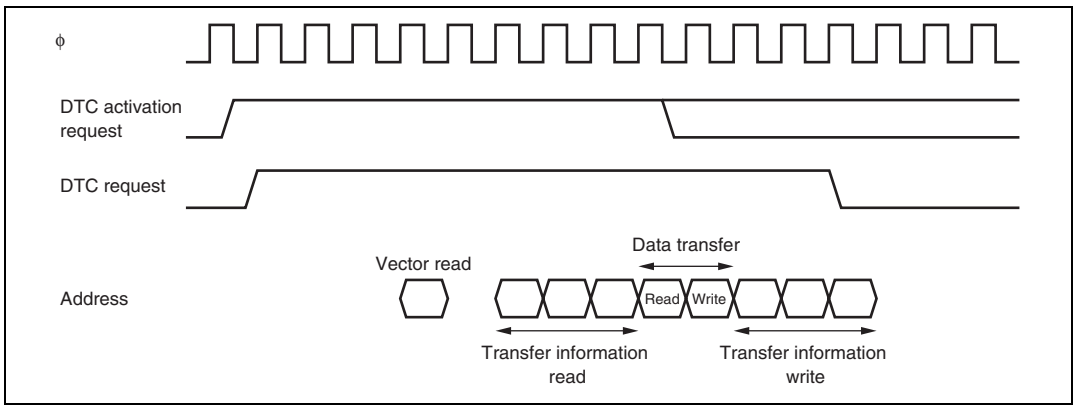


Figure 7.9 DTC Operation Timing (Example in Normal Mode or Repeat Mode)

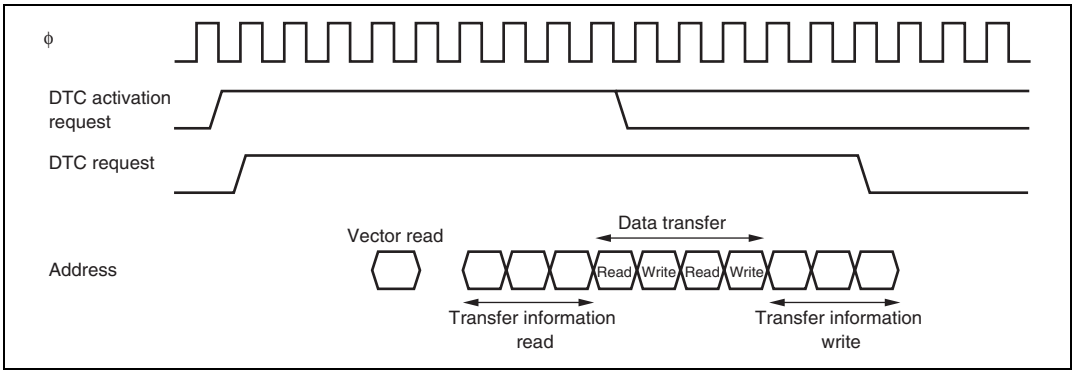


Figure 7.10 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)

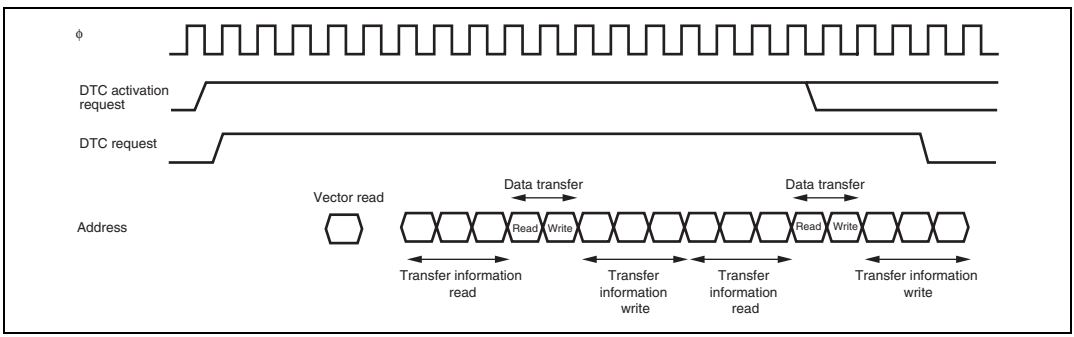


Figure 7.11 DTC Operation Timing (Example of Chain Transfer)

7.5.7 Number of DTC Execution States

Table 7.6 lists the execution status for a single DTC data transfer, and table 7.7 shows the number of states required for each execution status.

Table 7.6 DTC Execution Status

Mode	Vector Read I	Register Information Read/Write J	Data Read K	Data Write L	Internal Operations M
Normal	1	6	1	1	3
Repeat	1	6	1	1	3
Block transfer	1	6	N	N	3

[Legend]

N: Block size (initial setting of CRAH and CRAL)

Table 7.7 Number of States Required for Each Execution Status

Object to be Accessed		On-Chip RAM (H'(FF)EC00 to H'(FF)EFFF)	On-Chip RAM (On-chip RAM area other than H'(FF)EC00 to H'(FF)EFFF)	On-Chip ROM	On-Chip I/O Registers		External Device	
Bus width		32	16	16	8	16	8	8
Access states		1	1	1	2	2	2	3
Execution status	Vector read S_I	—	—	1	—	—	4	6 + 2m
	Register information read/write S_J	1	—	—	—	—	—	—
	Byte data read S_K	1	1	1	2	2	2	3 + m
	Word data read S_K	1	1	1	4	2	4	6 + 2m
	Byte data write S_L	1	1	1	2	2	2	3 + m
	Word data write S_L	1	1	1	4	2	4	6 + 2m
	Internal operation S_M	1	1	1	1	1	1	1

The number of execution states is calculated from using the formula below. Note that Σ is the sum of all transfers activated by one activation source (the number in which the CHNE bit is set to 1, plus 1).

$$\text{Number of execution states} = I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$$

For example, when the DTC vector address table is located in on-chip ROM, normal mode is set, and data is transferred from on-chip ROM to an internal I/O register, then the time required for the DTC operation is 13 states. The time from activation to the end of data write is 10 states.

7.6 Procedures for Using DTC

7.6.1 Activation by Interrupt

The procedure for using the DTC with interrupt activation is as follows:

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the enable bits for the interrupt sources to be used as the activation sources to 1. The DTC is activated when an interrupt used as an activation source is generated.
5. After one data transfer has been completed, or after the specified number of data transfers have been completed, the DTCE bit is cleared to 0 and a CPU interrupt is requested. If the DTC is to continue transferring data, set the DTCE bit to 1.

7.6.2 Activation by Software

The procedure for using the DTC with software activation is as follows:

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Check that the SWDTE bit is 0.
4. Write 1 to the SWDTE bit and the vector number to DTVECR.
5. Check the vector number written to DTVECR.
6. After one data transfer has been completed, if the DISEL bit is 0 and a CPU interrupt is not requested, the SWDTE bit is cleared to 0. If the DTC is to continue transferring data, set the SWDTE bit to 1. When the DISEL bit is 1 or after the specified number of data transfers have been completed, the SWDTE bit is held at 1 and a CPU interrupt is requested.

7.7 Examples of Use of the DTC

7.7.1 Normal Mode

An example is shown in which the DTC is used to receive 128 bytes of data via the SCI.

1. Set MRA to a fixed source address (SM1 = SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), normal mode (MD1 = MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one data transfer by one interrupt (CHNE = 0, DISEL = 0). Set the SCI, RDR address in SAR, the start address of the RAM area where the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any value.
2. Set the start address of the register information at the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the reception complete (RXI) interrupt. Since the generation of a receive error during the SCI reception operation will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.
5. Each time the reception of one byte of data has been completed on the SCI, the RDRF flag in SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive data is transferred from RDR to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag is automatically cleared to 0.
6. When CRA becomes 0 after 128 data transfers have been completed, the RDRF flag is held at 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The interrupt handling routine will perform wrap-up processing.

7.7.2 Software Activation

An example is shown in which the DTC is used to transfer a block of 128 bytes of data by means of software activation. The transfer source address is H'1000 and the transfer destination address is H'2000. The vector number is H'60, so the vector address is H'04C0.

1. Set MRA to incrementing source address (SM1 = 1, SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one block transfer by one interrupt (CHNE = 0). Set the transfer source address (H'1000) in SAR, the transfer destination address (H'2000) in DAR, and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.
2. Set the start address of the register information at the DTC vector address (H'04C0).
3. Check that the SWDTE bit in DTVECR is 0. Check that there is currently no transfer activated by software.
4. Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write data is H'E0.
5. Read DTVECR again and check that it is set to the vector number (H'60). If it is not, this indicates that the write failed. This is presumably because an interrupt occurred between steps 3 and 4 and led to a different software activation. To activate this transfer, go back to step 3.
6. If the write was successful, the DTC is activated and a block of 128 bytes of data is transferred.
7. After the transfer, an SWDTEND interrupt occurs. The interrupt handling routine should clear the SWDTE bit to 0 and perform wrap-up processing.

7.8 Usage Notes

7.8.1 Module Stop Mode Setting

DTC operation can be enabled or disabled by the module stop control register (MSTPCR). In the initial state, DTC operation is enabled. Access to DTC registers is disabled when module stop mode is set. Note that when the DTC is being activated, module stop mode can not be specified. For details, see section 22, Power-Down Modes.

7.8.2 On-Chip RAM

MRA, MRB, SAR, DAR, CRA, and CRB are all located in on-chip RAM. When the DTC is used, the RAME bit in SYSCR should not be cleared to 0.

7.8.3 DTCE Bit Setting

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR, for reading and writing. Multiple DTC activation sources can be set at one time (only at the initial setting) by masking all interrupts and writing data after executing a dummy read on the relevant register.

7.8.4 Setting Required on Entering Subactive Mode or Watch Mode

Set the MSTP14 bit in MSTPCRH to 1 to make the DTC enter module stop mode, then confirm that is set to 1 before making a transition to subactive mode or watch mode.

7.8.5 DTC Activation by Interrupt Sources of SCI, IIC, or A/D Converter

Interrupt sources of the SCI, IIC, or A/D converter are cleared when the DTC reads from or writes to the specified registers, and they cannot be cleared when the DTC reads from or writes to registers or memory that are not specified.

Section 8 I/O Ports

Table 8.1 is a summary of the port functions. The pins of each port also function as input/output pins of peripheral modules and interrupt input pins. Each input/output port includes a data direction register (DDR) that controls input/output and data registers (DR and ODR) that store output data. DDR, DR, and ODR are not provided for an input-only port.

Ports 1 to 3 and P43 to P45 have on-chip input pull-up MOSs. Port 1 to 3 can drive LEDs (with 5-mA current sink).

P47 and P52 in the H8S/2125 are NMOS push-pull output.

Table 8.1 Port Functions

Port	Description	Mode 1	Mode 2 (EXPE = 1)	Mode 3 (EXPE = 0)	I/O Status
Port 1	General I/O port also functioning as address output, PWM output pins, and PWMX output pins	A7	A7/P17/PW7	P17/PW7	On-chip input pull-up MOSs LED drive capability (sink current: 5 mA)
		A6	A6/P16/PW6	P16/PW6	
		A5	A5/P15/PW5	P15/PW5	
		A4	A4/P14/PW4	P14/PW4	
		A3	A3/P13/PW3	P13/PW3	
		A2	A2/P12/PW2	P12/PW2	
		A1	A1/P11/PW1/PWX1	P11/PW1/PWX1	
		A0	A0/P10/PW0/PWX0	P10/PW0/PWX0	
Port 2	General I/O port also functioning as address output pin, PWM output pin, SCI_1 I/O pin, and IIC_1 I/O pin	A15	A15/P27/PW15/SCK1	P27/PW15/SCK1	On-chip input pull-up MOSs LED drive capability (sink current: 5 mA)
		A14	A14/P26/PW14/RxD1	P26/PW14/RxD1	
		A13	A13/P25/PW13/TxD1	P25/PW13/TxD1	
		A12	A12/P24/PW12/SCL1	P24/PW12/SCL1	
		A11	A11/P23/PW11/SDA1	P23/PW11/SDA1	
		A10	A10/P22/PW10	P22/PW10	
		A9	A9/P21/PW9	P21/PW9	
		A8	A8/P20/PW8	P20/PW8	
Port 3	General I/O port also functioning as data bus I/O pin	D7		P37	On-chip input pull-up MOSs LED drive capability (sink current: 5 mA)
		D6		P36	
		D5		P35	
		D4		P34	
		D3		P33	
		D2		P32	
		D1		P31	
		D0		P30	

Port	Description	Mode 1	Mode 2 (EXPE = 1)	Mode 3 (EXPE = 0)	I/O Status
Port 4	General I/O port also functioning as extending data bus control I/O, IIC_0 I/O, subclock input, φ output, interrupt input, and A/D converter external trigger input pins	P47/$\overline{\text{WAIT}}$/SDA0		P47/SDA0	On-chip input pull-up MOSs (P45 to P43)
		P46/φ/EXCL		P46/φ/EXCL	
		$\overline{\text{AS}}$/$\overline{\text{IOS}}$		P45	
		$\overline{\text{WR}}$		P44	
		$\overline{\text{RD}}$		P43	
		P42/$\overline{\text{IRQ0}}$		P42/$\overline{\text{IRQ0}}$	
		P41/$\overline{\text{IRQ1}}$		P41/$\overline{\text{IRQ1}}$	
		P40/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$		P40/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$	
Port 5	General I/O port also functioning as SCI_0 input/output and IIC_0 input/output pins	P52/SCK0/SCL0			
		P51/RxD0			
		P50/TxD0			
Port 6	General I/O port also functioning as interrupt input, FRT I/O, TMR_0, TMR_1, TMR_X, and TMR_Y I/O pins	P67/$\overline{\text{IRQ3}}$/TMOX/TMO1			
		P66/FTOB/TMRI1			
		P65/FTID/TMCI1			
		P64/FTIC/TMO0			
		P63/FTIB/TMRI0			
		P62/FTIA/TMIY			
		P61/FTOA/TMOY			
		P60/FTCI/TMCI0/TMIX			
Port 7	General input port also functioning as A/D converter analog input, interrupt input, TCM_0, and TCM_1 input pins	P77/AN7/$\overline{\text{IRQ7}}$/TCMCKI0			
		P76/AN6/$\overline{\text{IRQ6}}$/TCMCYI0			
		P75/AN5/$\overline{\text{IRQ5}}$/TCMCKI1			
		P74/AN4/$\overline{\text{IRQ4}}$/TCMCYI1			
		P73/AN3			
		P72/AN2			
		P71/AN1			
		P70/AN0			

8.1 Port 1

Port 1 is an 8-bit I/O port. Port 1 pins also function as an address bus, PWM, and PWMX output pins. Port 1 functions change according to the operating mode. Port 1 has an on-chip input pull-up MOS function that can be controlled by software. Port 1 has the following registers.

- Port 1 data direction register (P1DDR)
- Port 1 data register (P1DR)
- Port 1 pull-up MOS control register (P1PCR)

8.1.1 Port 1 Data Direction Register (P1DDR)

P1DDR specifies input or output for the pins of port 1 on a bit-by-bit basis.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	In mode 1:
6	P16DDR	0	W	Each pin of port 1 is address output regardless of the set value of P1DDR.
5	P15DDR	0	W	
4	P14DDR	0	W	In modes 2 and 3 (EXPE=1):
3	P13DDR	0	W	The corresponding port 1 pins are address output or PWM output ports when P1DDR bits are set to 1, and input ports when cleared to 0.
2	P12DDR	0	W	
1	P11DDR	0	W	In modes 2 and 3 (EXPE=0):
0	P10DDR	0	W	The corresponding port 1 pins are output ports or PWM outputs when the P1DDR bits are set to 1, and input ports when cleared to 0.

8.1.2 Port 1 Data Register (P1DR)

P1DR stores output data for the port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DR	0	R/W	If a port 1 read is performed while the P1DDR bits are set to 1, the P1DR values are read. If a port 1 read is performed while the P1DDR bits are cleared to 0, the pin states are read.
6	P16DR	0	R/W	
5	P15DR	0	R/W	
4	P14DR	0	R/W	
3	P13DR	0	R/W	
2	P12DR	0	R/W	
1	P11DR	0	R/W	
0	P10DR	0	R/W	

8.1.3 Port 1 Pull-Up MOS Control Register (P1PCR)

P1PCR controls the on/off status of the port 1 on-chip input pull-up MOSs.

Bit	Bit Name	Initial Value	R/W	Description
7	P17PCR	0	R/W	When the pins are in input state, the corresponding input pull-up MOS is turned on when a P1PCR bit is set to 1.
6	P16PCR	0	R/W	
5	P15PCR	0	R/W	
4	P14PCR	0	R/W	
3	P13PCR	0	R/W	
2	P12PCR	0	R/W	
1	P11PCR	0	R/W	
0	P10PCR	0	R/W	

8.1.4 Pin Functions

- P17/A7/PW7, P16/A6/PW6, P15/A5/PW5, P14/A4/PW4, P13/A3/PW3, P12/A2/PW2

  The pin function is switched as shown below according to the combination of the OEn bit in PWOERA of PWM, the P1nDDR bit, and operating mode.

Operating Mode	Mode 1	Mode 2, 3 (EXPE = 1)			Mode 2, 3 (EXPE = 0)		
P1nDDR	—	0	1		0	1	
OEn	—	—	0	1	—	0	1
Pin Function	A7 to A2 output pins	P17 to P12 input pins	A7 to A2 output pins	PW7 to PW2 output pins	P17 to P12 input pins	P17 to P12 output pins	PW7 to PW2 output pins

Note: n = 7 to 2

- P11/A1/PW1/PWX1

  The pin function is switched as shown below according to the combination of the OE1 bit in PWOERA of PWM, the OEB bit in DACR of PWMX, the P11DDR bit, and operating mode.

Operating Mode	Mode 1	Mode 2, 3 (EXPE = 1)				Mode 2, 3 (EXPE = 0)			
OEB	—	0			1	0			1
P11DDR	—	0	1		—	0	1		—
OE1	—	—	0	1	—	—	0	1	—
Pin Function	A1 output pin	P11 input pin	A1 output pin	PW1 output pin	PWX1 output pin	P11 input pin	P11 output pin	PW1 output pin	PWX1 output pin

- P10/A0/PW0/PWX0

  The pin function is switched as shown below according to the combination of the OE0 bit in PWOERA of PWM, the OEA bit in DACR of PWMX, the P10DDR bit, and operating mode.

Operating Mode	Mode 1	Mode 2, 3 (EXPE = 1)				Mode 2, 3 (EXPE = 0)			
OEA	—	0			1	0			1
P10DDR	—	0	1		—	0	1		—
OE0	—	—	0	1	—	—	0	1	—
Pin Function	A0 output pin	P10 input pin	A0 output pin	PW0 output pin	PWX0 output pin	P10 input pin	P10 output pin	PW0 output pin	PWX0 output pin

8.1.5 Port 1 Input Pull-Up MOS

Port 1 has an on-chip input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be specified as on or off on a bit-by-bit basis.

Table 8.2 summarizes the input pull-up MOS states.

Table 8.2 Input Pull-Up MOS States (Port 1)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1	Off	Off	Off	Off
2, 3			On/Off	On/Off

[Legend]

Off: Input pull-up MOS is always off.

On/Off: On when the pin is in the input state, P1DDR = 0, and P1PCR = 1; otherwise off.

8.2 Port 2

Port 2 is an 8-bit I/O port. Port 2 pins also function as address bus output pins, PWM output pins, and SCI_1 and IIC_1 I/O pins. Port 2 functions change according to the operating mode. Port 2 has an on-chip input pull-up MOS function that can be controlled by software. Port 2 has the following registers.

- Port 2 data direction register (P2DDR)
- Port 2 data register (P2DR)
- Port 2 pull-up MOS control register (P2PCR)

8.2.1 Port 2 Data Direction Register (P2DDR)

P2DDR specifies input or output for the pins of port 2 on a bit-by-bit basis.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DDR	0	W	In Mode 1: The corresponding port 2 pins are address outputs, regardless of the P2DDR setting. Modes 2 and 3 (EXPE = 1): The corresponding port 2 pins are address outputs or PWM outputs when P2DDR bits are set to 1, and input ports when cleared to 0. P27 to P24 are switched from address outputs to output ports by setting the IOSE bit to 1. To ensure normal access to external space, P27 should not be set as an on-chip peripheral module output pin when port 2 pins are used as address output pins. Modes 2 and 3 (EXPE = 0): The corresponding port 2 pins are output ports or PWM outputs when P2DDR bits are set to 1, and input ports when cleared to 0.
6	P26DDR	0	W	
5	P25DDR	0	W	
4	P24DDR	0	W	
3	P23DDR	0	W	
2	P22DDR	0	W	
1	P21DDR	0	W	
0	P20DDR	0	W	

8.2.2 Port 2 Data Register (P2DR)

P2DR stores output data for port 2.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DR	0	R/W	If a port 2 read is performed while P2DDR bits are set to 1, the P2DR values are read directly, regardless of the actual pin states. If a port 2 read is performed while P2DDR bits are cleared to 0, the pin states are read.
6	P26DR	0	R/W	
5	P25DR	0	R/W	
4	P24DR	0	R/W	
3	P23DR	0	R/W	
2	P22DR	0	R/W	
1	P21DR	0	R/W	
0	P20DR	0	R/W	

8.2.3 Port 2 Pull-Up MOS Control Register (P2PCR)

P2PCR controls the port 2 on-chip input pull-up MOSs.

Bit	Bit Name	Initial Value	R/W	Description
7	P27PCR	0	R/W	In modes 2 and 3, the input pull-up MOS is turned on when a P2PCR bit is set to 1 in the input port state.
6	P26PCR	0	R/W	
5	P25PCR	0	R/W	
4	P24PCR	0	R/W	
3	P23PCR	0	R/W	
2	P22PCR	0	R/W	
1	P21PCR	0	R/W	
0	P20PCR	0	R/W	

8.2.4 Pin Functions

To ensure normal access to external space, P27 should not be set as an on-chip peripheral module output pin when port 2 pins are used as address output pins.

- P27/A15/PW15/SCK1

  The pin function is switched as shown below according to the combination of the IOSE bit in SYSCR, the C/$\overline{A}$ bit in SMR of SCI_1, the CKE0 and CKE1 bits in SCR of SCI_1, the OE15 bit in PWOERB of PWM, the P27DDR bit, and operating mode.

Operating Mode	Mode 1	Mode 2, 3 (EXPE = 1)							Mode 2, 3 (EXPE = 0)					
CKE1	—	0						1	0					1
C/$\overline{A}$	—	0					1	—	0				1	—
CKE0	—	0				1	—	—	0			1	—	—
P27DDR	—	0	1			—	—	—	0	1		—	—	—
OE15	—	—	0		1	—	—	—	—	0	1	—	—	—
IOSE	—	—	0	1	—	—	—	—	—					
Pin Function	A15 output pin	P27 input pin	A15 output pin	P27 output pin	PW15 output pin	SCK1 output pin		SCK1 input pin	P27 input pin	P27 output pin	PW15 output pin	SCK1 output pin		SCK1 input pin

- P26/A14/PW14/RxD1

  The pin function is switched as shown below according to the combination of the IOSE bit in SYSCR, the RE bit in SCR of SCI_1, the OE14 bit in PWOERB of PWM, the P26DDR bit, and operating mode.

Operating Mode	Mode 1	Mode 2, 3 (EXPE = 1)					Mode 2, 3 (EXPE = 0)			
RE	—	0				1	0			1
P26DDR	—	0	1			—	0	1		—
OE14	—	—	0		1	—	—	0	1	—
IOSE	—	—	0	1	—	—	—			
Pin Function	A14 output pin	P26 input pin	A14 output pin	P26 output pins	PW14 output pin	RxD1 input pin	P26 input pin	P26 output pin	PW14 output pin	RxD1 input pin

- P25/A13/PW13/TxD1

  The pin function is switched as shown below according to the combination of the IOSE bit in SYSCR, the TE bit in SCR of SCI_1, the OE13 bit in PWOERB of PWM, the P25DDR bit, and operating mode.

Operating Mode	Mode 1	Mode 2, 3 (EXPE = 1)					Mode 2, 3 (EXPE = 0)			
TE	—	0				1	0			1
P25DDR	—	0	1			—	0	1		—
OE13	—	—	0		1	—	—	0	1	—
IOSE	—	—	0	1	—	—	—			
Pin Function	A13 output pin	P25 input pin	A13 output pin	P25 output pin	PW13 output pin	TxD1 input pin	P25 input pin	P25 output pin	PW13 output pin	TxD1 input pin

- P24/A12/PW12/SCL1

  The pin function is switched as shown below according to the combination of the IOSE bit in SYSCR, the ICE bit in ICCR of IIC_1, the OE12 bit in PWOERB of PWM, the P24DDR bit, and operating mode.

Operating Mode	Mode 1	Mode 2, 3 (EXPE = 1)					Mode 2, 3 (EXPE = 0)			
ICE	—	0				1	0			1
P24DDR	—	0	1			—	0	1		—
OE12	—	—	0		1	—	—	0	1	—
IOSE	—	—	0	1	—	—	—			
Pin Function	A12 output pin	P24 input pin	A12 output pin	P24 output pin	PW12 output pin	SCL1 I/O pin	P24 input pin	P24 output pin	PW12 output pin	SCL1 I/O pin

- P23/A11/PW11/SDA1

  The pin function is switched as shown below according to the combination of the ICE bit in ICCR of IIC_1, the OE11 bit in PWOERB of PWM, the P23DDR bit, and operating mode.

Operating Mode	Mode 1	Mode 2, 3 (EXPE = 1)				Mode 2, 3 (EXPE = 0)			
ICE	—	0			1	0			1
P23DDR	—	0	1		—	0	1		—
OE11	—	—	0	1	—	—	0	1	—
Pin Function	A11 output pin	P23 input pin	A11 output pin	PW11 output pin	SDA1 I/O pin	P23 input pin	P23 output pin	PW11 output pin	SDA1 I/O pin

- P22/A10/PW10, P21/A9/PW9, P20/A8/PW8

  The pin function is switched as shown below according to the combination of the OEm bit in PWOERB of PWM, the P2nDDR bit, and operating mode.

Operating Mode	Mode 1	Mode 2, 3 (EXPE = 1)			Mode 2, 3 (EXPE = 0)		
P2nDDR	—	0	1		0	1	
OEm	—	—	0	1	—	0	1
Pin Function	A10 to A8 output pins	P22 to P20 input pins	A10 to A8 output pins	PW10 to PW8 output pins	P22 to P20 input pins	P22 to P20 output pins	PW10 to PW8 output pins

Note: n = 2 to 0
m = 10 to 8

8.2.5 Port 2 Input Pull-Up MOS

Port 2 has an on-chip input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be specified as on or off on a bit-by-bit basis.

Table 8.3 summarizes the input pull-up MOS states.

Table 8.3 Input Pull-Up MOS States (Port 2)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1	Off	Off	Off	Off
2, 3			On/Off	On/Off

[Legend]

Off: Input pull-up MOS is always off.

On/Off: On when the pin is in the input state, P2DDR = 0, and P2PCR = 1; otherwise off.

8.3 Port 3

Port 3 is an 8-bit I/O port. Port 3 pins also function as a bidirectional data bus. Port 3 functions change according to the operating mode. Port 3 has the following registers.

- Port 3 data direction register (P3DDR)
- Port 3 data register (P3DR)
- Port 3 pull-up MOS control register (P3PCR)

8.3.1 Port 3 Data Direction Register (P3DDR)

P3DDR specifies input or output for the pins of port 3 on a bit-by-bit basis.

Bit	Bit Name	Initial Value	R/W	Description
7	P37DDR	0	W	Modes 1, 2, and 3 (EXPE = 1) The input/output direction specified by P3DDR is ignored, and pins automatically function as data I/O pins. Modes 2 and 3 (EXPE = 0) The corresponding port 3 pins are output ports when P3DDR bits are set to 1, and input ports when cleared to 0.
6	P36DDR	0	W	
5	P35DDR	0	W	
4	P34DDR	0	W	
3	P33DDR	0	W	
2	P32DDR	0	W	
1	P31DDR	0	W	
0	P30DDR	0	W	

8.3.2 Port 3 Data Register (P3DR)

P3DR stores output data of port 3.

Bit	Bit Name	Initial Value	R/W	Description
7	P37DR	0	R/W	If a port 3 read is performed while P3DDR bits are set to 1, the P3DR values are read directly, regardless of the actual pin states. If a port 3 read is performed while P3DDR bits are cleared to 0, the pin states are read.
6	P36DR	0	R/W	
5	P35DR	0	R/W	
4	P34DR	0	R/W	
3	P33DR	0	R/W	
2	P32DR	0	R/W	
1	P31DR	0	R/W	
0	P30DR	0	R/W	

8.3.3 Port 3 Pull-Up MOS Control Register (P3PCR)

P3PCR controls the port 3 on-chip input pull-up MOSs on a bit-by-bit basis.

Bit	Bit Name	Initial Value	R/W	Description
7	P37PCR	0	R/W	In modes 2 and 3 (when EXPE = 0), the input pull-up MOS is turned on when a P3PCR bit is set to 1 in the input port state.
6	P36PCR	0	R/W	
5	P35PCR	0	R/W	
4	P34PCR	0	R/W	
3	P33PCR	0	R/W	
2	P32PCR	0	R/W	
1	P31PCR	0	R/W	
0	P30PCR	0	R/W	

8.3.4 Pin Functions

- P37/D7, P36/D6, P35/D5, P34/D4, P33/D3, P32/D2, P31/D1, P30/D0

  The pin function is switched as shown below according to the combination of the P3nDDR bit and operating mode.

Operating Mode	Mode 1, 2, 3 (EXPE = 1)	Mode 2, 3 (EXPE = 0)	
P3nDDR	—	0	1
Pin Function	D7 to D0 input/output pins	P37 to P30 input pins	P37 to P30 output pins

Note: n = 7 to 0

8.3.5 Port 3 Input Pull-Up MOS

Port 3 has an on-chip input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be specified as on or off on a bit-by-bit basis.

Table 8.4 summarizes the input pull-up MOS states.

Table 8.4 Input Pull-Up MOS States (Port 3)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1, 2, 3 (EXPE = 1)	Off	Off	Off	Off
2, 3 (EXPE = 0)			On/Off	On/Off

[Legend]

Off: Input pull-up MOS is always off.

On/Off: On when the pin is in the input state, P3DDR = 0, and P3PCR = 1; otherwise off.

8.4 Port 4

Port 4 is an 8-bit I/O port. Port 4 pins also function as external interrupt input pins, the A/D converter input pins, the IIC_0 I/O pins, the subclock input pins, bus control signal I/O pins, and the system clock (ϕ) output pins. P47 is an NMOS push-pull output. SDA0 is an NMOS open-drain output, and has direct bus drive capability. Port 4 has the following registers.

- Port 4 data direction register (P4DDR)
- Port 4 data register (P4DR)
- Port 4 pull-up MOS control register (P4PCR)
- Port 4 noise canceller enable register (P4NCE)
- Port 4 noise canceller determine control register (P4NCMC)
- Port 4 noise canceller cycle setting register (P4NCCS)

8.4.1 Port 4 Data Direction Register (P4DDR)

P4DDR specifies input or output for the pins of port 4 on a bit-by-bit basis.

Bit	Bit Name	Initial Value	R/W	Description
7	P47DDR	0	W	P4DDR is initialized to H'40 (mode 1) or H'00 (modes 2 and 3). Modes 1, 2, and 3 (EXPE = 1): Pin P47 functions as a bus control input ($\overline{\text{WAIT}}$), the IIC_0 I/O pin (SDA0), or an I/O port, according to the wait mode setting. When P97 functions as an I/O port, it becomes an output port when P97DDR is set to 1, and an input port when P97DDR is cleared to 0. Pin P46 functions as the ϕ output pin when P46DDR is set to 1, and as the subclock input (EXCL) or an input port when P96DDR is cleared to 0. Pins P45 to P43 automatically become bus control outputs ($\overline{\text{AS}}$/$\overline{\text{IOS}}$, $\overline{\text{WR}}$, $\overline{\text{RD}}$), regardless of the input/output direction indicated by P45DDR to P43DDR. Pins P42 to P40 become output ports when P42DDR to P40DDR are set to 1, and input ports when P42DDR to P40DDR are cleared to 0. Modes 2 and 3 (EXPE = 0): When the corresponding P4DDR bits are set to 1, pin P46 functions as the ϕ output pin and pins P47 and P45 to P40 become output ports. When P4DDR bits are cleared to 0, the corresponding pins become input ports.
6	P46DDR	1/0*	W	
5	P45DDR	0	W	
4	P44DDR	0	W	
3	P43DDR	0	W	
2	P42DDR	0	W	
1	P41DDR	0	W	
0	P40DDR	0	W	

Note: * The initial value of P46DDR is 1 (mode 1) or 0 (modes 2 and 3).

8.4.2 Port 4 Data Register (P4DR)

P4DR stores output data for the port 4 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P47DR	0	R/W	P4DR stores output data other than the bit 6 for the port 4 pins that are used as the general output port. If a port 4 read is performed while the P4DDR bits are set to 1, the P4DR values are read. If a port 4 read is performed while the P4DDR bits are cleared to 0, the pin states are read.
6	P46DR	Undefined*	R/W	
5	P45DR	0	R/W	
4	P44DR	0	R/W	
3	P43DR	0	R/W	
2	P42DR	0	R/W	
1	P41DR	0	R/W	
0	P40DR	0	R/W	

Note: * Undefined value is determined depends on P46 pin state.

8.4.3 Port 4 Pull-Up MOS Control Register (P4PCR)

P4PCR controls the on/off state of the input pull-up MOS for port 4 pins.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved The initial value should not be changed.
5	P45PCR	0	R/W	When the pins are in input state, the corresponding input pull-up MOS is turned on when a P4PCR bit is set to 1.
4	P44PCR	0	R/W	
3	P43PCR	0	R/W	
2 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

8.4.4 Port 4 Noise Canceller Enable Register (P4NCE)

P4NCE enables or disables the noise cancel circuit at port 4 in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	P47NCE	0	R/W	Noise cancel circuit is enabled when P4NCE bit is set to 1, and the pin state is fetched in the P4DR in the sampling cycle set by the P4NCCS.
6	P46NCE	0	R/W	
5	P45NCE	0	R/W	
4	P44NCE	0	R/W	
3	P43NCE	0	R/W	
2	P42NCE	0	R/W	
1	P41NCE	0	R/W	
0	P40NCE	0	R/W	

8.4.5 Port 4 Noise Canceller Mode Control Register (P4NCMC)

P4NCMC controls whether 1 or 0 is expected for the input signal to port 4 in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	P47NCMC	0	R/W	1 expected: 1 is stored in the port data register when 1 is input stably 0 expected: 0 is stored in the port data register when 0 is input stably
6	P46NCMC	0	R/W	
5	P45NCMC	0	R/W	
4	P44NCMC	0	R/W	
3	P43NCMC	0	R/W	
2	P42NCMC	0	R/W	
1	P41NCMC	0	R/W	
0	P40NCMC	0	R/W	

8.4.6 Port 4 Noise Cancel Cycle Setting Register (P4NCCS)

P4NCCS controls the sampling cycles of the noise canceller.

Bit	Bit Name	Initial Value	R/W	Description		
7 to 3	—	Undefined	R/W	Reserved The read data is undefined. The write value should always be 0.		
2	P4NCCK2	0	R/W	These bits set the sampling cycles of the noise canceller.		
1	P4NCCK1	0	R/W	When ϕ is 10 MHz		
0	P4NCCK0	0	R/W	000:	0.80 µs	ϕ/2
				001:	12.8 µs	ϕ/32
				010:	3.3 ms	ϕ/8192
				011:	6.6 ms	ϕ/16384
				100:	13.1 ms	ϕ/32768
				101:	26.2 ms	ϕ/65536
				110:	52.4 ms	ϕ/131072
				111:	104.9 ms	ϕ/262144

8.4.7 Pin Functions

- P47/$\overline{\text{WAIT}}$/SDA0

  The pin function is switched as shown below according to the combination of operating mode, the WMS1 bit in WSCR, the ICE bit in ICCR of IIC_0, and the P47DDR bit.

Operating Mode	Modes 1, 2, 3 (EXPE = 1)				Modes 2, 3 (EXPE = 0)		
WMS1	0			1	—		
ICE	0		1	—	0		1
P47DDR	0	1	—	—	0	1	—
Pin Function	P47 input pin	P47 output pin	SDA0 I/O pin	$\overline{\text{WAIT}}$ input pin	P47 input pin	P47 output pin	SDA0 I/O pin

Note: When this pin is set as the P47 output pin, it is an NMOS push-pull output. SDA0 is an NMOS open-drain output, and has direct bus drive capability.

- P46/φ/EXCL

  The pin function is switched as shown below according to the combination of the EXCLE bit in LPWRCR and the P46DDR bit.

P46DDR	0		1
EXCLE	0	1	0
Pin Function	P46 input pin	EXCL input pin	φ output pin

Note: When this pin is used as the EXCL input pin, P46DDR should be cleared to 0.

- P45/$\overline{AS}$/$\overline{IOS}$

  The pin function is switched as shown below according to the combination of operating mode, the IOSE bit in SYSCR and the P45DDR bit.

Operating Mode	Modes 1, 2, 3 (EXPE = 1)		Modes 2, 3 (EXPE = 0)	
P45DDR	—		0	1
IOSE	0	1	—	—
Pin Function	$\overline{AS}$ output pin	$\overline{IOS}$ output pin	P45 input pin	P45 output pin

- P44/$\overline{WR}$

  The pin function is switched as shown below according to the combination of operating mode and the P44DDR bit.

Operating Mode	Modes 1, 2, 3 (EXPE = 1)	Modes 2, 3 (EXPE = 0)	
P44DDR	—	0	1
Pin Function	$\overline{WR}$ output pin	P44 input pin	P44 output pin

- P43/$\overline{RD}$

  The pin function is switched as shown below according to the combination of operating mode and the P43DDR bit.

Operating Mode	Modes 1, 2, 3 (EXPE = 1)	Modes 2, 3 (EXPE = 0)	
P43DDR	—	0	1
Pin Function	$\overline{RD}$ output pin	P43 input pin	P43 output pin

- P42/$\overline{\text{IRQ0}}$

P42DDR	0	1
Pin Function	P42 input pin	P42 output pin
	$\overline{\text{IRQ0}}$ input pin*	

Note: * When bit IRQ0E in IER is set to 1, this pin is used as the $\overline{\text{IRQ0}}$ input pin.

- P41/$\overline{\text{IRQ1}}$

P41DDR	0	1
Pin Function	P41 input pin	P41 output pin
	$\overline{\text{IRQ1}}$ input pin*	

Note: * When the bit IRQ1E in IER is set to 1, this pin is used as the $\overline{\text{IRQ1}}$ input pin.

- P40/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$

  The pin function is switched as shown below according to the combination of the P40DDR bit.

P40DDR	0	1
Pin Function	P40 input pin	P40 output pin
	$\overline{\text{IRQ2}}$ input pin, $\overline{\text{ADTRG}}$ input pin*	

Note: * When the IRQ2E bit in IER is set to 1, this pin is used as the $\overline{\text{IRQ2}}$ input pin. When both of the TRGS1 and TRGS0 bits in ADCR of the A/D converter are set to 1, this pin is used as the $\overline{\text{ADTRG}}$ input pin.

8.5 Port 5

Port 5 is a 3-bit I/O port. Port 5 pins also function as SCI_0 I/O pins, and the IIC_0 I/O pin. P52 and SCK0 are NMOS push-pull outputs, and SCL0 is an NMOS open-drain output. Port 5 has the following registers.

- Port 5 data direction register (P5DDR)
- Port 5 data register (P5DR)

8.5.1 Port 5 Data Direction Register (P5DDR)

P5DDR specifies input or output for the pins of port 5 on a bit-by-bit basis.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 1	—	Reserved The initial value must not be changed.
2	P52DDR	0	W	The corresponding port 5 pins are output ports when P5DDR bits are set to 1, and input ports when cleared to 0. As SCI_0 is initialized in software standby mode, the pin states are determined by the IIC_0 ICCR, P5DDR, and P5DR specifications.
1	P51DDR	0	W	
0	P50DDR	0	W	

8.5.2 Port 5 Data Register (P5DR)

P5DR stores output data for port 5 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 1	—	Reserved The initial value must not be changed.
2	P52DR	0	R/W	If a port 5 read is performed while P5DDR bits are set to 1, the P5DR values are read directly, regardless of the actual pin states. If a port 5 read is performed while P5DDR bits are cleared to 0, the pin states are read.
1	P51DR	0	R/W	
0	P50DR	0	R/W	

8.5.3 Pin Functions

- P52/SCK0/SCL0

  The pin function is switched as shown below according to the combination of the CKE1 and CKE0 bits in SCR of SCI_0, the C/$\overline{A}$ bit in SMR of SCI_0, the ICE bit in ICCR of IIC_0, and the P52DDR bit.

ICE	0					1
CKE1	0				1	0
C/$\overline{A}$	0			1	—	0
CKE0	0		1	—	—	0
P52DDR	0	1	—	—	—	—
Pin Function	P52 input pin	P52 output pin	SCK0 output pin	SCK0 output pin	SCK0 input pin	SCL0 I/O pin

Note: When this pin is used as the SCL0 I/O pin, bits CKE1 and CKE0 in SCR of SCI0 and bit C/$\overline{A}$ in SMR of SCI0 must all be cleared to 0.
SCL0 is an NMOS open-drain output, and has direct bus drive capability.
When set as the P52 output pin or SCK0 output pin, this pin is an NMOS push-pull output.

- P51/RxD0

  The pin function is switched as shown below according to the combination of the RE bit in SCR of SCI_0 and the P51DDR bit.

RE	0		1
P51DDR	0	1	—
Pin Function	P51 input pin	P51 output pin	RxD0 input pin

- P50/TxD0

  The pin function is switched as shown below according to the combination of the TE bit in SCR of SCI_0 and the P50DDR bit.

TE	0		1
P50DDR	0	1	—
Pin Function	P50 input pin	P50 output pin	TxD0 output pin

8.6 Port 6

Port 6 is an 8-bit I/O port. Port 6 pins also function as the noise cancel pins and I/O pins for FRT, TMR_0, TMR_1, TMR_X, and TMR_Y. Port 6 has the following registers.

- Port 6 data direction register (P6DDR)
- Port 6 data register (P6DR)
- Port 6 noise canceller enable register (P6NCE)
- Port 6 noise canceller mode control register (P6NCMC)
- Port 6 noise cancel cycle setting register (P6NCCS)

8.6.1 Port 6 Data Direction Register (P6DDR)

P6DDR specifies input or output for the pins of port 6 on a bit-by-bit basis.

Bit	Bit Name	Initial Value	R/W	Description
7	P67DDR	0	W	The corresponding port 6 pins are output ports when P6DDR bits are set to 1, and input ports when cleared to 0.
6	P66DDR	0	W	
5	P65DDR	0	W	
4	P64DDR	0	W	
3	P63DDR	0	W	
2	P62DDR	0	W	
1	P61DDR	0	W	
0	P60DDR	0	W	

8.6.2 Port 6 Data Register (P6DR)

P6DR stores output data for port 6.

Bit	Bit Name	Initial Value	R/W	Description
7	P67DR	0	R/W	If a port 6 read is performed while P6DDR bits are set to 1, the P6DR values are read directly, regardless of the actual pin states. If a port 6 read is performed while P6DDR bits are cleared to 0, the pin states are read.
6	P66DR	0	R/W	
5	P65DR	0	R/W	
4	P64DR	0	R/W	
3	P63DR	0	R/W	
2	P62DR	0	R/W	
1	P61DR	0	R/W	
0	P60DR	0	R/W	

8.6.3 Port 6 Noise Canceller Enable Register (P6NCE)

P6NCE specifies enable or disable for noise cancel circuit for the pins of port 6 on a bit-by-bit basis.

Bit	Bit Name	Initial Value	R/W	Description
7	P67NCE	0	R/W	Noise cancel circuit is enabled when P6NCCS bit is set to 1, and the pin state is fetched in the P6DR in the sampling cycle set by the P6NCCS.
6	P66NCE	0	R/W	
5	P65NCE	0	R/W	
4	P64NCE	0	R/W	
3	P63NCE	0	R/W	
2	P62NCE	0	R/W	
1	P61NCE	0	R/W	
0	P60NCE	0	R/W	

8.6.4 Port 6 Noise Canceller Mode Control Register (P6NCMC)

P6NCMC controls whether 1 or 0 is expected for the input signal to port 6 in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	P67NCMC	0	R/W	1 expected: 1 is stored in the port data register when 1 is input stably 0 expected: 0 is stored in the port data register when 0 is input stably
6	P66NCMC	0	R/W	
5	P65NCMC	0	R/W	
4	P64NCMC	0	R/W	
3	P63NCMC	0	R/W	
2	P62NCMC	0	R/W	
1	P61NCMC	0	R/W	
0	P60NCMC	0	R/W	

8.6.5 Port 6 Noise Cancel Cycle Setting Register (P6NCCS)

P6NCCS controls the sampling cycles of the noise canceller.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	R/W	Reserved The read data is undefined. The write value should always be 0.
2	P6NCCK2	0	R/W	These bits set the sampling cycles of the noise canceller.
1	P6NCCK1	0	R/W	When ϕ is 10 MHz
0	P6NCCK0	0	R/W	000: 0.80 µs ϕ/2 001: 12.8 µs ϕ/32 010: 3.3 ms ϕ/8192 011: 6.6 ms ϕ/16384 100: 13.1 ms ϕ/32768 101: 26.2 ms ϕ/65536 110: 52.4 ms ϕ/131072 111: 104.9 ms ϕ/262144

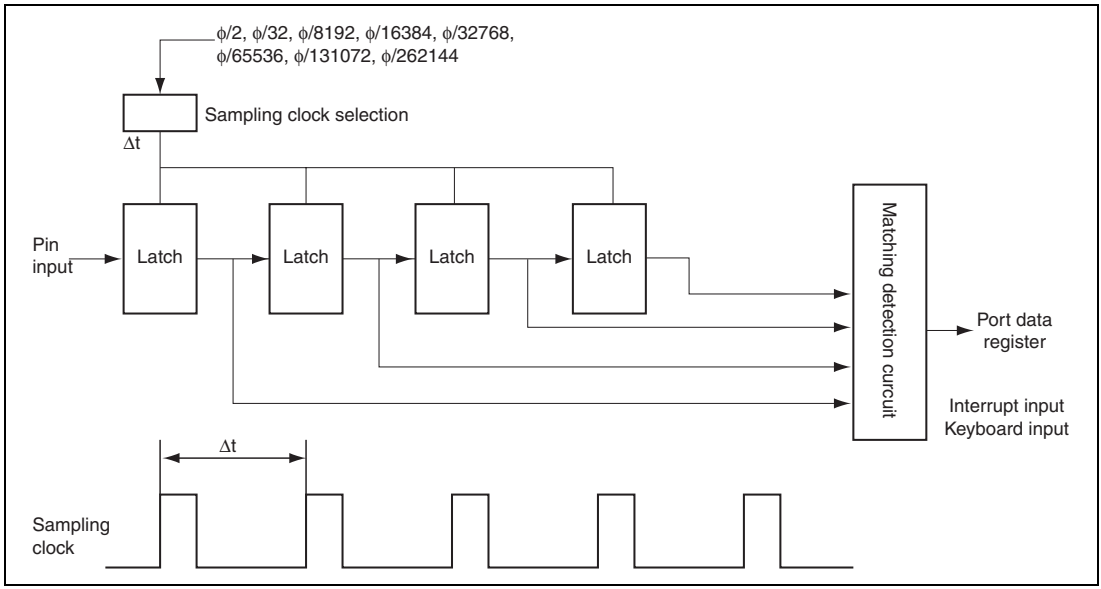


Figure 8.1 Noise Cancel Circuit

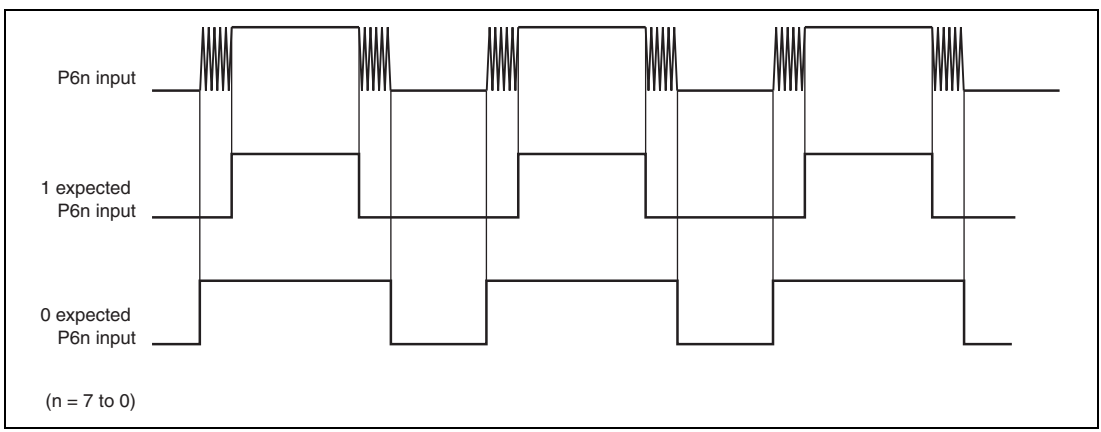


Figure 8.2 Conceptual Diagram of Noise Cancel Operation

8.6.6 Pin Functions

- P67/$\overline{\text{IRQ3}}$/TMOX/TMO1

  The pin function is switched as shown below according to the OS3 to OS0 bits in TCSR of TMR_1 and TMR_X and the P67DDR bit. When the IRQ3E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ3}}$ input pin.

OS3 to OS0 (TMR_X)	All 0			Not all 0
OS3 to OS0 (TMR_1)	All 0		Not all 0	—
P67DDR	0	1	—	—
Pin Function	P67 input pin	P67 output pin	TMO1 output pin	TMOX output pin
	$\overline{\text{IRQ3}}$ input pin			

- P66/FTOB/TMRI1

  The pin function is switched as shown below according to the combination of the OEB bit in TOCR of FRT and the P66DDR bit. When the CCLR1 and CCLR0 bits in TCR of TMR_1 are both set to 1, this pin can be used as the TMRI1 input pin.

OEB	0		1
P66DDR	0	1	—
Pin Function	P66 input pin	P66 output pin	FTOB output pin
	TMRI1 input pin		

- P65/FTID/TMCI1

  The pin function is switched as shown below according to the P65DDR bit. When the ICIDE bit in TIER of FRT is set to 1, this pin can be used as the FTID input pin. When the external clock is selected with the CKS2 to CKS0 bits in TCR of TMR_1, this pin can be used as the TMCI1 input pin.

P65DDR	0	1
Pin Function	P65 input pin	P65 output pin
	FTID input pin/TMCI1 input pin	

- P64/FTIC/TMO0

  The pin function is switched as shown below according to the combination of the OS3 to OS0 bits in TCSR of TMR_0 and the P64DDR bit. When the ICICE bit in TIER of FRT is set to 1, this pin can be used as the FTIC input pin.

OS3 to OS0	All 0		Not all 0
P64DDR	0	1	—
Pin Function	P64 input pin	P64 output pin	TMO0 output pin
	FTIC input pin		

- P63/FTIB/TMRI0

  The pin function is switched as shown below according to the P63DDR bit. When the ICIBE bit in TIER of FRT is set to 1, this pin is used as the FTIB input pin. When the CCLR1 and CCLR0 bits in TCR of TMR_0 are both set to 1, this pin can be used as the TMRI0 input pin.

P63DDR	0	1
Pin Function	P63 input pin	P63 output pin
	FTIB input pin/TMRI0 input pin	

- P62/FTIA/TMIY

  The pin function is switched as shown below according to the P62DDR bit. When the ICIAE bit in TIER of FRT is set to 1, this pin can be used as the FTIA input pin. When the CCLR1 and CCLR0 bits in TCR of TMR_Y are both set to 1, this pin can be used as the TMIY (TMRIY) input pin.

P62DDR	0	1
Pin Function	P62 input pin	P62 output pin
	FTIA input pin/TMIY input pin	

- P61/FTOA/TMOY

  The pin function is switched as shown below according to the combination of the OEA bit in TOCR of FRT, the OS3 to OS0 bits in TCSR of TMR_Y, and the P61DDR bit.

OS3 to OS0	All 0			Not all 0
OEA	0		1	—
P61DDR	0	1	—	—
Pin Function	P61 input pin	P61 output pins	FTOA output pin	TMOY output pin

- P60/FTCI/TMCI0/TMIX

  The pin function is switched as shown below according to the P60DDR bit. When the CKS1 and CKS0 bits in TCR of FRT are both set to 1, this pin can be used as the FTCI input pin. When the CCLR1 and CCLR0 bits in TCR of TMR_X are both set to 1, this pin can be used as the TMIX (TMRIX) input pin. When the external clock is selected with the CKS2 to CKS0 bits in TCR of TMR_0, this pin can be used as the TMCI0 input pin.

P60DDR	0	1
Pin Function	P60 input pin	P60 output pin
	FTCI input pin/TMCI0 input pin/TMIX input pin	

8.7 Port 7

Port 7 is an 8-bit input port. Port 7 pins also function as the interrupt input pins, A/D converter analog input pins, TCM_0 input pin, and TCM_1 input pin. Port 7 has the following register.

- Port 7 input data register (P7PIN)

8.7.1 Port 7 Input Data Register (P7PIN)

P7PIN reflects the pin states of port 7.

Bit	Bit Name	Initial Value	R/W	Description
7	P77PIN	Undefined*	R	When this register is read, the pin states are read.
6	P76PIN	Undefined*	R	
5	P75PIN	Undefined*	R	
4	P74PIN	Undefined*	R	
3	P73PIN	Undefined*	R	
2	P72PIN	Undefined*	R	
1	P71PIN	Undefined*	R	
0	P70PIN	Undefined*	R	

Note: * The initial value is determined in accordance with the pin states of P77 to P70.

8.7.2 Pin Functions

- P77/AN7/$\overline{\text{IRQ7}}$/TCMCKI0

  When the IRQ7E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ7}}$ input pin. When the external clock is selected with the CKS2 to CKS0 bits in TCMCR_0 of TCM_0, this pin can be used as the TCMCKI0 input pin.

Pin Function	P77 input pin/AN7 input pin/$\overline{\text{IRQ7}}$ input pin/TCMCKI0 input pin

- P76/AN6/$\overline{\text{IRQ6}}$/TCMCYI0

  When the IRQ6E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ6}}$ input pin. When the TCMIPE bit in TCMIER_0 of TCM_0 is set to 1, this pin can be used as the TCMCYI0 input pin.

Pin Function	P76 input pin/AN6 input pin/$\overline{\text{IRQ6}}$ input pin/TCMCYI0 input pin

- P75/AN5/$\overline{\text{IRQ5}}$/TCMCYI1

  When the IRQ5E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ5}}$ input pin. When the external clock is selected with the CKS2 to CKS0 bits in TCMCR_1 of TCM_1, this pin can be used as the TCMCKI1 input pin.

Pin Function	P75 input pin/AN5 input pin/$\overline{\text{IRQ5}}$ input pin/TCMCKI1 input pin

- P74/AN4/$\overline{\text{IRQ4}}$/TCMCYI1

  When the IRQ4E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ4}}$ input pin. When the TCMIPE bit in TCMIER_1 of TCM_1 is set to 1, this pin can be used as the TCMCYI1 input pin.

Pin Function	P74 input pin/AN4 input pin/$\overline{\text{IRQ4}}$ input pin/TCMCYI1 input pin

- P73/AN3, P72/AN2, P71/AN1, P70/AN0

Pin Function	P7n input pin/ANn input pin

Note: n = 3 to 0

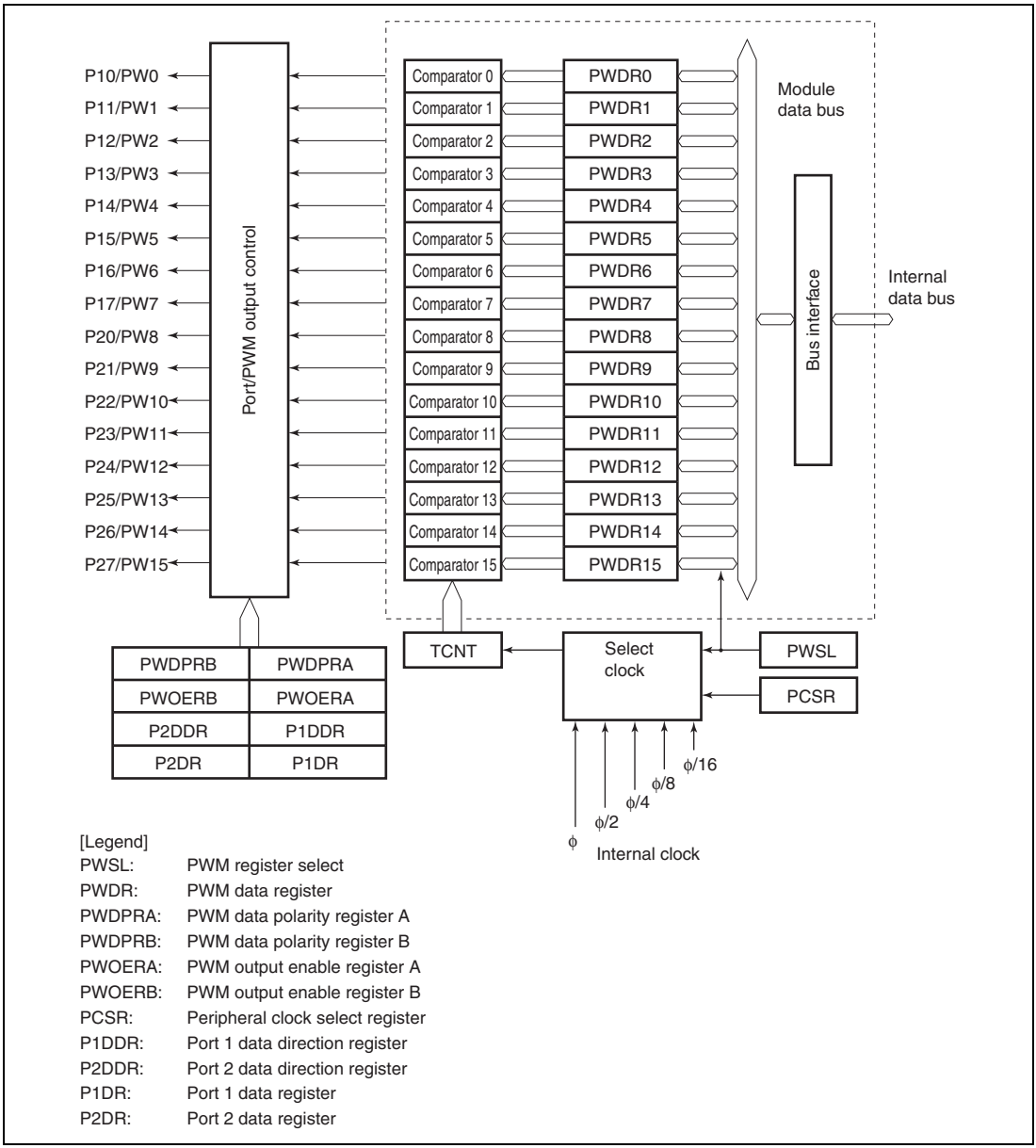
Section 9 8-Bit PWM Timer (PWM)

This LSI has an on-chip pulse width modulation (PWM) timer with sixteen outputs. Sixteen output waveforms are generated from a common time base, enabling PWM output with a high carrier frequency to be produced using pulse division.

9.1 Features

- Operable at a maximum carrier frequency of 1.25 MHz using pulse division (at 20 MHz operation)
- Duty cycles from 0 to 100% with 1/256 resolution (100% duty realized by port output)
- Direct or inverted PWM output, and PWM output enable/disable control

Figure 9.1 shows a block diagram of the PWM timer.



[Legend]

PWSL:	PWM register select
PWDR:	PWM data register
PWDPRA:	PWM data polarity register A
PWDPRB:	PWM data polarity register B
PWOERA:	PWM output enable register A
PWOERB:	PWM output enable register B
PCSR:	Peripheral clock select register
P1DDR:	Port 1 data direction register
P2DDR:	Port 2 data direction register
P1DR:	Port 1 data register
P2DR:	Port 2 data register

Figure 9.1 Block Diagram of PWM Timer

9.2 Input/Output Pin

Table 9.1 shows the PWM output pins.

Table 9.1 Pin Configuration

Name	Abbreviation	I/O	Function
PWM output 15 to 0	PW15 to PW0	Output	PWM timer pulse output 15 to 0

9.3 Register Descriptions

The PWM has the following registers. To access PCSR, the FLSHE bit in the serial timer control register (STCR) must be cleared to 0. For details on the serial timer control register (STCR), see section 3.2.3, Serial Timer Control Register (STCR).

- PWM register select (PWSL)
- PWM data registers 0 to 15 (PWDR0 to PWDR15)
- PWM data polarity register A (PWDPRA)
- PWM data polarity register B (PWDPRB)
- PWM output enable register A (PWOERA)
- PWM output enable register B (PWOERB)
- Peripheral clock select register (PCSR)

9.3.1 PWM Register Select (PWSL)

PWSL is used to select the input clock and the PWM data register.

Bit	Bit Name	Initial Value	R/W	Description
7	PWCKE	0	R/W	PWM Clock Enable
6	PWCKS	0	R/W	PWM Clock Select These bits, together with bits PWCKB and PWCKA in PCSR, select the internal clock input to TCNT in the PWM. For details, see table 9.2. The resolution, PWM conversion period, and carrier frequency depend on the selected internal clock, and can be obtained from the following equations. Resolution (minimum pulse width) = 1/internal clock frequency PWM conversion period = resolution × 256 Carrier frequency = 16/PWM conversion period With a 20 MHz system clock (ϕ), the resolution, PWM conversion period, and carrier frequency are as shown in table 9.3.
5	—	1	R	Reserved This bit is always read as 1 and cannot be modified.
4	—	0	R	Reserved This bit is always read as 0 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
3	RS3	0	R/W	Register Select
2	RS2	0	R/W	These bits select the PWM data register.
1	RS1	0	R/W	0000: PWDR0 selected
0	RS0	0	R/W	0001: PWDR1 selected
				0010: PWDR2 selected
				0011: PWDR3 selected
				0100: PWDR4 selected
				0101: PWDR5 selected
				0110: PWDR6 selected
				0111: PWDR7 selected
				1000: PWDR8 selected
				1001: PWDR9 selected
				1010: PWDR10 selected
				1011: PWDR11 selected
				1100: PWDR12 selected
				1101: PWDR13 selected
				1110: PWDR14 selected
				1111: PWDR15 selected

Table 9.2 Internal Clock Selection

PWSL		PCSR			
PWCKE	PWCKS	PWCKB	PWCKA	Description	
0	—	—	—	Clock input is disabled	(Initial value)
1	0	—	—	ϕ (system clock) is selected	
	1	0	0	ϕ/2 is selected	
			1	ϕ/4 is selected	
		1	0	ϕ/8 is selected	
			1	ϕ/16 is selected	

Table 9.3 Resolution, PWM Conversion Period and Carrier Frequency when ϕ = 20 MHz

Internal Clock Frequency	Resolution	PWM Conversion Period	Carrier Frequency
ϕ	50 ns	12.8 µs	1250 kHz
ϕ/2	100 ns	25.6 µs	625 kHz
ϕ/4	200 ns	51.2 µs	312.5 kHz
ϕ/8	400 ns	102.4 µs	156.3 kHz
ϕ/16	800 ns	204.8 µs	78.1 kHz

9.3.2 PWM Data Registers (PWDR0 to PWDR15)

PWDR are 8-bit readable/writable registers. The PWM has sixteen PWM data registers. Each PWDR specifies the duty cycle of the basic pulse to be output, and the number of additional pulses. The value set in PWDR corresponds to a 0 or 1 ratio in the conversion period. The upper four bits specify the duty cycle of the basic pulse as 0/16 to 15/16 with a resolution of 1/16. The lower four bits specify how many extra pulses are to be added within the conversion period comprising 16 basic pulses. Thus, a specification of 0/256 to 255/256 is possible for 0/1 ratios within the conversion period. For 256/256 (100%) output, port output should be used. PWDR0 to PWDR15 are initialized to H'00.

9.3.3 PWM Data Polarity Registers A and B (PWDPRA, PWDPRB)

Each PWDPR selects the PWM output phase.

- PWDPRA

Bit	Bit Name	Initial Value	R/W	Description
7	OS7	0	R/W	Output Select 7 to 0
6	OS6	0	R/W	These bits select the PWM output phase. Bits OS7 to OS0 correspond to outputs PW7 to PW0.
5	OS5	0	R/W	
4	OS4	0	R/W	0: PWM direct output (PWDR value corresponds to high width of output)
3	OS3	0	R/W	
2	OS2	0	R/W	1: PWM inverted output (PWDR value corresponds to low width of output)
1	OS1	0	R/W	
0	OS0	0	R/W	

- PWDPRB

Bit	Bit Name	Initial Value	R/W	Description
7	OS15	0	R/W	Output Select 15 to 8
6	OS14	0	R/W	These bits select the PWM output phase. Bits OS15 to OS8 correspond to outputs PW15 to PW8.
5	OS13	0	R/W	
4	OS12	0	R/W	0: PWM direct output (PWDR value corresponds to high width of output)
3	OS11	0	R/W	
2	OS10	0	R/W	1: PWM inverted output (PWDR value corresponds to low width of output)
1	OS9	0	R/W	
0	OS8	0	R/W	

9.3.4 PWM Output Enable Registers A and B (PWOERA, PWOERB)

Each PWOER switches between PWM output and port output.

- PWOERA

Bit	Bit Name	Initial Value	R/W	Description
7	OE7	0	R/W	Output Enable 7 to 0
6	OE6	0	R/W	These bits, together with P1DDR, specify the P1n/PWn pin state. Bits OE7 to OE0 correspond to outputs PW7 to PW0.
5	OE5	0	R/W	
4	OE4	0	R/W	P1nDDR OEn: Pin state
3	OE3	0	R/W	0X: Port input
2	OE2	0	R/W	10: Port output or PWM 256/256 output
1	OE1	0	R/W	11: PWM output (0 to 255/256 output)
0	OE0	0	R/W	

[Legend]
X: Don't care

- PWOERB

Bit	Bit Name	Initial Value	R/W	Description
7	OE15	0	R/W	Output Enable 15 to 8
6	OE14	0	R/W	These bits, together with P2DDR, specify the P2n/PWn pin state. Bits OE15 to OE8 correspond to outputs PW15 to PW8.
5	OE13	0	R/W	
4	OE12	0	R/W	P2nDDR OEn: Pin state
3	OE11	0	R/W	0X: Port input
2	OE10	0	R/W	10: Port output or PWM 256/256 output
1	OE9	0	R/W	11: PWM output (0 to 255/256 output)
0	OE8	0	R/W	

[Legend]
X: Don't care

To perform PWM 256/256 output when DDR = 1 and OE = 0, the corresponding pin should be set to port output. The corresponding pin can be set as port output in single-chip mode (PW15 to PW0) or when IOSE = 1 (PW15 to PW12) in SYSCR in extended mode with on-chip ROM. Otherwise, it should be noted that an address bus is output to the corresponding pin.

DR data is output when the corresponding pin is used as port output. A value corresponding to PWM 256/256 output is determined by the OE bit, so the value should have been set to DR beforehand.

9.3.5 Peripheral Clock Select Register (PCSR)

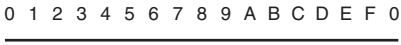
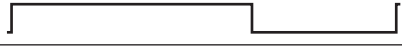
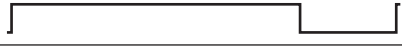
PCSR selects the PWM input clock.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved
6	—	0	R/W	The initial value should not be changed.
5	PWCKXB		R/W	PWMX Clock Select
4	PWCKXA		R/W	For details, see section 10.3.4, Peripheral Clock Select Register (PCSR).
3	—	All 0	R	Reserved The initial value should not be changed.
2	PWCKB	0	R/W	PWM Clock Select B, A
1	PWCKA	0	R/W	Together with bits PWCKE and PWCKS in PWSL, these bits select the internal clock input to the clock counter. For details, see table 9.2.
0	PWCKXC	0	R/W	PWMX Clock Select For details, see section 10.3.4, Peripheral Clock Select Register (PCSR).

9.4 Operation

The upper four bits of PWDR specify the duty cycle of the basic pulse as 0/16 to 15/16 with a resolution of 1/16. Table 9.4 shows the duty cycles of the basic pulse.

Table 9.4 Duty Cycle of Basic Pulse

Upper 4 Bits	Basic Pulse Waveform (Internal)
0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

The lower four bits of PWDR specify the position of pulses added to the 16 basic pulses. An additional pulse adds a high period (when OS = 0) with a width equal to the resolution before the rising edge of a basic pulse. When the upper four bits of PWDR are 0000, there is no rising edge of the basic pulse, but the timing for adding pulses is the same. Table 9.5 shows the positions of the additional pulses added to the basic pulses, and figure 9.2 shows an example of additional pulse timing.

Table 9.5 Position of Pulses Added to Basic Pulses

	Basic Pulse No.															
Lower 4 Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0000																
0001																Yes
0010								Yes								Yes
0011								Yes				Yes				Yes
0100				Yes				Yes				Yes				Yes
0101				Yes				Yes				Yes		Yes		Yes
0110				Yes		Yes		Yes				Yes		Yes		Yes
0111				Yes		Yes		Yes		Yes		Yes		Yes		Yes
1000		Yes		Yes		Yes		Yes		Yes		Yes		Yes		Yes
1001		Yes		Yes		Yes		Yes		Yes		Yes		Yes	Yes	Yes
1010		Yes		Yes		Yes	Yes	Yes		Yes		Yes		Yes	Yes	Yes
1011		Yes		Yes		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes
1100		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes
1101		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Yes	Yes
1110		Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Yes	Yes
1111		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

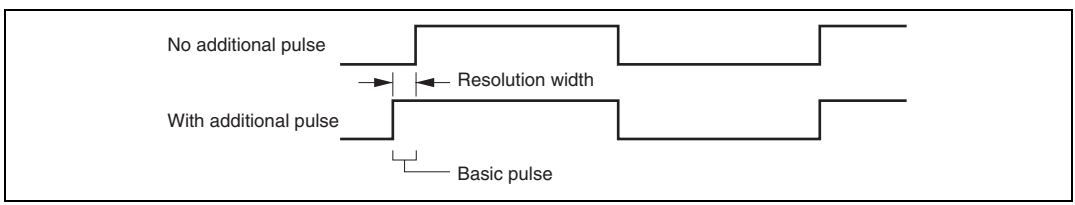


Figure 9.2 Example of Additional Pulse Timing (when Upper 4 Bits of PWDR = 1000)

9.4.1 PWM Setting Example (Pulse Division System)

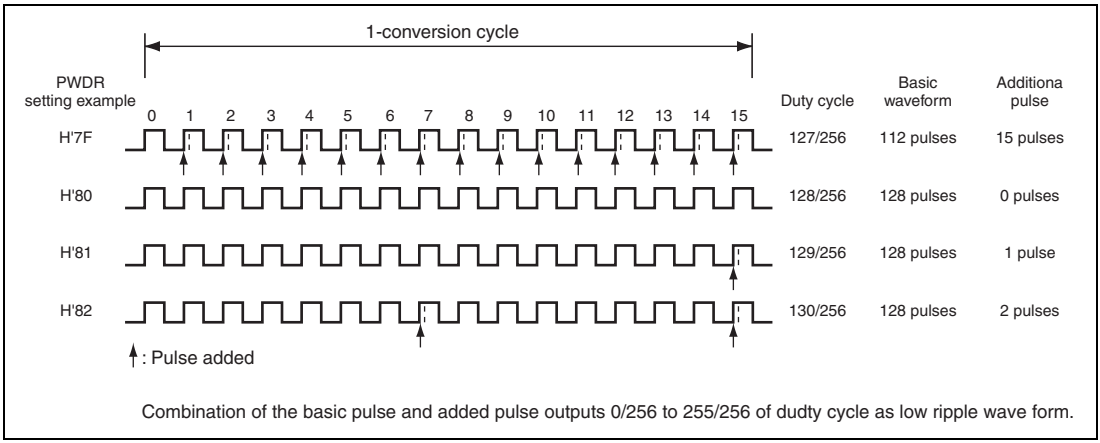


Figure 9.3 Example of PWM Setting

9.4.2 Diagram of PWM Used as D/A Converter

Figure 9.4 shows the diagram example when using the PWM pulse as the D/A converter. Analog signal with low ripple can be generated by connecting the low pass filter.

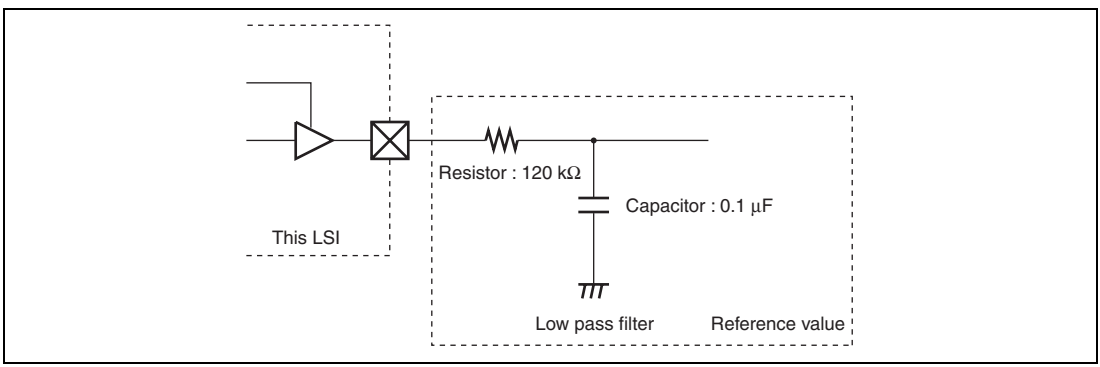


Figure 9.4 Example when PWM is Used as D/A Converter

9.5 Usage Note

9.5.1 Module Stop Mode Setting

PWM operation can be enabled or disabled using the module stop control register. The initial setting is for PWM operation to be halted. Register access is enabled by canceling the module stop mode. For details, see section 22, Power-Down Modes.

Section 10 14-Bit PWM Timer (PWMX)

This LSI has an on-chip 14-bit pulse-width modulator (PWM) timer with two output channels. It can be connected to an external low-pass filter to operate as a 14-bit D/A converter.

10.1 Features

- Division of pulse into multiple base cycles to reduce ripple
- Eight resolution settings
  The resolution can be set to 1, 2, 64, 128, 256, 1024, 4096, or 16384 system clock cycles.
- Two base cycle settings
  The base cycle can be set equal to T × 64 or T × 256, where T is the resolution.
- Sixteen operation clocks (by combination of eight resolution settings and two base cycle settings)

Figure 10.1 shows a block diagram of the PWM (D/A) module.

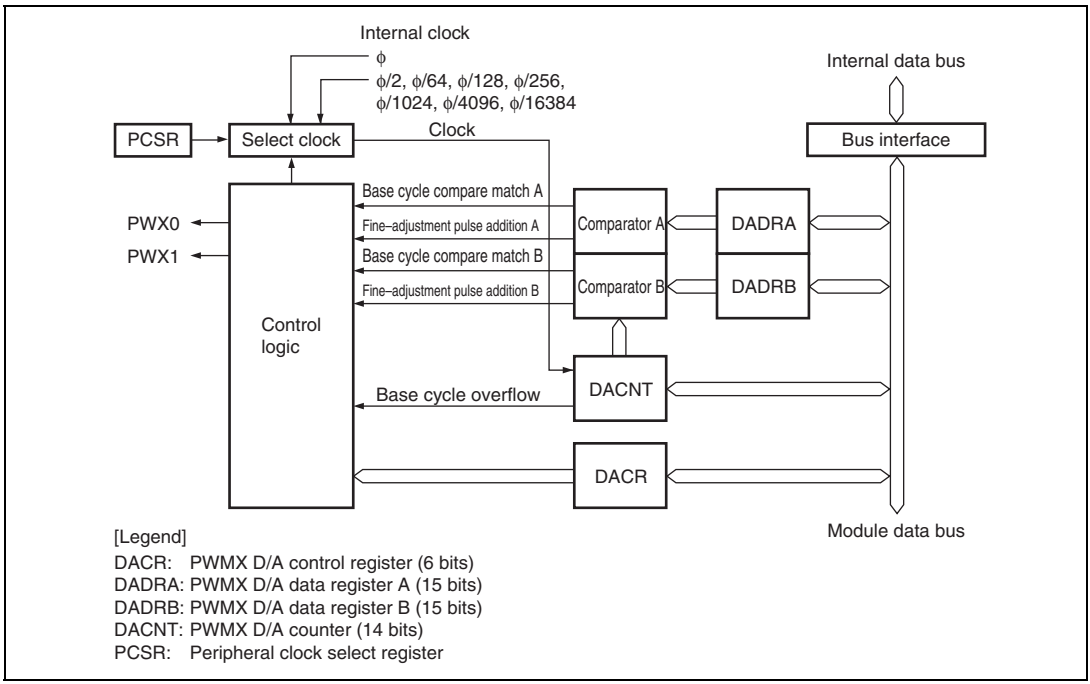


Figure 10.1 PWMX (D/A) Block Diagram

10.2 Input/Output Pins

Table 10.1 lists the PWMX (D/A) module input and output pins.

Table 10.1 Pin Configuration

Name	Abbreviation	I/O	Function
PWMX output pin 0	PWX0	Output	PWMX output of channel A
PWMX output pin 1	PWX1	Output	PWMX output of channel B

10.3 Register Descriptions

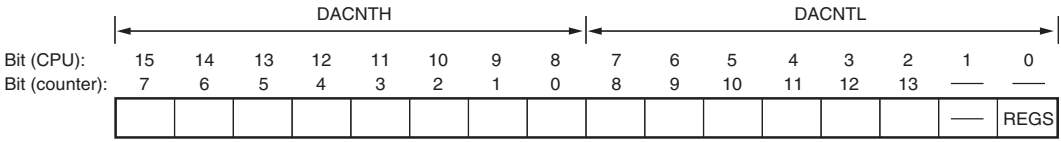
The PWMX (D/A) module has the following registers. The PWMX (D/A) registers are assigned to the same addresses with other registers. The registers are selected by the IICE bit in the serial timer control register (STCR). For details on the module stop control register, see section 22.1.3, Module Stop Control Register H, L, A, and B (MSTPCRH, MSTPCRL, MSTPCRA, and MSTPCRB).

- PWMX (D/A) counter (DACNT)
- PWMX (D/A) data register A (DADRA)
- PWMX (D/A) data register B (DADRB)
- PWMX (D/A) control register (DACR)
- Peripheral clock select register (PCSR)

Note: The same addresses are shared by DADRA and DACR, and by DADRB and DACNT. Switching is performed by the REGS bit in DACNT or DADRB.

10.3.1 PWMX (D/A) Counter (DACNT)

DACNT is a 14-bit readable/writable up-counter. The input clock is selected by the clock select bit (CKS) in DACR. DACNT functions as the time base for both PWMX (D/A) channels. When a channel operates with 14-bit precision, it uses all DACNT bits. When a channel operates with 12-bit precision, it uses the lower 12 bits and ignores the upper 2-bit counter. As DACNT is 16 bits, data transfer between the CPU is performed through the temporary register (TEMP). For details, see section 10.4, Bus Master Interface.



- DACNTH

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	DACNT7 to DACNT0	All 0	R/W	Upper Up-Counter

- DACNTL

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	DACNT 8 to DACNT 13	All 0	R/W	Lower Up-Counter
1	—	1	R	Reserved Always read as 1 and cannot be modified.
0	REGS	1	R/W	Register Select DADRA and DACR, and DADRB and DACNT, are located at the same addresses. The REGS bit specifies which registers can be accessed. 0: DADRA and DADRB can be accessed 1: DACR and DACNT can be accessed

10.3.2 PWMX (D/A) Data Registers A and B (DADRA and DADRB)

DADRA corresponds to PWMX (D/A) channel A, and DADRB to PWMX (D/A) channel B. As DACNT is 16 bits, data transfer between the CPU is performed through the temporary register (TEMP). For details, see section 10.4, Bus Master Interface.

- DADRA

Bit	Bit Name	Initial Value	R/W	Description
15	DA13	1	R/W	D/A Data 13 to 0 These bits set a digital value to be converted to an analog value. In each base cycle, the DACNT value is continually compared with the DADR value to determine the duty cycle of the output waveform, and to decide whether to output a fine-adjustment pulse equal in width to the resolution. To enable this operation, this register must be set within a range that depends on the CFS bit. If the DADR value is outside this range, the PWM output is held constant. A channel can be operated with 12-bit precision by fixing DA0 and DA1 to 0. The two data bits are not compared with DACNT12 and DACNT13 of DACNT.
14	DA12	1	R/W	
13	DA11	1	R/W	
12	DA10	1	R/W	
11	DA9	1	R/W	
10	DA8	1	R/W	
9	DA7	1	R/W	
8	DA6	1	R/W	
7	DA5	1	R/W	
6	DA4	1	R/W	
5	DA3	1	R/W	
4	DA2	1	R/W	
3	DA1	1	R/W	
2	DA0	1	R/W	
1	CFS	1	R/W	Carrier Frequency Select 0: Base cycle = resolution (T) × 64 The range of DA13 to DA0: H'0100 to H'3FFF 1: Base cycle = resolution (T) × 256 The range of DA13 to DA0: H'0040 to H'3FFF
0	—	1	R	Reserved Always read as 1 and cannot be modified.

- DADRB

Bit	Bit Name	Initial Value	R/W	Description
15	DA13	1	R/W	D/A Data 13 to 0 These bits set a digital value to be converted to an analog value. In each base cycle, the DACNT value is continually compared with the DADR value to determine the duty cycle of the output waveform, and to decide whether to output a fine-adjustment pulse equal in width to the resolution. To enable this operation, this register must be set within a range that depends on the CFS bit. If the DADR value is outside this range, the PWM output is held constant. A channel can be operated with 12-bit precision by fixing DA0 and DA1 to 0. The two data bits are not compared with DACNT12 and DACNT13 of DACNT.
14	DA12	1	R/W	
13	DA11	1	R/W	
12	DA10	1	R/W	
11	DA9	1	R/W	
10	DA8	1	R/W	
9	DA7	1	R/W	
8	DA6	1	R/W	
7	DA5	1	R/W	
6	DA4	1	R/W	
5	DA3	1	R/W	
4	DA2	1	R/W	
3	DA1	1	R/W	
2	DA0	1	R/W	
1	CFS	1	R/W	Carrier Frequency Select 0: Base cycle = resolution (T) × 64 DA13 to DA0 range = H'0100 to H'3FFF 1: Base cycle = resolution (T) × 256 DA13 to DA0 range = H'0040 to H'3FFF
0	REGS	1	R/W	Register Select DADRA and DACR, and DADRB and DACNT, are located at the same addresses. The REGS bit specifies which registers can be accessed. 0: DADRA and DADRB can be accessed 1: DACR and DACNT can be accessed

10.3.3 PWMX (D/A) Control Register (DACR)

DACR enables the PWM outputs, and selects the output phase and operating speed.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The initial value should not be changed.
6	PWME	0	R/W	PWMX Enable Starts or stops the PWM D/A counter (DACNT). 0: DACNT operates as a 14-bit up-counter 1: DACNT halts at H'0003
5	—	1	R	Reserved
4	—	1	R	Always read as 1 and cannot be modified.
3	OEB	0	R/W	Output Enable B Enables or disables output on PWMX (D/A) channel B. 0: PWMX (D/A) channel B output (at the PWX1 output pin) is disabled 1: PWMX (D/A) channel B output (at the PWX1 output pin) is enabled
2	OEA	0	R/W	Output Enable A Enables or disables output on PWMX (D/A) channel A. 0: PWMX (D/A) channel A output (at the PWX0 output pin) is disabled 1: PWMX (D/A) channel A output (at the PWX0 output pin) is enabled
1	OS	0	R/W	Output Select Selects the phase of the PWMX (D/A) output. 0: Direct PWMX (D/A) output 1: Inverted PWMX (D/A) output
0	CKS	0	R/W	Clock Select Selects the PWMX (D/A) resolution. Eight kinds of resolution can be selected. 0: Operates at resolution (T) = system clock cycle time (t_{cyc}) 1: Operates at resolution (T) = system clock cycle time (t_{cyc}) × 2, × 64, × 128, × 256, × 1024, × 4096, and × 16384.

10.3.4 Peripheral Clock Select Register (PCSR)

PCSR and the CKS bit of DACR select the operating speed.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved
6	—	0	R/W	The initial value should not be changed.
5	PWCKXB	0	R/W	PWMX clock select
4	PWCKXA	0	R/W	These bits select a clock cycle with the CKS bit of DACR of PWMX being 1. See table 10.2.
3	—	0	R/W	Reserved The initial value should not be changed.
2	PWCKB	0	R/W	PWM clock select B, A
1	PWCKA	0	R/W	See section 9.3.5, Peripheral Clock Select Register (PCSR).
0	PWCKXC	0	R/W	PWMX clock select This bit selects a clock cycle with the CKS bit of DACR of PWMX being 1. See table 10.2.

Table 10.2 Clock Select of PWMX

PWCKXC	PWCKXB	PWCKXA	Resolution (T)
0	0	0	Operates on the system clock cycle (t_{cyc}) x 2
0	0	1	Operates on the system clock cycle (t_{cyc}) x 64
0	1	0	Operates on the system clock cycle (t_{cyc}) x 128
0	1	1	Operates on the system clock cycle (t_{cyc}) x 256
1	0	0	Operates on the system clock cycle (t_{cyc}) x 1024
1	0	1	Operates on the system clock cycle (t_{cyc}) x 4096
1	1	0	Operates on the system clock cycle (t_{cyc}) x 16384
1	1	1	Setting prohibited

10.4 Bus Master Interface

DACNT, DADRA, and DADRB are 16-bit registers. The data bus linking the bus master and the on-chip peripheral modules, however, is only 8 bits wide. When the bus master accesses these registers, it therefore uses an 8-bit temporary register (TEMP).

These registers are written to and read from as follows.

- Write

  When the upper byte is written to, the upper-byte write data is stored in TEMP. Next, when the lower byte is written to, the lower-byte write data and TEMP value are combined, and the combined 16-bit value is written in the register.

- Read

  When the upper byte is read from, the upper-byte value is transferred to the CPU and the lower-byte value is transferred to TEMP. Next, when the lower byte is read from, the lower-byte value in TEMP is transferred to the CPU.

These registers should always be accessed 16 bits at a time with a MOV instruction, and the upper byte should always be accessed before the lower byte. Correct data will not be transferred if only the upper byte or only the lower byte is accessed. Also note that a bit manipulation instruction cannot be used to access these registers.

Example 1: Write to DACNT

```
MOV.W R0, @DACNT      ; Write R0 contents to DACNT
```

Example 2: Read DADRA

```
MOV.W @DADRA, R0      ; Copy contents of DADRA to R0
```

Table 10.3 Reading/Writing to 16-bit Registers

	Read		Write	
Register	**Word**	**Byte**	**Word**	**Byte**
DADRA, DADRB	O	O	O	×
DACNT	O	×	O	×

[Legend]

O: Enabled access.

Word-unit access includes accessing byte sequentially, first upper byte, and then lower byte.

×: The result of the access in the unit cannot be guaranteed.

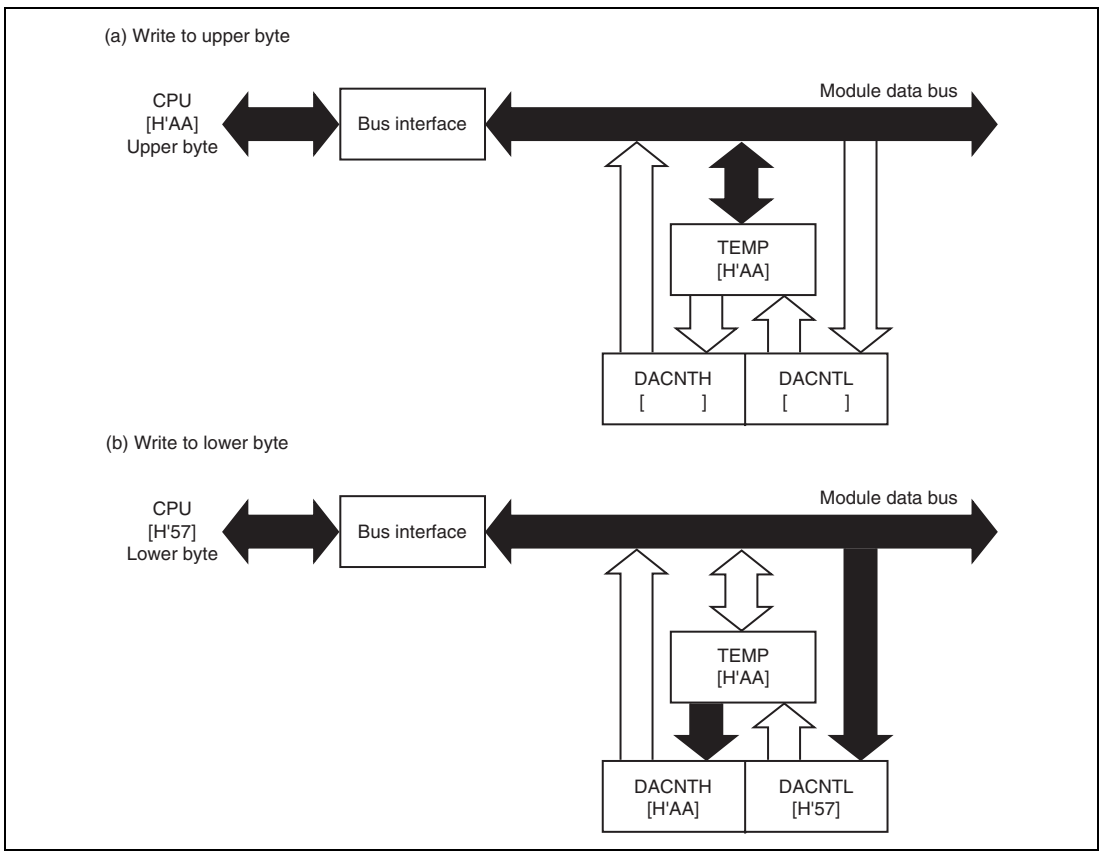


Figure 10.2 (1) DACNT Access Operation (1) [CPU → DACNT(H'AA57) Writing]

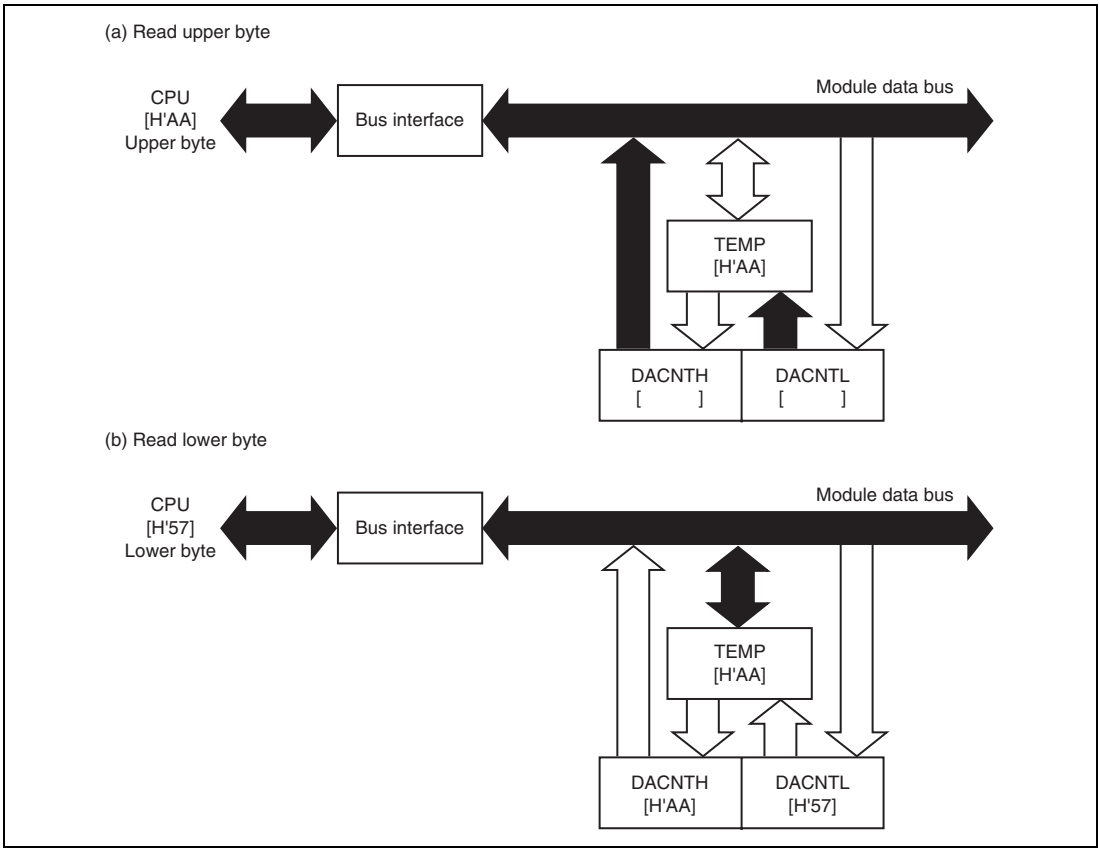


Figure 10.2 (2) DACNT Access Operation (2) [DACNT → CPU(H'AA57) Reading]

10.5 Operation

A PWM waveform like the one shown in figure 10.3 is output from the PWMX pin. DA13 to DA0 in DADR corresponds to the total width (T_L) of the low (0) pulses output in one conversion cycle (256 pulses when CFS = 0, 64 pulses when CFS = 1). When OS = 0, this waveform is directly output. When OS = 1, the output waveform is inverted, and DA13 to DA0 in DADR value corresponds to the total width (T_H) of the high (1) output pulses. Figures 10.4 and 10.5 show the types of waveform output available.

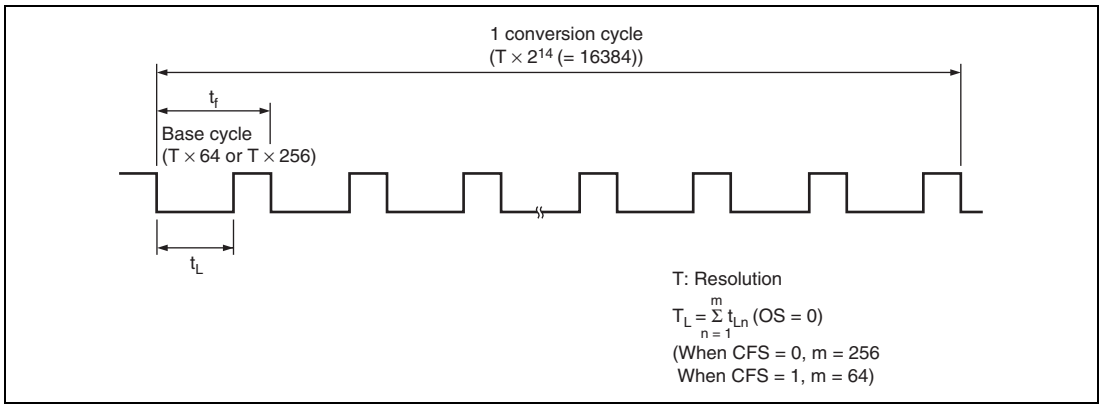


Figure 10.3 PWMX (D/A) Operation

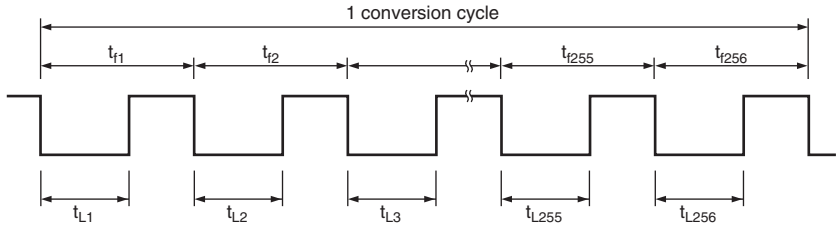
Table 10.4 summarizes the relationships between the CKS and CFS bit settings and the resolution, base cycle, and conversion cycle. The PWM output remains fixed unless DA13 to DA0 in DADR contain at least a certain minimum value. The relationship between the OS bit and the output waveform is shown in figures 10.4 and 10.5.

Table 10.4 Settings and Operation (Examples when ϕ = 20 MHz)

PCSR									Fixed DADR Bits				
PWCKX0 PWCKX1				Resolution			Conversion			Bit Data				
C	B	A	CK	T (μs)	CFS	Base Cycle	Cycle	TL/TH (OS = 0/OS = 1)	Accuracy (Bits)	DA3	DA2	DA1	DA0	Conversion Cycle*
—	—	—	0	0.05	0	3.2	819.2	Always low/high output	14					819.2 μs
						(μs)	(μs)	DA13 to 0 = H'0000 to H'00FF	12			0	0	204.8 μs
						/312.5kHz		(Data value) × T	10	0	0	0	0	51.2 μs
								DA13 to 0 = H'0100 to H'3FFF						
					1	12.8		Always low/high output	14					819.2 μs
						(μs)		DA13 to 0 = H'0000 to H'003F	12			0	0	204.8 μs
								(Data value) × T						
				(φ)		/78.1kHz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	51.2 μs
0	0	0	1	0.1	0	6.4	1.64	Always low/high output	14					1638.4 μs
						(μs)	(ms)	DA13 to 0 = H'0000 to H'00FF	12			0	0	409.6 μs
						/156.2kHz		(Data value) × T	10	0	0	0	0	102.4 μs
								DA13 to 0 = H'0100 to H'3FFF						
					1	25.6		Always low/high output	14					1638.4 μs
						(μs)		DA13 to 0 = H'0000 to H'003F	12			0	0	409.6 μs
								(Data value) × T						
				(φ/2)		/39.1kHz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	102.4 μs
0	0	1	1	3.2	0	204.8	52.4	Always low/high output	14					52.4 ms
						(μs)	(ms)	DA13 to 0 = H'0000 to H'00FF	12			0	0	13.1 ms
						/4.9kHz		(Data value) × T	10	0	0	0	0	3.3 ms
								DA13 to 0 = H'0100 to H'3FFF						
					1	819.2		Always low/high output	14					52.4 ms
						(μs)		DA13 to 0 = H'0000 to H'003F	12			0	0	13.1 ms
								(Data value) × T						
				(φ/64)		/1.2kHz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	3.3 ms
0	1	0	1	6.4	0	409.6	104.9	Always low/high output	14					104.9 ms
						(μs)	(ms)	DA13 to 0 = H'0000 to H'00FF	12			0	0	26.2 ms
						/2.4kHz		(Data value) × T	10	0	0	0	0	6.6 ms
								DA13 to 0 = H'0100 to H'3FFF						
					1	1638.4		Always low/high output	14					104.9 ms
						(μs)		DA13 to 0 = H'0000 to H'003F	12			0	0	26.2 ms
								(Data value) × T						
				(φ/128)		/610.4kHz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	6.6 ms

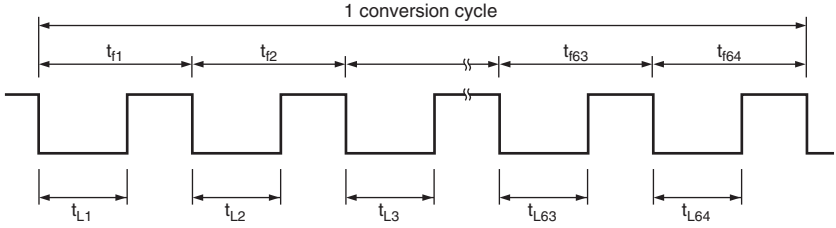
PCSR								Fixed DADR Bits					
PWCKX0 PWCKX1				Resolution T			Conversion		Accuracy	Bit Data				Conversion
C	B	A	CKS	(µs)	CFS	Base Cycle	Cycle	TL/TH (OS = 0/OS = 1)	(Bits)	DA3	DA2	DA1	DA0	Cycle*
0	1	1	1	12.8	0	819.2	209.7	Always low/high output	14					209.7 ms
						(µs)	(ms)	DA13 to 0 = H'0000 to H'00FF	12			0	0	52.4 ms
						/1.2kHz		(Data value) × T	10	0	0	0	0	13.1 ms
								DA13 to 0 = H'0100 to H'3FFF						
					1	3276.8		Always low/high output	14					209.7 ms
						(µs)		DA13 to 0 = H'0000 to H'003F	12			0	0	52.4 ms
				(ϕ/256)		/305.2kHz		(Data value) × T	10	0	0	0	0	13.1 ms
								DA13 to 0 = H'0040 to H'3FFF						
1	0	0	1	51.2	0	3.3	838.9	Always low/high output	14					838.9 ms
						(ms)	(ms)	DA13 to 0 = H'0000 to H'00FF	12			0	0	209.7 ms
						/305.2Hz		(Data value) × T	10	0	0	0	0	52.4 ms
								DA13 to 0 = H'0100 to H'3FFF						
					1	13.1		Always low/high output	14					838.9 ms
						(ms)		DA13 to 0 = H'0000 to H'003F	12			0	0	209.7 ms
				(ϕ/1024)		/76.3Hz		(Data value) × T	10	0	0	0	0	52.4 ms
								DA13 to 0 = H'0040 to H'3FFF						
1	0	1	1	204.8	0	13.1	2.03	Always low/high output	14					3.4 s
						(ms)	(s)	DA13 to 0 = H'0000 to H'00FF	12			0	0	838.9 ms
						/76.3Hz		(Data value) × T	10	0	0	0	0	209.7 ms
								DA13 to 0 = H'0100 to H'3FFF						
					1	52.4		Always low/high output	14					3.4 s
						(ms)		DA13 to 0 = H'0000 to H'003F	12			0	0	838.9 ms
				(ϕ/4096)		/19.1Hz		(Data value) × T	10	0	0	0	0	209.7 ms
								DA13 to 0 = H'0040 to H'3FFF						
1	1	0	1	496.48	0	52.4	8.13	Always low/high output	14					13.4 s
						(ms)	(s)	DA13 to 0 = H'0000 to H'00FF	12			0	0	3.4 s
						/19.1Hz		(Data value) × T	10	0	0	0	0	838.9 ms
								DA13 to 0 = H'0100 to H'3FFF						
					1	209.7		Always low/high output	14					13.4 s
						(ms)		DA13 to 0 = H'0000 to H'003F	12			0	0	3.4 s
				(ϕ/16384)		/4.8Hz		(Data value) × T	10	0	0	0	0	838.9 ms
								DA13 to 0 = H'0040 to H'3FFF						
1	1	1	1	Setting prohibited	—	—	—	—	—	—	—	—	—	—

Note: * Indicates the conversion cycle when specific DA3 to DA0 bits are fixed.



$t_{f1} = t_{f2} = t_{f3} = \cdots = t_{f255} = t_{f256} = T \times 64$
$t_{L1} + t_{L2} + t_{L3} + \cdots + t_{L255} + t_{L256} = T_L$

a. CFS = 0 [base cycle = resolution (T) × 64]



$t_{f1} = t_{f2} = t_{f3} = \cdots = t_{f63} = t_{f64} = T \times 256$
$t_{L1} + t_{L2} + t_{L3} + \cdots + t_{L63} + t_{L64} = T_L$

b. CFS = 1 [base cycle = resolution (T) × 256]

Figure 10.4 Output Waveform (OS = 0, DADR corresponds to T_L)

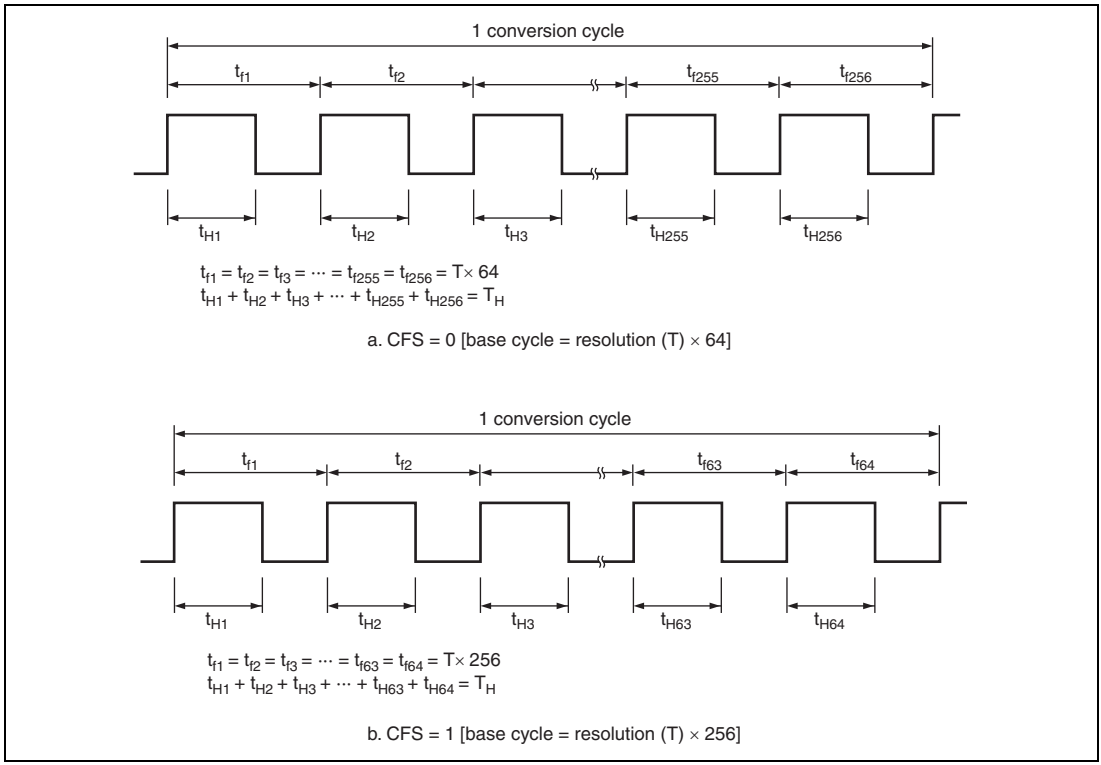


Figure 10.5 Output Waveform (OS = 1, DADR corresponds to T_H)

An example of the additional pulses when CFS = 1 (base cycle = resolution (T) × 256) and OS = 1 (inverted PWM output) is described below. When CFS = 1, the upper eight bits (DA13 to DA6) in DADR determine the duty cycle of the base pulse while the subsequent six bits (DA5 to DA0) determine the locations of the additional pulses as shown in figure 10.6.

Table 10.5 lists the locations of the additional pulses.

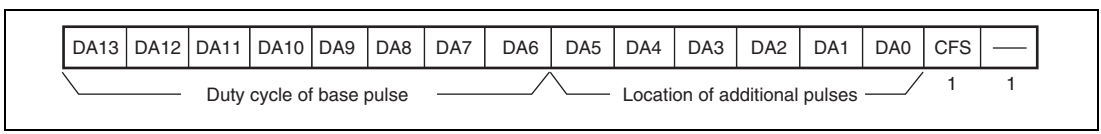


Figure 10.6 D/A Data Register Configuration when CFS = 1

In this example, DADR = H'0207 (B'0000 0010 0000 0111). The output waveform is shown in figure 10.7. Since CFS = 1 and the value of the upper eight bits is B'0000 0010, the high width of the base pulse duty cycle is 2/256 × (T).

Since the value of the subsequent six bits is B'0000 01, an additional pulse is output only at the location of base pulse No. 63 according to table 10.5. Thus, an additional pulse of 1/256 × (T) is to be added to the base pulse.

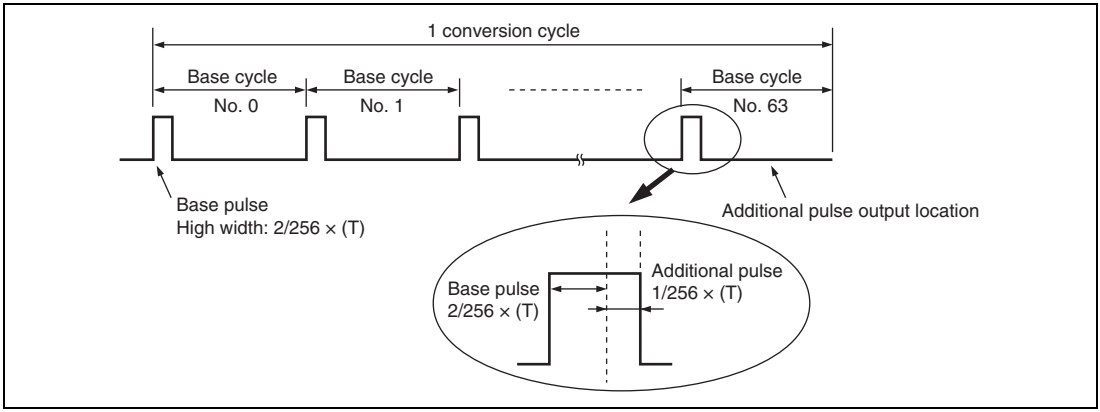
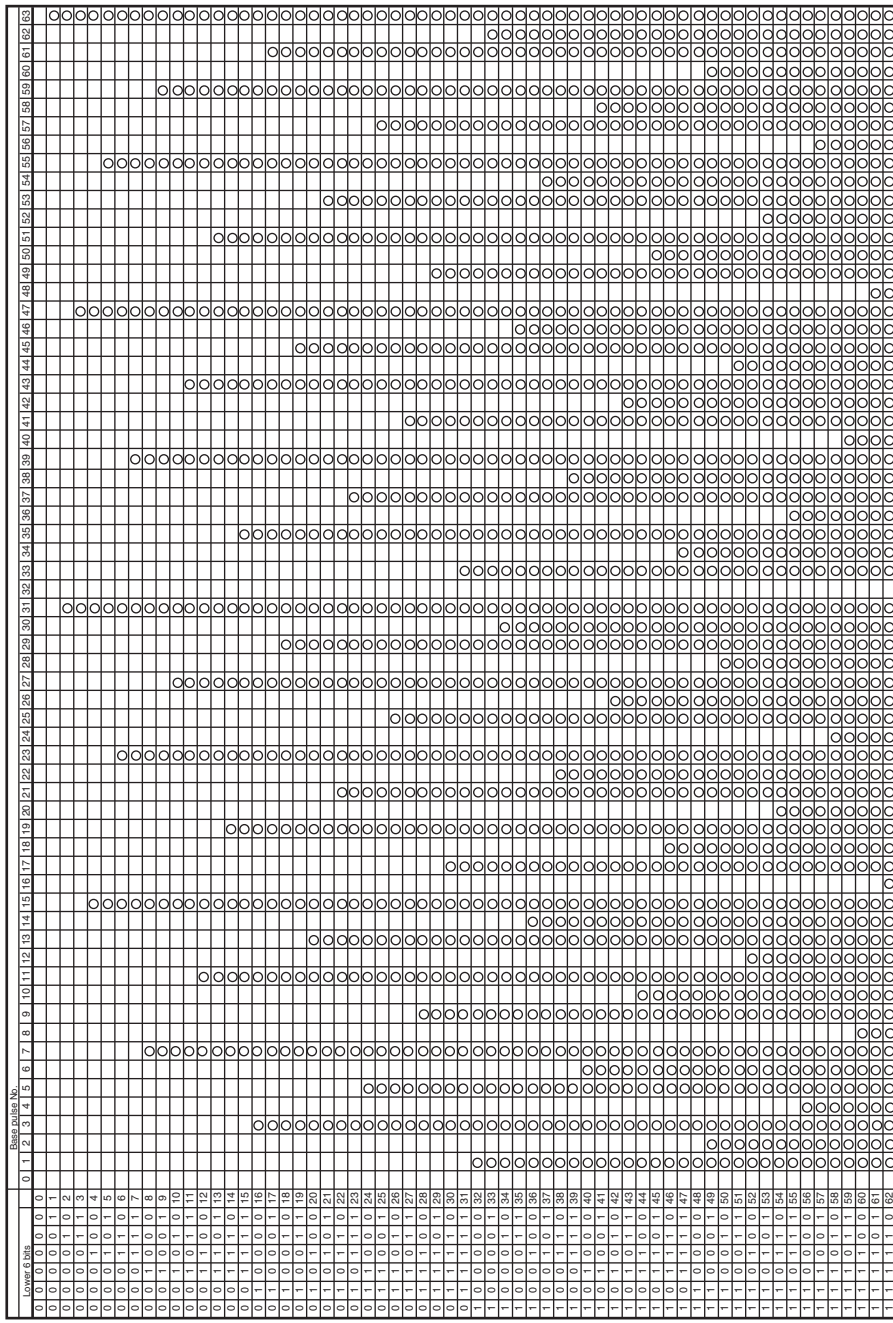


Figure 10.7 Output Waveform when DADR = H'0207 (OS = 1)

However, when CFS = 0 (base cycle = resolution (T) × 64), the duty cycle of the base pulse is determined by the upper six bits and the locations of the additional pulses by the subsequent eight bits with a method similar to as above.

