16

H8S/2437 Group

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer

H8S Family / H8S / 2600 Series

H8S/2437 HD64F2437

User's Manual

Rev.1.00
2003.9.19

Renesas Technology
www.renesas.com

Cautions

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)

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

This LSI is a microcomputer (MCU) made up of the H8S/2600 CPU with Renesas Technology-original architecture as its core, and the peripheral functions required to configure a system.

The H8S/2600 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/2600 CPU can handle a 16-Mbyte linear address space. The instruction set of the H8S/2600 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/2600 CPU.

This LSI is equipped with the flash memory, RAM, two kinds of PWM timers (PWM and PWMX), a 16-bit free-running timer (FRT), an 8-bit timer (TMR), a 16-bit timer pulse unit (TPU), a watchdog timer (WDT), a timer connection, a serial communication interface (SCI), an I^2C bus interface 3 (IIC3), an A/D converter, and I/O ports as on-chip peripheral modules required for system configuration.

A flash memory (F-ZTAT™*) version is available for this LSI's 256-kbyte ROM. The CPU and the flash memory 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 this LSI 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 this LSI to the target users.
Refer to the H8S/2600 Series, H8S/2000 Series Programming 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 Programming 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/eng/

H8S/2437 Group manuals:

Document Title	Document No.
H8S/2437 Group Hardware Manual	This manual
H8S/2600 Series, H8S/2000 Series Programming Manual	ADE-602-083

User's manuals for development tools:

Document Title	Document No.
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	ADE-702-247
H8S, H8/300 Series Simulator/Debugger User's Manual	ADE-702-282
H8S, H8/300 Series High-performance Embedded Workshop, High-performance Debugging Interface Tutorial	ADE-702-231
High-performance Embedded Workshop User's Manual	ADE-702-201

Contents

Figures

Section 8 8-Bit PWM Timer (PWM)

Section 9 14-Bit PWM Timer (PWMX)

Section 10 16-Bit Free-Running Timer (FRT)

Section 11 8-Bit Timer (TMR)

Section 12 16-Bit Timer Pulse Unit (TPU)

Section 13 Timer Connection

Section 14 Duty Measurement Circuit

Section 15 Watchdog Timer (WDT)

Section 16 Serial Communication Interface (SCI)

Section 17 I^2C Bus Interface 3 (IIC3)

Section 18 A/D Converter

Section 20 Flash Memory (0.18-μm F-ZTAT Version)

Appendix

Tables

Section 12 16-Bit Timer Pulse Unit (TPU)

Section 13 Timer Connection

Section 1 Overview

1.1 Features

- High-speed H8S/2600 central processing unit with an internal 16-bit architecture
  Upward-compatible with H8/300 and H8/300H CPUs on an object level
  Sixteen 16-bit general registers
  69 basic instructions
  Multiply-and-accumulate instruction
- Various peripheral functions
  8-bit PWM timer (PWM)
  14-bit PWM timer (PWMX)
  16-bit free-running timer (FRT)
  8-bit timer (TMR)
  16-bit timer pulse unit (TPU)
  Watchdog timer (WDT)
  Timer connection
  Duty measurement circuit
  Asynchronous or clocked synchronous serial communication interface (SCI)
  I^2C bus interface 3 (IIC3)
  10-bit A/D converter
- On-chip memory

ROM Type	Model	ROM	RAM	Remarks
Flash memory version	HD64F2437	256 kbytes	16 kbytes	

- General I/O ports
  I/O pins: 94
  Input-only pins: 16
- Supports various power-down modes
- Compact package

Package	Code	Body Size	Pin Pitch
QFP-128	FP-128B	14.0 × 20.0 mm	0.5 mm

1.2 Internal Block Diagram

Figure 1.1 shows the internal block diagram of the H8S/2437 Group.

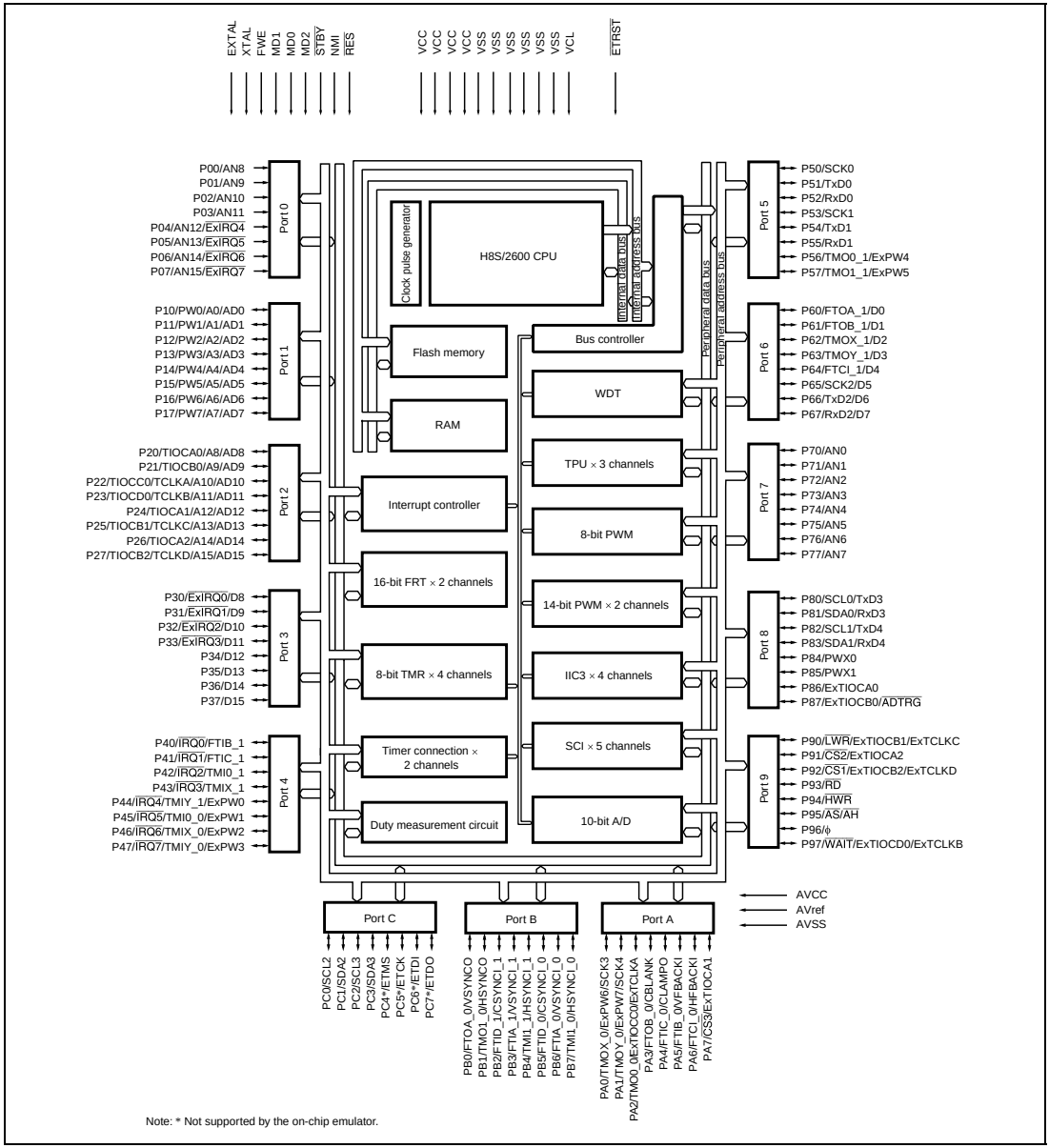


Figure 1.1 Internal Block Diagram of H8S/2437 Group

1.3 Pin Description

1.3.1 Pin Assignment

Figure 1.2 shows the pin assignment of the H8S/2437 Group.

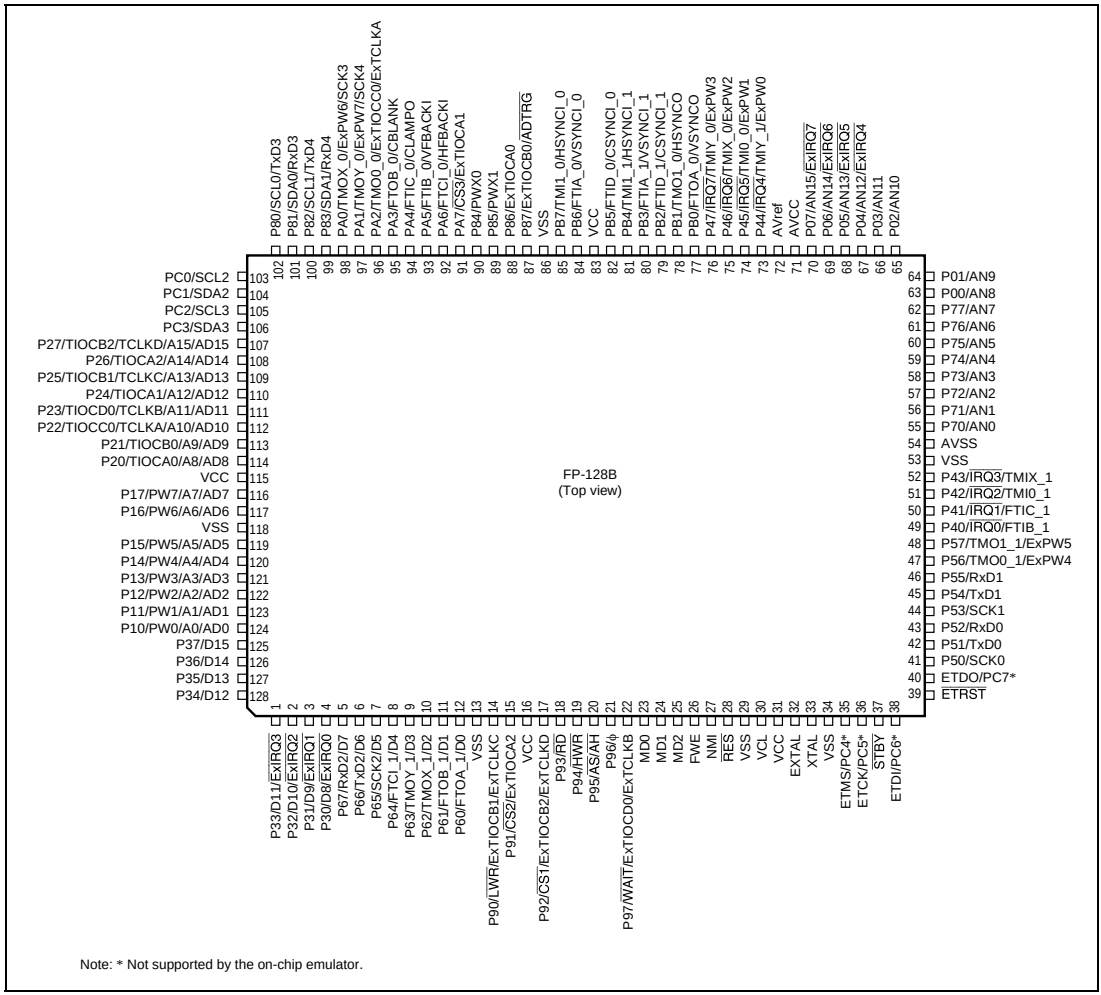


Figure 1.2 Pin Assignment of H8S/2437 Group (FP-128B)

1.3.2 Pin Assignment in Each Operating Mode

Table 1.1 Pin Assignment in Each Operating Mode

Pin No.	Pin Name			
QFP-128	Extended Mode (EXPE = 1)		Single-Chip Mode	Flash Memory Programmer Mode
	Normal	Multiplex	(EXPE = 0)	
1	D11	P33/$\overline{\text{ExIRQ3}}$	P33/$\overline{\text{ExIRQ3}}$	D3
2	D10	P32/$\overline{\text{ExIRQ2}}$	P32/$\overline{\text{ExIRQ2}}$	D2
3	D9	P31/$\overline{\text{ExIRQ1}}$	P31/$\overline{\text{ExIRQ1}}$	D1
4	D8	P30/$\overline{\text{ExIRQ0}}$	P30/$\overline{\text{ExIRQ0}}$	D0
5	D7	P67/RxD2	P67/RxD2	NC
6	D6	P66/TxD2	P66/TxD2	NC
7	D5	P65/SCK2	P65/SCK2	NC
8	D4	P64/FTCI_1	P64/FTCI_1	NC
9	D3	P63/TMOY_1	P63/TMOY_1	NC
10	D2	P62/TMOX_1	P62/TMOX_1	NC
11	D1	P61/FTOB_1	P61/FTOB_1	NC
12	D0	P60/FTOA_1	P60/FTOA_1	NC
13	VSS		VSS	VSS
14	P90/$\overline{\text{LWR}}$/ExTIOCB1/ExTCLKC		P90/ExTIOCB1/ExTCLKC	A16
15	P91/$\overline{\text{CS2}}$/ExTIOCA2		P91/ExTIOCA2	A17
16	VCC		VCC	VCC
17	P92/$\overline{\text{CS1}}$/ExTIOCB2/ExTCLKD		P92/ExTIOCB2/ExTCLKD	A18
18	$\overline{\text{RD}}$		P93	NC
19	$\overline{\text{HWR}}$		P94	NC
20	P95/$\overline{\text{AS}}$	$\overline{\text{AH}}$	P95	NC
21	P96/ϕ		P96/ϕ	NC
22	P97/$\overline{\text{WAIT}}$/ExTIOCD0/ExTCLKB		P97/ExTIOCD0/ExTCLKB	NC
23	MD0		MD0	VSS
24	MD1		MD1	VCC
25	MD2		MD2	VCC
26	FWE		FWE	VCC
27	NMI		NMI	VCC

Pin No.	Pin Name			
QFP-128	Extended Mode (EXPE = 1)		Single-Chip Mode	Flash Memory Programmer Mode
	Normal	Multiplex	(EXPE = 0)	
28	$\overline{\text{RES}}$		$\overline{\text{RES}}$	$\overline{\text{RES}}$
29	VSS		VSS	VSS
30	VCL		VCL	VCL
31	VCC		VCC	VCC
32	EXTAL		EXTAL	EXTAL
33	XTAL		XTAL	XTAL
34	VSS		VSS	VSS
35	ETMS/PC4*		ETMS/PC4*	NC
36	ETCK/PC5*		ETCK/PC5*	NC
37	$\overline{\text{STBY}}$		$\overline{\text{STBY}}$	VCC
38	ETDI/PC6*		ETDI/PC6*	NC
39	$\overline{\text{ETRST}}$		$\overline{\text{ETRST}}$	$\overline{\text{RES}}$
40	ETDO/PC7*		ETDO/PC7*	NC
41	P50/SCK0		P50/SCK0	NC
42	P51/TxD0		P51/TxD0	NC
43	P52/RxD0		P52/RxD0	NC
44	P53/SCK1		P53/SCK1	NC
45	P54/TxD1		P54/TxD1	NC
46	P55/RxD1		P55/RxD1	NC
47	P56/TMO0_1/ExPW4		P56/TMO0_1/ExPW4	NC
48	P57/TMO1_1/ExPW5		P57/TMO1_1/ExPW5	NC
49	P40/$\overline{\text{IRQ0}}$/FTIB_1		P40/$\overline{\text{IRQ0}}$/FTIB_1	VCC
50	P41/$\overline{\text{IRQ1}}$/FTIC_1		P41/$\overline{\text{IRQ1}}$/FTIC_1	VCC
51	P42/$\overline{\text{IRQ2}}$/TMI0_1		P42/$\overline{\text{IRQ2}}$/TMI0_1	VCC
52	P43/$\overline{\text{IRQ3}}$/TMIX_1		P43/$\overline{\text{IRQ3}}$/TMIX_1	VCC
53	VSS		VSS	VSS
54	AVSS		AVSS	VSS
55	P70/AN0		P70/AN0	NC
56	P71/AN1		P71/AN1	NC
57	P72/AN2		P72/AN2	NC
58	P73/AN3		P73/AN3	NC

Pin No.	Pin Name			
QFP-128	Extended Mode (EXPE = 1)		Single-Chip Mode	Flash Memory Programmer Mode
	Normal	Multiplex	(EXPE = 0)	
59	P74/AN4		P74/AN4	NC
60	P75/AN5		P75/AN5	NC
61	P75/AN6		P75/AN6	NC
62	P77/AN7		P77/AN7	NC
63	P00/AN8		P00/AN8	$\overline{WE}$
64	P01/AN9		P01/AN9	$\overline{OE}$
65	P02/AN10		P02/AN10	$\overline{CE}$
66	P03/AN11		P03/AN11	NC
67	P04/AN12/$\overline{ExIRQ4}$		P04/AN12/$\overline{ExIRQ4}$	NC
68	P05/AN13/$\overline{ExIRQ5}$		P05/AN13/$\overline{ExIRQ5}$	NC
69	P06/AN14/$\overline{ExIRQ6}$		P06/AN14/$\overline{ExIRQ6}$	NC
70	P07/AN15/$\overline{ExIRQ7}$		P07/AN15/$\overline{ExIRQ7}$	NC
71	AVCC		AVCC	VCC
72	AVref		AVref	VCC
73	P44/$\overline{IRQ4}$/TMIY_1/ExPW0		P44/$\overline{IRQ4}$/TMIY_1/ExPW0	VCC
74	P45/$\overline{IRQ5}$/TMI0_0/ExPW1		P45/$\overline{IRQ5}$/TMI0_0/ExPW1	VCC
75	P46/$\overline{IRQ6}$/TMIX_0/ExPW2		P46/$\overline{IRQ6}$/TMIX_0/ExPW2	VCC
76	P47/$\overline{IRQ7}$/TMIY_0/ExPW3		P47/$\overline{IRQ7}$/TMIY_0/ExPW3	VCC
77	PB0/FTOA_0/VSYNCO		PB0/FTOA_0/VSYNCO	NC
78	PB1/TMO1_0/HSYNCO		PB1/TMO1_0/HSYNCO	NC
79	PB2/FTID_1/CSYNCI_1		PB2/FTID_1/CSYNCI_1	NC
80	PB3/FTIA_1/VSYNCI_1		PB3/FTIA_1/VSYNCI_1	NC
81	PB4/TMI1_1/HSYNCI_1		PB4/TMI1_1/HSYNCI_1	NC
82	PB5/FTID_0/CSYNCI_0		PB5/FTID_0/CSYNCI_0	NC
83	VCC		VCC	VCC
84	PB6/FTIA_0/VSYNCI_0		PB6/FTIA_0/VSYNCI_0	NC
85	PB7/TMI1_0/HSYNCI_0		PB7/TMI1_0/HSYNCI_0	NC
86	VSS		VSS	VSS
87	P87/ExTIOCB0/$\overline{ADTRG}$		P87/ExTIOCB0/$\overline{ADTRG}$	NC
88	P86/ExTIOCA0		P86/ExTIOCA0	NC
89	P85/PWX1		P85/PWX1	NC

Pin No.	Pin Name			
QFP-128	Extended Mode (EXPE = 1)		Single-Chip Mode	Flash Memory Programmer Mode
	Normal	Multiplex	(EXPE = 0)	
90	P84/PWX0		P84/PWX0	NC
91	PA7/$\overline{CS3}$/ExTIOCA1		PA7/ExTIOCA1	NC
92	PA6/FTCI_0/HFBACKI		PA6/FTCI_0/HFBACKI	NC
93	PA5/FTIB_0/VFBACKI		PA5/FTIB_0/VFBACKI	NC
94	PA4/FTIC_0/CLAMPO		PA4/FTIC_0/CLAMPO	NC
95	PA3/FTOB_0/CBLANK		PA3/FTOB_0/CBLANK	NC
96	PA2/TMO0_0/ExTIOCC0/ ExTCLKA		PA2/TMO0_0/ExTIOCC0/ ExTCLKA	VSS
97	PA1/TMOY_0/ExPW7/SCK4		PA1/TMOY_0/ExPW7/SCK4	NC
98	PA0/TMOX_0/ExPW6/SCK3		PA0/TMOX_0/ExPW6/SCK3	VCC
99	P83/SDA1/RxD4		P83/SDA1/RxD4	NC
100	P82/SCL1/TxD4		P82/SCL1/TxD4	NC
101	P81/SDA0/RxD3		P81/SDA0/RxD3	NC
102	P80/SCL0/TxD3		P80/SCL0/TxD3	NC
103	PC0/SCL2		PC0/SCL2	NC
104	PC1/SDA2		PC1/SDA2	NC
105	PC2/SCL3		PC2/SCL3	NC
106	PC3/SDA3		PC3/SDA3	NC
107	P27/A15	AD15	P27/TIOCB2/TCLKD	A15
108	P26/A14	AD14	P26/TIOCA2	A14
109	P25/A13	AD13	P25/TIOCB1/TCLKC	A13
110	P24/A12	AD12	P24/TIOCA1	A12
111	P23/A11	AD11	P23/TIOCD0/TCLKB	A11
112	P22/A10	AD10	P22/TIOCC0/TCLKA	A10
113	P21/A9	AD9	P21/TIOCB0	A9
114	P20/A8	AD8	P20/TIOCA0	A8
115	VCC		VCC	VCC
116	P17/A7	AD7	P17/PW7	A7
117	P16/A6	AD6	P16/PW6	A6
118	VSS		VSS	VSS
119	P15/A5	AD5	P15/PW5	A5

Pin No.	Pin Name			
QFP-128	Extended Mode (EXPE = 1)		Single-Chip Mode	Flash Memory Programmer Mode
	Normal	Multiplex	(EXPE = 0)	
120	P14/A4	AD4	P14/PW4	A4
121	P13/A3	AD3	P13/PW3	A3
122	P12/A2	AD2	P12/PW2	A2
123	P11/A1	AD1	P11/PW1	A1
124	P10/A0	AD0	P10/PW0	A0
125	D15	P37	P37	D7
126	D14	P36	P36	D6
127	D13	P35	P35	D5
128	D12	P34	P34	D4

Note: * Not supported by the on-chip emulator.

1.3.3 Pin Functions

Table 1.2 Pin Functions

Type	Symbol	Pin No.	I/O	Name and Function
Power supply	VCC	16, 31, 83, 115	Input	Power supply pins. Connect all these pins to the system power supply.
	VCL	30	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.
	VSS	13, 29, 34, 53, 86, 118	Input	Ground pins. Connect all these pins to the system power supply (0V).
Clock	XTAL	33	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	32	Input	
	ϕ	21	Output	Supplies the system clock to external devices.
Operating mode control	MD2 MD1 MD0	25 24 23	Input	These pins set the operating mode. Inputs at these pins should not be changed during operation.
System control	$\overline{\text{RES}}$	28	Input	Reset pin. When this pin is low, the chip is reset.
	$\overline{\text{STBY}}$	37	Input	When this pin is low, a transition is made to hardware standby mode.
	FWE	26	Input	Pin for use by flash memory.
Address bus	A15 to A8 A7 to A0	107 to 114, 116, 117, 119 to 124	Output	Address output pins
Data bus	D15 to D8 D7 to D0	125 to 4 5 to 12	I/O	Bidirectional data bus
Address/ data multiplex bus	AD15 to AD8	107 to 114	I/O	Upper 8-bit, 16-bit bus
	AD7 to AD0	116, 117, 119 to 124	I/O	Lower 16-bit bus

Type	Symbol	Pin No.	I/O	Name and Function
Bus control	$\overline{\text{WAIT}}$	22	Input	Requests insertion of a wait state in the bus cycle when accessing an external 3-state address space.
	$\overline{\text{RD}}$	18	Output	This pin is low when the external address space is being read.
	$\overline{\text{HWR}}$	19	Output	This pin is low when the external address space is to be written to, and the upper half of the data bus is enabled.
	$\overline{\text{LWR}}$	14	Output	This pin is low when the external address space is to be written to, and the lower half of the data bus is enabled.
	$\overline{\text{AS}}$	20	Output	This pin is low when address output on the address bus is valid.
	$\overline{\text{CS3}}$ to $\overline{\text{CS1}}$	91, 15, 17	Output	Chip select signals for areas 3 to 1.
	$\overline{\text{AH}}$	20	Output	Address latch signal for address/data multiplex bus.
Interrupts	NMI	27	Input	Nonmaskable interrupt request input pin
	$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$	76 to 73 52 to 49	Input	These pins request a maskable interrupt. Selectable to which pin of IRQn or ExIRQn to input IRQ7 to IRQ0 interrupts.
	$\overline{\text{ExIRQ7}}$ to $\overline{\text{ExIRQ0}}$	70 to 67 1 to 4		
On-chip emulator	$\overline{\text{ETRST}}$*[2]	39	Input	Interface pins for the on-chip emulator. Reset by holding the $\overline{\text{ETRST}}$ pin to low when activating the H-UDI. At this time, the $\overline{\text{ETRST}}$ pin should be held low for 20 clocks of ETCK. For details, see section 24, Electrical Characteristics. Then, to activate the H-UDI, the $\overline{\text{ETRST}}$ pin should be set to 1 and desired values should be set to the ETCK, ETMS, and ETDI pins. When in the normal operation without activating the H-UDI, the $\overline{\text{ETRST}}$, ETCK, ETMS, and ETDI pins should be set to 1 or high-impedance. Since these pins are internally pulled up, care should be taken in the standby state.
	ETMS	35	Input	
	ETDO	40	Output	
	ETDI	38	Input	
	ETCK	36	Input	
8-bit PWM timer (PWM)	PW7 to PW0	116, 117, 119, 124	Output	Pulse output pins for the PWM timer. Selectable from which pin of PWn or ExPWn to output PW5 to PW0.
	ExPW7 to ExPW0	97, 98, 48, 47, 76 to 73		

Type	Symbol	Pin No.	I/O	Name and Function
14-bit PWM timer (PWMX)	PWX0 PWX1	90 89	Output	Pulse output pins for PWM D/A
16-bit free-running timer (FRT)	FTCI_0 FTCI_1	92 8	Input	External event input pins
	FTOA_0 FTOA_1 FTOB_0 FTOB_1	77 12 95 11	Output	Output compare output pins
	FTIA_0 to FTID_0 FTIA_1 to FTID_1	84, 93, 94, 82 80, 49, 50, 79	Input	Input capture input pins
8-bit timer (TMR0, TMR1, TMRX, TMRY)	TMO0_0 TMO0_1 TMO1_0 TMO1_1 TMOX_0 TMOX_1 TMOY_0 TMOY_1	96 47 78 48 98 10 97 9	Output	Waveform output pins with output compare function
	TMI0_0 TMI0_1 TMI1_0 TMI1_1 TMIX_0 TMIX_1 TMIY_0 TMIY_1	74 51 85 81 75 52 76 73	Input	External event input pins and counter reset input pins

Type	Symbol	Pin No.	I/O	Name and Function
16-bit timer pulse unit (TPU)	TCLKA to TCLKD ExTCLKA to ExTCLKD	107, 109, 111, 112 96, 22, 14, 17	Input	External clock input pins. Selectable to which pin of TCLKn or ExTCLKn to input external clocks.
	TIOCA0 TIOCB0 TIOCC0 TIOCD0 ExTIOCA0 ExTIOCB0 ExTIOCC0 ExTIOCD0	114 113 112 111 88 87 96 22	I/O	Input capture input/output compare output/PWM output pins for TGR0A to TGR0D. Selectable to/from which pin of TIOCn0 or ExTIOCn0 to input/output input capture, output compare, and PWM.
	TIOCA1 TIOCB1 ExTIOCA1 ExTIOCB1	110 109 91 14	I/O	Input capture input/output compare output/PWM output pins for TGR1A to TGR1D. Selectable to/from which pin of TIOCn1 or ExTIOCn1 to input/output input capture, output compare, and PWM.
	TIOCA2 TIOCB2 ExTIOCA2 ExTIOCB2	108 107 15 17	I/O	Input capture input/output compare output/PWM output pins for TGR2A to TGR2D. Selectable to/from which pin of TIOCn2 or ExTIOCn2 to input/output input capture, output compare, and PWM.
Timer connection	VSYNCI_0 VSYNCI_1 HSYNCI_0 HSYNCI_1 CSYNCI_0 CSYNCI_1 HFBACKI VFBACKI	84 80 85 81 82 79 92 93	Input	Synchronization signal input pins for the timer connection
	VSYNCO HSYNCO CLAMPO CBLANK	77 78 94 95	Output	Synchronization signal output pins for the timer connection
Serial communi-cation interface (SCI)	TxD0 to TxD4	42, 45, 6, 102, 100	Output	Transmit data output pins
	RxD0 to RxD4	43, 46, 5, 101, 99	Input	Receive data input pins
	SCK0 to SCK4	41, 44, 7, 79, 97	I/O	Clock input/output pins.

Type	Symbol	Pin No.	I/O	Name and Function
I^2C bus interface 3 (IIC3)	SCL0, SCL1 SCL2, SCL3	102, 100 103, 105	I/O	I^2C clock input/output pins. These pins can drive a bus directly with the NMOS open drain output.
	SDA0, SDA1 SDA2, SDA3	101, 99 104, 106	I/O	I^2C data input/output pins. These pins can drive a bus directly with the NMOS open drain output.
A/D converter	AN15 to AN0	70 to 55	Input	Analog input pins
	$\overline{\text{ADTRG}}$	87	Input	External trigger input pin to start A/D conversion
	AVCC	71	Input	Analog power supply pin for the A/D converter. When the A/D converter is not used, this pin should be connected to the system power supply (+3.3 V).
	AVref	72	Input	Reference power supply pin for the A/D converter. When the A/D converter is not used, this pin should be connected to the system power supply (+3.3 V).
	AVSS	54	Input	Ground pin for the A/D converter. This pin should be connected to the system power supply (0 V).
I/O ports	P07 to P00	70 to 63	Input	Eight input pins
	P17 to P10	116, 117, 119 to 124	I/O	Eight input/output pins
	P27 to P20	107 to 114	I/O	Eight input/output pins
	P37 to P30	125 to 4	I/O	Eight input/output pins
	P47 to P40	76 to 73, 52 to 49	I/O	Eight input/output pins
	P57 to P50	48 to 41	I/O	Eight input/output pins
	P67 to P60	5 to 12	I/O	Eight input/output pins
	P77 to P70	62 to 55	Input	Eight input pins
	P87 to P80	87 to 90 99 to 102	I/O	Eight input/output pins
	P97 to P90	22 to 14	I/O	Eight input/output pins
	PA7 to PA0	91 to 98	I/O	Eight input/output pins
	PB7 to PB0	85, 84, 82 to 77	I/O	Eight input/output pins
	PC7 to PC4*[1]	40, 38, 36, 35	Input	Four input pins
	PC3 to PC0	106 to 103	I/O	Four input/output pins

Notes: 1. Not supported by the on-chip emulator.

2. Following precautions are required on the power-on reset signal that is applied to the $\overline{\text{ETRST}}$ pin.
   The reset signal must be applied at a power-on.
   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.
   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.

Figure1.3 shows an example of design in which signals for reset do not affect each other.

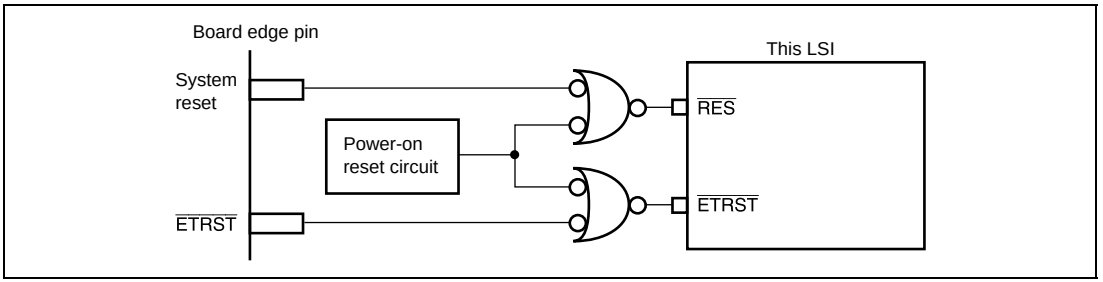


Figure 1.3 Sample Design of Reset Signals without Affection Each Other

Section 2 CPU

The H8S/2600 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/2600 CPU has sixteen 16-bit general registers, can address a 16-Mbyte linear address space, and is ideal for realtime control. This section describes the H8S/2600 CPU. The usable modes and address spaces differ depending on the product. For details on each product, refer to section 3, MCU Operating Modes.

2.1 Features

- Upward-compatible with H8/300 and H8/300H CPUs
  Can execute H8/300 and H8/300H object programs
- General-register architecture
  Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-bit registers
- Sixty-nine basic instructions
  8/16/32-bit arithmetic and logic instructions
  Multiply and divide instructions
  Powerful bit-manipulation instructions
  Multiply-accumulate instruction
- 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-Mbyte address space
  Program: 16 Mbytes
  Data: 16 Mbytes
- High-speed operation
  All frequently-used instructions execute in one or two states
  8/16/32-bit register-register add/subtract: 1 state
  8 × 8-bit register-register multiply: 3 states
  16 ÷ 8-bit register-register divide: 12 states
  16 × 16-bit register-register multiply: 4 states
  32 ÷ 16-bit register-register divide: 20 states

- Two CPU operating modes
  Normal mode*
  Advanced mode
- Power-down state
  Transition to power-down state by SLEEP instruction
  CPU clock speed selection

Note: Normal mode is not available in this LSI.

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/2600 CPU has the following enhancements.

- More general registers and control registers
  Eight 16-bit expanded registers, and one 8-bit and two 32-bit control registers, have been added.
- Expanded address space
  Normal mode supports the same 64-kbyte address space as the H8/300 CPU.
  Advanced mode supports a maximum 16-Mbyte address space.
- Enhanced addressing
  The addressing modes have been enhanced to make effective use of the 16-Mbyte address space.
- Enhanced instructions
  Addressing modes of bit-manipulation instructions have been enhanced.
  Signed multiply and divide instructions have been added.
  A multiply-accumulate instruction has been added.
  Two-bit shift and 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 execute twice as fast.

Note: Normal mode is not available in this LSI.

2.1.3 Differences from H8/300H CPU

In comparison to the H8/300H CPU, the H8S/2600 CPU has the following enhancements.

- Additional control register
  One 8-bit and two 32-bit control registers have been added.
- Enhanced instructions
  Addressing modes of bit-manipulation instructions have been enhanced.
  A multiply-accumulate instruction has been added.
  Two-bit shift and 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 execute twice as fast.

2.2 CPU Operating Modes

The H8S/2600 CPU has two operating modes: normal and advanced. Normal mode supports a maximum 64-kbyte address space. Advanced mode supports a maximum 16-Mbyte total address space. The mode is selected by the mode pins.

2.2.1 Normal Mode

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

- Address Space

  The H8S/2600 CPU provides linear access to a maximum 64-kbyte address space.
- 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 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 the general register is referenced in the register indirect addressing mode with pre-decrement (@–Rn) or post-increment (@Rn+) and a carry or borrow occurs, however, 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-handling Vector Table and Memory Indirect Branch Addresses

  In normal mode the top area starting at H'0000 is allocated to the exception-handling vector table. One branch address is stored per 16 bits. The exception-handling vector table in normal mode is shown in figure 2.1. For details of the exception-handling 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-handling vector table.
- Stack Structure

  When the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.2. EXR is not pushed onto the stack in interrupt control mode 0. For details, see section 4, Exception Handling.

Note: Normal mode is not available in this LSI.

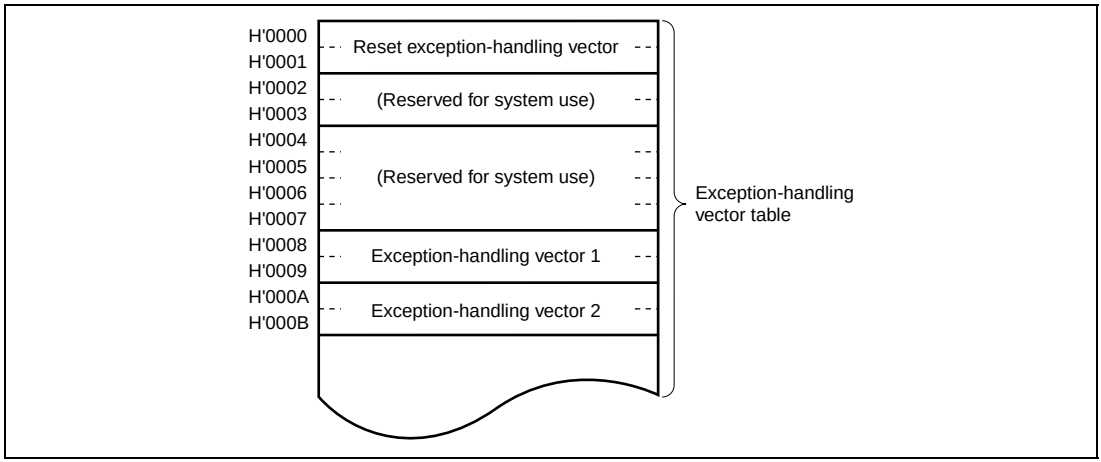
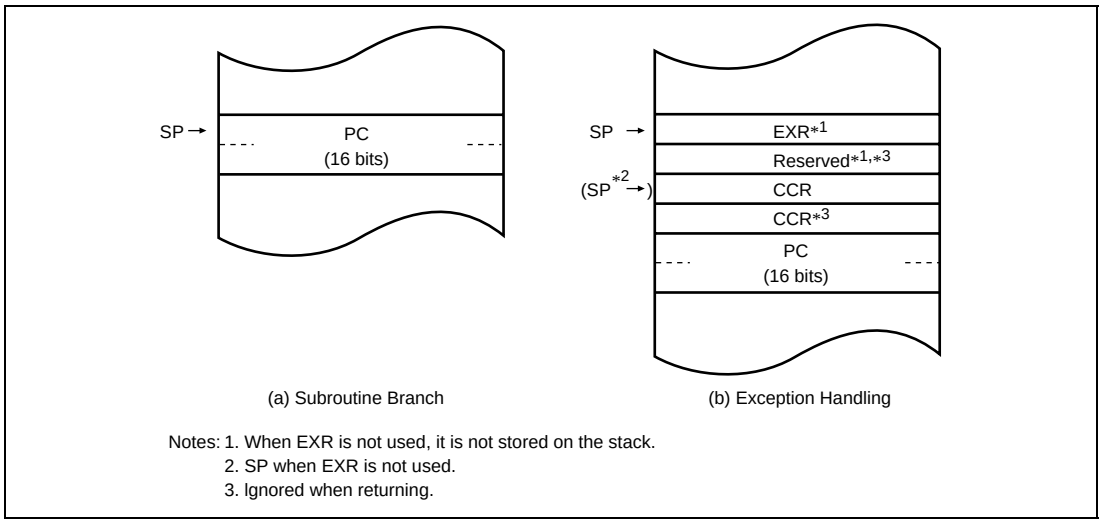


Figure 2.1 Exception-Handling Vector Table (Normal Mode)



Notes: 1. When EXR is not used, it is not stored on the stack.
2. SP when EXR is not used.
3. Ignored when returning.

Figure 2.2 Stack Structure in Normal Mode

2.2.2 Advanced Mode

- Address Space

  Linear access is provided to a 16-Mbyte maximum address space.
- 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 or address registers.
- Instruction Set

  All instructions and addressing modes can be used.
- Exception-handling Vector Table and Memory Indirect Branch Addresses

  In advanced mode the top area starting at H'00000000 is allocated to the exception-handling vector table in units of 32 bits. In each 32 bits, the upper 8 bits are ignored and a branch address is stored in the lower 24 bits (figure 2.3). For details of the exception-handling vector table, see section 4, Exception Handling.

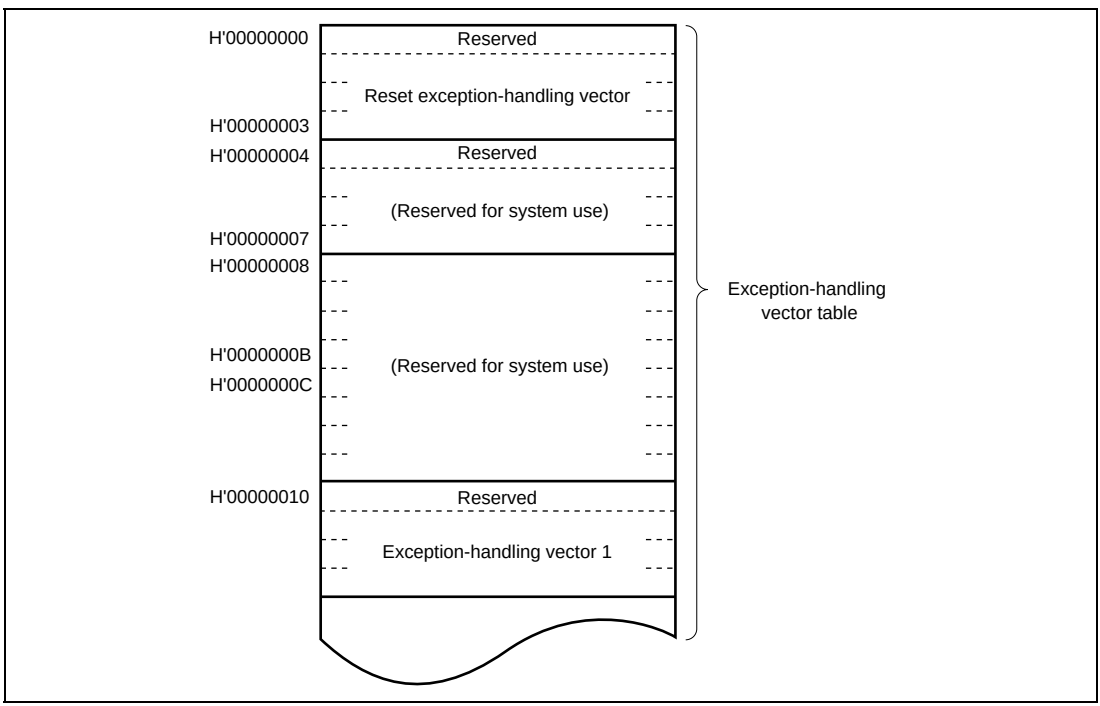


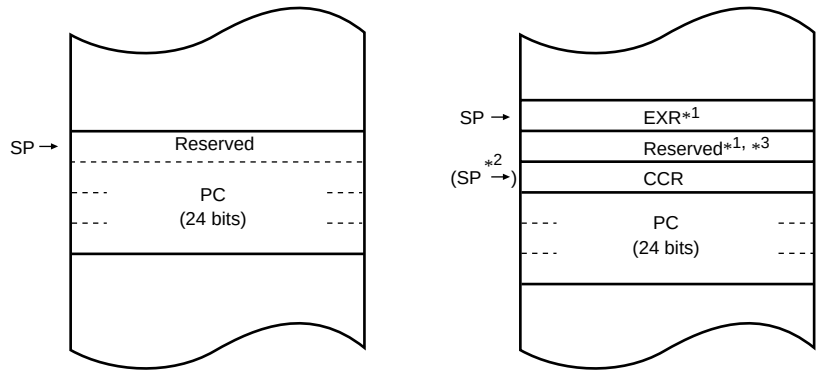
Figure 2.3 Exception-Handling Vector Table (Advanced Mode)

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 8 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 first part of this range is also used for the exception-handling vector table.

- Stack Structure

  In advanced mode, when the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.4. EXR is not pushed onto the stack in interrupt control mode 0. For details, see section 4, Exception Handling.



Notes: 1. When EXR is not used, it is not stored on the stack.
2. SP when EXR is not used.
3. Ignored when returning.

Figure 2.4 Stack Structure in Advanced Mode

2.3 Address Space

Figure 2.5 shows a memory map of the H8S/2600 CPU. The H8S/2600 CPU provides linear access to a maximum 64-kbyte address space in normal mode, and a maximum 16-Mbyte (architecturally 4-Gbyte) address space in advanced mode. The usable modes and address spaces differ depending on the product. For details on each product, refer to section 3, MCU Operating Modes.

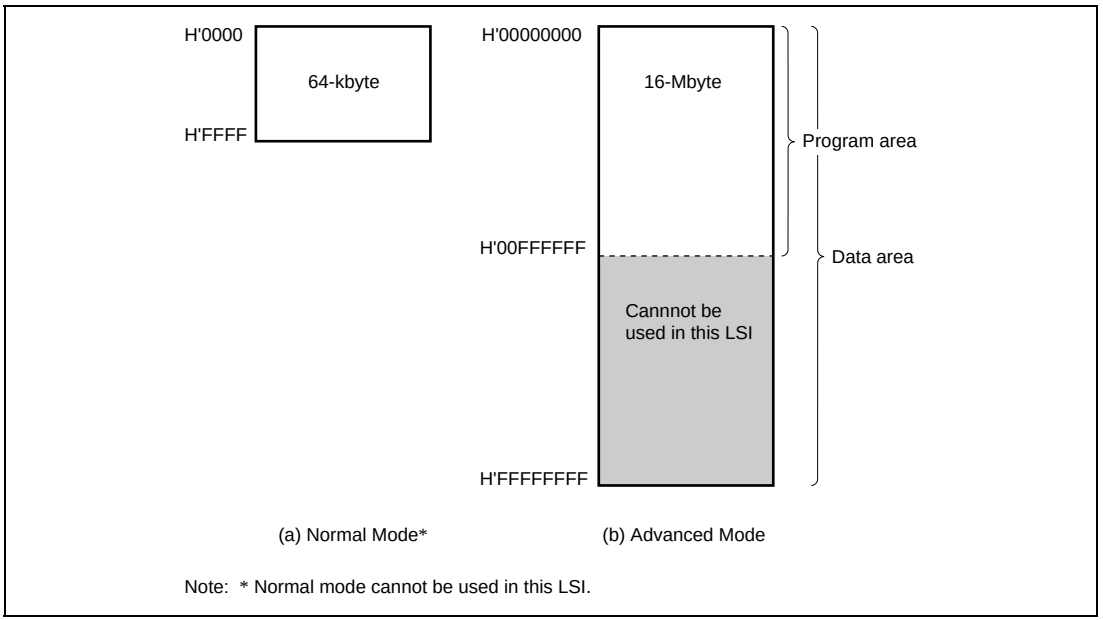


Figure 2.5 Memory Map

Note: Normal mode is not available in this LSI.

2.4 Registers

The H8S/2600 CPU has the internal registers shown in figure 2.6. There are two types of registers: general registers and control registers. Control registers are a 24-bit program counter (PC), an 8-bit extended register (EXR), an 8-bit condition code register (CCR), and a 64-bit multiply-accumulate register (MAC).

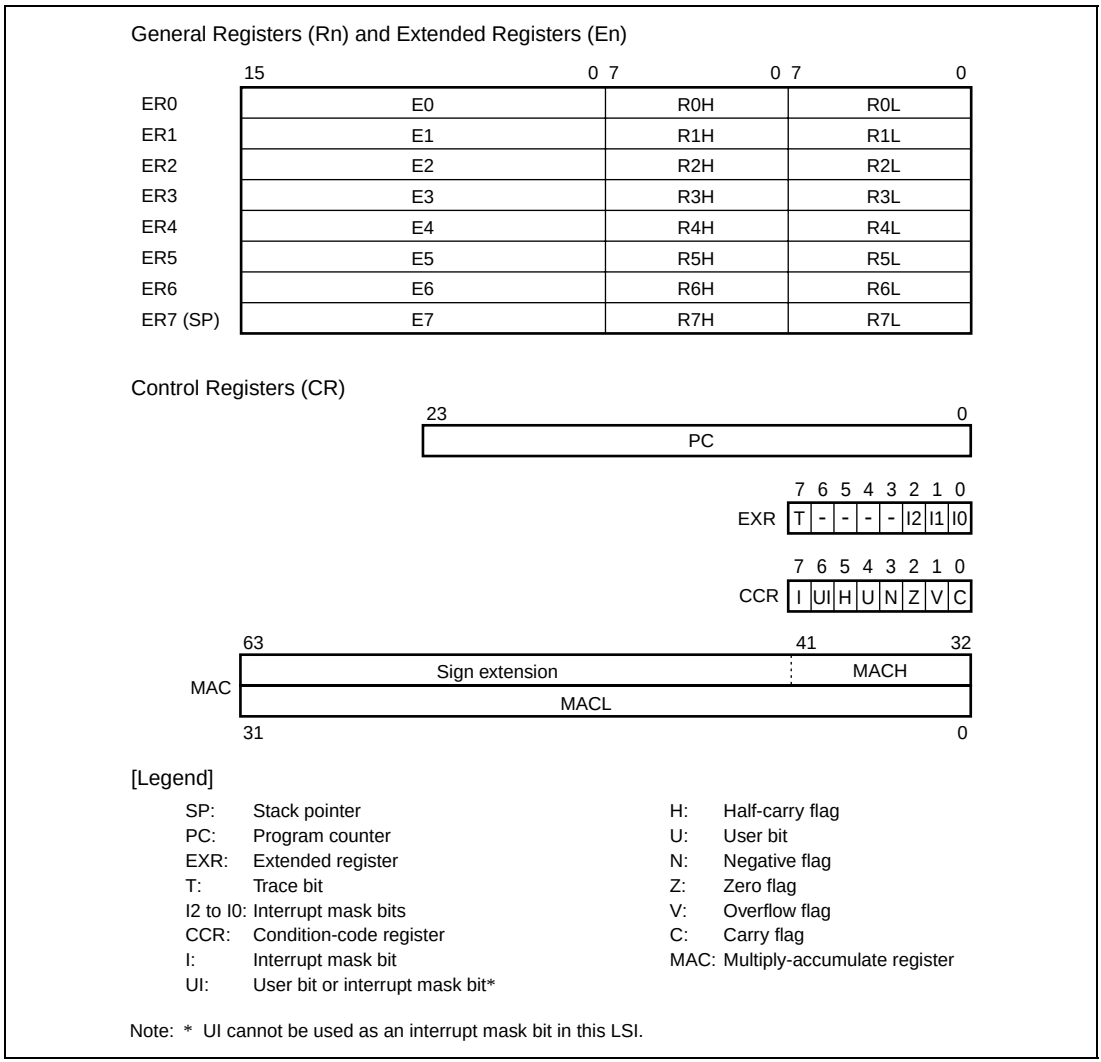


Figure 2.6 CPU Registers

2.4.1 General Registers

The H8S/2600 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).

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

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

The usage of each register can be selected independently.

General register ER7 has the function of stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2.8 shows the stack.

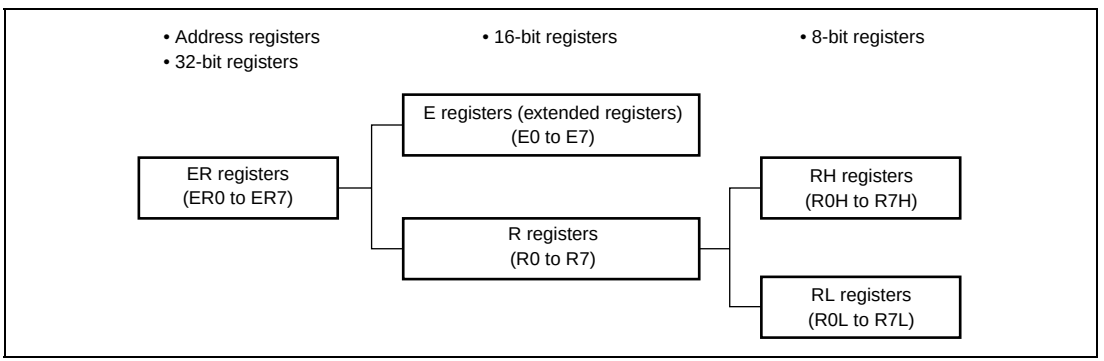


Figure 2.7 Usage of General Registers

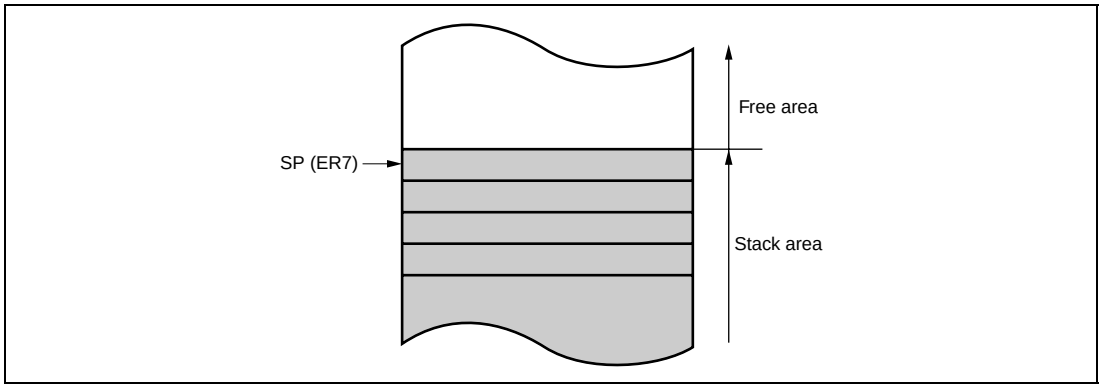


Figure 2.8 Stack

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

2.4.3 Extended Register (EXR)

EXR is an 8-bit register that can be manipulated by the LDC, STC, ANDC, ORC, and XORC instructions. When these instructions except for the STC instruction is executed, all interrupts including NMI will be masked for three states after execution is completed.

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit When this bit is set to 1, a trace exception-handling is started each time an instruction is executed. When this bit is cleared to 0, instructions are executed in sequence.
6 to 3	—	All 1	—	Reserved These bits are always read as 1.
2	I2	1	R/W	These bits designate the interrupt mask level (0 to 7). For details, refer to section 5, Interrupt Controller.
1	I1	1	R/W	
0	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 by hardware at the start of an exception-handling sequence. For details, refer to section 5, Interrupt Controller.
6	UI	Undefined	R/W	User Bit or Interrupt Mask Bit Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. This bit cannot be used as an interrupt mask bit in this LSI.
5	H	Undefined	R/W	Half-Carry Flag When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.
4	U	Undefined	R/W	User Bit Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit of data as a sign bit.
2	Z	Undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.

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 Multiply-Accumulate Register (MAC)

This 64-bit register stores the results of multiply-accumulate operations. It consists of two 32-bit registers denoted MACH and MACL. The lower 10 bits of MACH are valid; the upper bits are a sign extension.

2.4.6 Initial Values of CPU Internal Registers

When the reset exception handling loads the start address from the vector address, PC is initialized, the T bit in EXR is cleared to 0, and the I bits in EXR and CCR are set to 1. However, the general registers and the other CCR bits are not initialized. The initial value of SP (ER7) is undefined. SP should therefore be initialized by using the MOV.L instruction immediately after a reset.

2.5 Data Formats

The H8S/2600 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 in general registers.

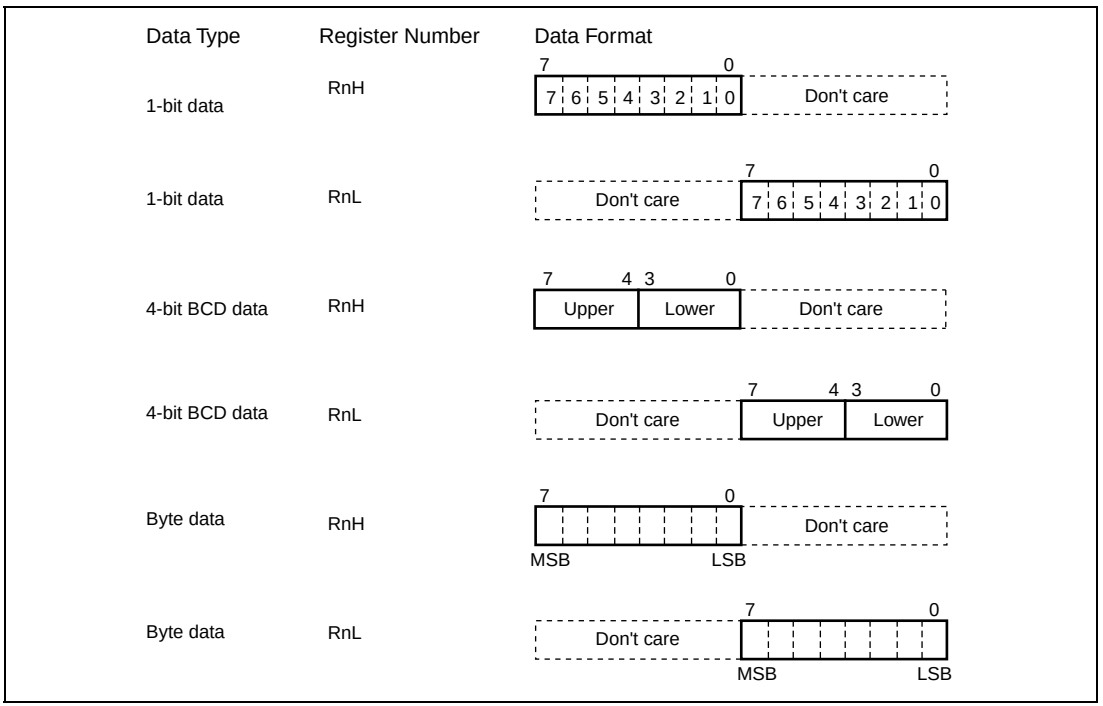
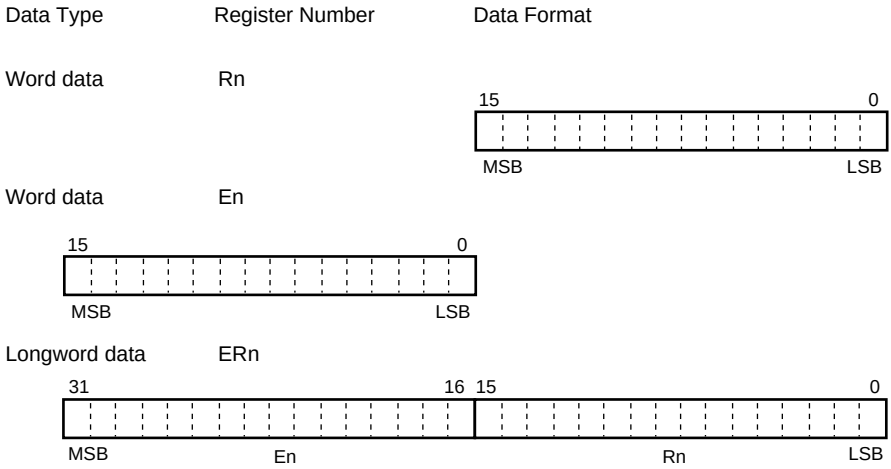


Figure 2.9 General Register Data Formats (1)



[Legend]

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

Figure 2.9 General Register Data Formats (2)

2.5.2 Memory Data Formats

Figure 2.10 shows the data formats in memory. The H8S/2600 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 ER7 is used as an address register to access the stack, the operand size should be word size or longword size.

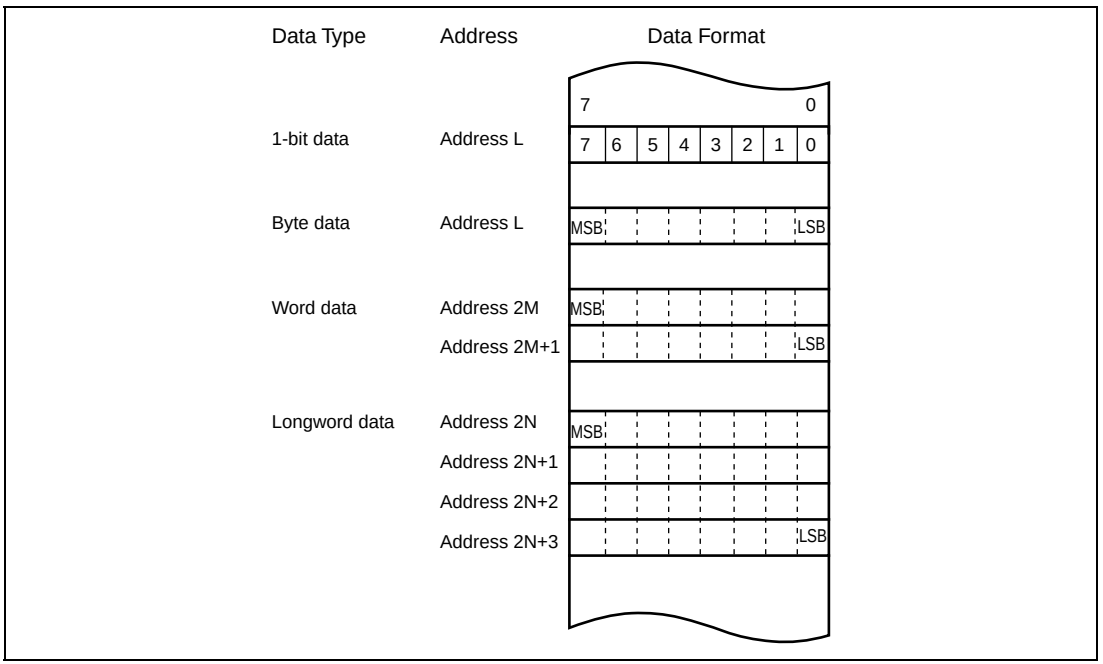


Figure 2.10 Memory Data Formats

2.6 Instruction Set

The H8S/2600 CPU has 69 types of instructions. The instructions are classified by function 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, STM	L	
	MOVFPE*[3], MOVTPE*[3]	B	
Arithmetic operations	ADD, SUB, CMP, NEG	B/W/L	23
	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	
	MAC, LDMAC, STMAC, CLRMAC	—	
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	Bcc*[2], JMP, BSR, JSR, RTS	—	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—	9
Block data transfer	EEPMOV	—	1
		Total:	69

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 respectively. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn, @-SP respectively.
2. Bcc is the general name for conditional branch instructions.
3. Cannot be used in this LSI.
4. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

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)
MAC	Multiply-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended 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*	Function
MOV	B/W/L	(EAs) → Rd, Rs → (EAd) Transfers data between two general registers or between a general register and memory, or transfers 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	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
STM	L	Rn (register list) → @–SP Pushes two or more general registers onto the stack.

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

Table 2.4 Arithmetic Operations Instructions (1)

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

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

Table 2.4 Arithmetic Operations Instructions (2)

Instruction	Size*[1]	Function
DIVXS	B/W	Rd ÷ Rs → Rd Performs signed division on data in two general registers. Either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.
CMP	B/W/L	Rd – Rs, Rd – #IMM Compares data in a general register with data in another general register or with immediate data, and sets CCR bits according to the result.
NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic complement) of data in a general register.
EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by padding with zeros on the left.
EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by extending the sign bit.
TAS*[2]	B	@ERd – 0, 1 → (<bit 7> of @ERd) Tests memory contents, and sets the most significant bit (bit 7) to 1.
MAC	—	(EAs) × (EAd) + MAC → MAC Performs signed multiplication on memory contents and adds the result to the multiply-accumulate register. The following operations can be performed: 16 bits × 16 bits + 32 bits → 32 bits, saturating 16 bits × 16 bits + 42 bits → 42 bits, non-saturating
CLRMAC	—	0 → MAC Clears the multiply-accumulate register to zero.
LDMAC STMAC	L	Rs → MAC, MAC → Rd Transfers data between a general register and a multiply-accumulate register.

Notes: 1. Size refers to the operand size.
B: Byte
W: Word
L: Longword
2. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

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 general register contents.

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

Table 2.6 Shift Instructions

Instruction	Size*	Function
SHAL SHAR	B/W/L	Rd (shift) → Rd Performs an arithmetic shift on general register contents. 1-bit or 2-bit shift is possible.
SHLL SHLR	B/W/L	Rd (shift) → Rd Performs a logical shift on general register contents. 1-bit or 2-bit shift is possible.
ROTL ROTR	B/W/L	Rd (rotate) → Rd Rotates general register contents. 1-bit or 2-bit rotation is possible.
ROTXL ROTXR	B/W/L	Rd (rotate) → Rd Rotates general register contents through the carry flag. 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 ANDs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIAND	B	C ∧ [~ (<bit-No.> of <EAd>)] → C ANDs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BOR	B	C ∨ (<bit-No.> of <EAd>) → C ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIOR	B	C ∨ [~ (<bit-No.> of <EAd>)] → C ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

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

Table 2.7 Bit Manipulation Instructions (2)

Instruction	Size*[1]	Function
BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C 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 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 Transfers the contents of a general register or memory, 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 8 bits are valid.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a general register or memory. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 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/2600 Series 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 bits, data registers by 3 bits or 4 bits. Some instructions have two register fields. Some have no register field.
- Effective Address Extension

  8, 16, or 32 bits specifying immediate data, an absolute address, or a displacement.
- Condition Field

  Specifies the branching condition of Bcc instructions.

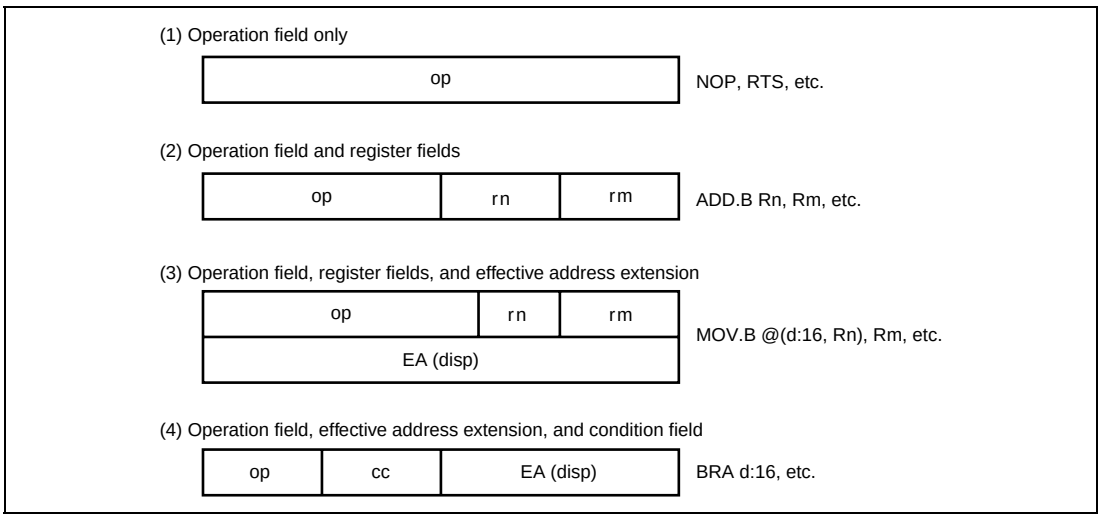


Figure 2.11 Instruction Formats (Examples)

2.7 Addressing Modes and Effective Address Calculation

The H8S/2600 CPU supports the eight addressing modes listed in table 2.11. The usable address modes are different in each instruction.

Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions use register direct, register indirect, or 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 Register indirect with pre-decrement	@ERn+ @–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 containing the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

2.7.2 Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn) which contains the address of the operand on memory. If the address is a program instruction address, the lower 24 bits are valid and the upper 8 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 is added to an address register (ERn) specified by the register field of the instruction code, 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 transfer instruction, or 4 for longword transfer instruction. For word or longword transfer instruction, 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 transfer instruction, or 4 for longword transfer instruction. For word or longword transfer instruction, 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. A 32-bit absolute address can access the entire address space.

A 24-bit absolute address (@aa:24) indicates the address of a program instruction. The upper 8 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)		

Note: * Not available in this LSI.

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

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

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some bit manipulation instructions contain 3-bit immediate data in the instruction code, specifying a bit number. 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 is used in the Bcc and BSR instructions. An 8-bit or 16-bit displacement contained in the instruction code is sign-extended and added to the 24-bit PC contents to generate a branch address. Only the lower 24 bits of this branch address are valid; the upper 8 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 bytes (–63 to +64 words) or –32766 to +32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting value should be an even number.

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. This memory operand contains a branch address. The upper bits of the absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF 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 all 0 (H'00). Note that the first part of the address range is also the exception-handling vector area. For further details, refer to 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 instruction code to be fetched at the address preceding the specified address. (For further information, see section 2.5.2, Memory Data Formats.)

Note: Normal mode is not available in this LSI.

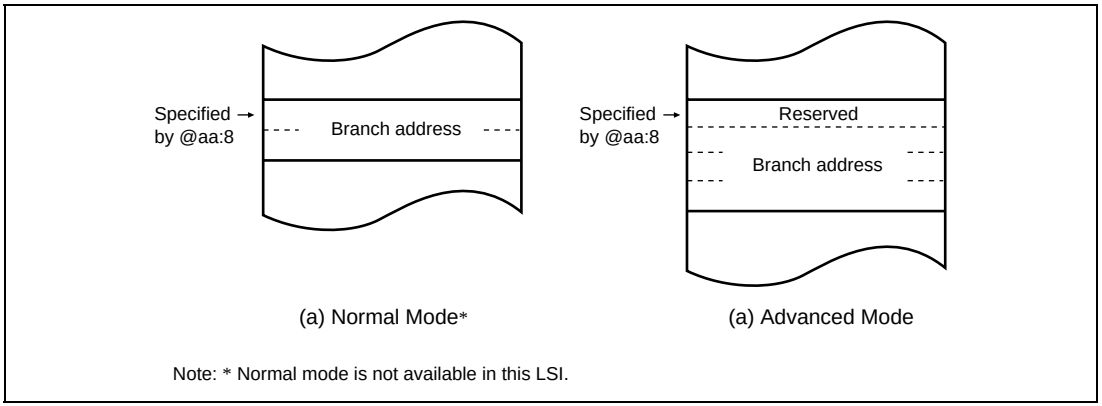


Figure 2.12 Branch Address Specification in Memory Indirect Mode

2.7.9 Effective Address Calculation

Table 2.13 indicates how effective addresses (EA) are calculated in each addressing mode. In normal mode the upper 8 bits of the effective address are ignored in order to generate a 16-bit address.

Note: Normal mode is not available in this LSI.

Table 2.13 Effective Address Calculation (1)

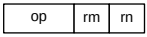
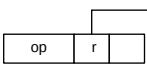
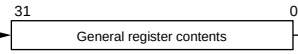
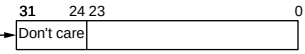
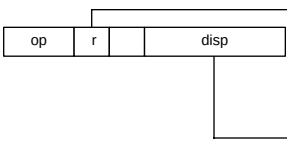
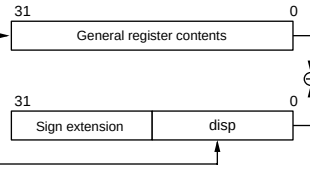
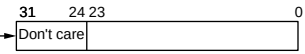
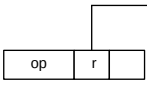
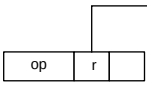
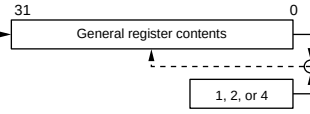
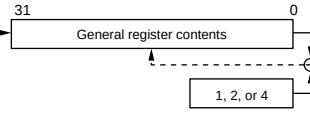
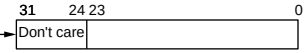
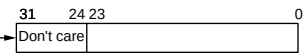
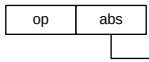
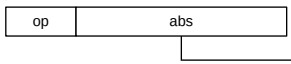
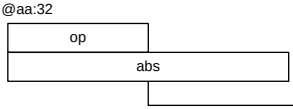
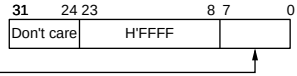
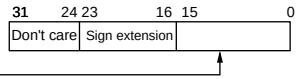
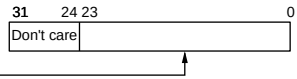
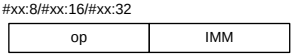
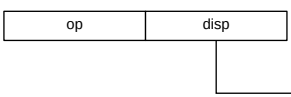
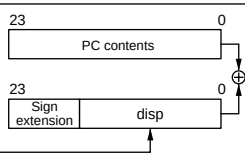
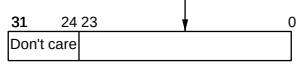
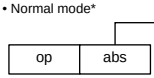
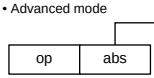
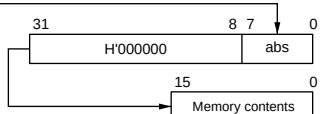
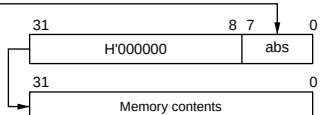
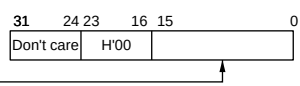
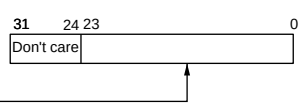
No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct (Rn) 		Operand is general register contents.
2	Register indirect (@ERn) 		
3	Register indirect with displacement @(d:16,ERn) or @(d:32,ERn) 		
4	Register indirect with post-increment or pre-decrement •Register indirect with post-increment @ERn+  •Register indirect with pre-decrement @-ERn 	  Operand Size / Offset Byte: 1 Word: 2 Longword: 4	 

Table 2.13 Effective Address Calculation (2)

No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
5	Absolute address @aa:8  @aa:16  @aa:24  @aa:32 		   
6	Immediate #xx:8/#xx:16/#xx:32 		Operand is immediate data.
7	Program-counter relative @(d:8,PC)/@(d:16,PC) 		
8	Memory indirect @@aa:8 • Normal mode*  • Advanced mode 	 	 

Note: * Normal mode is not available in this LSI.

2.8 Processing States

The H8S/2600 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

  The CPU and on-chip peripheral modules are all initialized and stop. 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, refer to 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-handling vector table and branches to that address. For further details, refer to section 4, Exception Handling.

- Program Execution State

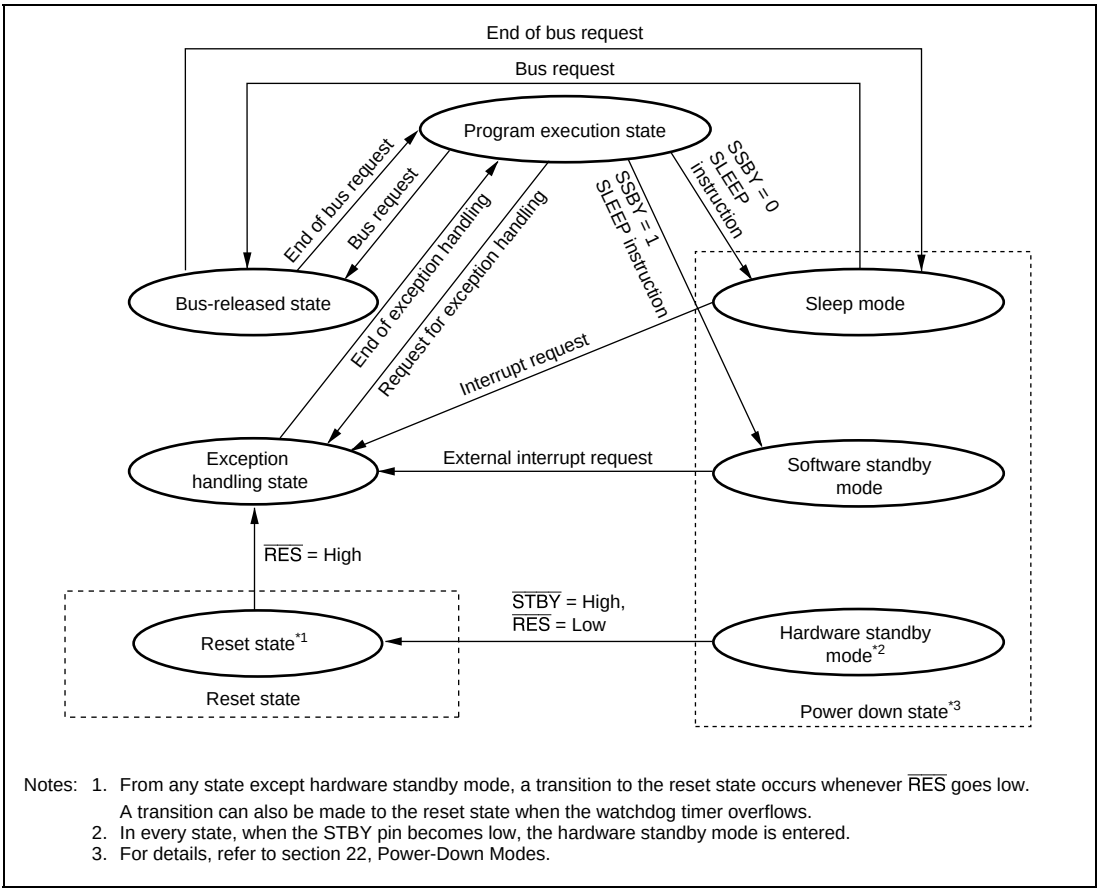
  In this state the CPU executes program instructions in sequence.

- Bus-Released State

  In a product which has a bus mastership other than the CPU, such as a direct memory access controller (DMAC) and a data transfer controller (DTC), the bus-released state occurs when the bus has been released in response to a bus request from a bus mastership other than the CPU. While the bus is released, the CPU halts operations.

- 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 further details, refer to 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. In every state, when the STBY pin becomes low, the hardware standby mode is entered.
3. For details, refer to section 22, Power-Down Modes.

Figure 2.13 State Transitions

2.9 Usage Note

2.9.1 Usage Notes on Bit-Wise Operation Instructions

The BSET, BCLR, BNOT, BST, and BIST instructions are used to read data in bytes, operate the data in bit units, and write the result of the bit unit operation in bits again. Therefore, special care is necessary to use these instructions for the registers and the ports that include write-only bit.

The BCLR instruction can be used to clear the flags in the internal I/O registers to 0. In this time, if it is obvious that the flag has been set to 1 in the interrupt handler, there is no need to read the flag beforehand.

Section 3 MCU Operating Modes

3.1 Operating Mode Selection

This LSI has four operating modes (modes 1, 3, 5, and 7). These modes are determined by the mode pin settings (MD2, MD1, and MD0). For normal program execution mode, the mode pins must be set to mode 7. Do not change the mode pins while in the middle of an operation. 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
1	0	0	1	Boot mode	Flash memory programming/erasing
3	0	1	1	Emulation	On-chip emulation mode
5	1	0	1	User boot mode	Flash memory programming/erasing
7	1	1	1	Advanced	Single-chip mode with on-chip ROM enable extended mode

Modes 0, 2, 4, and 6 are not available with this LSI.

After a reset in mode 7, the operation is started in single-chip mode. It is possible to shift to extended mode when the EXPE bit in MDCR is set to 1.

Modes 1 and 5 are boot modes for flash memory programming/erasing. For details, refer to section 20, Flash Memory (0.18-µm F-ZTAT Version).

Mode 3 is on-chip emulation mode. The JTAG interface is controlled by the on-chip emulator, on-chip emulation is possible.

3.2 Register Descriptions

The following registers are related to the operating mode.

- Mode control register (MDCR)
- System control register (SYSCR)

3.2.1 Mode Control Register (MDCR)

MDCR monitors the current operating mode and operating mode settings.

Bit	Bit Name	Initial Value	R/W	Descriptions
7	EXPE	0	R/W	Extended Mode Enable Extended Mode Set Up 0: Single-chip mode 1: Extended mode
6 to 3	—	All 0	R	Reserved
2	MDS2	—*	R	Mode Select 2 to 0 These bits indicate the input levels at pins MD2 to MD0 (the current operating mode). Bits MDS2 to MDS0 correspond to MD2 to MD0. MDS2 to MDS0 are read-only bits and they cannot be written to The mode pin (MD2 to MD0) input levels are latched into these bits when MDCR is read. These latches are canceled by a reset
1	MDS1	—*	R	
0	MDS0	—*	R	

Note: * Determined by pins MD2 to MD0.

3.2.2 System Control Register (SYSCR)

SYSCR selects saturating calculation for the MAC instruction, and controls reset source monitor, Ram address space, and on-chip flash memory control.

Bit	Bit Name	Initial Value	R/W	Descriptions
7	MACS	0	R/W	MAC Saturation Selects either saturating or non-saturating calculation for the MAC instruction. 0: Non-saturating calculation for MAC instruction 1: Saturating calculation for MAC instruction
6 to 4	—	All 0	R/W	Reserved The initial value should not be changed.
3	XRST	1	R	External Reset Indicates reset source. Reset occurs as external reset input or watchdog timer overflow. 0: Generated by watchdog timer overflow 1: Generated by external reset
2	FLASHE	0	R/W	Flash Memory Control Register Enable Controls CPU access to the flash memory control registers (FCCS, FPCS, FECS, FKEY, FMATS, and FTDAR). 0: Flash memory control registers are not selected 1: Flash memory control registers are selected
1	—	0	R/W	Reserved The initial value should not be changed.
0	RAME	1	R/W	RAM Enable Enables or disables the on-chip RAM. The RAME bit is initialized when the reset status is released. 0: On-chip RAM is disabled 1: On-chip RAM is enabled

3.3 Operating Mode Descriptions

3.3.1 Mode 7

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled. The initial mode after a reset is single-chip mode, to use the external address space, set the EXPE bit in MDCR to 1.

Normal Extended Mode:

After a reset, ports 1 and 2 become input ports.

The address bus can be output when the corresponding port data direction register (DDR) is set to 1. Port 3 is a data bus, part of port 9 and port A become a bus control signal. When the ABWn bit in BCRAn is cleared to 0, port 6 becomes the data bus. (n = 1 to 3)

Multiplex Extended Mode:

When using an 8-bit bus, regardless of the data direction register (DDR) setting of port 2, it becomes an address output and data input/output port. Port 1 can be used as a general port.

When using a 16-bit bus, regardless of the data direction register (DDR) setting of port 1 or 2, they become address output and data input/output ports.

3.3.2 Pin Functions

The pin functions of ports 1 to 3, 6, 9, and A change according to operating modes. Table 3.2 shows the pin functions in each operating mode.

Table 3.2 Pin Functions in Each Operating Mode

		Mode 7	
Port		**Normal Extended Mode**	**Multiplex Extended Mode**
Port 1		P*/A	P*/AD
Port 2		P*/A	P*/AD
Port 3		P*/D	P*
Port 6		P*/D	P*
Port 9		P*/C	P*/C
Port A	PA7	P*/C	P*/C

[Legend]

P: Input/output port

A: Address bus output

D: Data bus input/output

AD: Address data multiplex input/output

C: Control signals, clock input/output

Note: * After a reset

3.4 Memory Map

Figure 3.1 shows a memory map.

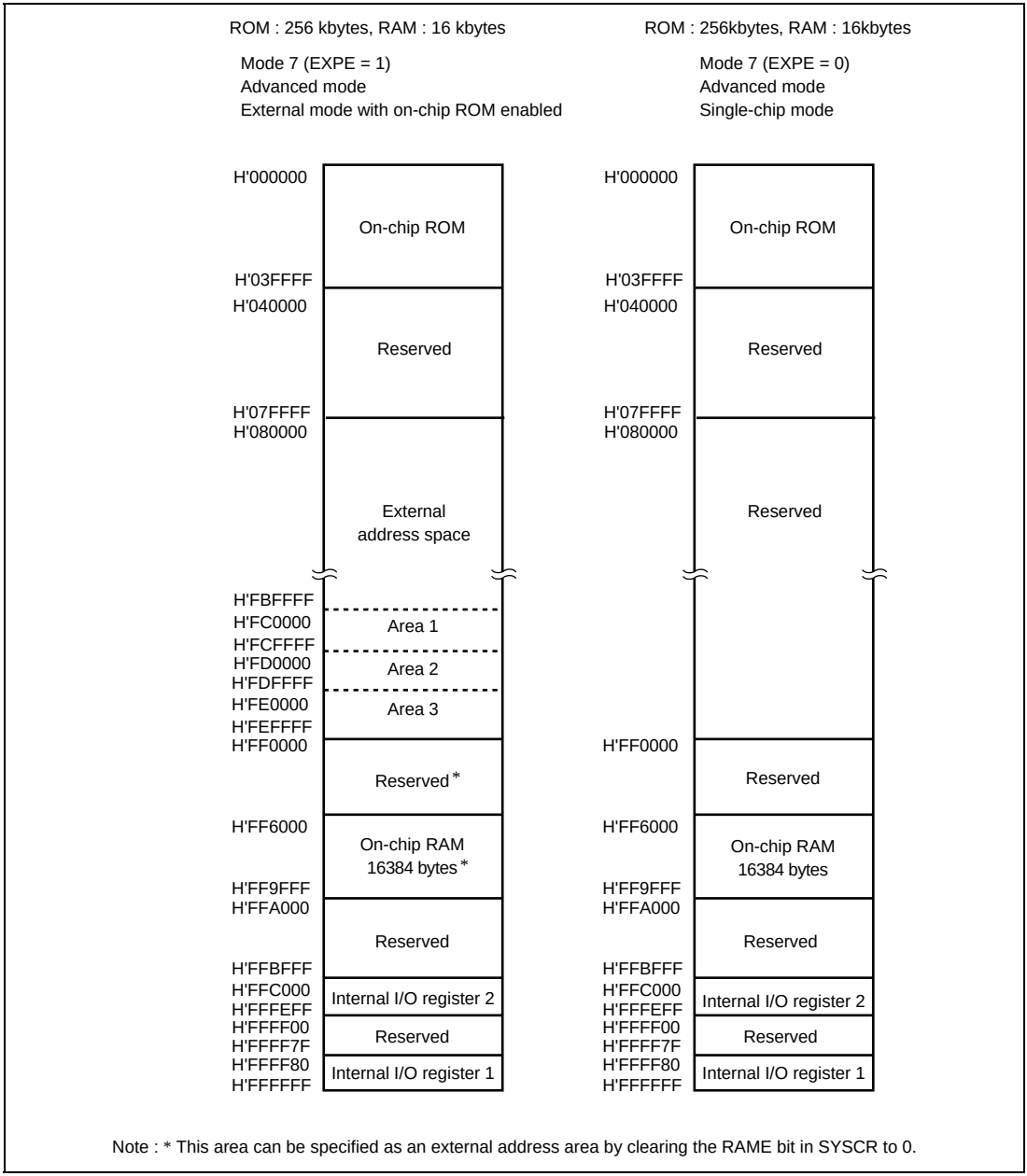


Figure 3.1 Memory Map

Section 4 Exception Handling

4.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, trace, interrupt, 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. Exception sources, the stack structure, and operation of the CPU vary depending on the interrupt control mode. For details on the interrupt control mode, refer to section 5, Interrupt Controller.

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High	Reset	Starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin, or when the watchdog timer overflows. The CPU enters the reset state when the $\overline{\text{RES}}$ pin is low
	Trace*[1]	Starts when execution of the current instruction or exception handling ends, if the trace (T) bit in the EXR is set to 1.
	Direct transition*[2]	Starts when the direct transition occurs by execution of the SLEEP instruction.
	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.
Low	Trap instruction	Started by execution of a trap instruction (TRAPA) Trap instruction exception handling requests are accepted at all times in program execution state.

Notes: 1. Traces are enabled only in interrupt control mode 2. Trace exception handling is not executed after execution of an RTE instruction.
2. Not available in this LSI.

4.2 Exception Sources and Exception Vector Table

Different vector addresses are assigned to different exception sources. Table 4.2 lists the exception sources and their vector addresses. Since the usable modes differ depending on the product, for details on each product, refer to section 3, MCU Operating Modes.