Table 10.5 Locations of Additional Pulses Added to Base Pulse (When CFS = 1)



10.6 Usage Notes

10.6.1 Module Stop Mode Setting

PWMX operation can be enabled or disabled by using the module stop control register. In the initial state, PWMX operation is disabled. Register access is enabled by clearing module stop mode. For details, see section 22, Power-Down Modes.

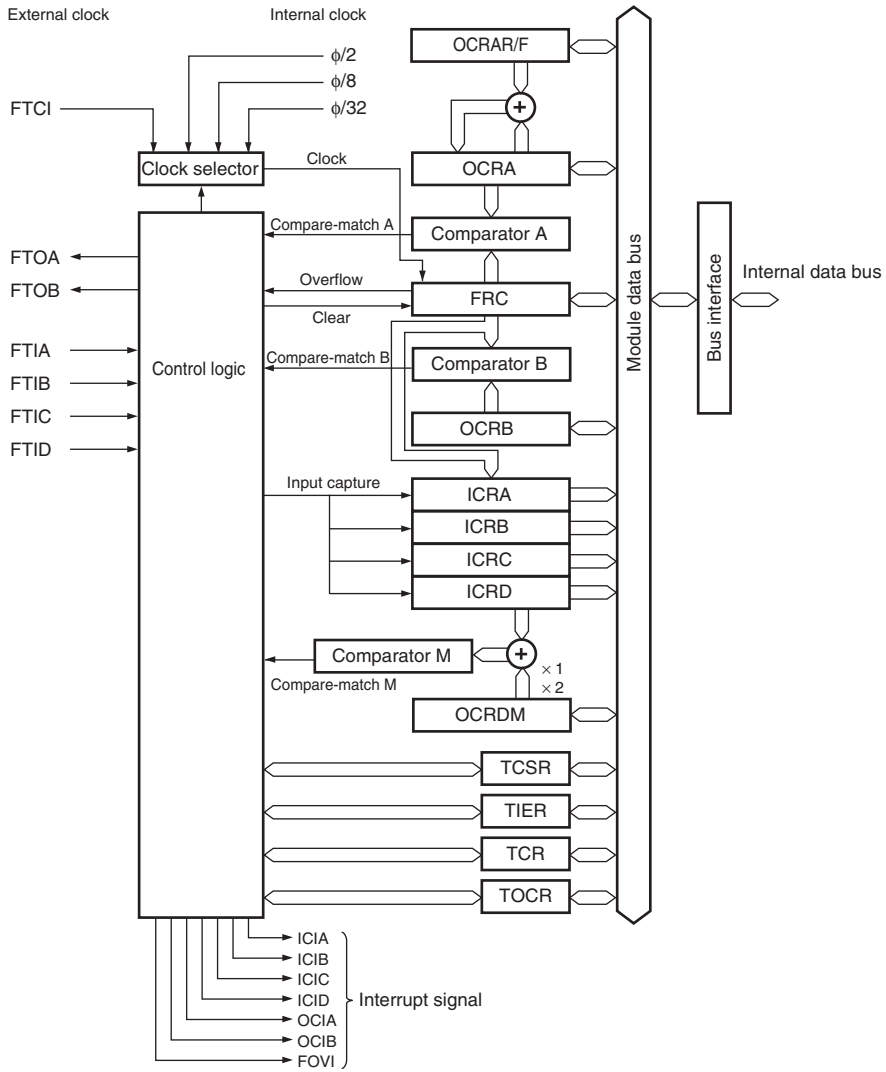
Section 11 16-Bit Free-Running Timer (FRT)

This LSI has an on-chip 16-bit free-running timer (FRT). The FRT operates on the basis of the 16-bit free-running counter (FRC), and outputs two independent waveforms, and measures the input pulse width and external clock periods.

11.1 Features

- Selection of four clock sources
  - One of the three internal clocks (φ/2, φ/8, or φ/32), or an external clock input can be selected (enabling use as an external event counter).
- Two independent comparators
  - Two independent waveforms can be output.
- Four independent input capture channels
  - The rising or falling edge can be selected.
  - Buffer modes can be specified.
- Counter clearing
  - The free-running counters can be cleared on compare-match A.
- Seven independent interrupts
  - Two compare-match interrupts, four input capture interrupts, and one overflow interrupt can be requested independently.
- Special functions provided by automatic addition function
  - The contents of OCRAR and OCRAF can be added to the contents of OCRA automatically, enabling a periodic waveform to be generated without software intervention. The contents of ICRD can be added automatically to the contents of OCRDM × 2, enabling input capture operations in this interval to be restricted.

Figure 11.1 shows a block diagram of the FRT.



[Legend]

OCRA, OCRB:	Output compare register A, B (16-bit)
OCRAR,OCRAF:	Output compare register AR, AF (16-bit)
OCRDM:	Output compare register DM (16-bit)
FRC:	Free-running counter (16-bit)
ICRA to ICRD:	Input capture registers A to D (16-bit)
TCSR:	Timer control/status register (8-bit)
TIER:	Timer interrupt enable register (8-bit)
TCR:	Timer control register (8-bit)
TOCR:	Timer output compare control register (8-bit)

Figure 11.1 Block Diagram of 16-Bit Free-Running Timer

11.2 Input/Output Pins

Table 11.1 lists the FRT input and output pins.

Table 11.1 Pin Configuration

Name	Abbreviation	I/O	Function
Counter clock input pin	FTCI	Input	FRC counter clock input
Output compare A output pin	FTOA	Output	Output compare A output
Output compare B output pin	FTOB	Output	Output compare B output
Input capture A input pin	FTIA	Input	Input capture A input
Input capture B input pin	FTIB	Input	Input capture B input
Input capture C input pin	FTIC	Input	Input capture C input
Input capture D input pin	FTID	Input	Input capture D input

11.3 Register Descriptions

The FRT has the following registers.

- Free-running counter (FRC)
- Output compare register A (OCRA)
- Output compare register B (OCRB)
- Input capture register A (ICRA)
- Input capture register B (ICRB)
- Input capture register C (ICRC)
- Input capture register D (ICRD)
- Output compare register AR (OCRAR)
- Output compare register AF (OCRAF)
- Output compare register DM (OCRDM)
- Timer interrupt enable register (TIER)
- Timer control/status register (TCSR)
- Timer control register (TCR)
- Timer output compare control register (TOCR)

Note: OCRA and OCRB share the same address. Register selection is controlled by the OCRS bit in TOCR. ICRA, ICRB, and ICRC share the same addresses with OCRAR, OCRAF, and OCRDM. Register selection is controlled by the ICRS bit in TOCR.

11.3.1 Free-Running Counter (FRC)

FRC is a 16-bit readable/writable up-counter. The clock source is selected by bits CKS1 and CKS0 in TCR. FRC can be cleared by compare-match A. When FRC overflows from H'FFFF to H'0000, the overflow flag bit (OVF) in TCSR is set to 1. FRC should always be accessed in 16-bit units; cannot be accessed in 8-bit units. FRC is initialized to H'0000.

11.3.2 Output Compare Registers A and B (OCRA and OCRB)

The FRT has two output compare registers, OCRA and OCRB, each of which is a 16-bit readable/writable register whose contents are continually compared with the value in FRC. When a match is detected (compare-match), the corresponding output compare flag (OCFA or OCFB) is set to 1 in TCSR. If the OEA or OEB bit in TOCR is set to 1, when the OCR and FRC values match, the output level selected by the OLVLA or OLVLB bit in TOCR is output at the output compare output pin (FTOA or FTOB). Following a reset, the FTOA and FTOB output levels are 0 until the first compare-match. OCR should always be accessed in 16-bit units; cannot be accessed in 8-bit units. OCR is initialized to H'FFFF.

11.3.3 Input Capture Registers A to D (ICRA to ICRD)

The FRT has four input capture registers, ICRA to ICRD, each of which is a 16-bit read-only register. When the rising or falling edge of the signal at an input capture input pin (FTIA to FTID) is detected, the current FRC value is transferred to the corresponding input capture register (ICRA to ICRD). At the same time, the corresponding input capture flag (ICFA to ICFD) in TCSR is set to 1. The FRC contents are transferred to ICR regardless of the value of ICF. The input capture edge is selected by the input edge select bits (IEDGA to IEDGD) in TCR.

ICRC and ICRD can be used as ICRA and ICRB buffer registers, respectively, by means of buffer enable bits A and B (BUFEA and BUFEB) in TCR. For example, if an input capture occurs when ICRC is specified as the ICRA buffer register, the FRC contents are transferred to ICRA, and then transferred to the buffer register ICRC. When IEDGA and IEDGC bits in TCR are set to different values, both rising and falling edges can be specified as the change of the external input signal. When IEDGA and IEDGC are set to the same value, either rising edge or falling edge can be specified as the change of the external input signal.

To ensure input capture, the input capture pulse width should be at least 1.5 system clocks (ϕ) for a single edge. When triggering is enabled on both edges, the input capture pulse width should be at least 2.5 system clocks (ϕ).

ICRA to ICRD should always be accessed in 16-bit units; cannot be accessed in 8-bit units. ICR is initialized to H'0000.

11.3.4 Output Compare Registers AR and AF (OCRAR and OCRAF)

OCRAR and OCRAF are 16-bit readable/writable registers. When the OCRAMS bit in TOCR is set to 1, the operation of OCRA is changed to include the use of OCRAR and OCRAF. The contents of OCRAR and OCRAF are automatically added alternately to OCRA, and the result is written to OCRA. The write operation is performed on the occurrence of compare-match A. In the 1st compare-match A after setting the OCRAMS bit to 1, OCRAF is added. The operation due to compare-match A varies according to whether the compare-match follows addition of OCRAR or OCRAF. The value of the OLVLA bit in TOCR is ignored, and 1 is output on a compare-match A following addition of OCRAF, while 0 is output on a compare-match A following addition of OCRAR.

When using the OCRA automatic addition function, do not select internal clock $\phi/2$ as the FRC input clock together with a set value of H'0001 or less for OCRAR (or OCRAF).

OCRAR and OCRAF should always be accessed in 16-bit units; cannot be accessed in 8-bit units. OCRAR and OCRAF are initialized to H'FFFF.

11.3.5 Output Compare Register DM (OCRDM)

OCRDM is a 16-bit readable/writable register in which the upper eight bits are fixed at H'00. When the ICRDMS bit in TOCR is set to 1 and the contents of OCRDM are other than H'0000, the operation of ICRD is changed to include the use of OCRDM. The point at which input capture D occurs is taken as the start of a mask interval. Next, twice the contents of OCRDM is added to the contents of ICRD, and the result is compared with the FRC value. The point at which the values match is taken as the end of the mask interval. New input capture D events are disabled during the mask interval. A mask interval is not generated when the contents of OCRDM are H'0000 while the ICRDMS bit is set to 1.

OCRDM should always be accessed in 16-bit units; cannot be accessed in 8-bit units. OCRDM is initialized to H'0000.

11.3.6 Timer Interrupt Enable Register (TIER)

TIER enables and disables interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	ICIAE	0	R/W	Input Capture Interrupt A Enable Selects whether to enable input capture interrupt A request (ICIA) when input capture flag A (ICFA) in TCSR is set to 1. 0: ICIA requested by ICFA is disabled 1: ICIA requested by ICFA is enabled
6	ICIBE	0	R/W	Input Capture Interrupt B Enable Selects whether to enable input capture interrupt B request (ICIB) when input capture flag B (ICFB) in TCSR is set to 1. 0: ICIB requested by ICFB is disabled 1: ICIB requested by ICFB is enabled
5	ICICE	0	R/W	Input Capture Interrupt C Enable Selects whether to enable input capture interrupt C request (ICIC) when input capture flag C (ICFC) in TCSR is set to 1. 0: ICIC requested by ICFC is disabled 1: ICIC requested by ICFC is enabled
4	ICIDE	0	R/W	Input Capture Interrupt D Enable Selects whether to enable input capture interrupt D request (ICID) when input capture flag D (ICFD) in TCSR is set to 1. 0: ICID requested by ICFD is disabled 1: ICID requested by ICFD is enabled
3	OCIAE	0	R/W	Output Compare Interrupt A Enable Selects whether to enable output compare interrupt A request (OCIA) when output compare flag A (OCFA) in TCSR is set to 1. 0: OCIA requested by OCFA is disabled 1: OCIA requested by OCFA is enabled

Bit	Bit Name	Initial Value	R/W	Description
2	OCIBE	0	R/W	Output Compare Interrupt B Enable Selects whether to enable output compare interrupt B request (OCIB) when output compare flag B (OCFB) in TCSR is set to 1. 0: OCIB requested by OCFB is disabled 1: OCIB requested by OCFB is enabled
1	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects whether to enable a free-running timer overflow request interrupt (FOVI) when the timer overflow flag (OVF) in TCSR is set to 1. 0: FOVI requested by OVF is disabled 1: FOVI requested by OVF is enabled
0	—	1	R	Reserved This bit is always read as 1 and cannot be modified.

11.3.7 Timer Control/Status Register (TCSR)

TCSR is used for counter clear selection and control of interrupt request signals.

Bit	Bit Name	Initial Value	R/W	Description
7	ICFA	0	R/(W)*	Input Capture Flag A This status flag indicates that the FRC value has been transferred to ICRA by means of an input capture signal. When BUFEA = 1, ICFA indicates that the old ICRA value has been moved into ICRC and the new FRC value has been transferred to ICRA. [Setting condition] When an input capture signal causes the FRC value to be transferred to ICRA [Clearing condition] Read ICFA when ICFA = 1, then write 0 to ICFA

Bit	Bit Name	Initial Value	R/W	Description
6	ICFB	0	R/(W)*	Input Capture Flag B This status flag indicates that the FRC value has been transferred to ICRB by means of an input capture signal. When BUFEB = 1, ICFB indicates that the old ICRB value has been moved into ICRD and the new FRC value has been transferred to ICRB. [Setting condition] When an input capture signal causes the FRC value to be transferred to ICRB [Clearing condition] Read ICFB when ICFB = 1, then write 0 to ICFB
5	ICFC	0	R/(W)*	Input Capture Flag C This status flag indicates that the FRC value has been transferred to ICRC by means of an input capture signal. When BUFEA = 1, on occurrence of an input capture signal specified by the IEDGC bit at the FTIC input pin, ICFC is set but data is not transferred to ICRC. In buffer operation, ICFC can be used as an external interrupt signal by setting the ICICE bit to 1. [Setting condition] When an input capture signal is received [Clearing condition] Read ICFC when ICFC = 1, then write 0 to ICFC
4	ICFD	0	R/(W)*	Input Capture Flag D This status flag indicates that the FRC value has been transferred to ICRD by means of an input capture signal. When BUFEB = 1, on occurrence of an input capture signal specified by the IEDGD bit at the FTID input pin, ICFD is set but data is not transferred to ICRD. In buffer operation, ICFD can be used as an external interrupt signal by setting the ICIDE bit to 1. [Setting condition] When an input capture signal is received [Clearing condition] Read ICFD when ICFD = 1, then write 0 to ICFD

Bit	Bit Name	Initial Value	R/W	Description
3	OCFA	0	R/(W)*	Output Compare Flag A This status flag indicates that the FRC value matches the OCRA value. [Setting condition] When FRC = OCRA [Clearing condition] Read OCFA when OCFA = 1, then write 0 to OCFA
2	OCFB	0	R/(W)*	Output Compare Flag B This status flag indicates that the FRC value matches the OCRB value. [Setting condition] When FRC = OCRB [Clearing condition] Read OCFB when OCFB = 1, then write 0 to OCFB
1	OVF	0	R/(W)*	Overflow Flag This status flag indicates that the FRC has overflowed. [Setting condition] When FRC overflows (changes from H'FFFF to H'0000) [Clearing condition] Read OVF when OVF = 1, then write 0 to OVF
0	CCLRA	0	R/W	Counter Clear A This bit selects whether the FRC is to be cleared at compare-match A (when the FRC and OCRA values match). 0: FRC clearing is disabled 1: FRC is cleared at compare-match A

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

11.3.8 Timer Control Register (TCR)

TCR selects the rising or falling edge of the input capture signals, enables the input capture buffer mode, and selects the FRC clock source.

Bit	Bit Name	Initial Value	R/W	Description
7	IEDGA	0	R/W	Input Edge Select A Selects the rising or falling edge of the input capture A signal (FTIA). 0: Capture on the falling edge of FTIA 1: Capture on the rising edge of FTIA
6	IEDGB	0	R/W	Input Edge Select B Selects the rising or falling edge of the input capture B signal (FTIB). 0: Capture on the falling edge of FTIB 1: Capture on the rising edge of FTIB
5	IEDGC	0	R/W	Input Edge Select C Selects the rising or falling edge of the input capture C signal (FTIC). 0: Capture on the falling edge of FTIC 1: Capture on the rising edge of FTIC
4	IEDGD	0	R/W	Input Edge Select D Selects the rising or falling edge of the input capture D signal (FTID). 0: Capture on the falling edge of FTID 1: Capture on the rising edge of FTID
3	BUFEA	0	R/W	Buffer Enable A Selects whether ICRC is to be used as a buffer register for ICRA. 0: ICRC is not used as a buffer register for ICRA 1: ICRC is used as a buffer register for ICRA
2	BUFEB	0	R/W	Buffer Enable B Selects whether ICRD is to be used as a buffer register for ICRB. 0: ICRD is not used as a buffer register for ICRB 1: ICRD is used as a buffer register for ICRB

Bit	Bit Name	Initial Value	R/W	Description
1	CKS1	0	R/W	Clock Select 1, 0
0	CKS0	0		Select clock source for FRC.
				00: ϕ/2 internal clock source
				01: ϕ/8 internal clock source
				10: ϕ/32 internal clock source
				11: External clock source (counting at FTCI rising edge)

11.3.9 Timer Output Compare Control Register (TOCR)

TOCR enables output from the output compare pins, selects the output levels, switches access between output compare registers A and B, controls the ICRD and OCRA operating modes, and switches access to input capture registers A, B, and C.

Bit	Bit Name	Initial Value	R/W	Description
7	ICRDMS	0	R/W	Input Capture D Mode Select
				Specifies whether ICRD is used in the normal operating mode or in the operating mode using OCRDM.
				0: The normal operating mode is specified for ICRD
				1: The operating mode using OCRDM is specified for ICRD
6	OCRAMS	0	R/W	Output Compare A Mode Select
				Specifies whether OCRA is used in the normal operating mode or in the operating mode using OCRAR and OCRAF.
				0: The normal operating mode is specified for OCRA
				1: The operating mode using OCRAR and OCRAF is specified for OCRA
5	ICRS	0	R/W	Input Capture Register Select
				The same addresses are shared by ICRA and OCRAR, by ICRB and OCRAF, and by ICRC and OCRDM. The ICRS bit determines which registers are selected when the shared addresses are read from or written to. The operation of ICRA, ICRB, and ICRC is not affected.
				0: ICRA, ICRB, and ICRC are selected
				1: OCRAR, OCRAF, and OCRDM are selected

Bit	Bit Name	Initial Value	R/W	Description
4	OCRS	0	R/W	Output Compare Register Select OCRA and OCRB share the same address. The OCRS determines which register is selected when the shared address is read from or written to. The operation of OCRA or OCRB is not affected. 0: OCRA is selected 1: OCRB is selected
3	OEA	0	R/W	Output Enable A Enables or disables output of the output compare A output pin (FTOA). 0: Output compare A output is disabled 1: Output compare A output is enabled
2	OEB	0	R/W	Output Enable B Enables or disables output of the output compare B output pin (FTOB). 0: Output compare B output is disabled 1: Output compare B output is enabled
1	OLVLA	0	R/W	Output Level A Selects the level to be output at the output compare A output pin (FTOA) in response to compare-match A (signal indicating a match between the FRC and OCRA values). When the OCRAMS bit is 1, this bit is ignored. 0: 0 is output at compare-match A 1: 1 is output at compare-match A
0	OLVLB	0	R/W	Output Level B Selects the level to be output at the output compare B output pin (FTOB) in response to compare-match B (signal indicating a match between the FRC and OCRB values). 0: 0 is output at compare-match B 1: 1 is output at compare-match B

11.4 Operation

11.4.1 Pulse Output

Figure 11.2 shows an example of 50%-duty pulses output with an arbitrary phase difference. When a compare match occurs while the CCLRA bit in TCSR is set to 1, the OLVLA and OLVLB bits are inverted by software.

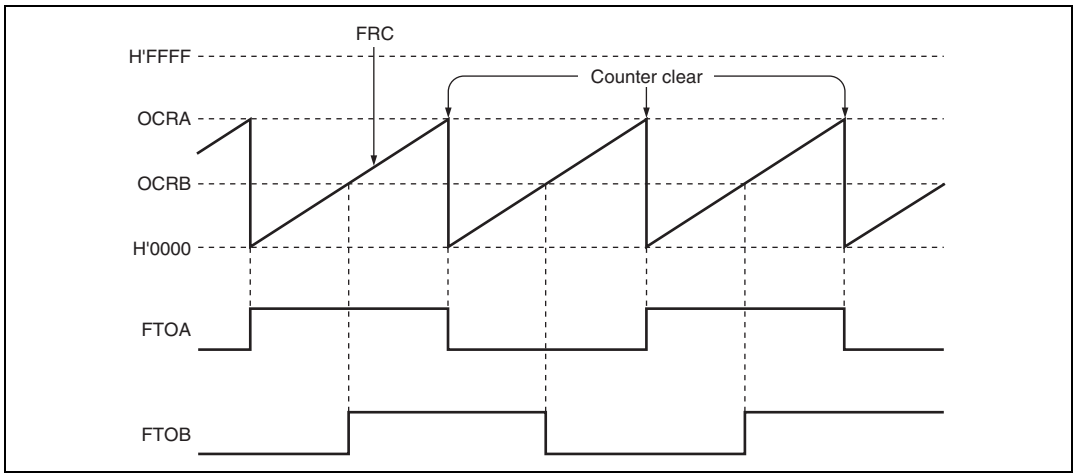


Figure 11.2 Example of Pulse Output

11.5 Operation Timing

11.5.1 FRC Increment Timing

Figure 11.3 shows the FRC increment timing with an internal clock source. Figure 11.4 shows the increment timing with an external clock source. The pulse width of the external clock signal must be at least 1.5 system clocks (ϕ). The counter will not increment correctly if the pulse width is shorter than 1.5 system clocks (ϕ).

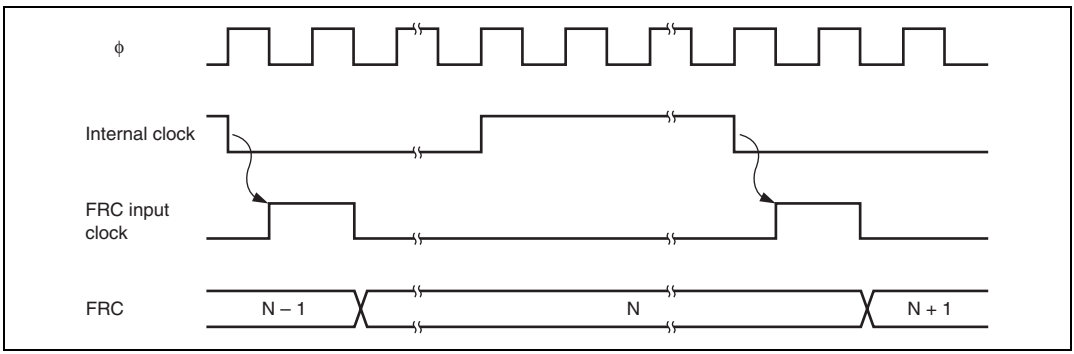


Figure 11.3 Increment Timing with Internal Clock Source

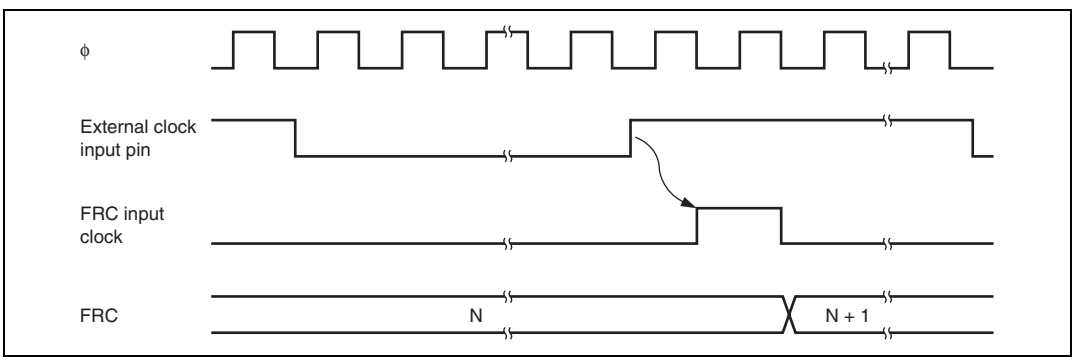


Figure 11.4 Increment Timing with External Clock Source

11.5.2 Output Compare Output Timing

A compare-match signal occurs at the last state when the FRC and OCR values match (at the timing when the FRC updates the counter value). When a compare-match signal occurs, the level selected by the OLVL bit in TOCR is output at the output compare output pin (FTOA or FTOB). Figure 11.5 shows the timing of this operation for compare-match A.

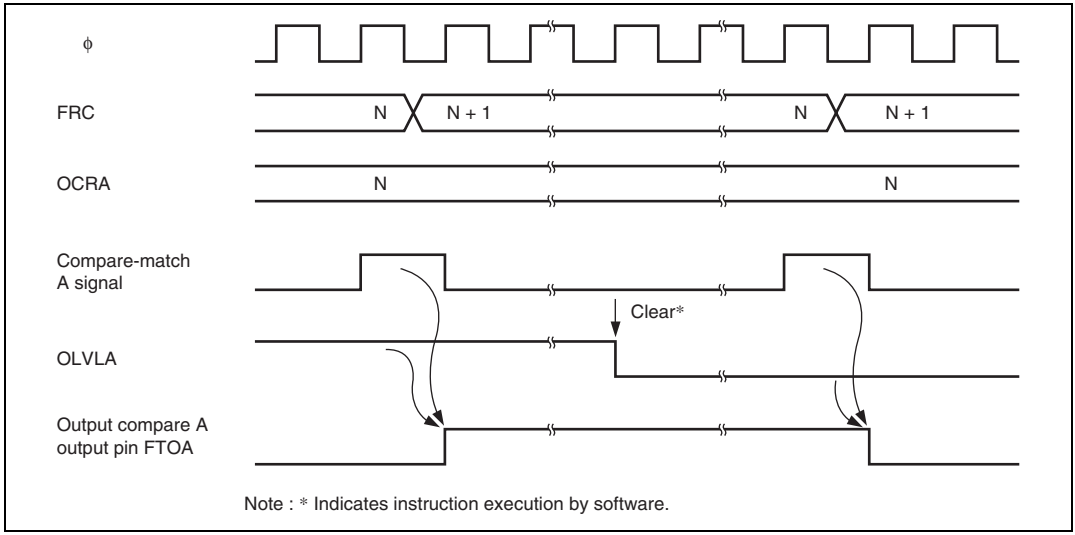


Figure 11.5 Timing of Output Compare A Output

11.5.3 FRC Clear Timing

FRC can be cleared when compare-match A occurs. Figure 11.6 shows the timing of this operation.

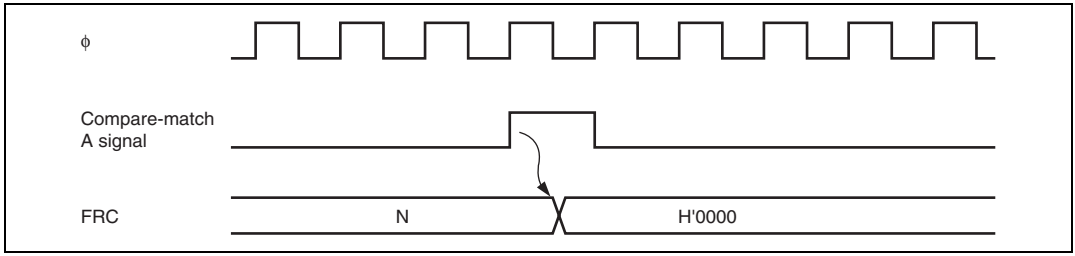


Figure 11.6 Clearing of FRC by Compare-Match A Signal

11.5.4 Input Capture Input Timing

The rising or falling edge can be selected for the input capture input timing by the IEDGA to IEDGD bits in TCR. Figure 11.7 shows the usual input capture timing when the rising edge is selected.

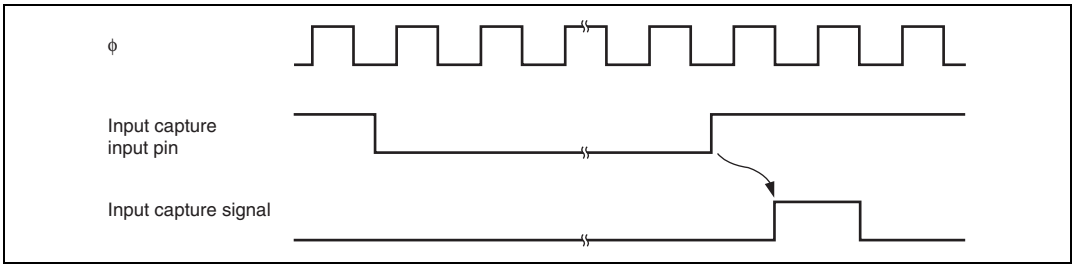


Figure 11.7 Input Capture Input Signal Timing (Usual Case)

If ICRA to ICRD are read when the corresponding input capture signal arrives, the internal input capture signal is delayed by one system clock (ϕ). Figure 11.8 shows the timing for this case.

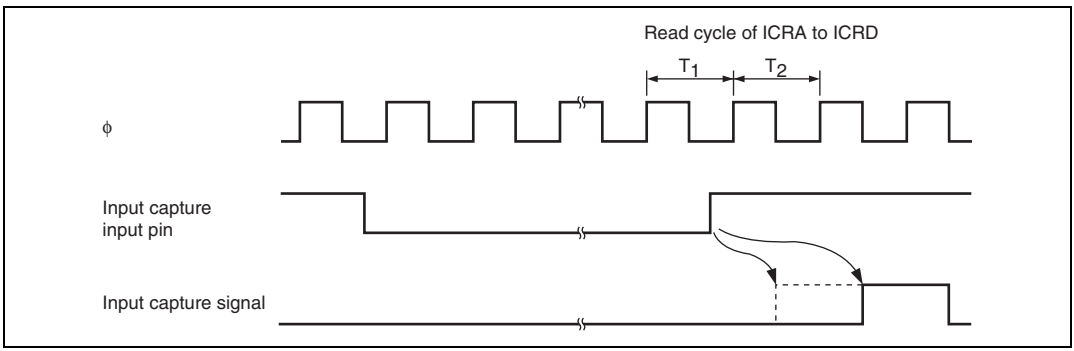


Figure 11.8 Input Capture Input Signal Timing (When ICRA to ICRD is Read)

11.5.5 Buffered Input Capture Input Timing

ICRC and ICRD can operate as buffers for ICRA and ICRB, respectively. Figure 11.9 shows how input capture operates when ICRC is used as ICRA's buffer register (BUFEA = 1) and IEDGA and IEDGC are set to different values (IEDGA = 0 and IEDGC = 1, or IEDGA = 1 and IEDGC = 0), so that input capture is performed on both the rising and falling edges of FTIA.

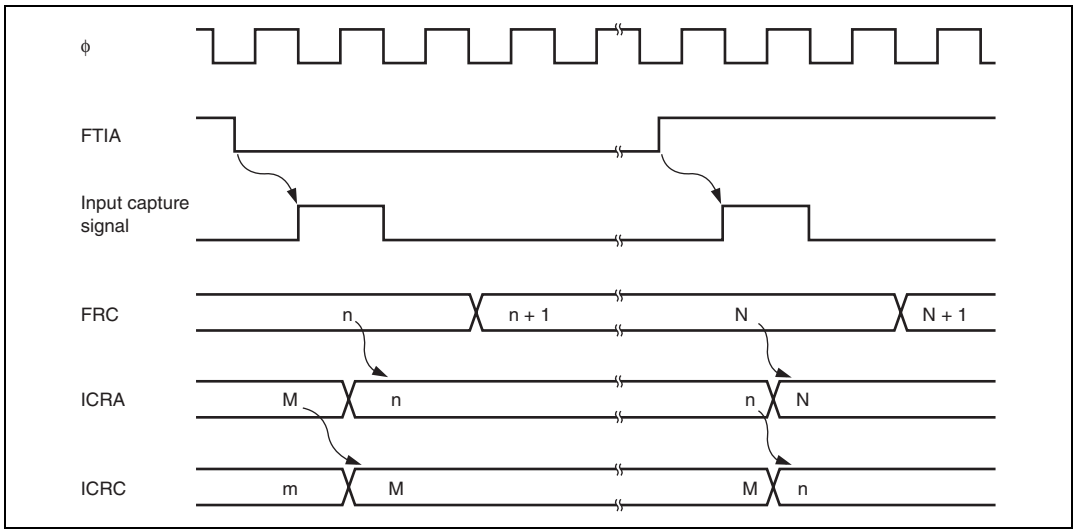


Figure 11.9 Buffered Input Capture Timing

Even when ICRC or ICRD is used as a buffer register, its input capture flag is set by the selected transition of its input capture signal. For example, if ICRC is used to buffer ICRA, when the edge transition selected by the IEDGC bit occurs on the FTIC input capture line, ICFC will be set, and if the ICICE bit is set at this time, an interrupt will be requested. The FRC value will not be transferred to ICRC, however. In buffered input capture, if either set of two registers to which data will be transferred (ICRA and ICRC, or ICRB and ICRD) is being read when the input capture input signal arrives, input capture is delayed by one system clock (φ). Figure 11.10 shows the timing when BUFEA = 1.

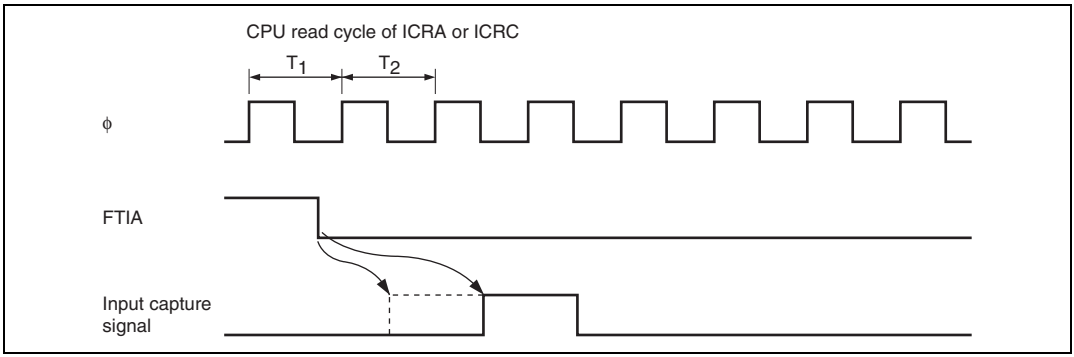


Figure 11.10 Buffered Input Capture Timing (BUFEA = 1)

11.5.6 Timing of Input Capture Flag (ICF) Setting

The input capture flag, ICFA to ICFD, is set to 1 by the input capture signal. The FRC value is simultaneously transferred to the corresponding input capture register (ICRA to ICRD). Figure 11.11 shows the timing of setting the ICFA to ICFD flag.

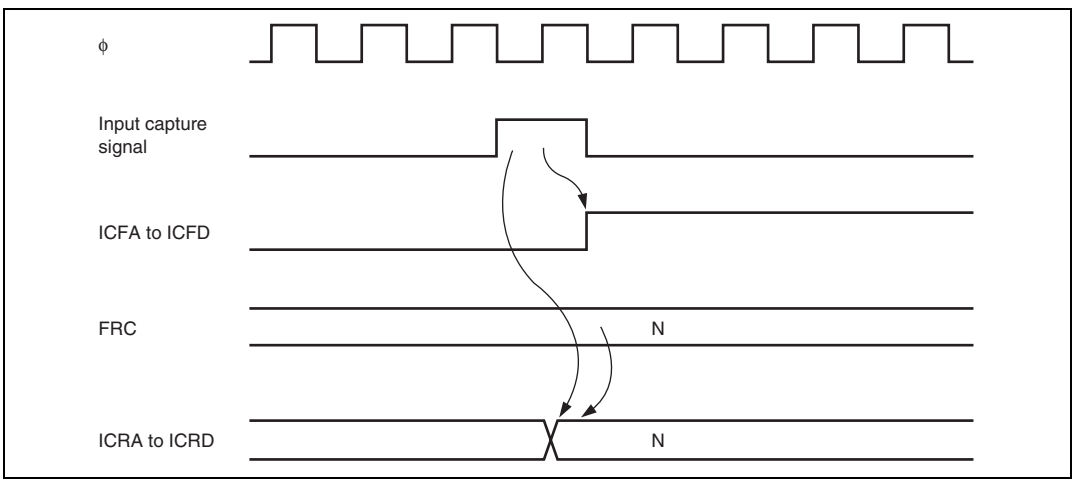


Figure 11.11 Timing of Input Capture Flag (ICFA, ICFB, ICFC, or ICFD) Setting

11.5.7 Timing of Output Compare Flag (OCF) setting

The output compare flag, OCFA or OCFB, is set to 1 by a compare-match signal generated when the FRC value matches the OCRA or OCRB value. This compare-match signal is generated at the last state in which the two values match, just before FRC increments to a new value. When the FRC and OCRA or OCRB value match, the compare-match signal is not generated until the next cycle of the clock source. Figure 11.12 shows the timing of setting the OCFA or OCFB flag.

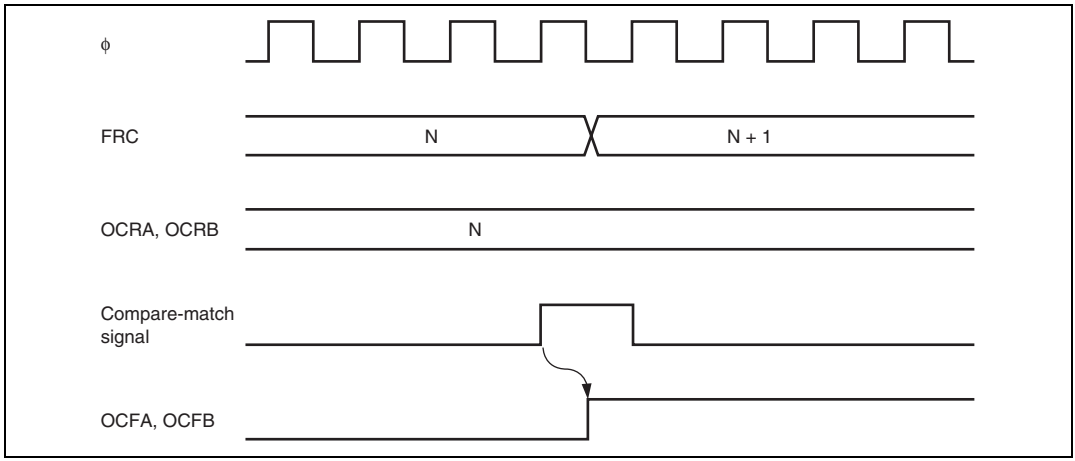


Figure 11.12 Timing of Output Compare Flag (OCFA or OCFB) Setting

11.5.8 Timing of FRC Overflow Flag Setting

The FRC overflow flag (OVF) is set to 1 when FRC overflows (changes from H'FFFF to H'0000). Figure 11.13 shows the timing of setting the OVF flag.

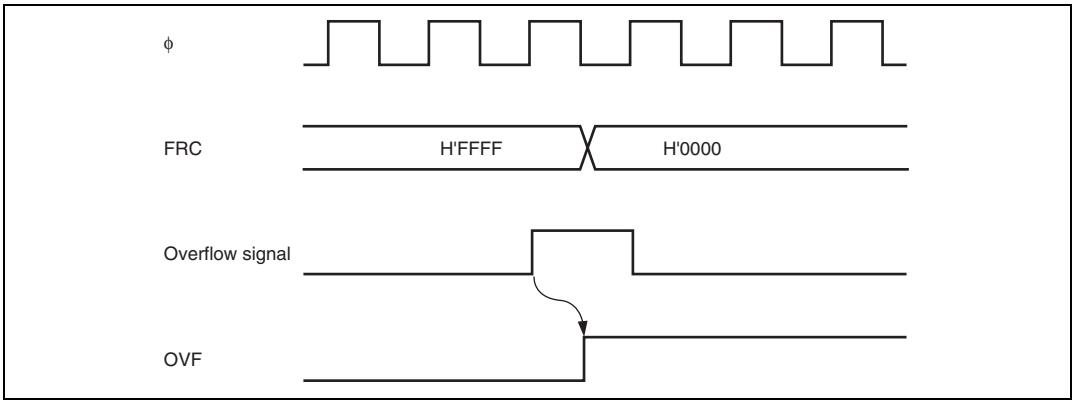


Figure 11.13 Timing of Overflow Flag (OVF) Setting

11.5.9 Automatic Addition Timing

When the OCRAMS bit in TOCR is set to 1, the contents of OCRAR and OCRAF are automatically added to OCRA alternately, and when an OCRA compare-match occurs a write to OCRA is performed. Figure 11.14 shows the OCRA write timing.

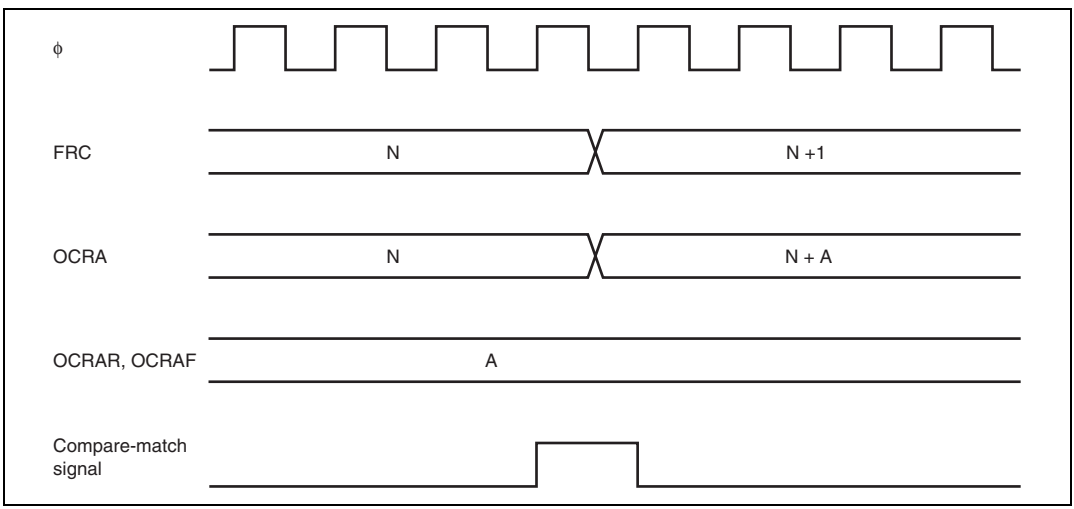


Figure 11.14 OCRA Automatic Addition Timing

11.5.10 Mask Signal Generation Timing

When the ICRDMS bit in TOCR is set to 1 and the contents of OCRDM are other than H'0000, a signal that masks the ICRD input capture signal is generated. The mask signal is set by the input capture signal. The mask signal is cleared by the sum of the ICRD contents and twice the OCRDM contents, and an FRC compare-match. Figure 11.15 shows the timing of setting the mask signal. Figure 11.16 shows the timing of clearing the mask signal.

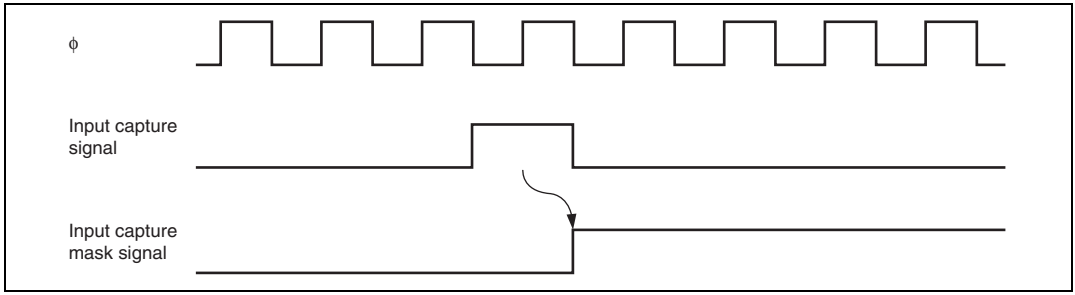


Figure 11.15 Timing of Input Capture Mask Signal Setting

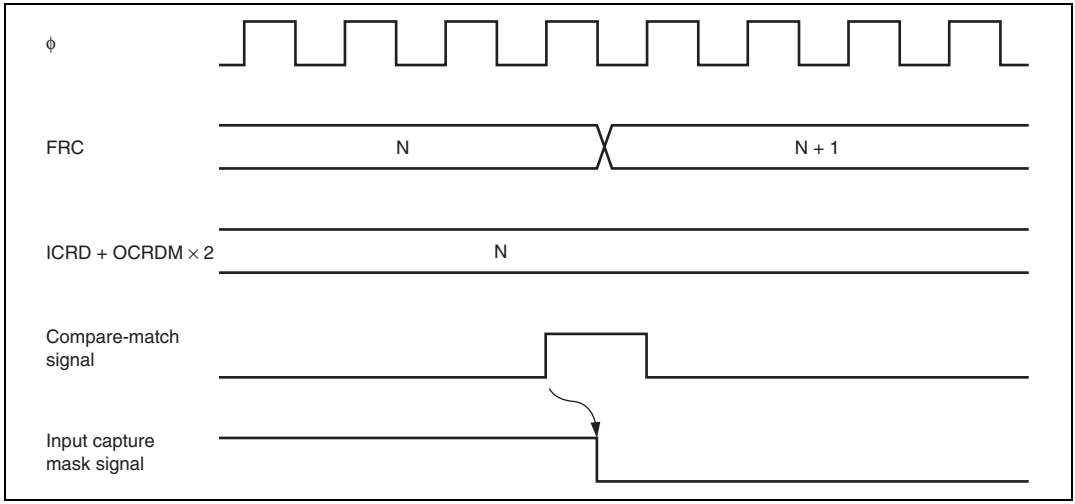


Figure 11.16 Timing of Input Capture Mask Signal Clearing

11.6 Interrupt Sources

The free-running timer can request seven interrupts: ICIA to ICID, OCIA, OCIB, and FOVI. Each interrupt can be enabled or disabled by an enable bit in TIER. Independent signals are sent to the interrupt controller for each interrupt. Table 11.2 lists the sources and priorities of these interrupts.

The ICIA, ICIB, OCIA, and OCIB interrupts can be used as the on-chip DTC activation sources.

Table 11.2 FRT Interrupt Sources

Interrupt	Interrupt Source	Interrupt Flag	DTC Activation	Priority
ICIA	Input capture of ICRA	ICFA	Enable	High
ICIB	Input capture of ICRB	ICFB	Enable	
ICIC	Input capture of ICRC	ICFC	Disable	
ICID	Input capture of ICRD	ICFD	Disable	
OCIA	Compare match of OCRA	OCFA	Enable	
OCIB	Compare match of OCRB	OCFB	Enable	
FOVI	Overflow of FRC	OVF	Disable	Low

11.7 Usage Notes

11.7.1 Conflict between FRC Write and Clear

If an internal counter clear signal is generated during the state after an FRC write cycle, the clear signal takes priority and the write is not performed. Figure 11.17 shows the timing for this type of conflict.

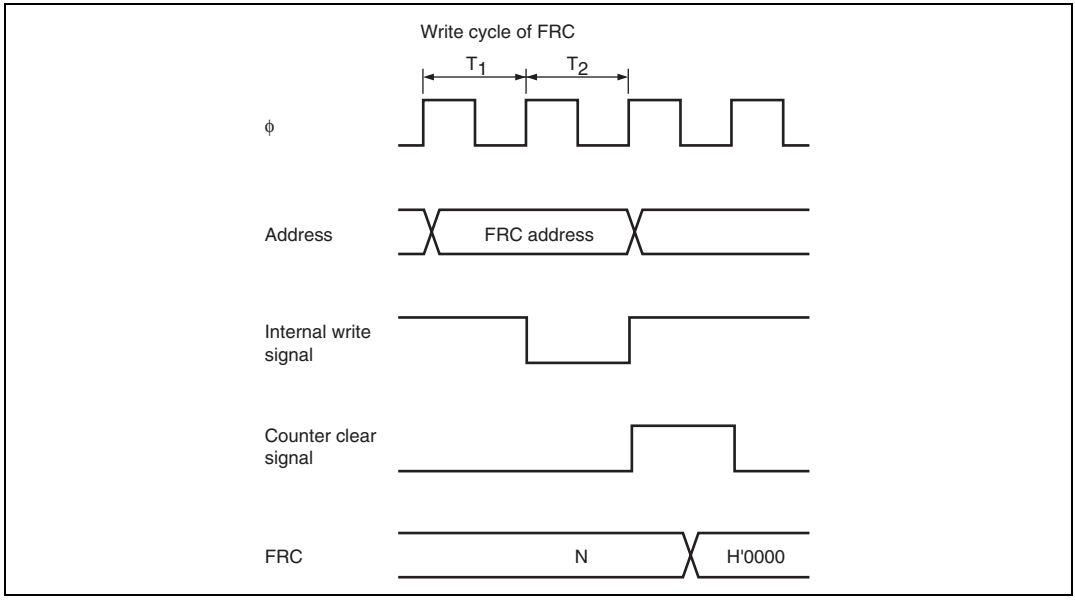


Figure 11.17 Conflict between FRC Write and Clear

11.7.2 Conflict between FRC Write and Increment

If an FRC increment pulse is generated during the state after an FRC write cycle, the write takes priority and FRC is not incremented. Figure 11.18 shows the timing for this type of conflict.

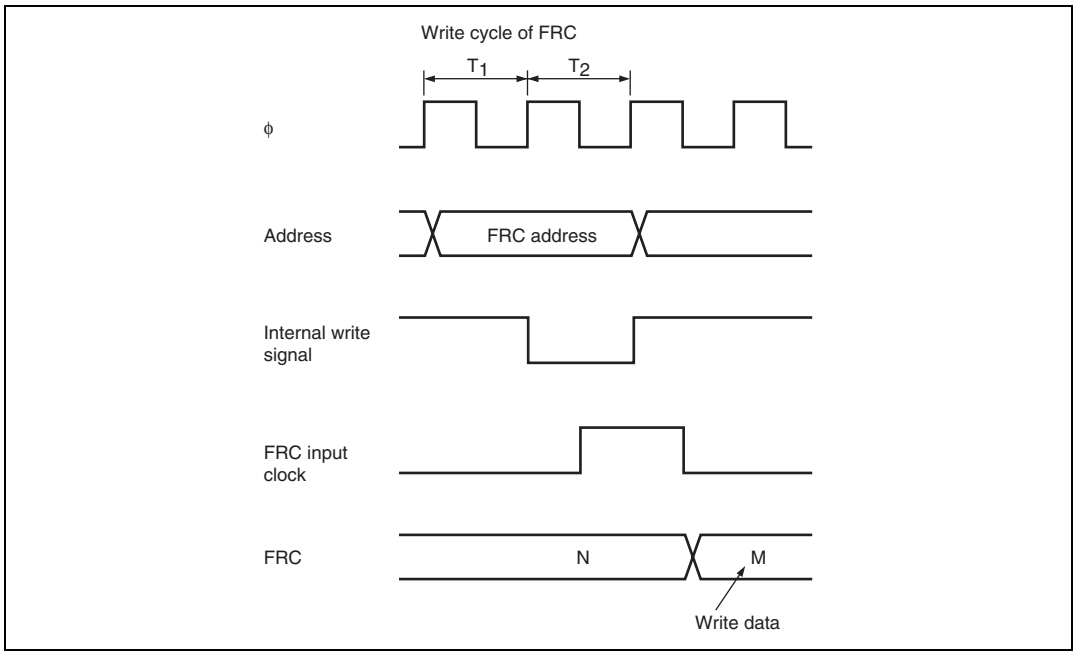


Figure 11.18 Conflict between FRC Write and Increment

11.7.3 Conflict between OCR Write and Compare-Match

If a compare-match occurs during the state after an OCRA or OCRB write cycle, the write takes priority and the compare-match signal is disabled. Figure 11.19 shows the timing for this type of conflict.

If automatic addition of OCRAR and OCRAF to OCRA is selected, and a compare-match occurs in the cycle following the OCRA, OCRAR, and OCRAF write cycle, the OCRA, OCRAR and OCRAF write takes priority and the compare-match signal is disabled. Consequently, the result of the automatic addition is not written to OCRA. Figure 11.20 shows the timing for this type of conflict.

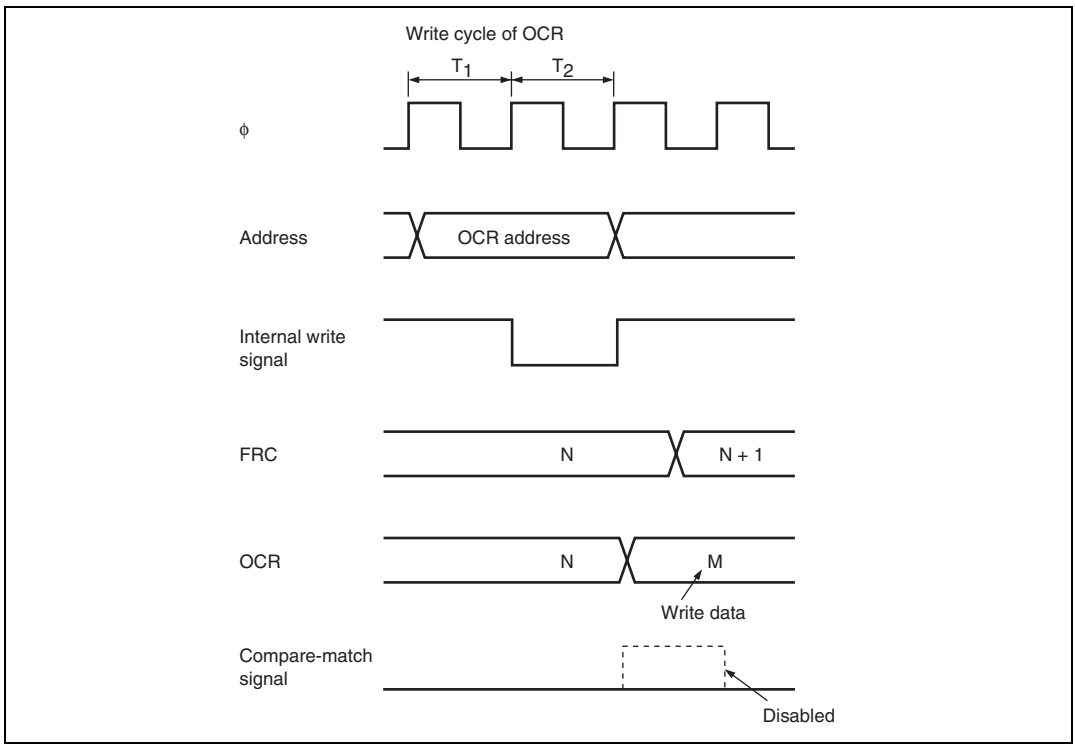


Figure 11.19 Conflict between OCR Write and Compare-Match (When Automatic Addition Function is Not Used)

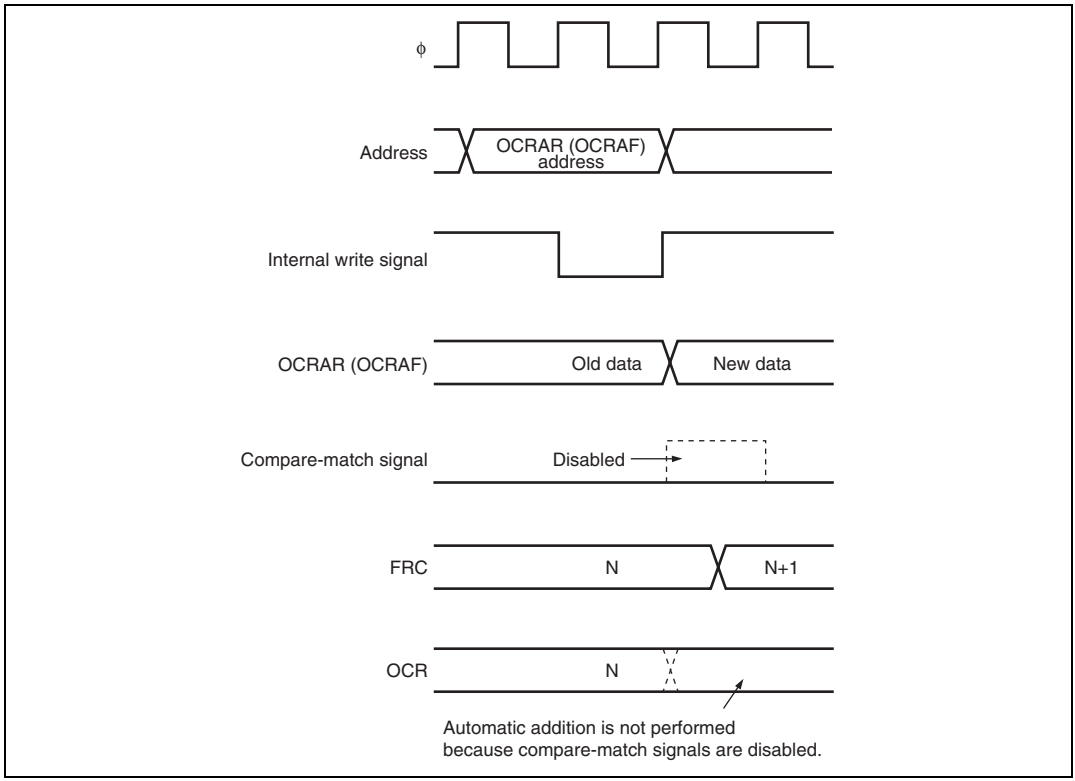


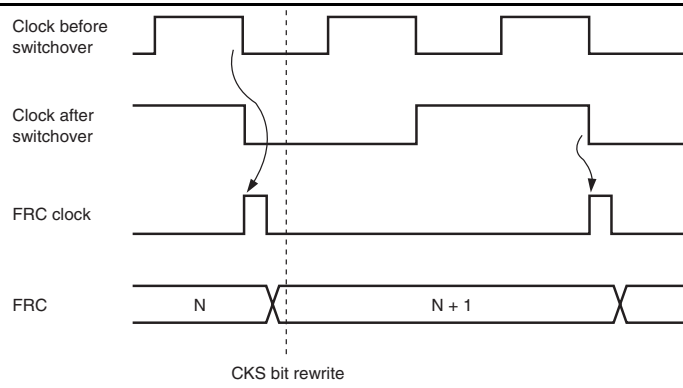
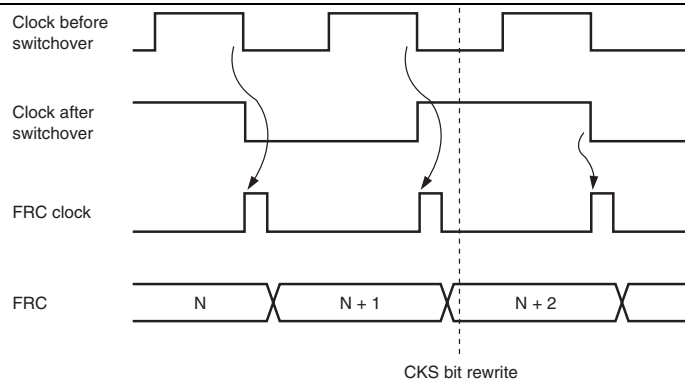
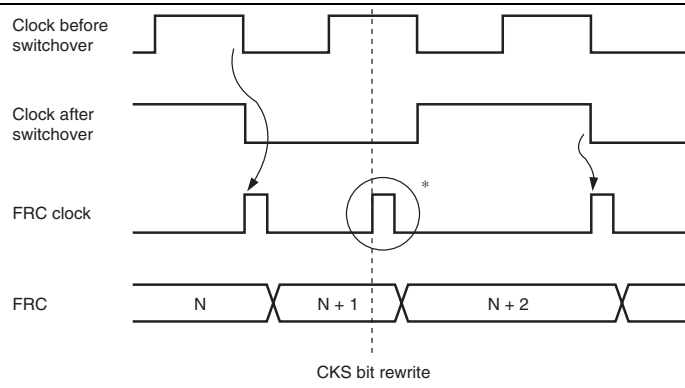
Figure 11.20 Conflict between OCR Write and Compare-Match (When Automatic Addition Function is Used)

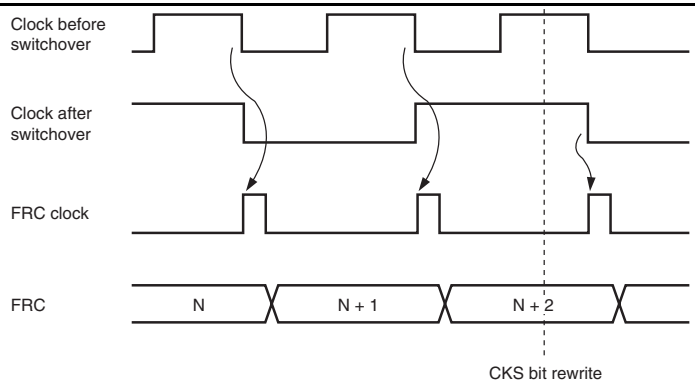
11.7.4 Switching of Internal Clock and FRC Operation

When the internal clock is changed, the changeover may source FRC to increment. This depends on the time at which the clock is switched (bits CKS1 and CKS0 are rewritten), as shown in table 11.3.

When an internal clock is used, the FRC clock is generated on detection of the falling edge of the internal clock scaled from the system clock (φ). If the clock is changed when the old source is high and the new source is low, as in case no. 3 in table 11.3, the changeover is regarded as a falling edge that triggers the FRC clock, and FRC is incremented. Switching between an internal clock and external clock can also source FRC to increment.

Table 11.3 Switching of Internal Clock and FRC Operation

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	FRC Operation
1	Switching from low to low	
2	Switching from low to high	
3	Switching from high to low	

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	FRC Operation
4	Switching from high to high	

Note: * Generated on the assumption that the switchover is a falling edge; FRC is incremented.

11.7.5 Module Stop Mode Setting

FRT operation can be enabled or disabled by the module stop control register. In the initial state, FRT operation is disabled. Access to FRT registers is enabled when module stop mode is cancelled. For details, see section 22, Power-Down Modes.

Section 12 16-Bit Cycle Measurement Timer (TCM)

This LSI has two channels on-chip 16-bit cycle measurement timers (TCM). Each TCM has a 16-bit counter that provides the basis for measuring the periods of input waveforms from fans.

12.1 Features

- Capable of measuring the periods of input waveforms from fans
- Sensed edge is selectable
- 16-bit compare match
- 16-bit resolution
- Selectable counter clock
  - Any of seven internal clocks or an external clock
- Four interrupt sources
  - Counter overflow
  - Cycle limit overflow
  - Compare match
  - Triggering of input capture

Figure 12.1 is a block diagram of the TCM.

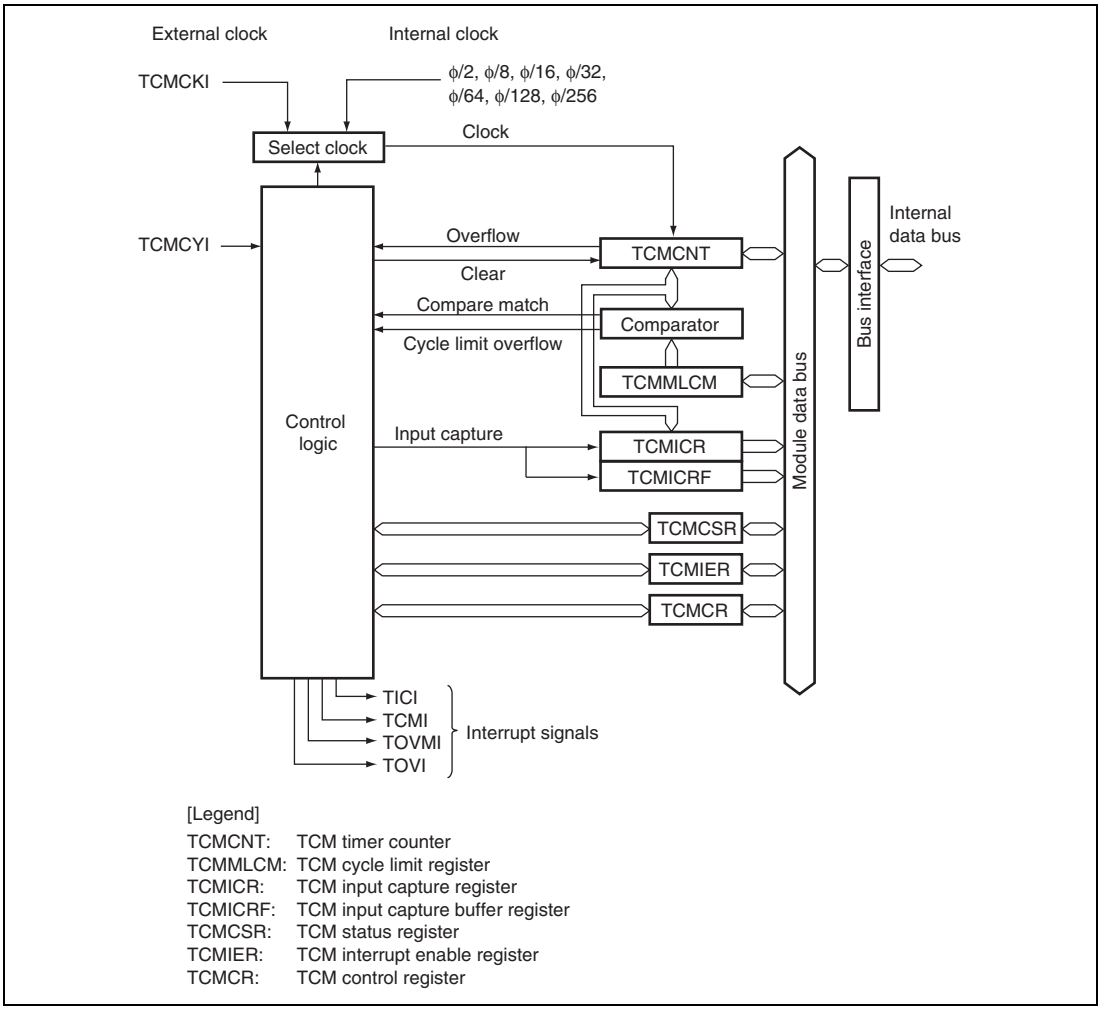


Figure 12.1 Block Diagram of the TCM

12.2 Input/Output Pins

Table 12.1 lists the input and output pins for the TCMs.

Table 12.1 Pin Configuration

Channel	Pin Name	I/O	Function
0	TCMCKI0	Input	External counter clock input
	TCMCYI0	Input	External event input
1	TCMCKI1	Input	External counter clock input
	TCMCYI1	Input	External event input

12.3 Register Descriptions

The TCMs have the following registers.

- TCM timer counter_0 (TCMCNT_0)
- TCM cycle limit register_0 (TCMMLCM_0)
- TCM input capture register_0 (TCMICR_0)
- TCM input capture buffer register_0 (TCMICRF_0)
- TCM status register_0 (TCMCSR_0)
- TCM control register_0 (TCMCR_0)
- TCM interrupt enable register_0 (TCMIER_0)
- TCM timer counter_1 (TCMCNT_1)
- TCM cycle limit register_1 (TCMMLCM_1)
- TCM input capture register_1 (TCMICR_1)
- TCM input capture buffer register_1 (TCMICRF_1)
- TCM status register_1 (TCMCSR_1)
- TCM control register_1 (TCMCR_1)
- TCM interrupt enable register_1 (TCMIER_1)

12.3.1 TCM Timer Counter (TCMCNT)

TCMCNT is a 16-bit readable/writable up-counter. The input clock is selected by the bits CKS2 to CKS0 in TCMCR. When CKS2 to CKS0 are set to B'111, the external clock is selected. In this case, the rising or falling edge is selected by CKSEG in TCMCR.

When TCMCNT overflows (counting changes the value from H'FFFF to H'0000), OVF in TCMCSR is set to 1. When the CST bit in TCMCR is cleared in timer mode, TCMCR is initialized to H'0000. In speed measurement mode, TCMCNT is cleared by detection of the first edge (the edge selected with the IEDG bit in TCMCR) of the measurement period (two periods of the input waveform form one measurement period).

In timer mode, TCMCNT is always writable. TCMCNT cannot be modified in speed measurement mode. TCMCNT should always be accessed in 16-bit units and cannot be accessed in 8-bit units. TCMCNT is initialized to H'0000.

12.3.2 TCM Cycle Limit Register (TCMMLCM)

TCMMLCM is a 16-bit readable/writable register. TCMMLCM is available as a compare match register when the TCMMDS bit in TCMCR is cleared (operation is in timer mode). TCMMLCM is available as a cycle limit register when the TCMMDS bit in TCMCR is set to 1 (operation is in speed measurement mode).

In timer mode, the value in TCMMLCM is constantly compared with that in TCMCNT, when the values match, CMF in TCMCSR is set to 1. However, comparison is disabled in the second half of a cycle of writing to TCMMLCM.

In speed measurement mode, a value that sets an upper limit on the measurement period (which is formed by two periods of the input waveform) can be set in TCMMLCM. When the third edge (rising or falling as selected with the IEDG bit in TCMCR) of the measurement period is detected, the value in TCMCNT is transferred to TCMICR. At this time, the values in TCMICR and TCMMLCM are compared. The MAXOVF flag in TCMCSR is set to 1 if the value in TCMICR is greater than that in TCMMLCM. TCMMLCM should always be accessed in 16-bit units and cannot be accessed in 8-bit units. TCMMLCM is initialized to H'FFFF.

12.3.3 TCM Input Capture Register (TCMICR)

TCMICR is a 16-bit read only register. In timer mode, the value in TCMCNT is transferred to TCMICR on the edge selected by the IEDG bit in TCMCR. At the same time, the ICPF flag in TCMCSR is set to 1. In speed measurement mode, the value in TCMCNT is transferred to TCMICR on detection of the third edge of the measurement period. At this time, the ICPF flag in TCMCSR is set to 1. TCMICR should always be accessed in 16-bit units and cannot be accessed in 8-bit units. TCMICR is initialized to H'0000.

12.3.4 TCM Input Capture Buffer Register (TCMICRF)

TCMICRF is a 16-bit read only register. TCMICRF can be used as TCMICR buffer register. When input capture is generated, the value in TCMICR is transferred to TCMICRF.

TCMICR and TCMICRF should always be accessed in 16-bit units and cannot be accessed in 8-bit units. TCMICRF is initialized to H'0000.

12.3.5 TCM Status Register (TCMCSR)

TCMCSR is an 8-bit readable/writable register that controls operation of the interrupt sources.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Timer Overflow This flag indicates that the TCMCNT has overflowed. Only 0 can be written to clear the flag. [Setting condition] Overflow of TCMCNT (change in value from H'FFFF to H'0000) [Clearing condition] Reading OVF when OVF = 1 and then writing 0 to OVF.

Bit	Bit Name	Initial Value	R/W	Description
6	MAXOVF	0	R/(W)*	Measurement Period Limit Overflow This flag indicates that the number of cycles measured in one measurement period (two periods of the input waveform form one measurement period) in speed measurement mode has exceeded the upper limit for measurement periods, as set in TCMMLCM. Only 0 can be written to clear the flag. [Setting condition] When TCMICR is greater than TCMMLCM [Clearing condition] Reading MAXOVF when MAXOVF = 1 and then writing 0 to MAXOVF
5	CMF	0	R/(W)*	Compare Match Flag (only valid in timer mode) [Setting condition] When the values in TCMCNT and TCMMLCM match. Note: CMF is not set in speed measurement mode, even when the values in TCMCNT and TCMMLCM match. [Clearing condition] Reading CMF when CMF = 1 and then writing 0 to CMF
4	CKSEG	0	R/(W)*	External Clock Edge Select If bits CKS2 to CKS0 in TCMCR are set to B'111, this bit selects the edge for counting of external count clock edge. 0: Count falling edges of the external clock. 1: Count rising edges of the external clock.

Bit	Bit Name	Initial Value	R/W	Description
3	ICPF	0	R/(W)*	Input Capture Generation Timer mode: The flag indicates that the value in TCMCNT has been transferred to TCMICR on generation of an input capture signal. This flag is set when the input capture signal is generated, i.e. on detection of the edge selected by the IEDGD bit on the TCMCYI input pin. Speed measurement mode: The flag indicates that the value in TCMCNT has been transferred to TCMICR on detection of the third edge (rising or falling as determined by the IEDG bit in TCMCR) during the measurement period. Only 0 can be written to clear the flag. [Setting condition] Generation of the input capture signal [Clearing condition] Reading ICPF when ICPF = 1 and then writing 0 to ICPF
2 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

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

12.3.6 TCM Control Register (TCMCR)

TCMCR is an 8-bit readable/writable register. TCMCR selects input capture input edge, counter start, and counter clock, and controls operation mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CST	0	R/W	Counter Start In timer mode, setting this bit to 1 starts counting by TCMCNT; clearing this bit stops counting by TCMCNT. Then, the counter is initialized to H'0000, and input-capture operation stops. Clear this bit and thus return TCMCNT to H'0000 in initialization for speed measurement mode.

Bit	Bit Name	Initial Value	R/W	Description
6	POCTL	0	R/W	TCMCYI Input Polarity Reversal 0: Use the TCMCYI input directly 1: Use the inverted TCMCYI input Note: Modify this bit while CST = 0 and TCMMDS = 0
5	CPSPE	0	R/W	Input Capture Stop Enable Controls whether or not counting up by TCMCNT and input-capture operation stop or continue when MAXOVF is set to 1 in speed measurement mode. The bit does not affect operation in timer mode. 0: Counting up and input-capture operation continue when the MAXOVF flag is set to 1. 1: Counting up and input-capture operation stop when the MAXOVF flag is set to 1.
4	IEDG	0	R/W	Input Edge Select In timer mode, selects the falling or rising edge of the TCMCYI input for use in input capture, in combination with the value of the POCTL bit. In speed-measurement mode, selects the falling or rising edge of the TCMCYI input for use in measurement, in combination with the value of the POCTL bit. POCTL = 0 0: Selects the rising edge of the TCMCYI input 1: Selects the falling edge of the TCMCYI input POCTL = 1 0: Selects the falling edge of the TCMCYI input 1: Selects the rising edge of the TCMCYI input
3	TCMMDS	0	R/W	TCM Mode Select Selects the TCM operating mode. 0: Timer mode The TCM provides compare match and input capture facilities. 1: Speed measurement mode TCMCNT should be initialized to H'0000. Clear the CST in TCMCR to 0 before setting to speed measurement mode.

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 2, 1, 0
1	CKS1	0	R/W	Selects the clock signal for input to TCMCNT.
0	CKS0	0	R/W	Note: Modify this bit when CST = 0 and TCMMDS = 0
				000: Count ϕ/2 internal clock
				001: Count ϕ/8 internal clock
				010: Count ϕ/16 internal clock
				011: Count ϕ/32 internal clock
				100: Count ϕ/64 internal clock
				101: Count ϕ/128 internal clock
				110: Count ϕ/256 internal clock
				111: Count external clock (select the external clock edge with CKSEG in TCMCSR.)

12.3.7 TCM Interrupt Enable Register (TCMIER)

TCMIER is an 8-bit readable/writable register that enables or disables interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	OVIE	0	R/W	Counter Overflow Interrupt Enable
				Enables or disables the issuing of interrupt requests on setting of the OVF flag in TCMCSR to 1.
				0: Disable interrupt requests by OVF
				1: Enable interrupt requests by OVF
6	MAXOVIE	0	R/W	Measurement Period Limit Overflow Interrupt Enable
				Enables or disables the issuing of interrupt requests on setting of the MAXOVF flag in TCMCSR to 1.
				0: Disable interrupt requests by MAXOVF
				1: Enable interrupt requests by MAXOVF

Bit	Bit Name	Initial Value	R/W	Description
5	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables the issuing of interrupt requests when the CMF bit in TCMCSR is set to 1. 0: Disable interrupt requests by CMF 1: Enable interrupt requests by CMF
4	TCMIPE	0	R/W	Input Capture Input Enable Enables input to the pin. 0: Disable input 1: Enable input Note: Modify this bit when CST = 0 and TCMMDS = 0.
3	ICPIE	0	R/W	Input Capture Interrupt Enable Enables or disables interrupt requests when the ICPF flag in TCMCSR is set to 1. 0: Disable interrupt requests by ICPF 1: Enable interrupt requests by ICPF
2	—	0	R/W	Reserved
1	—	0	R/W	The initial value should not be changed.
0	—	0	R	Reserved This bit is always read as 0 and cannot be modified.

12.4 Operation

TCMCNT operates in timer mode or speed measurement mode. TCMCNT is in timer mode after a reset.

12.4.1 Timer Mode

(1) Counter Operation

TCMCNT operates as a free running counter in timer mode. TCMCNT starts counting up when the CST bit in TCMCR is set to 1. When TCMCNT overflows (the value changes from H'FFFF to H'0000), the OVF bit in TCMCSR is set to 1 and an interrupt request is generated if the OVIE bit in TCMIER is 1. Figure 12.2 shows an example of free running counter operation. In addition, figure 12.3 shows TCMCNT count timing of external clock operation. The external clock should have a pulse width of no less than 1.5 cycles. The counter will not operate correctly if the pulses are narrower than this.

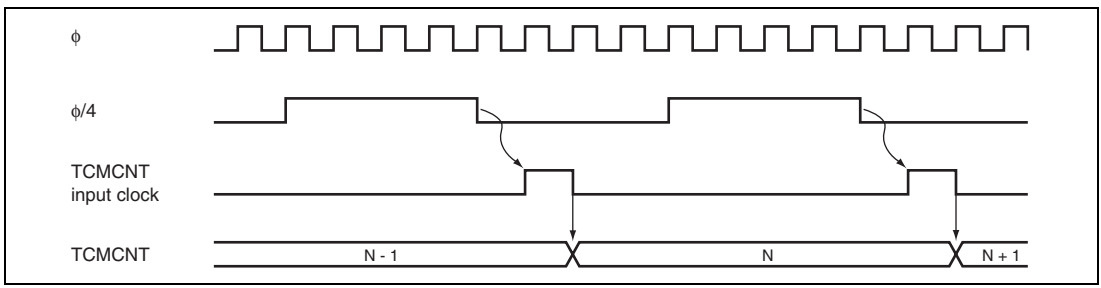


Figure 12.2 Example of Free Running Counter Operation

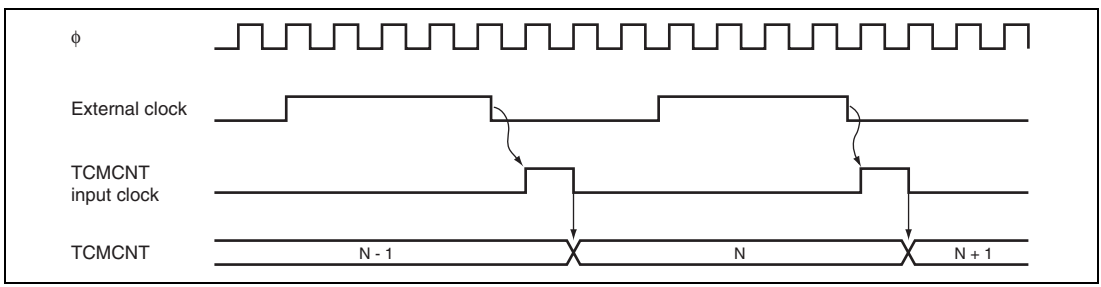


Figure 12.3 Count Timing of External Clock Operation (Falling Edges)

(2) Input Capture

The value in TCMCNT is transferred to TCMICR by detecting input edge of TCMCYI pin in timer mode. At this time, the ICPF flag in TCMCSR is set. Detection of rising or falling edges is selectable. Figure 12.4 shows an example of the timing of input capture operations and figure 12.5 shows buffer operation of input capture.

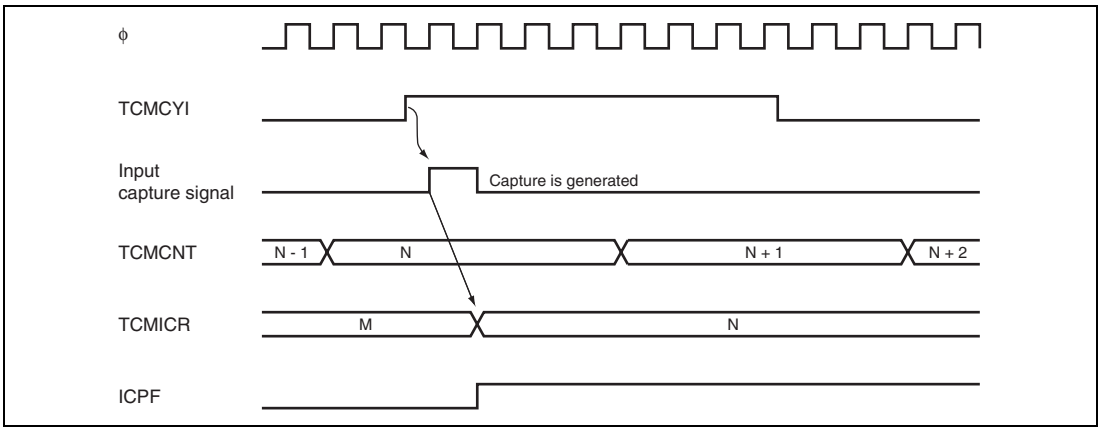


Figure 12.4 Input Capture Operation Timing (Sensing of Rising Edges)

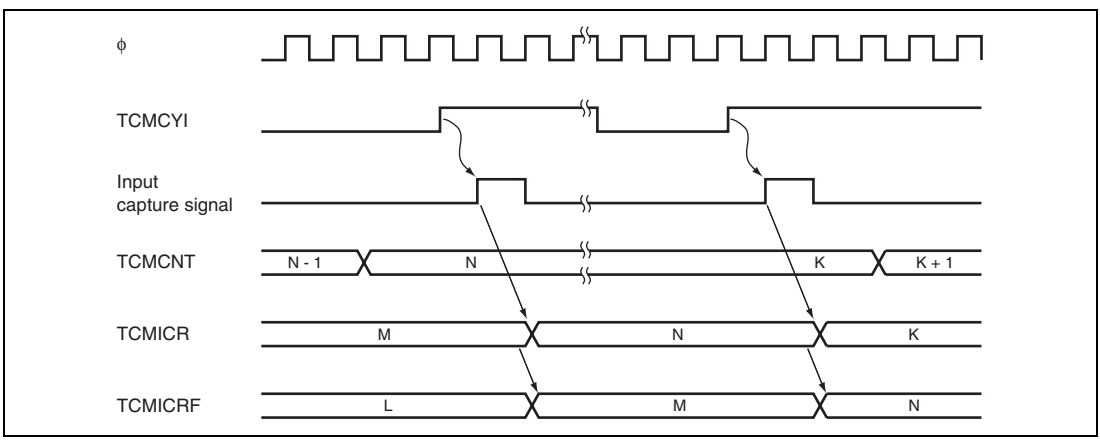


Figure 12.5 Buffer Operation of Input Capture

(3) CMF Set Timing when a Compare Match occurs

The CMF flag in TCMCSR is set in the last state where the values in TCMCNT and TCMMLCM match in timer mode. Therefore, a compare match signal is not generated until a further cycle of the TCMCNT input clock is generated after a match between the values in TCMCNT and TCMMLCM. Figure 12.6 shows the timing with which the CMF flag is set.

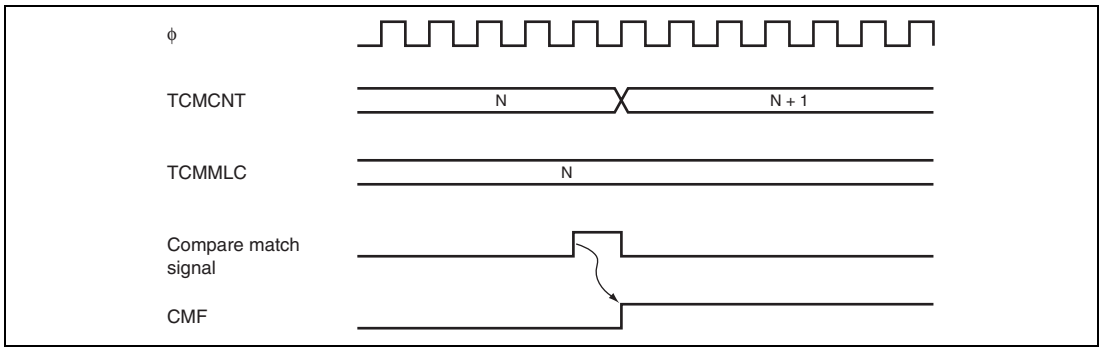


Figure 12.6 Timing of CMF Flag Setting on a Compare Match

12.4.2 Speed Measurement Mode

(1) Counter Operation

Setting the TCMMDS bit in TCMCR to 1 selects speed measurement mode, in which counting up proceeds regardless of the setting of the CST bit in TCMCR. TCMCNT is cleared to H'0000 on detection of the first edge in the measurement period and counts up from there. Figure 12.7 shows an example of counter operation in speed measurement mode.

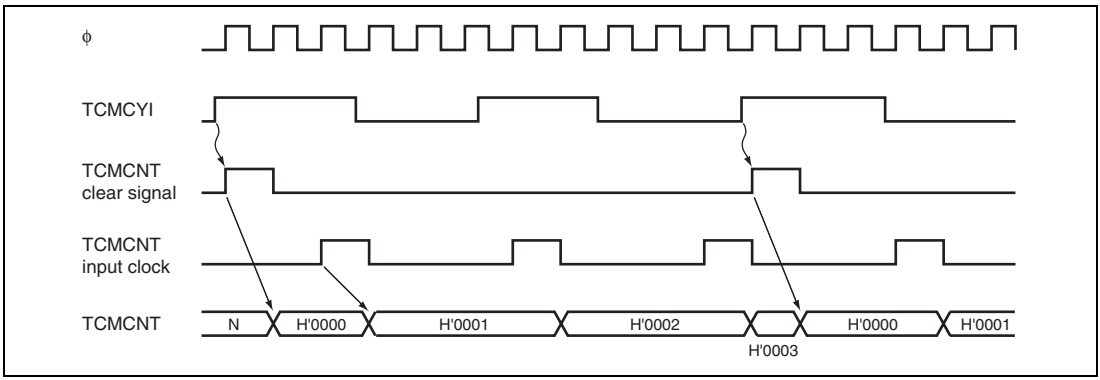


Figure 12.7 Example of Counter Operation in Speed Measurement Mode

(2) Measuring a Speed

In speed measurement mode, two cycles of the input waveform for TCM form one measurement cycle. Start by setting TCMMDS = 0 and then set CST = 0, which clears TCMCNT to H'0000. After that, set an upper limit on the measurement cycle in the TCMMLCM register. Finally, place the timer in speed measurement mode by setting the TCMMDS bit in TCMCR to 1. TCMCNT will count cycles of the selected clock. On detection of the first edge (either rising or falling as selected with the IEDG bit in TCMCR) of the measurement cycle, TCMCNT is automatically cleared to H'0000. On detection of the third edge, the value in TCMCNT is transferred to TCMICR. At this time, the value in TCMICR is compared with the value in TCMMLCM. If TCMICR is larger than TCMMLCM, the MAXOVF bit in TCMCSR is set to 1. If generation of the corresponding interrupt request is enabled by the setting in TCMIER, the request is generated. In addition, on detection of the third edge, TCMCNT is cleared to H'0000, and the next round of measurement starts.

When the CPSPE bit in TCMCR has been cleared to 0, the next round of speed measurement will start, even if the MAXOVF flag is set to 1.

If the MAXOVF flag is set to 1 while the CPSPE bit in TCMCR is set to 1, counting up by TCMCNT stops and so does speed measurement. Subsequently clearing MAXOVF to 0 automatically clears TCMCNT to H'0000, and counting up for speed measurement is then restarted.

Figure 12.8 shows an example of timing in speed measurement.

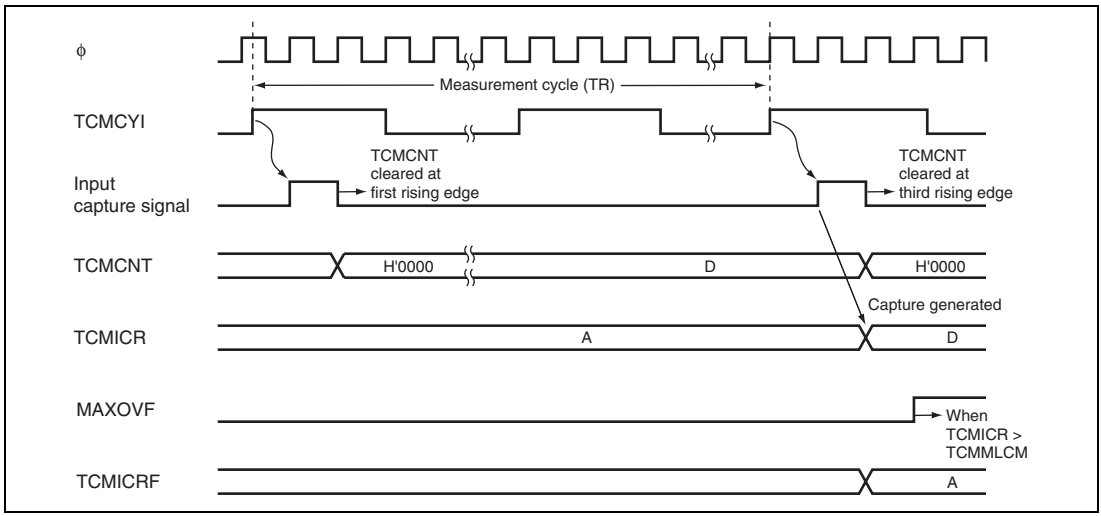


Figure 12.8 Example of Timing in Speed Measurement

(3) Determination of Fan Stoppage

Either of two sets of conditions can be considered to represent fan-stopped states.

The fan can be considered to have stopped when a timer overflow is generated within the period from the start of speed measurement mode to detection of the first edge (rising or falling as selected with the IEDG bit in TCMCR).

Figure 12.9 shows an example of the timing of fan-stopped state (1).

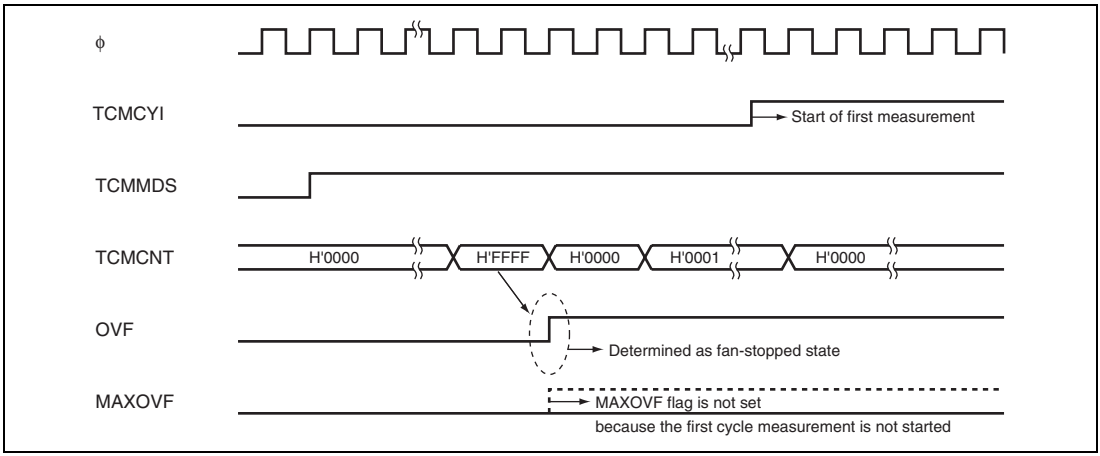


Figure 12.9 Example of Timing in Fan-Stopped State (1)

Speed measurement stops if MAXOVF is set to 1 while the CPSPE bit in TCMCR is set to 1. Subsequently clearing MAXOVF to 0 restarts speed measurement. In this case, the fan can be considered to have stopped if a timer overflow is generated before detection of the first edge.

Figure 12.10 shows an example of the timing of the fan-stopped state (2).

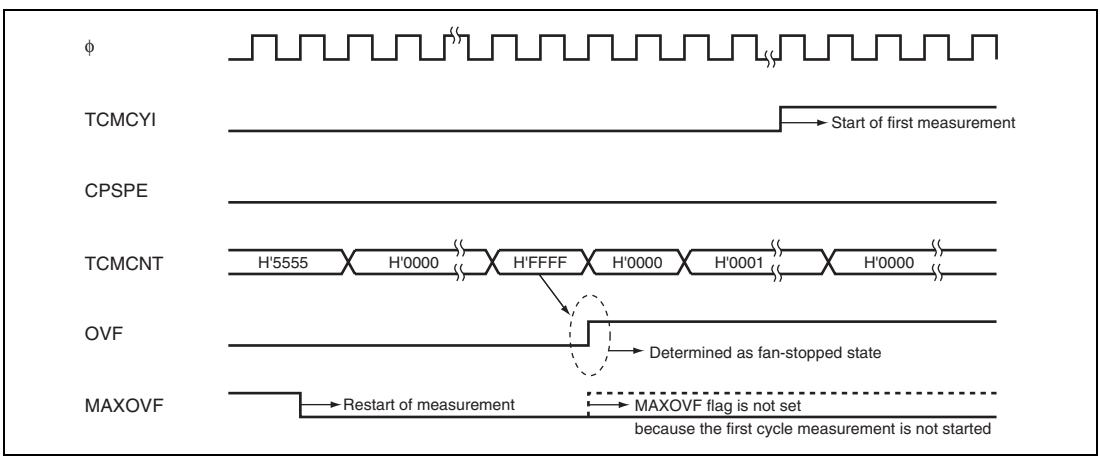


Figure 12.10 Example of Timing in Fan-Stopped State (2)

(4) Example of Settings for Speed Measurement Mode

Figure 12.11 shows an example of the flow when speed measurement mode is to be used.

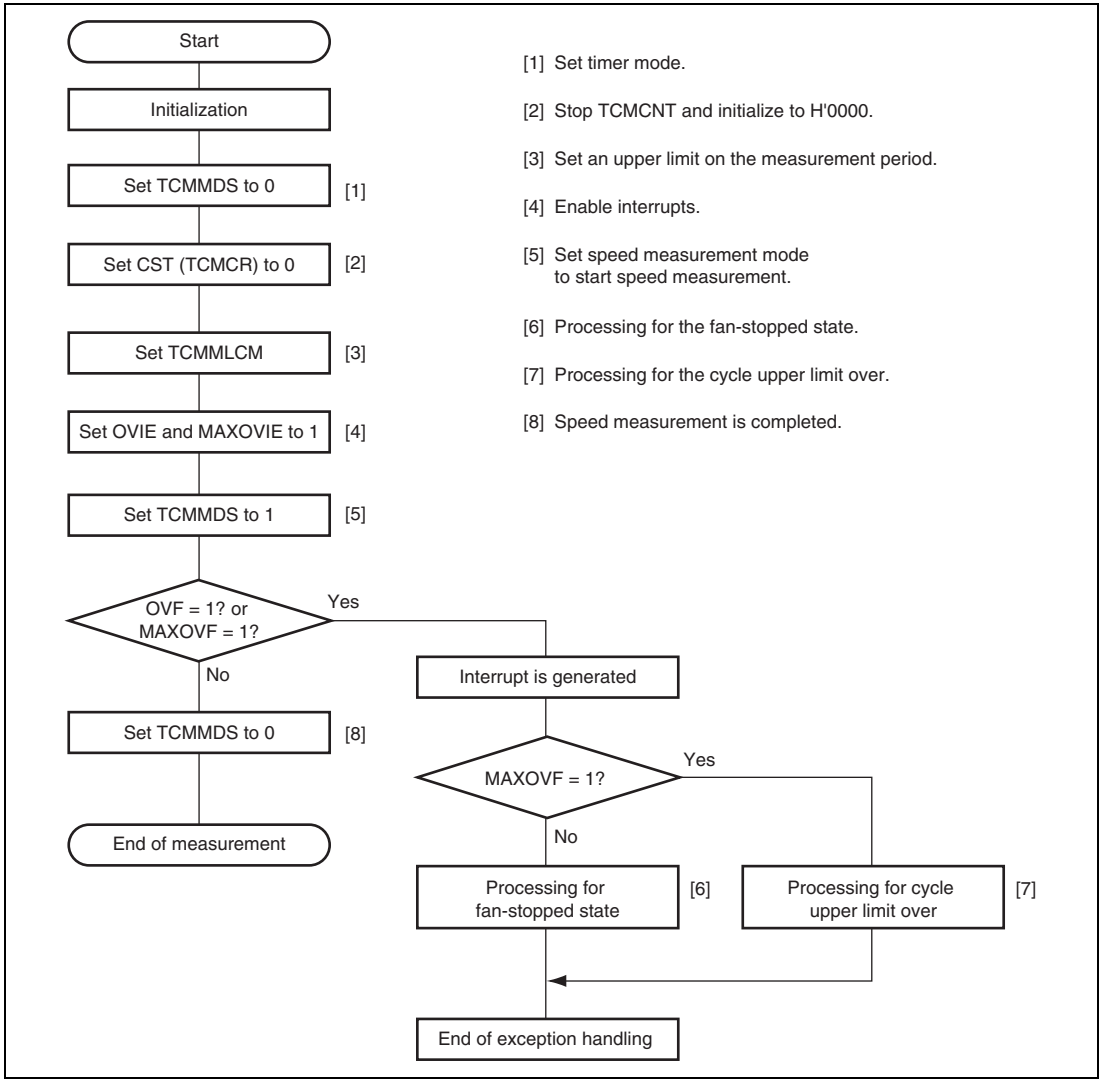


Figure 12.11 Example of Speed Measurement Mode Settings

(5) Formula for Calculating Fan Speed from Results in Speed Measurement Mode

Two cycles of the input waveform from a fan form a single measurement cycle in speed measurement mode. The rotation speed of a fan is given by the following formula.

- When the fan motor generates two pulses per rotation:

$$\text{Fan rotation speed (RPM)} = \frac{\text{TCMCNT input clock frequency} \times 60 \times 2}{\text{Number of captures counted by TCMCNT}}$$

(6) Measurement Error and Range in Speed Measurement Mode

When the TCM is used in speed measurement mode, take the range of error in measurement and the range of measurable frequencies into account in selecting the source clock to drive counting.

When measuring the pulses of a signal input via the TCMCYI pin, the possible error in the number counted due to the conditions of measurement is N ± 1 (N: the correct number of input TCM cycles). Therefore, the error in measurement is given by the following formula.

$$\text{Measurement error} = \left| \frac{N \pm 1}{N} \right| \times 100\% - 1 = \frac{1}{N} \times 100\%$$

Table 12.2 lists the ranges of error in measurement when the TCM period is from 60 to 3.75 ms and the frequency of the system clock ϕ is from 8 to 20 MHz.

Table 12.2 Range of Error in Measurement

Input Clock Frequency of TCMCNT	Number Counted by TCMCNT		Range of Error (%)
ϕ/2	15000 to 65535	H'3A98 to H'FFFF	0.001 to 0.006
ϕ/8	3750 to 65535	H'EA6 to H'FFFF	0.001 to 0.002
ϕ/16	1875 to 65535	H'753 to H'FFFF	0.001 to 0.05
ϕ/32	937±1 to 65535	H'3A9±1 to H'FFFF	0.001 to 0.11
ϕ/64	468±1 to 37500	H'1D4±1 to H'927C	0.002 to 0.21
ϕ/128	234±1 to 18750	H'EA±1 to H'493E	0.005 to 0.43
ϕ/256	117±1 to 9375	H'75±1 to H'249F	0.01 to 0.85

Table 12.3 lists the measurable ranges of frequency for an error of measurement of 5% or less when the frequency of the system clock φ is 8, 10, or 20 MHz.

Table 12.3 Range of Measurement Speed

Input Clock Frequency of TCMCNT		Number Counted by TCMCNT	Measurable Range (RMP)
φ/2	8 MHz	20 to 65535	24000000 to 7324
	10 MHz	20 to 65535	30000000 to 9155
	20 MHz	20 to 65535	60000000 to 18310
φ/8	8 MHz	20 to 65535	6000000 to 1830
	10 MHz	20 to 65535	7500000 to 2288
	20 MHz	20 to 65535	15000000 to 4576
φ/16	8 MHz	20 to 65535	3000000 to 914
	10 MHz	20 to 65535	3750000 to 1144
	20 MHz	20 to 65535	7500000 to 2288
φ/32	8 MHz	20 to 65535	1500000 to 456
	10 MHz	20 to 65535	1875000 to 572
	20 MHz	20 to 65535	3750000 to 1144
φ/64	8 MHz	20 to 65535	750000 to 228
	10 MHz	20 to 65535	937500 to 286
	20 MHz	20 to 65535	1875000 to 572
φ/128	8 MHz	20 to 65535	375000 to 114
	10 MHz	20 to 65535	468750 to 143
	20 MHz	20 to 65535	937500 to 286
φ/256	8 MHz	20 to 65535	187500 to 56
	10 MHz	20 to 65535	234375 to 71
	20 MHz	20 to 65535	468750 to 143

12.5 Interrupt Sources

TCM has four interrupt sources: TICI, TCMI, TOVMI, and TOVI. Each interrupt source is either enabled or disabled by the corresponding interrupt enable bit in TCMIER and independently transferred to the interrupt controller. Table 12.4 lists the interrupt sources in priority order.

TICI and TCMI interrupts can be set up as activation sources for the on-chip DTC.

Table 12.4 TCM Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority
TCM_0	TICI0	TCMICR_0 input capture	ICPF_0	Available	High
	TCMI0	TCMMLCM_0 compare match	CMF_0	Available	
	TOVMI0	TCMMLCM_0 overflow	MAXOVF_0	Not available	
	TOVI0	TCMCNT_0 overflow	OVF_0	Not available	
TCM_1	TICI1	TCMICR_1 input capture	ICPF_1	Available	
	TCMI1	TCMMLCM_1 compare match	CMF_1	Available	
	TOVMI1	TCMMLCM_1 overflow	MAXOVF_1	Not available	
	TOVI1	TCMCNT_1 overflow	OVF_1	Not available	Low

12.6 Usage Notes

12.6.1 Conflict between TCMCNT Write and Count-Up Operation

When a conflict between TCMCNT write and count-up operation occurs in the second half of the TCMCNT write cycle, TCMCNT is not incremented and writing to TCMCNT takes priority. Figure 12.12 shows the timing of this conflict.

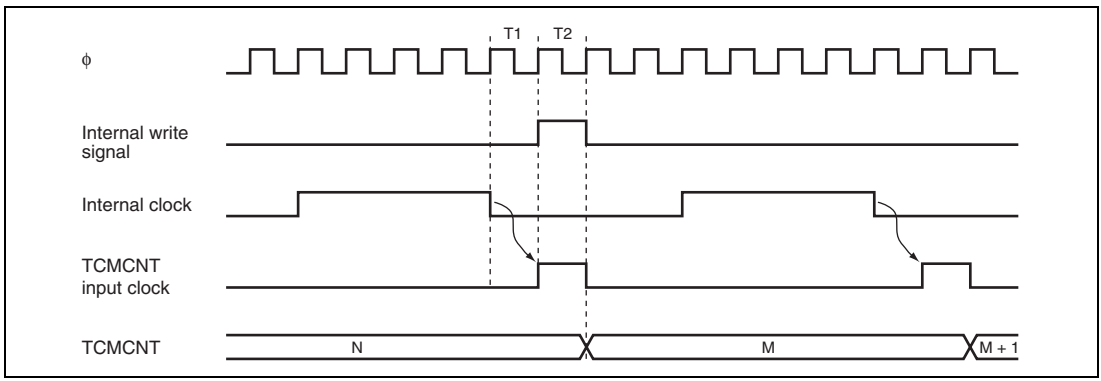


Figure 12.12 Conflict between TCMCNT Write and Count-Up Operation

12.6.2 Conflict between TCMMLCM Write and Compare Match

When a conflict between TCMMLCM write and a compare match should occur in the second half of a cycle of writing to TCMMLCM, writing to TCMMLCM takes priority and the compare match signal is inhibited. Figure 12.13 shows the timing of this conflict.

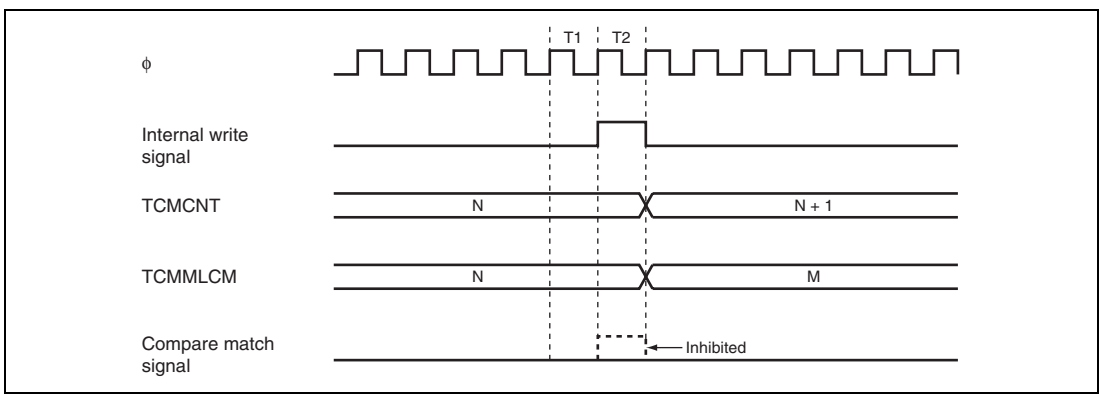


Figure 12.13 Conflict between TCMMLCM Write and Compare Match

12.6.3 Conflict between TCMICR Read and Input Capture

When operation is in timer mode and the corresponding input capture signal is detected during reading of TCMICR, the input capture signal is delayed by one system clock (ϕ). Figure 12.14 shows the timing of this conflict.

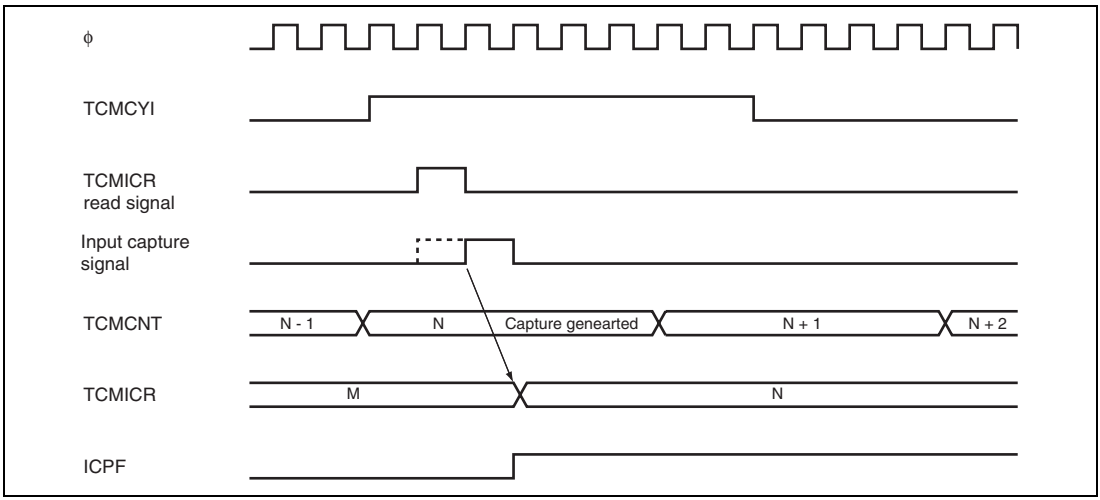


Figure 12.14 Conflict between TCMICR Read and Input Capture

12.6.4 Conflict between Edge Detection in Speed Measurement Mode and Writing to TCMMLCM

If the selected edge of TCMCYI is detected in the second half of a cycle of writing to the register (TCMMLCM) in speed measurement mode, the detected edge signal is delayed by one cycle of the system clock (ϕ).

Figure 12.15 shows the timing of this conflict.

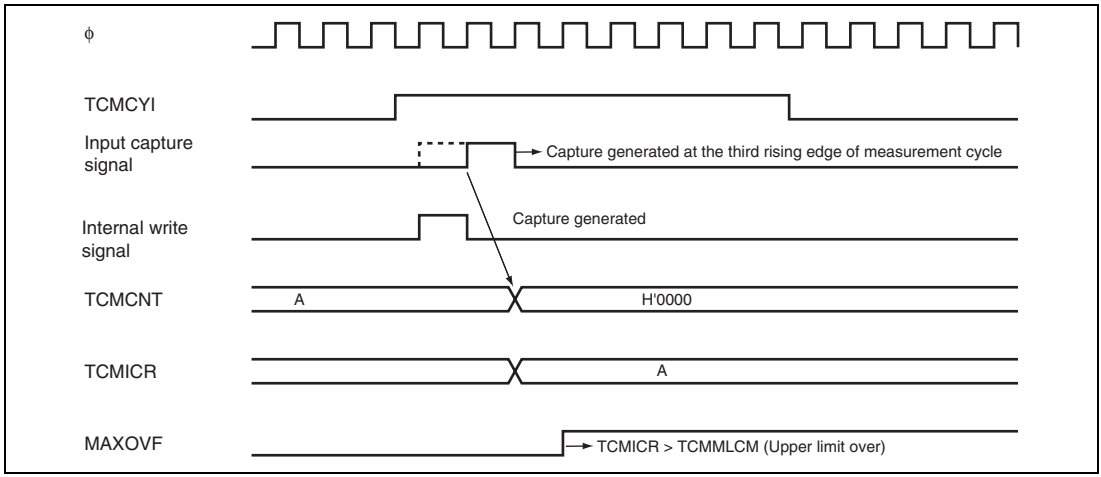


Figure 12.15 Conflict between Edge Detection and Register Write (Speed Measurement Mode)

12.6.5 Conflict between Edge Detection in Speed Measurement Mode and Clearing of TCMMDS Bit in TCMCR

If the CST bit in TCMCR is set to 1 in speed measurement mode, and the TCMMDS bit in TCMCR is cleared, but the selected edge from TCMCYI is detected at the same time, detection of the selected edge will cause the timer to continue to operate in speed measurement mode. The timer will not make the transition to timer mode until the next detection of the selected edge. Thus, ensure that the CST bit is cleared to 0 in speed measurement mode.

Figure 12.16 shows the timing of this conflict.

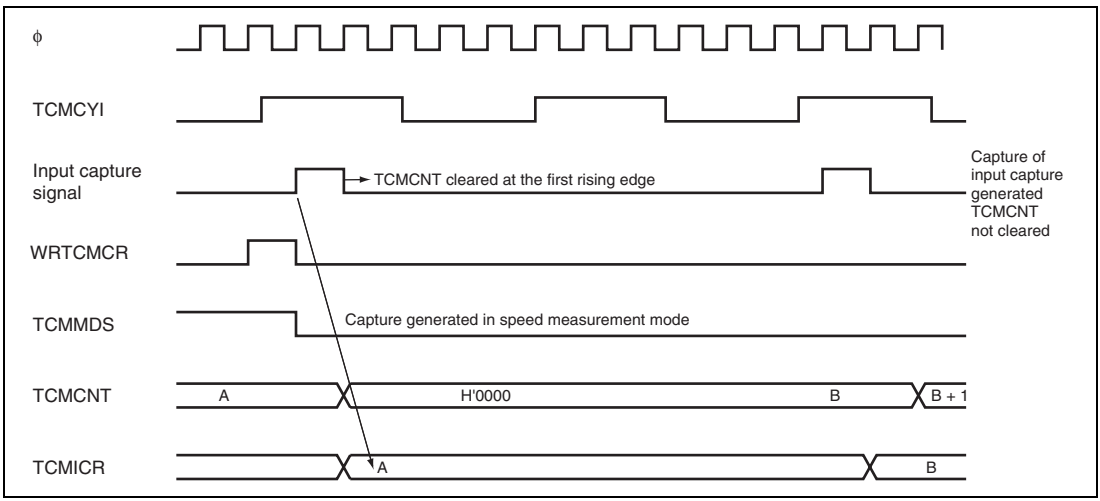


Figure 12.16 Conflict between Edge Detection and Clearing of TCMMDS (to Switch from Speed Measurement Mode to Timer Mode)

12.6.6 Setting for Module Stop Mode

The module-stop control register can be used to select either continuation or stoppage of TCM operation in module-stopped mode. The default setting is for TCM operation to stop. TCM registers become accessible on release from module-stopped mode. For details, see section 22, Power-Down Modes.

Section 13 8-Bit Timer (TMR)

This LSI has an on-chip 8-bit timer module (TMR_0, TMR_1, TMR_Y, and TMR_X) with four channels operating on the basis of an 8-bit counter. The 8-bit timer module can be used as a multifunction timer in a variety of applications, such as generation of counter reset, interrupt requests, and pulse output with an arbitrary duty cycle using a compare-match signal with two registers.

13.1 Features

- Selection of clock sources

  The counter input clock can be selected from six internal clocks and an external clock
- Selection of three ways to clear the counters

  The counters can be cleared on compare-match A, compare-match B, or by an external reset signal.
- Timer output controlled by two compare-match signals

  The timer output signal in each channel is controlled by two independent compare-match signals, enabling the timer to be used for various applications, such as the generation of pulse output or PWM output with an arbitrary duty cycle.
- Cascading of two channels
  - Cascading of TMR_0 and TMR_1

    Operation as a 16-bit timer can be performed using TMR_0 as the upper half and TMR_1 as the lower half (16-bit count mode).

    TMR_1 can be used to count TMR_0 compare-match occurrences (compare-match count mode).
  - Cascading of TMR_Y and TMR_X

    Operation as a 16-bit timer can be performed using TMR_Y as the upper half and TMR_X as the lower half (16-bit count mode).

    TMR_X can be used to count TMR_Y compare-match occurrences (compare-match count mode).
- Multiple interrupt sources for each channel

  TMR_0, TMR_1, and TMR_Y: Three types of interrupts: Compare-match A, compare-match B, and overflow

  TMR_X: Four types of interrupts: Compare-match A, compare match B, overflow, and input capture

Figures 13.1 and 13.2 show block diagrams of 8-bit timers.

An input capture function is added to TMR_X.

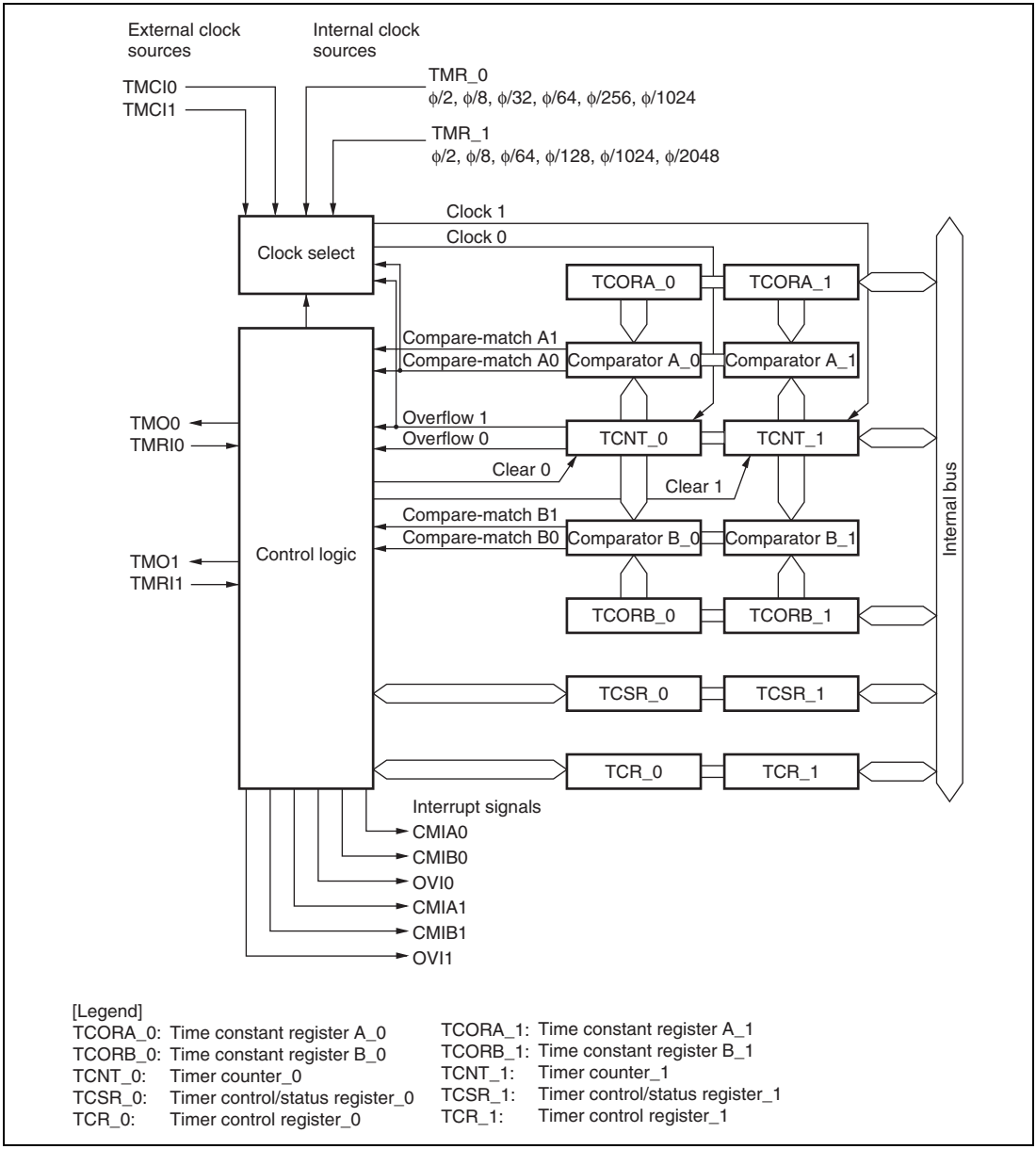
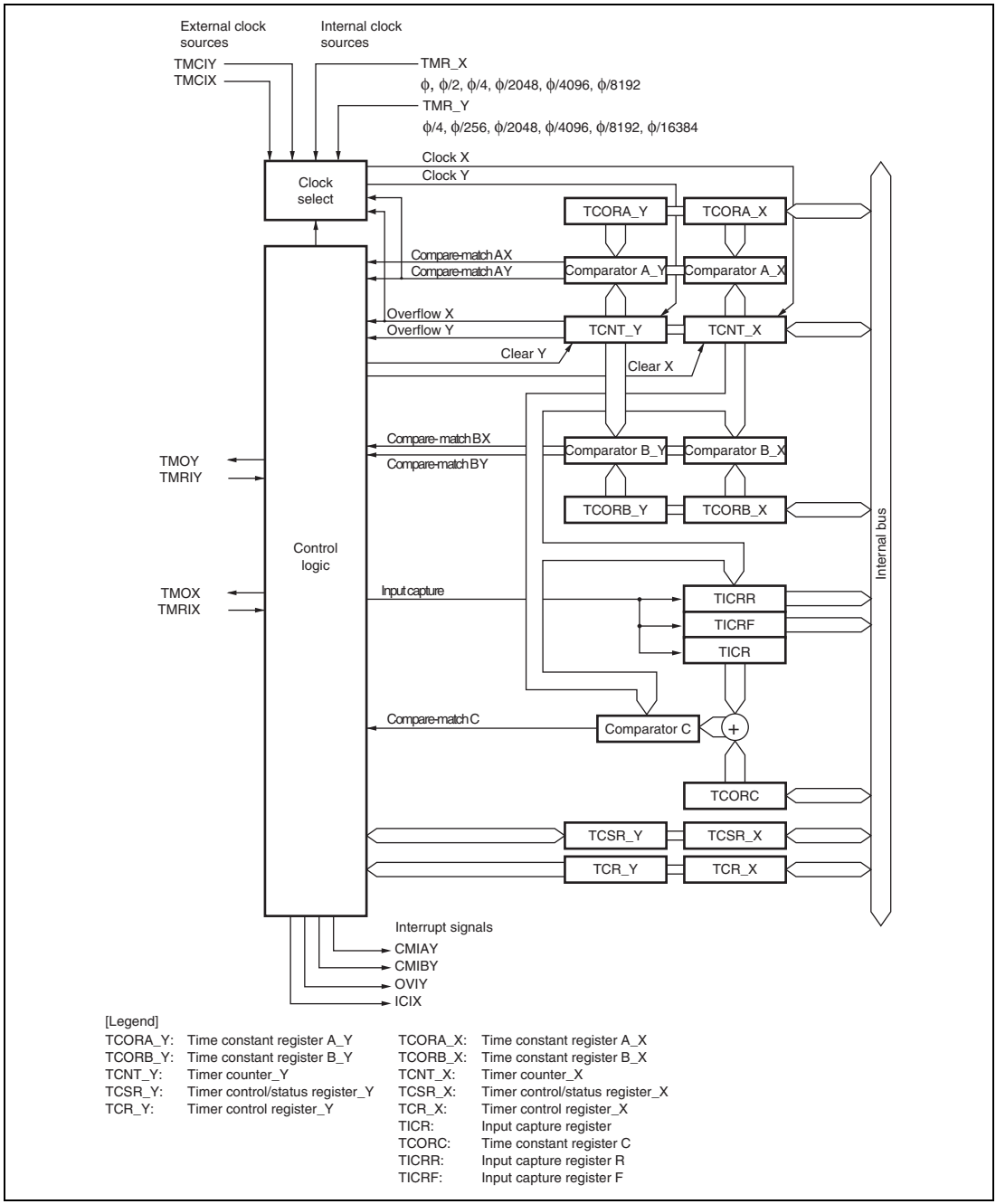


Figure 13.1 Block Diagram of 8-Bit Timer (TMR_0 and TMR_1)



[Legend]

TCORA_Y:	Time constant register A_Y	TCORA_X:	Time constant register A_X
TCORB_Y:	Time constant register B_Y	TCORB_X:	Time constant register B_X
TCNT_Y:	Timer counter_Y	TCNT_X:	Timer counter_X
TCSR_Y:	Timer control/status register_Y	TCSR_X:	Timer control/status register_X
TCR_Y:	Timer control register_Y	TCR_X:	Timer control register_X
		TICR:	Input capture register
		TCORC:	Time constant register C
		TICRR:	Input capture register R
		TICRF:	Input capture register F

Figure 13.2 Block Diagram of 8-Bit Timer (TMR_Y and TMR_X)

13.2 Input/Output Pins

Table 13.1 summarizes the input and output pins of the TMR.

Table 13.1 Pin Configuration

Channel	Name	Symbol	I/O	Function
TMR_0	Timer output	TMO0	Output	Output controlled by compare-match
	Timer clock input	TMCI0	Input	External clock input for the counter
	Timer reset input	TMRI0	Input	External reset input for the counter
TMR_1	Timer output	TMO1	Output	Output controlled by compare match
	Timer clock input	TMCI1	Input	External clock input for the counter
	Timer reset input	TMRI1	Input	External reset input for the counter
TMR_Y	Timer clock/reset input	TMIY (TMCIY/TMRIY)	Input	External clock input/external reset input for the counter
	Timer output	TMOY	Output	Output controlled by compare-match
TMR_X	Timer output	TMOX	Output	Output controlled by compare-match
	Timer clock/reset input	TMIX (TMCIX/TMRIX)	Input	External clock input/external reset input for the counter

13.3 Register Descriptions

The TMR has the following registers. For details on the serial timer control register, see section 3.2.3, Serial Timer Control Register (STCR).

TMR_0

- Timer counter_0 (TCNT_0)
- Time constant register A_0 (TCORA_0)
- Time constant register B_0 (TCORB_0)
- Timer control register_0 (TCR_0)
- Timer control/status register_0 (TCSR_0)

TMR_1

- Timer counter_1 (TCNT_1)
- Time constant register A_1 (TCORA_1)
- Time constant register B_1 (TCORB_1)
- Timer control register_1 (TCR_1)
- Timer control/status register_1 (TCSR_1)

TMR_Y

- Timer counter_Y (TCNT_Y)
- Time constant register A_Y (TCORA_Y)
- Time constant register B_Y (TCORB_Y)
- Timer control register_Y (TCR_Y)
- Timer control/status register_Y (TCSR_Y)
- Timer connection register S (TCONRS)

TMR_X

- Timer counter_X (TCNT_X)
- Time constant register A_X (TCORA_X)
- Time constant register B_X (TCORB_X)
- Timer control register_X (TCR_X)
- Timer control/status register_X (TCSR_X)
- Input capture register (TICR)
- Time constant register (TCORC)
- Input capture register R (TICRR)
- Input capture register F (TICRF)

- Timer connection register I (TCONRI)

For both TMR_Y and TMR_X

- Timer XY control register (TCRXY)

Note: Some of the registers of TMR_X and TMR_Y use the same address. The registers can be switched by the TMRX/Y bit in TCONRS.
TCNT_Y, TCORA_Y, TCORB_Y, and TCR_Y can be accessed when the TMRX/Y bit in TCONRS is set to 1. TCNT_X, TCORA_X, TCORB_X, and TCR_X can be accessed when the TMRX/Y bit in TCONRS are cleared to 0.

13.3.1 Timer Counter (TCNT)

Each TCNT is an 8-bit readable/writable up-counter. TCNT_0 and TCNT_1 (or TCNT_X and TCNT_Y) comprise a single 16-bit register, so they can be accessed together by word access. The clock source is selected by the CKS2 to CKS0 bits in TCR. TCNT can be cleared by an external reset input signal, compare-match A signal or compare-match B signal. The method of clearing can be selected by the CCLR1 and CCLR0 bits in TCR. When TCNT overflows (changes from H'FF to H'00), the OVF bit in TCSR is set to 1. TCNT is initialized to H'00.

13.3.2 Time Constant Register A (TCORA)

TCORA is an 8-bit readable/writable register. TCORA_0 and TCORA_1 (or TCORA_X and TCORA_Y) comprise a single 16-bit register, so they can be accessed together by word access. TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding compare-match flag A (CMFA) in TCSR is set to 1. Note however that comparison is disabled during the T_2 state of a TCORA write cycle. The timer output from the TMO pin can be freely controlled by these compare-match A signals and the settings of output select bits OS1 and OS0 in TCSR. TCORA is initialized to H'FF.

13.3.3 Time Constant Register B (TCORB)

TCORB is an 8-bit readable/writable register. TCORB_0 and TCORB_1 (or TCORB_X and TCORB_Y) comprise a single 16-bit register, so they can be accessed together by word access. TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare-match flag B (CMFB) in TCSR is set to 1. Note however that comparison is disabled during the T_2 state of a TCORB write cycle. The timer output from the TMO pin can be freely controlled by these compare-match B signals and the settings of output select bits OS3 and OS2 in TCSR. TCORB is initialized to H'FF.

13.3.4 Timer Control Register (TCR)

TCR selects the TCNT clock source and the condition by which TCNT is cleared, and enables/disables interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	Compare-Match Interrupt Enable B Selects whether the CMFB interrupt request (CMIB) is enabled or disabled when the CMFB flag in TCSR is set to 1. 0: CMFB interrupt request (CMIB) is disabled 1: CMFB interrupt request (CMIB) is enabled
6	CMIEA	0	R/W	Compare-Match Interrupt Enable A Selects whether the CMFA interrupt request (CMIA) is enabled or disabled when the CMFA flag in TCSR is set to 1. 0: CMFA interrupt request (CMIA) is disabled 1: CMFA interrupt request (CMIA) is enabled
5	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects whether the OVF interrupt request (OVI) is enabled or disabled when the OVF flag in TCSR is set to 1. 0: OVF interrupt request (OVI) is disabled 1: OVF interrupt request (OVI) is enabled
4	CCLR1	0	R/W	Counter Clear 1, 0
3	CCLR0	0	R/W	These bits select the method by which the timer counter is cleared. 00: Clearing is disabled 01: Cleared on compare-match A 10: Cleared on compare-match B 11: Cleared on rising edge of external reset input
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	These bits select the clock input to TCNT and count condition, together with the ICKS1 and ICKS0 bits in STCR. For details, see table 13.2.
0	CKS0	0	R/W	

Table 13.2 Clock Input to TCNT and Count Condition (1)

	TCR			STCR		
Channel	**CKS2**	**CKS1**	**CKS0**	**ICKS1**	**ICKS0**	**Description**
TMR_0	0	0	0	—	—	Disables clock input
	0	0	1	—	0	Increments at falling edge of internal clock φ/8
	0	0	1	—	1	Increments at falling edge of internal clock φ/2
	0	1	0	—	0	Increments at falling edge of internal clock φ/64
	0	1	0	—	1	Increments at falling edge of internal clock φ/32
	0	1	1	—	0	Increments at falling edge of internal clock φ/1024
	0	1	1	—	1	Increments at falling edge of internal clock φ/256
	1	0	0	—	—	Increments at overflow signal from TCNT_1*
TMR_1	0	0	0	—	—	Disables clock input
	0	0	1	0	—	Increments at falling edge of internal clock φ/8
	0	0	1	1	—	Increments at falling edge of internal clock φ/2
	0	1	0	0	—	Increments at falling edge of internal clock φ/64
	0	1	0	1	—	Increments at falling edge of internal clock φ/128
	0	1	1	0	—	Increments at falling edge of internal clock φ/1024
	0	1	1	1	—	Increments at falling edge of internal clock φ/2048
	1	0	0	—	—	Increments at compare-match A from TCNT_0*

	TCR			STCR		
Channel	**CKS2**	**CKS1**	**CKS0**	**ICKS1**	**ICKS0**	**Description**
Common	1	0	1	—	—	Increments at rising edge of external clock
	1	1	0	—	—	Increments at falling edge of external clock
	1	1	1	—	—	Increments at both rising and falling edges of external clock

Note: * If the TMR_0 clock input is set as the TCNT_1 overflow signal and the TMR_1 clock input is set as the TCNT_0 compare-match signal simultaneously, a count-up clock cannot be generated. These settings should not be made.

Table 13.2 Clock Input to TCNT and Count Condition (2)

	TCR			TCRXY		
Channel	**CKS2**	**CKS1**	**CKS0**	**CKSX**	**CKSY**	**Description**
TMR_Y	0	0	0	—	0	Disables clock input
	0	0	1	—	0	Increments at ϕ/4
	0	1	0	—	0	Increments at ϕ/256
	0	1	1	—	0	Increments at ϕ/2048
	1	0	0	—	0	Disables clock input
	0	0	0	—	1	Disables clock input
	0	0	1	—	1	Increments at ϕ/4096
	0	1	0	—	1	Increments at ϕ/8192
	0	1	1	—	1	Increments at ϕ/16384
	1	0	0	—	1	Increments at overflow signal from TCNT_X*
	1	0	1	—	x	Increments at rising edge of external clock
	1	1	0	—	x	Increments at falling edge of external clock
	1	1	1	—	x	Increments at both rising and falling edges of external clock

	TCR			TCRXY		
Channel	**CKS2**	**CKS1**	**CKS0**	**CKSX**	**CKSY**	**Description**
TMR_X	0	0	0	0	—	Disables clock input
	0	0	1	0	—	Increments at ϕ
	0	1	0	0	—	Increments at $\phi/2$
	0	1	1	0	—	Increments at $\phi/4$
	1	0	0	0	—	Disables clock input
	0	0	0	1	—	Disables clock input
	0	0	1	1	—	Increments at $\phi/2048$
	0	1	0	1	—	Increments at $\phi/4096$
	0	1	1	1	—	Increments at $\phi/8192$
	1	0	0	1	—	Increments at compare-match A from TCNT_Y*
	1	0	1	x	—	Increments at rising edge of external clock
	1	1	0	x	—	Increments at falling edge of external clock
	1	1	1	x	—	Increments at both rising and falling edges of external clock

Note: * If the TMR_Y clock input is set as the TCNT_X overflow signal and the TMR_X clock input is set as the TCNT_Y compare-match signal simultaneously, a count-up clock cannot be generated. These settings should not be made.

[Legend]

x: Don't care

—: Invalid

13.3.5 Timer Control/Status Register (TCSR)

TCSR indicates the status flags and controls compare-match output.

- TCSR_0

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT_0 and TCORB_0 match [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_0 and TCORA_0 match [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_0 overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4	ADTE	0	R/W	A/D Trigger Enable Enables or disables A/D converter start requests by compare-match A. 0: A/D converter start requests by compare-match A are disabled 1: A/D converter start requests by compare-match A are enabled

Bit	Bit Name	Initial Value	R/W	Description
3	OS3	0	R/W	Output Select 3, 2
2	OS2	0	R/W	These bits specify how the TMO0 pin output level is to be changed by compare-match B of TCORB_0 and TCNT_0. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)
1	OS1	0	R/W	Output Select 1, 0
0	OS0	0	R/W	These bits specify how the TMO0 pin output level is to be changed by compare-match A of TCORA_0 and TCNT_0. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Note: * Only 0 can be written, for flag clearing.

- TCSR_1

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT_1 and TCORB_1 match [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_1 and TCORA_1 match [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA

Bit	Bit Name	Initial Value	R/W	Description
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_1 overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4	—	1	R	Reserved This bit is always read as 1 and cannot be modified.
3	OS3	0	R/W	Output Select 3, 2
2	OS2	0	R/W	These bits specify how the TMO1 pin output level is to be changed by compare-match B of TCORB_1 and TCNT_1. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)
1	OS1	0	R/W	Output Select 1, 0
0	OS0	0	R/W	These bits specify how the TMO1 pin output level is to be changed by compare-match A of TCORA_1 and TCNT_1. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Note: * Only 0 can be written, for flag clearing.

- TCSR_X

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT_X and TCORB_X match [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB

Bit	Bit Name	Initial Value	R/W	Description
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_X and TCORA_X match [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_X overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4	ICF	0	R/(W)*	Input Capture Flag [Setting condition] When a rising edge and falling edge is detected in the external reset signal in that order. [Clearing condition] Read ICF when ICF = 1, then write 0 in ICF
3	OS3	0	R/W	Output Select 3, 2
2	OS2	0	R/W	These bits specify how the TMOX pin output level is to be changed by compare-match B of TCORB_X and TCNT_X. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)
1	OS1	0	R/W	Output Select 1, 0
0	OS0	0	R/W	These bits specify how the TMOX pin output level is to be changed by compare-match A of TCORA_X and TCNT_X. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Note: * Only 0 can be written, for flag clearing.

- TCSR_Y

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT_Y and TCORB_Y match [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_Y and TCORA_Y match [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_Y overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4	ICIE	0	R/W	Input Capture Interrupt Enable Enables or disables the ICF interrupt request (ICIX) when the ICF bit in TCSR_X is set to 1. 0: ICF interrupt request (ICIX) is disabled 1: ICF interrupt request (ICIX) is enabled
3	OS3	0	R/W	Output Select 3, 2
2	OS2	0	R/W	These bits specify how the TMOY pin output level is to be changed by compare-match B of TCORB_Y and TCNT_Y. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Bit	Bit Name	Initial Value	R/W	Description
1	OS1	0	R/W	Output Select 1, 0
0	OS0	0	R/W	These bits specify how the TMOY pin output level is to be changed by compare-match A of TCORA_Y and TCNT_Y. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Note: * Only 0 can be written, for flag clearing.

13.3.6 Time Constant Register C (TCORC)

TCORC is an 8-bit readable/writable register. The sum of contents of TCORC and TICR is always compared with TCNT. When a match is detected, a compare-match C signal is generated. However, comparison at the T_2 state in the write cycle to TCORC and at the input capture cycle of TICR is disabled. TCORC is initialized to H'FF.

13.3.7 Input Capture Registers R and F (TICRR and TICRF)

TICRR and TICRF are 8-bit read-only registers. While the ICST bit in TCONRI is set to 1, the contents of TCNT are transferred at the rising edge and falling edge of the external reset input (TMRIX) in that order. The ICST bit is cleared to 0 when one capture operation ends. TICRR and TICRF are initialized to H'00.

13.3.8 Timer Connection Register I (TCONRI)

TCONRI controls the input capture function.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R/W	Reserved The initial value should not be changed.
4	ICST	0	R/W	Input Capture Start Bit TMR_X has input capture registers (TICRR and TICRF). TICRR and TICRF can measure the width of a pulse by means of a single capture operation under the control of the ICST bit. When a rising edge followed by a falling edge is detected on TMRIX after the ICST bit is set to 1, the contents of TCNT at those points are captured into TICRR and TICRF, respectively, and the ICST bit is cleared to 0. [Clearing condition] When a rising edge followed by a falling edge is detected on TMRIX [Setting condition] When 1 is written in ICST after reading ICST = 0
3 to 0	—	All 0	R/W	Reserved The initial values should not be modified.

13.3.9 Timer Connection Register S (TCONRS)

TCONRS selects whether to access TMR_X or TMR_Y registers.

Bit	Bit Name	Initial Value	R/W	Description
7	TMRX/Y	0	R/W	TMR_X/TMR_Y Access Select For details, see table 13.3. 0: The TMR_X registers are accessed at addresses H'(FF)FFF0 to H'(FF)FFF5 1: The TMR_Y registers are accessed at addresses H'(FF)FFF0 to H'(FF)FFF5
6 to 0	—	All 0	R/W	Reserved The initial values should not be modified.

Table 13.3 Registers Accessible by TMR_X/TMR_Y

TMRX/Y	H'FFF0	H'FFF1	H'FFF2	H'FFF3	H'FFF4	H'FFF5	H'FFF6	H'FFF7
0	TMR_X	TMR_X	TMR_X	TMR_X	TMR_X	TMR_X	TMR_X	TMR_X
	TCR_X	TCSR_X	TICRR	TICRF	TCNT	TCORC	TCORA_X	TCORB_X
1	TMR_Y	TMR_Y	TMR_Y	TMR_Y	TMR_Y	TMR_Y		
	TCR_Y	TCSR_Y	TCORA_Y	TCORB_Y	TCNT_Y			

13.3.10 Timer XY Control Register (TCRXY)

TCRXY selects the TMR_X and TMR_Y output pins and internal clock.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved The initial value should not be changed.
5	CKSX	0	R/W	TMR_X Clock Select For details about selection, see table 13.2.
4	CKSY	0	R/W	TMR_Y Clock Select For details about selection, see table 13.2.
3 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

13.4 Operation

13.4.1 Pulse Output

Figure 13.3 shows an example for outputting an arbitrary duty pulse.

1. Clear the CCLR1 bit in TCR to 0, and set the CCLR0 bit in TCR to 1 so that TCNT is cleared according to the compare match of TCORA.
2. Set the OS3 to OS0 bits in TCSR to B'0110 so that 1 is output according to the compare match of TCORA and 0 is output according to the compare match of TCORB.

According to the above settings, the waveforms with the TCORA cycle and TCORB pulse width can be output without the intervention of software.

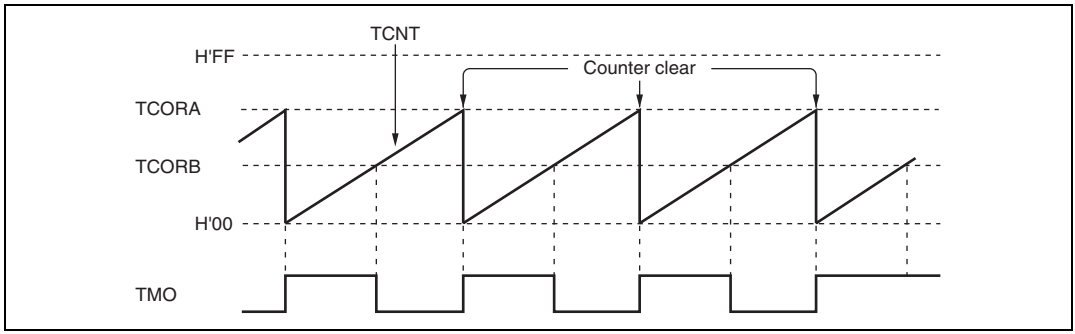


Figure 13.3 Pulse Output Example

13.5 Operation Timing

13.5.1 TCNT Count Timing

Figure 13.4 shows the TCNT count timing with an internal clock source. Figure 13.5 shows the TCNT count timing with an external clock source. The pulse width of the external clock signal must be at least 1.5 system clocks (ϕ) for a single edge and at least 2.5 system clocks (ϕ) for both edges. The counter will not increment correctly if the pulse width is less than these values.

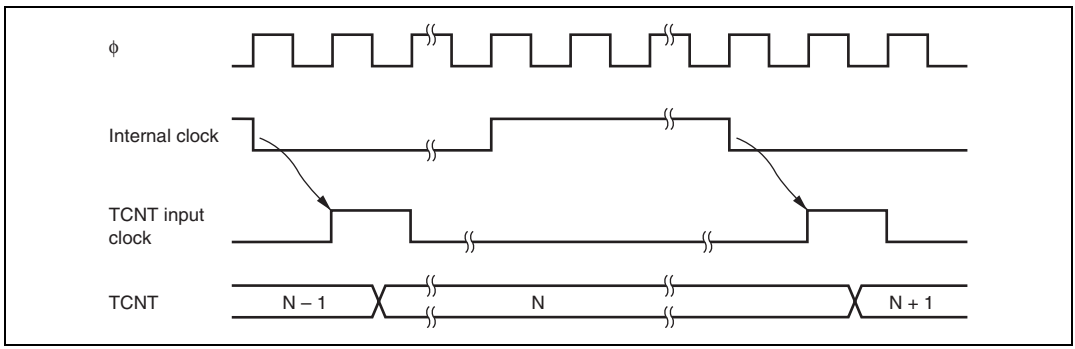


Figure 13.4 Count Timing for Internal Clock Input

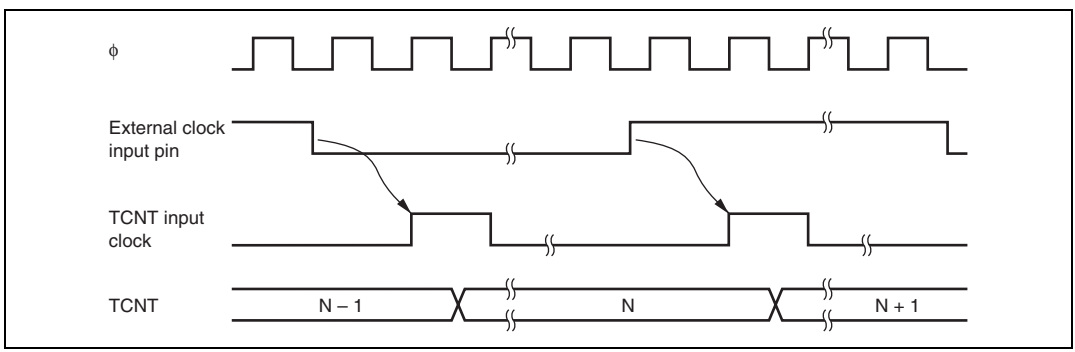


Figure 13.5 Count Timing for External Clock Input (Both Edges)

13.5.2 Timing of CMFA and CMFB Setting at Compare-Match

The CMFA and CMFB flags in TCSR are set to 1 by a compare-match signal generated when the TCNT and TCOR values match. The compare-match signal is generated at the last state in which the match is true, just when the timer counter is updated. Therefore, when TCNT and TCOR match, the compare-match signal is not generated until the next TCNT input clock. Figure 13.6 shows the timing of CMF flag setting.

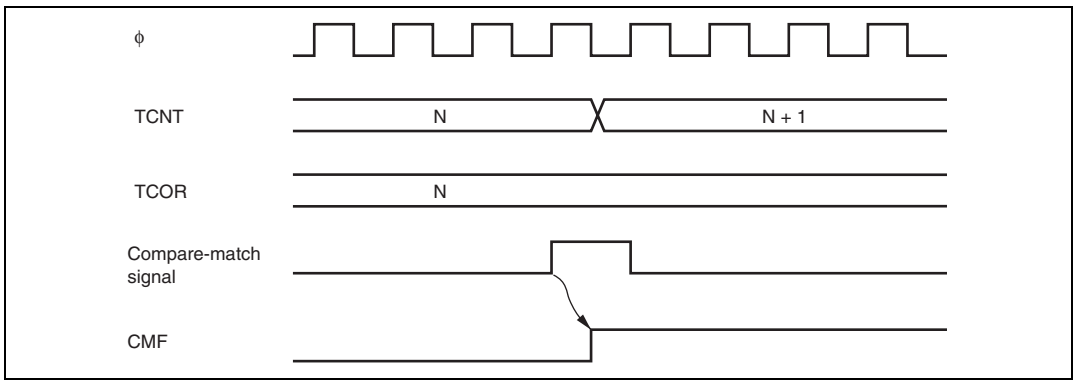


Figure 13.6 Timing of CMF Setting at Compare-Match

13.5.3 Timing of Timer Output at Compare-Match

When a compare-match signal occurs, the timer output changes as specified by the OS3 to OS0 bits in TCSR. Figure 13.7 shows the timing of timer output when the output is set to toggle by a compare-match A signal.

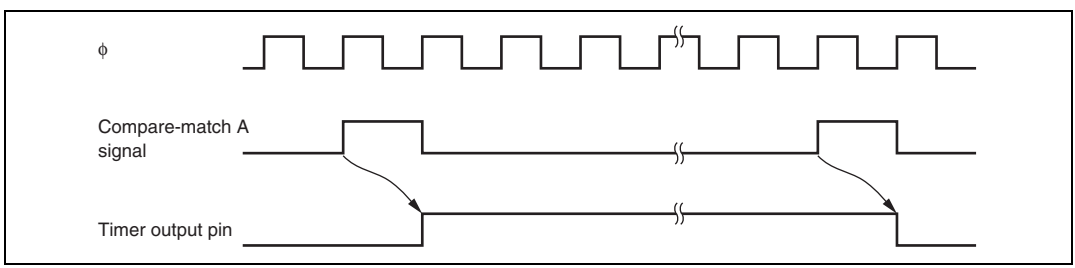


Figure 13.7 Timing of Toggled Timer Output by Compare-Match A Signal

13.5.4 Timing of Counter Clear at Compare-Match

TCNT is cleared when compare-match A or compare-match B occurs, depending on the setting of the CCLR1 and CCLR0 bits in TCR. Figure 13.8 shows the timing of clearing the counter by a compare-match.

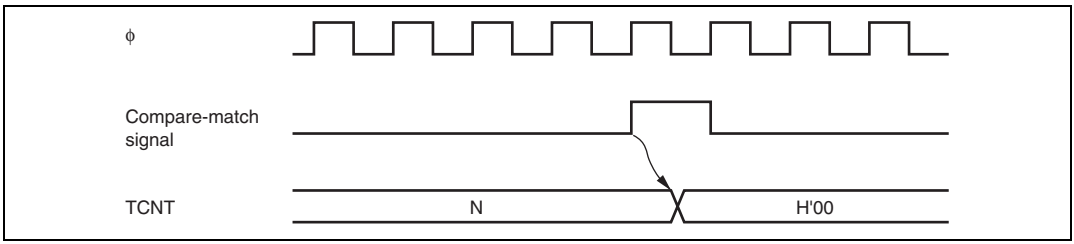


Figure 13.8 Timing of Counter Clear by Compare-Match

13.5.5 TCNT External Reset Timing

TCNT is cleared at the rising edge of an external reset input, depending on the settings of the CCLR1 and CCLR0 bits in TCR. The width of the clearing pulse must be at least 1.5 states. Figure 13.9 shows the timing of clearing the counter by an external reset input.

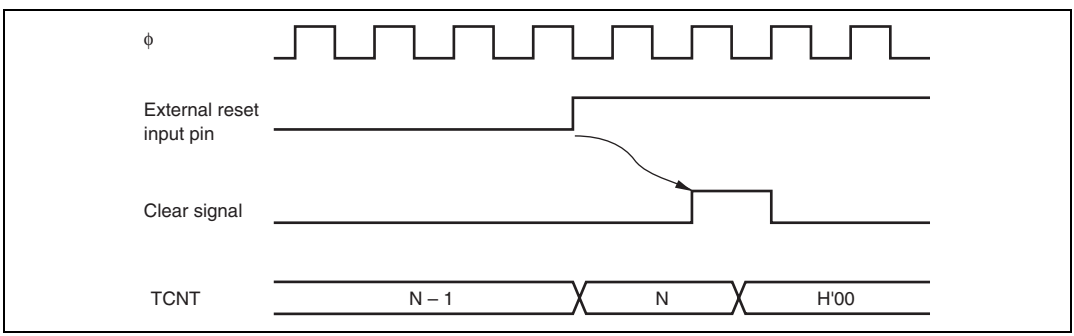


Figure 13.9 Timing of Counter Clear by External Reset Input

13.5.6 Timing of Overflow Flag (OVF) Setting

The OVF bit in TCSR is set to 1 when the TCNT overflows (changes from H'FF to H'00). Figure 13.10 shows the timing of OVF flag setting.

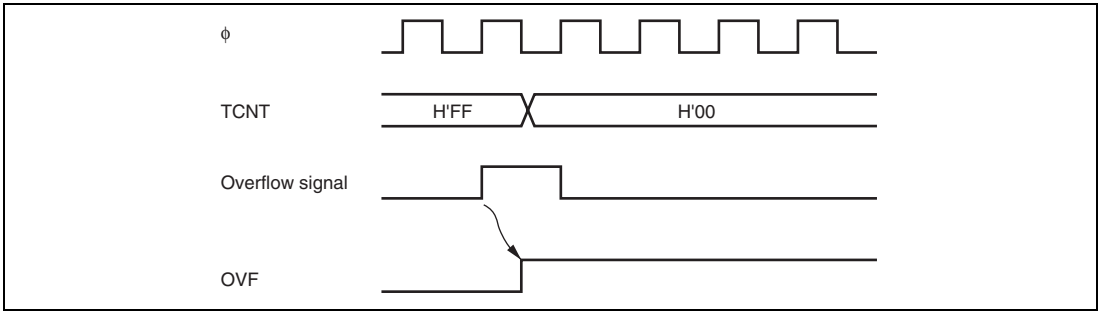


Figure 13.10 Timing of OVF Flag Setting

13.6 TMR_0 and TMR_1 Cascaded Connection

If bits CKS2 to CKS0 in either TCR_0 or TCR_1 are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, the 16-bit count mode or compare-match count mode is available.

13.6.1 16-Bit Count Mode

When bits CKS2 to CKS0 in TCR_0 are set to B'100, the timer functions as a single 16-bit timer with TMR_0 occupying the upper 8 bits and TMR_1 occupying the lower 8 bits.

(1) Setting of compare-match flags

- The CMF flag in TCSR_0 is set to 1 when a 16-bit compare-match occurs.
- The CMF flag in TCSR_1 is set to 1 when a lower 8-bit compare-match occurs.

(2) Counter clear specification

- If the CCLR1 and CCLR0 bits in TCR_0 have been set for counter clear at compare-match, the 16-bit counter (TCNT_0 and TCNT_1 together) is cleared when a 16-bit compare-match occurs. The 16-bit counter (TCNT_0 and TCNT_1 together) is also cleared when counter clear by the TMI0 pin has been set.
- The settings of the CCLR1 and CCLR0 bits in TCR_1 are ignored. The lower 8 bits cannot be cleared independently.

(3) Pin output

- Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR_0 is in accordance with the 16-bit compare-match conditions.
- Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR_1 is in accordance with the lower 8-bit compare-match conditions.

13.6.2 Compare-Match Count Mode

When bits CKS2 to CKS0 in TCR_1 are B'100, TCNT_1 counts the occurrence of compare-match A for TMR_0. TMR_0 and TMR_1 are controlled independently. Conditions such as setting of the CMF flag, generation of interrupts, output from the TMO pin, and counter clearing are in accordance with the settings for each or TMR_0 and TMR_1.

13.7 TMR_Y and TMR_X Cascaded Connection

If bits CKS2 to CKS0 in either TCR_Y or TCR_X are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, 16-bit count mode or compare-match count mode can be selected by the settings of the CKSX and CKSY bits in TCRXY.

13.7.1 16-Bit Count Mode

When bits CKS2 to CKS0 in TCR_Y are set to B'100 and the CKSY bit in TCRXY is set to 1, the timer functions as a single 16-bit timer with TMR_Y occupying the upper eight bits and TMR_X occupying the lower 8 bits.

(1) Setting of compare-match flags

- The CMF flag in TCSR_Y is set to 1 when an upper 8-bit compare-match occurs.
- The CMF flag in TCSR_X is set to 1 when a lower 8-bit compare-match occurs.

(2) Counter clear specification

- If the CCLR1 and CCLR0 bits in TCR_Y have been set for counter clear at compare-match, only the upper eight bits of TCNT_Y are cleared. The upper eight bits of TCNT_Y are also cleared when counter clear by the TMRIY pin has been set.
- The settings of the CCLR1 and CCLR0 bits in TCR_X are enabled, and the lower 8 bits of TCNT_X can be cleared by the counter.

(3) Pin output

- Control of output from the TMOY pin by bits OS3 to OS0 in TCSR_Y is in accordance with the upper 8-bit compare-match conditions.
- Control of output from the TMOX pin by bits OS3 to OS0 in TCSR_X is in accordance with the lower 8-bit compare-match conditions.

13.7.2 Compare-Match Count Mode

When bits CKS2 to CKS0 in TCR_X are set to B'100 and the CKSX bit in TCRXY is set to 1, TCNT_X counts the occurrence of compare-match A for TMR_Y. TMR_X and TMR_Y are controlled independently. Conditions such as setting of the CMF flag, generation of interrupts, output from the TMO pin, and counter clearing are in accordance with the settings for each channel.

13.7.3 Input Capture Operation

TMR_X has input capture registers (TICRR and TICRF). A narrow pulse width can be measured with TICRR and TICRF, using a single capture. If the falling edge of TMRIX (TMR_X input capture input signal) is detected after its rising edge has been detected, the value of TCNT_X at that time is transferred to both TICRR and TICRF.

(1) Input Capture Signal Input Timing

Figure 13.11 shows the timing of the input capture operation.

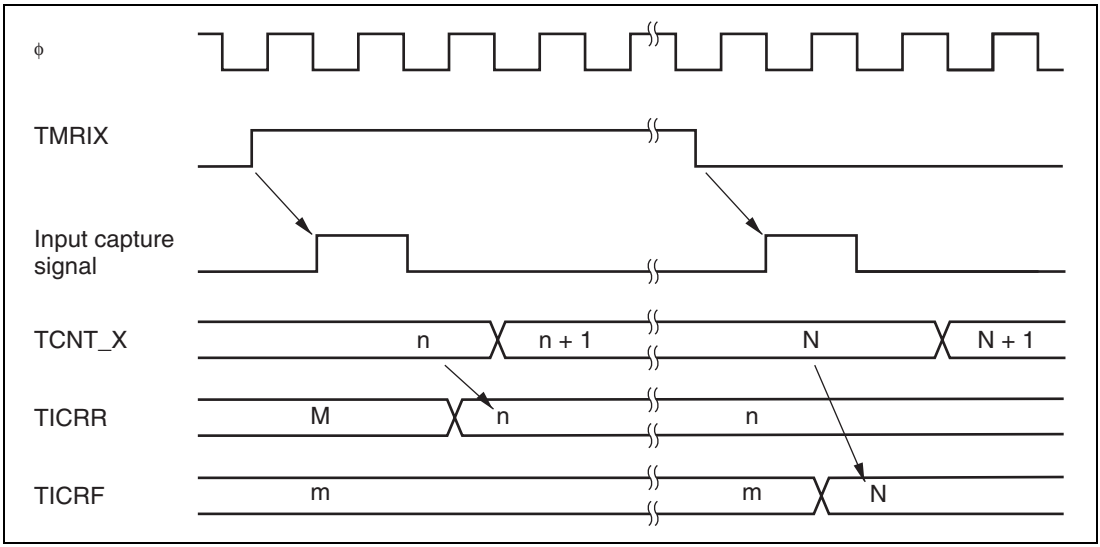


Figure 13.11 Timing of Input Capture Operation

If the input capture signal is input while TICRR and TICRF are being read, the input capture signal is delayed by one system clock (ϕ) cycle. Figure 13.12 shows the timing of this operation.

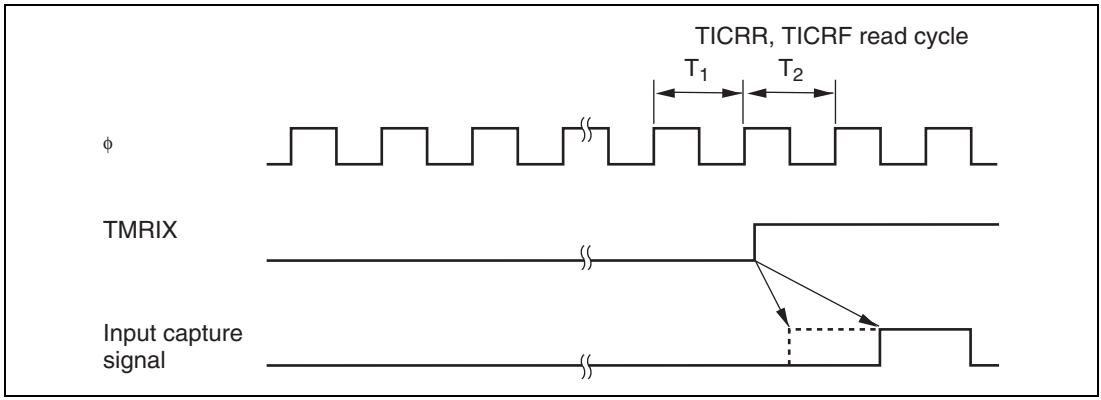


Figure 13.12 Timing of Input Capture Signal (Input capture signal is input during TICRR and TICRF read)

(2) Selection of Input Capture Signal Input

TMRIX (input capture input signal of TMR_X) is selected according to the setting of the ICST bit in TCONRI. The input capture signal selection is shown in table 13.4.

Table 13.4 Input Capture Signal Selection

TCONRI	
Bit 4	
ICST	Description
0	Input capture function not used
1	TMIX pin input selection

13.8 Interrupt Sources

TMR_0, TMR_1, and TMR_Y can generate three types of interrupts: CMIA, CMIB, and OVI. TMR_X can generate four types of interrupts: CMIA, CMIB, OVI, and ICIX. Table 13.5 shows the interrupt sources and priorities. Each interrupt source can be enabled or disabled independently by interrupt enable bits in TCR or TCSR. Independent signals are sent to the interrupt controller for each interrupt.

Table 13.5 Interrupt Sources of 8-Bit Timers TMR_0, TMR_1, TMR_Y, and TMR_X

Channel	Name	Interrupt Source	Interrupt Flag	Interrupt Priority
TMR_0	CMIA0	TCORA_0 compare-match	CMFA	High
	CMIB0	TCORB_0 compare-match	CMFB	
	OVI0	TCNT_0 overflow	OVF	
TMR_1	CMIA1	TCORA_1 compare-match	CMFA	
	CMIB1	TCORB_1 compare-match	CMFB	
	OVI1	TCNT_1 overflow	OVF	
TMR_Y	CMIAY	TCORA_Y compare-match	CMFA	
	CMIBY	TCORB_Y compare-match	CMFB	
	OVIY	TCNT_Y overflow	OVF	
TMR_X	ICIX	Input capture	ICF	
	CMIAX	TCORA_X compare-match	CMFA	
	CMIBX	TCORB_X compare-match	CMFB	
	OVIX	TCNT_X overflow	OVF	Low

13.9 Usage Notes

13.9.1 Conflict between TCNT Write and Counter Clear

If a counter clear signal is generated during the T_2 state of a TCNT write cycle as shown in figure 13.13, clearing takes priority and the counter write is not performed.

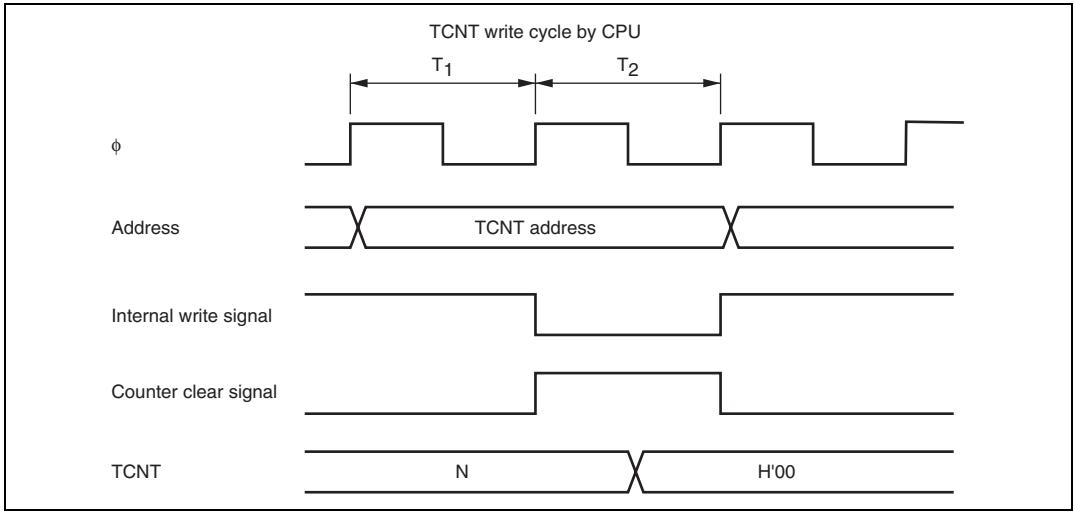


Figure 13.13 Conflict between TCNT Write and Clear

13.9.2 Conflict between TCNT Write and Count-Up

If a count-up occurs during the T_2 state of a TCNT write cycle as shown in figure 13.14, the counter write takes priority and the counter is not incremented.

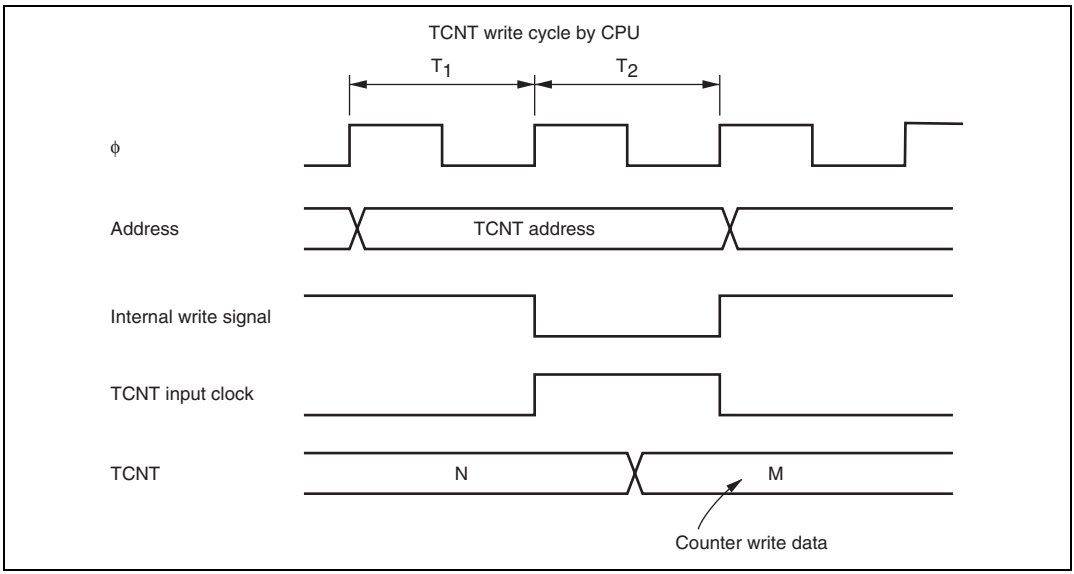


Figure 13.14 Conflict between TCNT Write and Count-Up

13.9.3 Conflict between TCOR Write and Compare-Match

If a compare-match occurs during the T_2 state of a TCOR write cycle as shown in figure 13.15, the TCOR write takes priority and the compare-match signal is disabled. With TMR_X, a TICR input capture conflicts with a compare-match in the same way as with a write to TCORC. In this case also, the input capture takes priority and the compare-match signal is disabled.

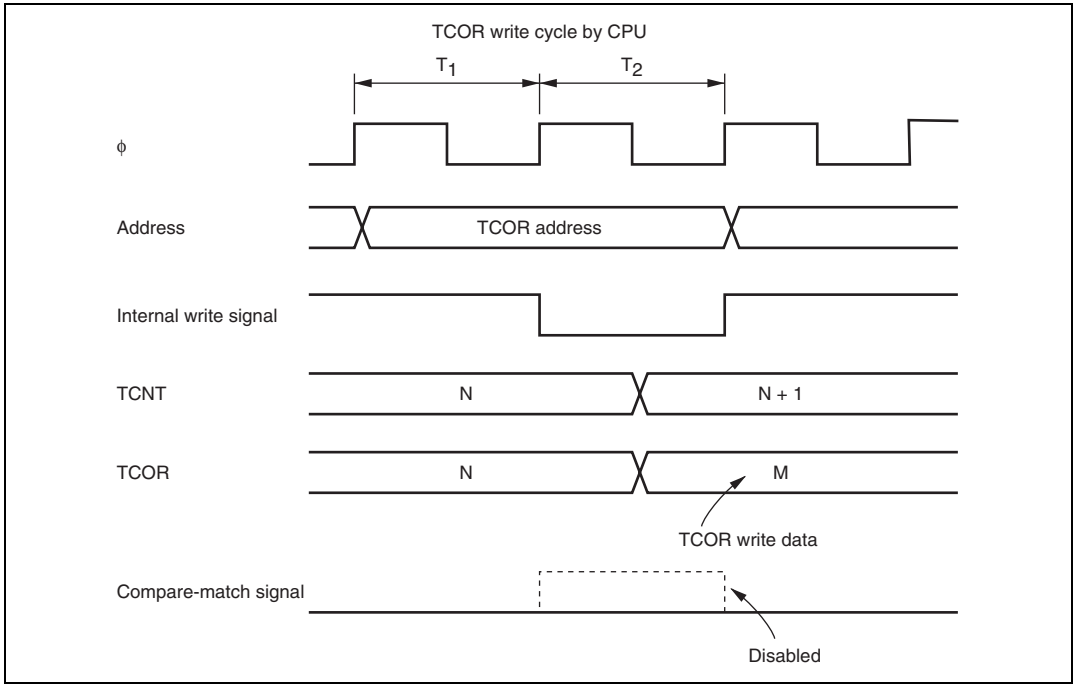


Figure 13.15 Conflict between TCOR Write and Compare-Match

13.9.4 Conflict between Compare-Matches A and B

If compare-matches A and B occur at the same time, the operation follows the output status that is defined for compare-match A or B, according to the priority of the timer output shown in table 13.6.

Table 13.6 Timer Output Priorities

Output Setting	Priority
Toggle output	High
1 output	
0 output	
No change	Low

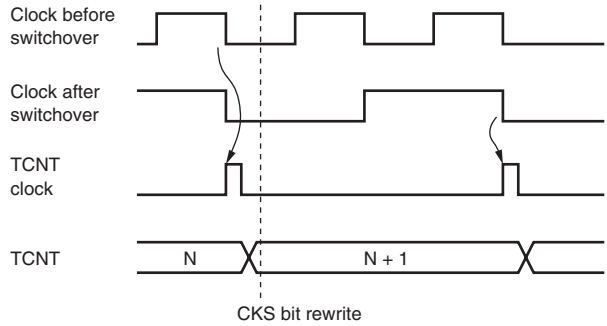
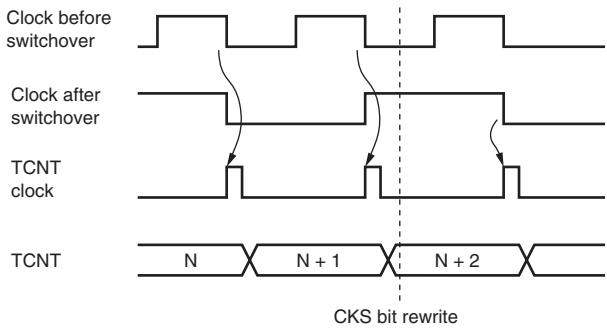
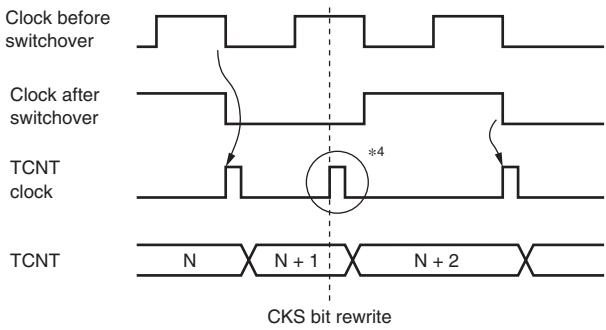
13.9.5 Switching of Internal Clocks and TCNT Operation

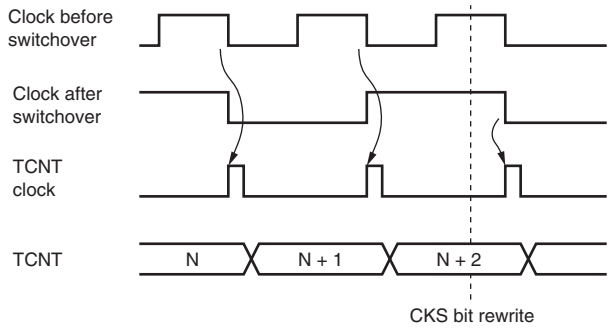
TCNT may increment erroneously when the internal clock is switched over. Table 13.7 shows the relationship between the timing at which the internal clock is switched (by writing to the CKS1 and CKS0 bits) and the TCNT operation.

When the TCNT clock is generated from an internal clock, the falling edge of the internal clock pulse is detected. If clock switching causes a change from high to low level, as shown in no. 3 in table 13.7, a TCNT clock pulse is generated on the assumption that the switchover is a falling edge, and TCNT is incremented.

Erroneous incrementation can also happen when switching between internal and external clocks.

Table 13.7 Switching of Internal Clocks and TCNT Operation

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
1	Clock switching from low to low level*[1]	
2	Clock switching from low to high level*[2]	
3	Clock switching from high to low level*[3]	

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
4	Clock switching from high to high level	

Notes: 1. Includes switching from low to stop, and from stop to low.
2. Includes switching from stop to high.
3. Includes switching from high to stop.
4. Generated on the assumption that the switchover is a falling edge; TCNT is incremented.

13.9.6 Mode Setting with Cascaded Connection

If the 16-bit count mode and compare-match count mode are set simultaneously, the input clock pulses for TCNT_0 and TCNT_1, and TCNT_X and TCNT_Y are not generated, and thus the counters will stop operating. Simultaneous setting of these two modes should therefore be avoided.

13.9.7 Module Stop Mode Setting

TMR operation can be enabled or disabled using the module stop control register. The initial setting is for TMR operation to be halted. Register access is enabled by canceling the module stop mode. For details, see section 22, Power-Down Modes.

Section 14 Watchdog Timer (WDT)

This LSI incorporates two watchdog timer channels (WDT_0 and WDT_1). The watchdog timer is an 8-bit timer that can generate an internal reset signal or an internal NMI interrupt signal if a system crash prevents the CPU from writing to the timer counter, thus allowing it to overflow.

When this watchdog 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. A block diagram of the WDT_0 and WDT_1 are shown in figure 14.1.

14.1 Features

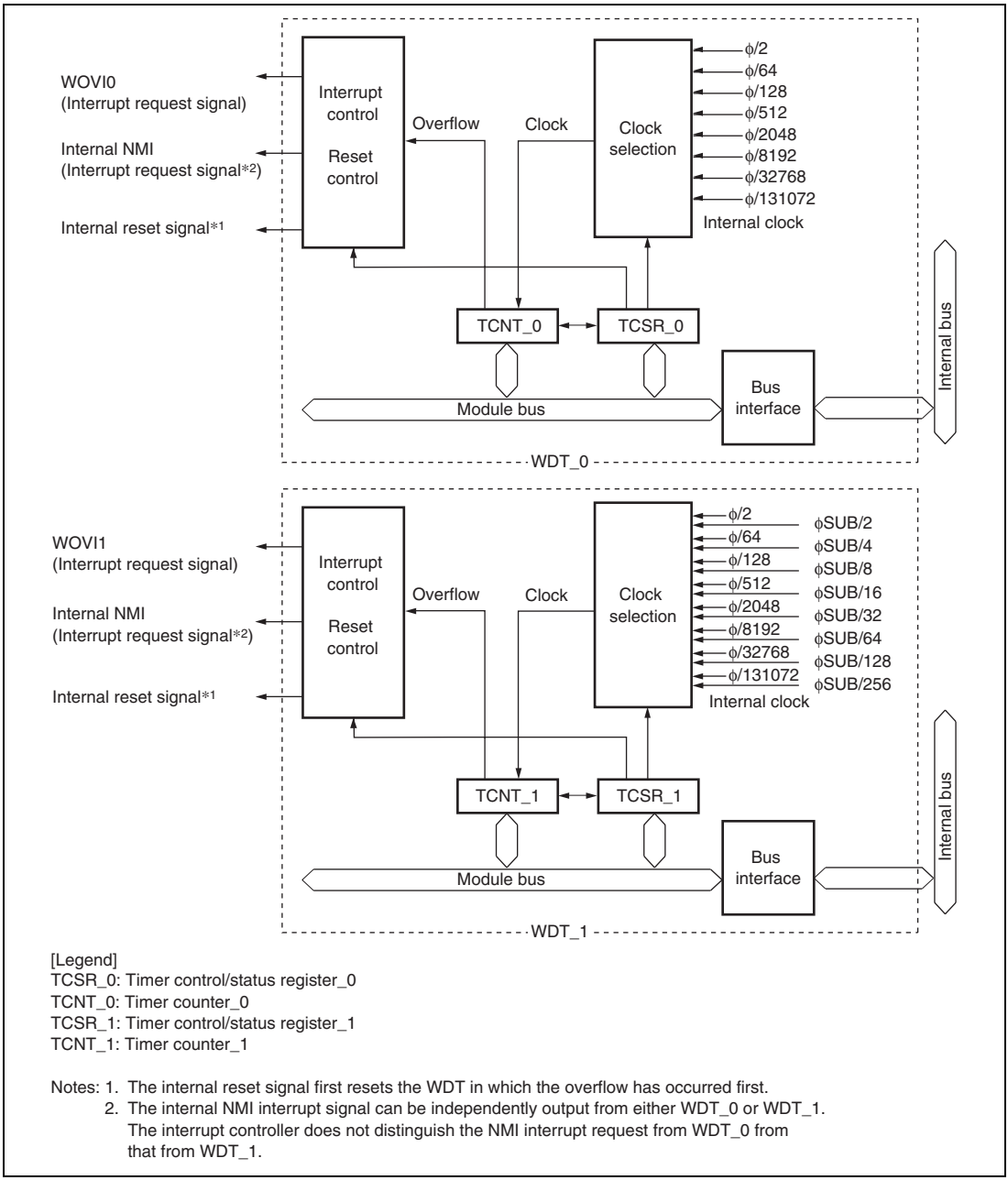
- Selectable from eight (WDT_0) or 16 (WDT_1) counter input clocks.
- Switchable between watchdog timer mode and interval timer mode

Watchdog Timer Mode:

- If the counter overflows, an internal reset or an internal NMI interrupt is generated.

Internal Timer Mode:

- If the counter overflows, an internal timer interrupt (WOVI) is generated.



[Legend]
TCSR_0: Timer control/status register_0
TCNT_0: Timer counter_0
TCSR_1: Timer control/status register_1
TCNT_1: Timer counter_1

Notes: 1. The internal reset signal first resets the WDT in which the overflow has occurred first.
2. The internal NMI interrupt signal can be independently output from either WDT_0 or WDT_1. The interrupt controller does not distinguish the NMI interrupt request from WDT_0 from that from WDT_1.

Figure 14.1 Block Diagram of WDT

14.2 Input/Output Pins

The WDT has the pins listed in table 14.1.

Table 14.1 Pin Configuration

Name	Symbol	I/O	Function
External sub-clock input pin	EXCL	Input	Inputs the clock pulses to the WDT_1 prescaler counter

14.3 Register Descriptions

The WDT has the following registers. To prevent accidental overwriting, TCSR and TCNT have to be written to in a method different from normal registers. For details, see section 14.6.1, Notes on Register Access. For details on the system control register, see section 3.2.2, System Control Register (SYSCR).

- Timer counter (TCNT)
- Timer control/status register (TCSR)

14.3.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter.

TCNT is initialized to H'00 when the TME bit in timer control/status register (TCSR) is cleared to 0.

14.3.2 Timer Control/Status Register (TCSR)

TCSR selects the clock source to be input to TCNT, and the timer mode.

- TCSR_0

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Overflow Flag Indicates that TCNT has overflowed (changes from H'FF to H'00). [Setting condition] When TCNT overflows (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset. [Clearing conditions] • When TCSR is read when OVF = 1, then 0 is written to OVF • When 0 is written to TME
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog timer or interval timer. 0: Interval timer mode 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting and is initialized to H'00.
4	—	0	R/W	Reserved The initial value should not be changed.
3	RST/$\overline{NMI}$	0	R/W	Reset or NMI Selects to request an internal reset or an NMI interrupt when TCNT has overflowed. 0: An NMI interrupt is requested 1: An internal reset is requested

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	Selects the clock source to be input to TCNT. The overflow frequency for ϕ = 20 MHz is enclosed in parentheses.
0	CKS0	0	R/W	000: ϕ/2 (frequency: 25.6 µs)
				001: ϕ/64 (frequency: 819.2 µs)
				010: ϕ/128 (frequency: 1.6 ms)
				011: ϕ/512 (frequency: 6.6 ms)
				100: ϕ/2048 (frequency: 26.2 ms)
				101: ϕ/8192 (frequency: 104.9 ms)
				110: ϕ/32768 (frequency: 419.4 ms)
				111: ϕ/131072 (frequency: 1.68 s)

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

- TCSR_1

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*[1]	Overflow Flag Indicates that TCNT has overflowed (changes from H'FF to H'00). [Setting condition] When TCNT overflows (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset. [Clearing conditions] • When TCSR is read when OVF = 1*[2], then 0 is written to OVF • When 0 is written to TME
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog timer or interval timer. 0: Interval timer mode 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting and is initialized to H'00.
4	PSS	0	R/W	Prescaler Select Selects the clock source to be input to TCNT. 0: Counts the divided cycle of ϕ–based prescaler (PSM) 1: Counts the divided cycle of ϕSUB–based prescaler (PSS)
3	RST/$\overline{NMI}$	0	R/W	Reset or NMI Selects to request an internal reset or an NMI interrupt when TCNT has overflowed. 0: An NMI interrupt is requested 1: An internal reset is requested

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	Selects the clock source to be input to TCNT. The overflow cycle for ϕ = 20 MHz and ϕSUB = 32.768 kHz is enclosed in parentheses.
0	CKS0	0	R/W	
				When PSS = 0:
				000: ϕ/2 (frequency: 25.6 µs)
				001: ϕ/64 (frequency: 819.2 µs)
				010: ϕ/128 (frequency: 1.6 ms)
				011: ϕ/512 (frequency: 6.6 ms)
				100: ϕ/2048 (frequency: 26.2 ms)
				101: ϕ/8192 (frequency: 104.9 ms)
				110: ϕ/32768 (frequency: 419.4 ms)
				111: ϕ/131072 (frequency: 1.68 s)
				When PSS = 1:
				000: ϕSUB/2 (cycle: 15.6 ms)
				001: ϕSUB/4 (cycle: 31.3 ms)
				010: ϕSUB/8 (cycle: 62.5 ms)
				011: ϕSUB/16 (cycle: 125 ms)
				100: ϕSUB/32 (cycle: 250 ms)
				101: ϕSUB/64 (cycle: 500 ms)
				110: ϕSUB/128 (cycle: 1 s)
				111: ϕSUB/256 (cycle: 2 s)

Notes: 1. Only 0 can be written, to clear the flag.
2. When OVF is polled with the interval timer interrupt disabled, OVF = 1 must be read at least twice.

14.4 Operation

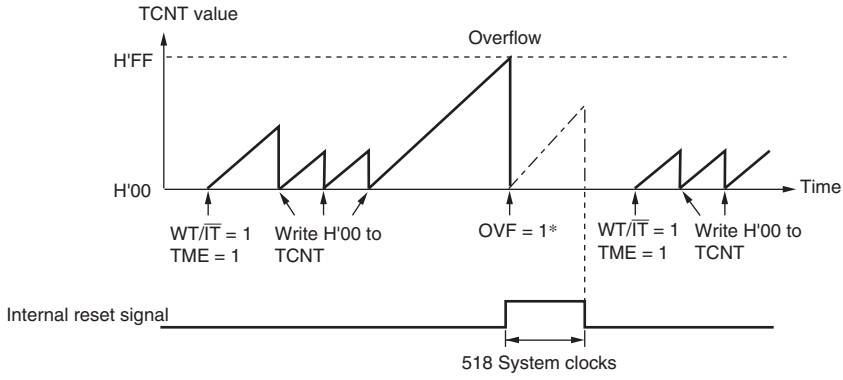
14.4.1 Watchdog Timer Mode

To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ bit and the TME bit in TCSR to 1. While the WDT is used as a watchdog timer, if TCNT overflows without being rewritten because of a system malfunction or another error, an internal reset or NMI interrupt request is generated. TCNT does not overflow while the system is operating normally. Software must prevent TCNT overflows by rewriting the TCNT value (normally be writing H'00) before overflows occurs.

If the RST/$\overline{NMI}$ bit of TCSR is set to 1, when the TCNT overflows, an internal reset signal for this LSI is issued for 518 system clocks as shown in figure 14.2. If the RST/$\overline{NMI}$ bit is cleared to 0, when the TCNT overflows, an NMI interrupt request is generated.

An internal reset request from the watchdog timer and a reset input from the $\overline{RES}$ pin are processed in the same vector. Reset source can be identified by the XRST bit status in SYSCR. If a reset caused by a signal input to the $\overline{RES}$ pin occurs at the same time as a reset caused by a WDT overflow, the $\overline{RES}$ pin reset has priority and the XRST bit in SYSCR is set to 1.

An NMI interrupt request from the watchdog timer and an interrupt request from the NMI pin are processed in the same vector. Do not handle an NMI interrupt request from the watchdog timer and an interrupt request from the NMI pin at the same time.



WT/$\overline{IT}$: Timer mode select bit
TME: Timer enable bit
OVF: Overflow flag

Note: * After the OVF bit becomes 1, it is cleared to 0 by an internal reset.
The XRST bit is also cleared to 0.

Figure 14.2 Watchdog Timer Mode (RST/$\overline{NMI}$ = 1) Operation

14.4.2 Interval Timer Mode

When the WDT is used as an interval timer, an interval timer interrupt (WOVI) is generated each time the TCNT overflows, as shown in figure 14.3. Therefore, an interrupt can be generated at intervals. When the TCNT overflows in interval timer mode, an interval timer interrupt (WOVI) is requested at the same time the OVF flag of TCSR is set to 1. The timing is shown figure 14.4.

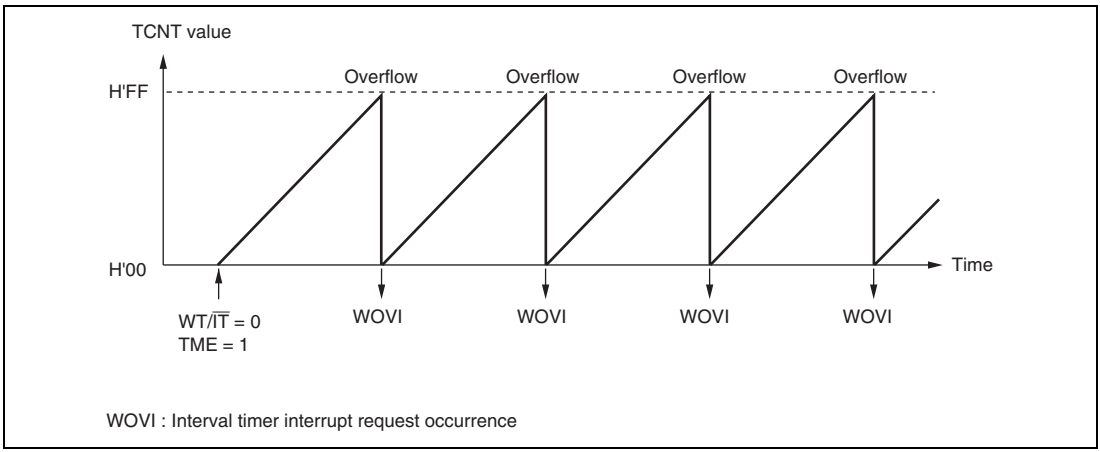


Figure 14.3 Interval Timer Mode Operation

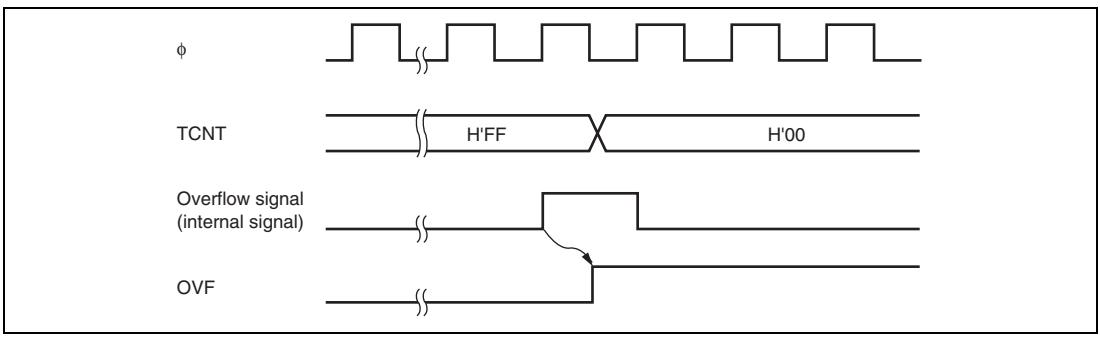


Figure 14.4 OVF Flag Set Timing

14.5 Interrupt Sources

During interval timer mode operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. OVF must be cleared to 0 in the interrupt handling routine.

When the NMI interrupt request is selected in watchdog timer mode, an NMI interrupt request is generated by an overflow

Table 14.2 WDT Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation
WOVI	TCNT overflow	OVF	Disable

14.6 Usage Notes

14.6.1 Notes on Register Access

The watchdog timer's registers, TCNT and TCSR differ from other registers in being more difficult to write to. The procedures for writing to and reading from these registers are given below.

(1) Writing to TCNT and TCSR (Example of WDT_0)

These registers must be written to by a word transfer instruction. They cannot be written to by a byte transfer instruction.

TCNT and TCSR both have the same write address. Therefore, satisfy the relative condition shown in figure 14.5 to write to TCNT or TCSR. To write to TCNT, the higher bytes must contain the value H'5A and the lower bytes must contain the write data before the transfer instruction execution. To write to TCSR, the higher bytes must contain the value H'A5 and the lower bytes must contain the write data.

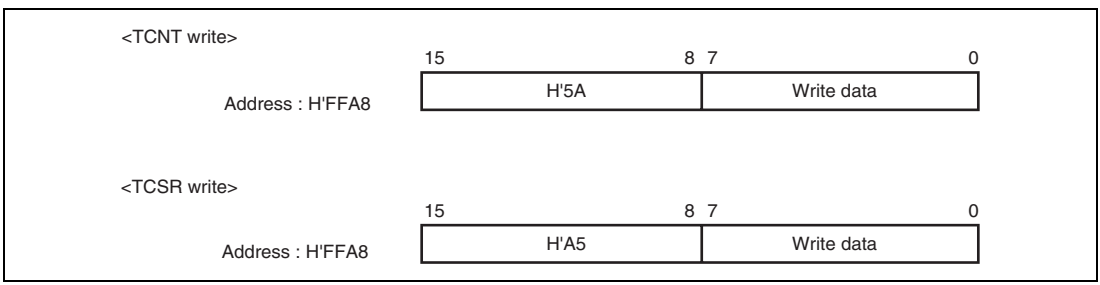


Figure 14.5 Writing to TCNT and TCSR (WDT_0)

(2) Reading from TCNT and TCSR (Example of WDT_0)

These registers are read in the same way as other registers. The read address is H'FFA8 for TCSR and H'FFA9 for TCNT.

14.6.2 Conflict between Timer Counter (TCNT) Write and Increment

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the write takes priority and the timer counter is not incremented. Figure 14.6 shows this operation.

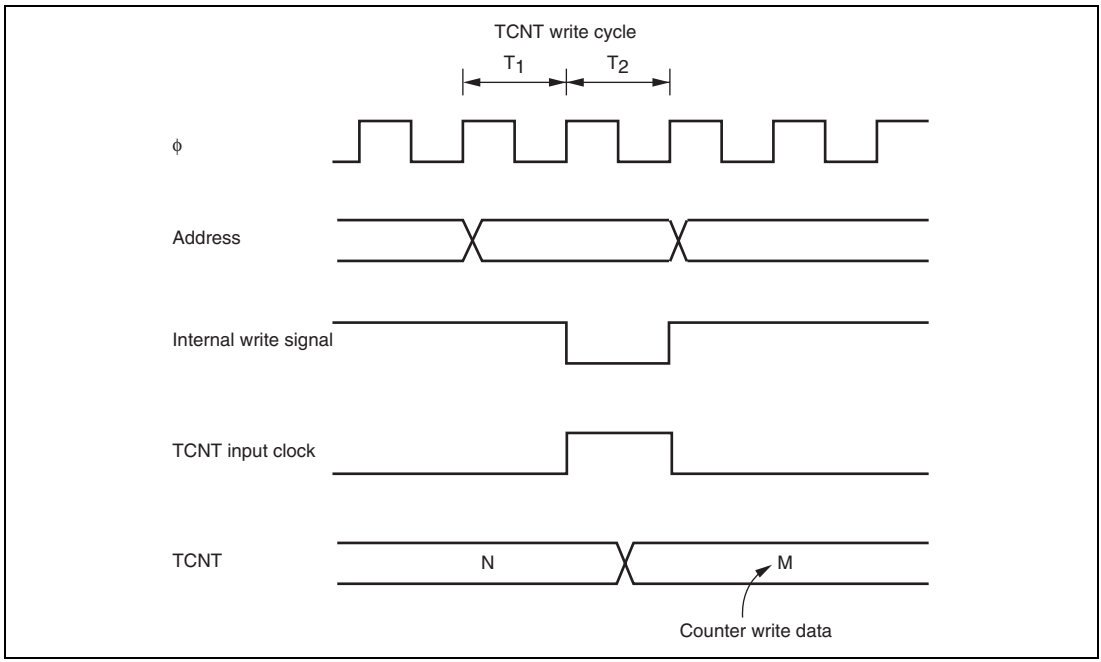


Figure 14.6 Conflict between TCNT Write and Increment

14.6.3 Changing Values of CKS2 to CKS0 Bits

If CKS2 to CKS0 bits in TCSR are written to while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before changing the values of CKS2 to CKS0 bits.

14.6.4 Changing Value of PSS Bit

If the PSS bit in TCSR_1 is written to while the WDT is operating, errors could occur in the operation. Stop the watchdog timer (by clearing the TME bit to 0) before changing the values of PSS bit.

14.6.5 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from/to watchdog timer to/from interval timer, while the WDT is operating, errors could occur in the operation. Software must stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

Section 15 Serial Communication Interface (SCI)

This LSI has two independent serial communication interface (SCI) channels. The SCI can handle both asynchronous and clocked synchronous serial communication. Asynchronous serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function) in asynchronous mode.

15.1 Features

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

  The transmitter and receiver are mutually independent, enabling transmission and 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.
- The on-chip baud rate generator allows any bit rate to be selected

  An external clock can be selected as a transfer clock source.
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode 7-bit data)
- Four interrupt sources

  Four interrupt sources — transmit-end, transmit-data-empty, receive-data-full, and receive error — that can issue requests.

  The transmit-data-empty and receive-data-full interrupt sources can activate the DTC.

Asynchronous Mode:

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

Clocked Synchronous Mode:

- Data length: 8 bits
- Receive error detection: Overrun errors
- Serial data communication with other LSIs that have the clock synchronized communication function

A block diagram of the SCI is shown in figure 15.1.

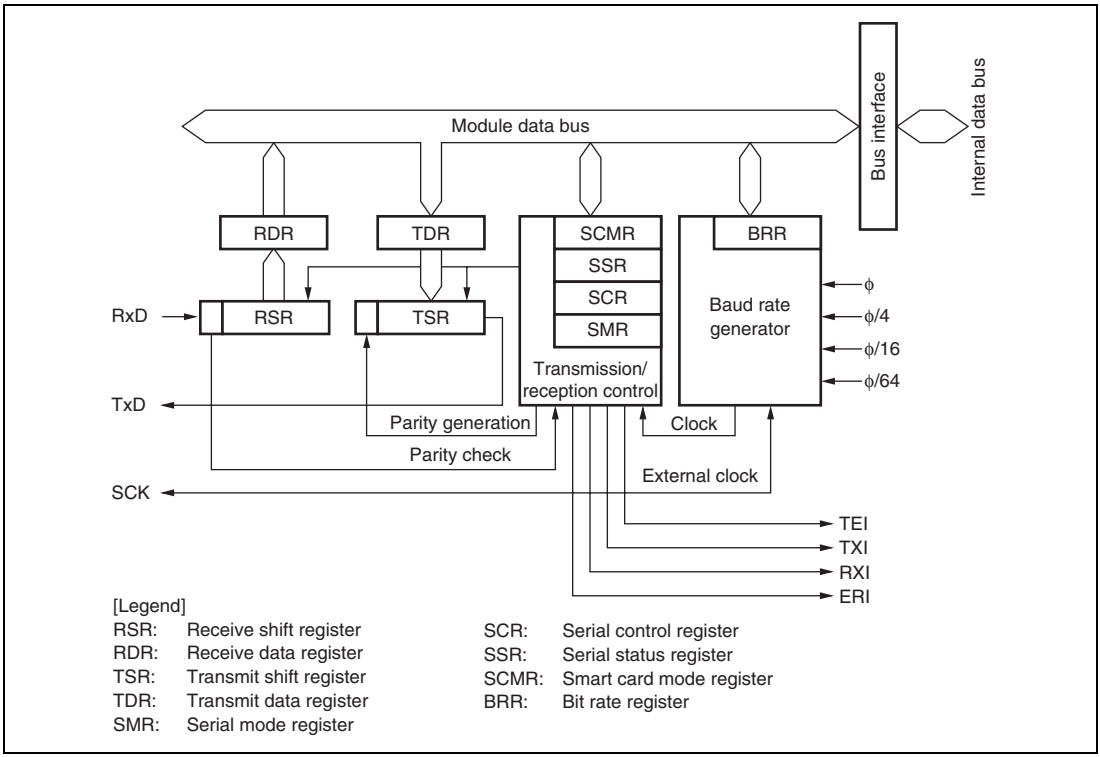


Figure 15.1 Block Diagram of SCI

15.2 Input/Output Pins

Table 15.1 shows the input/output pins for each SCI channel.

Table 15.1 Pin Configuration

Channel	Symbol*	Input/Output	Function
0	SCK0	Input/Output	Channel 0 clock input/output
	RxD0	Input	Channel 0 receive data input
	TxD0	Output	Channel 0 transmit data output
1	SCK1	Input/Output	Channel 1 clock input/output
	RxD1	Input	Channel 1 receive data input
	TxD1	Output	Channel 1 transmit data output

Note: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting the channel designation.

15.3 Register Descriptions

The SCI has the following registers for each channel.

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

15.3.1 Receive Shift Register (RSR)

RSR is a shift register used to receive serial data that converts it into parallel data. When one frame of data has been received, it is transferred to RDR automatically. RSR cannot be directly accessed by the CPU.

15.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores receive data. When the SCI has received one frame of serial data, it transfers the received serial data from RSR to RDR where it is stored. After this, RSR can receive the next data. Since RSR and RDR function as a double buffer in this way, continuous receive operations can be performed. After confirming that the RDRF bit in SSR is set to 1, read RDR for only once. RDR cannot be written to by the CPU. RDR is initialized to H'00.

15.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores transmit data. When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structures of TDR and TSR enables continuous serial transmission. If the next transmit data has already been written to TDR when one frame of data is transmitted, the SCI transfers the written data to TSR to continue transmission. Although TDR can be read from or written to by the CPU at all times, to achieve reliable serial transmission, write transmit data to TDR for only once after confirming that the TDRE bit in SSR is set to 1. TDR is initialized to H'FF.

15.3.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first transfers transmit data from TDR to TSR, then sends the data to the TxD pin. TSR cannot be directly accessed by the CPU.

15.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI's serial transfer format and select clock source for the on-chip baud rate generator. SMR can always be read from by the CPU. Writing to SMR by the CPU is enabled only in the initial setting, and not enabled during the receive, transmit, or transfer operation.

Bit	Bit Name	Initial Value	R/W	Description
7	C/$\overline{A}$	0	R/W	Communication Mode 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous mode) 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first is fixed and the MSB of TDR is not transmitted in transmission. In clocked synchronous mode, a fixed data length of 8 bits is used.
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. For a multiprocessor format, parity bit addition and checking are not performed regardless of the PE bit setting.
4	O/$\overline{E}$	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity.
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 In reception, only the first stop bit is checked. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

Bit	Bit Name	Initial Value	R/W	Description
2	MP	0	R/W	Multiprocessor Mode (enabled only in asynchronous mode) When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and O/$\overline{E}$ bit settings are invalid in multiprocessor mode.
1	CKS1	0	R/W	Clock Select 1,0
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/4 clock (n = 1) 10: ϕ/16 clock (n = 2) 11: ϕ/64 clock (n = 3) For the relation between the bit rate register setting and the baud rate, see section 15.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR.

15.3.6 Serial Control Register (SCR)

SCR is a register that performs enabling or disabling of SCI transfer operations and interrupt requests, and selection of the transfer clock source. SCR can always be read from by the CPU. Writing to SCR by the CPU is enabled only in the initial setting, and not enabled during the receive, transmit, or transfer operation.

For details on interrupt requests, refer to section 15.7, Interrupt Sources.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, a TXI interrupt request is enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled.

Bit	Bit Name	Initial Value	R/W	Description
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.
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 ORER status flags in SSR is disabled. 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 15.5, Multiprocessor Communication Function.
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, a TEI interrupt request is enabled.
1	CKE1	0	R/W	Clock Enable 1, 0
0	CKE0	0	R/W	These bits select the clock source and SCK pin function. Asynchronous mode 00: Internal clock (SCK pin functions as I/O port.) 01: Internal clock (Outputs a clock of the same frequency as the bit rate from the SCK pin.) 1X: External clock (Inputs a clock with a frequency 16 times the bit rate from the SCK pin.) Clocked synchronous mode 0X: Internal clock (SCK pin functions as clock output.) 1X: External clock (SCK pin functions as clock input.)

[Legend]

X: Don't care

15.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI and multiprocessor bits for transfer. TDRE, RDRF, ORER, PER, and FER can only be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Indicates whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and TDR is ready for data write [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When a TXI interrupt request is issued allowing the DTC to write data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that receive data is stored in RDR. [Setting condition] • When serial reception ends normally and receive data is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When an RXI interrupt request is issued allowing the DTC to read data from RDR The RDRF flag is not affected and retains its previous value when the RE bit in SCR is cleared to 0.
5	ORER	0	R/(W)*	Overrun Error [Setting condition] • When the next data is received while RDRF = 1 [Clearing condition] • When 0 is written to ORER after reading ORER = 1

Bit	Bit Name	Initial Value	R/W	Description
4	FER	0	R/(W)*	Framing Error [Setting condition] • When the stop bit is 0 [Clearing condition] • When 0 is written to FER after reading FER = 1 In 2-stop-bit mode, only the first stop bit is checked.
3	PER	0	R/(W)*	Parity Error [Setting condition] • When a parity error is detected during reception [Clearing condition] • When 0 is written to PER after reading PER = 1
2	TEND	1	R	Transmit End [Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When a TXI interrupt request is issued allowing the DTC to write data to TDR
1	MPB	0	R	Multiprocessor Bit MPB stores the multiprocessor bit in the receive frame. 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 frame.

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

15.3.8 Serial Interface Mode Register (SCMR)

SCMR selects SCI functions and its format. SCMR can always be read from by the CPU. Writing to SCMR by the CPU is enabled only in the initial setting, and not enabled during the receive, transmit, or transfer operation.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R	Reserved These bits are always read as 1 and cannot be modified.
3	SDIR	0	R/W	Data Transfer Direction Selects the serial/parallel conversion format. 0: TDR contents are transmitted with LSB-first. Receive data is stored as LSB first in RDR. 1: TDR contents are transmitted with MSB-first. Receive data is stored as MSB first in RDR. The SDIR bit is valid only when the 8-bit data format is used for transmission/reception; when the 7-bit data format is used, data is always transmitted/received with LSB-first.
2	SINV	0	R/W	Data Invert Specifies inversion of the data logic level. The SINV bit does not affect the logic level of the parity bit. When the parity bit is inverted, invert the O/$\overline{E}$ bit in SMR. 0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR. 1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR.
1	—	1	R	Reserved This bit is always read as 1 and cannot be modified.
0	SMIF	0	R/W	Serial Communication Interface Mode Select: 0: Normal asynchronous or clocked synchronous mode 1: Reserved mode

15.3.9 Bit Rate Register (BRR)

BRR is an 8-bit register that adjusts the bit rate. As the SCI performs baud rate generator control independently for each channel, different bit rates can be set for each channel. Table 15.2 shows the relationships between the N setting in BRR and bit rate B for normal asynchronous mode and clocked synchronous mode. The initial value of BRR is H'FF. BRR can always be read from by the CPU. Writing to BRR by the CPU is enabled only in the initial setting, and not enabled during the receive, transmit, or transfer operation.

Table 15.2 Relationships between N Setting in BRR and Bit Rate B

Mode	Bit Rate	Error
Asynchronous mode	$B = \frac{\phi \times 10^6}{64 \times 2^{2n-1} \times (N+1)}$	$\text{Error (\%)} = \{ \frac{\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \} \times 100$
Clocked synchronous mode	$B = \frac{\phi \times 10^6}{8 \times 2^{2n-1} \times (N+1)}$	—

[Legend]

B: Bit rate (bit/s)

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

ϕ: Operating frequency (MHz)

n: Determined by the SMR settings shown in the following table.

SMR Setting

CKS1	CKS0	n
0	0	0
0	1	1
1	0	2
1	1	3

Table 15.3 shows sample N settings in BRR in normal asynchronous mode. Table 15.4 shows the maximum bit rate settable for each frequency. Table 15.6 shows sample N settings in BRR in clocked synchronous mode. Tables 15.5 and 15.7 show the maximum bit rates with external clock input.

Table 15.3 BRR Settings for Various Bit Rates (Asynchronous Mode)

	Operating Frequency ϕ (MHz)											
	8			9.8304			10			12		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	–2.34

[Legend]

—: Can be set, but there will be a degree of error.

Note: * Make the settings so that the error does not exceed 1%.

	Operating Frequency ϕ (MHz)											
	12.288			14			14.7456			16		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	2	248	–0.17	3	64	0.70	3	70	0.03
150	2	159	0.00	2	181	0.16	2	191	0.00	2	207	0.16
300	2	79	0.00	2	90	0.16	2	95	0.00	2	103	0.16
600	1	159	0.00	1	181	0.16	1	191	0.00	1	207	0.16
1200	1	79	0.00	1	90	0.16	1	95	0.00	1	103	0.16
2400	0	159	0.00	0	181	0.16	0	191	0.00	0	207	0.16
4800	0	79	0.00	0	90	0.16	0	95	0.00	0	103	0.16
9600	0	39	0.00	0	45	–0.93	0	47	0.00	0	51	0.16
19200	0	19	0.00	0	22	–0.93	0	23	0.00	0	25	0.16
31250	0	11	2.40	0	13	0.00	0	14	–1.70	0	15	0.00
38400	0	9	0.00	—	—	—	0	11	0.00	0	12	0.16

[Legend]

—: Can be set, but there will be a degree of error.

Note: * Make the settings so that the error does not exceed 1%.

	Operating Frequency ϕ (MHz)											
	17.2032			18			19.6608			20		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	75	0.48	3	79	–0.12	3	86	0.31	3	88	–0.25
150	2	223	0.00	2	233	0.16	2	255	0.00	3	64	0.16
300	2	111	0.00	2	116	0.16	2	127	0.00	2	129	0.16
600	1	223	0.00	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	111	0.00	1	166	0.16	1	127	0.00	1	129	0.16
2400	0	223	0.00	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	111	0.00	0	166	0.16	0	127	0.00	0	129	0.16
9600	0	55	0.00	0	58	–0.69	0	63	0.00	0	64	0.16
19200	0	27	0.00	0	28	1.02	0	31	0.00	0	32	–1.36
31250	0	16	1.20	0	17	0.00	0	19	–1.70	0	19	0.00
38400	0	16	0.00	0	14	–2.34	0	15	0.00	0	15	1.73

[Legend]

—: Can be set, but there will be a degree of error.

Note: * Make the settings so that the error does not exceed 1%.

Table 15.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
19.6608	614400	0	0
20	625000	0	0

Table 15.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
19.6608	4.9152	307200
20	5.0000	312500

Table 15.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

Bit Rate (bit/s)	Operating Frequency ϕ (MHz)							
	8		10		16		20	
	n	N	n	N	n	N	n	N
110								
250	3	124	—	—	3	249		
500	2	249	—	—	3	124	—	—
1k	2	124	—	—	2	249	—	—
2.5k	1	199	1	249	2	99	2	124
5k	1	99	1	124	1	199	1	249
10k	0	199	0	249	1	99	1	124
25k	0	79	0	99	0	159	0	199
50k	0	39	0	49	0	79	0	99
100k	0	19	0	24	0	39	0	49
250k	0	7	0	9	0	15	0	19
500k	0	3	0	4	0	7	0	9
1M	0	1			0	3	0	4
2.5M			0	0*			0	1
5M							0	0*

[Legend]

Blank: Cannot be set.

—: Can be set, but there will be a degree of error.

*: Continuous transfer or reception is not possible.

Table 15.7 Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
8	1.3333	1333333.3	16	2.6667	2666666.7
10	1.6667	1666666.7	18	3.0000	3000000.0
12	2.0000	2000000.0	20	3.3333	3333333.3
14	2.3333	2333333.3			

15.4 Operation in Asynchronous Mode

Figure 15.2 shows the general format for asynchronous serial communication. One frame consists of a start bit (low level), followed by transmit/receive data, a parity bit, and finally stop bits (high level). In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer and reception.

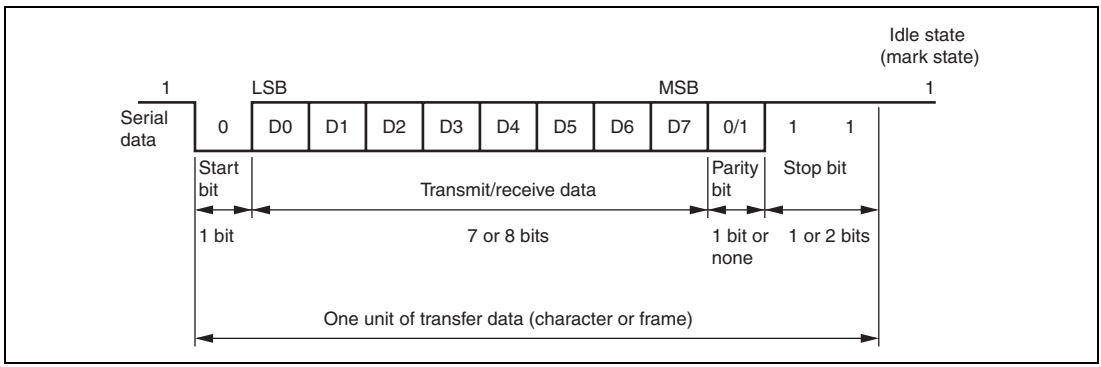


Figure 15.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

15.4.1 Data Transfer Format

Table 15.8 shows the data transfer formats that can be used in asynchronous mode. Any of 12 transfer formats can be selected according to the SMR setting. For details on the multiprocessor bit, refer to section 15.5, Multiprocessor Communication Function.

Table 15.8 Serial Transfer Formats (Asynchronous Mode)

SMR Settings				Serial Transmit/Receive Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	S 8-bit data STOP
0	0	0	1	S 8-bit data STOP STOP
0	1	0	0	S 8-bit data P STOP
0	1	0	1	S 8-bit data P STOP STOP
1	0	0	0	S 7-bit data STOP
1	0	0	1	S 7-bit data STOP STOP
1	1	0	0	S 7-bit data P STOP
1	1	0	1	S 7-bit data P STOP STOP
0	—	1	0	S 8-bit data MPB STOP
0	—	1	1	S 8-bit data MPB STOP STOP
1	—	1	0	S 7-bit data MPB STOP
1	—	1	1	S 7-bit data MPB STOP STOP

[Legend]
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

15.4.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a basic clock with a frequency of 16 times the bit rate. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Since receive data is latched internally at the rising edge of the 8th pulse of the basic clock, data is latched at the middle of each bit, as shown in figure 15.3. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \{ (0.5 - \frac{1}{2N}) - \frac{D - 0.5}{N} (1 + F) - (L - 0.5) F \} \times 100 \quad [\%] \quad \cdots \quad \text{Formula (1)}$$

M: Reception margin (%)
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 = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

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

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

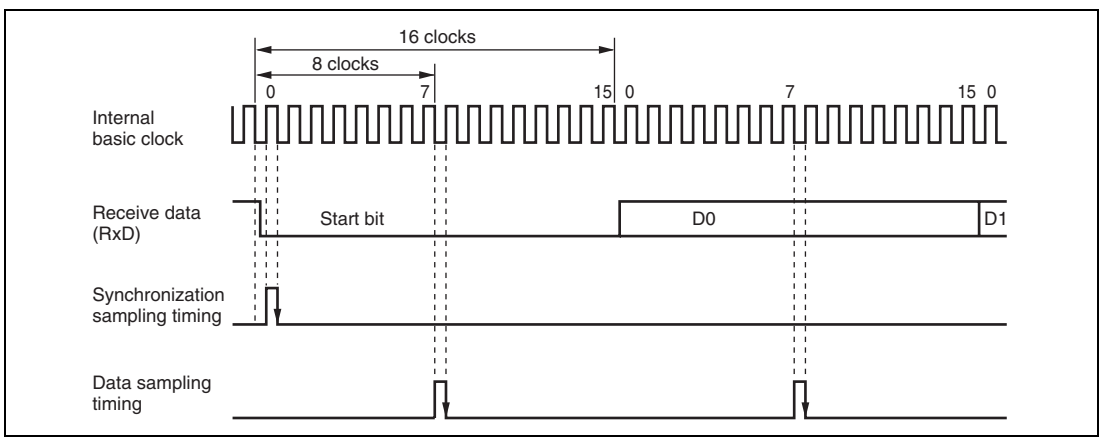


Figure 15.3 Receive Data Sampling Timing in Asynchronous Mode

15.4.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK pin can be selected as the SCI's transfer clock, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 15.4.

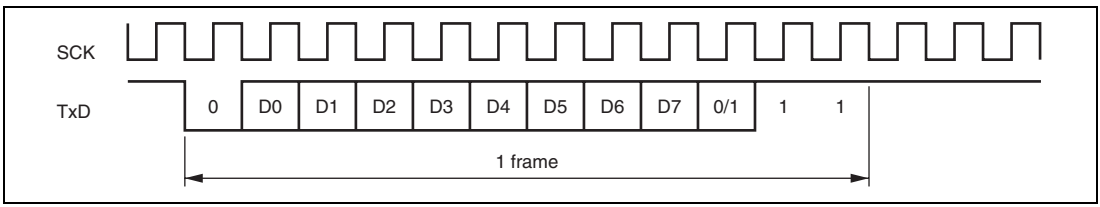


Figure 15.4 Relation between Output Clock and Transmit Data Phase (Asynchronous Mode)

15.4.4 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as shown in figure 15.5. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag in SSR is set to 1. Note that clearing the RE bit to 0 does not initialize the contents of the RDRF, PER, FER, and ORER flags in SSR, or the contents of RDR. When an external clock is used in asynchronous mode, the clock must be supplied even during initialization.

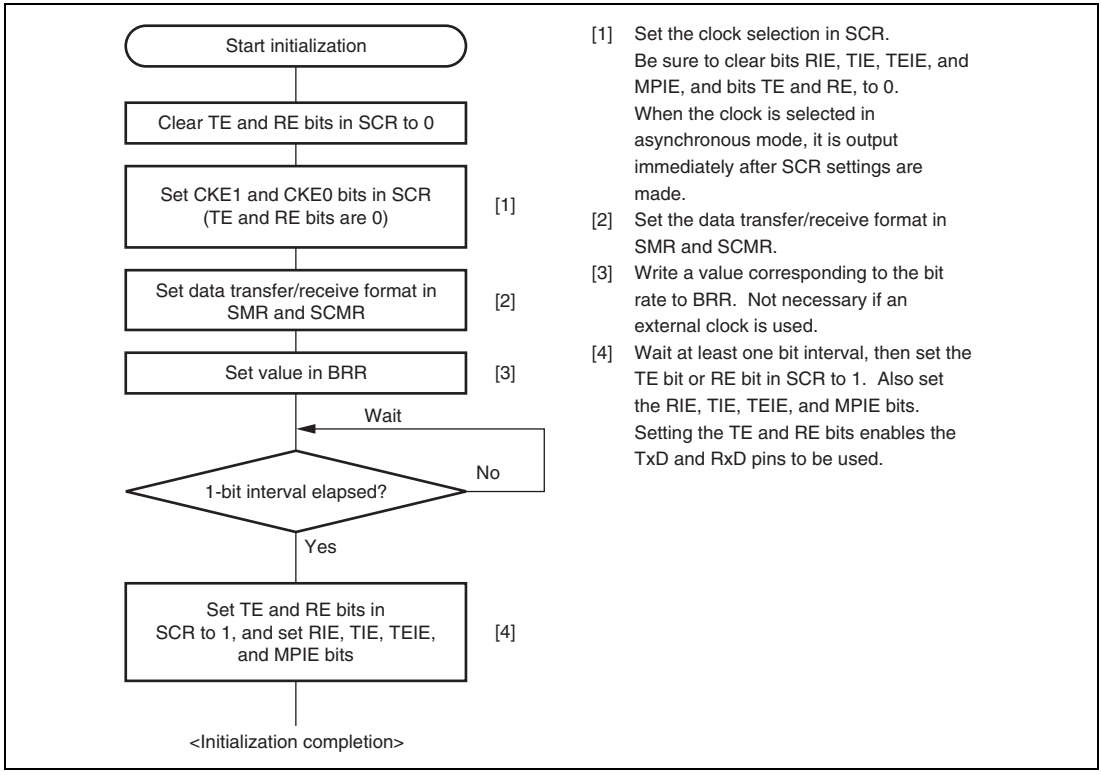


Figure 15.5 Sample SCI Initialization Flowchart

15.4.5 Data Transmission (Asynchronous Mode)

Figure 15.6 shows an example of the operation for transmission in asynchronous mode. In transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is cleared to 0, 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 SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a transmit data empty interrupt request (TXI) is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit or multiprocessor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks the TDRE flag at the timing for sending the stop bit.
5. 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.
6. 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 TEI interrupt request is generated.

Figure 15.7 shows a sample flowchart for transmission in asynchronous mode.

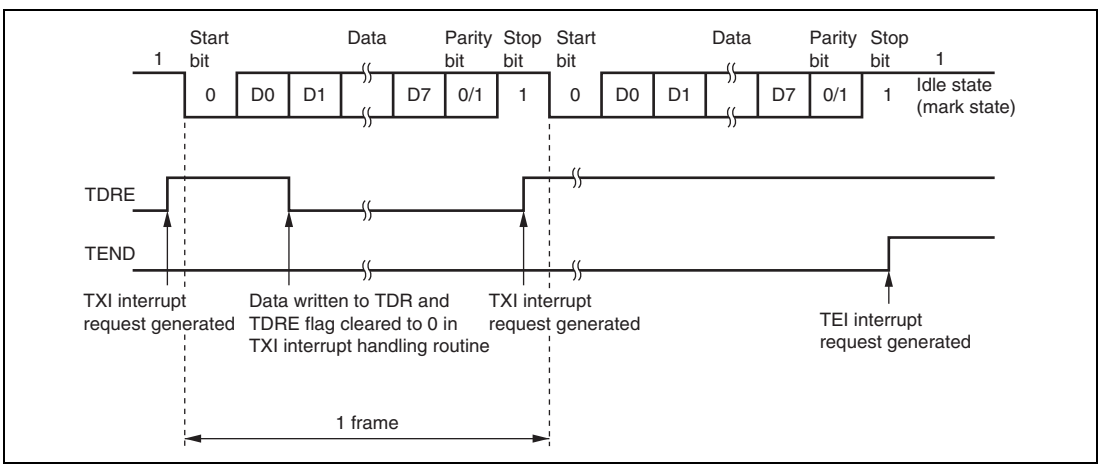
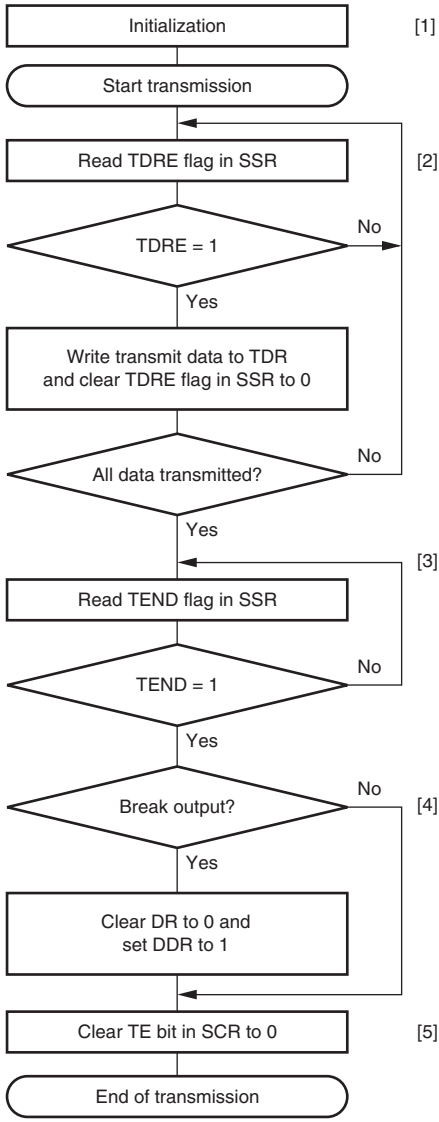


Figure 15.6 Example of SCI Transmit Operation in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and clear the TDRE flag to 0.
However, the TDRE flag is checked and cleared automatically when the DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Note: The SMR, SCR, SCMR, and BRR registers should not be written to during the period from the start to the end of transmission. This does not apply to the processing at step [5].

Figure 15.7 Sample Serial Transmission Flowchart

15.4.6 Serial Data Reception (Asynchronous Mode)

Figure 15.8 shows an example of the operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

1. The SCI monitors the communication line, and if a start bit is detected, performs internal synchronization, receives receive data in RSR, and checks the parity bit and stop bit.
2. If an overrun error (when reception of the next data is completed while the RDRF flag in SSR is still set to 1) occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF flag remains to be set to 1.
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 ERI interrupt request is generated.
4. If a framing error (when the stop bit is 0) is detected, the FER 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 ERI interrupt request is generated.
5. If reception finishes 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 RXI interrupt request is generated. Because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.

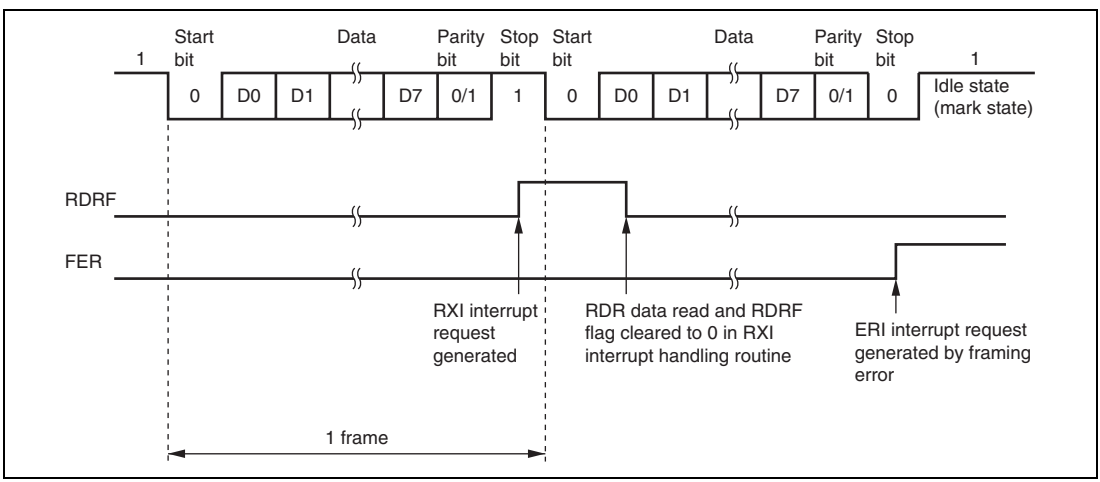


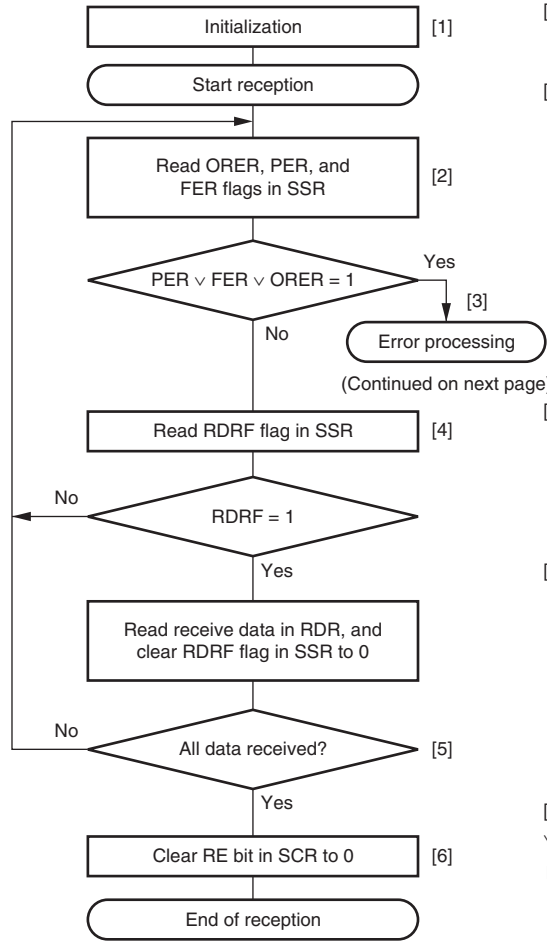
Figure 15.8 Example of SCI Receive Operation in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

Table 15.9 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 ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 15.9 shows a sample flow chart for serial data reception.

Table 15.9 SSR Status Flags and Receive Data Handling

	SSR Status Flag				
RDRF*	**ORER**	**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.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error processing and break detection:
If a receive error occurs, read the ORER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER, 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 RxD pin.

[4] SCI status check and receive data read:
Read SSR and check that RDRF = 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0.
However, the RDRF flag is cleared automatically when the DTC is initiated by an RXI interrupt and reads data from RDR.

[Legend]
∨ : Logical OR

Note: The SMR, SCR, SCMR, and BRR registers should not be written to during the period from the start to the end of reception. This does not apply to the processing at step [6].