Table 4.2 Exception HandlingVector Table

			Vector Address*[1]	
Exception Source		**Vector Number**	**Normal Mode*[2]**	**Advanced Mode**
Power-on reset		0	H'0000 to H'0001	H'0000 to H'0003
Manual reset*[3]		1	H'0002 to H'0003	H'0004 to H'0007
Reserved for system use		2	H'0004 to H'0005	H'0008 to H'000B
		3	H'0006 to H'0007	H'000C to H'000F
		4	H'0008 to H'0009	H'0010 to H'0013
Trace		5	H'000A to H'000B	H'0014 to H'0017
Interrupt (direct transition)*[3]		6	H'000C to H'000D	H'0018 to H'001B
Interrupt (NMI)		7	H'000E to H'000F	H'001C to H'001F
Trap instruction (#0)		8	H'0010 to H'0011	H'0020 to H'0023
(#1)		9	H'0012 to H'0013	H'0024 to H'0027
(#2)		10	H'0014 to H'0015	H'0028 to H'002B
(#3)		11	H'0016 to H'0017	H'002C to H'002F
Reserved for system use		12	H'0018 to H'0019	H'0030 to H'0033
		13	H'001A to H'001B	H'0034 to H'0037
		14	H'001C to H'001D	H'0038 to H'003B
		15	H'001E to H'001F	H'003C to H'003F
External interrupt	IRQ0	16	H'0020 to H'0021	H'0040 to H'0043
	IRQ1	17	H'0022 to H'0023	H'0044 to H'0047
	IRQ2	18	H'0024 to H'0025	H'0048 to H'004B
	IRQ3	19	H'0026 to H'0027	H'004C to H'004F
	IRQ4	20	H'0028 to H'0029	H'0050 to H'0053
	IRQ5	21	H'002A to H'002B	H'0054 to H'0057
	IRQ6	22	H'002C to H'002D	H'0058 to H'005B
	IRQ7	23	H'002E to H'002F	H'005C to H'005F
Internal interrupt*[4]		24 ⋮ 127	H'0030 to H'0031 ⋮ H'00FE to H'00FF	H'0060 to H'0063 ⋮ H'01FC to H'01FF

Notes: 1. Lower 16 bits of the address.
2. Not available in this LSI.
3. Not available in this LSI. Becomes reserved for system use.
4. For details on internal interrupt vectors, 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. To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-up. 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 15, Watchdog Timer (WDT).

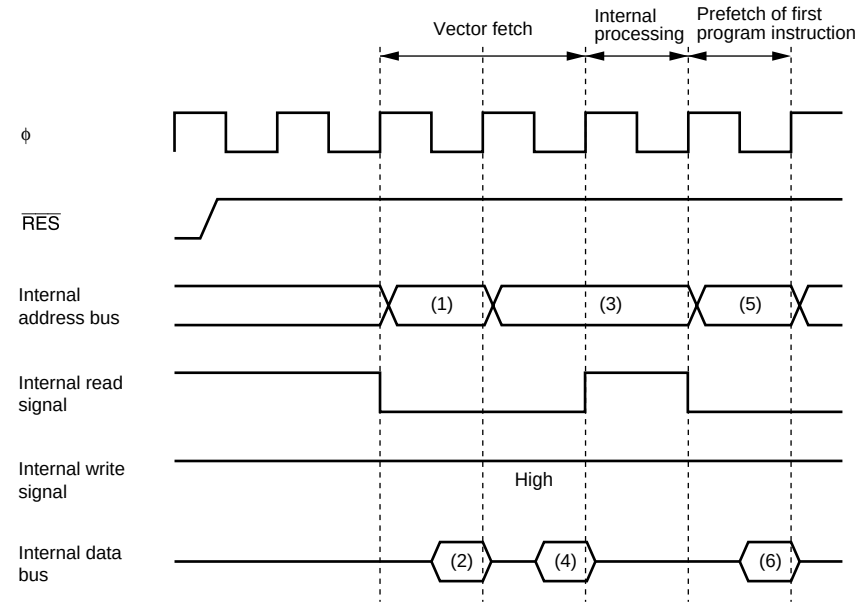
The interrupt control mode is 0 immediately after reset.

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, the T bit is cleared to 0 in EXR, and the I bit is set to 1 in EXR and CCR.
2. The reset exception-handling vector address is read and transferred to the PC, and program execution starts from the address indicated by the PC.

Figure 4.1 shows an example of the reset sequence.



(1)(3) Reset exception-handling vector address (when reset, (1)=H'000000, (3)=H'000002)
(2)(4) Start address (contents of reset exception-handling vector address)
(5) Start address ((5)=(2)(4))
(6) First program instruction

Figure 4.1 Reset Sequence

4.3.2 Interrupts after Reset

If an interrupt is accepted after a reset but 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 the reset state ends, make sure that this instruction initializes the stack pointer (example: MOV.L #xx: 32, SP).

4.3.3 On-Chip Peripheral Functions after Reset Release

After reset release, the module stop control register (MSTPCR, EXMSTPCR) is initialized and all modules enter module stop mode.

Consequently, on-chip peripheral module registers cannot be read or written to. Register reading and writing is enabled when module stop mode is exited.

4.4 Traces

Traces are enabled in interrupt control mode 2. Trace mode is not activated in interrupt control mode 0, irrespective of the state of the T bit. For details on interrupt control modes, see section 5, Interrupt Controller.

If the T bit in EXR is set to 1, trace mode is activated. In trace mode, a trace exception occurs on completion of each instruction. Trace mode is not affected by interrupt masking. Table 4.3 shows the state of CCR and EXR after execution of trace exception handling. Trace mode is canceled by clearing the T bit in EXR to 0. The T bit saved on the stack retains its value of 1, and when control is returned from the trace exception handling routine by the RTE instruction, trace mode resumes. Trace exception-handling is not carried out after execution of the RTE instruction.

Interrupts are accepted even within the trace exception-handling routine.

Table 4.3 Status of CCR and EXR after Trace Exception Handling

	CCR		EXR	
Interrupt Control Mode	**I**	**UI**	**I2 to I0**	**T**
0	Trace exception handling cannot be used.			
2	1	—	—	0

[Legend]
1: Set to 1
0: Cleared to 0
—: Retains value prior to execution.

4.5 Interrupts

Interrupts are controlled by the interrupt controller. The interrupt controller has two interrupt control modes and can assign interrupts other than NMI to eight priority/mask levels to enable multiplexed interrupt control. For details on the source that starts interrupt exception handling and the vector address, refer to section 5, Interrupt Controller.

The interrupt exception handling is as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended register (EXR) are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. 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.

4.6 Trap Instruction

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.

The trap instruction exception handling is as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended register (EXR) are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. 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 entry corresponding to a vector number from 0 to 3, as specified in the instruction code.

Table 4.4 shows the status of CCR and EXR after execution of trap instruction exception handling.

Table 4.4 Status of CCR and EXR after Trap Instruction Exception Handling

	CCR		EXR	
Interrupt Control Mode	**I**	**UI**	**I2 to I0**	**T**
0	1	—	—	—
2	1	—	—	0

[Legend]

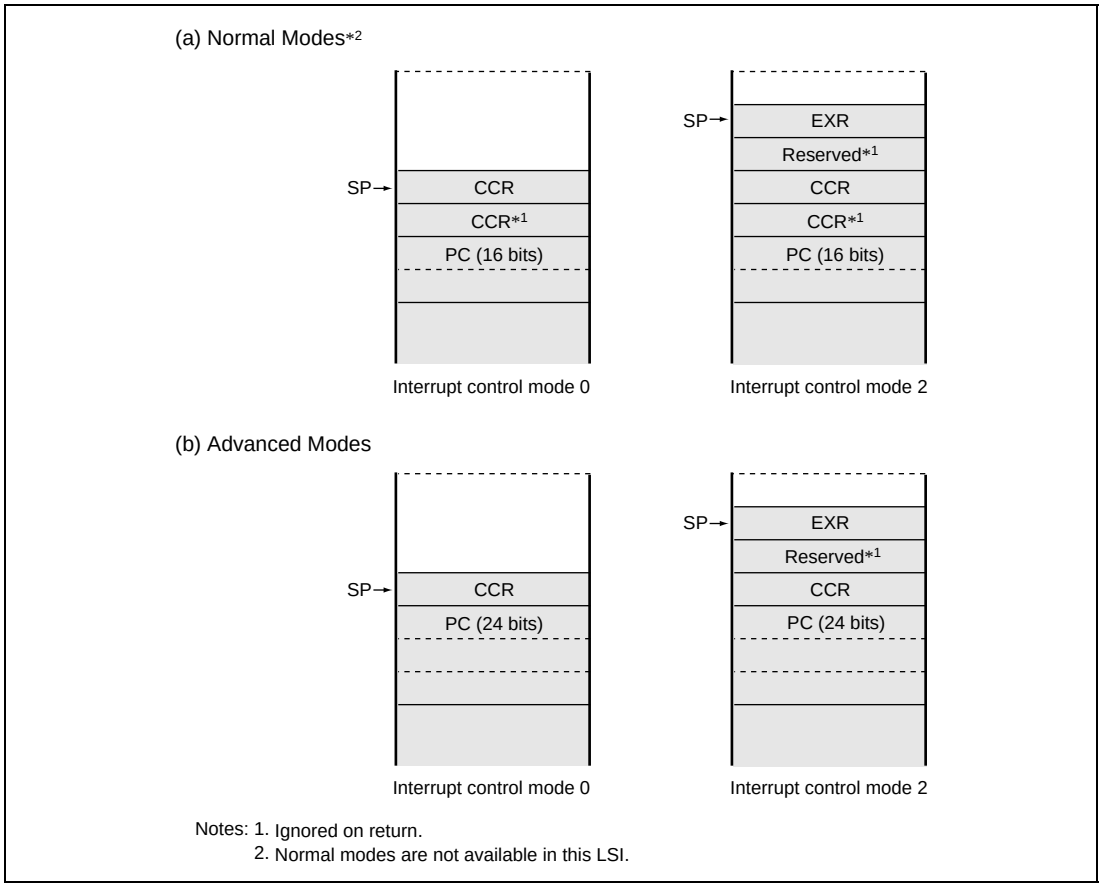
1: Set to 1

0: Cleared to 0

—: Retains value prior to execution.

4.7 Stack Status after Exception Handling

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



Notes: 1. Ignored on return.
2. Normal modes are not available in this LSI.

Figure 4.2 Stack Status after Exception Handling

4.8 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 by word size or longword size 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 operation when the SP value is odd.

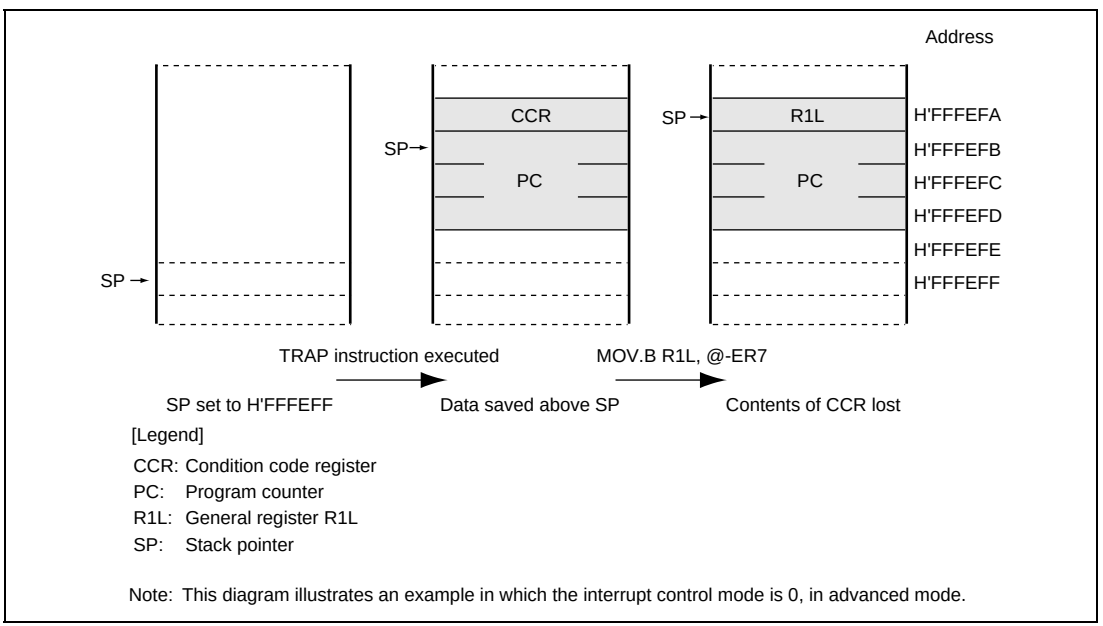


Figure 4.3 Operation when SP Value is Odd

Section 5 Interrupt Controller

5.1 Features

- Two interrupt control modes

  Any of two interrupt control modes can be set by means of the INTM1 and INTM0 bits in the interrupt control register (INTCR).
- Priorities settable with IPR

  An interrupt priority register (IPR) is provided for setting interrupt priorities. Eight priority levels can be set for each module for all interrupts except NMI. NMI is assigned the highest priority level of 8, and can be accepted at all times.
- 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 interrupts

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

A block diagram of the interrupt controller is shown in figure 5.1.

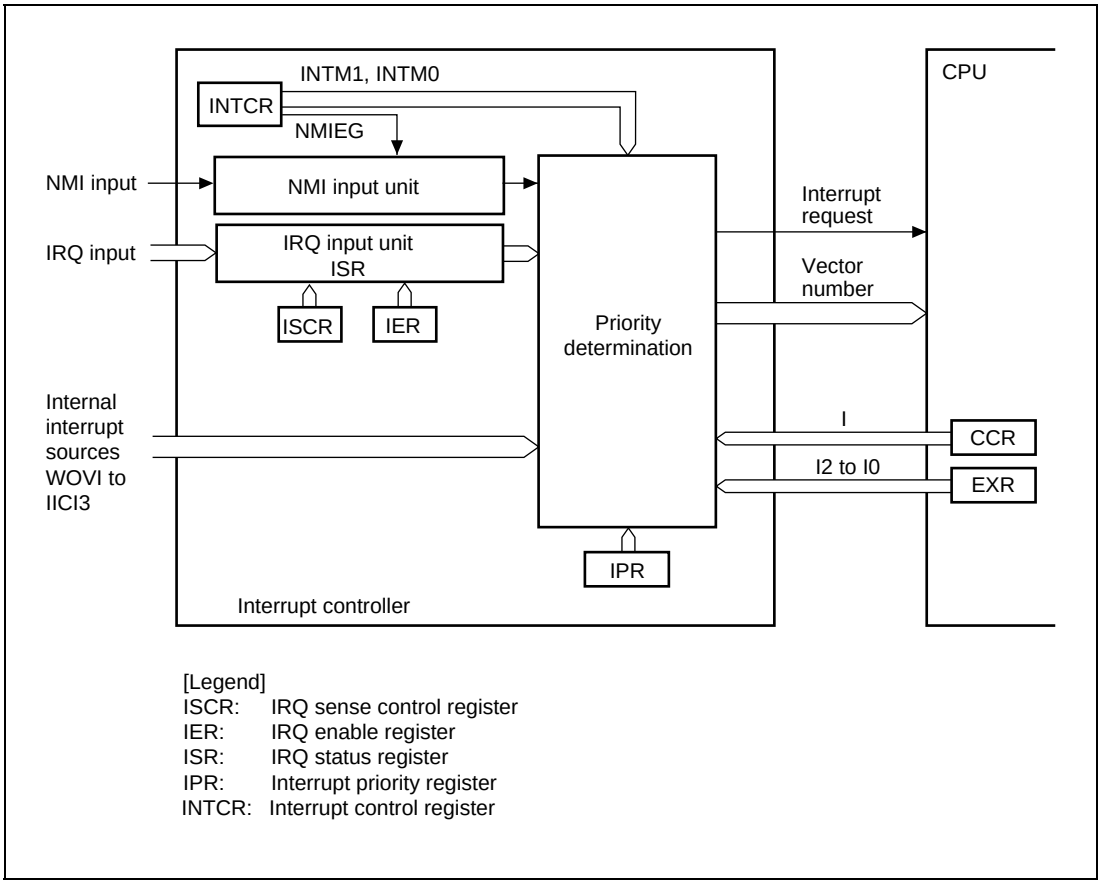


Figure 5.1 Block Diagram of Interrupt Controller

5.2 Input/Output Pins

Table 5.1 shows the pin configuration of the interrupt controller.

Table 5.1 Pin Configuration

Name	I/O	Function
NMI	Input	Nonmaskable external interrupt Rising or falling edge can be selected.
$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$	Input	Maskable external interrupts Rising, falling, or both edges, or level sensing, can be selected.

5.3 Register Descriptions

The interrupt controller has the following registers.

- Interrupt control register (INTCR)
- IRQ sense control register H (ISCR)
- IRQ enable register (IER)
- IRQ status register (ISR)
- Software standby release IRQ enable register (SSIER)
- Interrupt priority register A to K (IPRA to IPRK)

5.3.1 Interrupt Control Register (INTCR)

INTCR selects the interrupt control mode, and the detected edge for NMI.

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	INTM1	0	R/W	Interrupt Control Select Mode 1 and 0
4	INTM0	0	R/W	These bits select either of two interrupt control modes for the interrupt controller. 00: Interrupt control mode 0 Interrupts are controlled by I bit. 01: Setting prohibited 10: Interrupt control mode 2 Interrupts are controlled by bits I2 to I0 and IPR. 11: Setting prohibited.
3	NMIEG	0	R/W	NMI Edge Select Selects the input edge for the NMI pin. 0: Interrupt request generated at falling edge of NMI input 1: Interrupt request generated at rising edge of NMI input
2 to 0	—	All 0	R/(W)	Reserved The initial value should not be changed.

5.3.2 Interrupt Priority Registers A to K (IPRA to IPRK)

IPR are eleven 16-bit readable/writable registers that set priorities (levels 7 to 0) for interrupts other than NMI.

The correspondence between interrupt sources and IPR settings is shown in table 5.2. Setting a value in the range from H'0 to H'7 in the 3-bit groups of bits 14 to 12, 10 to 8, 6 to 4, and 2 to 0 sets the priority of the corresponding interrupt.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	—	Reserved This bit is always read as 0. Write is invalid.
14 13 12	IPR14 IPR13 IPR12	1 1 1	R/W R/W R/W	Sets the priority of the corresponding interrupt source. 000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)
11	—	0	—	Reserved This bit is always read as 0. Write is invalid.
10 9 8	IPR10 IPR9 IPR8	1 1 1	R/W R/W R/W	Sets the priority of the corresponding interrupt source. 000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)
7	—	0	—	Reserved This bit is always read as 0. Write is invalid.

Bit	Bit Name	Initial Value	R/W	Description
6 5 4	IPR6 IPR5 IPR4	1 1 1	R/W R/W R/W	Sets the priority of the corresponding interrupt source. 000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)
3	—	0	—	Reserved This bit is always read as 0. Write is invalid.
2 1 0	IPR2 IPR1 IPR0	1 1 1	R/W R/W R/W	Sets the priority of the corresponding interrupt source. 000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)

5.3.3 IRQ Enable Register (IER)

IER controls enabling and disabling of interrupt requests IRQ7 to IRQ0.

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7E	0	R/W	IRQ7 Enable The IRQ7 interrupt request is enabled when this bit is 1
6	IRQ6E	0	R/W	IRQ6 Enable The IRQ6 interrupt request is enabled when this bit is 1
5	IRQ5E	0	R/W	IRQ5 Enable The IRQ5 interrupt request is enabled when this bit is 1
4	IRQ4E	0	R/W	IRQ4 Enable The IRQ4 interrupt request is enabled when this bit is 1
3	IRQ3E	0	R/W	IRQ3 Enable The IRQ3 interrupt request is enabled when this bit is 1
2	IRQ2E	0	R/W	IRQ2 Enable The IRQ2 interrupt request is enabled when this bit is 1
1	IRQ1E	0	R/W	IRQ1 Enable The IRQ1 interrupt request is enabled when this bit is 1
0	IRQ0E	0	R/W	IRQ0 Enable The IRQ0 interrupt request is enabled when this bit is 1

5.3.4 IRQ Sense Control Registers (ISCR)

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

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ7SCB	0	R/W	IRQ7 Sense Control B
14	IRQ7SCA	0	R/W	IRQ7 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ7}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ7}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ7}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ7}}$ input
13	IRQ6SCB	0	R/W	IRQ6 Sense Control B
12	IRQ6SCA	0	R/W	IRQ6 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ6}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ6}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ6}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ6}}$ input
11	IRQ5SCB	0	R/W	IRQ5 Sense Control B
10	IRQ5SCA	0	R/W	IRQ5 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ5}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ5}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ5}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ5}}$ input

Bit	Bit Name	Initial Value	R/W	Description
9	IRQ4SCB	0	R/W	IRQ4 Sense Control B
8	IRQ4SCA	0	R/W	IRQ4 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ4}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ4}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ4}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ4}}$ input
7	IRQ3SCB	0	R/W	IRQ3 Sense Control B
6	IRQ3SCA	0	R/W	IRQ3 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ3}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ3}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ3}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ3}}$ input
5	IRQ2SCB	0	R/W	IRQ2 Sense Control B
4	IRQ2SCA	0	R/W	IRQ2 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ2}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ2}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ2}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ2}}$ input

Bit	Bit Name	Initial Value	R/W	Description
3	IRQ1SCB	0	R/W	IRQ1 Sense Control B
2	IRQ1SCA	0	R/W	IRQ1 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ1}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ1}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ1}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ1}}$ input
1	IRQ0SCB	0	R/W	IRQ0 Sense Control B
0	IRQ0SCA	0	R/W	IRQ0 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ0}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ0}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ0}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ0}}$ input

5.3.5 IRQ Status Register (ISR)

ISR is an IRQ7 to IRQ0 interrupt request flag register.

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7F	0	R/(W)*	[Setting condition] • When the interrupt source selected by ISCR occurs [Clearing conditions] • Cleared after reading condition1, when written as 0 • When interrupt exception handling is executed when low-level detection is set and $\overline{\text{IRQn}}$ input is high • When IRQn interrupt exception handling is executed while detecting the falling edge, rising edge, or both
6	IRQ6F	0	R/(W)*	
5	IRQ5F	0	R/(W)*	
4	IRQ4F	0	R/(W)*	
3	IRQ3F	0	R/(W)*	
2	IRQ2F	0	R/(W)*	
1	IRQ1F	0	R/(W)*	
0	IRQ0F	0	R/(W)*	

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

5.3.6 Software Standby Release IRQ Enable Register (SSIER)

SSIER selects the $\overline{\text{IRQ}}$ pins used to recover from the software standby state.

Bit	Bit Name	Initial Value	R/W	Description
7	SSI7	0	R/W	Software Standby Release IRQ Setting
6	SSI6	0	R/W	These bits select the $\overline{\text{IRQn}}$ pins used to recover from the software standby state.
5	SSI5	0	R/W	
4	SSI4	0	R/W	0: IRQn requests are not sampled in the software standby state (Initial value when n = 7 to 3).
3	SSI3	0	R/W	1: When IRQn request occurs in the software standby state, the chip recovers from the software standby state after the elapse of the oscillation settling time (Initial value when n = 2 to 0).
2	SSI2	1	R/W	
1	SSI1	1	R/W	
0	SSI0	1	R/W	

5.4 Interrupt Sources

5.4.1 External Interrupt Sources

There are nine external interrupts: NMI and IRQ7 to IRQ0. These interrupts can be used to restore the chip from software standby mode.

NMI Interrupt: Nonmaskable interrupt request (NMI) is the highest-priority interrupt, and is always accepted by the CPU regardless of the interrupt control mode or the status of the CPU interrupt mask bits. The NMIEG bit in INTCR can be used to select whether an interrupt is requested at a rising edge or a falling edge on the NMI pin.

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.

- 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 interrupt priority level can be set with IPR.
- The status of interrupt requests IRQ7 to IRQ0 is indicated in ISR. ISR flags can be cleared to 0 by software.

When IRQ7 to IRQ0 interrupt requests occur at low level of $\overline{\text{IRQn}}$, the corresponding $\overline{\text{IRQ}}$ should be held low until an interrupt handling starts. Then the corresponding $\overline{\text{IRQ}}$ should be set to high in the interrupt handling routine and clear the IRQnF bit (n = 0 to 7) in ISR to 0. Interrupts may not be executed when the corresponding IRQ is set to high before the interrupt handling starts.

Detection of IRQ7 to IRQ0 interrupts does not depend on whether the relevant pin has been set for input or output. However, when a pin is used as an external interrupt input pin, do not clear the corresponding DDR to 0 and use the pin as an I/O pin for another function.

A block diagram of interrupts IRQ7 to IRQ0 is shown in figure 5.2.

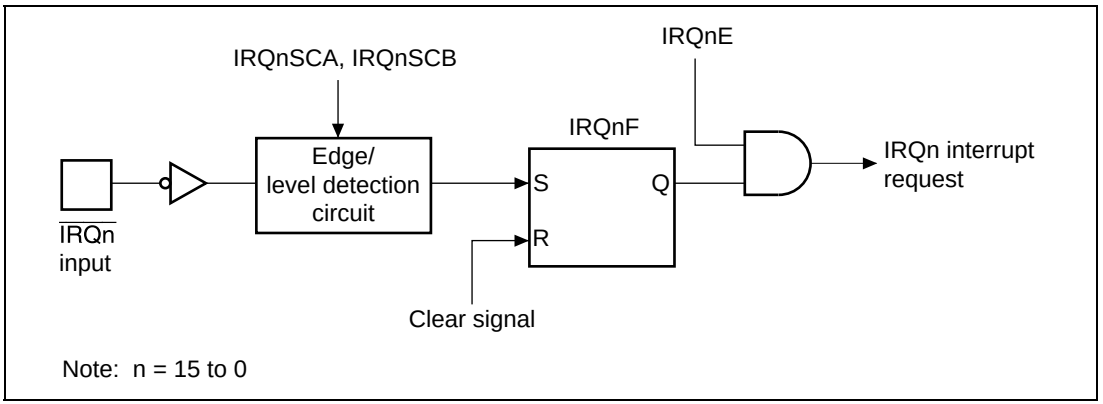


Figure 5.2 Block Diagram of Interrupts IRQ7 to IRQ0

5.4.2 Internal Interrupts

The sources for internal interrupts from on-chip peripheral modules have the following features:

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

5.5 Interrupt Exception Handling Vector Table

Table 5.2 shows interrupt exception handling sources, vector addresses, and interrupt priorities.

For default priorities, the lower the vector number, the higher the priority. When interrupt control mode 2 is set, priorities among modules can be changed by means of the IPR. Modules set at the same priority will conform to their default priorities. Priorities within a module are fixed.

Table 5.2 Interrupt Sources, Vector Addresses, and Interrupt Priorities

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*	IPR	Priority
			Advanced Mode		
External pin	NMI	7	H'001C	—	High
	IRQ0	16	H'0040	IPRA14 to IPRA12	
	IRQ1	17	H'0044	IPRA10 to IPRA8	
	IRQ2	18	H'0048	IPRA6 to IPRA4	
	IRQ3	19	H'004C	IPRA2 to IPRA0	
	IRQ4	20	H'0050	IPRB14 to IPRB12	
	IRQ5	21	H'0054	IPRB10 to IPRB8	
	IRQ6	22	H'0058	IPRB6 to IPRB4	
	IRQ7	23	H'005C	IPRB2 to IPRB0	
—	Reserved for system use	24	H'0060	IPRC14 to IPRC12	
		25	H'0064	IPRC10 to IPRC8	
		26	H'0068	IPRC6 to IPRC4	
		27	H'006C	IPRC2 to IPRC0	
		28	H'0070	IPRD14 to IPRD12	
		29	H'0074		
		30	H'0078	IPRD10 to IPRD8	
		31	H'007C		
		32	H'0080	IPRD6 to IPRD4	
WDT	WOVI	33	H'0084	IPRD2 to IPRD0	
—	Reserved for system use	34	H'0088	IPRE14 to IPRE12	
		35	H'008C		
		36	H'0090		
		37	H'0094		
		38	H'0098		
		39	H'009C		
		40	H'00A0		
		41	H'00A4		
		42	H'00A8	IPRE10 to IPRE8	
		43	H'00AC	IPRE10 to IPRE8	
		44	H'00B0	IPRE6 to IPRE4	
		45	H'00B4		Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*	IPR	Priority
			Advanced Mode		
A/D	ADI	46	H'00B8	IRPE2 to IRPE0	High
TPU_0	TGI0A	47	H'00BC	IPRF14 to IPRF12	
	TGI0B	48	H'00C0		
	TGI0C	49	H'00C4		
	TCI0D	50	H'00C8		
	TCI0V	51	H'00CC		
TPU_1	TGI1A	52	H'00D0	IPRF10 to IPRF8	
	TGI1B	53	H'00D4		
	TCI1V	54	H'00D8		
	TCI1U	55	H'00DC		
TPU_2	TGI2A	56	H'00E0	IPRF6 to IPRF4	
	TGI2B	57	H'00E4		
	TCI2V	58	H'00E8		
	TCI2U	59	H'00EC		
TMRX_0	CMIAX0	60	H'00F0	IPRF2 to IPRF0	
	CMIBX0	61	H'00F4		
	OVIX0	62	H'00F8		
	ICIX0	63	H'00FC		
FRT_0	ICIA0	64	H'0100	IPRG14 to IPRG12	
	ICIB0	65	H'0104		
	ICIC0	66	H'0108		
	ICID0	67	H'010C		
	OCIA0	68	H'0110		
	OCIB0	69	H'0114		
	FOVI0	70	H'0118		
TMR0_0	CMIA00	71	H'011C	IPRG10 to IPRG8	
	CMIB00	72	H'0120		
	OVI00	73	H'0124		
TMR1_0	CMIA10	74	H'0128	IPRG6 to IPRG4	
	CMIB10	75	H'012C	IPRG6 to IPRG4	
	OVI10	76	H'0130		Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority
TMRY_0	CMIAY0	77	H'0134	IPRG2 to IPRG0	High
	CMIBY0	78	H'0138		
	OVIY0	79	H'013C		
TMRX_1	CMIAX1	80	H'0140	IPRH14 to IPRH12	
	CMIBX1	81	H'0144		
	OVIX1	82	H'0148		
	ICIX1	83	H'014C		
FRT_1	ICIA1	84	H'0150	IPRH10 to IPRH8	
	ICIB1	85	H'0154		
	ICIC1	86	H'0158		
	ICID1	87	H'015C		
	OCIA1	88	H'0160		
	OCIB1	89	H'0164		
	FOVI1	90	H'0168		
TMR0_1	CMIA01	91	H'016C	IPRH6 to IPRH4	
	CMIB01	92	H'0170		
	OVI01	93	H'0174		
TMR1_1	CMIA11	94	H'0178	IPRH2 to IPRH0	
	CMIB11	95	H'017C		
	OVI11	96	H'0180		
TMRY_1	CMIAY1	97	H'0184	IPRI14 to IPRI12	
	CMIBY1	98	H'0188		
	OVIY1	99	H'018C		
Duty measure-ment circuit	TWOVI	100	H'0190	IPRI10 to IPRI8	
	TWENDI	101	H'0194		
SCI_0	ERI0	102	H'0198	IPRI6 to IPRI4	
	RXI0	103	H'019C		
	TXI0	104	H'01A0	IPRI6 to IPRI4	
	TEI0	105	H'01A4		Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*	IPR	Priority
			Advanced Mode		
SCI_1	ERI1	106	H'01A8	IPRI2 to IPRI0	High
	RXI1	107	H'01AC		
	TXI1	108	H'01B0		
	TEI1	109	H'01B4		
SCI_2	ERI2	110	H'01B8	IPRJ14 to IPRJ12	
	RXI2	111	H'01BC		
	TXI2	112	H'01C0		
	TEI2	113	H'01C4		
SCI_3	ERI3	114	H'01C8	IPRJ10 to IPRJ8	
	RXI3	115	H'01CC		
	TXI3	116	H'01D0		
	TEI3	117	H'01D4		
SCI_4	ERI4	118	H'01D8	IPRJ6 to IPRJ4	
	RXI4	119	H'01DC		
	TXI4	120	H'01E0		
	TEI4	121	H'01E4		
IIC3_0	IICI0	122	H'01E8	IPRJ2 to IPRJ0	
IIC3_1	IICI1	123	H'01EC	IPRK14 to IPRK12	
IIC3_2	IICI2	124	H'01F0	IPRK10 to IPRK8	
IIC3_3	IICI3	125	H'01F4	IPRK6 to IPRK4	
—	Reserved for system use	126	H'01F8	IPRK2 to IPRK0	
	Reserved for system use	127	H'01EC		Low

Note: * Lower 16 bits of the start address.

5.6 Interrupt Control Modes and Interrupt Operation

The interrupt controller has two modes: interrupt control mode 0 and interrupt control mode 2. Interrupt operations differ depending on the interrupt control mode. The interrupt control mode is selected by INTCR. Table 5.3 shows the differences between interrupt control mode 0 and interrupt control mode 2.

Table 5.3 Interrupt Control Modes

Interrupt Control Mode	Priority Setting Registers	Interrupt Mask Bits	Description
0	Default	I	The priorities of interrupt sources are fixed at the default settings. Interrupt sources except for NMI is masked by the I bit.
2	IPR	I2 to I0	8 priority levels except for NMI can be set with IPR. 8-level interrupt mask control is performed by bits I2 to I0.

5.6.1 Interrupt Control Mode 0

In interrupt control mode 0, the I bit in CCR of the CPU controls whether interrupts except for the NMI are masked or not. Figure 5.3 shows a flowchart of the interrupt acceptance operation in this case.

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. If the I bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending. If the I bit is cleared, an interrupt request is accepted.
3. Interrupt requests are sent to the interrupt controller, the highest-ranked interrupt according to the priority system is accepted, and other interrupt requests are held pending.
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 NMI.
7. The CPU generates a vector address for the accepted interrupt and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

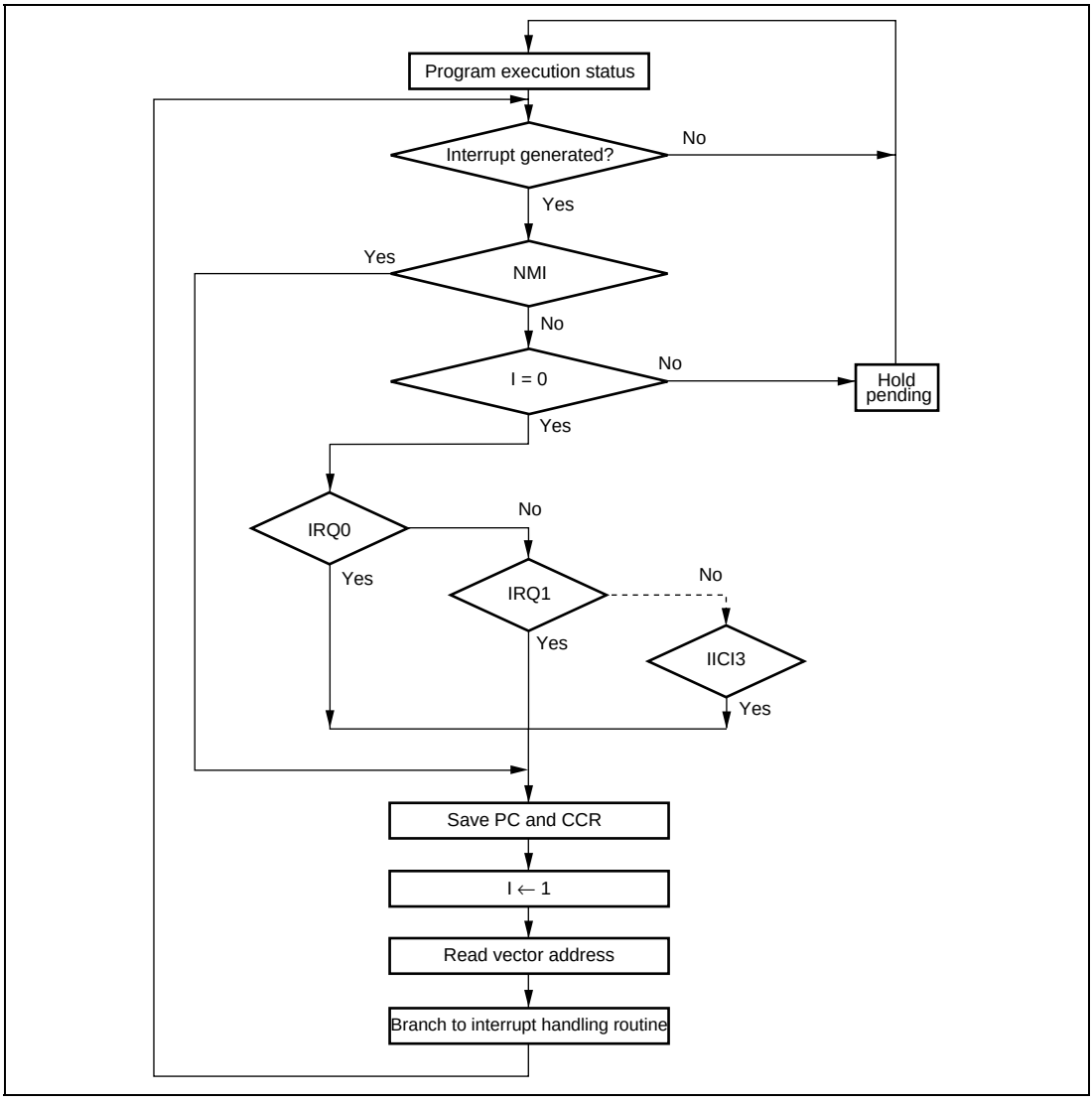


Figure 5.3 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

5.6.2 Interrupt Control Mode 2

In interrupt control mode 2, mask control is done in eight levels for interrupt requests except for NMI by comparing the EXR interrupt mask level (I2 to I0 bits) in the CPU and the IPR setting. Figure 5.4 shows a flowchart of the interrupt acceptance operation in this case.

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. When interrupt requests are sent to the interrupt controller, the interrupt with the highest priority according to the interrupt priority levels set in IPR is selected, and lower-priority interrupt requests are held pending. If a number of interrupt requests with the same priority are generated at the same time, the interrupt request with the highest priority according to the priority system shown in table 5.2 is selected.
3. Next, the priority of the selected interrupt request is compared with the interrupt mask level set in EXR. An interrupt request with a priority no higher than the mask level set at that time is held pending, and only an interrupt request with a priority higher than the interrupt mask level 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, CCR, and EXR 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 T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priority level of the accepted interrupt.

   If the accepted interrupt is NMI, the interrupt mask level is set to H'7.
7. The CPU generates a vector address for the accepted interrupt and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

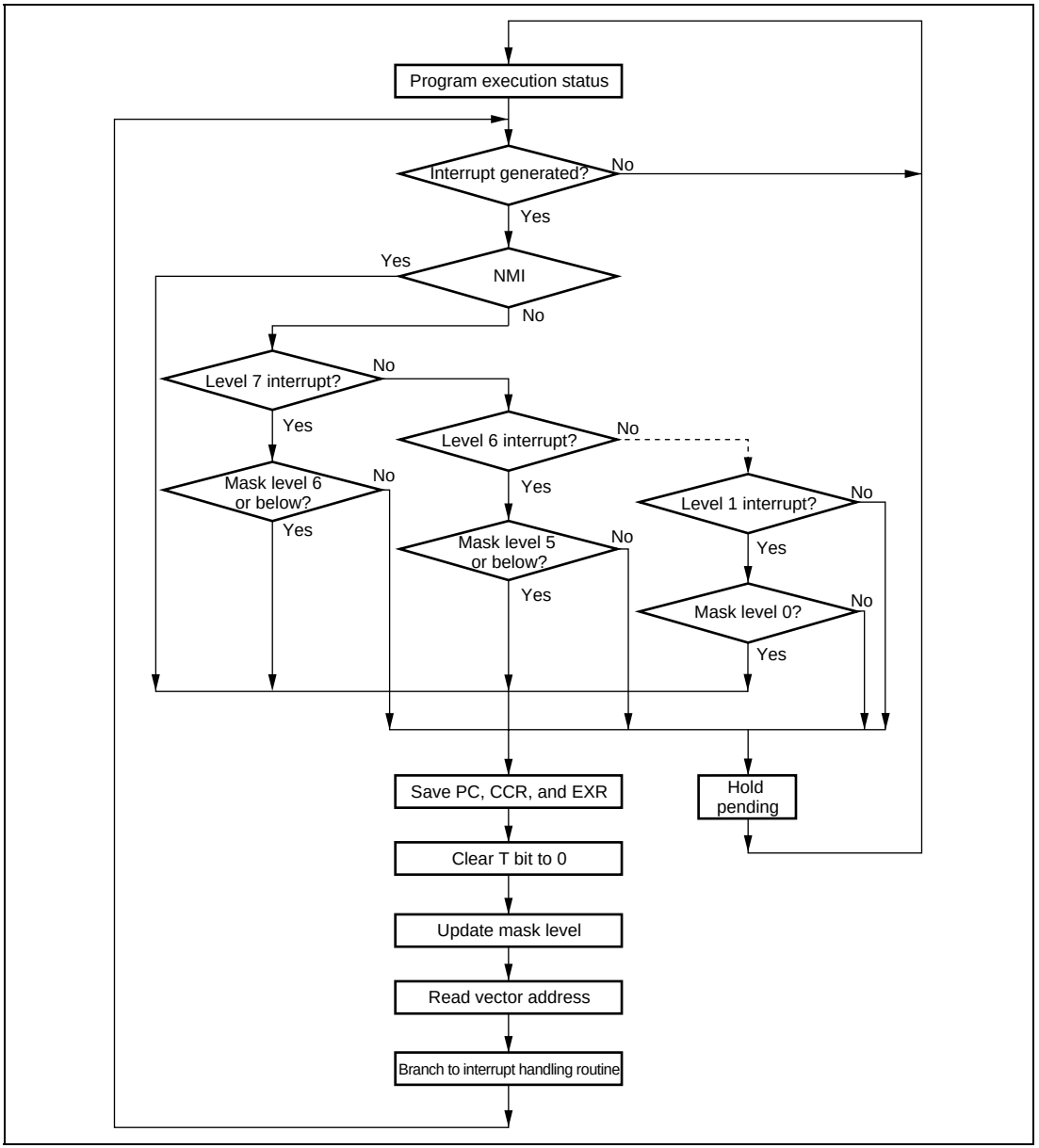


Figure 5.4 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

5.6.3 Interrupt Exception Handling Sequence

Figure 5.5 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.

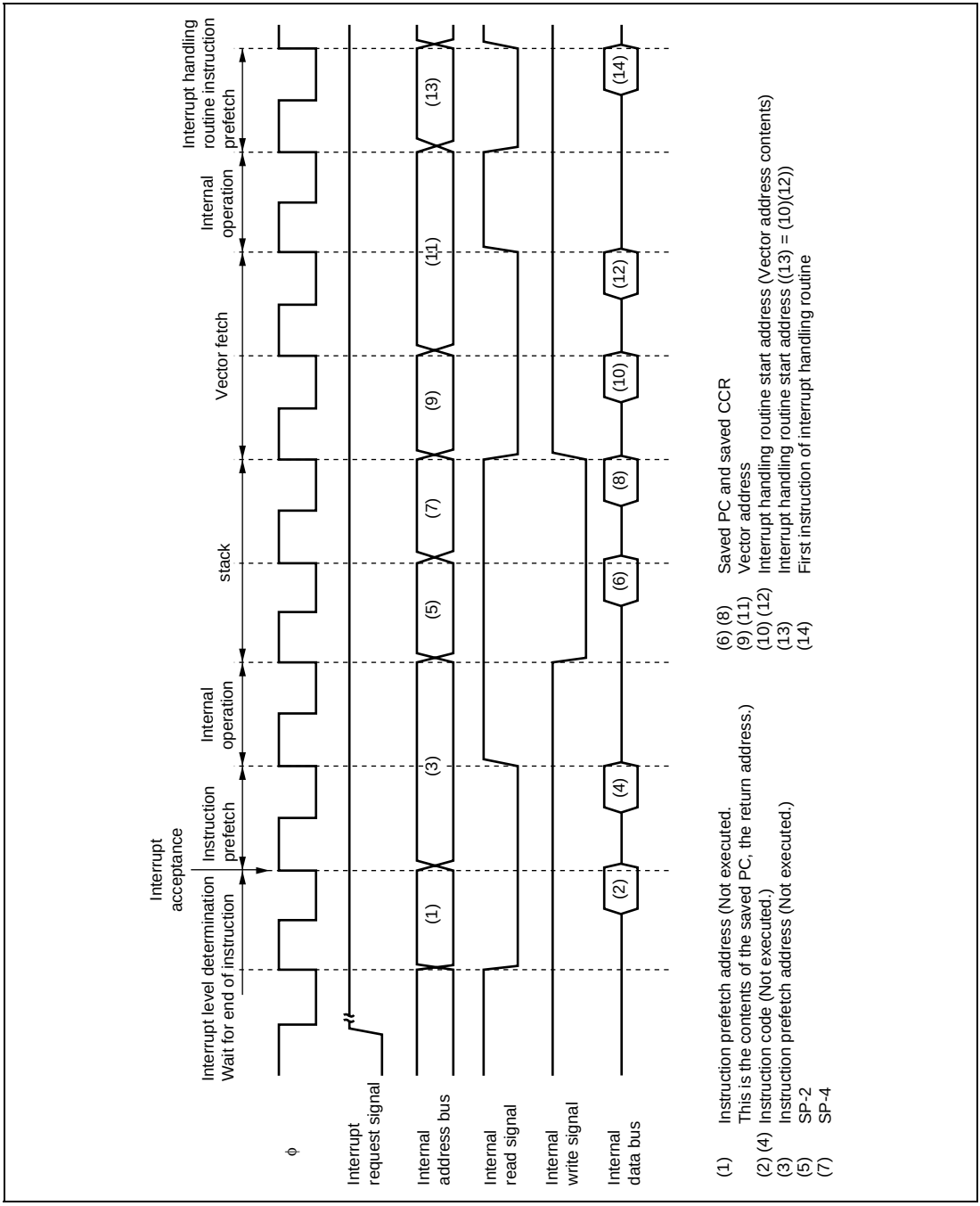


Figure 5.5 Interrupt Exception Handling

5.6.4 Interrupt Response Times

Table 5.4 shows interrupt response times - the interval between generation of an interrupt request and execution of the first instruction in the interrupt handling routine. The execution status symbols used in table 5.4 are explained in table 5.5.

This LSI is capable of fast word transfer to on-chip memory, and have the program area in on-chip ROM and the stack area in on-chip RAM, enabling high-speed processing.

Table 5.4 Interrupt Response Times

		Normal Mode*[5]		Advanced Mode	
No.	**Execution Status**	**Interrupt control mode 0**	**Interrupt control mode 2**	**Interrupt control mode 0**	**Interrupt control mode 2**
1	Interrupt priority determination*[1]	3	3	3	3
2	Number of wait states until executing instruction ends*[2]	1 to 19 +2·S_I	1 to 19+2·S_I	1 to 19+2·S_I	1 to 19+2·S_I
3	PC, CCR, EXR stack save	2·S_K	3·S_K	2·S_K	3·S_K
4	Vector fetch	S_I	S_I	2·S_I	2·S_I
5	Instruction fetch*[3]	2·S_I	2·S_I	2·S_I	2·S_I
6	Internal processing*[4]	2	2	2	2
Total (using on-chip memory)		11 to 31	12 to 32	12 to 32	13 to 33

Notes: 1. Two states in case of internal interrupt.
2. Refers to DIVXS instructions.
3. Prefetch after interrupt acceptance and interrupt handling routine prefetch.
4. Internal processing after interrupt acceptance and internal processing after vector fetch.
5. Not available in this LSI.

Table 5.5 Number of States in Interrupt Handling Routine Execution Statuses

	Object of Access				
		External Device			
		8 Bit Bus		16 Bit Bus	
Symbol	**Internal Memory**	**2-State Access**	**3-State Access**	**2-State Access**	**3-State Access**
Instruction fetch S_I	1	4	6+2m	2	3+m
Branch address read S_J					
Stack manipulation S_K					

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

5.7 Usage Notes

5.7.1 Contention between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to mask interrupts, the masking becomes effective after execution of the instruction.

When an interrupt enable bit is cleared by an instruction such as BCLR or MOV, if an interrupt is generated during execution of the instruction, the interrupt concerned will still be enabled on completion of the instruction, and so interrupt exception handling for that interrupt will be executed on completion of the instruction. However, if there is an interrupt request 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 also applies when an interrupt source flag is cleared to 0. Figure 5.6 shows an example in which the TCIEV bit in the TPU's TIER_0 register is cleared to 0. The above contention will not occur if an enable bit or interrupt source flag is cleared to 0 while the interrupt is masked.

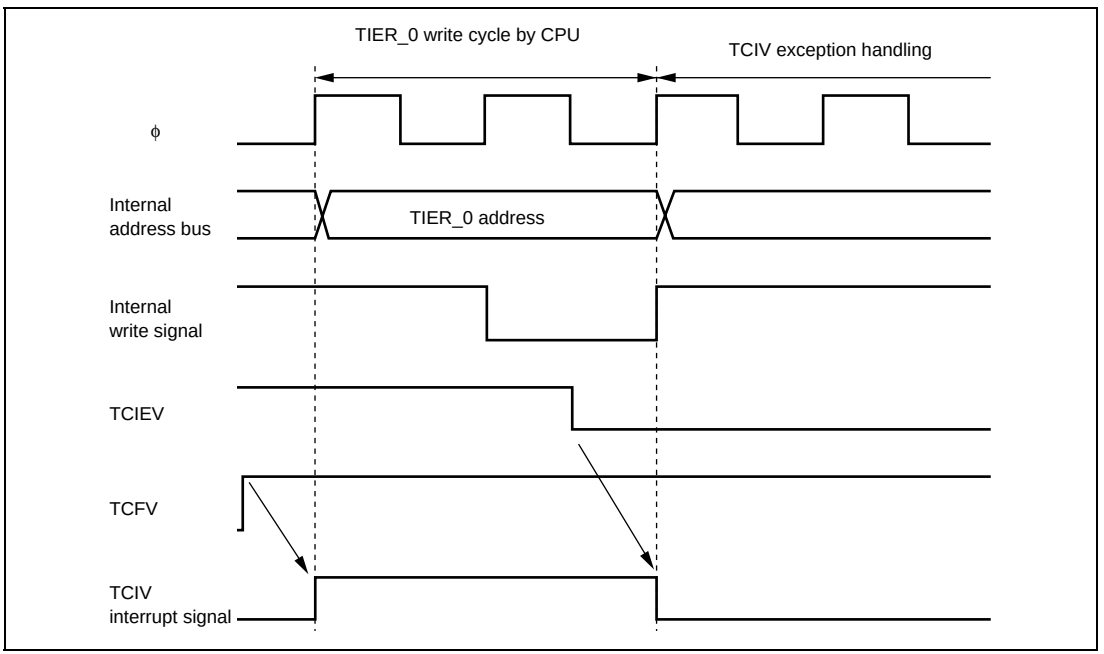


Figure 5.6 Contention between Interrupt Generation and Disabling

5.7.2 Instructions that Disable Interrupts

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

5.7.3 Times when Interrupts are Disabled

There are times when interrupt acceptance is disabled by the interrupt controller.

The interrupt controller disables interrupt acceptance for a 3-state period after the CPU has updated the mask level with an LDC, ANDC, ORC, or XORC instruction.

5.7.4 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 the transfer is not accepted until the transfer is completed.

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

5.7.5 $\overline{\text{IRQ}}$ Pin Select

IRQ input pins can be selected from port control register 1 (PTCNT1). For details on selectable pins, refer to section 7, I/O Ports.

When the PTCNT1 setting is changed, an edge occurs internally and the IRQnF bit (n = 0 to 7) of ISR may be set to 1 at the unintended timing if the selected pin level before the change is different from the selected pin level after the change. If the IRQn interrupt request (n = 0 to 7) is enabled, the interrupt exception handling is executed. To prevent the unintended interrupt, ITSR setting should be changed while the IRQn interrupt request is disabled, then the IRQnF bit should be cleared to 0.

5.7.6 Note on IRQ Status Register (ISR)

Since IRQnF flags may be set to 1 depending on the pin states after a reset, be sure to read from ISR after a reset and then write 0 to clear the IRQnF flags.

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.

6.1 Features

- Extended modes

  Two modes for external extension

  Normal extended mode: Normal extension (when the ADMXE bit in BCR is 0)

  Address-data multiplex extended mode: Multiplex extension (when the ADMXE bit in BCR is 1)
- Extended area division

  The external address space is divided into a basic area, and three 64-kbyte areas

  The basic area and area 1 are for common settings. Area 2 and 3 bus specifications can be set independently

  Areas 1, 2, and 3 enable chip-select ($\overline{CS1}$ to $\overline{CS3}$) output

  A maximum of 16 addresses can be output
- Area select signal, address strobe/hold signal polarity control
- It is possible to reverse the output polarity of $\overline{CS1}$ to $\overline{CS3}$ and $\overline{AS}$/$\overline{AH}$ by the PNCCS bit in BCRAn or the PNCASH bit in BCR

Normal Extension:

Address output pins (A15 to A0) and data input/output pins (D15 to D0) are separate

- Usable areas

  Basic area and areas 1, 2, and 3 are all usable
- Normal extended bus interface

  Selection between 2-state access area and 3-state access area is possible

  Program wait state insert is possible
- Idle cycle insertion

  Idle cycle insert is possible during the external write cycle, directly after external read cycle.

Multiplex Extension:

The address output pins and data input/output pins are multiplex pins

- Minimization of number of pins
  It is possible to minimize the number of pins necessary for expansion by multiplexing the address output pins and data input/output pins.
- Usable areas
  Areas 1, 2, and 3 are all usable
- Multiplex extended bus interface
  In the address cycle there are 2-state access fixed areas
  In the data cycle 2-state access areas or 3-state access areas are able to be selected
  The address cycle or data cycle can be independently inserted into the program wait state
- Idle cycle insert
  Idle cycle insert is possible during the external write cycle, directly after external read cycle

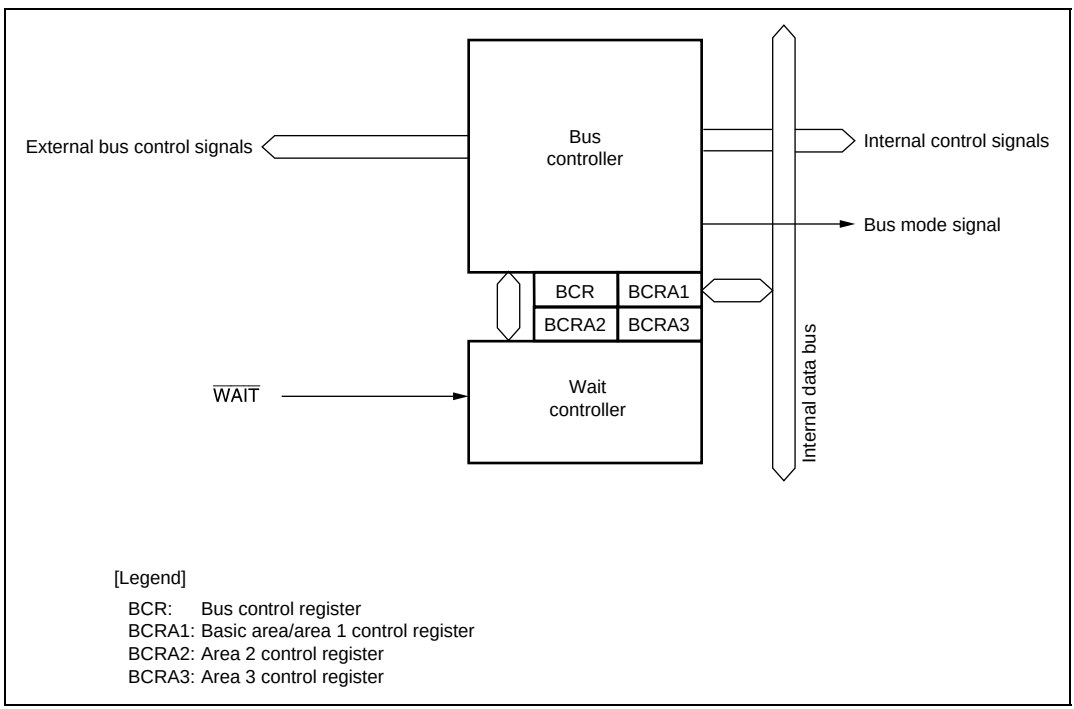


Figure 6.1 Block Diagram of Bus Controller

6.2 Input/Output Pins

Table 6.1 summarizes the pin configuration of the bus controller.

Table 6.1 Pin Configuration

Symbol	I/O	Function
$\overline{AS}$	Output	Strobe signal indicating that address output on address bus is enabled, during normal expansion
$\overline{CS1}$	Output	Chip select signal indicating that area 1 is accessed
$\overline{CS2}$	Output	Chip select signal indicating that area 2 is accessed
$\overline{CS3}$	Output	Chip select signal indicating that area 3 is accessed
$\overline{RD}$	Output	Strobe signal indicating that the external address area is being read
$\overline{HWR}$	Output	Strobe signal indicating that external address space is written to, and upper half (D15 to D8/AD15 to AD8) of data bus is enabled
$\overline{LWR}$	Output	Strobe signal indicating that basic bus interface space is written to, and lower half (D7 to D0/AD7 to AD0) of data bus is enabled
$\overline{WAIT}$	Input	Wait request signal when accessing external space
$\overline{AH}$	Output	Indicates the address fetch timing signal when in multiplex extension
AD15 to AD0	I/O	Address output and data input/output pins

6.3 Register Descriptions

Registers related to the bus controller are as follows.

- Bus control register (BCR)
- Basic area/area 1 control register (BCRA1)
- Area 2 control register (BCRA2)
- Area 3 control register (BCRA3)

6.3.1 Bus Control Register (BCR)

BCR is used to specify the external extended selection, inversion control of $\overline{CS1}$ to $\overline{CS3}$, $\overline{AS}$, and $\overline{AH}$ pins, as well as idle cycle insertion.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	R/(W)	Reserved The initial value should not be changed.
6	ICIS	1	R/W	Idle Cycle Insert When external read cycle and external write cycle continue, it selects whether idle cycle 1-state is inserted or idle cycle is not inserted. 0: Idle cycle is not inserted 1: Idle cycle 1-state is inserted
5	—	1	R/(W)	Reserved The initial value should not be changed.
4	—	0	R/(W)	Reserved The initial value should not be changed.
3	—	0	R/(W)	Reserved The initial value should not be changed.
2	—	0	R/(W)	Reserved The initial value should not be changed.
1	PNCASH	0	R/W	Address Strobe/Hold Polarity Control Controls output polarity of address strobe signal ($\overline{AS}$) and address hold signal ($\overline{AH}$). 0: $\overline{AS}$/$\overline{AH}$ output 1: $\overline{AS}$/$\overline{AH}$ output

Bit	Bit Name	Initial Value	R/W	Description
0	ADMXE	0	R/W	Address and Data Multiplex Bus Interface Enable Selects the type of external extended bus interface. 0: Normal extended bus interface 1: Address and data multiplex extended bus interface

6.3.2 Area Control Register (BCRA)

BCRA designates the access mode in area 1 to area 3. The basic area indicates the setting of area 1.

Bit	Bit Name	Initial Value	R/W	Description
7	ABWn	1	R/W	Area Bus Width Control Selects the bus width for area n. 0: 16-bit 1: 8-bit
6	ASTn	1	R/W	Area Access State Control Designates the number of access states in area n. Simultaneously permits or prohibits wait state insertion. Normal extension (ADMXE = 0): 0: 2-state access area, wait state insertion prohibited 1: 3-state access area, wait state insertion permitted Multiplex extension (ADMXE = 1): 0: Data 2-state access area, wait state insertion prohibited 1: Data 3-state access area, wait state insertion permitted
5	PNCCSn	0	R/W	Chip Select Polarity Control Controls the output polarity of the chip select signal ($\overline{\text{CSn}}$) for area n. 0: $\overline{\text{CSn}}$ output 1: CSn output

Bit	Bit Name	Initial Value	R/W	Description
4	AWn	1	R/W	Multiplex Extended Address Wait Selects the number of address cycle program waits in area n. Normal extension (ADMXE = 0): Ignored Multiplex extension (ADMXE = 1): 0: Program wait is not inserted 1: 1-state program wait is inserted into the address cycle
3	WMSn1	0	R/W	Area n Wait Mode Select 1, 0 When the ASTn bit is set to 1 and area n is accessed, wait mode is selected. 00: Program wait mode 01: Wait prohibited mode 10: Pin wait mode 11: Pin auto wait mode
2	WMSn0	0		
1	WCn1	1	R/W	Area n Wait Count 1, 0 Selects the number of data cycle program waits, when area n is accessed. 00: Program wait is not inserted 01: 1-state is inserted into program wait 10: 2-state is inserted into program wait 11: 3-state is inserted into program wait
0	WCn0	1		

[Legend]

n = 1 to 3

6.4 Bus Control

6.4.1 Bus Specifications

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

Normal Extended Mode:

1. Bus Width

   A bus width of 8 or 16 bits can be selected with the ABWn bit in BCRAn.
2. Number of Access States

   The number of access states for data access , 2-state or 3-state can be selected with the ASTn bit in BCRAn. When 2-state access space is designated, wait state insertion is disabled.
3. Wait Mode and Number of Program Wait States

   When 3-state access space is designated by the ASTn bit in BCRAn, the number of program wait states to be inserted automatically is selected with WMSn1, WMSn0, WCn1, and WCn0 in the BCRAn. From 0 to 3 program wait states can be selected.

   The external extended wait function is effective when the low-speed device is connected to the external address area.

   For details on normal extended address range, external address area, as well as bus interface specifications, refer to tables 6.2 and 6.3.

Table 6.2 Address Range and External Address Area (Normal Extended Mode)

Address Range	Area
H'080000 to H'FBFFFF (15 Mbytes)	Basic area (shares bus specification with area 1)
H'FC0000 to H'FCFFFF (64 kbytes)	Area 1
H'FD0000 to H'FDFFFF (64 kbytes)	Area 2
H'FE0000 to H'FEFFFF (64 kbytes)	Area 3
H'FF0000 to H'FF9FFF (40 kbytes)	Integrated with basic area when RAME = 0

Table 6.3 Bus Specifications for Normal Extended Bus Interface

ASTn	WMSn1	WMSn0	WCn1	WCn0	Number of Access States	Number of Program Wait States
0	—	—	—	—	2	0
1	0	1	—	—	3	0
	0	0	0	0	3	0
	1	*		1	3	1
			1	0	3	2
				1	3	3

[Legend]
n = 1 to 3
*: Don't care.

Multiplex Extended Mode:

The bus access is used as both the address bus and the data bus, but not simultaneously.

1. Bus Width

   A bus width of 8 or 16 bits can be selected with the ABWn bit in BCRAn. The bus width of the address cycle and the data cycle are the same.
2. Number of Access States
   - — Address cycle state

     The address cycle state is 2-state.
   - — Data cycle state

     The number of access states for data access , 2-state or 3-state can be selected with the ASTn bit in BCRAn. When 2-state access space is designated, wait state insertion is disabled.
3. Wait Mode and Number of Program Wait States
   - — Address cycle wait

     The number of program wait states to be inserted into the address cycle are selected by the AWn bit in BCRAn. 0 or 1 address cycle program wait states can be selected. The address cycle wait is not affected by the number of data access wait states or the wait mode.
   - — Data cycle wait

     When 3-state access space is designated by the ASTn bit in BCRAn, the number of program wait states to be inserted automatically are selected with WMSn1, WMSn0, WCn1, and WCn0 in the BCRAn. From 0 to 3 data cycle wait states can be selected.

The external extended wait function is effective when the low-speed device is connected to the external address area.

For details on multiplex extended address range, external address area, as well as bus interface specifications, refer to tables 6.4 to 6.6.

Table 6.4 Address Range and External Address Area (Multiplex Extended Mode)

Address Range	Area
H'080000 to H'FBFFFF (15 Mbytes)	Basic area
H'FC0000 to H'FCFFFF (64 kbytes)	Area 1
H'FD0000 to H'FDFFFF (64 kbytes)	Area 2
H'FE0000 to H'FEFFFF (64 kbytes)	Area 3
H'FF0000 to H'FF9FFF (40 kbytes)	Integrated with basic area when RAME = 0 (use prohibited)

Table 6.5 Bus Specifications for Multiplex Extended Bus Interface (Address Cycle)

ASTn	AWn	WMSn1	WMSn0	WCn1	WCn0	Number of Access States	Number of Program Wait States
—	0	—	—	—	—	2	0
	1	—	—	—	—	2	1

[Legend]

n = 1 to 3

Table 6.6 Bus Specifications for Multiplex Extended Bus Interface (Data Cycle)

ASTn	AWn	WMSn1	WMSn0	WCn1	WCn0	Number of Access States	Number of Program Wait States
0	—	—	—	—	—	2	0
1	—	0	1	—	—	3	0
	—	0	0	0	0	3	0
		1	*		1	3	1
				1	0	3	2
					1	3	3

[Legend]

n = 1 to 3

*: Don’t care.

6.4.2 External Address Area

The initial condition of the external address space is normal extended 3-state access space. The space outside the on-chip ROM, on-chip RAM, internal I/O register, and their reserved areas are available as the external address spaces. When the RAME bit in SYSCR is set to 1, the on-chip RAM and its reserved area are enabled. However if the RAME bit is cleared to 0, the on-chip RAM and its reserved area are ignored. When the RAME bit is 0, H'FF0000 to H'FF9FFF of the on-chip RAM and its reserved area, becomes an external address area.

6.4.3 Chip Select Signals

This LSI can output chip select signals ($\overline{CS1}$ to $\overline{CS3}$) for areas 0 to 3 respectively. The $\overline{CS1}$ to $\overline{CS3}$ signal outputs low or high level when the corresponding external space area is accessed. The chip select signal's output polarity can be controlled by the PNCCSn bit in BCRAn. Figure 6.2 shows an example of $\overline{CS1}$ to $\overline{CS3}$ signal's output polarity and output timing.

Selection of $\overline{CS1}$ to $\overline{CS3}$ signal output and I/O port input/output is set by the port function control register (PFCR) bit for the port corresponding to the $\overline{CS1}$ to $\overline{CS3}$ pins. In external extended mode, all of the $\overline{CS1}$ to $\overline{CS3}$ pins function as I/O ports after a reset. Therefore the corresponding PFCR bits should be set to 1 when outputting signals $\overline{CS1}$ to $\overline{CS3}$. For details, refer to section 7, I/O Ports.

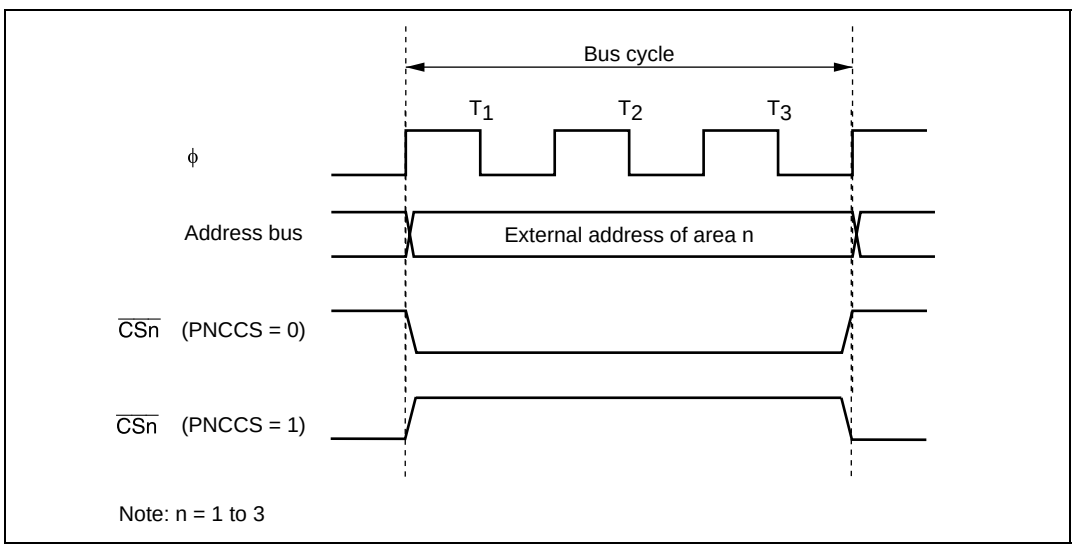


Figure 6.2 $\overline{CSn}$ Signal Output Polarity and Output Timing

6.4.4 Address Strobe/Hold Signal

In normal extended mode, the address above the bus address is enabled, which is indicated by the output strobe signal ($\overline{AS}$). In multiplex extended mode, the hold signal ($\overline{AH}$) which indicates the address fetch timing, is output. Output polarity of the $\overline{AS}$/ $\overline{AH}$ signals can be controlled by the PNCASH bit in BCR.

6.4.5 Address Output

This LSI can output a maximum of 16 addresses.

In normal extended mode, enabling or disabling of A15 to A0 signal output is set by the data direction register (DDR) bit for the port corresponding to the A15 to A0 pins. In external extended mode, the A15 to A0 pins are placed in the input state after a reset, the corresponding DDR bits should be set to 1 when outputting signals A15 to A0. For details, refer to section 7, I/O Ports.

In multiplex extended mode, the address output is decided by the access area bus width. If the access area is 16 bits, the lower 16-bit internal addresses are output from A15 to A0 in the address cycle. If the access area is 8 bits, the lower 8-bit internal addresses are output from A15 to A8 in the address cycle.

6.5 Bus Interface

The normal extended bus interface enables direct connection between the ROM and SRAM. For details on the basic area and areas 1 to 3 bus specification selection, refer to tables 6.2 and 6.3.

For multiplex extended bus interface, only products compatible with this bus system can be directly connected. For details on areas 1 to 3 bus specification selection, refer to tables 6.4 to 6.6

6.5.1 Data Size and Data Alignment

Data sizes for the CPU are byte, word, and longword. The BSC has a data alignment function, and controls whether the upper data bus (D15 to D8/AD15 to AD8) or lower data bus (D7 to D0/AD7 to AD0) 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. The multiplex extended address cycle is fixed to the bus specifications of the area being accessed (8-bit access space or 16-bit access space).

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/AD15 to AD8) 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.

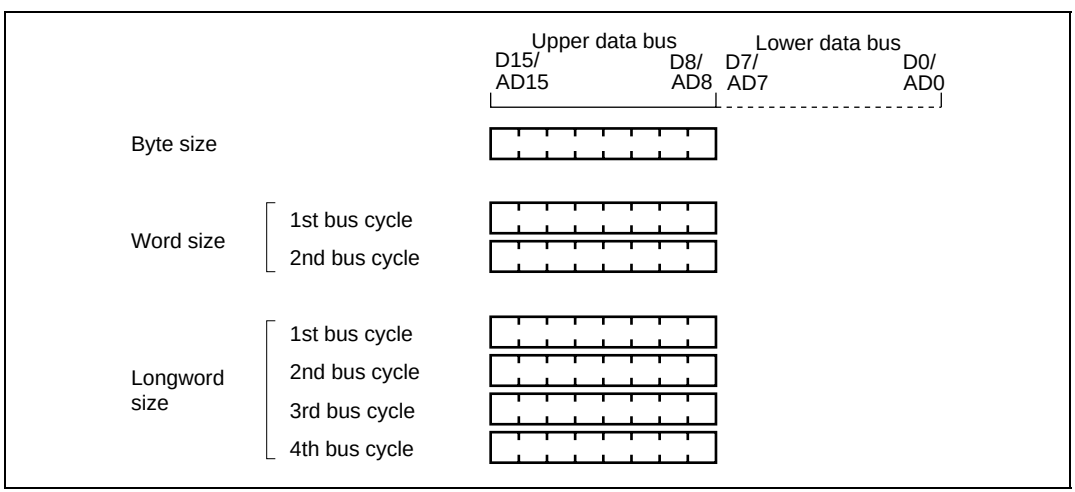


Figure 6.3 Access Sizes and Data Alignment Control (8-Bit Access Space)

16-Bit Access Space:

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/AD15 to AD8) and lower data bus (D7 to D0/AD7 to AD0) are used for accesses. The amount of data that can be accessed at one time is one byte or one word. 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 even addresses, and the lower data bus for odd addresses.

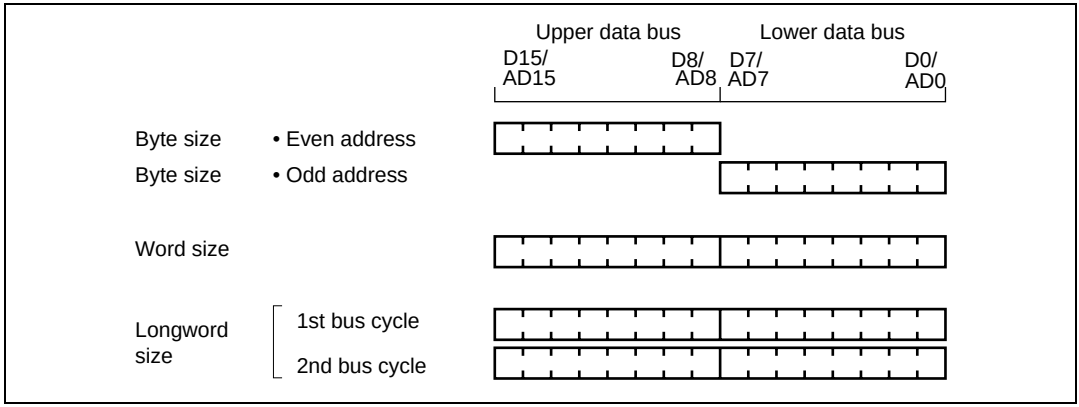


Figure 6.4 Access Sizes and Data Alignment Control (16-Bit Access Space)

6.5.2 Valid Strobes

Table 6.7 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.

Table 6.7 Data Buses Used and Valid Strobes

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D15 to D8/AD15 to AD8)	Lower Data Bus (D7 to D0/AD7 to AD0)
8-bit access space	Byte	Read	—	$\overline{RD}$	Valid	Ports or others
		Write	—	$\overline{HWR}$		Ports or others
16-bit access space	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

Note: Undefined: Undefined data is output.
Invalid: Input state with the input value ignored.
Ports or others: Input/output pins for ports or on-chip peripheral devices cannot be used as data buses.

6.5.3 Basic Operation Timing in Normal Extended Mode

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.

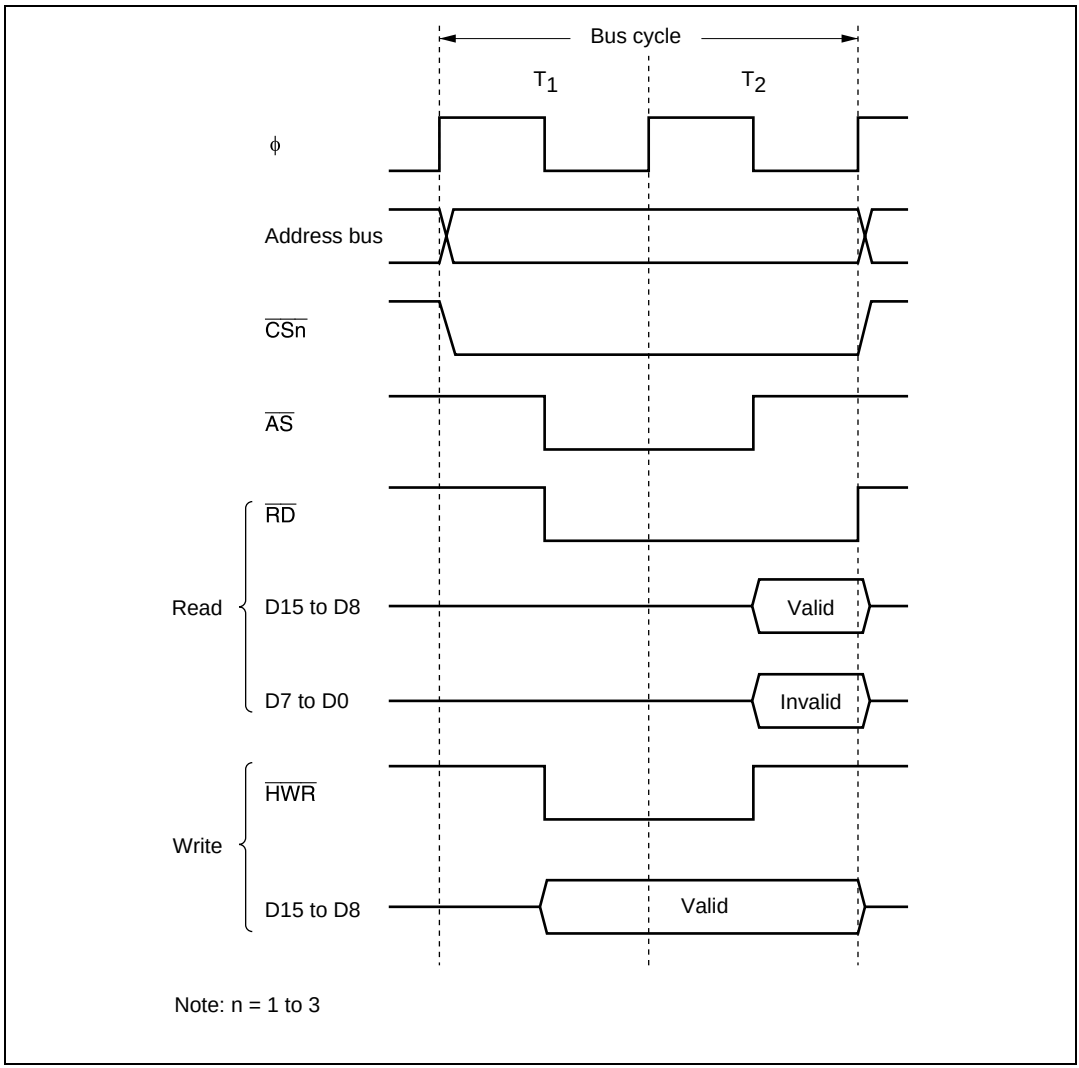


Figure 6.5 Bus Timing for 8-Bit, 2-State Access Space

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.

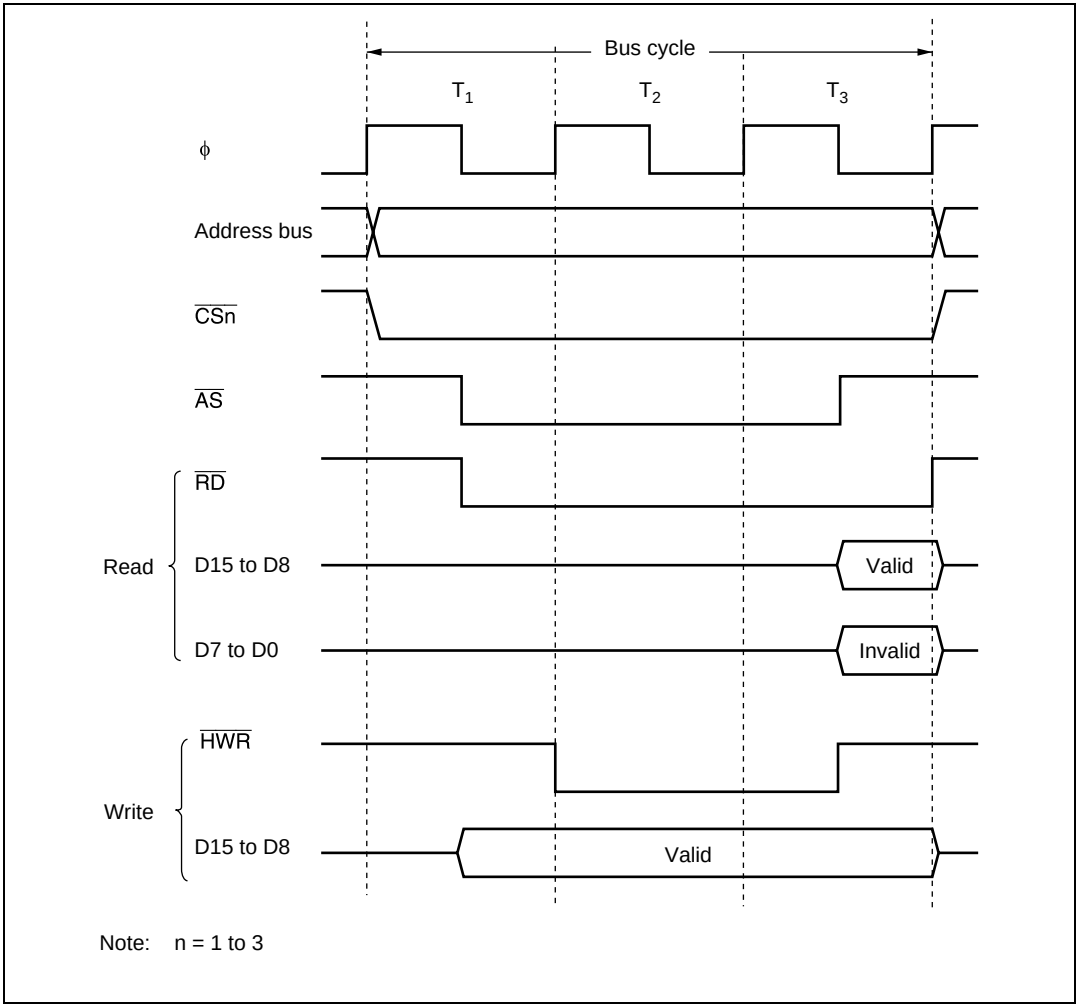


Figure 6.6 Bus Timing for 8-Bit, 3-State Access Space

16-Bit, 2-State Access Space:

Figures 6.7 to 6.9 show bus timings for a 16-bit, 2-state access space. When a 16-bit access space is accessed, the upper half (D15 to D8) of the data bus is used for even addresses, and the lower half (D7 to D0) for odd addresses. Wait states cannot be inserted.