Figure 15.9 Sample Serial Reception Flowchart (1)

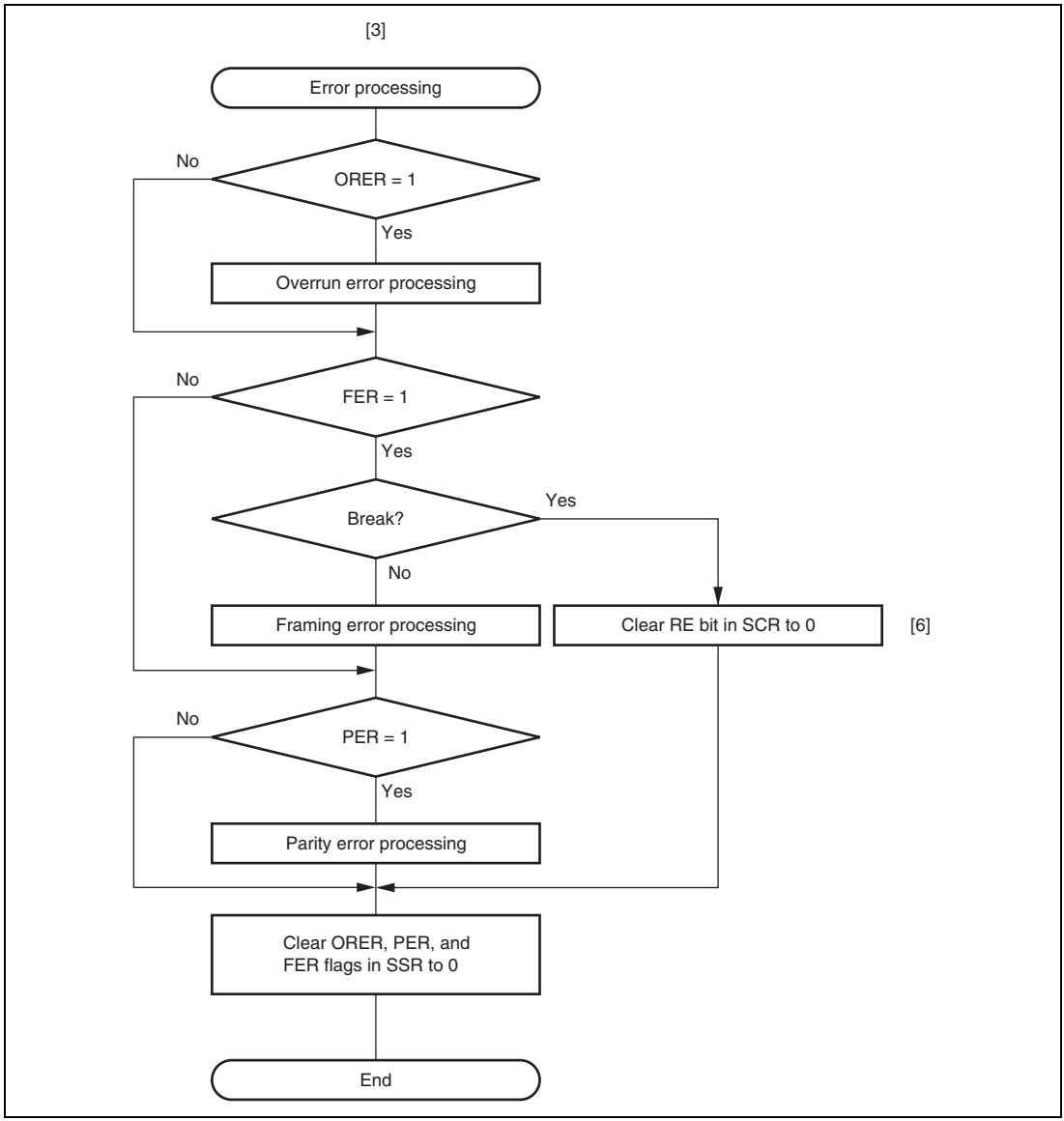


Figure 15.9 Sample Serial Reception Flowchart (2)

15.5 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer to be performed among a number of processors sharing communication lines by means of asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station, and a data transmission cycle for the specified receiving station. 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, and if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 15.10 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. The receiving station skips data until data with a 1 multiprocessor bit is sent. 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 ID does not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI 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 ORER in SSR to 1 are prohibited until data with a 1 multiprocessor bit is received. On reception of a receive character with a 1 multiprocessor bit, the MPB 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 RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is 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.

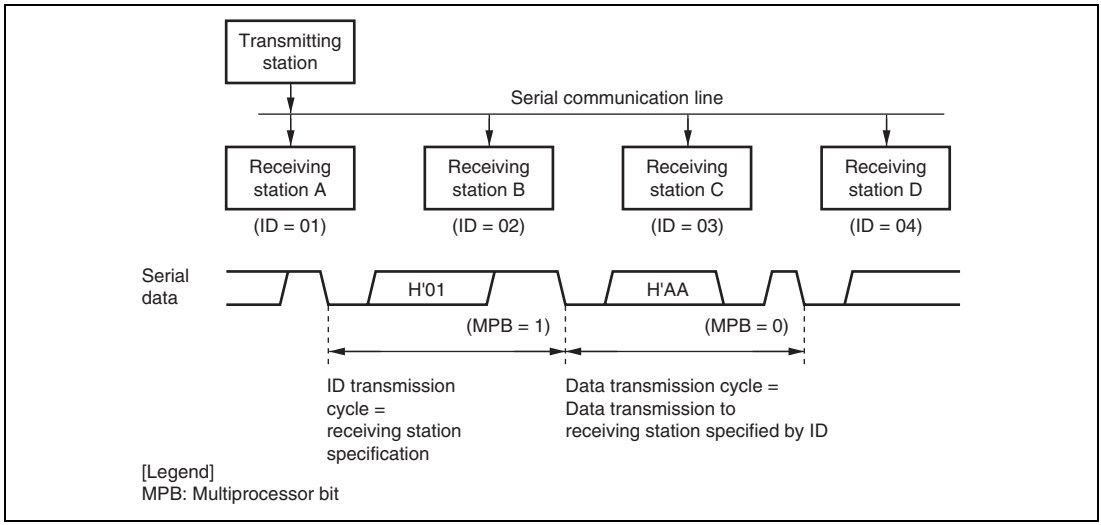
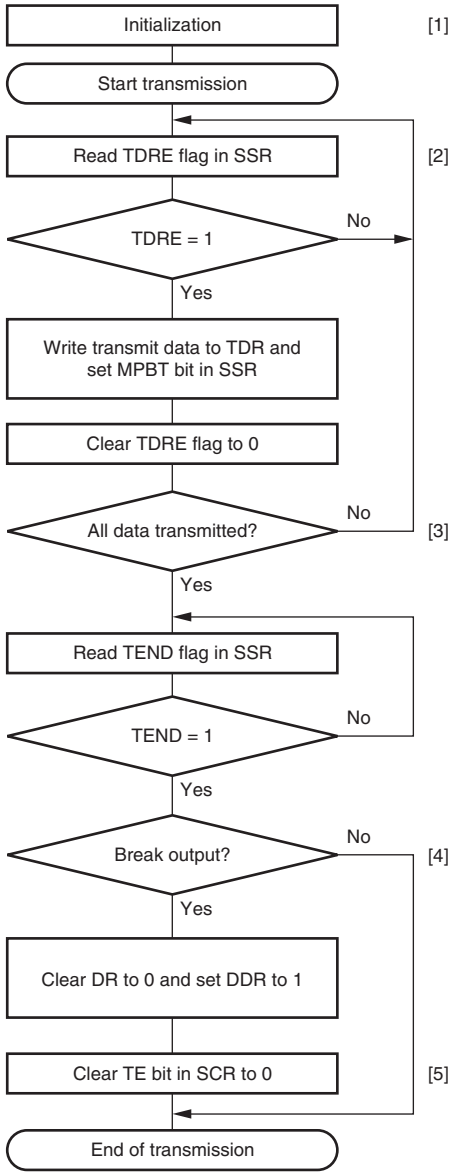


Figure 15.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

15.5.1 Multiprocessor Serial Data Transmission

Figure 15.11 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 SCI operations are the same as those in asynchronous mode.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1. Finally, clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.
However, the TDRE flag is checked and cleared automatically when the DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set port DDR to 1, clear DR to 0, and then clear the TE bit in SCR to 0.

Note: The SMR, SCR, SCMR, and BRR registers should not be written to during the period from the start to the end of transmission. This does not apply to the processing at step [5].

Figure 15.11 Sample Multiprocessor Serial Transmission Flowchart

15.5.2 Multiprocessor Serial Data Reception

Figure 15.13 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 sent. On receiving data with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI interrupt request is generated at this time. All other SCI operations are the same as in asynchronous mode. Figure 15.12 shows an example of SCI operation for multiprocessor format reception.

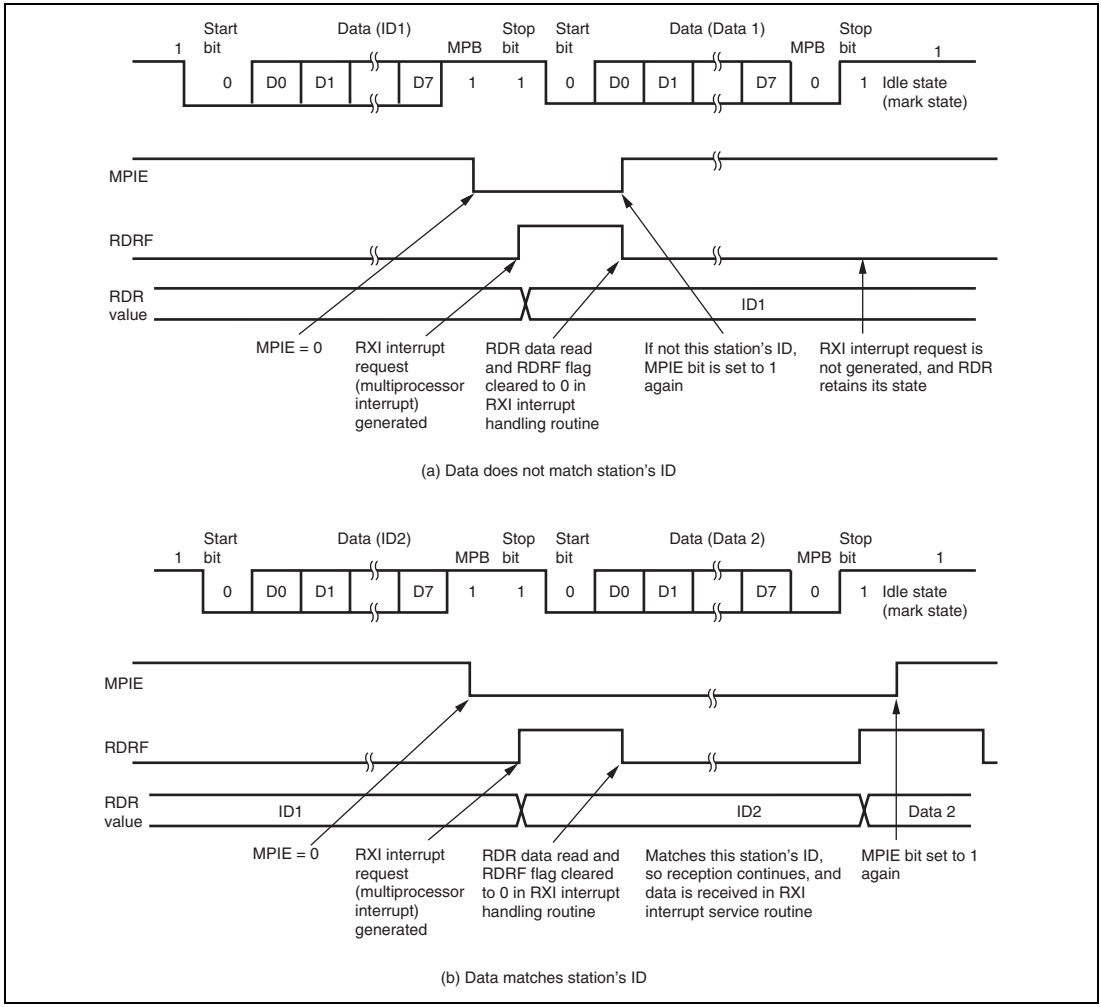
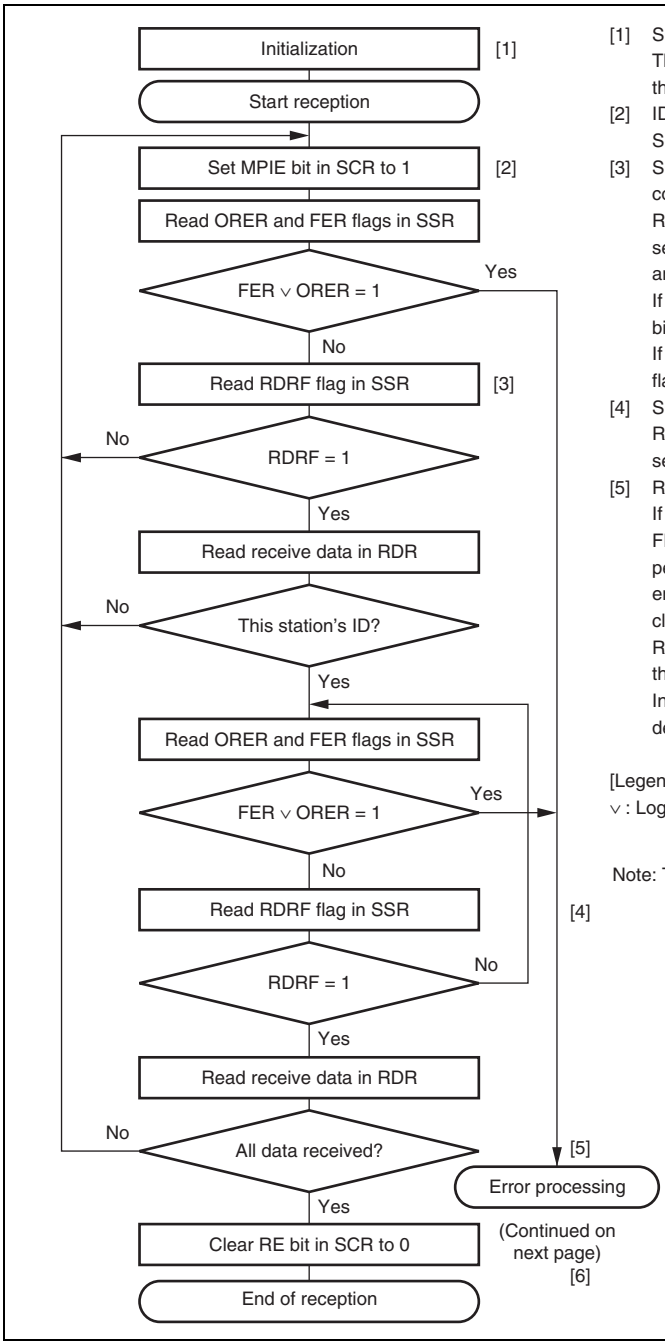


Figure 15.12 Example of SCI Receive Operation
(Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] ID reception cycle:
Set the MPIE bit in SCR to 1.

[3] SCI status check, ID reception and comparison:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and compare it with this station's ID.
If the data is not this station's ID, set the MPIE bit to 1 again, and clear the RDRF flag to 0.
If the data is this station's ID, clear the RDRF flag to 0.

[4] SCI status check and data reception:
Read SSR and check that the RDRF flag is set to 1, then read the data in RDR.

[5] Receive error processing and break detection:
If a receive error occurs, read the ORER and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER and FER flags are all cleared to 0.
Reception cannot be resumed if either of these flags is set to 1.
In the case of a framing error, a break can be detected by reading the RxD pin value.

[Legend]
∨ : Logical OR

Note: The SMR, SCR, SCMR, and BRR registers should not be written to during the period from the start to the end of reception. This does not apply to the processing at step [6].

Figure 15.13 Sample Multiprocessor Serial Reception Flowchart (1)

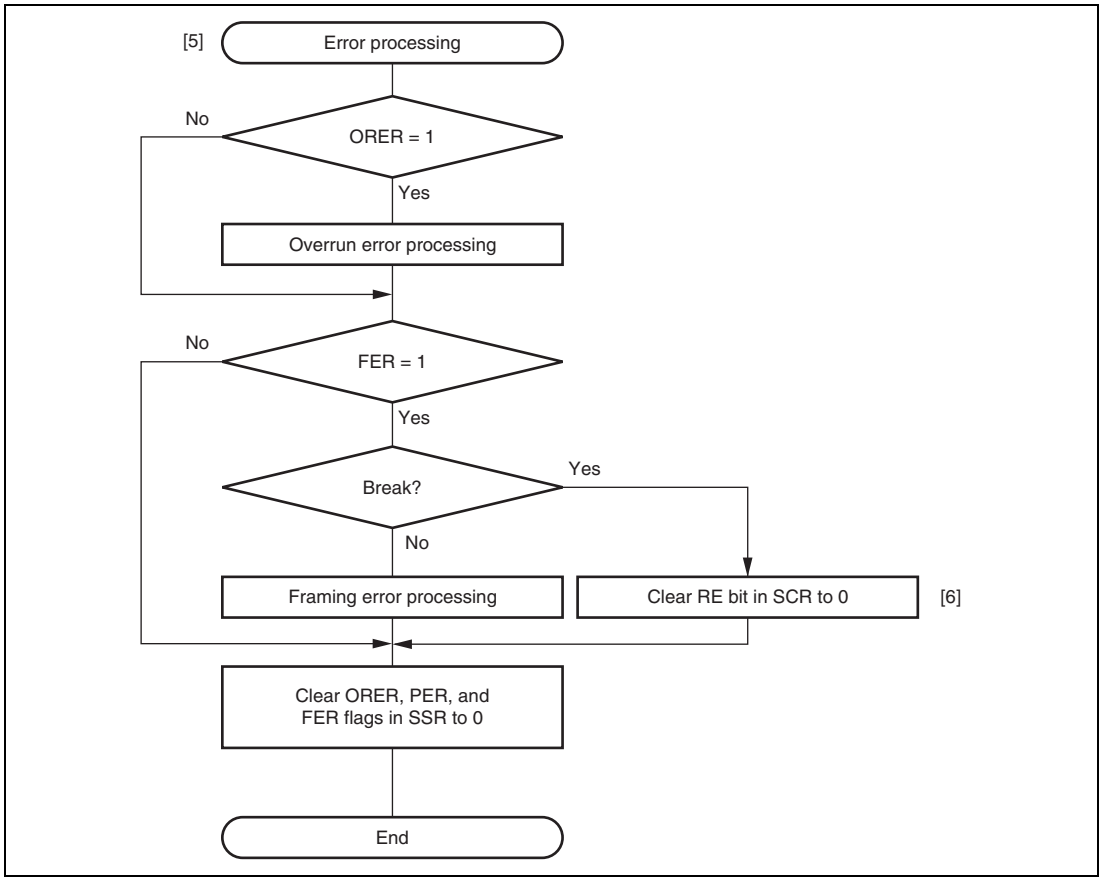


Figure 15.13 Sample Multiprocessor Serial Reception Flowchart (2)

15.6 Operation in Clocked Synchronous Mode

Figure 15.14 shows the general format for clocked synchronous communication. In clocked synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the transmission line holds the MSB state. In clocked synchronous mode, no parity or multiprocessor bit is added. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

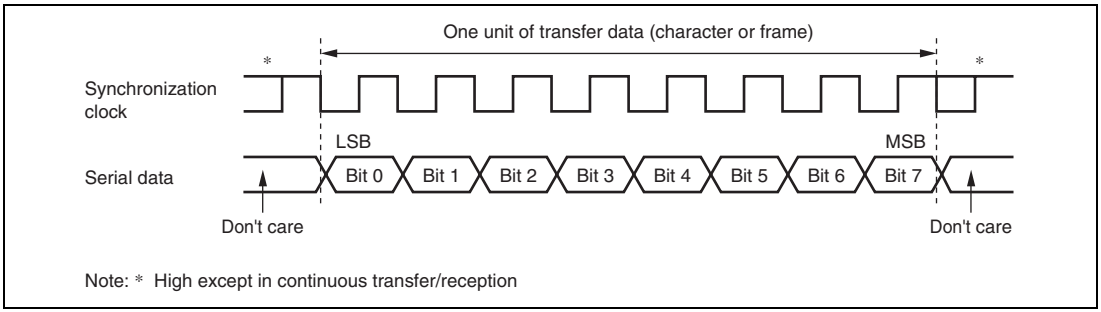


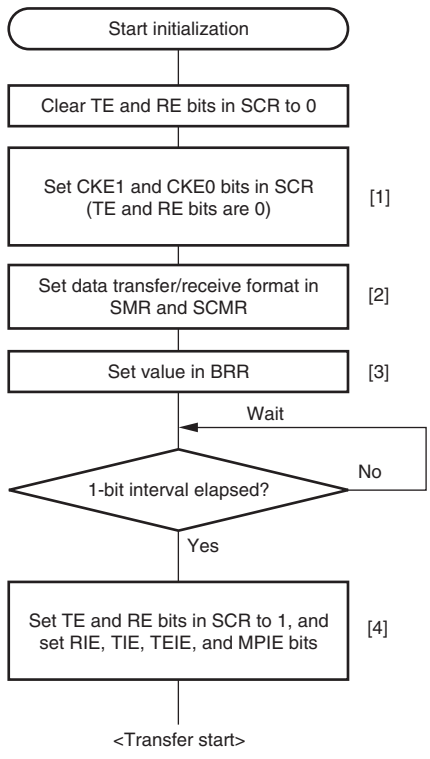
Figure 15.14 Data Format in Clocked Synchronous Communication (LSB-First)

15.6.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK pin can be selected, according to the setting of the CKE1 and CKE0 bits in SCR. When the SCI is operated on an internal clock, the synchronization clock is output from the SCK pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high.

15.6.2 SCI Initialization (Clocked Synchronous Mode)

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described in a sample flowchart in figure 15.15. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag in SSR is set to 1. However, clearing the RE bit to 0 does not initialize the RDRF, PER, FER, and ORER flags in SSR, or RDR.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, TE and RE to 0.
[2] Set the data transfer/receive format in SMR and SCMR.
[3] Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.
[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1.
Also set the RIE, TIE TEIE, and MPIE bits.
Setting the TE and RE bits enables the TxD and RxD pins to be used.

Note: * In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 15.15 Sample SCI Initialization Flowchart

15.6.3 Serial Data Transmission (Clocked Synchronous Mode)

Figure 15.16 shows an example of SCI operation for transmission in clocked synchronous mode. In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is 0, 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 SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a TXI interrupt request is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. 8-bit data is sent from the TxD 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.
4. The SCI checks the TDRE flag at the timing for sending the last bit.
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 TxD pin maintains the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated. The SCK pin is fixed high.

Figure 15.17 shows a sample flow chart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) is set to 1. Make sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.

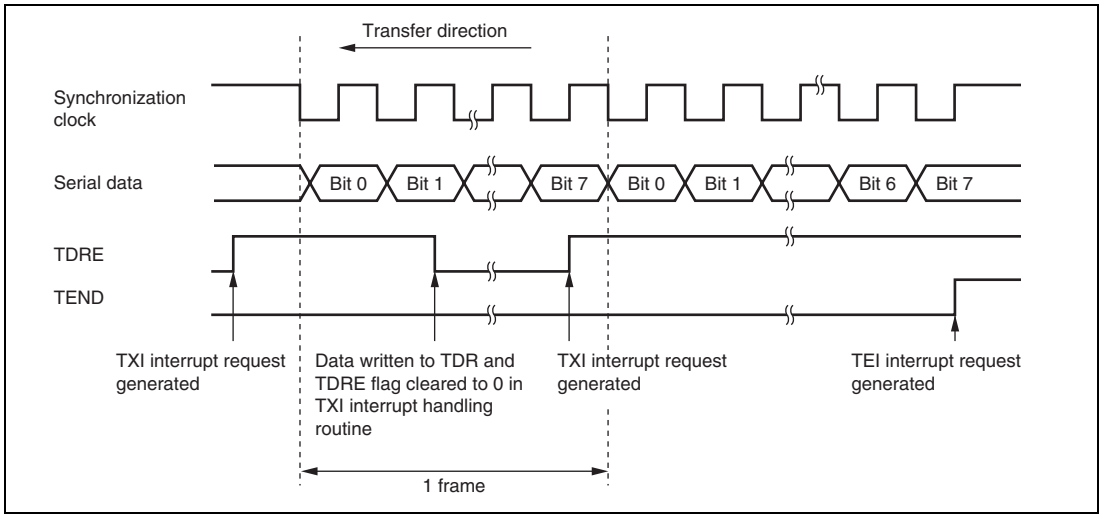
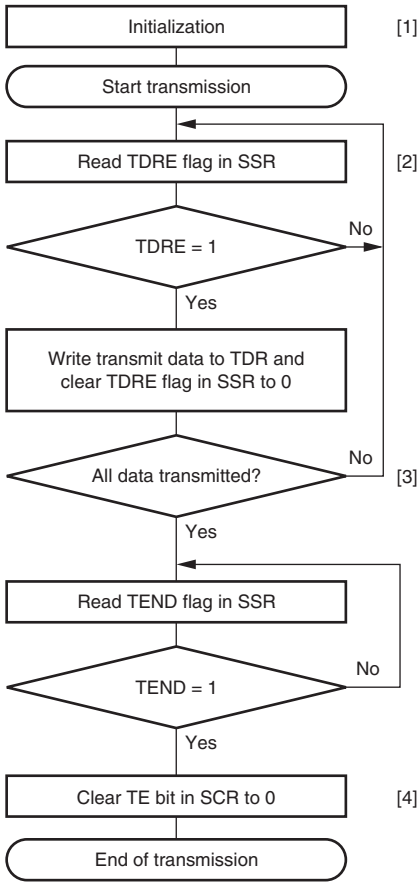


Figure 15.16 Example of SCI Transmit Operation in Clocked Synchronous Mode



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.
However, the TDRE flag is checked and cleared automatically when the DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

Note: The SMR, SCR, SCMR, and BRR registers should not be written to during the period from the start to the end of transmission. This does not apply to the processing at step [4].

Figure 15.17 Sample Serial Transmission Flowchart

15.6.4 Serial Data Reception (Clocked Synchronous Mode)

Figure 15.18 shows an example of SCI operation for reception in clocked synchronous mode. In serial reception, the SCI operates as described below.

1. The SCI performs internal initialization in synchronization with a synchronization clock input or output, starts receiving data, and stores the receive data in RSR.
2. If an overrun error (when reception of the next data is completed while the RDRF flag is still set to 1) occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF flag remains to be set to 1.
3. If reception finishes 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 RXI interrupt request is generated. Because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.

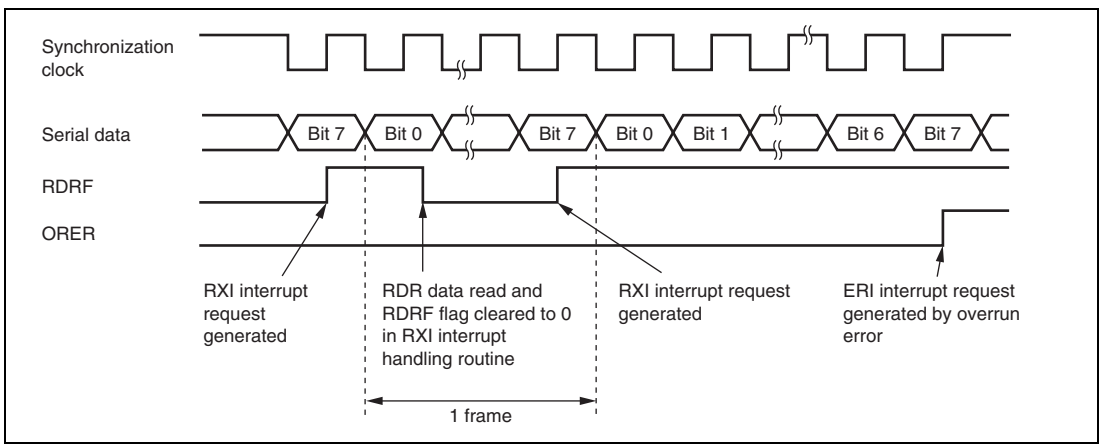


Figure 15.18 Example of SCI Receive Operation in Clocked Synchronous Mode

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 15.19 shows a sample flowchart for serial data reception.

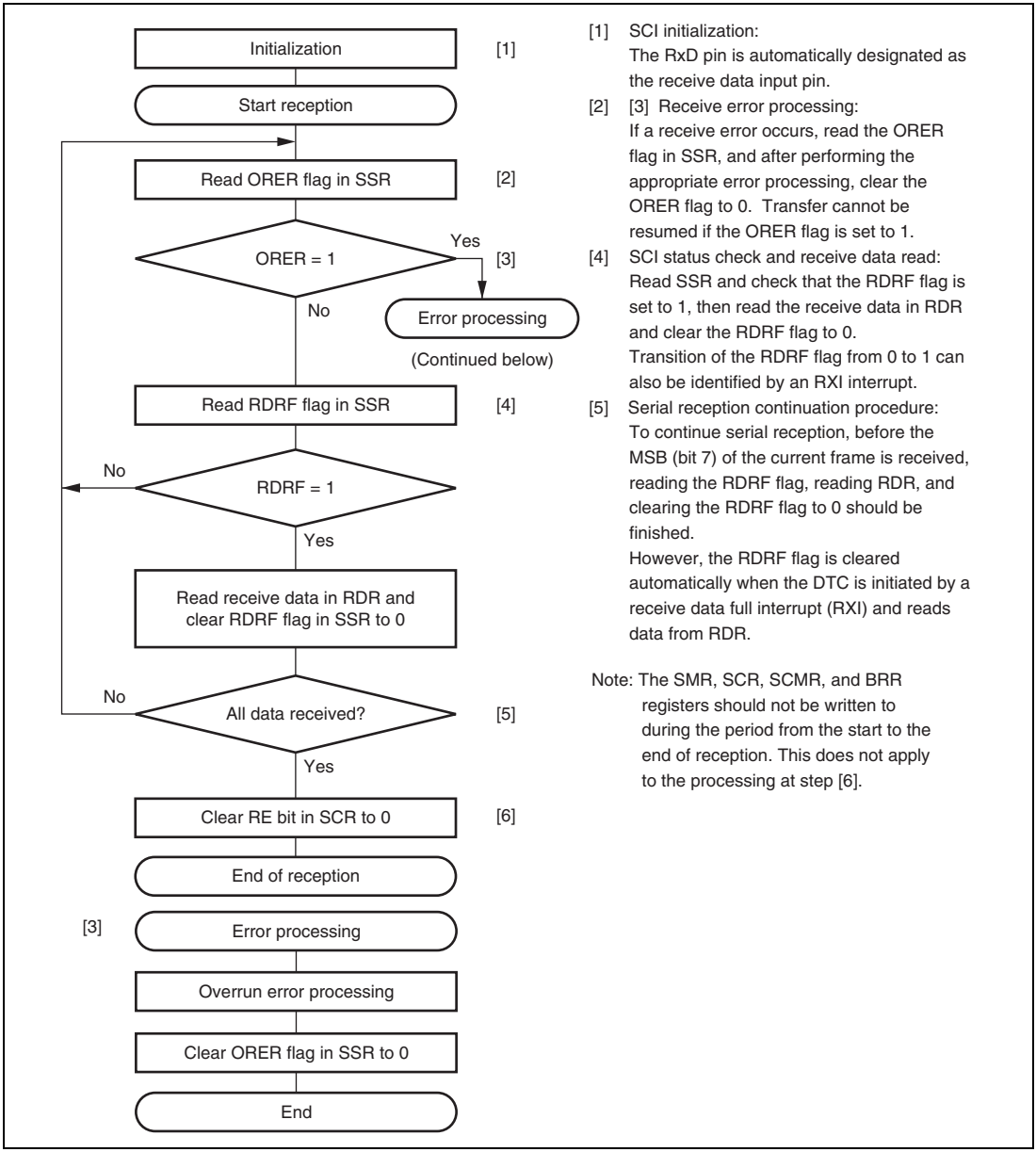
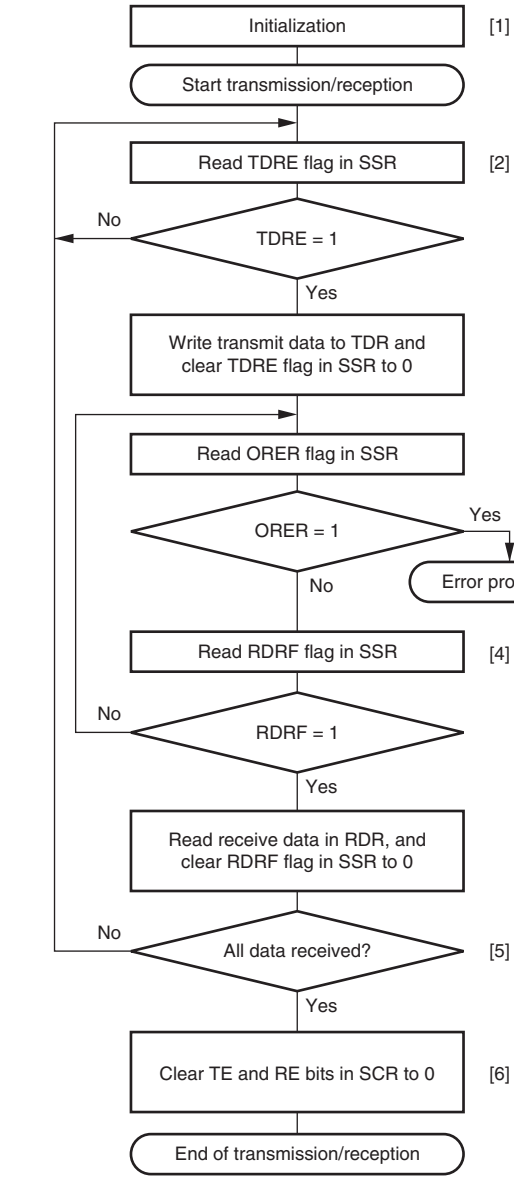


Figure 15.19 Sample Serial Reception Flowchart

15.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode)

Figure 15.20 shows a sample flowchart for simultaneous serial transmit and receive operations. After initializing the SCI, 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, check that the SCI has finished transmission and the TDRE and TEND flags in SSR are set to 1, clear the TE bit in SCR to 0, and then set the TE and RE bits to 1 simultaneously with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, check that the SCI has finished reception, and clear the RE bit to 0. Then after checking that the RDRF bit in SSR and receive error flags (ORER, FER, and PER) are cleared to 0, set the TE and RE bits to 1 simultaneously with a single instruction.



Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.
The SMR, SCR, SCMR, and BRR registers should not be written to during the period from the start to the end of transmission/reception. This does not apply to the processing at step [6].

[1] SCI initialization:
The TxD pin is designated as the transmit data output pin, and the RxD pin is designated as the receive data input pin, enabling simultaneous transmit and receive operations.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0. Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error processing, clear the ORER flag to 0. Transmission/reception cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR and clear the TDRE flag to 0.
However, the TDRE flag is checked and cleared automatically when the DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR. Similarly, the RDRF flag is cleared automatically when the DTC is initiated by a receive data full interrupt (RXI) and reads data from RDR.

Figure 15.20 Sample Flowchart of Simultaneous Serial Transmission and Reception

15.7 Interrupt Sources

Table 15.10 shows the interrupt sources in serial communication interface. A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled using the enable bits in SCR.

When the TDRE flag in SSR is set to 1, a TXI interrupt request is generated. When the TEND flag in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the DTC to allow data transfer. The TDRE flag is automatically cleared to 0 at data transfer by the DTC.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the ORER, PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interrupt can activate the DTC to allow data transfer. The RDRF flag is automatically cleared to 0 at data transfer by the DTC.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to 1. If a TEI interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt has priority for acceptance. However, note that if the TDRE and TEND flags are cleared simultaneously by the TXI interrupt routine, the SCI cannot branch to the TEI interrupt routine later.

Table 15.10 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority
0	ERI0	Receive error	ORER, FER, PER	Not possible	High
	RXI0	Receive data full	RDRF	Possible	
	TXI0	Transmit data empty	TDRE	Possible	
	TEI0	Transmit end	TEND	Not possible	
1	ERI1	Receive error	ORER, FER, PER	Not possible	
	RXI1	Receive data full	RDRF	Possible	
	TXI1	Transmit data empty	TDRE	Possible	
	TEI1	Transmit end	TEND	Not possible	Low

15.8 Usage Notes

15.8.1 Module Stop Mode Setting

SCI operation can be disabled or enabled using the module stop control register. The initial setting is for SCI operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

15.8.2 Break Detection and Processing

When framing error detection is performed, a break can be detected by reading the RxD pin value directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag in SSR is set, and the PER flag may also be set. Note that, since the SCI continues the receive operation even after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

15.8.3 Mark State and Break Detection

When the TE bit in SCR is 0, the TxD pin is used as an I/O port whose direction (input or output) and level are determined by DR and DDR of the port. This can be used to set the TxD pin to the 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 DDR and DR to 1. Since the TE bit is cleared to 0 at this point, the TxD pin becomes an I/O port, and 1 is output from the TxD pin. To send a break during serial transmission, first set DDR to 1 and DR to 0, and then clear the TE bit to 0. When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, the TxD pin becomes an I/O port, and 0 is output from the TxD pin.

15.8.4 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only)

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

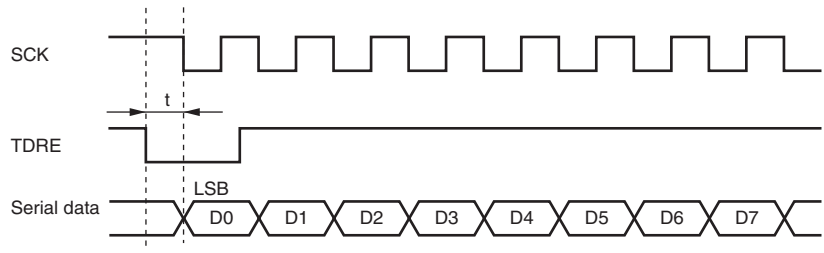
15.8.5 Relation between Writing to TDR and TDRE Flag

Data can be written to TDR irrespective of the TDRE flag status in SSR. However, if the new data is written to TDR when the TDRE flag is 0, that is, when the previous data has not been transferred to TSR yet, the previous data in TDR is lost. Be sure to write transmit data to TDR after verifying that the TDRE flag is set to 1.

15.8.6 Restrictions on Using DTC

When an external clock source is used as a synchronization clock, update TDR by the DTC and wait for at least five ϕ clock cycles before allowing the transmit clock to be input. If the transmit clock is input within four clock cycles after TDR modification, the SCI may malfunction (figure 15.21).

When using the DTC to read RDR, be sure to set the receive end interrupt source (RXI) as a DTC activation source.



Note: * When external clock is supplied, t must be more than four clock cycles.

Figure 15.21 Example of Transmission using DTC in Clocked Synchronous Mode

15.8.7 SCI Operations during Mode Transitions

(1) Transmission

Before making a transition to module stop, software standby, or sub-sleep mode, stop all transmit operations (TE = TIE = TEIE = 0). TSR, TDR, and SSR are reset. The states of the output pins during each mode depend on the port settings, and the pins output a high-level signal after mode cancellation. If a transition is made during data transmission, the data being transmitted will be undefined.

To transmit data in the same transmission mode after mode cancellation, set TE to 1, read SSR, write to TDR, clear TDRE in this order, and then start transmission. To transmit data in a different transmission mode, initialize the SCI first.

Figure 15.22 shows a sample flowchart for mode transition during transmission. Figures 15.23 and 15.24 show the pin states during transmission.

Before making a transition from the transmission mode using DTC transfer to module stop, software standby, or sub-sleep mode, stop all transmit operations (TE = TIE = TEIE = 0). Setting

TE and TIE to 1 after mode cancellation generates a TXI interrupt request to start transmission using the DTC.

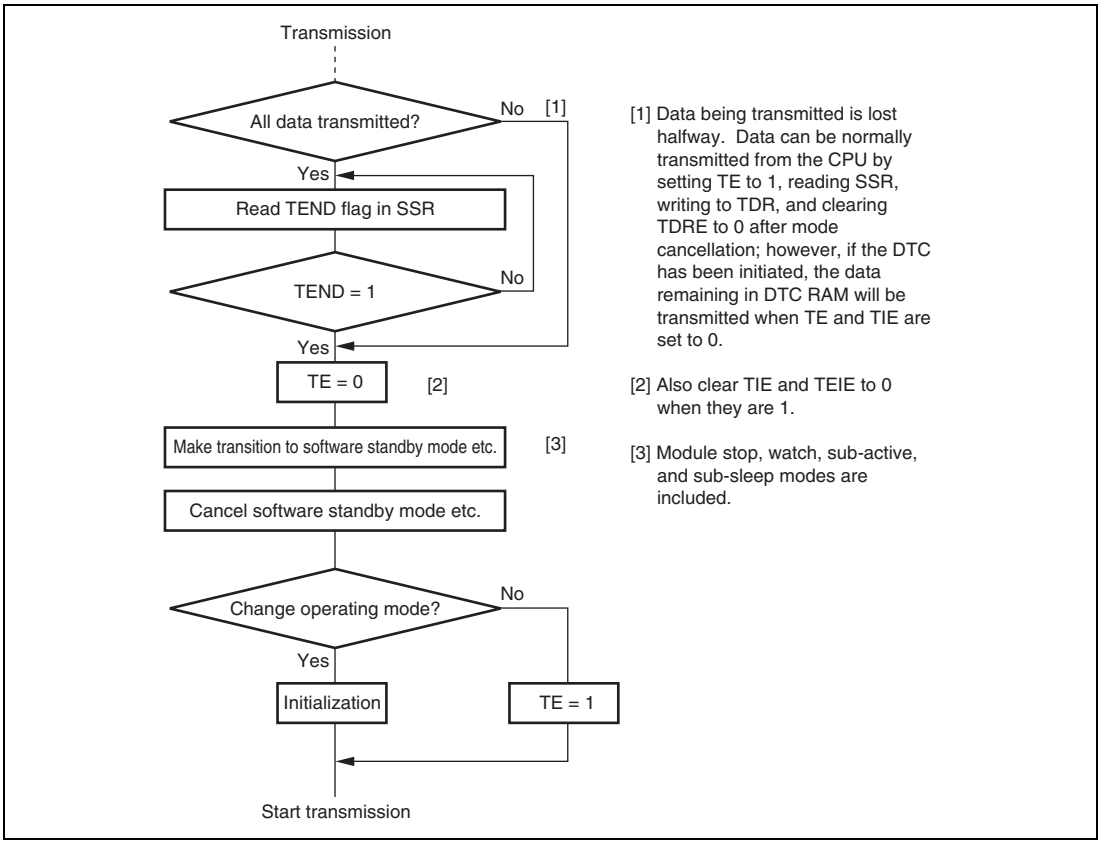


Figure 15.22 Sample Flowchart for Mode Transition during Transmission

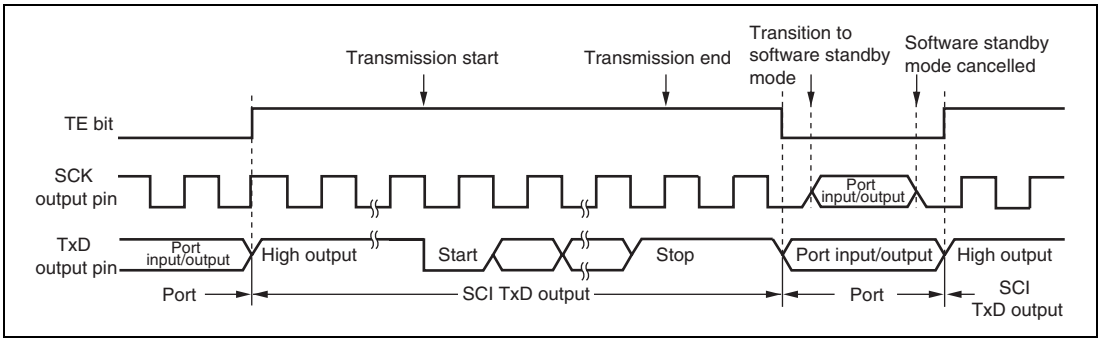


Figure 15.23 Pin States during Transmission in Asynchronous Mode (Internal Clock)

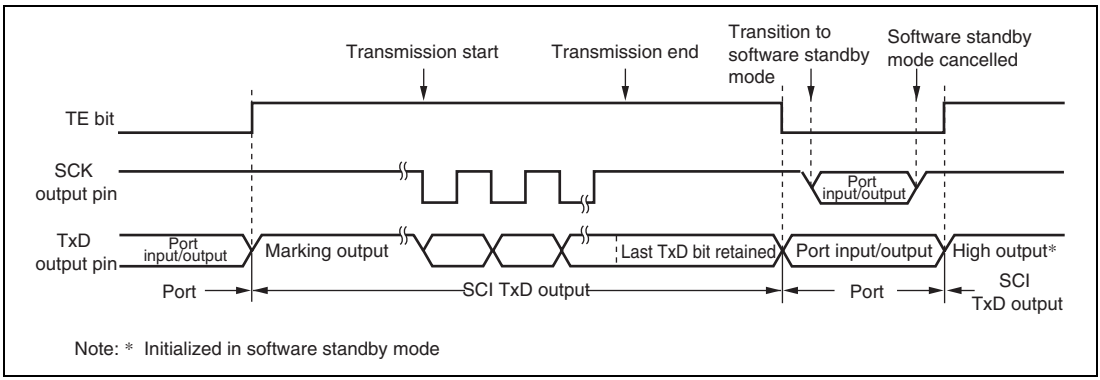


Figure 15.24 Pin States during Transmission in Clocked Synchronous Mode (Internal Clock)

(2) Reception

Before making a transition to module stop, software standby, watch, sub-active, or sub-sleep mode, stop reception (RE = 0). RSR, RDR, and SSR are reset. If a transition is made during data reception, the data being received will be invalid.

To receive data in the same reception mode after mode cancellation, set RE to 1, and then start reception. To receive data in a different reception mode, initialize the SCI first.

Figure 15.25 shows a sample flowchart for mode transition during reception.

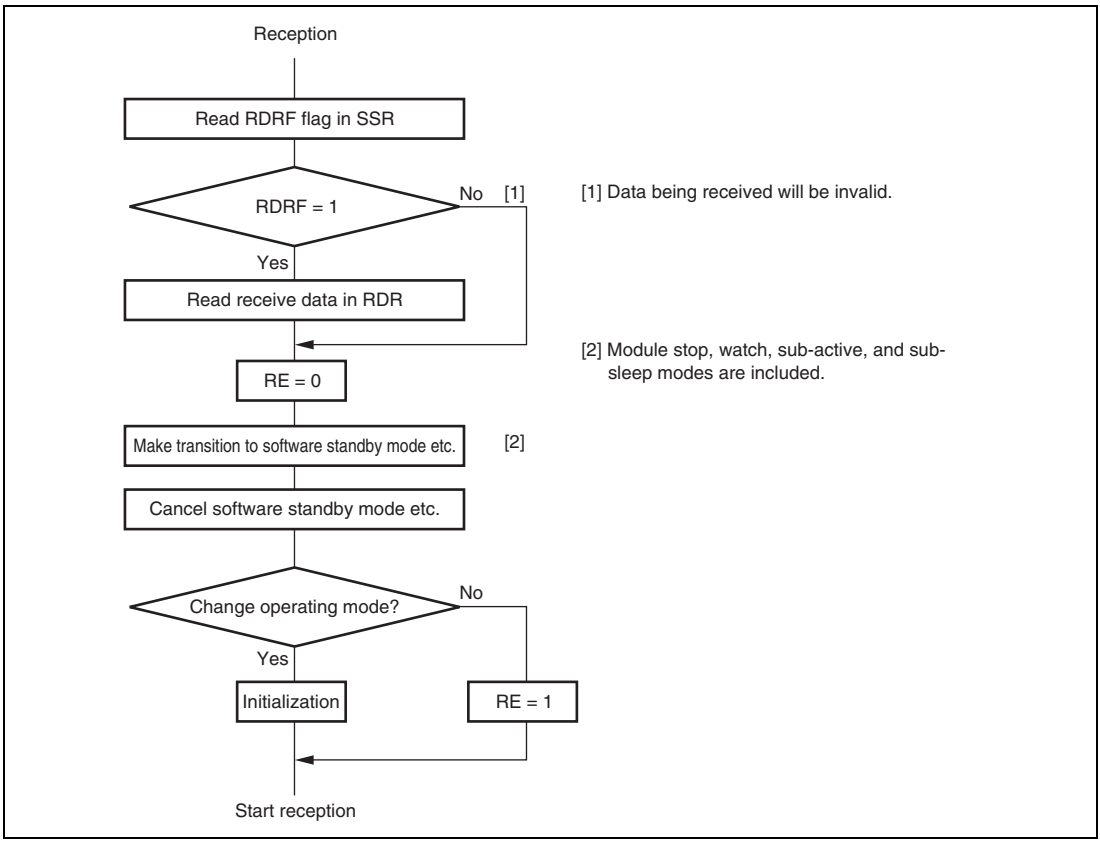


Figure 15.25 Sample Flowchart for Mode Transition during Reception

15.8.8 Notes on Switching from SCK Pins to Port Pins

When SCK pins are switched to port pins after transmission has completed, pins are enabled for port output after outputting a low pulse of half a cycle as shown in figure 15.26.

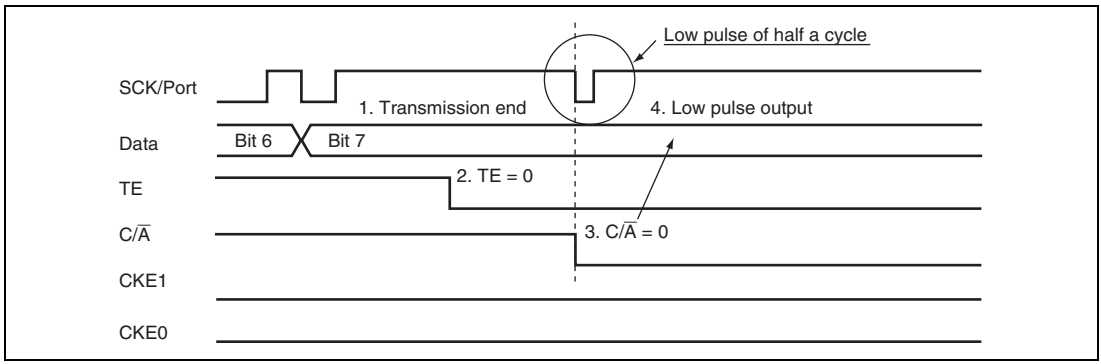


Figure 15.26 Switching from SCK Pins to Port Pins

To prevent the low pulse output that is generated when switching the SCK pins to the port pins, specify the SCK pins for input (pull up the SCK/port pins externally), and follow the procedure below with DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE1 = 0, and TE = 1.

1. End serial data transmission
2. TE bit = 0
3. CKE1 bit = 1
4. C/$\overline{A}$ bit = 0 (switch to port output)
5. CKE1 bit = 0

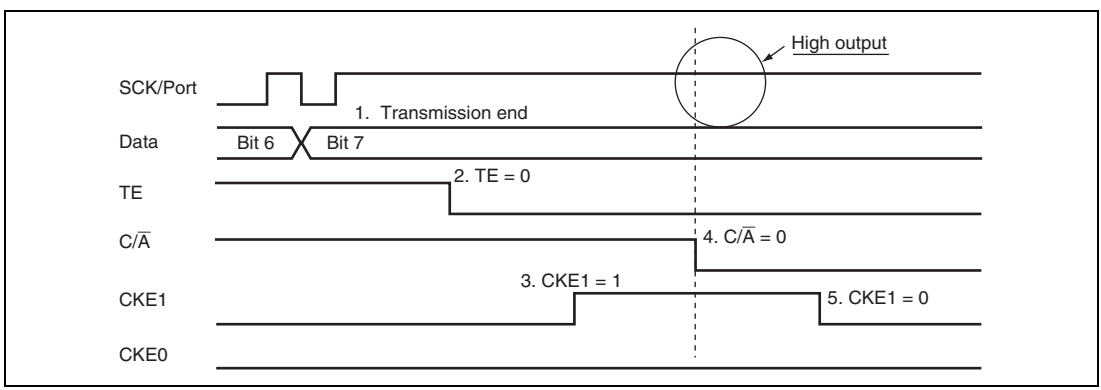


Figure 15.27 Prevention of Low Pulse Output at Switching from SCK Pins to Port Pins

15.8.9 Notes on Register Writing during the Receive, Transmit, or Transfer Operation

Once the TE or RE bit of the SCR register has been set to 1 to start the receive, transmit, or transfer operation, the SMR, SCR, SCMR, and BRR registers should not be written to. Rewriting the set value to these registers is also prohibited. However, this does not apply to the clearing of the TE or RE bit at the end of the receive, transmit, or transfer operation.

These registers can always be read from.

Section 16 I²C Bus Interface (IIC)

This LSI has a two-channel I²C bus interface. The I²C bus interface 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.

16.1 Features

- Selection of addressing format or non-addressing format
  - I²C bus format: addressing format with an acknowledge bit, for master/slave operation
  - Clocked synchronous serial format: non-addressing format without an acknowledge bit, for master operation only
- Conforms to Philips I²C bus interface (I²C bus format)
- Two ways of setting slave address (I²C bus format)
- Start and stop conditions generated automatically in master mode (I²C bus format)
- Selection of the acknowledge output level in reception (I²C bus format)
- Automatic loading of an acknowledge bit in transmission (I²C bus format)
- Wait function in master mode (I²C bus format)
  - A wait can be inserted by driving the SCL pin low after data transfer, excluding acknowledgement.
  - The wait can be cleared by clearing the interrupt flag.
- Wait function (I²C bus format)
  - A wait request can be generated by driving the SCL pin low after data transfer.
  - The wait request is cleared when the next transfer becomes possible.
- Interrupt sources
  - Data transfer end (including when a transition to transmit mode with I²C bus format occurs, when ICDR data is transferred, or during a wait state)
  - Address match: When any slave address matches or the general call address is received in slave receive mode with I²C bus format (including address reception after loss of master arbitration)
  - Start condition detection (in master mode)
  - Stop condition detection (in slave mode)
- Selection of 16 internal clocks (in master mode)

- Direct bus drive (SCL/SDA pin)
  - — Two pins—P52/SCL0 and P47/SDA0—(normally NMOS push-pull outputs) function as NMOS open-drain outputs when the bus drive function is selected.
  - — Two pins—P24/SCL1 and P23/SDA1—(normally CMOS pins) function as NMOS outputs when the bus drive function is selected. Voltage exceeding Vcc cannot be applied.

Figure 16.1 shows a block diagram of the I²C bus interface. Figure 16.2 shows an example of I/O pin connections to external circuits. Since I²C bus interface I/O pins are different in structure from normal port pins, they have different specifications for permissible applied voltages. For details, see section 24, Electrical Characteristics.

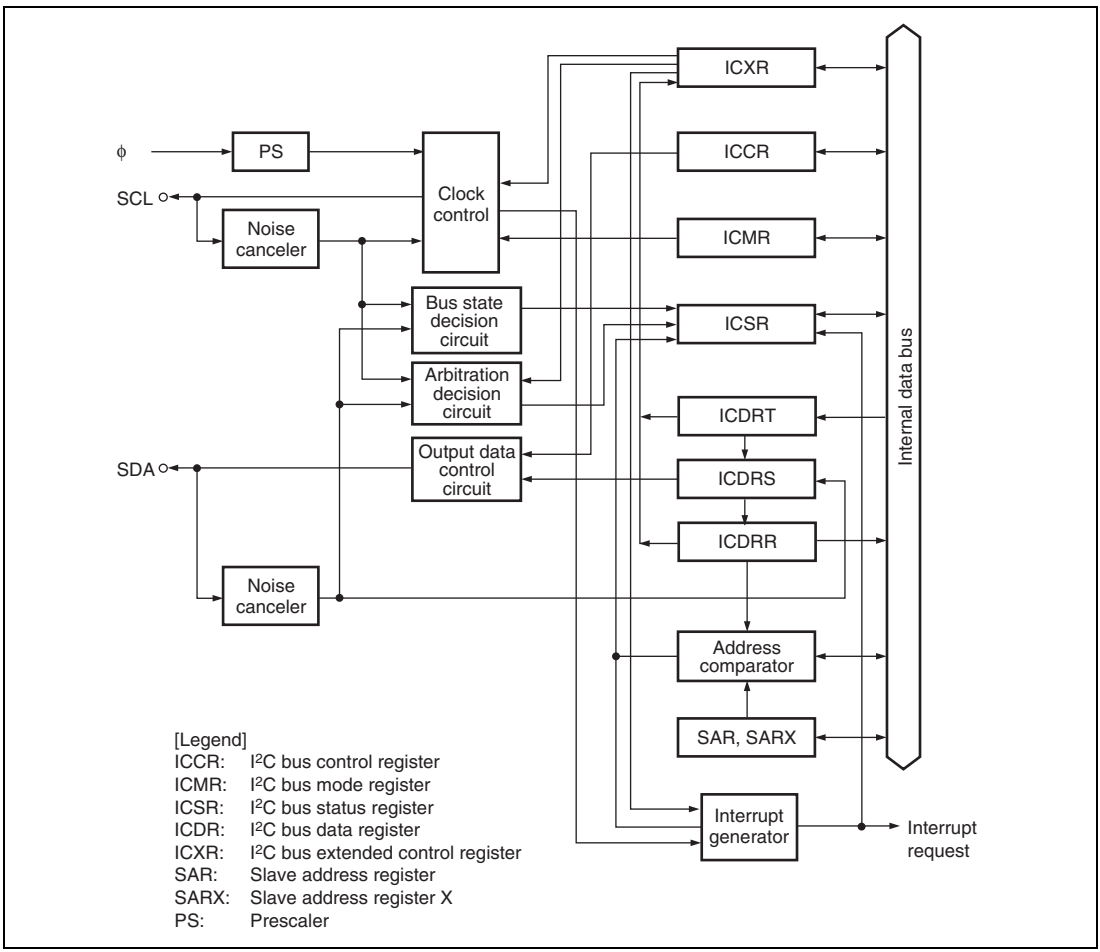


Figure 16.1 Block Diagram of I²C Bus Interface

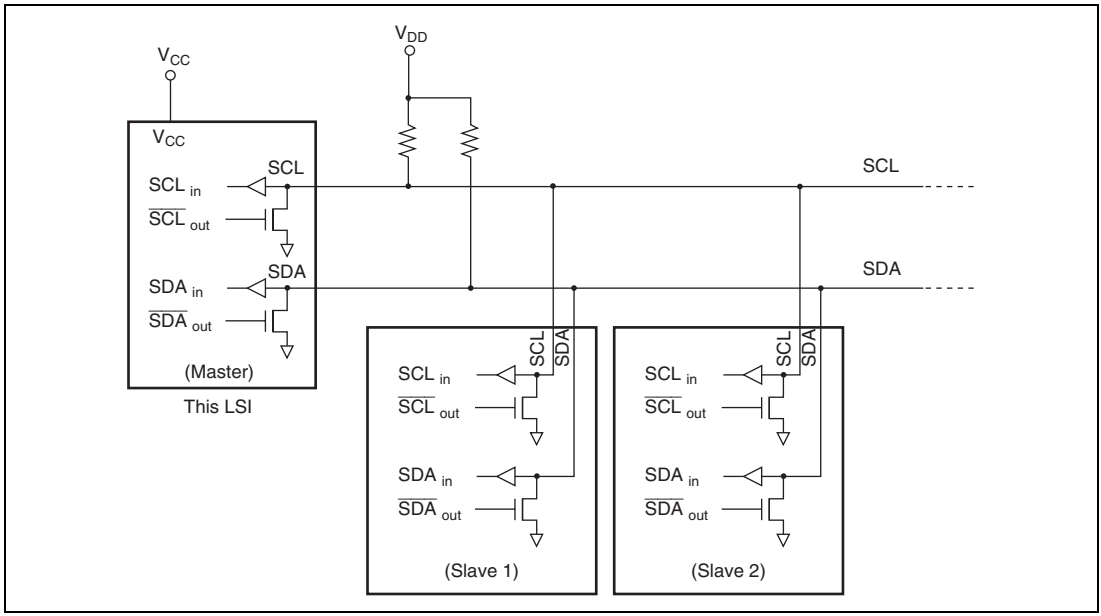


Figure 16.2 I²C Bus Interface Connections (Example: This LSI as Master)

16.2 Input/Output Pins

Table 16.1 summarizes the input/output pins used by the I²C bus interface.

Table 16.1 Pin Configuration

Channel	Symbol*	Input/Output	Function
0	SCL0	Input/Output	Serial clock input/output pin of IIC_0
	SDA0	Input/Output	Serial data input/output pin of IIC_0
1	SCL1	Input/Output	Serial clock input/output pin of IIC_1
	SDA1	Input/Output	Serial data input/output pin of IIC_1

Note: * In the text, the channel subscript is omitted, and only SCL and SDA are used.

16.3 Register Descriptions

The I²C bus interface has the following registers. Registers ICDR and SARX and registers ICMR and SAR are allocated to the same addresses. Accessible registers differ depending on the ICE bit in ICCR. When the ICE bit is cleared to 0, SAR and SARX can be accessed, and when the ICE bit is set to 1, ICMR and ICDR can be accessed. For details on the serial timer control register, refer to section 3.2.3, Serial Timer Control Register (STCR).

- I²C bus control register (ICCR)
- I²C bus status register (ICSR)
- I²C bus data register (ICDR)
- I²C bus mode register (ICMR)
- Slave address register (SAR)
- Second slave address register (SARX)
- I²C bus extended control register (ICXR)
- DDC switch register (DDCSWR) (for IIC_0 only)

16.3.1 I²C Bus Data Register (ICDR)

ICDR is an 8-bit readable/writable register that is used as a transmit data register when transmitting and a receive data register when receiving. ICDR is internally divided into a shift register (ICDRS), receive buffer (ICDRR), and transmit buffer (ICDRT). Data transfers among these three registers are performed automatically in accordance with changes in the bus state, and they affect the status of internal flags such as ICDRE and ICDRF.

In master transmit mode with the I²C bus format, writing transmit data to ICDR should be performed after start condition detection. When the start condition is detected, previous write data is ignored. In slave transmit mode, writing should be performed after the slave addresses match and the TRS bit is automatically changed to 1.

If the IIC is in transmit mode (TRS = 1) and ICDRT has the next transmit data (the ICDRE flag is 0) after successful transmission/reception of one frame of data using ICDRS, data is transferred automatically from ICDRT to ICDRS.

If the IIC is in transmit mode (TRS = 1) and ICDRT has the next data (the ICDRE flag is 0), data is transferred automatically from ICDRT to ICDRS, following transmission of one frame of data using ICDRS. When the ICDRE flag is 1 and the next transmit data writing is waited, data is transferred automatically from ICDRT to ICDRS by writing to ICDR. If I²C is in receive mode

(TRS = 0), no data is transferred from ICDRT to ICDRS. Note that data should not be written to ICDR in receive mode.

Reading receive data from ICDR is performed after data is transferred from ICDRS to ICDRR.

If I²C is in receive mode and no previous data remains in ICDRR (the ICDRF flag is 0), data is transferred automatically from ICDRS to ICDRR, following reception of one frame of data using ICDRS. If additional data is received while the ICDRF flag is 1, data is transferred automatically from ICDRS to ICDRR by reading from ICDR. In transmit mode, no data is transferred from ICDRS to ICDRR. Always set I²C to receive mode before reading from ICDR.

If the number of bits in a frame, excluding the acknowledge bit, is less than eight, transmit data and receive data are stored differently. Transmit data should be written justified toward the MSB side when MLS = 0 in ICMR, and toward the LSB side when MLS = 1. Receive data bits should be read from the LSB side when MLS = 0, and from the MSB side when MLS = 1.

ICDR can be written to and read from only when the ICE bit is set to 1 in ICCR. The initial value of ICDR is undefined.

16.3.2 Slave Address Register (SAR)

SAR sets the slave address and selects the communication format. If the LSI is in slave mode with the I²C bus format selected, when the FS bit is set to 0 and the upper 7 bits of SAR match the upper 7 bits of the first frame received after a start condition, the LSI operates as the slave device specified by the master device. SAR can be accessed only when the ICE bit in ICCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7	SVA6	0	R/W	Slave Address 6 to 0
6	SVA5	0	R/W	Set a slave address.
5	SVA4	0	R/W	
4	SVA3	0	R/W	
3	SVA2	0	R/W	
2	SVA1	0	R/W	
1	SVA0	0	R/W	
0	FS	0	R/W	Format Select Selects the communication format with the combination of the FSX bit in SARX. Refer to table 16.2. This bit should be set to 0 when general call address recognition is performed.

16.3.3 Second Slave Address Register (SARX)

SARX sets the second slave address and selects the communication format. In slave mode, transmit/receive operations by the DTC are possible when the received address matches the second slave address. If the LSI is in slave mode with the I²C bus format selected, when the FSX bit is set to 0 and the upper 7 bits of SARX match the upper 7 bits of the first frame received after a start condition, the LSI operates as the slave device specified by the master device. SARX can be accessed only when the ICE bit in ICCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7	SVAX6	0	R/W	Second Slave Address 6 to 0
6	SVAX5	0	R/W	Set the second slave address.
5	SVAX4	0	R/W	
4	SVAX3	0	R/W	
3	SVAX2	0	R/W	
2	SVAX1	0	R/W	
1	SVAX0	0	R/W	
0	FSX	1	R/W	Format Select X Selects the communication format together with the FS bit in SAR and the SW bit in DDCSWR. Refer to table 16.2.

Table 16.2 Communication Format

SAR	SARX	
FS	**FSX**	**Operating Mode**
0	0	I²C bus format • SAR and SARX slave addresses recognized • General call address recognized
	1	I²C bus format • SAR slave address recognized • SARX slave address ignored • General call address recognized
1	0	I²C bus format • SAR slave address ignored • SARX slave address recognized • General call address ignored
	1	Clocked synchronous serial format • SAR and SARX slave addresses ignored • General call address ignored

- I²C bus format: addressing format with an acknowledge bit
- Clocked synchronous serial format: non-addressing format without an acknowledge bit, for master mode only

16.3.4 I²C Bus Mode Register (ICMR)

ICMR sets the communication format and transfer rate. It can only be accessed when the ICE bit in ICCR is set to 1.

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 This bit is valid only in master mode with the I²C bus format. 0: Data and the acknowledge bit are transferred consecutively with no wait inserted. 1: After the fall of the clock for the final data bit (8th clock), the IRIC flag is set to 1 in ICCR, and a wait state begins (with SCL at the low level). When the IRIC flag is cleared to 0 in ICCR, the wait ends and the acknowledge bit is transferred. For details, refer to section 16.4.7, IRIC Setting Timing and SCL Control.
5	CKS2	0	R/W	Transfer Clock Select 2 to 0 These bits are used only in master mode.
4	CKS1	0	R/W	
3	CKS0	0	R/W	These bits select the required transfer rate, together with the IICX1 (IIC_1) and IICX0 (IIC_0) bits in STCR. Refer to table 16.3.

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. 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 line is low. The bit counter is initialized to 000 when a start condition is detected. The value returns to 000 at the end of a data transfer. I²C Bus Format / Clocked Synchronous Serial Mode 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
0	BC0	0	R/W	

Table 16.3 I²C Transfer Rate

STCR	ICMR							
Bits 5 and 6	Bit 5	Bit 4	Bit 3		Transfer Rate			
IICX	CKS2	CKS1	CKS0	Clock	φ = 8 MHz	φ = 10 MHz	φ = 16 MHz	φ = 20 MHz
0	0	0	0	φ/28	286 kHz	357 kHz	517 kHz*	714 kHz*
0	0	0	1	φ/40	200 kHz	250 kHz	400 kHz	500 kHz*
0	0	1	0	φ/48	167 kHz	208 kHz	333 kHz	417 kHz*
0	0	1	1	φ/64	125 kHz	156 kHz	250 kHz	313 kHz
0	1	0	0	φ/80	100 kHz	125 kHz	200 kHz	250 kHz
0	1	0	1	φ/100	80.0 kHz	100 kHz	160 kHz	200 kHz
0	1	1	0	φ/112	71.4 kHz	89.3 kHz	143 kHz	179 kHz
0	1	1	1	φ/128	62.5 kHz	78.1 kHz	125 kHz	156 kHz
1	0	0	0	φ/56	143 kHz	179 kHz	286 kHz	357 kHz
1	0	0	1	φ/80	100 kHz	125 kHz	200 kHz	250 kHz
1	0	1	0	φ/96	83.3 kHz	104 kHz	167 kHz	208 kHz
1	0	1	1	φ/128	62.5 kHz	78.1 kHz	125 kHz	156 kHz
1	1	0	0	φ/160	50.0 kHz	62.5 kHz	100 kHz	125 kHz
1	1	0	1	φ/200	40.0 kHz	50.0 kHz	80.0 kHz	100 kHz
1	1	1	0	φ/224	35.7 kHz	44.6 kHz	71.4 kHz	89.3 kHz
1	1	1	1	φ/256	31.3 kHz	39.1 kHz	62.5 kHz	78.1 kHz

Note: * Outside the I²C bus interface specifications (standard mode: max. 100 kHz; high-speed mode: max. 400 kHz)

16.3.5 I²C Bus Control Register (ICCR)

ICCR controls the I²C bus interface and performs interrupt flag confirmation.

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface Enable 0: I²C bus interface modules are stopped and I²C bus interface module internal state is initialized. SAR and SARX can be accessed. 1: I²C bus interface modules can perform transfer operation, and the ports function as the SCL and SDA input/output pins. ICMR and ICDR can be accessed.
6	IEIC	0	R/W	I²C Bus Interface Interrupt Enable 0: Disables interrupts from the I²C bus interface to the CPU 1: Enables interrupts from the I²C bus interface to the CPU.
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode Both these bits will be cleared by hardware when they lose in a bus contention in master mode with the I²C bus format. In slave receive mode with I²C bus format, the R/W bit in the first frame immediately after the start condition sets these bits in receive mode or transmit mode automatically by hardware. Modification of the TRS bit during transfer is deferred until transfer is completed, and the changeover is made after completion of the transfer.

Bit	Bit Name	Initial Value	R/W	Description
5	MST	0	R/W	[MST clearing conditions]
4	TRS	0		1. When 0 is written by software
				2. When lost in bus contention in I²C bus format master mode
				[MST setting conditions]
				1. When 1 is written by software (for MST clearing condition 1)
				2. When 1 is written in MST after reading MST = 0 (for MST clearing condition 2)
				[TRS clearing conditions]
				1. When 0 is written by software (except for TRS setting condition 3)
				2. When 0 is written in TRS after reading TRS = 1 (for TRS setting condition 3)
				3. When lost in bus contention in I²C bus format master mode
				4. When the SW bit in DDCSWR is changed from 1 to 0
				[TRS setting conditions]
				1. When 1 is written by software (except for TRS clearing conditions 3 and 4)
				2. When 1 is written in TRS after reading TRS = 0 (for TRS clearing conditions 3 and 4)
				3. When 1 is received as the R/$\overline{W}$ bit after the first frame address matching in I²C bus format slave mode
3	ACKE	0	R/W	Acknowledge Bit Decision and Selection
				0: The value of the acknowledge bit is ignored, and continuous transfer is performed. The value of the received acknowledge bit is not indicated by the ACKB bit in ICSR, which is always 0.
				1: If the received acknowledge bit is 1, continuous transfer is halted.
				Depending on the receiving device, the acknowledge bit may be significant, in indicating completion of processing of the received data, for instance, or may be fixed at 1 and have no significance.

Bit	Bit Name	Initial Value	R/W	Description
2	BBSY	0	R/W	Bus Busy
0	SCP	1	W	Start Condition/Stop Condition Prohibit In master mode: • Writing 0 in BBSY and 0 in SCP: A stop condition is issued • Writing 1 in BBSY and 0 in SCP: A start condition and a restart condition are issued In slave mode: • Writing to the BBSY flag is disabled. [BBSY setting condition] When the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued. [BBSY clearing condition] When the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued. To issue a start/stop condition, use the MOV instruction. The I²C bus interface must be set in master transmit mode before the issue of a start condition. Set MST to 1 and TRS to 1 before writing 1 in BBSY and 0 in SCP. The BBSY flag can be read to check whether the I²C bus (SCL, SDA) is busy or free. The SCP bit is always read as 1. If 0 is written, the data is not stored.

Bit	Bit Name	Initial Value	R/W	Description
1	IRIC	0	R/(W)*	I²C Bus Interface Interrupt Request Flag Indicates that the I²C bus interface has issued an interrupt request to the CPU. IRIC is set at different times depending on the FS bit in SAR, the FSX bit in SARX, and the WAIT bit in ICMR. See section 16.4.7, IRIC Setting Timing and SCL Control. The conditions under which IRIC is set also differ depending on the setting of the ACKE bit in ICCR. [Setting conditions] I²C bus format master mode: • When a start condition is detected in the bus line state after a start condition is issued (when the ICDRE flag is set to 1 because of first frame transmission) • When a wait is inserted between the data and acknowledge bit when the WAIT bit is 1 (fall of the 8th transmit/receive clock) • At the end of data transfer (rise of the 9th transmit/receive clock while no wait is inserted) • When a slave address is received after bus arbitration is lost (the first frame after the start condition) • If 1 is received as the acknowledge bit (when the ACKB bit in ICSR is set to 1) when the ACKE bit is 1 • When the AL flag is set to 1 after bus arbitration is lost while the ALIE bit is 1 I²C bus format slave mode: • When the slave address (SVA or SVAX) matches (when the AAS or AASX flag in ICSR is set to 1) and at the end of data transfer up to the subsequent retransmission start condition or stop condition detection (rise of the 9th transmit/receive clock) • When the general call address is detected (when 0 is received as the R/$\overline{W}$ bit and the ADZ flag in ICSR is set to 1) and at the end of data reception up to the subsequent retransmission start condition or stop condition detection (rise of the 9th receive clock) • If 1 is received as the acknowledge bit (when the ACKB bit in ICSR is set to 1) while the ACKE bit is 1 • When a stop condition is detected (when the STOP or ESTP flag in ICSR is set to 1) while the STOPIM bit is 0

Bit	Bit Name	Initial Value	R/W	Description
1	IRIC	0	R/(W)*	Clocked synchronous serial format mode: • At the end of data transfer (rise of the 8th transmit/receive clock) • When a start condition is detected When the ICDRE or ICDRF flag is set to 1 in any operating mode: • When a start condition is detected in transmit mode (when a start condition is detected in transmit mode and the ICDRE flag is set to 1) • When data is transferred among the ICDR register and buffer (when data is transferred from ICDRT to ICDRS in transmit mode and the ICDRE flag is set to 1, or when data is transferred from ICDRS to ICDRR in receive mode and the ICDRF flag is set to 1) [Clearing conditions] • When 0 is written in IRIC after reading IRIC = 1 • When ICDR is read from or written to by the DTC (This may not function as a clearing condition depending on the situation. For details, see the description of the DTC operation given below.)

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

When the DTC is used, IRIC is cleared automatically and transfer can be performed continuously without CPU intervention.

When, with the I²C bus format selected, IRIC is set to 1 and an interrupt is generated, other flags must be checked in order to identify the source that set IRIC to 1. Although each source has a corresponding flag, caution is needed at the end of a transfer.

When the ICDRE or ICDRF flag is set, the IRTR flag may or may not be set. The IRTR flag (the DTC start request flag) is not set at the end of a data transfer up to detection of a retransmission start condition or stop condition after a slave address (SVA) or general call address match in I²C bus format slave mode.

Even when the IRIC flag and IRTR flag are set, the ICDRE or ICDRF flag may not be set. The IRIC and IRTR flags are not cleared at the end of the specified number of transfers in continuous transfer using the DTC. The ICDRE or ICDRF flag is cleared, however, since the specified number of ICDR reads or writes have been completed.

Tables 16.4 and 16.5 show the relationship between the flags and the transfer states.

Table 16.4 Flags and Transfer States (Master Mode)

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	ICDRF	ICDRE	State
1	1	0	0	0	0	0↓	0	0↓	0↓	0	—	0	Idle state (flag clearing required)
1	1	1↑	0	0	1↑	0	0	0	0	0	—	1↑	Start condition detected
1	—	1	0	0	—	0	0	0	0	—	—	—	Wait state
1	1	1	0	0	—	0	0	0	0	1↑	—	—	Transmission end (ACKE = 1 and ACKB = 1)
1	1	1	0	0	1↑	0	0	0	0	0	—	1↑	Transmission end with ICDRE = 0
1	1	1	0	0	—	0	0	0	0	0	—	0↓	ICDR write with the above state
1	1	1	0	0	—	0	0	0	0	0	—	1	Transmission end with ICDRE = 1
1	1	1	0	0	—	0	0	0	0	0	—	0↓	ICDR write with the above state or after start condition detected
1	1	1	0	0	1↑	0	0	0	0	0	—	1↑	Automatic data transfer from ICDRT to ICDRS with the above state
1	0	1	0	0	1↑	0	0	0	0	—	1↑	—	Reception end with ICDRF = 0
1	0	1	0	0	—	0	0	0	0	—	0↓	—	ICDR read with the above state
1	0	1	0	0	—	0	0	0	0	—	1	—	Reception end with ICDRF = 1
1	0	1	0	0	—	0	0	0	0	—	0↓	—	ICDR read with the above state
1	0	1	0	0	1↑	0	0	0	0	—	1↑	—	Automatic data transfer from ICDRS to ICDRR with the above state
0↓	0↓	1	0	0	—	0	1↑	0	0	—	—	—	Arbitration lost
1	—	0↓	0	0	—	0	0	0	0	—	—	0↓	Stop condition detected

[Legend]

0: 0-state retained

1: 1-state retained

—: Previous state retained

0↓: Cleared to 0

1↑: Set to 1

Table 16.5 Flags and Transfer States (Slave Mode)

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	ICDRF	ICDRE	State
0	0	0	0	0	0	0	0	0	0	0	—	0	Idle state (flag clearing required)
0	0	1↑	0	0	0	0↓	0	0	0	0	—	1↑	Start condition detected
0	1↑/0 *1	1	0	0	0	0	—	1↑	0	0	1↑	1	SAR match in first frame (SARX ≠ SAR)
0	0	1	0	0	0	0	—	1↑	1↑	0	1↑	1	General call address match in first frame (SARX ≠ H'00)
0	1↑/0 *1	1	0	0	1↑	1↑	—	0	0	0	1↑	1	SARS match in first frame (SAR ≠ SARX)
0	1	1	0	0	—	—	—	—	0	1↑	—	—	Transmission end (ACKE = 1 and ACKB = 1)
0	1	1	0	0	1↑/0 *1	—	—	—	0	0	—	1↑	Transmission end with ICDRE = 0
0	1	1	0	0	—	—	0↓	0↓	0	0	—	0↓	ICDR write with the above state
0	1	1	0	0	—	—	—	—	1	0		1	Transmission end with ICDRE = 1
0	1	1	0	0	—	—	0↓	0↓	0	0		0↓	ICDR write with the above state
0	1	1	0	0	1↑/0 *2	—	0	0	0	0		1↑	Automatic data transfer from ICDRT to ICDRS with the above state
0	0	1	0	0	1↑/0 *2	—	—	—	—	—	1↑	—	Reception end with ICDRF = 0
0	0	1	0	0	—	—	0↓	0↓	0↓	—	0↓	—	ICDR read with the above state

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	ICDRF	ICDRE	State
0	0	1	0	0	—	—	—	—	—	—	1	—	Reception end with ICDRF = 1
0	0	1	0	0	—	—	0↓	0↓	0↓	—	0↓	—	ICDR read with the above state
0	0	1	0	0	1↑/0 *2	—	0	0	0	—	1↑	—	Automatic data transfer from ICDRS to ICDRR with the above state
0	—	0↓	1↑/0 *3	0/1↑ *3	—	—	—	—	—	—	—	0↓	Stop condition detected

[Legend]

0: 0-state retained

1: 1-state retained

—: Previous state retained

0↓: Cleared to 0

1↑: Set to 1

Notes: 1. Set to 1 when 1 is received as a R/$\overline{W}$ bit following an address.

2. Set to 1 when the AASX bit is set to 1.

3. When ESTP = 1, STOP is 0, or when STOP = 1, ESTP is 0.

16.3.6 I²C Bus Status Register (ICSR)

ICSR consists of status flags. Also see tables 16.4 and 16.5.

Bit	Bit Name	Initial Value	R/W	Description
7	ESTP	0	R/(W)*	Error Stop Condition Detection Flag This bit is valid in I²C bus format slave mode. [Setting condition] When a stop condition is detected during frame transfer. [Clearing conditions] • When 0 is written in ESTP after reading ESTP = 1 • When the IRIC flag in ICCR is cleared to 0
6	STOP	0	R/(W)*	Normal Stop Condition Detection Flag This bit is valid in I²C bus format slave mode. [Setting condition] When a stop condition is detected after frame transfer completion. [Clearing conditions] • When 0 is written in STOP after reading STOP = 1 • When the IRIC flag is cleared to 0

Bit	Bit Name	Initial Value	R/W	Description
5	IRTR	0	R/(W)*	I²C Bus Interface Continuous Transfer Interrupt Request Flag Indicates that the I²C bus interface has issued an interrupt request to the CPU, and the source is completion of reception/transmission of one frame in continuous transmission/reception for which DTC activation is possible. When the IRTR flag is set to 1, the IRIC flag is also set to 1 at the same time. [Setting conditions] I²C bus format slave mode: • When the ICDRE or ICDRF flag in ICDR is set to 1 when AASX = 1 Master mode or clocked synchronous serial format mode with I²C bus format, or formatless mode: • When the ICDRE or ICDRF flag is set to 1 [Clearing conditions] • When 0 is written after reading IRTR = 1 • When the IRIC flag is cleared to 0 while ICE is 1
4	AASX	0	R/(W)*	Second Slave Address Recognition Flag In I²C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVAX6 to SVAX0 in SARX. [Setting condition] When the second slave address is detected in slave receive mode and FSX = 0 in SARX [Clearing conditions] • When 0 is written in AASX after reading AASX = 1 • When a start condition is detected • In master mode

Bit	Bit Name	Initial Value	R/W	Description
3	AL	0	R/(W)*	Arbitration Lost Flag Indicates that arbitration was lost in master mode. [Setting conditions] When ALSL = 0 • If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode • If the internal SCL line is high at the fall of SCL in master transmit mode When ALSL = 1 • If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode • If the SDA pin is driven low by another device before the I²C bus interface drives the SDA pin low, after the start condition instruction was executed in master transmit mode [Clearing conditions] • When ICDR is written to (transmit mode) or read from (receive mode) • When 0 is written in AL after reading AL = 1
2	AAS	0	R/(W)*	Slave Address Recognition Flag In I²C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR, or if the general call address (H'00) is detected. [Setting condition] When the slave address or general call address (one frame including a R/$\overline{W}$ bit is H'00) is detected in slave receive mode and FS = 0 in SAR [Clearing conditions] • When ICDR is written to (transmit mode) or read from (receive mode) • When 0 is written in AAS after reading AAS = 1 • In master mode

Bit	Bit Name	Initial Value	R/W	Description
1	ADZ	0	R/(W)*	General Call Address Recognition Flag In I²C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition is the general call address (H'00). [Setting condition] When the general call address (one frame including a R/$\overline{W}$ bit is H'00) is detected in slave receive mode and FS = 0 or FSX = 0 [Clearing conditions] • When ICDR is written to (transmit mode) or read from (receive mode) • When 0 is written in ADZ after reading ADZ = 1 • In master mode If a general call address is detected while FS=1 and FSX=0, the ADZ flag is set to 1; however, the general call address is not recognized (AAS flag is not set to 1).

Bit	Bit Name	Initial Value	R/W	Description
0	ACKB	0	R/W	Acknowledge Bit Stores acknowledge data. Transmit mode: [Setting condition] When 1 is received as the acknowledge bit when ACKE = 1 in transmit mode [Clearing conditions] • When 0 is received as the acknowledge bit when ACKE = 1 in transmit mode • When 0 is written to the ACKE bit Receive mode: 0: Returns 0 as acknowledge data after data reception 1: Returns 1 as acknowledge data after data reception When this bit is read, the value loaded from the bus line (returned by the receiving device) is read in transmission (when TRS = 1). In reception (when TRS = 0), the value set by internal software is read. When this bit is written, acknowledge data that is returned after receiving is rewritten regardless of the TRS value. If the ICSR register bit is written using bit-manipulation instructions, the acknowledge data should be re-set since the acknowledge data setting is rewritten by the ACKB bit reading value. Write the ACKE bit to 0 to clear the ACKB flag to 0, before transmission is ended and a stop condition is issued in master mode, or before transmission is ended and SDA is released to issue a stop condition by a master device.

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

16.3.7 DDC Switch Register (DDCSWR)

DDCSWR controls the IIC internal latch clearance.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R/W	Reserved The initial value should not be changed.
4	—	0	R	DDC Mode Switch Interrupt Flag Indicates an interrupt request to the CPU is generated when automatic format switching is executed for IIC_0. [Setting condition] When a falling edge is detected on the SCL pin when SWE = 1 [Clearing condition] When 0 is written in IF after reading IF = 1
3	CLR3	1	W*	IIC Clear 3 to 0 Controls initialization of the internal state of IIC_0 and IIC_1. 00--: Setting prohibited 0100: Setting prohibited 0101: IIC_0 internal latch cleared 0110: IIC_1 internal latch cleared 0111: IIC_0 and IIC_1 internal latches cleared 1---: Invalid setting When a write operation is performed on these bits, a clear signal is generated for the internal latch circuit of the corresponding module, and the internal state of the IIC module is initialized. These bits can only be written to; they are always read as 1. Write data to this bit is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR. When clearing is required again, all the bits must be written to in accordance with the setting.
2	CLR2	1	W*	
1	CLR1	1	W*	
0	CLR0	1	W*	

Note: * This bit is always read as 1.

16.3.8 I²C Bus Extended Control Register (ICXR)

ICXR enables or disables the I²C bus interface interrupt generation and continuous receive operation, and indicates the status of receive/transmit operations.

Bit	Bit Name	Initial Value	R/W	Description
7	STOPIM	0	R/W	Stop Condition Interrupt Source Mask Enables or disables the interrupt generation when the stop condition is detected in slave mode. 0: Enables IRIC flag setting and interrupt generation when the stop condition is detected (STOP = 1 or ESTP = 1) in slave mode. 1: Disables IRIC flag setting and interrupt generation when the stop condition is detected.
6	HNDS	0	R/W	Handshake Receive Operation Select Enables or disables continuous receive operation in receive mode. 0: Enables continuous receive operation 1: Disables continuous receive operation When the HNDS bit is cleared to 0, receive operation is performed continuously after data has been received successfully while ICDRF flag is 0. When the HNDS bit is set to 1, SCL is fixed to the low level and the next data transfer is disabled after data has been received successfully while the ICDRF flag is 0. The bus line is released and next receive operation is enabled by reading the receive data in ICDR.

Bit	Bit Name	Initial Value	R/W	Description
5	ICDRF	0	R	Receive Data Read Request Flag Indicates the ICDR (ICDRR) status in receive mode. 0: Indicates that the data has been already read from ICDR (ICDRR) or ICDR is initialized. 1: Indicates that data has been received successfully and transferred from ICDRS to ICDRR, and the data is ready to be read out. [Setting conditions] • When data is received successfully and transferred from ICDRS to ICDRR. 1. When data is received successfully while ICDRF = 0 (at the rise of the 9th clock pulse). 2. When ICDR is read successfully in receive mode after data was received while ICDRF = 1. [Clearing conditions] • When ICDR (ICDRR) is read. • When 0 is written to the ICE bit. • When the IIC is internally initialized using the CLR3 to CLR0 bits in DDCSWR. When ICDRF is set due to the condition (2) above, ICDRF is temporarily cleared to 0 when ICDR (ICDRR) is read; however, since data is transferred from ICDRS to ICDRR immediately, ICDRF is set to 1 again. Note that ICDR cannot be read successfully in transmit mode (TRS = 1) because data is not transferred from ICDRS to ICDRR. Be sure to read data from ICDR in receive mode (TRS = 0).

Bit	Bit Name	Initial Value	R/W	Description
4	ICDRE	0	R	Transmit Data Write Request Flag Indicates the ICDR (ICDRT) status in transmit mode. 0: Indicates that the data has been already written to ICDR (ICDRT) or ICDR is initialized. 1: Indicates that data has been transferred from ICDRT to ICDRS and is being transmitted, or the start condition has been detected or transmission has been complete, thus allowing the next data to be written to. [Setting conditions] • When the start condition is detected from the bus line state with I²C bus format or serial format. • When data is transferred from ICDRT to ICDRS. 1. When data transmission completed while ICDRE = 0 (at the rise of the 9th clock pulse). 2. When data is written to ICDR in transmit mode after data transmission was completed while ICDRE = 1. [Clearing conditions] • When data is written to ICDR (ICDRT). • When the stop condition is detected with I²C bus format or serial format. • When 0 is written to the ICE bit. • When the IIC is internally initialized using the CLR3 to CLR0 bits in DDCSWR. Note that if the ACKE bit is set to 1 with I²C bus format thus enabling acknowledge bit decision, ICDRE is not set when data transmission is completed while the acknowledge bit is 1. When ICDRE is set due to the condition (2) above, ICDRE is temporarily cleared to 0 when data is written to ICDR (ICDRT); however, since data is transferred from ICDRT to ICDRS immediately, ICDRE is set to 1 again. Do not write data to ICDR when TRS = 0 because the ICDRE flag value is invalid during the time.

Bit	Bit Name	Initial Value	R/W	Description
3	ALIE	0	R/W	Arbitration Lost Interrupt Enable Enables or disables IRIC flag setting and interrupt generation when arbitration is lost. 0: Disables interrupt request when arbitration is lost. 1: Enables interrupt request when arbitration is lost.
2	ALSL	0	R/W	Arbitration Lost Condition Select Selects the condition under which arbitration is lost. 0: When the SDA pin state disagrees with the data that IIC bus interface outputs at the rise of SCL, or when the SCL pin is driven low by another device. 1: When the SDA pin state disagrees with the data that IIC bus interface outputs at the rise of SCL, or when the SDA line is driven low by another device in idle state or after the start condition instruction was executed.
1	FNC1	0	R/W	Function Bit
0	FNC0	0	R/W	Cancels some restrictions on usage. For details, refer to section 16.6, Usage Notes. 00: Restrictions on operation remaining in effect 01: Setting prohibited 10: Setting prohibited 11: Restrictions on operation canceled

16.4 Operation

The I²C bus interface has an I²C bus format and a serial format.

16.4.1 I²C Bus Data Format

The I²C bus format is an addressing format with an acknowledge bit. This is shown in figure 16.3. The first frame following a start condition always consists of 9 bits.

The serial format is a non-addressing format with no acknowledge bit. This is shown in figure 16.4.

Figure 16.5 shows the I²C bus timing.

The symbols used in figures 16.3 to 16.5 are explained in table 16.6.

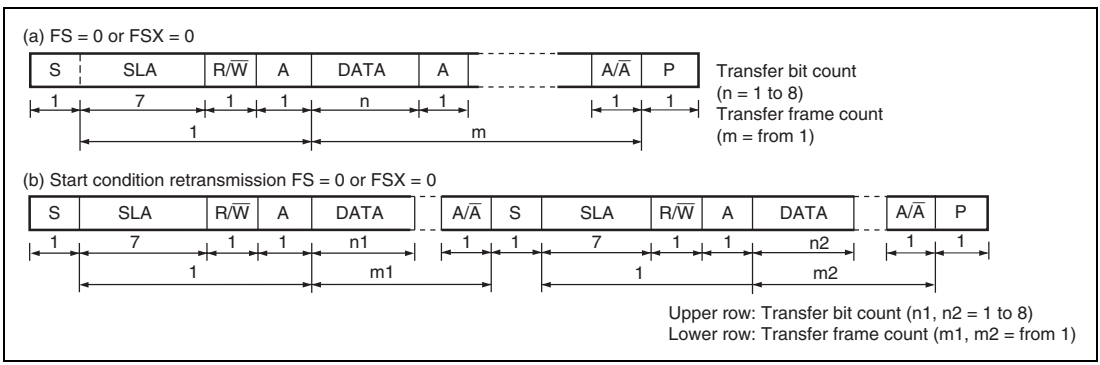


Figure 16.3 I²C Bus Data Format (I²C Bus Format)

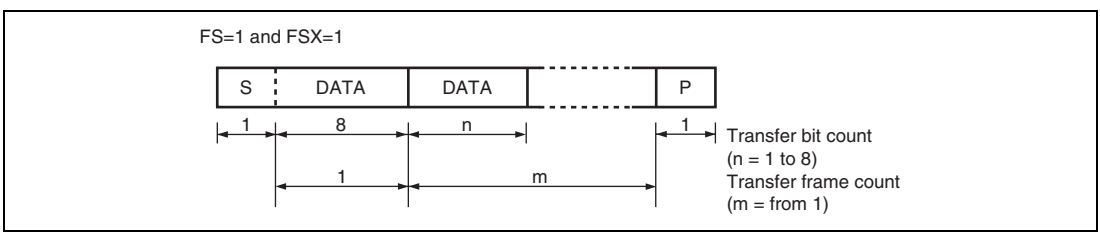


Figure 16.4 I²C Bus Data Format (Serial Format)

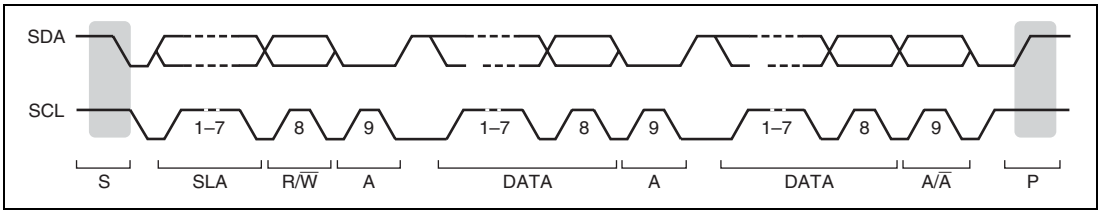


Figure 16.5 I²C Bus Timing

Table 16.6 I²C Bus Data Format Symbols

Legend	
S	Start condition. The master device drives SDA from high to low while SCL is high
SLA	Slave address. The master device selects the slave device.
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 receiving device drives SDA low to acknowledge a transfer. (The slave device returns acknowledge in master transmit mode, and the master device returns acknowledge in master receive mode.)
DATA	Transferred data. The bit length of transferred data is set with the BC2 to BC0 bits in ICMR. The MSB first or LSB first is switched with the MLS bit in ICMR.
P	Stop condition. The master device drives SDA from low to high while SCL is high

16.4.2 Initialization

Initialize the IIC by the procedure shown in figure 16.6 before starting transmission/reception of data.

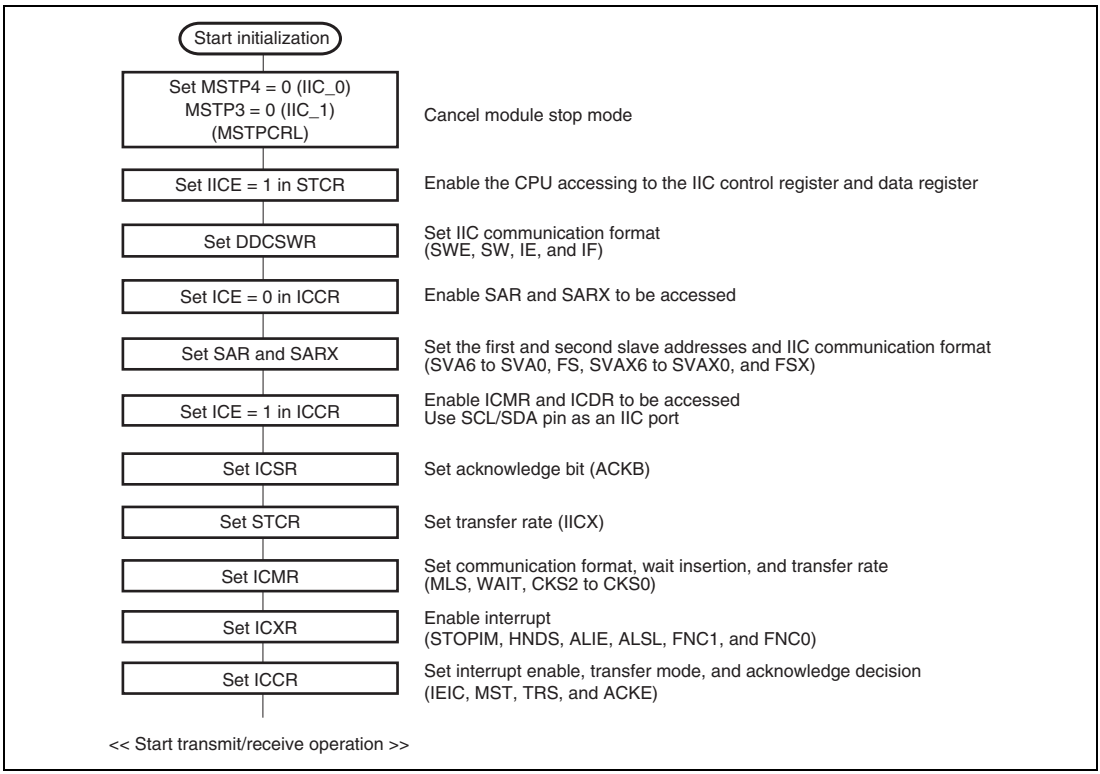


Figure 16.6 Sample Flowchart for IIC Initialization

Note: Be sure to modify the ICMR register after transmit/receive operation has been completed. If the ICMR register is modified during transmit/receive operation, bit counter BC2 to BC0 will be modified erroneously, thus causing incorrect operation.

16.4.3 Master Transmit Operation

In I²C bus format master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal.

Figure 16.7 shows the sample flowchart for the operations in master transmit mode.

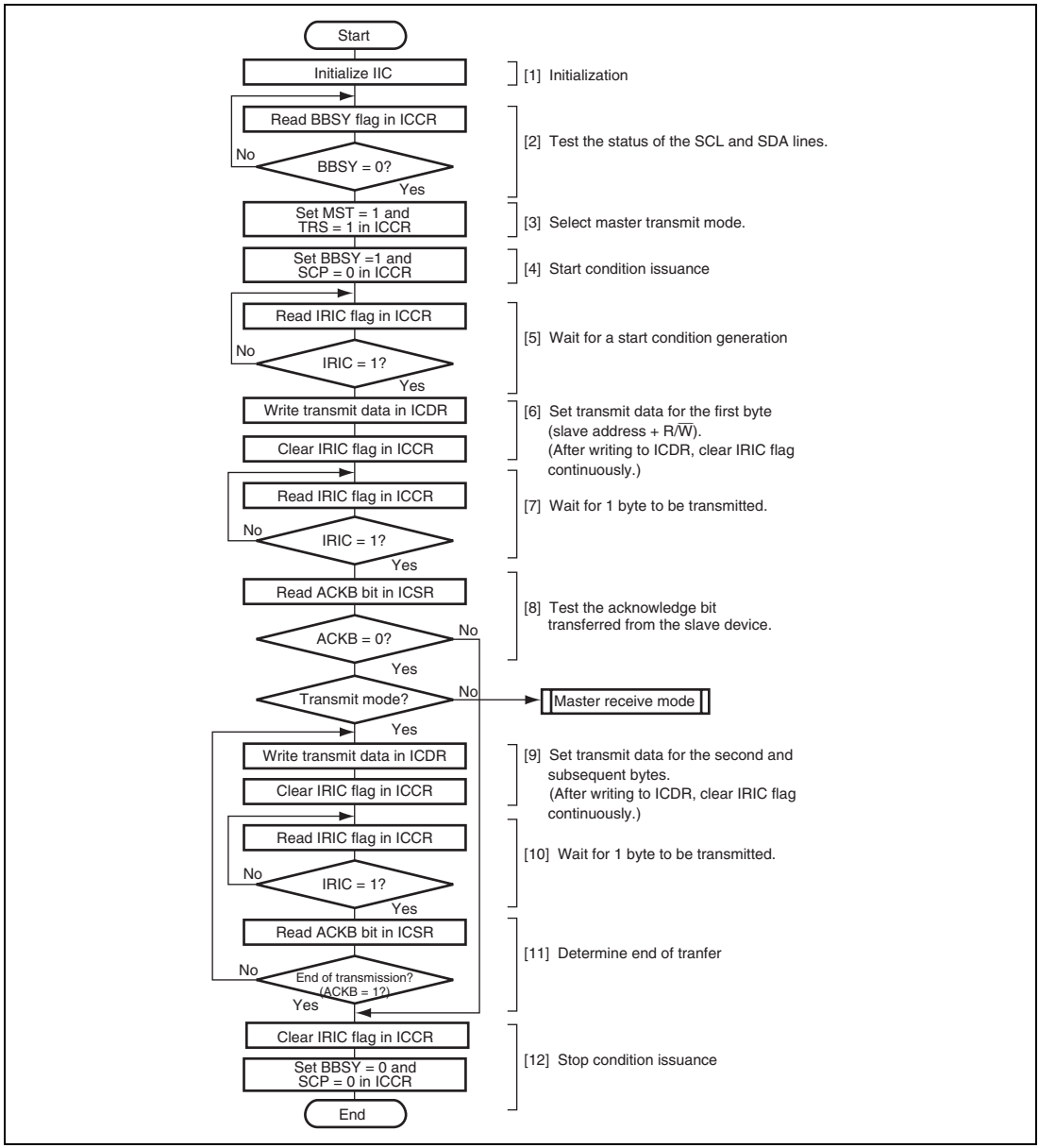


Figure 16.7 Sample Flowchart for Operations in Master Transmit Mode

The transmission procedure and operation, by which data is sequentially transmitted in synchronization with ICDR (ICDRT) write operations, are described below.

1. Initialize the IIC as described in section 16.4.2, Initialization.
2. Read the BBSY flag in ICCR to confirm that the bus is free.
3. Set bits MST and TRS to 1 in ICCR to select master transmit mode.
4. Write 1 to BBSY and 0 to SCP in ICCR. This changes SDA from high to low when SCL is high, and generates the start condition.
5. Then the IRIC and IRTR flags are set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU.
6. Write the data (slave address + R/$\overline{W}$) to ICDR.

   With the I²C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the first frame data following the start condition indicates the 7-bit slave address and transmit/receive direction (R/$\overline{W}$).

   To determine the end of the transfer, the IRIC flag is cleared to 0. After writing to ICDR, clear IRIC continuously so no other interrupt handling routine is executed. If the time for transmission of one frame of data has passed before the IRIC clearing, the end of transmission cannot be determined. The master device sequentially sends the transmission clock and the data written to ICDR. The selected slave device (i.e. the slave device with the matching slave address) drives SDA low at the 9th transmit clock pulse and returns an acknowledge signal.
7. When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of the 9th transmit clock pulse. After one frame has been transmitted, SCL is automatically fixed low in synchronization with the internal clock until the next transmit data is written.
8. Read the ACKB bit in ICSR to confirm that ACKB is cleared to 0. When the slave device has not acknowledged (ACKB bit is 1), operate step [12] to end transmission, and retry the transmit operation.
9. Write the transmit data to ICDR.

   As indicating the end of the transfer, the IRIC flag is cleared to 0. Perform the ICDR write and the IRIC flag clearing sequentially, just as in step [6]. Transmission of the next frame is performed in synchronization with the internal clock.
10. When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of the 9th transmit clock pulse. After one frame has been transmitted, SCL is automatically fixed low in synchronization with the internal clock until the next transmit data is written.
11. Read the ACKB bit in ICSR.

    Confirm that the slave device has been acknowledged (ACKB bit is 0). When there is still data to be transmitted, go to step [9] to continue the next transmission operation. When the slave device has not acknowledged (ACKB bit is set to 1), operate step [12] to end transmission.

12. Clear the IRIC flag to 0.

    Write 0 to ACKE in ICCR, to clear received ACKB contents to 0.

    Write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.

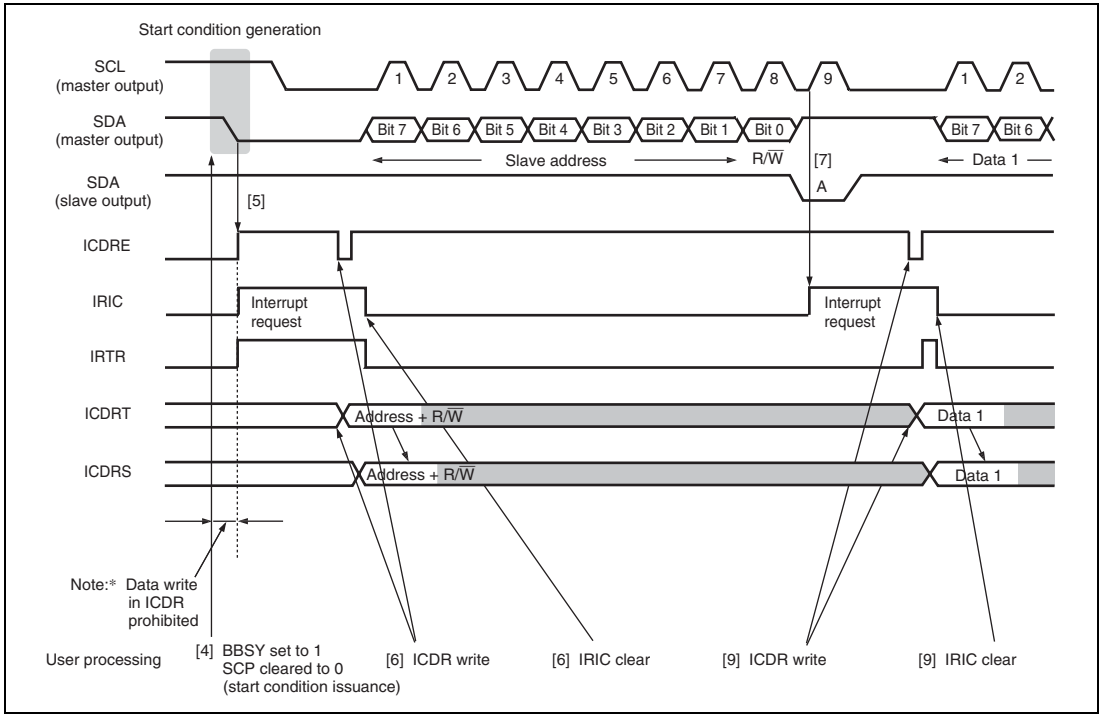


Figure 16.8 Example of Operation Timing in Master Transmit Mode (MLS = WAIT = 0)

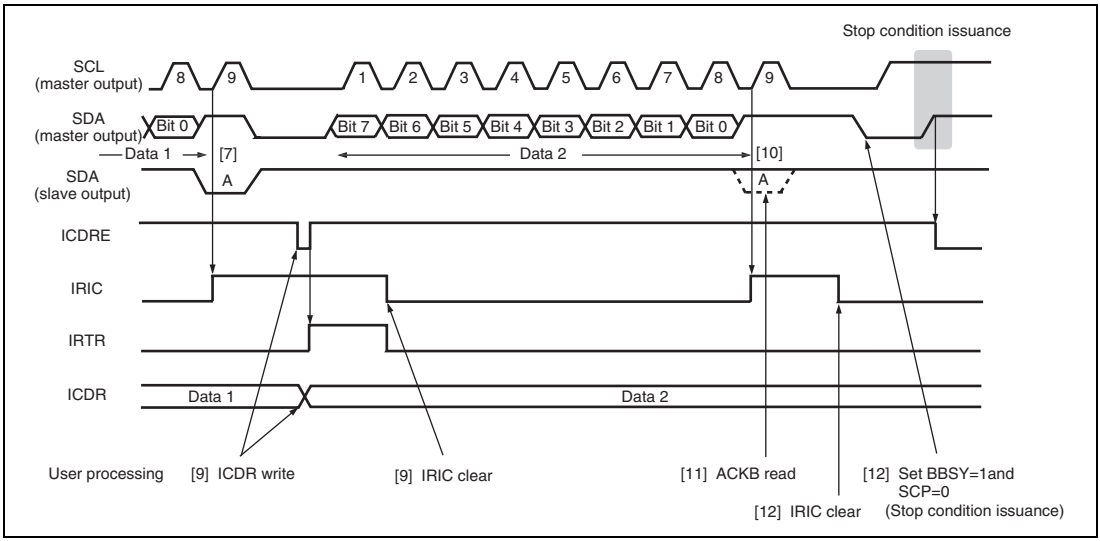


Figure 16.9 Example of Stop Condition Issuance Operation Timing in Master Transmit Mode (MLS = WAIT = 0)

16.4.4 Master Receive Operation

In I²C bus format master receive mode, the master device outputs the receive clock, receives data, and returns an acknowledge signal. The slave device transmits data.

The master device transmits data containing the slave address and R/$\overline{W}$ (1: read) in the first frame following the start condition issuance in master transmit mode, selects the slave device, and then switches the mode for receive operation.

(1) Receive Operation Using the HNDS Function (HNDS = 1)

Figure 16.10 shows the sample flowchart for the operations in master receive mode (HNDS = 1).

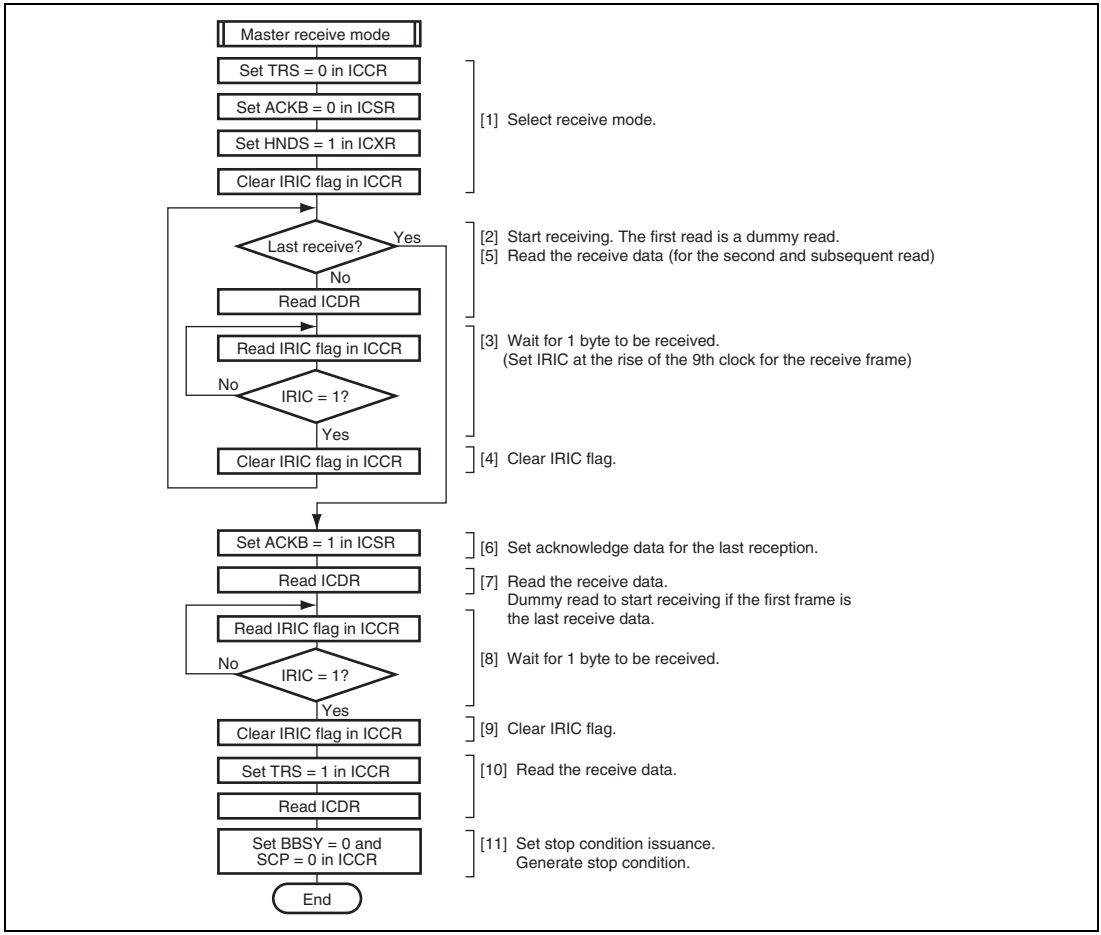


Figure 16.10 Sample Flowchart for Operations in Master Receive Mode (HNDS = 1)

The reception procedure and operations using the HNDS function, by which the data reception process is provided in 1-byte units with SCL fixed low at each data reception, are described below.

1. Clear the TRS bit in ICCR to 0 to switch from transmit mode to receive mode.
   Clear the ACKB bit in ICSR to 0 (acknowledge data setting).
   Set the HNDS bit in ICXR to 1.
   Clear the IRIC flag to 0 to determine the end of reception.
   Go to step [6] to halt reception operation if the first frame is the last receive data.
2. When ICDR is read (dummy data read), reception is started, the receive clock is output in synchronization with the internal clock, and data is received. (Data from the SDA pin is sequentially transferred to ICDRS in synchronization with the rise of the receive clock pulses.)
3. The master device drives SDA low to return the acknowledge data at the 9th receive clock pulse. The receive data is transferred from ICDRS to ICDRR at the rise of the 9th clock pulse, setting the ICDRF, IRIC, and IRTR flags to 1. If the IEIC bit has been set to 1, an interrupt request is sent to the CPU.
   The master device drives SCL low from the fall of the 9th receive clock pulse to the ICDR data reading.
4. Clear the IRIC flag to determine the next interrupt.
   Go to step [6] to halt reception operation if the next frame is the last receive data.
5. Read ICDR receive data. This clears the ICDRF flag to 0. The master device outputs the receive clock continuously to receive the next data.

Data can be received continuously by repeating steps [3] to [5].

6. Set the ACKB bit to 1 so as to return the acknowledge data for the last reception.
7. Read ICDR receive data. This clears the ICDRF flag to 0. The master device outputs the receive clock to receive data.
8. When one frame of data has been received, the ICDRF, IRIC, and IRTR flags are set to 1 at the rise of the 9th receive clock pulse.
9. Clear the IRIC flag to 0.
10. Read ICDR receive data after setting the TRS bit. This clears the ICDRF flag to 0.
11. Clear the BBSY bit and SCP bit to 0 in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.

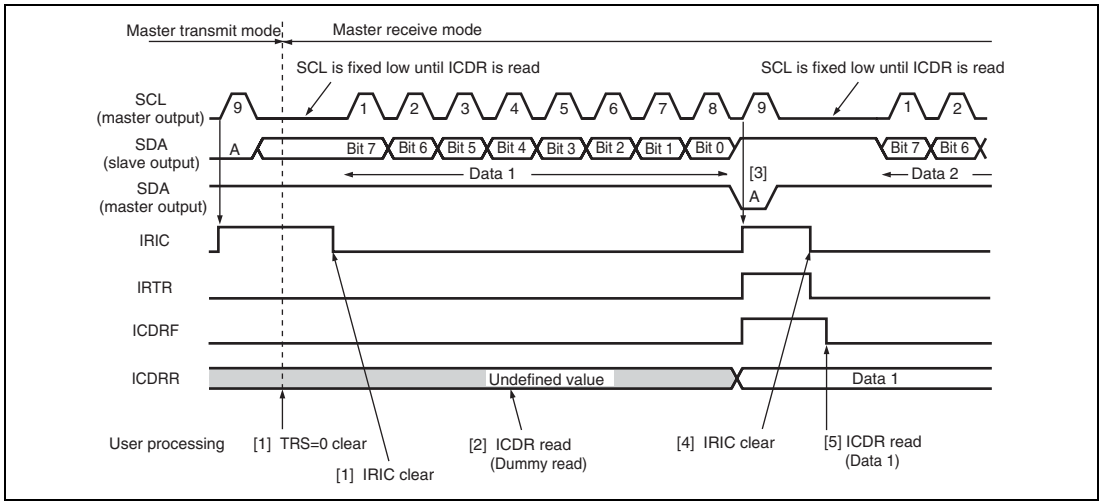


Figure 16.11 Example of Operation Timing in Master Receive Mode (MLS = WAIT = 0, HNDS = 1)

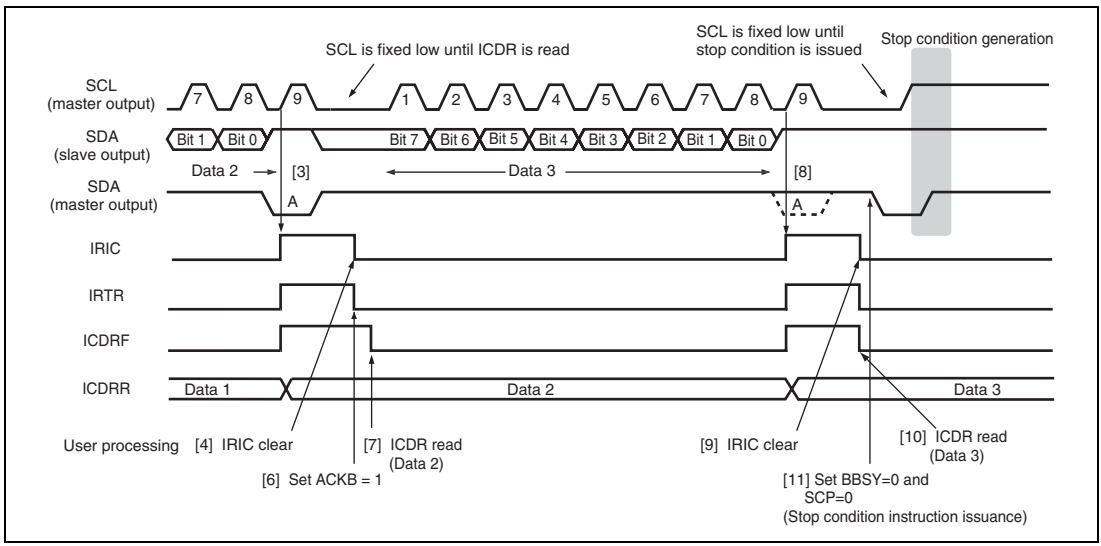


Figure 16.12 Example of Stop Condition Issuance Operation Timing in Master Receive Mode (MLS = WAIT = 0, HNDS = 1)

(2) Receive Operation Using the Wait Function

Figures 16.13 and 16.14 show the sample flowcharts for the operations in master receive mode (WAIT = 1).

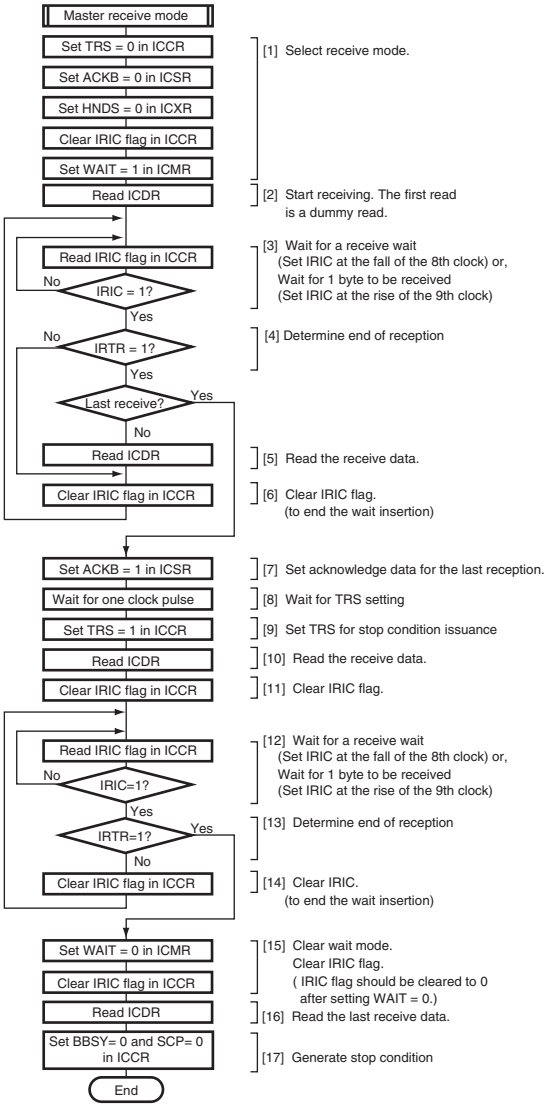


Figure 16.13 Sample Flowchart for Operations in Master Receive Mode (receiving multiple bytes) (WAIT = 1)

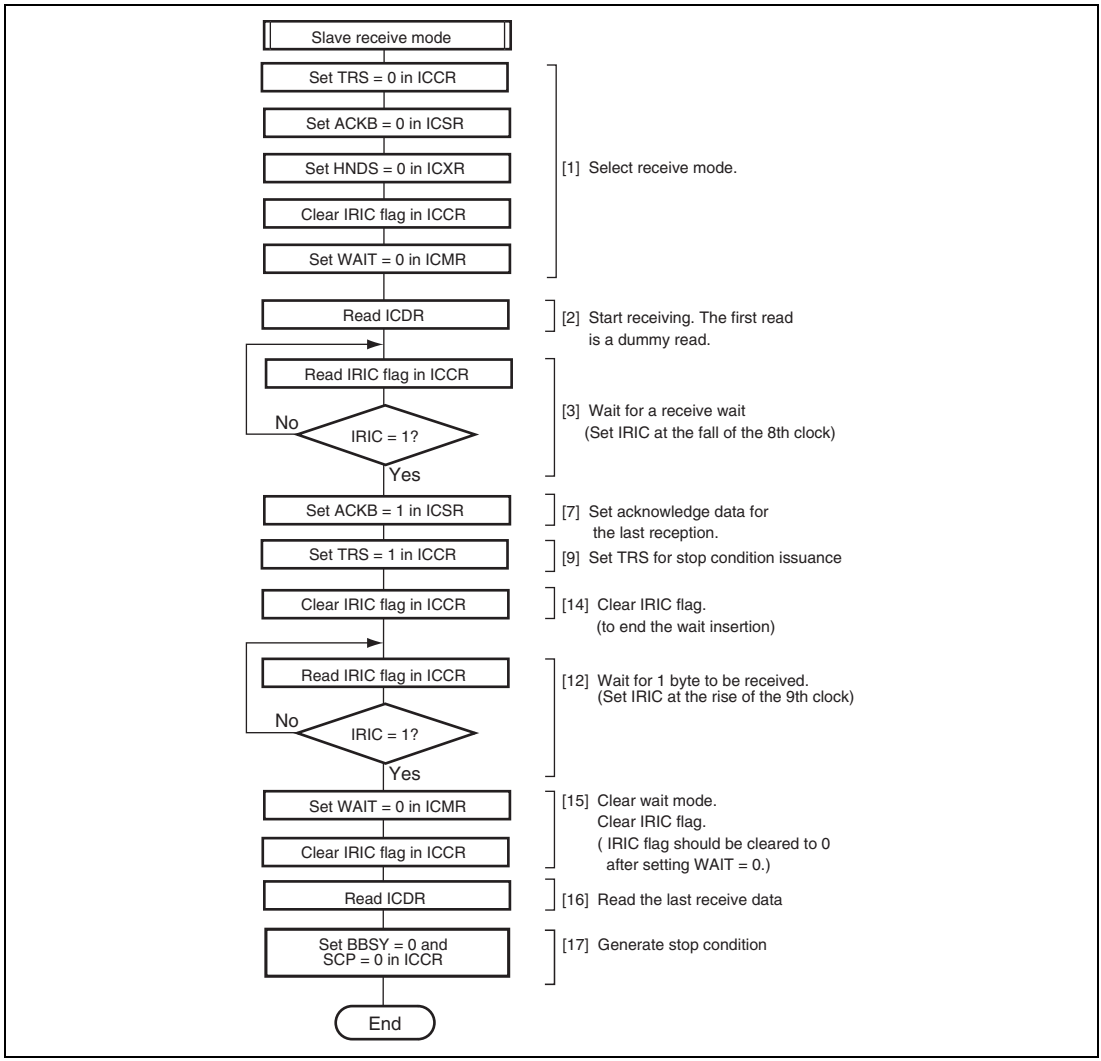


Figure 16.14 Sample Flowchart for Operations in Master Receive Mode (receiving a single byte) (WAIT = 1)

The reception procedure and operations using the wait function (WAIT bit), by which data is sequentially received in synchronization with ICDR (ICDRR) read operations, are described below.

The following describes the multiple-byte reception procedure. In single-byte reception, some steps of the following procedure are omitted. At this time, follow the procedure shown in figure 16.14.

1. Clear the TRS bit in ICCR to 0 to switch from transmit mode to receive mode.
   Clear the ACKB bit in ICSR to 0 to set the acknowledge data.
   Clear the HNDS bit in ICXR to 0 to cancel the handshake function.
   Clear the IRIC flag to 0, and then set the WAIT bit in ICMR to 1.
2. When ICDR is read (dummy data is read), reception is started, the receive clock is output in synchronization with the internal clock, and data is received.
3. The IRIC flag is set to 1 in either of the following cases. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU.
   — At the fall of the 8th receive clock pulse for one frame
     SCL is automatically fixed low in synchronization with the internal clock until the IRIC flag clearing.
   — At the rise of the 9th receive clock pulse for one frame
     The IRTR and ICDRF flags are set to 1, indicating that one frame of data has been received. The master device outputs the receive clock continuously to receive the next data.
4. Read the IRTR flag in ICSR.
   If the IRTR flag is 0, execute step [6] to clear the IRIC flag to 0 to release the wait state.
   If the IRTR flag is 1 and the next data is the last receive data, execute step [7] to halt reception.
5. If IRTR flag is 1, read ICDR receive data.
6. Clear the IRIC flag. When the flag is set as the first case in step [3], the master device outputs the 9th clock and drives SDA low at the 9th receive clock pulse to return an acknowledge signal.

Data can be received continuously by repeating steps [3] to [6].

7. Set the ACKB bit in ICSR to 1 so as to return the acknowledge data for the last reception.
8. After the IRIC flag is set to 1, wait for at least one clock pulse until the rise of the first clock pulse for the next receive data.
9. Set the TRS bit in ICCR to 1 to switch from receive mode to transmit mode. The TRS bit value becomes valid when the rising edge of the next 9th clock pulse is input.
10. Read the ICDR receive data.

11. Clear the IRIC flag to 0.
12. The IRIC flag is set to 1 in either of the following cases.
    - At the fall of the 8th receive clock pulse for one frame

      SCL is automatically fixed low in synchronization with the internal clock until the IRIC flag is cleared.
    - At the rise of the 9th receive clock pulse for one frame

      The IRTR and ICDRF flags are set to 1, indicating that one frame of data has been received. The master device outputs the receive clock continuously to receive the next data.
13. Read the IRTR flag in ICSR.

    If the IRTR flag is 0, execute step [14] to clear the IRIC flag to 0 to release the wait state.

    If the IRTR flag is 1 and data reception is complete, execute step [15] to issue the stop condition.
14. If IRTR flag is 0, clear the IRIC flag to 0 to release the wait state.

    Execute step [12] to read the IRIC flag to detect the end of reception.
15. Clear the WAIT bit in ICMR to cancel the wait mode.

    Then, clear the IRIC flag. Clearing of the IRIC flag should be done while WAIT = 0. (If the WAIT bit is cleared to 0 after clearing the IRIC flag and then an instruction to issue a stop condition is executed, the stop condition may not be issued correctly.)
16. Read the last ICDR receive data.
17. Clear the BBSY bit and SCP bit to 0 in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.

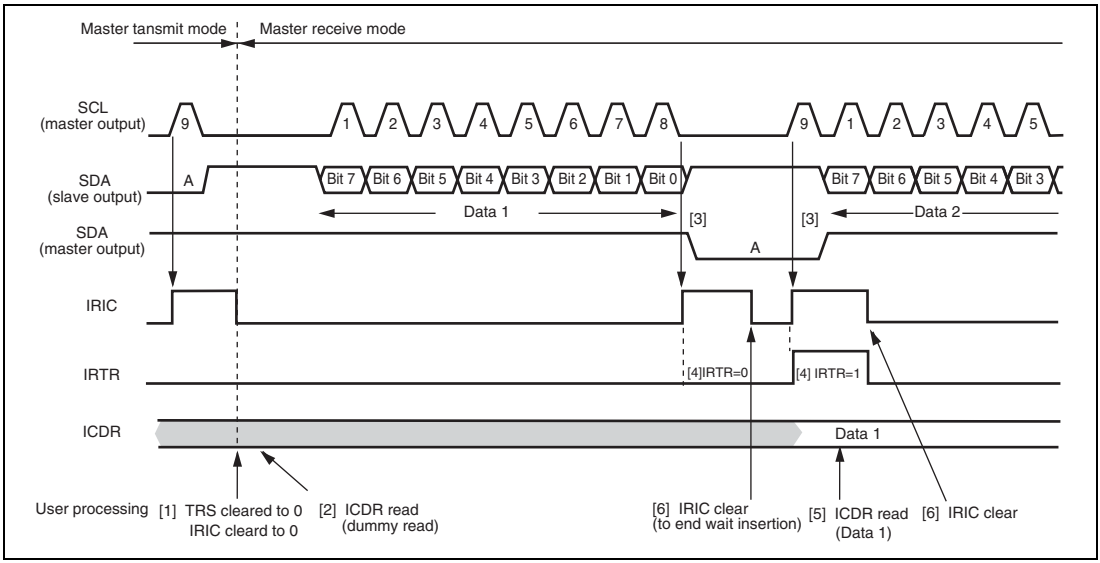


Figure 16.15 Example of Master Receive Mode Operation Timing (MLS = ACKB = 0, WAIT = 1)

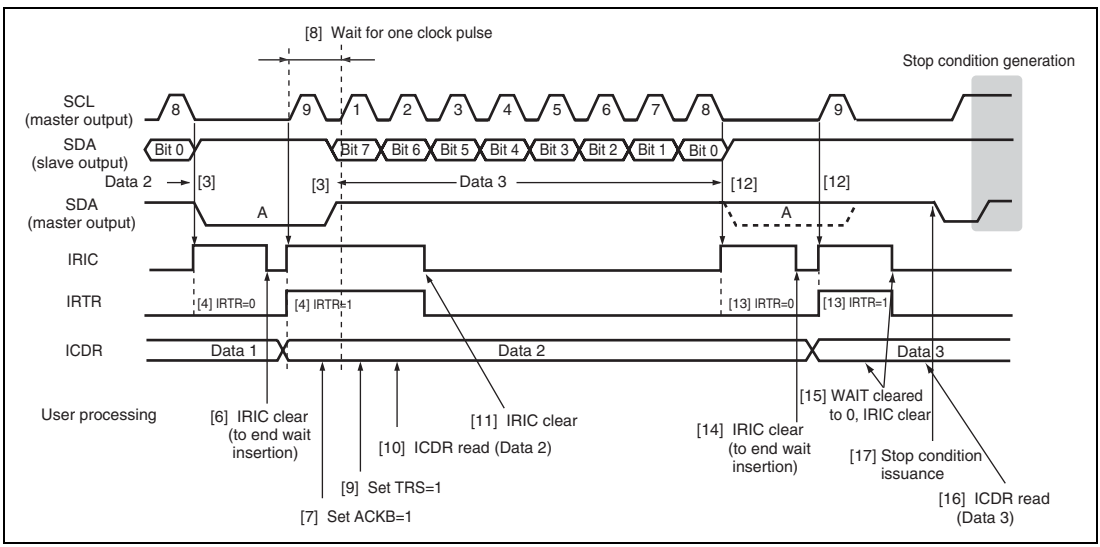


Figure 16.16 Example of Stop Condition Issuance Timing in Master Receive Mode (MLS = ACKB = 0, WAIT = 1)

16.4.5 Slave Receive Operation

In I²C bus format slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal.

The slave device operates as the device specified by the master device when the slave address in the first frame following the start condition that is issued by the master device matches its own address.

(1) Receive Operation Using the HNDS Function (HNDS = 1)

Figure 16.17 shows the sample flowchart for the operations in slave receive mode (HNDS = 1).

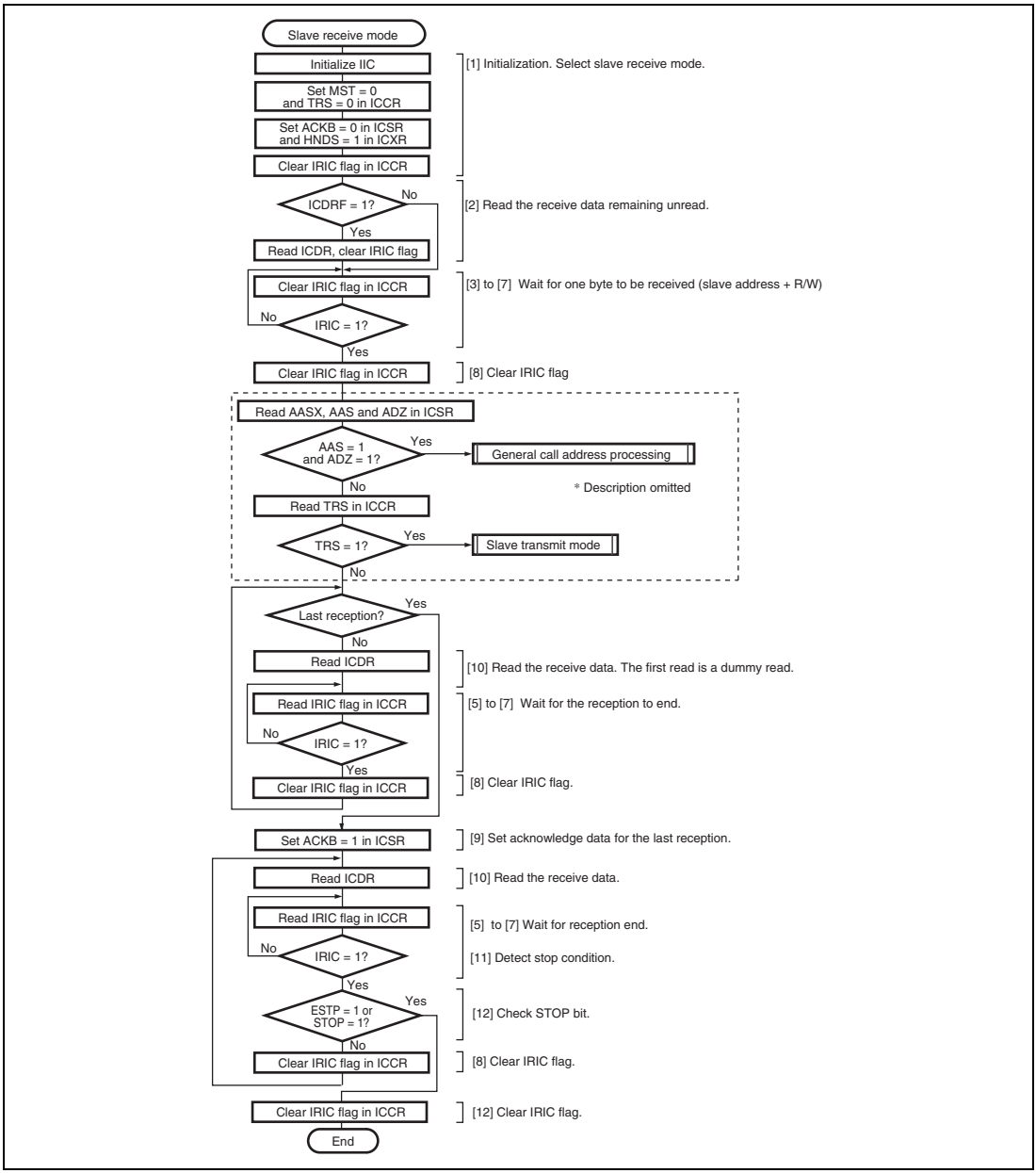


Figure 16.17 Sample Flowchart for Operations in Slave Receive Mode (HNDS = 1)

The reception procedure and operations using the HNDS bit function, by which data reception process is provided in 1-byte unit with SCL being fixed low at every data reception, are described below.

1. Initialize the IIC as described in section 16.4.2, Initialization.

   Clear the MST and TRS bits to 0 to set slave receive mode, and set the HNDS bit to 1 and the ACKB bit to 0. Clear the IRIC flag in ICCR to 0 to see the end of reception.
2. Confirm that the ICDRF flag is 0. If the ICDRF flag is set to 1, read the ICDR and then clear the IRIC flag to 0.
3. When the start condition output by the master device is detected, the BBSY flag in ICCR is set to 1. The master device then outputs the 7-bit slave address and transmit/receive direction (R/W), in synchronization with the transmit clock pulses.
4. When the slave address matches in the first frame following the start condition, the device operates as the slave device specified by the master device. If the 8th data bit (R/W) is 0, the TRS bit remains cleared to 0, and slave receive operation is performed. If the 8th data bit (R/W) is 1, the TRS bit is set to 1, and slave transmit operation is performed. When the slave address does not match, receive operation is halted until the next start condition is detected.
5. At the 9th clock pulse of the receive frame, the slave device returns the data in the ACKB bit as an acknowledge signal.
6. At the rise of the 9th clock pulse, the IRIC flag is set to 1. If the IEIC bit has been set to 1, an interrupt request is sent to the CPU.

   If the AASX bit has been set to 1, IRTR flag is also set to 1.
7. At the rise of the 9th clock pulse, the receive data is transferred from ICDRS to ICDRR, setting the ICDRF flag to 1. The slave device drives SCL low from the fall of the 9th receive clock pulse until data is read from ICDR.
8. Confirm that the STOP bit is cleared to 0, and clear the IRIC flag to 0.
9. If the next frame is the last receive frame, set the ACKB bit to 1.
10. If ICDR is read, the ICDRF flag is cleared to 0, releasing the SCL bus line. This enables the master device to transfer the next data.

Receive operations can be performed continuously by repeating steps [5] to [10].

11. When the stop condition is detected (SDA is changed from low to high when SCL is high), the BBSY flag is cleared to 0 and the STOP bit is set to 1. If the STOPIM bit has been cleared to 0, the IRIC flag is set to 1.
12. Confirm that the STOP bit is set to 1, and clear the IRIC flag to 0.

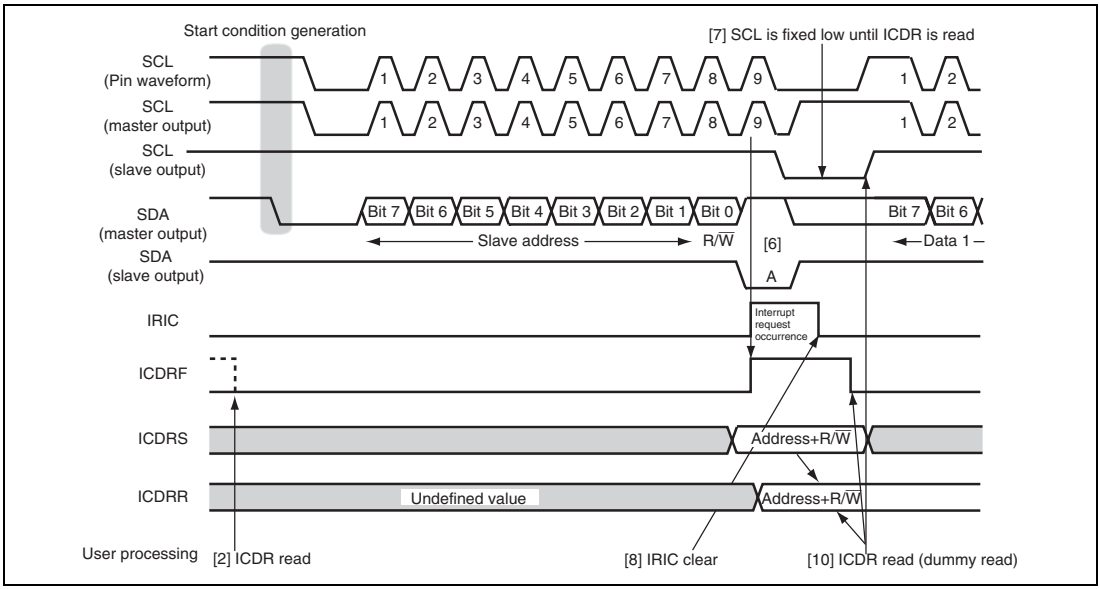


Figure 16.18 Example of Slave Receive Mode Operation Timing (1) (MLS = 0, HNDS= 1)

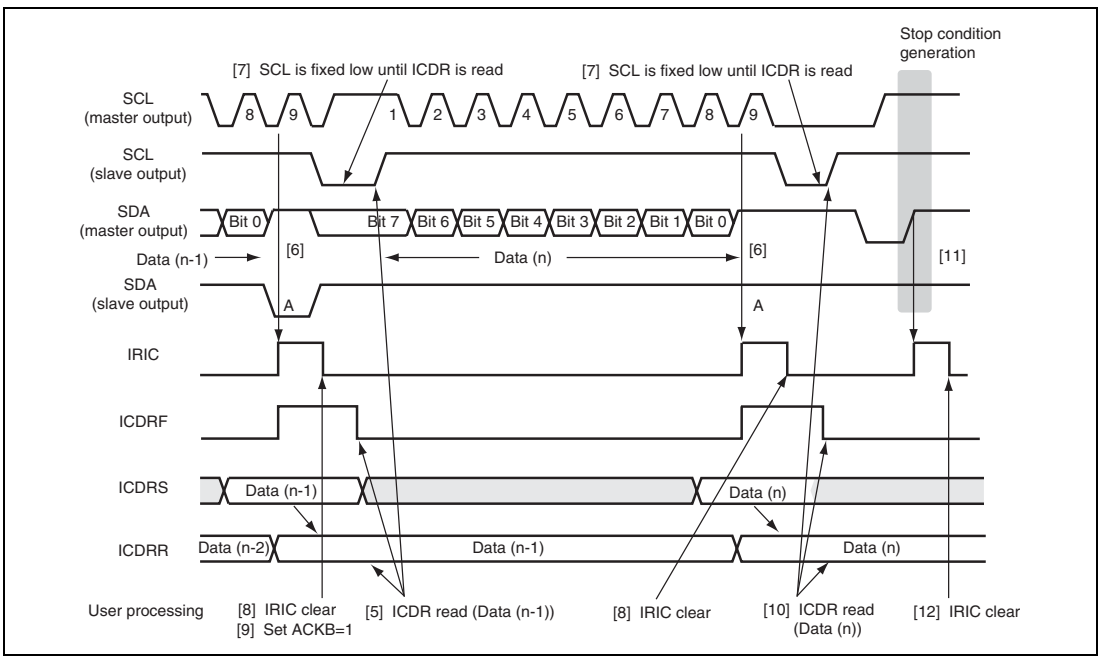


Figure 16.19 Example of Slave Receive Mode Operation Timing (2) (MLS = 0, HNDS= 1)

(2) Continuous Receive Operation

Figure 16.20 shows the sample flowchart for the operations in slave receive mode (HNDS = 0).

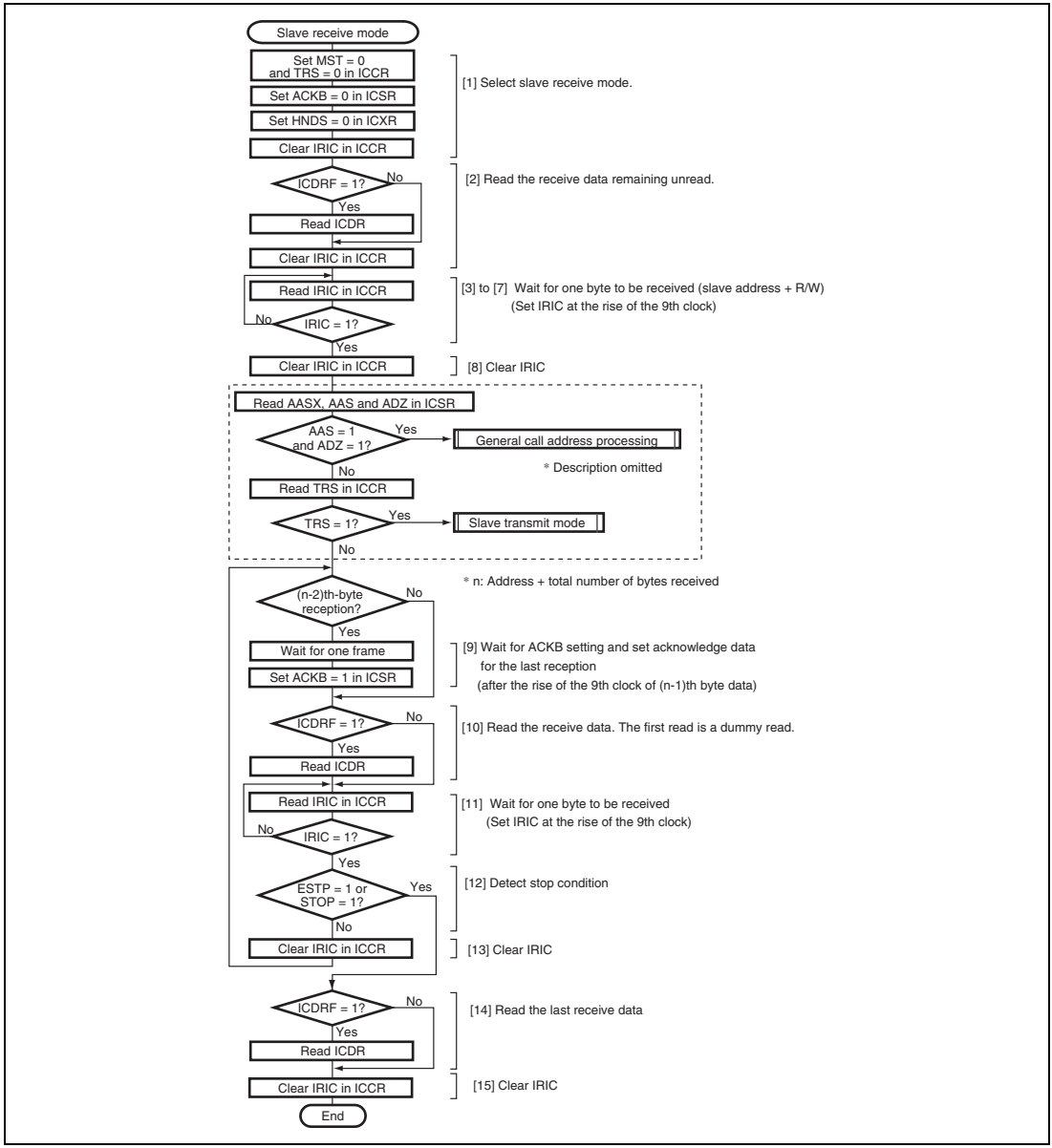


Figure 16.20 Sample Flowchart for Operations in Slave Receive Mode (HNDS = 0)

The reception procedure and operations in slave receive are described below.

1. Initialize the IIC as described in section 16.4.2, Initialization.

   Clear the MST and TRS bits to 0 to set slave receive mode, and set the HNDS and ACKB bits to 0. Clear the IRIC flag in ICCR to 0 to see the end of reception.
2. Confirm that the ICDRF flag is 0. If the ICDRF flag is set to 1, read the ICDR and then clear the IRIC flag to 0.
3. When the start condition output by the master device is detected, the BBSY flag in ICCR is set to 1. The master device then outputs the 7-bit slave address and transmit/receive direction (R/W) in synchronization with the transmit clock pulses.
4. When the slave address matches in the first frame following the start condition, the device operates as the slave device specified by the master device. If the 8th data bit (R/$\overline{W}$) is 0, the TRS bit remains cleared to 0, and slave transmit operation is performed. When the slave address does not match, receive operation is halted until the next start condition is detected.
5. At the 9th clock pulse of the receive frame, the slave device returns the data in the ACKB bit as an acknowledge signal.
6. At the rise of the 9th clock pulse, the IRIC flag is set to 1. If the IEIC bit has been set to 1, an interrupt request is sent to the CPU.

   If the AASX bit has been set to 1, the IRTR flag is also set to 1.
7. At the rise of the 9th clock pulse, the receive data is transferred from ICDRS to ICDRR, setting the ICDRF flag to 1.
8. Confirm that the STOP bit is cleared to 0 and clear the IRIC flag to 0.
9. If the next read data is the third last receive frame, wait for at least one frame time to set the ACKB bit. Set the ACKB bit after the rise of the 9th clock pulse of the second last receive frame.
10. Confirm that the ICDRF flag is set to 1 and read ICDR. This clears the ICDRF flag to 0.
11. At the rise of the 9th clock pulse or when the receive data is transferred from IRDRS to ICDRR due to ICDR read operation, the IRIC and ICDRF flags are set to 1.
12. When the stop condition is detected (SDA is changed from low to high when SCL is high), the BBSY flag is cleared to 0 and the STOP or ESTP flag is set to 1. If the STOPIM bit has been cleared to 0, the IRIC flag is set to 1. In this case, execute step [14] to read the last receive data.
13. Clear the IRIC flag to 0.

Receive operations can be performed continuously by repeating steps [9] to [13].

14. Confirm that the ICDRF flag is set to 1, and read ICDR.
15. Clear the IRIC flag.

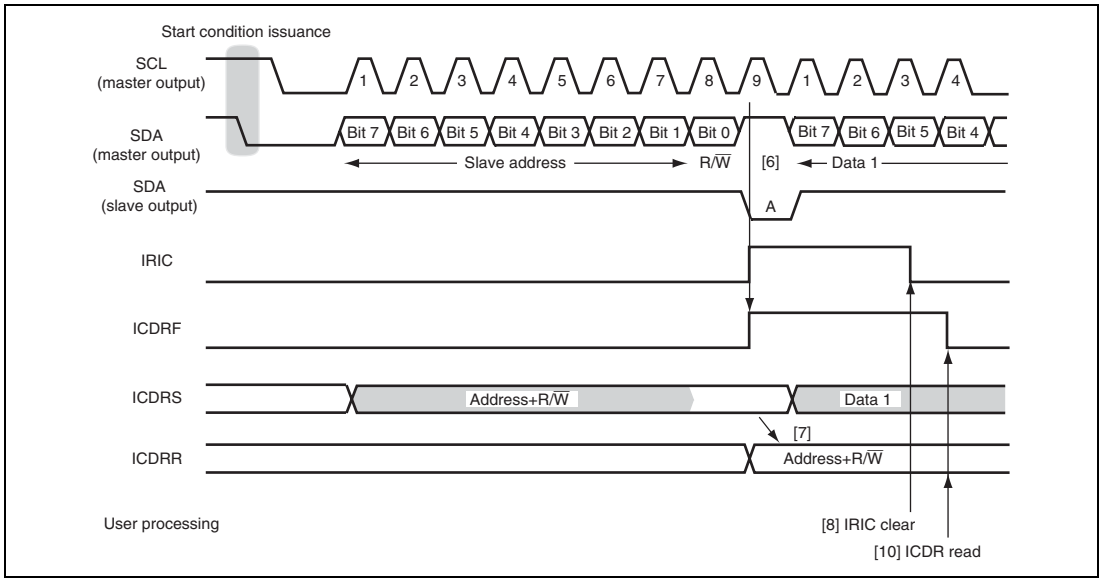


Figure 16.21 Example of Slave Receive Mode Operation Timing (1)
(MLS = ACKB = 0, HNDS = 0)

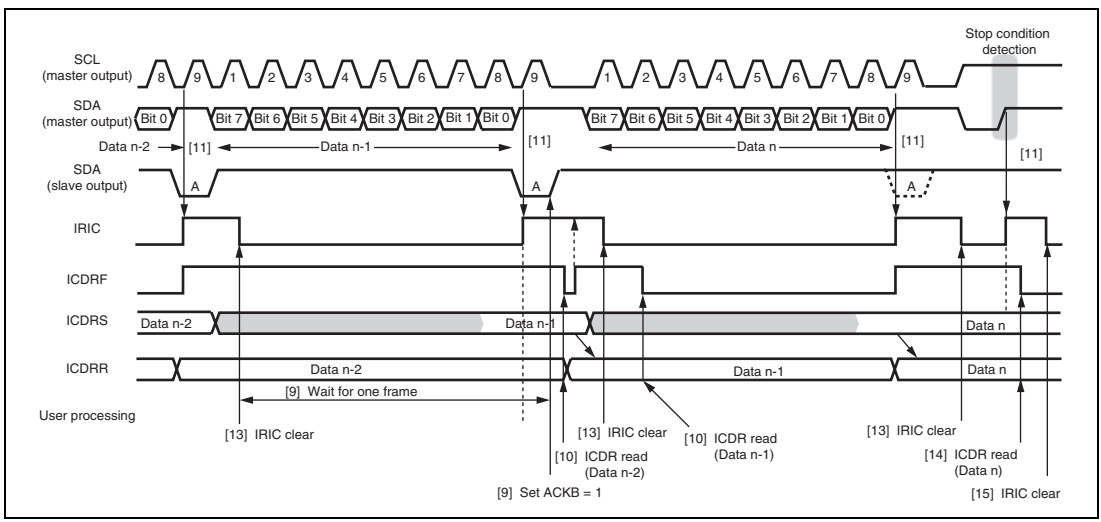


Figure 16.22 Example of Slave Receive Mode Operation Timing (2)
(MLS = ACKB = 0, HNDS = 0)

16.4.6 Slave Transmit Operation

If the slave address matches to the address in the first frame (address reception frame) following the start condition detection when the 8th bit data (R/$\overline{W}$) is 1 (read), the TRS bit in ICCR is automatically set to 1 and the mode changes to slave transmit mode.

Figure 16.23 shows the sample flowchart for the operations in slave transmit mode.

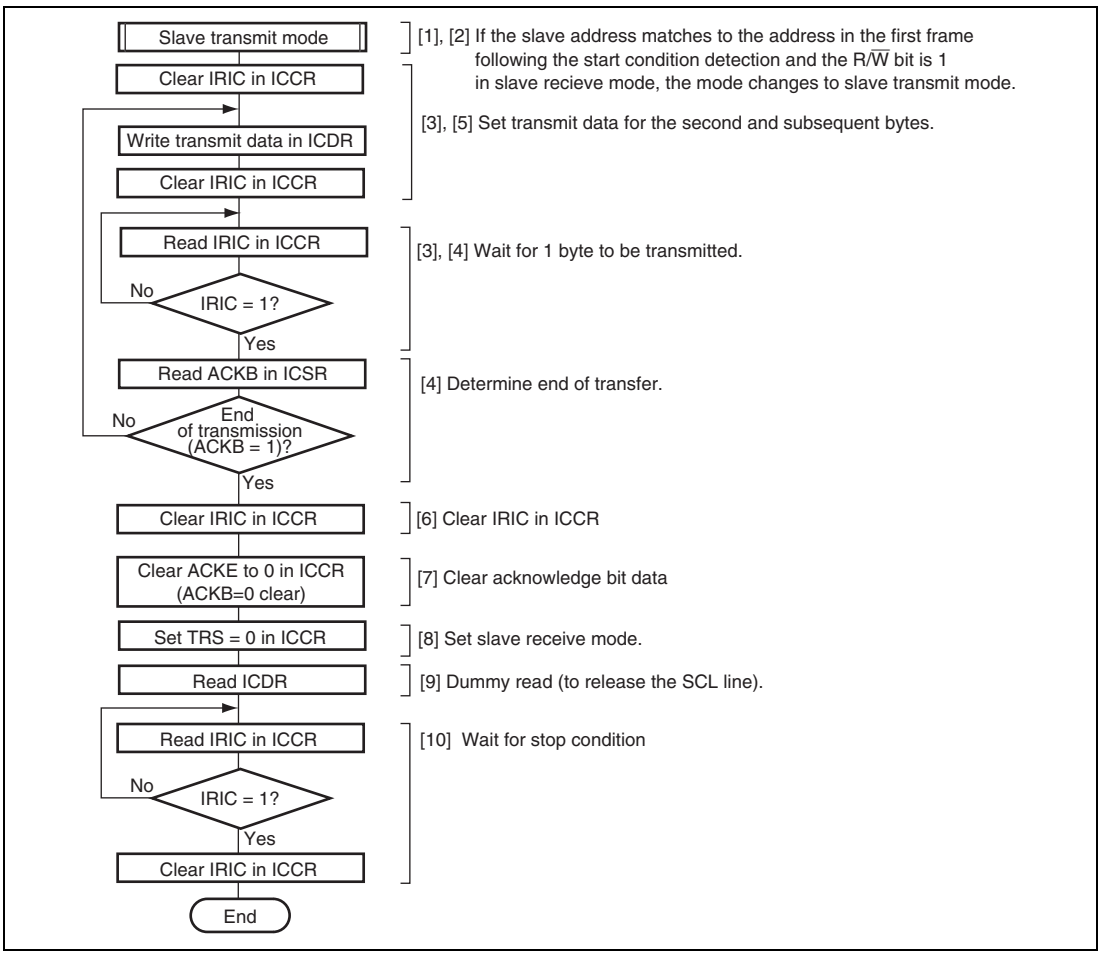


Figure 16.23 Sample Flowchart for Slave Transmit Mode

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. The transmission procedure and operations in slave transmit mode are described below.

1. Initialize slave receive mode and wait for slave address reception.
2. When the slave address matches in the first frame following detection of the start condition, the slave device drives SDA low at the 9th clock pulse and returns an acknowledge signal. If the 8th data bit (R/W) is 1, the TRS bit in ICCR is set to 1, and the mode changes to slave transmit mode automatically. The IRIC flag is set to 1 at the rise of the 9th clock. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. At the same time, the ICDRE flag is set to 1. The slave device drives SCL low from the fall of the transmit 9th clock until ICDR data is written, to disable the master device to output the next transfer clock.
3. After clearing the IRIC flag to 0, write data to ICDR. At this time, the ICDRE flag is cleared to 0. The written data is transferred to ICDRS, and the ICDRE and IRIC flags are set to 1 again. The slave device sequentially sends the data written into ICDRS in accordance with the clock output by the master device.

   The IRIC flag is cleared to 0 to detect the end of transmission. Processing from the ICDR register writing to the IRIC flag clearing should be performed continuously. Prevent any other interrupt processing from being inserted.
4. The master device drives SDA low at the 9th clock pulse, and returns an acknowledge signal. As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used to determine whether the transfer operation was performed successfully. When one frame of data has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit clock pulse. When the ICDRE flag is 0, the data written into ICDR is transferred to ICDRS, transmission starts, and the ICDRE and IRIC flags are set to 1 again. If the ICDRE flag has been set to 1, this slave device drives SCL low from the fall of the 9th transmit clock until data is written to ICDR.
5. To continue transmission, write the next data to be transmitted into ICDR. The ICDRE flag is cleared to 0. The IRIC flag is cleared to 0 to detect the end of transmission. Processing from the ICDR writing to the IRIC flag clearing should be performed continuously. Prevent any other interrupt processing from being inserted.

Transmit operations can be performed continuously by repeating steps [4] and [5].

6. Clear the IRIC flag to 0.
7. To end transmission, clear the ACKE bit in ICCR to 0, to clear the acknowledge bit stored in the ACKB bit to 0.
8. Clear the TRS bit to 0 for the next address reception, to set slave receive mode.
9. Dummy-read ICDR to release SCL on the slave side.

10. When the stop condition is detected, that is, when SDA is changed from low to high when SCL is high, the BBSY flag in ICCR is cleared to 0 and the STOP flag in ICSR is set to 1. When the STOPIM bit in ICXR is 0, the IRIC flag is set to 1. If the IRIC flag has been set, it is cleared to 0.

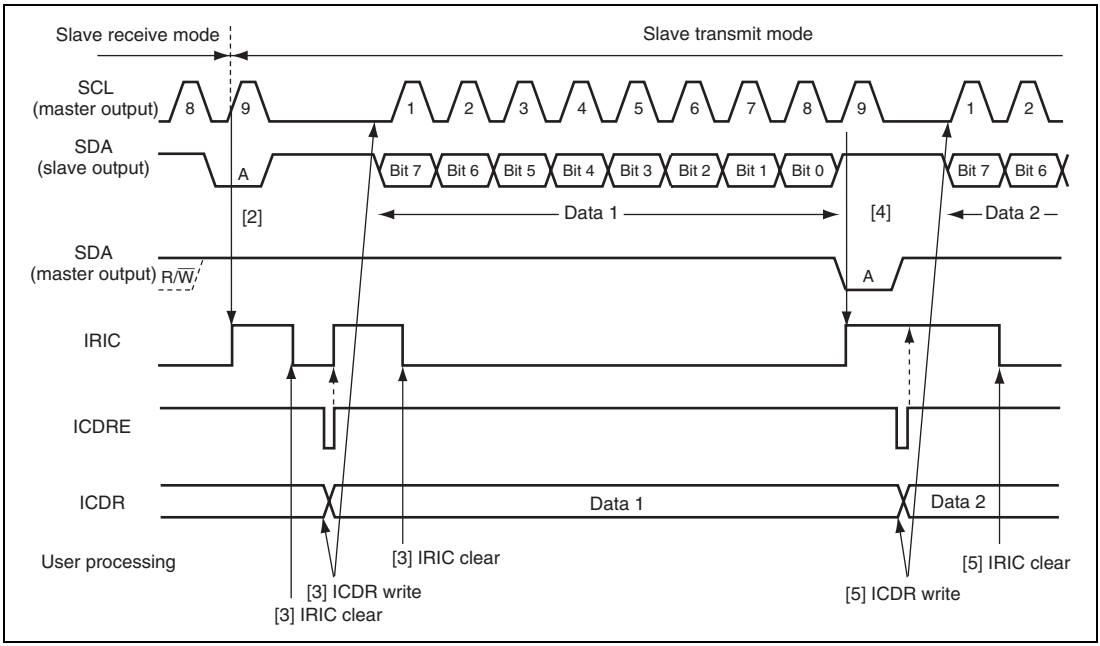


Figure 16.24 Example of Slave Transmit Mode Operation Timing (MLS = 0)

16.4.7 IRIC Setting Timing and SCL Control

The interrupt request flag (IRIC) is set at different times depending on the WAIT bit in ICMR, the FS bit in SAR, and the FSX bit in SARX. If the ICDRE or ICDRF flag is set to 1, SCL is automatically held low after one frame has been transferred in synchronization with the internal clock. Figures 16.25 to 16.27 show the IRIC set timing and SCL control.

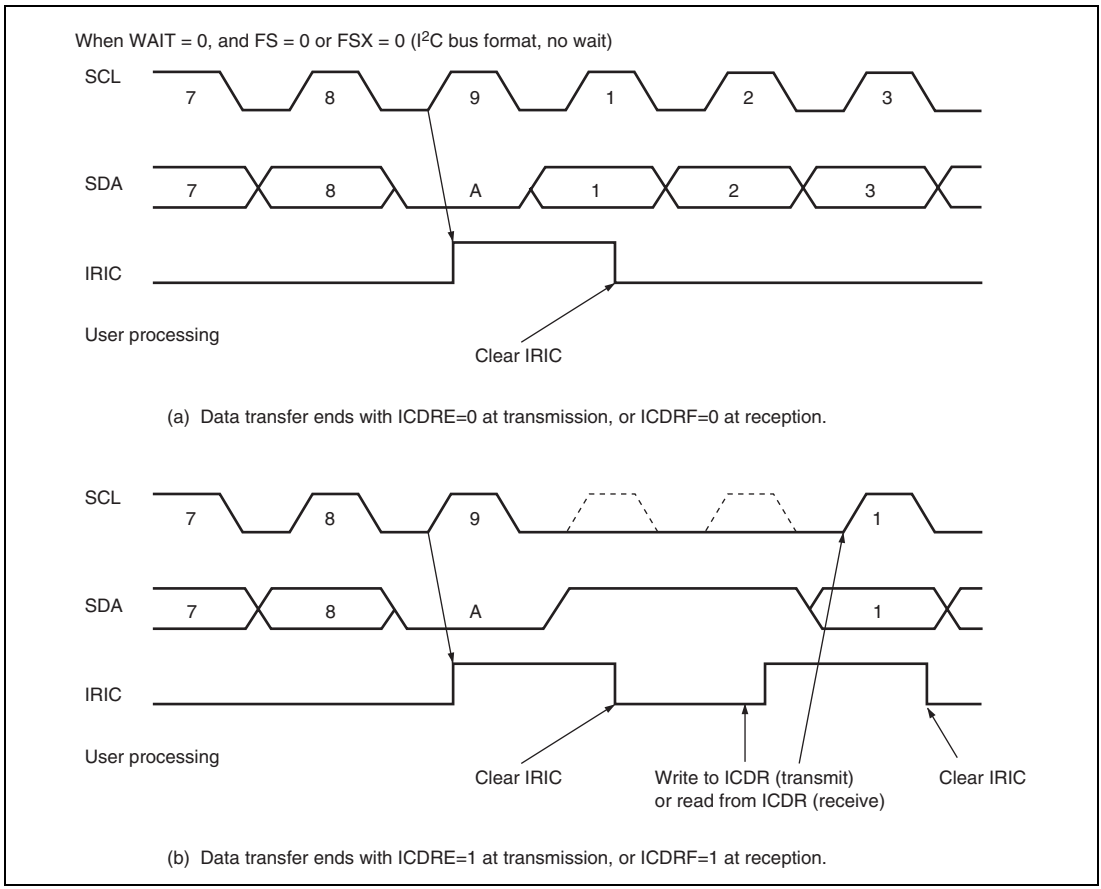
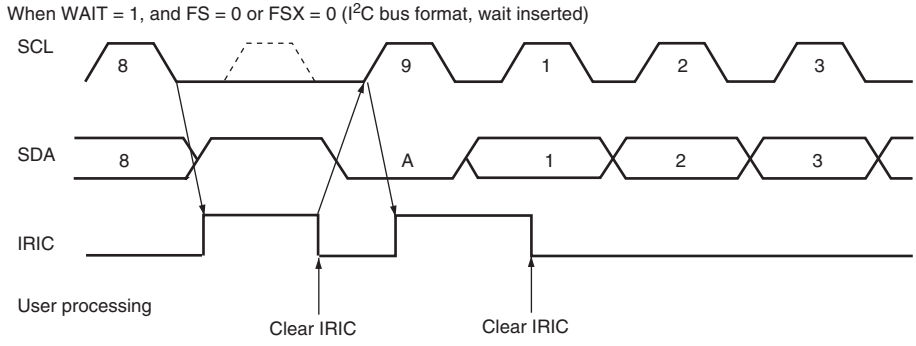
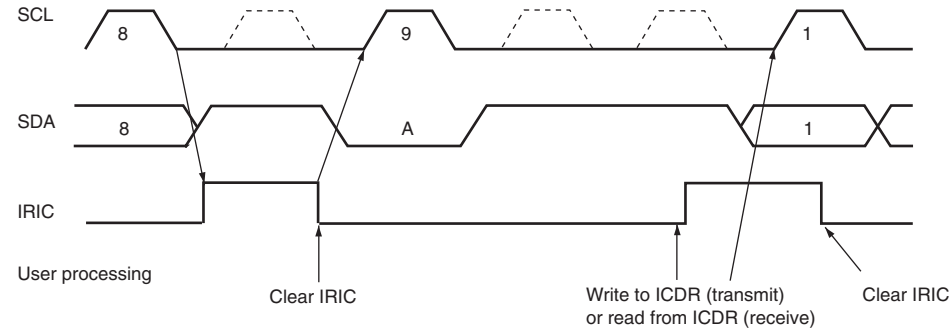


Figure 16.25 IRIC Setting Timing and SCL Control (1)



(a) Data transfer ends with ICDRE=0 at transmission, or ICDRF=0 at reception.



(b) Data transfer ends with ICDRE=1 at transmission, or ICDRF=1 at reception.

Figure 16.26 IRIC Setting Timing and SCL Control (2)

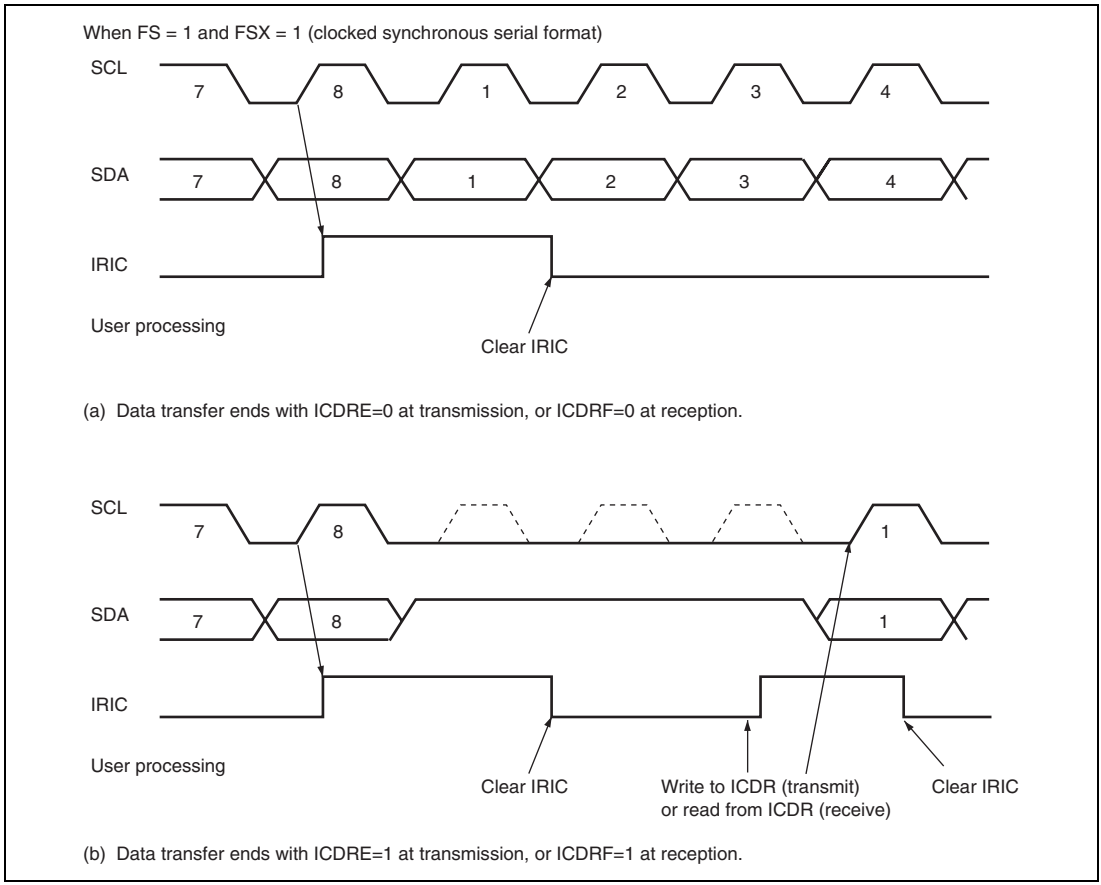


Figure 16.27 IRIC Setting Timing and SCL Control (3)

16.4.8 Operation Using DTC

This LSI provides the DTC to allow continuous data transfer. The DTC is initiated when the IRTR flag is set to 1, which is one of the two interrupt flags (IRTR and IRIC). When the ACKE bit is 0, the ICDRE, IRIC, and IRTR flags are set at the end of data transmission regardless of the acknowledge bit value. If the ACKE bit is 1, the ICDRE, IRIC, and IRTR flags are set when data transmission is completed with the acknowledge bit value of 0, and if the ACKE bit is 1, only the IRIC flag is set when data transmission is completed with the acknowledge bit value of 1.

When initiated, the DTC transfers specified number of bytes, clears the ICDRE, IRIC, and IRTR flags to 0. Therefore, no interrupt is generated during continuous data transfer; however, if data transmission is completed with the acknowledge bit value of 1 when the ACKE bit is 1, the DTC is not initiated, thus allowing an interrupt to be generated if enabled.

The acknowledge bit may indicate specific events such as completion of receive data processing for some receiving devices, and for other receiving devices, the acknowledge bit may be fixed at 1, indicating no specific events.

The I²C bus format provides for selection of the slave device and transfer direction by means of the slave address and the R/$\overline{W}$ bit, confirmation of reception with the acknowledge bit, indication of the last frame, and so on. Therefore, continuous data transfer using the DTC must be carried out in conjunction with CPU processing by means of interrupts.

Table 16.7 shows some examples of processing using the DTC. These examples assume that the number of transfer data bytes is known in slave mode.

Table 16.7 Examples of Operation Using DTC

Item	Master Transmit Mode	Master Receive Mode	Slave Transmit Mode	Slave Receive Mode
Slave address + R/$\overline{W}$ bit transmission/ reception	Transmission by DTC (ICDR write)	Transmission by CPU (ICDR write)	Reception by CPU (ICDR read)	Reception by CPU (ICDR read)
Dummy data read	—	Processing by CPU (ICDR read)	—	—
Actual data transmission/ reception	Transmission by DTC (ICDR write)	Reception by DTC (ICDR read)	Transmission by DTC (ICDR write)	Reception by DTC (ICDR read)
Dummy data (H'FF) write	—	—	Processing by DTC (ICDR write)	—
Last frame processing	Not necessary	Reception by CPU (ICDR read)	Not necessary	Reception by CPU (ICDR read)
Transfer request processing after last frame processing	1st time: Clearing by CPU 2nd time: Stop condition issuance by CPU	Not necessary	Automatic clearing on detection of stop condition during transmission of dummy data (H'FF)	Not necessary
Setting of number of DTC transfer data frames	Transmission: Actual data count + 1 (+1 equivalent to slave address + R/$\overline{W}$ bits)	Reception: Actual data count	Transmission: Actual data count + 1 (+1 equivalent to dummy data (H'FF))	Reception: Actual data count

16.4.9 Noise Canceller

The logic levels at the SCL and SDA pins are routed through noise cancelers before being latched internally. Figure 16.28 shows a block diagram of the noise canceler.

The noise canceler consists of two cascaded latches and a match detector. The SCL (or SDA) pin 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.

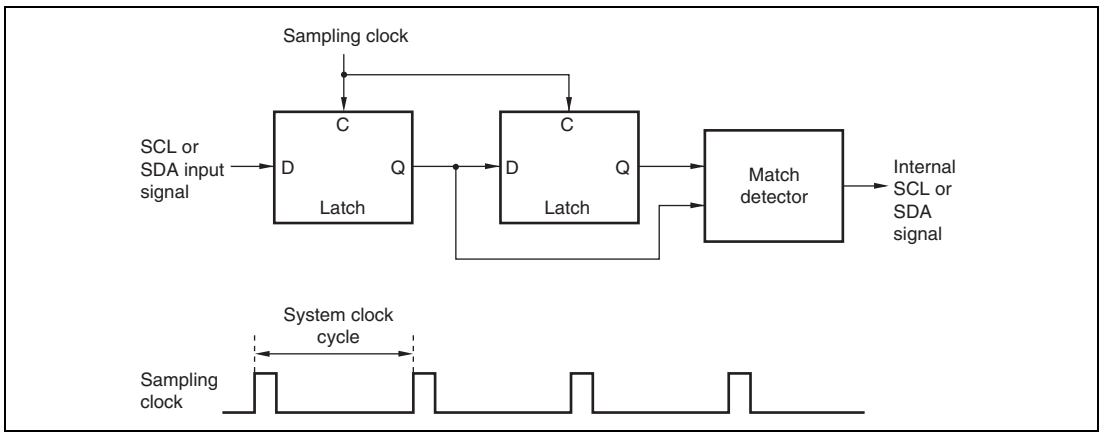


Figure 16.28 Block Diagram of Noise Canceler

16.4.10 Initialization of Internal State

The IIC has a function for forcible initialization of its internal state if a deadlock occurs during communication.

Initialization is executed in accordance with the setting of bits CLR3 to CLR0 in DDCSWR or clearing ICE bit. For details on the setting of bits CLR3 to CLR0, see section 16.3.7, DDC Switch Register (DDCSWR).

(1) Scope of Initialization

The initialization executed by this function covers the following items:

- ICDRE and ICDRF internal flags
- Transmit/receive sequencer and internal operating clock counter
- Internal latches for retaining the output state of the SCL and SDA pins (wait, clock, data output, etc.)

The following items are not initialized:

- Actual register values (ICDR, SAR, SARX, ICMR, ICCR, ICSR, and ICXR (except for the ICDRE and ICDRF flags))
- Internal latches used to retain register read information for setting/clearing flags in ICMR, ICCR, and ICSR
- The value of the ICMR bit counter (BC2 to BC0)
- Generated interrupt sources (interrupt sources transferred to the interrupt controller)

(2) Notes on Initialization

- Interrupt flags and interrupt sources are not cleared, and so flag clearing measures must be taken as necessary.
- Basically, other register flags are not cleared either, and so flag clearing measures must be taken as necessary.
- When initialization is executed by DDCSWR, the write data for bits CLR3 to CLR0 is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR.
- Similarly, when clearing is required again, all the bits must be written to simultaneously in accordance with the setting.
- If a flag clearing setting is made during transmission/reception, the IIC module will stop transmitting/receiving at that point and the SCL and SDA pins will be released. When transmission/reception is started again, register initialization, etc., must be carried out as necessary to enable correct communication as a system.

The value of the BBSY bit cannot be modified directly by this module clear function, but since the stop condition pin waveform is generated according to the state and release timing of the SCL and SDA pins, the BBSY bit may be cleared as a result. Similarly, state switching of other bits and flags may also have an effect.

To prevent problems caused by these factors, the following procedure should be used when initializing the IIC state.

1. Execute initialization of the internal state according to the setting of bits CLR3 to CLR0 or ICE bit clearing.
2. Execute a stop condition issuance instruction (write 0 to BBSY and SCP) to clear the BBSY bit to 0, and wait for two transfer rate clock cycles.
3. Re-execute initialization of the internal state according to the setting of bits CLR3 to CLR0 or ICE bit clearing.
4. Initialize (re-set) the IIC registers.

16.5 Interrupt Sources

The IIC has interrupt sources, IICI0 and IICI1. Table 16.8 shows the interrupt sources and priority. Individual interrupt sources can be enabled or disabled using the enable bit in ICCR, and are sent to the interrupt controller independently.

An IICI interrupt can activate the DTC to allow data transfer.

Table 16.8 IIC Interrupt Sources

Channel	Name	Enable Bit	Interrupt Source	Interrupt Flag	DTC Activation	Priority
0	IICI0	IEIC	I²C bus interface interrupt request	IRIC	Possible	High ↑
1	IICI1	IEIC	I²C bus interface interrupt request	IRIC	Possible	↓ Low

16.6 Usage Notes

1. In master mode, if an instruction to generate a start condition is issued and then an instruction to generate a stop condition is issued before the start condition is output to the I²C bus, neither condition will be output correctly. To output the stop condition followed by the start condition*, after issuing the instruction that generates the start condition, read DR in each I²C bus output pin, and check that SCL and SDA are both low. The pin states can be monitored by reading DR even if the ICE bit is set to 1. Then issue the instruction that generates the stop condition. Note that SCL may not yet have gone low when BBSY is cleared to 0.

Note: * An illegal procedure in the I²C bus specification.

2. Either of the following two conditions will start the next transfer. Pay attention to these conditions when accessing to ICDR.
   - Write to ICDR when ICE = 1 and TRS = 1 (including automatic transfer from ICDRT to ICDRS)
   - Read from ICDR when ICE = 1 and TRS = 0 (including automatic transfer from ICDRS to ICDRR)
3. Table 16.9 shows the timing of SCL and SDA outputs in synchronization with the internal clock. Timings on the bus are determined by the rise and fall times of signals affected by the bus load capacitance, series resistance, and parallel resistance.

Table 16.9 I²C Bus Timing (SCL and SDA Outputs)

Item	Symbol	Output Timing	Unit	Notes
SCL output cycle time	t_{SCLO}	$28t_{cyc}$ to $256t_{cyc}$	ns	See figure 24.26.
SCL output high pulse width	t_{SCLHO}	$0.5t_{SCLO}$	ns	
SCL output low pulse width	t_{SCLLO}	$0.5t_{SCLO}$	ns	
SDA output bus free time	t_{BUFO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Start condition output hold time	t_{STAHO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Retransmission start condition output setup time	t_{STASO}	$1t_{SCLO}$	ns	
Stop condition output setup time	t_{STOSO}	$0.5t_{SCLO} + 2t_{cyc}$	ns	
Data output setup time (master)	t_{SDASO}	$1t_{SCLLO} – 3t_{cyc}$	ns	
Data output setup time (slave)		$1t_{SCLL} – (6t_{cyc}$ or $12t_{cyc}$ *)		
Data output hold time	t_{SDAHO}	$3t_{cyc}$	ns	

Note: * $6t_{cyc}$ when IICX is 0, $12t_{cyc}$ when 1.

4. SCL and SDA inputs are sampled in synchronization with the internal clock. The AC timing therefore depends on the system clock cycle tcyc, as shown in section 24, Electrical Characteristics. Note that the I²C bus interface AC timing specifications will not be met with a system clock frequency of less than 5 MHz.
5. The I²C bus interface specification for the SCL rise time tsr is 1000 ns or less (300 ns for high-speed mode). In master mode, the I²C bus interface monitors the SCL line and synchronizes one bit at a time during communication. If tsr (the time for SCL to go from low to VIH) exceeds the time determined by the input clock of the I²C bus interface, the high period of SCL is extended. The SCL rise time is determined by the pull-up resistance and load capacitance of the SCL line. To insure proper operation at the set transfer rate, adjust the pull-up resistance and load capacitance so that the SCL rise time does not exceed the values given in table 16.10.

Table 16.10 Permissible SCL Rise Time (t_{sr}) Values

			Time Indication [ns]				
IICX	t_{cyc} Indication		I²C Bus Specification (Max.)	φ = 8 MHz	φ = 10 MHz	φ = 16 MHz	φ = 20 MHz
0	7.5 t_{cyc}	Standard mode	1000	937	750	468	375
		High-speed mode	300	←	←	←	←
1	17.5 t_{cyc}	Standard mode	1000	←	←	←	875
		High-speed mode	300	←	←	←	←

6. The I²C bus interface specifications for the SCL and SDA rise and fall times are under 1000 ns and 300 ns. The I²C bus interface SCL and SDA output timing is prescribed by t_{cyc}, as shown in table 16.9. However, because of the rise and fall times, the I²C bus interface specifications may not be satisfied at the maximum transfer rate. Table 16.11 shows output timing calculations for different operating frequencies, including the worst-case influence of rise and fall times.

   t_{BUFO} fails to meet the I²C bus interface specifications at any frequency. The solution is either (a) to provide coding to secure the necessary interval (approximately 1 µs) between issuance of a stop condition and issuance of a start condition, or (b) to select devices whose input timing permits this output timing for use as slave devices connected to the I²C bus.

   t_{SCLLO} in high-speed mode and t_{STASO} in standard mode fail to satisfy the I²C bus interface specifications for worst-case calculations of t_{Sr}/t_{Sf}. Possible solutions that should be investigated include (a) adjusting the rise and fall times by means of a pull-up resistor and capacitive load, (b) reducing the transfer rate to meet the specifications, or (c) selecting devices whose input timing permits this output timing for use as slave devices connected to the I²C bus.

Table 16.11 I²C Bus Timing (with Maximum Influence of t_{Sr}/t_{Sf})

Item	t_{cyc} Indication		Time Indication (at Maximum Transfer Rate) [ns]					
			t_{Sr}/t_{Sf} Influence (Max.)	I²C Bus Specification (Min.)	ϕ = 8 MHz	ϕ = 10 MHz	ϕ = 16 MHz	ϕ = 20 MHz
t_{SCLHO}	0.5 t_{SCLO} ($-t_{Sr}$)	Standard mode	–1000	4000	4000	4000	4000	4000
		High-speed mode	–300	600	950	950	950	950
t_{SCLLO}	0.5 t_{SCLO} ($-t_{Sf}$)	Standard mode	–250	4700	4750	4750	4750	4750
		High-speed mode	–250	1300	1000*1	1000*1	1000*1	1000*1
t_{BUFO}	0.5 t_{SCLO}–1 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	4700	3875*1	3900*1	3938*1	3950*1
		High-speed mode	–300	1300	825*1	850*1	888*1	900*1
t_{STAHO}	0.5 t_{SCLO}–1 t_{cyc} ($-t_{Sf}$)	Standard mode	–250	4000	4625	4650	4688	4700
		High-speed mode	–250	600	875	900	938	950
t_{STASO}	1 t_{SCLO} ($-t_{Sr}$)	Standard mode	–1000	4700	9000	9000	9000	9000
		High-speed mode	–300	600	2200	2200	2200	2200
t_{STOSO}	0.5 t_{SCLO} + 2 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	4000	4250	4200	4125	4100
		High-speed mode	–300	600	1200	1150	1075	1050
t_{SDASO} (master)	1 t_{SCLLO}*3 –3 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	250	3325	3400	3513	3550
		High-speed mode	–300	100	625	700	813	850
t_{SDASO} (slave)	1 t_{SCLL}*3 –12 t_{cyc}*2 ($-t_{Sr}$)	Standard mode	–1000	250	2200	2500	2950	3100
		High-speed mode	–300	100	–500*1	–200*1	250	400

Item	t_{cyc} Indication		Time Indication (at Maximum Transfer Rate) [ns]: t_{Sr}/t_{Sf} Influence (Max.)	I²C Bus Specification (Min.)	ϕ = 8 MHz	ϕ = 10 MHz	ϕ = 16 MHz	ϕ = 20 MHz
t_{SDAHO}	3 t_{cyc}	Standard mode	0	0	375	300	188	150
		High-speed mode	0	0	375	300	188	150

Notes: 1. Does not meet the I²C bus interface specification. Remedial action such as the following is necessary: (a) secure a start/stop condition issuance interval; (b) adjust the rise and fall times by means of a pull-up resistor and capacitive load; (c) reduce the transfer rate; (d) select slave devices whose input timing permits this output timing.

The values in the above table will vary depending on the settings of the IICX bit and bits CKS0 to CKS2. Depending on the frequency it may not be possible to achieve the maximum transfer rate; therefore, whether or not the I²C bus interface specifications are met must be determined in accordance with the actual setting conditions.

2. Value when the IICX bit is set to 1. When the IICX bit is cleared to 0, the value is ($t_{SCLL} - 6\ t_{cyc}$).
3. Calculated using the I²C bus specification values (standard mode: 4700 ns min.; high-speed mode: 1300 ns min.).

7. Notes on ICDR read at end of master reception

To halt reception at the end of a receive operation in master receive mode, set the TRS bit to 1 and write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition. After this, receive data can be read by means of an ICDR read, but if data remains in the buffer the ICDRS receive data will not be transferred to ICDR (ICDRR), and so it will not be possible to read the second byte of data.

If it is necessary to read the second byte of data, issue the stop condition in master receive mode (i.e. with the TRS bit cleared to 0). When reading the receive data, first confirm that the BBSY bit in ICCR is cleared to 0, the stop condition has been generated, and the bus has been released, then read ICDR with TRS cleared to 0.

Note that if the receive data (ICDR data) is read in the interval between execution of the instruction for issuance of the stop condition (writing of 0 to BBSY and SCP in ICCR) and the actual generation of the stop condition, the clock may not be output correctly in subsequent master transmission.

Clearing of the MST bit after completion of master transmission/reception, or other modifications of IIC control bits to change the transmit/receive operating mode or settings, must be carried out during interval (a) in figure 16.29 (after confirming that the BBSY bit in ICCR has been cleared to 0).

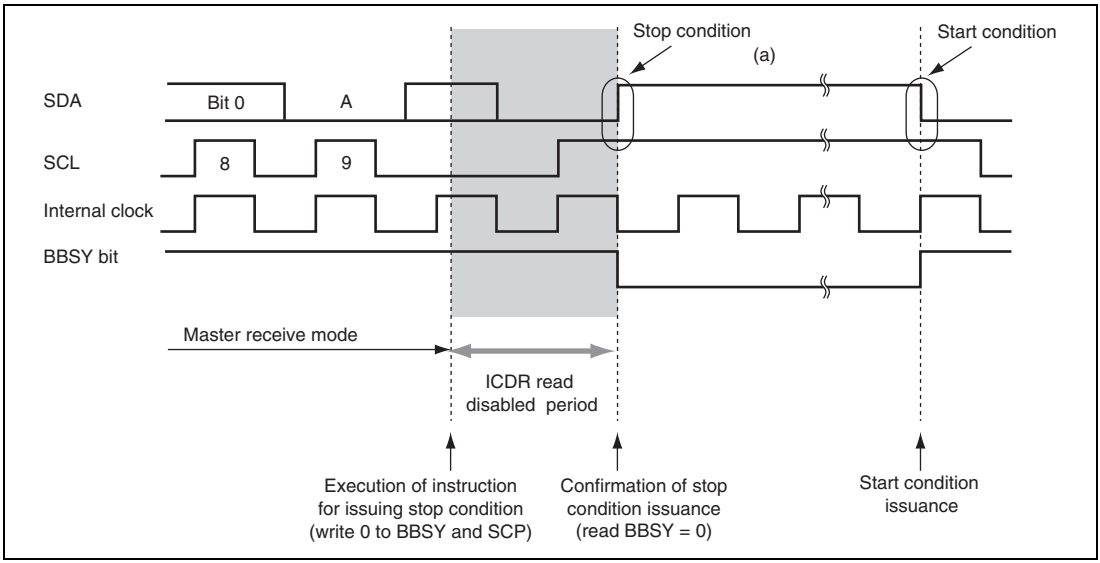


Figure 16.29 Notes on Reading Master Receive Data

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

8. Notes on start condition issuance for retransmission

   Figure 16.30 shows the timing of start condition issuance for retransmission, and the timing for subsequently writing data to ICDR, together with the corresponding flowchart. Write the transmit data to ICDR after the start condition for retransmission is issued and then the start condition is actually generated.

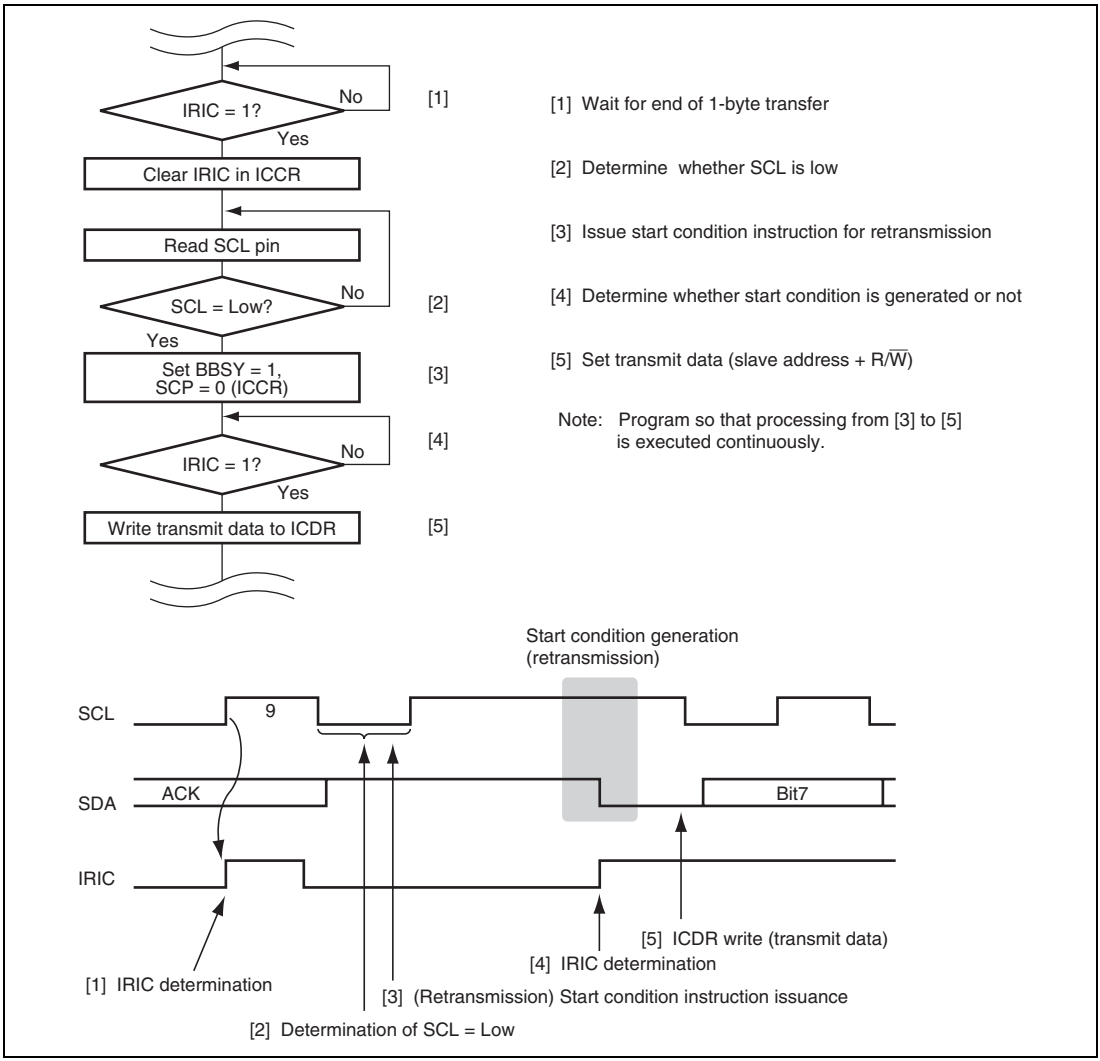


Figure 16.30 Flowchart for Start Condition Issuance Instruction for Retransmission and Timing

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

9. Note on when I²C bus interface stop condition instruction is issued

   In cases where the rise time of the 9th clock of SCL exceeds the stipulated value because of a large bus load capacity or where a slave device in which a wait can be inserted by driving the SCL pin low is used, the stop condition instruction should be issued after reading SCL after the rise of the 9th clock pulse and determining that it is low.

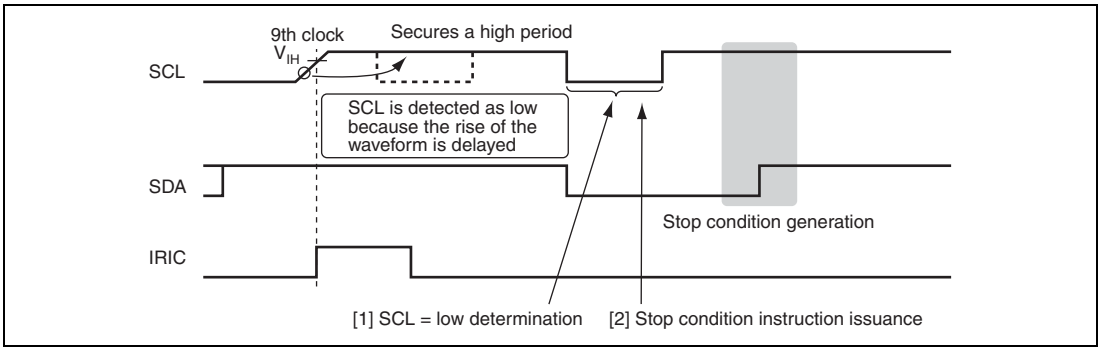


Figure 16.31 Stop Condition Issuance Timing

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

10. Note on IRIC flag clear when the wait function is used

    If the rise time of SCL exceeds the stipulated value or a slave device in which a wait can be inserted by driving the SCL pin low is used when the wait function is used in I²C bust interface master mode, the IRIC flag should be cleared after determining that the SCL is low, as described below.

    If the IRIC flag is cleared to 0 when WAIT = 1 while the SCL is extending the high level time, the SDA level may change before the SCL goes low, which may generate a start or stop condition erroneously.

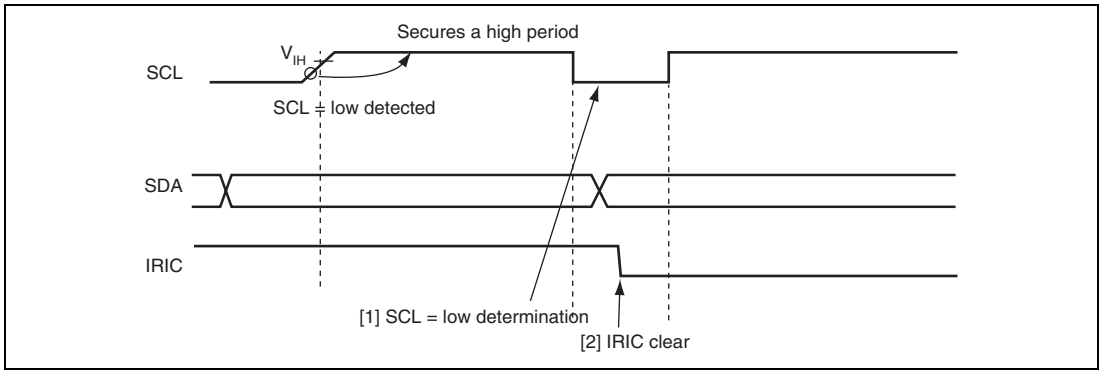


Figure 16.32 IRIC Flag Clearing Timing When WAIT = 1

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

11. Note on ICDR read and ICCR access in slave transmit mode

In I²C bus interface slave transmit mode, do not read ICDR or do not read/write from/to ICCR during the time shaded in figure 16.33. However, such read and write operations cause no problem in interrupt handling processing that is generated in synchronization with the rising edge of the 9th clock pulse because the shaded time has passed before making the transition to interrupt handling.

To handle interrupts securely, be sure to keep either of the following conditions.

— Read ICDR data that has been received so far or read/write from/to ICCR before starting the receive operation of the next slave address.

— Monitor the BC2 to BC0 bit counter in ICMR; when the count is B'000 (8th or 9th clock pulse), wait for at least two transfer clock times in order to read ICDR or read/write from/to ICCR during the time other than the shaded time.

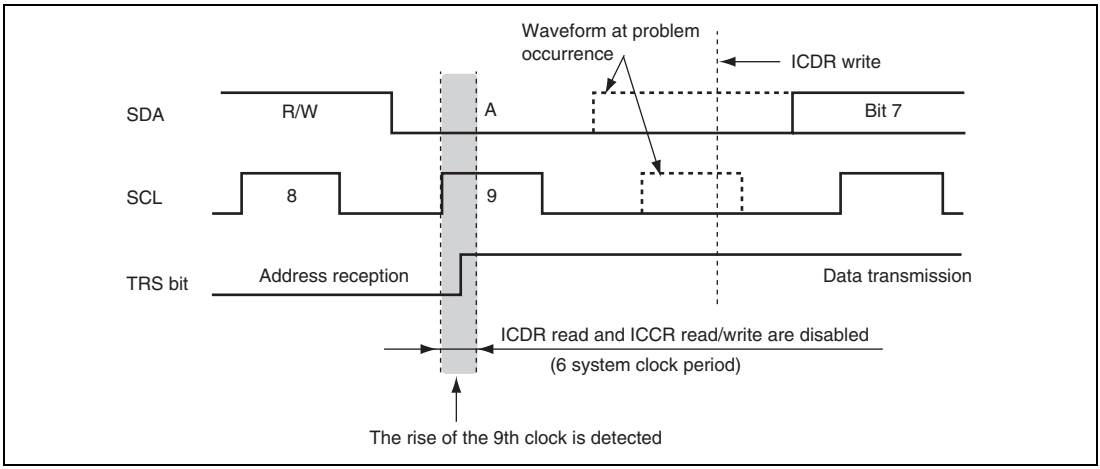


Figure 16.33 ICDR Read and ICCR Access Timing in Slave Transmit Mode

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

12. Note on TRS bit setting in slave mode

In I²C bus interface slave mode, if the TRS bit value in ICCR is set after detecting the rising edge of the 9th clock pulse or the stop condition before detecting the next rising edge on the SCL pin (the time indicated as (a) in figure 16.34), the bit value becomes valid immediately when it is set. However, if the TRS bit is set during the other time (the time indicated as (b) in figure 16.34), the bit value is suspended and remains invalid until the rising edge of the 9th clock pulse or the stop condition is detected. Therefore, when the address is received after the restart condition is input without the stop condition, the effective TRS bit value remains 1 (transmit mode) internally and thus the acknowledge bit is not transmitted after the address has been received at the 9th clock pulse.

To receive the address in slave mode, clear the TRS bit to 0 during the time indicated as (a) in figure 16.34. To release the SCL low level that is held by means of the wait function in slave mode, clear the TRS bit to and then dummy-read ICDR.

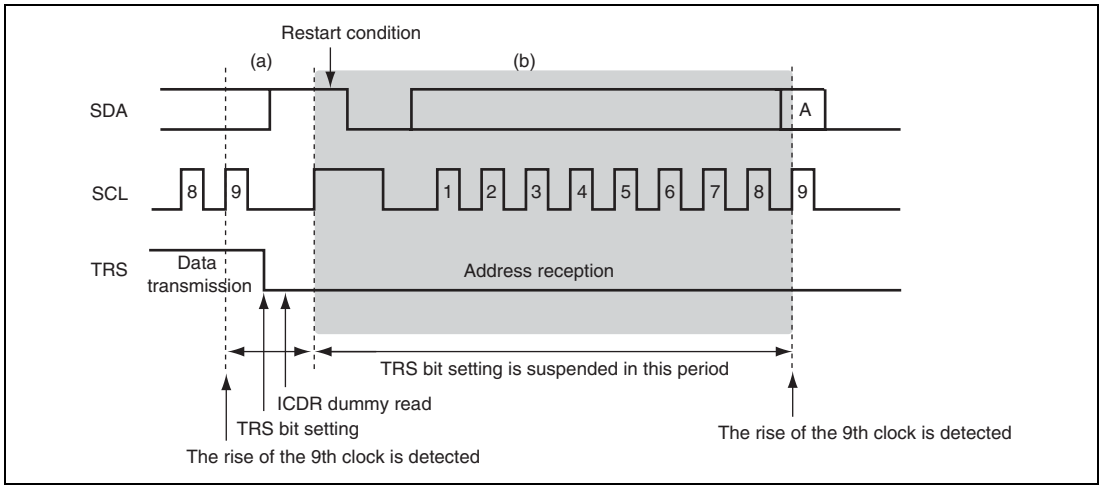


Figure 16.34 TRS Bit Set Timing in Slave Mode

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

13. Note on ICDR read in transmit mode and ICDR write in receive mode

    If ICDR is read in transmit mode (TRS = 1) or ICDR is written to in receive mode (TRS = 0), the SCL pin may not be held low in some cases after transmit/receive operation has been completed, thus inconveniently allowing clock pulses to be output on the SCL bus line before ICDR is accessed correctly. To access ICDR correctly, read ICDR after setting receive mode or write to ICDR after setting transmit mode.

14. Note on ACKE and TRS bits in slave mode

    In the I²C bus interface, if 1 is received as the acknowledge bit value (ACKB = 1) in transmit mode (TRS = 1) and then the address is received in slave mode without performing appropriate processing, interrupt handling may start at the rising edge of the 9th clock pulse even when the address does not match. Similarly, if the start condition or address is transmitted from the master device in slave transmit mode (TRS = 1), the IRIC flag may be set after the ICDRE flag is set and 1 received as the acknowledge bit value (ACKB = 1), thus causing an interrupt source even when the address does not match.

    To use the I²C bus interface module in slave mode, be sure to follow the procedures below.

    A. When having received 1 as the acknowledge bit value for the last transmit data at the end of a series of transmit operation, clear the ACKE bit in ICCR once to initialize the ACKB bit to 0.

B. Set receive mode (TRS = 0) before the next start condition is input in slave mode. Complete transmit operation by the procedure shown in figure 16.23, in order to switch from slave transmit mode to slave receive mode.

15. Note on Arbitration Lost in Master Mode

The I²C bus interface recognizes the data in transmit/receive frame as an address when arbitration is lost in master mode and a transition to slave receive mode is automatically carried out.

When arbitration is lost not in the first frame but in the second frame or subsequent frame, transmit/receive data that is not an address is compared with the value set in the SAR or SARX register as an address. If the receive data matches with the address in the SAR or SARX register, the I²C bus interface erroneously recognizes that the address call has occurred. (See figure 16.35.)

In multi-master mode, a bus conflict could happen. When the I²C bus interface is operated in master mode, check the state of the AL bit in the ICSR register every time after one frame of data has been transmitted or received.

When arbitration is lost during transmitting the second frame or subsequent frame, take avoidance measures.

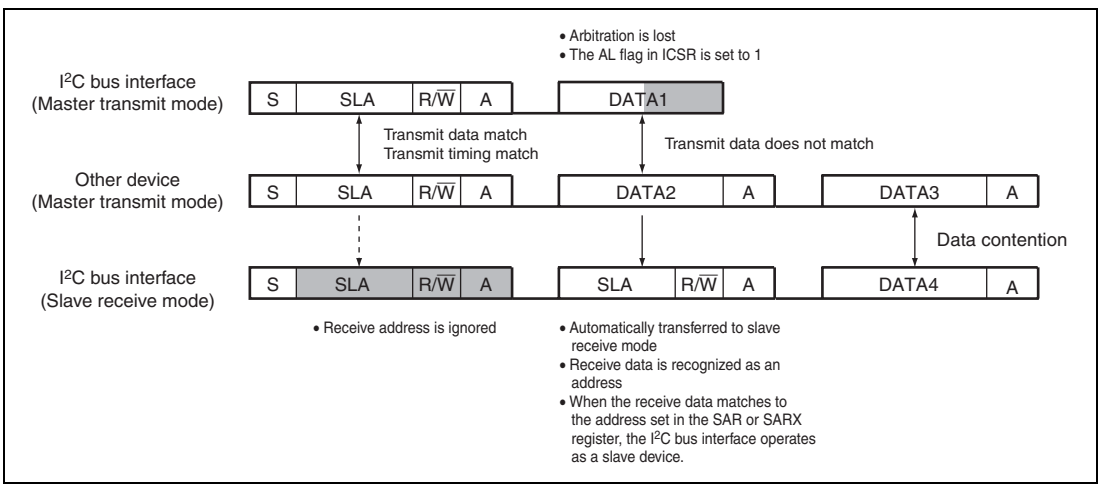


Figure 16.35 Diagram of Erroneous Operation when Arbitration is Lost

16.6.1 Note on Wait Function in Master Mode

While the WAIT bit in ICMR is set to 1 and WAIT in master mode, if the interrupt flag of the IRIC bit is cleared from 1 to 0 between the falling edge of the 7th clock and the falling edge of the 8th clock, the clock pulse of the 9th clock may be output continuously due to the failure to insert a wait after the falling edge of the 8th clock.

When the wait function is used in master mode, clear the IRIC flag after the IRIC flag is set to 1 on the falling edge of the 9th clock and before the rising edge of the 7th clock (the counter value of BC2 to BC0 should be 2 or greater).

If the clearing of the IRIC flag is delayed due to the interrupt or other processes and the value of the RC counter is changed to 1 or 0, confirm that the SCL pins are in the L state after the counter values of BC2 to BC0 are cleared to 0, and then clear the IRIC flag (see figure 16.36).

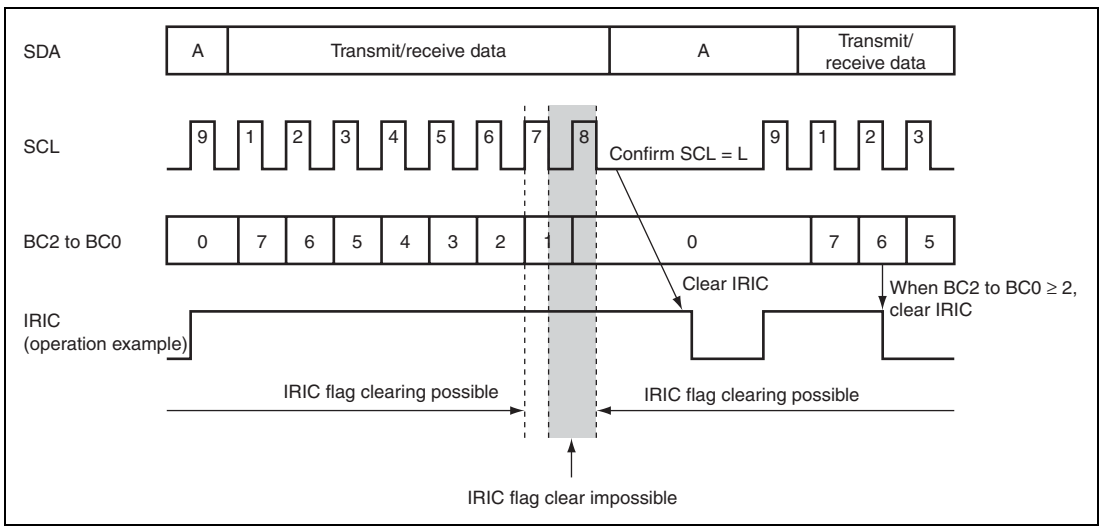


Figure 16.36 IRIC Flag Clear Timing in Wait Operation

Note: This limitation on use can be cleared by setting the FNC1 and FNC0 bits in ICXR to B'11.

16.6.2 Module Stop Mode Setting

The IIC operation can be enabled or disabled using the module stop control register. The initial setting is for the IIC operation to be halted. Register access is enabled by canceling module stop mode. For details, refer to section 22, Power-Down Modes.

Section 17 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.

17.1 Features

- 10-bit resolution
- Input channels: Eight analog input channels
- Analog conversion voltage range can be specified using the analog power supply voltage pin (AVCC) as an analog reference voltage.
- Conversion time: 13.4 μs per channel (at 20-MHz operation)
- Two kinds of operating modes
  Single mode: Single-channel A/D conversion
  Scan mode: Continuous A/D conversion on one to four channels
- Four data registers
  Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of A/D conversion start
  Software
  Timer (8-bit timer) conversion start trigger
  External trigger signal ($\overline{\text{ADTRG}}$)
- Interrupt source
  A/D conversion end interrupt (ADI) request can be generated

A block diagram of the A/D converter is shown in figure 17.1.

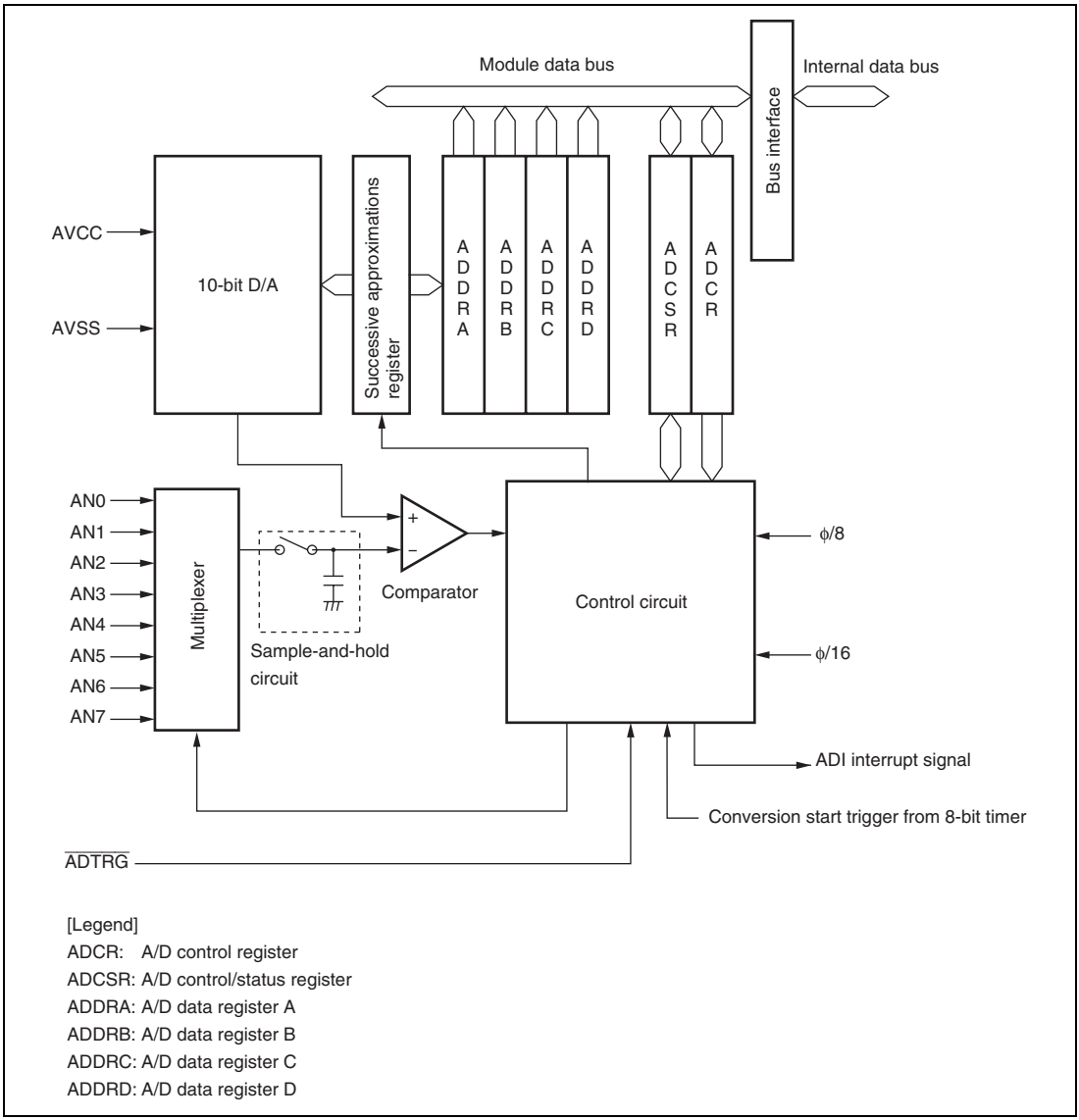


Figure 17.1 Block Diagram of A/D Converter

17.2 Input/Output Pins

Table 17.1 summarizes the pins used by the A/D converter. The eight analog input pins are divided into two groups consisting of four channels. Analog input pins 0 to 3 (AN0 to AN3) comprising group 0 and analog input pins 4 to 7 (AN4 to AN7) comprising group1. The AVCC and AVSS pins are the power supply pins for the analog block in the A/D converter.

Table 17.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVCC	Input	Analog block power supply
Analog ground pin	AVSS	Input	Analog block ground and reference voltage
Analog input pin 0	AN0	Input	Group 0 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	Group 1 analog input pins
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input pin for starting A/D conversion

17.3 Register Descriptions

The A/D converter has the following registers.

- A/D data register A (ADDRA)
- A/D data register B (ADDRB)
- A/D data register C (ADDRC)
- A/D data register D (ADDRD)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

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

There are four 16-bit read-only ADDR registers, ADDRA to ADDRD, used to store the results of A/D conversion. The ADDR registers which store a conversion result for each channel are shown in table 17.2.

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

The data bus between the CPU and A/D converter is eight bits wide. The upper byte can be read directly from the CPU. However, when the lower byte is read from, data that was transferred to a temporary register at reading of the upper byte is read. Accordingly, when reading from ADDR, access in word units or access upper byte first, and then lower byte.

Table 17.2 Analog Input Channels and Corresponding ADDR

Analog Input Channel		A/D Data Register to Store A/D Conversion Results
Group 0	Group 1	
AN0	AN4	ADDRA
AN1	AN5	ADDRB
AN2	AN6	ADDRC
AN3	AN7	ADDRD

17.3.2 A/D Control/Status Register (ADCSR)

ADCSR controls A/D converter operation.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the end of A/D conversion. [Setting conditions] • When A/D conversion ends in single mode • When A/D conversion ends on all channels specified in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 • When DTC is activated by an ADI interrupt and ADDR is read
6	ADIE	0	R/W	A/D Interrupt Enable Enables ADI interrupt by ADF when this bit is set to 1.
5	ADST	0	R/W	A/D Start Setting this bit to 1 starts A/D conversion. In single mode, this bit is cleared to 0 automatically when conversion on the specified channel ends. In scan mode, conversion continues sequentially on the specified channels until this bit is cleared to 0 by software, a reset, or a transition to standby mode or module stop mode.
4	SCAN	0	R/W	Scan Mode Selects the A/D converter operating mode. 0: Single mode 1: Scan mode Switch the operating mode when ADST = 0.
3	CKS	0	R/W	Clock Select Sets A/D conversion time. 0: Conversion time is 266 states (max) 1: Conversion time is 134 states (max) (when the system clock (φ) is 16 MHz or lower) Switch conversion time while the ADST bit is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description	
2	CH2	0	R/W	Channel Select 2 to 0	
1	CH1	0	R/W	Select analog input channels.	
0	CH0	0	R/W	When SCAN = 0	When SCAN = 1
				000: AN0	000: AN0
				001: AN1	001: AN0 and AN1
				010: AN2	010: AN0 to AN2
				011: AN3	011: AN0 to AN3
				100: AN4	100: AN4
				101: AN5	101: AN4 and AN5
				110: AN6	110: AN4 to AN6
				111: AN7	111: AN4 to AN7
				Switch input channels when ADST = 0.	

Note: * Only 0 can be written for clearing the flag.

17.3.3 A/D Control Register (ADCR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 1 and 0
6	TRGS0	0	R/W	Enable the start of A/D conversion by a trigger signal. Set these bits only while A/D conversion is stopped (ADST = 0).
				00: A/D conversion start by external trigger is disabled
				01: Setting prohibited
				10: A/D conversion start by conversion trigger from TMR
				11: A/D conversion start by $\overline{\text{ADTRG}}$ pin
5 to 0	—	All 1	R/W	Reserved
				The initial value should not be changed.

17.4 Operation

The A/D converter operates by successive approximation with 10-bit resolution. It has two operating modes: single mode and scan mode. When changing the operating mode or analog input channel, to prevent incorrect operation, first clear the ADST bit in ADCSR to 0 to halt A/D conversion. The ADST bit can be set at the same time the operating mode or analog input channel is changed.

17.4.1 Single Mode

In single mode, A/D conversion is to be performed only once on the specified single channel. Operations are as follows.

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

17.4.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the specified channels (max. four channels). Operations are as follows.

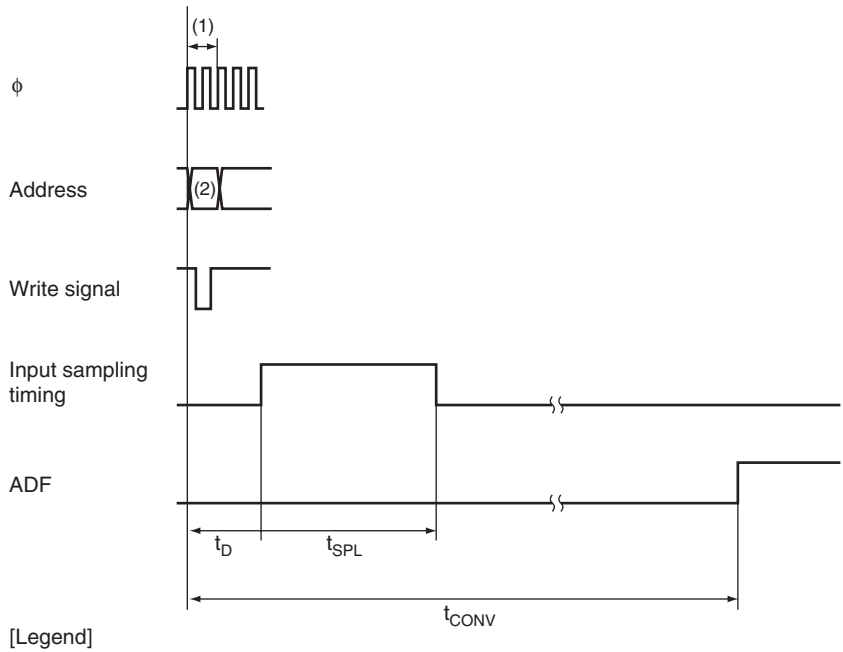
1. When the ADST bit in ADCSR is set to 1 by software or an external trigger input, A/D conversion starts on the first channel in the group (AN0 when the CH2 bit in ADCSR is 0, or AN4 when the CH2 bit in ADCSR is 1).
2. When A/D conversion for each channel is completed, the result is sequentially transferred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

17.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input when the A/D conversion start delay time (t_D) passes after the ADST bit in ADCSR is set to 1, then starts A/D conversion. Figure 17.2 shows the A/D conversion timing. Table 17.3 indicates the A/D conversion time.

As indicated in figure 17.2, the A/D conversion time (t_{CONV}) includes t_D and the input sampling time (t_{SPL}). The length of t_D varies depending on the timing of write to ADCSR. The total conversion time therefore varies within the ranges indicated in table 17.3.

In scan mode, the values shown in table 17.3 become those for the first conversion time. For the second and subsequent conversions, the conversion time is 266 states (fixed) when CKS = 0 and 134 states (fixed) when CKS = 1. Use the conversion time of 134 states only when the system clock (ϕ) is 16 MHz or lower.



[Legend]
(1): ADCSR write cycle
(2): ADCSR address
t_D: A/D conversion start delay
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 17.2 A/D Conversion Timing

Table 17.3 A/D Conversion Time (Single Mode)

		CKS = 0			CKS = 1*		
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**
A/D conversion start delay time	t_D	10	—	17	6	—	9
Input sampling time	t_{SPL}	—	63	—	—	31	—
A/D conversion time	t_{CONV}	259	—	266	131	—	134

Notes: Values in the table indicate the number of states.
* in the table indicates that the system clock (φ) is 16 MHz or lower.

17.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS1 and TRGS0 bits are set to B'11 in ADCR, an external trigger is input to the $\overline{\text{ADTRG}}$ pin. The ADST bit in ADCSR is set to 1 at the falling edge of the $\overline{\text{ADTRG}}$ pin, thus starting A/D conversion. Other operations, in both single and scan modes, are the same as when the ADST bit has been set to 1 by software. Figure 17.3 shows the timing.

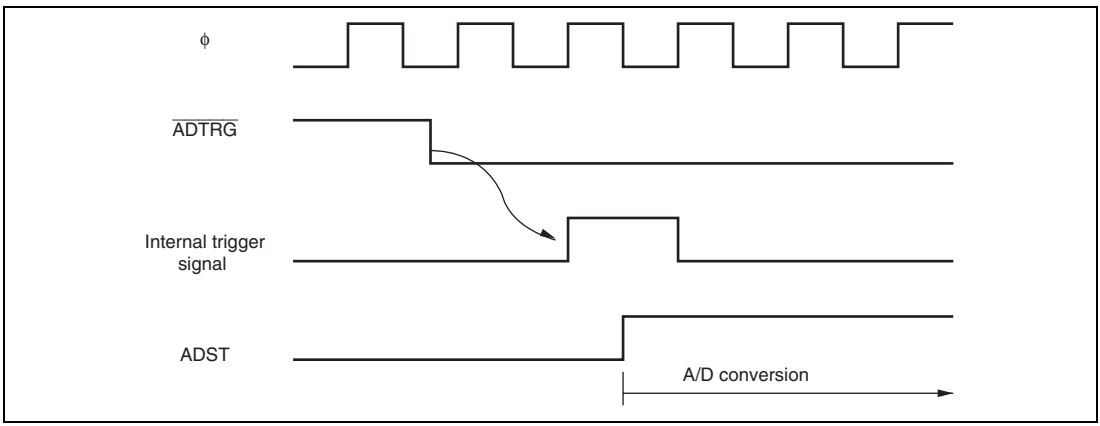


Figure 17.3 External Trigger Input Timing

17.5 Interrupt Source

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. If the ADF bit in ADCSR has been set to 1 after A/D conversion ends and the ADIE bit is set to 1, an ADI interrupt request is enabled.

The ADI interrupt can be used to activate the on-chip DTC.

Table 17.4 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation
ADI	A/D conversion end	ADF	Enable

17.6 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 17.4).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from the minimum voltage value B'00 0000 0000 (H'000) to B'00 0000 0001 (H'001) (see figure 17.5).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from B'11 1111 1110 (H'3FE) to B'11 1111 1111 (H'3FF) (see figure 17.5).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristics between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error (see figure 17.5).
- Absolute accuracy

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

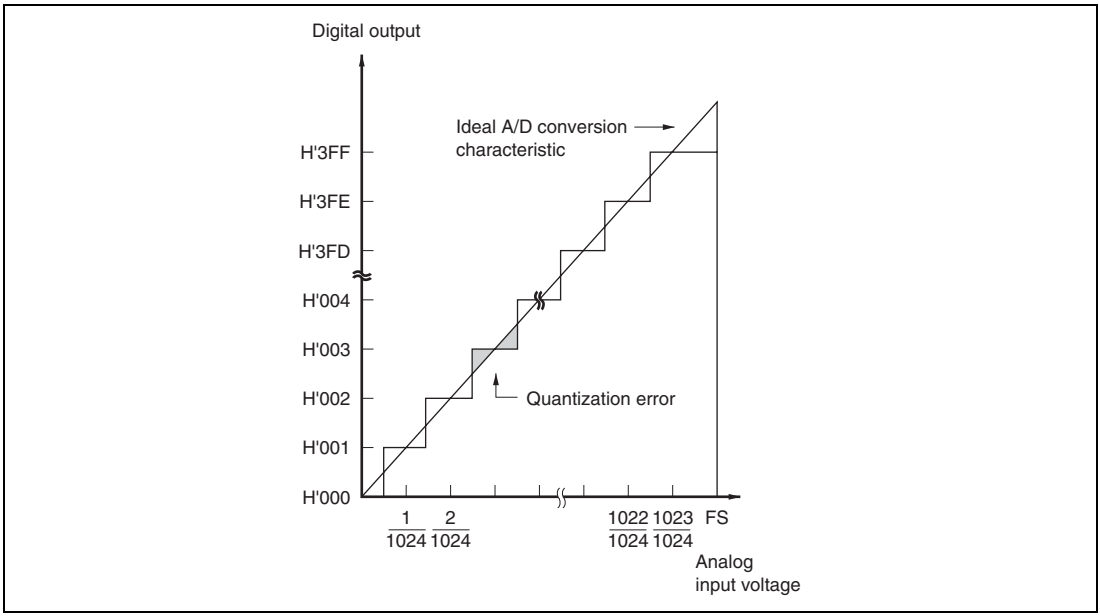


Figure 17.4 A/D Conversion Accuracy Definitions

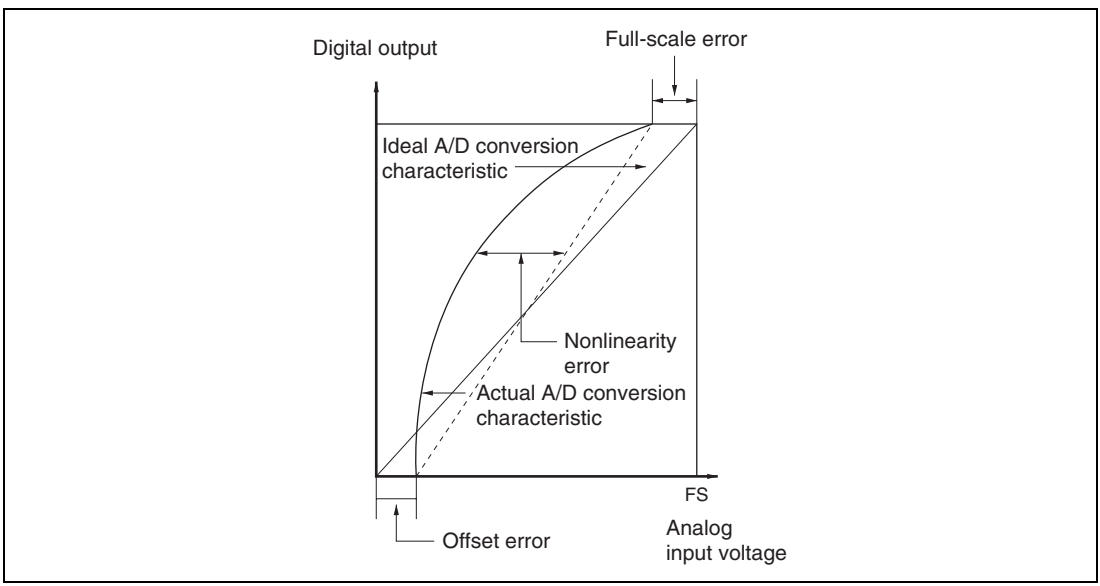


Figure 17.5 A/D Conversion Accuracy Definitions

17.7 Usage Notes

17.7.1 Permissible Signal Source Impedance

This LSI's analog input is designed so that the conversion accuracy is guaranteed for an input signal for which the signal source impedance is 5 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 5 kΩ, charging may be insufficient and it may not be possible to guarantee the A/D conversion accuracy. However, if a large capacitance is provided externally in single mode, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored. However, since 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., voltage fluctuation ratio of 5 mV/μs or greater) (see figure 17.6). When converting a high-speed analog signal or converting in scan mode, a low-impedance buffer should be inserted.