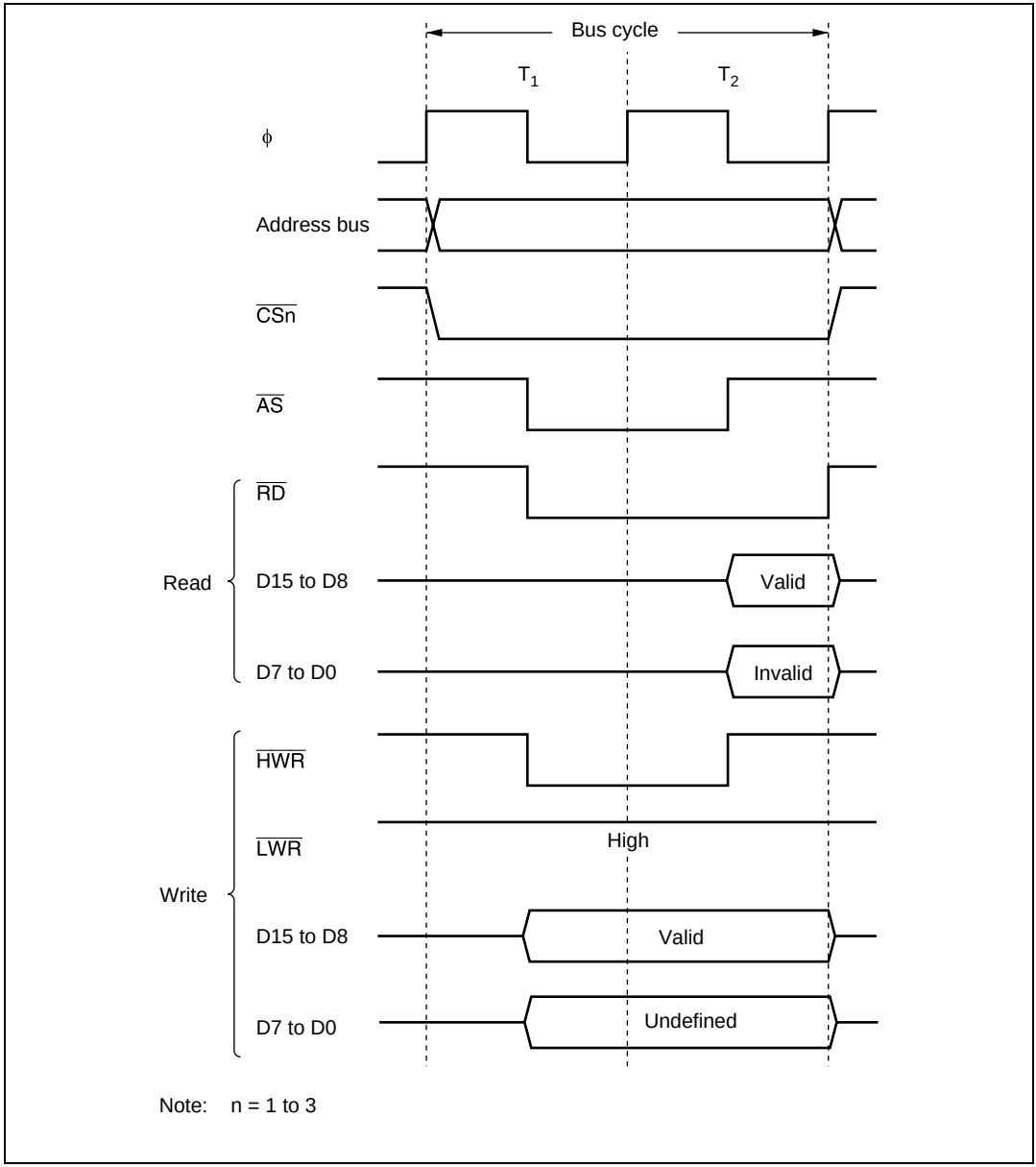


Figure 6.7 Bus Timing for 16-Bit, 2-State Access Space (Even Byte Access)

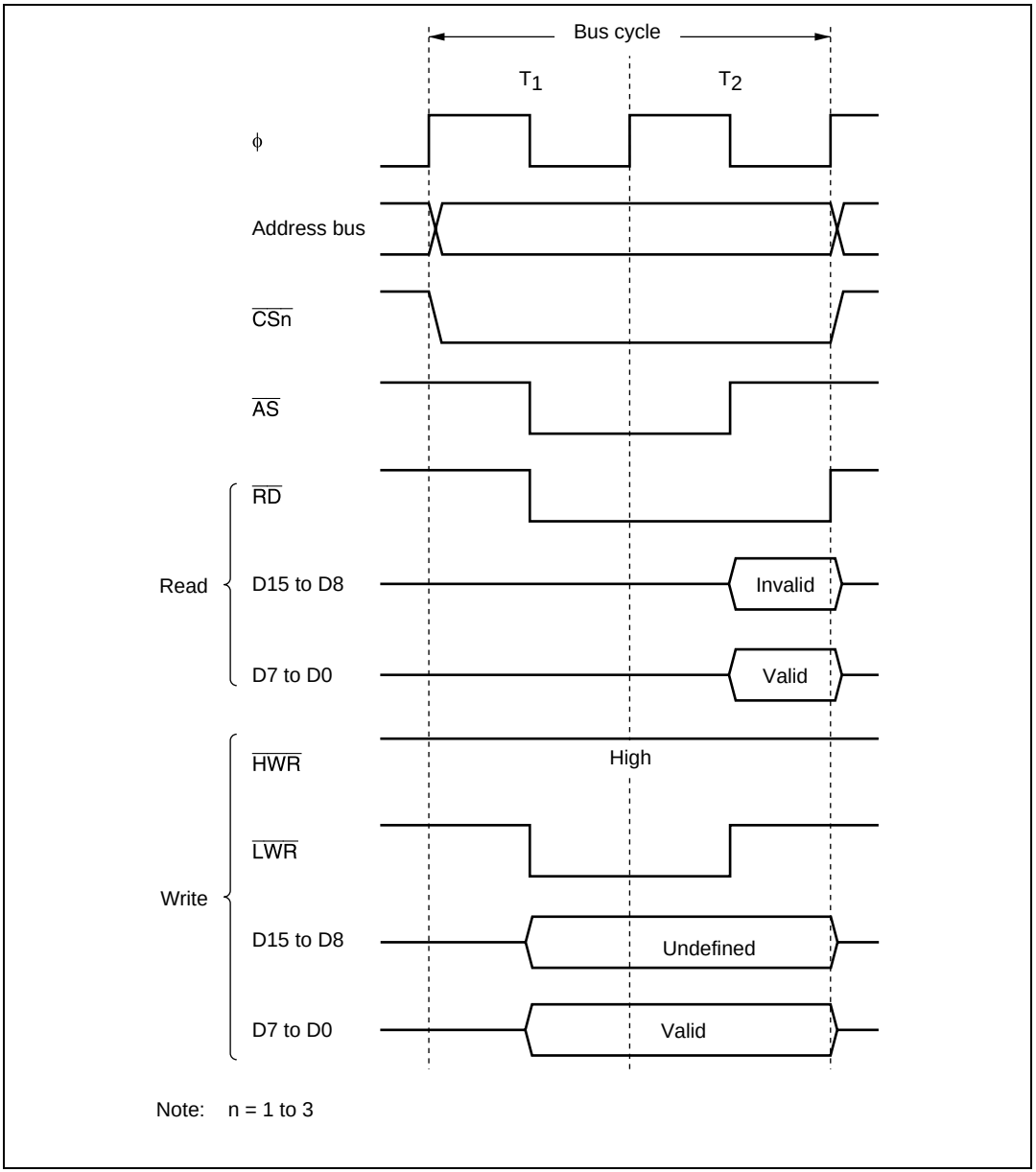


Figure 6.8 Bus Timing for 16-Bit, 2-State Access Space (Odd Byte Access)

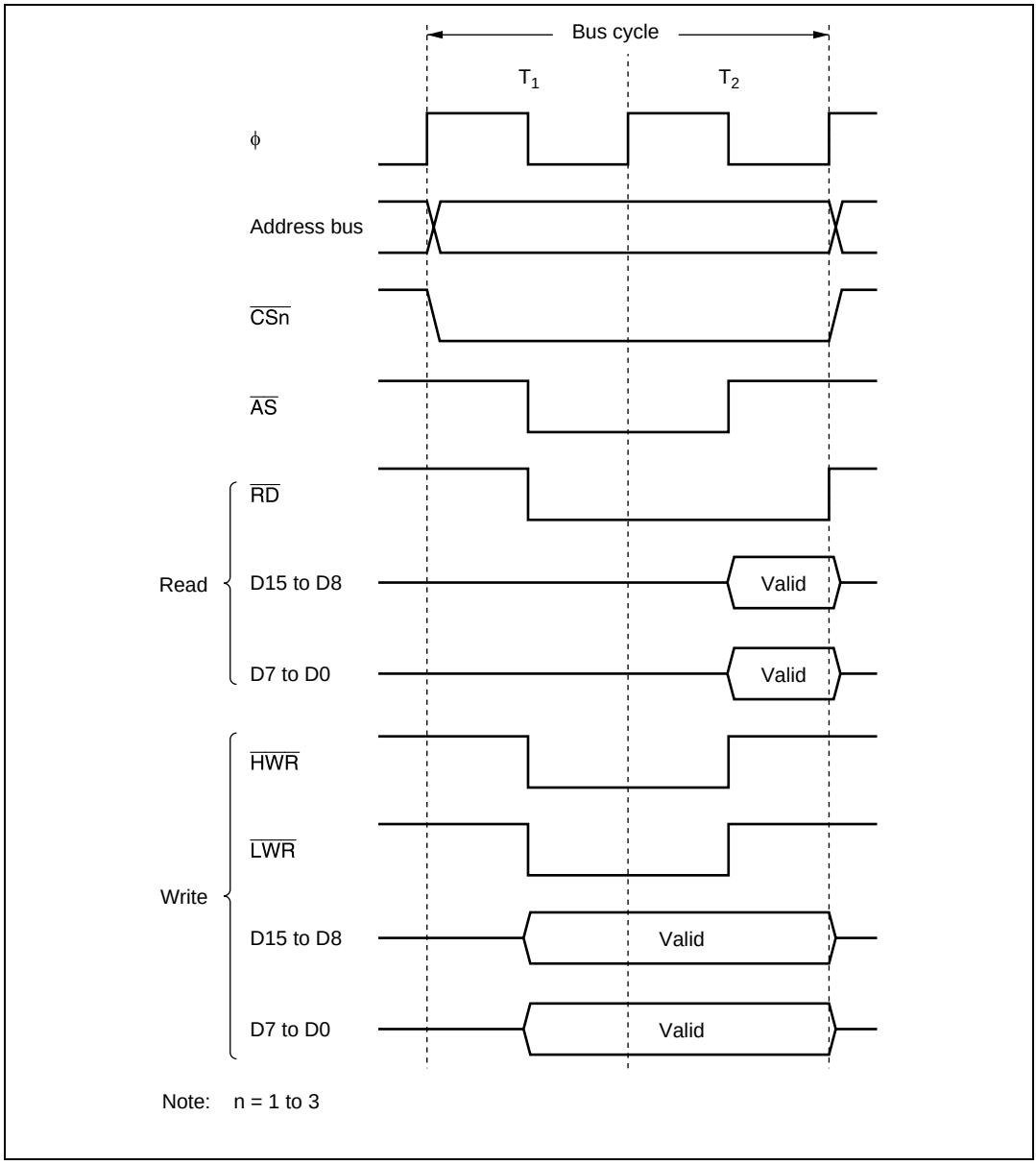


Figure 6.9 Bus Timing for 16-Bit, 2-State Access Space (Word Access)

16-Bit, 3-State Access Space:

Figures 6.10 to 6.12 show bus timings for a 16-bit, 3-state access space. When a 16-bit access space is accessed, the upper half (D15 to D8) of the data bus is used for even addresses, and the lower half (D7 to D0) for odd addresses. Wait states can be inserted.

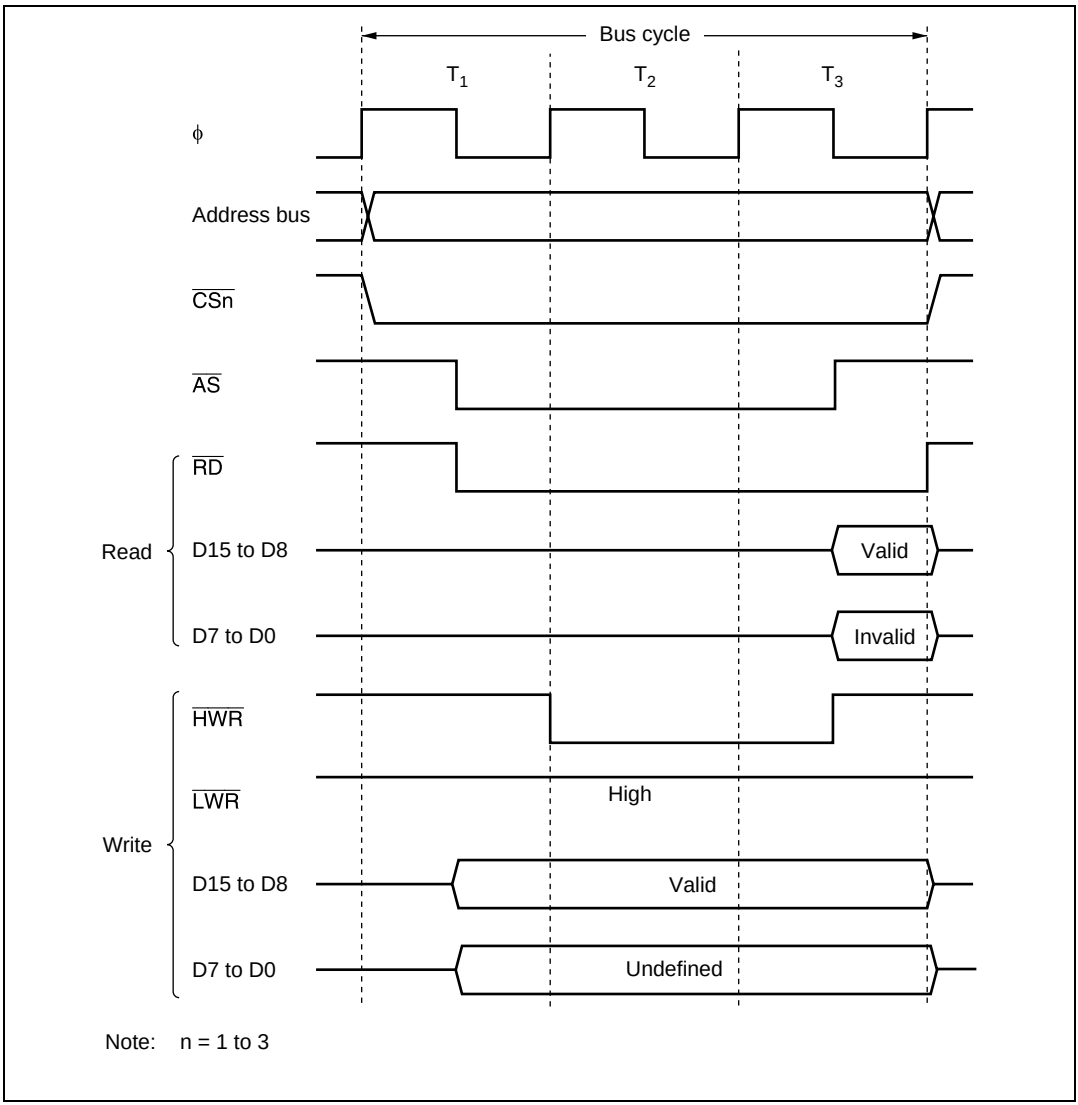


Figure 6.10 Bus Timing for 16-Bit, 3-State Access Space (Even Byte Access)

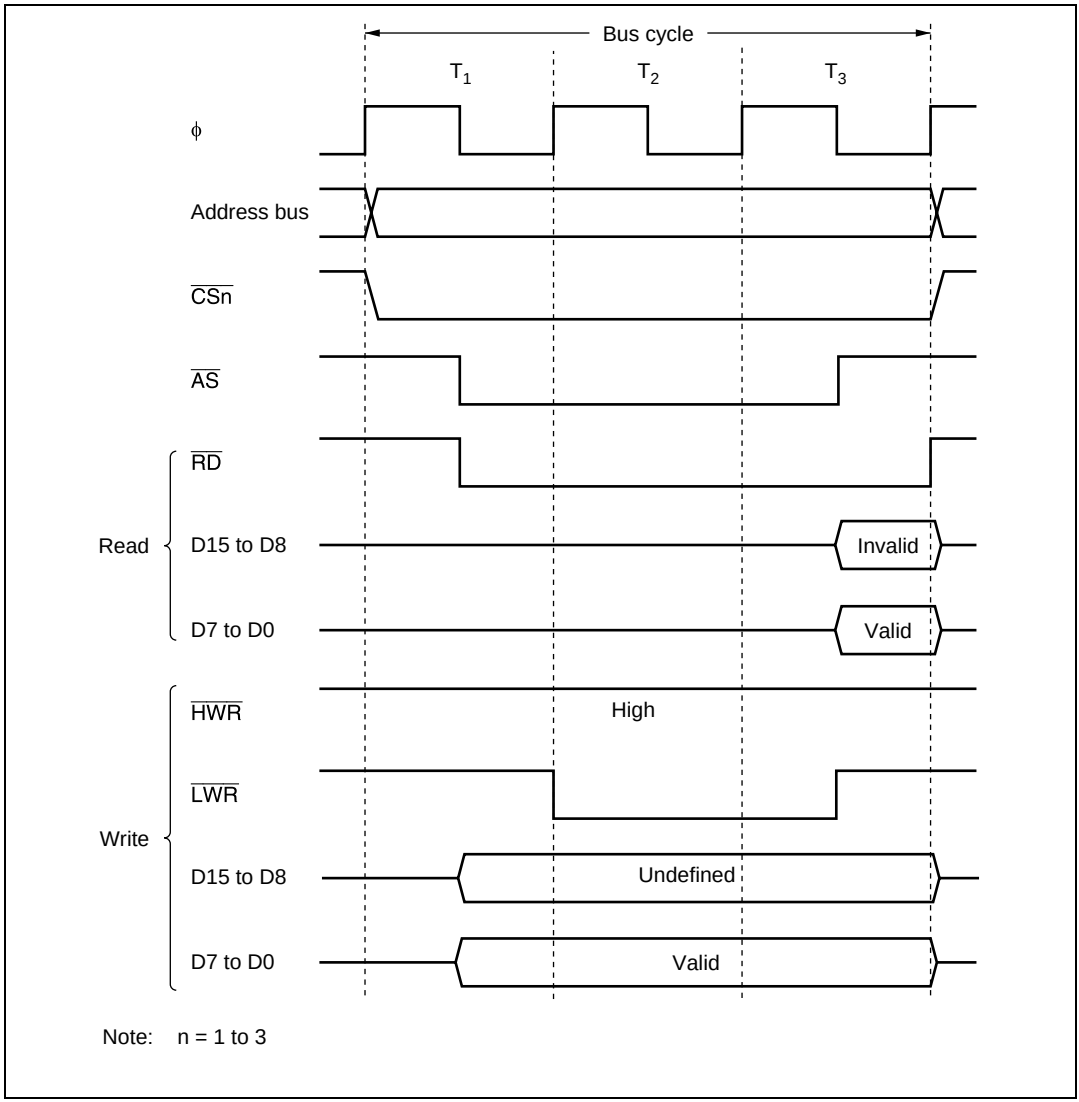


Figure 6.11 Bus Timing for 16-Bit, 3-State Access Space (Odd Byte Access)

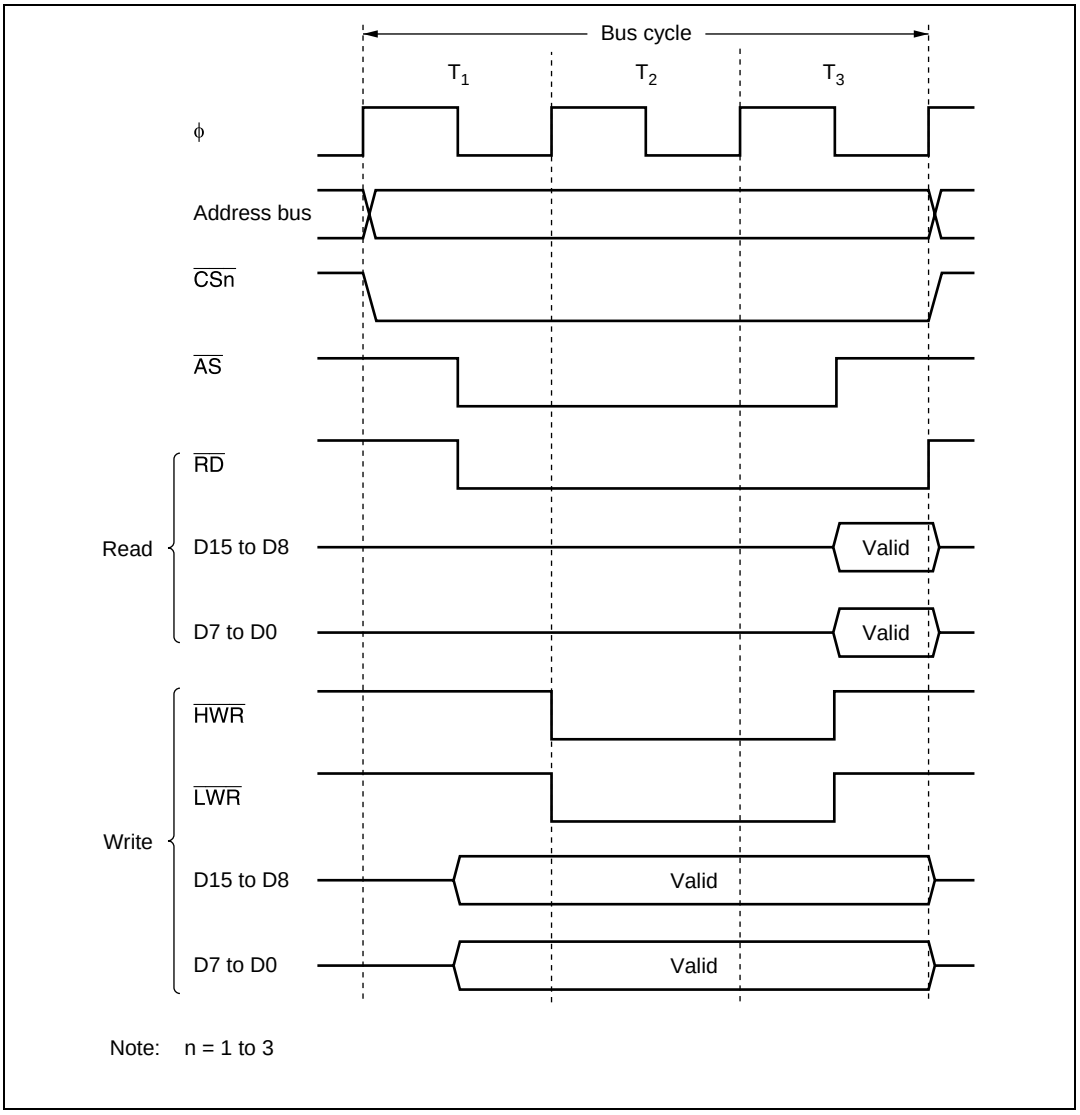


Figure 6.12 Bus Timing for 16-Bit, 3-State Access Space (Word Access)

6.5.4 Basic Operation Timing in Multiplex Extended Mode

8-Bit, 2-State Data Access Space:

Figures 6.13 and 6.14 show the bus timing for an 8-bit, 2-state access space. When an 8-bit access space is accessed, the upper half (AD15 to AD8) of the address bus and the data bus is used. Wait states cannot be inserted.

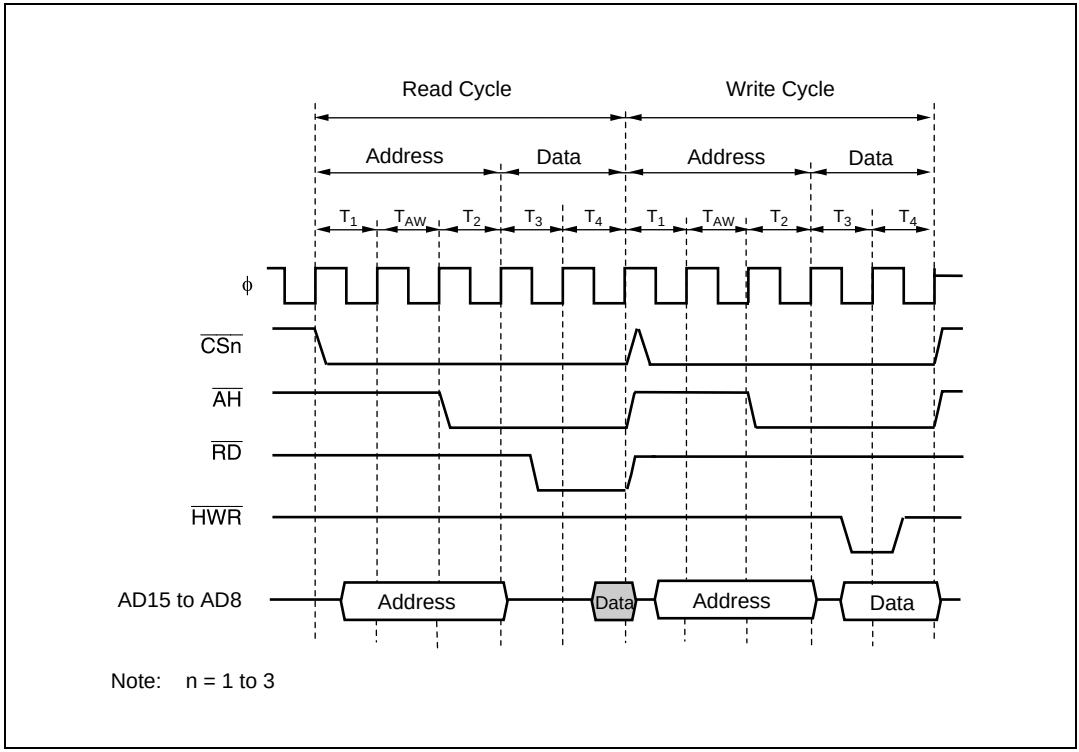


Figure 6.13 Bus Timing for 8-Bit, 2-State Data Access Space (With Address Wait)

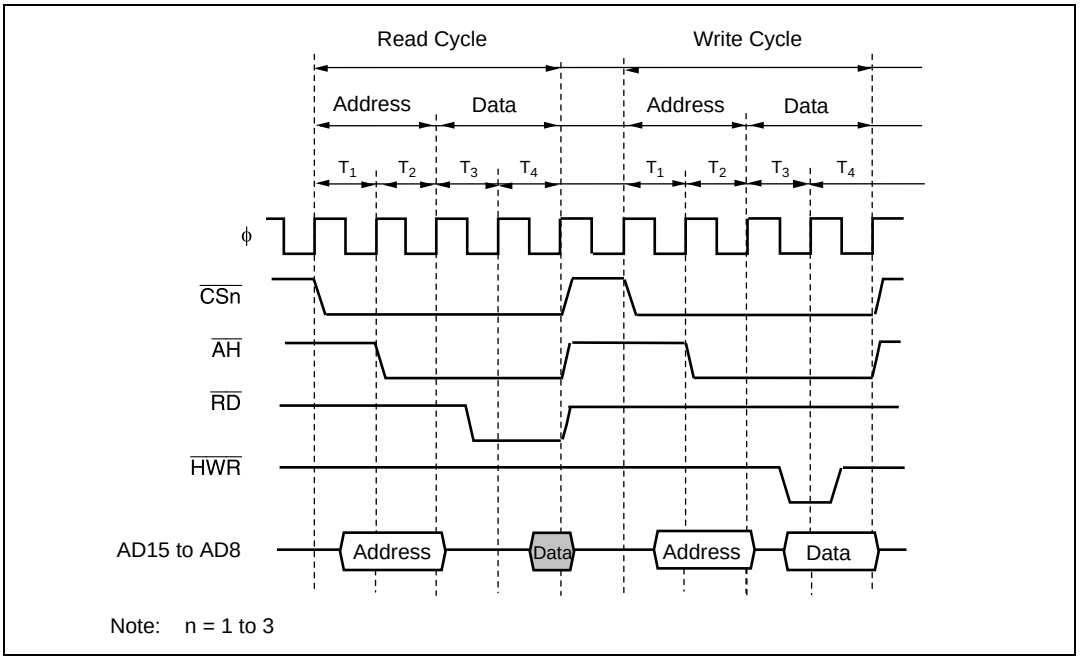


Figure 6.14 Bus Timing for 8-Bit, 2-State Data Access Space (Without Address Wait)

8-Bit, 3-State Data Access Space:

Figure 6.15 shows the bus timing for an 8-bit, 3-state access space. When an 8-bit access space is accessed, the upper half (AD15 to AD8) of the address bus and the data bus are used. Wait states can be inserted.

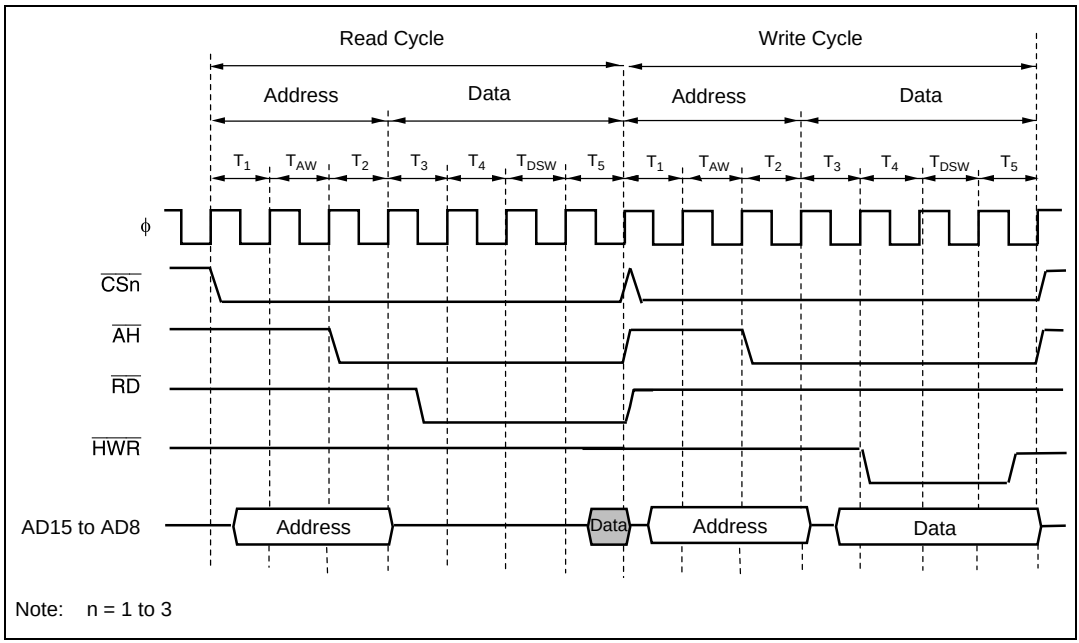


Figure 6.15 Bus Timing for 8-Bit, 3-State Data Access Space (With Address Wait)

16-Bit, 2-State Data Access Space:

Figures 6.16 to 6.21 show bus timings for a 16-bit, 2-state access space. When a 16-bit access space is accessed, the address bus uses all buses (AD15 to AD0), the upper half (AD15 to AD8) of the data bus is used for even addresses, and the lower half (AD7 to AD0) for odd addresses. Data cycle wait states cannot be inserted.

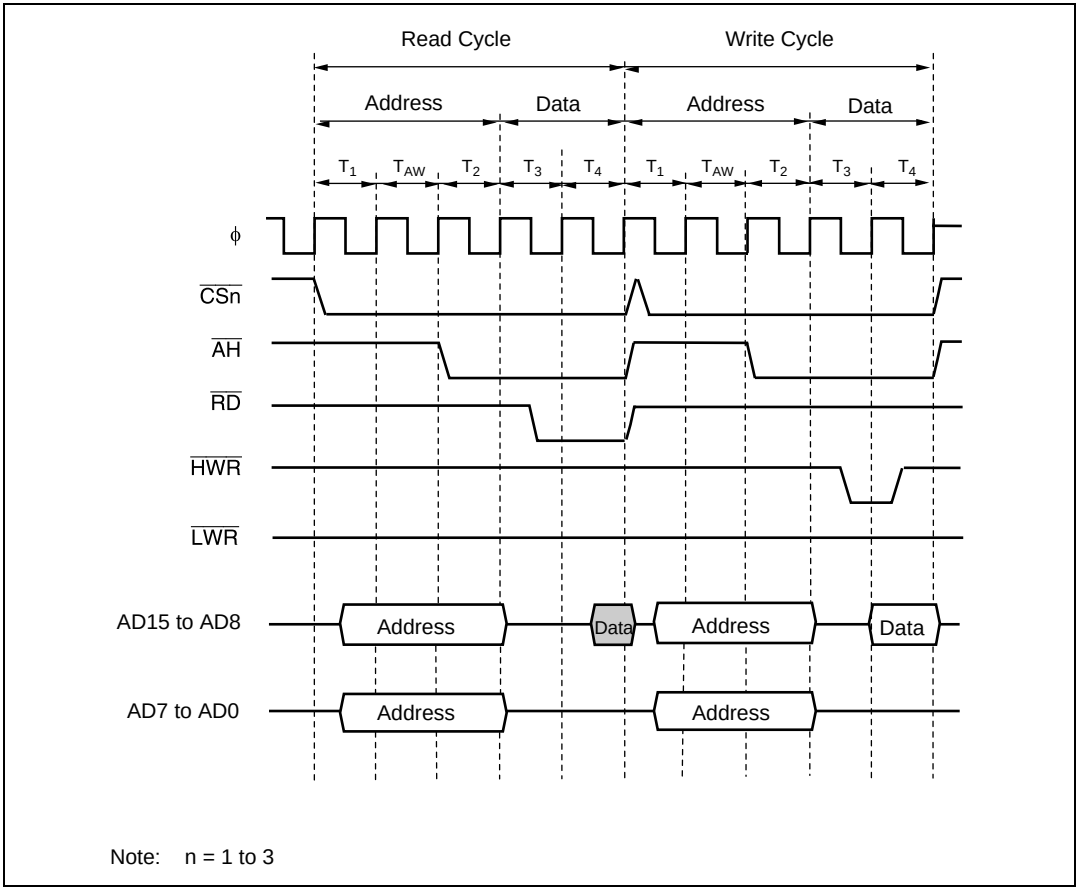


Figure 6.16 Bus Timing for 16-Bit, 2-State Data Access Space (1) (Even Byte Access, with Address Wait)

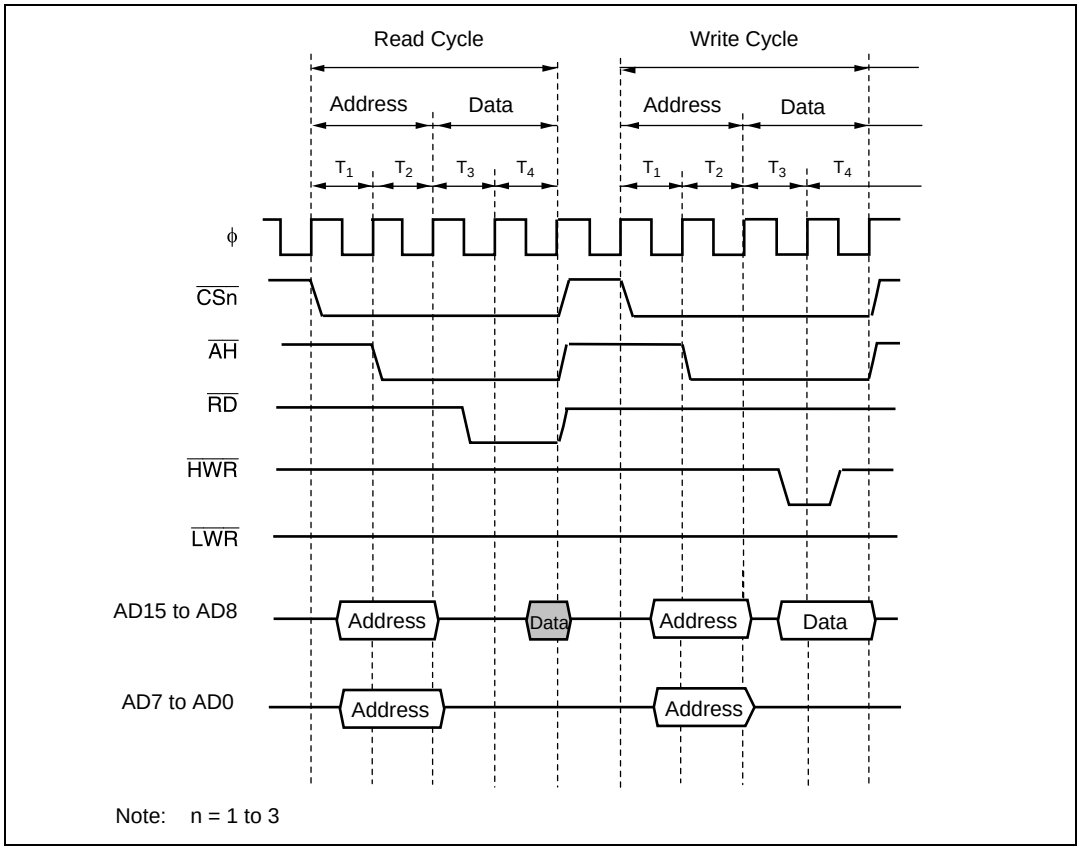


Figure 6.17 Bus Timing for 16-Bit, 2-State Data Access Space (2)
(Even Byte Access, without Address Wait)

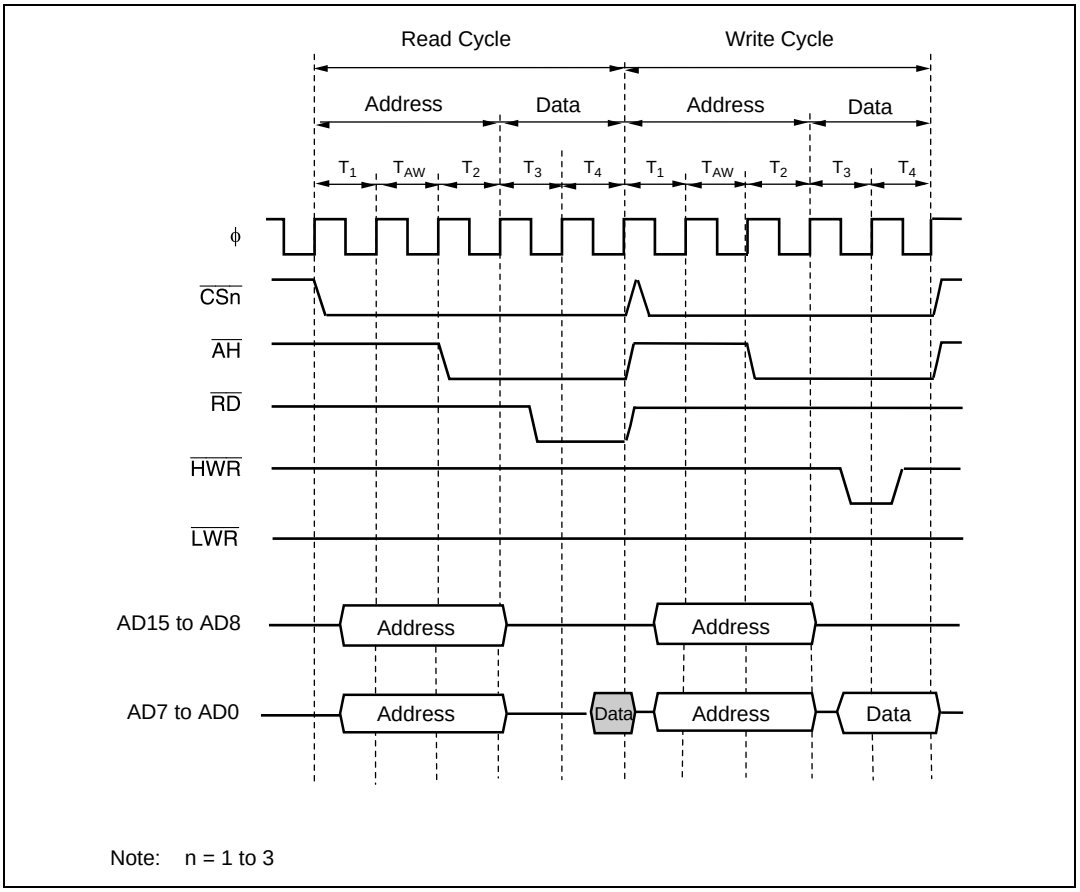


Figure 6.18 Bus Timing for 16-Bit, 2-State Access Space (3)
(Odd Byte Access, with Address Wait)

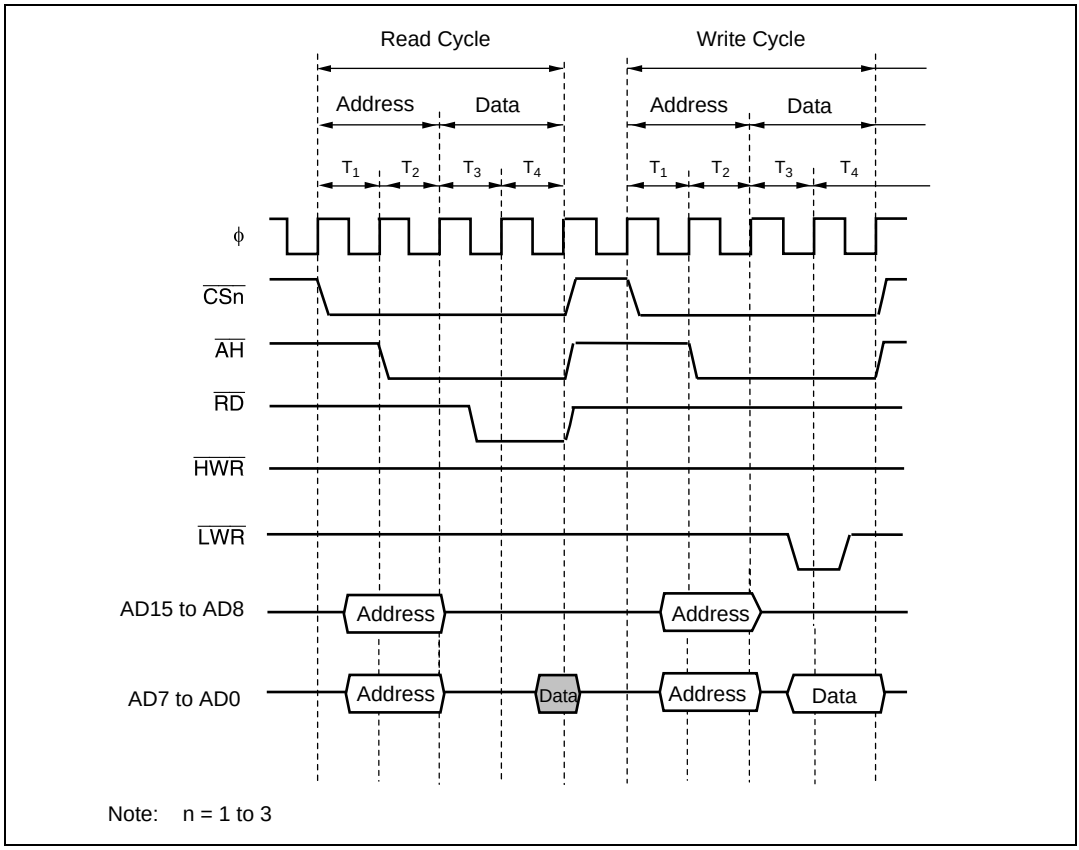


Figure 6.19 Bus Timing for 16-Bit, 2-State Data Access Space (4)
(Odd Byte Access, without Address Wait)

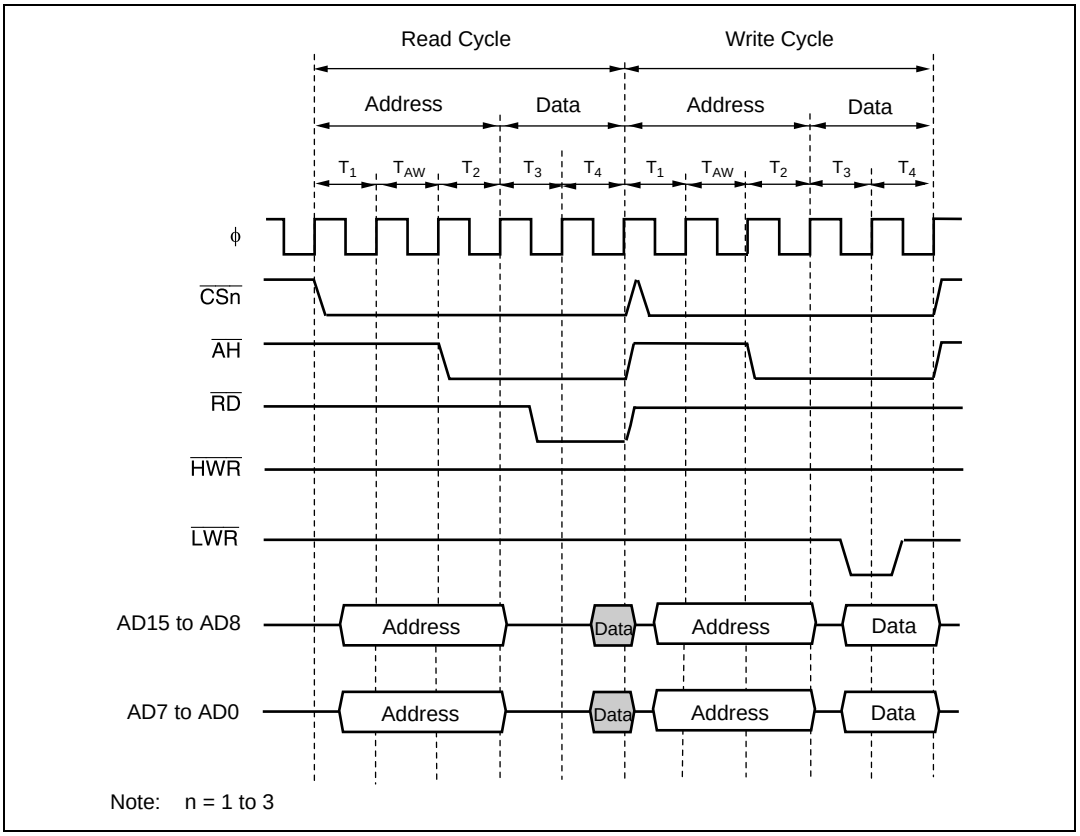


Figure 6.20 Bus Timing for 16-Bit, 2-State Data Access Space (5)
(Word Access, with Address Wait)

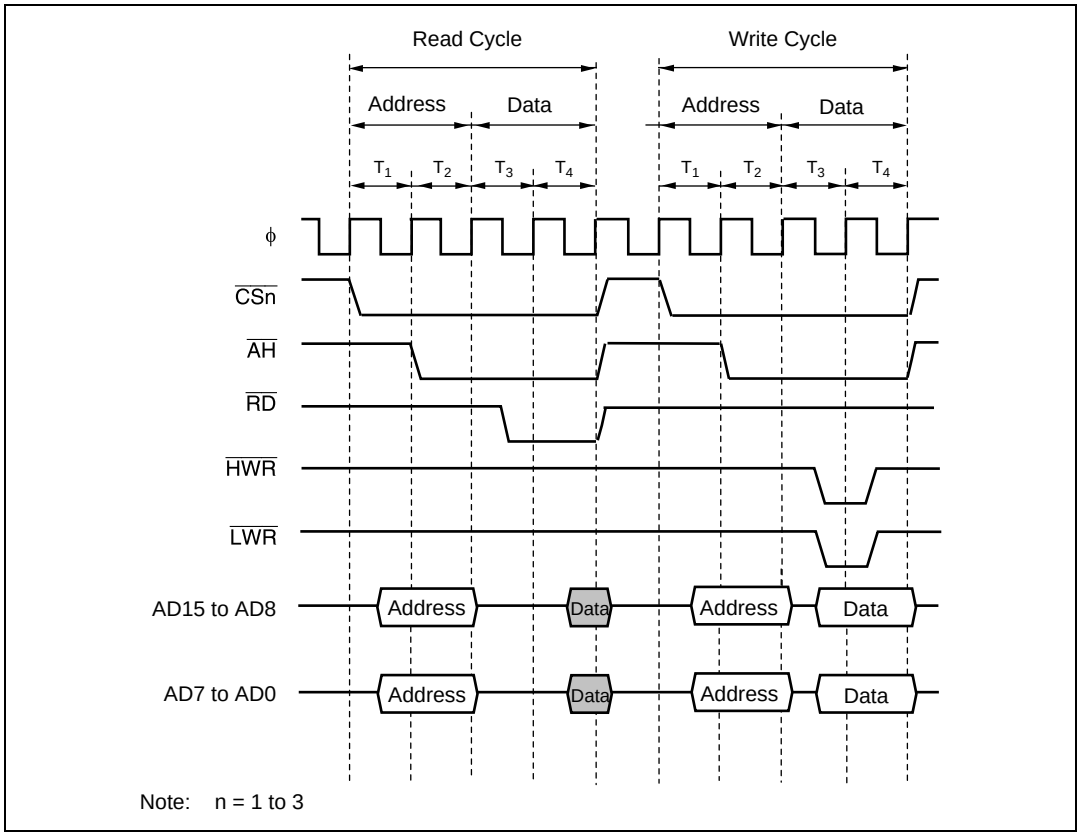


Figure 6.21 Bus Timing for 16-Bit, 2-State Data Access Space (6)
(Word Access, without Address Wait)

16-Bit, 3-State Data Access Space:

Figures 6.22 to 6.24 show bus timings for a 16-bit, 3-state access space. When a 16-bit access space is accessed, the address bus uses all buses (AD15 to AD0), the upper half (AD15 to AD8) of the data bus is used for even addresses, and the lower half (AD7 to AD0) for odd addresses. Data cycle wait states can be inserted.

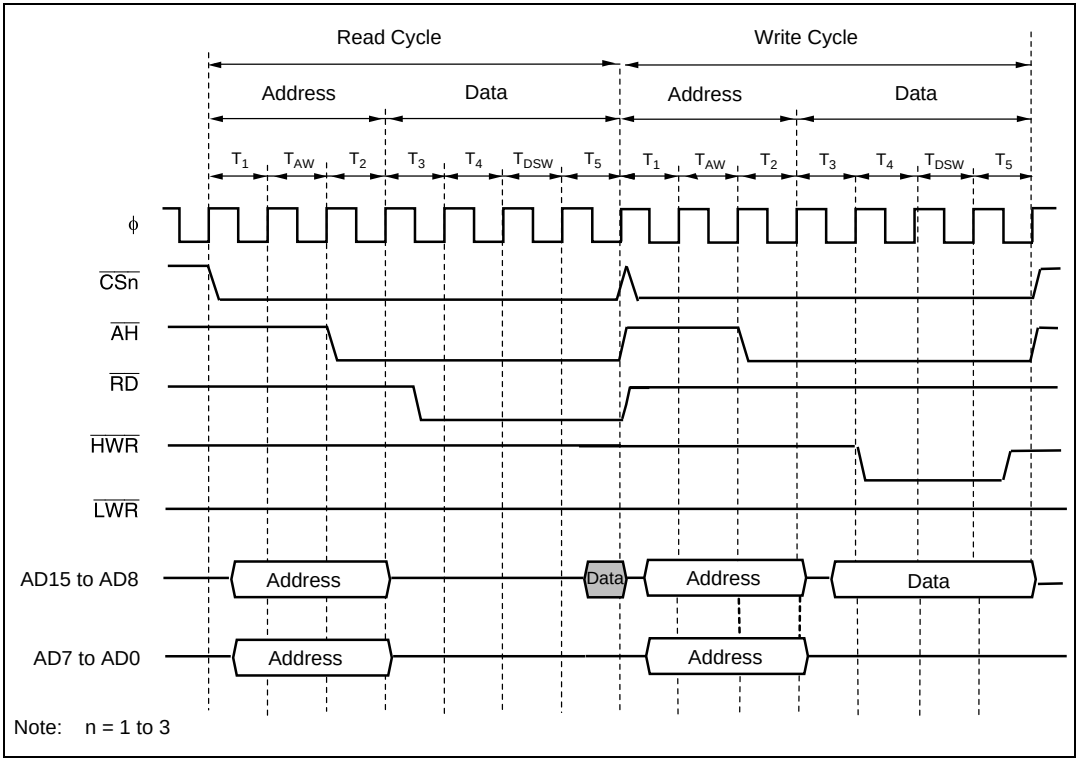


Figure 6.22 Bus Timing for 16-Bit, 3-State Data Access Space (1)
(Even Byte Access, with Address Wait)

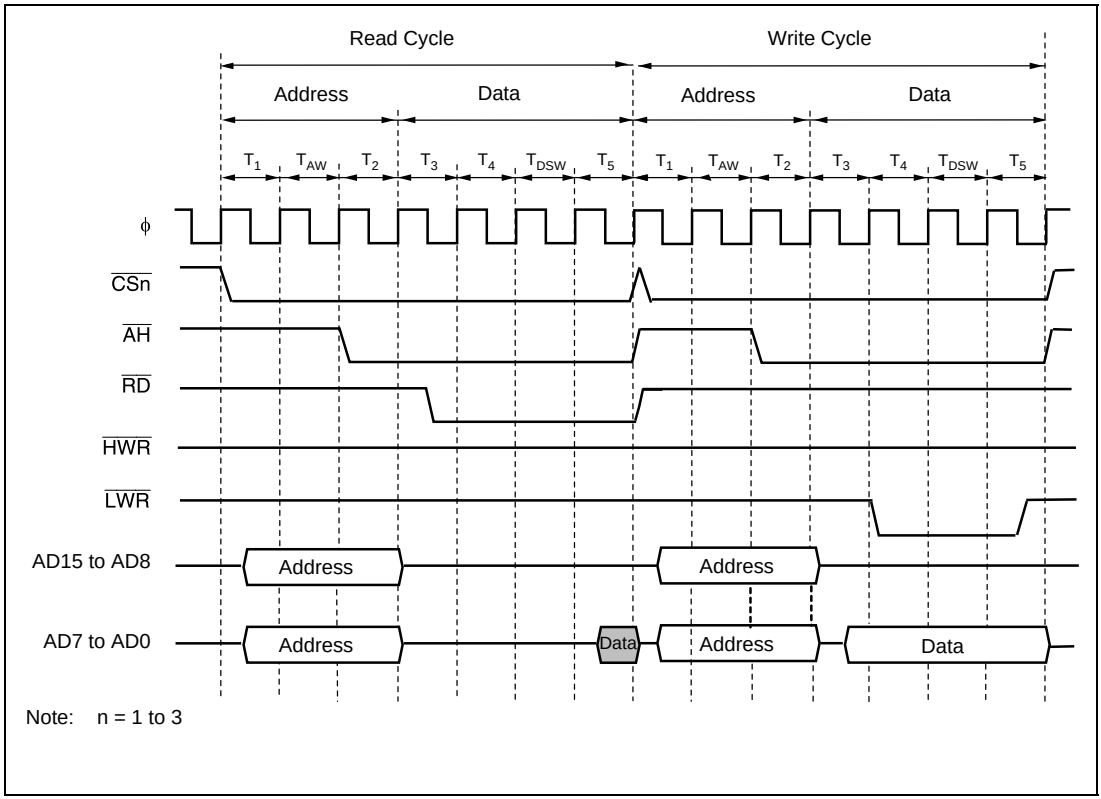


Figure 6.23 Bus Timing for 16-Bit, 3-State Data Access Space (2)
(Odd Byte Access, with Address Wait)

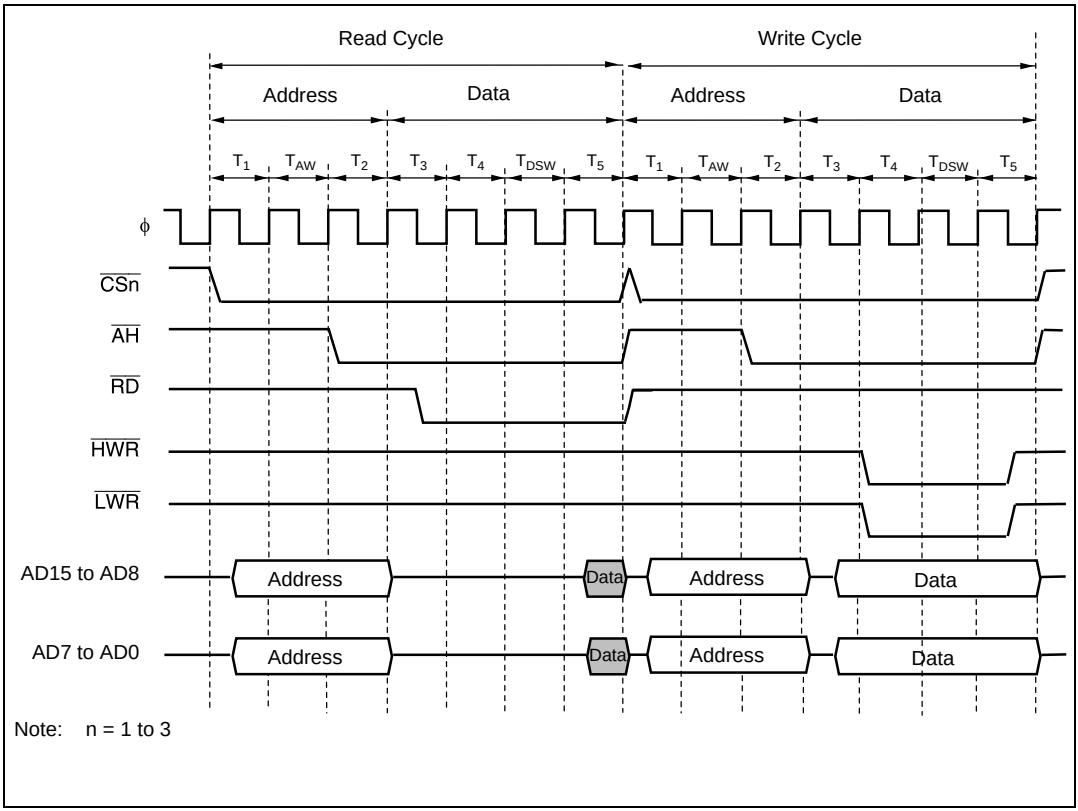


Figure 6.24 Bus Timing for 16-Bit, 3-State Data Access Space (3)
(Word Access, with Address Wait)

6.5.5 Wait Control

When accessing the external address space, this LSI can extend the bus cycle by inserting the wait states (T_W). 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.

In Normal Extended Mode:

1. Program Wait Mode

   When the external address is accessed in the program wait mode, the T_W of the number of states which are set by the WCn1 and WCn0 pins of the BCRAn, is always inserted between the T_2 and the T_3 states.

2. Pin Wait Mode

   A specified number of wait states T_W are always inserted between the T_2 state and T_3 state when accessing the external address space. The number of wait states T_W is specified by the settings of the WCn1 and WCn0 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.

   Pin wait mode 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.

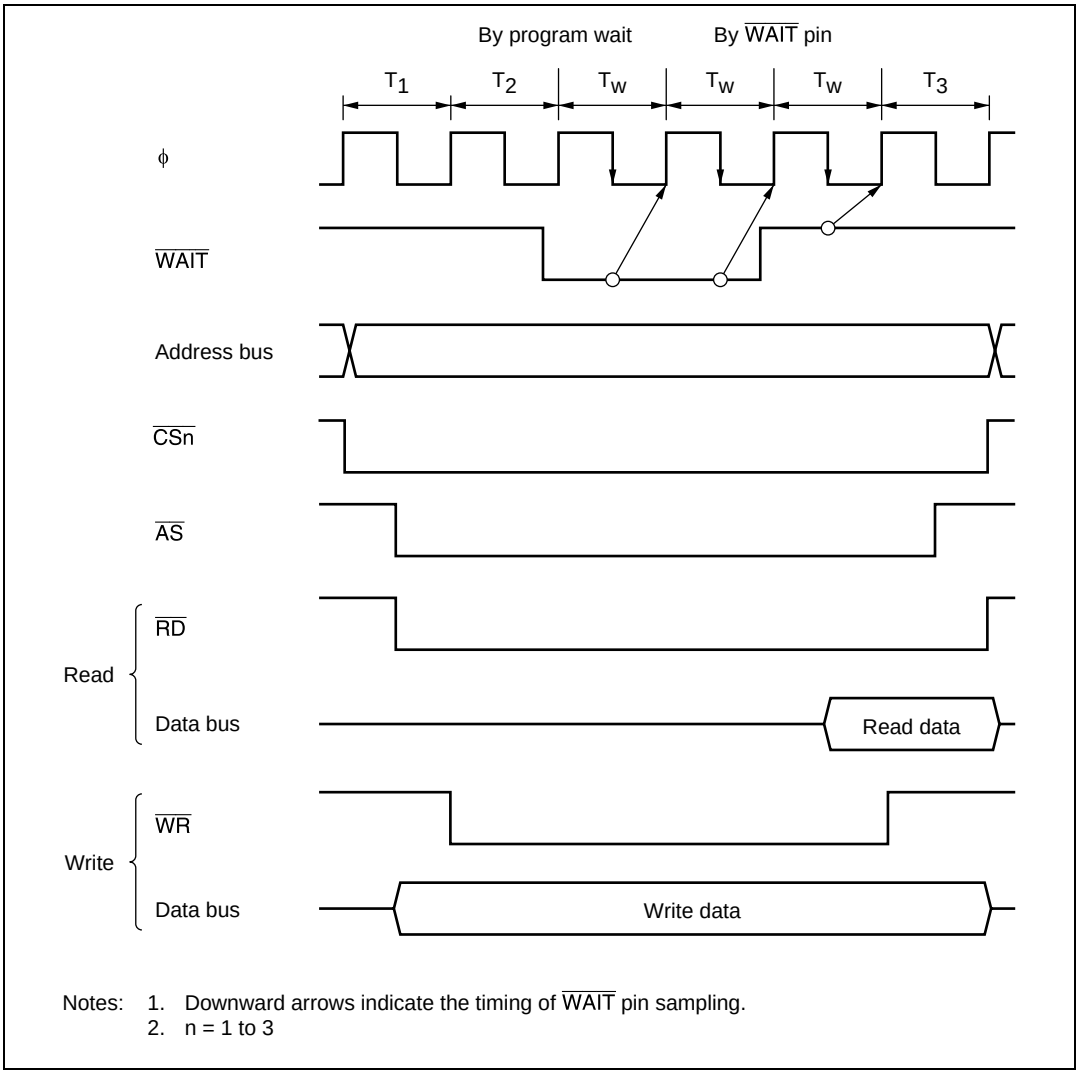
3. Pin Auto-Wait Mode

   A specified number of wait states T_W are inserted between the T_2 state and T_3 state when accessing the external address space if the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_2 state. The number of wait states T_W is specified by the settings of the WCn1 and WCn0 bits. Even if the $\overline{\text{WAIT}}$ pin is held low, T_W states are inserted only up to the specified number of states.

   Pin auto-wait mode enables the low-speed memory interface only by inputting the chip select signal to the $\overline{\text{WAIT}}$ pin.

   Figure 6.25 shows an example of wait state insertion timing in pin wait mode.

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



Notes: 1. Downward arrows indicate the timing of $\overline{WAIT}$ pin sampling.
2. n = 1 to 3

Figure 6.25 Example of Wait State Insertion Timing (Normal Extended Pin Wait Mode)

In Multiplex Extended Mode:

1. Program Wait Mode

   Program wait mode includes address wait and data wait.

   Zero to one state of address wait T_{AW} is inserted between T_1 and T_2 states. Zero to three states of data wait T_{DSW} are inserted between T_4 and T_5 states.

   The address cycle always operates in program wait mode.

2. Pin Wait Mode

   When accessing the external address space, a specified number of wait states T_{DOW} can be inserted between the T_4 state and T_5 state of data state. If the $\overline{WAIT}$ pin is low at the falling edge of ϕ in the last T_4 or T_{DOW} state, another T_{DOW} state is inserted. If the $\overline{WAIT}$ pin is held low, T_{DOW} states are inserted until it goes high.

   Pin wait mode is useful when inserting four or more T_{DOW} states, or when changing the number of T_{DOW} states to be inserted for each external device.

3. Pin Auto-Wait Mode

   A specified number of wait states T_{DOW} are inserted between the T_4 state and T_5 state when accessing the external address space if the $\overline{WAIT}$ pin is low at the falling edge of ϕ in the last T_4 state. The number of wait states T_{DOW} is specified by the settings of the WCn1 and WCn0 bits. Even if the $\overline{WAIT}$ pin is held low, T_{DOW} states are inserted only up to the specified number of states.

   Pin auto-wait mode enables the low-speed memory interface only by inputting the chip select signal to the $\overline{WAIT}$ pin.

   Figure 6.26 shows an example of wait state insertion timing in pin wait mode.

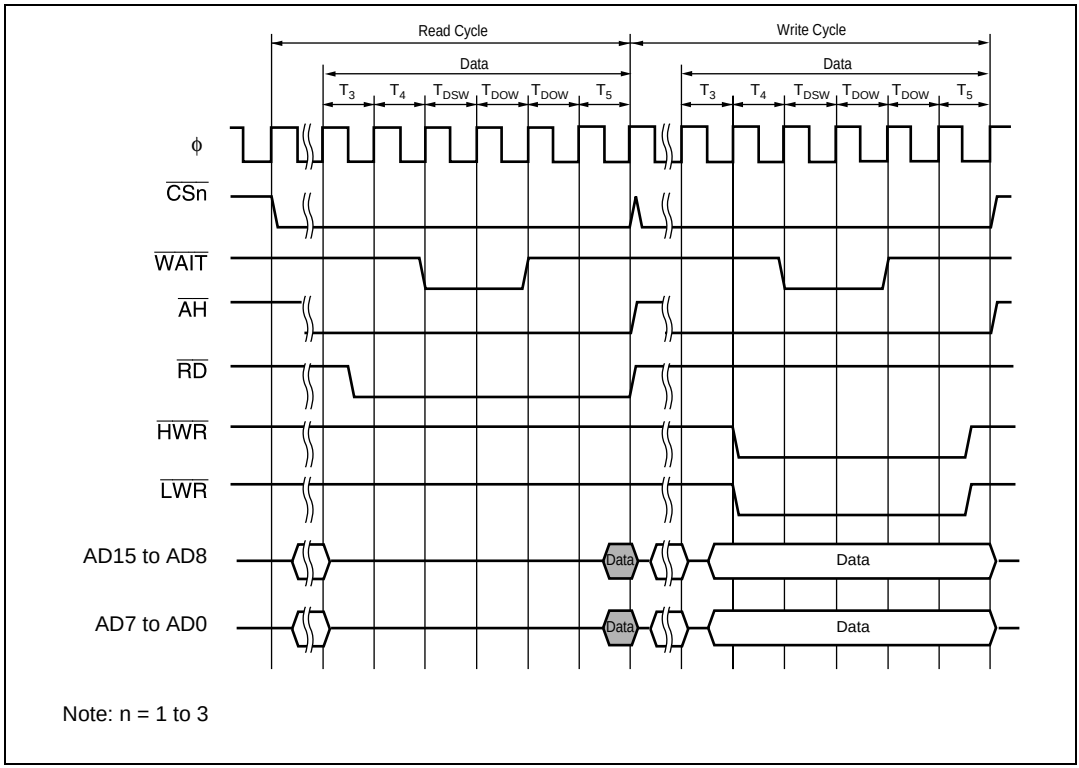


Figure 6.26 Example of Wait State Insertion Timing (Multiplex Extended Mode)

6.6 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.

In the case of normal extended mode if an external write occurs after an external read while the ICIS bit is set to 1 in BCR, an idle cycle is inserted at the start of the write cycle.

In the case of multiplex extended mode if an external cycle occurs after an external read while the ICIS bit is set to 1 in BCR, an idle cycle is inserted at the start of the external cycle after the external read.

Figure 6.27 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 6.27 (a), with no idle cycle inserted, a conflict occurs in bus cycle B between the read data from ROM and the CPU write data. In figure 6.27 (b), an idle cycle is inserted, thus preventing data conflict.

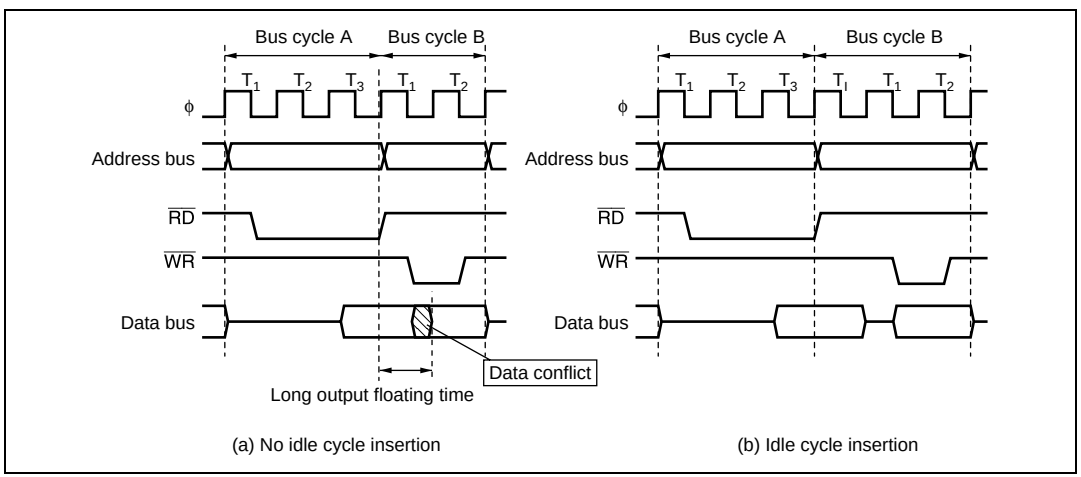


Figure 6.27 Examples of Idle Cycle Operation

Table 6.8 shows the pin states in an idle cycle.

Table 6.8 Pin States in Idle Cycle

Pins	Pin State
A15 to A0	Contents of immediately following bus cycle
D15 to D0	High impedance
AD15 to AD0	High impedance
$\overline{AS}$/$\overline{AH}$	High when PNCASH = 0. Low when PNCASH = 1.
$\overline{CSn}$	High when PNCCSn = 0. Low when PNCCSn = 1.
$\overline{RD}$	High
$\overline{HWR}$, $\overline{LWR}$	High

[Legend]

n = 1 to 3

Section 7 I/O Ports

Table 7.1 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, a data register (DR) that stores output data, and a port register (PORT) that reads a pin state. DDR and DR are not provided for ports 0, 7, and C.

Ports 1 to 3, and 6, have on-chip input pull-up MOSs. DDR and an pull-up MOS control register (PCR) can be used to control the on/off status of the input pull-up MOSs.

Port 6 has an on-chip open-drain control register (ODR) that can be used to control the on/off status of the output buffer PMOS.

Ports 1 to 6, and 9, can drive a single TTL load and 30-pF capacitive load.

All the I/O ports can drive a Darlington transistor in output mode. P80 to P83 and PC0 to PC3 are NMOS push-pull output.

Table 7.1 Port Functions (1)

Port	Description	Extended Mode (EXPE = 1)		Single-Chip Mode (EXPE = 0)	I/O Status
		Normal	Multiplex		
Port 0	General input port also functioning as an A/D converter analog input and interrupt input.	P07/AN15/$\overline{\text{ExIRQ7}}$			
		P06/AN14/$\overline{\text{ExIRQ6}}$			
		P05/AN13/$\overline{\text{ExIRQ5}}$			
		P04/AN12/$\overline{\text{ExIRQ4}}$			
		P03/AN11			
		P02/AN10			
		P01/AN9			
		P00/AN8			
Port 1	General I/O port also functioning as an address output, address/data multiplex input/output, and PWM output.	P17/A7	AD7*[1]	P17/PW7*[2]	On-chip input pull-up MOS
		P16/A6	AD6*[1]	P16/PW6*[2]	
		P15/A5	AD5*[1]	P15/PW5*[2]	
		P14/A4	AD4*[1]	P14/PW4*[2]	
		P13/A3	AD3*[1]	P13/PW3*[2]	
		P12/A2	AD2*[1]	P12/PW2*[2]	
		P11/A1	AD1*[1]	P11/PW1*[2]	
		P10/A0	AD0*[1]	P10/PW0*[2]	
Port 2	General I/O port also functioning as an address output, address/data multiplex input/output, and TPU input/output.	P27/A15	AD15	P27/TIOCB2/TCLKD	On-chip input pull-up MOS
		P26/A14	AD14	P26/TIOCA2	
		P25/A13	AD13	P25/TIOCB1/TCLKC	
		P24/A12	AD12	P24/TIOCA1	
		P23/A11	AD11	P23/TIOCD0/TCLKB	
		P22/A10	AD10	P22/TIOCC0/TCLKA	
		P21/A9	AD9	P21/TIOCB0	
		P20/A8	AD8	P20/TIOCA0	

Table 7.1 Port Functions (2)

Port	Description	Extended Mode(EXPE = 1)		Single-Chip Mode (EXPE = 0)	I/O Status
		Normal	Multiplex		
Port 3	General I/O port also functioning as a bidirectional data bus and interrupt input.	D15	P37		On-chip input pull-up MOS
		D14	P36		
		D13	P35		
		D12	P34		
		D11	P33/$\overline{\text{ExIRQ3}}$		
		D10	P32/$\overline{\text{ExIRQ2}}$		
		D9	P31/$\overline{\text{ExIRQ1}}$		
		D8	P30/$\overline{\text{ExIRQ0}}$		
Port 4	General I/O port also functioning as an interrupt input, TMR0_0, TMR0_1, TMRX_0, TMRX_1, TMRY_0, TMRY_1, and FRT_1 input, and PWM output.	P47/$\overline{\text{IRQ7}}$/TMIY_0/ExPW3			
		P46/$\overline{\text{IRQ6}}$/TMIX_0/ExPW2			
		P45/$\overline{\text{IRQ5}}$/TMI0_0/ExPW1			
		P44/$\overline{\text{IRQ4}}$/TMIY_1/ExPW0			
		P43/$\overline{\text{IRQ3}}$/TMIX_1			
		P42/$\overline{\text{IRQ2}}$/TMI0_1			
		P41/$\overline{\text{IRQ1}}$/FTIC_1			
		P40/$\overline{\text{IRQ0}}$/FTIB_1			
Port 5	General I/O port also functioning as a TMR0_1 and TMR1_1 output, SCI_0 and SCI_1 input/output, and PWMX output.	P57/TMO1_1/ExPW5			
		P56/TMO0_1/ExPW4			
		P55/RxD1			
		P54/TxD1			
		P53/SCK1			
		P52/RxD0			
		P51/TxD0			
		P50/SCK0			

Table 7.1 Port Functions (3)

Port	Description	Extended Mode (EXPE = 1) Normal	Extended Mode (EXPE = 1) Multiplex	Single-Chip Mode (EXPE = 0)	I/O Status
Port 6	General I/O port also functioning as a bidirectional data bus, FRT_1 input/output, TMRX_1 and TMRY_1 output, and SCI_2 input/output.	D7*[3]	P67/RxD2*[4]		On-chip input pull-up MOS Open-drain output enabled
		D6*[3]	P66/TxD2*[4]		
		D5*[3]	P65/SCK2*[4]		
		D4*[3]	P64/FTCI_1*[4]		
		D3*[3]	P63/TMOY_1*[4]		
		D2*[3]	P62/TMOX_1*[4]		
		D1*[3]	P61/FTOB_1*[4]		
		D0*[3]	P60/FTOA_1*[4]		
Port 7	General input port also functioning as an A/D converter analog input.	P77/AN7			
		P76/AN6			
		P75/AN5			
		P74/AN4			
		P73/AN3			
		P72/AN2			
		P71/AN1			
		P70/AN0			

Table 7.1 Port Functions (4)

Port	Description	Extended Mode (EXPE = 1) Normal	Extended Mode (EXPE = 1) Multiplex	Single-Chip Mode (EXPE = 0)	I/O Status
Port 8	General I/O port also functioning as an A/D converter external trigger input, PWMX output, SCI_3, SCI_4, IIC3_0, and IIC3_1 input/output, and TPU input/output.	P87/$\overline{\text{ADTRG}}$/ExTIOCB0			NMOS push-pull output (P80 to P83)
		P86/ExTIOCA0			
		P85/PWX1			
		P84/PWX0			
		P83/SDA1/RxD4			
		P82/SCL1/TxD4			
		P81/SDA0/RxD3			
		P80/SCL0/TxD3			
Port 9	General I/O port also functioning as a bus control input/output, system clock output, and TPU input/output.	P97/$\overline{\text{WAIT}}$/ExTIOCD0/ ExTCLKB		P97/ExTIOCD0/ ExTCLKB	
		P96/ϕ		P96	
		P95/$\overline{\text{AS}}$	$\overline{\text{AH}}$	P95	
		$\overline{\text{HWR}}$		P94	
		$\overline{\text{RD}}$		P93	
		P92/$\overline{\text{CS1}}$/ExTIOCB2/ ExTCLKD		P92/ExTIOCB2/ ExTCLKD	
		P91/$\overline{\text{CS2}}$/ExTIOCA2		P91/ExTIOCA2	
		P90/$\overline{\text{LWR}}$/ExTIOCB1/ ExTCLKC		P90/ExTIOCB1/ ExTCLKC	

Table 7.1 Port Functions (5)

Port	Description	Extended Mode (EXPE = 1) Normal	Extended Mode (EXPE = 1) Multiplex	Single-Chip Mode (EXPE = 0)	I/O Status
Port A	General I/O port also functioning as a bus control output, FRT_0 input/output, TMX_0, TMY_0, and TM0_0 output, timer connection input/output, SCI_3 and SCI_4 input/output, PWM output, and TPU input/output.	PA7/$\overline{CS3}$/ExTIOCA1		PA7/ExTIOCA1	
		PA6/FTCI_0/HFBACKI			
		PA5/FTIB_0/VFBACKI			
		PA4/FTIC_0/CLAMPO			
		PA3/FTOB_0/CBLANK			
		PA2/TMO0_0/ExTIOCC0/ExTCLKA			
		PA1/TMOY_0/ExPW7/SCK4			
		PA0/TMOX_0/ExPW6/SCK3			
Port B	General I/O port also functioning as an FRT_0 and TMR1_0 input/output, FRT_1 and TMR1_1 input, and timer connection input/output.	PB7/TMI1_0/HSYNCI_0			
		PB6/FTIA_0/VSYNCI_0			
		PB5/FTID_0/CSYNCI_0			
		PB4/TMI1_1/HSYNCI_1			
		PB3/FTIA_1/VSYNCI_1			
		PB2/FTID_1/CSYNCI_1			
		PB1/TMO1_0/HSYNCO			
		PB0/FTOA_0/VSYNCO			
Port C	General I/O port also functioning as an on-chip emulator input/output, and IIC3_2 and IIC3_3 input/output.	PC7*[5]/ETDO			NMOS push-pull output (PC0 to PC3)
		PC6*[5]/ETDI			
		PC5*[5]/ETCK			
		PC4*[5]/ETMS			
		PC3/SDA3			
		PC2/SCL3			
		PC1/SDA2			
		PC0/SCL2			

Notes: 1. When multiplex extended 16-bit bus is specified.
2. When single-chip mode and multiplex extended 8-bit bus are specified.
3. When normal extended 16-bit bus is specified.
4. When normal extended 8-bit bus, multiplex extended, and single-chip mode are specified.
5. Not supported by the on-chip emulator.

7.1 Port 0

Port 0 is an 8-bit input port. Port 0 pins also function as A/D converter analog input pins and $\overline{\text{EXIRQ}}$ input pins.

Port 0 has the following register.

- Port 0 register (PORT0)

7.1.1 Port 0 Register (PORT0)

PORT0 is an 8-bit read-only register, that reflects the pin state in port 0. PORT0 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P07	—*	R	When this register is read, the pin state is always read.
6	P06	—*	R	
5	P05	—*	R	
4	P04	—*	R	
3	P03	—*	R	
2	P02	—*	R	
1	P01	—*	R	
0	P00	—*	R	

Note: * Determined by the states of the P07 to P00 pins.

7.1.2 Pin Functions

If the corresponding bit in PTCNT1 is set to 1, the port 0 pins can be used as interrupt input pins ($\overline{\text{ExIRQ7}}$ to $\overline{\text{ExIRQ4}}$).

Pin function relationships are listed below.

Note: When these pins are set as the interrupt input pins, do not use them as the A/D converter pins.

- P07/AN15/$\overline{\text{ExIRQ7}}$

Pin function	P07 input pin
	AN15 input pin/$\overline{\text{ExIRQ7}}$ input pin*

Note: * When the IRQ7S bit in PTCNT1 is set to 1, it functions as the $\overline{\text{ExIRQ7}}$ input.

- P06/AN14/$\overline{\text{ExIRQ6}}$

Pin function	P06 input pin
	AN14 input pin/$\overline{\text{ExIRQ6}}$ input pin*

Note: * When the IRQ6S bit in PTCNT1 is set to 1, it functions as the $\overline{\text{ExIRQ6}}$ input.

- P05/AN13/$\overline{\text{ExIRQ5}}$

Pin function	P05 input pin
	AN13 input pin/$\overline{\text{ExIRQ5}}$ input pin*

Note: * When the IRQ5S bit in PTCNT1 is set to 1, it functions as the $\overline{\text{ExIRQ5}}$ input.

- P04/AN12/$\overline{\text{ExIRQ4}}$

Pin function	P04 input pin
	AN12 input pin/$\overline{\text{ExIRQ4}}$ input pin*

Note: * When the IRQ4S bit in PTCNT1 is set to 1, it functions as the $\overline{\text{ExIRQ4}}$ input.

- P03/AN11

Pin function	P03 input pin
	AN11 input pin

- P02/AN10

Pin function	P02 input pin
	AN10 input pin

- P01/AN9

Pin function	P01 input pin
	AN9 input pin

- P00/AN8

Pin function	P00 input pin
	AN8 input pin

7.2 Port 1

Port 1 is an 8-bit I/O port. Port 1 pins also function as address bus pins, address/data multiplex bus pins, and PWM output pins. Pin functions change according to the operating mode.

Port 1 has the following registers.

- Port 1 data direction register (P1DDR)
- Port 1 data register (P1DR)
- Port 1 register (PORT1)
- Port 1 pull-up MOS control register (P1PCR)

7.2.1 Port 1 Data Direction Register (P1DDR)

The individual bits in P1DDR specify input or output for the pins of port 1. The read value is undefined.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	• In normal extended mode The corresponding port 1 pins are address outputs when P1DDR bits are set to 1, and input ports when cleared to 0. • In multiplex extended mode (16-bit bus width) Operation is not affected • In multiplex extended mode (8-bit bus width) Operates as single-chip mode • In single-chip mode 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.
6	P16DDR	0	W	
5	P15DDR	0	W	
4	P14DDR	0	W	
3	P13DDR	0	W	
2	P12DDR	0	W	
1	P11DDR	0	W	
0	P10DDR	0	W	

7.2.2 Port 1 Data Register (P1DR)

P1DR stores output data for port 1.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DR	0	R/W	P1DR stores output data for the port 1 pins that are used as the general output ports.
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	

7.2.3 Port 1 Register (PORT1)

PORT1 reflects the pin state in port 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P17	—*	R	When this register is read, the bit that is set in P1DDR is read as the value of P1DR. The bit that is cleared in P1DDR is read as the pin state.
6	P16	—*	R	
5	P15	—*	R	
4	P14	—*	R	
3	P13	—*	R	
2	P12	—*	R	
1	P11	—*	R	
0	P10	—*	R	

Note: * Determined by the states of the P17 to P10 pins.

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

P1PCR controls the on or off state of input pull-up MOSs for port 1.

Bit	Bit Name	Initial Value	R/W	Description
7	P17PCR	0	R/W	• In normal extended and single-chip modes When the pins are in the input states, the corresponding input pull-up MOS is turned on when a P1PCR bit is set to 1. • In multiplex extended mode When using 16-bit bus width, operation is not affected. When using 8-bit bus width, operates as single-chip mode.
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	

7.2.5 Pin Functions

When the PWnS bit in PTCNT0 is cleared to 0, port 1 pins can be used as PWM output pins (PW7 to PW0). The pin function relationships are listed below.

According to the combinations of operating modes and the OEn bit, PWnS bit, and P1nDDR bit in PWOER of the PWM, pin functions change as follows.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)				
Bus width*	—		16-bit	8-bit			
P1nDDR	0	1	—	0	1		
PWnS	—	—	—	—	1	0	
OEn	—	—	—	—	—	0	1
Pin function	P1n input pin	A7 to A0 output pin	AD7 to AD0 I/O pin	P1n input pin	P1n output pin	P1n output pin	PWn output pin

[Legend]

n = 7 to 0

Note: * When the ABW3 to ABW1 bits in BCRA3 to BCRA1 are all set to 1, bus width is 8 bits, if any are cleared to 0, bus width is 16 bits.

- Single-Chip Mode (EXPE = 0)

P1nDDR	0	1		
PWnS	—	1	0	
OEn	—	—	0	1
Pin function	P1n input pin	P1n output pin	P1n output pin	PWn output pin

[Legend]
n = 7 to 0

7.2.6 Port 1 Input Pull-Up MOS States

Port 1 has an on-chip input pull-up MOS that can be controlled by software. This input pull-up MOS can be used regardless of the operating mode. Table 7.2 summarizes the input pull-up MOS states.

Table 7.2 Port 1 Input Pull-Up MOS States

Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
Off	Off	On/Off	On/Off

[Legend]
Off: Always off.
On/Off : On when P1DDR = 0 and P1PCR = 1; otherwise off.

7.3 Port 2

Port 2 is an 8-bit I/O port. Port 2 pins also function as address bus pins, address/data multiplex bus pins, and TPU I/O pins. Pin functions change according to the operating mode.

Port 2 has the following registers.

- Port 2 data direction register (P2DDR)
- Port 2 data register (P2DR)
- Port 2 register (PORT2)
- Port 2 pull-up MOS control register (P2PCR)

7.3.1 Port 2 Data Direction Register (P2DDR)

The individual bits in P2DDR specify input or output for the pins of port 2. The read value is undefined.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DDR	0	W	• In normal extended mode The corresponding port 2 pin is an address output when a P2DDR bit is set to 1, and an input port when cleared to 0. • In multiplex extended mode Operation is not affected. • In single-chip mode While a general I/O port function is selected, the corresponding port 2 pin is an output port when a P2DDR bit is set to 1, and an input port 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	

7.3.2 Port 2 Data Register (P2DR)

P2DR stores output data for the port 2 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DR	0	R/W	P2DR stores output data for the port 2 pins that are used as the general output ports.
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	

7.3.3 Port 2 Register (PORT2)

PORT2 reflects the pin state in port 2 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P27	—*	R	When this register is read, the bit that is set in P2DDR is read as the value of P2DR. The bit that is cleared in P2DDR is read as the pin state.
6	P26	—*	R	
5	P25	—*	R	
4	P24	—*	R	
3	P23	—*	R	
2	P22	—*	R	
1	P21	—*	R	
0	P20	—*	R	

Note: * Determined by the states of the P27 to P20 pins.

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

P2PCR controls the on or off state of input pull-up MOSs for port 2.

Bit	Bit Name	Initial Value	R/W	Description
7	P27PCR	0	R/W	• In normal extended and single-chip modes When the pins are in the input states, the corresponding input pull-up MOS is turned on when a P2PCR bit is set to 1. • In multiplex extended mode Operation is not affected.
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	

7.3.5 Pin Functions

When the corresponding bit in PTCNT2 is cleared to 0, port 2 pins can be used as TPU I/O pins. The relationship between register setting values and pin functions is as follows.

- P27/TIOCB2/TCLKD/A15/AD15

  When the TIOCB2/TCLKDS bit in PTCNT2 is cleared to 0, this pin can be used as the TIOCB2/TCLKD pin.

  According to operating modes, the TPU channel 2 settings by the MD3 to MD0 bits in TMDR_2, the IOB3 to IOB0 bits in TIOR_2, and the CCLR1 and CCLR0 bits in TCR_2, and the combination of the TPSC2 to TPSC0 bits in TCR_0, the TIOCB2/TCLKDS bit, and the P27DDR bit, pin functions change as follows.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
P27DDR	0	1	—
Pin function	P27 input pin	A15 output pin	AD15 I/O pin
	TIOCB2 input pin*[1]		
	TCLKD input pin*[2]		

- Single-Chip Mode (EXPE = 0)

TIOCB2/ TCLKDS	0			1	
TPU channel 2 setting	Table below (2)		Table below (1)	—	
P27DDR	0	1	—	0	1
Pin function	P27 input pin	P27 output pin	TIOCB2 output pin	P27 input pin	P27 output pin
	TIOCB2 input pin*[1]				
	TCLKD input pin*[2]				

TPU channel 2 setting	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. When TIOCB2/TCLKDS = 0, MD3 to MD0 = B'0000 or B'01xx, and IOB3 = 1, this pin functions as the TIOCB2 input pin.

2. When TIOCB2/TCLKDS = 0 and TPSC2 to TPSC0 in TCR_0 = B'111, this pin functions as the TCLKD input pin. When TIOCB2/TCLKDS = 0 and phase-count mode is set to the TCR channel 2, this pin functions as the TCLKD input pin.

- P26/TIOCA2/A14/AD14

  When the TIOCA2S bit in PTCNT2 is cleared to 0, this pin can be used as the TIOCA2 pin.

  According to operating modes, the TPU channel 2 settings by the MD3 to MD0 bits in TMDR_2, the IOA3 to IOA0 bits in TIOR_2, and the CCLR1 and CCLR0 bits in TCR_2, and the combination of the TIOCA2S bit and the P26DDR bit, pin functions change as follows.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
P26DDR	0	1	—
Pin function	P26 input pin	A14 output pin	AD14 I/O pin
	TIOCA2 input pin*[1]		

- Single-Chip Mode (EXPE = 0)

TIOCA2S	0			1	
TPU channel 2 setting	Table below (2)		Table below (1)		—
P26DDR	0	1	—	0	1
Pin function	P26 input pin	P26 output pin	TIOCA2 output pin	P26 input pin	P26 output pin
	TIOCA2 input pin*[1]				

TPU channel 2 setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*[2] mode 1 output	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. When TIOCA2S = 0, MD3 to MD0 = B'0000 or B'01xx, and IOA3 = 1, this pin functions as the TIOCA2 input pin.

2. Output is disabled for TIOCB2.

- P25/TIOCB1/TCLKC/A13/AD13

  When the TIOCB1/TCLKCS bit in PTCNT2 is cleared to 0, this pin can be used as the TIOCB1/TCLKC pin.

  According to operating modes, the TPU channel 1 settings by the MD3 to MD0 bits in TMDR_1, the IOB3 to IOB0 bits in TIOR_1, and the CCLR1 and CCLR0 bits in TCR_1, and the combination of the TPSC2 to TPSC0 bits in TCR_0 and TCR_2, the TIOCB1/TCLKCS bit, and the P25DDR bit, pin functions change as follows.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
P25DDR	0	1	—
Pin function	P25 input pin	A13 output pin	AD13 I/O pin
	TIOCB1 input pin*[1]		
	TCLKC input pin*[2]		

- Single-Chip Mode (EXPE = 0)

TIOCB1/ TCLKCS	0			1	
TPU channel 1 setting	Table below (2)		Table below (1)		—
P25DDR	0	1	—	0	1
Pin function	P25 input pin	P25 output pin	TIOCB1 output pin	P25 input pin	P25 output pin
	TIOCB1 input pin*[1]				
	TCLKC input pin*[2]				

TPU channel 1 setting	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. When TIOCB1/TCLKCS = 0, MD3 to MD0 = B'0000 or B'01xx, and IOB3 to IOB0 = B'10xx, this pin functions as the TIOCB1 input pin.

2. When TIOCB1/TCLKCS = 0 and TPSC2 to TPSC0 in TCR_0 or TCR2 = B'111, this pin functions as the TCLKC input pin. When TIOCB1/TCLKCS = 0 and phase-count mode is set to the TCR channel 2, this pin functions as the TCLKC input pin.

- P24/TIOCA1/A12/AD12

  When the TIOCA1S bit in PTCNT2 is cleared to 0, this pin can be used as the TIOCA1 pin.

  According to operating modes, the TPU channel 1 settings by the MD3 to MD0 bits in TMDR_1, the IOA3 to IOA0 bits in TIOR_1, and the CCLR1 and CCLR0 bits in TCR_1, and the combination of the TIOCA1S bit and the P24DDR bit, pin functions change as follows.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
P24DDR	0	1	—
Pin function	P24 input pin	A12 output pin	AD12 I/O pin
	TIOCA1 input pin*[1]		

- Single-Chip Mode (EXPE = 0)

TIOCA2S	0			1	
TPU channel 1 setting	Table below (2)		Table below (1)		—
P24DDR	0	1	—	0	1
Pin function	P24 input pin	P24 output pin	TIOCA1 output pin	P24 input pin	P24 output pin
	TIOCA1 input pin*[1]				

TPU channel 1 setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*[2] mode 1 output	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. When TIOCA1S = 0, MD3 to MD0 = B'0000 or B'01xx, and IOA3 to IOA0 = B'10xx, this pin functions as the TIOCA1 input pin.