17.7.2 Influences on Absolute Accuracy

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

Care is also required to insure that filter circuits do not interfere with digital signals on the mounting board, so acting as antennas.

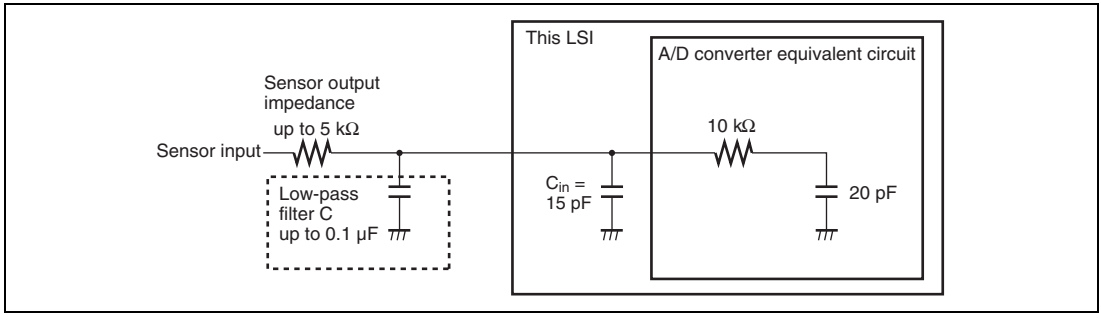


Figure 17.6 Example of Analog Input Circuit

17.7.3 Setting Range of Analog Power Supply and Other Pins

If conditions shown below are not met, the reliability of this LSI may be adversely affected.

- Analog input voltage range
  The voltage applied to analog input pin ANn during A/D conversion should be in the range AVSS ≤ ANn ≤ AVCC (n = 0 to 7).
- Relation between AVCC, VSS and VCC, VSS
  For the relationship between AVCC, AVSS and VCC, VSS, set AVSS = VSS, but AVCC = VCC is not necessary and which one is greater does not matter. Even when the A/D converter is not used, the ACC and AVSS pins must on no account be left open.

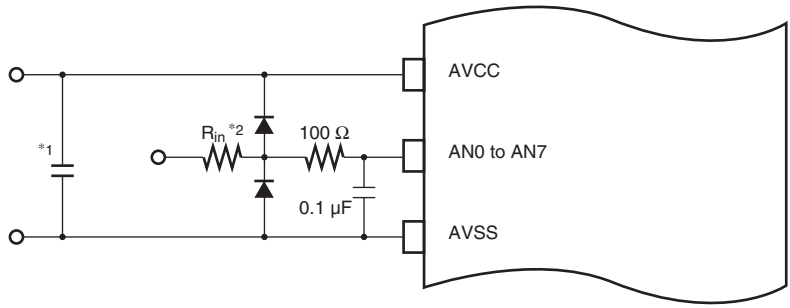
17.7.4 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values. Also, digital circuitry must be isolated from the analog input pins (AN0 to AN7) and analog power supply voltage (AVCC) by the analog ground (AVSS). Also, the analog ground (AVSS) should be connected at one point to a stable ground (VSS) on the board.

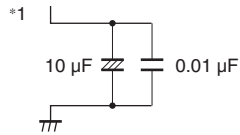
17.7.5 Notes on Noise Countermeasures

A protection circuit connected to prevent damage of the analog input pins (AN0 to AN7) due to an abnormal voltage such as an excessive surge should be connected between AVCC and AVSS, as shown in figure 17.7. Also, the bypass capacitors connected to AVCC and the filter capacitors connected to AN0 to AN7 must be connected to AVSS.

If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN7) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance (R_{in}), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding the circuit constants.

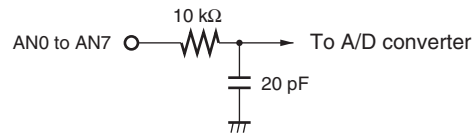


Notes: Values are reference values.



*2 R_{in}: Input impedance

Figure 17.7 Example of Analog Input Protection Circuit



Note: Values are reference values.

Figure 17.8 Analog Input Pin Equivalent Circuit

17.7.6 Module Stop Mode Setting

A/D converter operation can be enabled or disabled by the module stop control register. In the initial state, A/D converter operation is disabled. Access to A/D converter registers is enabled when module stop mode is cancelled. For details, see section 22, Power-Down Modes.

Section 18 RAM

This LSI has 8 Kbytes of on-chip high-speed static RAM. The RAM is connected to the CPU by a 16-bit data bus, enabling one-state access by the CPU for both byte data and word data.

The on-chip RAM can be enabled or disabled by means of the RAME bit in the system control register (SYSCR). For details on SYSCR, see section 3.2.2, System Control Register (SYSCR).

Section 19 Flash Memory (0.18-µm F-ZTAT Version)

The flash memory has the following features. Figure 19.1 shows a block diagram of the flash memory.

19.1 Features

- Size

Product Classification		ROM Size	ROM Addresses
H8S/2125	R4F2125	512 Kbytes	H'000000 to H'07FFFF (mode 2) H'0000 to H'DFFF (mode 3)

- Two flash-memory MATs according to LSI initiation mode

  The on-chip flash memory has two memory spaces in the same address space (hereafter referred to as memory MATs). The mode setting at initiation determines which memory MAT is initiated first. The MAT can be switched by using the bank-switching method after initiation.

  — The user MAT is initiated at a power-on reset in user mode: 512 Kbytes

  — The user boot memory MAT is initiated at a power-on reset in user boot mode: 8 Kbytes
- Programming/erasing interface by the download of on-chip program

  This LSI has a dedicated programming/erasing program. After downloading this program to the on-chip RAM, programming/erasing can be performed by setting the argument parameter.
- Programming/erasing time

  The flash memory programming time is 3 ms (typ) in 128-byte simultaneous programming, and approximately 25 µs per byte. The erasing time is 1000 ms (typ) per 64-Kbyte block.
- Number of programming

  The number of flash memory programming can be up to 100 times at the minimum. (The value ranged from 1 to 100 is guaranteed.)
- Three on-board programming modes

  — Boot mode

  This mode is a program mode that uses an on-chip SCI interface. The user MAT and user boot MAT can be programmed. In this mode, the bit rate between the host and this LSI can be automatically adjusted.

  — User program mode

  The user MAT can be programmed by using the optional interface.

— User boot mode

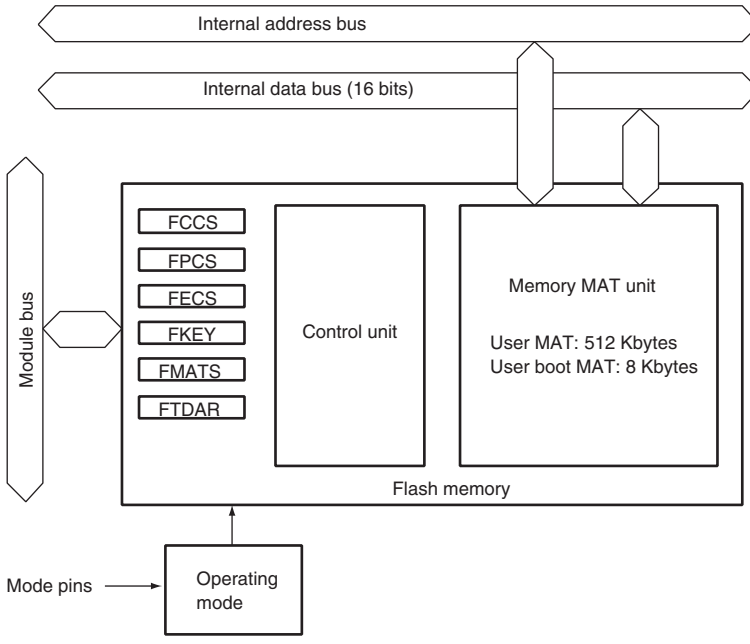
The user boot program of the optional interface can be made and the user MAT can be programmed.

- Programming/erasing protection

  Sets protection against flash memory programming/erasing via hardware, software, or error protection.

- Programmer mode

  This mode uses the PROM programmer. The user MAT and user boot MAT can be programmed.



[Legend]
FCCS: Flash code control status register
FPCS: Flash program code select register
FECS: Flash erase code select register
FKEY: Flash key code register
FMATS: Flash MAT select register
FTDAR: Flash transfer destination address register

Note: To read from or write to the registers, the FLSHE bit in the serial timer control register (STCR) must be set to 1.

Figure 19.1 Block Diagram of Flash Memory

19.1.1 Mode Transitions

When each mode pin is set in the reset state and the reset is started, this LSI enters each operating mode as shown in figure 19.2.

1. Flash memory can be read in user mode, but cannot be programmed or erased.
2. Flash memory can be read, programmed, or erased on the board only in user program mode, user boot mode, and boot mode.
3. Flash memory can be read, programmed, or erased by means of the PROM programmer in programmer mode.

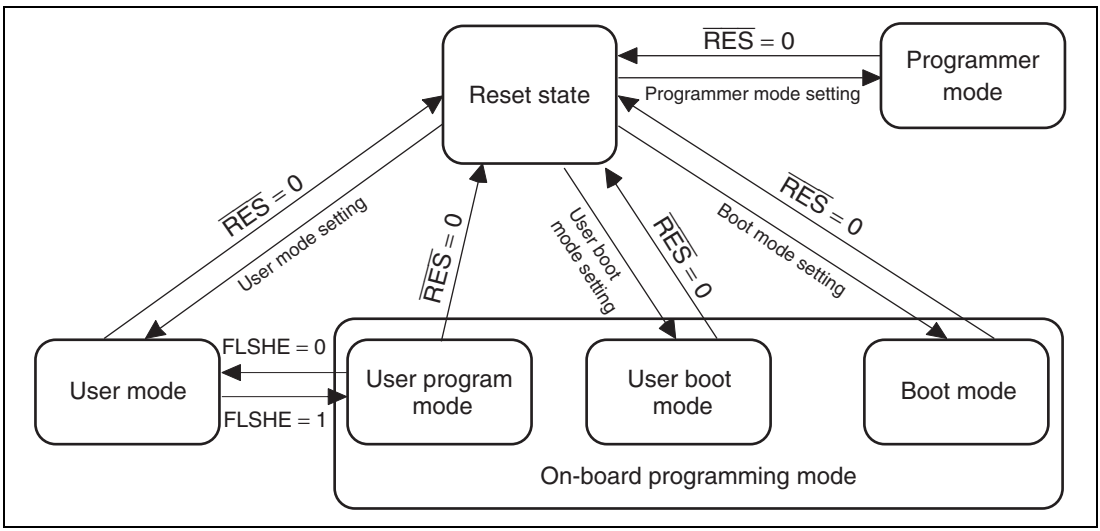


Figure 19.2 Mode Transition for Flash Memory

19.1.2 Mode Comparison

The comparison table of programming and erasing related items about boot mode, user program mode, user boot mode, and programmer mode is shown in table 19.1.

Table 19.1 Comparison of Programming Modes

	Boot Mode	User Program Mode	User Boot Mode	Programmer Mode
Programming/ erasing environment	On-board	On-board	On-board	PROM programmer
Programming/ erasing enable MAT	User MAT User boot MAT	User MAT	User MAT	User MAT User boot MAT
All erasure	O (Automatic)	O	O	O (Automatic)
Block division erasure	O*[1]	O	O	×
Program data transfer	From host via SCI	Via optional device	Via optional device	Via programmer
Reset initiation MAT	Embedded program storage MAT	User MAT	User boot MAT*[2]	—
Transition to user mode	Changing mode setting and reset	Changing FLSHE bit setting	Changing mode setting and reset	—

Notes: 1. All erasure is performed. After that, the specified block can be erased.
2. First, the reset vector is fetched from the embedded program storage MAT. After the flash memory related registers are checked, the reset vector is fetched from the user boot MAT.

- The user boot MAT can be programmed or erased only in boot mode and programmer mode.
- In boot mode, the user MAT and user boot MAT are totally erased. Then, the user MAT or user boot MAT can be programmed by means of commands. Note that the contents of the MAT cannot be read until this state.

  Boot mode can be used for programming only the user boot MAT and then programming the user MAT in user boot mode. Another way is to program only the user MAT since user boot mode is not used.
- In user boot mode, boot operation of the optional interface can be performed with mode pin settings different from those in user program mode.

19.1.3 Flash Memory MAT Configuration

This LSI's flash memory is configured by the 512-Kbyte user MAT and 8-Kbyte user boot MAT.

The start address is allocated to the same address in the user MAT and user boot MAT. Therefore, when program execution or data access is performed between two MATs, the MAT must be switched by using FMATS.

The user MAT or user boot MAT can be read in all modes. However, the user boot MAT can be programmed only in boot mode and programmer mode.

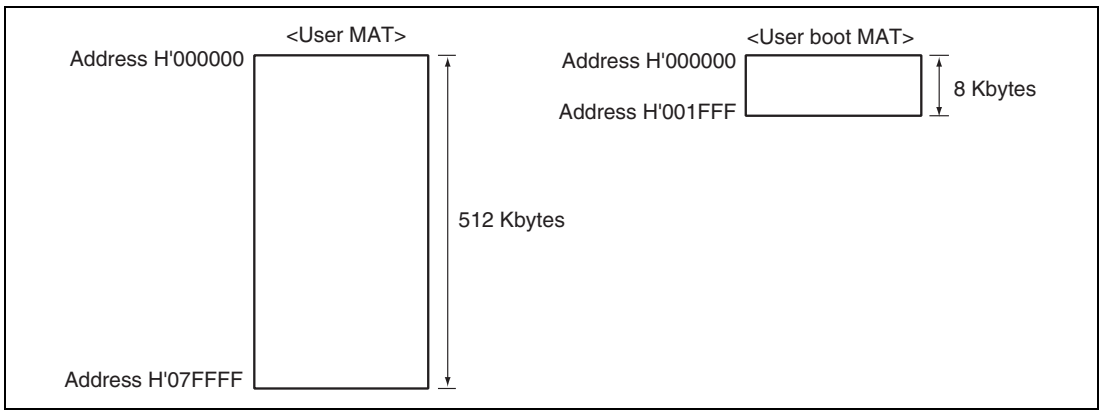


Figure 19.3 Flash Memory Configuration

The size of the user MAT is different from that of the user boot MAT. An address that exceeds the size of the 8-Kbyte user boot MAT should not be accessed. If the attempt is made, data is read as an undefined value.

19.1.4 Block Division

The user MAT is divided into 64 Kbytes (seven blocks), 32 Kbytes (one block), and 4 Kbytes (eight blocks) as shown in figure 19.4. The user MAT can be erased in this divided-block units by specifying the erase-block number of EB0 to EB15 when erasing.

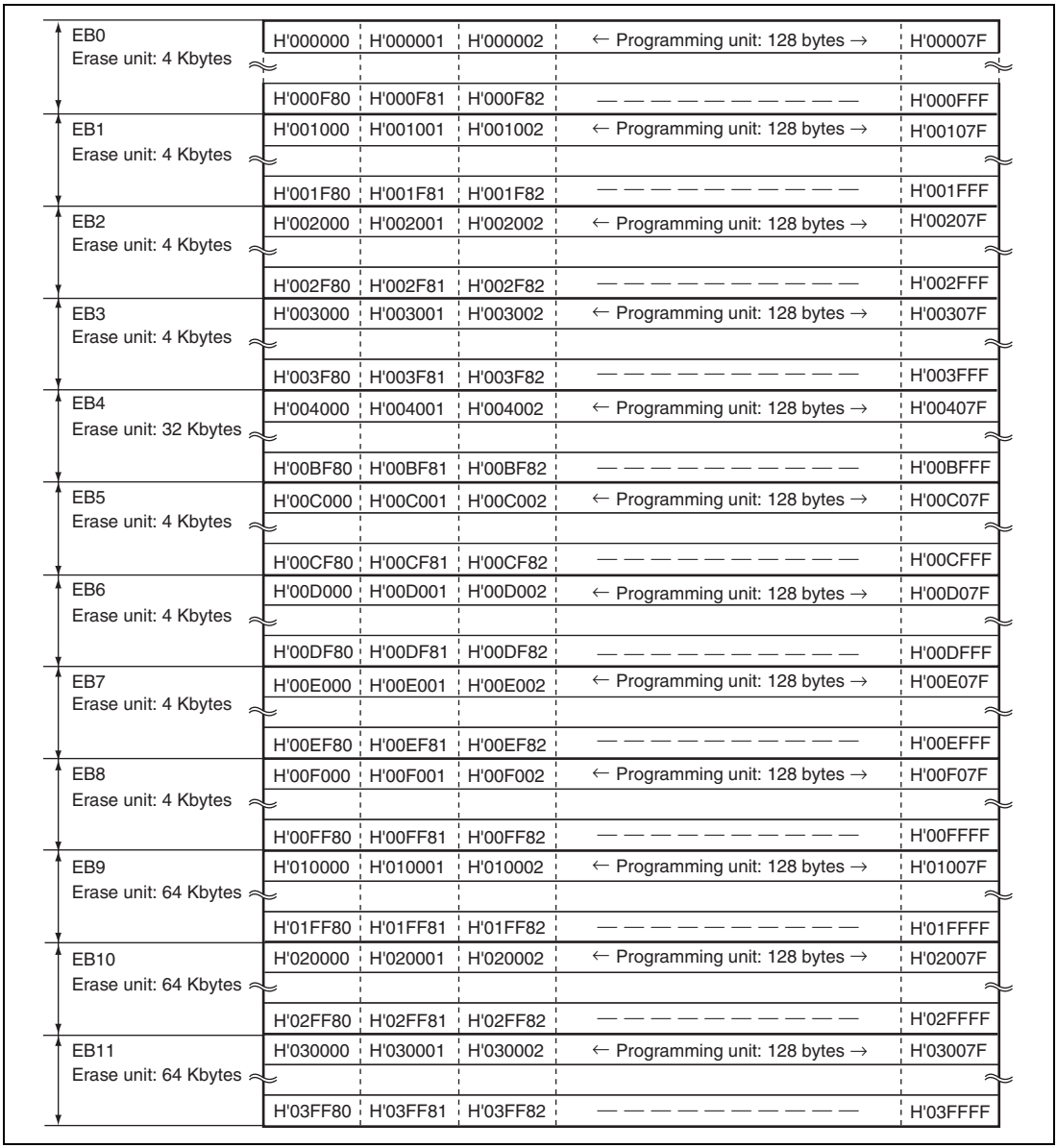


Figure 19.4 Block Division of User MAT (1)

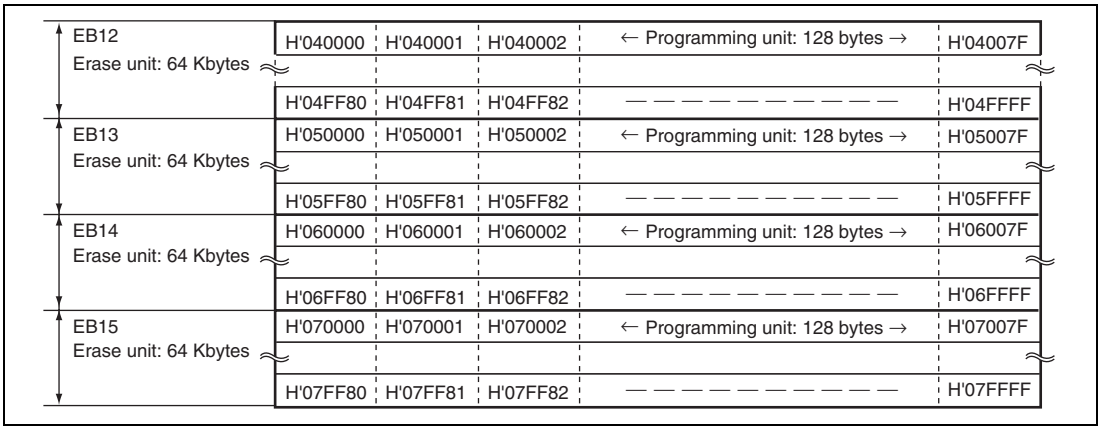


Figure 19.4 Block Division of User MAT (2)

19.1.5 Programming/Erasing Interface

Programming/erasing is executed by downloading the on-chip program to the on-chip RAM and specifying the program address/data and erase block by using the interface register/parameter.

The procedure program is made by the user in user program mode and user boot mode. An overview of the procedure is given as follows. For details, see section 19.4.2, User Program Mode.

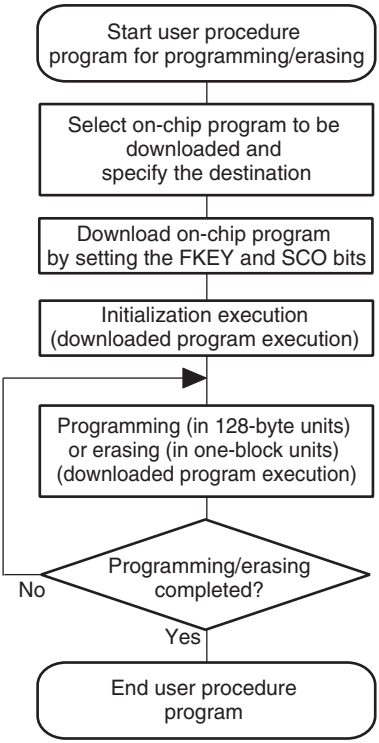


Figure 19.5 Overview of User Procedure Program

1. Selection of on-chip program to be downloaded

   For programming/erasing execution, set the FLSHE bit in STCR to 1 to make a transition to user program mode.

   This LSI has programming/erasing programs that can be downloaded to the on-chip RAM. The on-chip program to be downloaded is selected by setting the corresponding bits in the programming/erasing interface register. The address of the download destination is specified by the flash transfer destination address register (FTDAR).

2. Download of on-chip program

   The on-chip program is automatically downloaded by setting the flash key code register (FKEY) and the SCO bit in the flash code control status register (FCCS), which are programming/erasing interface registers.

   The flash memory MAT is replaced with the embedded program storage MAT during downloading. Since the flash memory cannot be read during programming/erasing, the procedure program that executes download to completion of programming/erasing must be executed in a space other than flash memory (for example, on-chip RAM).

   Since the result of download is returned to the programming/erasing interface parameter, whether download has succeeded or not can be confirmed.

3. Initialization of programming/erasing

   Set the operating frequency before execution of programming/erasing. This setting is performed by using the programming/erasing interface parameter.

4. Execution of programming/erasing

   For programming/erasing execution, set the FLSHE bit in STCR to 1 to make a transition to user program mode.

   The program data/programming destination address is specified in 128-byte units for programming. The block to be erased is specified in erase-block units for erasing.

   Make these specifications by using the programming/erasing interface parameter, and then initiate the on-chip program. The on-chip program is executed by using the JSR or BSR instruction to execute the subroutine call of the specified address in the on-chip RAM. The execution result is returned to the programming/erasing interface parameter.

   The area to be programmed must be erased in advance when programming flash memory. All interrupts must be disabled during programming and erasing. Interrupts must be masked within the user system.

5. Consecutive execution of programming/erasing

   When the 128-byte programming or one-block erasure does not end the processing, the program address/data and erase-block number must be updated and consecutive programming/erasing is required.

   Since the downloaded on-chip program remains in the on-chip RAM even after the processing ends, download and initialization are not required when the same processing is executed consecutively.

19.2 Input/Output Pins

Flash memory is controlled by the pins listed in table 19.2.

Table 19.2 Pin Configuration

Pin Name	Input/Output	Function
$\overline{\text{RES}}$	Input	Reset
MD2*	Input	Sets operating mode of this LSI
MD1	Input	Sets operating mode of this LSI
MD0	Input	Sets operating mode of this LSI
TxD1	Output	Serial transmit data output (used in boot mode)
RxD1	Input	Serial receive data input (used in boot mode)

Note: * MD2 is not supported in SDIP-64 and QFP-64.

19.3 Register Descriptions

The registers/parameters that control flash memory are shown below. To read from or write to these registers/parameters, the FLSHE bit in STCR must be set to 1. For details on STCR, see section 3.2.3, Serial Timer Control Register (STCR).

- Flash code control status register (FCCS)
- Flash program code select register (FPCS)
- Flash erase code select register (FECS)
- Flash key code register (FKEY)
- Flash MAT select register (FMATS)
- Flash transfer destination address register (FTDAR)
- Download pass/fail result (DPFR)
- Flash pass/fail result (FPFR)
- Flash multipurpose address area (FMPAR)
- Flash multipurpose data destination area (FMPDR)
- Flash erase block select (FEBS)
- Flash programming/erasing frequency control (FPEFEQ)

There are several operating modes for accessing flash memory, for example, read mode/program mode.

There are two memory MATs: user MAT and user boot MAT. The dedicated registers/parameters are allocated for each operating mode and MAT selection. The correspondence between operating modes and registers/parameters for use is shown in table 19.3.

Table 19.3 Register/Parameter and Target Mode

		Download	Initialization	Programming	Erasure	Read
Programming/ erasing interface registers	FCCS	O	—	—	—	—
	FPCS	O	—	—	—	—
	FECS	O	—	—	—	—
	FKEY	O	—	O	O	—
	FMATS	—	—	O*[1]	O*[1]	O*[2]
	FTDAR	O	—	—	—	—
Programming/ erasing interface parameters	DPFR	O	—	—	—	—
	FPFR	—	O	O	O	—
	FPEFEQ	—	O	—	—	—
	FMPAR	—	—	O	—	—
	FMPDR	—	—	O	—	—
	FEBS	—	—	—	O	—

Notes: 1. The setting is required when programming or erasing the user MAT in user boot mode.
2. The setting may be required according to the combination of initiation mode and read target MAT.

19.3.1 Programming/Erasing Interface Registers

The programming/erasing interface registers are all 8-bit registers that can be accessed in bytes. These registers are initialized at a reset or in hardware standby mode.

- Flash Code Control Status Register (FCCS)

  FCCS requests monitoring error occurrence during programming or erasing flash memory, and the download of an on-chip program.

Bit	Bit Name	Initial Value	R/W	Description
7	FWE	1	R	Flash Write Enable This bit is always read as 1 and cannot be modified.
6	—	0	R/W	Reserved
5	—	0	R/W	The initial value should not be changed.

Bit	Bit Name	Initial Value	R/W	Description
4	FLER	0	R	Flash Memory Error Indicates an error has occurred during programming or erasing flash memory. When this bit is set to 1, flash memory enters the error-protection state. In case this bit is set to 1, high voltage is applied to the internal flash memory. To reduce the damage to flash memory, the reset must be released after a reset period of 100 µs which is longer than normal. 0: Flash memory operates normally. Programming/erasing protection (error protection) for flash memory is invalid. [Clearing condition] • At a reset or in hardware standby mode 1: An error occurs during programming/erasing flash memory. Programming/erasing protection (error protection) for flash memory is valid. [Setting conditions] • When an interrupt, such as NMI, occurs during programming/erasing flash memory. • When flash memory is read during programming/erasing flash memory (including a vector read or an instruction fetch). • When the SLEEP instruction is executed during programming/erasing flash memory (including software standby mode) • When a bus master other than the CPU, such as the DTC or LPC, gets bus mastership during programming/erasing flash memory.
3 to 1	—	All 0	R/W	Reserved The initial value should not be changed.

Bit	Bit Name	Initial Value	R/W	Description
0	SCO	0	(R)/W*	Source Program Copy Operation Requests the on-chip programming/erasing program to be downloaded to the on-chip RAM. When this bit is set to 1, the on-chip program which is selected by FPCS/FECS is automatically downloaded in the on-chip RAM specified by FTDAR. In order to set this bit to 1, H'A5 must be written to FKEY and this operation must be executed in the on-chip RAM. Immediately after setting this bit to 1, four NOP instructions must be executed. Since this bit is cleared to 0 when download is completed, this bit cannot be read as 1. All interrupts must be disabled during downloading. Interrupts must be masked within the user system. 0: Download of the on-chip programming/erasing program to the on-chip RAM is not executed. [Clearing condition] When download is completed 1: Request to download the on-chip programming/erasing program to the on-chip RAM has occurred. [Setting conditions] When all of the following conditions are satisfied and this bit is set to 1 • H'A5 is written to FKEY • During execution in the on-chip RAM

Note: * This bit is a write only bit. This bit is always read as 0.

- Flash Program Code Select Register (FPCS)

  FPCS selects the on-chip programming program to be downloaded.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	PPVS	0	R/W	Program Pulse Verify Selects the programming program. 0: On-chip programming program is not selected. [Clearing condition] When transfer is completed 1: On-chip programming program is selected.

- Flash Erase Code Select Register (FECS)

  FECS selects the on-chip erasing program to be downloaded.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	EPVB	0	R/W	Erase Pulse Verify Block Selects the erasing program. 0: On-chip erasing program is not selected. [Clearing condition] When transfer is completed 1: On-chip erasing program is selected.

- Flash Key Code Register (FKEY)

  FKEY is for software protection that enables download of an on-chip program and programming/erasing of flash memory. Before setting the SCO bit to 1 to download an on-chip program or before executing the downloaded programming/erasing program, the key code must be written, otherwise the processing cannot be executed.

Bit	Bit Name	Initial Value	R/W	Description
7	K7	0	R/W	Key Code
6	K6	0	R/W	Only when H'A5 is written, writing to the SCO bit is valid. When a value other than H'A5 is written to FKEY, 1 cannot be set to the SCO bit. Therefore downloading to the on-chip RAM cannot be executed. Only when H'5A is written, programming/erasing can be executed. Even if the on-chip programming/erasing program is executed, the flash memory cannot be programmed or erased when a value other than H'5A is written to FKEY. H'A5: Writing to the SCO bit is enabled. (The SCO bit cannot be set by a value other than H'A5.) H'5A: Programming/erasing is enabled. (Software protection state is entered for a value other than H'5A.) H'00: Initial value
5	K5	0	R/W	
4	K4	0	R/W	
3	K3	0	R/W	
2	K2	0	R/W	
1	K1	0	R/W	
0	K0	0	R/W	

- Flash MAT Select Register (FMATS)

  FMATS specifies whether the user MAT or user boot MAT is selected.

Bit	Bit Name	Initial Value	R/W	Description
7	MS7	0/1*	R/W	MAT Select
6	MS6	0	R/W	The user MAT is selected when a value other than H'AA is written, and the user boot MAT is selected when H'AA is written. The MAT is switched by writing a value in FMATS. When the MAT is switched, follow section 19.6, Switching between User MAT and User Boot MAT. (The user boot MAT cannot be programmed in user program mode even if the user boot MAT is selected by FMATS. The user boot MAT must be programmed in boot mode or programmer mode.) H'AA: User boot MAT is selected (user MAT is selected when the value of these bits is other than H'AA). Initial value when initiated in user boot mode. H'00: Initial value when initiated in a mode except for user boot mode (user MAT is selected) [Programmable condition] In the execution state in the on-chip RAM
5	MS5	0/1*	R/W	
4	MS4	0	R/W	
3	MS3	0/1*	R/W	
2	MS2	0	R/W	
1	MS1	0/1*	R/W	
0	MS0	0	R/W	

Note: * Set to 1 in user boot mode, otherwise cleared to 0.

- Flash Transfer Destination Address Register (FTDAR)

  FTDAR specifies the on-chip RAM address where an on-chip program is downloaded. This register must be specified before setting the SCO bit in FCCS to 1.

Bit	Bit Name	Initial Value	R/W	Description
7	TDER	0	R/W	Transfer Destination Address Setting Error This bit is set to 1 when the address specified by bits TDA6 to TDA0, which is the start address where an on-chip program is downloaded, is over the range. Whether or not the address specified by bits TDA6 to TDA0 is within the range of H'00 to H'02 is determined when an on-chip program is downloaded by setting the SCO bit in FCCS to 1. Make sure that this bit is cleared to 0 and the value specified by bits TDA6 to TDA0 is within the range of H'00 to H'02 before setting the SCO bit to 1. 0: The value specified by bits TDA6 to TDA0 is within the range. 1: The value specified by bits TDA6 to TDA0 is outside the range (H'03 to H'7F) and download is stopped.
6	TDA6	0	R/W	Transfer Destination Address Specifies the start address where an on-chip program is downloaded. A value of H'00 can be specified as the download start address in the on-chip RAM. H'00: H'FFD080 is specified as the download start address. H'01: H'FFD880 is specified as the download start address. H'02: H'FFE080 is specified as the download start address. H'03 to H'7F: Setting prohibited. Specifying this value sets the TDER bit to 1 during downloading and stops the download.
5	TDA5	0	R/W	
4	TDA4	0	R/W	
3	TDA3	0	R/W	
2	TDA2	0	R/W	
1	TDA1	0	R/W	
0	TDA0	0	R/W	

19.3.2 Programming/Erasing Interface Parameters

The programming/erasing interface parameters specify the operating frequency, storage place for program data, programming destination address, and erase block and exchanges the processing result for the downloaded on-chip program. These parameters use the CPU general registers (ER0 and ER1) or the on-chip RAM area. The initial value is undefined at a reset or in hardware standby mode.

In download, initialization, or execution of the on-chip program, registers of the CPU except for R0L are stored. The return value of the processing result is written in R0L. Since the stack area is used for storing the registers except for R0L, the stack area must be saved at the processing start. (A maximum size of a stack area to be used is 128 bytes.)

The programming/erasing interface parameters is used for the following four functions:

1. Download control
2. Initialization before programming or erasing
3. Programming
4. Erasing

These items use different parameters. The correspondence table is shown in table 19.4.

The meaning of bits in FPFR varies in each processing: initialization, programming, or erasure. For details, see descriptions of FPFR for each processing.

Table 19.4 Parameters and Target Modes

Parameter Name	Abbreviation	Download	Initialization	Programming	Erasure	R/W	Initial Value	Allocation
Download pass/fail result	DPFR	O	—	—	—	R/W	Undefined	On-chip RAM*
Flash pass/fail result	FPFR	—	O	O	O	R/W	Undefined	R0L of CPU
Flash programming/ erasing frequency control	FPEFEQ	—	O	—	—	R/W	Undefined	ER0 of CPU
Flash multipurpose address area	FMPAR	—	—	O	—	R/W	Undefined	ER1 of CPU
Flash multipurpose data destination area	FMPDR	—	—	O	—	R/W	Undefined	ER0 of CPU
Flash erase block select	FEBS	—	—	—	O	R/W	Undefined	R0L of CPU

Note: * A single byte of the download start address specified by FTDAR.

(1) Download Control

The on-chip program is automatically downloaded by setting the SCO bit to 1. The on-chip RAM area where the program is to be downloaded is the 2-Kbyte area starting from the address specified by FTDAR.

Download control is set by the programming/erasing interface registers, and the DPFR parameter indicates the return value.

(a) Download pass/fail result parameter (DPFR: single byte of start address specified by FTDAR)

This parameter indicates the return value of the download result. The value of this parameter can be used to determine if downloading was executed or not. Since confirmation whether the SCO bit is set to 1 or not is difficult, certain determination must be gained by setting a value other than the return value of download (for example, H'FF) to the single byte of the start address specified by FTDAR before download starts (before setting the SCO bit to 1).

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Unused The return value is 0.
2	SS	—	R/W	Source Select Error Detect Only one type can be specified for the on-chip program that can be downloaded. When more than two types of programs are selected, the program is not selected, or the program is selected without mapping, an error occurs. 0: Download program selection is normal 1: Download error has occurred (multi-selection or program which is not mapped is selected)
1	FK	—	R/W	Flash Key Register Error Detect Returns the check result whether the FKEY value is set to H'A5. 0: FKEY setting is normal (FKEY = H'A5) 1: FKEY setting is abnormal (FKEY = value other than H'A5)
0	SF	—	R/W	Success/Fail Returns the result whether download has ended normally or not. Determines the result whether the program was correctly downloaded to the on-chip RAM by way of the confirming reading of it. 0: Download to on-chip program has ended normally (no error) 1: Download to on-chip program has ended abnormally (error occurred)

(2) Programming/Erasing Initialization

The on-chip programming/erasing program to be downloaded includes the initialization program.

A pulse of the specified width must be applied when programming or erasing. The specified pulse width is made by the method in which a wait loop is configured by CPU instructions. The operating frequency of the CPU must be set too.

The initialization program is used to set the above values as parameters of the programming/erasing program that was downloaded.

(a) Flash programming/erasing frequency control parameter (FPEFEQ: general register ER0 of CPU)

This parameter sets the operating frequency of the CPU. The settable range of the operating frequency in this LSI is 8 to 20 MHz.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	—	Unused These bits should be cleared to 0.
15 to 0	F15 to F0	—	R/W	Frequency Set These bits set the operating frequency of the CPU. The setting value must be calculated with the following procedure. 1. The operating frequency shown in MHz units must be rounded off to two decimals. 2. The value multiplied by 100 is converted to the hexadecimal numeral and written to the FPEFEQ parameter (general register ER0). For example, when the operating frequency of the CPU is 20.000 MHz, the setting value is as follows: 1. 20.000 is rounded off to two decimals, thus becoming 20.00. 2. The formula of 20.00 × 100 = 2000 is converted to the hexadecimal numeral and H'07D0 is set to ER0.

(b) Flash pass/fail result parameter (FPFR: general register R0L of CPU)

This parameter indicates the return value of the initialization result.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	—	—	Unused The return value is 0.
1	FQ	—	R/W	Frequency Error Detect Returns the check result whether the specified CPU operating frequency is in the range of the supported operating frequency. 0: Setting of operating frequency is normal 1: Setting of operating frequency is abnormal
0	SF	—	R/W	Success/Fail Indicates whether initialization has ended normally or not. 0: Initialization has ended normally (no error) 1: Initialization has ended abnormally (error occurred)

(3) Programming Execution

When flash memory is programmed, the programming destination address on the user MAT must be passed to the programming program in which the program data has been downloaded.

1. The start address of the programming destination on the user MAT must be set in general register ER1. This parameter is called the flash multipurpose address area parameter (FMPAR).

   Since the program data is always in 128-byte units, the lower eight bits (A7 to A0) must be H'00 or H'80 as the boundary of the programming start address on the user MAT.
2. The program data for the user MAT must be prepared in a consecutive area. The program data must be in the consecutive space that can be accessed by using the MOV.B instruction of the CPU and in an address space other than flash memory.

   When data to be programmed does not satisfy 128 bytes, 128-byte program data must be prepared by filling in the dummy code H'FF.

   The start address of the area in which the prepared program data is stored must be set in general register ER0. This parameter is called the flash multipurpose data destination area parameter (FMPDR).

For details on the programming procedure, see section 19.4.2, User Program Mode.

(a) Flash multipurpose address area parameter (FMPAR: general register ER1 of CPU)

This parameter stores the start address of the programming destination on the user MAT.

When the address in an area other than the flash memory space is set, an error occurs.

The start address of the programming destination must be at the 128-byte boundary. If this boundary condition is not satisfied, an error occurs. The error occurrence is indicated by the WA bit (bit 1) in the FPFR parameter.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOA31 to MOA0	—	R/W	These bits store the start address of the programming destination on the user MAT. Consecutive 128-byte programming is executed starting from the specified start address of the user MAT. Therefore, the specified programming start address becomes a 128-byte boundary and the MOA6 to MOA0 bits are always 0.

(b) Flash multipurpose data destination area parameter (FMPDR: general register ER0 of CPU)

This parameter stores the start address of the area which stores the data to be programmed in the user MAT. When the storage destination of the program data is in flash memory, an error occurs. The error occurrence is indicated by the WD bit in the FPFR parameter.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOD31 to MOD0	—	R/W	These bits store the start address of the area which stores the program data for the user MAT. Consecutive 128-byte data is programmed to the user MAT starting from the specified start address.

(c) Flash pass/fail result parameter (FPFR: general register R0L of CPU)

This parameter indicates the return value of the programming processing result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused The return value is 0.
6	MD	—	R/W	Programming Mode Related Setting Error Detect Returns the check result of the error-protection state not being entered. When a low-level signal is input to the FWE pin or the error-protection state is entered, 1 is written to this bit. These states can be confirmed with the FWE and FLER bits in FCCS. For conditions to enter the error-protection state, see section 19.5.3, Error Protection. 0: FWE and FLER settings are normal (FWE = 1, FLER = 0) 1: Programming cannot be performed because FWE = 0 or FLER = 1
5	EE	—	R/W	Programming Execution Error Detect 1 is returned to this bit when the specified data could not be written because the user MAT was not erased. If this bit is set to 1, there is a high possibility that the user MAT is partially rewritten. In this case, after removing the error factor, erase the user MAT. If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when programming is performed. In this case, both the user MAT and user boot MAT are not rewritten. Programming of the user boot MAT should be performed in boot mode or programmer mode. 0: Programming has ended normally 1: Programming has ended abnormally and programming result is not guaranteed

Bit	Bit Name	Initial Value	R/W	Description
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of the FKEY value before the start of the programming processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is abnormal (FKEY = value other than H'5A)
3	—	—	—	Unused The return value is 0.
2	WD	—	R/W	Write Data Address Detect When an address in the flash memory area is specified as the start address of the storage destination of the program data, an error occurs. 0: Setting of program data address is normal 1: Setting of program data address is abnormal
1	WA	—	R/W	Write Address Error Detect When the following items are specified as the start address of the programming destination, an error occurs. • When the specified programming destination address is in an area other than flash memory • When the specified address is not at a 128-byte boundary (the lower eight bits of the address are other than H'00 or H'80) 0: Setting of programming destination address is normal 1: Setting of programming destination address is abnormal
0	SF	—	R/W	Success/Fail Indicates whether the programming processing has ended normally or not. 0: Programming has ended normally (no error) 1: Programming has ended abnormally (error occurred)

(4) Erasure Execution

When flash memory is erased, the erase-block number on the user MAT must be passed to the erasing program that is downloaded. This is set to the FEBS parameter (general register ER0).

One block is specified from the block numbers 0 to 23.

For details on the erasing procedure, see section 19.4.2, User Program Mode.

(a) Flash erase block select parameter (FEBS: general register ER0 of CPU)

This parameter specifies the erase-block number. Several block numbers cannot be selected at one time.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	—	Unused These bits should be cleared to 0.
7	EB7	—	R/W	Erase Block
6	EB6	—	R/W	These bits set the erase-block number in the range from 0 to 15. 0 corresponds to the EB0 block and 15 corresponds to the EB15 block. An error occurs when a number other than 0 to 15 (H'00 to H'0F) is set.
5	EB5	—	R/W	
4	EB4	—	R/W	
3	EB3	—	R/W	
2	EB2	—	R/W	
1	EB1	—	R/W	
0	EB0	—	R/W	

(b) Flash pass/fail result parameter (FPFR: general register R0L of CPU)

This parameter indicates the return value of the erasing processing result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused The return value is 0.
6	MD	—	R/W	Erasing Mode Related Setting Error Detect Returns the check result of the error-protection state not being entered. When the error-protection state is entered, 1 is written to this bit. This state can be confirmed with the FLER bit in FCCS. For conditions to enter the error-protection state, see section 19.5.3, Error Protection. 0: FLER setting is normal (FLER = 0) 1: Erasing cannot be performed because FLER = 1
5	EE	—	R/W	Erasure Execution Error Detect 1 is returned to this bit when the user MAT could not be erased or when flash-memory related register settings are partially changed. If this bit is set to 1, there is a high possibility that the user MAT is partially erased. In this case, after removing the error factor, erase the user MAT. If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when erasure is performed. In this case, both the user MAT and user boot MAT are not erased. Erasing of the user boot MAT should be performed in boot mode or programmer mode. 0: Erasure has ended normally 1: Erasure has ended abnormally and erasure result is not guaranteed
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of the FKEY value before the start of the erasing processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is abnormal (FKEY = value other than H'5A)

Bit	Bit Name	Initial Value	R/W	Description
3	EB	—	R/W	Erase Block Select Error Detect Returns the check result whether the specified erase-block number is in the block range of the user MAT. 0: Setting of erase-block number is normal 1: Setting of erase-block number is abnormal
2, 1	—	—	—	Unused The return value is 0.
0	SF	—	R/W	Success/Fail Indicates whether the erasing processing has ended normally or not. 0: Erasure has ended normally (no error) 1: Erasure has ended abnormally (error occurred)

19.4 On-Board Programming

When the pins are set to on-board programming mode and the reset start is executed, a transition is made to an on-board programming state in which the on-chip flash memory can be programmed/erased. On-board programming mode has three operating modes: boot mode, user program mode, and user boot mode.

For details on the pin setting for entering each mode, see table 19.5. For details of the state transition of each mode for flash memory, see figure 19.2.

Table 19.5 On-Board Programming Mode Setting

Mode Setting	MD2*[2]	MD1	MD0	NMI	P42	P41	P40
Boot mode	0	0	0	1	1	1	1
User program mode*[1]	0	1	0/1	0/1	—	—	—
User boot mode	0	0	0	0	1	1	1

Notes: 1. Before downloading a programming/erasing program, the FLSHE bit must be set to 1 to make a transition to user program mode.
2. MD2 is not supported in SDIP-64 and QFP-64.

19.4.1 Boot Mode

Boot mode executes programming/erasing of the user MAT and user boot MAT by means of the control commands and program data transmitted from the host via the on-chip SCI. The tool for transmitting the control commands, and program data must be prepared in the host. The SCI communication mode is set to asynchronous mode. When reset start is executed after this LSI's pins have been set to boot mode, the boot program built in the microcomputer beforehand is initiated. After the SCI bit rate is automatically adjusted, communication with the host is executed by means of control commands.

A system configuration diagram in boot mode is shown in figure 19.6. For details on the pin settings in boot mode, see table 19.5. The NMI and other interrupts are ignored in boot mode. However, the NMI and other interrupts should be disabled within the user system.

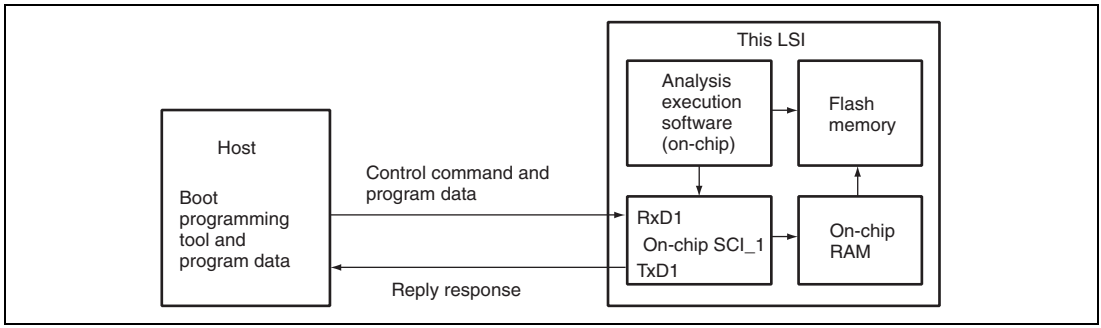


Figure 19.6 System Configuration in Boot Mode

(1) SCI Interface Setting by Host

When boot mode is initiated, this LSI measures the low period of asynchronous SCI communication data (H'00) which is transmitted consecutively from the host. The SCI transmit/receive format is set to 8-bit data, 1 stop bit, and no parity. This LSI calculates the bit rate of transmission by the host by means of the measured low period and transmits the bit adjustment end sign (1 byte of H'00) to the host. The host must confirm that this bit adjustment end sign (H'00) has been received normally and then transmits 1 byte of H'55 to this LSI. When reception has not been executed normally, boot mode is initiated again (reset) and the operation described above must be performed. The bit rates of the host and this LSI do not match due to the bit rate of transmission by the host and the system clock frequency of this LSI. To operate the SCI normally, the transfer bit rate of the host must be set to 4,800 bps, 9,600 bps, or 19,200 bps.

The system clock frequency, which can automatically adjust the transfer bit rate of the host and the bit rate of this LSI, is shown in table 19.6. Boot mode must be initiated in the range of this system clock.

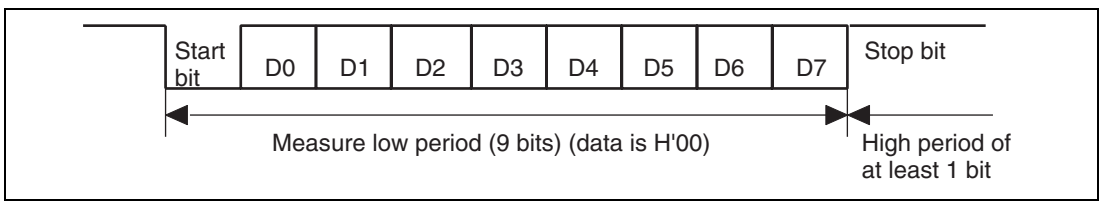


Figure 19.7 Automatic-Bit-Rate Adjustment Operation of SCI

Table 19.6 System Clock Frequency for Automatic-Bit-Rate Adjustment by This LSI

Bit Rate of Host	System Clock Frequency for Automatic-Bit-Rate Adjustment by This LSI
4,800 bps	8 to 20 MHz
9,600 bps	8 to 20 MHz
19,200 bps	8 to 20 MHz

(2) State Transition Diagram

The overview of the state transition diagram after boot mode is initiated is shown in figure 19.8.

1. Bit rate adjustment
   After boot mode is initiated, the bit rate of the SCI interface is adjusted with that of the host.
2. Waiting for inquiry set command
   For inquiries about the user MAT size and configuration, MAT start address, and support state, the required information is transmitted to the host.
3. Automatic erasure of all user MATs and user boot MATs
   After inquiries have finished, all user MATs and user boot MATs are automatically erased.
4. Waiting for programming/erasing command
   - When the program preparation notice is received, the state for waiting for program data is entered. The programming start address and program data must be transmitted following the programming command. When programming is finished, the programming start address must be set to H'FFFFFFFF and transmitted. Then the state of program data wait is returned to the state of programming/erasing command wait.
   - When the erasure preparation notice is received, the state for waiting for erase-block data is entered. The erase-block number must be transmitted following the erasing command. When the erasure is finished, the erase-block number must be set to H'FF and transmitted. Then the state of erase-block data wait is returned to the state of programming/erasing command wait. This erasing operation should be used in a case where after programming has been executed in boot mode, a specific block is to be reprogrammed without a reset start. When programming can be executed by only one operation, since all blocks are erased before entering the state for waiting for a programming/erasing/other command, the erasing operation is not required.
   - There are many commands other than programming/erasing. For example, sum check, blank check (erasure check), and memory read of the user MAT and user boot MAT, and acquisition of current status information.

Note that memory read of the user MAT or user boot MAT can only read out the programmed data after all user MATs or user boot MATs have been automatically erased.

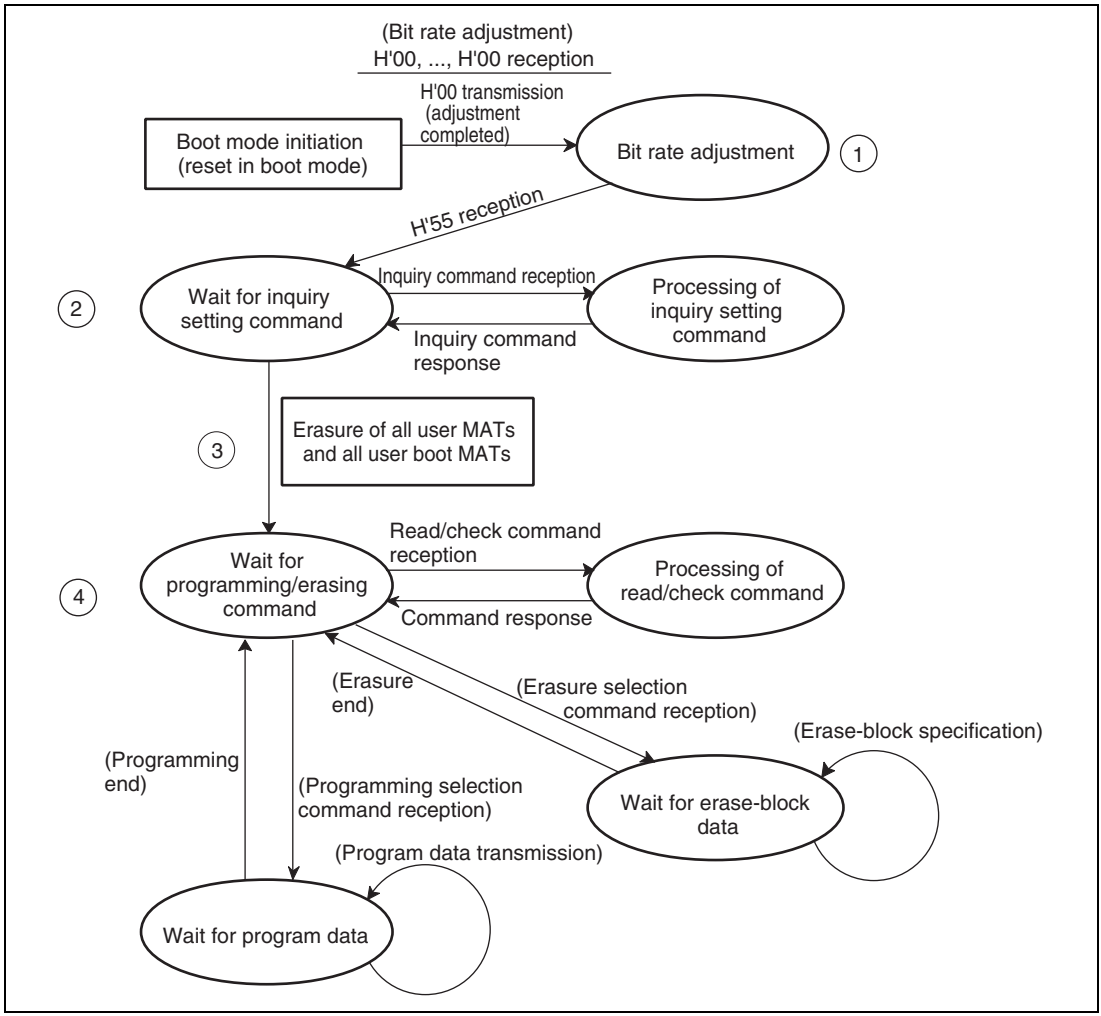


Figure 19.8 Overview of Boot Mode State Transition Diagram

19.4.2 User Program Mode

The user MAT can be programmed/erased in user program mode. (The user boot MAT cannot be programmed/erased.)

Programming/erasing is executed by downloading the program built in the microcomputer beforehand.

The programming/erasing overview flow is shown in figure 19.9.

High voltage is applied to internal flash memory during the programming/erasing processing. Therefore, a transition to the reset state or hardware standby mode must not be made. Doing so may damage and destroy flash memory. If a reset is executed accidentally, the reset must be released after a reset input period of 100 µs which is longer than normal.

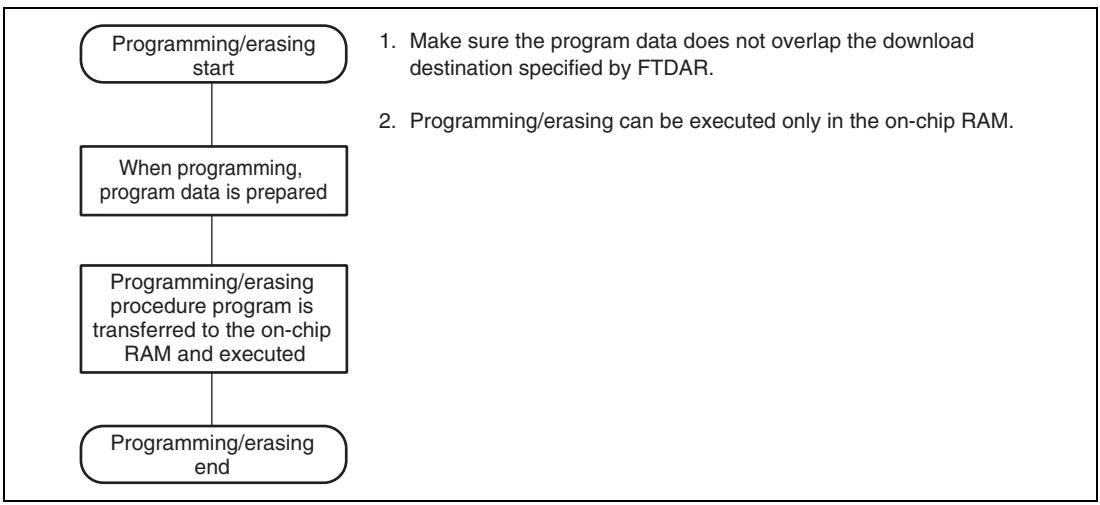


Figure 19.9 Programming/Erasing Overview Flow

(1) On-Chip RAM Address Map when Programming/Erasing is Executed

Part of the procedure program that is made by the user, like the download request, programming/erasing procedure, and determination of the result, must be executed in the on-chip RAM. The on-chip program that is to be downloaded is all in the on-chip RAM. Note that areas in the on-chip RAM must be controlled so that these parts do not overlap.

Figure 19.10 shows the area where a program is downloaded.

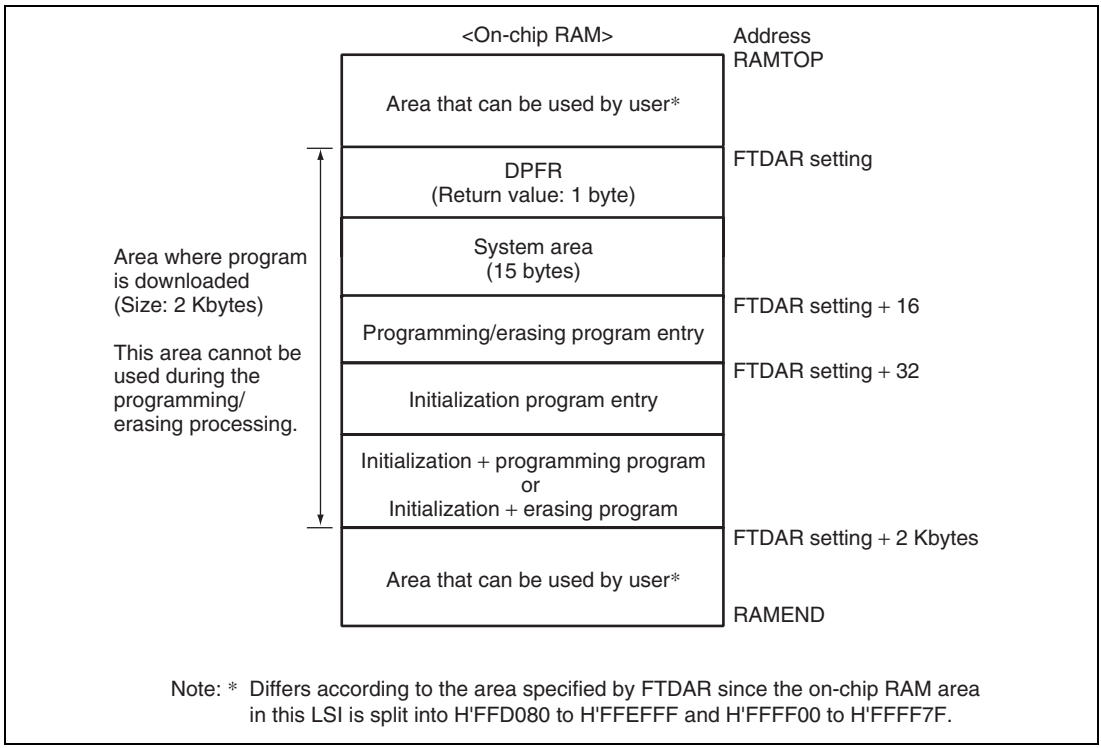


Figure 19.10 RAM Map when Programming/Erasing is Executed

(2) Programming Procedure in User Program Mode

The procedures for download, initialization, and programming are shown in figure 19.11.

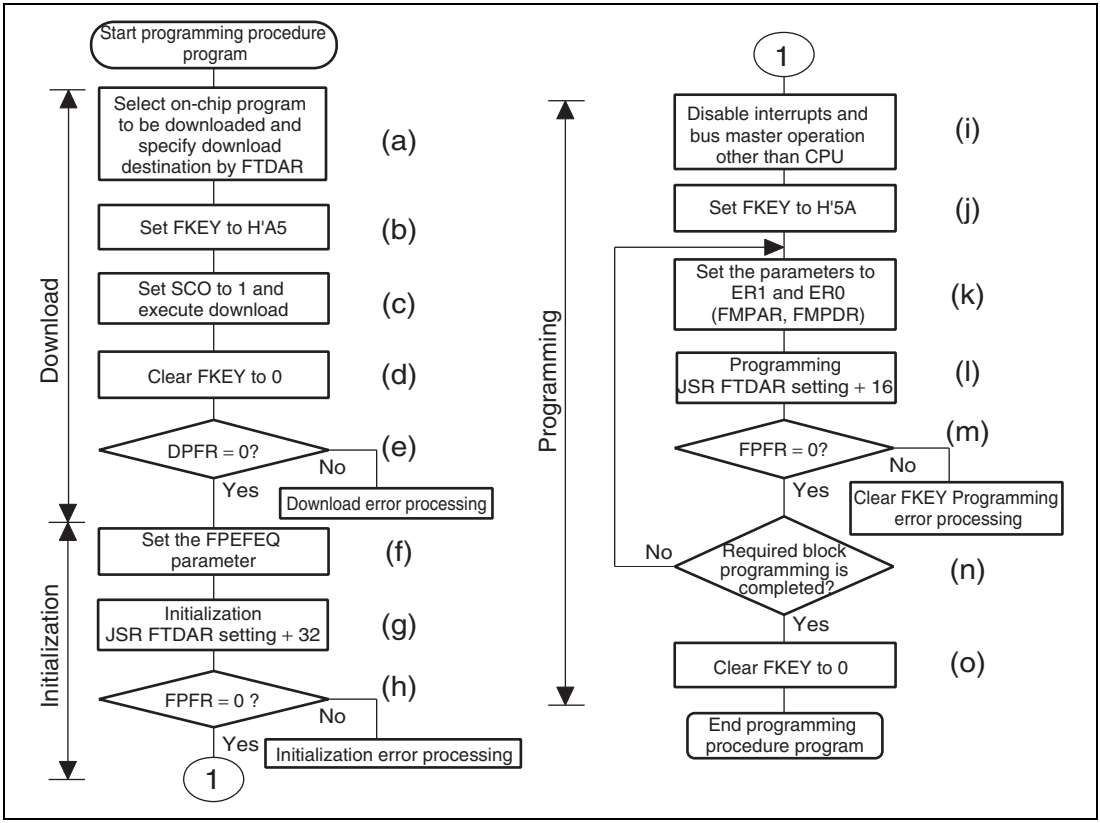


Figure 19.11 Programming Procedure

The procedure program must be executed in an area other than the flash memory to be programmed. Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in the on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM and user MAT) is shown in section 19.4.4, Storable Areas for Procedure Program and Program Data.

The following description assumes the area to be programmed on the user MAT is erased and program data is prepared in the consecutive area. When erasing has not been done yet, execute erasing before writing.

128-byte programming is performed in one programming processing. To program more than 128 bytes, update the programming destination address/program data parameter in 128-byte units and repeat programming.

When less than 128 bytes of programming is performed, the program data must amount to 128 bytes by filling in invalid data. If the invalid data to be added is H'FF, the programming processing time can be shortened.

(a) Select the on-chip program to be downloaded and specify a download destination

When the PPVS bit in FPCS is set to 1, the programming program is selected.

Several programming/erasing programs cannot be selected at one time. If several programs are set, download is not performed and a download error is returned to the SS bit in DPFR. The start address of the download destination is specified by FTDAR.

(b) Write H'A5 in FKEY

If H'A5 is not written to FKEY for protection, 1 cannot be set to the SCO bit for a download request.

(c) Set the SCO bit in FCCS to 1 to execute download.

To set 1 to the SCO bit, the following conditions must be satisfied.

- H'A5 is written to FKEY.
- The SCO bit writing is executed in the on-chip RAM.

When the SCO bit is set to 1, download is started automatically. When execution returns to the user procedure program, the SCO bit is already cleared to 0. Therefore, the SCO bit cannot be confirmed to be 1 in the user procedure program.

The download result can be confirmed only by the return value of DPFR. To prevent incorrect determination, before the SCO bit is set to 1, set the single byte of the on-chip RAM start address (to be used as the DPFR parameter) specified by FTDAR to a value (e.g. H'FF) other than the return value.

When download is executed, particular interrupt processing, which is accompanied by bank switchover as described below, is performed as a microcomputer internal processing. Execute four NOP instructions immediately after the instruction that sets the SCO bit to 1.

- The user MAT space is switched to the embedded program storage MAT.
- After the selection condition of the download program and the FTDAR address setting are checked, the transfer processing to the on-chip RAM specified by FTDAR is executed.
- The SCO bit in FPCS, FECS, and FCCS is cleared to 0.
- The return value is set to the DPFR parameter.
- After the embedded program storage MAT is returned to the user MAT space, execution returns to the user procedure program.
- In the download processing, the values of CPU general registers are retained.
- In the download processing, all interrupts are not accepted. However, interrupt requests except for NMI are held. Therefore, when execution returns to the user procedure program, the interrupts will occur.
- When the level-detection interrupt requests are to be held, interrupts must be input until the download is ended.
- When hardware standby mode is entered during the download processing, normal download to the on-chip RAM cannot be guaranteed. Therefore, download must be executed again.
- Since a stack area of 128 bytes at the maximum is used, the stack area must be allocated before setting the SCO bit to 1.
- If a flash memory access by the DTC is requested during downloading, the operation cannot be guaranteed. Therefore, access by the DTC must not occur.

(d) Clear FKEY to H'00 for protection.

(e) Check the value of the DPFR parameter to confirm the download result.

- Check the value of the DPFR parameter (single byte of start address of the download destination specified by FTDAR). If the value is H'00, download has been performed normally. If the value is not H'00, the source that caused download to fail can be investigated by the description below.
- If the value of the DPFR parameter is the same as before downloading (e.g. H'FF), the address setting of the download destination in FTDAR may be abnormal. In this case, confirm the setting of the TDER bit in FTDAR.
- If the value of the DPFR parameter is different from before downloading, check the SS bit and FK bit in the DPFR parameter to ensure that the download program selection and FKEY setting were normal, respectively.

(f) Set the operating frequency to the FPEFEQ parameter for initialization.

The current frequency of the CPU clock is set to the FPEFEQ parameter (general register ER0).

The settable range of the FPEFEQ parameter is 8 to 20 MHz. When the frequency is set out of this range, an error is returned to the FPFR parameter of the initialization program and initialization is not performed. For details on the frequency setting, see the description in section 19.3.2 (2) (a), Flash programming/erasing frequency control parameter (FPEFEQ: general register ER0 of CPU).

(g) Initialization

When a programming program is downloaded, the initialization program is also downloaded to the on-chip RAM. There is an entry point for the initialization program in the area from the start address of a download destination specified by FTDAR + 32 bytes. The subroutine is called and initialization is executed by using the following steps.

```
MOV.L   #DLTOP+32,ER2          ; Set entry address to ER2
JSR     @ER2                   ; Call initialization routine
NOP
```

- The general registers other than R0L are saved in the initialization program.
- R0L is a return value of the FPFR parameter.
- Since the stack area is used in the initialization program, a 128-byte stack area at the maximum must be allocated in RAM.
- Interrupts can be accepted during the execution of the initialization program. Note however that the program storage area and stack area in the on-chip RAM, and register values must not be rewritten.

(h) The return value in the initialization program, FPFR (general register R0L) is determined.

(i) All interrupts and the use of a bus master other than the CPU are prohibited.

The stipulated voltage is applied for the stipulated time when programming or erasing. If interrupts occur or a bus master other than the CPU gets the bus during this period, a voltage pulse exceeding the specification may be applied, thus damaging flash memory. Accordingly, interrupts must be disabled and a bus master other than the CPU, such as the DTC, must not be allowed.

To disable interrupts, bit 7 (I) in the condition code register (CCR) of the CPU should be set to B'1 in interrupt control mode 0, or bits 7 and 6 (I and UI) in the condition code register (CCR) of the CPU should be set to B'11 in interrupt control mode 1. This enables interrupts other than NMI to be held and not executed.

The NMI interrupt must be masked within the user system.

The interrupts that are held must be executed after all programming processings.

When a bus master other than the CPU, such as the DTC, acquires the bus, the error-protection state is entered. Therefore, acquisition of the bus by the DTC must also be prohibited.

(j) Set H'5A in FKEY and prepare the user MAT for programming.

(k) Set the parameters required for programming.

The start address of the programming destination of the user MAT (FMPAR) is set to general register ER1, and the start address of the program data area (FMPDR) is set to general register ER0.

- Example of FMPAR setting

  FMPAR specifies the programming destination address. When an address other than one in the user MAT area is specified, even if the programming program is executed, programming is not executed and an error is returned to the return value parameter FPFR. Since the programming unit is 128 bytes, the lower eight bits of the address must be at the 128-byte boundary of H'00 or H'80.
- Example of FMPDR setting

  When the storage destination of the program data is flash memory, even if the programming execution routine is executed, programming is not executed and an error is returned to the FPFR parameter. In this case, the program data must be transferred to the on-chip RAM before programming is executed.

(l) Programming

There is an entry point for the programming program in the area from the start address of a download destination specified by FTDAR + 16 bytes. The subroutine is called and programming is executed by using the following steps.

```
MOV.L   #DLTOP+16,ER2       ; Set entry address to ER2
JSR     @ER2                ; Call programming routine
NOP
```

- The general registers other than R0L are saved in the programming program.
- R0L is a return value of the FPFR parameter.
- Since the stack area is used in the programming program, a 128-byte stack area at the maximum must be allocated in RAM.

(m) The return value in the programming program, FPFR (general register R0L) is determined.

(n) Determine whether programming of the necessary data has finished.

If more than 128 bytes of data are to be programmed, specify FMPAR and FMPDR in 128-byte units, and repeat steps (l) to (n). Increment the programming destination address by 128 bytes and update the programming data pointer correctly. If an address that has already been programmed is written to again, not only will a programming error occur, but also flash memory will be damaged.

(o) After programming finishes, clear FKEY and specify software protection.

If this LSI is restarted by a reset immediately after user MAT programming has finished, secure a reset period (period of $\overline{RES}$ = 0) of 100 µs which is longer than normal.

(3) Erasing Procedure in User Program Mode

The procedures for download, initialization, and erasing are shown in figure 19.12.

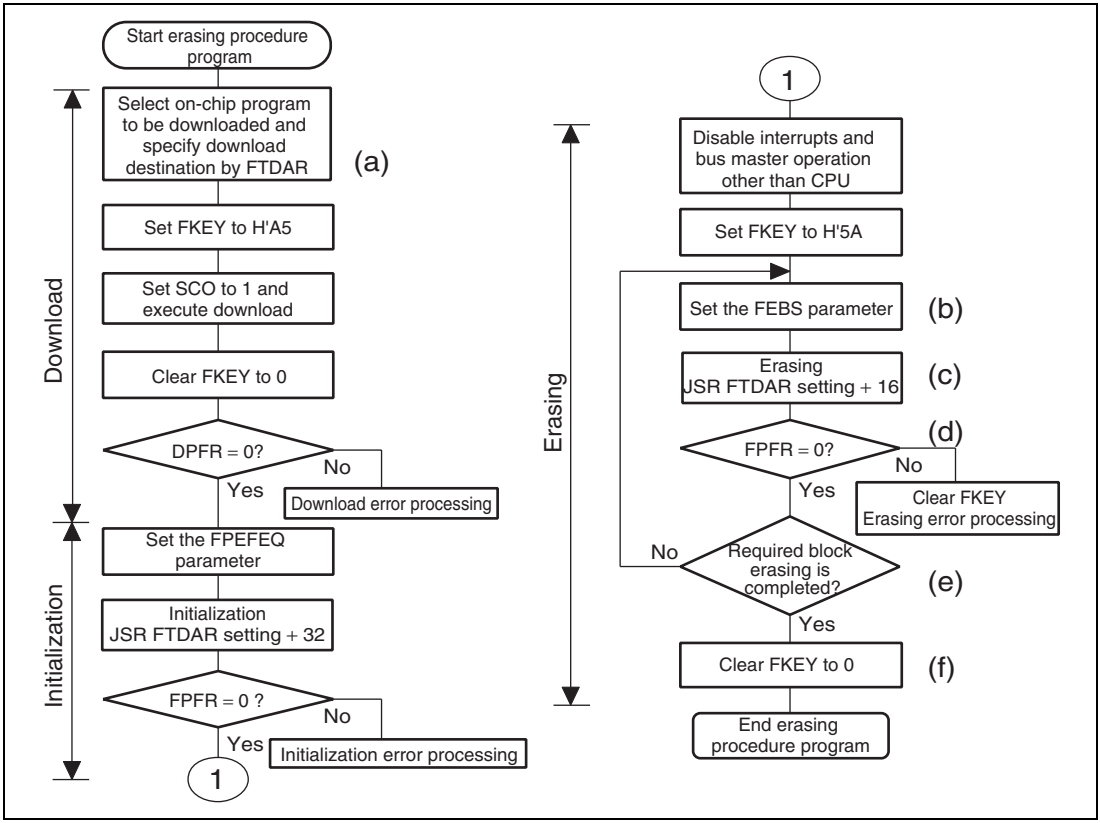


Figure 19.12 Erasing Procedure

The procedure program must be executed in an area other than the user MAT to be erased. Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in the on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM and user MAT) is shown in section 19.4.4, Storable Areas for Procedure Program and Program Data.

For the downloaded on-chip program area, see the RAM map for programming/erasing in figure 19.10.

A single divided block is erased by one erasing processing. For block divisions, refer to figure 19.4. To erase two or more blocks, update the erase-block number and perform the erasing processing for each block.

(a) Select the on-chip program to be downloaded

Set the EPVB bit in FECS to 1.

Several programming/erasing programs cannot be selected at one time. If several programs are set, download is not performed and a download error is reported to the SS bit in the DPFR parameter.

Specify the start address of the download destination by FTDAR.

The procedures to be carried out after setting FKEY, e.g. download and initialization, are the same as those in the programming procedure. For details, see section 19.4.2 (2), Programming Procedure in User Program Mode.

The procedures after setting parameters for erasing programs are as follows:

(b) Set the FEBS parameter necessary for erasure

Set the erase-block number of the user MAT in the flash erase block select parameter FEBS (general register ER0). If a value other than an erase-block number of the user MAT is set, no block is erased even though the erasing program is executed, and an error is returned to the return value parameter FPFR.

(c) Erasure

Similar to as in programming, there is an entry point for the erasing program in the area from the start address of a download destination specified by FTDAR + 16 bytes. The subroutine is called and erasing is executed by using the following steps.

```
MOV.L   #DLTOP+16,ER2       ; Set entry address to ER2
JSR     @ER2                ; Call erasing routine
NOP
```

- The general registers other than R0L are saved in the erasing program.
- R0L is a return value of the FPFR parameter.
- Since the stack area is used in the erasing program, a 128-byte stack area at the maximum must be allocated in RAM.

(d) The return value in the erasing program, FPFR (general register R0L) is determined.

(e) Determine whether erasure of the necessary blocks has completed.

If more than one block is to be erased, update the FEBS parameter and repeat steps (b) to (e). Blocks that have already been erased can be erased again.

(f) After erasure completes, clear FKEY and specify software protection.

If this LSI is restarted by a reset immediately after user MAT erasure has completed, secure a reset period (period of $\overline{RES}$ = 0) of 100 µs which is longer than normal.

(4) Erasing and Programming Procedure in User Program Mode

By changing the on-chip RAM address of the download destination in FTDAR, the erasing program and programming program can be downloaded to separate on-chip RAM areas.

Figure 19.13 shows a repeating procedure of erasing and programming.

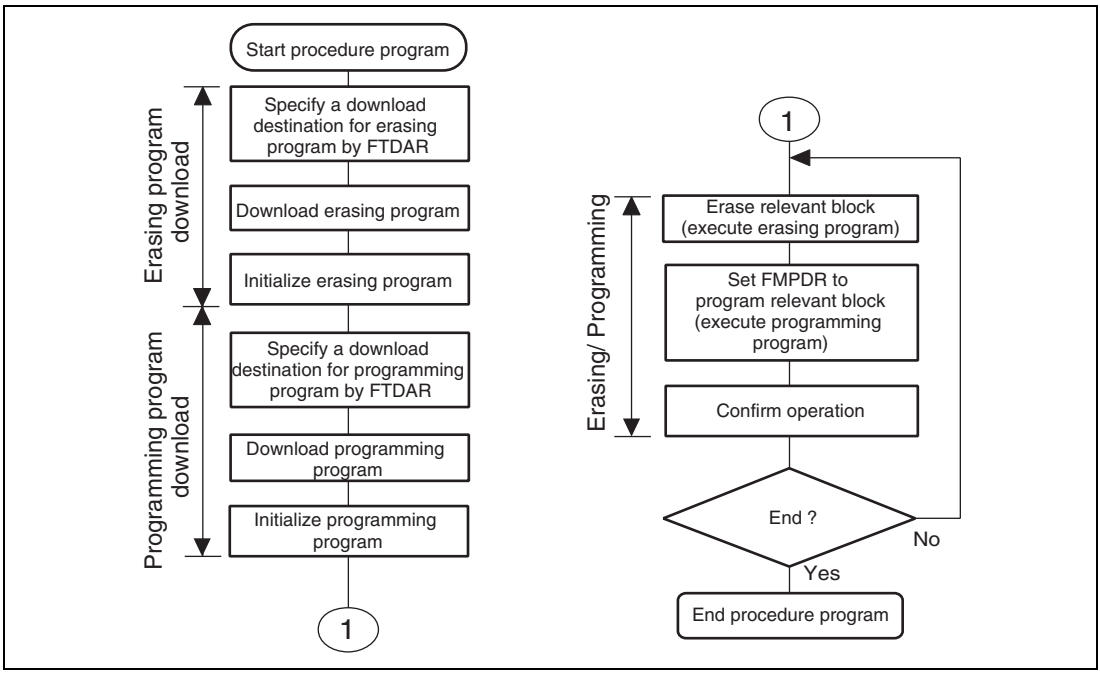


Figure 19.13 Repeating Procedure of Erasing and Programming

In the above procedure, download and initialization are performed only once at the beginning.

In this kind of operation, note the following:

- Be careful not to damage on-chip RAM with overlapped settings.
  In addition to the erasing program area and programming program area, areas for the user procedure programs, work area, and stack area are allocated in the on-chip RAM. Do not make settings that will overwrite data in these areas.
- Be sure to initialize both the erasing program and programming program.
  Initialization by setting the FPEFEQ parameter must be performed for both the erasing program and programming program. Initialization must be executed for both entry addresses: (download start address for erasing program) + 32 bytes and (download start address for programming program) + 32 bytes.

19.4.3 User Boot Mode

This LSI has user boot mode that is initiated with different mode pin settings than those in boot mode or user program mode. User boot mode is a user-arbitrary boot mode, unlike boot mode that uses the on-chip SCI.

Only the user MAT can be programmed/erased in user boot mode. Programming/erasing of the user boot MAT is only enabled in boot mode or programmer mode.

(1) User Boot Mode Initiation

For the mode pin settings to start up user boot mode, see table 19.5.

When the reset start is executed in user boot mode, the built-in check routine runs. The user MAT and user boot MAT states are checked by this check routine.

While the check routine is running, NMI and all other interrupts cannot be accepted.

Next, processing starts from the execution start address of the reset vector in the user boot MAT. At this point, H'AA is set to FMATS because the execution target MAT is the user boot MAT.

(2) User MAT Programming in User Boot Mode

For programming the user MAT in user boot mode, additional processing made by setting FMATS is required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after programming completes.

Figure 19.14 shows the procedure for programming the user MAT in user boot mode.

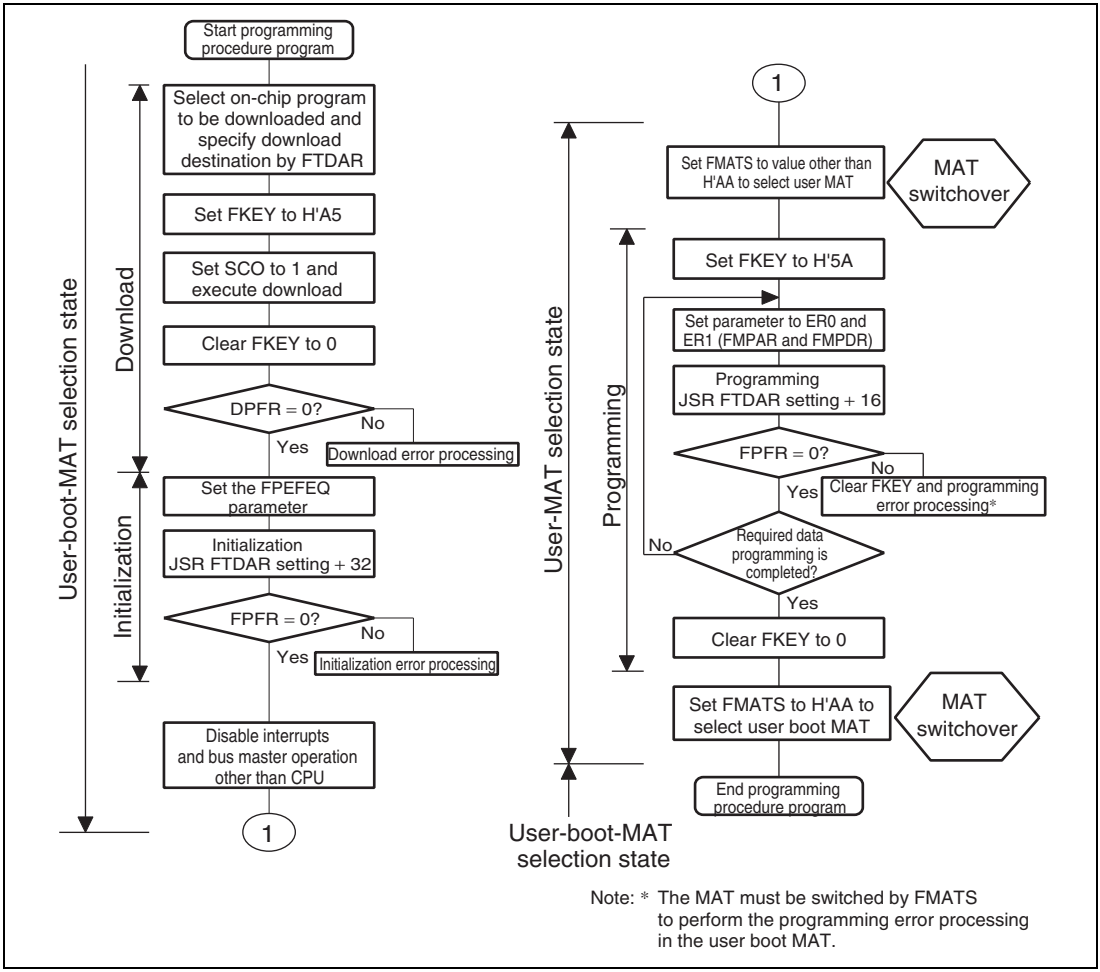


Figure 19.14 Procedure for Programming User MAT in User Boot Mode

The difference between the programming procedures in user program mode and user boot mode is whether the MAT is switched or not as shown in figure 19.14.

In user boot mode, the user boot MAT can be seen in the flash memory space with the user MAT hidden in the background. The user MAT and user boot MAT are switched only while the user MAT is being programmed. Because the user boot MAT is hidden while the user MAT is being programmed, the procedure program must be executed in an area other than flash memory. After the programming procedure completes, switch the MATs again to return to the first state.

MAT switching is enabled by writing a specific value to FMATS. Note however that while the MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allowed until MAT switching is completed, and if an interrupt occurs, from which MAT the interrupt vector is read is undetermined. Perform MAT switching in accordance with the description in section 19.6, Switching between User MAT and User Boot MAT.

Except for MAT switching, the programming procedure is the same as that in user program mode.

The area that can be executed in the steps of the user procedure program (on-chip RAM and user MAT) is shown in section 19.4.4, Storable Areas for Procedure Program and Program Data.

(3) User MAT Erasing in User Boot Mode

For erasing the user MAT in user boot mode, additional processing made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after erasing completes.

Figure 19.15 shows the procedure for erasing the user MAT in user boot mode.

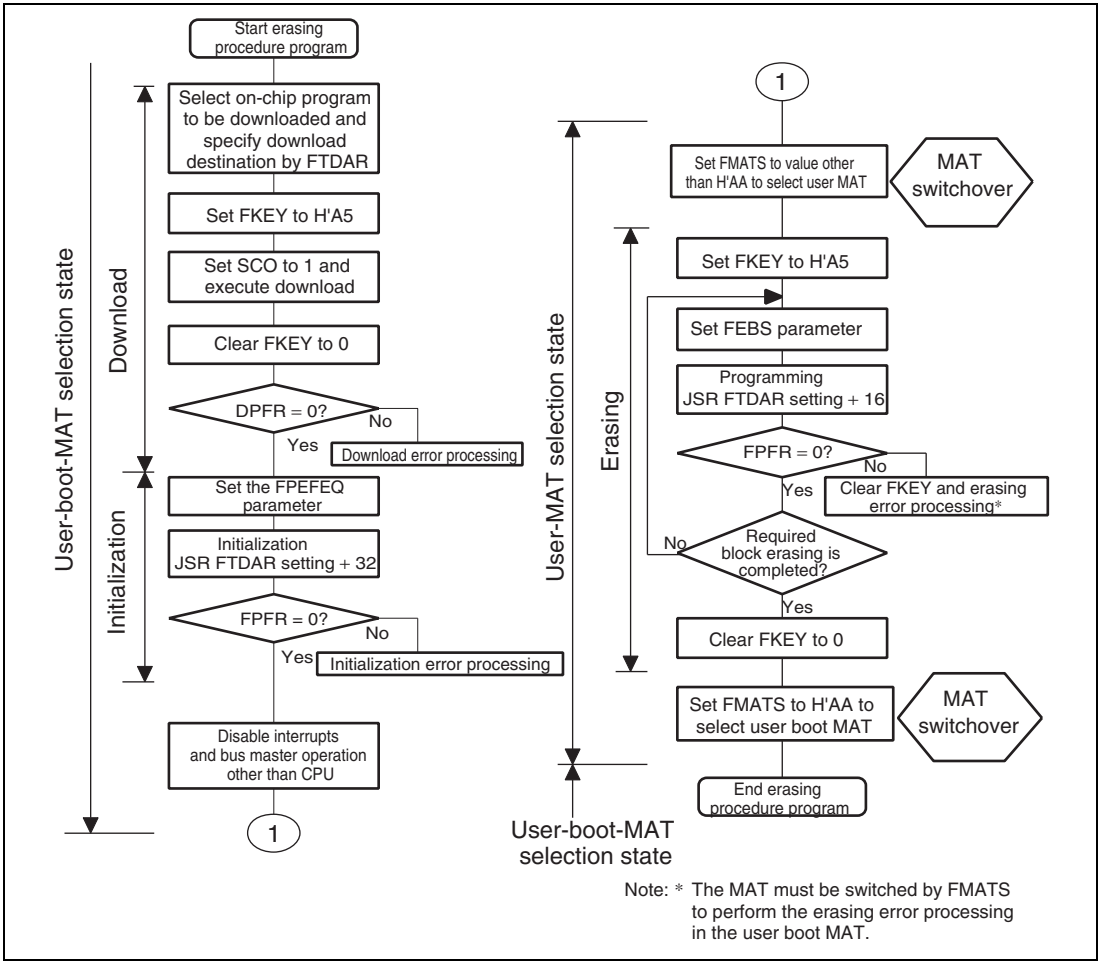


Figure 19.15 Procedure for Erasing User MAT in User Boot Mode

The difference between the erasing procedures in user program mode and user boot mode depends on whether the MAT is switched or not as shown in figure 19.15.

MAT switching is enabled by writing a specific value to FMATS. Note however that while the MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allowed until MAT switching is completed, and if an interrupt occurs, from which MAT the interrupt vector is read is undetermined. Perform MAT switching in accordance with the description in section 19.6, Switching between User MAT and User Boot MAT.

Except for MAT switching, the erasing procedure is the same as that in user program mode.

The area that can be executed in the steps of the user procedure program (on-chip RAM and user MAT) is shown in section 19.4.4, Storable Areas for Procedure Program and Program Data.

19.4.4 Storable Areas for Procedure Program and Program Data

In the descriptions in the previous section, the storable areas for the programming/erasing procedure programs and program data are assumed to be in the on-chip RAM. However, the procedure programs and program data can be stored in and executed from other areas, such as part of flash memory which is not to be programmed or erased.

(1) Conditions that Apply to Programming/Erasing

1. The on-chip programming/erasing program is downloaded from the address in the on-chip RAM specified by FTDAR, therefore, this area is not available for use.
2. The on-chip programming/erasing program will use 128 bytes at the maximum as a stack. So, make sure that this area is allocated.
3. Download by setting the SCO bit to 1 will lead to switching of the MATs. Therefore, if this operation is used, it should be executed from the on-chip RAM.
4. The flash memory is accessible until the start of programming or erasing, that is, until the result of downloading has been determined. The required procedure programs, NMI handling vector, and NMI handling routine should be transferred to the on-chip RAM before programming/erasing of the flash memory starts.
5. Since flash memory is not accessible during programming/erasing processing, programs downloaded to the on-chip RAM are executed. The procedure programs that initiate programming/erasing processing, and execution areas for the NMI interrupt vector table and NMI interrupt handling program must be stored in on-chip RAM.

6. After programming/erasing, access to the flash memory is prohibited until FKEY is cleared.

   In case the LSI mode is changed to generate a reset on completion of a programming/erasing operation, a reset state ($\overline{RES}$ = 0) of 100 µs or more must be secured.

   Transitions to the reset state or hardware standby mode are prohibited during programming/erasing operations. However, when the reset signal is accidentally input to the chip, the reset must be released after a reset period of 100 µs that is longer than normal.
7. Switching of the MATs by FMATS should be required when programming/erasing of the user MAT is operated in user boot mode. The program that switches the MATs should be executed from the on-chip RAM. (For details, see section 19.6, Switching between User MAT and User Boot MAT.) Make sure you know which MAT is currently selected when switching them.
8. When the program data storable area indicated by the programming parameter FMPDR is in flash memory, an error will occur even when the program data stored is normal. Therefore, the program data should be temporarily transferred to the on-chip RAM to set an address other than flash memory in FMPDR.

In consideration of these conditions, the following tables show areas where program data can be stored and executed for different combinations of operating mode, user MAT bank configuration, and processing type.

Table 19.7 Executable MAT

	Initiated Mode	
Processing	**User Program Mode**	**User Boot Mode***
Programming	Table 19.8 (1)	Table 19.8 (3)
Erasing	Table 19.8 (2)	Table 19.8 (4)

Note: * Programming/Erasing is possible to the user MAT.

Table 19.8 (1) Usable Area for Programming in User Program Mode

	Storable/Executable Area		Selected MAT	
Item	**On-chip RAM**	**User MAT**	**User MAT**	**Embedded Program Storage MAT**
Storage area for program data	O	×*	—	—
Selecting on-chip program to be downloaded	O	O	O	
Writing H'A5 to FKEY	O	O	O	
Writing 1 to SCO in FCCS (download)	O	×		O
FKEY clearing	O	O	O	
Determination of download result	O	O	O	
Download error processing	O	O	O	
Setting initialization parameter	O	O	O	
Initialization	O	×	O	
Determination of initialization result	O	O	O	
Initialization error processing	O	O	O	
NMI handling routine	O	×	O	
Disabling interrupts	O	O	O	
Writing H'5A to FKEY	O	O	O	
Setting programming parameter	O	×	O	

	Storable/Executable Area		Selected MAT	
Item	**On-chip RAM**	**User MAT**	**User MAT**	**Embedded Program Storage MAT**
Programming	O	×	O	
Determination of programming result	O	×	O	
Programming error processing	O	×	O	
FKEY clearing	O	×	O	

Note: * Transferring the data to the on-chip RAM in advance enables this area to be used.

Table 19.8 (2) Usable Area for Erasure in User Program Mode

	Storable/Executable Area		Selected MAT	
Item	**On-chip RAM**	**User MAT**	**User MAT**	**Embedded Program Storage MAT**
Selecting on-chip program to be downloaded	O	O	O	
Writing H'A5 to FKEY	O	O	O	
Writing 1 to SCO in FCCS (download)	O	×		O
FKEY clearing	O	O	O	
Determination of download result	O	O	O	
Download error processing	O	O	O	
Setting initialization parameter	O	O	O	
Initialization	O	×	O	
Determination of initialization result	O	O	O	
Initialization error processing	O	O	O	
NMI handling routine	O	×	O	
Disabling interrupts	O	O	O	
Writing H'5A to FKEY	O	O	O	
Setting erasure parameter	O	×	O	
Erasure	O	×	O	
Determination of erasure result	O	×	O	
Erasing error processing	O	×	O	
FKEY clearing	O	×	O	

Table 19.8 (3) Usable Area for Programming in User Boot Mode

	Storable/Executable Area			Selected MAT	
Item	On-chip RAM	User Boot MAT	User MAT	User Boot MAT	Embedded Program Storage MAT
Storage area for program data	O	×*1	—	—	—
Selecting on-chip program to be downloaded	O	O		O	
Writing H'A5 to FKEY	O	O		O	
Writing 1 to SCO in FCCS (download)	O	×			O
FKEY clearing	O	O		O	
Determination of download result	O	O		O	
Download error processing	O	O		O	
Setting initialization parameter	O	O		O	
Initialization	O	×		O	
Determination of initialization result	O	O		O	
Initialization error processing	O	O		O	
NMI handling routine	O	×		O	
Disabling interrupts	O	O		O	
Switching MATs by FMATS	O	×	O		
Writing H'5A to FKEY	O	×	O		

	Storable/Executable Area			Selected MAT	
Item	**On-chip RAM**	**User Boot MAT**	**User MAT**	**User Boot MAT**	**Embedded Program Storage MAT**
Setting programming parameter	O	×	O		
Programming	O	×	O		
Determination of programming result	O	×	O		
Programming error processing	O	×*2	O		
FKEY clearing	O	×	O		
Switching MATs by FMATS	O	×		O	

Notes: 1. Transferring the data to the on-chip RAM in advance enables this area to be used.
2. Switching FMATS by a program in the on-chip RAM enables this area to be used.

Table 19.8 (4) Usable Area for Erasure in User Boot Mode

	Storable/Executable Area			Selected MAT	
Item	**On-chip RAM**	**User Boot MAT**	**User MAT**	**User Boot MAT**	**Embedded Program Storage MAT**
Selecting on-chip program to be downloaded	O	O		O	
Writing H'A5 to FKEY	O	O		O	
Writing 1 to SCO in FCCS (download)	O	×			O
FKEY clearing	O	O		O	
Determination of download result	O	O		O	
Download error processing	O	O		O	
Setting initialization parameter	O	O		O	
Initialization	O	×		O	
Determination of initialization result	O	O		O	
Initialization error processing	O	O		O	
NMI handling routine	O	×		O	
Disabling interrupts	O	O		O	
Switching MATs by FMATS	O	×		O	
Writing H'5A to FKEY	O	×	O		
Setting erasure parameter	O	×	O		

	Storable/Executable Area		Selected MAT		
Item	**On-chip RAM**	**User Boot MAT**	**User MAT**	**User Boot MAT**	**Embedded Program Storage MAT**
Erasure	O	×	O		
Determination of erasure result	O	×	O		
Erasing error processing	O	×*	O		
FKEY clearing	O	×	O		
Switching MATs by FMATS	O	×	O		

Note: * Switching FMATS by a program in the on-chip RAM enables this area to be used.

19.5 Protection

There are two kinds of flash memory programming/erasing protection: hardware and software protection.

19.5.1 Hardware Protection

Programming and erasing of flash memory is forcibly disabled or suspended by hardware protection. In this state, the downloading of an on-chip program and initialization are possible. However, even though a programming/erasing program is initiated, the user MAT cannot be programmed/erased, and a programming/erasing error is reported with the FPFR parameter.

Table 19.9 Hardware Protection

		Function to be Protected	
Item	**Description**	**Download**	**Programming/ Erasure**
Reset, standby protection	• The programming/erasing interface registers are initialized in the reset state (including a reset by the WDT) and hardware standby mode, and the programming/erasing protection state is entered. • The reset state will not be entered by a reset using the $\overline{\text{RES}}$ pin unless the $\overline{\text{RES}}$ pin is held low until oscillation has stabilized after the power is supplied. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the RES pulse width that is specified by the AC characteristics. If a reset is input during programming or erasure, values in the flash memory are not guaranteed. In this case, execute erasure and then execute programming again.	O	O

19.5.2 Software Protection

Software protection is set up by disabling download of on-chip programming/erasing programs or by means of a key code.

Table 19.10 Software Protection

		Function to be Protected	
Item	**Description**	**Download**	**Programming/ Erasure**
Protection by SCO bit	• The programming/erasing protection state is entered by clearing the SCO bit in FCCS to 0 to disable downloading of the programming/erasing programs.	O	O
Protection by FKEY	• Downloading and programming/erasing are disabled unless the required key code is written in FKEY. Different key codes are used for downloading and programming/erasing.	O	O

19.5.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when an error occurs, in the form of the microcomputer entering runaway during programming/erasing of the flash memory or operations that are not following the stipulated procedures for programming/erasing. Aborting programming or erasure in such cases prevents damage to the flash memory due to excessive programming or erasing.

If the microcomputer malfunctions during programming/erasing of the flash memory, the FLER bit in FCCS is set to 1 and the error-protection state is entered, and this aborts the programming or erasure.

The FLER bit is set to 1 in the following conditions:

1. When an interrupt such as NMI occurs during programming/erasing.
2. When the flash memory is read during programming/erasing (including a vector read or an instruction fetch).
3. When a SLEEP instruction (including software-standby mode) is executed during programming/erasing.

4. When a bus master other than the CPU, such as the DTC, acquires the bus during programming/erasing

Error protection is cancelled only by a reset or a transition to hardware-standby mode.

Note that the reset should be released after a reset period of 100 µs which is longer than normal. Since high voltages are applied during programming/erasing of the flash memory, some voltage may remain after the error-protection state has been entered. For this reason, it is necessary to reduce the risk of damage to the flash memory by extending the reset period so that the charge is released.

The state transition diagram in figure 19.16 shows transitions to and from the error-protection state.

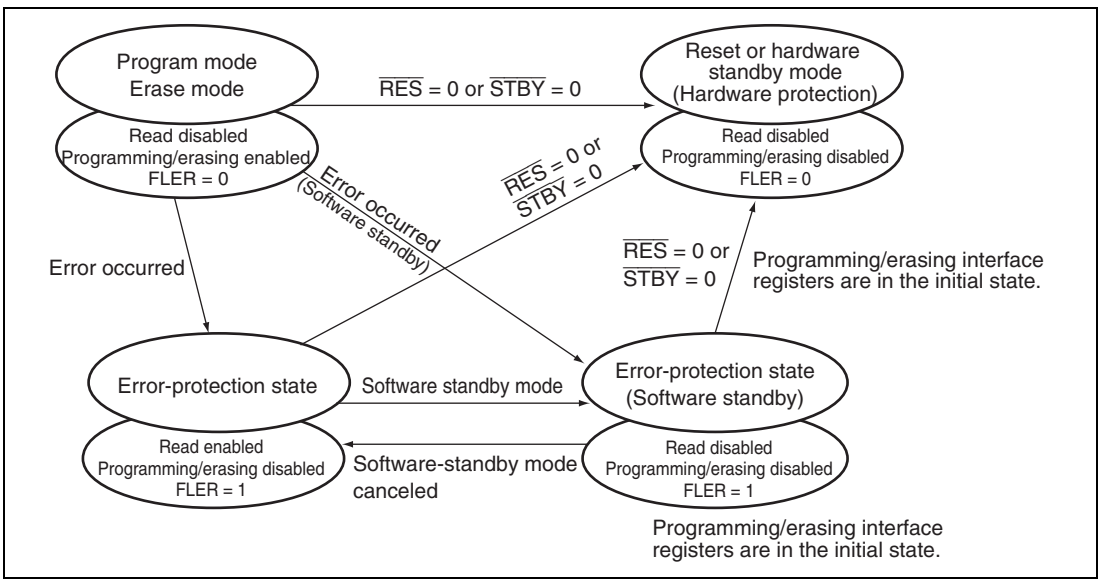


Figure 19.16 Transitions to Error-Protection State

19.6 Switching between User MAT and User Boot MAT

It is possible to switch between the user MAT and user boot MAT. However, the following procedure is required because both of these MATs are allocated to address 0.
(Switching to the user boot MAT disables programming and erasing. Programming of the user boot MAT should take place in boot mode or programmer mode.)

1. MAT switching by FMATS should always be executed from the on-chip RAM.
2. To ensure that switching has finished and access is made to the newly switched MAT, execute four NOP instructions in the same on-chip RAM immediately after writing to FMATS (this prevents access to the flash memory during MAT switching).
3. If an interrupt has occurred during switching, there is no guarantee of which memory MAT is being accessed.

   Always mask the maskable interrupts before switching between MATs. In addition, configure the system so that NMI interrupts do not occur during MAT switching.
4. After the MATs have been switched, take care because the interrupt vector table will also have been switched.

   If interrupt handling is to be the same before and after MAT switching, transfer the interrupt handling routines to the on-chip RAM and set the WEINTE bit in FCCS to place the interrupt-vector table in the on-chip RAM.
5. Memory sizes of the user MAT and user boot MAT are different. Do not access a user boot MAT in a space of 8 Kbytes or more. If access goes beyond the 8-Kbyte space, the values read are undefined.

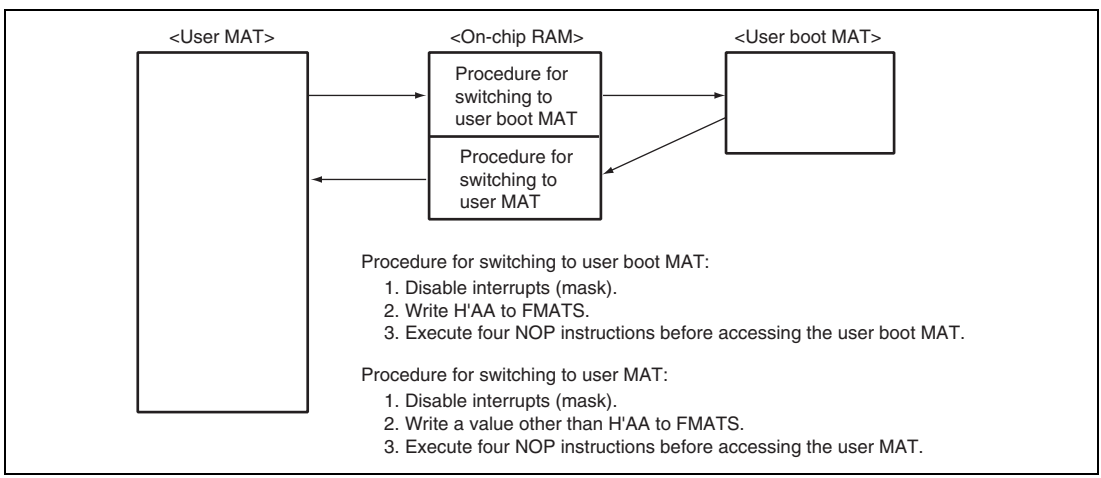


Figure 19.17 Switching between User MAT and User Boot MAT

19.7 Programmer Mode

Along with its on-board programming mode, this LSI also has a programmer mode as another mode for programming/erasing of programs and data. In programmer mode, a general PROM programmer that supports Renesas microcomputers with 512-Kbyte flash memory as a device type*[1] can be used to freely write programs to the on-chip ROM. Programming/erasing is possible on the user MAT and user boot MAT*[2]. Figure 19.18 shows a memory map in programmer mode.

A status-polling system is adopted for operation in automatic programming, automatic erasure, and status-read modes. In status-read mode, details of the internal signals are output after execution of automatic programming or automatic erasure. In programmer mode, a 12-MHz clock signal must be input.

Notes: 1. In this LSI, set the programming voltage of the PROM programmer to 3.3 V.
2. For the PROM programmer and the version of its program, see the instruction manuals for socket adapter.

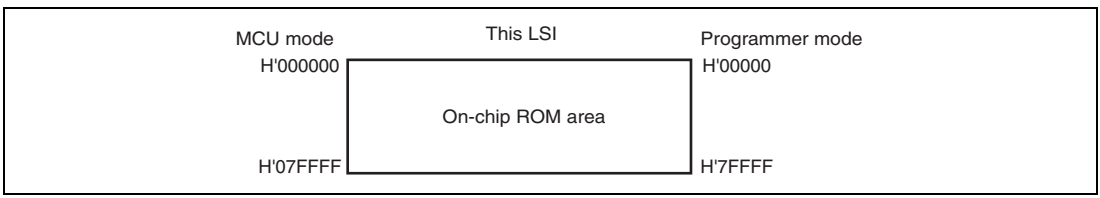


Figure 19.18 Memory Map in Programmer Mode

19.8 Serial Communication Interface Specifications for Boot Mode

The boot program initiated in boot mode performs transmission and reception with the host PC via the on-chip SCI. The serial communication interface specifications for the host and boot program are shown below.

(1) Status

The boot program has three states.

1. Bit-rate-adjustment state

   In this state, the boot program adjusts the bit rate to communicate with the host. Initiating boot mode enables starting of the boot program and transition to the bit-rate-adjustment state. The boot program receives the command from the host to adjust the bit rate. After adjusting the bit rate, the boot program enters the inquiry/selection state.
2. Inquiry/Selection state

   In this state, the boot program responds to inquiry commands from the host. The device name, clock mode, and bit rate are selected in this state. After selection of these settings, the boot program makes a transition to the programming/erasing state by the command for a transition to the programming/erasing state. The boot program transfers the libraries required for erasure to the on-chip RAM and erases the user MATs and user boot MATs before the transition to the programming/erasing state.
3. Programming/erasing state

   Programming and erasure by the boot program take place in this state. The boot program is made to transfer the programming/erasing programs to the on-chip RAM by commands from the host. Sum check and blank check are executed by sending commands from the host.

The boot program states are shown in figure 19.19.

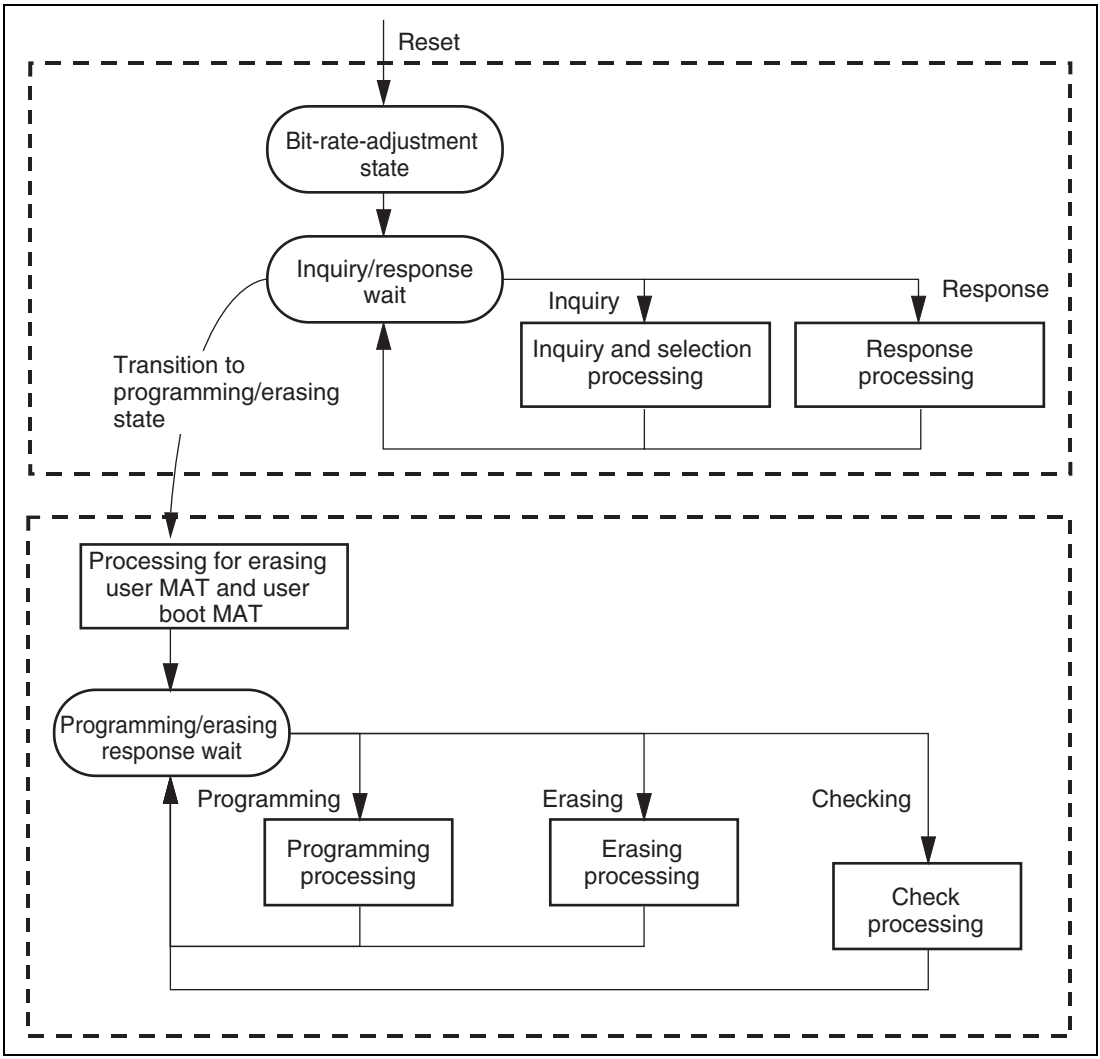


Figure 19.19 Boot Program States

(2) Bit-Rate-Adjustment State

The bit rate is adjusted by measuring the period of a low-level byte (H'00) transmitted from the host. The bit rate can be changed by the command for a new bit rate selection. After the bit rate has been adjusted, the boot program enters the inquiry/selection state. The bit-rate-adjustment sequence is shown in figure 19.20.

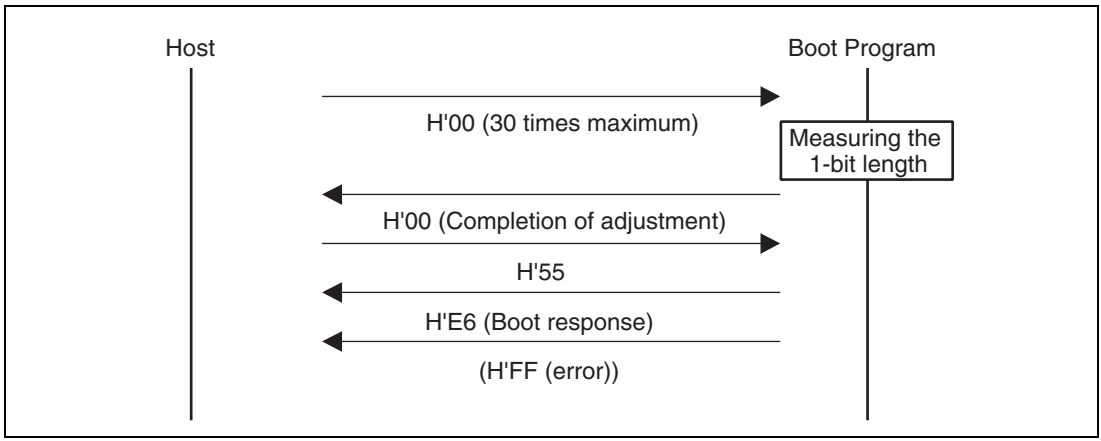


Figure 19.20 Bit-Rate-Adjustment Sequence

(3) Communications Protocol

After adjustment of the bit rate, the protocol for communications between the host and the boot program is as shown below.

1. 1-byte commands and 1-byte responses

   These commands and responses are comprised of a single byte. They are the inquiries and the ACK for successful completion.
2. n-byte commands or n-byte responses

   These commands and responses are comprised of n bytes of data. They are selection commands and responses to inquiries.

   The size of program data is not included under this heading because it is determined in another command.
3. Error response

   This response is an error response to the commands. It is two bytes of data, and consists of an error response and an error code.

4. Programming of 128 bytes

   The size is not specified in the commands. The data size is indicated in the response to the programming unit inquiry.

5. Memory read response

   This response consists of r4 bytes of data.