2. Output is disabled for TIOCB1.

- P23/TIOCD0/TCLKB/A11/AD11

  When the TIOCD0/TCLKBS bit in PTCNT2 is cleared to 0, this pin can be used as the TIOCD0/TCLKB pin.

  According to operating modes, the TPU channel 0 settings by the MD3 to MD0 bits in TMDR_0, the IOD3 to IOD0 bits in TIORL_0, and the CCLR2 to CCLR0 bits in TCR_0, and the combination of the TPSC2 to TPSC0 bits in TCR_0 to TCR_2, the TIOCD0/TCLKBS bit, and the P23DDR bit, pin functions change as follows.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
P23DDR	0	1	—
Pin function	P23 input pin	A11 output pin	AD11 I/O pin
	TIOCD0 input pin*[1]		
	TCLKB input pin*[2]		

- Single-Chip Mode (EXPE = 0)

TIOCD0/ TCLKBS	0			1	
TPU channel 0 setting	Table below (2)		Table below (1)		—
P23DDR	0	1	—	0	1
Pin function	P23 input pin	P23 output pin	TIOCD0 output pin	P23 input pin	P23 output pin
	TIOCD0 input pin*[1]				
	TCLKB input pin*[2]				

TPU channel 0 setting	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOD3 to IOD0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. When TIOCD0/TCLKBS = 0, MD3 to MD0 = B'0000, and IOD3 to IOD0 = B'10xx, this pin functions as the TIOCD0 input pin.

2. When TIOCD0/TCLKBS = 0 and TPSC2 to TPSC0 in one of TCR_0 to TCR2 = B′101, this pin functions as the TCLKB input pin. When TIOCB1/TCLKCS = 0 and phase-count mode is set to the TCR channel 1, this pin functions as the TCLKB input pin.

- P22/TIOCC0/TCLKA/A10/AD10

  When the TIOCC0/TCLKAS bit in PTCNT2 is cleared to 0, this pin can be used as the TIOCC0/TCLKA pin.

  According to operating modes, the TPU channel 0 settings by the MD3 to MD0 bits in TMDR_0, the IOC3 to IOC0 bits in TIORL_0, and the CCLR2 to CCLR0 bits in TCR_0, and the combination of the TPSC2 to TPSC0 bits in TCR_0 to TCR_2, the TIOCC0/TCLKAS bit, and the P22DDR bit, pin functions change as follows.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
P22DDR	0	1	—
Pin function	P22 input pin	A10 output pin	AD10 I/O pin
	TIOCC0 input pin*[1]		
	TCLKA input pin*[1]		

- Single-Chip Mode (EXPE = 0)

TIOCC0/ TCLKAS	0			1	
TPU channel 0 setting	Table below (2)		Table below (1)		—
P22DDR	0	1	—	0	1
Pin function	P22 input pin	P22 output pin	TIOCC0 output pin	P22 input pin	P22 output pin
	TIOCC0 input pin*[1]				
	TCLKA input pin*[2]				

TPU channel 0 setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOC3 to IOC0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM*[3] mode 1 output	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. When TIOCC0/TCLKAS = 0, MD3 to MD0 = B'0000, and IOC3 to IOC0 = B'10xx, this pin functions as the TIOCC0 input pin.

2. When TIOCC0/TCLKAS = 0 and TPSC2 to TPSC0 in one of TCR_0 to TCR2 = B'100, this pin functions as the TCLKA input pin. When TIOCC0/TCLKAS = 0 and phase-count mode is set to the TCR channel 1, this pin functions as the TCLKA input pin.

3. Output is disabled for TIOCD0. When BFA = 1 or BFB = 1 in TMDR0, output is disabled and the setting is the same as (2).

- P21/TIOCB0/A9/AD9

  When the TIOCB0S bit in PTCNT2 is cleared to 0, this pin can be used as the TIOCB0 pin.

  According to operating modes, the TPU channel 0 settings by the MD3 to MD0 bits in TMDR_0 and the IOB3 to IOB0 bits in TIORH_0, and the combination of the TIOCB0S bit and the P21DDR bit, pin functions change as follows.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
P21DDR	0	1	—
Pin function	P21 input pin	A9 output pin	AD9 I/O pin
	TIOCB0 input pin*		

- Single-Chip Mode (EXPE = 0)

TIOCB0S	0			1	
TPU channel 0 setting	Table below (2)		Table below (1)		—
P21DDR	0	1	—	0	1
Pin function	P21 input pin	P21 output pin	TIOCB0 output pin	P21 input pin	P21 output pin
	TIOCB0 input pin*				

TPU channel 0 setting	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B′0011 B'0101 to B′0111	—	B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

[Legend]

x: Don't care

Note: * When TIOCB0S = 0, MD3 to MD0 = B'0000, and IOB3 to IOB0 = B'10xx, this pin functions as the TIOCB0 input pin.

- P20/TIOCA0/A8/AD8

  When the TIOCA0S bit in PTCNT2 is cleared to 0, this pin can be used as the TIOCA0 pin.

  According to operating modes, the TPU channel 0 settings by the MD3 to MD0 bits in TMDR_0, the IOA3 to IOA0 bits in TIORH_0, and the CCLR2 to CCLR0 bits in TCR_0, and the combination of the TIOCA0S bit and the P20DDR bit, pin functions change as follows.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
P20DDR	0	1	—
Pin function	P20 input pin	A8 output pin	AD8 I/O pin
	TIOCA0 input pin*[1]		

- Single-Chip Mode (EXPE = 0)

TIOCA0S	0			1	
TPU channel 0 setting	Table below (2)		Table below (1)		—
P20DDR	0	1	—	0	1
Pin function	P20 input pin	P20 output pin	TIOCA0 output pin	P20 input pin	P20 output pin
	TIOCA0 input pin*[1]				

TPU channel 0 setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM*[2] mode 1 output	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. When TIOCA0S = 0, MD3 to MD0 = B'0000, and IOA3 to IOA0 = B'10xx, this pin functions as the TIOCA0 input pin.

2. Output is disabled for TIOCB0.

7.3.6 Port 2 Input Pull-Up MOS States

Port 2 has an on-chip input pull-up MOS that can be controlled by software. This input pull-up MOS can be used regardless of the operating mode. Table 7.3 summarizes the input pull-up MOS states.

Table 7.3 Port 2 Input Pull-Up MOS States

Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
Off	Off	On/Off	On/Off

[Legend]

Off: Always off.

On/Off : On when P2DDR = 0 and P2PCR = 1; otherwise off.

7.4 Port 3

Port 3 is an 8-bit I/O port. Port 3 pins also function as bidirectional data bus and $\overline{\text{ExIRQ}}$ input pins. 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 register (PORT3)
- Port 3 pull-up MOS control register (P3PCR)

7.4.1 Port 3 Data Direction Register (P3DDR)

The individual bits in P3DDR specify input or output for the pins of port 3. The read value is undefined.

Bit	Bit Name	Initial Value	R/W	Description
7	P37DDR	0	W	• In normal extended mode Operation is not affected. • In multiplex extended mode Operates as single-chip mode • In single-chip mode The corresponding port 3 pins are output ports when the 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	

7.4.2 Port 3 Data Register (P3DR)

P3DR stores output data for the port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P37DR	0	R/W	P3DR stores output data for the port 3 pins that are used as the general output ports.
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	

7.4.3 Port 3 Register (PORT3)

PORT3 reflects the pin state in port 3 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P37	—*	R	When this register is read, the bit that is set in P3DDR is read as the value of P3DR. The bit that is cleared in P3DDR is read as the pin state.
6	P36	—*	R	
5	P35	—*	R	
4	P34	—*	R	
3	P33	—*	R	
2	P32	—*	R	
1	P31	—*	R	
0	P30	—*	R	

Note: * Determined by the states of the P37 to P30 pins.

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

P3PCR controls the on or off state of input pull-up MOSs for port 3.

Bit	Bit Name	Initial Value	R/W	Description
7	P37PCR	0	R/W	• In normal extended mode Operation is not affected. • In multiplex extended mode Operates as single-chip mode • In single-chip mode When the pins are in the input states, the corresponding input pull-up MOS is turned on when a P3PCR bit is set to 1.
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	

7.4.5 Pin Functions

When the corresponding bit in PTCNT1 is set to 1, port 3 pins can be used as interrupt input pins ($\overline{\text{ExIRQ3}}$ to $\overline{\text{ExIRQ0}}$). In normal extended mode, port 3 pins function as data I/O pins. In multiplex extended mode, operation of port 3 pins is the same as that in single-chip mode. The relationship between register setting values and pin functions is as follows.

- P37/D15

  The pin function is switched as shown below according to the operating mode and the P37DDR bit.

	Extended Mode (EXPE = 1)			Single-Chip Mode (EXPE = 0)	
	Normal Extended Mode (ADMXE = 0)	Multiplex Extended Mode (ADMXE = 1)			
P37DDR	—	0	1	0	1
Pin function	D15 I/O pin	P37 input pin	P37 output pin	P37 input pin	P37 output pin

- P36/D14

  The pin function is switched as shown below according to the operating mode and the P36DDR bit.

	Extended Mode (EXPE = 1)			Single-Chip Mode (EXPE = 0)	
	Normal Extended Mode (ADMXE = 0)	Multiplex Extended Mode (ADMXE = 1)			
P36DDR	—	0	1	0	1
Pin function	D14 I/O pin	P36 input pin	P36 output pin	P36 input pin	P36 output pin

- P35/D13

  The pin function is switched as shown below according to the operating mode and the P35DDR bit.

	Extended Mode (EXPE = 1)			Single-Chip Mode (EXPE = 0)	
	Normal Extended Mode (ADMXE = 0)	Multiplex Extended Mode (ADMXE = 1)			
P35DDR	—	0	1	0	1
Pin function	D13 I/O pin	P35 input pin	P35 output pin	P35 input pin	P35 output pin

- P34/D12

  The pin function is switched as shown below according to the operating mode and the P34DDR bit.

	Extended Mode (EXPE = 1)			Single-Chip Mode (EXPE = 0)	
	Normal Extended Mode (ADMXE = 0)	Multiplex Extended Mode (ADMXE = 1)			
P34DDR	—	0	1	0	1
Pin function	D12 I/O pin	P34 input pin	P34 output pin	P34 input pin	P34 output pin

- P33/D11/$\overline{\text{ExIRQ3}}$

  When the IRQ3S bit in PTCNT1 is set to 1, this pin can be used as the $\overline{\text{ExIRQ3}}$ pin.

  The pin function is switched as shown below according to the operating mode, the IRQ3S bit, and the P33DDR bit.

	Extended Mode (EXPE = 1)				
	Normal Extended Mode (ADMXE = 0)	Multiplex Extended Mode (ADMXE = 1)		Single-Chip Mode (EXPE = 0)	
P33DDR	—	0	1	0	1
Pin function	D11 I/O pin	P33 input pin	P33 output pin	P33 input pin	P33 output pin
	ExIRQ3 input pin*				

Note: * When the IRQ3S bit in PTCNT1 is set to 1, this pin functions as the $\overline{\text{ExIRQ3}}$ input pin.

- P32/D10/$\overline{\text{ExIRQ2}}$

  When the IRQ2S bit in PTCNT1 is set to 1, this pin can be used as the $\overline{\text{ExIRQ2}}$ pin.

  The pin function is switched as shown below according to the operating mode, the IRQ2S bit, and the P32DDR bit.

	Extended Mode (EXPE = 1)				
	Normal Extended Mode (ADMXE = 0)	Multiplex Extended Mode (ADMXE = 1)		Single-Chip Mode (EXPE = 0)	
P32DDR	—	0	1	0	1
Pin function	D10 I/O pin	P32 input pin	P32 output pin	P32 input pin	P32 output pin
	$\overline{\text{ExIRQ2}}$ input pin*				

Note: * When the IRQ2S bit in PTCNT1 is set to 1, this pin functions as the $\overline{\text{ExIRQ2}}$ input pin.

- P31/D9/$\overline{\text{ExIRQ1}}$

  When the IRQ1S bit in PTCNT1 is set to 1, this pin can be used as the $\overline{\text{ExIRQ1}}$ pin.

  The pin function is switched as shown below according to the operating mode, the IRQ1S bit, and the P31DDR bit.

	Extended Mode (EXPE = 1)			Single-Chip Mode (EXPE = 0)	
	Normal Extended Mode (ADMXE = 0)	Multiplex Extended Mode (ADMXE = 1)			
P31DDR	—	0	1	0	1
Pin function	D9 I/O pin	P31 input pin	P31 output pin	P31 input pin	P31 output pin
	$\overline{\text{ExIRQ1}}$ input pin*				

Note: * When the IRQ1S bit in PTCNT1 is set to 1, this pin functions as the $\overline{\text{ExIRQ1}}$ input pin.

- P30/D8/$\overline{\text{ExIRQ0}}$

  When the IRQ0S bit in PTCNT1 is set to 1, this pin can be used as the $\overline{\text{ExIRQ0}}$ pin.

  The pin function is switched as shown below according to the operating mode, the IRQ0S bit, and the P30DDR bit.

	Extended Mode (EXPE = 1)			Single-Chip Mode (EXPE = 0)	
	Normal Extended Mode (ADMXE = 0)	Multiplex Extended Mode (ADMXE = 1)			
P30DDR	—	0	1	0	1
Pin function	D8 I/O pin	P30 input pin	P30 output pin	P30 input pin	P30 output pin
	$\overline{\text{ExIRQ0}}$ input pin*				

Note: * When the IRQ0S bit in PTCNT1 is set to 1, this pin functions as the $\overline{\text{ExIRQ0}}$ input pin.

7.4.6 Port 3 Input Pull-Up MOS States

Port 3 has an on-chip input pull-up MOS that can be controlled by software. This input pull-up MOS can be used in single-chip and multiplex extended modes. Table 7.4 summarizes the input pull-up MOS states.

Table 7.4 Port 3 Input Pull-Up MOS States

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
Normal extended mode (EXPE = 1, ADMXE = 0)	Off	Off	Off	Off
Single-chip mode (EXPE = 0)	Off	Off	On/Off	On/Off
Multiplex extended mode (EXPE = 1, ADMXE = 1)				

[Legend]

Off: Always off.

On/Off : On when input state and P3PCR = 1; otherwise off.

7.5 Port 4

Port 4 is an 8-bit I/O port. Port 4 pins also function as external interrupt pins, TMR0_0, TMR0_1, TMRX_0, TMRY_0, TMRX_1, TMRY_1, and FRT_1 input pins, and PWM output pins.

Port 4 has the following registers.

- Port 4 data direction register (P4DDR)
- Port 4 data register (P4DR)
- Port 4 register (PORT4)

7.5.1 Port 4 Data Direction Register (P4DDR)

The individual bits in P4DDR specify input or output for the pins of port 4. The read value is undefined.

Bit	Bit Name	Initial Value	R/W	Description
7	P47DDR	0	W	While a general I/O port function is selected, the corresponding port 4 pin is an output port when a P4DDR bit is set to 1, and an input port when cleared to 0.
6	P46DDR	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	

7.5.2 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 for the port 4 pins that are used as the general output ports.
6	P46DR	0	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	

7.5.3 Port 4 Register (PORT4)

PORT4 reflects the pin state in port 4 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P47	—*	R	When this register is read, the bit that is set in P4DDR is read as the value of P4DR. The bit that is cleared in P4DDR is read as the pin state.
6	P46	—*	R	
5	P45	—*	R	
4	P44	—*	R	
3	P43	—*	R	
2	P42	—*	R	
1	P41	—*	R	
0	P40	—*	R	

Note: * Determined by the states of the P47 to P40 pins.

7.5.4 Pin Functions

When the corresponding bit in PTCNT1 is cleared to 0, port 4 pins can be used as interrupt input pins ($\overline{IRQ7}$ to $\overline{IRQ0}$). When the corresponding bit in PTCNT0 is set to 1, port 4 pins can be used as PWM output pins (ExPW3 to ExPW0). The relationship between register setting values and pin functions is as follows.

- P47/$\overline{IRQ7}$/TMIY_0/ExPW3

  When the IRQ7S bit in PTCNT1 is cleared to 0, this pin can be used as an $\overline{IRQ7}$ pin.

  When the PW3S bit in PTCNT0 is set to 1, this pin can be used as an ExPW3 pin.

  The pin function is switched as shown below according to the combination of the OE3 bit in PWOER of the PWM, the IRQ7S bit, the PW3S bit, and the P47DDR bit.

  When the external clock is selected by the CKS2 to CKS0 bits in TCR of the TMRY_0, this pin functions as a TMCIY input pin. When the CCLR1 and CCLR0 bits in TCR of the TMRY_0 are both set to 1, this pin functions as a TMRIY input pin.

P47DDR	0		1	
PW3S	—	0	1	
OE3	—	—	0	1
Pin function	P47 input pin	P47 output pin	P47 output pin	PW3 output pin
	TMIY_0 (TMCIY/TMRIY) input pin			
	$\overline{IRQ7}$ input pin*			

Note: * When the IRQ7S bit in PTCNT1 is cleared to 0, this pin functions as the $\overline{IRQ7}$ input pin.

- P46/$\overline{IRQ6}$/TMIX_0/ExPW2

  When the IRQ6S bit in PTCNT1 is cleared to 0, this pin can be used as an $\overline{IRQ6}$ pin.

  When the PW2S bit in PTCNT0 is set to 1, this pin can be used as an ExPW2 pin.

  The pin function is switched as shown below according to the combination of the OE2 bit in PWOER of the PWM, the IRQ6S bit, the PW2S bit, and the P46DDR bit.

  When the external clock is selected by the CKS2 to CKS0 bits in TCR of the TMRX_0, this pin functions as a TMCIX input pin. When the CCLR1 and CCLR0 bits in TCR of the TMRX_0 are both set to 1, this pin functions as a TMRIX input pin.

P46DDR	0	1		
PW2S	—	0	1	
OE2	—	—	0	1
Pin function	P46 input pin	P46 output pin	P46 output pin	PW2 output pin
	TMIX_0 (TMCIX/TMRIX) input pin			
	$\overline{\text{IRQ6}}$ input pin*			

Note: * When the IRQ6S bit in PTCNT1 is cleared to 0, this pin functions as the $\overline{\text{IRQ6}}$ input pin.

- P45/$\overline{\text{IRQ5}}$/TMI0_0/ExPW1

  When the IRQ5S bit in PTCNT1 is cleared to 0, this pin can be used as an $\overline{\text{IRQ5}}$ pin.

  When the PW1S bit in PTCNT0 is set to 1, this pin can be used as an ExPW1 pin.

  The pin function is switched as shown below according to the combination of the OE1 bit in PWOER of the PWM, the IRQ5S bit, the PW1S bit, and the P45DDR bit.

  When the external clock is selected by the CKS2 to CKS0 bits in TCR of the TMR0_0, this pin functions as a TMCI0 input pin. When the CCLR1 and CCLR0 bits in TCR of the TMR0_0 are both set to 1, this pin functions as a TMRI0 input pin.

P45DDR	0	1		
PW1S	—	0	1	
OE1	—	—	0	1
Pin function	P45 input pin	P45 output pin	P45 output pin	PW1 output pin
	TMI0_0 (TMCI0/TMRI0) input pin			
	$\overline{\text{IRQ5}}$ input pin*			

Note: * When the IRQ5S bit in PTCNT1 is cleared to 0, this pin functions as the $\overline{\text{IRQ5}}$ input pin.

- P44/$\overline{\text{IRQ4}}$/TMIY_1/ExPW0

  When the IRQ4S bit in PTCNT1 is cleared to 0, this pin can be used as an $\overline{\text{IRQ4}}$ pin.

  When the PW0S bit in PTCNT0 is set to 1, this pin can be used as an ExPW0 pin.

  The pin function is switched as shown below according to the combination of the OE0 bit in PWOER of the PWM, the IRQ4S bit, the PW0S bit, and the P44DDR bit.

  When the external clock is selected by the CKS2 to CKS0 bits in TCR of the TMRY_1, this pin functions as a TMCIY input pin. When the CCLR1 and CCLR0 bits in TCR of the TMRY_1 are both set to 1, this pin functions as a TMRIY input pin.

P44DDR	0	1		
PW0S	—	0	1	
OE0	—	—	0	1
Pin function	P44 input pin	P44 output pin	P44 output pin	PW0 output pin
	TMIY_1 (TMCIY/TMRIY) input pin			
	$\overline{\text{IRQ4}}$ input pin*			

Note: * When the IRQ4S bit in PTCNT1 is cleared to 0, this pin functions as the $\overline{\text{IRQ4}}$ input pin.

- P43/$\overline{\text{IRQ3}}$/TMIX_1

  When the IRQ3S bit in PTCNT1 is cleared to 0, this pin can be used as an $\overline{\text{IRQ3}}$ pin.

  The pin function is switched as shown below according to the combination of the IRQ3S bit and the P43DDR bit.

  When the external clock is selected by the CKS2 to CKS0 bits in TCR of the TMRX_1, this pin functions as a TMCIX input pin. When the CCLR1 and CCLR0 bits in TCR of the TMRX_1 are both set to 1, this pin functions as a TMRIX input pin.

P43DDR	0	1
Pin function	P44 input pin	P44 output pin
	TMIX_1 (TMCIX/TMRIX) input pin	
	$\overline{\text{IRQ3}}$ input pin*	

Note: * When the IRQ3S bit in PTCNT1 is cleared to 0, this pin functions as the $\overline{\text{IRQ3}}$ input pin.

- P42/$\overline{\text{IRQ2}}$/TMI0_1

  When the IRQ2S bit in PTCNT1 is cleared to 0, this pin can be used as an $\overline{\text{IRQ2}}$ pin.

  The pin function is switched as shown below according to the combination of the IRQ2S bit and the P42DDR bit.

  When the external clock is selected by the CKS2 to CKS0 bits in TCR of the TMR0_1, this pin functions as a TMCI0 input pin. When the CCLR1 and CCLR0 bits in TCR of the TMR0_1 are both set to 1, this pin functions as a TMRI0 input pin.

P42DDR	0	1
Pin function	P42 input pin	P42 output pin
	TMI0_1 (TMCI0/TMRI0) input pin	
	$\overline{\text{IRQ2}}$ input pin*	

Note: * When the IRQ2S bit in PTCNT1 is cleared to 0, this pin functions as the $\overline{\text{IRQ2}}$ input pin.

- P41/$\overline{\text{IRQ1}}$/FTIC_1

  When the IRQ1S bit in PTCNT1 is cleared to 0, this pin can be used as an $\overline{\text{IRQ1}}$ pin.

  The pin function is switched as shown below according to the combination of the IRQ1S bit and the P41DDR bit.

  When the ICICE bit in TIER of the FRT_1 is set to 1, this pin functions as an FTIC_1 input pin.

P41DDR	0	1
Pin function	P41 input pin	P41 output pin
	FTIC_1 input pin	
	$\overline{\text{IRQ1}}$ input pin*	

Note: * When the IRQ1S bit in PTCNT1 is cleared to 0, this pin functions as the $\overline{\text{IRQ1}}$ input pin.

- P40/$\overline{\text{IRQ0}}$/FTIB_1

  When the IRQ0S bit in PTCNT1 is cleared to 0, this pin can be used as an $\overline{\text{IRQ0}}$ pin.

  The pin function is switched as shown below according to the combination of the IRQ0S bit and the P40DDR bit.

  When the ICIBE bit in TIER of the FRT_1 is set to 1, this pin functions as an FTIB_1 input pin.

P40DDR	0	1
Pin function	P40 input pin	P40 output pin
	FTIB_1 input pin	
	$\overline{\text{IRQ0}}$ input pin*	

Note: * When the IRQ0S bit in PTCNT1 is cleared to 0, this pin functions as the $\overline{\text{IRQ0}}$ input pin.

7.6 Port 5

Port 5 is an 8-bit I/O port. Port 5 pins also function as TMR0_1 and TMR1_1 output pins, SCI_0 and SCI_1 I/O pins, and PWM output pins.

Port 5 has the following registers.

- Port 5 data direction register (P5DDR)
- Port 5 data register (P5DR)
- Port 5 register (PORT5)

7.6.1 Port 5 Data Direction Register (P5DDR)

The individual bits in P5DDR specify input or output for the pins of port 5. The read value is undefined.

Bit	Bit Name	Initial Value	R/W	Description
7	P57DDR	0	W	While a general I/O port function is selected, the corresponding port 5 pin is an output port when a P5DDR bit is set to 1, and an input port when cleared to 0.
6	P56DDR	0	W	
5	P55DDR	0	W	
4	P54DDR	0	W	
3	P53DDR	0	W	
2	P52DDR	0	W	
1	P51DDR	0	W	
0	P50DDR	0	W	

7.6.2 Port 5 Data Register (P5DR)

P5DR stores output data for the port 5 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P57DR	0	R/W	P5DR stores output data for the port 5 pins that are used as the general output ports.
6	P56DR	0	R/W	
5	P55DR	0	R/W	
4	P54DR	0	R/W	
3	P53DR	0	R/W	
2	P52DR	0	R/W	
1	P51DR	0	R/W	
0	P50DR	0	R/W	

7.6.3 Port 5 Register (PORT5)

PORT5 reflects the pin state in port 5 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P57	—*	R	When this register is read, the bit that is set in P5DDR is read as the value of P5DR. The bit that is cleared in P5DDR is read as the pin state.
6	P56	—*	R	
5	P55	—*	R	
4	P54	—*	R	
3	P53	—*	R	
2	P52	—*	R	
1	P51	—*	R	
0	P50	—*	R	

Note: * Determined by the states of the P57 to P50 pins.

7.6.4 Pin Functions

When the corresponding bit in PTCNT0 is set to 1, port 5 pins can be used as PWM output pins (ExPW5 and ExPW4). The relationship between register setting values and pin functions is as follows.

- P57/TMO1_1/ExPW5

  When the PW5S bit in PTCNT0 is set to 1, this pin can be used as an ExPW5 pin.

  The pin function is switched as shown below according to the combination of the OS3 to OS0 bits in TCSR of the TMR1_1, the OE5 bit in PWOER of the PWM, the PW5S bit, and the P57DDR bit.

OS3 to OS0	All 0				At lease one bit is set to 1
P57DDR	0	1			—
PW5S	—	0	1		—
OE5	—	—	0	1	—
Pin function	P57 input pin	P57 output pin	P57 output pin	ExPW5 output pin	TMO1_1 output pin

- P56/TMO0_1/ExPW4

  When the PW4S bit in PTCNT0 is set to 1, this pin can be used as an ExPW4 pin.

  The pin function is switched as shown below according to the combination of the OS3 to OS0 bits in TCSR of the TMR0_1, the OE4 bit in PWOER of the PWM, the PW4S bit, and the P56DDR bit.

OS3 to OS0	All 0				At lease one bit is set to 1
P56DDR	0	1			—
PW4S	—	0	1		—
OE5	—	—	0	1	—
Pin function	P56 input pin	P56 output pin	P56 output pin	ExPW4 output pin	TMO0_1 output pin

- P55/RxD1

  The pin function is switched as shown below according to the combination of the RE bit in SCR of the SCI_1 and the P55DDR bit.

RE	0		1
P55DDR	0	1	—
Pin function	P55 input pin	P55 output pin	RxD1 input pin

- P54/TxD1

  The pin function is switched as shown below according to the combination of the TE bit in SCR of the SCI_1 and the P54DDR bit.

TE	0		1
P54DDR	0	1	—
Pin function	P54 input pin	P54 output pin	TxD1 output pin

- P53/SCK1

  The pin function is switched as shown below according to the combination of the C/$\overline{A}$ bit in SMR of the SCI_1, the CKE0 and CKE1 bits in SCR, and the P53DDR bit.

CKE1	0				1
C/$\overline{A}$	0			1	—
CKE0	0		1	—	—
P53DDR	0	1	—	—	—
Pin function	P53 input pin	P53 output pin	SCK1 output pin	SCK1 output pin	SCK1 input pin

- P52/RxD0

  The pin function is switched as shown below according to the combination of the RE bit in SCR of the SCI_0 and the P52DDR bit.

RE	0		1
P52DDR	0	1	—
Pin function	P52 input pin	P52 output pin	RxD0 input pin

- P51/TxD0

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

TE	0		1
P51DDR	0	1	—
Pin function	P51 input pin	P51 output pin	TxD0 output pin

- P50/SCK0

  The pin function is switched as shown below according to the combination of the C/$\overline{A}$ bit in SMR of the SCI_0, the CKE0 and CKE1 bits in SCR, and the P50DDR bit.

CKE1	0				1
C/$\overline{A}$	0			1	—
CKE0	0		1	—	—
P50DDR	0	1	—	—	—
Pin function	P50 input pin	P50 output pin	SCK0 output pin	SCK0 output pin	SCK0 input pin

7.7 Port 6

Port 6 is an 8-bit I/O port. Port 6 pins also function as bidirectional data bus, SCI_2 I/O pins, FRT_1 I/O pins, and TMRX_1 and TMRY_1 output pins. Port 6 functions change according to the operating mode.

Port 6 has the following registers.

- Port 6 data direction register (P6DDR)
- Port 6 data register (P6DR)
- Port 6 register (PORT6)
- Port 6 pull-up MOS control register (P6PCR)
- Port 6 open-drain control register (P6ODR)

7.7.1 Port 6 Data Direction Register (P6DDR)

The individual bits in P6DDR specify input or output for the pins of port 6. The read value is undefined.

Bit	Bit Name	Initial Value	R/W	Description
7	P67DDR	0	W	• In normal extended mode (16-bit data bus) Operation is not affected. • In normal extended mode (8-bit data bus)/multiplex extended mode Operates as single-chip mode • In single-chip mode While a general I/O port function is selected, the corresponding port 6 pin is an output port when a P6DDR bit is set to 1, and an input port 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	

7.7.2 Port 6 Data Register (P6DR)

P6DR stores output data for the port 6 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P67DR	0	R/W	P6DR stores output data for the port 6 pins that are used as the general output ports.
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	

7.7.3 Port 6 Register (PORT6)

PORT6 reflects the pin state in port 6 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P67	—*	R	When this register is read, the bit that is set in P6DDR is read as the value of P6DR. The bit that is cleared in P6DDR is read as the pin state.
6	P66	—*	R	
5	P65	—*	R	
4	P64	—*	R	
3	P63	—*	R	
2	P62	—*	R	
1	P61	—*	R	
0	P60	—*	R	

Note: * Determined by the states of the P67 to P60 pins.

7.7.4 Port 6 Pull-Up MOS Control Register (P6PCR)

P6PCR controls the on or off state of input pull-up MOSs for port 6.

Bit	Bit Name	Initial Value	R/W	Description
7	P67PCR	0	R/W	• In normal extended mode (16-bit data bus) Operation is not affected. • In normal extended mode (8-bit data bus)/multiplex extended mode Operates as single-chip mode • In single-chip mode When the pins are in the input states, the corresponding input pull-up MOS is turned on when a P6PCR bit is set to 1.
6	P66PCR	0	R/W	
5	P65PCR	0	R/W	
4	P64PCR	0	R/W	
3	P63PCR	0	R/W	
2	P62PCR	0	R/W	
1	P61PCR	0	R/W	
0	P60PCR	0	R/W	

7.7.5 Port 6 Open-Drain Control Register (P6ODR)

P6ODR specifies the output type of port 6.

Bit	Bit Name	Initial Value	R/W	Description
7	P67ODR	0	R/W	Setting a bit to 1 specifies the PMOS of the corresponding pin to the off state. When a pin function is specified as an output port, open-drain output is enabled. Push-pull output is enabled when a pin cleared to 0 is specified as an output port.
6	P66ODR	0	R/W	
5	P65ODR	0	R/W	
4	P64ODR	0	R/W	
3	P63ODR	0	R/W	
2	P62ODR	0	R/W	
1	P61ODR	0	R/W	
0	P60ODR	0	R/W	

7.7.6 Pin Functions

The relationship between register setting values and pin functions is as follows.

- P67/RxD2/D7

  The pin function is switched as shown below according to the combination of the operating mode, the RE bit in SCR of the SCI_2, and the P67DDR bit.
- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
Bus width*	16-bit	8-bit	—
RE	—	Single-chip operation	Single-chip operation
P67DDR	—		
Pin function	D7 I/O pin		

Note: * When the ABW3 to ABW1 bits in BCRA3 to BCRA1 are all set to 1, bus width is 8 bits, if any are cleared to 0, bus width is 16 bits.

- Single-Chip Mode (EXPE = 0)

RE	0		1
P67DDR	0	1	—
Pin function	P67 input pin	P67 output pin	RxD2 input pin

- P66/TxD2/D6

  The pin function is switched as shown below according to the combination of the operating mode, the TE bit in SCR of the SCI_2, and the P66DDR bit.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
Bus width*	16-bit	8-bit	—
TE	—	Single-chip operation	Single-chip operation
P66DDR	—		
Pin function	D6 I/O pin		

Note: * When the ABW3 to ABW1 bits in BCRA3 to BCRA1 are all set to 1, bus width is 8 bits, if any are cleared to 0, bus width is 16 bits.

- Single-Chip Mode (EXPE = 0)

TE	0		1
P66DDR	0	1	—
Pin function	P66 input pin	P66 output pin	TxD2 output pin

- P65/SCK2/D5

  The pin function is switched as shown below according to the combination of the operating mode, the C/$\overline{A}$ bit in SMR of the SCI_2, the CKE0 and CKE1 bits in SCR, and the P65DDR bit.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
Bus width*	16-bit	8-bit	—
CKE1	—	Single-chip operation	Single-chip operation
C/$\overline{A}$	—		
CKE0	—		
P65DDR	—		
Pin function	D5 I/O pin		

Note: * When the ABW3 to ABW1 bits in BCRA3 to BCRA1 are all set to 1, bus width is 8 bits, if any are cleared to 0, bus width is 16 bits.

- Single-Chip Mode (EXPE = 0)

CKE1	0				1
C/$\overline{A}$	0			1	—
CKE0	0		1	—	—
P65DDR	0	1	—	—	—
Pin function	P65 input pin	P65 output pin	SCK2 output pin	SCK2 output pin	SCK2 input pin

- P64/FTCI_1/D4

  The pin function is switched as shown below according to the combination of the operating mode and the P64DDR bit.

  When the CKS1 and CKS0 bits in TCR of the FRT_1 are all set to 1, this pin functions as an FTCI_1 input pin.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
Bus width*	16-bit	8-bit	—
P64DDR	—	Single-chip operation	Single-chip operation
Pin function	D4 I/O pin		
	FTCI_1 input pin		

Note: * When the ABW3 to ABW1 bits in BCRA3 to BCRA1 are all set to 1, bus width is 8 bits, if any are cleared to 0, bus width is 16 bits.

- Single-Chip Mode (EXPE = 0)

P64DDR	0	1
Pin function	P64 input pin	P64 output pin
	FTCI_1 input pin	

- P63/TMOY_1/D3

  The pin function is switched as shown below according to the combination of the operating mode, the OS3 to OS0 bits in TCSR of the TMRY_1, and the P63DDR bit.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
Bus width*	16-bit	8-bit	—
OS3 to OS0	—	Single-chip operation	Single-chip operation
P63DDR	—		
Pin function	D3 I/O pin		

Note: * When the ABW3 to ABW1 bits in BCRA3 to BCRA1 are all set to 1, bus width is 8 bits, if any are cleared to 0, bus width is 16 bits.

- Single-Chip Mode (EXPE = 0)

OS3 to OS0	All 0		At least one bit is set to 1
P63DDR	0	1	—
Pin function	P63 input pin	P63 output pin	TMOY_1 output pin

- P62/TMOX_1/D2

  The pin function is switched as shown below according to the combination of the operating mode, the OS3 to OS0 bits in TCSR of the TMRX_1, and the P62DDR bit.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
Bus width*	16-bit	8-bit	—
OS3 to OS0	—	Single-chip operation	Single-chip operation
P62DDR	—		
Pin function	D2 I/O pin		

Note: * When the ABW3 to ABW1 bits in BCRA3 to BCRA1 are all set to 1, bus width is 8 bits, if any are cleared to 0, bus width is 16 bits.

- Single-Chip Mode (EXPE = 0)

OS3 to OS0	All 0		At least one bit is set to 1
P62DDR	0	1	—
Pin function	P62 input pin	P62 output pin	TMOX_1 output pin

- P61/FTOB_1/D1

  The pin function is switched as shown below according to the combination of the operating mode, the OEB bit in TOCR of the FRT_1, and the P61DDR bit.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
Bus width*	16-bit	8-bit	—
OEB	—	Single-chip operation	Single-chip operation
P61DDR	—		
Pin function	D1 I/O pin		

Note: * When the ABW3 to ABW1 bits in BCRA3 to BCRA1 are all set to 1, bus width is 8 bits, if any are cleared to 0, bus width is 16 bits.

- Single-Chip Mode (EXPE = 0)

OEB	All 0		At least one bit is set to 1
P61DDR	0	1	—
Pin function	P61 input pin	P61 output pin	FTOB_1 output pin

- P60/FTOA_1/D0

  The pin function is switched as shown below according to the combination of the operating mode, the OEA bit in TOCR of the FRT_1, and the P60DDR bit.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)		Multiplex Extended Mode (ADMXE = 1)
Bus width*	16-bit	8-bit	—
OEA	—	Single-chip operation	Single-chip operation
P60DDR	—		
Pin function	D0 I/O pin		

Note: * When the ABW3 to ABW1 bits in BCRA3 to BCRA1 are all set to 1, bus width is 8 bits, if any are cleared to 0, bus width is 16 bits.

- Single-Chip Mode (EXPE = 0)

OEA	All 0		At least one bit is set to 1
P60DDR	0	1	—
Pin function	P60 input pin	P60 output pin	FTOA_1 output pin

7.7.7 Port 6 Input Pull-Up MOS States

Port 6 has an on-chip input pull-up MOS that can be controlled by software. Table 7.5 summarizes the input pull-up MOS states.

Table 7.5 Port 6 Input Pull-Up MOS States

Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
Off	Off	On/Off	On/Off

[Legend]
Off: Always off.
On/Off : On when input state and P6PCR = 1; otherwise off.

7.8 Port 7

Port 7 is an 8-bit input port. Port 7 pins also function as A/D converter analog input pins.

Port 7 has the following register.

- Port 7 register (PORT7)

7.8.1 Port 7 Register (PORT7)

PORT7 is an 8-bit read-only register, that reflects the pin state in port 7. PORT7 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P77	—*	R	When this register is read, the pin state is always read.
6	P76	—*	R	
5	P75	—*	R	
4	P74	—*	R	
3	P73	—*	R	
2	P72	—*	R	
1	P71	—*	R	
0	P70	—*	R	

Note: * Determined by the states of the P77 to P70 pins.

7.8.2 Pin Functions

Pin function relationships are listed below.

- P77/AN7

Pin function	P77 input pin
	AN7 input pin

- P76/AN6

Pin function	P76 input pin
	AN6 input pin

- P75/AN5

Pin function	P75 input pin
	AN5 input pin

- P74/AN4

Pin function	P74 input pin
	AN4 input pin

- P73/AN3

Pin function	P73 input pin
	AN3 input pin

- P72/AN2

Pin function	P72 input pin
	AN2 input pin

- P71/AN1

Pin function	P71 input pin
	AN1 input pin

- P70/AN0

Pin function	P70 input pin
	AN0 input pin

7.9 Port 8

Port 8 is an 8-bit I/O port. Port 8 pins also function as external trigger input pins for the A/D converter, PWMX output pins, SCI_3, SCI_4, IIC3_0, and IIC3_1 I/O pins, and TPU I/O pins. The output format for P80 to P83 which are general I/O ports is NMOS push-pull output.

Port 8 has the following registers.

- Port 8 data direction register (P8DDR)
- Port 8 data register (P8DR)
- Port 8 register (PORT8)

7.9.1 Port 8 Data Direction Register (P8DDR)

The individual bits in P8DDR specify input or output for the pins of port 8. The read value is undefined.

Bit	Bit Name	Initial Value	R/W	Description
7	P87DDR	0	W	While a general I/O port function is selected, the corresponding port 8 pin is an output port when a P8DDR bit is set to 1, and an input port when cleared to 0.
6	P86DDR	0	W	
5	P85DDR	0	W	
4	P84DDR	0	W	
3	P83DDR	0	W	
2	P82DDR	0	W	
1	P81DDR	0	W	
0	P80DDR	0	W	

7.9.2 Port 8 Data Register (P8DR)

P8DR stores output data for the port 8 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P87DR	0	R/W	P8DR stores output data for the port 8 pins that are used as the general output ports.
6	P86DR	0	R/W	
5	P85DR	0	R/W	
4	P84DR	0	R/W	
3	P83DR	0	R/W	
2	P82DR	0	R/W	
1	P81DR	0	R/W	
0	P80DR	0	R/W	

7.9.3 Port 8 Register (PORT8)

PORT8 reflects the pin state in port 8 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P87	—*	R	When this register is read, the bit that is set in P8DDR is read as the value of P8DR. The bit that is cleared in P8DDR is read as the pin state.
6	P86	—*	R	
5	P85	—*	R	
4	P84	—*	R	
3	P83	—*	R	
2	P82	—*	R	
1	P81	—*	R	
0	P80	—*	R	

Note: * Determined by the states of the P87 to P80 pins.

7.9.4 Pin Functions

When the corresponding bit in PTCNT2 is set to 1, port 8 pins can be used as TPU I/O pins (ExTIOCB0 and ExTIOCA0). The relationship between register setting values and pin functions is as follows.

- P87/ExTIOCB0/$\overline{\text{ADTRG}}$

  When the TIOCB0S bit in PTCNT2 is set to 1, this pin can be used as an ExTIOCB0 pin.

  The pin function is switched as shown below according to the combination of the TPU channel 0 settings by the MD3 to MD0 bits in TMDR_0 and the IOB3 to IOB0 bits in TIORH_0, the TIOCB0S bit, and the P87DDR bit.

  When the TRGS1 and TRGS0 bits in ADCR are all set to 1, this pin functions as an $\overline{\text{ADTRG}}$ input pin.

TIOCB0S	0		1		
TIOCB0 output	—		Table below (2)		Table below (1)
P87DDR	0	1	0	1	—
Pin function	P87 input pin	P87 output pin	P87 input pin	P87 output pin	ExTIOCB0 output pin
			ExTIOCB0 input pin*		
	$\overline{\text{ADTRG}}$ input pin				

TPU channel 0 setting	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

[Legend]

x: Don't care

Note: * When TIOCB0S = 1, MD3 to MD0 = B'0000, and IOB3 to IOB0 = B'10xx, this pin functions as the TIOCB0 input pin.

- P86/ExTIOCA0

  When the TIOCA0S bit in PTCNT2 is set to 1, this pin can be used as an ExTIOCA0 pin.

  The pin function is switched as shown below according to the combination of the TPU channel 0 settings by the MD3 to MD0 bits in TMDR_0, the IOA3 to IOA0 bits in TIORH_0, and the CCLR2 to CCLR0 bits in TCR_0, the TIOCA0S bit, and the P86DDR bit.

TIOCA0S	0		1		
TPU channel 0 setting	—		Table below (2)		Table below (1)
P86DDR	0	1	0	1	—
Pin function	P86 input pin	P86 output pin	P86 input pin	P86 output pin	ExTIOCA0 output pin
			ExTIOCA0 input pin*[1]		

TPU channel 0 setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM*[2] mode 1 output	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. When TIOCA0S = 1, MD3 to MD0 = B'0000, and IOA3 to IOA0 = B'10xx, this pin functions as the TIOCA0 input pin.

2. Output is disabled for TIOCB0.

- P85/PWX1

  The pin function is switched as shown below according to the combination of the OEB bit in DACR of the PWMX and the P85DDR bit.

OEB	0	1	
P85DDR	0	1	—
Pin function	P85 input pin	P85 output pin	PWX1 output pin

- P84/PWX0

  The pin function is switched as shown below according to the combination of the OEA bit in DACR of the PWMX and the P84DDR bit.

OEA	0		1
P84DDR	0	1	—
Pin function	P84 input pin	P84 output pin	PWX0 output pin

- P83/SDA1/RxD4

  The pin function is switched as shown below according to the combination of the RE bit in SCR of the SCI_4, the ICE bit in ICCRA of the IIC3_1, and the P83DDR bit. When this pin is used as the P83 output pin, the output format is NMOS push-pull output. The output format for SDA1 is NMOS open-drain output, and direct bus drive is possible.

ICE	0			1
RE	0		1	—
P83DDR	0	1	—	—
Pin function	P83 input pin	P83 output pin	RxD4 input pin	SDA1 I/O pin

- P82/SCL1/TxD4

  The pin function is switched as shown below according to the combination of the TE bit in SCR of the SCI_4, the ICE bit in ICCRA of the IIC3_1, and the P82DDR bit. When this pin is used as the TxD4 or P82 output pin, the output format is NMOS push-pull output. The output format for SCL1 is NMOS open-drain output, and direct bus drive is possible.

ICE	0			1
TE	0		1	—
P82DDR	0	1	—	—
Pin function	P82 input pin	P82 output pin	TxD4 output pin	SCL1 I/O pin

- P81/SDA0/RxD3

  The pin function is switched as shown below according to the combination of the RE bit in SCR of the SCI_3, the ICE bit in ICCRA of the IIC3_0, and the P81DDR bit. When this pin is used as the P81 output pin, the output format is NMOS push-pull output. The output format for SDA0 is NMOS open-drain output, and direct bus drive is possible.

ICE	0			1
RE	0		1	—
P81DDR	0	1	—	—
Pin function	P81 input pin	P81 output pin	RxD3 input pin	SDA0 I/O pin

- P80/SCL0/TxD3

  The pin function is switched as shown below according to the combination of the TE bit in SCR of the SCI_3, the ICE bit in ICCRA of the IIC3_0, and the P80DDR bit. When this pin is used as the TxD3 or P80 output pin, the output format is NMOS push-pull output. The output format for SCL0 is NMOS open-drain output, and direct bus drive is possible.

ICE	0			1
TE	0		1	—
P80DDR	0	1	—	—
Pin function	P80 input pin	P80 output pin	TxD3 output pin	SCL0 I/O pin

7.10 Port 9

Port 9 is an 8-bit I/O port. Port 9 pins also function as bus control I/O pins, system clock output pins, and TPU I/O pins. Port 9 functions change according to the operating mode.

Port 9 has the following registers.

- Port 9 data direction register (P9DDR)
- Port 9 data register (P9DR)
- Port 9 register (PORT9)
- Port function control register (PFCR)

7.10.1 Port 9 Data Direction Register (P9DDR)

The individual bits in P9DDR specify input or output for the pins of port 9. The read value is undefined.

Bit	Bit Name	Initial Value	R/W	Description
7	P97DDR	0	W	If port 9 pins are specified for use as the general I/O port, the corresponding port 9 pins are output ports when the P9DDR bits are set to 1, and input ports when cleared to 0.
6	P96DDR	0	W	When this bit is set to 1, the corresponding port 9 pin is the system clock output pin (φ), and as a general input port when cleared to 0.
5	P95DDR	0	W	If port 9 pins are specified for use as the general I/O port, the corresponding port 9 pins are output ports when the P9DDR bits are set to 1, and input ports when cleared to 0.
4	P94DDR	0	W	
3	P93DDR	0	W	
2	P92DDR	0	W	
1	P91DDR	0	W	
0	P90DDR	0	W	

7.10.2 Port 9 Data Register (P9DR)

P9DR stores output data for the port 9 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P97DR	0	R/W	P9DR stores output data for the port 9 pins that are used as the general output ports.
6	P96DR	0	R/W	
5	P95DR	0	R/W	
4	P94DR	0	R/W	
3	P93DR	0	R/W	
2	P92DR	0	R/W	
1	P91DR	0	R/W	
0	P90DR	0	R/W	

7.10.3 Port 9 Register (PORT9)

PORT9 reflects the pin state in port 9 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P97	—*	R	When this register is read, the bit that is set in P9DDR is read as the value of P9DR. The bit that is cleared in P9DDR is read as the pin state.
6	P96	—*	R	
5	P95	—*	R	
4	P94	—*	R	
3	P93	—*	R	
2	P92	—*	R	
1	P91	—*	R	
0	P90	—*	R	

Note: * Determined by the states of the P97 to P90 pins.

7.10.4 Port Function Control Register (PFCR)

PFCR controls the I/O port.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
4	CS3E	0	R/W	$\overline{CS3}$ Output Enable Selects to enable or disable the $\overline{CS3}$ output. 0: PA7 is designated as I/O port 1: PA7 is designated as $\overline{CS3}$ output pin
3	CS2E	0	R/W	$\overline{CS2}$ Output Enable Selects to enable or disable the $\overline{CS2}$ output. 0: P91 is designated as I/O port 1: P91 is designated as $\overline{CS2}$ output pin
2	CS1E	0	R/W	$\overline{CS1}$ Output Enable Selects to enable or disable the $\overline{CS1}$ output. 0: P92 is designated as I/O port 1: P92 is designated as $\overline{CS1}$ output pin
1	LWROE	0	R/W	$\overline{LWR}$ Output Enable Selects to enable or disable the $\overline{LWR}$ output. 0: P90 is designated as I/O port 1: P90 is designated as $\overline{LWR}$ output pin
0	ASOE	0	R/W	$\overline{AS}$ Output Enable Selects to enable or disable the $\overline{AS}$ output. 0: P95 is designated as I/O port 1: P95 is designated as $\overline{AS}$ output pin

7.10.5 Pin Functions

When the corresponding bit in PTCNT2 is set to 1, port 9 pins can be used as TPU I/O pins (ExTIOCD0/ExTCLKB, $\overline{\text{ExTIOCB2}}$/$\overline{\text{ExTCLKD}}$, ExTIOCA2, and ExTIOCB1/ExTCLKC). The relationship between register setting values and pin functions is as follows.

- P97/$\overline{\text{WAIT}}$/ExTIOCD0/ExTCLKB

  When the TIOCD0/TCLKBS bit in PTCNT2 is set to 1, this pin can be used as the ExTIOCD0/ExTCLKB pin.

  According to operating modes, the TPU channel 0 settings by the WMSn1 (n = 3 to 1) bit in BCRAn, the MD3 to MD0 bits in TMDR_0, the IOD3 to IOD0 bits in TIORL_0, and the CCLR2 to CCLR0 bits in TCR_0, and the combination of the TPSC2 to TPSC0 bits in TCR_0 to TCR_2, the TIOCD0/TCLKBS bit, and the P97DDR bit, the pin function is switched as shown below.

- Extended Mode (EXPE = 1)

<table>
<tr><td>WMS11, WMS21, WMS31</td><td>All 0</td><td colspan="3">At least one bit is set to 1</td></tr>
<tr><td>TIOCD0/TCLKBS</td><td>0</td><td>0</td><td colspan="2">1</td></tr>
<tr><td>TPU channel 0 setting</td><td rowspan="4">Single-chip operation</td><td>—</td><td>Table below (2)</td><td>Table below (1)</td></tr>
<tr><td>P97DDR</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td rowspan="2">Pin function</td><td rowspan="3">$\overline{\text{WAIT}}$ input pin</td><td>$\overline{\text{WAIT}}$ input pin</td><td rowspan="2">$\overline{\text{WAIT}}$ input pin</td></tr>
<tr><td>ExTIOCD0 input pin*[1]</td></tr>
<tr><td colspan="2">ExTCLKB input pin*[2]</td></tr>
</table>

- Single-Chip Mode (EXPE = 0)

<table>
<tr><td>TIOCD0/ TCLKBS</td><td colspan="2">0</td><td colspan="3">1</td></tr>
<tr><td>TPU channel 0 setting</td><td colspan="2">—</td><td colspan="2">Table below (2)</td><td>Table below (1)</td></tr>
<tr><td>P97DDR</td><td>0</td><td>1</td><td>0</td><td>1</td><td>—</td></tr>
<tr><td rowspan="3">Pin function</td><td rowspan="3">P97 input pin</td><td rowspan="3">P97 output pin</td><td>P97 input pin</td><td>P97 output pin</td><td rowspan="2">ExTIOCD0 output pin</td></tr>
<tr><td colspan="2">ExTIOCD0 input pin*[1]</td></tr>
<tr><td colspan="3">ExTCLKB input pin*[2]</td></tr>
</table>

TPU channel 0 setting	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. When TIOCD0/TCLKBS = 1, MD3 to MD0 = B'0000, and IOD3 to IOD0 = B'10xx, this pin functions as the TIOCD0 input pin.

2. When TIOCD0/TCLKBS = 1 and TPSC2 to TPSC0 in one of TCR_0 to TCR2 = B'101, this pin functions as the TCLKB input pin. When TIOCB1/TCLKCS = 1 and phase-count mode is set to the TCR channel 1, this pin functions as the TCLKB input pin.

- P96/ϕ

According to the setting of the P96DDR bit, the pin function is switched as shown below.

P96DDR	0	1
Pin function	P96 input pin	ϕ output pin

- P95/$\overline{AS}$/$\overline{AH}$

  According to the operating mode and combination of the ASOE bit and the P95DDR bit, the pin function is switched as shown below.

- Extended Mode (EXPE = 1)

	Normal Extended Mode (ADMXE = 0)			Multiplex Extended Mode (ADMXE = 1)
ASOE	0		1	—
P95DDR	0	1	—	—
Pin function	P95 input pin	P95 output pin	$\overline{AS}$ output pin	$\overline{AH}$ output pin

- Single-Chip Mode (EXPE = 0)

P95DDR	0	1
Pin function	P95 input pin	P95 output pin

- P94/$\overline{\text{HWR}}$

  According to the operating mode and the setting of the P94DDR bit, the pin function is switched as shown below.

	Extended Mode (EXPE = 1)	Single-Chip Mode (EXPE = 0)	
P94DDR	—	0	1
Pin function	$\overline{\text{HWR}}$ output pin	P94 input pin	P94 output pin

- P93/$\overline{\text{RD}}$

  According to the operating mode and the setting of the P93DDR bit, the pin function is switched as shown below.

	Extended Mode (EXPE = 1)	Single-Chip Mode (EXPE = 0)	
P93DDR	—	0	1
Pin function	$\overline{\text{RD}}$ output pin	P93 input pin	P93 output pin

- P92/$\overline{\text{CS1}}$/ExTIOCB2/ExTCLKD

  When the TIOCB2/TCLKDS bit in PTCNT2 is set to 1, this pin can be used as the ExTIOCB2/ExTCLKD2 pin.

  According to operating modes, the TPU channel 2 settings by the CS1E bit, the MD3 to MD0 bits in TMDR_2, the IOB3 to IOB0 bits in TIOR_2, and the CCLR1 and CCLR0 bits in TCR_2, and the combination of the TPSC2 to TPSC0 bits in TCR_0, the TIOCB2/TCLKDS bit, and the P92DDR bit, the pin function is switched as shown below.

- Extended Mode (EXPE = 1)

<table>
<tr><td>CS1E</td><td>0</td><td colspan="3">1</td></tr>
<tr><td>TIOCB2/TCLKDS</td><td rowspan="5">Single-chip operation</td><td>0</td><td colspan="2">1</td></tr>
<tr><td>TPU channel 2 setting</td><td>—</td><td>Table below (2)</td><td>Table below (1)</td></tr>
<tr><td>P92DDR</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td rowspan="3">Pin function</td><td rowspan="3">$\overline{\text{CS1}}$ output pin</td><td>$\overline{\text{CS1}}$ output pin</td><td rowspan="2">$\overline{\text{CS1}}$ output pin</td></tr>
<tr><td>ExTIOCB2 input pin*[1]</td></tr>
<tr><td colspan="2">ExTCLKD input pin*[2]</td></tr>
</table>

- Single-Chip Mode (EXPE = 0)

TIOCB2/TCLKDS	0		1		
TPU channel 2 setting	—		Table below (2)		Table below (1)
P92DDR	0	1	0	1	—
Pin function	P92 input pin	P92 output pin	P92 input pin	P92 output pin	ExTIOCB2 output pin
			ExTIOCB2 input pin*[1]		
			ExTCLKD input pin*[2]		

TPU channel 2 setting	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B′10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. When TIOCB2/TCLKDS = 1, MD3 to MD0 = B'0000 or B'01xx, and IOB3 = 1, this pin functions as the TIOCB2 input pin.

2. When TIOCB2/TCLKDS = 1 and TPSC2 to TPSC0 in TCR_0 = B'111, this pin functions as the TCLKD input pin. When TIOCB2/TCLKDS = 1 and phase-count mode is set to the TCR channel 2, this pin functions as the TCLKD input pin.

- P91/$\overline{\text{CS2}}$/ExTIOCA2

  When the TIOCA2S bit in PTCNT2 is set to 1, this pin can be used as the ExTIOCA2 pin.

  According to operating modes, the TPU channel 2 settings by the CS2E bit, the MD3 to MD0 bits in TMDR_2, the IOA3 to IOA0 bits in TIOR_2, and the CCLR1 and CCLR0 bits in TCR_2, and the combination of the TIOCA2S bit and the P91DDR bit, the pin function is switched as shown below.

- Extended Mode (EXPE = 1)

CS2E	0	1		
TIOCA2S	Single-chip operation	0	1	
TPU channel 2 setting		—	Table below (2)	Table below (1)
P91DDR		—	—	—
Pin function		$\overline{CS2}$ output pin	$\overline{CS2}$ output pin	$\overline{CS2}$ output pin
			ExTIOCA2 input pin*[1]	

- Single-Chip Mode (EXPE = 0)

TIOCA2S	0		1		
TPU channel 2 setting	—		Table below (2)		Table below (1)
P91DDR	0	1	0	1	—
Pin function	P91 input pin	P91 output pin	P91 input pin	P91 output pin	ExTIOCA2 output pin
			ExTIOCA2 input pin*[1]		

TPU channel 2 setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM mode 1 output*[2]	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. When TIOCA2S = 1, MD3 to MD0 = B'0000 or B'01xx, and IOA3 = 1, this pin functions as the TIOCA2 input pin.

2. Output is disabled for TIOCB2.

- P90/$\overline{LWR}$/ExTIOCB1/ExTCLKC

  When the TIOCB1/TCLKCS bit in PTCNT2 is set to 1, this pin can be used as the ExTIOCB1/ExTCLKC pin.

  According to operating modes, the TPU channel 1 settings by the LWROE bit, the MD3 to MD0 bits in TMDR_1, the IOB3 to IOB0 bits in TIOR_1, and the CCLR1 and CCLR0 bits in TCR_1, and the combination of the TPSC2 to TPSC0 bits in TCR_0 and TCR_2, the TIOCB1/TCLKCS bit, and the P90DDR bit, the pin function is switched as shown below.

- Extended Mode (EXPE = 1)

LWROE	0	1		
TIOCB1/TCLKCS	Single-chip operation	0	1	
TPU channel 1 setting		—	Table below (2)	Table below (1)
P90DDR		—	—	—
Pin function		$\overline{\text{LWR}}$ output pin	$\overline{\text{LWR}}$ output pin	$\overline{\text{LWR}}$ output pin
			ExTIOCB1 input pin*[1]	
			ExTCLKC input pin*[2]	

- Single-Chip Mode (EXPE = 0)

TIOCB1/ TCLKCS	0		1		
TPU channel 1 setting	—		Table below (2)		Table below (1)
P90DDR	0	1	0	1	—
Pin function	P90 input pin	P90 output pin	P90 input pin	P90 output pin	ExTIOCB1 output pin
			ExTIOCB1 input pin*[1]		
			ExTCLKC input pin*[2]		

TPU channel 1 setting	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. When TIOCB1/TCLKCS = 1, MD3 to MD0 = B'0000 or B'01xx, and IOB3 to IOB0 = B'10xx, this pin functions as the TIOCB1 input pin.

2. When TIOCB1/TCLKCS = 1 and TPSC2 to TPSC0 in TCR_0 or TCR_2 = B'111, this pin functions as the TCLKC input pin. When TIOCB1/TCLKCS = 0 and phase-count mode is set to the TCR channel 2, this pin functions as the TCLKC input pin.

7.11 Port A

Port A is an 8-bit I/O port. Port A pins also function as bus control output pins, SCI_3 and SCI_4 I/O pins, TMX_0, TMY_0, TM0_0, and PWM output pins, and FRT_0 and timer connection I/O pins.

Port A has the following registers. For details on the port function control register, refer to section 7.10.4, Port Function Control Register (PFCR).

- Port A data direction register (PADDR)
- Port A data register (PADR)
- Port A register (PORTA)
- Port function control register (PFCR)

7.11.1 Port A Data Direction Register (PADDR)

The individual bits in PADDR specify input or output for the pins of port A. The read value is undefined.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DDR	0	W	While a general I/O port function is selected, the corresponding port A pin is an output port when a PADDR bit is set to 1, and an input port when cleared to 0.
6	PA6DDR	0	W	
5	PA5DDR	0	W	
4	PA4DDR	0	W	
3	PA3DDR	0	W	
2	PA2DDR	0	W	
1	PA1DDR	0	W	
0	PA0DDR	0	W	

7.11.2 Port A Data Register (PADR)

PADR stores output data for the port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DR	0	R/W	PADR stores output data for the port A pins that are used as the general output ports.
6	PA6DR	0	R/W	
5	PA5DR	0	R/W	
4	PA4DR	0	R/W	
3	PA3DR	0	R/W	
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

7.11.3 Port A Register (PORTA)

PORTA reflects the pin state in port A and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7	—*	R	When this register is read, the bit that is set in PADDR is read as the value of PADR. The bit that is cleared in PADDR is read as the pin state.
6	PA6	—*	R	
5	PA5	—*	R	
4	PA4	—*	R	
3	PA3	—*	R	
2	PA2	—*	R	
1	PA1	—*	R	
0	PA0	—*	R	

Note: * Determined by the states of the PA7 to PA0 pins.

7.11.4 Pin Functions

When the corresponding bit in PTCNT2 is set to 1, port A pins can be used as TPU I/O pins (ExTIOCA1 and ExTIOCC0/ExTCLKA). The relationship between register setting values and pin functions is as follows.

- PA7/$\overline{CS3}$/ExTIOCA1

  When the TIOCA1S bit in PTCNT2 is set to 1, this pin can be used as the ExTIOCA1 pin.

  According to operating modes, the TPU channel 1 settings by the CS3E bit in PFCR, the MD3 to MD0 bits in TMDR_1, the IOA3 to IOA0 bits in TIOR_1, and the CCLR1 and CCLR0 bits in TCR_1, and the combination of the TIOCA1S bit and the PA7DDR bit, the pin function is switched as shown below.

- Extended Mode (EXPE = 1)

CS3E	0	1		
TIOCA1S	Single-chip operation	0	1	
TPU channel 1 setting		—	Table below (2)	Table below (1)
PA7DDR		—	—	—
Pin function		$\overline{CS3}$ output pin	$\overline{CS3}$ output pin	$\overline{CS3}$ output pin
			ExTIOCA1 input pin*[1]	

- Single-Chip Mode (EXPE = 0)

TIOCA1S	0		1		
TPU channel 1 setting	—		Table below (2)		Table below (1)
PA7DDR	0	1	0	1	—
Pin function	PA7 input pin	PA7 output pin	PA7 input pin	PA7 output pin	ExTIOCA1 output pin
			ExTIOCA1 input pin*[1]		

TPU channel 1 setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM mode 1 output*[2]	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. When TIOCA1S = 1, MD3 to MD0 = B'0000 or B'01xx, and IOA3 to IOA0 = B'10xx, this pin functions as the TIOCA1 input pin.

2. Output is disabled for TIOCB1.

- PA6/FTCI_0/HFBACKI

  According to the setting of the PA6DDR bit, the pin function is switched as shown below.

  When the CKS1 and CKS0 bits in TCR of the FRT_0 are all set to 1, this pin functions as the FTCI_0 input pin. When the SIMOD1 and SIMOD0 bits (IHI signal) in TCONRI of the timer connection_0 are cleared to B′00, this pin functions as the HFBACKI input pin.

PA6DDR	0	1
Pin function	PA6 input pin	PA6 output pin
	FTCI_0 input pin	
	HFBACKI input pin	

- PA5/FTIB_0/VFBACKI

  According to the setting of the PA5DDR bit, the pin function is switched as shown below.

  When the ICIBE bit in TIER of the FRT_0 is set to 1, this pin functions as the FTIB_0 input pin. When the SIMOD1 and SIMOD0 bits (IVI signal) in TCONRI of the timer connection_0 are cleared to B′00, this pin functions as the VFBACKI input pin.

PA5DDR	0	1
Pin function	PA5 input pin	PA5 output pin
	FTIB_0 input pin	
	VFBACKI input pin	

- PA4/FTIC_0/CLAMPO

  According to the combination of the CLOE bit in TCONRO of the timer connection_0 and the PA4DDR bit, the pin function is switched as shown below.

  When the ICICE bit in TIER of the FRT_0 is set to 1, this pin functions as the FTIC_0 input pin.

CLOE	0		1
PA4DDR	0	1	—
Pin function	PA4 input pin	PA4 output pin	CLAMPO output pin
	FTIC_0 input pin		

- PA3/FTOB_0/CBLANK

  According to the combination of the CBOE bit in TCONRO of the timer connection_0, the OEB bit in TOCR of the FRT_0, and the PA3DDR bit, the pin function is switched as shown below.

CBOE	0			1
OEB	0		1	—
PA3DDR	0	1	—	—
Pin function	PA3 input pin	PA3 output pin	FTOB_0 output pin	CBLANK output pin

- PA2/TMO0_0/ExTIOCC0/ExTCLKA

  When the TIOCC0/TCLKAS bit in PTCNT2 is set to 1, this pin can be used as the ExTIOCC0/ExTCLKA pin.

  According to the TPU channel 0 settings by the OS3 to OS0 bits in TCSR of the TMR0_0, the MD3 to MD0 bits in TMDR_0, the IOC3 to IOC0 bits in TIORL_0, and the CCLR2 to CCLR0 bits in TCR_0, and the combination of the TPSC2 to TPSC0 bits in TCR_0 to TCR_2, the TIOCC0/TCLKAS bit, and the PA2DDR bit, the pin function is switched as shown below.

TIOCC0/ TCLKAS	0			1		
TPU channel 0 setting	—			Table (2)		Table (1)
OS3 to OS0	All 0		At least one bit is set to 1	—		
PA2DDR	0	1	—	0	1	—
Pin function	PA2 input pin	PA2 output pin	TMO0_0 output pin	PA2 input pin	PA2 output pin	ExTIOCC0 output pin
				ExTIOCC0 input pin*[1]		
				ExTCLKA input pin*[2]		

TPU channel 0 setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM mode 1 output*[2]	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. When TIOCC0/TCLKAS = 1, MD3 to MD0 = B'0000, and IOC3 to IOC0 = B'10xx, this pin functions as the TIOCC0 input pin.

2. When TIOCC0/TCLKAS = 1 and TPSC2 to TPSC0 in one of TCR_0 to TCR_2 = B'100, this pin functions as the TCLKA input pin. When TIOCC0/TCLKAS = 1 and phase-count mode is set to the TCR channel 1, this pin functions as the TCLKA input pin.

3. Output is disabled for TIOCD0. When BFA = 1 or BFB = 1 in TMDR0, output is disabled and the setting is the same as (2).

- PA1/TMOY_0/ExPW7/SCK4

  When the PW7S bit in PTCNT0 is set to 1, this pin can be used as the ExPW7 pin.

  According to the combination of the C/$\overline{A}$ bit in SMR of the SCI_4, the CKE0 and CKE1 bits in SCR, the OS3 to OS0 bits in TCSR of the TMRY_0, the OE7 bit in PWOER of the PWM, the PW7S bit, and the PA1DDR bit, the pin function is switched as shown below.

CKE1	0							1
C/$\overline{A}$	0						1	—
CKE0	0					1	—	—
OS3 to OS0	All 0				At least one bit is set to 1	—	—	—
PA1DDR	0	1			—	—	—	—
PW7S	—	0	1		—	—	—	—
OE7	—	—	0	1	—	—	—	—
Pin function	PA1 input pin	PA1 output pin	PA1 output pin	PW7 output pin	TMOY_0 output pin	SCK4 output pin	SCK4 output pin	SCK4 input pin

- PA0/TMOX_0/ExPW6/SCK3

  When the PW6S bit in PTCNT0 is set to 1, this pin can be used as the ExPW6 pin.

  According to the combination of the C/$\overline{A}$ bit in SMR of the SCI_3, the CKE0 and CKE1 bits in SCR, the OS3 to OS0 bits in TCSR of the TMRX_0, the OE6 bit in PWOER of the PWM, the PW6S bit, and the PA0DDR bit, the pin function is switched as shown below.

CKE1	0							1
C/$\overline{A}$	0						1	—
CKE0	0					1	—	—
OS3 to OS0	All 0				At least one bit is set to 1	—	—	—
PA0DDR	0		1		—	—	—	—
PW6S	—	0	1		—	—	—	—
OE6	—	—	0	1	—	—	—	—
Pin function	PA0 input pin	PA0 output pin	PA0 output pin	PW6 output pin	TMOX_0 output pin	SCK3 output pin	SCK3 output pin	SCK3 input pin

7.12 Port B

Port B is an 8-bit I/O port. Port B pins also function as TMR1_1 input pins, TMR1_0 output pins, FRT_1 and timer connection_0 I/O pins, and timer connection_1 input pins.

Port B has the following registers.

- Port B data direction register (PBDDR)
- Port B data register (PBDR)
- Port B register (PORTB)

7.12.1 Port B Data Direction Register (PBDDR)

The individual bits in PBDDR specify input or output for the pins of port B. The read value is undefined.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DDR	0	W	While a general I/O port function is selected, the corresponding port B pin is an output port when a PBDDR bit is set to 1, and an input port when cleared to 0.
6	PB6DDR	0	W	
5	PB5DDR	0	W	
4	PB4DDR	0	W	
3	PB3DDR	0	W	
2	PB2DDR	0	W	
1	PB1DDR	0	W	
0	PB0DDR	0	W	

7.12.2 Port B Data Register (PBDR)

PBDR stores output data for the port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DR	0	R/W	PBDR stores output data for the port B pins that are used as the general output ports.
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

7.12.3 Port B Register (PORTB)

PORTB reflects the pin state in port B and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7	—*	R	When this register is read, the bit that is set in PBDDR is read as the value of PBDR. The bit that is cleared in PBDDR is read as the pin state.
6	PB6	—*	R	
5	PB5	—*	R	
4	PB4	—*	R	
3	PB3	—*	R	
2	PB2	—*	R	
1	PB1	—*	R	
0	PB0	—*	R	

Note: * Determined by the states of the PB7 to PB0 pins.

7.12.4 Pin Functions

The relationship between register setting values and pin functions is as follows.

- PB7/TMI1_0/HSYNCI_0

  According to the setting of the PB7DDR bit, the pin function is switched as shown below.

  When the external clock is selected by the CKS2 to CKS0 bits in TCR of the TMR1_0, this pin functions as the TMCI1_0 input pin. When the CCLR1 and CCLR0 bits in TCR of the TMR1_0 are all set to 1, this pin functions as the TMRI1_0 input pin. When the SIMOD1 bit (IHI signal) in TCONRI of the timer connection_0 is set to 1, this pin functions as the HSYNCI_0 input pin.

PB7DDR	0	1
Pin function	PB7 input pin	PB7 output pin
	TMI1_0 input pin	
	HSYNCI_0 input pin	

- PB6/FTIA_0/VSYNCI_0

  According to the setting of the PB6DDR bit, the pin function is switched as shown below.

  When the ICIAE bit in TIER of the FRT_0 is set to 1, this pin functions as the FTIA_0 input pin. When the SIMOD1 and SIMOD0 bits (IVI signal) in TCONRI of the timer connection_0 are all set to 1, this pin functions as the VSYNCI_0 input pin.

PB6DDR	0	1
Pin function	PB6 input pin	PB6 output pin
	FTIA_0 input pin	
	VSYNCI_0 input pin	

- PB5/FTID_0/CSYNCI_0

  According to the setting of the PB5DDR bit, the pin function is switched as shown below.

  When the ICIDE bit in TIER of the FRT_0 is set to 1, this pin functions as the FTID_0 input pin. When the SIMOD1 and SIMOD0 bits (IHI signal) in TCONRI of the timer connection_0 are set to 01, this pin functions as the CSYNCI_0 input pin.

PB5DDR	0	1
Pin function	PB5 input pin	PB5 output pin
	FTID_0 input pin	
	CSYNCI_0 input pin	

- PB4/TMI1_1/HSYNCI_1

  According to the setting of the PB4DDR bit, the pin function is switched as shown below.

  When the external clock is selected by the CKS2 to CKS0 bits in TCR of the TMR1_1, this pin functions as the TMCI1_1 input pin. When the CCLR1 and CCLR0 bits in TCR of the TMR1_1 are all set to 1, this pin functions as the TMRI1_1 input pin. When the SIMOD1 bit (IHI signal) in TCONRI of the timer connection_1 is set to 1, this pin functions as the HSYNCI_1 input pin.

PB4DDR	0	1
Pin function	PB4 input pin	PB4 output pin
	TMI1_1 input pin	
	HSYNCI_1 input pin	

- PB3/FTIA_1/VSYNCI_1

  According to the setting of the PB3DDR bit, the pin function is switched as shown below.

  When the ICIAE bit in TIER of the FRT_1 is set to 1, this pin functions as the FTIA_1 input pin. When the SIMOD1 and SIMOD0 bits (IVI signal) in TCONRI of the timer connection_1 are all set to 1, this pin functions as the VSYNCI_1 input pin.

PB3DDR	0	1
Pin function	PB3 input pin	PB3 output pin
	FTIA_1 input pin	
	VSYNCI_1 input pin	

- PB2/FTID_1/CSYNCI_1

  According to the setting of the PB2DDR bit, the pin function is switched as shown below.

  When the ICIDE bit in TIER of the FRT_1 is set to 1, this pin functions as the FTID_1 input pin. When the SIMOD1 and SIMOD0 bits (IHI signal) in TCONRI of the timer connection_1 are set to 01, this pin functions as the CSYNCI_1 input pin.

PB2DDR	0	1
Pin function	PB2 input pin	PB2 output pin
	FTID_1 input pin	
	CSYNCI_1 input pin	

- PB1/TMO1_0/HSYNCO

  According to the combination of the HOE bit in TCONRO of the timer connection_0, the OS3 to OS0 bits in TCSR of the TMR1_0, and the PB1DDR bit, the pin function is switched as shown below.

HOE	0			1
OS3 to OS0	All 0		At least one bit is set to 1	—
PB1DDR	0	1	—	—
Pin function	PB1 input pin	PB1 output pin	TMO1_0 output pin	HSYNCO output pin

- PB0/FTOA_0/VSYNCO

  According to the combination of the VOE bit in TCONRO of the timer connection_0, the OEA bit in TOCR of the FRT_0, and the PB0DDR bit, the pin function is switched as shown below.

VOE	0		1	
OEA	0		1	—
PB0DDR	0	1	—	—
Pin function	PB0 input pin	PB0 output pin	FTOA_0 output pin	VSYNCO output pin

7.13 Port C

Port C is an 8-bit I/O port. Port C pins also function as IIC3_2 and IIC3_3 I/O pins and on-chip emulator I/O pins. The output format for PC0 to PC3 which are general I/O ports is NMOS push-pull output. PC4 to PC7 which are general input ports are not supported by the on-chip emulator.

Port C has the following registers.

- Port C data direction register (PCDDR)
- Port C data register (PCDR)
- Port C register (PORTC)

7.13.1 Port C Data Direction Register (PCDDR)

The individual bits in PCDDR specify input or output for the pins of port C. The read value is undefined.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	W	Reserved These bits cannot be modified.
3	PC3DDR	0	W	While a general I/O port function is selected, the corresponding port C pin is an output port when a PCDDR bit is set to 1, and an input port when cleared to 0.
2	PC2DDR	0	W	
1	PC1DDR	0	W	
0	PC0DDR	0	W	

7.13.2 Port C Data Register (PCDR)

PCDR stores output data for the port C pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	W	Reserved The initial value should not be changed.
3	PC3DR	0	R/W	PCDR stores output data for the port C pins that are used as the general output ports.
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

7.13.3 Port C Register (PORTC)

PORTC reflects the pin state in port C and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7	—*	R	These bits are always read as pin states.
6	PC6	—*	R	
5	PC5	—*	R	
4	PC4	—*	R	
3	PC3	—*	R	When this register is read, the bit that is set in PCDDR is read as the value of PCDR. The bit that is cleared in PCDDR is read as the pin state.
2	PC2	—*	R	
1	PC1	—*	R	
0	PC0	—*	R	

Note: * Determined by the states of the PC7 to PC0 pins.

7.13.4 Pin Functions

The relationship between register setting values and pin functions is as follows.

- PC7/ETDO

Pin function	PC7 input pin

Note: When the on-chip emulator is used, this pin functions as the ETDO output pin. When the on-chip emulator is not used (normal operation), a high or low level signal should be input to this pin.

- PC6/ETDI

Pin function	PC6 input pin

Note: When the on-chip emulator is used, this pin functions as the ETDI input pin. When no signal is input, this pin is fixed to 1 by the internal pull-up.

- PC5/ETCK

Pin function	PC5 input pin

Note: When the on-chip emulator is used, this pin functions as the ETCK input pin. When no signal is input, this pin is fixed to 1 by the internal pull-up.

- PC4/ETMS

Pin function	PC4 input pin

Note: When the on-chip emulator is used, this pin functions as the ETMS input pin. When no signal is input, this pin is fixed to 1 by the internal pull-up.

- PC3/SDA3

  The pin function is switched as shown below according to the combination of the ICE bit in ICCRA of the IIC3_3 and the PC3DDR bit. When this pin is used as the PC3 output pin, the output format is NMOS push-pull output. The output format for SDA3 is NMOS open-drain output, and direct bus drive is possible.

ICE	0		1
PC3DDR	0	1	—
Pin function	PC3 input pin	PC3 output pin	SDA3 I/O pin

- PC2/SCL3

  The pin function is switched as shown below according to the combination of the ICE bit in ICCRA of the IIC3_3 and the PC2DDR bit. When this pin is used as the PC2 output pin, the output format is NMOS push-pull output. The output format for SCL3 is NMOS open-drain output, and direct bus drive is possible.

ICE	0		1
PC2DDR	0	1	—
Pin function	PC2 input pin	PC2 output pin	SCL3 I/O pin

- PC1/SDA2

  The pin function is switched as shown below according to the combination of the ICE bit in ICCRA of the IIC3_2 and the PC1DDR bit. When this pin is used as the PC1 output pin, the output format is NMOS push-pull output. The output format for SDA2 is NMOS open-drain output, and direct bus drive is possible.

ICE	0		1
PC1DDR	0	1	—
Pin function	PC1 input pin	PC1 output pin	SDA2 I/O pin

- PC0/SCL2

  The pin function is switched as shown below according to the combination of the ICE bit in ICCRA of the IIC3_2 and the PC0DDR bit. When this pin is used as the PC0 output pin, the output format is NMOS push-pull output. The output format for SCL2 is NMOS open-drain output, and direct bus drive is possible.

ICE	0		1
PC0DDR	0	1	—
Pin function	PC0 input pin	PC0 output pin	SCL2 I/O pin

7.14 Change of Peripheral Function Pins

I/O ports that also function as peripheral modules, such as the 8-bit PWM timer output, external interrupts, and TPU I/O, and, can be changed. They are changed according to the setting of PTCNT0 to PTCNT2. The pin name of the peripheral function is indicated by adding 'Ex' at the head of the original pin name. In each peripheral function description, the original pin name is used.

7.14.1 Port Control Register 0 (PTCNT0)

PTCNT0 selects ports that also function as 8-bit PWM timer output pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PW7S	0	R/W	Selects the PW7 output pin for the 8-bit PWM timer. 0: P17/$\overline{\text{PW7}}$ is selected 1: PA1/$\overline{\text{ExPW7}}$ is selected
6	PW6S	0	R/W	Selects the PW6 output pin for the 8-bit PWM timer. 0: P16/$\overline{\text{PW6}}$ is selected 1: PA0/$\overline{\text{ExPW6}}$ is selected
5	PW5S	0	R/W	Selects the PW5 output pin for the 8-bit PWM timer. 0: P15/$\overline{\text{PW5}}$ is selected 1: P57/$\overline{\text{ExPW5}}$ is selected
4	PW4S	0	R/W	Selects the PW4 output pin for the 8-bit PWM timer. 0: P14/$\overline{\text{PW4}}$ is selected 1: P56/$\overline{\text{ExPW4}}$ is selected
3	PW3S	0	R/W	Selects the PW3 output pin for the 8-bit PWM timer. 0: P13/$\overline{\text{PW3}}$ is selected 1: P47/$\overline{\text{ExPW3}}$ is selected
2	PW2S	0	R/W	Selects the PW2 output pin for the 8-bit PWM timer. 0: P12/$\overline{\text{PW2}}$ is selected 1: P46/$\overline{\text{ExPW2}}$ is selected
1	PW1S	0	R/W	Selects the PW1 output pin for the 8-bit PWM timer. 0: P11/$\overline{\text{PW1}}$ is selected 1: P45/$\overline{\text{ExPW1}}$ is selected
0	PW0S	0	R/W	Selects the PW0 output pin for the 8-bit PWM timer. 0: P10/$\overline{\text{PW0}}$ is selected 1: P44/$\overline{\text{ExPW0}}$ is selected

7.14.2 Port Control Register 1 (PTCNT1)

PTCNT1 selects ports that also function as $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input pins.

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7S	0	R/W	Selects the $\overline{\text{IRQ7}}$ input pin. 0: P47/$\overline{\text{IRQ7}}$ is selected 1: P07/$\overline{\text{ExIRQ7}}$ is selected
6	IRQ6S	0	R/W	Selects the $\overline{\text{IRQ6}}$ input pin. 0: P46/$\overline{\text{IRQ6}}$ is selected 1: P06/$\overline{\text{ExIRQ6}}$ is selected
5	IRQ5S	0	R/W	Selects the $\overline{\text{IRQ5}}$ input pin. 0: P45/$\overline{\text{IRQ5}}$ is selected 1: P05/$\overline{\text{ExIRQ5}}$ is selected
4	IRQ4S	0	R/W	Selects the $\overline{\text{IRQ4}}$ input pin. 0: P44/$\overline{\text{IRQ4}}$ is selected 1: P04/$\overline{\text{ExIRQ4}}$ is selected
3	IRQ3S	0	R/W	Selects the $\overline{\text{IRQ3}}$ input pin. 0: P43/$\overline{\text{IRQ3}}$ is selected 1: P33/$\overline{\text{ExIRQ3}}$ is selected
2	IRQ2S	0	R/W	Selects the $\overline{\text{IRQ2}}$ input pin. 0: P42/$\overline{\text{IRQ2}}$ is selected 1: P32/$\overline{\text{ExIRQ2}}$ is selected
1	IRQ1S	0	R/W	Selects the $\overline{\text{IRQ1}}$ input pin. 0: P41/$\overline{\text{IRQ1}}$ is selected 1: P31/$\overline{\text{ExIRQ1}}$ is selected
0	IRQ0S	0	R/W	Selects the $\overline{\text{IRQ0}}$ input pin. 0: P40/$\overline{\text{IRQ0}}$ is selected 1: P30/$\overline{\text{ExIRQ0}}$ is selected

7.14.3 Port Control Register 2 (PTCNT2)

PTCNT2 selects ports that also function as TPU I/O pins.

Bit	Bit Name	Initial Value	R/W	Description
7	TIOCB2/ TCLKDS	0	R/W	Selects the TIOCB2/TCLKD I/O pin for the TPU. 0: P27/TIOCB2/TCLKD is selected 1: P92/ExTIOCB2/ExTCLKD is selected
6	TIOCA2S	0	R/W	Selects the TIOCA2 I/O pin for the TPU. 0: P26/TIOCA2 is selected 1: P91/ExTIOCA2 is selected
5	TIOCB1/ TCLKCS	0	R/W	Selects the TIOCB1/TCLKC I/O pin for the TPU. 0: P25/TIOCB1/TCLKC is selected 1: P90/ExTIOCB1/ExTCLKC is selected
4	TIOCA1S	0	R/W	Selects the TIOCA1 I/O pin for the TPU. 0: P24/TIOCA1 is selected 1: PA7/ExTIOCA1 is selected
3	TIOCD0/ TCLKBS	0	R/W	Selects the TIOCD0/TCLKB I/O pin for the TPU. 0: P23/TIOCD0/TCLKB is selected 1: P97/ExTIOCD0/ExTCLKB is selected
2	TIOCC0/ TCLKAS	0	R/W	Selects the TIOCC0/TCLKA I/O pin for the TPU. 0: P22/TIOCC0/TCLKA is selected 1: PA2/ExTIOCC0/ExTCLKA is selected
1	TIOCB0S	0	R/W	Selects the TIOCB0 I/O pin for the TPU. 0: P21/TIOCB0 is selected 1: P87/ExTIOCB0 is selected
0	TIOCA0S	0	R/W	Selects the TIOCA0 I/O pin for the TPU. 0: P20/TIOCA0 is selected 1: P86/ExTIOCA0 is selected

Section 8 8-Bit PWM Timer (PWM)

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

8.1 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 8.1 shows a block diagram of the PWM timer.

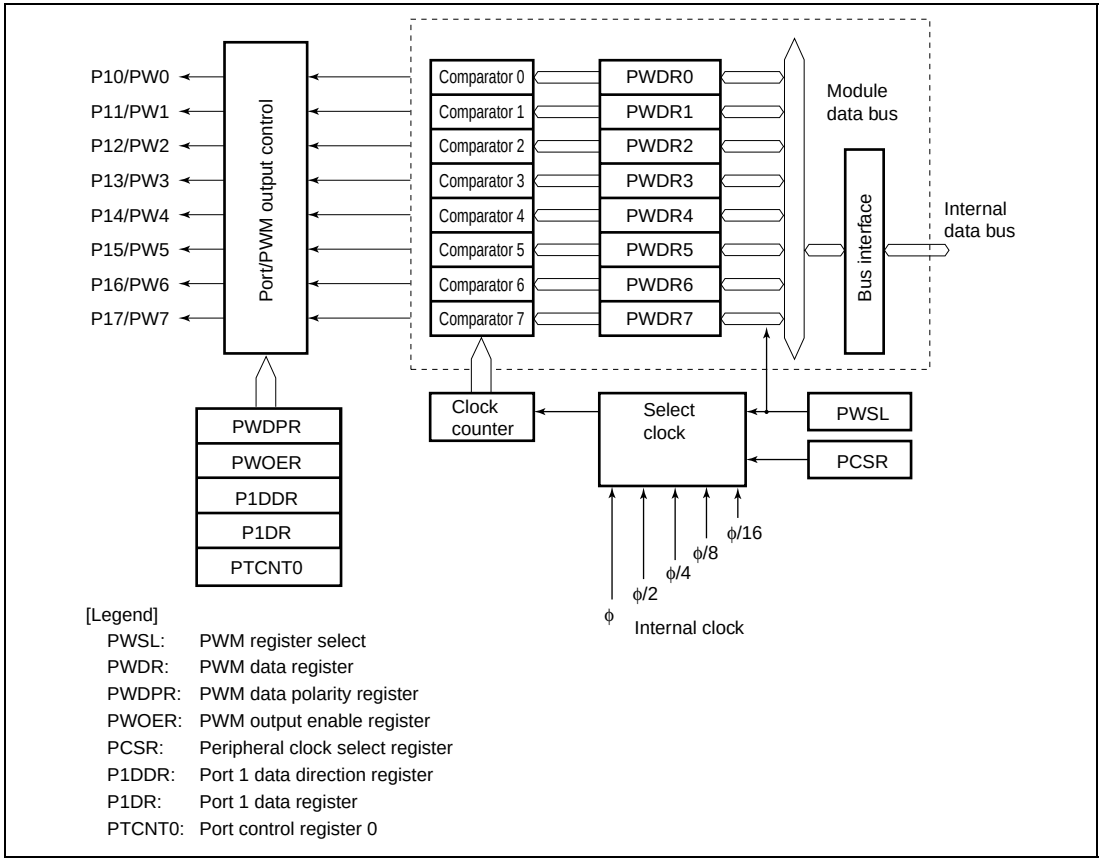


Figure 8.1 Block Diagram of PWM Timer

8.2 Input/Output Pin

Table 8.1 shows the PWM output pin.