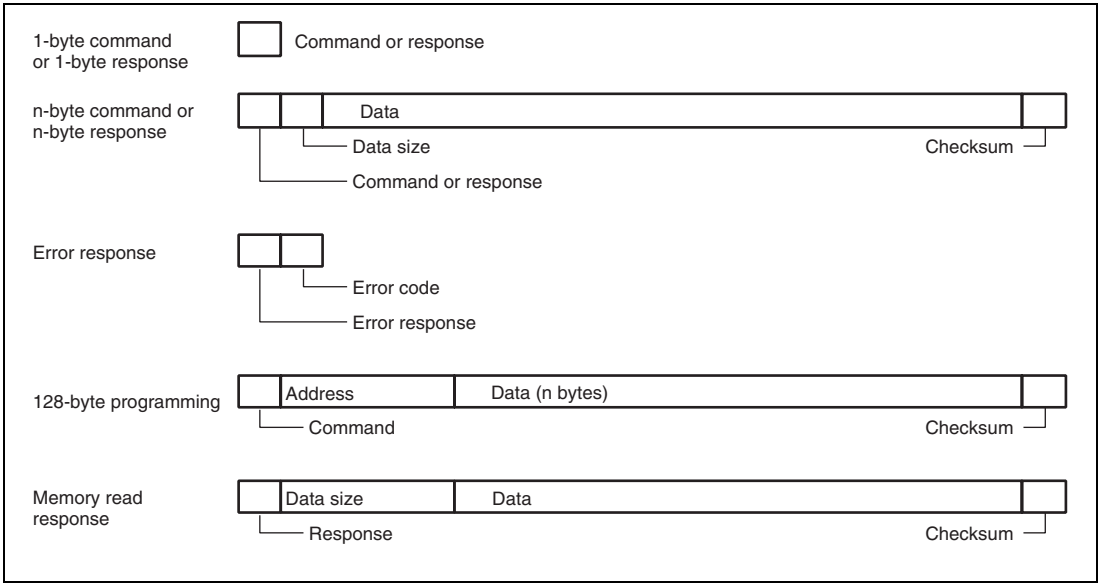


Figure 19.21 Communication Protocol Format

- Command (1 byte): Commands for inquiries, selection, programming, erasing, and checking
- Response (1 byte): Response to an inquiry
- Size (1 byte): The amount of transfer data excluding the command, size, and checksum
- Data (n bytes): Detailed data of a command or response
- Checksum (1 byte): The checksum is calculated so that the total of all values from the command byte to the SUM byte becomes H'00.
- Error response (1 byte): Error response to a command
- Error code (1 byte): Type of the error
- Address (4 bytes): Address for programming
- Data (n bytes): Data to be programmed (n is indicated in the response to the programming unit inquiry.)
- Data size (4 bytes): Four-byte response to a memory read

(4) Inquiry/Selection State

The boot program returns information from the flash memory in response to the host's inquiry commands and sets the device code, clock mode, and bit rate in response to the host's selection command.

Inquiry and selection commands are listed in table 19.11.

Table 19.11 Inquiry and Selection Commands

Command	Command Name	Description
H'20	Supported Device Inquiry	Inquiry regarding device code and product name
H'10	Device Selection	Selection of device code
H'21	Clock Mode Inquiry	Inquiry regarding number of clock modes and values of each mode
H'11	Clock Mode Selection	Indication of the selected clock mode
H'22	Division Ratio Inquiry	Inquiry regarding the number of types of division ratios, and the number and values of each ratio type
H'23	Operating Clock Frequency Inquiry	Inquiry regarding the maximum and minimum values of the main clock and peripheral clock
H'24	User Boot MAT Information Inquiry	Inquiry regarding the number of user boot MATs and the start and last addresses of each MAT
H'25	User MAT Information Inquiry	Inquiry regarding the number of user MATs and the start and last addresses of each MAT
H'26	Erased Block Information Inquiry	Inquiry regarding the number of blocks and the start and last addresses of each block
H'27	Programming Unit Inquiry	Inquiry regarding the size of program data
H'3F	New Bit Rate Selection	Selection of the new bit rate
H'40	Transition to Programming/Erasing State	Erasure of user MAT and user boot MAT, and transition to programming/erasing state
H'4F	Boot Program Status Inquiry	Inquiry into the processing status of the boot program

The selection commands, which are device selection (H'10), clock mode selection (H'11), and new bit rate selection (H'3F), should be transmitted from the host in that order. These commands are needed in all cases. When two or more selection commands are transmitted at the same time, the last command will be valid.

All of these commands, except for boot program status inquiry (H'4F), will be valid until the boot program receives the programming/erasing state transition command (H'40). The host can choose the needed commands out of the above commands and make inquiries. The boot program status inquiry command (H'4F) remains valid even after the boot program has received the programming/erasing state transition command (H'40).

(a) Supported Device Inquiry

The boot program will return the device codes of the supported devices and the product names in response to the supported device inquiry command.

Command

H'20

- Command, H'20 (1 byte): Inquiry regarding supported devices

Response

H'30	Size	Number of devices	
Number of characters	Device code		Product name
...			
SUM			

- Response, H'30 (1 byte): Response to the supported device inquiry
- Size (1 byte): The number of bytes to be transferred, excluding the command, size, and checksum, that is, the total amount of data consisting the number of devices, the number of characters, device codes, and product names
- Number of devices (1 byte): The number of device types supported by the boot program in the microcomputer
- Number of characters (1 byte): The number of characters in the device codes and boot program's name
- Device code (4 bytes): ASCII code of the supported product name
- Product name (n bytes): ASCII code of the boot program type name
- SUM (1 byte): Checksum

  The checksum is calculated so that the total number of all values from the command byte to the SUM byte becomes H'00.

(b) Device Selection

The boot program will set the specified supported device in response to the device selection command. The program will return information on the selected device in response to the inquiry after this setting has been made.

Command	H'10	Size	Device code	SUM

- Command, H'10 (1 byte): Device selection
- Size (1 byte): The number of characters in the device code. Fixed at 4.
- Device code (4 bytes): Device code (ASCII code) returned in response to the supported device inquiry
- SUM (1 byte): Checksum

Response	H'06

- Response, H'06 (1 byte): Response to the device selection command.
  The boot program will return ACK when the device code matches.

Error Response	H'90	ERROR

- Error response, H'90 (1 byte): Error response to the device selection command
  ERROR (1 byte): Error code
  H'11: Checksum error
  H'21: Device code error, that is, the device code does not match

(c) Clock Mode Inquiry

The boot program will return the supported clock modes in response to the clock mode inquiry command.

Command	H'21

- Command, H'21 (1 byte): Inquiry regarding clock mode

Response	H'31	Size	Number of modes	Mode	···	SUM

- Response, H'31 (1 byte): Response to the clock mode inquiry
- Size (1 byte): Amount of data that represents the number of modes and modes
- Number of clock modes (1 byte): The number of supported clock modes.
  H'00 indicates no clock mode or the device allows the clock mode to be read.
- Mode (1 byte): Values of the supported clock modes (i.e. H'01 means clock mode 1.)
- SUM (1 byte): Checksum

(d) Clock Mode Selection

The boot program will set the specified clock mode in response to the clock mode selection command. The program will return information on the selected clock mode in response to the inquiry after this setting has been made.

The clock mode selection command should be sent after the device selection command.

Command	H'11	Size	Mode	SUM

- Command, H'11 (1 byte): Selection of clock mode
- Size (1 byte): The number of characters that represents the modes. Fixed at 1.
- Mode (1 byte): A clock mode returned in response to the clock mode inquiry.
- SUM (1 byte): Checksum

Response	H'06

- Response, H'06 (1 byte): Response to the clock mode selection command.
  The boot program will return ACK when the clock mode matches.

Error Response	H'91	ERROR

- Error response, H'91 (1 byte): Error response to the clock mode selection command
- ERROR (1 byte): Error code

  H'11: Checksum error

  H'22: Clock mode error, that is, the clock mode does not match

  Even if the number of clock modes is H'00 or H'01 by a clock mode inquiry, the clock mode must be selected using the respective value.

(e) Division Ratio Inquiry

The boot program will return the supported division ratios in response to the division ratio inquiry command.

Command	H'22

- Command, H'22 (1 byte): Inquiry regarding division ratio

Response	H'32	Size	Number of types					
	Number of division ratios	Division ratio	···					
	...							
	SUM							

- Response, H'32 (1 byte): Response to the division ratio inquiry
- Size (1 byte): The amount of data that represents the number of types, number of division ratios, and division ratios
- Number of types (1 byte): The number of supported division ratio types
  (e.g. H'02 when there are two types: main operating frequency and peripheral module operating frequency)
- Number of division ratios (1 byte): The number of supported division ratios for each operating frequency.
  The number of division ratios supported in the main module and peripheral modules.
- Division ratio (1 byte)
  - Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the clock is divided by two, the value will be H'FE[–2])

  The number of division ratios returned is the same as the number of division ratios and as many groups of data are returned as there are types.
- SUM (1 byte): Checksum

(f) Operating Clock Frequency Inquiry

The boot program will return the number of operating clock frequencies, and the maximum and minimum values in response to the operating clock frequency inquiry command.

Command	H'23

- Command, H'23 (1 byte): Inquiry regarding operating clock frequencies

Response	H'33	Size	Number of operating clock frequencies
	Minimum value of operating clock frequency		Maximum value of operating clock frequency
	...		
	SUM		

- Response, H'33 (1 byte): Response to operating clock frequency inquiry
- Size (1 byte): The amount of data that represents the number of operating clock frequencies, and the minimum and maximum values of the operating clock frequencies
- Number of operating clock frequencies (1 byte): The number of supported operating clock frequency types
  (e.g. H'02 when there are two types: main operating frequency and peripheral module operating frequency)
- Minimum value of operating clock frequency (2 bytes): Minimum value among the divided clock frequencies.

  The minimum and maximum values of operating clock frequency represent the frequency values (MHz), valid to the hundredths place, and multiplied by 100.

  (e.g. when the value is 20.00 MHz, it will be 2000, which is H'07D0)
- Maximum value of operating clock frequency (2 bytes): Maximum value among the divided clock frequencies.

  There are as many pairs of minimum and maximum values as there are operating clock frequencies.
- SUM (1 byte): Checksum

(g) User Boot MAT Information Inquiry

The boot program will return the number of user boot MATs and their addresses in response to the user boot MAT information inquiry command.

Command	H'24

- Command, H'24 (1 byte): Inquiry regarding user boot MAT information

Response	H'34	Size	Number of areas	
	Area start address			Area last address
	...			
	SUM			

- Response, H'34 (1 byte): Response to user boot MAT information inquiry
- Size (1 byte): The amount of data that represents the number of areas, area start address, and area last address
- Number of areas (1 byte): The number of consecutive user boot MAT areas.
  H'01 when the user boot MAT areas are consecutive.
- Area start address (4 bytes): Start address of the area
- Area last address (4 bytes): Last address of the area.
  There are as many groups of data representing the start and last addresses as there are areas.
- SUM (1 byte): Checksum

(h) User MAT Information Inquiry

The boot program will return the number of user MATs and their addresses in response to the user MAT information inquiry command.

Command	H'25

- Command, H'25 (1 byte): Inquiry regarding user MAT information

Response	H'35	Size	Number of areas	
	Area start address			Area last address
	...			
	SUM			

- Response, H'35 (1 byte): Response to the user MAT information inquiry
- Size (1 byte): The amount of data that represents the number of areas, area start address, and area last address
- Number of areas (1 byte): The number of consecutive user MAT areas.
  H'01 when the user MAT areas are consecutive.

- Area start address (4 bytes): Start address of the area
- Area last address (4 bytes): Last address of the area.
  There are as many groups of data representing the start and last addresses as there are areas.
- SUM (1 byte): Checksum

(i) Erased Block Information Inquiry

The boot program will return the number of erased blocks and their addresses in response to the erased block information inquiry command.

Command	H'26

- Command, H'26 (1 byte): Inquiry regarding erased block information

Response	H'36	Size	Number of blocks	
	Block start address			Block last address
	...			
	SUM			

- Response, H'36 (1 byte): Response to the erased block information inquiry
- Size (2 bytes): The amount of data that represents the number of blocks, block start address, and block last address.
- Number of blocks (1 byte): The number of erased blocks of flash memory
- Block start address (4 bytes): Start address of a block
- Block last address (4 bytes): Last address of a block
  There are as many groups of data representing the start and last addresses as there are blocks.
- SUM (1 byte): Checksum

(j) Programming Unit Inquiry

The boot program will return the programming unit used to program data in response to the programming unit inquiry command.

Command	H'27

- Command, H'27 (1 byte): Inquiry regarding programming unit

Response	H'37	Size	Programming unit	SUM

- Response, H'37 (1 byte): Response to programming unit inquiry
- Size (1 byte): The number of characters that indicate the programming unit. Fixed at 2.
- Programming unit (2 bytes): A unit for programming.
  This is the unit for reception of program data.

- SUM (1 byte): Checksum

(k) New Bit Rate Selection

The boot program will set a new bit rate in response to the new bit rate selection command, and return the new bit rate in response to the confirmation.

This new bit rate selection command should be sent after sending the clock mode selection command.

Command				
	H'3F	Size	Bit rate	Input frequency
	Number of division ratios	Division ratio 1	Division ratio 2	
	SUM			

- Command, H'3F (1 byte): Selection of new bit rate
- Size (1 byte): The amount of data that represents the bit rate, input frequency, number of division ratios, and division ratios
- Bit rate (2 bytes): New bit rate
  One hundredth of the value (e.g. when the value is 19200 bps, it will be 192, which is H'00C0.)
- Input frequency (2 bytes): Frequency of the clock input to the boot program.
  This is valid to the hundredths place and represents the frequency value (MHz) multiplied by 100. (e.g. when the value is 20.00 MHz, it will be 2000, which is H'07D0)
- Number of division ratios (1 byte): The number of supported division ratios.
  Normally the number is two: one for the main operating frequency and one for peripheral module operating frequency.
- Division ratio 1 (1 byte): The division ratio for the main operating frequency
  - Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the clock is divided by two, the value will be H'FE[–2])
- Division ratio 2 (1 byte): The division ratio for the peripheral module operating frequency
  - Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the clock is divided by two, the value will be H'FE[–2])
- SUM (1 byte): Checksum

Response	
	H'06

- Response, H'06 (1 byte): Response to selection of a new bit rate.
  The boot program will return ACK when the new bit rate can be set.

Error Response		
	H'BF	ERROR

- Error response, H'BF (1 byte): Error response to selection of a new bit rate

- ERROR (1 byte): Error code
  - H'11: Checksum error
  - H'24: Bit rate selection error
    The rate is not available.
  - H'25: Input frequency error
    The input frequency is not within the specified range.
  - H'26: Division ratio error
    The division ratio does not match an available ratio.
  - H'27: Operating frequency error
    The operating frequency is not within the specified range.

(5) Receive Data Check

The methods for checking received data are listed below.

1. Input frequency

   The received value of the input frequency is checked to ensure that it is within the range of the minimum to maximum frequencies which are available with the clock modes of the specified device. When the value is out of this range, an input frequency error is generated.
2. Division ratio

   The received value of the division ratio is checked to ensure that it matches the division for the clock modes of the specified device. When the value is out of this range, a division ratio error is generated.
3. Operating frequency

   Operating frequency is calculated from the received value of the input frequency and the division ratio. The input frequency is the frequency input to the LSI, and the operating frequency is the frequency at which the LSI is actually operated. The formula is given below.

   Operating frequency = Input frequency ÷ Division ratio

   The calculated operating frequency should be checked to ensure that it is within the range of the minimum to maximum frequencies which are available with the clock modes of the specified device. When it is out of this range, an operating frequency error is generated.
4. Bit rate

   To facilitate error checking, the value (n) of clock select (CKS) in the serial mode register (SMR), and the value (N) in the bit rate register (BRR), which are found from the peripheral operating clock frequency (ϕ) and bit rate (B), are used to calculate the error rate to ensure that it is less than 4%. If the error is more than 4%, a bit rate selection error is generated. The error is calculated using the following formula:

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

When the new bit rate is selectable, the rate will be set in the register after sending ACK in response. The host will send an ACK with the new bit rate for confirmation and the boot program will response with that rate.

Confirmation	H'06

- Confirmation, H'06 (1 byte): Confirmation of a new bit rate

Response	H'06

- Response, H'06 (1 byte): Response to confirmation of a new bit rate

The sequence of new bit rate selection is shown in figure 19.22.

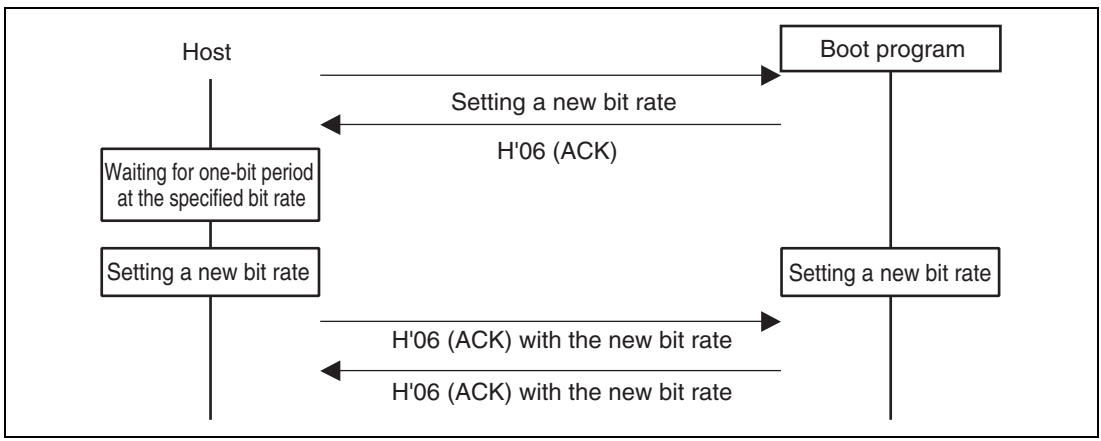


Figure 19.22 Sequence of New Bit Rate Selection

(6) Transition to Programming/Erasing State

The boot program will transfer the erasing program, and erase the user MATs and user boot MATs in that order in response to the transition to the programming/erasing state command. On completion of this erasure, ACK will be returned and a transition made to the programming/erasing state.

Before sending the programming selection command or program data, the host should select the LSI device with the device selection command, the clock mode with the clock mode selection command, and the new bit rate with the new bit rate selection command, and then send the transition to programming/erasing state command.

Command	H'40

- Command, H'40 (1 byte): Transition to programming/erasing state

Response	H'06

- Response, H'06 (1 byte): Response to transition to programming/erasing state.
  The boot program will return ACK when the user MAT and user boot MAT have been erased normally by the transferred erasing program.

Error Response	H'C0	H'51

- Error response, H'C0 (1 byte): Error response to blank check of user boot MAT
- Error code, H'51 (1 byte): Erasing error
  An error occurred and erasure was not completed.

(7) Command Error

A command error will occur when a command is undefined, the order of commands is incorrect, or a command is unacceptable. Issuing a clock mode selection command before a device selection command, or an inquiry command after the transition to programming/erasing state command, are such examples.

Error Response	H'80	H'xx

- Error response, H'80 (1 byte): Command error
- Command, H'xx (1 byte): Received command

(8) Command Order

The order for commands in the inquiry/selection state is shown below.

1. A supported device inquiry (H'20) should be made to inquire about the supported devices.
2. The device should be selected from among those described by the returned information and set with a device selection (H'10) command.
3. A clock mode inquiry (H'21) should be made to inquire about the supported clock modes.
4. The clock mode should be selected from among those described by the returned information and set.
5. After selection of the device and clock mode, inquiries for other required information should be made, such as the division ratio inquiry (H'22) or operating frequency inquiry (H'23), which are needed for a new bit rate selection.
6. A new bit rate should be selected with the new bit rate selection (H'3F) command, according to the returned information on division ratios and operating frequencies.

7. After selection of the device and clock mode, programming/erasing information of the user boot MAT and user MAT should be inquired using the user boot MAT information inquiry (H'24), user MAT information inquiry (H'25), erased block information inquiry (H'26), and programming unit inquiry (H'27).
8. After making inquiries and selecting a new bit rate, issue the transition to programming/erasing state command (H'40). The boot program will then enter the programming/erasing state.

(9) Programming/Erasing State

In the programming/erasing state, a programming selection command makes the boot program select the programming method, a 128-byte programming command makes it program the memory with data, and an erasing selection command and block erasing command make it erase the block. The programming/erasing commands are listed in table 19.12.

Table 19.12 Programming/Erasing Commands

Command	Command Name	Description
H'42	User boot MAT programming selection	Transfers the user boot MAT programming program
H'43	User MAT programming selection	Transfers the user MAT programming program
H'50	128-byte programming	Programs 128 bytes of data
H'48	Erasing selection	Transfers the erasing program
H'58	Block erasing	Erases a block of data
H'52	Memory read	Reads the contents of memory
H'4A	User boot MAT sum check	Checks the sum of the user boot MAT
H'4B	User MAT sum check	Checks the sum of the user MAT
H'4C	User boot MAT blank check	Checks whether the contents of the user boot MAT are blank
H'4D	User MAT blank check	Checks whether the contents of the user MAT are blank
H'4F	Boot program status inquiry	Inquires into the boot program's processing status

Programming: Programming is executed by a programming-selection command and a 128-byte programming command.

First, the host should send the programming-selection command, and select the programming method and programming MATs. There are two programming selection commands according to the area and method for programming.

1. User boot MAT programming selection
2. User MAT programming selection

After issuing the programming selection command, the host should send the 128-byte programming command. The 128-byte programming command that follows the selection command represents the program data according to the method specified by the selection command. When more than 128 bytes of data are to be programmed, 128-byte programming commands should be executed repeatedly. Sending from the host a 128-byte programming command with H'FFFFFFFF as the address will stop the programming. On completion of programming, the boot program will wait for selection of programming or erasing.

In case of continuing programming with another method or programming of another MAT, the procedure must be repeated from the programming selection command.

The sequence for the programming selection and 128-byte programming commands is shown in figure 19.23.

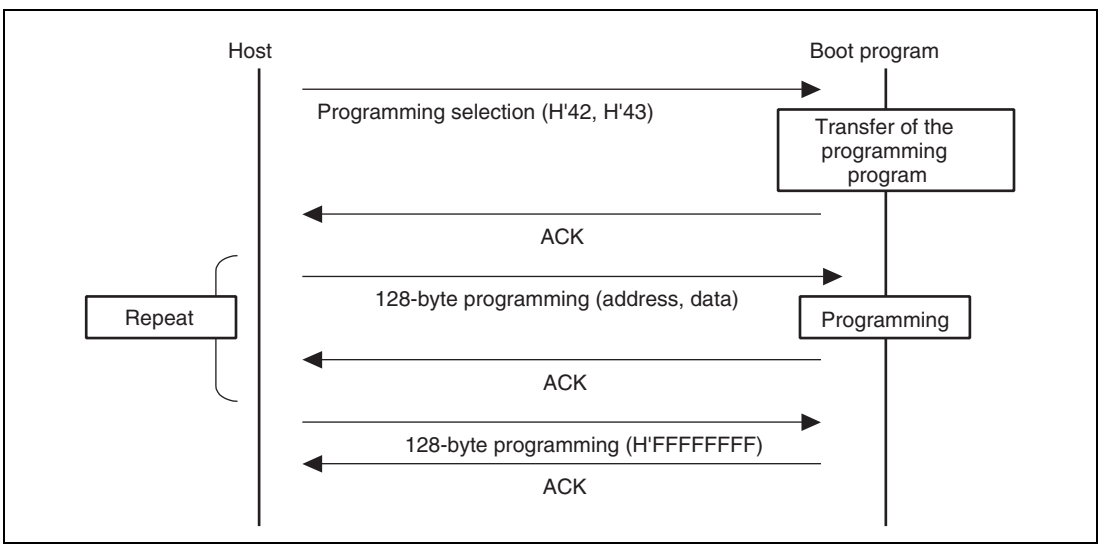


Figure 19.23 Programming Sequence

(a) User Boot MAT Programming Selection

The boot program will transfer a programming program in response to the user boot MAT programming selection command. The data is programmed to the user boot MAT by the transferred programming program.

Command	H'42

- Command, H'42 (1 byte): User boot MAT programming selection

Response	H'06

- Response, H'06 (1 byte): Response to user boot MAT programming selection.
  When the programming program has been transferred, the boot program will return ACK.

Error Response	H'C2	ERROR

- Error response, H'C2 (1 byte): Error response to user boot MAT programming selection
- ERROR (1 byte): Error code
  H'54: Selection processing error (transfer error occurs and processing is not completed)

User MAT Programming Selection: The boot program will transfer a programming program in response to the user MAT programming selection command. The data is programmed to the user MAT by the transferred programming program.

Command	H'43

- Command, H'43 (1 byte): User MAT programming selection

Response	H'06

- Response, H'06 (1 byte): Response to user MAT programming selection.
  When the programming program has been transferred, the boot program will return ACK.

Error Response	H'C3	ERROR

- Error response, H'C3 (1 byte): Error response to user MAT programming selection
- ERROR (1 byte): Error code
  H'54: Selection processing error (transfer error occurs and processing is not completed)

(b) 128-Byte Programming

The boot program will use the programming program transferred by the programming selection command for programming the user boot MAT or user MAT in response to the 128-byte programming command.

Command

H'50	Address						
Data	···						
···							
SUM							

- Command, H'50 (1 byte): 128-byte programming
- Programming address (4 bytes): Start address for programming.
  Multiple of the size specified in response to the programming unit inquiry command.
  (e.g. H'00, H'01, H'00, H'00: H'010000)
- Program data (128 bytes): Data to be programmed.
  The size is specified in response to the programming unit inquiry command.
- SUM (1 byte): Checksum

Response

H'06

- Response, H'06 (1 byte): Response to 128-byte programming.
  On completion of programming, the boot program will return ACK.

Error Response

H'D0	ERROR

- Error response, H'D0 (1 byte): Error response to 128-byte programming
- ERROR (1 byte): Error code

  H'11: Checksum Error
  H'2A: Address error
  H'53: Programming error
  A programming error has occurred and programming cannot be continued.

The specified address should match the boundary of the programming unit. For example, when the programming unit is 128 bytes, the lower eight bits of the address should be H'00 or H'80.
When the program data is less than 128 bytes, the host should fill the rest with H'FF.

Sending the 128-byte programming command with the address of H'FFFFFFFF will stop the programming operation. The boot program will interpret this as the end of programming and wait for selection of programming or erasing.

Command

H'50	Address	SUM

- Command, H'50 (1 byte): 128-byte programming
- Programming address (4 bytes): End code (H'FF, H'FF, H'FF, H'FF)
- SUM (1 byte): Checksum

Response	H'06

- Response, H'06 (one byte): Response to 128-byte programming.
  On completion of programming, the boot program will return ACK.

Error Response	H'D0	ERROR

- Error response, H'D0 (1 byte): Error response to 128-byte programming
- ERROR (1 byte): Error code

  H'11: Checksum error
  H'2A: Address error
  H'53: Programming error
  An error has occurred in programming and programming cannot be continued.

(10) Erasure

Erasure is performed with the erasure selection and block erasure commands.

First, erasure is selected by the erasure selection command and the boot program then erases the specified block. The command should be repeatedly executed if two or more blocks are to be erased. Sending a block erasure command from the host with the block number H'FF will stop the erasure processing. On completion of erasing, the boot program will wait for selection of programming or erasing.

The sequence for the erasure selection command and block erasure command is shown in figure 19.24.

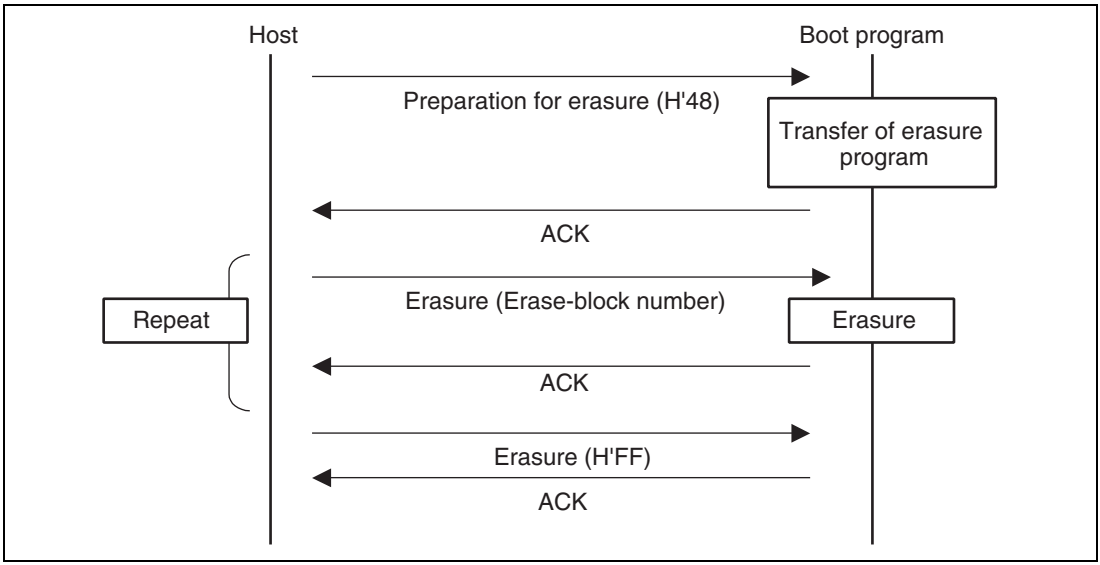


Figure 19.24 Erasure Sequence

(a) Erasure Selection

The boot program will transfer the erasing program in response to the erasure selection command. User MAT data is erased by the transferred erasing program.

Command	H'48

- Command, H'48 (1 byte): Erasure selection

Response	H'06

- Response, H'06 (1 byte): Response to erasure selection.
  After the erasing program has been transferred, the boot program will return ACK.

Error Response	H'C8	ERROR

- Error response, H'C8 (1 byte): Error response to erasure selection
- ERROR (1 byte): Error code

  H'54: Selection processing error (transfer error occurs and processing is not completed)

(b) Block Erasure

The boot program will erase the contents of the specified block in response to the block erasure command.

Command	H'58	Size	Block number	SUM

- Command, H'58 (1 byte): Erasure
- Size (1 byte): The number of characters that represents the erase-block number.
  Fixed at 1.
- Block number (1 byte): Number of the block to be erased
- SUM (1 byte): Checksum

Response	H'06

- Response, H'06 (1 byte): Response to erasure
  On completion of erasure, the boot program will return ACK.

Error Response	H'D8	ERROR

- Error response, H'D8 (1 byte): Response to erasure
- ERROR (1 byte): Error code

  H'11: Checksum error
  H'29: Block number error
  Block number is incorrect.
  H'51: Erasing error
  An error has occurred during erasure.

On receiving block number H'FF, the boot program will stop erasure and wait for a selection command.

Command	H'58	Size	Block number	SUM

- Command, H'58 (1 byte): Erasure
- Size (1 byte): The number of characters that represents the block number.
  Fixed at 1.
- Block number (1 byte): H'FF
  Stop code for erasure
- SUM (1 byte): Checksum

Response	H'06

- Response, H'06 (1 byte): Response to end of erasure (ACK will be returned)

When erasure is to be performed again after the block number H'FF has been sent, the procedure should be executed from the erasure selection command.

(11) Memory Read

The boot program will return the data in the specified address in response to the memory read command.

Command

H'52	Size	Area	Read address				
Read size				SUM			

- Command, H'52 (1 byte): Memory read
- Size (1 byte): Amount of data that represents the area, read address, and read size (fixed at 9)
- Area (one byte)

  H'00: User boot MAT
  H'01: User MAT
  An address error occurs when the area setting is incorrect.
- Read address (4 bytes): Start address to be read from
- Read size (4 bytes): Size of data to be read
- SUM (1 byte): Checksum

Response

H'52	Read size						
Data	···						
SUM							

- Response: H'52 (1 byte): Response to memory read
- Read size (4 bytes): Size of data to be read
- Data (n bytes): Data of the read size from the read address
- SUM (1 byte): Checksum

Error Response

H'D2	ERROR

- Error response: H'D2 (1 byte): Error response to memory read
- ERROR (1 byte): Error code

  H'11: Checksum error
  H'2A: Address error
  The read address is not in the MAT.
  H'2B: Size error
  The read size exceeds the MAT.

(12) User Boot MAT Sum Check

The boot program will return the total amount of bytes of the user boot MAT contents in response to the user boot MAT sum check command.

Command	H'4A

- Command, H'4A (1 byte): Sum check for user boot MAT

Response	H'5A	Size	Checksum of MAT	SUM

- Response, H'5A (1 byte): Response to the checksum of user boot MAT
- Size (1 byte): The number of characters that represents the checksum. Fixed at 4.
- Checksum of MAT (4 bytes): Checksum of user boot MATs. The total amount of data is obtained in byte units.
- SUM (1 byte): Checksum (for transmit data)

(13) User MAT Sum Check

The boot program will return the total amount of bytes of the user MAT contents in response to the user MAT sum check command.

Command	H'4B

- Command, H'4B (1 byte): Checksum for user MAT

Response	H'5B	Size	Checksum of MAT	SUM

- Response, H'5B (1 byte): Response to the checksum of the user MAT
- Size (1 byte): The number of characters that represents the checksum. Fixed at 4.
- Checksum of MAT (4 bytes): Checksum of user MATs. The total amount of data is obtained in byte units.
- SUM (1 byte): Checksum (for transmit data)

(14) User Boot MAT Blank Check

The boot program will check whether or not all user boot MATs are blank and return the result in response to the user boot MAT blank check command.

Command	H'4C

- Command, H'4C (1 byte): Blank check for user boot MATs

Response	H'06

- Response, H'06 (1 byte): Response to blank check of user boot MATs. If all user boot MATs are blank (H'FF), the boot program will return ACK.

Error Response	H'CC	H'52

- Error response, H'CC (1 byte): Error response to blank check for user boot MATs
- Error code, H'52 (1 byte): Erasure incomplete error

(15) User MAT Blank Check

The boot program will check whether or not all user MATs are blank and return the result in response to the user MAT blank check command.

Command	H'4D

- Command, H'4D (1 byte): Blank check for user MATs

Response	H'06

- Response, H'06 (1 byte): Response to blank check for user MATs.
  If all user MATs are blank (H'FF), the boot program will return ACK.

Error Response	H'CD	H'52

- Error response, H'CD (1 byte): Error response to blank check for user MATs
- Error code, H'52 (1 byte): Erasure incomplete error

(16) Boot Program State Inquiry

The boot program will return indications of its present state and error condition in response to the boot program state inquiry command. This inquiry can be made in either the inquiry/selection state or the programming/erasing state.

Command	H'4F

- Command, H'4F (1 byte): Inquiry regarding boot program's state

Response	H'5F	Size	Status	ERROR	SUM

- Response, H'5F (1 byte): Response to boot program state inquiry
- Size (1 byte): The number of characters. Fixed at 2.
- Status (1 byte): State of the standard boot program
- ERROR (1 byte): Error status

  ERROR = 0 indicates normal operation.
  ERROR = 1 indicates error has occurred.
- SUM (1 byte): Checksum

Table 19.13 Status Code

Code	Description
H'11	Device Selection Wait
H'12	Clock Mode Selection Wait
H'13	Bit Rate Selection Wait
H'1F	Programming/Erasing State Transition Wait (Bit rate selection is completed)
H'31	Programming/Erasing State
H'3F	Programming/Erasing Selection Wait (Erasure is completed)
H'4F	Program Data Receive Wait (Programming is completed)
H'5F	Erase Block Specification Wait (Erasure is completed)

Table 19.14 Error Code

Code	Description
H'00	No Error
H'11	Checksum Error
H'12	Program Size Error
H'21	Device Code Mismatch Error
H'22	Clock Mode Mismatch Error
H'24	Bit Rate Selection Error
H'25	Input Frequency Error
H'26	Division Ratio Error
H'27	Operating Frequency Error
H'29	Block Number Error
H'2A	Address Error
H'2B	Data Length Error
H'51	Erasing Error
H'52	Erasure Incomplete Error
H'53	Programming Error
H'54	Selection Processing Error
H'80	Command Error
H'FF	Bit-Rate-Adjustment Confirmation Error

19.9 Usage Notes

1. The initial state of a Renesas product at shipment is the erased state. For a product whose history of erasing is undefined, automatic erasure for checking the initial state (erased state) and compensating is recommended.
2. For the PROM programmer suitable for programmer mode in this LSI and its program version, refer to the instruction manual of the socket adapter.
3. If the socket, socket adapter, or product index of the PROM programmer does not match the specifications, excessive current flows and the product may be damaged.
4. If a voltage higher than the rated voltage is applied, the product may be fatally damaged. Use a PROM programmer that supports a programming voltage of 3.3 V for Renesas microcomputers with 512-Kbyte flash memory. Do not set the programmer to HN28F101 or a programming voltage of 5.0 V. Use only the specified socket adapter. If other adapters are used, the product may be damaged.
5. Do not remove the chip from the PROM programmer nor input a reset signal during programming/erasing. As a high voltage is applied to the flash memory during programming/erasing, doing so may damage flash memory permanently. If a reset is input accidentally, the reset must be released after a reset period of 100 µs which is longer than normal.
6. After programming/erasing, access to the flash memory is prohibited until FKEY is cleared. In case the LSI mode is changed to generate a reset on completion of a programming/erasing operation, a reset state ($\overline{RES}$ = 0) of 100 µs or more must be secured. Transitions to the reset state or hardware standby mode are prohibited during programming/erasing operations. However, when the reset signal is accidentally input to the chip, the reset must be released after a reset period of 100 µs that is longer than normal.
7. At turning on or off the VCC power supply, fix the $\overline{RES}$ pin to low and set the flash memory to the hardware protection state. This power-on or power-off timing must also be satisfied at a power-off or power-on caused by a power failure and other factors.
8. Perform programming to a 128-byte programming-unit block only once in on-board programming or programmer mode.

   Perform programming in the state where the programming-unit block is fully erased.
9. When a chip is to be reprogrammed with the programmer after it has already been programmed or erased in on-board programming mode, automatic programming is recommended to be performed after automatic erasure.
10. To write data or programs to the flash memory, program data and programs must be allocated to addresses higher than that of the external interrupt vector table (in normal mode: H'0020, in advanced mode: H'000040), and H'FF must be written to the areas that are reserved for the system in the exception handling vector table.

11. If data other than H'FF (4 bytes) is written to the key code area (in normal mode: H'001E to H'001F, in advance mode: H'00003C to H'00003F) of flash memory, reading cannot be performed in programmer mode. (In this case, data is read as H'00. Rewrite is possible after erasing the data.) For reading in programmer mode, make sure to write H'FF to the entire key code area.

    If data other than H'FF is to be written to the key code area in programmer mode, a verification error will occur unless a software countermeasure is taken for the PROM programmer and version of program.
12. The code size of the programming program that includes the initialization routine or the erasing program that includes the initialization routine is 2 Kbytes or less. Accordingly, when the CPU clock frequency is 20 MHz, the download for each program takes approximately 200 µs at the maximum.
13. While an instruction in the on-chip RAM is being executed, the DTC can write to the SCO bit in FCCS that is used for a download request or FMATS that is used for MAT switching. Make sure that these registers are not accidentally written to, otherwise an on-chip program may be downloaded and damage the on-chip RAM or a MAT switchover may occur and the CPU get out of control. Do not use the DTC to write to flash memory related registers.
14. A programming/erasing program for flash memory used in the conventional H8S F-ZTAT microcomputer which does not support download of the on-chip program by a SCO transfer request cannot run in this LSI. Be sure to download the on-chip program to execute programming/erasing of flash memory in this H8S F-ZTAT microcomputer.
15. Unlike the conventional H8S F-ZTAT microcomputer, no countermeasures are available for a runaway by the WDT during programming/erasing. Prepare countermeasures (e.g. use of periodic timer interrupts) for the WDT with taking the programming/erasing time into consideration as required.

Section 20 PROM (OTP Version)

The R4P2125 has 512-Kbyte one-time programmable on-chip flash memory. The PROM is connected to the bus master and a 16-bit width data bus. The CPU accesses both byte and word data in one state, enabling faster instruction fetches and higher processing speed.

The mode pins (MD2*, MD1, and MD0) and the EXPE bit in MDCR can be set to enable or disable the on-chip PROM. For details on MDCR, see section 3.2.1, Mode Control Register (MDCR).

The R4P/2125 is programmable using the PROM programmer.

Note: * MD2 is not supported in SDIP-64 and QFP-64.

Figure 20.1 shows a block diagram of the PROM.

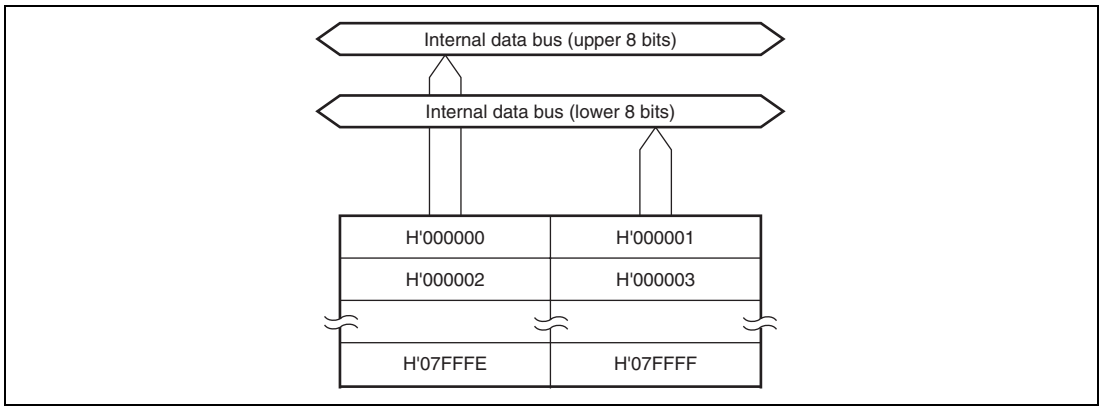


Figure 20.1 PROM Block Diagram (R4P2125)

20.1 Programmer Mode

20.1.1 Programmer Mode Setting

Programs and data can be written in programmer mode. In programmer mode, the on-chip PROM can be freely programmed using a PROM programmer that supports Renesas microcomputer device types with 512-Kbyte on-chip flash memory.

20.1.2 Socket Adapters and Memory Map

In programmer mode, a socket adapter is mounted on the PROM programmer to match the packages concerned. For the socket adapter, confirm with a programmer manufacturer supporting the Renesas microcomputer device type with 512-Kbyte on-chip flash memory.

Figure 20.2 shows the memory map in programmer mode. For pin names in programmer mode, see table 1.1.

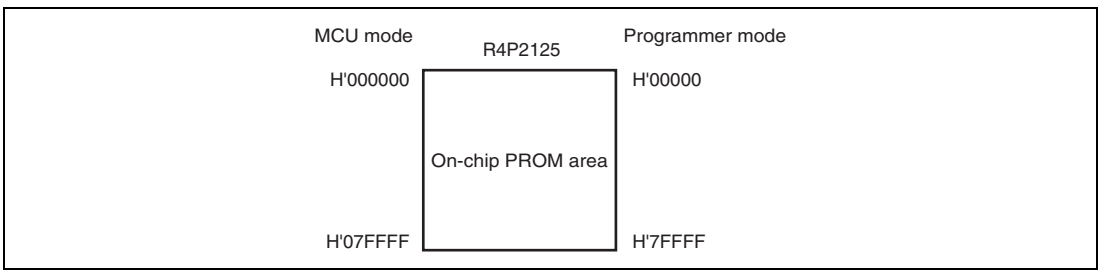


Figure 20.2 Memory Map in Programmer Mode

20.2 Usage Notes

1. The initial state of a Renesas product at shipment is the erased state. For a product whose history of erasing is undefined, automatic erasure for checking the initial state (erased state) and compensating is recommended.
2. For the PROM programmer suitable for programmer mode in this LSI and its program version, refer to the instruction manual of the socket adapter.
3. If the socket, socket adapter, or product index of the PROM programmer does not match the specifications, excessive current flows and the product may be damaged.
4. If a voltage higher than the rated voltage is applied, the product may be fatally damaged. Use a PROM programmer that supports a programming voltage of 3.3 V for Renesas microcomputers with 512-Kbyte flash memory. Do not set the programmer to HN28F101 or a programming voltage of 5.0 V. Use only the specified socket adapter. If other adapters are used, the product may be damaged.
5. Do not remove the chip from the PROM programmer nor input a reset signal during programming/erasing. As a high voltage is applied to the flash memory during programming/erasing, doing so may damage flash memory permanently. If a reset is input accidentally, the reset must be released after a reset period of 100 µs which is longer than normal.

Section 21 Clock Pulse Generator

This LSI incorporates a clock pulse generator which generates the system clock (ϕ), internal clock, bus master clock, and subclock (ϕSUB). The clock pulse generator consists of an oscillator, duty correction circuit, system clock select circuit, medium-speed clock divider, bus master clock select circuit, subclock input circuit, and subclock waveform forming circuit. Figure 21.1 shows a block diagram of the clock pulse generator.

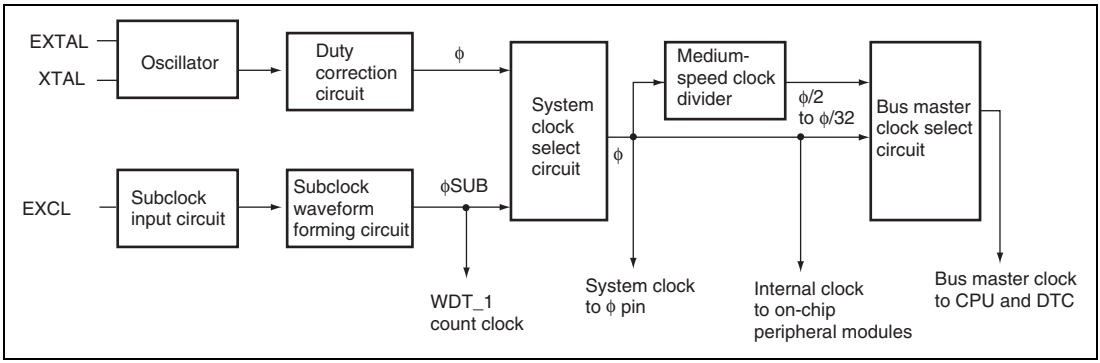


Figure 21.1 Block Diagram of Clock Pulse Generator

In high-speed mode or medium-speed mode, the bus master clock is selected by software according to the settings of the SCK2 to SCK0 bits in the standby control register (SBYCR). For details on SBYCR, see section 22.1.1, Standby Control Register (SBYCR).

The subclock input is controlled by software according to the setting of the EXCLE bit in the low power control register (LPWRCR). For details on LPWRCR, see section 22.1.2, Low-Power Control Register (LPWRCR).

21.1 Oscillator

Clock pulses can be supplied either by connecting a crystal resonator or by providing external clock input.

21.1.1 Connecting Crystal Resonator

Figure 21.2 shows a typical method for connecting a crystal resonator. An appropriate damping resistance R_d, given in table 21.1 should be used. An AT-cut parallel-resonance crystal resonator should be used.

Figure 21.3 shows an equivalent circuit of a crystal resonator. A crystal resonator having the characteristics given in table 21.2 should be used.

The frequency of the crystal resonator should be the same as that of the system clock (ϕ).

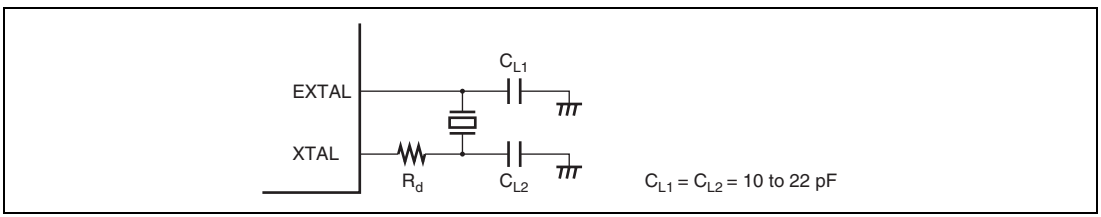


Figure 21.2 Typical Connection to Crystal Resonator

Table 21.1 Damping Resistor Values

Frequency (MHz)	8	10	12	16	20
R_d (Ω)	200	0	0	0	0

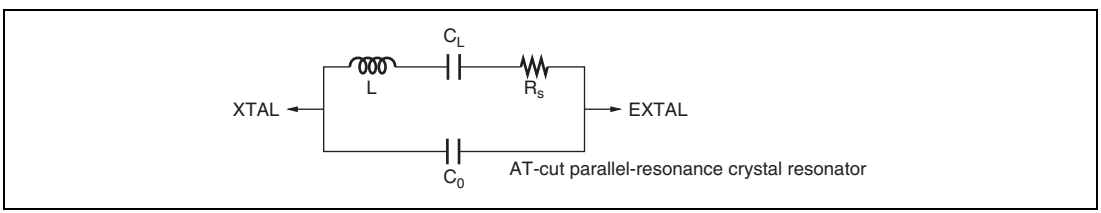


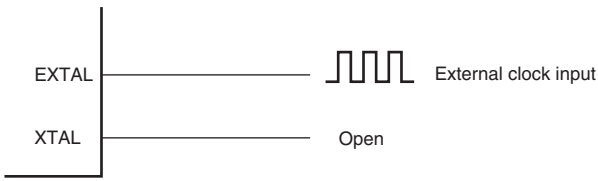
Figure 21.3 Equivalent Circuit of Crystal Resonator

Table 21.2 Crystal Resonator Parameters

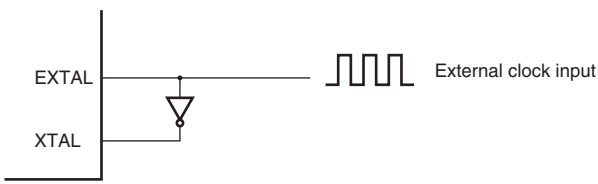
Frequency (MHz)	8	10	12	16	20
R_S (max.) (Ω)	80	70	60	50	40
C_0 (max.) (pF)	7				

21.1.2 External Clock Input Method

Figure 21.4 shows a typical method of inputting an external clock signal. To leave the XTAL pin open, incidental capacitance should be 10 pF or less. To input an inverted clock to the XTAL pin, the external clock should be set to high in standby mode, subactive mode, subsleep mode, and watch mode. External clock input conditions are shown in table 21.3. The frequency of the external clock should be the same as that of the system clock (ϕ).



(a) Example of external clock input when XTAL pin is left open



(b) Example of external clock input when an inverted clock is input to XTAL pin

Figure 21.4 Example of External Clock Input

Table 21.3 External Clock Input Conditions

		VCC = 3.0 to 3.6 V			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Test Conditions**
External clock input pulse width low level	t_{EXL}	20	—	ns	Figure 21.5
External clock input pulse width high level	t_{EXH}	20	—	ns	
External clock rising time	t_{EXr}	—	5	ns	
External clock falling time	t_{EXf}	—	5	ns	
Clock pulse width low level	t_{CL}	0.4	0.6	t_{cyc}	Figure 24.4
Clock pulse width high level	t_{CH}	0.4	0.6	t_{cyc}	

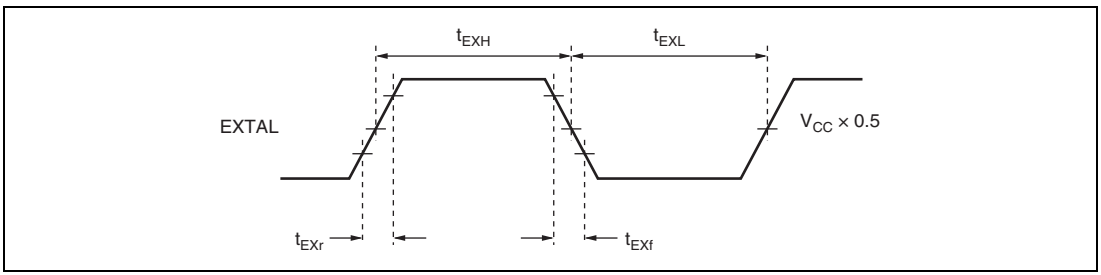


Figure 21.5 External Clock Input Timing

The oscillator and duty correction circuit can adjust the waveform of the external clock input that is input from the EXTAL pin.

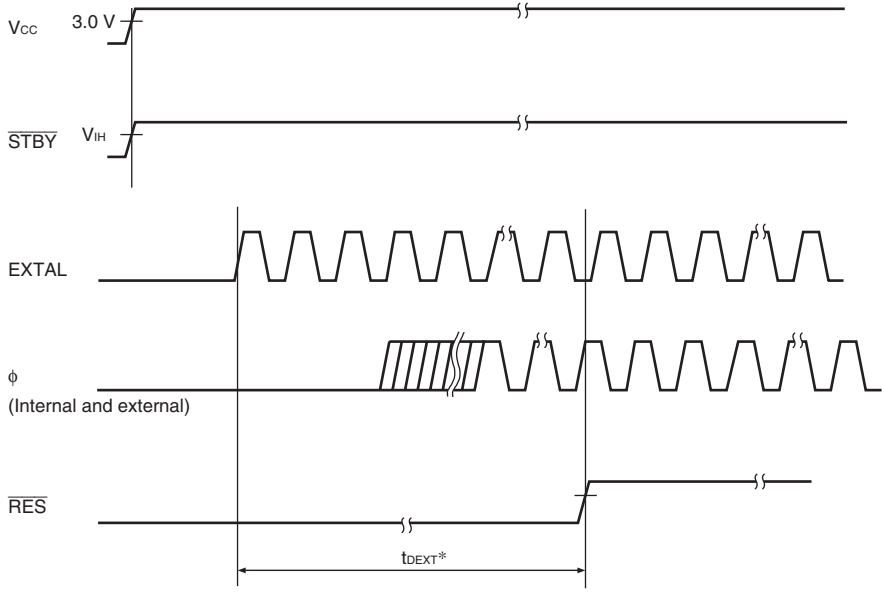
When a specified clock signal is input to the EXTAL pin, internal clock signal output is determined after the external clock output stabilization delay time (t_{DEXT}) has passed. As the clock signal output is not determined during the t_{DEXT} cycle, a reset signal should be set to low to maintain the reset state. Table 21.4 shows the external clock output stabilization delay time. Figure 21.6 shows the timing of the external clock output stabilization delay time.

Table 21.4 External Clock Output Stabilization Delay Time

Condition: VCC = 3.0 V to 3.6 V, AVCC = 3.0 V to 3.6 V, VSS = AVSS = 0 V

Item	Symbol	Min.	Max.	Unit	Remarks
External clock output stabilization delay time	t_{DEXT}*	500	—	µs	Figure 21.6

Note: * t_{DEXT} includes a $\overline{RES}$ pulse width (t_{RESW}).



Note: The external clock output stabilization delay time (t_{DEXT}) includes a $\overline{RES}$ pulse width (t_{RESW}).

Figure 21.6 Timing of External Clock Output Stabilization Delay Time

21.2 Duty Correction Circuit

The duty correction circuit generates the system clock (ϕ) by correcting the duty of the clock output from the oscillator.

21.3 Medium-Speed Clock Divider

The medium-speed clock divider divides the system clock (ϕ), and generates ϕ/2, ϕ/4, ϕ/8, ϕ/16, and ϕ/32 clocks.

21.4 Bus Master Clock Select Circuit

The bus master clock select circuit selects a clock to supply to the bus master from either the system clock (ϕ) or medium-speed clock (ϕ/2, ϕ/4, ϕ/8, ϕ/16, or ϕ/32) by the SCK2 to SCK0 bits in SBYCR.

21.5 Subclock Input Circuit

The subclock input circuit controls subclock input from the EXCL pin. To use the subclock, a 32.768-kHz external clock should be input from the EXCL pin.

Figure 21.7 shows the relationship of subclock input from the EXCL pin. When using a pin to input the subclock, specify input for the pin by clearing the DDR bit of the pin to 0. The subclock input is enabled by setting the EXCLE bit in LPWRCR to 1.

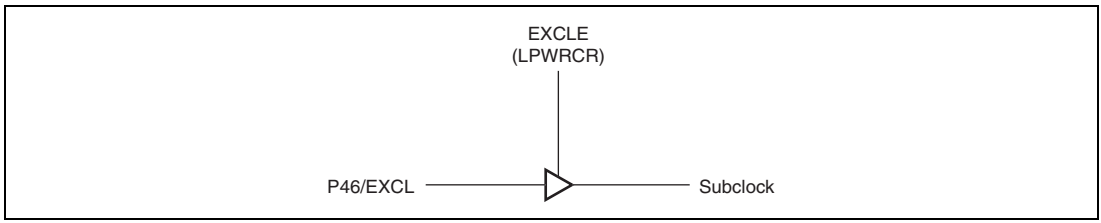


Figure 21.7 Subclock Input from EXCL Pin

Subclock input conditions are shown in table 21.5. When the subclock is not used, subclock input should not be enabled.

Table 21.5 Subclock Input Conditions

		VCC = 3.0 to 3.6 V				
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Unit**	**Test Conditions**
Subclock input pulse width low level	t_{EXCLL}	—	15.26	—	µs	Figure 21.8
Subclock input pulse width high level	t_{EXCLH}	—	15.26	—	µs	
Subclock input rising time	t_{EXCLr}	—	—	10	ns	
Subclock input falling time	t_{EXCLf}	—	—	10	ns	

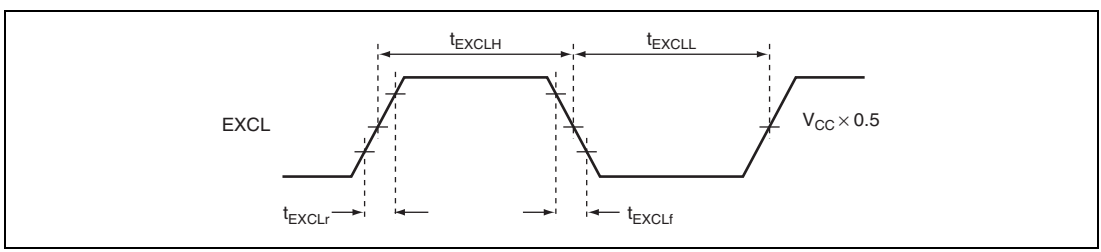


Figure 21.8 Subclock Input Timing

21.6 Subclock Waveform Forming Circuit

To remove noise from the subclock input at the EXCL pin, the subclock waveform forming circuit samples the subclock using a divided ϕ clock. The sampling frequency is set by the NESEL bit in LPWRCR.

The subclock is not sampled in subactive mode, subsleep mode, or watch mode.

21.7 Clock Select Circuit

The clock select circuit selects the system clock that is used in this LSI.

A clock generated by the oscillator to which the XTAL and EXTAL pins are connected is selected as a system clock (ϕ) when returning from high-speed mode, medium-speed mode, sleep mode, the reset state, or standby mode.

In subactive mode, subsleep mode, or watch mode, a subclock input from the EXCL pin is selected as a system clock when the EXCLE bit in LPWRCR is 1. At this time, on-chip peripheral modules such as the CPU, TMR_0, TMR_1, WDT_0, WDT_1, I/O ports, and interrupt controller and their functions operate on the ϕSUB clock. The count clock and sampling clock for each timer are divided ϕSUB clocks.

21.8 Usage Notes

21.8.1 Notes on Resonator

Since all kinds of characteristics of the resonator are closely related to the board design by the user, use the example of resonator connection in this document for only reference; be sure to use an resonator that has been sufficiently evaluated by the user. Consult with the resonator manufacturer about the resonator circuit ratings that vary depending on the stray capacitances of the resonator and installation circuit. Make sure the voltage applied to the oscillation pins do not exceed the maximum rating.

21.8.2 Notes on Board Design

When using a crystal resonator, the crystal resonator and its load capacitors should be placed as close as possible to the XTAL and EXTAL pins. Other signal lines should be routed away from the oscillator to prevent inductive interference with correct oscillation as shown in figure 21.9.

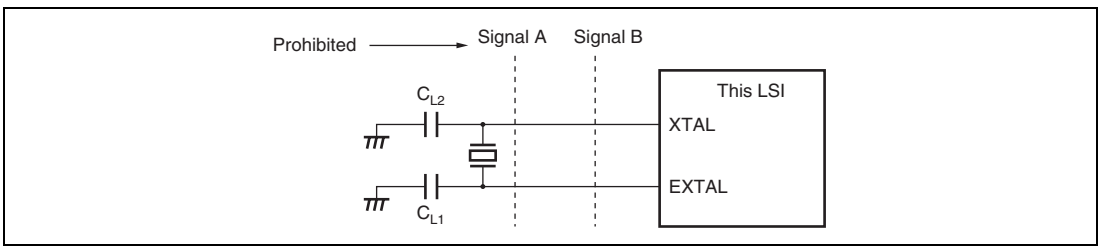


Figure 21.9 Note on Board Design of Oscillator Section

Section 22 Power-Down Modes

For operating modes after the reset state is cancelled, this LSI has not only the normal program execution state but also seven power-down modes in which power consumption is significantly reduced. In addition, there is also module stop mode in which reduced power consumption can be achieved by individually stopping on-chip peripheral modules.

- Medium-speed mode

  System clock frequency for the CPU operation can be selected as $\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, or $\phi/32$.
- Subactive mode

  The CPU operates based on the subclock, and on-chip peripheral modules TMR_0, TMR_1, WDT_0, and WDT_1 continue operating.
- Sleep mode

  The CPU stops but on-chip peripheral modules continue operating.
- Subsleep mode

  The CPU stops but on-chip peripheral modules TMR_0, TMR_1, WDT_0, and WDT_1 continue operating.
- Watch mode

  The CPU stops but on-chip peripheral module WDT_1 continue operating.
- Software standby mode

  The clock pulse generator stops, and the CPU and on-chip peripheral modules stop operating.
- Hardware standby mode

  The clock pulse generator stops, and the CPU and on-chip peripheral modules enter the reset state.
- Module stop mode

  Independently of above operating modes, on-chip peripheral modules that are not used can be stopped individually.

22.1 Register Descriptions

Power-down modes are controlled by the following registers. To access SBYCR, LPWRCR, SYSCR2, MSTPCRH, and MSTPCRL the FLSHE bit in the serial timer control register (STCR) must be cleared to 0. For details on STCR, see section 3.2.3, Serial Timer Control Register (STCR). For details on the PSS bit in TSCR_1 (WDT_1), see TCSR_1 in section 14.3.2, Timer Control/Status Register (TCSR).

- Standby control register (SBYCR)
- Low power control register (LPWRCR)
- Module stop control register H (MSTPCRH)
- Module stop control register L (MSTPCRL)
- Module stop control register A (MSTPCRA)
- Module stop control register B (MSTPCRB)

22.1.1 Standby Control Register (SBYCR)

SBYCR controls power-down modes.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby Specifies the operating mode to be entered after executing the SLEEP instruction. When the SLEEP instruction is executed in high-speed mode or medium-speed mode: 0: Shifts to sleep mode 1: Shifts to software standby mode, subactive mode, or watch mode When the SLEEP instruction is executed in subactive mode: 0: Shifts to subsleep mode 1: Shifts to watch mode or high-speed mode Note that the SSBY bit is not changed even if a mode transition occurs by an interrupt.

Bit	Bit Name	Initial Value	R/W	Description
6	STS2	0	R/W	Standby Timer Select 2 to 0
5	STS1	0	R/W	On canceling software standby mode, watch mode, or subactive mode, these bits select the wait time for clock stabilization from clock oscillation start. Select a wait time of 8 ms (oscillation stabilization time) or more, depending on the operating frequency. Table 22.1 shows the relationship between the STS2 to STS0 values and wait time. With an external clock, an arbitrary wait time can be selected. For normal cases, the minimum value is recommended.
4	STS0	0	R/W	
3	—	0	R/W	Reserved The initial value should not be changed.
2	SCK2	0	R/W	System Clock Select 2 to 0
1	SCK1	0	R/W	These bits select a clock for the bus master in high-speed mode or medium-speed mode. When making a transition to subactive mode or watch mode, these bits must be cleared to B'000. 000: High-speed mode 001: Medium-speed clock: ϕ/2 010: Medium-speed clock: ϕ/4 011: Medium-speed clock: ϕ/8 100: Medium-speed clock: ϕ/16 101: Medium-speed clock: ϕ/32 11X: Setting prohibited
0	SCK0	0	R/W	

[Legend]

X: Don't care

Table 22.1 Operating Frequency and Wait Time

STS2	STS1	STS0	Wait Time	20 MHz	10 MHz	8 MHz	Unit
0	0	0	8192 states	0.4	0.8	1.0	ms
0	0	1	16384 states	0.8	1.6	2.0	
0	1	0	32768 states	1.6	3.3	4.1	
0	1	1	65536 states	3.3	6.6	8.2	
1	0	0	131072 states	6.6	13.1	16.4	
1	0	1	262144 states	13.1	26.2	32.8	
1	1	0	Reserved	—	—	—	—
1	1	1	16 states*	0.8	1.6	2.0	µs

Recommended specification

Note: * Setting prohibited.

22.1.2 Low-Power Control Register (LPWRCR)

LPWRCR controls power-down modes.

Bit	Bit Name	Initial Value	R/W	Description
7	DTON	0	R/W	Direct Transfer On Flag Specifies the operating mode to be entered after executing the SLEEP instruction. When the SLEEP instruction is executed in high-speed mode or medium-speed mode: 0: Shifts to sleep mode, software standby mode, or watch mode 1: Shifts directly to subactive mode, or shifts to sleep mode or software standby mode When the SLEEP instruction is executed in subactive mode: 0: Shifts to subsleep mode or watch mode 1: Shifts directly to high-speed mode, or shifts to subsleep mode

Bit	Bit Name	Initial Value	R/W	Description
6	LSON	0	R/W	Low-Speed On Flag Specifies the operating mode to be entered after executing the SLEEP instruction. This bit also controls whether to shift to high-speed mode or subactive mode when watch mode is cancelled. When the SLEEP instruction is executed in high-speed mode or medium-speed mode: 0: Shifts to sleep mode, software standby mode, or watch mode 1: Shifts to watch mode or subactive mode When the SLEEP instruction is executed in subactive mode: 0: Shifts directly to watch mode or high-speed mode 1: Shifts to subsleep mode or watch mode When watch mode is cancelled: 0: Shifts to high-speed mode 1: Shifts to subactive mode
5	NESEL	0	R/W	Noise Elimination Sampling Frequency Select Selects the frequency by which the subclock (ϕSUB) input from the EXCL pin is sampled using the clock (ϕ) generated by the system clock pulse generator. 0: Sampling using ϕ/32 clock 1: Sampling using ϕ/4 clock
4	EXCLE	0	R/W	Subclock Input Enable Enables or disables subclock input from the EXCL pin. 0: Disables subclock input from the EXCL pin 1: Enables subclock input from the EXCL pin
3 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

22.1.3 Module Stop Control Registers H, L, A, and B (MSTPCRH, MSTPCRL, MSTPCRA, and MSTPCRB)

MSTPCR specifies on-chip peripheral modules to shift to module stop mode in module units. Each module can enter module stop mode by setting the corresponding bit to 1.

- MSTPCRH

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTP15	0	R/W	Reserved The initial value should not be changed.
6	MSTP14	0	R/W	Data transfer controller (DTC)
5	MSTP13	1	R/W	16-bit free-running timer (FRT)
4	MSTP12	1	R/W	8-bit timers (TMR_0 and TMR_1)
3	MSTP11	1	R/W	8-bit PWM timer (PWM), 14-bit PWM timer (PWMX)
2	MSTP10	1	R/W	Reserved The initial value should not be changed.
1	MSTP9	1	R/W	A/D converter
0	MSTP8	1	R/W	8-bit timers (TMR_X and TMR_Y)

- MSTPCRL

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTP7	1	R/W	Serial communication interface 0 (SCI_0)
6	MSTP6	1	R/W	Serial communication interface 1 (SCI_1)
5	MSTP5	1	R/W	Reserved The initial value should not be changed.
4	MSTP4	1	R/W	I^2C bus interface channel 0 (IIC_0)
3	MSTP3	1	R/W	I^2C bus interface channel 1 (IIC_1)
2	MSTP2	1	R/W	Reserved The initial value should not be changed.
1	MSTP1	1	R/W	Reserved The initial value should not be changed.
0	MSTP0	1	R/W	Reserved The initial value should not be changed.

- MSTPCRA

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTPA7	0	R/W	Reserved The initial value should not be changed.
6	MSTPA6	0	R/W	Reserved The initial value should not be changed.
5	MSTPA5	0	R/W	Reserved The initial value should not be changed.
4	MSTPA4	0	R/W	Reserved The initial value should not be changed.
3	MSTPA3	0	R/W	Reserved The initial value should not be changed.
2	MSTPA2	0	R/W	Reserved The initial value should not be changed.
1	MSTPA1	0	R/W	14-bit PWM timer (PWMX)
0	MSTPA0	0	R/W	8-bit PWM timer (PWM)

MSTPCRH and MSTPCRA set operation or stop by a combination of bits as follows:

MSTPCRH: MSTP11	MSTPCRA: MSTPA1	Function
0	0	14-bit PWM timer (PWMX) operates.
0	1	14-bit PWM timer (PWMX) stops.
1	0	14-bit PWM timer (PWMX) stops.
1	1	14-bit PWM timer (PWMX) stops.

MSTPCRH: MSTP11	MSTPCRA: MSTPA0	Function
0	0	8-bit PWM timer (PWM) operates.
0	1	8-bit PWM timer (PWM) stops.
1	0	8-bit PWM timer (PWM) stops.
1	1	8-bit PWM timer (PWM) stops.

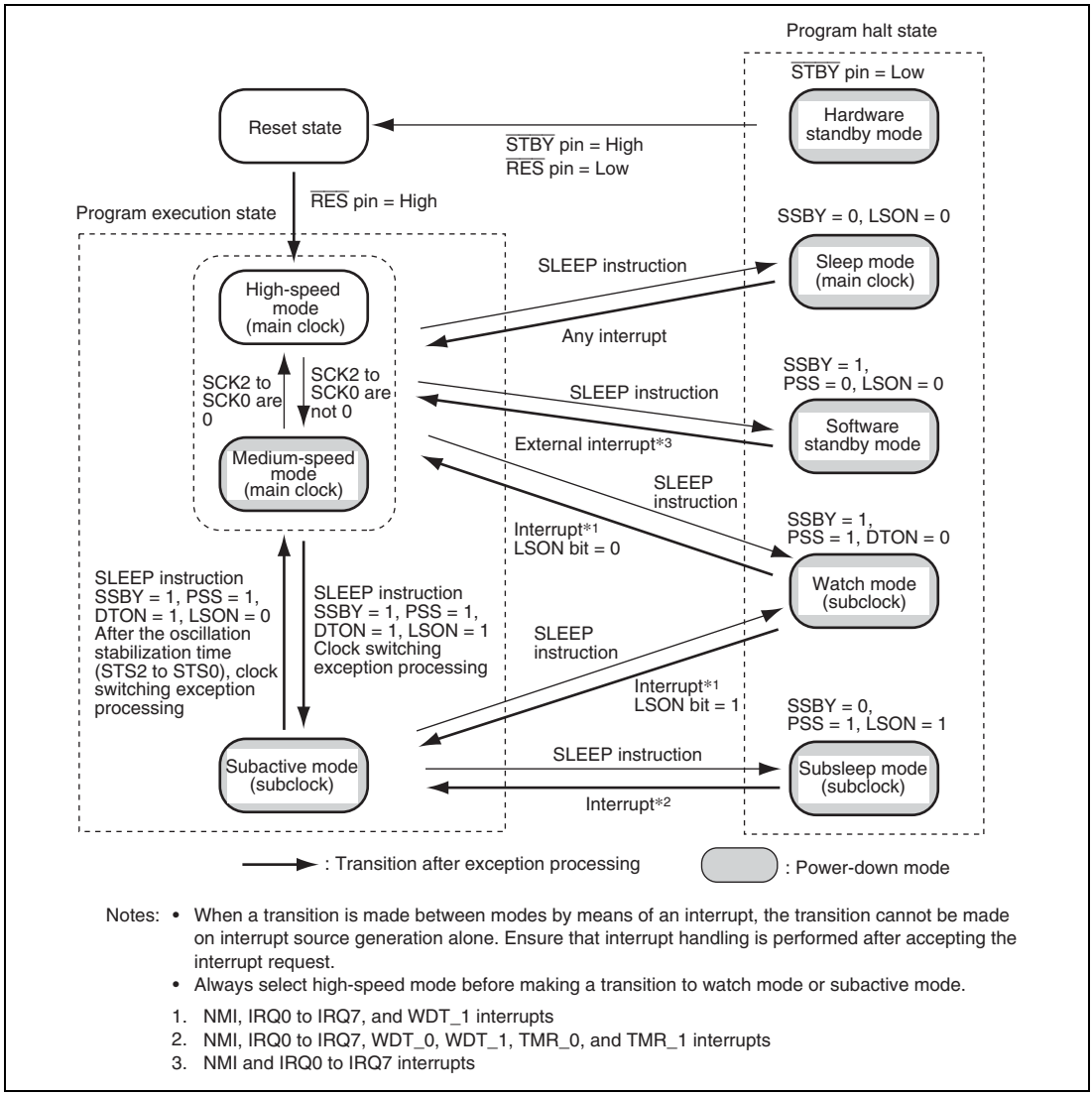
Note: The MSTP11 bit in MSTPCRH is the module stop bit of PWM and PWMX.

- MSTPCRB

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTPB7	1	R/W	Reserved The initial value should not be changed.
6	MSTPB6	1	R/W	Reserved The initial value should not be changed.
5	MSTPB5	1	R/W	Reserved The initial value should not be changed.
4	MSTPB4	1	R/W	Reserved The initial value should not be changed.
3	MSTPB3	1	R/W	Reserved The initial value should not be changed.
2	MSTPB2	1	R/W	Reserved The initial value should not be changed.
1	MSTPB1	1	R/W	16-bit cycle measurement timer 1 (TCM_1)
0	MSTPB0	1	R/W	16-bit cycle measurement timer 0 (TCM_0)

22.2 Mode Transitions and LSI States

Figure 22.1 shows the possible mode transition diagram. The mode transition from program execution state to program halt state is performed by the SLEEP instruction. The mode transition from program halt state to program execution state is performed by an interrupt. The $\overline{\text{STBY}}$ input causes a mode transition from any state to hardware standby mode. The $\overline{\text{RES}}$ input causes a mode transition from a state other than hardware standby mode to the reset state. Table 22.2 shows the LSI internal states in each operating mode.



Notes: • When a transition is made between modes by means of an interrupt, the transition cannot be made on interrupt source generation alone. Ensure that interrupt handling is performed after accepting the interrupt request.
• Always select high-speed mode before making a transition to watch mode or subactive mode.

1. NMI, IRQ0 to IRQ7, and WDT_1 interrupts
2. NMI, IRQ0 to IRQ7, WDT_0, WDT_1, TMR_0, and TMR_1 interrupts
3. NMI and IRQ0 to IRQ7 interrupts

Figure 22.1 Mode Transition Diagram

Table 22.2 LSI Internal States in Each Operating Mode

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Sub-active	Sub-sleep	Software Standby	Hardware Standby
System clock pulse generator		Functioning	Functioning	Functioning	Functioning	Halted	Halted	Halted	Halted	Halted
Subclock input		Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Halted	Halted
CPU	Instruction execution	Functioning	Functioning in medium-speed mode	Halted	Functioning	Halted	Subclock operation	Halted	Halted	Halted
	Registers			Retained		Retained		Retained	Retained	Undefined
External interrupts	NMI	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Halted
	IRQ0 to IRQ7									
On-chip peripheral modules	DTC	Functioning	Functioning in medium-speed mode/ Functioning	Functioning	Functioning/Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	WDT_1	Functioning	Functioning	Functioning	Functioning	Subclock operation	Subclock operation	Subclock operation	Halted (retained)	Halted (reset)
	WDT_0					Halted (retained)				
	TMR_0, TMR_1				Functioning/Halted (retained)					
	FRT						Halted (retained)	Halted (retained)		
	TCM									
	TMR_X, TMR_Y									
	IIC_0									
	IIC_1									

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Sub-active	Sub-sleep	Software Standby	Hardware Standby
On-chip peripheral modules	PWM	Function-ing	Function-ing	Function-ing	Function-ing/Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)
	PWMX									
	SCI_0									
	SCI_1									
	A/D converter									
	RAM	Function-ing	Function-ing	Function-ing (DTC)	Function-ing	Retained	Function-ing	Retained	Retained	Retained
	I/O	Function-ing	Function-ing	Function-ing	Function-ing	Retained	Function-ing	Function-ing	Retained	High impedance

Note: Halted (retained) means that the internal register values are retained and the internal state is operation suspended.

Halted (reset) means that the internal register values and the internal state are initialized.

In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).

22.3 Medium-Speed Mode

The CPU makes a transition to medium-speed mode as soon as the current bus cycle ends according to the setting of the SCK2 to SCK0 bits in SBYCR. In medium-speed mode, the operating clock can be selected from $\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, or $\phi/32$. On-chip peripheral modules other than the bus masters operate on the system clock (ϕ).

In medium-speed mode, a bus access is executed in the specified number of states with respect to the bus master operating clock. For example, if $\phi/4$ is selected as the operating clock, on-chip memory is accessed in four states, and internal I/O registers in eight states.

By clearing all of bits SCK2 to SCK0 to 0 in medium-speed mode, a transition is made to high-speed mode at the end of the current bus cycle.

When the SLEEP instruction is executed with the SSBY bit in SBYCR cleared to 0 and the LSON bit in LPWRCR cleared to 0, a transition is made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode is restored. When the SLEEP instruction is executed with the SSBY bit in SBYCR set to 1, the LSON bit in LPWRCR cleared to 0, and the PSS bit in TCSR (WDT_1) cleared to 0, a transition is made to software standby mode. When software standby mode is cleared by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is driven low, medium-speed mode is cancelled and a transition is made to the reset state. The same applies in the case of a reset caused by overflow of the watchdog timer.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 22.2 shows an example of medium-speed mode timing.

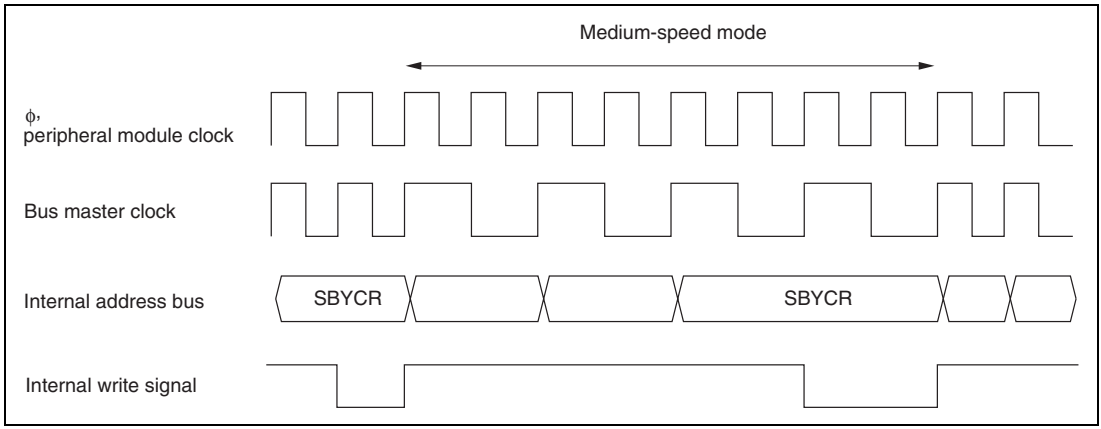


Figure 22.2 Medium-Speed Mode Timing

22.4 Sleep Mode

The CPU makes a transition to sleep mode if the SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0 and the LSON bit in LPWRCR is cleared to 0. In sleep mode, CPU operation stops but the on-chip peripheral modules do not. The contents of the CPU's internal registers are retained.

Sleep mode is cleared by any interrupt, the $\overline{\text{RES}}$ pin input, or the $\overline{\text{STBY}}$ pin input.

When an interrupt occurs, sleep mode is cleared and interrupt exception handling starts. Sleep mode is not cleared if the interrupt is disabled, or interrupts other than NMI have been masked by the CPU.

When the $\overline{\text{RES}}$ pin is driven low and sleep mode is cleared, a transition is made to the reset state. After the specified reset input time has elapsed, driving the $\overline{\text{RES}}$ pin high causes the CPU to start reset exception handling.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

22.5 Software Standby Mode

The CPU makes a transition to software standby mode when the SLEEP instruction is executed with the SSBY bit in SBYCR set to 1, the LSON bit in LPWRCR cleared to 0, and the PSS bit in TCSR (WDT_1) cleared to 0. In software standby mode, the CPU, on-chip peripheral modules, and clock pulse generator all stop. However, the contents of the CPU registers, on-chip RAM data, I/O ports, and the states of on-chip peripheral modules other than the SCI, PWM, PWMX, and A/D converter are retained as long as the prescribed voltage is supplied.

Software standby mode is cleared by an external interrupt (NMI, IRQ0 to IRQ7), $\overline{\text{RES}}$ pin input, or $\overline{\text{STBY}}$ pin input.

When an external interrupt request signal is input, system clock oscillation starts, and after the elapse of the time set in bits STS2 to STS0 in SBYCR, software standby mode is cleared, and interrupt exception handling is started. When clearing software standby mode with an IRQ0 to IRQ7 interrupt, set the corresponding enable bit to 1. In the case of an IRQ0 to IRQ7 interrupt, software standby mode is not cleared if the corresponding enable bit is cleared to 0 or if the interrupt has been masked by the CPU.

When the $\overline{\text{RES}}$ pin is driven low, the clock pulse generator starts oscillation. Simultaneously with the start of system clock oscillation, the system clock is supplied to the entire LSI. Note that the $\overline{\text{RES}}$ pin must be held low until clock oscillation is stabilized. If the $\overline{\text{RES}}$ pin is driven high after the clock oscillation stabilization time has elapsed, the CPU starts reset exception handling.

When the $\overline{\text{STBY}}$ pin is driven low, software standby mode is cleared and a transition is made to hardware standby mode.

Figure 22.3 shows an example in which a transition is made to software standby mode at the falling edge of the NMI pin, and software standby mode is cleared at the rising edge of the NMI pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in SYSCR cleared to 0 (falling edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY bit is set to 1, and a SLEEP instruction is executed, causing a transition to software standby mode.

Software standby mode is then cleared at the rising edge of the NMI pin.

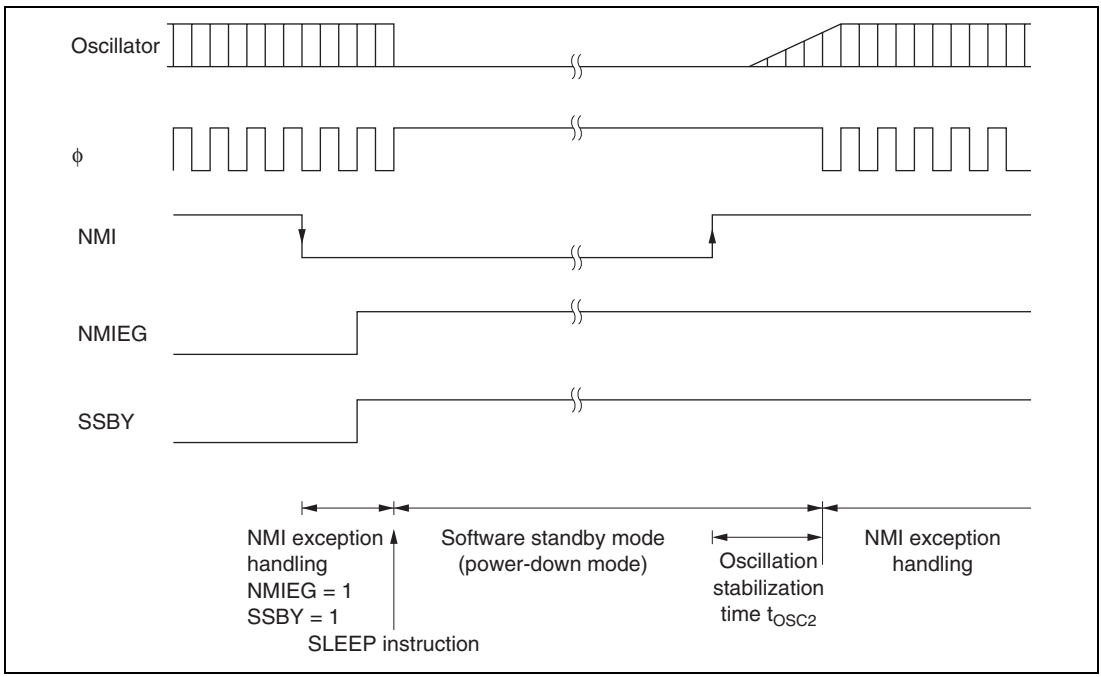


Figure 22.3 Software Standby Mode Application Example

22.6 Hardware Standby Mode

The CPU makes a transition to hardware standby mode from any mode when the $\overline{\text{STBY}}$ pin is driven low.

In hardware standby mode, all functions enter the reset state. As long as the prescribed voltage is supplied, on-chip RAM data is retained. The I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 before driving the $\overline{\text{STBY}}$ pin low. Do not change the state of the mode pins (MD2*, MD1, and MD0) while this LSI is in hardware standby mode.

Hardware standby mode is cleared by the $\overline{\text{STBY}}$ pin input or the $\overline{\text{RES}}$ pin input.

When the $\overline{\text{STBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, the clock pulse generator starts oscillation. Ensure that the $\overline{\text{RES}}$ pin is held low until system clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin is subsequently driven high after the clock oscillation stabilization time has elapsed, reset exception handling starts.

Figure 22.4 shows an example of hardware standby mode timing.

Note: * MD2 is not supported in SDIP-64 and QFP-64.

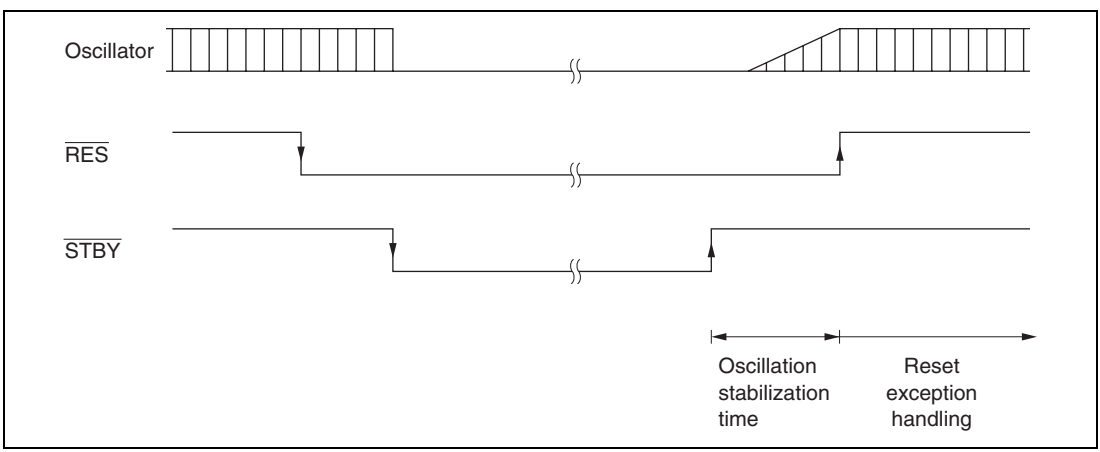


Figure 22.4 Hardware Standby Mode Timing

22.7 Watch Mode

The CPU makes a transition to watch mode when the SLEEP instruction is executed in high-speed mode or subactive mode with the SSBY bit in SBYCR set to 1, the DTON bit in LPWRCR cleared to 0, and the PSS bit in TCSR (WDT_1) set to 1.

In watch mode, the CPU is stopped and on-chip peripheral modules other than WDT_1 are also stopped. The contents of the CPU's internal registers, several on-chip peripheral module registers, and on-chip RAM data are retained and the I/O ports retain their values before transition as long as the prescribed voltage is supplied.

Watch mode is cleared by an interrupt (WOVI1, NMI, IRQ0 to IRQ7), $\overline{\text{RES}}$ pin input, or $\overline{\text{STBY}}$ pin input.

When an interrupt occurs, watch mode is cleared and a transition is made to high-speed mode or medium-speed mode when the LSON bit in LPWRCR cleared to 0, or a transition is made to subactive mode when the LSON bit is set to 1. When a transition is made to high-speed mode, a stable clock is supplied to the entire LSI and interrupt exception handling starts after the time set in the STS2 to STS0 bits in SBYCR has elapsed. In the case of an IRQ0 to IRQ7 interrupt, watch mode is not cleared if the corresponding enable bit has been cleared to 0 or the interrupt has been masked by the CPU. In the case of an interrupt from an on-chip peripheral module, watch mode is not cleared if the interrupt enable register has been set to disable the reception of that interrupt or the interrupt has been masked by the CPU.

When the $\overline{\text{RES}}$ pin is driven low, the clock pulse generator starts oscillation. Simultaneously with the start of system clock oscillation, the system clock is supplied to the entire LSI. Note that the $\overline{\text{RES}}$ pin must be held low until clock oscillation is stabilized. If the $\overline{\text{RES}}$ pin is driven high after the clock oscillation stabilization time has elapsed, the CPU starts reset exception handling.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

22.8 Subsleep Mode

The CPU makes a transition to subsleep mode when the SLEEP instruction is executed in subactive mode with the SSBY bit in SBYCR cleared to 0, the LSON bit in LPWRCR set to 1, and the PSS bit in TCSR (WDT_1) set to 1.

In subsleep mode, the CPU is stopped. On-chip peripheral modules other than TMR_0, TMR_1, WDT_0, and WDT_1 are also stopped. The contents of the CPU registers, several on-chip peripheral module registers, and on-chip RAM data are retained and the I/O ports retain their values before transition as long as the prescribed voltage is supplied.

Subsleep mode is cleared by an interrupt (interrupts by on-chip peripheral modules, NMI, IRQ0 to IRQ7), $\overline{\text{RES}}$ pin input, or $\overline{\text{STBY}}$ pin input.

When an interrupt occurs, subsleep mode is cleared and interrupt exception handling starts.

In the case of an IRQ0 to IRQ7 interrupt, subsleep mode is not cleared if the corresponding enable bit has been cleared to 0 or the interrupt has been masked by the CPU. In the case of an interrupt from an on-chip peripheral module, subsleep mode is not cleared if the interrupt enable register has been set to disable the reception of that interrupt or the interrupt has been masked by the CPU.

When the $\overline{\text{RES}}$ pin is driven low, the clock pulse generator starts oscillation. Simultaneously with the start of system clock oscillation, the system clock is supplied to the entire LSI. Note that the $\overline{\text{RES}}$ pin must be held low until clock oscillation is stabilized. If the $\overline{\text{RES}}$ pin is driven high after the clock oscillation stabilization time has elapsed, the CPU starts reset exception handling.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

22.9 Subactive Mode

The CPU makes a transition to subactive mode when the SLEEP instruction is executed in high-speed mode with the SSBY bit in SBYCR set to 1, the DTON bit and LSON bit in LPWRCR both set to 1, and the PSS bit in TCSR (WDT_1) set to 1. When an interrupt occurs in watch mode with the LSON bit in LPWRCR set to 1, a direct transition is made to subactive mode. Similarly, if an interrupt occurs in subsleep mode, a transition is made to subactive mode.

In subactive mode, the CPU operates at a low speed based on the subclock and sequentially executes programs. On-chip peripheral modules other than TMR_0, TMR_1, WDT_0, and WDT_1 are also stopped.

When operating the CPU in subactive mode, the SCK2 to SCK0 bits in SBYCR must all be cleared to 0.

Subactive mode is cleared by the SLEEP instruction, $\overline{\text{RES}}$ pin input, or $\overline{\text{STBY}}$ pin input.

When the SLEEP instruction is executed with the SSBY bit in SBYCR set to 1, the DTON bit in LPWRCR cleared to 0, and the PSS bit in TCSR (WDT_1) set to 1, subactive mode is cleared and a transition is made to watch mode. When the SLEEP instruction is executed with the SSBY bit in SBYCR cleared to 0, the LSON bit in LPWRCR set to 1, and the PSS bit in TCSR (WDT_1) set to 1, a transition is made to subsleep mode. When the SLEEP instruction is executed with the SSBY bit in SBYCR set to 1, the DTON bit in LPWRCR set to 1, the LSON bit in LPWRCR cleared to 0, and the PSS bit in TCSR (WDT_1) set to 1, a direct transition is made to high-speed mode.

For details on direct transitions, see section 22.11, Direct Transitions.

When the $\overline{\text{RES}}$ pin is driven low, the clock pulse generator starts oscillation. Simultaneously with the start of system clock oscillation, the system clock is supplied to the entire LSI. Note that the $\overline{\text{RES}}$ pin must be held low until clock oscillation is stabilized. If the $\overline{\text{RES}}$ pin is driven high after the clock oscillation stabilization time has elapsed, the CPU starts reset exception handling.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

22.10 Module Stop Mode

Module stop mode can be individually set for each on-chip peripheral module.

When the corresponding MSTP bit in MSTPCR is set to 1, module operation stops at the end of the bus cycle and a transition is made to module stop mode. In turn, when the corresponding MSTP bit is cleared to 0, module stop mode is cleared and module operation resumes at the end of the bus cycle. In module stop mode, the internal states of on-chip peripheral modules other than the SCI, PWM, PWMX, and A/D converter are retained.

After the reset state is cancelled, all on-chip peripheral modules other than the DTC are in module stop mode.

While an on-chip peripheral module is in module stop mode, its registers cannot be read from or written to.

22.11 Direct Transitions

The CPU executes programs in three modes: high-speed, medium-speed, and subactive. When a direct transition is made from high-speed mode to subactive mode and vice versa, there is no interruption of program execution. A direct transition is enabled by executing the SLEEP instruction after setting the DTON bit in LPWRCR to 1. After a transition, direct transition exception handling starts.

When the SLEEP instruction is executed in high-speed mode with the SSBY bit in SBYCR set to 1, the LSON bit and DTON bit in LPWRCR both set to 1, and the PSS bit in TSCR (WDT_1) set to 1, the CPU makes a direct transition to subactive mode.

When the SLEEP instruction is executed in subactive mode with the SSBY bit in SBYCR set to 1, the LSON bit in LPWRCR cleared to 0, the DTON bit in LPWRCR set to 1, and the PSS bit in TSCR (WDT_1) set to 1, after the time set in the STS2 to STS0 bits in SBYCR has elapsed, the CPU makes a direct transition to high-speed mode.

22.12 Usage Notes

22.12.1 I/O Port Status

The status of the I/O ports is retained in software standby mode. Therefore, while a high level is output or the pull-up MOS is on, the current consumption is not reduced by the amount of current to support the high level output.

22.12.2 Current Consumption when Waiting for Oscillation Stabilization

The current consumption increases during oscillation stabilization.

22.12.3 DTC Module Stop Mode

If the DTC module stop mode specification and DTC bus request occur simultaneously, the bus is released to the DTC and the MSTP bit cannot be set to 1.

After completing the DTC bus cycle, set the MSTP bit to 1 again.

Section 23 List of Registers

The list of registers gives information on the on-chip register addresses, how the register bits are configured, the register states in each operating mode, the register selection condition, and the register address of each module. The information is given as shown below.

1. Register addresses (address order)
- Registers are listed from the lower allocation addresses.
- For the addresses of 16 bits, the MSB is described.
- Registers are classified by functional modules.
- The access size 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.
- The bit number in the bit-name column indicates that the whole register is allocated as a counter or for holding data.
- Each line covers eight bits, and 16-bit register is shown as 2 lines, respectively.

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 module, see the section on that on-chip module.

4. Register selection conditions
- Register selection conditions are described in the same order as the register addresses.
- Register selection conditions are described in section 3.2.2, System Control Register (SYSCR), section 3.2.3, Serial Timer Control Register (STCR), section 21.1.3, Module Stop Control Register H, L, A, and B (MSTPCRH, MSTPCRL, MSTPCRA, and MSTPCRB), or register descriptions for each module.

5. Register addresses (classification by type of module)
- The register addresses are described by modules.
- The register addresses are described in channel order when the module has multiple channels.

23.1 Register Addresses (Address Order)

The data bus width indicates the numbers 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	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
TCM timer counter_0	TCMCNT_0	16	H'FBC0	TCM_0	16	2
TCM cycle limit register_0	TCMMLCM_0	16	H'FBC2	TCM_0	16	2
TCM input capture register_0	TCMICR_0	16	H'FBC4	TCM_0	16	2
TCM input capture buffer register_0	TCMICRF_0	16	H'FBC6	TCM_0	16	2
TCM status register_0	TCMCSR_0	8	H'FBC8	TCM_0	8	2
TCM control register_0	TCMCR_0	8	H'FBC9	TCM_0	8	2
TCM interrupt enable register_0	TCMIER_0	8	H'FBCA	TCM_0	8	2
TCM timer counter_1	TCMCNT_1	16	H'FBD0	TCM_1	16	2
TCM cycle limit register_1	TCMMLCM_1	16	H'FBD2	TCM_1	16	2
TCM input capture register_1	TCMICR_1	16	H'FBD4	TCM_1	16	2
TCM input capture buffer register_1	TCMICRF_1	16	H'FBD6	TCM_1	16	2
TCM status register_1	TCMCSR_1	8	H'FBD8	TCM_1	8	2
TCM control register_1	TCMCR_1	8	H'FBD9	TCM_1	8	2
TCM interrupt enable register_1	TCMIER_1	8	H'FBDA	TCM_1	8	2
Port 6 noise canceller enable register	P6NCE	8	H'FE00	PORT	8	2
Port 6 noise canceller mode control register	P6NCMC	8	H'FE01	PORT	8	2
Port 6 noise cancel cycle setting register	P6NCCS	8	H'FE02	PORT	8	2
Port 4 noise canceller enable register	P4NCE	8	H'FE09	PORT	8	2
Port 4 noise canceller mode control register	P4NCMC	8	H'FE0A	PORT	8	2
Port 4 noise cancel cycle setting register	P4NCCS	8	H'FE0B	PORT	8	2
Module stop control register A	MSTPCRA	8	H'FE7E	SYSTEM	8	2
Module stop control register B	MSTPCRB	8	H'FE7F	SYSTEM	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Interrupt control register D	ICRD	8	H'FE87	INT	8	2
Flash code control status register	FCCS	8	H'FEA8	ROM	8	2
Flash program code select register	FPCS	8	H'FEA9	ROM	8	2
Flash erase code select register	FECS	8	H'FEAA	ROM	8	2
Flash key code register	FKEY	8	H'FEAC	ROM	8	2
Flash MAT select register	FMATS	8	H'FEAD	ROM	8	2
Flash transfer destination address register	FTDAR	8	H'FEAE	ROM	8	2
Timer XY control register	TCRXY	8	H'FEC6	TMR_XY	8	2
I^2C bus extended control register_0	ICXR_0	8	H'FED4	IIC_0	8	2
I^2C bus extended control register_1	ICXR_1	8	H'FED5	IIC_1	8	2
DDC switch register	DDCSWR	8	H'FEE6	IIC_0, IIC_1	8	2
Interrupt control register A	ICRA	8	H'FEE8	INT	8	2
Interrupt control register B	ICRB	8	H'FEE9	INT	8	2
Interrupt control register C	ICRC	8	H'FEEA	INT	8	2
IRQ status register	ISR	8	H'FEEB	INT	8	2
IRQ sense control register H	ISCRH	8	H'FEEC	INT	8	2
IRQ sense control register L	ISCRL	8	H'FEED	INT	8	2
DTC enable register A	DTCERA	8	H'FEEE	DTC	8	2
DTC enable register B	DTCERB	8	H'FEEF	DTC	8	2
DTC enable register C	DTCERC	8	H'FEF0	DTC	8	2
DTC enable register D	DTCERD	8	H'FEF1	DTC	8	2
DTC enable register E	DTCERE	8	H'FEF2	DTC	8	2
DTC vector register	DTVECR	8	H'FEF3	DTC	8	2
Address break control register	ABRKCR	8	H'FEF4	INT	8	2
Break address register A	BARA	8	H'FEF5	INT	8	2
Break address register B	BARB	8	H'FEF6	INT	8	2
Break address register C	BARC	8	H'FEF7	INT	8	2
Peripheral clock select register	PCSR	8	H'FF82	PWM, PWMX	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Standby control register	SBYCR	8	H'FF84	SYSTEM	8	2
Low power control register	LPWRCR	8	H'FF85	SYSTEM	8	2
Module stop control register H	MSTPCRH	8	H'FF86	SYSTEM	8	2
Module stop control register L	MSTPCRL	8	H'FF87	SYSTEM	8	2
Serial mode register_1	SMR_1	8	H'FF88	SCI_1	8	2
I^2C bus control register_1	ICCR_1	8	H'FF88	IIC_1	8	2
Bit rate register_1	BRR_1	8	H'FF89	SCI_1	8	2
I^2C bus status register_1	ICSR_1	8	H'FF89	IIC_1	8	2
Serial control register_1	SCR_1	8	H'FF8A	SCI_1	8	2
Transmit data register_1	TDR_1	8	H'FF8B	SCI_1	8	2
Serial status register_1	SSR_1	8	H'FF8C	SCI_1	8	2
Receive data register_1	RDR_1	8	H'FF8D	SCI_1	8	2
Serial interface mode register_1	SCMR_1	8	H'FF8E	SCI_1	8	2
I^2C bus data register_1	ICDR_1	8	H'FF8E	IIC_1	8	2
Second slave address register_1	SARX_1	8	H'FF8E	IIC_1	8	2
I^2C bus mode register_1	ICMR_1	8	H'FF8F	IIC_1	8	2
Slave address register_1	SAR_1	8	H'FF8F	IIC_1	8	2
Timer interrupt enable register	TIER	8	H'FF90	FRT	8	2
Timer control/status register	TCSR	8	H'FF91	FRT	8	2
Free-running counter	FRC	16	H'FF92	FRT	16	2
Output control register A	OCRA	16	H'FF94	FRT	16	2
Output control register B	OCRB	16	H'FF94	FRT	16	2
Timer control register	TCR	8	H'FF96	FRT	8	2
Timer output compare control register	TOCR	8	H'FF97	FRT	8	2
Input capture register A	ICRA	16	H'FF98	FRT	16	2
Output control register AR	OCRAR	16	H'FF98	FRT	16	2
Input capture register B	ICRB	16	H'FF9A	FRT	16	2
Output control register AF	OCRAF	16	H'FF9A	FRT	16	2
Input capture register C	ICRC	16	H'FF9C	FRT	16	2
Output compare register DM	OCRDM	16	H'FF9C	FRT	16	2
Input capture register D	ICRD	16	H'FF9E	FRT	16	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
PWMX (D/A) control register	DACR	8	H'FFA0	PWMX	8	2
PWMX (D/A) data register AH	DADRAH	8	H'FFA0	PWMX	8	2
PWMX (D/A) data register AL	DADRAL	8	H'FFA1	PWMX	8	2
PWMX (D/A) counter H	DACNTH	8	H'FFA6	PWMX	8	2
PWMX (D/A) data register BH	DADRBH	8	H'FFA6	PWMX	8	2
PWMX (D/A) counter L	DACNTL	8	H'FFA7	PWMX	8	2
PWMX (D/A) data register BL	DADRBL	8	H'FFA7	PWMX	8	2
Timer control/status register_0	TCSR_0	8	H'FFA8 (write)	WDT_0	16	2
Timer control/status register_0	TCSR_0	8	H'FFA8 (read)	WDT_0	8	2
Timer counter_0	TCNT_0	8	H'FFA8 (write)	WDT_0	16	2
Timer counter_0	TCNT_0	8	H'FFA9 (read)	WDT_0	8	2
Port 1 pull-up MOS control register	P1PCR	8	H'FFAC	PORT	8	2
Port 2 pull-up MOS control register	P2PCR	8	H'FFAD	PORT	8	2
Port 3 pull-up MOS control register	P3PCR	8	H'FFAE	PORT	8	2
Port 1 data direction register	P1DDR	8	H'FFB0	PORT	8	2
Port 2 data direction register	P2DDR	8	H'FFB1	PORT	8	2
Port 1 data register	P1DR	8	H'FFB2	PORT	8	2
Port 2 data register	P2DR	8	H'FFB3	PORT	8	2
Port 3 data direction register	P3DDR	8	H'FFB4	PORT	8	2
Port 4 data direction register	P4DDR	8	H'FFB5	PORT	8	2
Port 3 data register	P3DR	8	H'FFB6	PORT	8	2
Port 4 data register	P4DR	8	H'FFB7	PORT	8	2
Port 5 data direction register	P5DDR	8	H'FFB8	PORT	8	2
Port 6 data direction register	P6DDR	8	H'FFB9	PORT	8	2
Port 5 data register	P5DR	8	H'FFBA	PORT	8	2
Port 6 data register	P6DR	8	H'FFBB	PORT	8	2
Port 7 input data register	P7PIN	8	H'FFBE	PORT	8	2
Interrupt enable register	IER	8	H'FFC2	INT	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Serial timer control register	STCR	8	H'FFC3	SYSTEM	8	2
System control register	SYSCR	8	H'FFC4	SYSTEM	8	2
Mode control register	MDCR	8	H'FFC5	SYSTEM	8	2
Bus control register	BCR	8	H'FFC6	BSC	8	2
Wait state control register	WSCR	8	H'FFC7	BSC	8	2
Timer control register_0	TCR_0	8	H'FFC8	TMR_0	8	2
Timer control register_1	TCR_1	8	H'FFC9	TMR_1	8	2
Timer control/status register_0	TCSR_0	8	H'FFCA	TMR_0	8	2
Timer control/status register_1	TCSR_1	8	H'FFCB	TMR_1	16	2
Time constant register A_0	TCORA_0	8	H'FFCC	TMR_0	16	2
Time constant register A_1	TCORA_1	8	H'FFCD	TMR_1	16	2
Time constant register B_0	TCORB_0	8	H'FFCE	TMR_0	16	2
Time constant register B_1	TCORB_1	8	H'FFCF	TMR_1	16	2
Timer counter_0	TCNT_0	8	H'FFD0	TMR_0	16	2
Timer counter_1	TCNT_1	8	H'FFD1	TMR_1	16	2
PWM output enable register B	PWOERB	8	H'FFD2	PWM	8	2
PWM output enable register A	PWOERA	8	H'FFD3	PWM	8	2
PWM data polarity register B	PWDPRB	8	H'FFD4	PWM	8	2
PWM data polarity register A	PWDPRA	8	H'FFD5	PWM	8	2
PWM register select	PWSL	8	H'FFD6	PWM	8	2
PWM data register 15 to 0	PWDR15 to 0	8	H'FFD7	PWM	8	2
Serial mode register_0	SMR_0	8	H'FFD8	SCI_0	8	2
I^2C bus control register_0	ICCR_0	8	H'FFD8	IIC_0	8	2
Bit rate register_0	BRR_0	8	H'FFD9	SCI_0	8	2
I^2C bus status register_0	ICSR_0	8	H'FFD9	IIC_0	8	2
Serial control register_0	SCR_0	8	H'FFDA	SCI_0	8	2
Transmit data register_0	TDR_0	8	H'FFDB	SCI_0	8	2
Serial status register_0	SSR_0	8	H'FFDC	SCI_0	8	2
Receive data register_0	RDR_0	8	H'FFDD	SCI_0	8	2
Serial interface mode register_0	SCMR_0	8	H'FFDE	SCI_0	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
I^2C bus data register_0	ICDR_0	8	H'FFDE	IIC_0	8	2
Second slave address register_0	SARX_0	8	H'FFDE	IIC_0	8	2
I^2C bus mode register_0	ICMR_0	8	H'FFDF	IIC_0	8	2
Slave address register_0	SAR_0	8	H'FFDF	IIC_0	8	2
A/D data register AH	ADDRAH	8	H'FFE0	A/D Converter	8	2
A/D data register AL	ADDRAL	8	H'FFE1	A/D Converter	8	2
A/D data register BH	ADDRBH	8	H'FFE2	A/D Converter	8	2
A/D data register BL	ADDRBL	8	H'FFE3	A/D Converter	8	2
A/D data register CH	ADDRCH	8	H'FFE4	A/D Converter	8	2
A/D data register CL	ADDRCL	8	H'FFE5	A/D Converter	8	2
A/D data register DH	ADDRDH	8	H'FFE6	A/D Converter	8	2
A/D data register DL	ADDRDL	8	H'FFE7	A/D Converter	8	2
A/D control/status register	ADCSR	8	H'FFE8	A/D Converter	8	2
A/D control register	ADCR	8	H'FFE9	A/D Converter	8	2
Timer control/status register_1	TCSR_1	8	H'FFEA (write)	WDT_1	16	2
Timer control/status register_1	TCSR_1	8	H'FFEA (read)	WDT_1	8	2
Timer counter_1	TCNT_1	8	H'FFEA (write)	WDT_1	16	2
Timer counter_1	TCNT_1	8	H'FFEB (read)	WDT_1	8	2
Timer control register_X	TCR_X	8	H'FFF0	TMR_X	8	2
Timer control register_Y	TCR_Y	8	H'FFF0	TMR_Y	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Timer control/status register_X	TCSR_X	8	H'FFF1	TMR_X	8	2
Timer control/status register_Y	TCSR_Y	8	H'FFF1	TMR_Y	8	2
Input capture register R	TICRR	8	H'FFF2	TMR_X	8	2
Time constant register A_Y	TCORA_Y	8	H'FFF2	TMR_Y	8	2
Input capture register F	TICRF	8	H'FFF3	TMR_X	8	2
Time constant register B_Y	TCORB_Y	8	H'FFF3	TMR_Y	8	2
Timer counter_X	TCNT_X	8	H'FFF4	TMR_X	8	2
Timer counter_Y	TCNT_Y	8	H'FFF4	TMR_Y	8	2
Time constant register C	TCORC	8	H'FFF5	TMR_X	8	2
Time constant register A_X	TCORA_X	8	H'FFF6	TMR_X	8	2
Time constant register B_X	TCORB_X	8	H'FFF7	TMR_X	8	2
Timer connection register I	TCONRI	8	H'FFFC	TMR_X	8	2
Timer connection register S	TCONRS	8	H'FFFE	TMR_X, TMR_Y	8	2

23.2 Register Bits

Register addresses and bit names of the on-chip peripheral modules are described below.

Each line covers eight bits, and 16-bit registers are shown as 2 lines.