Table 8.1 Pin Configuration

Name	Symbol	I/O	Function
PWM output pins 7 to 0	PW7 to PW0	Output	PWM timer pulse output 7 to 0

8.3 Register Descriptions

The PWM has the following registers.

- PWM register select (PWSL)
- PWM data registers 7 to 0 (PWDR7 to PWDR0)
- PWM data polarity register (PWDPR)
- PWM output enable register (PWOER)
- Peripheral clock select register (PCSR)

8.3.1 PWM Register Select (PWSL)

PWSL selects 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 of the PWM. For details, see table 8.2. 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 the 20-MHz system clock (ϕ), the resolution, PWM conversion period, and carrier frequency are as shown in table 8.3.
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.
3	—	0	R/W	Reserved The initial value should not be changed.
2	RS2	0	R/W	Register Select These bits select the PWM data register. 000: PWDR0 selected 001: PWDR1 selected 010: PWDR2 selected 011: PWDR3 selected 100: PWDR4 selected 101: PWDR5 selected 110: PWDR6 selected 111: PWDR7 selected
1	RS1	0	R/W	
0	RS0	0	R/W	

Table 8.2 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 8.3 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	78.13 kHz

8.3.2 PWM Data Registers 7 to 0 (PWDR7 to PWDR0)

PWDR are 8-bit readable/writable registers. The PWM has eight 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 the 0/1 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 additional 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 the 0/1 ratio within the conversion period. For 256/256 (100%) output, port output should be used.

8.3.3 PWM Data Polarity Register (PWDPR)

PWDPR selects the PWM output phase.

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	OS1	0	R/W	1: PWM inverted output (PWDR value corresponds to low width of output)
0	OS0	0	R/W	

8.3.4 PWM Output Enable Register (PWOER)

PWOER switches between PWM output and port output.

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	
3	OE3	0	R/W	P1nDDR OEn: Pin state
2	OE2	0	R/W	
1	OE1	0	R/W	0x: Port input
0	OE0	0	R/W	10: Port output or PWM 256/256 output
				11: PWM output (0 to 255/256 output)

[Legend]
n = 7 to 0
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 when IOSE = 1 and CS256E = 0 in SYSCR in single-chip mode or in extended mode with on-chip ROM enabled. 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 OS bit, so the value should be set to DR beforehand.

8.3.5 Peripheral Clock Select Register (PCSR)

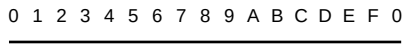
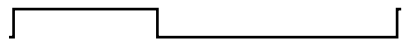
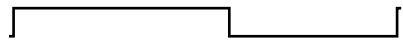
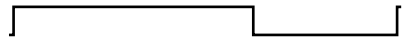
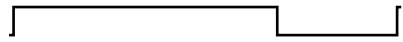
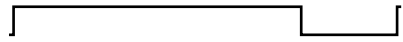
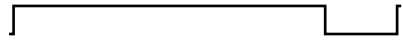
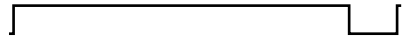
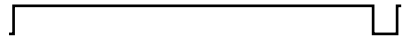
PCSR selects the PWM input clock.

Bit	Bit Name	Initial Value	R/W	Description
7	PWCKXC	0	R/W	See section 9.3.4, Peripheral Clock Select Register (PCSR).
6	PWCKXB	0	R/W	
5	PWCKXA	0	R/W	
4 to 2	—	All 0	R/W	Reserved The initial value should not be changed.
1	PWCKB	0	R/W	PWM Clock Select B, A
0	PWCKA	0	R/W	Together with bits PWCKE and PWCKS in PWSL, these bits select the internal clock input to TCNT of the PWM. For details, see table 8.2.

8.4 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 8.4 shows the duty cycles of the basic pulse.

Table 8.4 Duty Cycle of Basic Pulse

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

The lower four bits in 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 in PWDR are 0000, there is no rising edge of the basic pulse, but the timing for adding pulses is the same. Table 8.5 shows the positions of the additional pulses added to the basic pulses, and figure 8.2 shows an example of additional pulse timing.

Table 8.5 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

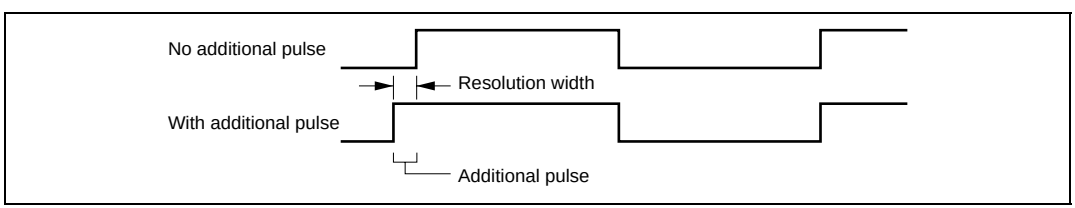


Figure 8.2 Example of Additional Pulse Timing (When Upper 4 Bits in PWDR = 1000)

Section 9 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.

9.1 Features

- Division of pulse into multiple base cycles to reduce ripple
- Eight resolution settings
  The resolution can be set to 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 operating clocks (by combination of eight resolution settings and two base cycle settings)

Figure 9.1 shows a block diagram of the PWMX (D/A) module.

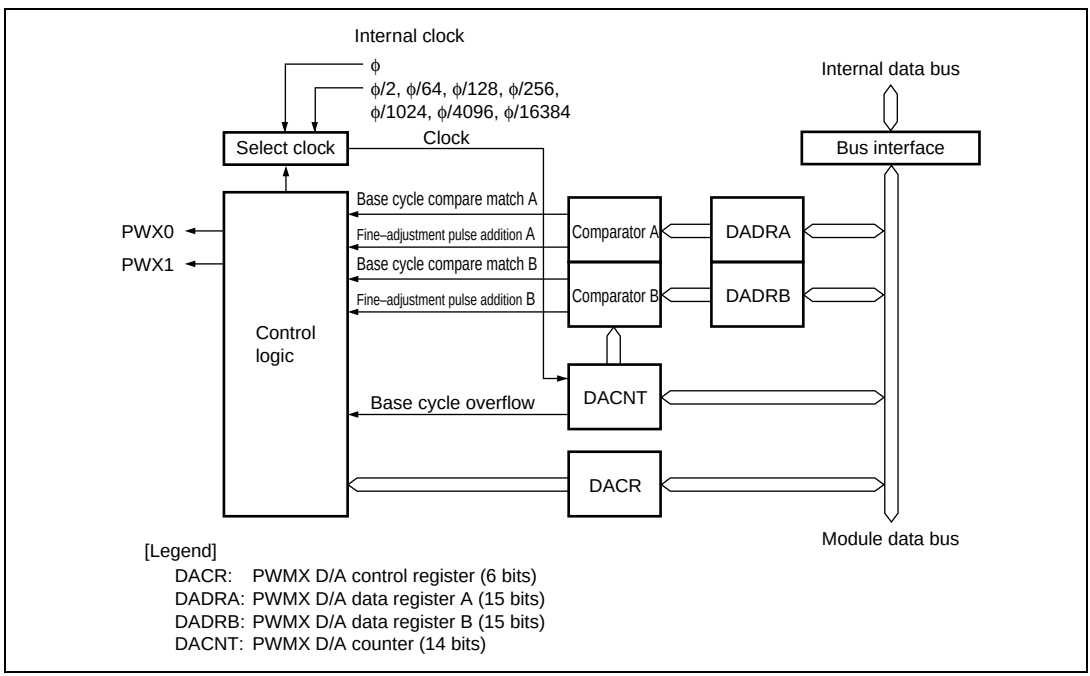


Figure 9.1 Block Diagram of PWMX (D/A)

9.2 Input/Output Pins

Table 9.1 lists the PWMX (D/A) input and output pins.

Table 9.1 Pin Configuration

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

9.3 Register Descriptions

The PWMX (D/A) has the following registers.

- PWMX (D/A) counters H and L (DACNTH and DACNTL)
- 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.

9.3.1 PWMX (D/A) Counters H and L (DACNTH and DACNTL)

DACNT is a 14-bit readable/writable up-counter. The input clock is selected by the CKS bit in DACR. DACNT functions as the timebase for both PWMX (D/A) channels. When a channel operates with 14-bit accuracy, it uses all DACNT bits. When a channel operates with 12-bit accuracy, it uses the lower 12 bits and ignores the upper two bits. DACNT cannot be accessed in 8-bit units. DACNT should always be accessed in 16-bit units. For details, see section 9.4, Bus Master Interface.

- DACNTH

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	UC7 to UC0	All 0	R/W	Lower Up-Counter

- DACNTL

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	UC8 to UC13	All 0	R/W	Upper Up-Counter
1	—	1	R/W	Reserved This bit is 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. When changing the register to be accessed, set this bit in advance. 0: DADRA and DADRB can be accessed 1: DACR and DACNT can be accessed

9.3.2 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. DADR cannot be accessed in 8-bit units. DADR should always be accessed in 16-bit units. For details, see section 9.4, Bus Master Interface.

- DADRA

Bit	Bit Name	Initial Value	R/W	Description
15	DA13	1	R/W	D/A Data 13 to 0 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 fixed. A channel can be operated with 12-bit accuracy by fixing DA0 and DA1 to 0. The lower two data bits are not compared with UC12 and UC13 in 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/W	Reserved This bit is 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
14	DA12	1	R/W	Set a digital value to be converted to an analog value.
13	DA11	1	R/W	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 fixed.
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	A channel can be operated with 12-bit accuracy by fixing DA0 and DA1 to 0. The lower two data bits are not compared with UC12 and UC13 in DACNT.
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	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. When changing the register to be accessed, set this bit in advance. 0: DADRA and DADRB can be accessed 1: DACR and DACNT can be accessed

9.3.3 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 DACNT. 0: DACNT operates as a 14-bit up-counter 1: DACNT halts at H'0003
5	—	1	—	Reserved These bits are always read as 1 and cannot be modified.
4	—	1	—	
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.

9.3.4 Peripheral Clock Select Register (PCSR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	PWCKXC	0	R/W	PWMX Clock Select
6	PWCKXB	0	R/W	Select a clock cycle with the CKS bit in DACR of the PWMX being 1. See table 9.2.
5	PWCKXA	0	R/W	
4 to 2	—	0	R/W	Reserved The initial value should not be changed.
1	PWCKB	0	R/W	PWM Clock Select
0	PWCKA	0	R/W	

Table 9.2 Clock Selection 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

9.4 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.

(1) 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 to the register.

(2) 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 in 16-bit units 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 9.3 Access Method for Reading/Writing 16-Bit Registers

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

[Legend]

○: Indicates the allowed access. Word access includes continuous access to upper bytes and lower bytes in that order.

×: The result is not guaranteed in that access.

9.5 Operation

A PWM waveform like the one shown in figure 9.2 is output from the PWX pin. Data 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 data in DADR value corresponds to the total width (T_H) of the high (1) output pulses. Figures 9.3 and 9.4 show the types of output waveform.

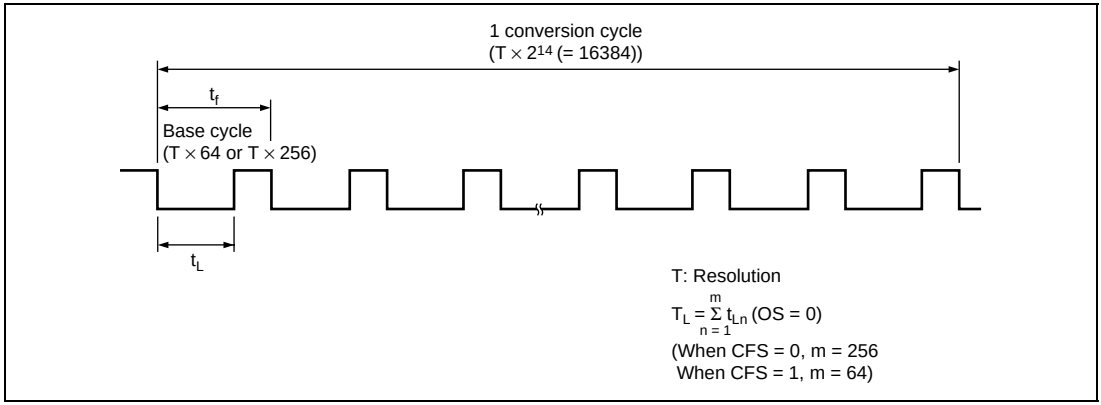


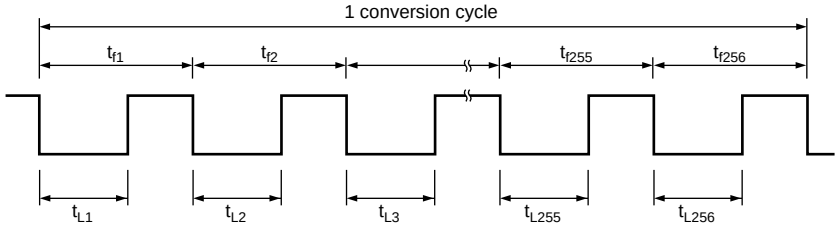
Figure 9.2 PWMX (D/A) Operation

Table 9.4 summarizes the relationships between the CKS and CFS bit settings and the resolution, base cycle, and conversion cycle. The PWM output remains fixed unless the contents of DADR are at least a certain minimum value. The relationship between the OS bit and the output waveform is shown in figures 9.3 and 9.4.

Table 9.4 Settings and Operation (Examples when ϕ = 20 MHz)

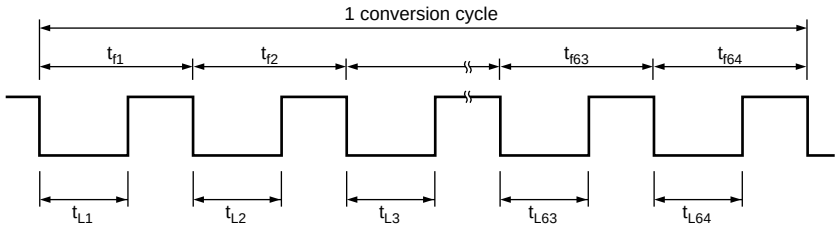
CKS	Resolution*2 T (µs)	CFS	Base Cycle (µs)	Conver-sion Cycle (µs)	TL/TH (OS = 0/OS = 1)	Fixed DADR Bits: Conversion Accuracy (Bits)	Bit Data DA3	DA2	DA1	DA0	Conversion Cycle*1 (µs)
0	0.05	0	3.2	819.2	Always low/high output	14					819.2
					DA13 to DA0 = H'0000 to H'00FF	12			0	0	204.8
					(Data value) × T						
					DA13 to DA0 = H'0100 to H'3FFF	10	0	0	0	0	51.2
		1	12.8	819.2	Always low/high output	14					819.2
					DA13 to DA0 = H'0000 to H'003F	12			0	0	204.8
					(Data value) × T						
					DA13 to DA0 = H'0040 to H'3FFF	10	0	0	0	0	51.2
1	0.1	0	6.4	1638.4	Always low/high output	14					1638.4
					DA13 to DA0 = H'0000 to H'00FF	12			0	0	409.6
					(Data value) × T						
					DA13 to DA0 = H'0100 to H'3FFF	10	0	0	0	0	102.4
		1	25.6	1638.4	Always low/high output	14					1638.4
					DA13 to DA0 = H'0000 to H'003F	12			0	0	409.6
					(Data value) × T						
					DA13 to DA0 = H'0040 to H'3FFF	10	0	0	0	0	102.4

Notes: 1. Indicates the conversion cycle when specific bits in DADR are fixed.
2. Indicates the resolution when PWCKXC = PWCKXB = PWCKXA = 0.



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

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 9.3 Output Waveform (OS = 0, DADR Corresponds to T_L)

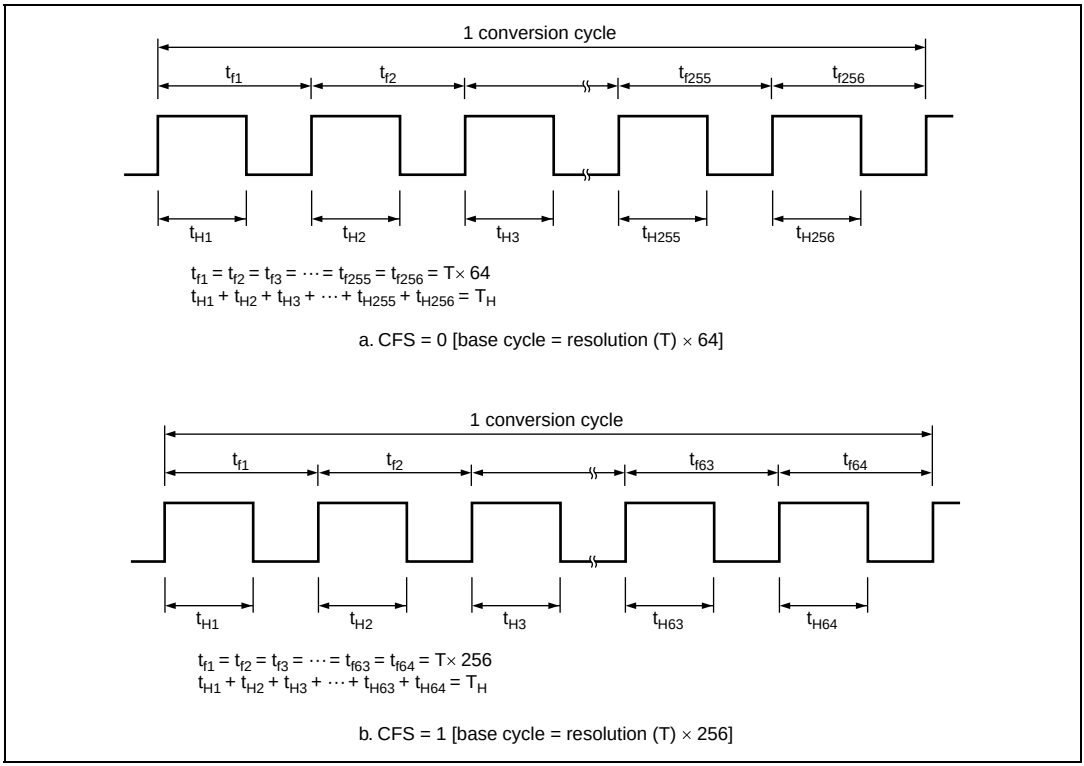


Figure 9.4 Output Waveform (OS = 1, DADR Corresponds to T_H)

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 9.5.

Table 9.5 lists the locations of the additional pulses.

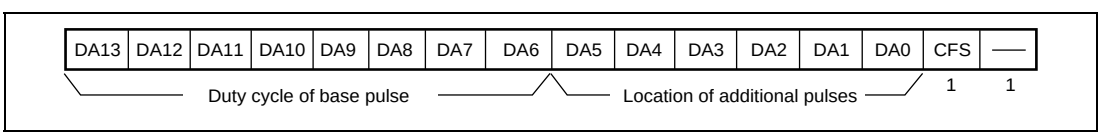


Figure 9.5 D/A Data Register Configuration when CFS = 1

In this example, DADR = H'0207 (B'0000 0010 0000 0111). The output waveform is shown in figure 9.6. 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 9.5. Thus, an additional pulse of 1/256 × (T) is to be added to the base pulse.

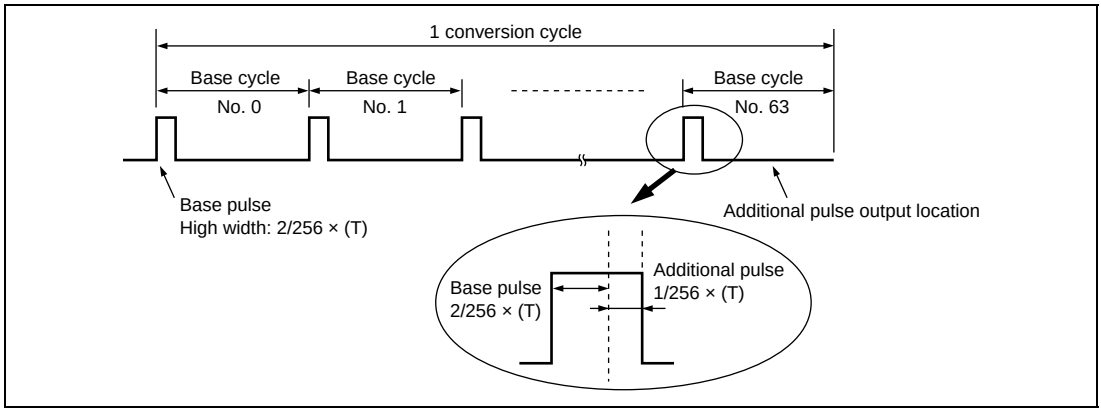
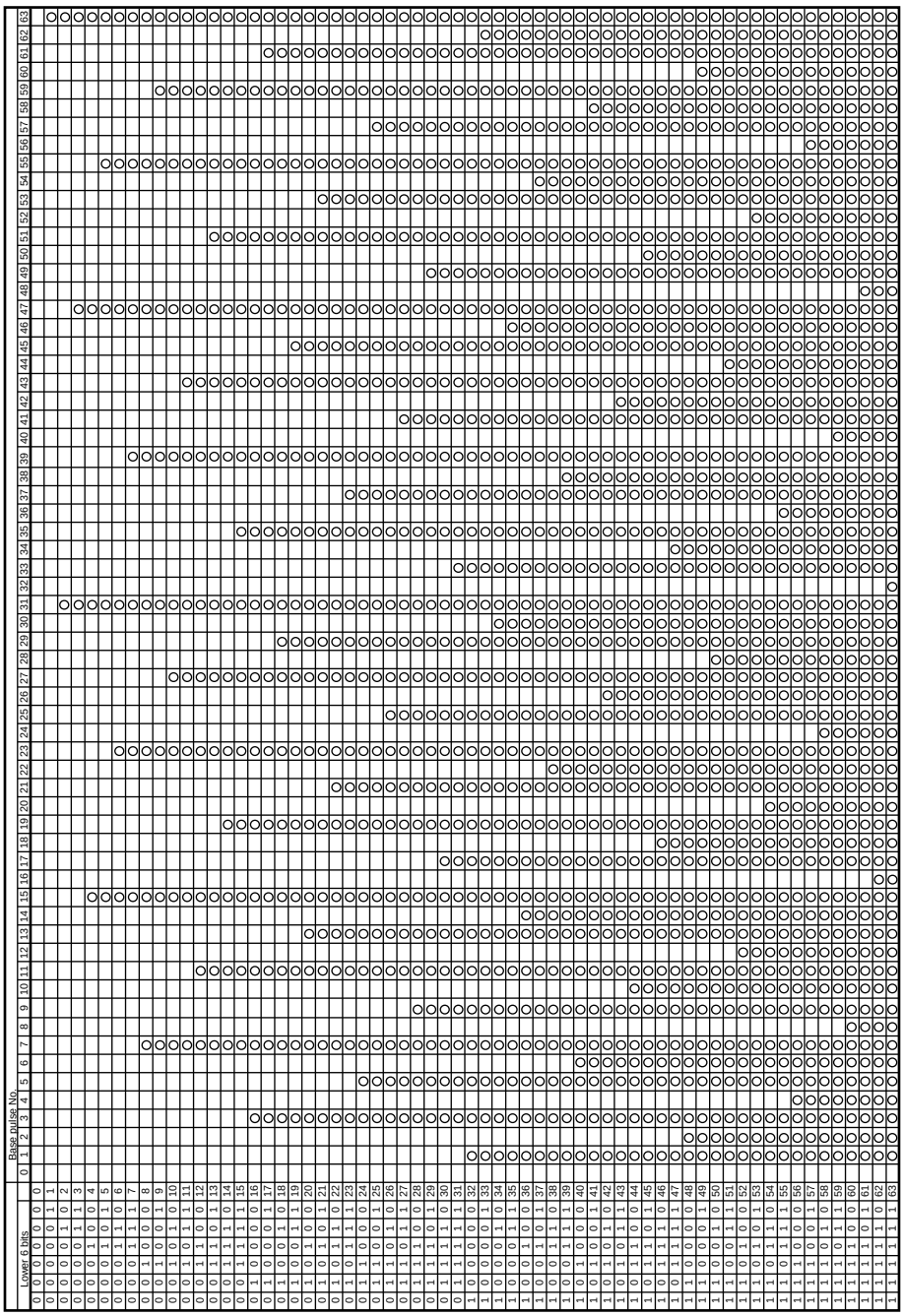


Figure 9.6 Output Waveform when DADR = H'0207 (OS = 1)

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 9.5 Locations of Additional Pulses Added to Base Pulse (when CFS = 1)



Section 10 16-Bit Free-Running Timer (FRT)

This LSI has an on-chip 16-bit free-running timer (FRT) with two channels. 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.

10.1 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 captures

  The rising or falling edge can be selected.

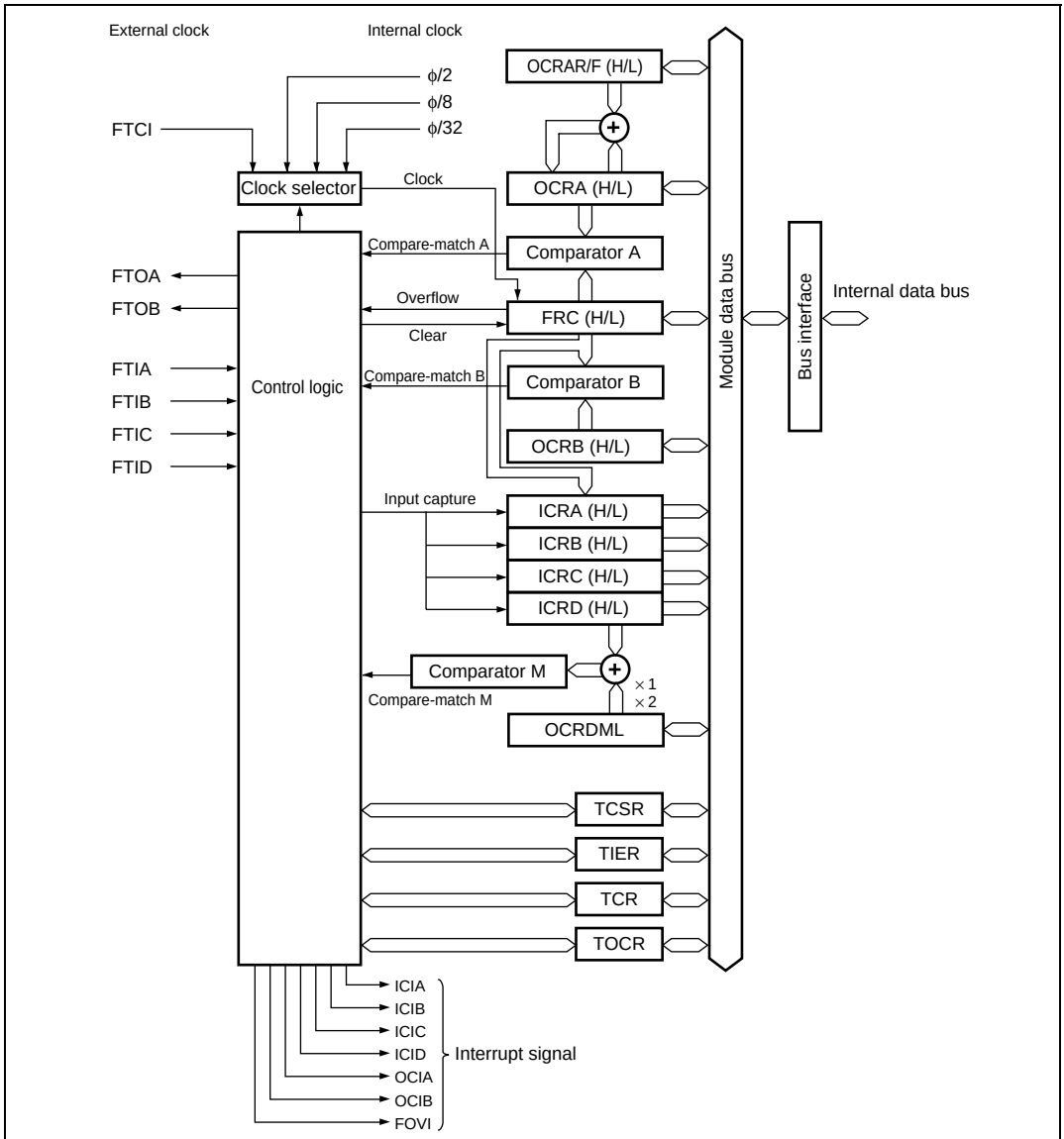
  Buffer modes can be specified.
- Counter clearing

  The free-running counters can be cleared on compare-match A.
- Fourteen independent interrupts

  Two compare-match interrupts, four input capture interrupts, and one overflow interrupt can be requested independently for each channel.
- 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 10.1 shows a block diagram of the FRT.



[Legend]

OCRA, OCRB:	Output compare registers A, B (16-bit)
OCRAR, OCRAF:	Output compare registers 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 10.1 Block Diagram of 16-Bit Free-Running Timer

10.2 Input/Output Pins

Table 10.1 lists the FRT input and output pins.

Table 10.1 Pin Configuration

Channel	Name	Symbol	I/O	Function
0	Counter clock input pin	FTCI_0	Input	FRC counter clock input
	Output compare A output pin	FTOA_0	Output	Output compare A output
	Output compare B output pin	FTOB_0	Output	Output compare B output
	Input capture A input pin	FTIA_0	Input	Input capture A input
	Input capture B input pin	FTIB_0	Input	Input capture B input
	Input capture C input pin	FTIC_0	Input	Input capture C input
	Input capture D input pin	FTID_0	Input	Input capture D input
1	Counter clock input pin	FTCI_1	Input	FRC counter clock input
	Output compare A output pin	FTOA_1	Output	Output compare A output
	Output compare B output pin	FTOB_1	Output	Output compare B output
	Input capture A input pin	FTIA_1	Input	Input capture A input
	Input capture B input pin	FTIB_1	Input	Input capture B input
	Input capture C input pin	FTIC_1	Input	Input capture C input
	Input capture D input pin	FTID_1	Input	Input capture D input

Note: Channels 0 and 1 are omitted in this manual.

10.3 Register Descriptions

The FRT has the following registers for each channel.

- 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_0 (OCRA_1) and OCRB_0 (OCRB_1) share the same address. Register selection is controlled by the OCRS bit in TOCR_0 (TOCR_1). ICRA_0 (ICRA_1), ICRB_0 (ICRB_1), and ICRC_0 (ICRC_1) share the same addresses with OCRAR_0 (OCRAR_1), OCRAF_0 (OCRAF_1), and OCRDM_0 (OCRDM_1). Register selection is controlled by the ICRS bit in TOCR_0 (TOCR_1).

10.3.1 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 OVF flag 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.

10.3.2 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 OCFA or OCFB flag in TCSR is set to 1. 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.

10.3.3 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 ICRA to ICRD. At the same time, the ICFA to ICFD flags in TCSR are set to 1. The FRC contents are transferred to ICR regardless of the value of ICF. The input capture edge is selected by the IEDGA to IEDGD bits in TCR.

ICRC and ICRD can be used as ICRA and ICRB buffer registers, respectively, by means of the BUFEA and BUFEB bits in TCR.

For example, if an input capture occurs when ICRA is specified as the input capture register and ICRC is specified as the ICRA buffer register, the FRC contents are transferred to ICRA, and then transferred to the buffer register ICRC. In this case, setting the IEDGA and IEDGC bits in TCR to the different values enables the rising- or falling-edge sensing to be specified.

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.

10.3.4 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 first 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 ϕ/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.

10.3.5 Output Compare Register DM (OCRDM)

OCRDM is a 16-bit readable/writable register in which the upper eight bits are fixed to 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.

10.3.6 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 an interrupt request (ICIA) by the ICFA flag when the ICFA flag 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 an interrupt request (ICIB) by the ICFB flag when the ICFB flag 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 an interrupt request (ICIC) by the ICFC flag when the ICFC flag 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 an interrupt request (ICID) by the ICFD flag when the ICFD flag 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 an interrupt request (OCIA) by the OCFA flag when the OCFA flag 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 an interrupt request (OCIB) by the OCFB flag when the OCFB flag 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 an interrupt request (FOVI) by the OVF flag when the OVF flag in TCSR is set to 1. 0: FOVI requested by OVF is disabled 1: FOVI requested by OVF is enabled
0	—	0	R	Reserved This bit is always read as 1 and cannot be modified.

10.3.7 Timer Control/Status Register (TCSR)

TCSR selects whether the counter operates or not and controls 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 new FRC value has been transferred to ICRA by an input capture signal and the old ICRA value has been moved into ICRC. [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 new FRC value has been transferred to ICRB by an input capture signal and the old ICRB value has been moved into ICRD. [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 occurs [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 occurs [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 Selects whether 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.

10.3.8 Timer Control Register (TCR)

TCR selects the rising or falling edge of the input capture signals, specifies the buffer operation, 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	R/W	Select clock source for FRC. 00: Count on internal clock ϕ/2 01: Count on internal clock ϕ/8 10: Count on internal clock ϕ/32 11: Count on rising edge of external clock input signal (FTCI)

10.3.9 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 normal operating mode or in operating mode using OCRDM. 0: Normal operating mode is specified for ICRD 1: Operating mode using OCRDM is specified for ICRD
6	OCRAMS	0	R/W	Output Compare A Mode Select Specifies whether OCRA is used in normal operating mode or in operating mode using OCRAR and OCRAF. 0: Normal operating mode is specified for OCRA 1: 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. When this address is accessed, the OCRS bit selects which register is accessed. 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

10.4 Operation

10.4.1 Pulse Output

Figure 10.2 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.

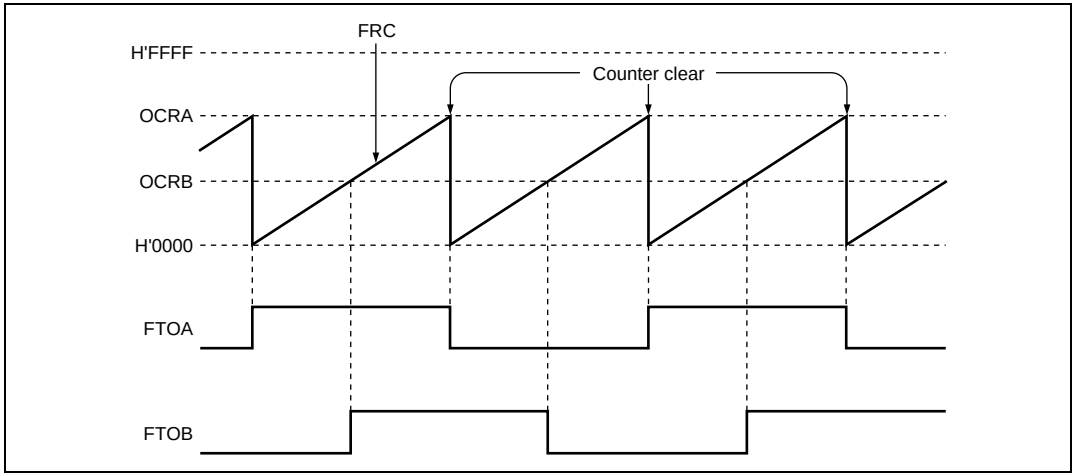


Figure 10.2 Example of Pulse Output

10.5 Operation Timing

10.5.1 FRC Increment Timing

Figure 10.3 shows the FRC increment timing with an internal clock source. Figure 10.4 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).

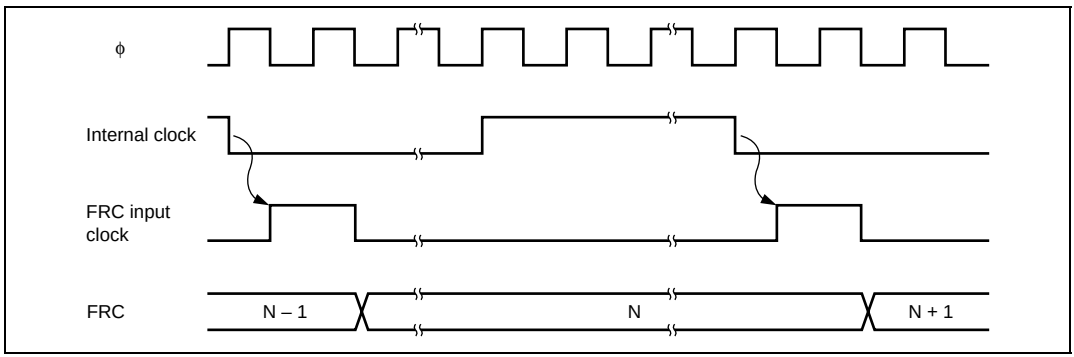


Figure 10.3 Increment Timing with Internal Clock Source

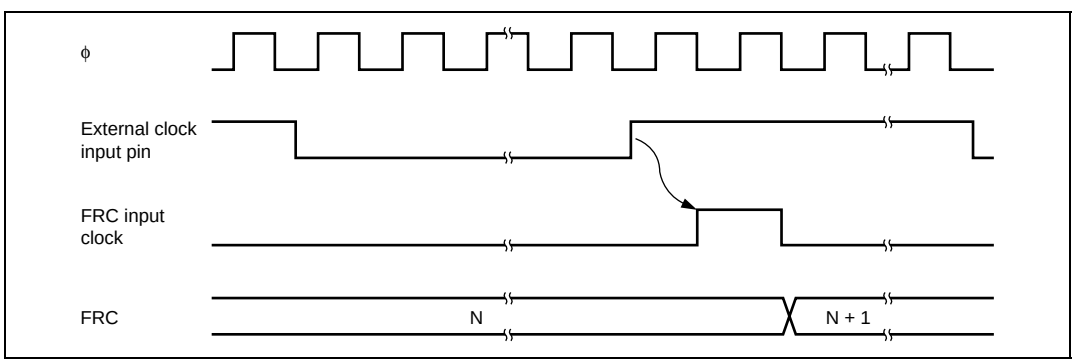


Figure 10.4 Increment Timing with External Clock Source

10.5.2 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 10.5 shows the timing of this operation for compare-match A.

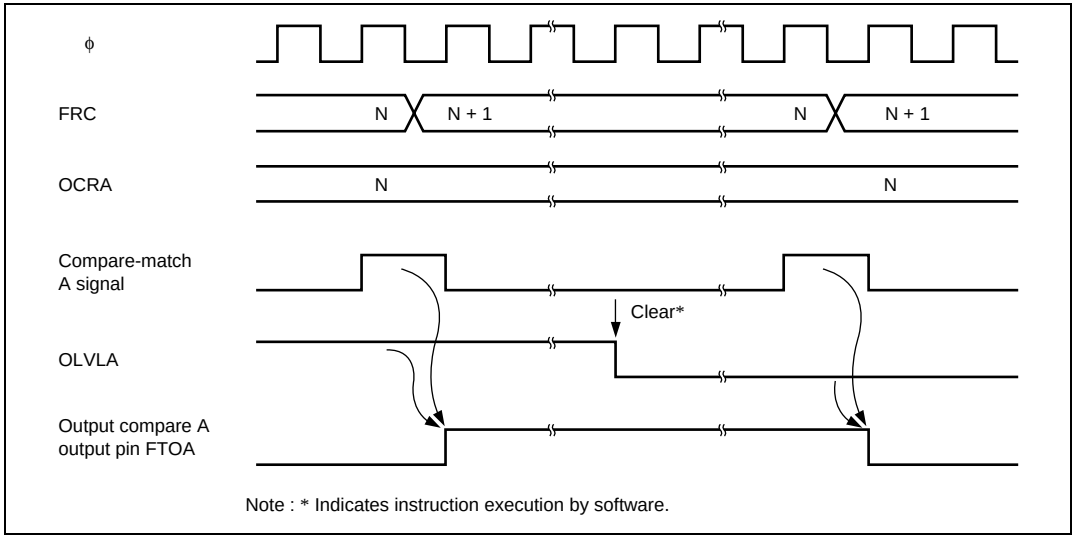


Figure 10.5 Timing of Output Compare A Output

10.5.3 FRC Clear Timing

FRC can be cleared when compare-match A occurs. Figure 10.6 shows the timing of this operation.

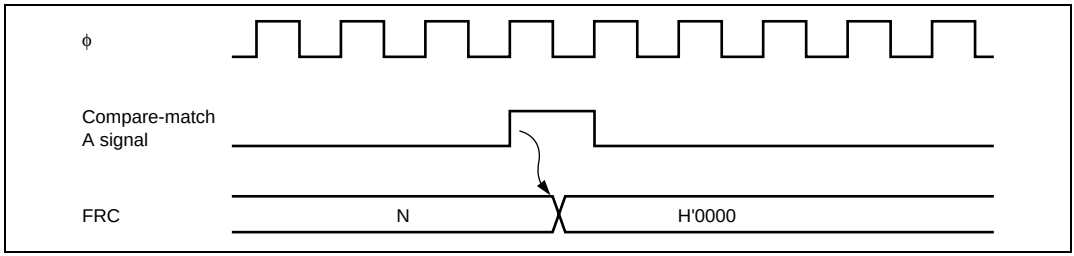


Figure 10.6 Clearing of FRC by Compare-Match A Signal

10.5.4 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 10.7 shows the usual input capture timing when the rising edge is selected.

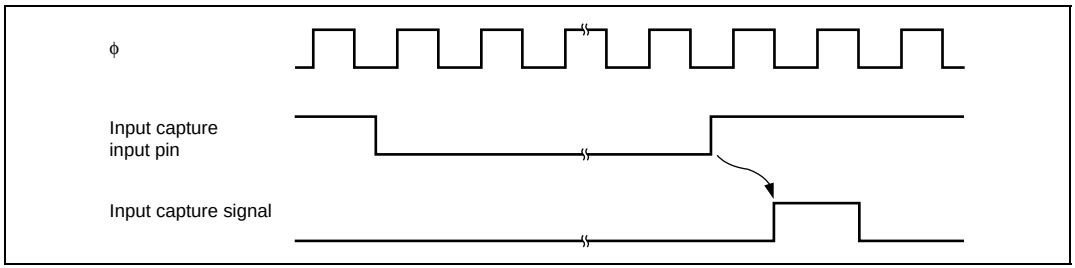


Figure 10.7 Timing of Input Capture Input Signal (Usual Case)

If the corresponding input capture signal is input when ICRA to ICRD are read, the input capture signal is delayed by one system clock (ϕ). Figure 10.8 shows the timing for this case.

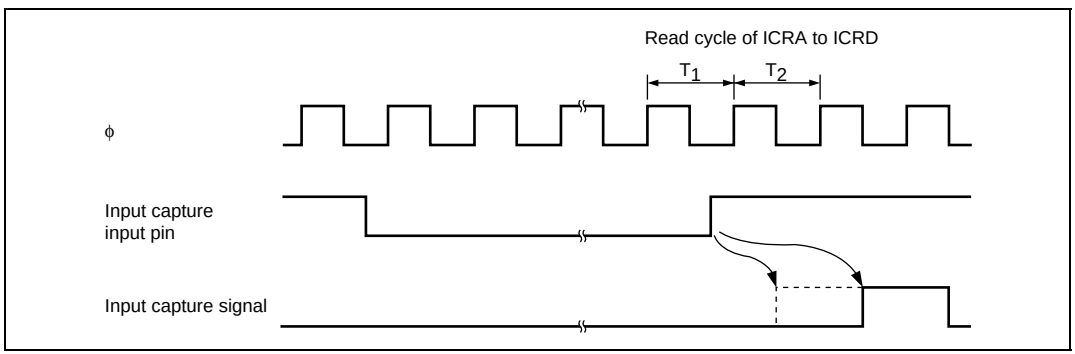


Figure 10.8 Timing of Input Capture Input Signal (When ICRA to ICRD are Read)

10.5.5 Buffered Input Capture Input Timing

ICRC and ICRD can operate as buffers for ICRA and ICRB, respectively. Figure 10.9 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 = 1 and IEDGC = 0, or IEDGA = 0 and IEDGC = 1), so that input capture is performed on both the rising and falling edges of FTIA.

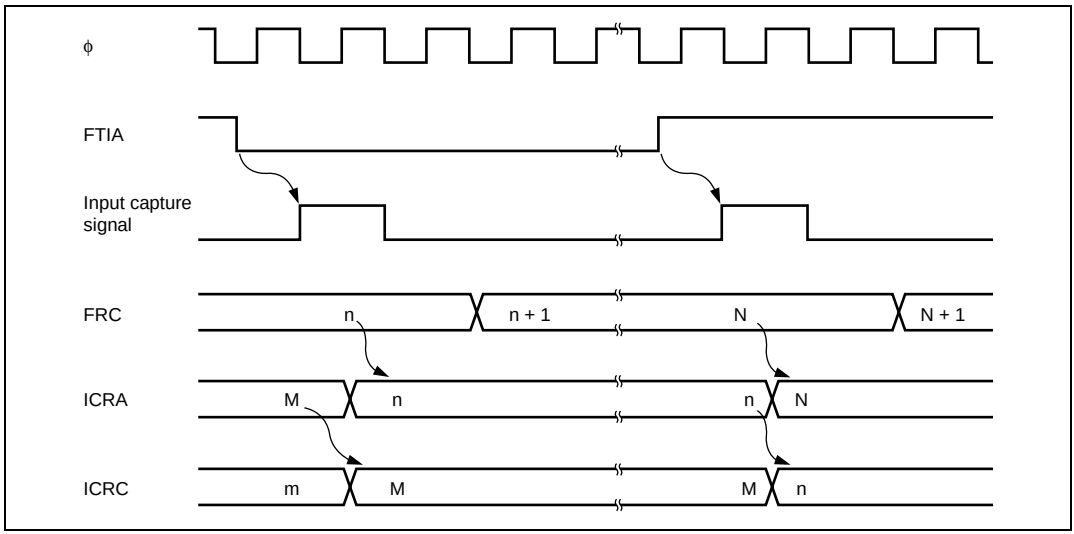


Figure 10.9 Buffered Input Capture Timing

Even when ICRC or ICRD is used as a buffer register, its input capture flag is set according to the selected edge of its input capture signal. For example, if ICRC is used as the ICRA buffer register, when the edge transition selected by the IEDGC bit occurs on the FTIC input capture line, the ICFC flag 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 occurs, input capture is delayed by one system clock (φ). Figure 10.10 shows the timing when BUFEA = 1.

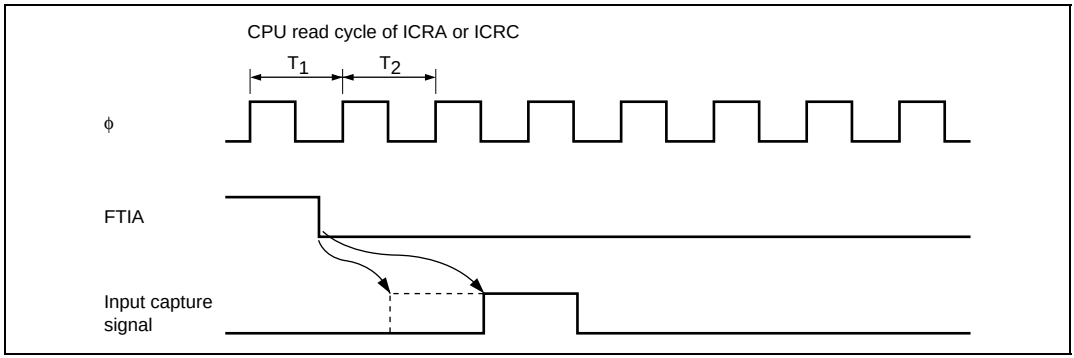


Figure 10.10 Buffered Input Capture Timing (BUFEA = 1)

10.5.6 Timing of Input Capture Flag Setting

The input capture flag, ICFA, ICFB, ICFC, or ICFD, is set to 1 by the input capture signal. The FRC value is simultaneously transferred to the corresponding input capture register (ICRA, ICRB, ICRC, or ICRD). Figure 10.11 shows the timing of setting the ICFA to ICFD flags.

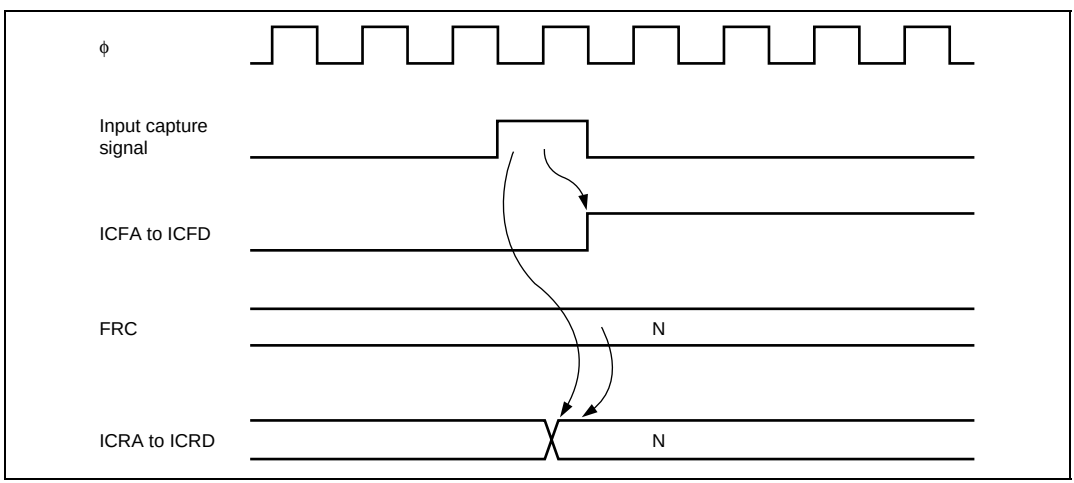


Figure 10.11 Timing of Input Capture Flags (ICFA to ICFD) Setting

10.5.7 Timing of Output Compare Flag 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 values match, the compare-match signal is not generated until the next cycle of the clock source. Figure 10.12 shows the timing of setting the OCFA or OCFB flag.

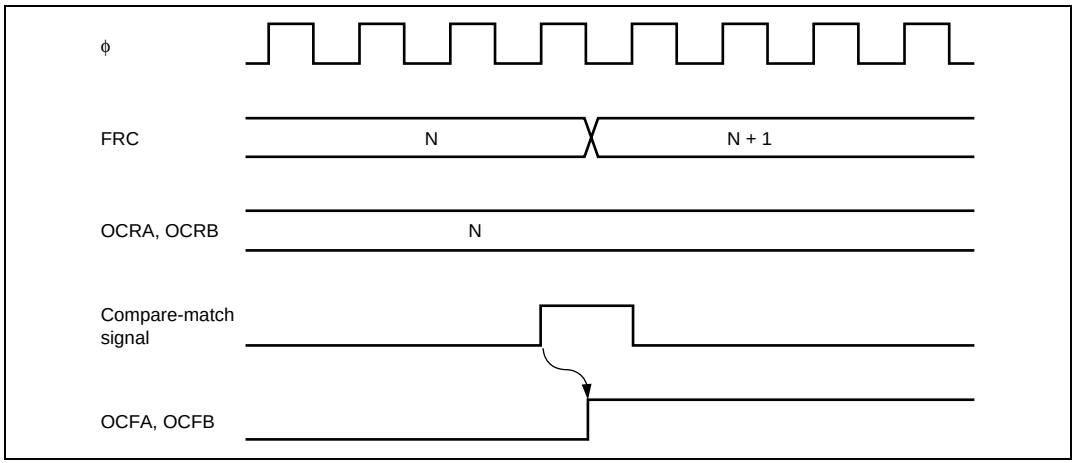


Figure 10.12 Timing of Output Compare Flag (OCFA or OCFB) Setting

10.5.8 Timing of Overflow Flag Setting

The OVF flag is set to 1 when FRC overflows (changes from H'FFFF to H'0000). Figure 10.13 shows the timing of setting the OVF flag.

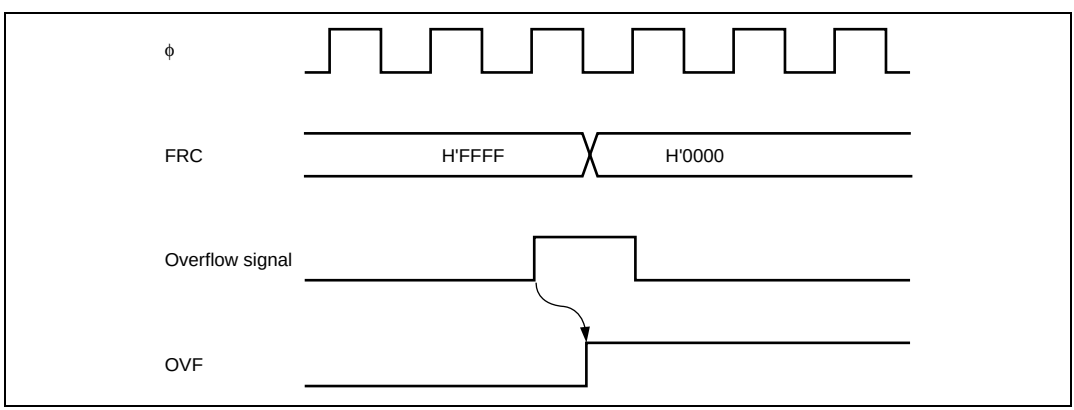


Figure 10.13 Timing of OVF Flag Setting

10.5.9 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 10.14 shows the OCRA write timing.

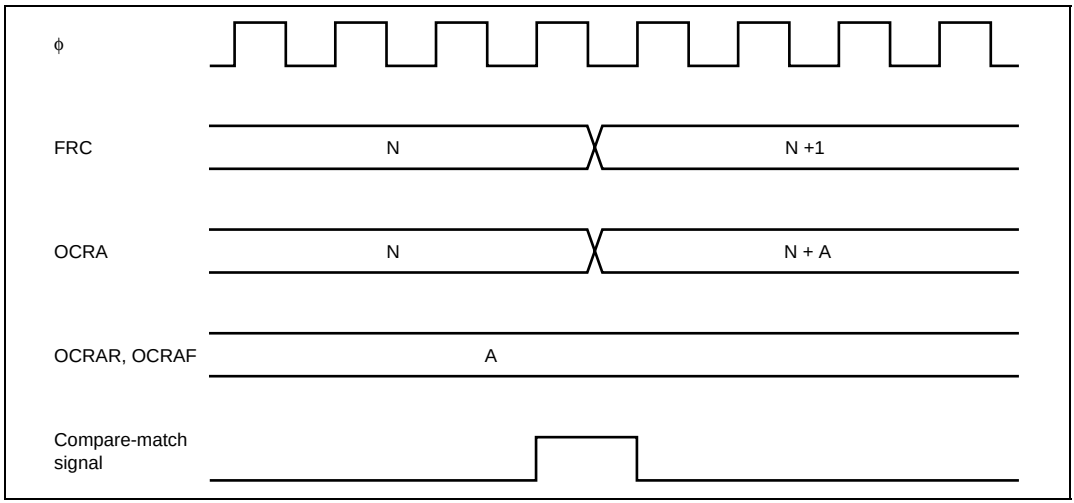


Figure 10.14 OCRA Automatic Addition Timing

10.5.10 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 10.15 shows the timing of setting the mask signal. Figure 10.16 shows the timing of clearing the mask signal.

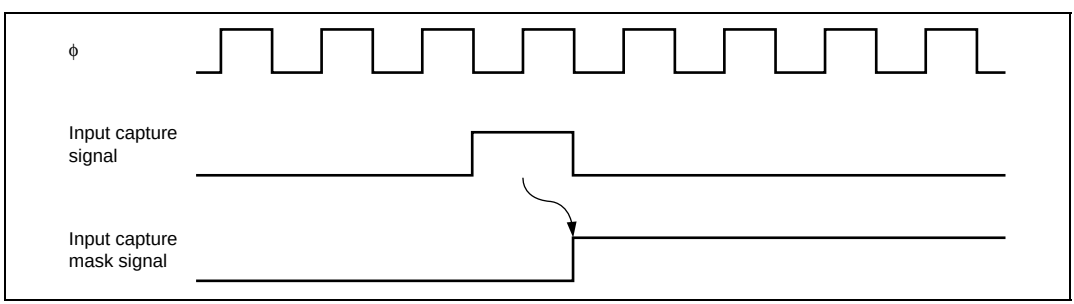


Figure 10.15 Timing of Input Capture Mask Signal Setting

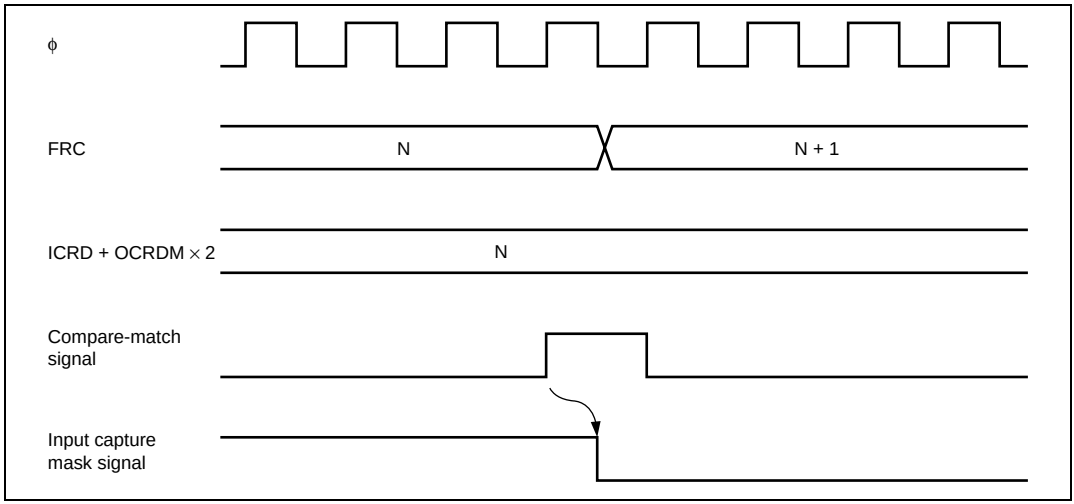


Figure 10.16 Timing of Input Capture Mask Signal Clearing

10.6 Interrupt Sources

The FRT can request seven interrupts: ICIA to ICID, OCIA, OCIB, and FOVI. Each interrupt can be enabled or disabled by an interrupt enable bit in TIER. Independent signals are sent to the interrupt controller for each interrupt. Table 10.2 lists the sources and priorities of these interrupts.

Table 10.2 FRT Interrupt Sources

Channel	Interrupt	Interrupt Source	Interrupt Flag	Priority
0	ICIA_0	Input capture of ICRA	ICFA	High
	ICIB_0	Input capture of ICRB	ICFB	
	ICIC_0	Input capture of ICRC	ICFC	
	ICID_0	Input capture of ICRD	ICFD	
	OCIA_0	Compare match of OCRA	OCFA	
	OCIB_0	Compare match of OCRB	OCFB	
	FOV_0	Overflow of FRC	OVF	
1	ICIA_1	Input capture of ICRA	ICFA	
	ICIB_1	Input capture of ICRB	ICFB	
	ICIC_1	Input capture of ICRC	ICFC	
	ICID_1	Input capture of ICRD	ICFD	
	OCIA_1	Compare match of OCRA	OCFA	
	OCIB_1	Compare match of OCRB	OCFB	
	FOV_1	Overflow of FRC	OVF	Low

10.7 Usage Notes

10.7.1 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 10.17 shows the timing for this type of conflict.

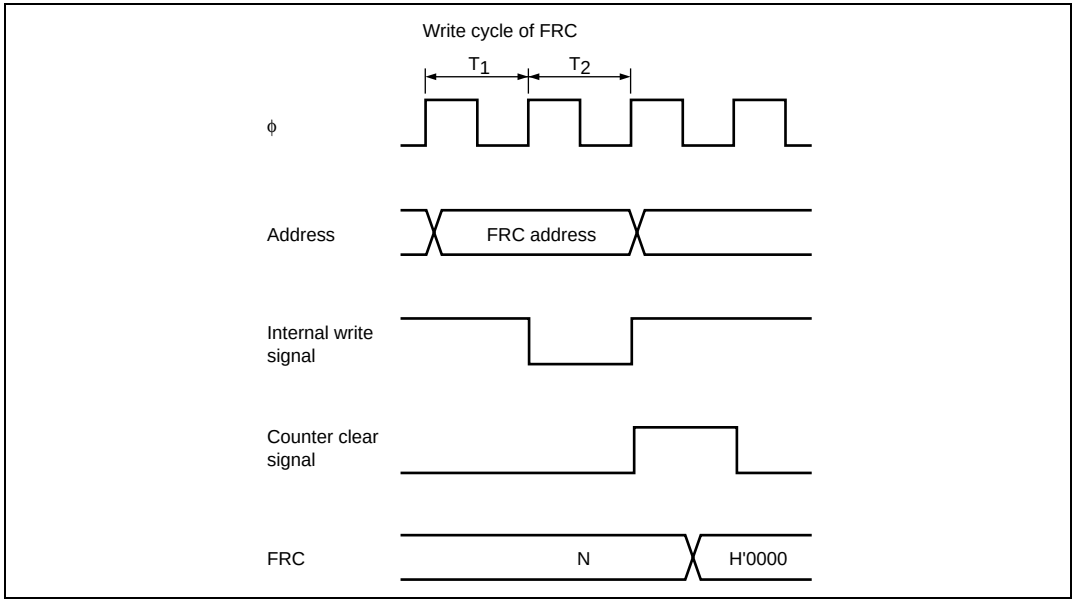


Figure 10.17 FRC Write-Clear Conflict

10.7.2 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 10.18 shows the timing for this type of conflict.

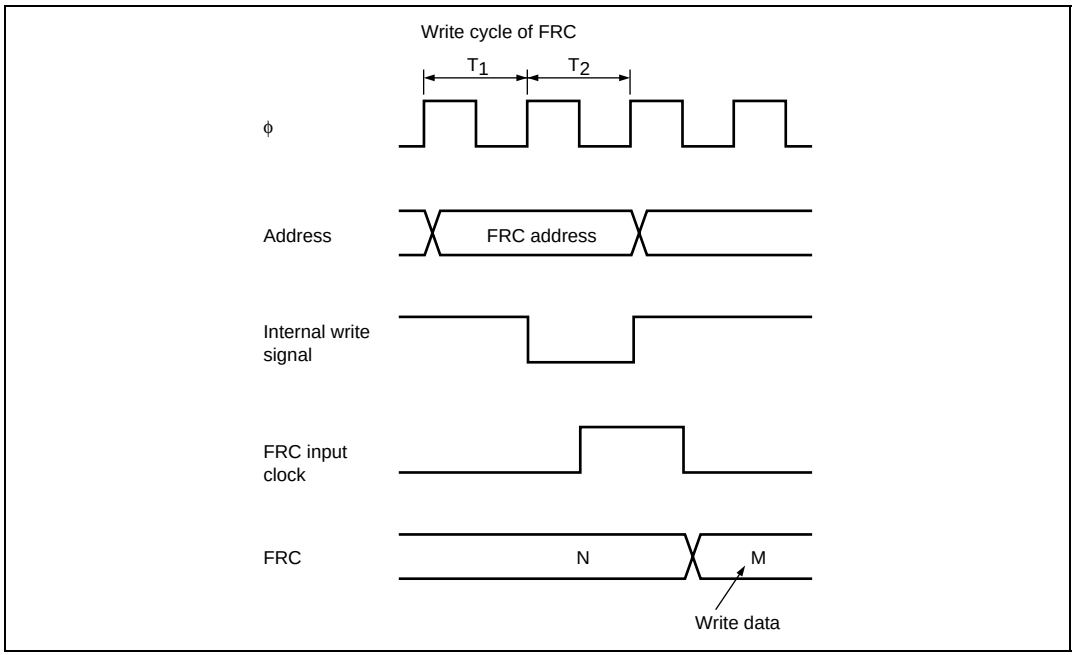


Figure 10.18 FRC Write-Increment Conflict

10.7.3 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 10.19 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 10.20 shows the timing for this type of conflict.

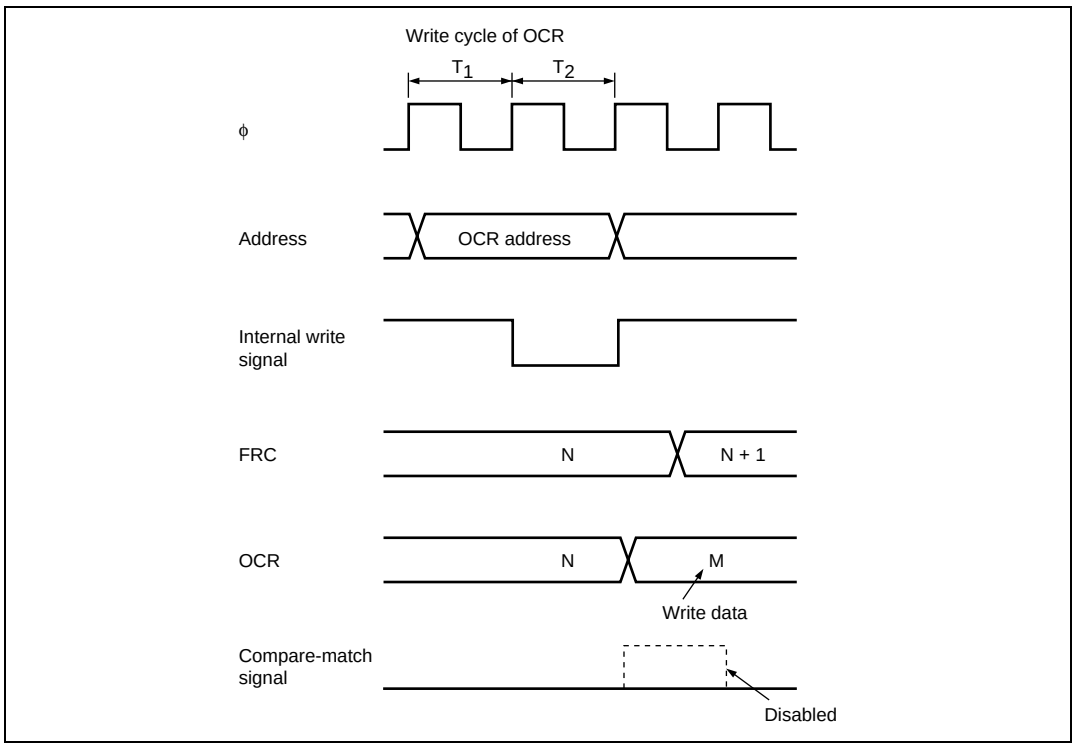


Figure 10.19 Conflict between OCR Write and Compare-Match (When Automatic Addition Function is not Used)

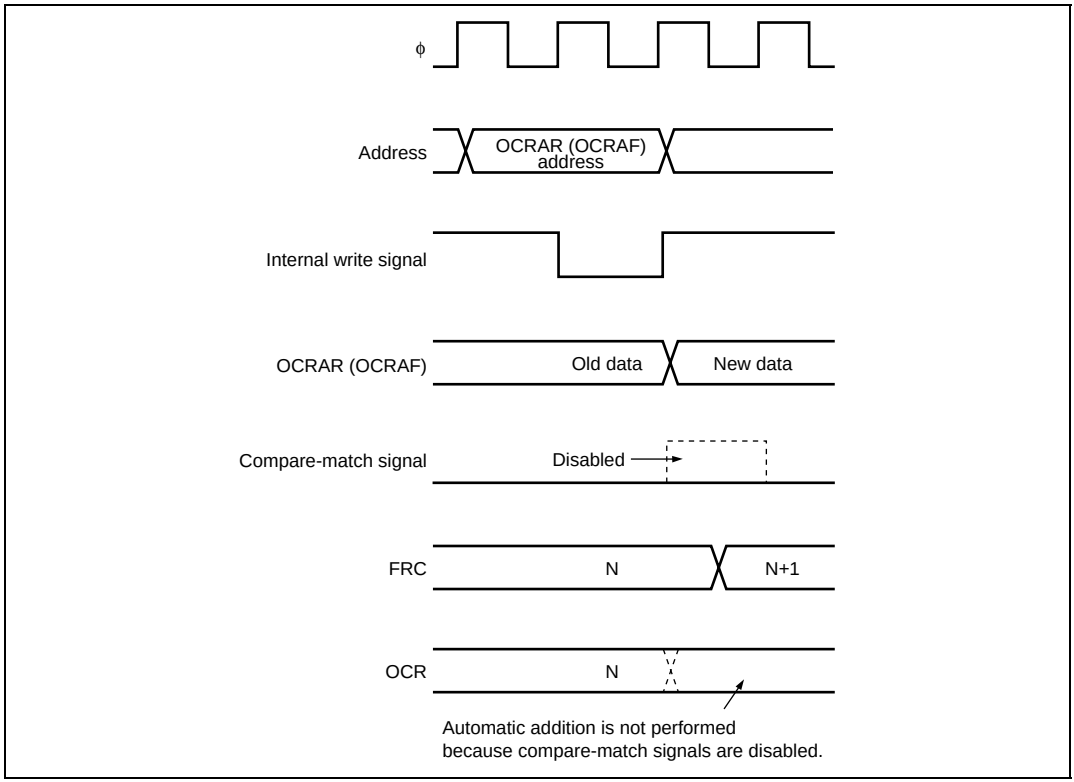


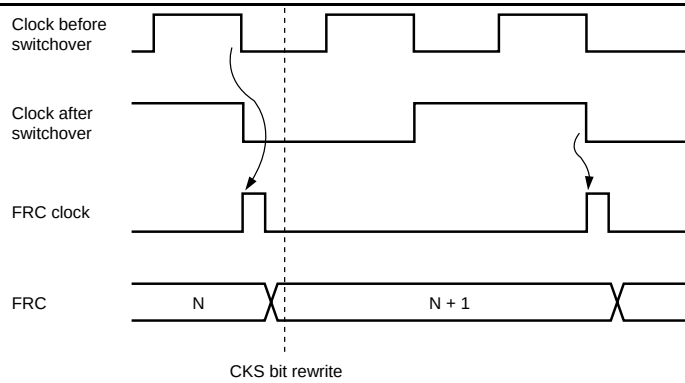
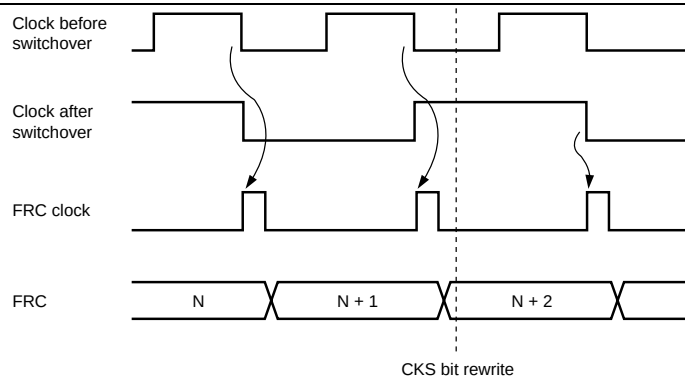
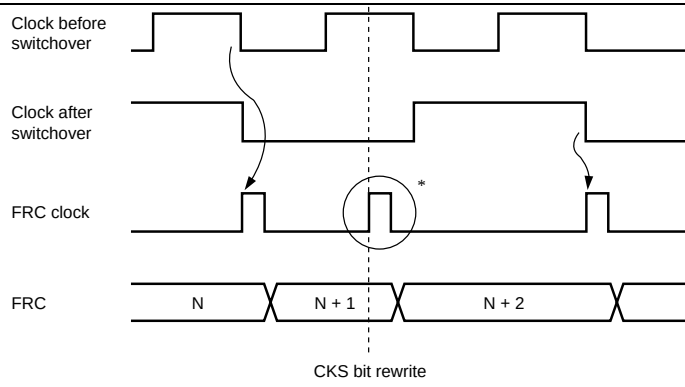
Figure 10.20 Conflict between OCRAR/OCRAF Write and Compare-Match (When Automatic Addition Function is Used)

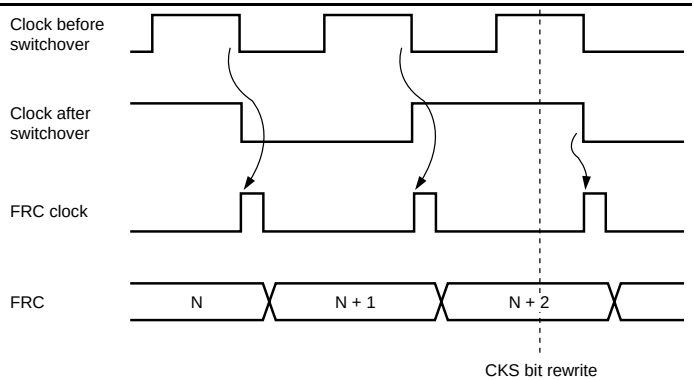
10.7.4 Switching of Internal Clock and FRC Operation

When the internal clock is changed, the changeover may cause FRC to be incremented. This depends on the timing at which the clock is switched (bits CKS1 and CKS0 are rewritten). Table 10.3 shows the relationship between the timing and the FRC operation.

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 10.3, 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 cause FRC to be incremented.

Table 10.3 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.

Section 11 8-Bit Timer (TMR)

This LSI has an on-chip 2-system 8-bit timer module (TMR0 and TMR1) with two channels operating on the basis of an 8-bit counter. In addition to external event counting, 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.

This LSI also has a similar on-chip 2-system 8-bit timer module (TMRY and TMRX) with two channels.

11.1 Features

- Selection of clock sources

  TMR0, TMR1: The counter input clock can be selected from six internal clocks and an external clock

  TMRY, TMRX: The counter input clock can be selected from three internal clocks and an external clock
- Selection of three ways to clear the counters

  The counters can be cleared on compare-match A or 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 systems

  Cascading of TMR0 and TMR1:

  Operation as a 16-bit timer can be performed using the TMR0 as the upper half and TMR1 as the lower half (16-bit count mode).

  The TMR1 can be used to count TMR0 compare-match occurrences (compare-match count mode).

  Cascading of TMRY and TMRX:

  Operation as a 16-bit timer can be performed using the TMRY as the upper half and TMRX as the lower half (16-bit count mode).

  The TMRX can be used to count TMRY compare-match occurrences (compare-match count mode).
- Multiple interrupt sources for each channel

  TMR0, TMR1, and TMRY: Three types of interrupts: Compare-match A, compare-match B, and overflow

TMRX: Four types of interrupts: Compare-match A, compare-match B, overflow, and input capture

Figures 11.1 and 11.2 show block diagrams of 8-bit timers.

An input capture function is added to the TMRX.

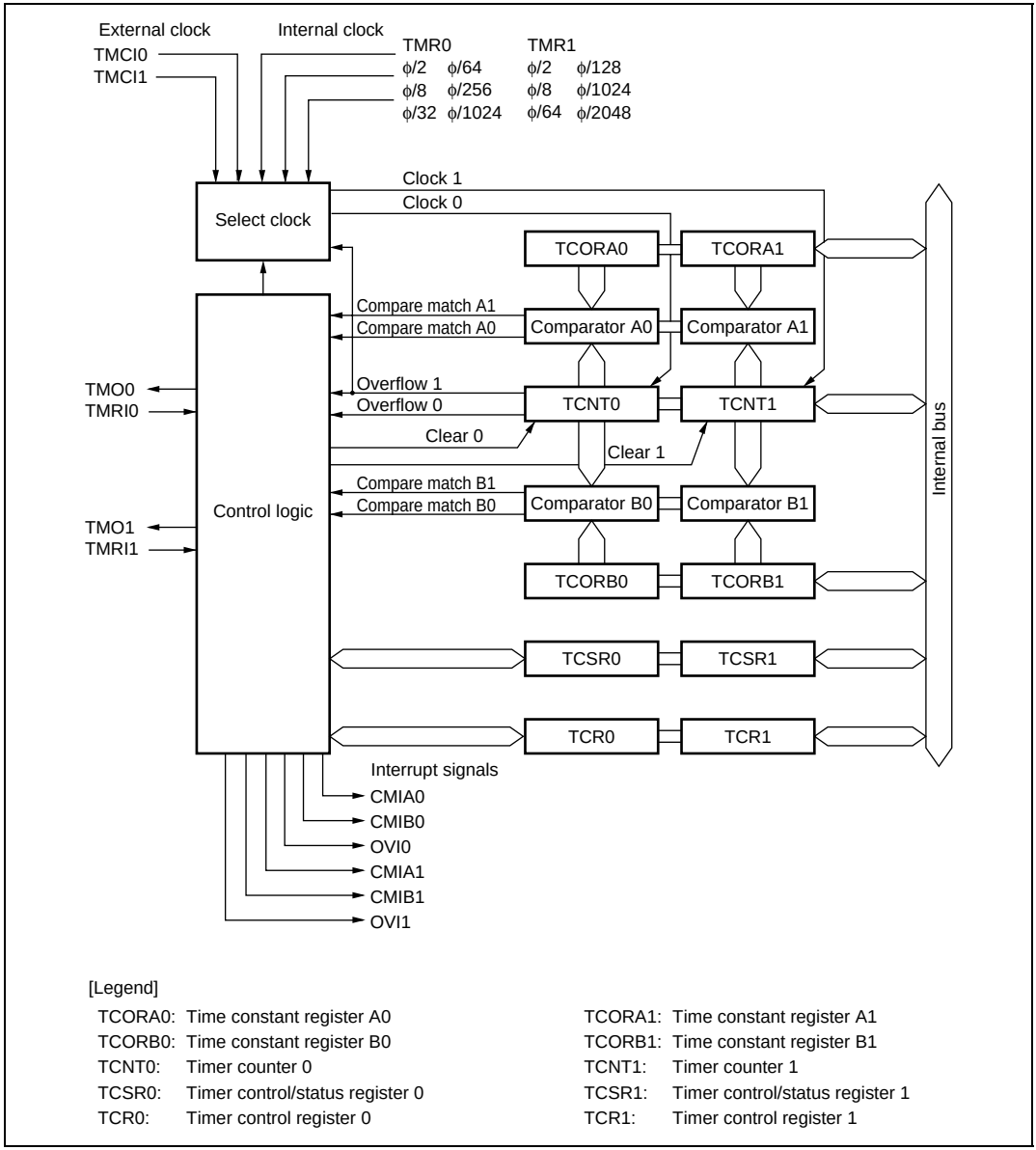
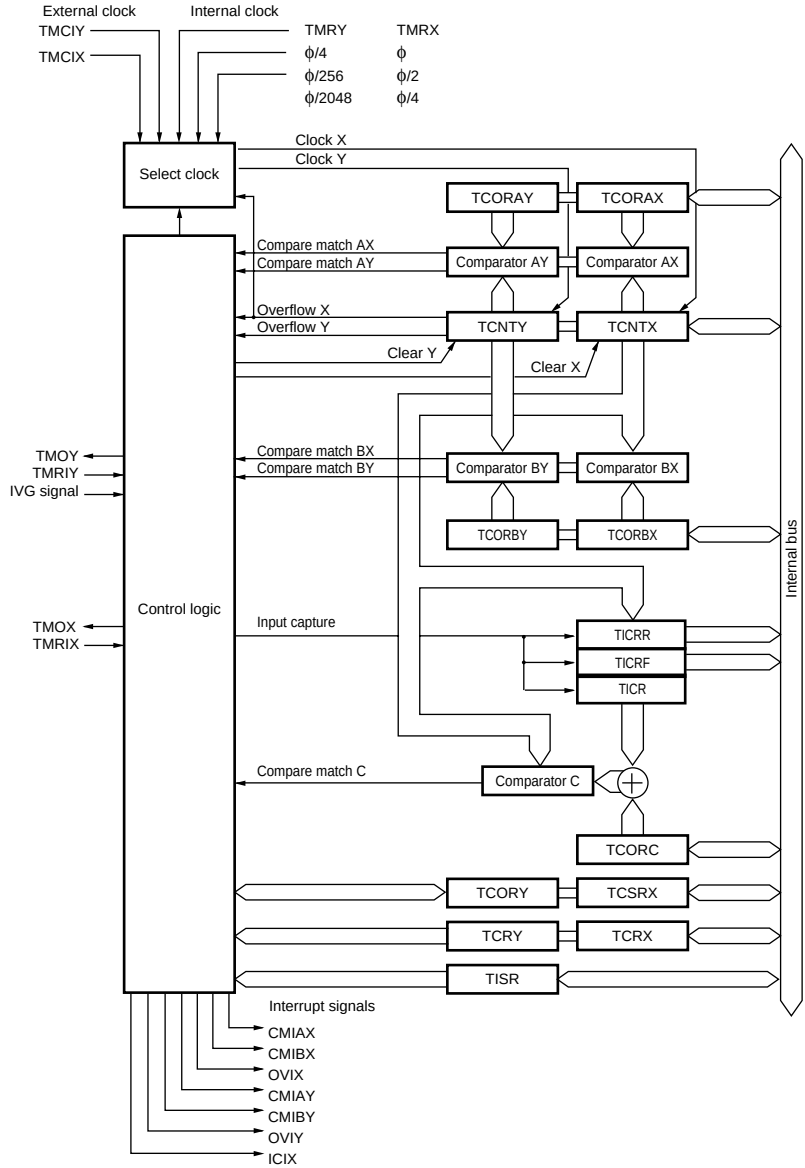


Figure 11.1 Block Diagram of 8-Bit Timer (TMR0 and TMR1)



[Legend]

TCORAY:	Time constant register AY	TCORAX:	Time constant register AX
TCORBY:	Time constant register BY	TCORBX:	Time constant register BX
TCNTY:	Timer counter Y	TCNTX:	Timer counter X
TCSRY:	Timer control/status register Y	TCSRX:	Timer control/status register X
TCRY:	Timer control register Y	TCRX:	Timer control register X
TISR:	Timer input select register	TICR:	Input capture register
		TCORC:	Time constant register
		TICRR:	Input capture register R
		TICRF:	Input capture register F

Figure 11.2 Block Diagram of 8-Bit Timer (TMRY and TMRX)

11.2 Input/Output Pins

Table 11.1 summarizes the input and output pins of the TMR.

Table 11.1 Pin Configuration

Channel	System	Name	Symbol	I/O	Function
0	TMR0	Timer output	TMO0_0	Output	Output controlled by compare-match
		Timer clock/reset input	TMI0_0/ ExTMI0_0	Input	External clock input (TMCI0)/external reset input (TMRI0) for the counter
	TMR1	Timer output	TMO1_0	Output	Output controlled by compare-match
		Timer clock/reset input	TMI1_0/ ExTMI1_0	Input	External clock input (TMCI1)/external reset input (TMRI1) for the counter
	TMRY	Timer output	TMOY_0	Output	Output controlled by compare-match
		Timer clock/reset input	TMIY_0/ ExTMIY_0	Input	External clock input (TMCIY)/external reset input (TMRIY) for the counter
	TMRX	Timer output	TMOX_0	Output	Output controlled by compare-match
		Timer clock/reset input	TMIX_0/ ExTMIX_0	Input	External clock input (TMCIX)/external reset input (TMRIX) for the counter

Channel	System	Name	Symbol	I/O	Function
1	TMR0	Timer output	TMO0_1	Output	Output controlled by compare-match
		Timer clock/reset input	TMI0_1/ ExTMI0_1	Input	External clock input (TMCI0)/external reset input (TMRI0) for the counter
	TMR1	Timer output	TMO1_1	Output	Output controlled by compare-match
		Timer clock/reset input	TMI1_1/ ExTMI1_1	Input	External clock input (TMCI1)/external reset input (TMRI1) for the counter
	TMRY	Timer output	TMOY_1	Output	Output controlled by compare-match
		Timer clock/reset input	TMIY_1/ ExTMIY_1	Input	External clock input (TMCIY)/external reset input (TMRIY) for the counter
	TMRX	Timer output	TMOX_1	Output	Output controlled by compare-match
		Timer clock/reset input	TMIX_1/ ExTMIX_1	Input	External clock input (TMCIX)/external reset input (TMRIX) for the counter

11.3 Register Descriptions

The TMR has the following registers. For details on the timer extended control register, see section 13.3.5, Timer Extended Control Register (TECR).

TMR0_0:

- Timer counter 0_0 (TCNT0_0)
- Time constant register A0_0 (TCORA0_0)
- Time constant register B0_0 (TCORB0_0)
- Timer control register 0_0 (TCR0_0)
- Timer control/status register 0_0 (TCSR0_0)

TMR1_0:

- Timer counter 1_0 (TCNT1_0)
- Time constant register A1_0 (TCORA1_0)
- Time constant register B1_0 (TCORB1_0)
- Timer control register 1_0 (TCR1_0)
- Timer control/status register 1_0 (TCSR1_0)

TMRY_0:

- Timer counter Y_0 (TCNTY_0)
- Time constant register AY_0 (TCORAY_0)
- Time constant register BY_0 (TCORBY_0)
- Timer control register Y_0 (TCRY_0)
- Timer control/status register Y_0 (TCSRY_0)
- Timer input select register_0 (TISR_0)

TMRX_0:

- Timer counter X_0 (TCNTX_0)
- Time constant register AX_0 (TCORAX_0)
- Time constant register BX_0 (TCORBX_0)
- Timer control register X_0 (TCRX_0)
- Timer control/status register X_0 (TCSRX_0)
- Input capture register_0 (TICR_0)
- Time constant register_0 (TCORC_0)
- Input capture register R_0 (TICRR_0)
- Input capture register F_0 (TICRF_0)

TMR0_1:

- Timer counter 0_1 (TCNT0_1)
- Time constant register A0_1 (TCORA0_1)
- Time constant register B0_1 (TCORB0_1)
- Timer control register 0_1 (TCR0_1)
- Timer control/status register 0_1 (TCSR0_1)

TMR1_1:

- Timer counter 1_1 (TCNT1_1)
- Time constant register A1_1 (TCORA1_1)
- Time constant register B1_1 (TCORB1_1)
- Timer control register 1_1 (TCR1_1)
- Timer control/status register 1_1 (TCSR1_1)

TMRY_1:

- Timer counter Y_1 (TCNTY_1)
- Time constant register AY_1 (TCORAY_1)

- Time constant register BY_1 (TCORBY_1)
- Timer control register Y_1 (TCRY_1)
- Timer control/status register Y_1 (TCSRY_1)
- Timer input select register_1 (TISR_1)

TMRX_1:

- Timer counter X_1 (TCNTX_1)
- Time constant register AX_1 (TCORAX_1)
- Time constant register BX_1 (TCORBX_1)
- Timer control register X_1 (TCRX_1)
- Timer control/status register X_1 (TCSRX_1)
- Input capture register_1 (TICR_1)
- Time constant register_1 (TCORC_1)
- Input capture register R_1 (TICRR_1)
- Input capture register F_1 (TICRF_1)

11.3.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter. TCNT0 and TCNT1 (or TCNTY and TCNTX) comprise a single 16-bit register, so they can be accessed together in word units. 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.

11.3.2 Time Constant Register A (TCORA)

TCORA is an 8-bit readable/writable register. TCORA0 and TCORA1 (or TCORAY and TCORAX) comprise a single 16-bit register, so they can be accessed together in word units. TCORA is continually compared with the value in TCNT. When a match is detected, the CMFA flag in TCSR is set to 1. Note however that comparison is disabled during the T2 state of a TCORA write cycle. The timer output from the TMO pin can be freely controlled by this compare-match A signal and the settings of the OS1 and OS0 bits in TCSR. TCORA is initialized to H'FF.

11.3.3 Time Constant Register B (TCORB)

TCORB is an 8-bit readable/writable register. TCORB0 and TCORB1 (or TCORBY and TCORBX) comprise a single 16-bit register, so they can be accessed together in word units. TCORB is continually compared with the value in TCNT. When a match is detected, the CMFB

flag in TCSR is set to 1. Note however that comparison is disabled during the T2 state of a TCORB write cycle. The timer output from the TMO pin can be freely controlled by this compare-match B signal and the settings of the OS3 and OS2 bits in TCSR. TCORB is initialized to H'FF.

11.3.4 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	Select the method by which TCNT 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	Select the clock input to TCNT and count condition, together with the ICKS1 and ICKS0 bits in TECR (ICKS1_1 and ICKS0_1 in channel 1, and ICKS1_0 and ICKS0_0 in channel 0). For details, see table 11.2.
0	CKS0	0	R/W	

Table 11.2 Clock Input to TCNT and Count Condition

System	TCR			TECR		Description
	CKS2	CKS1	CKS0	ICKS1	ICKS0	
TMR0	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 TCNT1*
TMR1	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 TCNT0*
TMRY	0	0	0	—	—	Disables clock input
	0	0	1	—	—	Increments at falling edge of internal clock φ/4
	0	1	0	—	—	Increments at falling edge of internal clock φ/256

System	TCR			TECR		Description
	CKS2	CKS1	CKS0	ICKS1	ICKS0	
TMRY	0	1	1	—	—	Increments at falling edge of internal clock ϕ/2048
	1	0	0	—	—	Increments at overflow signal from TCNTX*
TMRX	0	0	0	—	—	Disables clock input
	0	0	1	—	—	Increments at falling edge of internal clock ϕ
	0	1	0	—	—	Increments at falling edge of internal clock ϕ/2
	0	1	1	—	—	Increments at falling edge of internal clock ϕ/4
	1	0	0	—	—	Increments at compare-match A from TCNTY*
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 TMR0 clock input is set as the TCNT1 overflow signal and the TMR1 clock input is set as the TCNT0 compare-match signal simultaneously, a count-up clock cannot be generated. Similarly, If the TMRY clock input is set as the TCNTX overflow signal and the TMRX clock input is set as the TCNTY compare-match signal simultaneously, a count-up clock cannot be generated. Simultaneous setting of these two conditions should therefore be avoided.

11.3.5 Timer Control/Status Register (TCSR)

TCSR indicates the status flags and controls compare-match output.

- **TCSR0**

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*[1]	Compare-Match Flag B [Setting condition] When the values of TCNT0 and TCORB0 match [Clearing condition] Read CMFB when CMFB = 1, then write 0 to CMFB
6	CMFA	0	R/(W)*[1]	Compare-Match Flag A [Setting condition] When the values of TCNT0 and TCORA0 match [Clearing condition] Read CMFA when CMFA = 1, then write 0 to CMFA
5	OVF	0	R/(W)*[1]	Timer Overflow Flag [Setting condition] When TCNT0 overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 to OVF
4	ADTE	0	R/W	A/D Trigger Enable*[2] Enables or disables A/D conversion 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
3	OS3	0	R/W	Output Select 3, 2
2	OS2	0	R/W	Specify how the TMO0 pin output level is to be changed by compare-match B of TCORB0 and TCNT0. 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	Specify how the TMO0 pin output level is to be changed by compare-match A of TCORA0 and TCNT0. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Notes: 1. Only 0 can be written to clear the flag.
2. This bit is reserved in channel 1.

- **TCSR1**

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT1 and TCORB1 match [Clearing condition] Read CMFB when CMFB = 1, then write 0 to CMFB
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT1 and TCORA1 match [Clearing condition] Read CMFA when CMFA = 1, then write 0 to CMFA
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT1 overflows from H′FF to H′00 [Clearing condition] Read OVF when OVF = 1, then write 0 to OVF
4	—	1	R	Reserved This bit is always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
3	OS3	0	R/W	Output Select 3, 2
2	OS2	0	R/W	Specify how the TMO1 pin output level is to be changed by compare-match B of TCORB1 and TCNT1. 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	Specify how the TMO1 pin output level is to be changed by compare-match A of TCORA1 and TCNT1. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

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

- **TCSRY**

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNTY and TCORBY match [Clearing condition] Read CMFB when CMFB = 1, then write 0 to CMFB
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNTY and TCORAY match [Clearing condition] Read CMFA when CMFA = 1, then write 0 to CMFA

Bit	Bit Name	Initial Value	R/W	Description
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNTY overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 to OVF
4	ICIE	0	R/W	Input Capture Interrupt Enable Enables or disables the ICF interrupt request (ICIX) when the ICF bit in TCSRX 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	Specify how the TMOY pin output level is to be changed by compare-match B of TCORBY and TCNTY. 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	Specify how the TMOY pin output level is to be changed by compare-match A of TCORAY and TCNTY. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

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

- **TCSRX**

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNTX and TCORBX match [Clearing condition] Read CMFB when CMFB = 1, then write 0 to CMFB
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNTX and TCORAX match [Clearing condition] Read CMFA when CMFA = 1, then write 0 to CMFA
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNTX overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 to 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 to ICF
3	OS3	0	R/W	Output Select 3, 2
2	OS2	0	R/W	Specify how the TMOX pin output level is to be changed by compare-match B of TCORBX and TCNTX. 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	Specify how the TMOX pin output level is to be changed by compare-match A of TCORAX and TCNTX. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

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

11.3.6 Input Capture Register (TICR)

TICR is an 8-bit register. The contents of TCNT are transferred to TICR at the falling edge of the external reset input. TICR cannot be directly accessed by the CPU.

11.3.7 Time Constant Register (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 T2 state of the TCORC write cycle and at the input capture cycle of TICR is disabled. TCORC is initialized to H'FF.

11.3.8 Input Capture Registers R and F (TICRR and TICRF)

TICRR and TICRF are 8-bit read-only registers. 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.

11.3.9 Timer Input Select Register (TISR)

TISR selects a signal source of external clock/reset input for the counter.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 1	R/(W)	Reserved The initial value should not be changed.
0	IS	0	R/W	Input Select Selects the internal synchronization signal (IVG signal) or timer clock/reset input pin (TMIY or ExTMIY) as the signal source of the external clock/reset input for the TMRY counter. 0: IVG signal is selected 1: TMIY is selected

11.4 Operation

11.4.1 Pulse Output

Figure 11.3 shows an example for outputting an arbitrary duty pulse.

1. Clear the CCLR1 bit in TCR to 0 and then set the CCLR0 bit 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.

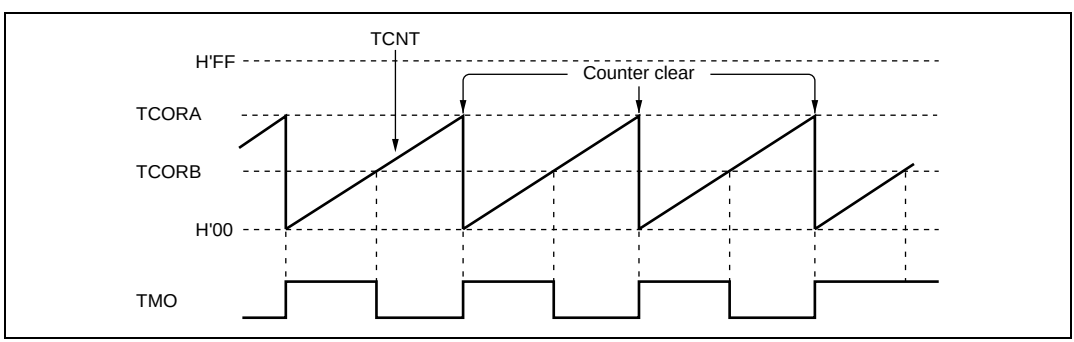


Figure 11.3 Pulse Output Example

11.5 Operation Timing

11.5.1 TCNT Count Timing

Figure 11.4 shows the TCNT count timing with an internal clock source. Figure 11.5 shows the TCNT count timing with an external clock source. The pulse width of the external clock signal must be at least 1.5 states for a single edge and at least 2.5 states for both edges. The counter will not increment correctly if the pulse width is less than these values.

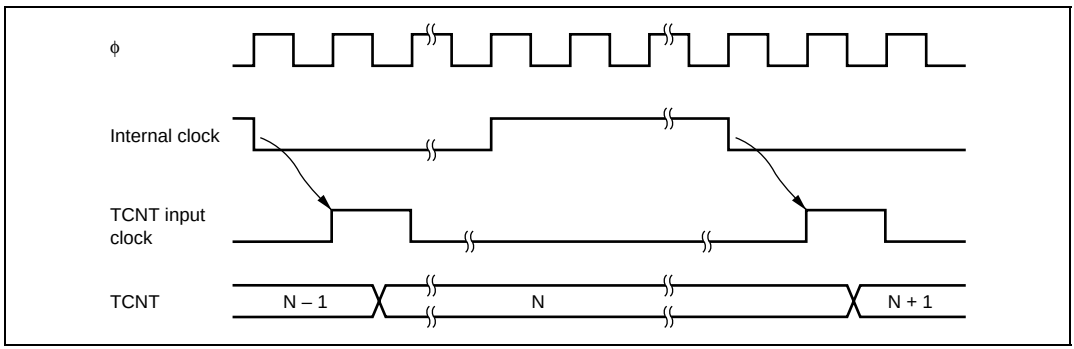


Figure 11.4 Count Timing for Internal Clock Input

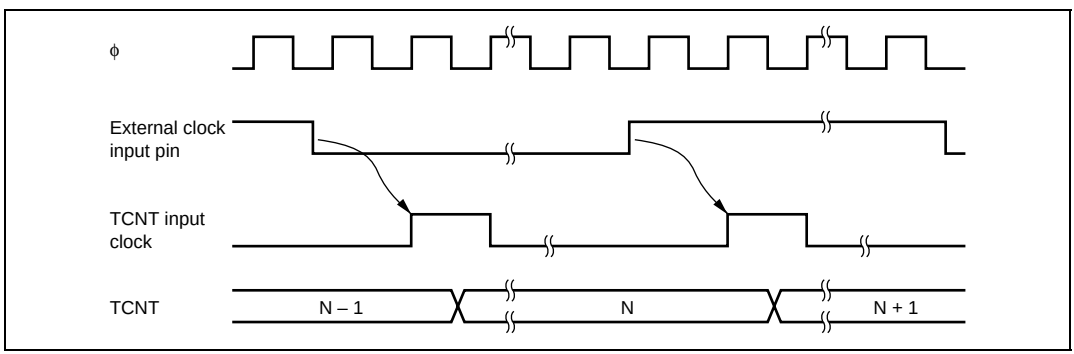


Figure 11.5 Count Timing for External Clock Input

11.5.2 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 values match, the compare-match signal is not generated until the next TCNT input clock. Figure 11.6 shows the timing of CMF flag setting.

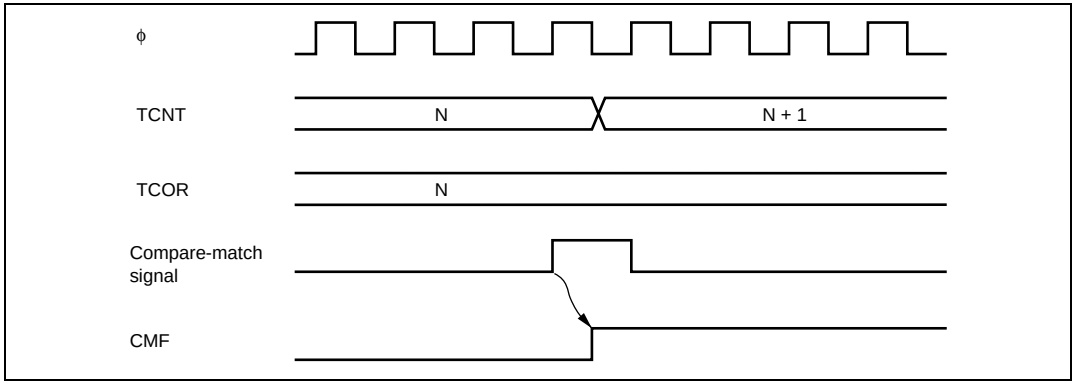


Figure 11.6 Timing of CMF Setting at Compare-Match

11.5.3 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 11.7 shows the timing of timer output when the output is set to toggle by a compare-match A signal.

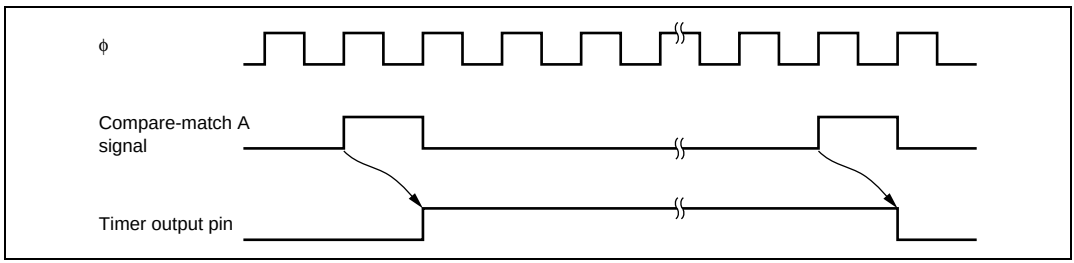


Figure 11.7 Timing of Toggled Timer Output by Compare-Match A Signal

11.5.4 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 11.8 shows the timing of clearing the counter by a compare-match.

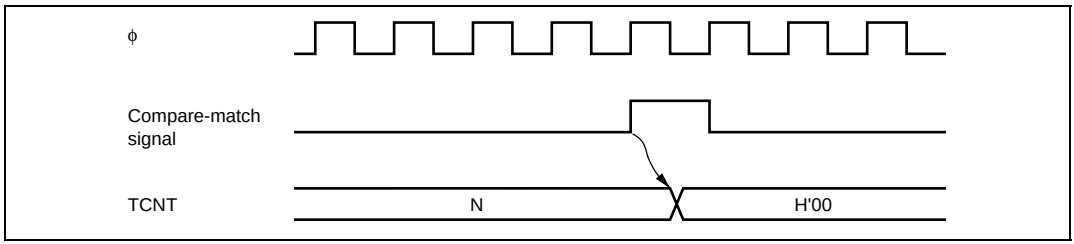


Figure 11.8 Timing of Counter Clear by Compare-Match

11.5.5 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 11.9 shows the timing of clearing the counter by an external reset input.

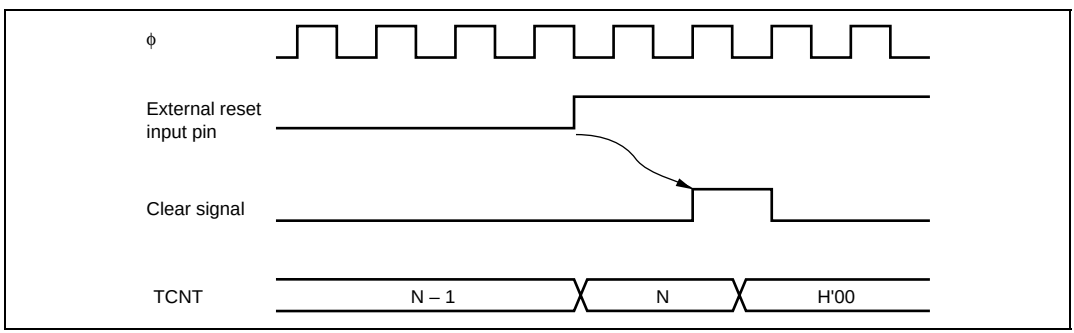


Figure 11.9 Timing of Counter Clear by External Reset Input

11.5.6 Timing of Overflow Flag (OVF) Setting

The OVF flag in TCSR is set to 1 by an overflow signal output when the TCNT overflows (changes from H'FF to H'00). Figure 11.10 shows the timing of OVF flag setting.

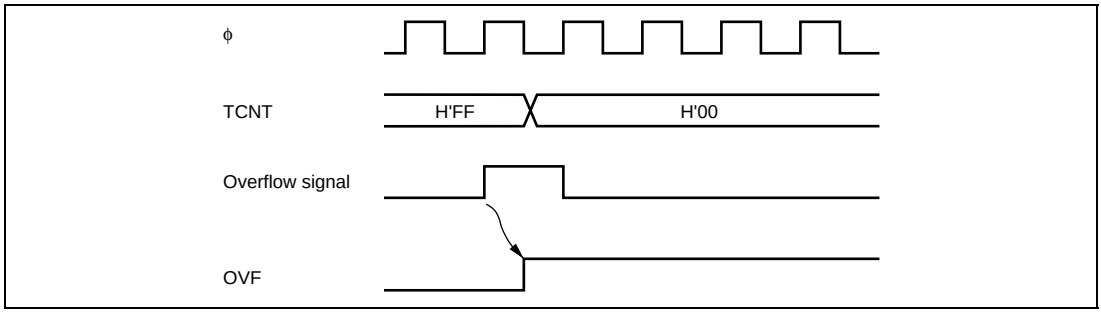


Figure 11.10 Timing of OVF Flag Setting

11.6 TMR0 and TMR1 Cascaded Connection

If bits CKS2 to CKS0 in either TCR0 or TCR1 are set to B'100, the 2-system 8-bit timers are cascaded. With this configuration, 16-bit count mode in which the TMR0 and TMR1 are used as a single 16-bit timer or compare-match count mode in which the compare-match of the 8-bit timer (TMR0) is counted by the TMR1 can be selected.

11.6.1 16-Bit Count Mode

When bits CKS2 to CKS0 in TCR0 are set to B'100, the timer functions as a single 16-bit timer with TMR0 occupying the upper eight bits and TMR1 occupying the lower 8 bits.

Setting of Compare-Match Flags:

- The CMF flag in TCSR0 is set to 1 when a 16-bit compare-match occurs.
- The CMF flag in TCSR1 is set to 1 when a lower 8-bit compare-match occurs.

Counter Clear Specification:

- If the CCLR1 and CCLR0 bits in TCR0 have been set for counter clear at compare-match, the 16-bit counter (TCNT0 and TCNT1 together) is cleared when a 16-bit compare-match occurs. The 16-bit counter (TCNT0 and TCNT1 together) is also cleared when counter clear by the TMI0 pin has been set.
- The settings of the CCLR1 and CCLR0 bits in TCR1 are invalid. The lower 8 bits cannot be cleared independently.

Pin Output:

- Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR0 is in accordance with the 16-bit compare-match conditions.
- Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR1 is in accordance with the lower 8-bit compare-match conditions.

11.6.2 Compare-Match Count Mode

When bits CKS2 to CKS0 in TCR1 are B'100, TCNT1 counts the occurrence of compare-match A for the TMR0. The TMR0 and TMR1 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 the TMR0 and TMR1.

11.7 TMRY and TMRX Cascaded Connection

If bits CKS2 to CKS0 in either TCRY or TCRX are set to B′100, the 2-system 8-bit timers are cascaded. With this configuration, 16-bit count mode in which the TMRY and TMRX are used as a single 16-bit timer or compare-match count mode in which the compare-match of the 8-bit timer (TMRY) is counted by the TMRX can be selected.

11.7.1 16-Bit Count Mode

When bits CKS2 to CKS0 in TCRY are set to B'100, the timer functions as a single 16-bit timer with TMRY occupying the upper eight bits and TMRX occupying the lower 8 bits.

Setting of Compare-Match Flags:

- The CMF flag in TCSRY is set to 1 when an upper 8-bit compare-match occurs.
- The CMF flag in TCSRX is set to 1 when a lower 8-bit compare-match occurs.

Counter Clear Specification:

- If the CCLR1 and CCLR0 bits in TCRY have been set for counter clear at compare-match, only the upper eight bits of TCNTY are cleared. The upper eight bits of TCNTY are also cleared when counter clear by the TMRIY pin has been set.
- The settings of the CCLR1 and CCLR0 bits in TCRX are valid, and the lower 8 bits of TCNTX can be cleared.

Pin Output:

- Control of output from the TMOY pin by bits OS3 to OS0 in TCSRY is in accordance with the upper 8-bit compare-match conditions.
- Control of output from the TMOX pin by bits OS3 to OS0 in TCSRX is in accordance with the lower 8-bit compare-match conditions.

11.7.2 Compare-Match Count Mode

When bits CKS2 to CKS0 in TCRX are B'100, TCNTX counts the occurrence of compare-match A for the TMRY. TMRY and TMRX 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 the TMRY and TMRX.

11.7.3 Input Capture Operation

The TMRX has input capture registers (TICR, TICRR, and TICRF). A narrow pulse width can be measured with TICRR and TICRF, using a single capture. If the falling edge of TMRIX (TMRX input capture input signal) is detected after its rising edge has been detected, the value of TCNTX at that time is transferred to both TICRR and TICRF.

Input Timing of Input Capture Signal:

Figure 11.11 shows the timing of the input capture operation.

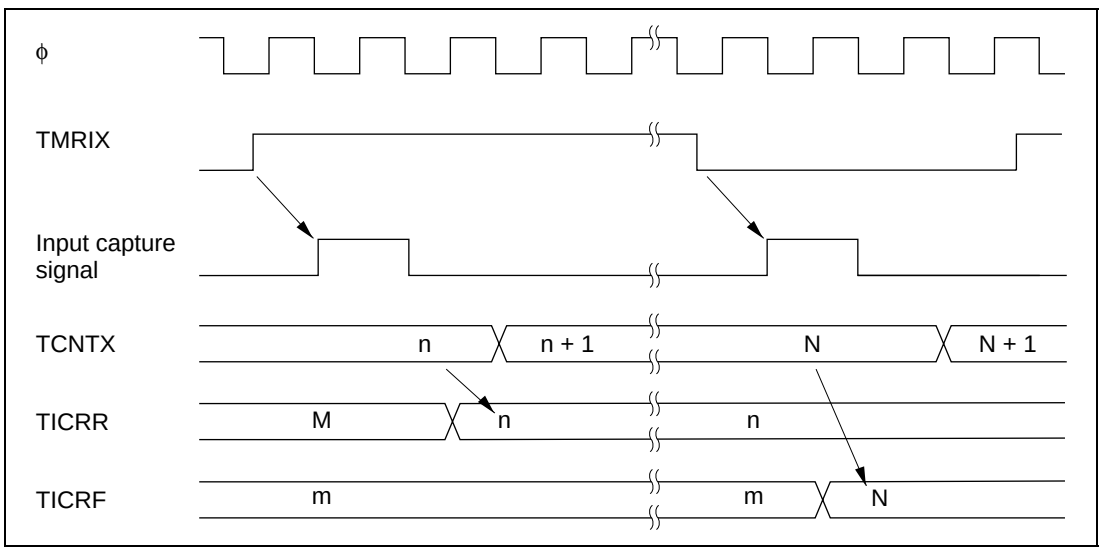


Figure 11.11 Timing of Input Capture Operation

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 11.12 shows the timing of this operation.

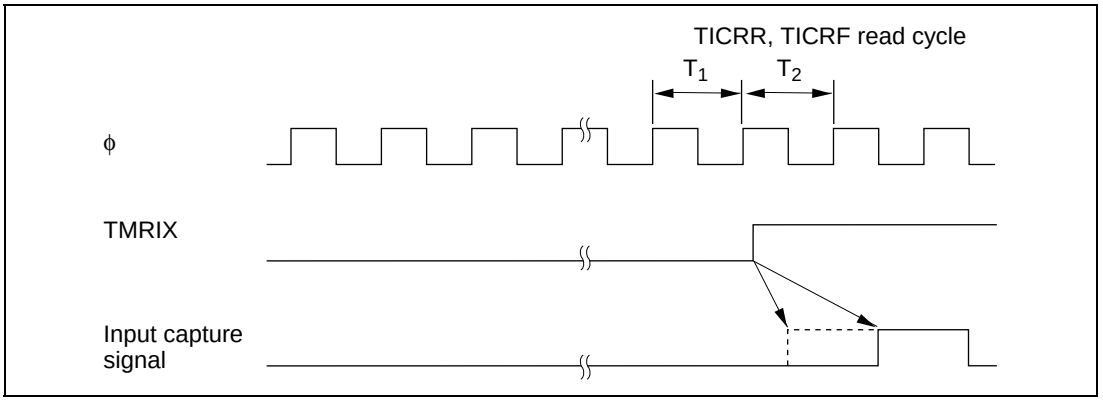


Figure 11.12 Timing of Input Capture Signal (When Input Capture Signal is Input during TICRR and TICRF Read)

11.8 Interrupt Sources

The TMR0, TMR1, and TMRY can generate three types of interrupts: CMIA, CMIB, and OVI. The TMRX can generate four types of interrupts: CMIA, CMIB, OVI, and ICIX. Table 11.3 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 11.3 Interrupt Sources of 8-Bit Timers TMR0, TMR1, TMRY, and TMRX

Channel	System	Name	Interrupt Source	Interrupt Flag	Interrupt Priority
0	TMRX	CMIAX0	TCORAX compare-match	CMFA	High
		CMIBX0	TCORBX compare-match	CMFB	
		OVIX0	TCNTX overflow	OVF	
		ICIX0	Input capture	ICF	
	TMR0	CMIA00	TCORA0 compare-match	CMFA	
		CMIB00	TCORB0 compare-match	CMFB	
		OVI00	TCNT0 overflow	OVF	
	TMR1	CMIA10	TCORA1 compare-match	CMFA	
		CMIB10	TCORB1 compare-match	CMFB	
		OVI10	TCNT1 overflow	OVF	
	TMRY	CMIAY0	TCORAY compare-match	CMFA	
		CMIBY0	TCORBY compare-match	CMFB	
		OVIY0	TCNTY overflow	OVF	
1	TMRX	CMIAX1	TCORAX compare-match	CMFA	
		CMIBX1	TCORBX compare-match	CMFB	
		OVIX1	TCNTX overflow	OVF	
		ICIX1	Input capture	ICF	
	TMR0	CMIA01	TCORA0 compare-match	CMFA	
		CMIB01	TCORB0 compare-match	CMFB	
		OVI01	TCNT0 overflow	OVF	
	TMR1	CMIA11	TCORA1 compare-match	CMFA	
		CMIB11	TCORB1 compare-match	CMFB	
		OVI11	TCNT1 overflow	OVF	
	TMRY	CMIAY1	TCORAY compare-match	CMFA	
		CMIBY1	TCORBY compare-match	CMFB	
		OVIY1	TCNTY overflow	OVF	Low

11.9 Usage Notes

11.9.1 Conflict between TCNT Write and Clear

If TCNT is cleared during the T_2 state of a TCNT write cycle as shown in figure 11.13, the clear takes priority and TCNT is not written.

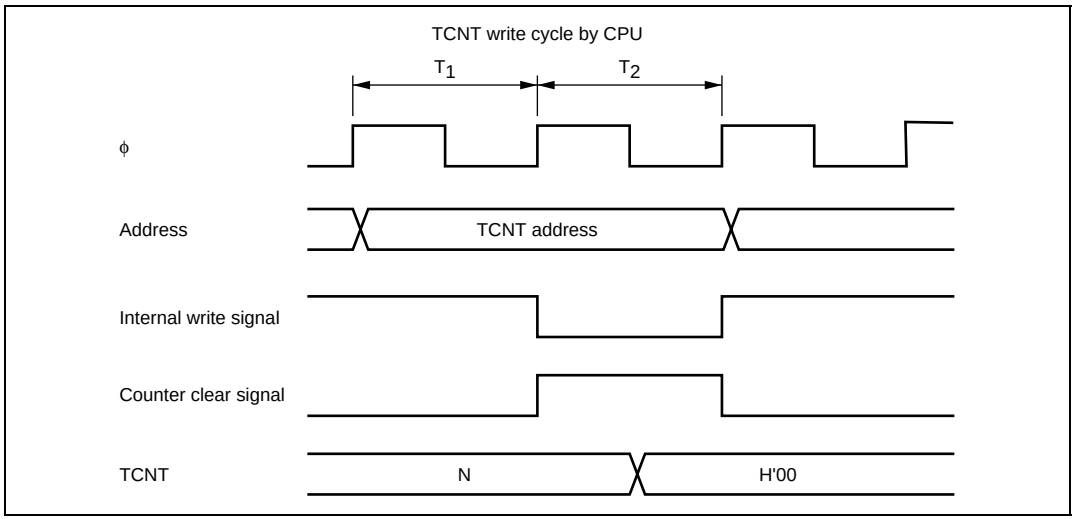


Figure 11.13 Conflict between TCNT Write and Clear

11.9.2 Conflict between TCNT Write and Increment

If a TCNT input clock is generated during the T_2 state of a TCNT write cycle as shown in figure 11.14, the write takes priority and the counter is not incremented.

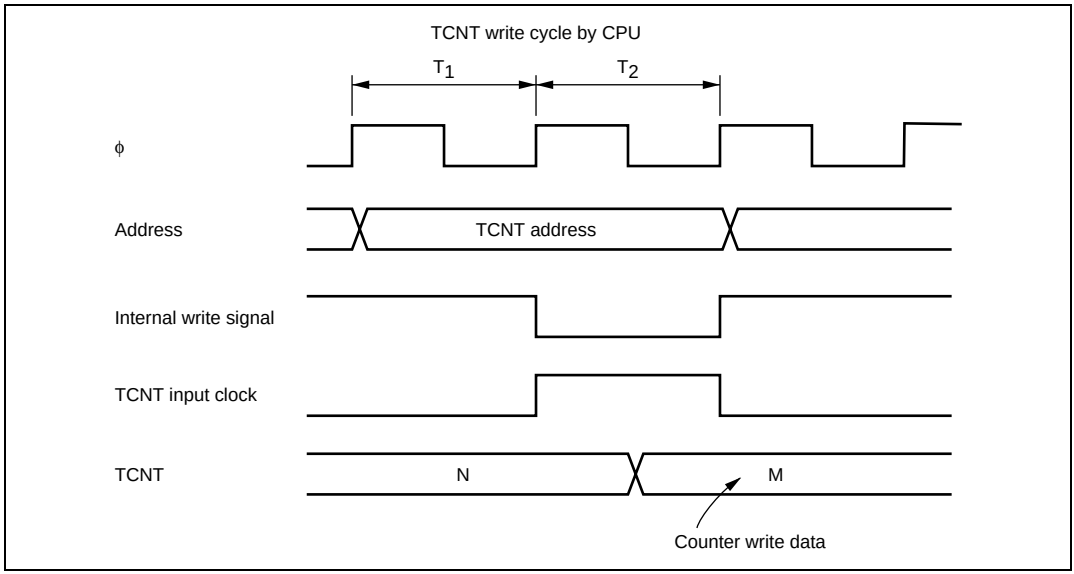


Figure 11.14 Conflict between TCNT Write and Increment

11.9.3 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 11.15, the TCOR write takes priority and the compare-match signal is disabled. With the TMRX, 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.

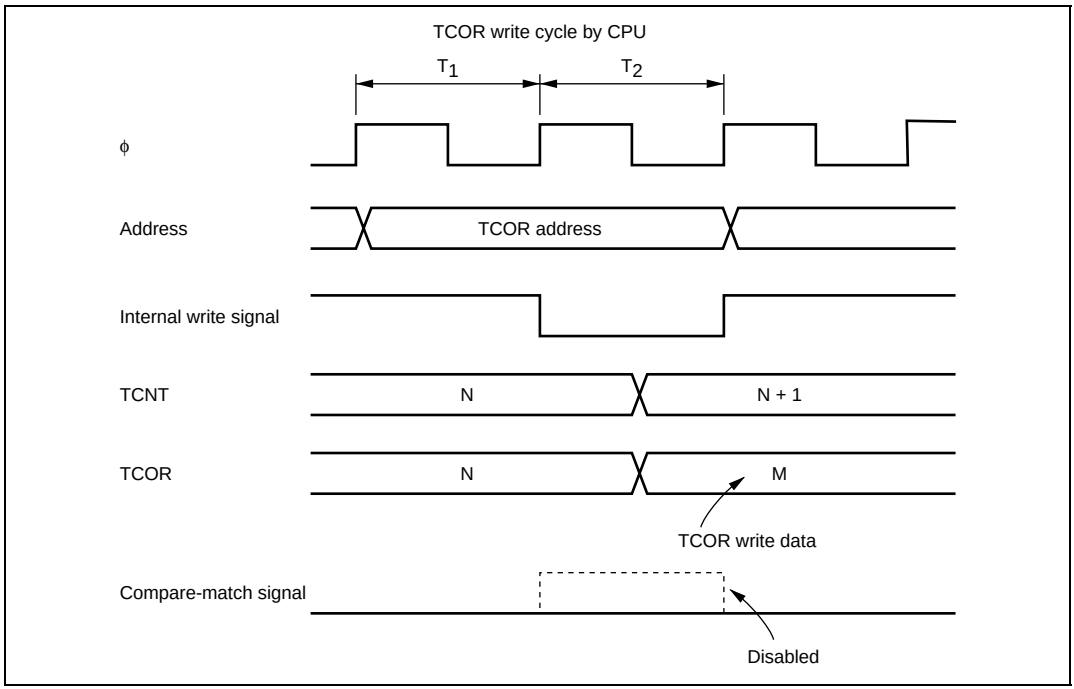


Figure 11.15 Conflict between TCOR Write and Compare-Match

11.9.4 Conflict between Compare-Matches A and B

If compare-matches A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output states set for compare-match A and compare-match B, as shown in table 11.4.

Table 11.4 Timer Output Priorities

Output Setting	Priority
Toggle output	High
1 output	
0 output	
No change	Low

11.9.5 Switching of Internal Clocks and TCNT Operation

TCNT may be incremented erroneously when the internal clock is switched over. Table 11.5 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 11.5, 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 11.5 Switching of Internal Clocks and TCNT Operation

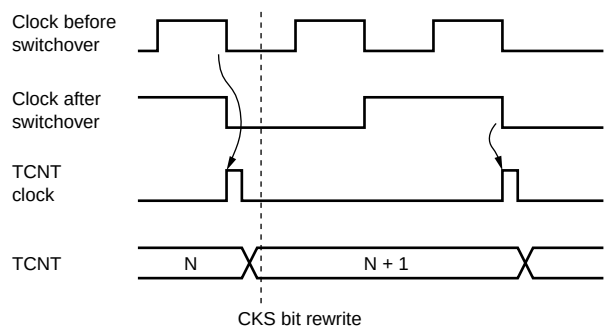
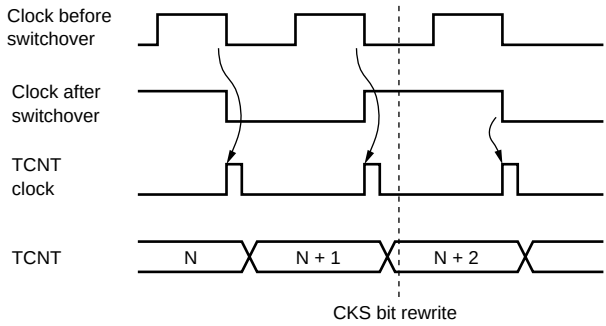
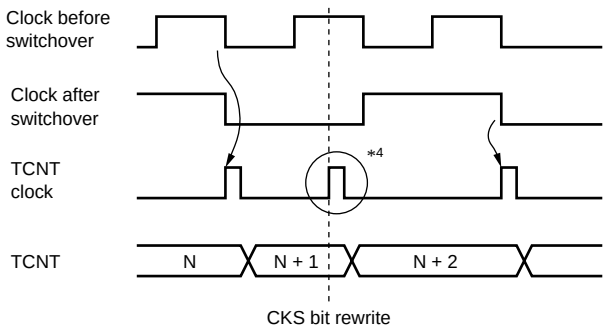
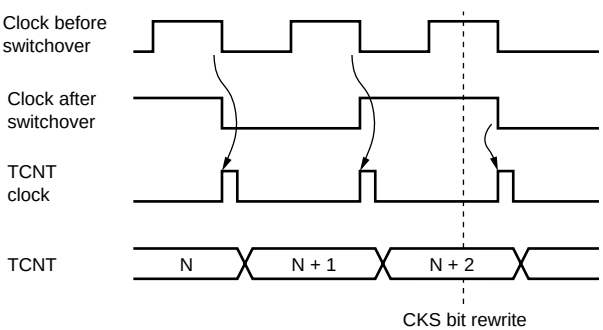
No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
1	Clock switching from low to low level*[1]	

Table 11.5 Switching of Internal Clocks and TCNT Operation (cont)

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
2	Clock switching from low to high level*[2]	
3	Clock switching from high to low level*[3]	
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.

11.9.6 Mode Setting with Cascaded Connection

If 16-bit count mode and compare-match count mode are set simultaneously, the input clock pulses for TCNT0 and TCNT1 (or TCNTY and TCNTX) are not generated, and thus the counters will stop operating. Simultaneous setting of these two modes should therefore be avoided.

Section 12 16-Bit Timer Pulse Unit (TPU)

This LSI has an on-chip 16-bit timer pulse unit (TPU) that comprises three 16-bit timer channels. The function list of the 16-bit timer pulse unit and its block diagram are shown in table 12.1 and figure 12.1, respectively.

12.1 Features

- Maximum 8-pulse input/output
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  Waveform output at compare match
  Input capture function
  Counter clear operation
  Synchronous operation:
  Multiple timer counters (TCNT) can be written to simultaneously
  Simultaneous clearing by compare match and input capture possible
  Register simultaneous input/output possible by counter synchronous operation
  Maximum of 7-phase PWM output possible by combination with synchronous operation
- Buffer operation settable for channel 0
- Phase counting mode settable independently for each of channels 1 and 2
- Cascaded operation
- Fast access via internal 16-bit bus
- 13 interrupt sources
- Automatic transfer of register data
- A/D conversion start trigger can be generated
- Module stop mode can be set

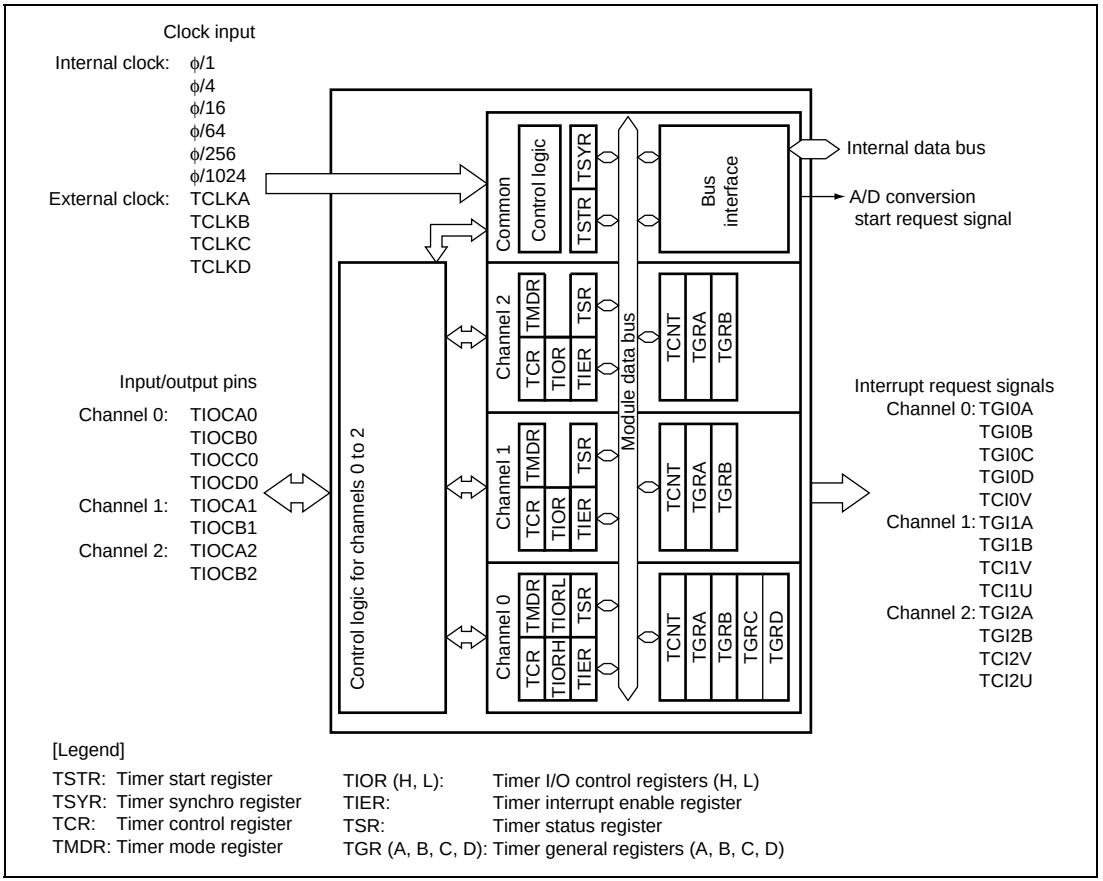


Figure 12.1 Block Diagram of TPU

Table 12.1 TPU Functions

Item		Channel 0	Channel 1	Channel 2
Count clock		φ/1	φ/1	φ/1
		φ/4	φ/4	φ/4
		φ/16	φ/16	φ/16
		φ/64	φ/64	φ/64
		TCLKA	φ/256	φ/1024
		TCLKB	TCLKA	TCLKA
		TCLKC	TCLKB	TCLKB
		TCLKD		TCLKC
General registers (TGR)		TGRA_0	TGRA_1	TGRA_2
		TGRB_0	TGRB_1	TGRB_2
General registers/buffer registers		TGRC_0	—	—
		TGRD_0		
I/O pins		TIOCA0	TIOCA1	TIOCA2
		TIOCB0	TIOCB1	TIOCB2
		TIOCC0		
		TIOCD0		
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	O	O	O
	1 output	O	O	O
	Toggle output	O	O	O
Input capture function		O	O	O
Synchronous operation		O	O	O
PWM mode		O	O	O
Phase counting mode		—	O	O
Buffer operation		O	—	—

Item	Channel 0	Channel 1	Channel 2
A/D conversion start trigger	TGRA_0 compare match or input capture	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture
Interrupt sources	5 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow

[Legend]

O: Possible

—: Not possible

12.2 Input/Output Pins

Table 12.2 Pin Configuration

Channel	Symbol	I/O	Function
All	TCLKA	Input	External clock A input pin (Channel 1 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (Channel 1 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channel 2 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channel 2 phase counting mode B phase input)
0	TIOCA0	I/O	TGRA_0 input capture input/output compare output/PWM output pin
	TIOCB0	I/O	TGRB_0 input capture input/output compare output/PWM output pin
	TIOCC0	I/O	TGRC_0 input capture input/output compare output/PWM output pin
	TIOCD0	I/O	TGRD_0 input capture input/output compare output/PWM output pin
1	TIOCA1	I/O	TGRA_1 input capture input/output compare output/PWM output pin
	TIOCB1	I/O	TGRB_1 input capture input/output compare output/PWM output pin
2	TIOCA2	I/O	TGRA_2 input capture input/output compare output/PWM output pin
	TIOCB2	I/O	TGRB_2 input capture input/output compare output/PWM output pin

12.3 Register Descriptions

The TPU has the following registers for each channel.

Channel 0:

- Timer control register_0 (TCR_0)
- Timer mode register_0 (TMDR_0)
- Timer I/O control register H_0 (TIORH_0)
- Timer I/O control register L_0 (TIORL_0)
- Timer interrupt enable register_0 (TIER_0)
- Timer status register_0 (TSR_0)
- Timer counter_0 (TCNT_0)
- Timer general register A_0 (TGRA_0)
- Timer general register B_0 (TGRB_0)
- Timer general register C_0 (TGRC_0)
- Timer general register D_0 (TGRD_0)

Channel 1:

- Timer control register_1 (TCR_1)
- Timer mode register_1 (TMDR_1)
- Timer I/O control register _1 (TIOR_1)
- Timer interrupt enable register_1 (TIER_1)
- Timer status register_1 (TSR_1)
- Timer counter_1 (TCNT_1)
- Timer general register A_1 (TGRA_1)
- Timer general register B_1 (TGRB_1)

Channel 2:

- Timer control register_2 (TCR_2)
- Timer mode register_2 (TMDR_2)
- Timer I/O control register_2 (TIOR_2)
- Timer interrupt enable register_2 (TIER_2)
- Timer status register_2 (TSR_2)
- Timer counter_2 (TCNT_2)
- Timer general register A_2 (TGRA_2)
- Timer general register B_2 (TGRB_2)

Common Registers:

- Timer start register (TSTR)
- Timer synchro register (TSYR)

12.3.1 Timer Control Register (TCR)

TCR controls the TCNT operation for each channel. The TPU has a total of three TCR registers, one for each channel (channels 0 to 2). TCR settings should be made only when TCNT operation is stopped.

Bit	Bit Name	Initial Value	R/W	Description
7	CCLR2	0	R/W	Counter Clear 2 to 0
6	CCLR1	0	R/W	Select the TCNT counter clearing source. See tables 12.3 and 12.4 for details.
5	CCLR0	0	R/W	
4	CKEG1	0	R/W	Clock Edge 1, 0
3	CKEG0	0	R/W	Select the input clock edge. When the internal clock is counted using both edges, the input clock cycle is 1/2 (example: ϕ/4 both edges = ϕ/2 rising edge). If phase counting mode is used on channels 1 and 2, this setting is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is ϕ/4 or slower. This setting is ignored if the input clock is ϕ/1 and the rising edge counting is selected. 00: Count at rising edge 01: Count at falling edge 1x: Count at both edges [Legend] x: Don't care
2	TPSC2	0	R/W	Timer Prescaler 2 to 0
1	TPSC1	0	R/W	Select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 12.5 to 12.7 for details.
0	TPSC0	0	R/W	

Table 12.3 CCLR2 to CCLR0 (Channel 0)

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*[1]
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture*[2]
		1	0	TCNT cleared by TGRD compare match/input capture*[2]
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*[1]

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture dose not occur.

Table 12.4 CCLR2 to CCLR0 (Channels 1 and 2)

Channel	Bit 7 Reserved*[2]	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*[1]

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.
2. Bit 7 is reserved in channels 1 and 2. It is always read as 0 and cannot be modified.

Table 12.5 TPSC2 to TPSC0 (Channel 0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on φ/1
			1	Internal clock: counts on φ/4
		1	0	Internal clock: counts on φ/16
			1	Internal clock: counts on φ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	External clock: counts on TCLKD pin input

Table 12.6 TPSC2 to TPSC0 (Channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on φ/1
			1	Internal clock: counts on φ/4
		1	0	Internal clock: counts on φ/16
			1	Internal clock: counts on φ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	Internal clock: counts on φ/256
			1	Counts on TCNT2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

Table 12.7 TPSC2 to TPSC0 (Channel 2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on φ/1
			1	Internal clock: counts on φ/4
		1	0	Internal clock: counts on φ/16
			1	Internal clock: counts on φ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	Internal clock: counts on φ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

12.3.2 Timer Mode Register (TMDR)

TMDR sets the operating mode for each channel. The TPU has a total of three TMDR registers, one for each channel. TMDR settings should be made only when TCNT operation is stopped.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare is not generated. In channels 1 and 2, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: TGRB operates normally 1: TGRB and TGRD used together for buffer operation

Bit	Bit Name	Initial Value	R/W	Description
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare is not generated. In channels 1 and 2, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified. 0: TGRA operates normally 1: TGRA and TGRC used together for buffer operation
3	MD3	0	R/W	Modes 3 to 0 Set the timer operating mode. MD3 is a reserved bit. The write value should always be 0. See table 12.8 for details.
2	MD2	0	R/W	
1	MD1	0	R/W	
0	MD0	0	R/W	

Table 12.8 MD3 to MD0

Bit 3 MD3*[1]	Bit 2 MD2*[2]	Bit 1 MD1	Bit 0 MD0	Description
0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	×	×	×	—

[Legend] x: Don't care

Notes: 1. MD3 is a reserved bit. The write value should always be 0.

2. Phase counting mode cannot be set for channel 0. In this case, 0 should always be written to the MD2 bit.

12.3.3 Timer I/O Control Register (TIOR)

TIOR controls TGR. The TPU has a total of four TIOR registers, two for channel 0, and one each for channels 1 and 2. Care is required since TIOR is affected by the TMDR setting.

The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and TIOR operates as a buffer register.

- **TIORH_0, TIOR_1, TIOR_2**

Bit	Bit Name	Initial Value	R/W	Description
7	IOB3	0	R/W	I/O Control B3 to B0
6	IOB2	0	R/W	Specify the function of TGRB.
5	IOB1	0	R/W	
4	IOB0	0	R/W	
3	IOA3	0	R/W	I/O Control A3 to A0
2	IOA2	0	R/W	Specify the function of TGRA.
1	IOA1	0	R/W	
0	IOA0	0	R/W	

- **TIORL_0**

Bit	Bit Name	Initial Value	R/W	Description
7	IOD3	0	R/W	I/O Control D3 to D0
6	IOD2	0	R/W	Specify the function of TGRD.
5	IOD1	0	R/W	
4	IOD0	0	R/W	
3	IOC3	0	R/W	I/O Control C3 to C0
2	IOC2	0	R/W	Specify the function of TGRC.
1	IOC1	0	R/W	
0	IOC0	0	R/W	

Table 12.9 TIORH_0 (Channel 0)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_0 Function**	**TIOCB0 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB0 pin Input capture at rising edge
			1		Capture input source is TIOCB0 pin Input capture at falling edge
		1	×		Capture input source is TIOCB0 pin Input capture at both edges
	1	×	×		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down*

[Legend] x: Don't care

Note: * When bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and ϕ/1 is used as the TCNT_1 count clock, this setting is invalid and input capture is not generated.

Table 12.10 TIORH_0 (Channel 0)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_0 Function**	**TIOCA0 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA0 pin Input capture at rising edge
			1		Capture input source is TIOCA0 pin Input capture at falling edge
		1	×		Capture input source is TIOCA0 pin Input capture at both edges
	1	×	×		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend] x: Don't care

Table 12.11 TIORL_0 (Channel 0)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_0 Function**	**TIOCD0 Pin Function**
0	0	0	0	Output compare register*2	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD0 pin Input capture at rising edge
			1		Capture input source is TIOCD0 pin Input capture at falling edge
		1	×		Capture input source is TIOCD0 pin Input capture at both edges
	1	×	×		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down*1

[Legend] x: Don't care

Notes: 1. When bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and φ/1 is used as the TCNT_1 count clock, this setting is invalid and input capture is not generated.

2. When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 12.12 TIORL_0 (Channel 0)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_0 Function**	**TIOCC0 Pin Function**
0	0	0	0	Output compare register*	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is TIOCC0 pin Input capture at rising edge
			1		Capture input source is TIOCC0 pin Input capture at falling edge
		1	×		Capture input source is TIOCC0 pin Input capture at both edges
	1	×	×		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend] x: Don't care

Note: * When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 12.13 TIOR_1 (Channel 1)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_1 Function**	**TIOCB1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB1 pin Input capture at rising edge
			1		Capture input source is TIOCB1 pin Input capture at falling edge
		1	×		Capture input source is TIOCB1 pin Input capture at both edges
	1	×	×		Capture input source is TGRC_0 compare match/input capture Input capture at generation of channel 0/TGRC_0 compare match/input capture

[Legend] x: Don't care

Table 12.14 TIOR_1 (Channel 1)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_1 Function**	**TIOCA1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA1 pin Input capture at rising edge
			1		Capture input source is TIOCA1 pin Input capture at falling edge
		1	×		Capture input source is TIOCA1 pin Input capture at both edges
	1	×	×		Capture input source is TGRA_0 compare match/input capture Input capture at generation of channel 0/TGRA_0 compare match/input capture

[Legend] x: Don't care

Table 12.15 TIOR_2 (Channel 2)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_2 Function**	**TIOCB2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	×	0	0	Input capture register	Capture input source is TIOCB2 pin Input capture at rising edge
			1		Capture input source is TIOCB2 pin Input capture at falling edge
		1	×		Capture input source is TIOCB2 pin Input capture at both edges

[Legend] x: Don't care

Table 12.16 TIOR_2 (Channel 2)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_2 Function**	**TIOCA2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	×	0	0	Input capture register	Capture input source is TIOCA2 pin Input capture at rising edge
			1		Capture input source is TIOCA2 pin Input capture at falling edge
		1	×		Capture input source is TIOCA2 pin Input capture at both edges

[Legend] x: Don't care

12.3.4 Timer Interrupt Enable Register (TIER)

TIER controls enabling or disabling of interrupt requests for each channel. The TPU has a total of three TIER registers, one for each channel.

Bit	Bit Name	Initial Value	R/W	Description
7	TTGE	0	R/W	A/D Conversion Start Request Enable Enables or disables generation of A/D conversion start requests by TGRA input capture/compare match. 0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled
6	—	1	R	Reserved This bit is always read as 1 and cannot be modified.
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests (TCIU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1 and 2. In channel 0, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channel 0. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGID) by TGFD disabled 1: Interrupt requests (TGID) by TGFD enabled
2	TGIEC	0	R/W	TGR Interrupt Enable C Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channel 0. In channels 1 and 2, bit 2 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGIC) by TGFC disabled 1: Interrupt requests (TGIC) by TGFC enabled

Bit	Bit Name	Initial Value	R/W	Description
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB disabled 1: Interrupt requests (TGIB) by TGFB enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA disabled 1: Interrupt requests (TGIA) by TGFA enabled

12.3.5 Timer Status Register (TSR)

TSR indicates the status of each channel. The TPU has a total of three TSR registers, one for each channel.

Bit	Bit Name	Initial Value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that indicates the direction in which TCNT counts in channels 1 and 2. In channel 0, bit 7 is reserved. It is always read as 1 and cannot be modified. 0: TCNT counts down 1: TCNT counts up
6	—	1	R	Reserved This bit is always read as 1 and cannot be modified.
5	TCFU	0	R/(W)*	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 1 and 2 are set to phase counting mode. In channel 0, bit 5 is reserved. It is always read as 0 and cannot be modified. [Setting condition] When the TCNT value underflows (change from H'0000 to H'FFFF) [Clearing condition] When 0 is written to TCFU after reading TCFU = 1

Bit	Bit Name	Initial Value	R/W	Description
4	TCFV	0	R/(W)*	Overflow Flag Status flag that indicates that TCNT overflow has occurred. [Setting condition] When the TCNT value overflows (change from H'FFFF to H'0000) [Clearing condition] When 0 is written to TCFV after reading TCFV = 1
3	TGFD	0	R/(W)*	Input Capture/Output Compare Flag D Status flag that indicates the occurrence of TGRD input capture or compare match in channel 0. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register [Clearing condition] • When 0 is written to TGFD after reading TGFD = 1
2	TGFC	0	R/(W)*	Input Capture/Output Compare Flag C Status flag that indicates the occurrence of TGRC input capture or compare match in channel 0. In channels 1 and 2, bit 2 is reserved. It is always read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRC while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register [Clearing condition] • When 0 is written to TGFC after reading TGFC = 1

Bit	Bit Name	Initial Value	R/W	Description
1	TGFB	0	R/(W)*	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGRB input capture or compare match. [Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register [Clearing condition] • When 0 is written to TGFB after reading TGFB = 1
0	TGFA	0	R/(W)*	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TGRA input capture or compare match. [Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register [Clearing condition] • When 0 is written to TGFA after reading TGFA = 1

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

12.3.6 Timer Counter (TCNT)

TCNT is a 16-bit readable/writable counter. The TPU has a total of three TCNT counters, one for each channel. TCNT is initialized to H'0000 by a reset, and in hardware standby mode. TCNT cannot be accessed in 8-bit units; it must always be accessed in 16-bit units.

12.3.7 Timer General Register (TGR)

TGR is a 16-bit readable/writable register with a dual function as output compare and input capture registers. The TPU has a total of eight TGR registers, four for channel 0 and two each for channels 1 and 2. TGRC and TGRD for channel 0 can also be designated for operation as buffer registers. TGR is initialized to H'FFFF by a reset, and in hardware standby mode. TGR cannot be accessed in 8-bit units; it must always be accessed in 16-bit units. TGR and buffer register combinations are TGRA—TGRC and TGRB—TGRD.

12.3.8 Timer Start Register (TSTR)

TSTR selects TCNT operation/stop for channels 0 to 2. If the corresponding bit is set to 1, TCNT starts counting for the channel. When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	—	Reserved The write value should always be 0.
2	CST2	0	R/W	Counter Start 2 to 0
1	CST1	0	R/W	Select operation or stop for TCNT.
0	CST0	0	R/W	If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the output compare output level for the TIOC pin is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNTn count operation is stopped 1: TCNTn performs count operation (n = 2 to 0)

12.3.9 Timer Synchro Register (TSYR)

TSYR selects independent operation or synchronous operation for TCNT in channels 0 to 2. Synchronous operation is performed in the channel when the corresponding bit in TSYR is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	—	Reserved The write value should always be 0.
2	SYNC2	0	R/W	Timer Synchro 2 to 0
1	SYNC1	0	R/W	Select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, synchronous presetting of multiple channels and synchronous clearing due to counter clearing on another channel are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR2 to CCLR0 in TCR. 0: TCNTn operates independently (TCNT presetting/clearing is unrelated to other channels) 1: TCNTn performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible (n = 2 to 0)
0	SYNC0	0	R/W	

12.4 Interface to Bus Master

12.4.1 16-Bit Registers

TCNT and TGR are 16-bit registers. As the data bus to the bus master is 16 bits wide, these registers can be read from or written to in 16-bit units.

These registers cannot be read from or written to in 8-bit units; 16-bit access must always be used.

An example of 16-bit register access operation is shown in figure 12.2.

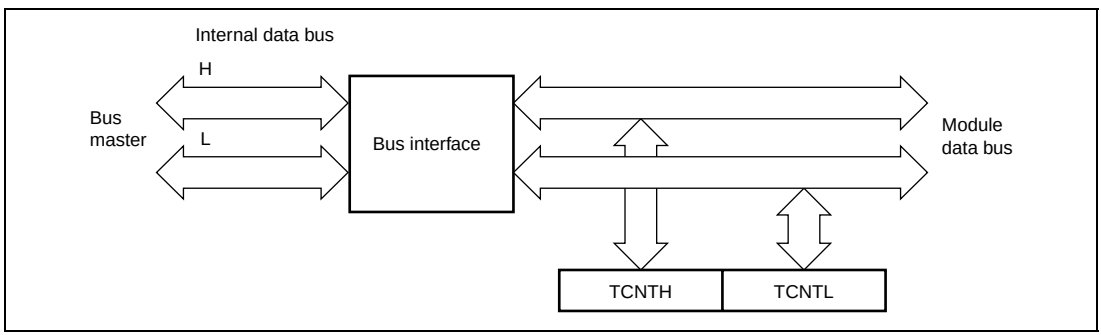


Figure 12.2 16-Bit Register Access Operation [Bus Master ↔ TCNT (16 Bits)]

12.4.2 8-Bit Registers

Registers other than TCNT and TGR are 8 bits. As the data bus to the bus master is 16 bits wide, these registers can be read from or written to in 16-bit units. They can also be read from or written to in 8-bit units.

Examples of 8-bit register access operation are shown in figures 12.3, 12.4, and 12.5.

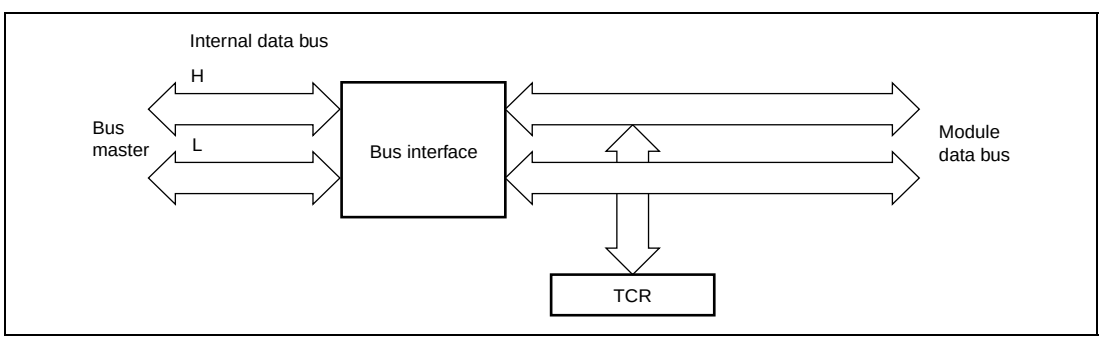


Figure 12.3 8-Bit Register Access Operation [Bus Master ↔ TCR (Upper 8 Bits)]

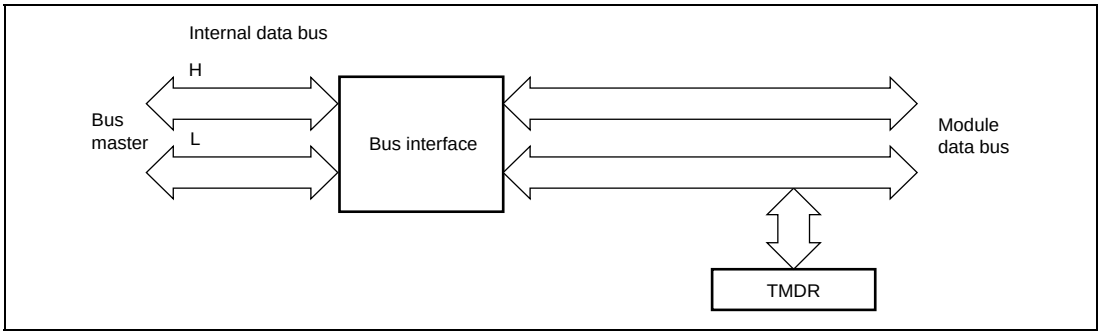


Figure 12.4 8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower 8 Bits)]

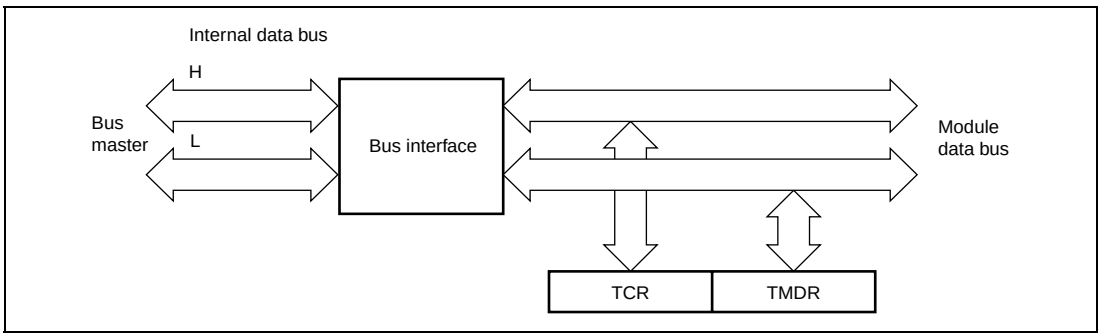


Figure 12.5 8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR (16 Bits)]

12.5 Operation

12.5.1 Basic Functions

Each channel has TCNT and TGR. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting. Each TGR can be used as an input capture register or output compare register.

Counter Operation:

When one of bits CST0 to CST2 in TSTR is set to 1, TCNT for the corresponding channel starts counting. TCNT can operate as a free-running counter, periodic counter, and so on.

- Example of Count Operation Setting Procedure
  Figure 12.6 shows an example of the count operation setting procedure.

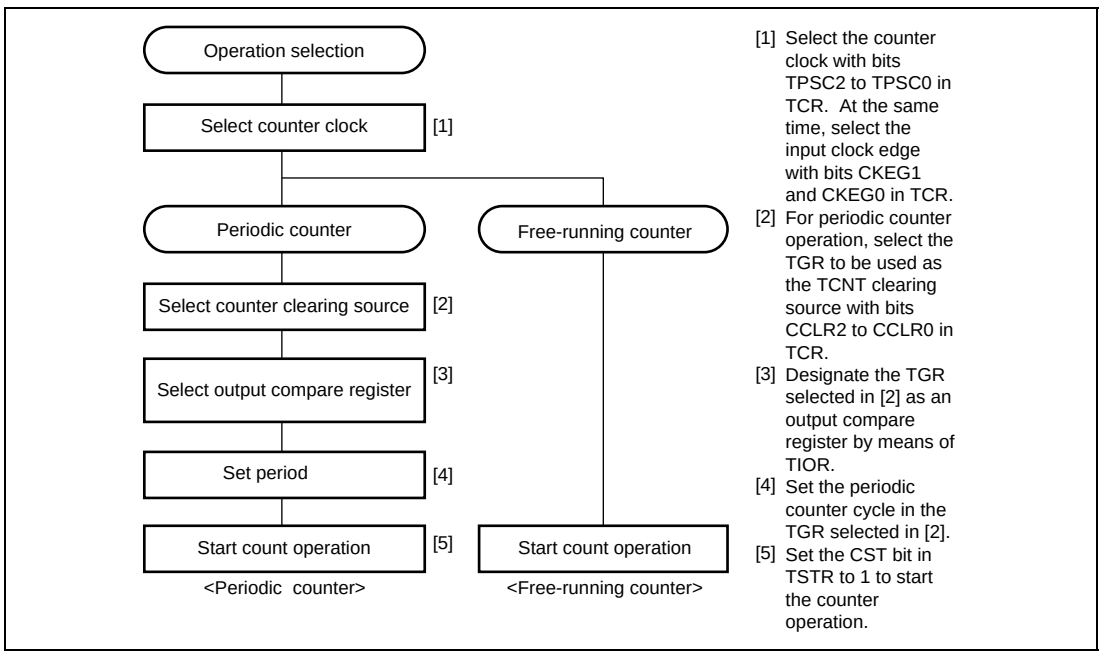


Figure 12.6 Example of Counter Operation Setting Procedure

- Free-Running Count Operation and Periodic Count Operation

  Immediately after a reset, the TPU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1, the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the TPU requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

  Figure 12.7 illustrates free-running counter operation.

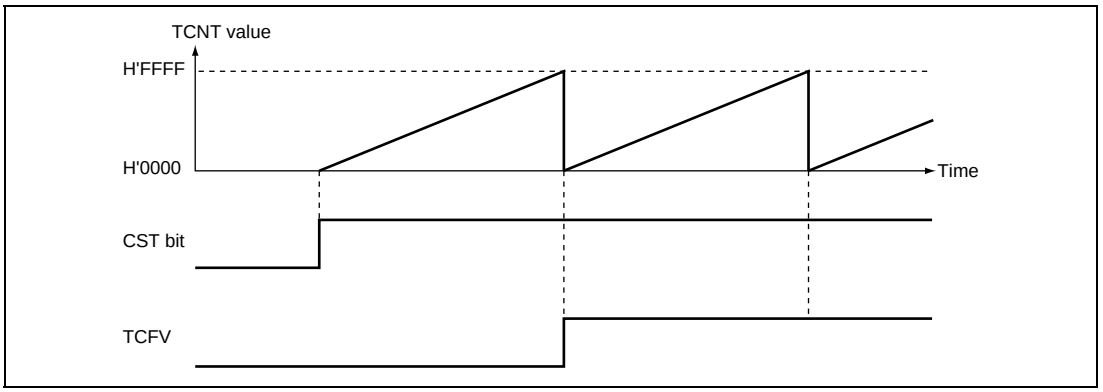


Figure 12.7 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, TCNT for the relevant channel performs periodic count operation. TGR for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 12.8 illustrates periodic counter operation.

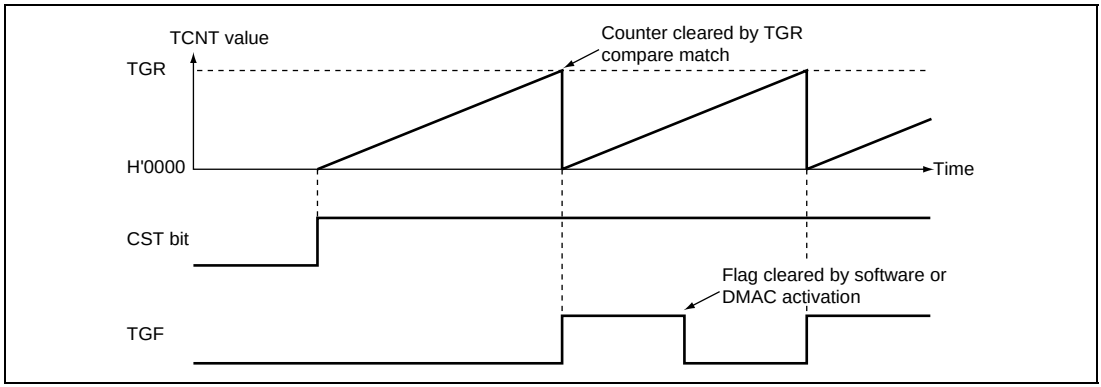


Figure 12.8 Periodic Counter Operation

Waveform Output by Compare Match:

The TPU can perform 0, 1, or toggle output from the corresponding output pin using compare match.

- Example of Setting Procedure for Waveform Output by Compare Match

  Figure 12.9 shows an example of the setting procedure for waveform output by compare match.

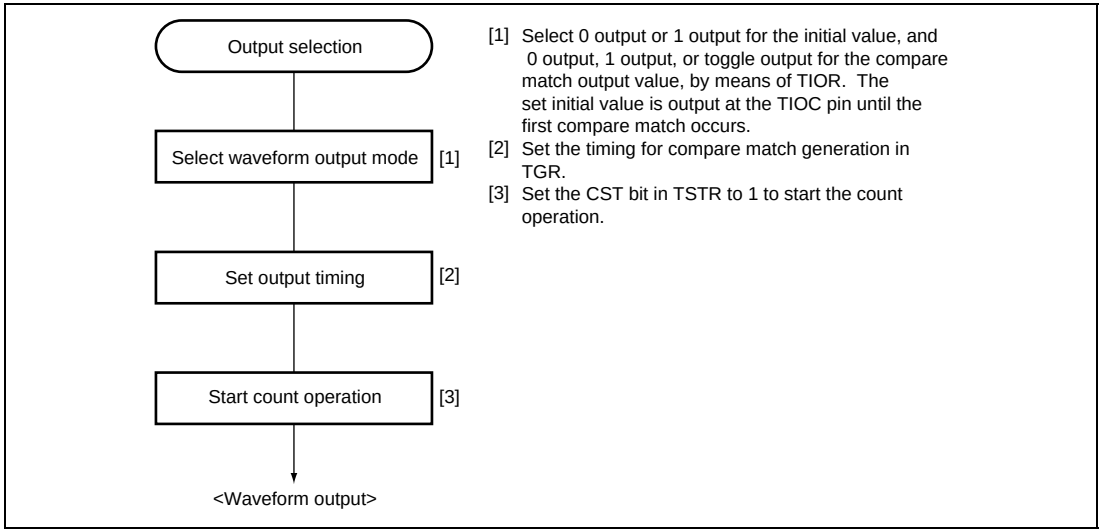


Figure 12.9 Example of Setting Procedure for Waveform Output by Compare Match

- Examples of Waveform Output Operation

  Figure 12.10 shows an example of 0 output/1 output.

  In this example TCNT has been designated as a free-running counter, and settings have been made so that 1 is output by compare match A, and 0 is output by compare match B. When the set level matches the pin level, the pin level does not change.

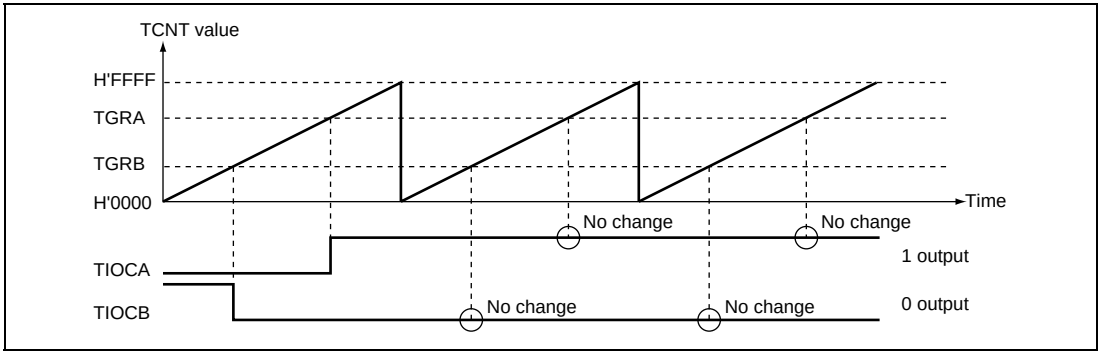


Figure 12.10 Example of 0 Output/1 Output Operation

Figure 12.11 shows an example of toggle output.

In this example TCNT has been designated as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

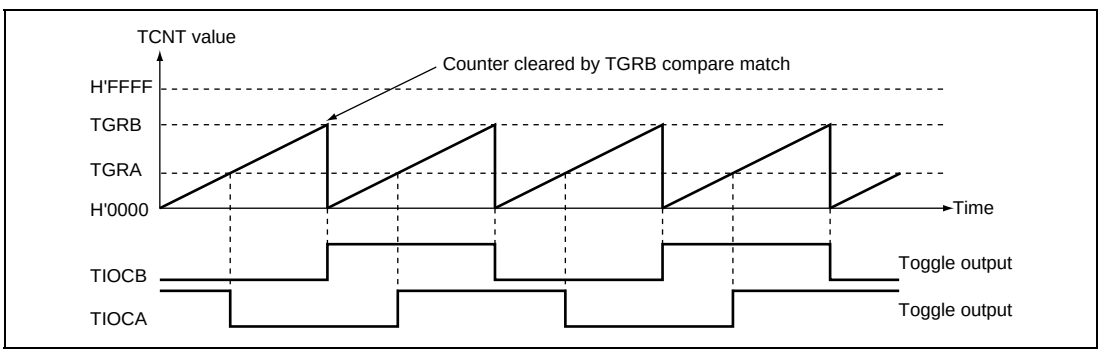


Figure 12.11 Example of Toggle Output Operation

Input Capture Function:

The TCNT value can be transferred to TGR on detection of the TIOC pin input edge. Rising edge, falling edge, or both edges can be selected as the detected edge.

Note: When another channel's counter input clock is used as the input capture input for channel 0, ϕ/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if ϕ/1 is selected. Another channel's counter input clock or compare-match signal can be used as the input capture source for channels 0 and 1.

- Example of Setting Procedure for Input Capture Operation

  Figure 12.12 shows an example of the setting procedure for input capture operation.

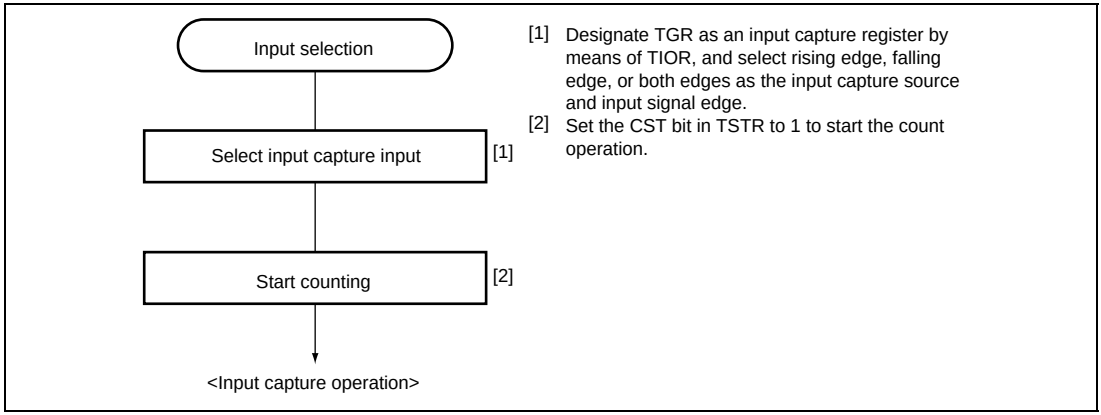


Figure 12.12 Example of Setting Procedure for Input Capture Operation

- Example of Input Capture Operation

  Figure 12.13 shows an example of input capture operation.

  In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

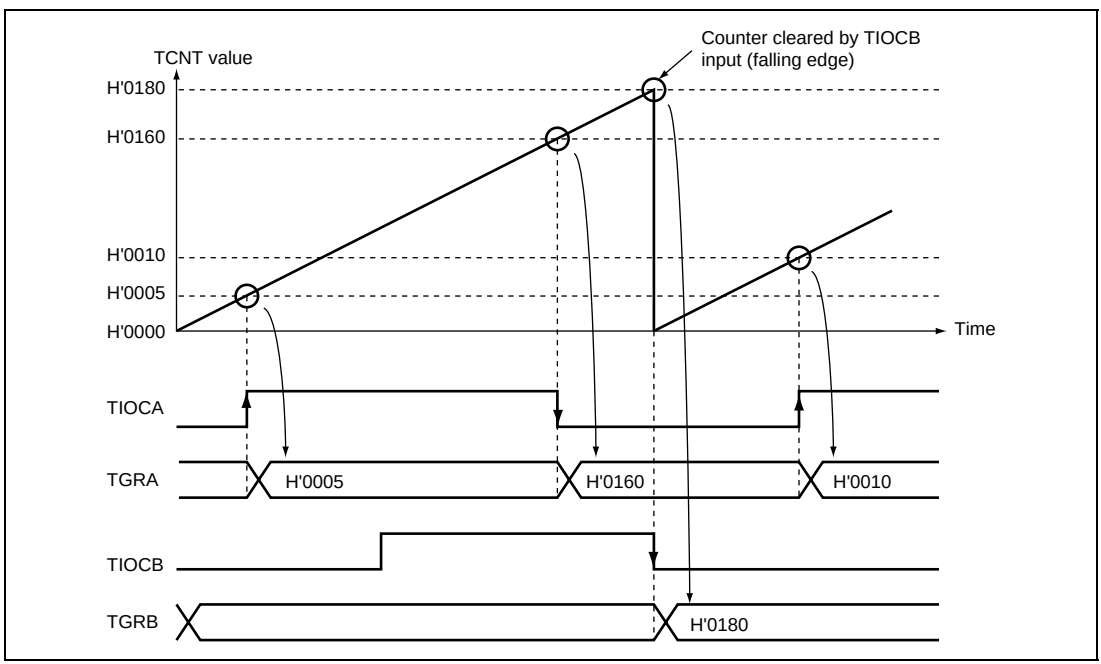


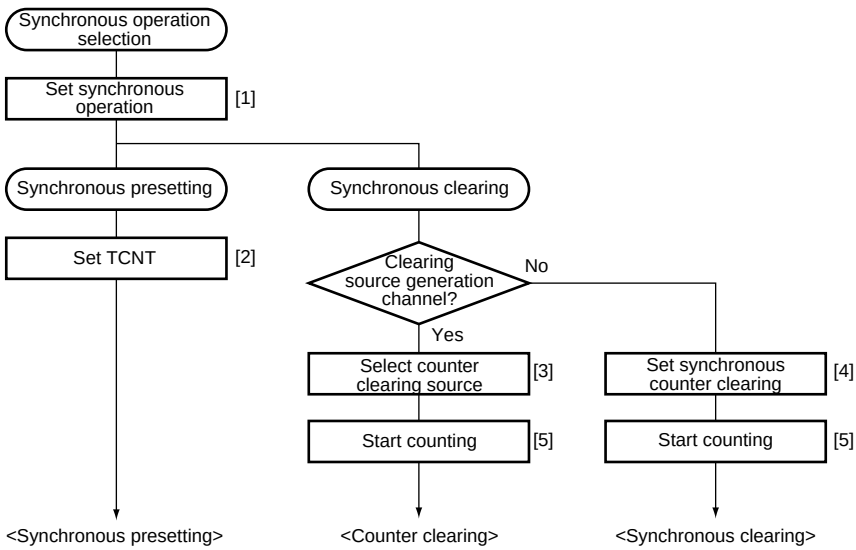
Figure 12.13 Example of Input Capture Operation

12.5.2 Synchronous Operation

In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing). Synchronous operation enables TGR to be incremented with respect to a single time base. Channels 0 to 2 can all be designated for synchronous operation.

Example of Synchronous Operation Setting Procedure:

Figure 12.14 shows an example of the synchronous operation setting procedure.



[1] Set 1 to the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.
[2] When TCNT of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT.
[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.
[5] Set 1 to the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 12.14 Example of Synchronous Operation Setting Procedure

Example of Synchronous Operation:

Figure 12.15 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGRB_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOCA0, TIOCA1, and TIOCA2. At this time, synchronous presetting, and synchronous clearing by TGRB_0 compare match, are performed for the TCNT counters in channels 0 to 2, and the data set in TGRB_0 is used as the PWM cycle.

For details on PWM modes, see section 12.5.5, PWM Modes.

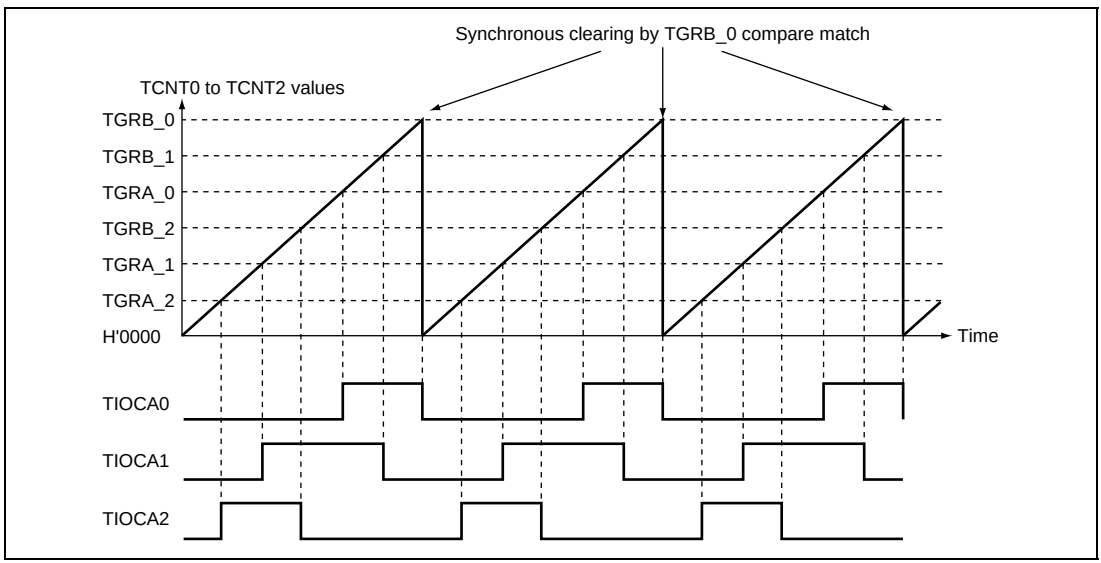


Figure 12.15 Example of Synchronous Operation

12.5.3 Buffer Operation

Buffer operation, provided for channel 0, enables TGRC and TGRD to be used as buffer registers. Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register. Table 12.17 shows the register combinations used in buffer operation.

Table 12.17 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register. This operation is illustrated in figure 12.16.

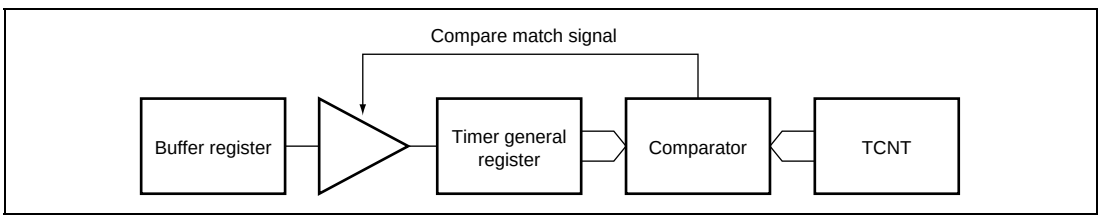


Figure 12.16 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously stored in the timer general register is transferred to the buffer register. This operation is illustrated in figure 12.17.

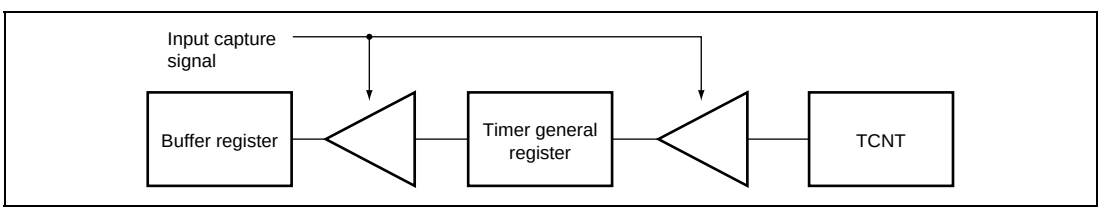


Figure 12.17 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure:

Figure 12.18 shows an example of the buffer operation setting procedure.

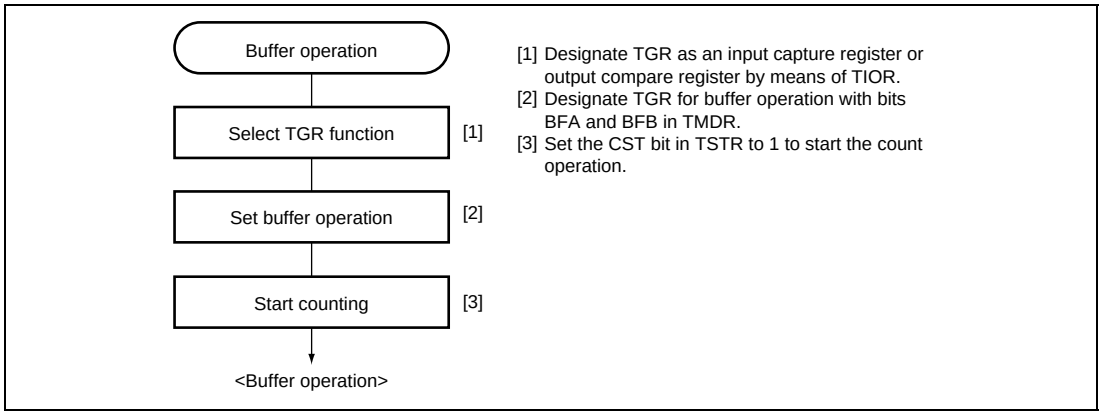


Figure 12.18 Example of Buffer Operation Setting Procedure

Examples of Buffer Operation:

- When TGR is Output Compare Register

  Figure 12.19 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B.

  As buffer operation has been set, when compare match A occurs the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

  For details on PWM modes, see section 12.5.5, PWM Modes.

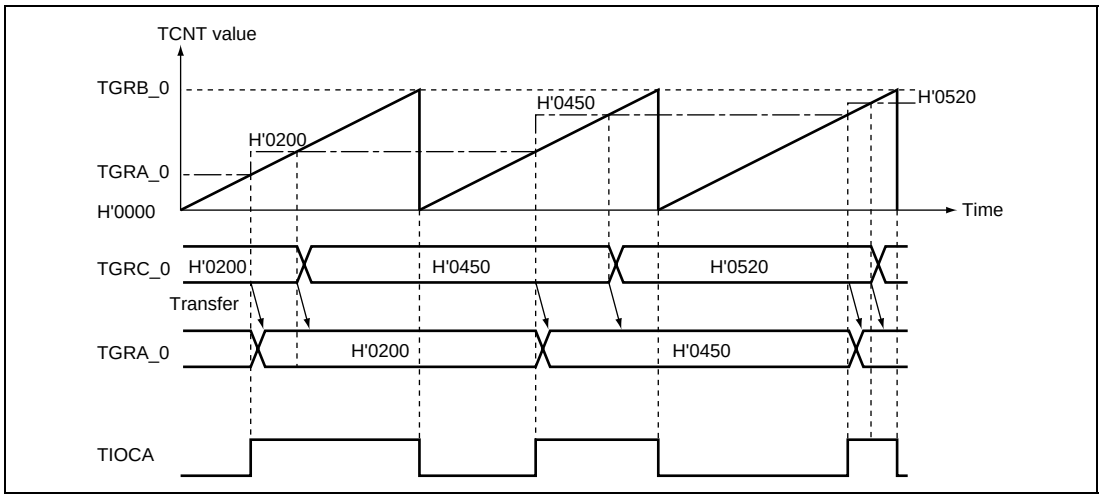


Figure 12.19 Example of Buffer Operation (1)

- When TGR is Input Capture Register

  Figure 12.20 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

  Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

  As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

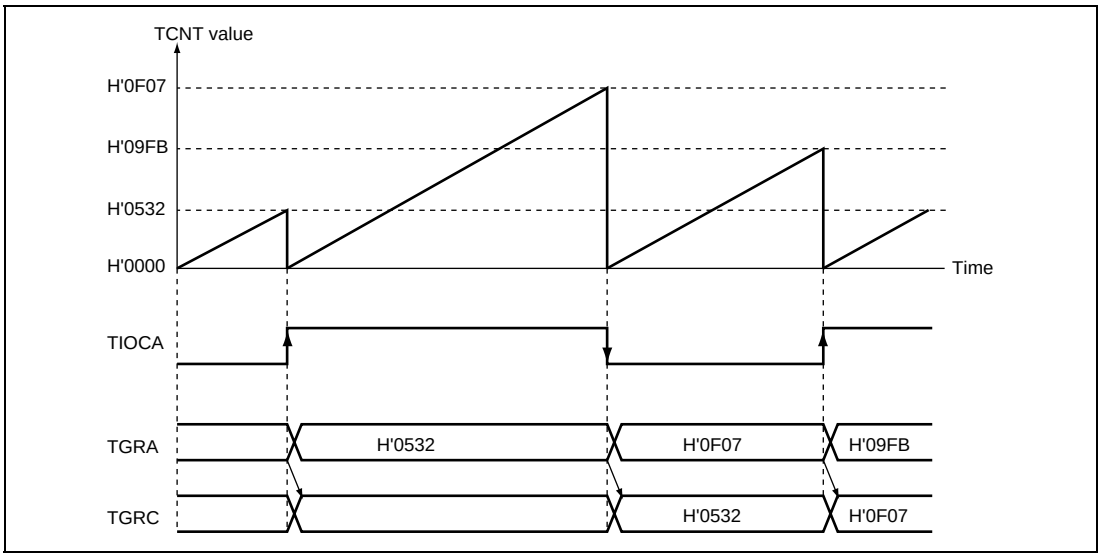


Figure 12.20 Example of Buffer Operation (2)

12.5.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.

This function works by counting the channel 1 counter clock at overflow/underflow of TCNT_2 as set in bits TPSC2 to TPSC0 in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase counting mode.

Table 12.18 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1, the counter clock setting is invalid and the counter operates independently in phase counting mode.

Table 12.18 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2

Example of Cascaded Operation Setting Procedure:

Figure 12.21 shows an example of the setting procedure for cascaded operation.

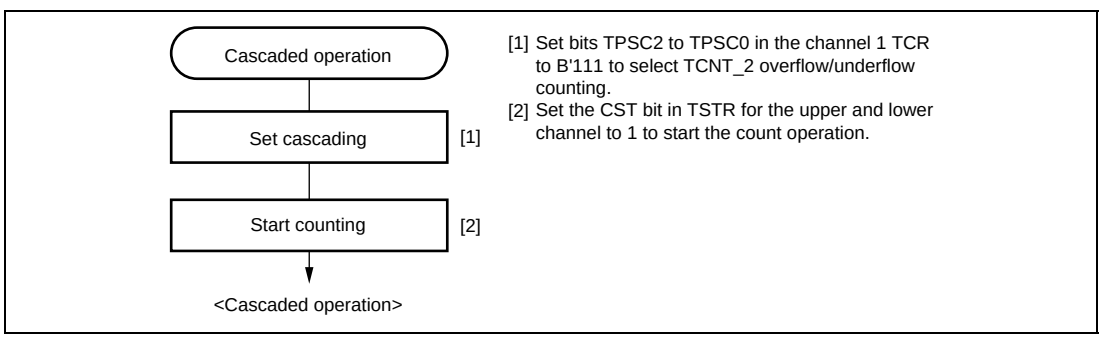


Figure 12.21 Cascaded Operation Setting Procedure

Examples of Cascaded Operation:

Figure 12.22 illustrates the operation when counting upon TCNT_2 overflow/underflow has been set for TCNT_1, TGRA_1 and TGRA_2 have been designated as input capture registers, and the TIOC pin rising edge has been selected.

When a rising edge is input to the TIOCA1 and TIOCA2 pins simultaneously, the upper 16 bits of the 32-bit data are transferred to TGRA_1, and the lower 16 bits to TGRA_2.

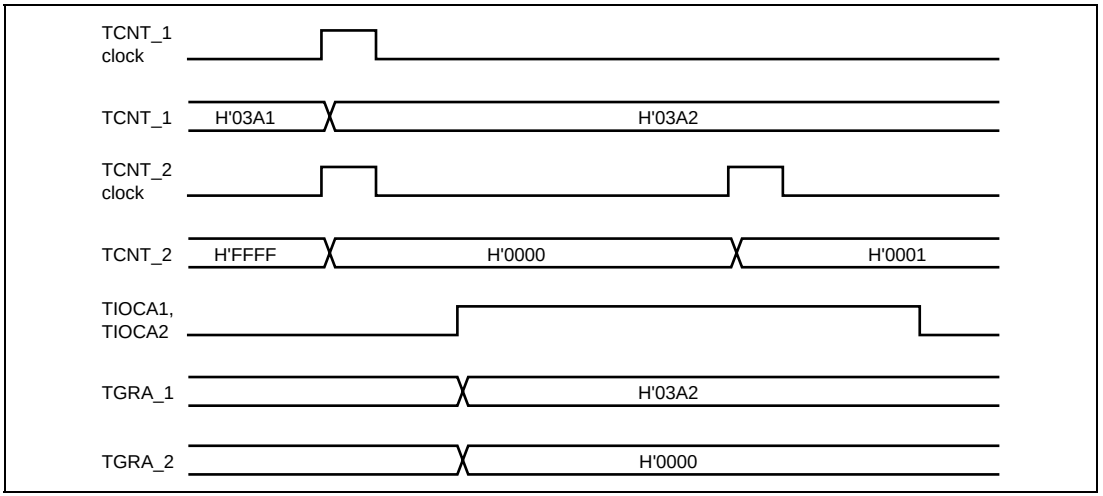


Figure 12.22 Example of Cascaded Operation (1)

Figure 12.23 illustrates the operation when counting upon TCNT_2 overflow/underflow has been set for TCNT_1, and phase counting mode has been designated for channel 2.

TCNT_1 is incremented by TCNT_2 overflow and decremented by TCNT_2 underflow.

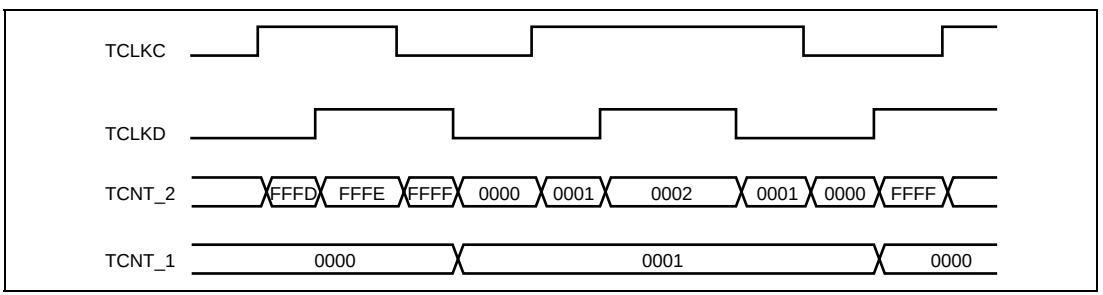


Figure 12.23 Example of Cascaded Operation (2)

12.5.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. 0, 1, or toggle output can be selected as the output level in response to compare match of each TGR.

Settings of TGR registers can output a PWM waveform in the range of 0% to 100% duty.

Designating TGR compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

PWM Mode 1:

PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The value specified by bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and the value specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR is output at compare matches B and D. The initial output value is the value set in TGRA or TGRC. When the set values of paired TGRs are identical, the output value does not change even if a compare match occurs.

In PWM mode 1, a maximum 4-phase PWM output is possible.

PWM Mode 2:

PWM output is generated using one TGR as the periodic register and the others as duty registers. The value specified in TIOR is output by means of compare matches. Upon counter clearing by a synchronous register compare match, the output value of each pin is the initial value set in TIOR. When the set values of the periodic and duty registers are identical, the output value does not change even if a compare match occurs.

In PWM mode 2, a maximum 7-phase PWM output is possible by combined use with synchronous operation.

The correspondence between PWM output pins and registers is shown in table 12.19.

Table 12.19 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Registers**	**PWM Mode 1**	**PWM Mode 2**
0	TGRA_0	TIOCA0	TIOCA0
	TGRB_0		TIOCB0
	TGRC_0	TIOCC0	TIOCC0
	TGRD_0		TIOCD0
1	TGRA_1	TIOCA1	TIOCA1
	TGRB_1		TIOCB1
2	TGRA_2	TIOCA2	TIOCA2
	TGRB_2		TIOCB2

Note: In PWM mode 2, PWM output is not possible for TGR in which the period is set.

- Example of PWM Mode Setting Procedure

  Figure 12.24 shows an example of the PWM mode setting procedure.

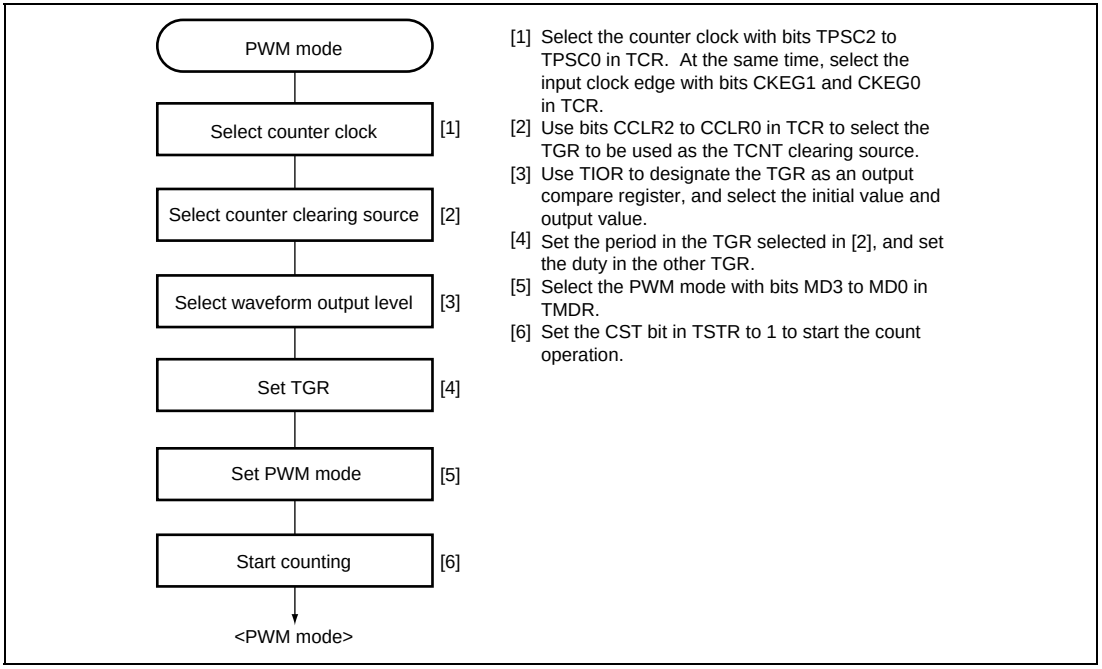


Figure 12.24 Example of PWM Mode Setting Procedure

- Examples of PWM Mode Operation

  Figure 12.25 shows an example of PWM mode 1 operation.

  In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value. In this case, the value set in TGRA is used as the period, and the value set in TGRB as the duty.

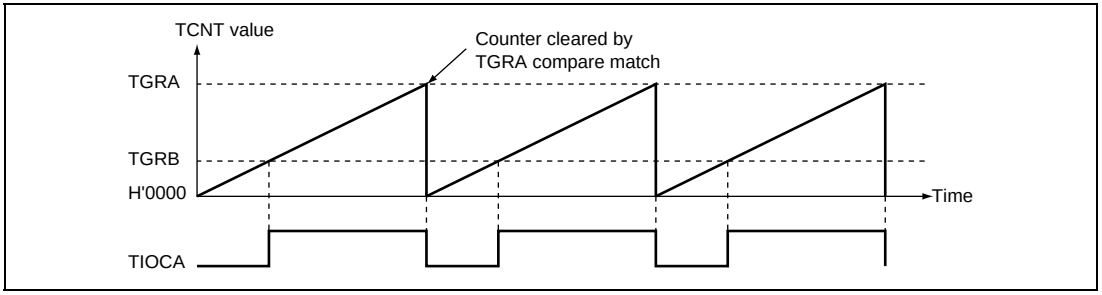


Figure 12.25 Example of PWM Mode Operation (1)

Figure 12.26 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGRB_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA_0 to TGRD_0, TGRA_1), to output a 5-phase PWM waveform. In this case, the value set in TGRB_1 is used as the period, and the values set in the other TGRs as the duty.

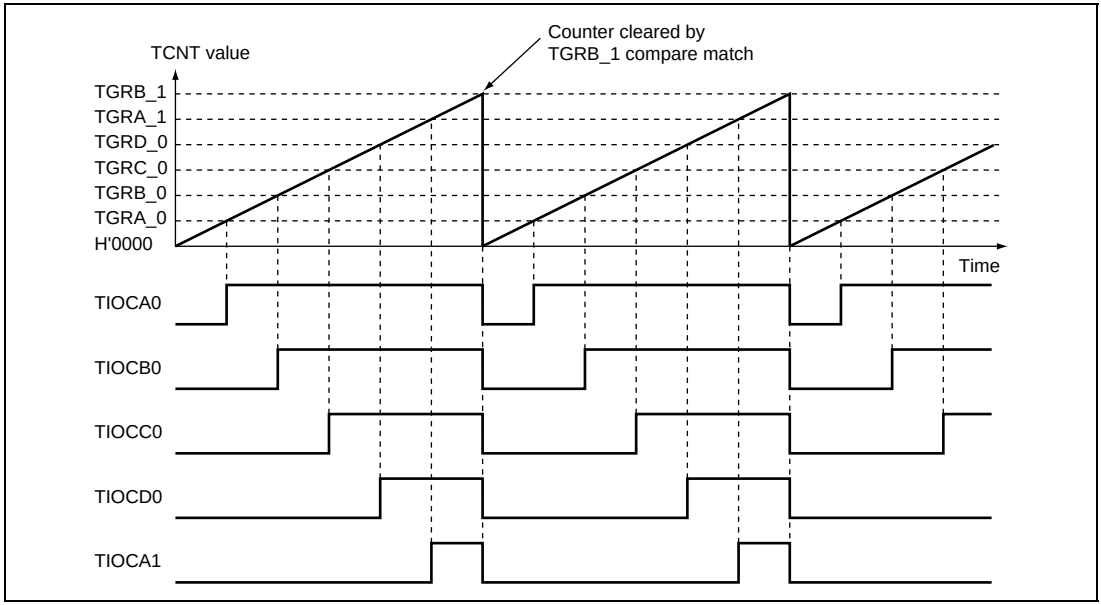


Figure 12.26 Example of PWM Mode Operation (2)

Figure 12.27 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

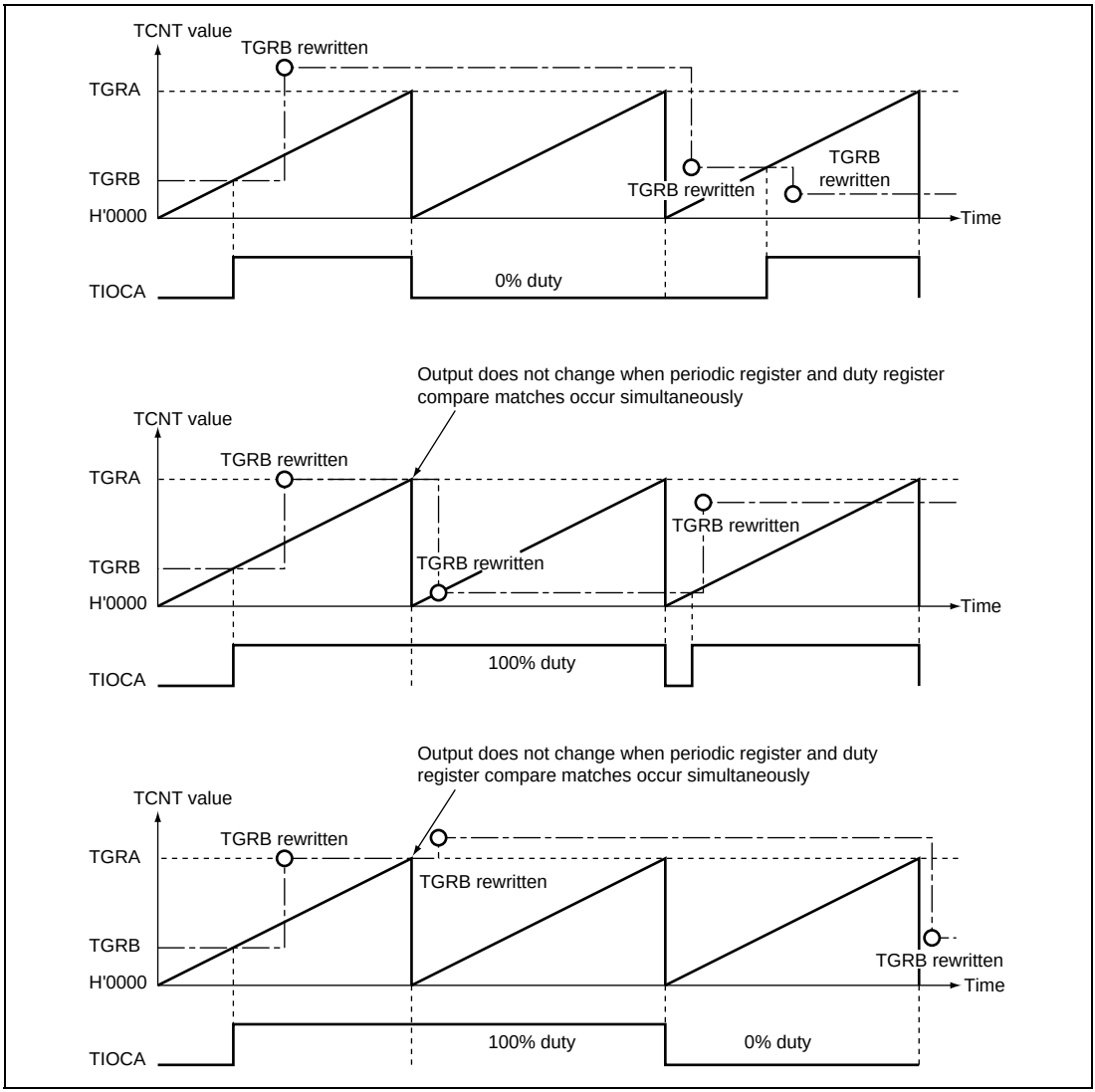


Figure 12.27 Example of PWM Mode Operation (3)

12.5.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1 and 2.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the settings of bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in TCR. However, since the functions of bits CCLR1 and CCLR0 in TCR, and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.

When an overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when an underflow occurs while TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.

Table 12.20 shows the correspondence between external clock pins and channels.

Table 12.20 Clock Input Pins for Phase Counting Mode

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 is set to phase counting mode	TCLKA	TCLKB
When channel 2 is set to phase counting mode	TCLKC	TCLKD

Example of Setting Procedure for Phase Counting Mode:

Figure 12.28 shows an example of the setting procedure for phase counting mode.

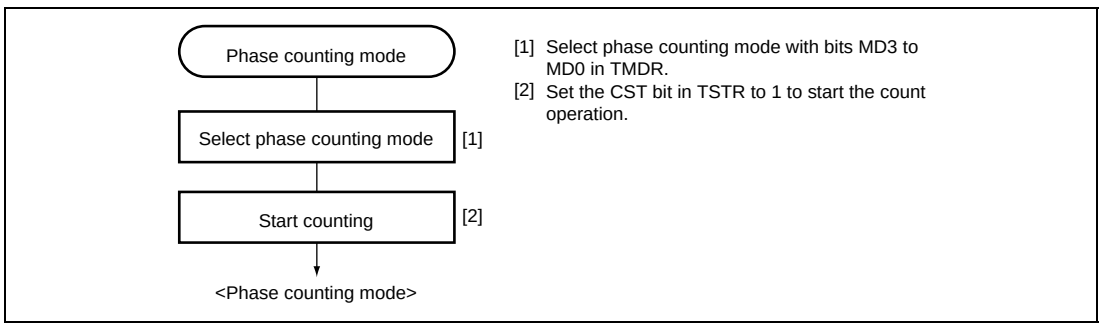


Figure 12.28 Example of Setting Procedure for Phase Counting Mode

Examples of Phase Counting Mode Operation:

In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

- Phase Counting Mode 1

  Figure 12.29 shows an example of phase counting mode 1 operation, and table 12.21 summarizes the TCNT up/down-count conditions.

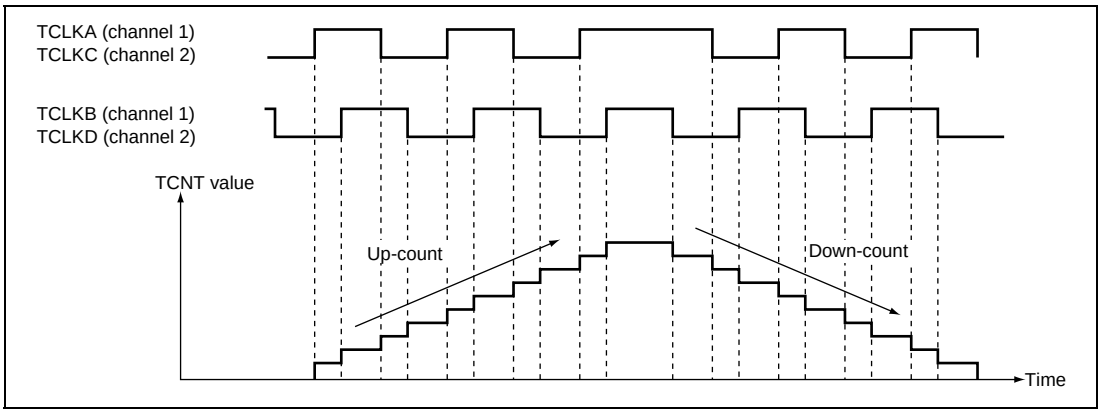


Figure 12.29 Example of Phase Counting Mode 1 Operation

Table 12.21 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	
↓	Low level	

[Legend]
↑ : Rising edge
↓ : Falling edge

- Phase Counting Mode 2

  Figure 12.30 shows an example of phase counting mode 2 operation, and table 12.22 summarizes the TCNT up/down-count conditions.

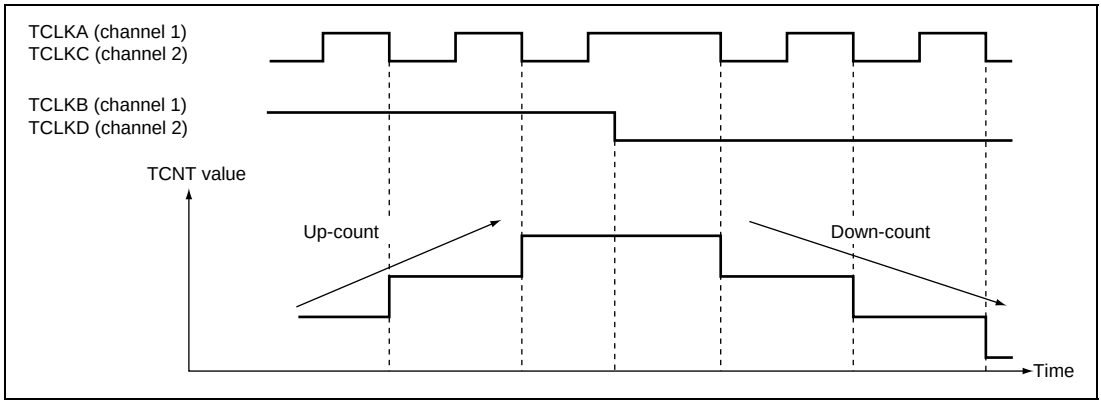


Figure 12.30 Example of Phase Counting Mode 2 Operation

Table 12.22 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Down-count

[Legend]
↑ : Rising edge
↓ : Falling edge

- Phase Counting Mode 3

  Figure 12.31 shows an example of phase counting mode 3 operation, and table 12.23 summarizes the TCNT up/down-count conditions.

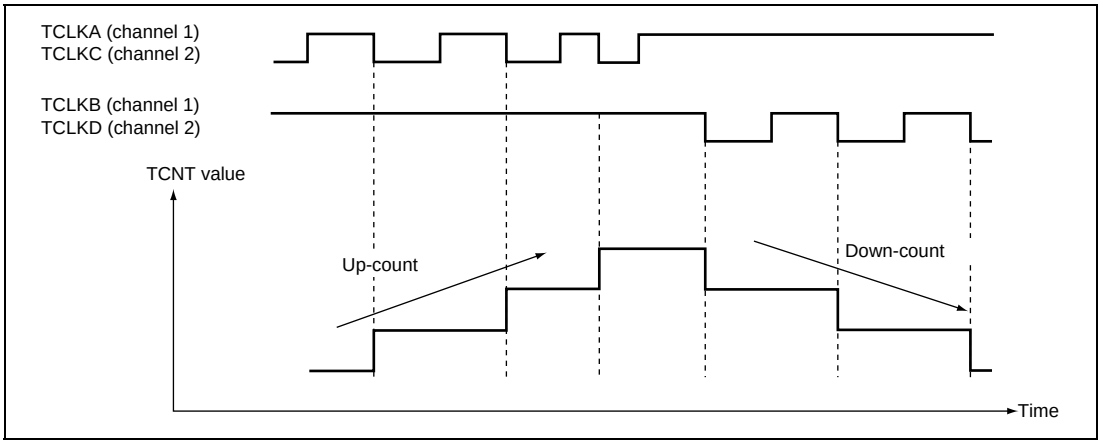


Figure 12.31 Example of Phase Counting Mode 3 Operation

Table 12.23 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Don't care

[Legend]
↑ : Rising edge
↓ : Falling edge

- Phase Counting Mode 4

  Figure 12.32 shows an example of phase counting mode 4 operation, and table 12.24 summarizes the TCNT up/down-count conditions.

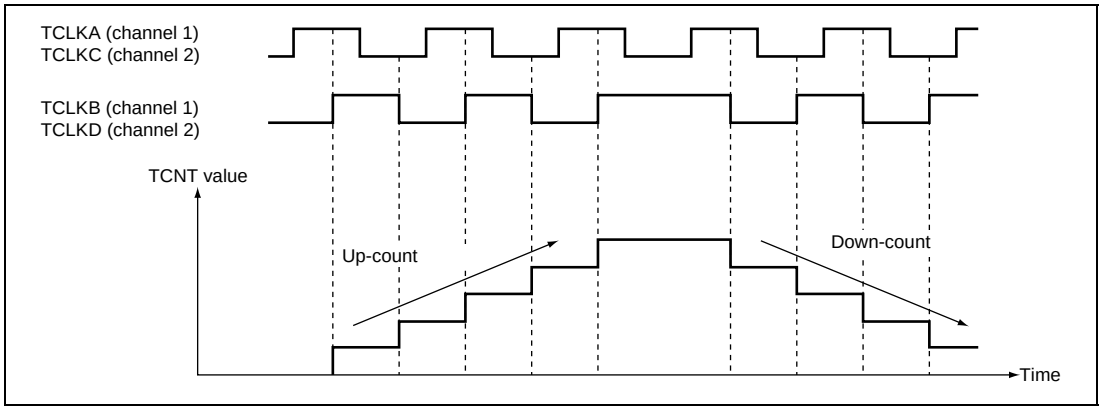


Figure 12.32 Example of Phase Counting Mode 4 Operation

Table 12.24 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channel 1) / TCLKC (Channel 2)	TCLKB (Channel 1) / TCLKD (Channel 2)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

[Legend]
↑ : Rising edge
↓ : Falling edge

12.6 Interrupt Sources

12.6.1 Interrupt Source and Priority

There are three kinds of TPU interrupt source: TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disable bit, allowing generation of interrupt request signals to be enabled or disabled individually.

When an interrupt source is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0. Relative channel priorities can be changed by the interrupt controller, but the priority within a channel is fixed. For details, see section 5, Interrupt Controller. Table 12.25 lists the TPU interrupt sources.

Table 12.25 TPU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	Priority*
0	TGI0A	TGRA_0 input capture/compare match	TGFA	High
	TGI0B	TGRB_0 input capture/compare match	TGFB	
	TGI0C	TGRC_0 input capture/compare match	TGFC	
	TGI0D	TGRD_0 input capture/compare match	TGFD	
	TCI0V	TCNT_0 overflow	TCFV	
1	TGI1A	TGRA_1 input capture/compare match	TGFA	
	TGI1B	TGRB_1 input capture/compare match	TGFB	
	TCI1V	TCNT_1 overflow	TCFV	
	TCI1U	TCNT_1 underflow	TCFU	
2	TGI2A	TGRA_2 input capture/compare match	TGFA	
	TGI2B	TGRB_2 input capture/compare match	TGFB	
	TCI2V	TCNT_2 overflow	TCFV	
	TCI2U	TCNT_2 underflow	TCFU	Low

Note: * This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Input Capture/Compare Match Interrupt:

An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has a total of eight input capture/compare match interrupts, four for channel 0, and two each for channels 1 and 2.

Overflow Interrupt:

An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The TPU has a total of three overflow interrupts, one for each channel.

Underflow Interrupt:

An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The TPU has a total of two underflow interrupts, one each for channels 1 and 2.

12.6.2 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match for a channel.

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to start A/D conversion is sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

In the TPU, a total of three TGRA input capture/compare match interrupts can be used as A/D conversion start sources, one for each channel.

12.7 Operation Timing

12.7.1 Input/Output Timing

TCNT Count Timing:

Figure 12.33 shows TCNT count timing in internal clock operation, and figure 12.34 shows TCNT count timing in external clock operation.

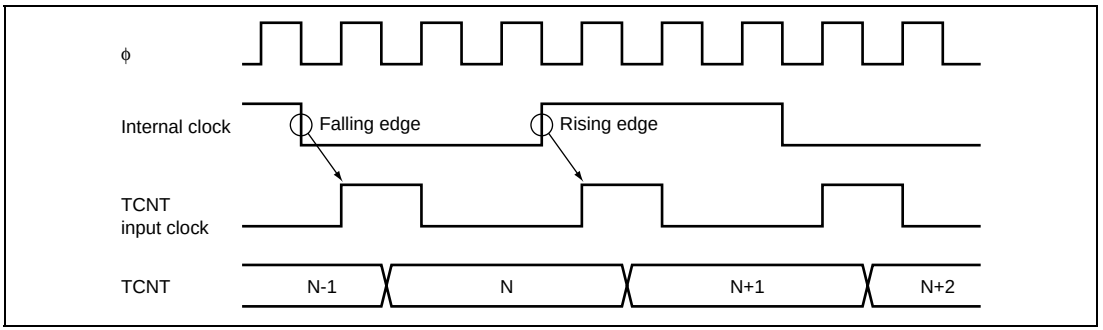


Figure 12.33 Count Timing in Internal Clock Operation

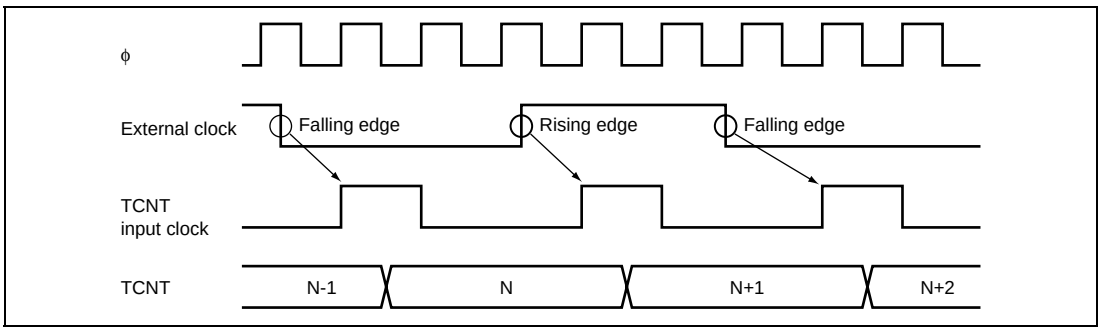


Figure 12.34 Count Timing in External Clock Operation

Output Compare Output Timing:

A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output (TIOC) pin. After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated. Figure 12.35 shows output compare output timing.

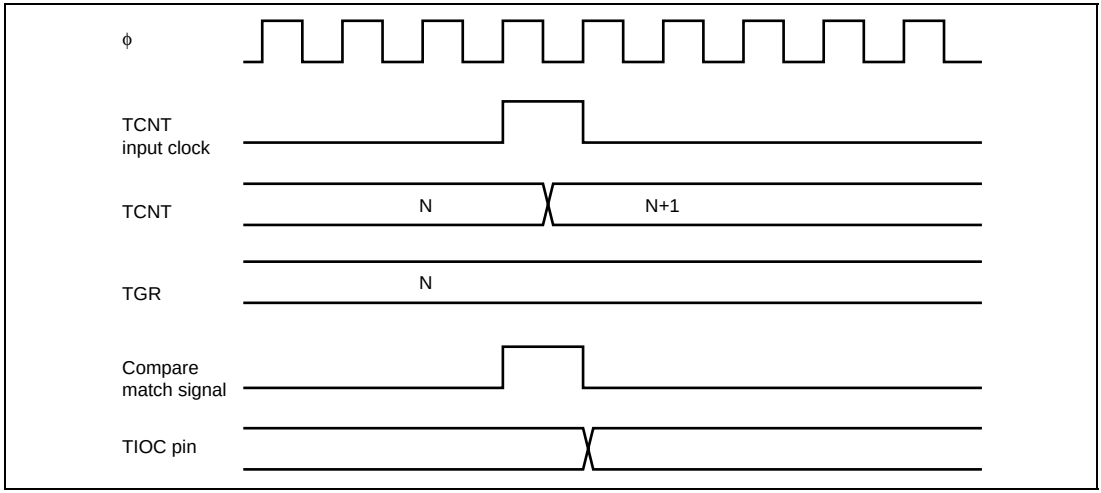


Figure 12.35 Output Compare Output Timing

Input Capture Signal Timing:

Figure 12.36 shows input capture signal timing.

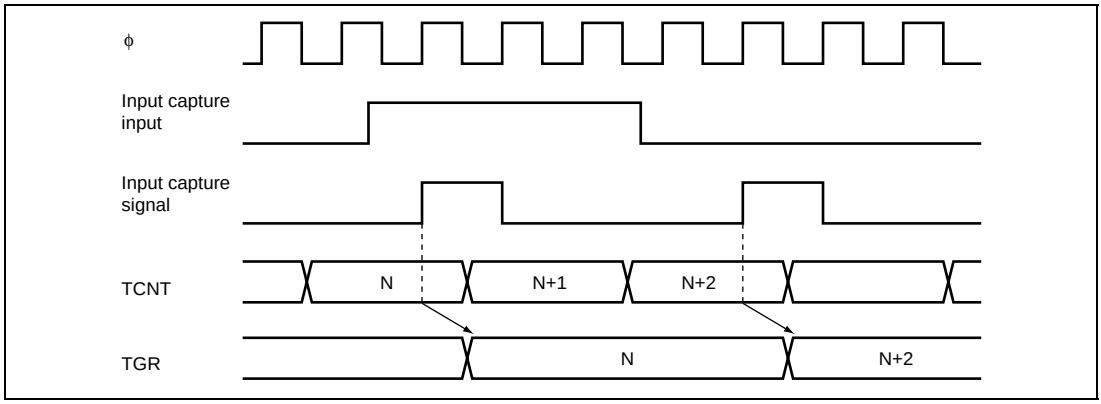


Figure 12.36 Input Capture Input Signal Timing

Timing for Counter Clearing by Compare Match/Input Capture:

Figure 12.37 shows the timing when counter clearing by compare match occurrence is specified, and figure 12.38 shows the timing when counter clearing by input capture occurrence is specified.

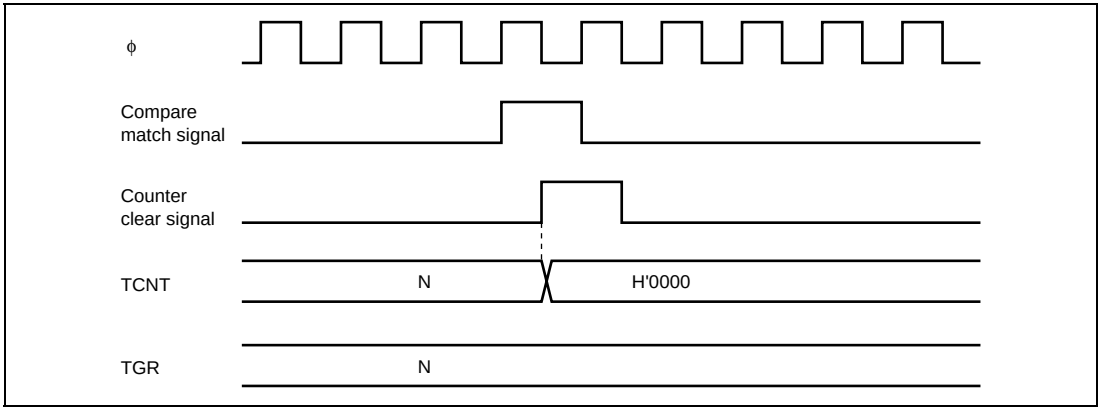


Figure 12.37 Counter Clear Timing (Compare Match)

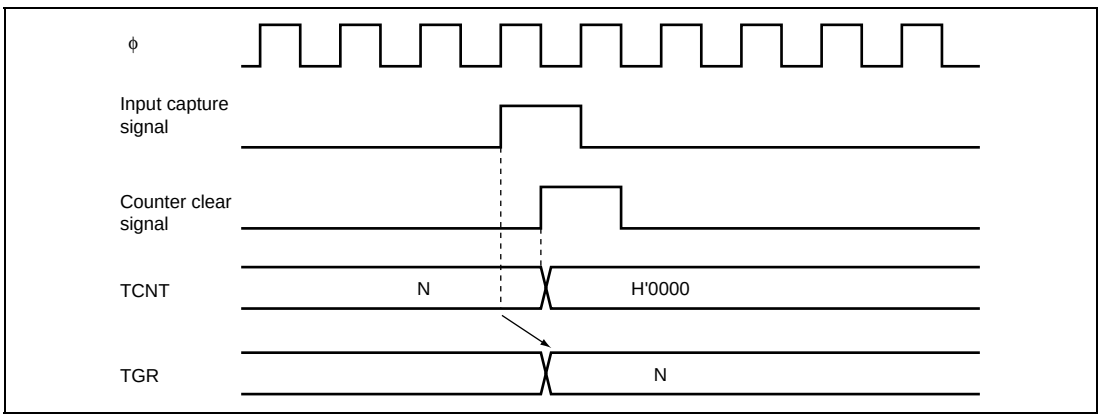


Figure 12.38 Counter Clear Timing (Input Capture)

Buffer Operation Timing:

Figures 12.39 and 12.40 show the timing in buffer operation.

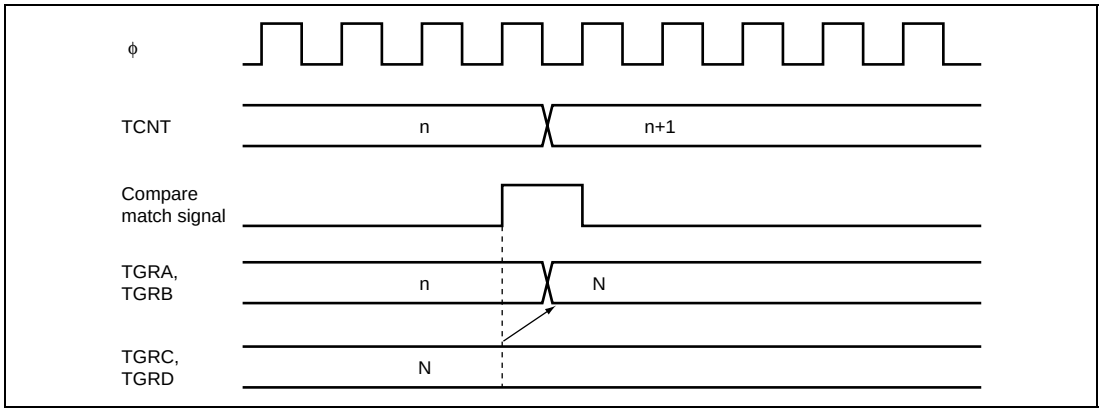


Figure 12.39 Buffer Operation Timing (Compare Match)

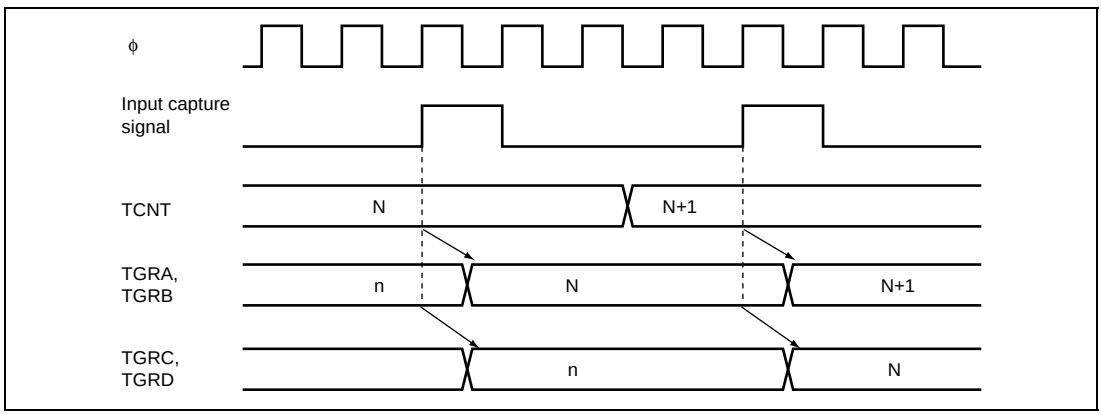


Figure 12.40 Buffer Operation Timing (Input Capture)

12.7.2 Interrupt Signal Timing

TGF Flag Setting Timing in Case of Compare Match:

Figure 12.41 shows the timing for setting of the TGF flag in TSR by compare match occurrence, and TGI interrupt request signal timing.

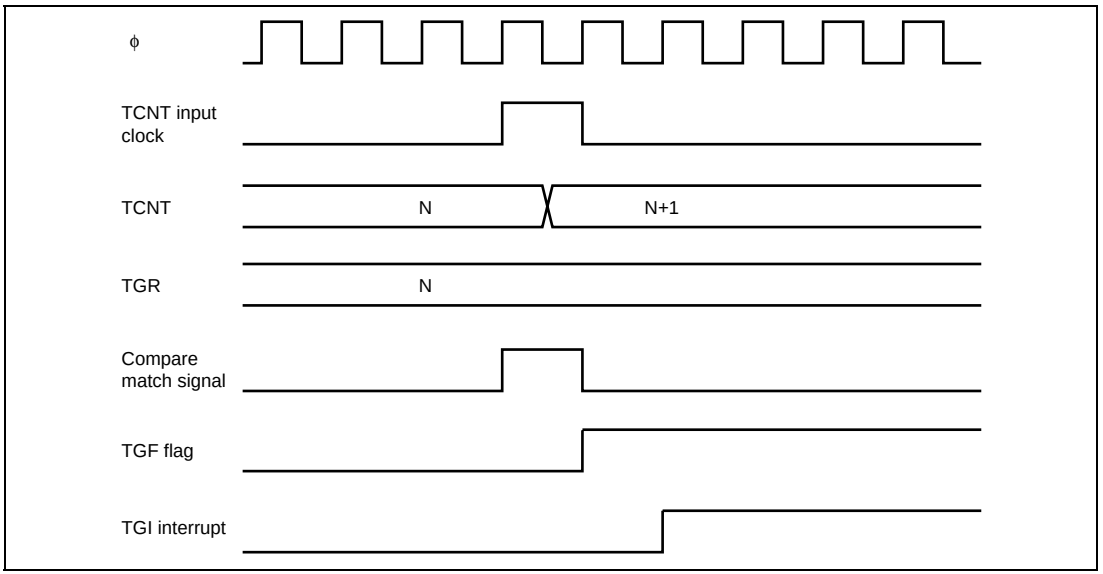


Figure 12.41 TGI Interrupt Timing (Compare Match)

TGF Flag Setting Timing in Case of Input Capture:

Figure 12.42 shows the timing for setting of the TGF flag in TSR by input capture occurrence, and TGI interrupt request signal timing.

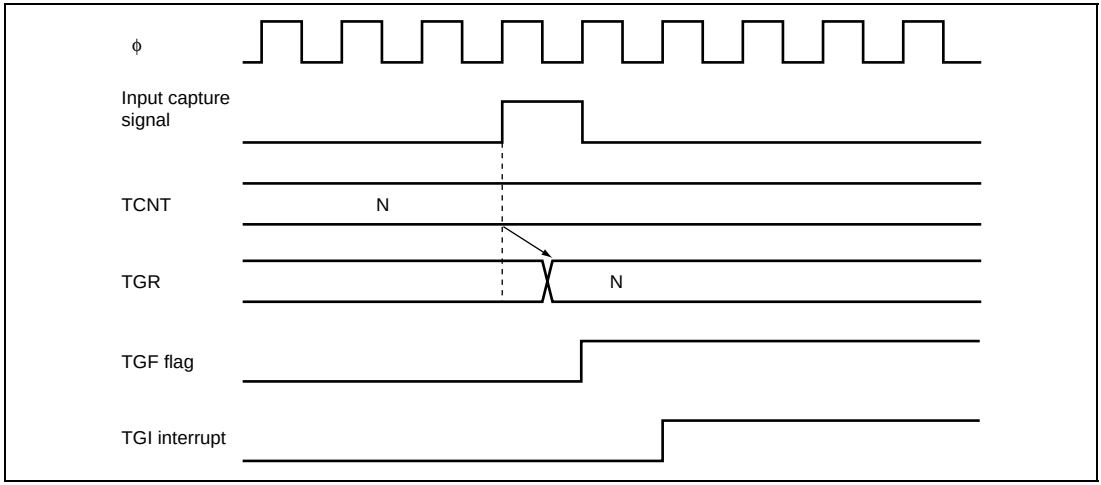


Figure 12.42 TGI Interrupt Timing (Input Capture)

TCFV Flag/TCFU Flag Setting Timing:

Figure 12.43 shows the timing for setting of the TCFV flag in TSR by overflow occurrence, and TCIV interrupt request signal timing. Figure 12.44 shows the timing for setting of the TCFU flag in TSR by underflow occurrence, and TCIU interrupt request signal timing.

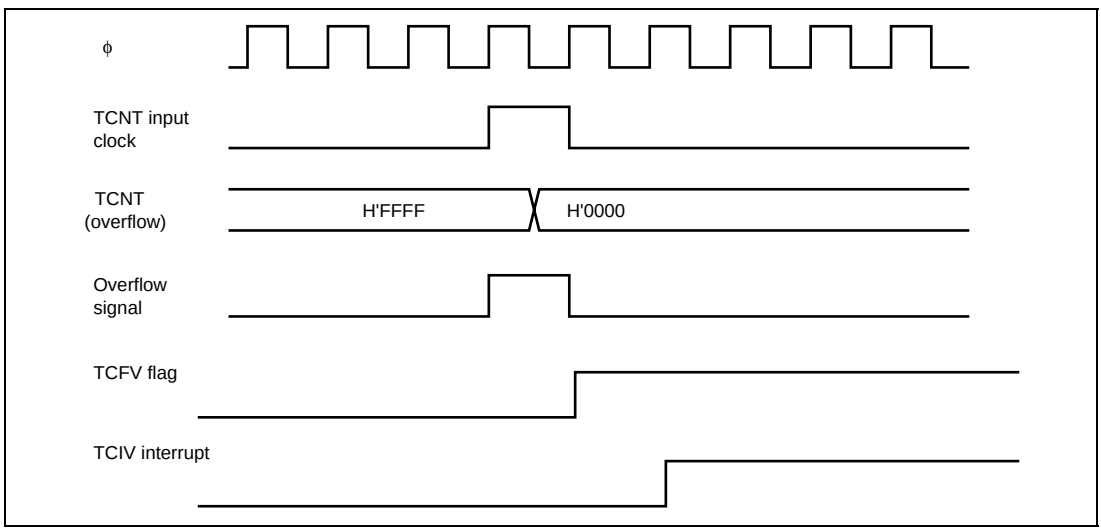


Figure 12.43 TCIV Interrupt Setting Timing

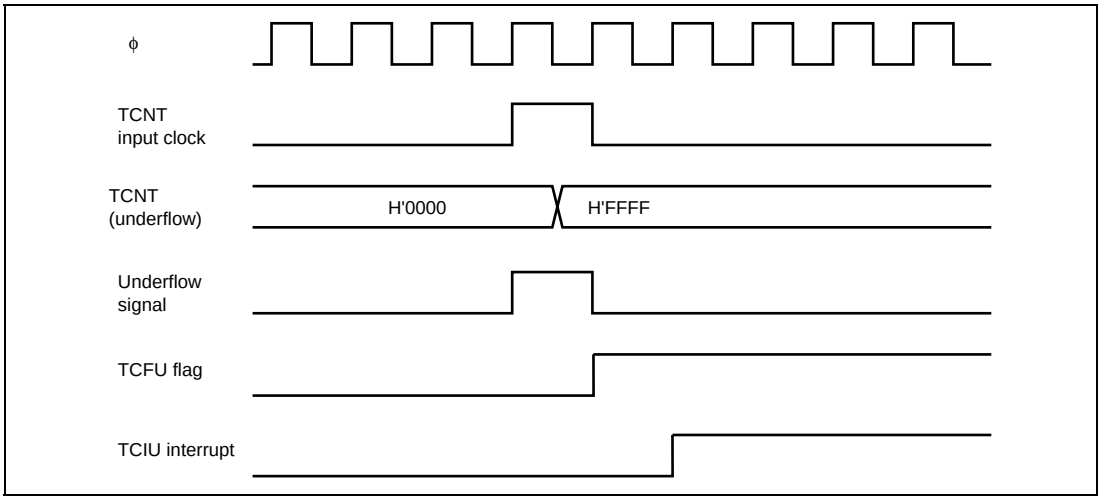


Figure 12.44 TCIU Interrupt Setting Timing

Status Flag Clearing Timing:

After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. Figure 12.45 shows the timing for status flag clearing by the CPU.

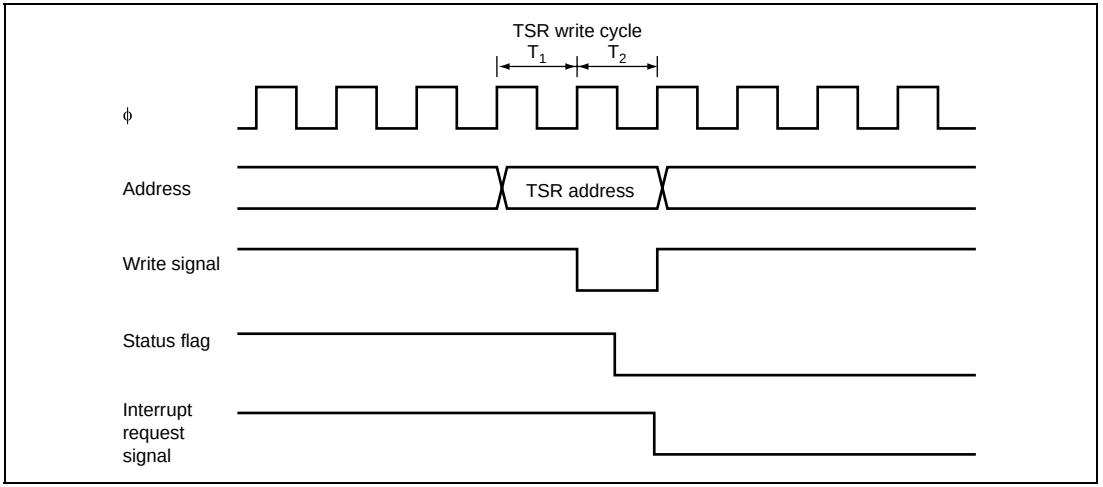


Figure 12.45 Timing for Status Flag Clearing by CPU

12.8 Usage Notes

Input Clock Restrictions:

The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 12.46 shows the input clock conditions in phase counting mode.

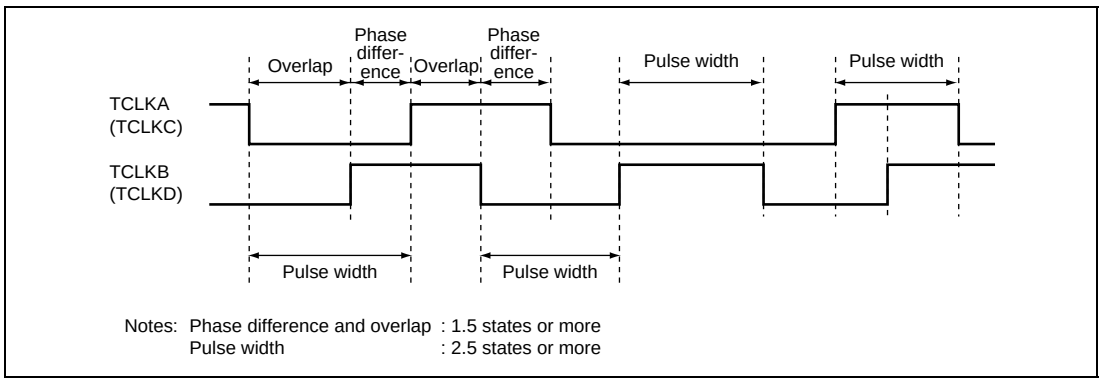


Figure 12.46 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

Caution on Period Setting:

When counter clearing by compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{\phi}{(N + 1)}$$

Where f: Counter frequency
ϕ: Operating frequency
N: TGR set value

Contention between TCNT Write and Clear Operations:

If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes priority and the TCNT write is not performed. Figure 12.47 shows the timing in this case.

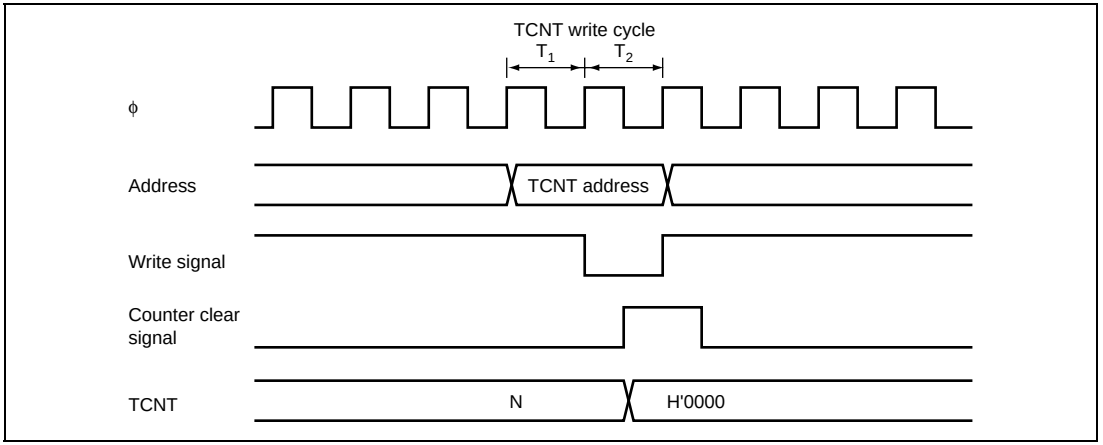


Figure 12.47 Contention between TCNT Write and Clear Operations

Contention between TCNT Write and Increment Operations:

If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes priority and TCNT is not incremented. Figure 12.48 shows the timing in this case.

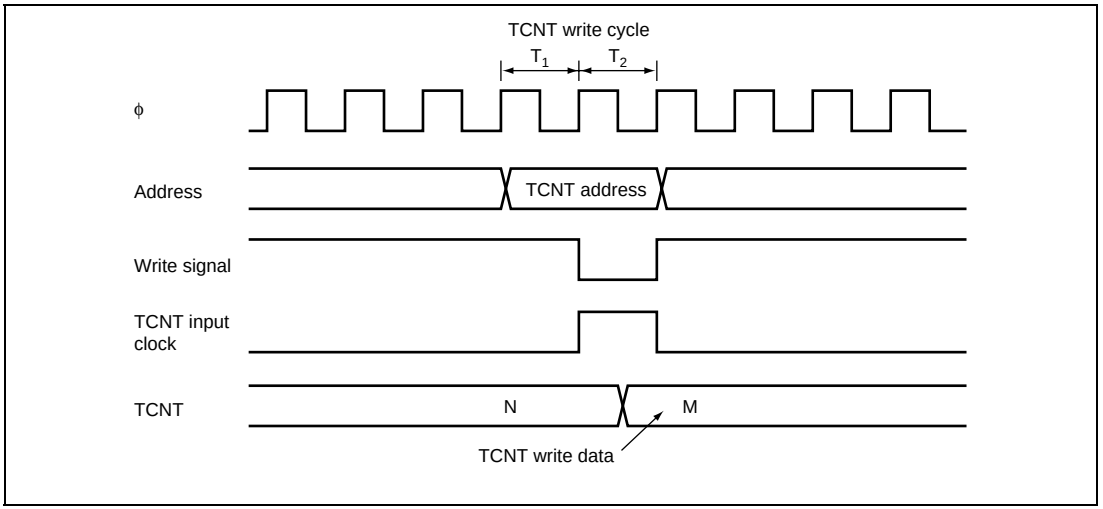


Figure 12.48 Contention between TCNT Write and Increment Operations

Contention between TGR Write and Compare Match:

If a compare match occurs in the T2 state of a TGR write cycle, the TGR write takes priority and the compare match signal is disabled. A compare match does not occur even if the same value as before is written to. Figure 12.49 shows the timing in this case.

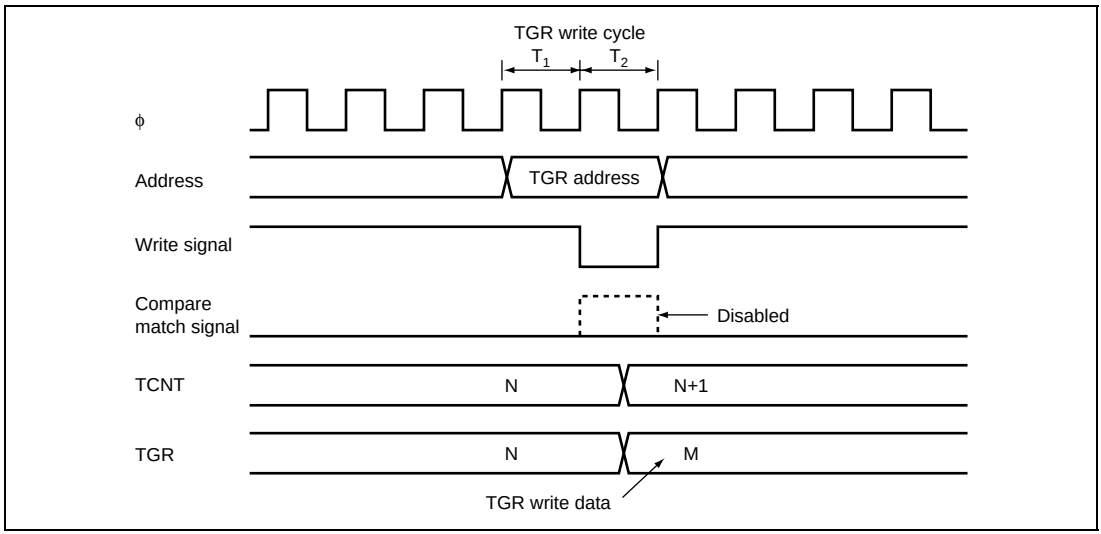


Figure 12.49 Contention between TGR Write and Compare Match

Contention between Buffer Register Write and Compare Match:

If a compare match occurs in the T2 state of a TGR write cycle, the data transferred to TGR by the buffer operation will be the write data. Figure 12.50 shows the timing in this case.

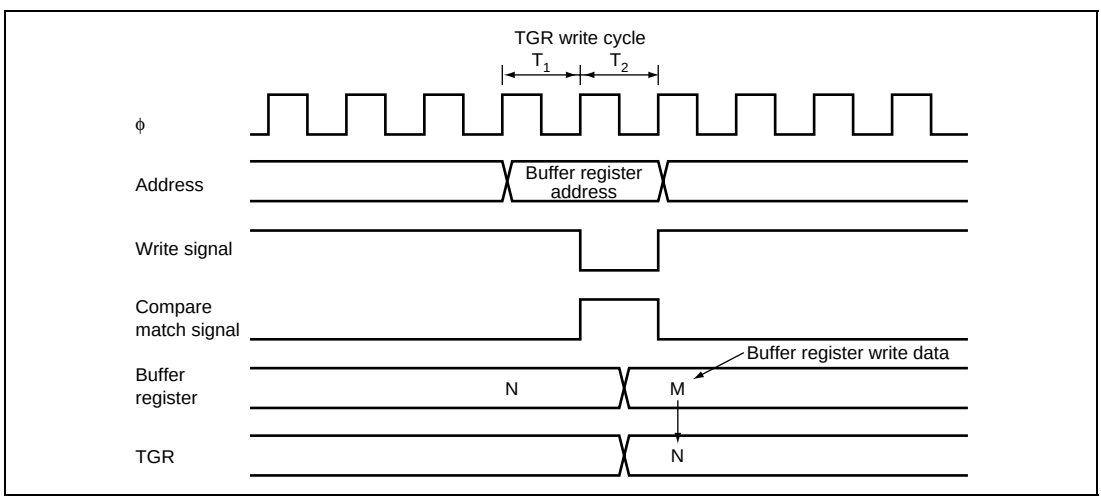


Figure 12.50 Contention between Buffer Register Write and Compare Match

Contention between TGR Read and Input Capture:

If the input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be the data after input capture transfer. Figure 12.51 shows the timing in this case.

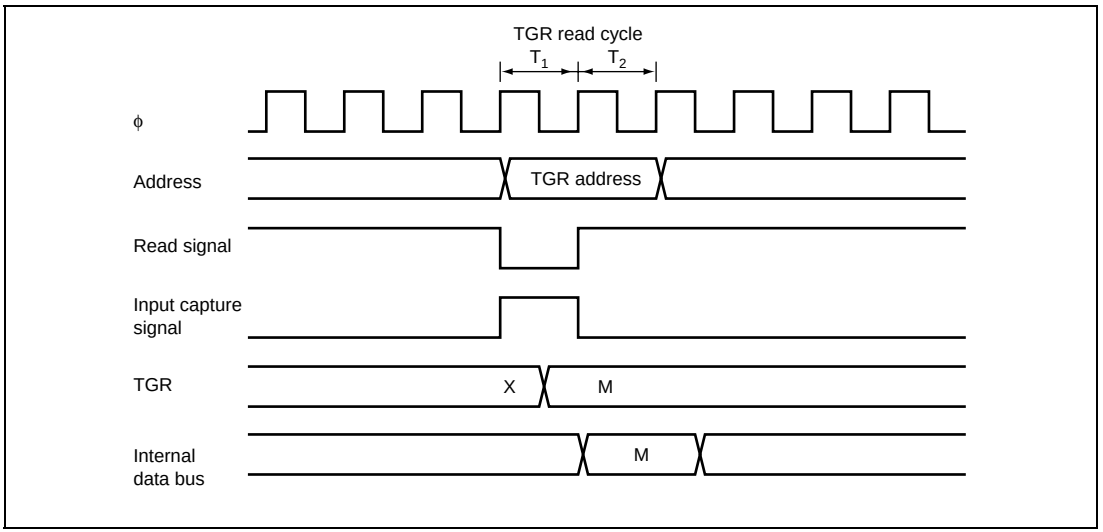


Figure 12.51 Contention between TGR Read and Input Capture

Contention between TGR Write and Input Capture:

If the input capture signal is generated in the T2 state of a TGR write cycle, the input capture operation takes priority and the write to TGR is not performed. Figure 12.52 shows the timing in this case.

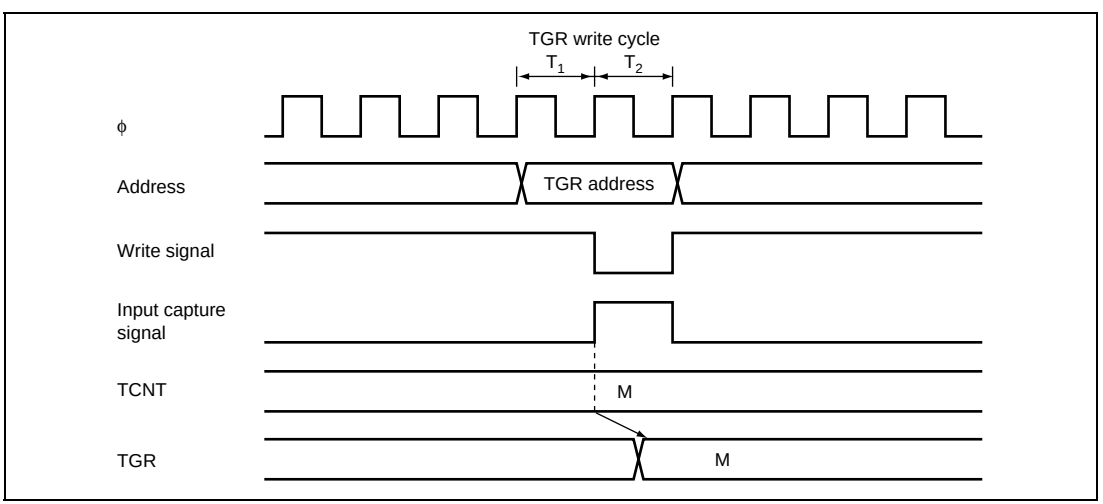


Figure 12.52 Contention between TGR Write and Input Capture

Contention between Buffer Register Write and Input Capture:

If the input capture signal is generated in the T2 state of a buffer register write cycle, the buffer operation takes priority and the write to the buffer register is not performed. Figure 12.53 shows the timing in this case.

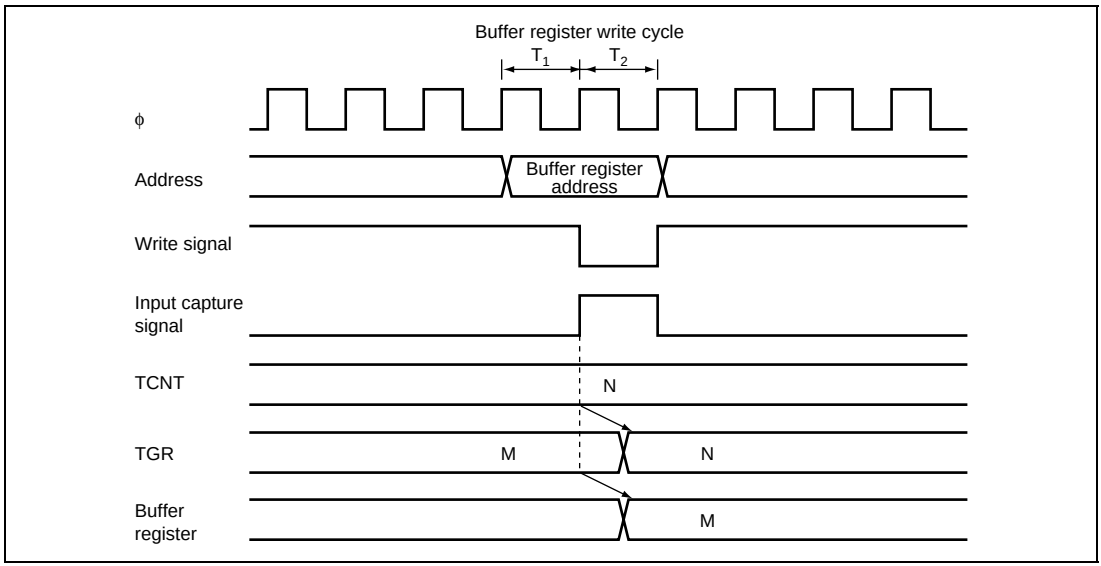


Figure 12.53 Contention between Buffer Register Write and Input Capture

Contention between Overflow/Underflow and Counter Clearing:

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes priority. Figure 12.54 shows the operation timing when a TGR compare match is specified as the clearing source, and H'FFFF is set in TGR.

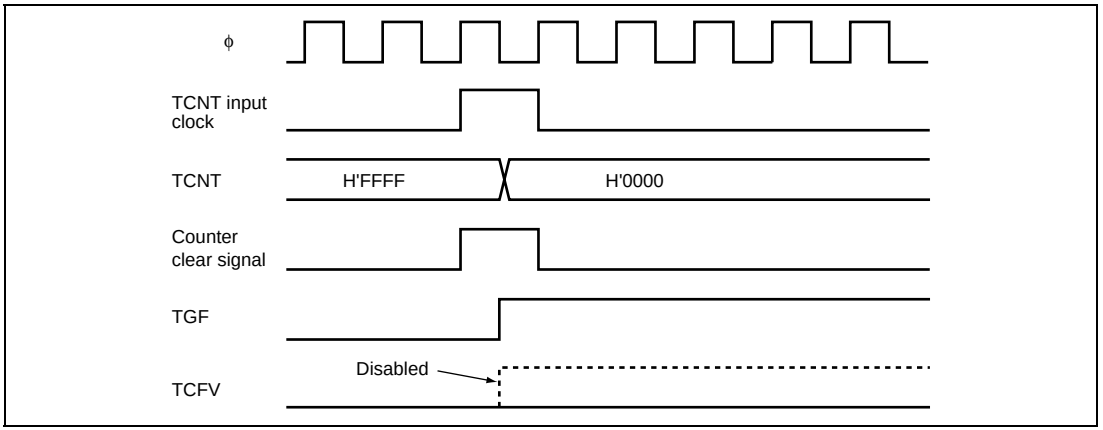


Figure 12.54 Contention between Overflow and Counter Clearing

Contention between TCNT Write and Overflow/Underflow:

If there is an up-count or down-count in the T2 state of a TCNT write cycle and overflow/underflow occurs, the TCNT write takes priority and the TCFV/TCFU flag in TSR is not set. Figure 12.55 shows the operation timing when there is contention between TCNT write and overflow.

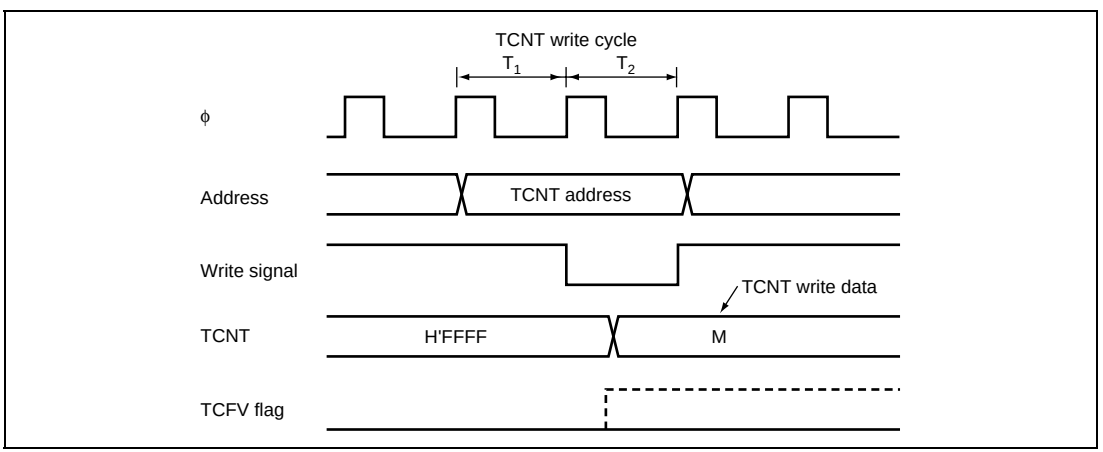


Figure 12.55 Contention between TCNT Write and Overflow

Multiplexing of I/O Pins:

In this LSI, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, and the TCLKD input pin with the TIOCB2 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

Interrupts in Module Stop Mode:

If module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source. Interrupts should therefore be disabled before entering module stop mode.

Section 13 Timer Connection

This LSI incorporates the timer connection with two channels. The timer connection allows interconnection by using the combination of input pins and I/O for a 16-bit free-running timer (FRT) and 8-bit timer (TMR1, TMRX, and TMRY). This capability can be used to implement complex functions such as PWM decoding and clamp waveform output.

13.1 Features

- Eight input pins and four output pins, all of which can be designated for phase.
  Five input pins for channel 0 and three input pins for channel 1
  Positive logic is assumed for all signals used within the timer connection facility.
- An edge-detection circuit is connected to the input pins, simplifying signal input detection.
- TMRX can be used for PWM input signal decoding.
- TMRX can be used for clamp waveform generation.
- An external clock signal divided by TMR1 can be used as the FRT capture input signal.
- An internal synchronization signal can be generated using the FRT and TMRY.
- A signal generated/modified using an input signal and timer connection can be selected and output.

This LSI incorporates the timer connection with two channels and some output pins are shared. Figure 13.1 shows a schematic diagram of the timer connection.

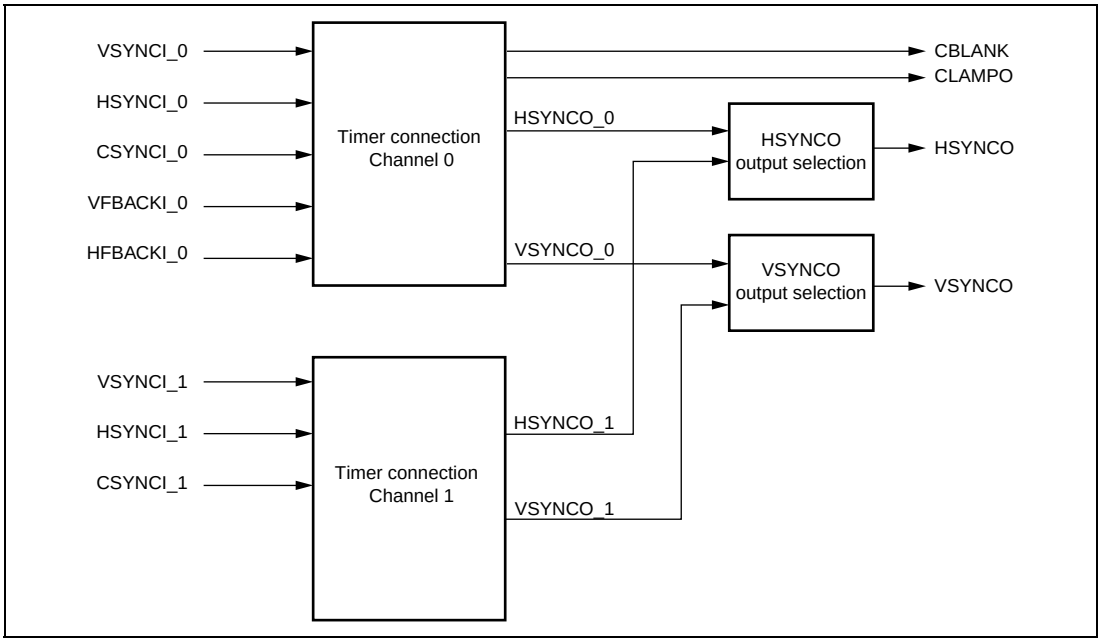


Figure 13.1 Schematic Diagram of Timer Connection

Figure 13.2 shows a block diagram of the timer connection. The configuration of the timer connection is the same in channels 0 and 1. However, the HFBACKI and VFBACKI inputs are not available in channel 1.

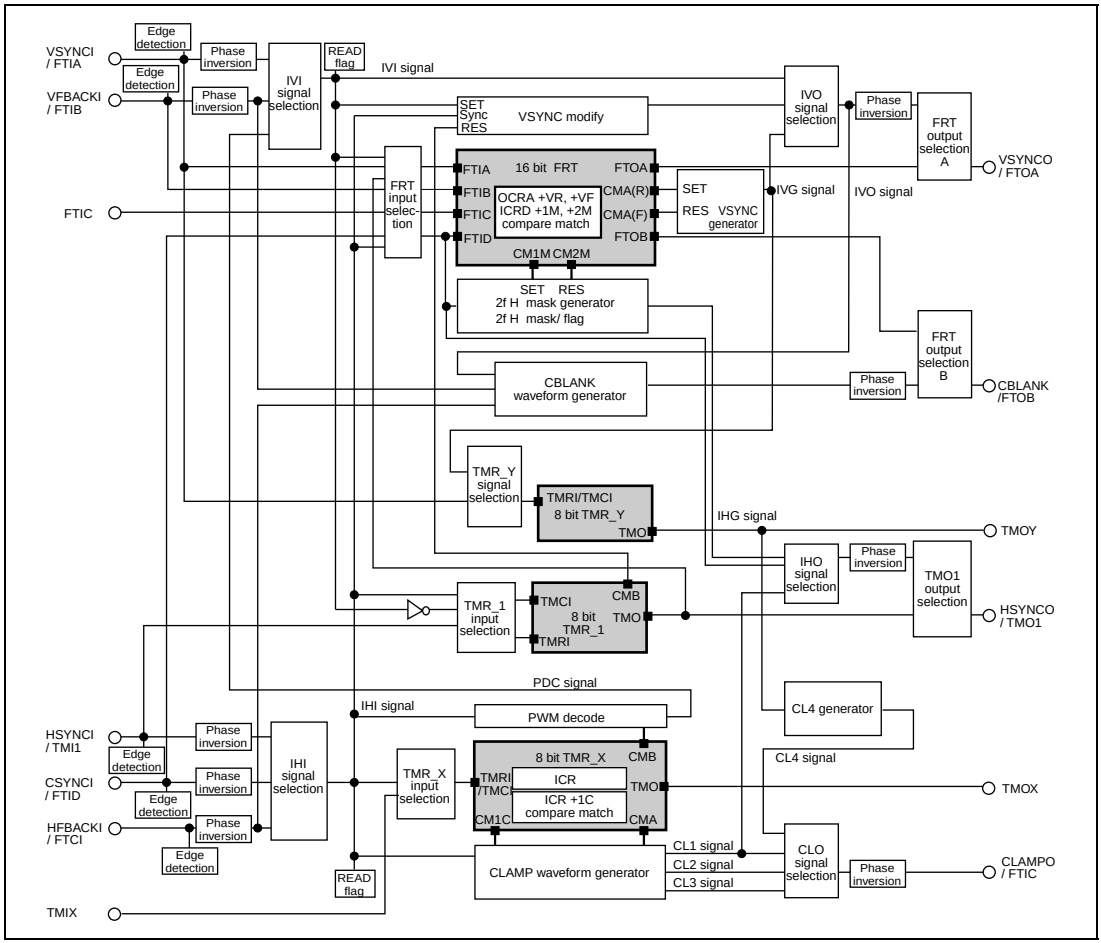


Figure 13.2 Block Diagram of Timer Connection

13.2 Input/Output Pins

Table 13.1 lists the timer connection input and output pins.

Table 13.1 Pin Configuration

Channel	Name	Abbreviation	I/O	Function
0	Vertical synchronization signal input pin	VSYNCI_0	Input	Vertical synchronization signal input pin or FTIA_0 input pin
	Horizontal synchronization signal input pin	HSYNCI_0	Input	Horizontal synchronization signal input pin or TMI1_0 input pin
	Composite synchronization signal input pin	CSYNCI_0	Input	Composite synchronization signal input pin or FTID_0 input pin
	Spare vertical synchronization signal input pin	VFBACKI_0	Input	Spare vertical synchronization signal input pin or FTIB_0 input pin
	Spare horizontal synchronization signal input pin	HFBACKI_0	Input	Spare horizontal synchronization signal input pin or FTCI_0 input pin
	Clamp waveform output pin	CLAMPO	Output	Clamp waveform output pin or FTIC_0 input pin
	Blanking waveform output pin	CBLANK	Output	Blanking waveform output pin or FTOB_0 output pin
1	Vertical synchronization signal input pin	VSYNCI_1	Input	Vertical synchronization signal input pin or FTIA_1 input pin
	Horizontal synchronization signal input pin	HSYNCI_1	Input	Horizontal synchronization signal input pin or TMI1_1 input pin
	Composite synchronization signal input pin	CSYNCI_1	Input	Composite synchronization signal input pin or FTID_1 input pin
Common	Vertical synchronization signal output pin	VSYNCO	Output	Vertical synchronization signal output pin or FTOA_0 output pin
	Horizontal synchronization signal output pin	HSYNCO	Output	Horizontal synchronization signal output pin or TMO1_0 output pin

Note: Channels 0 and 1 are omitted in this manual.

13.3 Register Descriptions

The timer connection has the following registers in each channel.

- Timer connection register I (TCONRI)
- Timer connection register O (TCONRO)
- Timer connection register S (TCONRS)
- Edge sense register (SEDGR)
- Timer extended control register (TECR)

13.3.1 Timer Connection Register I (TCONRI)

TCONRI controls connection between timers, the signal source for synchronization signal input, phase inversion, etc.

Bit	Bit Name	Initial Value	R/W	Description
7	SIMOD1	0	R/W	Input Synchronization Mode Select 1, 0
6	SIMOD0	0	R/W	Select the signal source of the IHI and IVI signals. • Mode 00: No signal 01: S-on-G mode 10: Composite mode 11: Separate mode • IHI Signal 00: HFBACKI input (setting prohibited for channel 1) 01: CSYNCI input 10: HSYNCI input 11: HSYNCI input • IVI Signal 00: VFBACKI input (setting prohibited for channel 1) 01: PDC input 10: PDC input 11: VSYNCI input
5	SCONE	0	R/W	Synchronization Signal Connection Enable Selects the signal source of the FTI input for the FRT, TMI1 input for the TMR1, and TMIX input for the TMRX. For details, see table 13.2.

Bit	Bit Name	Initial Value	R/W	Description
4	ICST	0	R/W	Input Capture Start Bit The TMRX external reset input (TMRIX) is connected to the IHI signal. The TMRX has input capture registers (TICR, TICRR, and TICRF). TICRR and TICRF can measure the width of a short 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 the 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. 0: Input capture function of TICRR and TICRF is halted [Clearing condition] When a rising edge followed by a falling edge is detected on TMRIX 1: Input capture function of TICRR and TICRF is operating (Waiting for the time when a rising edge followed by a falling edge is detected on TMRIX) [Setting condition] When 1 is written to ICST after reading ICST = 0
3	HFINV	0	R/W	Spare Horizontal Synchronization Signal Inversion Selects inversion of the input phase of the spare horizontal synchronization signal (HFBACKI). This bit is reserved in channel 1. The initial value should not be changed. 0: The HFBACKI pin state is used directly as the HFBACKI input 1: The HFBACKI pin state is inverted before use as the HFBACKI input
2	VFINV	0	R/W	Spare Vertical Synchronization Signal Inversion Selects inversion of the input phase of the spare vertical synchronization signal (VFBACKI). This bit is reserved in channel 1. The initial value should not be changed. 0: The VFBACKI pin state is used directly as the VFBACKI input 1: The VFBACKI pin state is inverted before use as the VFBACKI input

Bit	Bit Name	Initial Value	R/W	Description
1	HIINV	0	R/W	Horizontal and Composite Synchronization Signal Inversion Selects inversion of the input phase of the horizontal synchronization signal (HSYNCI) and composite synchronization signal (CSYNCI). 0: The HSYNCI and CSYNCI pin states are used directly as the HSYNCI and CSYNCI inputs 1: The HSYNCI and CSYNCI pin states are inverted before use as the HSYNCI and CSYNCI inputs
0	VIINV	0	R/W	Vertical Synchronization Signal Inversion Selects inversion of the input phase of the vertical synchronization signal (VSYNCI). 0: The VSYNCI pin state is used directly as the VSYNCI input 1: The VSYNCI pin state is inverted before use as the VSYNCI input

Table 13.2 Synchronization Signal Connection Enable

Bit 5		Description							
SCONE	**Mode**	**FTIA**	**FTIB**	**FTIC**	**FTID**	**TMCI1**	**TMRI1**	**TMCIX**	**TMRIX**
0	Normal connection (Initial value)	FTIA input	FTIB input	FTIC input	FTID input	TMI1 input	TMI1 input	TMIX input	TMIX input
1	Synchroniza-tion signal connection mode	IVI signal	TMO1 signal	VFBACKI input*	IHI signal	IHI signal	IVI inverse signal	IHI signal	IHI signal

Note: * Only FTIC input is available in channel 1.

13.3.2 Timer Connection Register O (TCONRO)

TCONRO controls output signal output, phase inversion, etc.

Bit	Bit Name	Initial Value	R/W	Description
7	HOE	0	R/W	Output Enable
6	VOE	0	R/W	Control enabling/disabling of output of the horizontal synchronization signal (HSYNCO), vertical synchronization signal (VSYNCO), and clamp waveform (CLAMPO) and blanking waveform (CBLANK) in channel 0. These bits are reserved in channel 1. The initial value should not be changed. • HOE 0: The PB1/TMO1_0/HSYNCO pin functions as the PB1/TMO1_0 pin 1: The PB1/TMO1_0/HSYNCO pin functions as the HSYNCO pin • VOE 0: The PB0/FTOA_0/VSYNCO pin functions as the PB0/FTOA_0 pin 1: The PB0/FTOA_0/VSYNCO pin functions as the VSYNCO pin • CLOE 0: The PA4/FTIC_0/CLAMPO pin functions as the PA4/FTIC_0 pin 1: The PA4/FTIC_0/CLAMPO pin functions as the CLAMPO pin • CBOE 0: The PA3/FTOB_0/CBLANK pin functions as the PA3/FTOB_0 pin 1: The PA3/FTOB_0/CBLANK pin functions as the CBLANK pin
5	CLOE	0	R/W	
4	CBOE	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
3	HOINV	0	R/W	Horizontal Synchronization Signal Output Inversion Selects the signal output from the HSYNCO with the settings of the HS2, HS1, and HS0 bits in TECR. See table 13.3.
2	VOINV	0	R/W	Vertical Synchronization Signal Output Inversion Selects the signal output from the VSYNCO with the setting of the VS0 bit in TECR. See table 13.4.
1	CLOINV	0	R/W	Output Synchronization Signal Inversion
0	CBOINV	0	R/W	Selects inversion of the output phase of the clamp waveform (CLAMPO) and blanking waveform (CBLANK) in channel 0. These bits are reserved in channel 1. The initial value should not be changed. • CLOINV 0: The CLO signal (CL1, CL2, CL3, or CL4 signal) is used directly as the CLAMPO output 1: The CLO signal (CL1, CL2, CL3, or CL4 signal) is inverted before use as the CLAMPO output • CBOINV 0: The CBLANK signal is used directly as the CBLANK output 1: The CBLANK signal is inverted before use as the CBLANK output

Table 13.3 HSYNCO Output Selection

TECR			TCONRO_1	TCONRO_0	HSYNCO Output Signal
HS2	HS1	HS0	HOINV	HOINV	
0	0	0	—	0	The IHO signal in channel 0 is used directly as the HSYNCO output
			—	1	The IHO signal in channel 0 is inverted before use as the HSYNCO output
		1	0	—	The IHO signal in channel 1 is used directly as the HSYNCO output
			1	—	The IHO signal in channel 1 is inverted before use as the HSYNCO output
	1	0	—	—	The input signal of the HSYNCI_0 is used as the HSYNCO output
		1	—	—	The input signal of the HSYNCI_1 is used as the HSYNCO output
1	0	0	—	—	The input signal of the CSYNCI_0 is used as the HSYNCO output
		1	—	—	The input signal of the CSYNCI_1 is used as the HSYNCO output
	1	0	—	—	Port output*
		1	—	—	Setting prohibited

Note: * The PB1DR value is output regardless of the PB1DDR setting.

Table 13.4 VSYNCO Output Selection

TECR	TCONRI_1	TCONRI_0	VSYNCO Output Signal
VS0	VOINV	VOINV	
0	—	0	The IVO signal in channel 0 is used directly as the VSYNCO output
	—	1	The IVO signal in channel 0 is inverted before use as the VSYNCO output
1	0	—	The IVO signal in channel 1 is used directly as the VSYNCO output
	1	—	The IVO signal in channel 1 is inverted before use as the VSYNCO output

13.3.3 Timer Connection Register S (TCONRS)

TCONRS selects whether to access TMRX or TMRY registers, and the signal source and generation method for the synchronization signal output.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The initial value should not be changed.
6	ISGENE	0	R/W	Internal Synchronization Signal Select Selects internal synchronization signals (IHG, IVG, and CL4 signals) as the signal sources for the IHO, IVO, and CLO signals together with the HOMOD1, HOMOD0, VOMOD1, VOMOD0, CLMOD1, and CLMOD0 bits.
5	HOMOD1	0	R/W	Horizontal Synchronization Output Mode Select 1, 0
4	HOMOD0	0	R/W	Select the signal source and generation method for the IHO signal. • ISGENE = 0 00: The IHI signal (without 2fH modification) is selected 01: The IHI signal (with 2fH modification) is selected 1X: The CL1 signal is selected • ISGENE = 1 XX: The IHG signal is selected
3	VOMOD1	0	R/W	Vertical Synchronization Output Mode Select 1, 0
2	VOMOD0	0	R/W	Select the signal source and generation method for the IVO signal. • ISGENE = 0 00: The IVI signal (without fall modification and IHI synchronization) is selected 01: The IVI signal (without fall modification, with IHI synchronization) is selected 10: The IVI signal (with fall modification, without IHI synchronization) is selected 11: The IVI signal (with fall modification and IHI synchronization) is selected • ISGENE = 1 XX: The IVG signal is selected

Bit	Bit Name	Initial Value	R/W	Description
1	CLMOD1	0	R/W	Clamp Waveform Mode Select 1, 0
0	CLMOD0	0	R/W	Select the signal source for the CLO signal (clamp waveform) in channel 0. These bits are reserved in channel 1. The initial value should not be changed. • ISGENE = 0 00: The CL1 signal is selected 01: The CL2 signal is selected 1X: The CL3 signal is selected • ISGENE = 1 XX: The CL4 signal is selected

[Legend]

X: Don’t care

13.3.4 Edge Sense Register (SEDGR)

SEDGR detects a rising edge on the timer connection input pins and the occurrence of 2fH modification, and determines the phase of the IVI and IHI signals.

Bit	Bit Name	Initial Value	R/W	Description
7	VEDG	0	R/(W)*[1]	VSYNCI Edge Detects a rising edge on the VSYNCI pin. 0: [Clearing condition] When 0 is written to VEDG after reading VEDG = 1 1: [Setting condition] When a rising edge is detected on the VSYNCI pin
6	HEDG	0	R/(W)*[1]	HSYNCI Edge Detects a rising edge on the HSYNCI pin. 0: [Clearing condition] When 0 is written to HEDG after reading HEDG = 1 1: [Setting condition] When a rising edge is detected on the HSYNCI pin
5	CEDG	0	R/(W)*[1]	CSYNCI Edge Detects a rising edge on the CSYNCI pin. 0: [Clearing condition] When 0 is written to CEDG after reading CEDG = 1 1: [Setting condition] When a rising edge is detected on the CSYNCI pin
4	HFEDG	0	R/(W)*[1]	HFBACKI Edge Detects a rising edge on the HFBACKI pin in channel 0. This bit is reserved in channel 1. This bit is always read as 0 and cannot be modified. 0: [Clearing condition] When 0 is written to HFEDG after reading HFEDG = 1 1: [Setting condition] When a rising edge is detected on the HFBACKI pin

Bit	Bit Name	Initial Value	R/W	Description
3	VFEDG	0	R/(W)*[1]	VFBACKI Edge Detects a rising edge on the VFBACKI pin in channel 0. This bit is reserved in channel 1. This bit is always read as 0 and cannot be modified. 0: [Clearing condition] When 0 is written to VFEDG after reading VFEDG = 1 1: [Setting condition] When a rising edge is detected on the VFBACKI pin
2	PREQF	0	R/(W)*[1]	Pre-Equalization Flag Detects the occurrence of a 2fH modification condition for the IHI signal. The generation of a falling/rising edge in the IHI signal during a mask interval is expressed as the occurrence of a 2fH modification condition. For details, see section 13.4.4, 2fH Modification of IHI Signal. 0: [Clearing condition] When 0 is written to PREQF after reading PREQF = 1 1: [Setting condition] When a 2fH modification condition for the IHI signal is detected
1	IHI	—*[2]	R	IHI Signal Level Indicates the current level of the IHI signal. A signal source and phase inversion are selected for the IHI signal depends on the contents of TCONRI. Read this bit to determine whether the input signal is positive or negative, then hold the IHI signal at positive phase by modifying TCONRI. 0: The IHI signal is low 1: The IHI signal is high
0	IVI	—*[2]	R	IVI Signal Level Indicates the current level of the IVI signal. A signal source and phase inversion are selected for the IVI signal depends on the contents of TCONRI. Read this bit to determine whether the input signal is positive or negative, then hold the IVI signal at positive phase by modifying TCONRI. 0: The IVI signal is low 1: The IVI signal is high

Notes: 1. Only 0 can be written, to clear the flag.
2. The initial value is undefined since it depends on the pin state.

13.3.5 Timer Extended Control Register (TECR)

TECR selects the HSYNCO and VSYNCO output signals and the count clock source for the TMR0 and TMR1.

Bit	Bit Name	Initial Value	R/W	Description
7	VS0	0	R/W	Vertical Synchronization Signal Output Selection Selects the signal output from the VSYNCO with the setting of the VOINV bit in TCONRO. See table 13.4.
6	HS2	0	R/W	Horizontal Synchronization Signal Output Select 2 to 0 Select the signal output from the HSYNCO with the settings of the HOINV bit in TCONRO. See table 13.3.
5	HS1	0	R/W	
4	HS0	0	R/W	
3	ICKS1_1	0	R/W	Internal Clock Source Select (Channel 1)
2	ICKS0_1	0	R/W	Select the clock input to the timer counter (TCNT) for the TMR0_1 and TMR1_1 and count condition with the settings of the CKS2 to CKS0 bits in the timer control register 1 (TCR_1). For details, see section 11.3.4, Timer Control Register (TCR).
1	ICKS1_0	0	R/W	Internal Clock Source Select (Channel 0)
0	ICKS0_0	0	R/W	Select the clock input to the timer counter (TCNT) for the TMR0_0 and TMR1_0 and count condition with the settings of the CKS2 to CKS0 bits in the timer control register 0 (TCR_0). For details, see section 11.3.4, Timer Control Register (TCR).

13.4 Operation

13.4.1 PWM Decoding (PDC Signal Generation)

The timer connection and TMRX can be used to decode a PWM signal in which 0 and 1 are represented by the pulse width. To do this, a signal in which a rising edge is generated at regular intervals must be selected as the IHI signal.

The timer counter (TCNT) in the TMRX is set to count the internal clock pulses and to be cleared on the rising edge of the external reset signal (IHI signal). The value to be used as the threshold for deciding the pulse width is written to TCORB. The PWM decoder contains a delay latch which uses the IHI signal as data and compare-match signal B (CMB) as a clock, and the state of the IHI signal (the result of the pulse width decision) at the first compare-match signal B timing after the TCNT is reset by the rise of the IHI signal is output as the PDC signal. Figure 13.3 shows a block diagram for the PWM decoding.

The pulse width setting using TICRR and TICRF of the TMRX can be used to determine the pulse width decision threshold.

Examples of TCR and TCORB settings of the TMRX are shown in tables 13.5 and 13.6, and the PWM decoding timing chart is shown in figure 13.4.

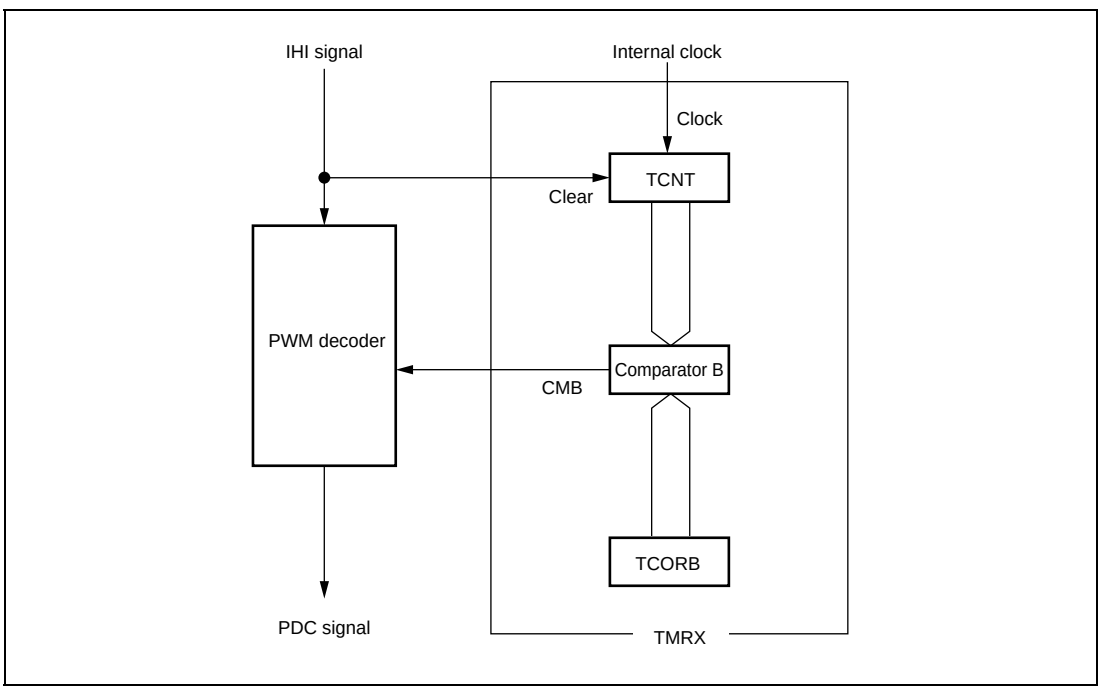


Figure 13.3 Block Diagram for PWM Decoding

Table 13.5 Examples of TCR Settings

Bit	Abbreviation	Contents	Description
7	CMIEB	0	Interrupts due to a compare-match and overflow are disabled
6	CMIEA	0	
5	OVIE	0	
4, 3	CCLR1, CCLR0	11	TCNT is cleared by the rising edge of the external reset signal (IHI signal)
2 to 0	CKS2 to CKS0	001	Internal clock: Incremented on ϕ

Table 13.6 Examples of TCORB (Pulse Width Threshold) Settings

	ϕ: 10 MHz	ϕ: 12 MHz	ϕ: 16 MHz	ϕ: 20 MHz
H'07	0.8 µs	0.67 µs	0.5 µs	0.4 µs
H'0F	1.6 µs	1.33 µs	1 µs	0.8 µs
H'1F	3.2 µs	2.67 µs	2 µs	1.6 µs
H'3F	6.4 µs	5.33 µs	4 µs	3.2 µs
H'7F	12.8 µs	10.67 µs	8 µs	6.4 µs

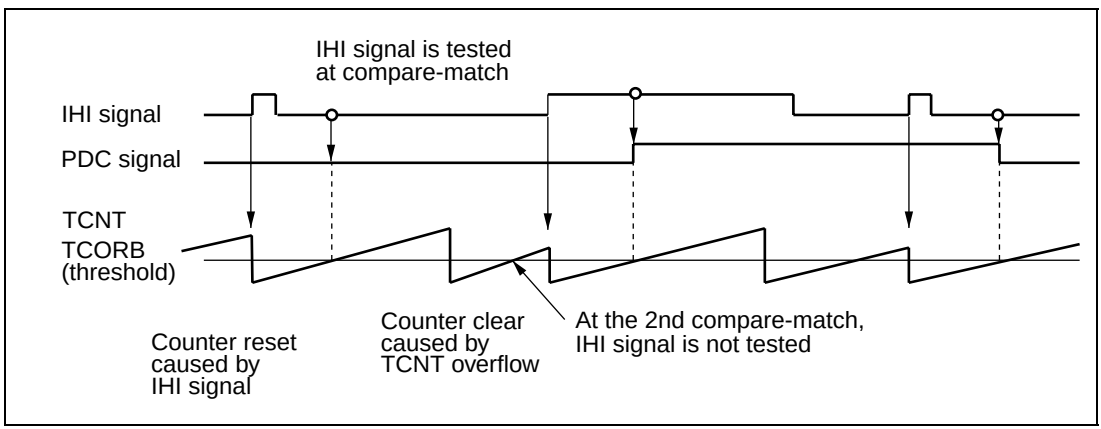


Figure 13.4 Timing Chart for PWM Decoding

13.4.2 Clamp Waveform Generation (CL1/CL2/CL3 Signal Generation)

The timer connection and TMRX can be used to generate signals with different duty cycles and rising/falling edges (clamp waveforms) in synchronization with the input signal (IHI signal). Three clamp waveforms can be generated: the CL1, CL2, and CL3 signals. In addition, the CL4 signal can be generated using the TMRY. Figure 13.5 shows a block diagram for clamp waveform generation.

The CL1 signal rises simultaneously with the rise of the IHI signal, and when the CL1 signal is high, the CL2 signal rises simultaneously with the fall of the IHI signal. The fall of both the CL1 and CL2 signals can be specified by TCORA.

The rise of the CL3 signal can be specified as simultaneous with the sampling of the fall of the IHI signal using the system clock, and the fall of the CL3 signal can be specified by TCORC. The CL3 signal can also fall when the IHI signal rises.

TCNT of the TMRX is set to count internal clock pulses and to be cleared on the rising edge of the external reset signal (IHI signal).

The value to be used as the CL1 signal pulse width is written to TCORA. Write a value of H'02 or more to TCORA when an internal clock ϕ is selected as the TMRX counter clock, and a value or H'01 or more when $\phi/2$ is selected. When an internal clock ϕ is selected, the CL1 signal pulse width is (TCORA set value + 3 ± 0.5). When the CL2 signal is used, the setting must be made so that this pulse width is greater than the IHI signal pulse width.

The value to be used as the CL3 signal pulse width is written to TCORC. TICR of the TMRX captures the value of TCNT at the inverse of the external reset signal edge (in this case, the falling edge of the IHI signal). The timing of the fall of the CL3 signal is determined by the sum of the contents of TICR and TCORC. Caution is required if the rising edge of the IHI signal precedes the fall timing set by the contents of TCORC, since the IHI signal will cause the CL3 signal to fall.

Examples of TCR settings of the TMRX are the same as those in table 13.5. The clamp waveform timing charts are shown in figures 13.6 and 13.7.

Since the rise of the CL1 and CL2 signals is synchronized with the edge of the IHI signal, and their fall is synchronized with the system clock, the pulse width variation is equivalent to the resolution of the system clock.

Both the rise and the fall of the CL3 signal are synchronized with the system clock and the pulse width is fixed, but there is a variation in the phase relationship with the IHI signal equivalent to the resolution of the system clock.

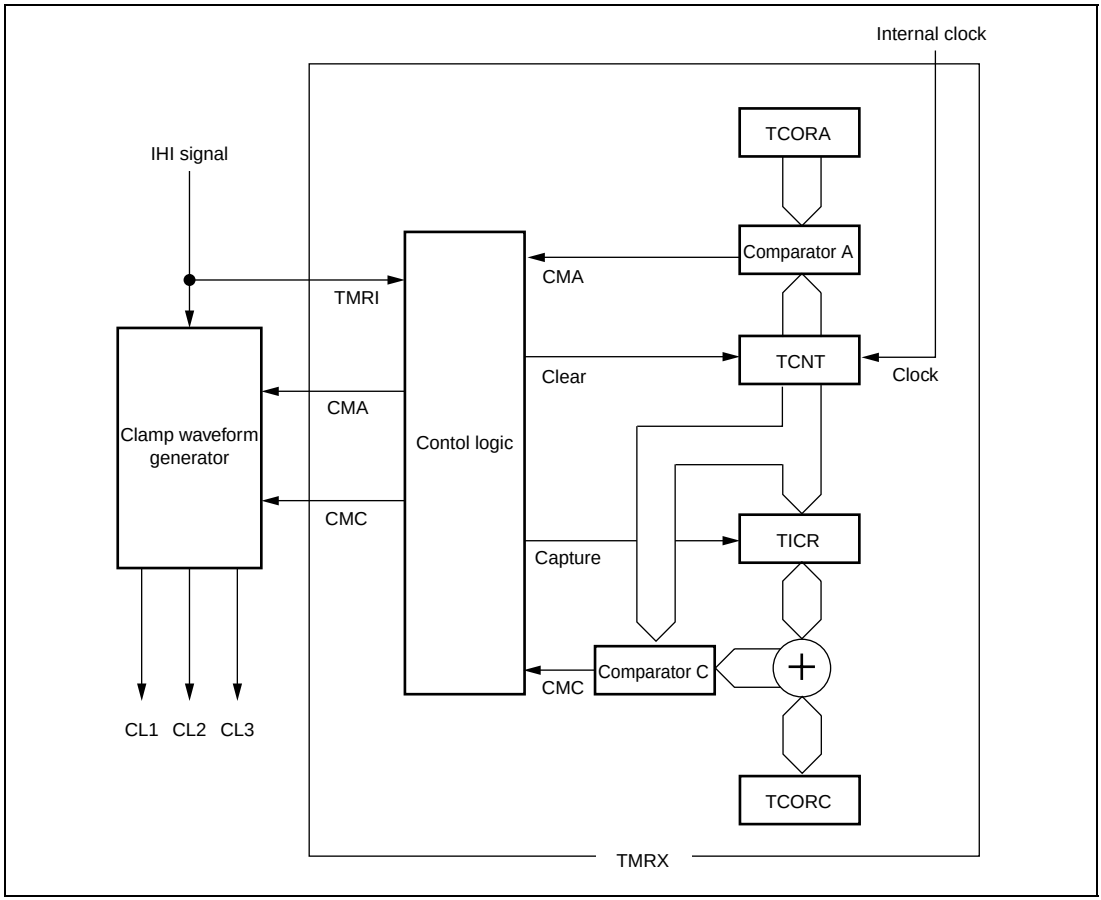


Figure 13.5 Block Diagram for Clamp Waveform Generation

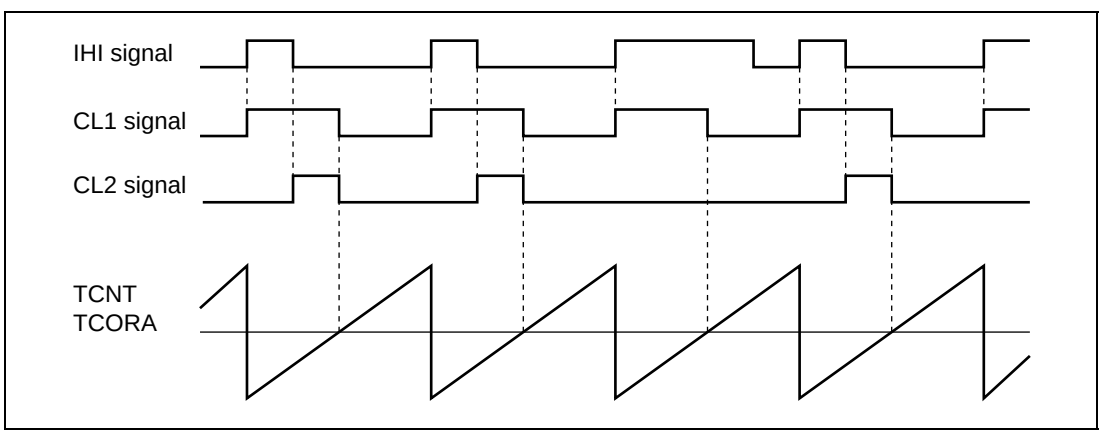


Figure 13.6 Timing Chart for Clamp Waveform Generation (CL1 and CL2 Signals)

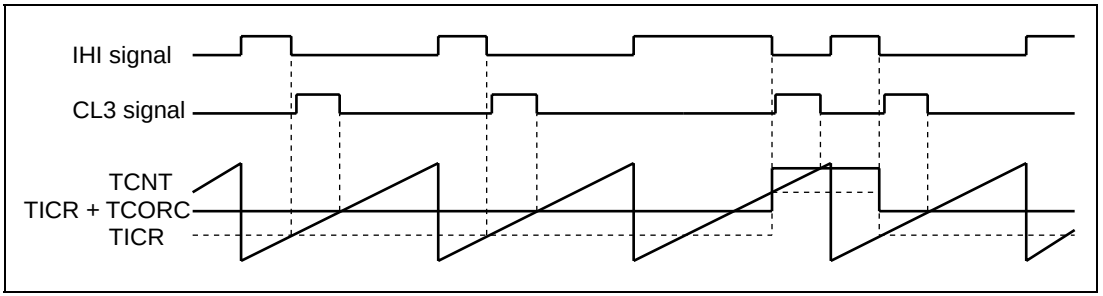


Figure 13.7 Timing Chart for Clamp Waveform Generation (CL3 Signal)

13.4.3 Measurement of 8-Bit Timer Divided Waveform Period

The timer connection, TMR1, and FRT can be used to measure the period of an IHI signal divided waveform. Since the TMR1 can be cleared by a rising edge of the inverted IVI signal, the rise and fall of the IHI signal divided waveform can be synchronized with the IVI signal. This enables period measurement to be carried out efficiently. Figure 13.8 shows a block diagram for the period measurement of the 8-bit timer divided waveform.

To measure the period of an IHI signal divided waveform, TCNT of the TMR1 is set to count the external clock (IHI signal) pulses and to be cleared on the rising edge of the external reset signal (inverse of the IVI signal). The value to be used as the division ratio is written to TCORA, and the TMO output method is specified by the OS bit in TCSR.

Examples of TCR and TCSR settings of the TMR1 and FRT are shown in table 13.7, and the timing chart for measurement of the IVI signal and IHI signal divided waveform periods is shown in figure 13.9. The period of the IHI signal divided waveform is given by (ICRD(3) – ICRD(2)) × resolution.

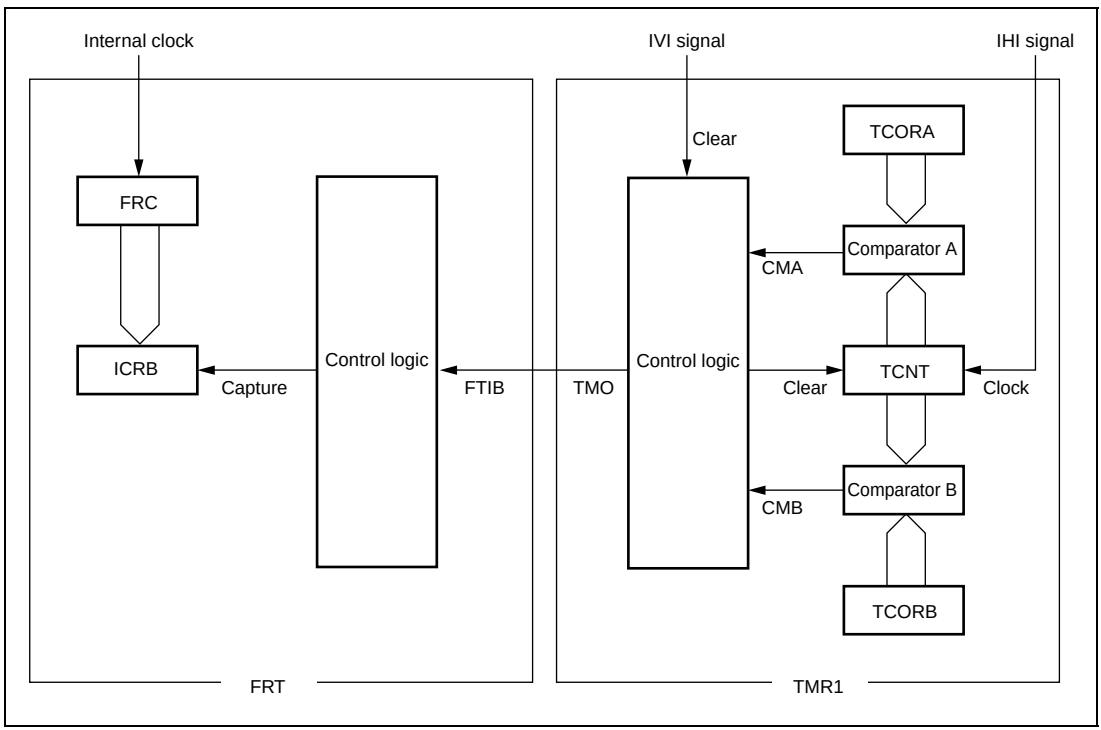


Figure 13.8 Block Diagram for Measurement of 8-Bit Timer Divided Waveform Period

Table 13.7 Examples of TCR and TCSR Settings

Register	Bit	Abbreviation	Contents	Description
TCR of TMR1	7	CMIEB	0	Interrupts due to compare-match and overflow are disabled
	6	CMIEA	0	
	5	OVIE	0	
	4, 3	CCLR1, CCLR0	11	TCNT is cleared by the rising edge of the external reset signal (inverse of the IVI signal)
	2 to 0	CKS2 to CKS0	101	TCNT is incremented on the rising edge of the external clock (IHI signal)
TCSR of TMR1	3 to 0	OS3 to OS0	0011	Not changed by compare-match B; output inverted by compare-match A (toggle output): Division by 512
			1001	When TCORB < TCORA, 1 output on compare-match B, and 0 output on compare-match A: Division by 256
TCR of FRT	6	IEDGB	0/1	0: FRC value is transferred to ICRB on falling edge of input capture input B (IHI divided signal waveform) 1: FRC value is transferred to ICRB on rising edge of input capture input B (IHI divided signal waveform)
	1, 0	CKS1, CKS0	01	FRC is incremented on internal clock: $\phi/8$
TCSR of FRT	0	CCLRA	0	FRC clearing is disabled

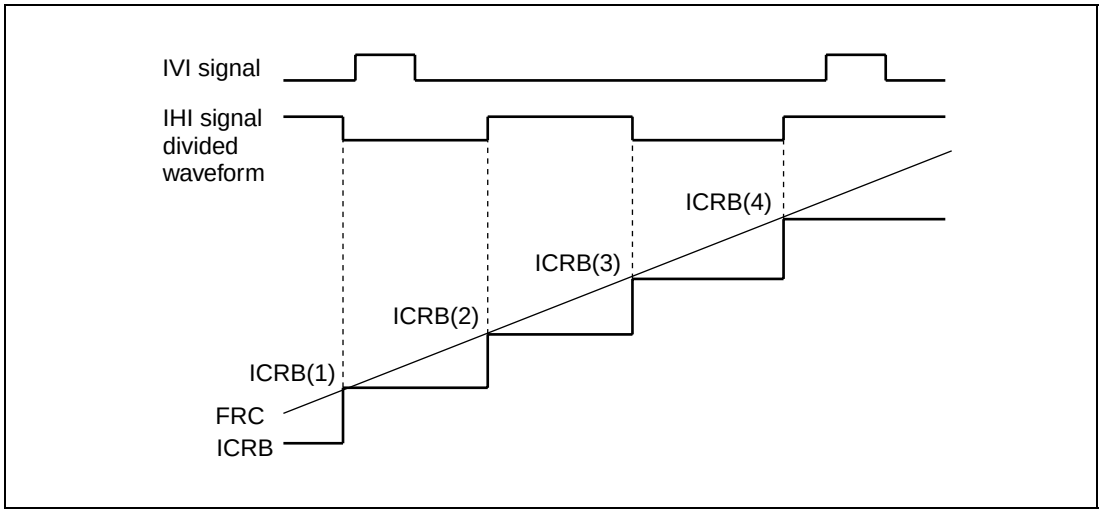


Figure 13.9 Timing Chart for Measurement of IVI Signal and IHI Signal Divided Waveform Periods

13.4.4 2fH Modification of IHI Signal

By using the timer connection and FRT, even if there is a part of the IHI signal with twice the frequency, this can be eliminated. In order for this function to operate properly, the duty cycle of the IHI signal must be approximately 30% or less, or approximately 70% or above.

The 8-bit OCRDM contents or twice the OCRDM contents can be added automatically to the data captured in ICRD of the FRT, and compare-matches generated at these points. The interval between the two compare-matches is called a mask interval. A value equivalent to approximately 1/3 the IHI signal period is written to OCRDM. ICRD is set so that capture is performed on the rise of the IHI signal. Figure 13.10 shows a block diagram for 2fH modification of the IHI signal.

Since the IHI signal supplied to the IHO signal selection circuit is normally set on the rise of the IHI signal and reset on the fall, its waveform is the same as that of the original IHI signal. When 2fH modification is selected, IHI signal edge detection is disabled during mask intervals. Capture is also disabled during these intervals.

Examples of TCR, TCSR, TOCR, and OCRDM settings of the FRT are shown in table 13.8, and the 2fH modification timing chart is shown in figure 13.11.

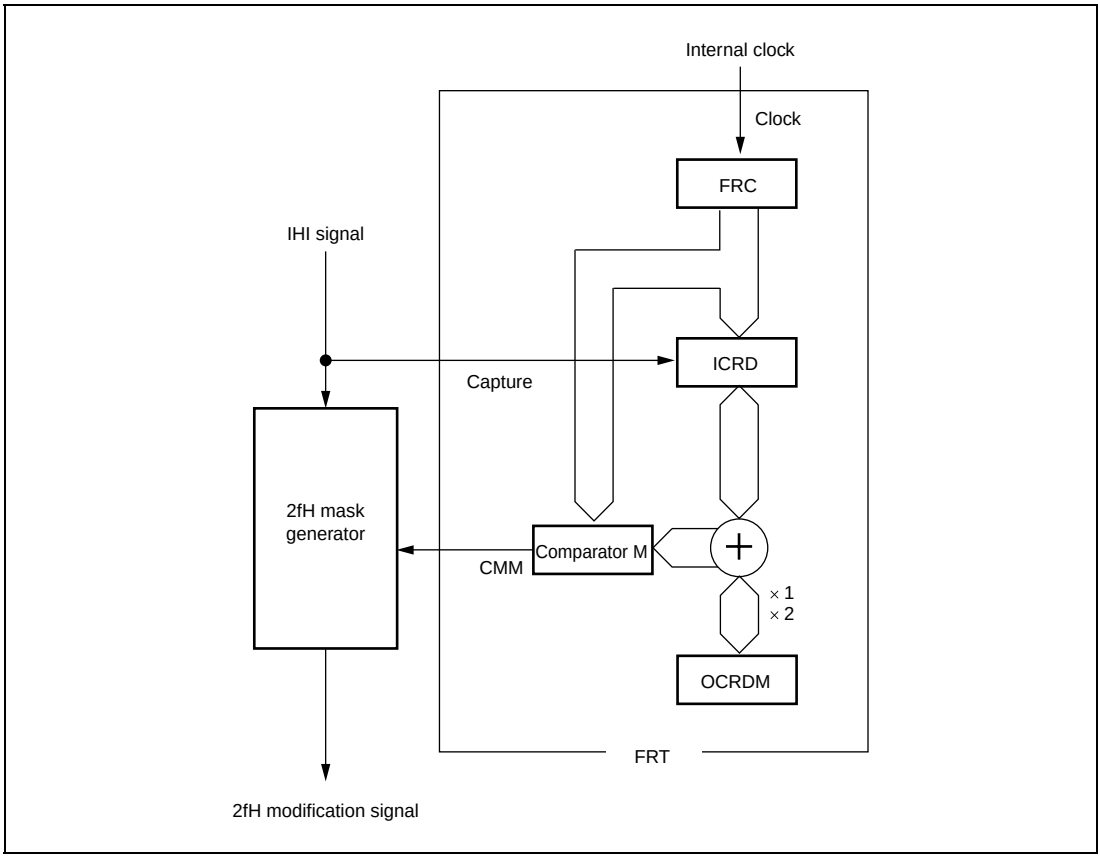


Figure 13.10 Block Diagram for 2fH Modification of IHI Signal

Table 13.8 Examples of TCR, TCSR, TCOR, and OCRDM Settings

Register	Bit	Abbreviation	Contents	Description
TCR of FRT	4	IEDGD	1	FRC value is transferred to ICRD on the rising edge of input capture input D (IHI signal)
	1, 0	CKS1, CKS0	01	FRC is incremented on internal clock: $\phi/8$
TCSR of FRT	0	CCLRA	0	FRC clearing is disabled
TCOR of FRT	7	ICRDMS	1	ICRD is set to the operating mode in which OCRDM is used
OCRDM of FRT	7 to 0	OCRDM7 to OCRDM0	H'01 to H'FF	Specifies the period during which ICRD operation is masked

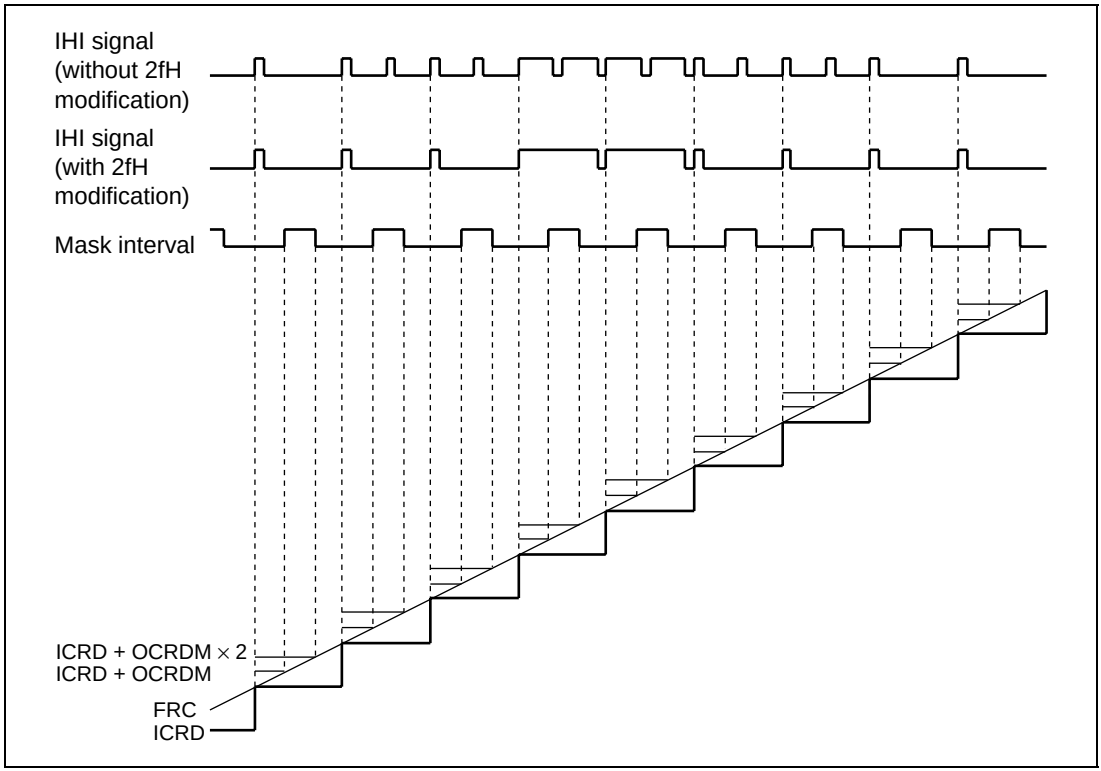


Figure 13.11 2fH Modification Timing Chart

13.4.5 IVI Signal Fall Modification and IHI Synchronization

By using the timer connection and TMR1, the fall of the IVI signal can be shifted backward by the specified number of IHI signal waveforms. Also, the fall of the IVI signal can be synchronized with the rise of the IHI signal. Figure 13.12 shows a block diagram for IVI signal fall modification and IHI signal operation.

To measure of the 8-bit timer divided waveform period, TCNT of the TMR1 is set to count external clock (IHI signal) pulses, and to be cleared on the rising edge of the external reset signal (inverse of the IVI signal). The number of IHI signal pulses until the fall of the IVI signal is written to TOCRB.

Since the IVI signal supplied to the IVO signal selection circuit is normally set on the rise of the IVI signal and reset on the fall, its waveform is the same as that of the original IVI signal. When fall modification is selected, a reset is performed on a TCORB compare-match of the TMR1.

The fall of the waveform generated in this way can be synchronized with the rise of the IHI signal, regardless of whether or not fall modification is selected.

Examples of TCR, TCSR, and TCORB settings of the TMR1 are shown in table 13.9, and the fall modification and IHI synchronization timing chart is shown in figure 13.13.

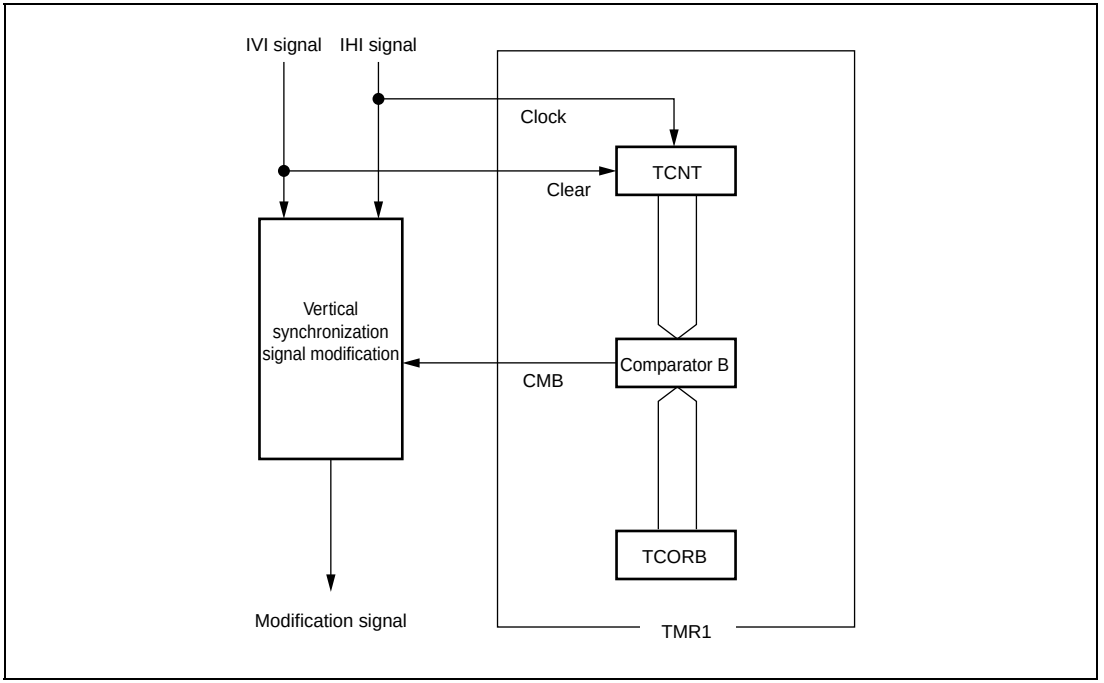


Figure 13.12 Block Diagram for IVI Signal Fall Modification and IHI Signal Operation

Table 13.9 Examples of TCR, TCSR, and TCORB Settings

Register	Bit	Abbreviation	Contents	Description
TCR of TMR1	7	CMIEB	0	Interrupts due to compare-match and overflow are disabled
	6	CMIEA	0	
	5	OVIE	0	
	4, 3	CCLR1, CCLR0	11	TCNT is cleared by the rising edge of the external reset signal (inverse of the IVI signal)
	2 to 0	CKS2 to CKS0	101	TCNT is incremented on the rising edge of the external clock (IHI signal)
TCSR of TMR1	3 to 0	OS3 to OS0	0011	Not changed by compare-match B; output inverted by compare-match A (toggle output)
			1001	When TCORB < TCORA, 1 output on compare-match B, 0 output on compare-match A
TCORB of TMR1	H'03 (example)	Compare-match on the 4th (example) rise of the IHI signal after the rise of the inverse of the IVI signal		

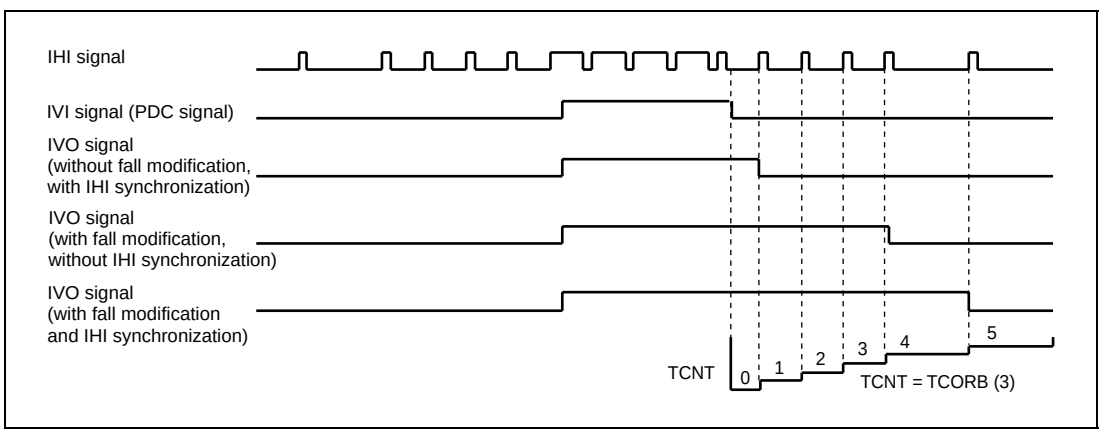


Figure 13.13 Fall Modification and IHI Synchronization Timing Chart

13.4.6 Internal Synchronization Signal Generation (IHG/IVG/CL4 Signal Generation)

By using the timer connection, FRT, and TMRY, it is possible to automatically generate internal signals (IHG and IVG signals) corresponding to the IHI and IVI signals. As the IHG signal is synchronized with the rise of the IVG signal, the IHG signal period must be made a divisor of the IVG signal period in order to keep it constant. In addition, the CL4 signal can be generated in synchronization with the IHG signal. Figure 13.14 shows a block diagram for IHG signal generation and figure 13.15 shows a block diagram for IVG signal generation.

The contents of OCRA of the FRT can be updated by the automatic addition of the contents of OCRAR or OCRAF, alternately, each time a compare-match occurs. A value corresponding to the 0 interval of the IVG signal is written to OCRAR, and a value corresponding to the 1 interval of the IVG signal is written to OCRAF. The IVG signal is set by a compare-match after an OCRAR addition, and reset by a compare-match after an OCRAF addition.

The IHG signal is the TMRY timer output. The TMRY is set to count internal clock pulses, and to be cleared on a TCORA compare-match, to fix the period and set the timer output. TCORB is set so as to reset the timer output. The IVG signal is connected as the TMRY reset input (TMRI), and the rise of the IVG signal can be treated in the same way as a TCORA compare-match.

The CL4 signal is a waveform that rises within one system clock period after the fall of the IHG signal, and has an interval of 1 for 6 system clock periods.

Examples of TCR, TCSR, TCORA, and TCORB settings of the TMRY, and TCR, OCRAR, OCRAF, and TOCR settings of the FRT are shown in table 13.10, and the IHG signal and IVG signal timing chart is shown in figure 13.16.

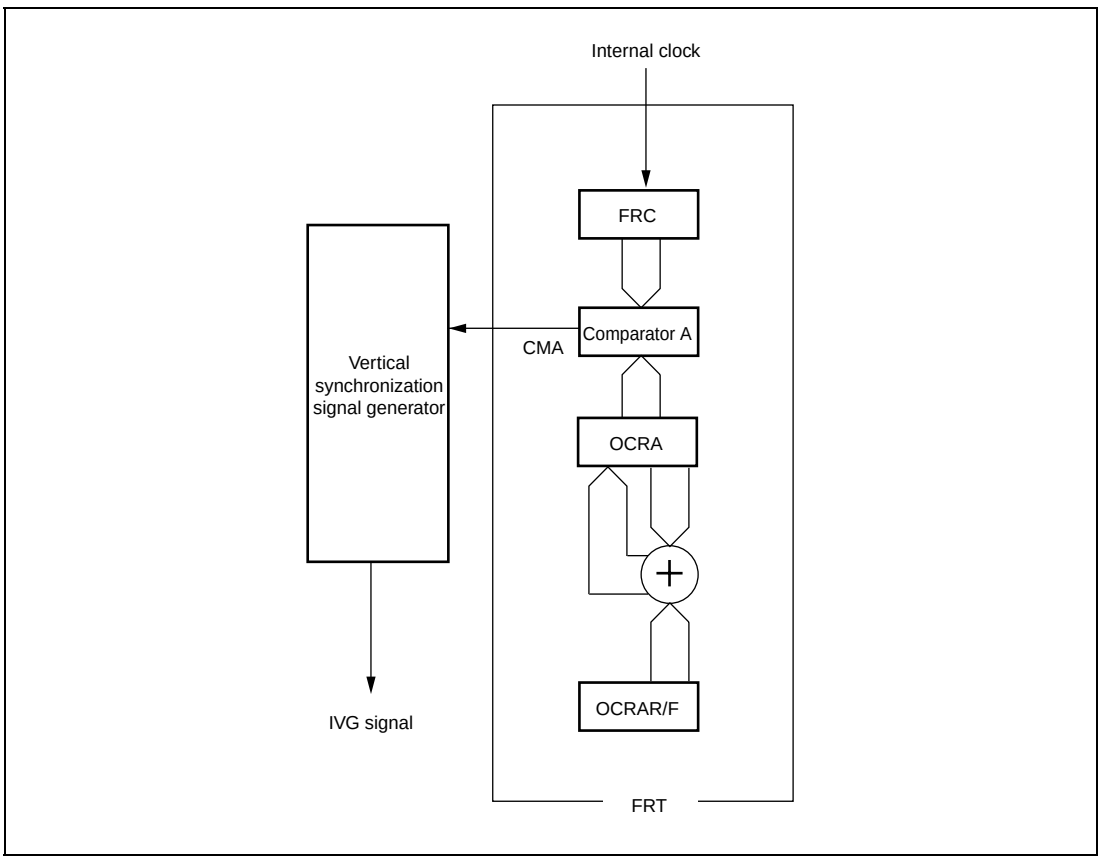


Figure 13.14 Block Diagram for IVG Signal Generation

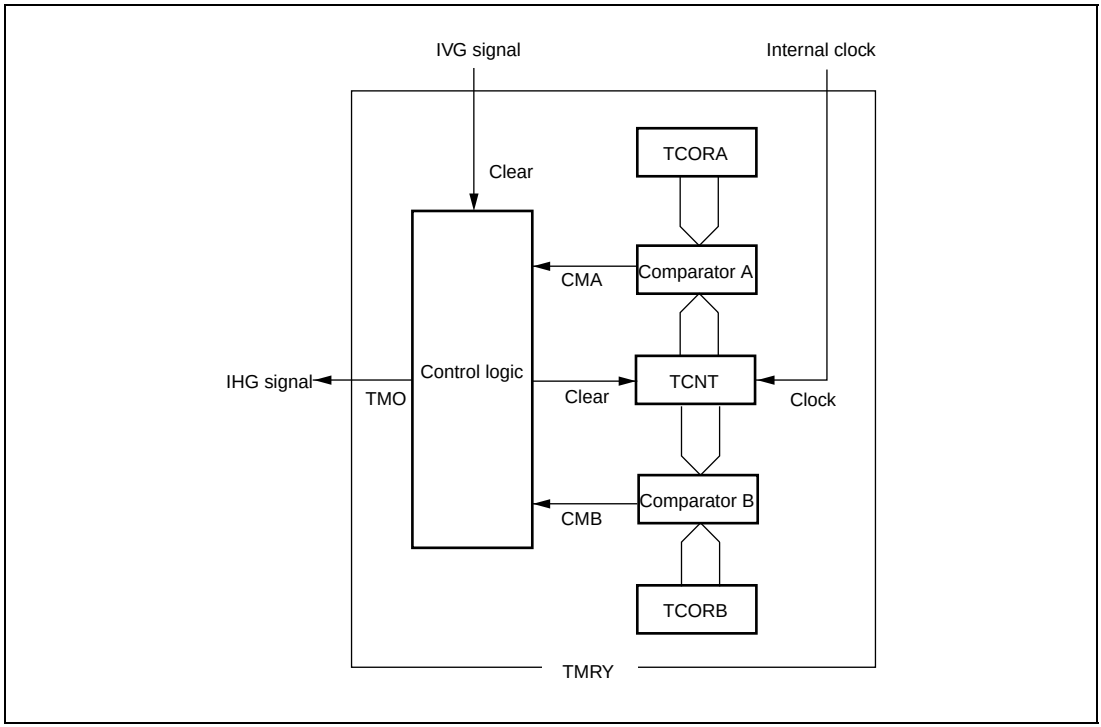


Figure 13.15 Block Diagram for IHG Signal Generation

Table 13.10 Examples of OCRAR, OCRAF, TOCR, TCORA, TCORB, TCR, and TCSR Settings

Register	Bit	Abbreviation	Contents	Description
TCR of TMRY	7	CMIEB	0	Interrupts due to compare-match and overflow are disabled
	6	CMIEA	0	
	5	OVIE	0	
	4, 3	CCLR1, CCLR0	01	TCNT is cleared by compare-match A
	2 to 0	CKS2 to CKS0	001	TCNT is incremented on internal clock: $\phi/4$
TCSR of TMRY	3 to 0	OS3 to OS0	0110	0 output on compare-match B 1 output on compare-match A
TCORA of TMRY	H'3F (example)	IHG signal period = $\phi \times 256$		
TCORB of TMRY	H'03 (example)	1 interval of IHG signal = $\phi \times 16$		
TCR of FRT	1, 0	CKS1, CKS0	01	FRC is incremented on internal clock: $\phi/8$
OCRAR of FRT	H'7FEF (example)	0 interval of IVG signal = $\phi \times 262016$		IVG signal period = $\phi \times 262144$ (1024 times IHG signal)
OCRAF of FRT	H'000F (example)	1 interval of IVG signal = $\phi \times 128$		
TOCR of FRT	6	OCRAMS	1	OCRA is set to the operating mode in which OCRAR and OCRAF are used

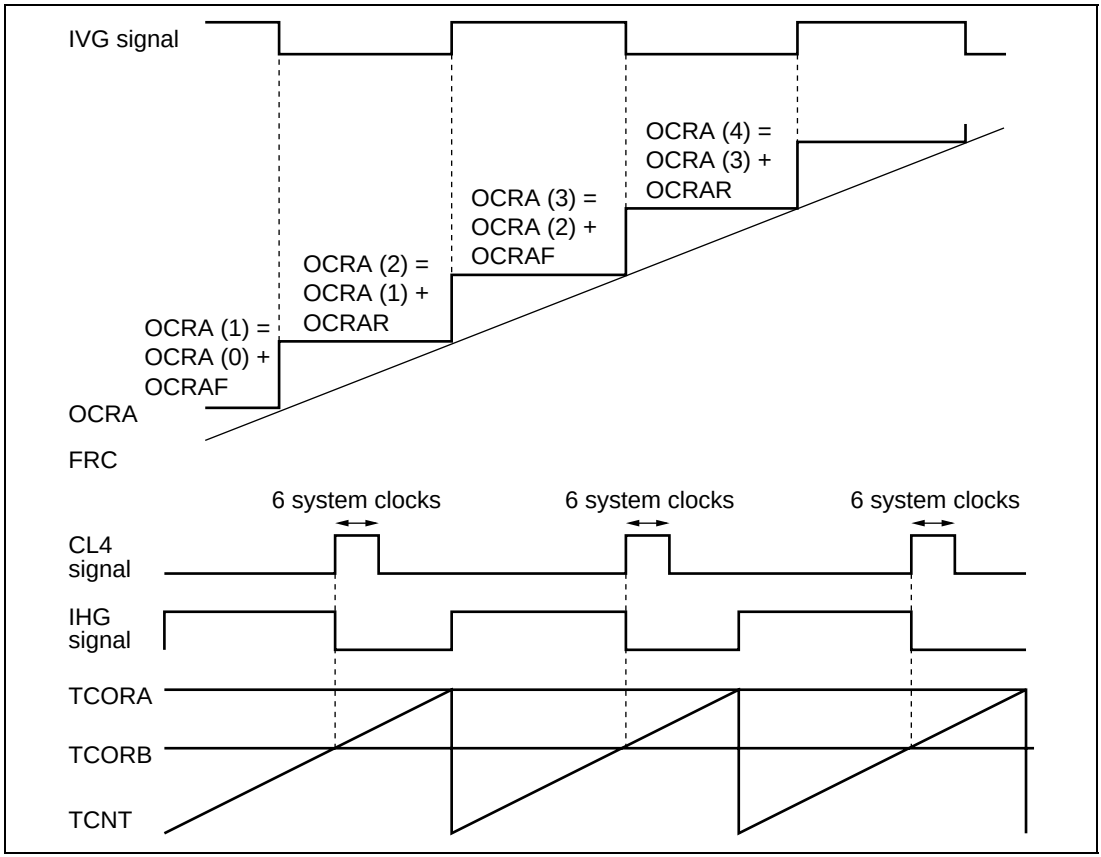


Figure 13.16 IVG Signal/IHG Signal/CL4 Signal Timing Chart

13.4.7 HSYNCO Output

With the HSYNCO output, the meaning of the signal source to be selected and use or non-use of modification varies according to the IHI signal source and the waveform required by an external circuitry. The HSYNCO output modes are shown in table 13.11.

Table 13.11 HSYNCO Output Modes

Mode	IHI Signal	IHO Signal	Meaning of IHO Signal
No signal	HFBACKI input	IHI signal (without 2fH modification)	HFBACKI input is output directly
		IHI signal (with 2fH modification)	Meaningless unless there is a double-frequency part in the HFBACKI input
		CL1 signal	1 interval of HFBACKI input is changed before output
		IHG signal	Internal synchronization signal is output
S-on-G mode	CSYNCI input	IHI signal (without 2fH modification)	CSYNCI input (composite synchronization signal) is output directly
		IHI signal (with 2fH modification)	Double-frequency part of CSYNCI input (composite synchronization signal) is eliminated before output
		CL1 signal	Horizontal synchronization signal part of CSYNCI input (composite synchronization signal) is separated before output
		IHG signal	Internal synchronization signal is output
Composite mode	HSYNCI input	IHI signal (without 2fH modification)	HSYNCI input (composite synchronization signal) is output directly
		IHI signal (with 2fH modification)	Double-frequency part of HSYNCI input (composite synchronization signal) is eliminated before output
		CL1 signal	Horizontal synchronization signal part of HSYNCI input (composite synchronization signal) is separated before output
		IHG signal	Internal synchronization signal is output
Separate mode	HSYNCI input	IHI signal (without 2fH modification)	HSYNCI input (horizontal synchronization signal) is output directly
		IHI signal (with 2fH modification)	Meaningless unless there is a double-frequency part in the HSYNCI input (horizontal synchronization signal)
		CL1 signal	1 interval of HSYNCI input (horizontal synchronization signal) is changed before output
		IHG signal	Internal synchronization signal is output

13.4.8 VSYNCO Output

With the VSYNCO output, the meaning of the signal source to be selected and use or non-use of modification varies according to the IVI signal source and the waveform required by an external circuitry. The VSYNCO output modes are shown in table 13.12.

Table 13.12 VSYNCO Output Modes

Mode	IVI Signal	IVO Signal	Meaning of IVO Signal
No signal	VFBACKI input	IVI signal (without fall modification or IHI synchronization)	VFBACKI input is output directly
		IVI signal (without fall modification, with IHI synchronization)	Meaningless if VFBACKI input is synchronized with HFBACKI input
		IVI signal (with fall modification, without IHI synchronization)	VFBACKI input fall is modified before output
		IVI signal (with fall modification and IHI synchronization)	VFBACKI input fall is modified and the signal is synchronized with HFBACKI input before output
		IVG signal	Internal synchronization signal is output
S-on-G mode or composite mode	PDC signal	IVI signal (without fall modification or IHI synchronization)	Vertical synchronization signal part of CSYNCI/HSYNCI input (composite synchronization signal) is separated before output
		IVI signal (without fall modification, with IHI synchronization)	Vertical synchronization signal part of CSYNCI/HSYNCI input (composite synchronization signal) is separated, and the signal is synchronized with CSYNCI/HSYNCI input before output
		IVI signal (with fall modification, without IHI synchronization)	Vertical synchronization signal part of CSYNCI/HSYNCI input (composite synchronization signal) is separated, and fall is modified before output
		IVI signal (with fall modification and IHI synchronization)	Vertical synchronization signal part of CSYNCI/HSYNCI input (composite synchronization signal) is separated, fall is modified, and the signal is synchronized with CSYNCI/HSYNCI input before output
		IVG signal	Internal synchronization signal is output

Mode	IVI Signal	IVO Signal	Meaning of IVO Signal
Separate mode	VSYNCI input	IVI signal (without fall modification or IHI synchronization)	VSYNCI input (vertical synchronization signal) is output directly
		IVI signal (without fall modification, with IHI synchronization)	Meaningless if VSYNCI input (vertical synchronization signal) is synchronized with HSYNCI input (horizontal synchronization signal)
		IVI signal (with fall modification, without IHI synchronization)	VSYNCI input (vertical synchronization signal) fall is modified before output
		IVI signal (with fall modification and IHI synchronization)	VSYNCI input (vertical synchronization signal) fall is modified and the signal is synchronized with HSYNCI input (horizontal synchronization signal) before output
		IVG signal	Internal synchronization signal is output

13.4.9 CBLANK Output

Using the signals generated/selected with the timer connection, it is possible to generate an waveform based on the composite synchronization signal (blanking waveform). This function is not available in channel 1.

One kind of blanking waveform is generated by combining HFBACKI and VFBACKI inputs, with the phase polarity made positive by means of bits HFINV and VFINV in TCONRI, and with the IVO signal.

The logic of CBLANK output waveform generation is shown in figure 13.17.

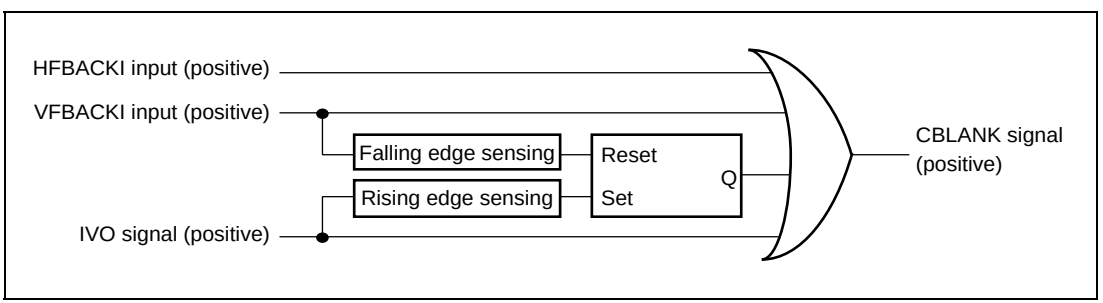


Figure 13.17 CBLANK Output Waveform Generation

Section 14 Duty Measurement Circuit

This LSI has an on-chip duty measurement circuit which consists of an edge detection circuit, 8-bit counter, and capture register. This circuit can measure the duty by detecting edges of an external event signal and capturing the high-level period and cycle.

Figure 14.1 shows a block diagram of the duty measurement circuit.

14.1 Features

- Selection of a counter operating signal from eight operating clocks

  One of the eight operating clocks (ϕ, ϕ/2, ϕ/4, ϕ/8, ϕ/32, ϕ/2048, ϕ/32768, or ϕ/65536) can be selected.
- Automatic duty measurement of eight external event signals for two systems

  Using an edge detection circuit enables both edges of the external event signal to be detected and capturing the counter value enables the duty to be measured. One of eight external event signals (2-system HSYNC, 2-system CSYNC, 2-system VSYNC, HFBACK, or VFBACK) can be selected.
- Two interrupt sources

  There are two interrupt sources: Duty measurement end and overflow.

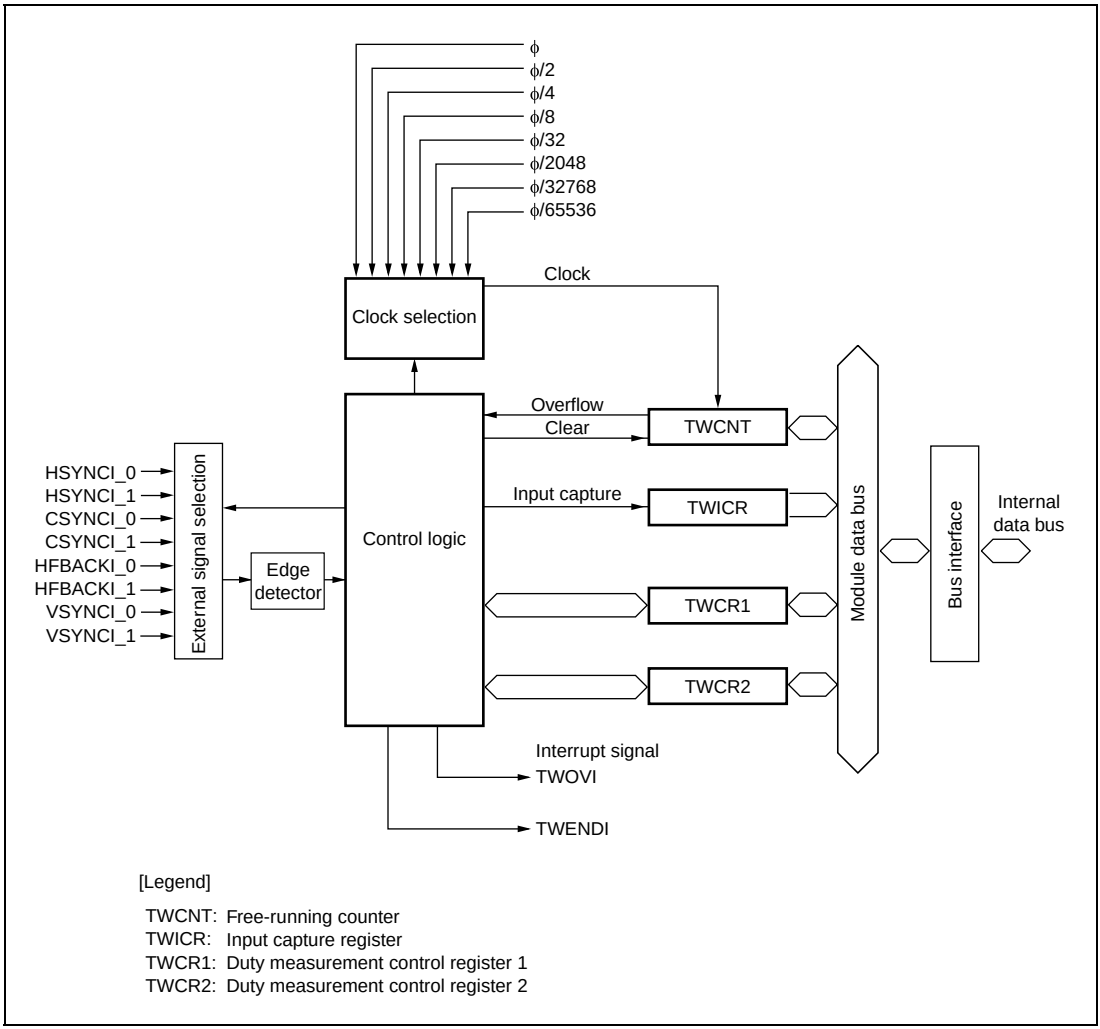


Figure 14.1 Block Diagram of Duty Measurement Circuit

14.2 Input/Output Pins

Table 14.1 lists the input pins for the duty measurement circuit.

Table 14.1 Pin Configuration

Name	Symbol	I/O	Function
Horizontal synchronization signal 0 input pin	HSYNCI_0	Input	Horizontal synchronization signal 0 input pin
Horizontal synchronization signal 1 input pin	HSYNCI_1	Input	Horizontal synchronization signal 1 input pin
Composite synchronization signal 0 input pin	CSYNCI_0	Input	Composite synchronization signal 0 input pin
Composite synchronization signal 1 input pin	CSYNCI_1	Input	Composite synchronization signal 1 input pin
Spare horizontal synchronization signal input pin	HFBACKI	Input	Spare horizontal synchronization signal input pin
Spare vertical synchronization signal input pin	VFBACKI	Input	Spare vertical synchronization signal input pin
Vertical synchronization signal 0 input pin	VSYNCI_0	Input	Vertical synchronization signal 0 input pin
Vertical synchronization signal 1 input pin	VSYNCI_1	Input	Vertical synchronization signal 1 input pin

14.3 Register Descriptions

The duty measurement circuit has the following registers.

- Free-running counter (TWCNT)
- Input capture register (TWICR)
- Duty measurement control register 1 (TWCR1)
- Duty measurement control register 2 (TWCR2)

14.3.1 Free-Running Counter (TWCNT)

TWCNT is an 8-bit readable/writable up-counter. The free-running count operation is performed by setting the FRC bit in TWCR1 to 1. When the START bit in TWCR2 is set to 1, TWCNT is cleared and duty measurement is started. For details on duty measurement operation, see section 14.4, Operation. The clock source is selected by bits CKS2 to CKS0 in TWCR1. When TWCNT overflows from H'FF to H'00, the OVF bit in TWCR2 is set to 1. TWCNT is initialized to H'00.

14.3.2 Input Capture Register (TWICR)

TWICR is an 8-bit read-only register. When the falling edge of the external event signal is detected during duty measurement, the current TWCNT value is transferred. For details on duty measurement operation, see section 14.4, Operation. TWICR is initialized to H'00.

14.3.3 Duty Measurement Control Register 1 (TWCR1)

TWCR1 controls the free-running counter (TWCNT) and selects the TWCNT input clock and an external event signal whose duty is to be measured.

Bit	Bit Name	Initial Value	R/W	Description
7	FRC	0	R/W	Free-Running Counter When 1 is written to this bit, TWCNT operates as a free-running counter regardless of the state of the START bit. 0: Duty measurement operation 1: Free-running counter operation
6	—	0	R/W	Reserved
5	CKS2	0	R/W	Clock Select 2 to 0 Select the clock input to TWCNT. 000: Count on internal clock φ 001: Count on internal clock φ/2 010: Count on internal clock φ/4 011: Count on internal clock φ/8 100: Count on internal clock φ/32 101: Count on internal clock φ/2048 110: Count on internal clock φ/32768 111: Count on internal clock φ/65536
4	CKS1	0	R/W	
3	CKS0	0	R/W	
2	IS2	0	R/W	Input Select 2 to 0 Select the external event signal whose duty is to be measured. 000: Measure the duty of HSYNCI_0 001: Measure the duty of HSYNCI_1 010: Measure the duty of CSYNCI_0 011: Measure the duty of CSYNCI_1 100: Measure the duty of HFBACKI 101: Measure the duty of VFBACKI 110: Measure the duty of VSYNCI_0 111: Measure the duty of VSYNCI_1
1	IS1	0	R/W	
0	IS0	0	R/W	

14.3.4 Duty Measurement Control Register 2 (TWCR2)

TWCR2 controls enabling or disabling interrupt request signals, status flag indication, and duty measurement operation.

Bit	Bit Name	Initial Value	R/W	Description
7	ENDIE	0	R/W	Duty Measurement End Interrupt Enable When the ENDF flag in TWCR2 is set to 1, this bit enables or disables an interrupt request by the ENDF flag. 0: Disables an interrupt request (TWENDI) by ENDF 1: Enables an interrupt request (TWENDI) by ENDF
6	OVIE	0	R/W	Overflow Interrupt Enable When the OVF flag in TWCR2 is set to 1, this bit enables or disables an interrupt request by the OVF flag. 0: Disables an interrupt request (TWOVI) by OVF 1: Enables an interrupt request (TWOVI) by OVF
5	ENDF	0	R/(W)*	Duty Measurement End A status flag indicating that duty measurement has ended. [Setting condition] When duty measurement ends [Clearing condition] When 0 is written to ENDF after reading ENDF = 1
4	OVF	0	R/(W)*	Overflow Flag A flag indicating that a TWCNT overflow has occurred. [Setting condition] When TWCNT overflows from H′FF to H′00 [Clearing condition] When 0 is written to OVF after reading OVF = 1
3 to 1	—	All 0	R/W	Reserved The initial value should not be changed.

Bit	Bit Name	Initial Value	R/W	Description
0	START	0	R/W	Start When the FRC bit in TWCR1 is 0 and 1 is written to this bit, duty measurement will start. If this bit is read during duty measurement, 1 is read from this bit. If duty measurement ends, this bit is automatically cleared to 0. If 0 is written during duty measurement, duty measurement forcibly ends. Even if 1 is written when the FRC bit in TWCR1 is 1, it is ignored.

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

14.4 Operation

14.4.1 Duty Measurement for External Event Signal

Figure 14.2 shows an example of duty measurement for the external event signal.

1. Select the external event signal whose duty needs to be measured by the IS2 to IS0 bits in TWCR1.
2. Select the count clock source for TWCNT by the CK2 to CK0 bits in TWCR1.
3. Write 1 to the START bit in TWCR2. At this time, TWCNT is cleared to H′00.
4. When a rising edge of the external event signal is detected, TWCNT starts counting. Then, if a falling edge of the external event signal is detected, the TWCNT value is transferred to TWICR. If the second rising edge of the external event signal is detected, TWCNT stops counting. At this time, the ENDF flag in TWCR2 is set to 1.

When TWICR and TWCNT values are read and then compared, duty measurement for the external event signal is accomplished.

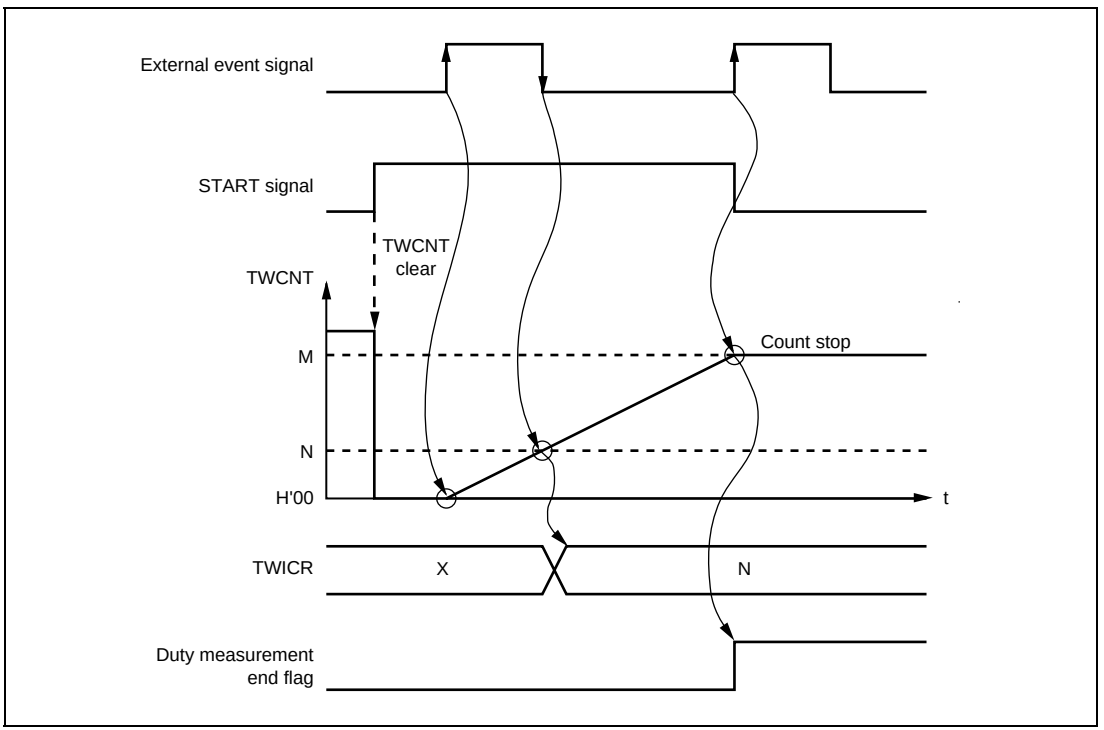


Figure 14.2 Example of Duty Measurement for External Event Signal

14.5 Operation Timing

14.5.1 TWCNT Count Timing

Figure 14.3 shows the TWCNT count timing.

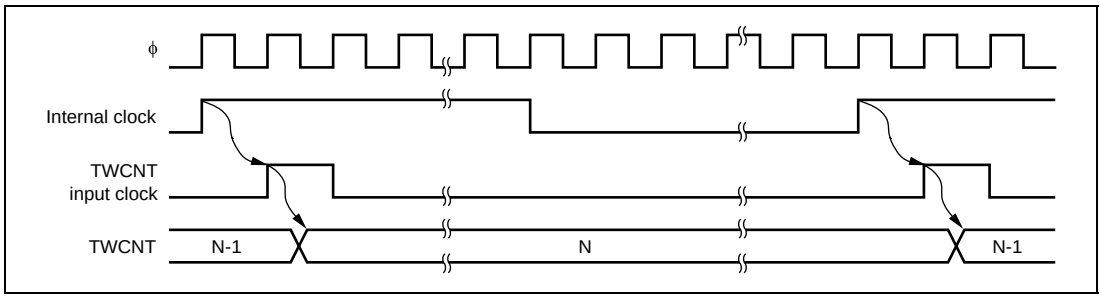


Figure 14.3 TWCNT Count Timing

14.5.2 TWCNT Clear Timing by Setting START Bit

Setting the START bit in TWCR2 to 1 starts duty measurement and then clears TWCNT. Figure 14.4 shows the TWCNT clear timing.

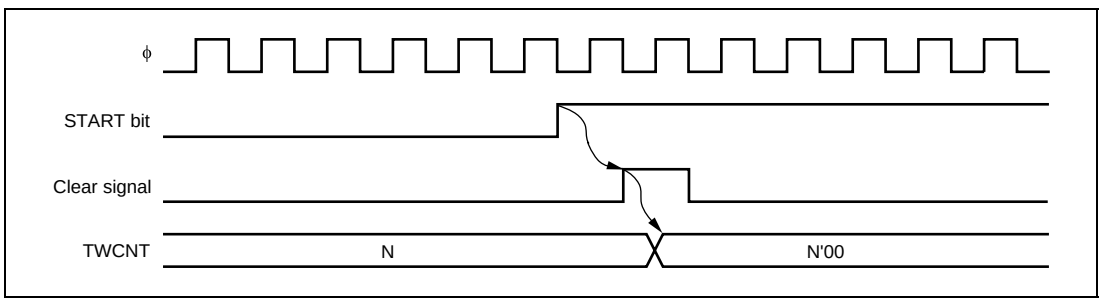


Figure 14.4 TWCNT Clear Timing by Setting START Bit

14.5.3 Count Start Timing for Duty Measurement

If a rising edge of the external event input is detected after the START bit in TWCR2 has been set to 1, a count enable signal is set. At this time, if the TWCNT input clock exists, TWCNT starts incrementing. Figure 14.5 shows the count start timing for duty measurement.

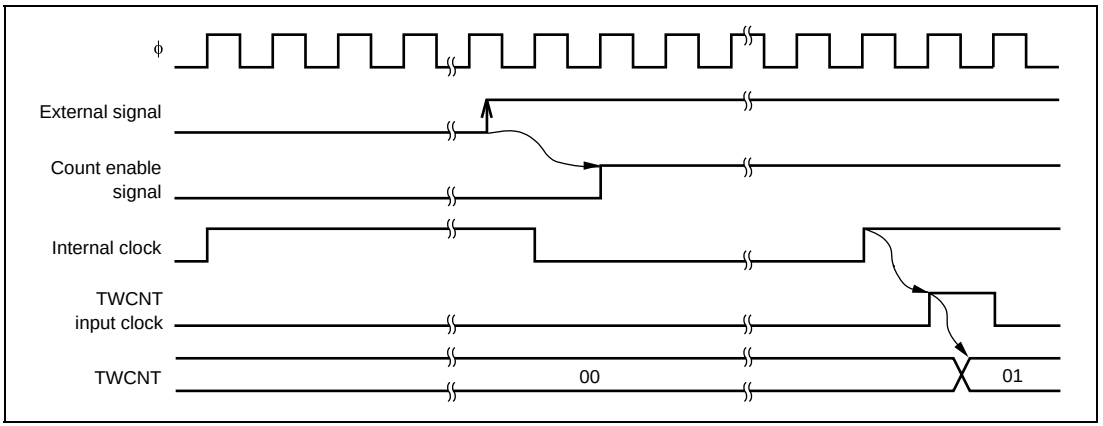


Figure 14.5 Count Start Timing for Duty Measurement

14.5.4 Capture Timing during Duty Measurement

When a falling edge of the external event signal is detected during duty measurement, the TWCNT value is transferred to TWICR. Figure 14.6 shows the capture timing during duty measurement.

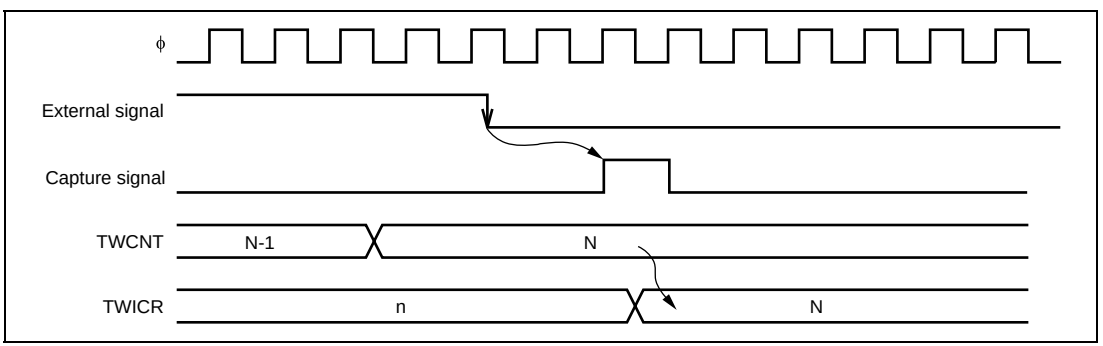


Figure 14.6 Input Capture Timing during Duty Measurement

14.5.5 Clear Timing for START Bit when Duty Measurement Ends

When duty measurement ends, the START bit in TWCR2 is cleared to 0. Figure 14.7 shows the clear timing for the START bit when duty measurement ends.

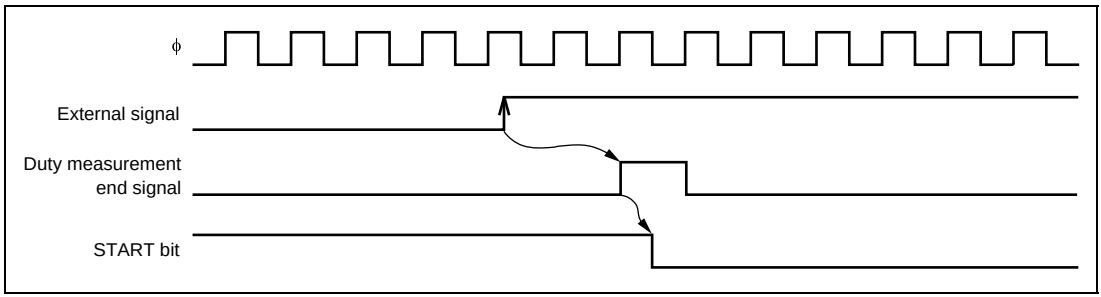


Figure 14.7 Clear Timing for START Bit when Duty Measurement Ends

14.5.6 Set Timing for Duty Measurement End Flag (ENDF)

When duty measurement ends, the duty measurement end flag (ENDF) in TWCR2 is set to 1. Figure 14.8 shows the ENDF set timing.

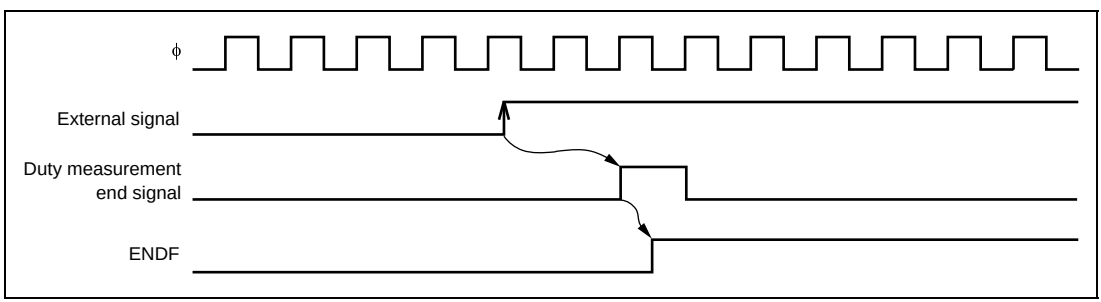


Figure 14.8 Set Timing for Duty Measurement End Flag (ENDF)

14.5.7 Set Timing for Overflow Flag (OVF)

The overflow flag (OVF) in TWCR2 is set to 1 by the overflow signal which is output when TCNT overflows from H′FF to H′00. Figure 14.9 shows the OVF set timing.

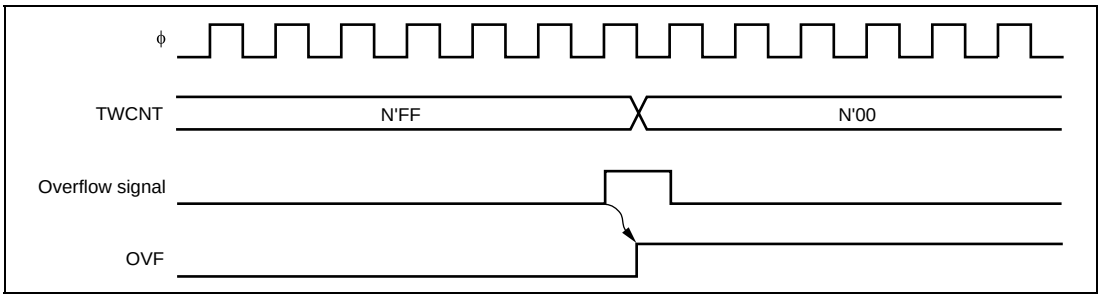


Figure 14.9 Set Timing for OVF Flag

14.6 Interrupt Sources

The duty measurement circuit can request two interrupts: TWOVI and TWENDI. Table 14.2 lists the sources and priorities of these interrupts. Each interrupt can be enabled or disabled by an interrupt enable bit in TCR or TCSR. Independent signals are sent to the interrupt controller for each interrupt.

Table 14.2 Interrupt Sources for Duty Measurement Circuit

Interrupt	Interrupt Source	Interrupt Flag	Priority
TWENDI	Duty measurement end	ENDF	High
TWOVI	TWCNT overflow	OVF	Low

14.7 Usage Notes

14.7.1 Conflict between TWCNT Write and Increment

If a TWCNT increment pulse is generated during the T_2 state of a TWCNT write cycle as shown in figure 14.10, the write takes priority and TWCNT is not incremented.

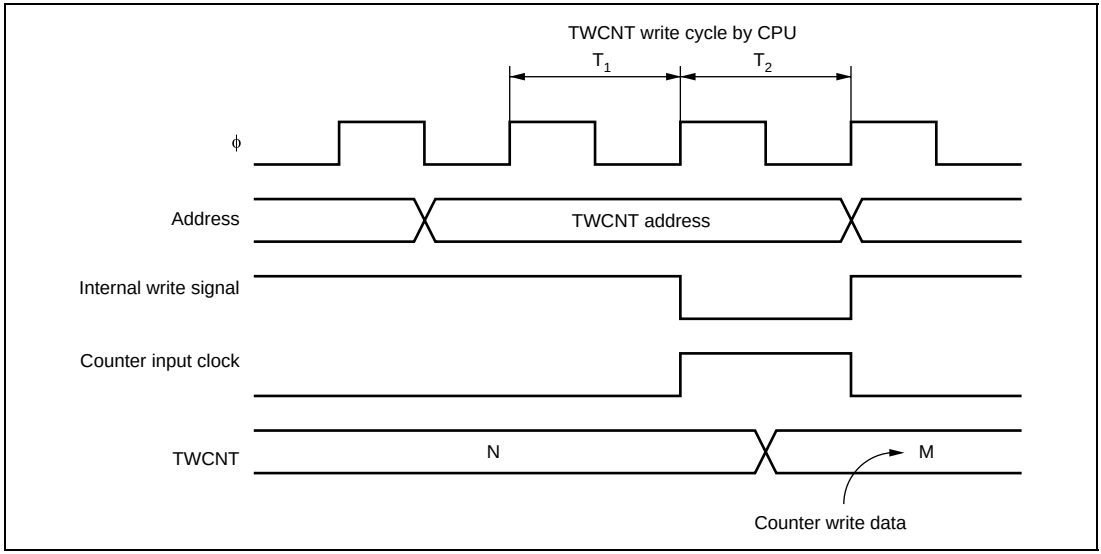


Figure 14.10 TWCNT Write-Increment Conflict

14.7.2 Write to START Bit during Free-Running Counter Operation

If 1 is written to the START bit in TWCR2 while the FRC bit in TWCR1 is 1 as shown in figure 14.11, duty measurement is ignored and the START bit is cleared to 0.