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCMCNT_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TCM_0
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCMMLCM_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCMICR_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCMICRF_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCMCSR_0	OVF	MAXOVF	CMF	CKSEG	ICPF	—	—	—	
TCMCR_0	CST	POCTL	CPSPE	IEDG	TCMMDS	CKS2	CKS1	CKS0	
TCMIER_0	OVIE	MAXOVIE	CMIE	TCMIPE	ICPIE	—	—	—	
TCMCNT_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TCM_1
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCMMLCM_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCMICR_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCMICRF_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCMCSR_1	OVF	MAXOVF	CMF	CKSEG	ICPF	—	—	—	
TCMCR_1	CST	POCTL	CPSPE	IEDG	TCMMDS	CKS2	CKS1	CKS0	
TCMIER_1	OVIE	MAXOVIE	CMIE	TCMIPE	ICPIE	—	—	—	
P6NCE	P67NCE	P66NCE	P65NCE	P64NCE	P63NCE	P62NCE	P61NCE	P60NCE	PORT
P6NCMC	P67NCMC	P66NCMC	P65NCMC	P64NCMC	P63NCMC	P62NCMC	P61NCMC	P60NCMC	
P6NCCS	—	—	—	—	—	P6NCCK2	P6NCCK1	P6NCCK0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
P4NCE	P47NCE	P46NCE	P45NCE	P44NCE	P43NCE	P42NCE	P41NCE	P40NCE	PORT
P4NCMC	P47NCMC	P46NCMC	P45NCMC	P44NCMC	P43NCMC	P42NCMC	P41NCMC	P40NCMC	
P4NCCS	—	—	—	—	—	P4NCCK2	P4NCCK1	P4NCCK0	
MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0	SYSTEM
MSTPCRB	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0	
ICRD	ICRD7	ICRD6	ICRD5	ICRD4	ICRD3	ICRD2	ICRD1	ICRD0	INT
FCCS	FWE	—	—	FLER	—	—	—	SCO	ROM
FPCS	—	—	—	—	—	—	—	PPVS	
FECS	—	—	—	—	—	—	—	EPVB	
FKEY	K7	K6	K5	K4	K3	K2	K1	K0	
FMATS	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0	
FTDAR	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0	
TCRXY	—	—	CKSX	CKSY	—	—	—	—	TMR_XY
ICXR_0	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0	IIC_0
ICXR_1	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0	IIC_1
DDCSWR	—	—	—	—	CLR3	CLR2	CLR1	CLR0	IIC_0, IIC_1
ICRA	ICRA7	ICRA6	ICRA5	ICRA4	ICRA3	ICRA2	ICRA1	ICRA0	INT
ICRB	ICRB7	ICRB6	ICRB5	ICRB4	ICRB3	ICRB2	ICRB1	ICRB0	
ICRC	ICRC7	ICRC6	ICRC5	ICRC4	ICRC3	ICRC2	ICRC1	ICRC0	
ISR	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
ISCRH	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	
ISCRL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA	
DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0	DTC
DTCERB	DTCEB7	DTCEB6	DTCEB5	DTCEB4	DTCEB3	DTCEB2	DTCEB1	DTCEB0	
DTCERC	DTCEC7	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0	
DTCERD	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0	
DTCERE	DTCEE7	DTCEE6	DTCEE5	DTCEE4	DTCEE3	DTCEE2	DTCEE1	DTCEE0	
DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	
ABRKCR	CMF	—	—	—	—	—	—	BIE	INT
BARA	A23	A22	A21	A20	A19	A18	A17	A16	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
BARB	A15	A14	A13	A12	A11	A10	A9	A8	INT
BARC	A7	A6	A5	A4	A3	A2	A1	—	
PCSR	—	—	PWCKXB	PWCKXA	—	PWCKB	PWCKA	PWCKXC	PWM, PWMX
SBYCR	SSBY	STS2	STS1	STS0	—	SCK2	SCK1	SCK0	SYSTEM
LPWRCR	DTON	LSON	NESEL	EXCLE	—	—	—	—	
MSTPCRH	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	
MSTPCRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0	
SMR_1	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_1
ICCR_1	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	IIC_1
BRR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SCI_1
ICSR_1	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	IIC_1
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	SCI_1
TDR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSR_1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
RDR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF	
ICDR_1	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0	IIC_1
SARX_1	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX	
ICMR_1	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0	
SAR_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
TIER	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—	FRT
TCSR	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA	
FRC	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
OCRA/ OCRB	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0	
TOCR	ICRDMS	OCRAMS	ICRS	OCRS	OEA	OEB	OLVLA	OLVLB	
ICRA/ OCRAR	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ICRB/ OCRAF	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	FRT
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ICRC/ OCRDM	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ICRD	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
DACR	—	PWME	—	—	OEB	OEA	OS	CKS	PWMX
DADRAH	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	
DADRAL	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—	
DACNTH	DACNT7	DACNT6	DACNT5	DACNT4	DACNT3	DACNT2	DACNT1	DACNT0	
DADRBH	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	
DACNTL	DACNT8	DACNT9	DACNT10	DACNT11	DACNT12	DACNT13	—	REGS	
DADRBL	DA5	DA4	DA3	DA2	DA1	DA0	CFS	REGS	
TCSR_0	OVF	WT/$\overline{IT}$	TME	—	RST/$\overline{NMI}$	CKS2	CKS1	CKS0	WDT_0
TCNT_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
P1PCR	P17PCR	P16PCR	P15PCR	P14PCR	P13PCR	P12PCR	P11PCR	P10PCR	PORT
P2PCR	P27PCR	P26PCR	P25PCR	P24PCR	P23PCR	P22PCR	P21PCR	P20PCR	
P3PCR	P37PCR	P36PCR	P35PCR	P34PCR	P33PCR	P32PCR	P31PCR	P30PCR	
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	
P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR	
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	
P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR	
P3DDR	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR	
P4DDR	P47DDR	P46DDR	P45DDR	P44DDR	P43DDR	P42DDR	P41DDR	P40DDR	
P3DR	P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR	
P4DR	P47DR	P46DR	P45DR	P44DR	P43DR	P42DR	P41DR	P40DR	
P5DDR	—	—	—	—	—	P52DDR	P51DDR	P50DDR	
P6DDR	P67DDR	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR	
P5DR	—	—	—	—	—	P52DR	P51DR	P50DR	
P6DR	P67DR	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR	
P7PIN	P77PIN	P76PIN	P75PIN	P74PIN	P73PIN	P72PIN	P71PIN	P70PIN	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	INT
STCR	—	IICX1	IICX0	IICE	FLSHE	—	ICKS1	ICKS0	SYSTEM
SYSCR	—	—	INTM1	INTM0	XRST	NMIEG	—	RAME	
MDCR	EXPE	IOSE	—	—	—	MDS2	MDS1	MDS0	
BCR	—	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	IOS1	IOS0	BSC
WSCR	—	—	ABW	AST	WMS1	WMS0	WC1	WC0	
TCR_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_0, TMR_1
TCR_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	
TCSR_0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0	
TCSR_1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
TCORA_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORA_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORB_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORB_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCNT_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCNT_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
PWOERB	OE15	OE14	OE13	OE12	OE11	OE10	OE9	OE8	PWM
PWOERA	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0	
PWDPRB	OS15	OS14	OS13	OS12	OS11	OS10	OS9	OS8	
PWDPRA	OS7	OS6	OS5	OS4	OS3	OS2	OS1	OS0	
PWSL	PWCKE	PWCKS	—	—	RS3	RS2	RS1	RS0	
PWDR15 to 0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SMR_0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_0
ICCR_0	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	IIC_0
BRR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SCI_0
ICSR_0	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	IIC_0
SCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	SCI_0
TDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSR_0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
RDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCMR_0	—	—	—	—	SDIR	SINV	—	SMIF	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ICDR_0	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0	IIC_0
SARX_0	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX	
ICMR_0	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0	
SAR_0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D Converter
ADDRAL	AD1	AD0	—	—	—	—	—	—	
ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRBL	AD1	AD0	—	—	—	—	—	—	
ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRCL	AD1	AD0	—	—	—	—	—	—	
ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRDL	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0	
ADCR	TRGS1	TRGS0	—	—	—	—	—	—	
TCSR_1	OVF	WT/$\overline{IT}$	TME	PSS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0	WDT_1
TCNT_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_X	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_X, TMR_Y
TCR_Y	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	
TCSR_X	CMFB	CMFA	OVF	ICF	OS3	OS2	OS1	OS0	
TCSR_Y	CMFB	CMFA	OVF	ICIE	OS3	OS2	OS1	OS0	
TICRR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORA_Y	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TICRF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORB_Y	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCNT_X	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCNT_Y	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORC	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORA_X	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORB_X	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCONRI	—	—	—	ICST	—	—	—	—	
TCONRS	TMRX/Y	—	—	—	—	—	—	—	

23.3 Register States in Each Operating Mode

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
TCMCNT_0	Initialized	—	—	—	—	—	—	—	Initialized	TCM_0
TCMMLCM_0	Initialized	—	—	—	—	—	—	—	Initialized	
TCMICR_0	Initialized	—	—	—	—	—	—	—	Initialized	
TCMICRF_0	Initialized	—	—	—	—	—	—	—	Initialized	
TCMCSR_0	Initialized	—	—	—	—	—	—	—	Initialized	
TCMCR_0	Initialized	—	—	—	—	—	—	—	Initialized	
TCMIER_0	Initialized	—	—	—	—	—	—	—	Initialized	
TCMCNT_1	Initialized	—	—	—	—	—	—	—	Initialized	TCM_1
TCMMLCM_1	Initialized	—	—	—	—	—	—	—	Initialized	
TCMICR_1	Initialized	—	—	—	—	—	—	—	Initialized	
TCMICRF_1	Initialized	—	—	—	—	—	—	—	Initialized	
TCMCSR_1	Initialized	—	—	—	—	—	—	—	Initialized	
TCMCR_1	Initialized	—	—	—	—	—	—	—	Initialized	
TCMIER_1	Initialized	—	—	—	—	—	—	—	Initialized	
P6NCE	Initialized	—	—	—	—	—	—	—	Initialized	PORT
P6NCMC	Initialized	—	—	—	—	—	—	—	Initialized	
P6NCCS	Initialized	—	—	—	—	—	—	—	Initialized	
P4NCE	Initialized	—	—	—	—	—	—	—	Initialized	
P4NCMC	Initialized	—	—	—	—	—	—	—	Initialized	
P4NCCS	Initialized	—	—	—	—	—	—	—	Initialized	
MSTPCRA	Initialized	—	—	—	—	—	—	—	Initialized	SYSTEM
MSTPCRB	Initialized	—	—	—	—	—	—	—	Initialized	
ICRD	Initialized	—	—	—	—	—	—	—	Initialized	INT
FCCS	Initialized	—	—	—	—	—	—	—	Initialized	ROM
FPCS	Initialized	—	—	—	—	—	—	—	Initialized	
FECS	Initialized	—	—	—	—	—	—	—	Initialized	
FKEY	Initialized	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
FMATS	Initialized	—	—	—	—	—	—	—	Initialized	ROM
FTDAR	Initialized	—	—	—	—	—	—	—	Initialized	
TCRXY	Initialized	—	—	—	—	—	—	—	Initialized	TMR_XY
ICXR_0	Initialized	—	—	—	—	—	—	—	Initialized	IIC_0
ICXR_1	Initialized	—	—	—	—	—	—	—	Initialized	IIC_1
DDCSWR	Initialized	—	—	—	—	—	—	—	Initialized	IIC_0, IIC_1
ICRA	Initialized	—	—	—	—	—	—	—	Initialized	INT
ICRB	Initialized	—	—	—	—	—	—	—	Initialized	
ICRC	Initialized	—	—	—	—	—	—	—	Initialized	
ISR	Initialized	—	—	—	—	—	—	—	Initialized	
ISCRH	Initialized	—	—	—	—	—	—	—	Initialized	
ISCRL	Initialized	—	—	—	—	—	—	—	Initialized	
DTCERA	Initialized	—	—	—	—	—	—	—	Initialized	DTC
DTCERB	Initialized	—	—	—	—	—	—	—	Initialized	
DTCERC	Initialized	—	—	—	—	—	—	—	Initialized	
DTCERD	Initialized	—	—	—	—	—	—	—	Initialized	
DTCERE	Initialized	—	—	—	—	—	—	—	Initialized	
DTVECR	Initialized	—	—	—	—	—	—	—	Initialized	
ABRKCR	Initialized	—	—	—	—	—	—	—	Initialized	INT
BARA	Initialized	—	—	—	—	—	—	—	Initialized	
BARB	Initialized	—	—	—	—	—	—	—	Initialized	
BARC	Initialized	—	—	—	—	—	—	—	Initialized	
PCSR	Initialized	—	—	—	—	—	—	—	Initialized	PWM, PWMX
SBYCR	Initialized	—	—	—	—	—	—	—	Initialized	SYSTEM
LPWRCR	Initialized	—	—	—	—	—	—	—	Initialized	
MSTPCRH	Initialized	—	—	—	—	—	—	—	Initialized	
MSTPCRL	Initialized	—	—	—	—	—	—	—	Initialized	
SMR_1	Initialized	—	—	—	—	—	—	—	Initialized	SCI_1

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
ICCR_1	Initialized	—	—	—	—	—	—	—	Initialized	IIC_1
BRR_1	Initialized	—	—	—	—	—	—	—	Initialized	SCI_1
ICSR_1	Initialized	—	—	—	—	—	—	—	Initialized	IIC_1
SCR_1	Initialized	—	—	—	—	—	—	—	Initialized	SCI_1
TDR_1	Initialized	—	—	—	—	—	—	—	Initialized	
SSR_1	Initialized	—	—	—	—	—	—	—	Initialized	
RDR_1	Initialized	—	—	—	—	—	—	—	Initialized	
SCMR_1	Initialized	—	—	—	—	—	—	—	Initialized	
ICDR_1	—	—	—	—	—	—	—	—	—	IIC_1
SARX_1	Initialized	—	—	—	—	—	—	—	Initialized	
ICMR_1	Initialized	—	—	—	—	—	—	—	Initialized	
SAR_1	Initialized	—	—	—	—	—	—	—	Initialized	
TIER	Initialized	—	—	—	—	—	—	—	Initialized	FRT
TCSR	Initialized	—	—	—	—	—	—	—	Initialized	
FRC	Initialized	—	—	—	—	—	—	—	Initialized	
OCRA	Initialized	—	—	—	—	—	—	—	Initialized	
OCRB	Initialized	—	—	—	—	—	—	—	Initialized	
TCR	Initialized	—	—	—	—	—	—	—	Initialized	
TOCR	Initialized	—	—	—	—	—	—	—	Initialized	
ICRA	Initialized	—	—	—	—	—	—	—	Initialized	
OCRAR	Initialized	—	—	—	—	—	—	—	Initialized	
ICRB	Initialized	—	—	—	—	—	—	—	Initialized	
OCRAF	Initialized	—	—	—	—	—	—	—	Initialized	
ICRC	Initialized	—	—	—	—	—	—	—	Initialized	
OCRDM	Initialized	—	—	—	—	—	—	—	Initialized	
ICRD	Initialized	—	—	—	—	—	—	—	Initialized	
DACR	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	PWMX
DADRA	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
DACNT	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
DADRB	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
TCSR_0	Initialized	—	—	—	—	—	—	—	Initialized	WDT_0
TCNT_0	Initialized	—	—	—	—	—	—	—	Initialized	
P1PCR	Initialized	—	—	—	—	—	—	—	Initialized	PORT
P2PCR	Initialized	—	—	—	—	—	—	—	Initialized	
P3PCR	Initialized	—	—	—	—	—	—	—	Initialized	
P1DDR	Initialized	—	—	—	—	—	—	—	Initialized	
P2DDR	Initialized	—	—	—	—	—	—	—	Initialized	
P1DR	Initialized	—	—	—	—	—	—	—	Initialized	
P2DR	Initialized	—	—	—	—	—	—	—	Initialized	
P3DDR	Initialized	—	—	—	—	—	—	—	Initialized	
P4DDR	Initialized	—	—	—	—	—	—	—	Initialized	
P3DR	Initialized	—	—	—	—	—	—	—	Initialized	
P4DR	Initialized	—	—	—	—	—	—	—	Initialized	
P5DDR	Initialized	—	—	—	—	—	—	—	Initialized	
P6DDR	Initialized	—	—	—	—	—	—	—	Initialized	
P5DR	Initialized	—	—	—	—	—	—	—	Initialized	
P6DR	Initialized	—	—	—	—	—	—	—	Initialized	
P7PIN	—	—	—	—	—	—	—	—	—	
IER	Initialized	—	—	—	—	—	—	—	Initialized	INT
STCR	Initialized	—	—	—	—	—	—	—	Initialized	SYSTEM
SYSCR	Initialized	—	—	—	—	—	—	—	Initialized	
MDCR	Initialized	—	—	—	—	—	—	—	Initialized	
BCR	Initialized	—	—	—	—	—	—	—	Initialized	BSC
WSCR	Initialized	—	—	—	—	—	—	—	Initialized	
TCR_0	Initialized	—	—	—	—	—	—	—	Initialized	TMR_0, TMR_1
TCR_1	Initialized	—	—	—	—	—	—	—	Initialized	
TCSR_0	Initialized	—	—	—	—	—	—	—	Initialized	
TCSR_1	Initialized	—	—	—	—	—	—	—	Initialized	
TCORA_0	Initialized	—	—	—	—	—	—	—	Initialized	
TCORA_1	Initialized	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
TCORB_0	Initialized	—	—	—	—	—	—	—	Initialized	TMR_0, TMR_1
TCORB_1	Initialized	—	—	—	—	—	—	—	Initialized	
TCNT_0	Initialized	—	—	—	—	—	—	—	Initialized	
TCNT_1	Initialized	—	—	—	—	—	—	—	Initialized	
PWOERB	Initialized	—	—	—	—	—	—	—	Initialized	PWM
PWOERA	Initialized	—	—	—	—	—	—	—	Initialized	
PWDPRB	Initialized	—	—	—	—	—	—	—	Initialized	
PWDPRA	Initialized	—	—	—	—	—	—	—	Initialized	
PWSL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
PWDR15 to 0	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
SMR_0	Initialized	—	—	—	—	—	—	—	Initialized	SCI_0
ICCR_0	Initialized	—	—	—	—	—	—	—	Initialized	IIC_0
BRR_0	Initialized	—	—	—	—	—	—	—	Initialized	SCI_0
ICSR_0	Initialized	—	—	—	—	—	—	—	Initialized	IIC_0
SCR_0	Initialized	—	—	—	—	—	—	—	Initialized	SCI_0
TDR_0	Initialized	—	—	—	—	—	—	—	Initialized	
SSR_0	Initialized	—	—	—	—	—	—	—	Initialized	
RDR_0	Initialized	—	—	—	—	—	—	—	Initialized	
SCMR_0	Initialized	—	—	—	—	—	—	—	Initialized	
ICDR_0	—	—	—	—	—	—	—	—	—	IIC_0
SARX_0	Initialized	—	—	—	—	—	—	—	Initialized	
ICMR_0	Initialized	—	—	—	—	—	—	—	Initialized	
SAR_0	Initialized	—	—	—	—	—	—	—	Initialized	
ADDRAH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	A/D Converter
ADDRAL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRBH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRBL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRCH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRCL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
ADDRDH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	A/D Converter
ADDRDL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
ADCSR	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
ADCR	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
TCSR_1	Initialized	—	—	—	—	—	—	—	Initialized	WDT_1
TCNT_1	Initialized	—	—	—	—	—	—	—	Initialized	
TCR_X	Initialized	—	—	—	—	—	—	—	Initialized	TMR_X, TMR_Y
TCR_Y	Initialized	—	—	—	—	—	—	—	Initialized	
TCSR_X	Initialized	—	—	—	—	—	—	—	Initialized	
TCSR_Y	Initialized	—	—	—	—	—	—	—	Initialized	
TICRR	Initialized	—	—	—	—	—	—	—	Initialized	
TCORA_Y	Initialized	—	—	—	—	—	—	—	Initialized	
TICRF	Initialized	—	—	—	—	—	—	—	Initialized	
TCORB_Y	Initialized	—	—	—	—	—	—	—	Initialized	
TCNT_X	Initialized	—	—	—	—	—	—	—	Initialized	
TCNT_Y	Initialized	—	—	—	—	—	—	—	Initialized	
TCORC	Initialized	—	—	—	—	—	—	—	Initialized	
TCORA_X	Initialized	—	—	—	—	—	—	—	Initialized	
TCORB_X	Initialized	—	—	—	—	—	—	—	Initialized	
TCONRI	Initialized	—	—	—	—	—	—	—	Initialized	
TCONRS	Initialized	—	—	—	—	—	—	—	Initialized	

23.4 Register Selection Condition

Lower Address	Register Abbreviation	Register selection condition	Module
H'FBC0	TCMCNT_0	MSTPB0 = 0	TCM_0
H'FBC2	TCMMLCM_0		
H'FBC4	TCMICR_0		
H'FBC6	TCMICRF_0		
H'FBC8	TCMCSR_0		
H'FBC9	TCMCR_0		
H'FBCA	TCMIER_0		
H'FBD0	TCMCNT_1	MSTPB1 = 0	TCM_1
H'FBD2	TCMMLCM_1		
H'FBD4	TCMICR_1		
H'FBD6	TCMICRF_1		
H'FBD8	TCMCSR_1		
H'FBD9	TCMCR_1		
H'FBDA	TCMIER_1		
H'FE00	P6NCE	No condition	PORT
H'FE01	P6NCMC		
H'FE02	P6NCCS		
H'FE09	P4NCE		
H'FE0A	P4NCMC		
H'FE0B	P4NCCS		
H'FE7E	MSTPCRA	No condition	SYSTEM
H'FE7F	MSTPCRB		
H'FE87	ICRD	No condition	INT
H'FEA8	FCCS	FLSHE = 1	ROM
H'FEA9	FPCS		
H'FEAA	FECS		
H'FEAC	FKEY		
H'FEAD	FMATS		
H'FEAE	FTDAR		

Lower Address	Register Abbreviation	Register selection condition	Module
H'FEC6	TCRXY	MSTP8 = 0	TMR_XY
H'FED4	ICXR_0	MSTP4 = 0	IIC_0
H'FED5	ICXR_1	MSTP3 = 0	IIC_1
H'FEE6	DDCSWR	MSTP4 = 0, IICE in STCR = 1	IIC_0, IIC_1
H'FEE8	ICRA	No condition	INT
H'FEE9	ICRB		
H'FEEA	ICRC		
H'FEEB	ISR		
H'FEEC	ISCRH		
H'FEED	ISCRL		
H'FEEE	DTCERA	No condition	DTC
H'FEEF	DTCERB		
H'FEF0	DTCERC		
H'FEF1	DTCERD		
H'FEF2	DTCERE		
H'FEF3	DTVECR		
H'FEF4	ABRKCR	No condition	INT
H'FEF5	BARA		
H'FEF6	BARB		
H'FEF7	BARC		
H'FF82	PCSR	No condition	PWM, PWMX
H'FF84	SBYCR	FLSHE in STCR = 0	SYSTEM
H'FF85	LPWRCR		
H'FF86	MSTPCRH		
H'FF87	MSTPCRL		
H'FF88	SMR_1	MSTP6 = 0, IICE in STCR = 0	SCI_1
	ICCR_1	MSTP3 = 0, IICE in STCR = 1	IIC_1
H'FF89	BRR_1	MSTP6 = 0, IICE in STCR = 0	SCI_1
	ICSR_1	MSTP3 = 0, IICE in STCR = 1	IIC_1
H'FF8A	SCR_1	MSTP6 = 0	SCI_1
H'FF8B	TDR_1		

Lower Address	Register Abbreviation	Register selection condition	Module
H'FF8C	SSR_1	MSTP6 = 0	SCI_1
H'FF8D	RDR_1		
H'FF8E	SCMR_1	MSTP6 = 0, IICE in STCR = 0	
	ICDR_1	MSTP3 = 0, IICE in STCR = 1	IIC_1
	SARX_1		
H'FF8F	ICMR_1		
	SAR_1		
H'FF90	TIER	MSTP13 = 0	FRT
H'FF91	TCSR		
H'FF92	FRC		
H'FF94	OCRA		
	OCRB		
H'FF96	TCR		
H'FF97	TOCR		
H'FF98	ICRA		
	OCRAR		
H'FF9A	ICRB		
	OCRAF		
H'FF9C	ICRC		
	OCRDM		
H'FF9E	ICRD		
H'FEA0	DACR	MSTP11 = 0, MSTPA0 = 1	PWMX
	DADRAH		
H'FEA1	DADRAL		
H'FEA6	DADRBH		
	DACNTH		
H'FEA7	DADRBL		
	DACNTL		
H'FFA8	TCSR_0	No condition	WDT_0
	TCNT_0 (write)		
H'FFA9	TCNT_0 (read)		

Lower Address	Register Abbreviation	Register selection condition	Module
H'FFAC	P1PCR	No condition	PORT
H'FFAD	P2PCR		
H'FFAE	P3PCR		
H'FFB0	P1DDR		
H'FFB1	P2DDR		
H'FFB2	P1DR		
H'FFB3	P2DR		
H'FFB4	P3DDR		
H'FFB5	P4DDR		
H'FFB6	P3DR		
H'FFB7	P4DR		
H'FFB8	P5DDR		
H'FFB9	P6DDR		
H'FFBA	P5DR		
H'FFBB	P6DR		
H'FFBE	P7PIN		
H'FFC2	IER	No condition	INT
H'FFC3	STCR	No condition	SYSTEM
H'FFC4	SYSCR		
H'FFC5	MDCR		
H'FFC6	BCR	No condition	BSC
H'FFC7	WSCR		
H'FFC8	TCR_0	MSTP12 = 0	TMR_0, TMR_1
H'FFC9	TCR_1		
H'FFCA	TCSR_0		
H'FFCB	TCSR_1		
H'FFCC	TCORA_0		
H'FFCD	TCORA_1		
H'FFCE	TCORB_0		
H'FFCF	TCORB_1		
H'FFD0	TCNT_0		
H'FFD1	TCNT_1		

Lower Address	Register Abbreviation	Register selection condition	Module
H'FFD2	PWOERB	MSTP11 = 0, MSTPA0 = 0	PWM
H'FFD3	PWOERA		
H'FFD4	PWDPRB		
H'FFD5	PWDPRA		
H'FFD6	PWSL		
H'FFD7	PWDR15 to 0		
H'FFD8	SMR_0	MSTP7 = 0, IICE in STCR = 0	SCI_0
	ICCR_0	MSTP4 = 0, IICE in STCR = 1	IIC_0
H'FFD9	BRR_0	MSTP7 = 0, IICE in STCR = 0	SCI_0
	ICSR_0	MSTP4 = 0, IICE in STCR = 1	IIC_0
H'FFDA	SCR_0	MSTP7 = 0	SCI_0
H'FFDB	TDR_0		
H'FFDC	SSR_0		
H'FFDD	RDR_0		
H'FFDE	SCMR_0	MSTP7 = 0, IICE in STCR = 0	
	ICDR_0	MSTP4 = 0, IICE in STCR = 1	IIC_0
	SARX_0		
H'FFDF	ICMR_0		
	SAR_0		
H'FFE0	ADDRAH	MSTP9 = 0	A/D Converter
H'FFE1	ADDRAL		
H'FFE2	ADDRBH		
H'FFE3	ADDRBL		
H'FFE4	ADDRCH		
H'FFE5	ADDRCL		
H'FFE6	ADDRDH		
H'FFE7	ADDRDL		
H'FFE8	ADCSR		
H'FFE9	ADCR		
H'FFEA	TCSR_1	No condition	WDT_1
	TCNT_1 (write)		
H'FFEB	TCNT_1 (read)		

Lower Address	Register Abbreviation	Register selection condition	Module
H'FFF0	TCR_X	MSTP8 = 0	TMR_X, TMR_Y
	TCR_Y		
H'FFF1	TCSR_X		
	TCSR_Y		
H'FFF2	TICRR		
	TCORA_Y		
H'FFF3	TICRF		
	TCORB_Y		
H'FFF4	TCNT_X		
	TCNT_Y		
H'FFF5	TCORC		
H'FFF6	TCORA_X		
H'FFF7	TCORB_X		
H'FFFC	TCONRI		
H'FFFE	TCONRS		

23.5 Register Addresses (Classification by Type of Module)

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
INT	ICRD	8	H'FE87	H'00	8	2
INT	ICRA	8	H'FEE8	H'00	8	2
INT	ICRB	8	H'FEE9	H'00	8	2
INT	ICRC	8	H'FEEA	H'00	8	2
INT	ISR	8	H'FEEB	H'00	8	2
INT	ISCRH	8	H'FEEC	H'00	8	2
INT	ISCRL	8	H'FEED	H'00	8	2
INT	ABRKCR	8	H'FEF4	—	8	2
INT	BARA	8	H'FEF5	H'00	8	2
INT	BARB	8	H'FEF6	H'00	8	2
INT	BARC	8	H'FEF7	H'00	8	2
INT	IER	8	H'FFC2	H'00	8	2
BSC	BCR	8	H'FFC6	H'D3	8	2
BSC	WSCR	8	H'FFC7	H'F3	8	2
DTC	DTCERA	8	H'FEEE	H'00	8	2
DTC	DTCERB	8	H'FEEF	H'00	8	2
DTC	DTCERC	8	H'FEF0	H'00	8	2
DTC	DTCERD	8	H'FEF1	H'00	8	2
DTC	DTCERE	8	H'FEF2	H'00	8	2
DTC	DTVECR	8	H'FEF3	H'00	8	2
PORT	P1PCR	8	H'FFAC	H'00	8	2
PORT	P1DDR	8	H'FFB0	H'00	8	2
PORT	P1DR	8	H'FFB2	H'00	8	2
PORT	P2PCR	8	H'FFAD	H'00	8	2
PORT	P2DDR	8	H'FFB1	H'00	8	2
PORT	P2DR	8	H'FFB3	H'00	8	2
PORT	P3PCR	8	H'FFAE	H'00	8	2
PORT	P3DDR	8	H'FFB4	H'00	8	2
PORT	P3DR	8	H'FFB6	H'00	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
PORT	P4NCE	8	H'FE09	H'00	8	2
PORT	P4NCMC	8	H'FE0A	H'00	8	2
PORT	P4NCCS	8	H'FE0B	H'00	8	2
PORT	P4DDR	8	H'FFB5	H'40/H'00	8	2
PORT	P4DR	8	H'FFB7	H'00	8	2
PORT	P5DR	8	H'FFBA	H'F8	8	2
PORT	P5DDR	8	H'FFB8	H'00	8	2
PORT	P6NCE	8	H'FE00	H'00	8	2
PORT	P6NCMC	8	H'FE01	H'00	8	2
PORT	P6NCCS	8	H'FE02	H'00	8	2
PORT	P6DR	8	H'FFBB	H'00	8	2
PORT	P6DDR	8	H'FFB9	H'00	8	2
PORT	P7PIN	8	H'FFBE	—	8	2
PWM	PWOERB	8	H'FFD2	H'00	8	2
PWM	PWOERA	8	H'FFD3	H'00	8	2
PWM	PWDPRB	8	H'FFD4	H'00	8	2
PWM	PWDPRA	8	H'FFD5	H'00	8	2
PWM	PWSL	8	H'FFD6	H'20	8	2
PWM	PWDR15 to 0	8	H'FFD7	H'00	8	2
PWM	PCSR	8	H'FF82	H'00	8	2
PWMX	DACR	8	H'FEA0	H'30	8	2
PWMX	DACR	8	H'FFA0	H'FF	8	2
PWMX	DADRAH	8	H'FEA0	H'00	8	2
PWMX	DADRAH	8	H'FFA0	H'FF	8	2
PWMX	DADRAL	8	H'FEA1	H'FF	8	2
PWMX	DADRAL	8	H'FFA1	H'FF	8	2
PWMX	DACNTH	8	H'FEA6	H'FF	8	2
PWMX	DACNTH	8	H'FFA6	H'00	8	2
PWMX	DADRBH	8	H'FEA6	H'FF	8	2
PWMX	DADRBH	8	H'FFA6	H'FF	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
PWMX	DACNTL	8	H'FEA7	H'03	8	2
PWMX	DACNTL	8	H'FFA7	H'03	8	2
PWMX	DADRBL	8	H'FEA7	H'FF	8	2
PWMX	DADRBL	8	H'FFA7	H'FF	8	2
PWMX	PCSR	8	H'FF82	H'00	8	2
FRT	TIER	8	H'FF90	H'01	8	2
FRT	TCSR	8	H'FF91	H'00	8	2
FRT	FRC	16	H'FF92	H'0000	16	2
FRT	OCRA	16	H'FF94	H'FFFF	16	2
FRT	OCRB	16	H'FF94	H'FFFF	16	2
FRT	TCR	8	H'FF96	H'00	8	2
FRT	TOCR	8	H'FF97	H'00	8	2
FRT	ICRA	16	H'FF98	H'0000	16	2
FRT	OCRAR	16	H'FF98	H'FFFF	16	2
FRT	ICRB	16	H'FF9A	H'0000	16	2
FRT	OCRAF	16	H'FF9A	H'FFFF	16	2
FRT	ICRC	16	H'FF9C	H'0000	16	2
FRT	OCRDM	16	H'FF9C	H'0000	16	2
FRT	ICRD	16	H'FF9E	H'0000	16	2
TPU_0	TCMCNT_0	16	H'FBC0	H'0000	16	2
TPU_0	TCMMLCM_0	16	H'FBC2	H'FFFF	16	2
TPU_0	TCMICR_0	16	H'FBC4	—	16	2
TPU_0	TCMICRF_0	16	H'FBC6	H'0000	16	2
TPU_0	TCMCSR_0	8	H'FBC8	H'00	8	2
TPU_0	TCMCR_0	8	H'FBC9	H'00	8	2
TPU_0	TCMIER_0	16	H'FBCA	H'00	8	2
TPU_1	TCMCNT_1	16	H'FBD0	H'0000	16	2
TPU_1	TCMMLCM_1	16	H'FBD2	H'FFFF	16	2
TPU_1	TCMICR_1	16	H'FBD4	—	16	2
TPU_1	TCMICRF_1	16	H'FBD6	H'0000	16	2
TPU_1	TCMCSR_1	8	H'FBD8	H'00	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
TPU_1	TCMCR_1	8	H'FBD9	H'00	8	2
TPU_1	TCMIER_1	16	H'FBDA	H'00	8	2
TMR_0	TCR_0	8	H'FFC8	H'00	8	2
TMR_0	TCSR_0	8	H'FFCA	H'00	8	2
TMR_0	TCORA_0	8	H'FFCC	H'FF	16	2
TMR_0	TCORB_0	8	H'FFCE	H'FF	16	2
TMR_0	TCNT_0	8	H'FFD0	H'00	16	2
TMR_1	TCR_1	8	H'FFC9	H'00	8	2
TMR_1	TCSR_1	8	H'FFCB	H'FF	16	2
TMR_1	TCORA_1	8	H'FFCD	H'FF	16	2
TMR_1	TCORB_1	8	H'FFCF	H'FF	16	2
TMR_1	TCNT_1	8	H'FFD1	H'00	16	2
TMR_X	TCR_X	8	H'FFF0	H'00	8	2
TMR_X	TCSR_X	8	H'FFF1	H'00	8	2
TMR_X	TICRR	8	H'FFF2	H'00	8	2
TMR_X	TICRF	8	H'FFF3	H'00	8	2
TMR_X	TCNT_X	8	H'FFF4	H'00	8	2
TMR_X	TCORC	8	H'FFF5	H'FF	8	2
TMR_X	TCORA_X	8	H'FFF6	H'FF	8	2
TMR_X	TCORB_X	8	H'FFF7	H'FF	8	2
TMR_X	TCONRI	8	H'FFFC	H'00	8	2
TMR_Y	TCR_Y	8	H'FEC8	H'00	8	2
TMR_Y	TCR_Y	8	H'FFF0	H'00	8	2
TMR_Y	TCSR_Y	8	H'FEC9	H'10	8	2
TMR_Y	TCSR_Y	8	H'FFF1	H'00	8	2
TMR_Y	TCORA_Y	8	H'FECA	H'FF	8	2
TMR_Y	TCORA_Y	8	H'FFF2	H'FF	8	2
TMR_Y	TCORB_Y	8	H'FECB	H'FF	8	2
TMR_Y	TCORB_Y	8	H'FFF3	H'FF	8	2
TMR_Y	TCNT_Y	8	H'FECC	H'00	8	2
TMR_Y	TCNT_Y	8	H'FFF4	H'00	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
TMR_X, TMR_Y	TCONRS	8	H'FFFE	H'00	8	2
TMR_XY	TCRXY	8	H'FEC6	H'00	8	2
WDT_0	TCSR_0	8	H'FFA8 (Write)	H'00	16	2
WDT_0	TCSR_0	8	H'FFA8 (Read)	H'00	8	2
WDT_0	TCNT_0	8	H'FFA8 (Write)	H'00	16	2
WDT_0	TCNT_0	8	H'FFA9 (Read)	H'00	8	2
WDT_1	TCSR_1	8	H'FFEA (Write)	H'00	16	2
WDT_1	TCSR_1	8	H'FFEA (Read)	H'00	8	2
WDT_1	TCNT_1	8	H'FFEA (Write)	H'00	16	2
WDT_1	TCNT_1	8	H'FFEB (Read)	H'00	8	2
SCI_0	SMR_0	8	H'FFD8	H'00	8	2
SCI_0	BRR_0	8	H'FFD9	H'FF	8	2
SCI_0	SCR_0	8	H'FFDA	H'00	8	2
SCI_0	TDR_0	8	H'FFDB	H'FF	8	2
SCI_0	SSR_0	8	H'FFDC	H'84	8	2
SCI_0	RDR_0	8	H'FFDD	H'00	8	2
SCI_0	SCMR_0	8	H'FFDE	H'F2	8	2
SCI_1	SMR_1	8	H'FF88	H'00	8	2
SCI_1	BRR_1	8	H'FF89	H'FF	8	2
SCI_1	SCR_1	8	H'FF8A	H'00	8	2
SCI_1	TDR_1	8	H'FF8B	H'FF	8	2
SCI_1	SSR_1	8	H'FF8C	H'84	8	2
SCI_1	RDR_1	8	H'FF8D	H'00	8	2
SCI_1	SCMR_1	8	H'FF8E	H'F2	8	2
IIC_0	ICXR_0	8	H'FED4	H'00	8	2
IIC_0	ICCR_0	8	H'FFD8	H'01	8	2
IIC_0	ICSR_0	8	H'FFD9	H'00	8	2
IIC_0	ICDR_0	8	H'FFDE	—	8	2
IIC_0	SARX_0	8	H'FFDE	H'01	8	2
IIC_0	ICMR_0	8	H'FFDF	H'00	8	2
IIC_0	SAR_0	8	H'FFDF	H'00	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
IIC_1	ICDR_1	8	H'FECE	—	8	2
IIC_1	SARX_1	8	H'FECE	H'01	8	2
IIC_1	ICMR_1	8	H'FECF	H'00	8	2
IIC_1	SAR_1	8	H'FECF	H'00	8	2
IIC_1	ICCR_1	8	H'FED0	H'01	8	2
IIC_1	ICSR_1	8	H'FED1	H'00	8	2
IIC_1	ICXR_1	8	H'FED5	H'00	8	2
IIC_1	ICCR_1	8	H'FF88	H'01	8	2
IIC_1	ICSR_1	8	H'FF89	H'00	8	2
IIC_1	ICDR_1	8	H'FF8E	—	8	2
IIC_1	SARX_1	8	H'FF8E	H'01	8	2
IIC_1	ICMR_1	8	H'FF8F	H'00	8	2
IIC_1	SAR_1	8	H'FF8F	H'00	8	2
IIC_0, IIC_1	DDCSWR	8	H'FEE6	H'0F	8	2
A/D Converter	ADDRAH	8	H'FFE0	H'00	8	2
A/D Converter	ADDRAL	8	H'FFE1	H'00	8	2
A/D Converter	ADDRBH	8	H'FFE2	H'00	8	2
A/D Converter	ADDRBL	8	H'FFE3	H'00	8	2
A/D Converter	ADDRCH	8	H'FFE4	H'00	8	2
A/D Converter	ADDRCL	8	H'FFE5	H'00	8	2
A/D Converter	ADDRDH	8	H'FFE6	H'00	8	2
A/D Converter	ADDRDL	8	H'FFE7	H'00	8	2
A/D Converter	ADCSR	8	H'FFE8	H'00	8	2
A/D Converter	ADCR	8	H'FFE9	H'3F	8	2
ROM	FCCS	8	H'FEA8	—	8	2
ROM	FPCS	8	H'FEA9	H'00	8	2
ROM	FECS	8	H'FEAA	H'00	8	2
ROM	FKEY	8	H'FEAC	H'00	8	2
ROM	FMATS	8	H'FEAD	—	8	2
ROM	FTDAR	8	H'FEAE	H'00	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
SYSTEM	MSTPCRA	8	H'FE7E	H'00	8	2
SYSTEM	MSTPCRB	8	H'FE7F	H'00	8	2
SYSTEM	SBYCR	8	H'FF84	H'01	8	2
SYSTEM	LPWRCR	8	H'FF85	H'00	8	2
SYSTEM	MSTPCRH	8	H'FF86	H'3F	8	2
SYSTEM	MSTPCRL	8	H'FF87	H'FF	8	2
SYSTEM	STCR	8	H'FFC3	H'00	8	2
SYSTEM	SYSCR	8	H'FFC4	H'09	8	2
SYSTEM	MDCR	8	H'FFC5	—	8	2

Section 24 Electrical Characteristics

24.1 Absolute Maximum Ratings

Table 24.1 lists the absolute maximum ratings.

Table 24.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage*	V_{CC}	–0.3 to +4.3	V
Input voltage (except port 7, P47, and P52)	V_{in}	–0.3 to V_{CC} +0.3	
Input voltage (P47 and P52)	V_{in}	–0.3 to +7.0	
Input voltage (port 7)	V_{in}	–0.3 to AV_{CC} +0.3	
Analog power supply voltage	AV_{CC}	–0.3 to +4.3	
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	
Operating temperature	T_{opr}	–20 to +75	°C
Operating temperature (when flash memory is programmed or erased)	T_{opr}	–20 to +75	
Storage temperature	T_{stg}	–55 to +125	

Caution: Permanent damage to this LSI may result if absolute maximum ratings are exceeded. Make sure the applied power supply does not exceed 4.3V.

Note: * Voltage applied to the VCC pin. The VCL pin should not be applied a voltage.

24.2 DC Characteristics

Table 24.2 lists the DC characteristics. Table 24.3 lists the permissible output currents. Table 24.4 lists the bus drive characteristics.

Table 24.2 DC Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC}*[1] = 3.0 V to 3.6 V, V_{SS} = AV_{SS}*[1] = 0 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	P67 to P60*[2] and $\overline{IRQ7}$ to $\overline{IRQ0}$*[3] (1)	V_T^-	V_{CC} × 0.2	—	—	V	
		V_T^+	—	—	V_{CC} × 0.7		
		$V_T^+ - V_T^-$	V_{CC} × 0.05	—	—		
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD2, MD1, MD0, and $\overline{ETRST}$ (2)	V_{IH}	V_{CC} × 0.9	—	V_{CC} + 0.3		
	EXTAL		V_{CC} × 0.7	—	V_{CC} + 0.3		
	Port 7		V_{CC} × 0.7	—	AV_{CC} + 0.3		
	P47 and P52		V_{CC} × 0.7	—	5.5		
	Input pins other than (1) and (2) above		V_{CC} × 0.7	—	V_{CC} + 0.3		
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD2, MD1, MD0, and $\overline{ETRST}$ (3)	V_{IL}	–0.3	—	V_{CC} × 0.1		
	NMI, EXTAL, and input pins other than (1) and (3) above		–0.3	—	V_{CC} × 0.2		
Output high voltage	All output pins (except P47 and P52)	V_{OH}	V_{CC} – 0.5	—	—		I_{OH} = –200 µA
			V_{CC} – 1.0	—	—		I_{OH} = –1 mA
	P47 and P52*[4]		0.5	—	—		I_{OH} = –200 µA
Output low voltage	All output pins*[5]	V_{OL}	—	—	0.4		I_{OL} = 1.6 mA
	Ports 1, 2, and 3		—	—	1.0		I_{OL} = 5 mA
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	10.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
	$\overline{STBY}$, NMI, MD2, MD1, and MD0		—	—	1.0		
	Port 7		—	—	1.0		V_{in} = 0.5 to AV_{CC} – 0.5 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Three-state leakage current (off state)	Ports 1 to 6	$\lvert I_{TSI} \rvert$	—	—	1.0		V_{in} = 0.5 to V_{CC} – 0.5 V
Input pull-up MOS current	Ports 1 to 3	$-I_P$	5	—	150		V_{in} = 0 V
Input capacitance	All pins	C_{in}	—	—	15	pF	V_{in} = 0 V f = 1MHz Ta = 25°C
Supply current*[6]	Normal operation	I_{CC}	—	20	30	mA	V_{CC} = 3.0 V to 3.6 V f = 20 MHz, all modules operating, high-speed mode
	Sleep mode		—	14	20		V_{CC} = 3.0 V to 3.6 V f = 20 MHz
	Standby mode		—	10	40	µA	Ta ≤ 50 °C
			—	—	80		50 °C < Ta
Analog power supply current	During A/D conversion	AI_{cc}	—	2	4	mA	
	A/D conversion standby		—	0.02	10	µA	AV_{CC} = 3.0 V to 3.6 V
VCC start voltage		VCC_{START}	—	0	0.8	V	
VCC rising edge		SVCC	—	—	20	ms/V	

Notes: 1. Do not leave the AVCC and AVSS pins open even if the A/D converter is not used.
Even if the A/D converter is not used, apply a voltage in the range from 3.0 V to 3.6 V to the AVCC pin by connecting to the power supply (V_{CC}).

2. Includes peripheral module inputs multiplexed on the pin.
3. $\overline{\text{IRQ2}}$ includes the $\overline{\text{ADTRG}}$ input multiplexed on the pin.
4. P47 and P52 and the peripheral module outputs multiplexed on these pins are NMOS push-pull outputs.
An external pull-up resistor is necessary to provide high-level output from SCL0 and SCL1 (ICE bit in ICCR is 1).
High levels of P47 and P52/SCK0 (ICE bit in ICCR is 0) are driven by NMOS. An external pull-up resistor is necessary to provide high-level output from these pins when they are used as an output.
5. Indicates values when ICE = 0. Low level output when the bus drive function is selected is rated separately.
6. Supply current values are for V_{IH} min = V_{CC} – 0.2 V and V_{IL} max = 0.2 V with all output pins unloaded and the on-chip pull-up MOSs in the off state.

Table 24.3 Permissible Output Currents

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0V

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	SCL0 and SDA0 (bus drive function selected)	I_{OL}	—	—	8	mA
	Ports 1, 2, and 3		—	—	5	
	Other output pins		—	—	2	
Permissible output low current (total)	Total of ports 1, 2, and 3	ΣI_{OL}	—	—	40	
	Total of all output pins, including the above		—	—	60	
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2	
Permissible output high current (total)	Total of all output pins	$\Sigma -I_{OH}$	—	—	30	

Notes: 1. To protect LSI reliability, do not exceed the output current values in table 24.3.
2. When driving a Darlington transistor or LED, always insert a current-limiting resistor in the output line, as show in figures 24.1 and 24.2.

Table 24.4 Bus Drive Characteristics

Conditions: V_{CC} = 3.0 V to 3.6V, V_{SS} = 0 V

Applicable Pins: SCL0, SDA0 (bus drive function selected)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	V_T^-	$V_{CC} \times 0.3$	—	—	V	
	V_T^+	—	—	$V_{CC} \times 0.7$		
	$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—		
Input high voltage	V_{IH}	$V_{CC} \times 0.7$	—	5.5		
Input low voltage	V_{IL}	– 0.5	—	$V_{CC} \times 0.3$		
Output low voltage	V_{OL}	—	—	0.5		I_{OL} = 8 mA
		—	—	0.4		I_{OL} = 3 mA
Input capacitance	C_{in}	—	—	10	pF	V_{in} = 0 V, f = 1 MHz, Ta = 25°C
Three-state leakage current (off state)	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V

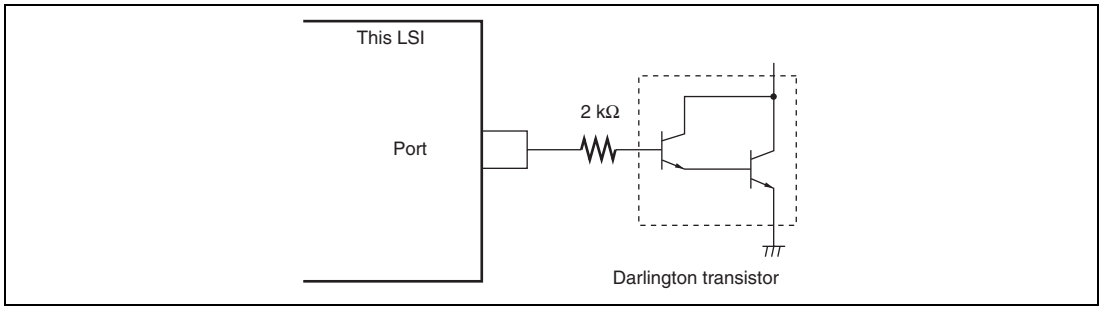


Figure 24.1 Darlington Transistor Drive Circuit (Example)

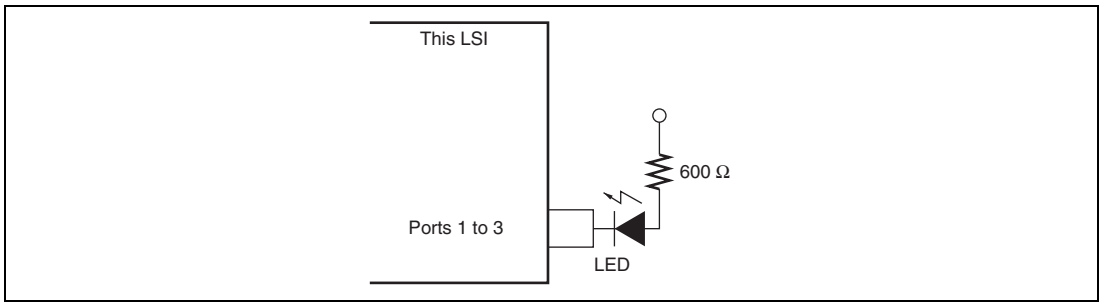


Figure 24.2 LED Drive Circuit (Example)

24.3 AC Characteristics

Figure 24.3 shows the test conditions for the AC characteristics.

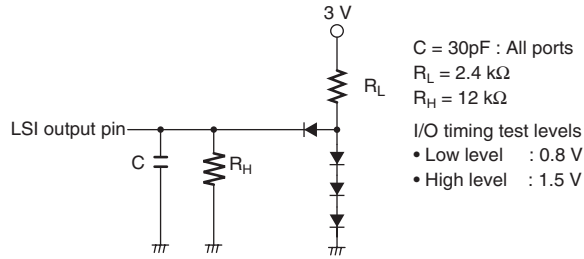


Figure 24.3 Output Load Circuit

24.3.1 Clock Timing

Table 24.5 shows the clock timing. The clock timing specified here covers clock output (ϕ) and clock pulse generator (crystal) and external clock input (EXTAL pin) oscillation stabilization times. For details of external clock input (EXTAL pin and EXCL pin) timing, see section 21, Clock Pulse Generator.

Table 24.5 Clock Timing

Condition A: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 8 MHz to 10 MHz

Condition B: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 8 MHz to 20 MHz

		Condition A		Condition B			
Item	**Symbol**	**Min.**	**Max.**	**Min.**	**Max.**	**Unit**	**Reference**
Clock cycle time	t_{cyc}	100	125	50	125	ns	Figure 24.4
Clock high pulse width	t_{CH}	30	—	20	—		
Clock low pulse width	t_{CL}	30	—	20	—		
Clock rise time	t_{Cr}	—	20	—	5		
Clock fall time	t_{Cf}	—	20	—	5		
Reset oscillation stabilization (crystal)	t_{OSC1}	20	—	20	—	ms	Figure 24.5
Software standby oscillation stabilization time (crystal)	t_{OSC2}	8	—	8	—		Figure 24.6
External clock output stabilization delay time	t_{DEXT}	500	—	500	—	µs	Figure 24.5

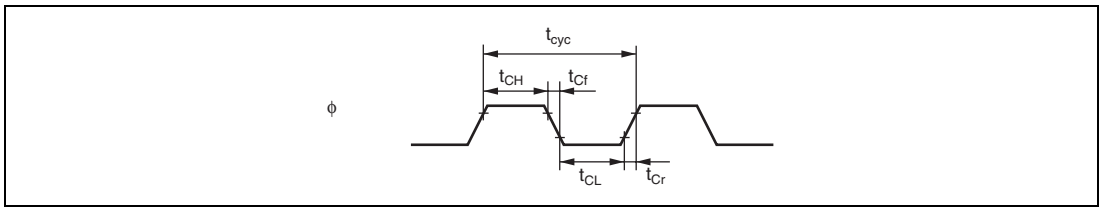


Figure 24.4 System Clock Timing

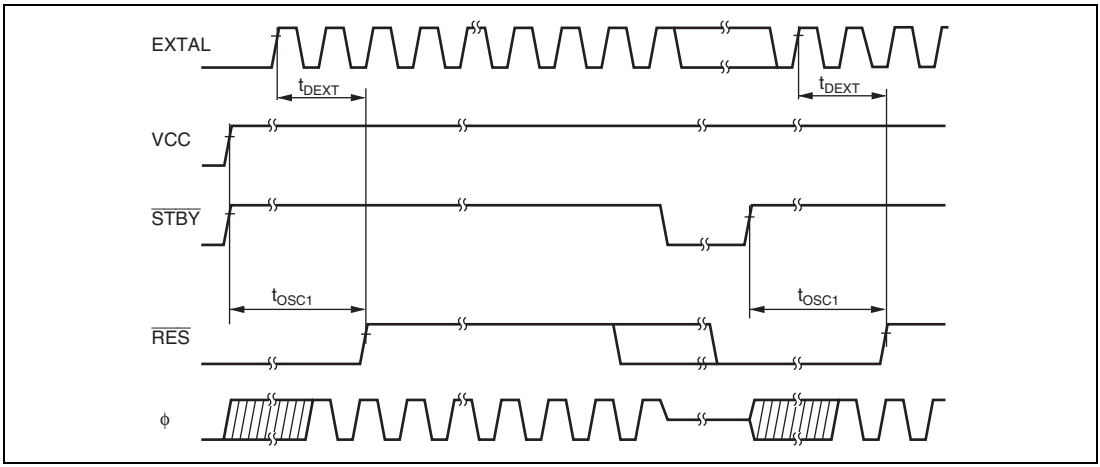


Figure 24.5 Oscillation Stabilization Timing

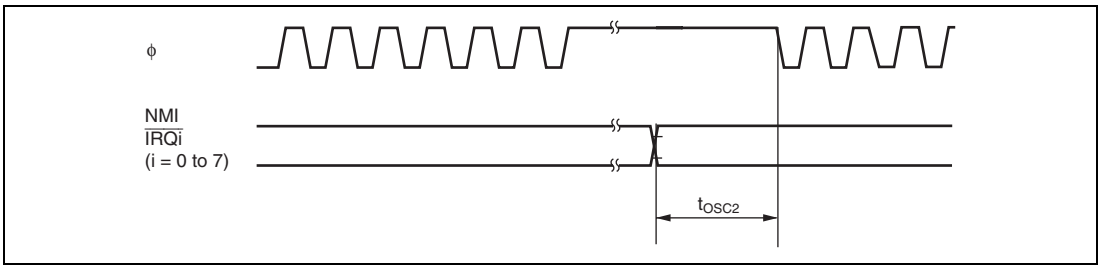


Figure 24.6 Oscillation Stabilization Timing (Exiting Software Standby Mode)

24.3.2 Control Signal Timing

Table 24.6 shows the control signal timing. Only external interrupts NMI, and IRQ0 to IRQ7 can be operated based on the subclock (ϕ = 32.768 kHz).

Table 24.6 Control Signal Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 32.768 kHz, 8 MHz to 20 MHz

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figure 24.7
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 24.8
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—		
IRQ setup time ($\overline{IRQ7}$ to $\overline{IRQ0}$)	t_{IRQS}	150	—		
IRQ hold time ($\overline{IRQ7}$ to $\overline{IRQ0}$)	t_{IRQH}	10	—		
IRQ pulse width ($\overline{IRQ7}$ to $\overline{IRQ0}$) (exiting software standby mode)	t_{IRQW}	200	—		

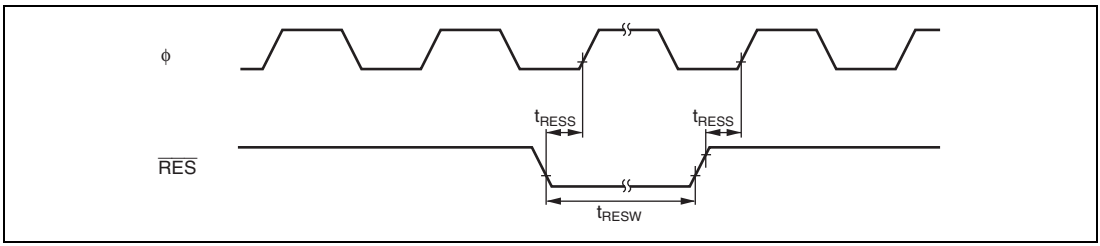


Figure 24.7 Reset Input Timing

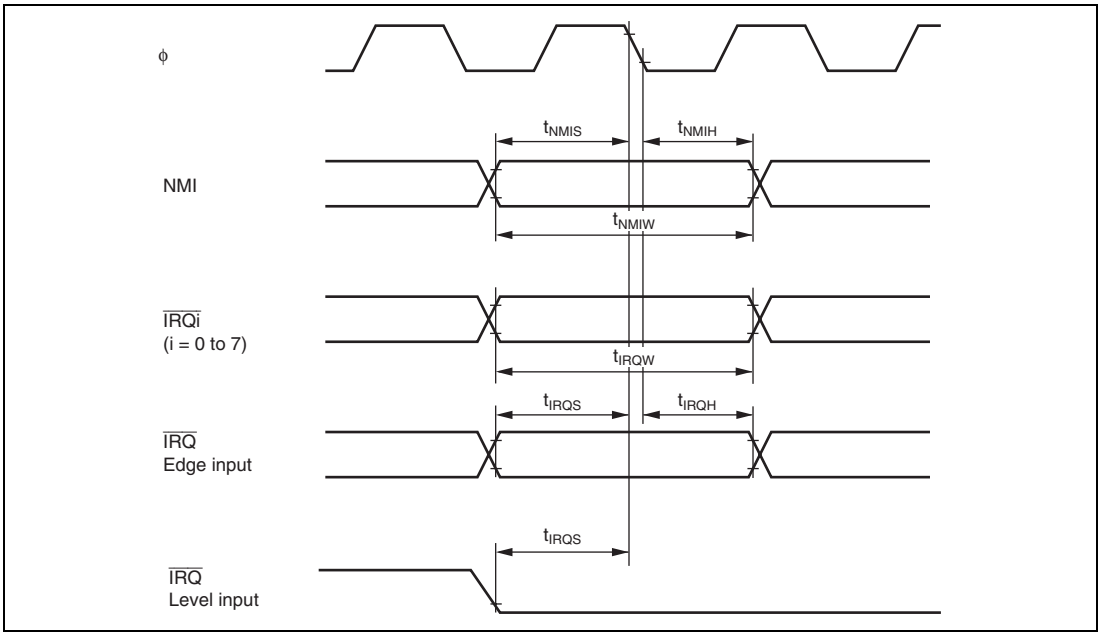


Figure 24.8 Interrupt Input Timing

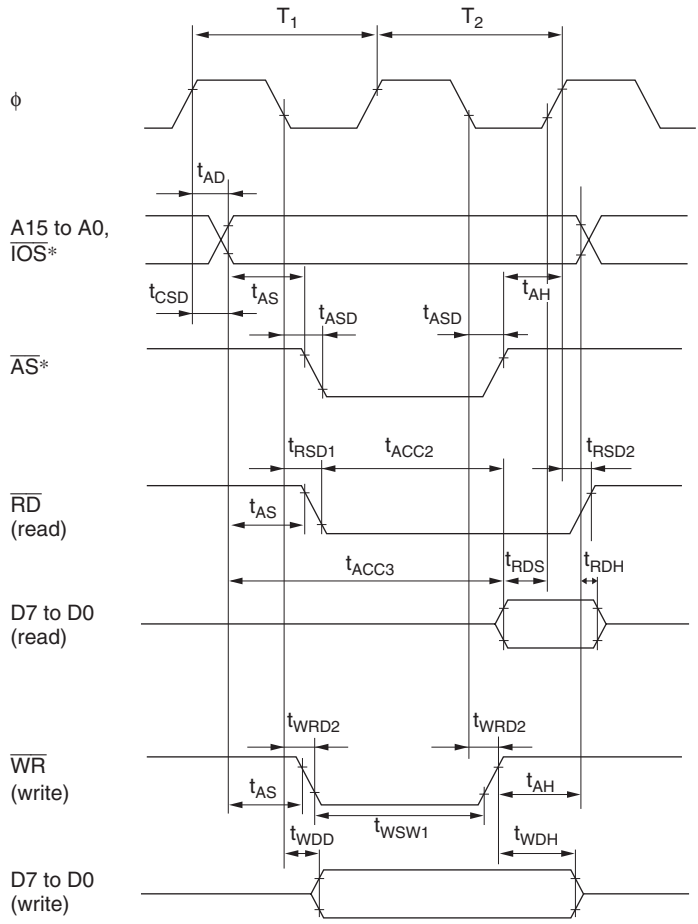
24.3.3 Bus Timing

Table 24.7 shows the bus timing. Operation in external expansion mode is not guaranteed when operating on the subclock (φ = 32.768 kHz).

Table 24.7 Bus Timing

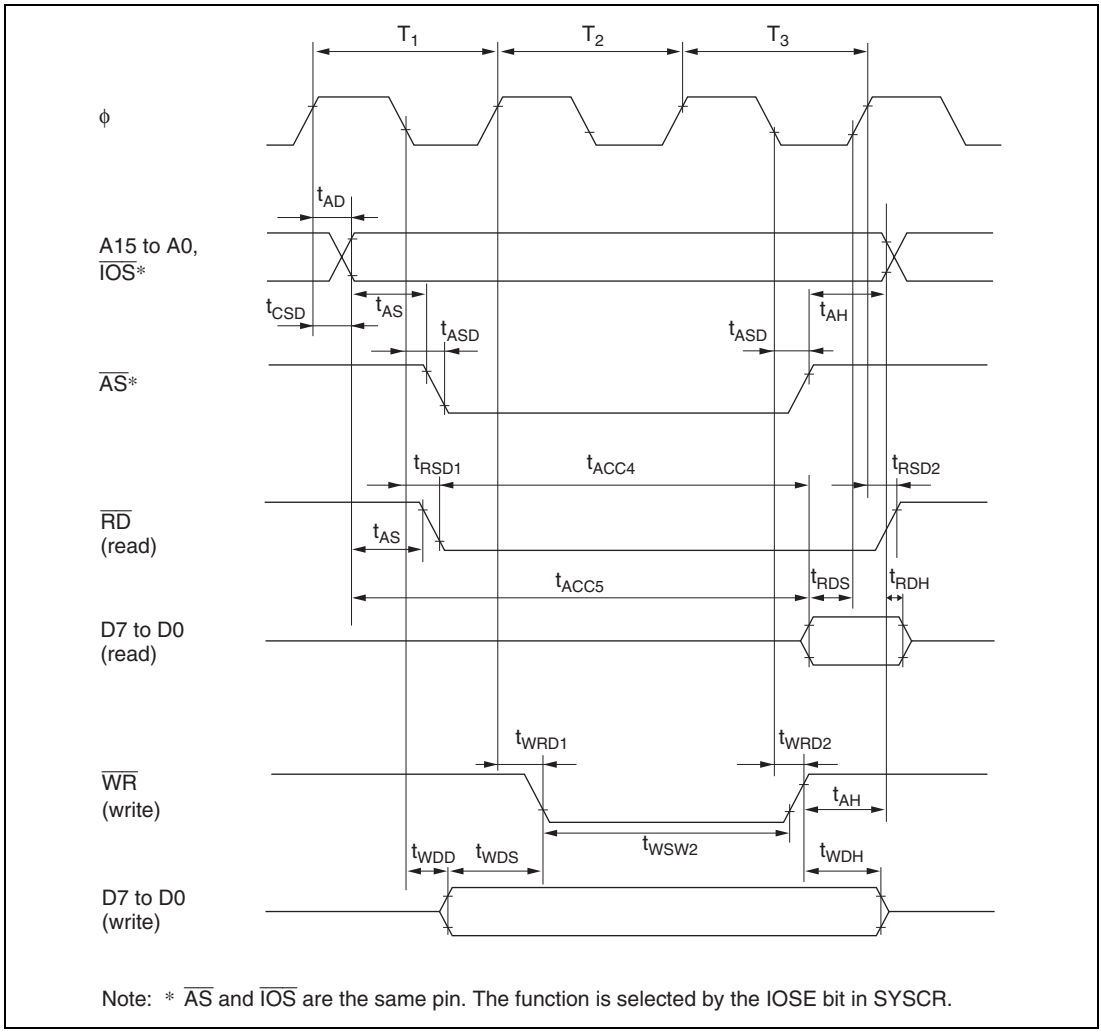
Condition: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, φ = 8 MHz to 20 MHz

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time	t_{AD}	—	20	ns	Figure 24.9 to figure 24.13
Address setup time	t_{AS}	$0.5 \times t_{cyc} - 15$	—	ns	
Address hold time	t_{AH}	$0.5 \times t_{cyc} - 10$	—	ns	
$\overline{CS}$ delay time ($\overline{IOS}$)	t_{CSD}	—	20	ns	
$\overline{AS}$ delay time	t_{ASD}	—	30	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	30	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	30	ns	
Read data setup time	t_{RDS}	15	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Read data access time 1	t_{ACC1}	—	$1.0 \times t_{cyc} - 30$	ns	
Read data access time 2	t_{ACC2}	—	$1.5 \times t_{cyc} - 25$	ns	
Read data access time 3	t_{ACC3}	—	$2.0 \times t_{cyc} - 30$	ns	
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 25$	ns	
Read data access time 5	t_{ACC5}	—	$3.0 \times t_{cyc} - 30$	ns	
$\overline{WR}$ delay time 1	t_{WRD1}	—	30	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	30	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 20$	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 20$	—	ns	
Write data delay time	t_{WDD}	—	30	ns	
Write data setup time	t_{WDS}	0	—	ns	
Write data hold time	t_{WDH}	10	—	ns	
$\overline{WAIT}$ setup time	t_{WTS}	30	—	ns	
$\overline{WAIT}$ hold time	t_{WTH}	5	—	ns	



Note: * AS and IOS are the same pin. The function is selected by the IOSE bit in SYSCR.

Figure 24.9 Basic Bus Timing (Two-State Access)



Note: * AS and IOS are the same pin. The function is selected by the IOSE bit in SYSCR.

Figure 24.10 Basic Bus Timing (Three-State Access)

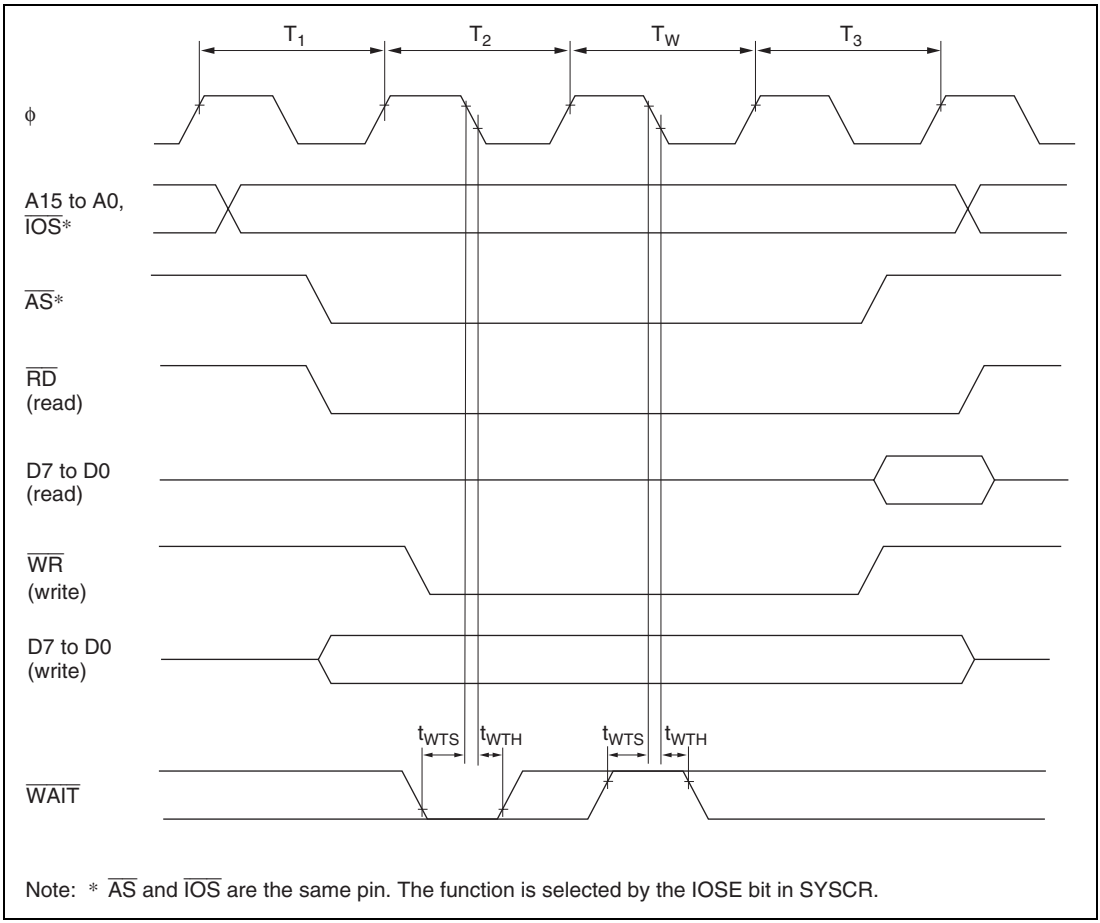
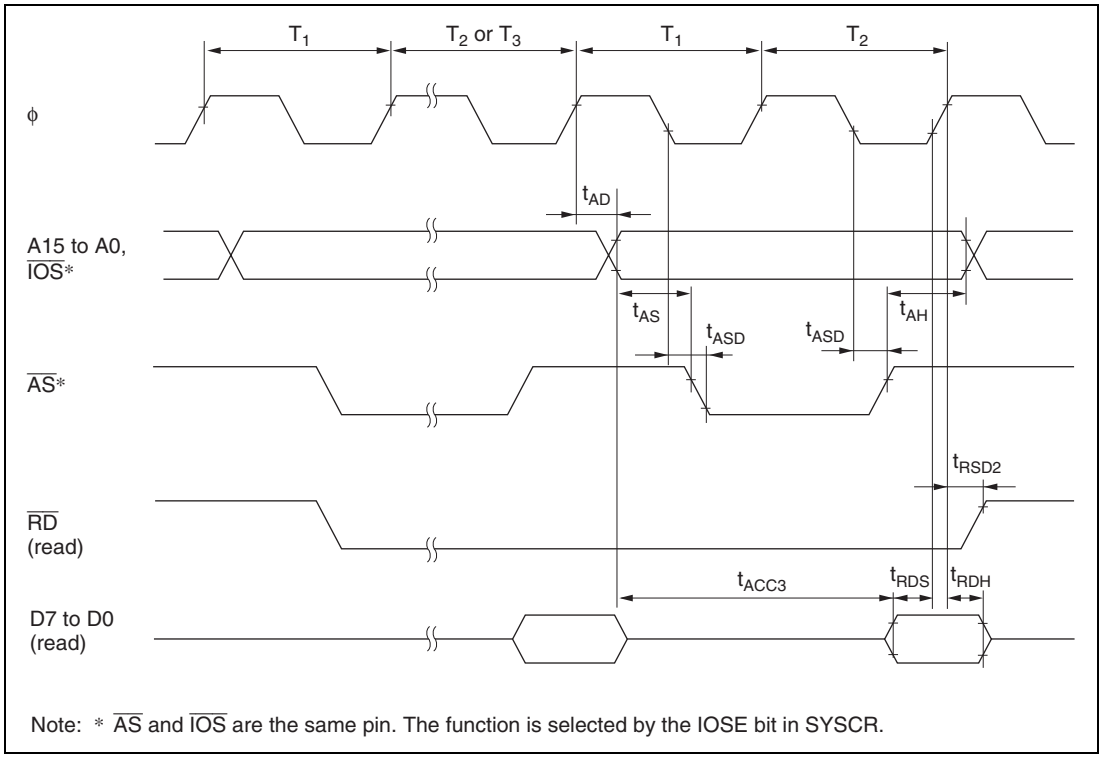
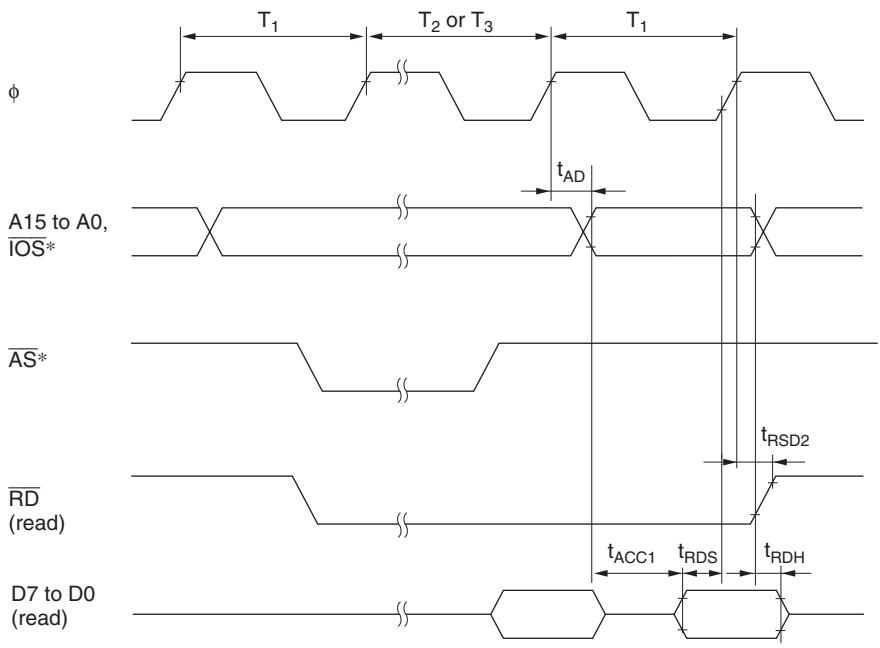


Figure 24.11 Basic Bus Timing (Three-State Access with One Wait State)



Note: * AS and IOS are the same pin. The function is selected by the IOSE bit in SYSCR.

Figure 24.12 Burst ROM Access Timing (Two-State Access)



Note: * $\overline{AS}$ and $\overline{IOS}$ are the same pin. The function is selected by the IOSE bit in SYSCR.

Figure 24.13 Burst ROM Access Timing (One-State Access)

24.3.4 Timing of On-Chip Peripheral Modules

Tables 24.8 and 24.9 show the on-chip peripheral module timing. The on-chip peripheral modules that can be operated by the subclock (ϕ = 32.768 kHz) are I/O ports, external interrupts (NMI, and IRQ0 to IRQ7), watchdog timer, and 8-bit timer (channels 0 and 1) only.

Table 24.8 Timing of On-Chip Peripheral Modules

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 32.768 kHz*, ϕ = 8 MHz to 20 MHz

Item			Symbol	Min.	Max.	Unit	Test Conditions
I/O ports	Output data delay time		t_{PWD}	—	50	ns	Figure 24.14
	Input data setup time		t_{PRS}	30	—		
	Input data hold time		t_{PRH}	30	—		
FRT	Timer output delay time		t_{FTOD}	—	50	ns	Figure 24.15
	Timer input setup time		t_{FTIS}	30	—		
	Timer clock input setup time		t_{FTCS}	30	—		Figure 24.16
	Timer clock pulse width	Single edge	t_{FTCWH}	1.5	—	t_{cyc}	
		Both edges	t_{FTCWL}	2.5	—		
TCM	Timer input setup time		t_{TCMIS}	30	—	ns	Figure 24.17
	Timer clock input setup time		t_{TCMCKS}	30	—		Figure 24.18
	Timer clock pulse width		t_{TCMCKW}	1.5	—	t_{cyc}	
TMR	Timer output delay time		t_{TMOD}	—	50	ns	Figure 24.19
	Timer reset input setup time		t_{TMRS}	30	—		Figure 24.21
	Timer clock input setup time		t_{TMCS}	30	—		Figure 24.20
	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	t_{cyc}	
		Both edges	t_{TMCWL}	2.5	—		
PWM, PWMX	Timer output delay time		t_{PWOD}	—	50	ns	Figure 24.22
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 24.23
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5		
	Transmit data delay time (synchronous)		t_{TXD}	—	50	ns	Figure 24.24
	Receive data setup time (synchronous)		t_{RXS}	50	—		
	Receive data hold time (synchronous)		t_{RXH}	50	—		
A/D converter	Trigger input setup time		t_{TRGS}	30	—	ns	Figure 24.25

Note: * Applied only for the peripheral modules that are available during subclock operation.

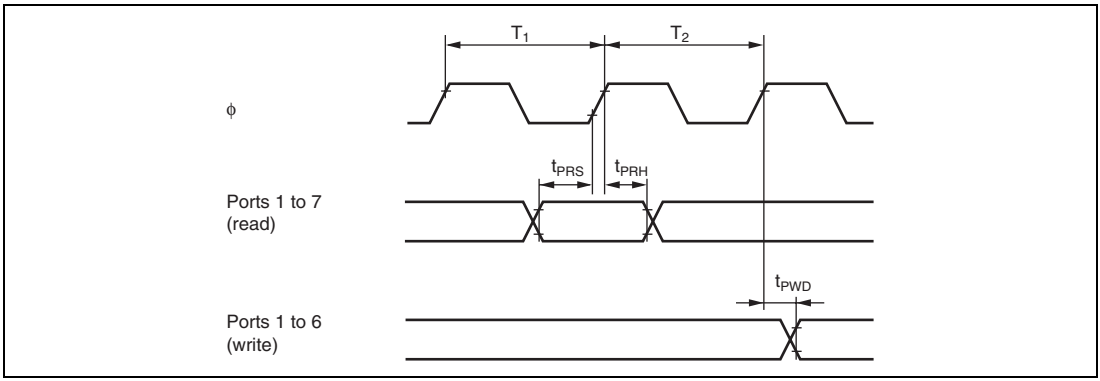


Figure 24.14 I/O Port Input/Output Timing

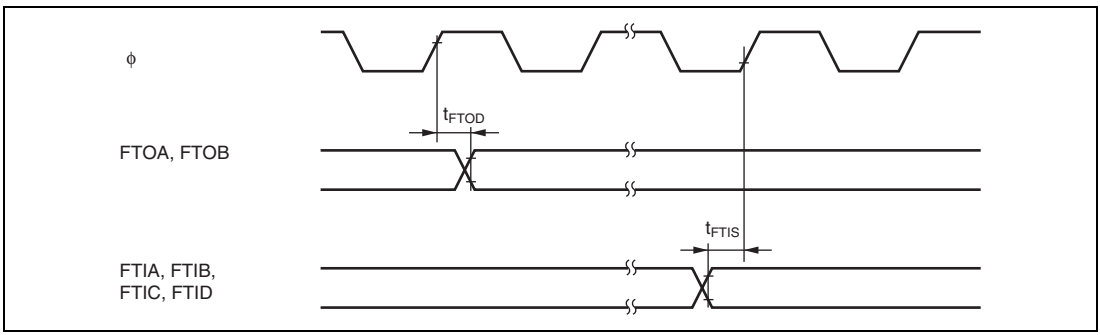


Figure 24.15 FRT Input/Output Timing

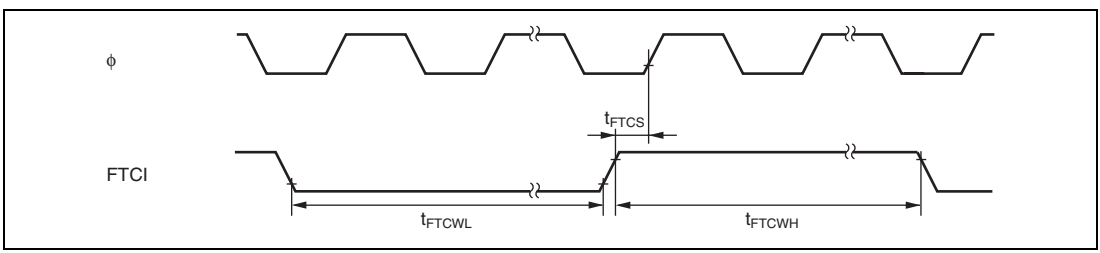


Figure 24.16 FRT Clock Input Timing

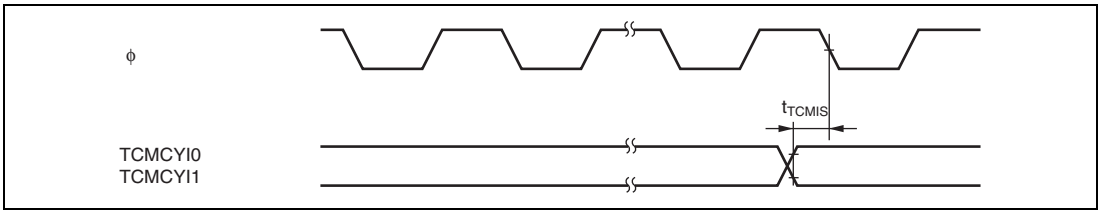


Figure 24.17 TCM Input/Output Timing

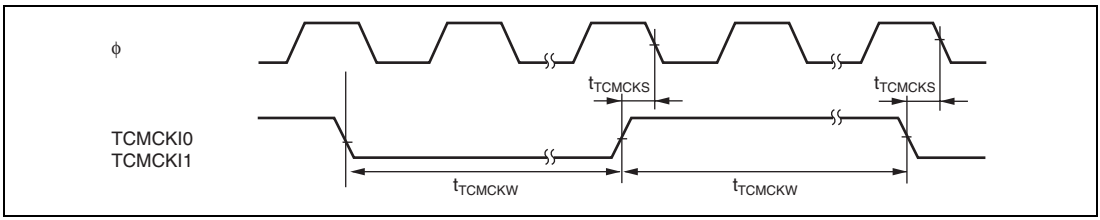


Figure 24.18 TCM Clock Input Timing

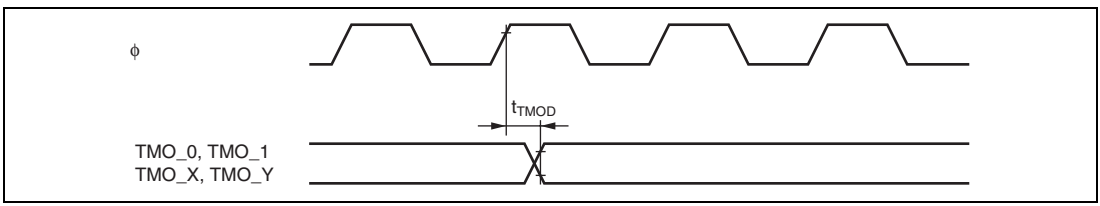


Figure 24.19 8-Bit Timer Output Timing

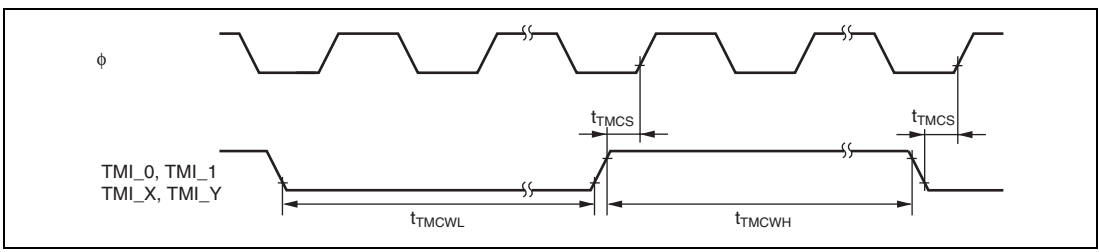


Figure 24.20 8-Bit Timer Clock Input Timing

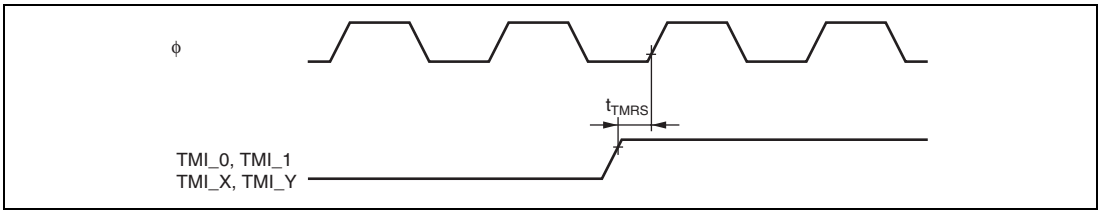


Figure 24.21 8-Bit Timer Reset Input Timing

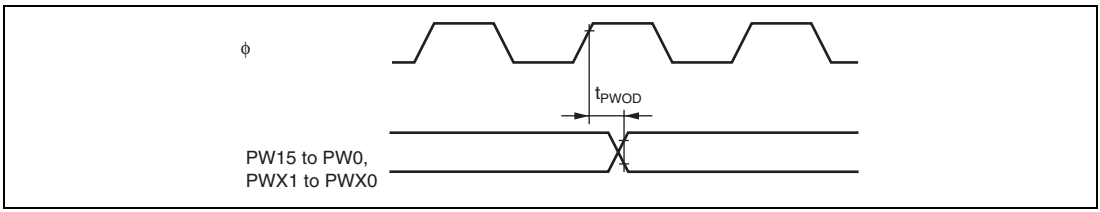


Figure 24.22 PWM, PWMX Output Timing

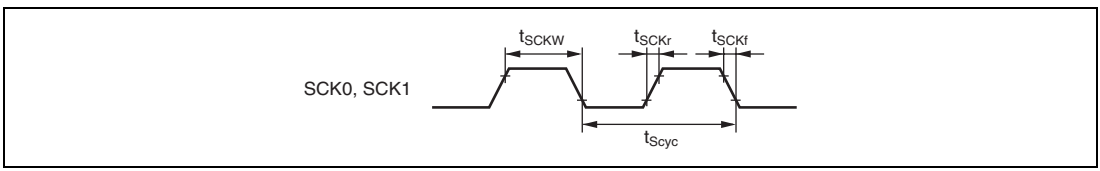


Figure 24.23 SCK Clock Input Timing

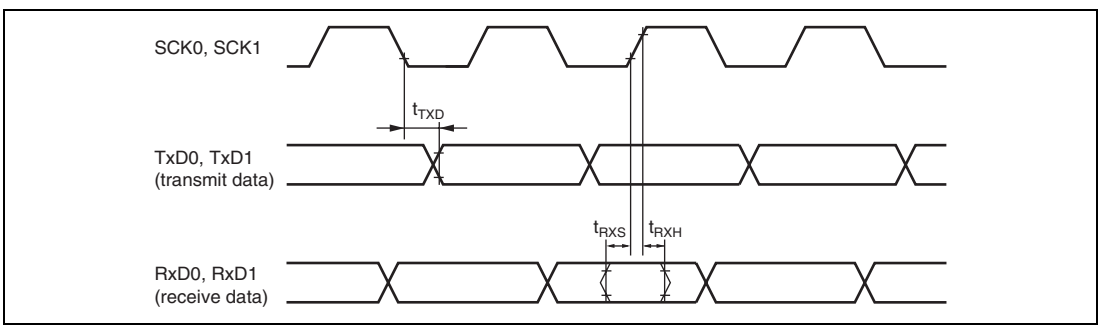


Figure 24.24 SCI Input/Output Timing (Clock Synchronous Mode)

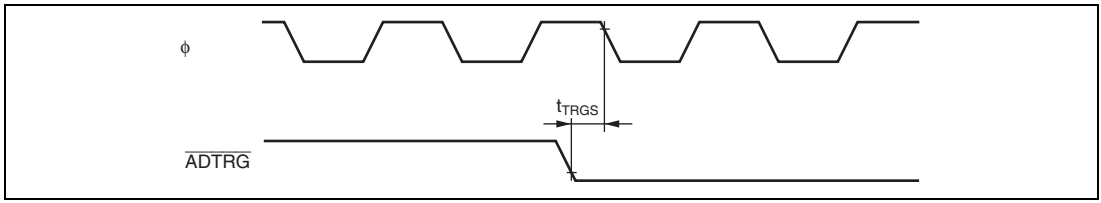


Figure 24.25 A/D Converter External Trigger Input Timing

Table 24.9 I^2C Bus Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 8 MHz to maximum operating frequency

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
SCL input cycle time	t_{SCL}	12	—	—	t_{cyc}	Figure 24.26
SCL input high pulse width	t_{SCLH}	3	—	—		
SCL input low pulse width	t_{SCLL}	5	—	—		
SCL, SDA input rise time	t_{Sr}	—	—	7.5*		
SCL, SDA input fall time	t_{Sf}	—	—	300	ns	
SCL, SDA output fall time	t_{Of}	—	—	250		
SCL, SDA input spike pulse elimination time	t_{SP}	—	—	1	t_{cyc}	
SDA input bus free time	t_{BUF}	5	—	—		
Start condition input hold time	t_{STAH}	3	—	—		
Retransmission start condition input setup time	t_{STAS}	3	—	—		
Stop condition input setup time	t_{STOS}	3	—	—		
Data input setup time	t_{SDAS}	0.5	—	—		
Data input hold time	t_{SDAH}	0	—	—	ns	
SCL, SDA capacitive load	C_b	—	—	400	pF	

Note: * 17.5 t_{cyc} can be set according to the clock selected for use by the I^2C module.

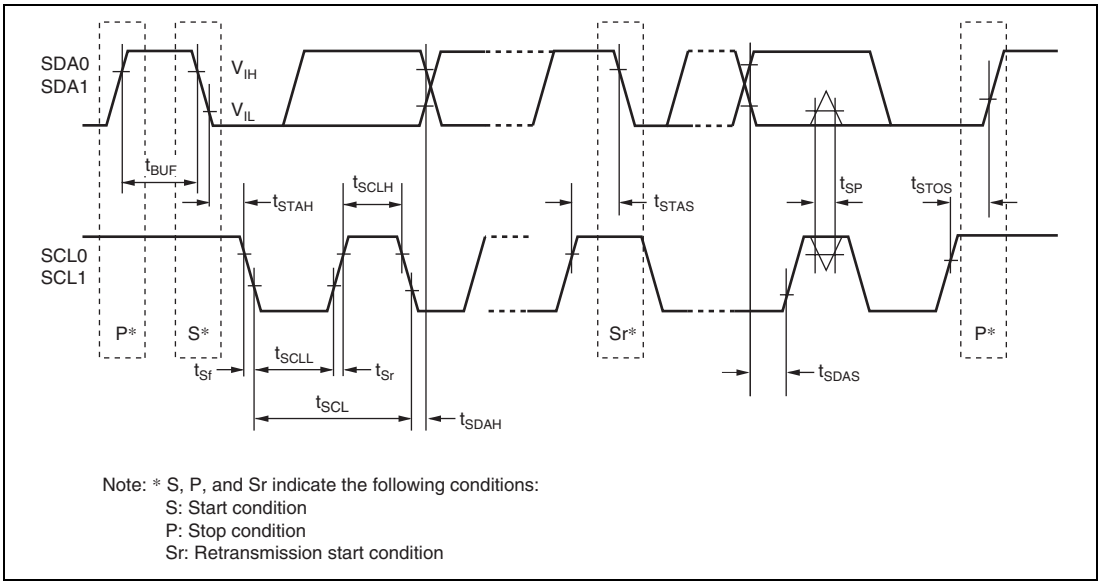


Figure 24.26 I²C Bus Interface Input/Output Timing

Table 24.10 H-UDI Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 8 MHz to 20 MHz

Item	Symbol	Min.	Max.	Unit	Test Conditions
ETCK clock cycle time	t_{TCKcyc}	50*	125*	ns	Figure 24.27
ETCK clock high pulse width	t_{TCKH}	20	—		
ETCK clock low pulse width	t_{TCKL}	20	—		
ETCK clock rise time	t_{TCKr}	—	5		
ETCK clock fall time	t_{TCKf}	—	5		
$\overline{\text{ETRST}}$ pulse width	t_{TRSTW}	20	—	t_{cyc}	Figure 24.28
Reset hold transition pulse width	t_{RSTHW}	3	—		
ETMS setup time	t_{TMSS}	20	—	ns	Figure 24.29
ETMS hold time	t_{TMSH}	20	—		
ETDI setup time	t_{TDIS}	20	—		
ETDI hold time	t_{TDIH}	20	—		
ETDO data delay time	t_{TDOD}	—	20		

Note: * When $t_{cyc} \leq t_{TCKcyc}$

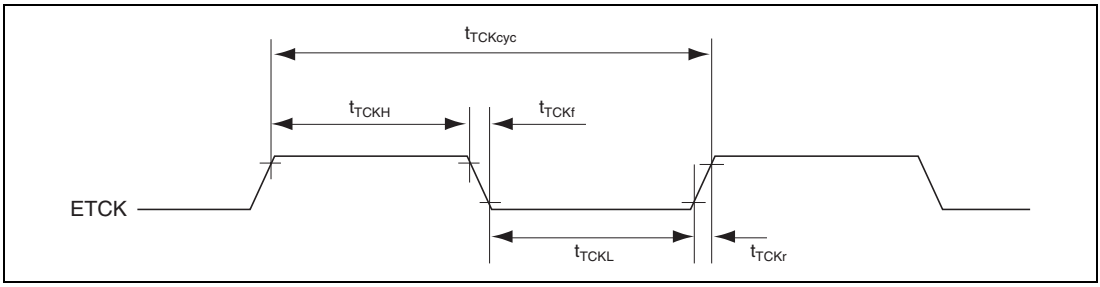


Figure 24.27 ETCK Timing

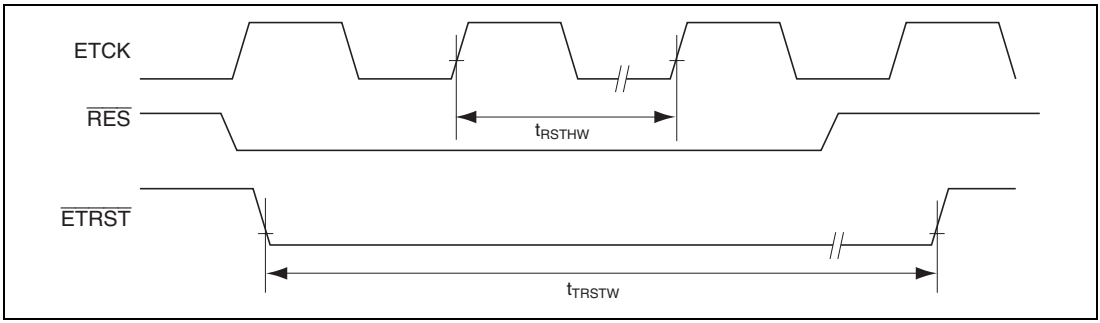


Figure 24.28 Reset Hold Timing

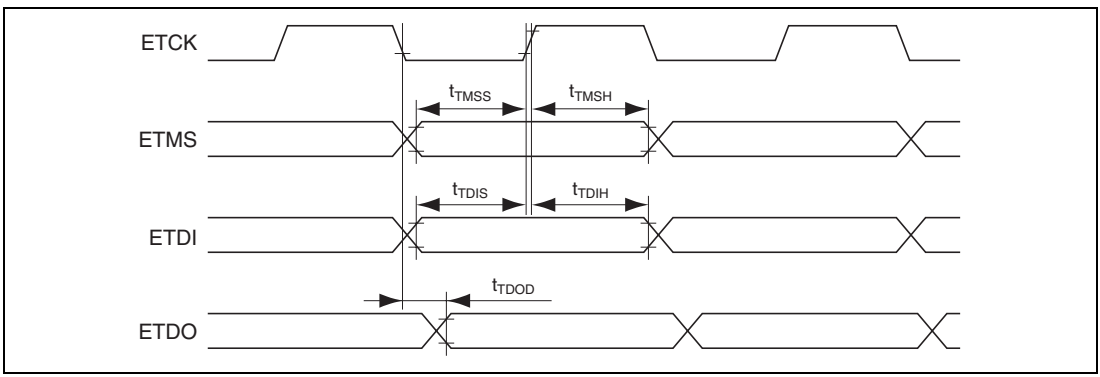


Figure 24.29 H-UDI Input/Output Timing

24.4 A/D Conversion Characteristics

Table 24.11 lists the A/D conversion characteristics.

Table 24.11 A/D Conversion Characteristics (AN7 to AN0 Input: 134/266-State Conversion)

Condition A: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 16 MHz

Condition B: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 20 MHz

	Condition A			Condition B			
Item	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Unit**
Resolution		10			10		Bits
Conversion time	—	—	8.38*1	—	—	13.4*2	µs
Analog input capacitance	—	—	20	—	—	20	pF
Permissible signal-source impedance	—	—	5	—	—	5	kΩ
Nonlinearity error	—	—	±7.0	—	—	±7.0	LSB
Offset error	—	—	±7.5	—	—	±7.5	
Full-scale error	—	—	±7.5	—	—	±7.5	
Quantization error	—	—	±0.5	—	—	±0.5	
Absolute accuracy	—	—	±8.0	—	—	±8.0	

Notes: 1. Value when using the maximum operating frequency in single mode of 134 states.
2. Value when using the maximum operating frequency in single mode of 266 states.

24.5 Flash Memory Characteristics

Table 24.12 lists the flash memory characteristics.

Table 24.12 Flash Memory Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{SS} = AV_{SS} = 0 V
Ta = 0°C to +75°C (operating temperature range for programming/erasing)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Programming time*1*2*4	t_P	—	3	30	ms/128 bytes	
Erase time*1*2*4	t_E	—	80	800	ms/4-Kbyte block	
		—	500	5000	ms/32-Kbyte block	
		—	1000	10000	ms/64-Kbyte block	
Programming time (total)*1*2*4	Σt_P	—	10	30	s/512-Kbyte	Ta = 25°C
Erase time (total)*1*2*4	Σt_E	—	10	30	s/512-Kbyte	Ta = 25°C
Programming and Erase time (total)*1*2*4	Σt_{PE}	—	20	60	s/512-Kbyte	Ta = 25°C
Reprogramming count	N_{WEC}	100*3	—	—	Times	
Data retention time*4	t_{DRP}	10	—	—	Years	

Notes: 1. Programming and erase time depends on the data.
2. Programming and erase time do not include data transfer time.
3. This value indicates the minimum number of which the flash memory are reprogrammed with all characteristics guaranteed. (The guaranteed value ranges from 1 to the minimum number.)
4. This value indicates the characteristics while the flash memory is reprogrammed within the specified range (including the minimum number).

24.6 Usage Notes

It is necessary to connect a bypass capacitor between the VCC pin and VSS pin, and a capacitor between the VCL pin and VSS pin for stable internal step-down power. An example of connection is shown in figure 24.30.

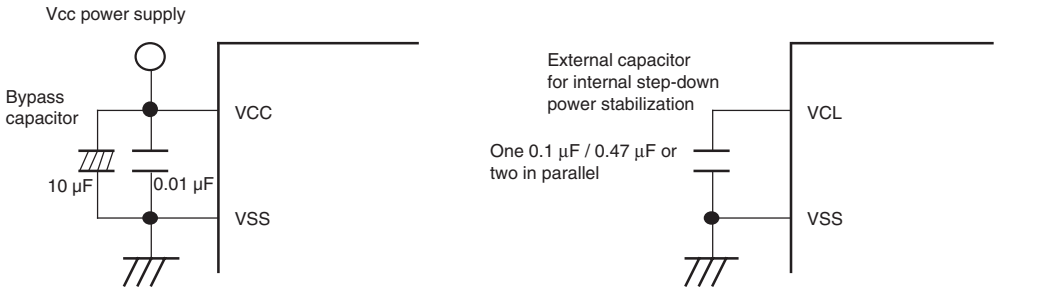


Figure 24.30 Connection of VCL Capacitor

Appendix

A. I/O Port States in Each Processing State

Table A.1 I/O Port States in Each Processing State

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Watch Mode	Sleep Mode	Sub-sleep Mode	Subactive Mode	Program Execution State
Port 1 A7 to A0	1	L	T	Keep*	Keep*	Keep*	Keep*	A7 to A0	A7 to A0
	2, 3 (EXPE = 1)	T						Address output/ input port	Address output/ input port
	2, 3 (EXPE = 0)							I/O port	I/O port
Port 2 A15 to A8	1	L	T	Keep*	Keep*	Keep*	Keep*	A15 to A8	A15 to A8
	2, 3 (EXPE = 1)	T						Address output/ input port	Address output/ input port
	2, 3 (EXPE = 0)							I/O port	I/O port
Port 3 D7 to D0	1	T	T	T	T	T	T	D7 to D0	D7 to D0
	2, 3 (EXPE = 1)								
	2, 3 (EXPE = 0)			Keep	Keep	Keep	Keep	I/O port	I/O port
Port 47 $\overline{\text{WAIT}}$	1	T	T	T/Keep	T/Keep	T/Keep	T/Keep	$\overline{\text{WAIT}}$/ I/O port	$\overline{\text{WAIT}}$/ I/O port
	2, 3 (EXPE = 1)								
	2, 3 (EXPE = 0)			Keep	Keep	Keep	Keep	I/O port	I/O port
Port 46 ϕ EXCL	1	Clock output	T	[DDR = 1] H [DDR = 0] T	EXCL input	[DDR = 1] clock output [DDR = 0] T	EXCL input	EXCL input	Clock output/ EXCL input/ input port
	2, 3 (EXPE = 1)	T							
	2, 3 (EXPE = 0)								
Ports 45 to 43 $\overline{\text{AS}}$, $\overline{\text{WR}}$, $\overline{\text{RD}}$	1	H	T	H	H	H	H	$\overline{\text{AS}}$, $\overline{\text{WR}}$, $\overline{\text{RD}}$	$\overline{\text{AS}}$, $\overline{\text{WR}}$, $\overline{\text{RD}}$
	2, 3 (EXPE = 1)	T							
	2, 3 (EXPE = 0)			Keep	Keep	Keep	Keep	I/O port	I/O port
Ports 42 to 40	1	T	T	Keep	Keep	Keep	Keep	I/O port	I/O port
	2, 3 (EXPE = 1)								
	2, 3 (EXPE = 0)								
Port 5	1	T	T	Keep	Keep	Keep	Keep	I/O port	I/O port
	2, 3 (EXPE = 1)								
	2, 3 (EXPE = 0)								

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Watch Mode	Sleep Mode	Sub-sleep Mode	Subactive Mode	Program Execution State
Port 6	1	T	T	Keep	Keep	Keep	Keep	I/O port	I/O port
	2, 3 (EXPE = 1)								
	2, 3 (EXPE = 0)								
Port 7	1	T	T	T	T	T	T	I/O port	I/O port
	2, 3 (EXPE = 1)								
	2, 3 (EXPE = 0)								

[Legend]

H: High

L: Low

T: High-impedance state

Keep: Input ports are in the high-impedance state (when DDR = 0 and PCR = 1, input pull-up MOSs remain on).

Output ports maintain their previous state.

Depending on the pins, the on-chip peripheral modules may be initialized and the I/O port function determined by DDR and DR used.

DDR: Data direction register

Note: * In the case of address output, the last address accessed is retained.

B. Product Codes

Product Type		Product Code	Mark Code	Package (Package Code)
H8S/2125	F-ZTAT Version	R4F2125	F2125VPS20	PRDP0064BB-A (DP-64S)
			F2125VFA20	PRQP0064GB-A (FP-64A)
			F2125VTF20	PTQP0080KC-A (TFP-80C)
	PROM (OTP Version)	R4P2125	P2125VPS20	PRDP0064BB-A (DP-64S)

C. Package Dimensions

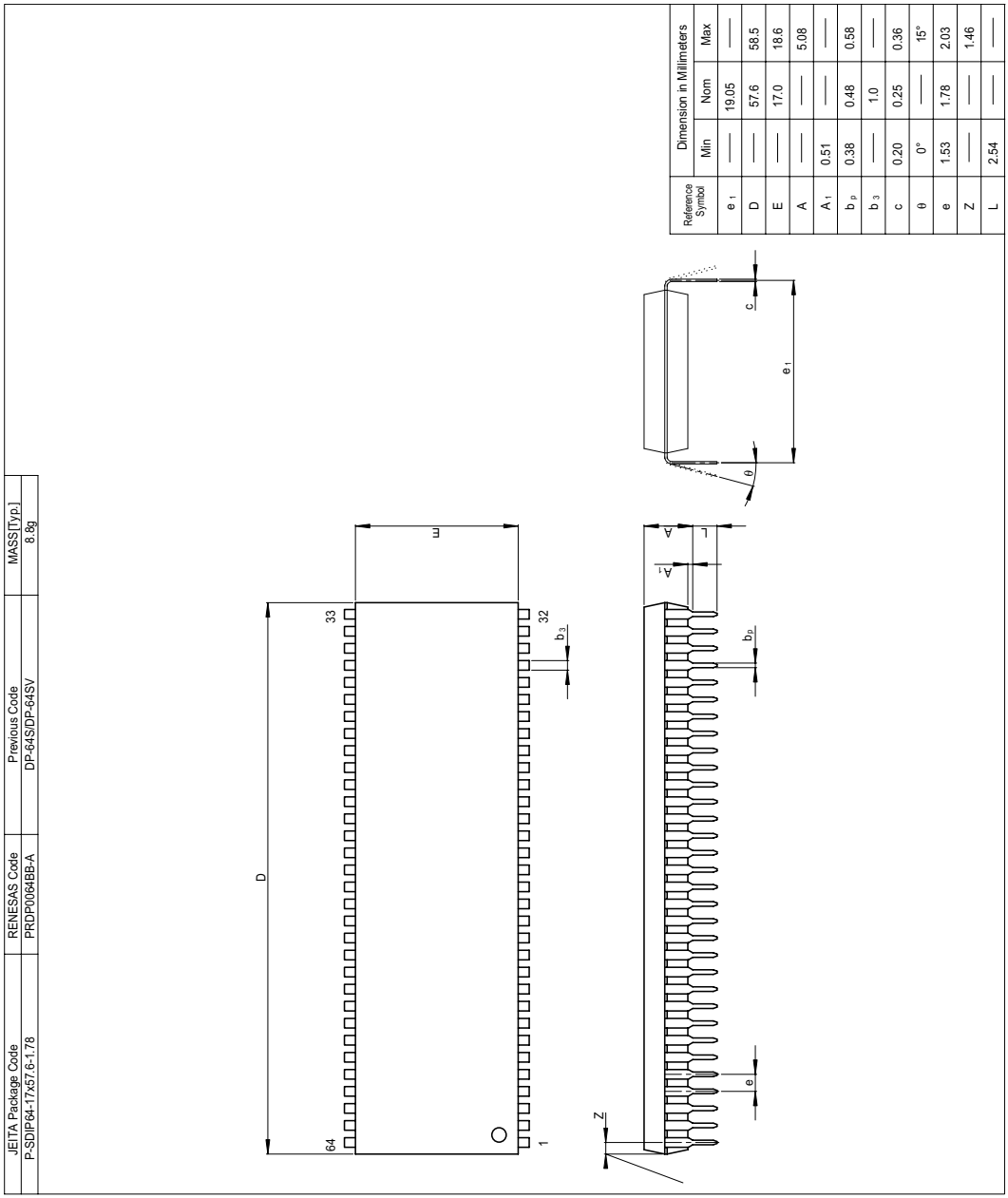


Figure C.1 Package Dimensions (SDIP-64)

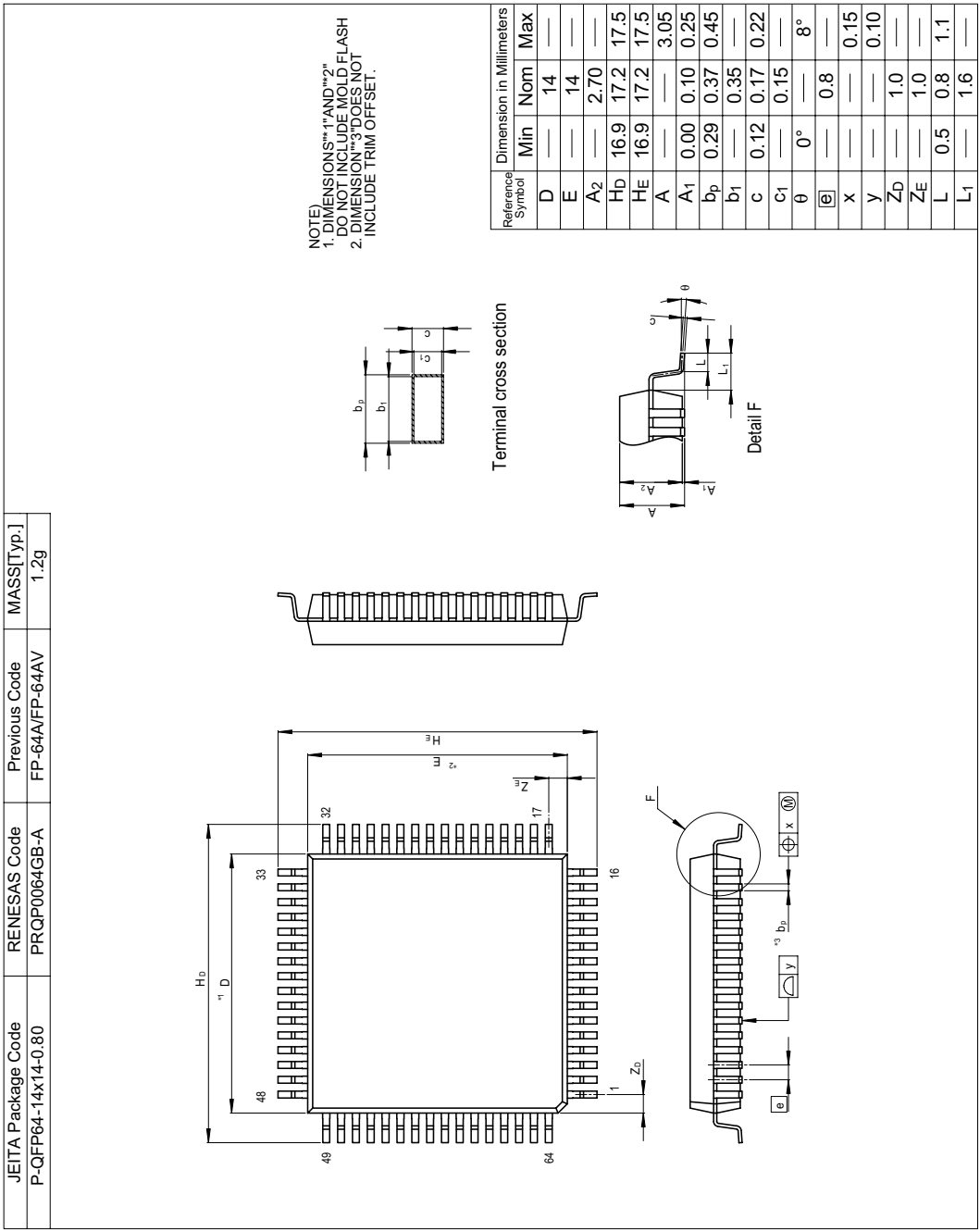


Figure C.2 Package Dimensions (QFP-64)

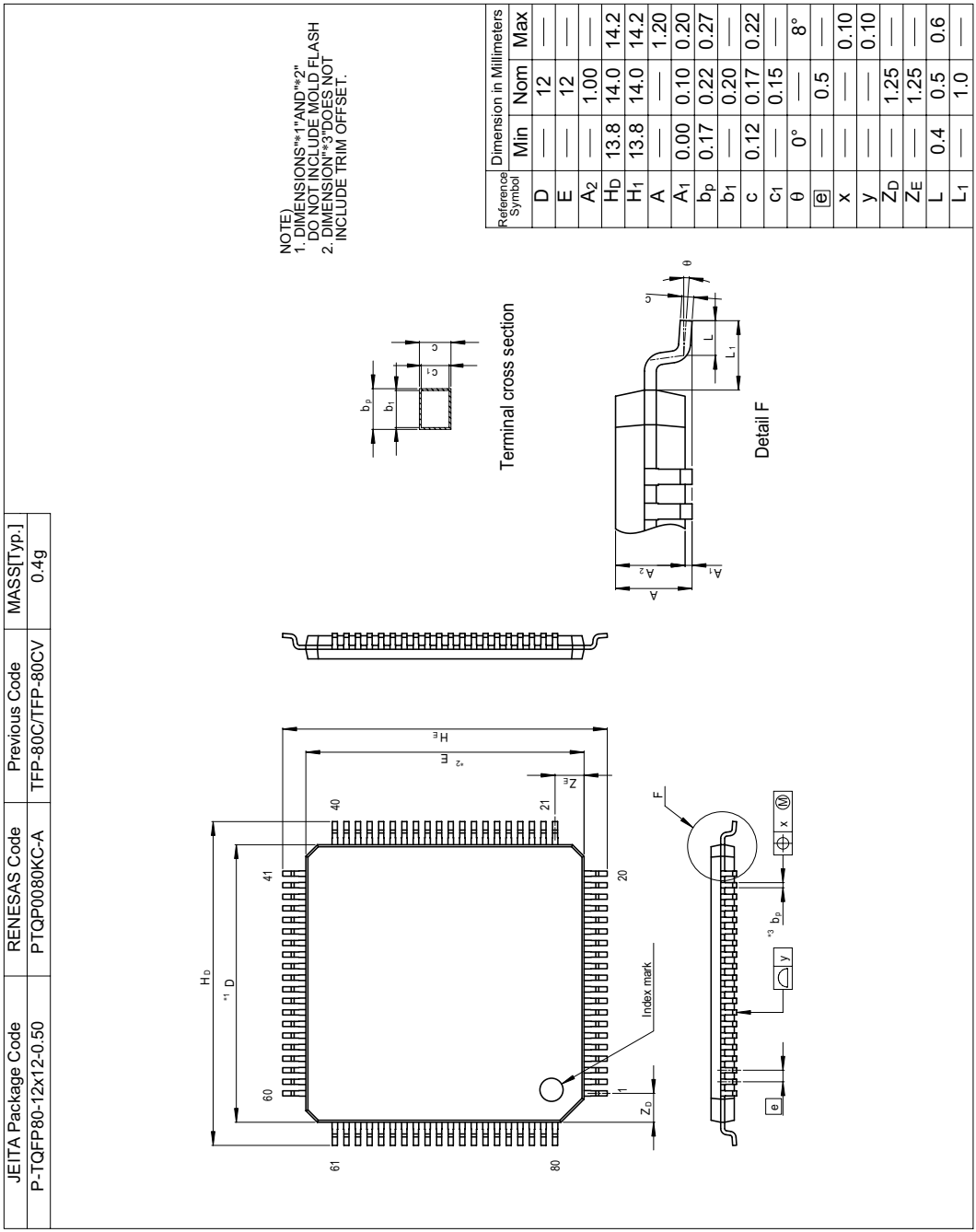


Figure C.3 Package Dimensions (TQFP-80)

Index

P

R

S

T

U

V

W

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2125 Group

Publication Date: Rev.1.00, Sep. 21, 2006
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

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

H8S/2125 Group
Hardware Manual

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan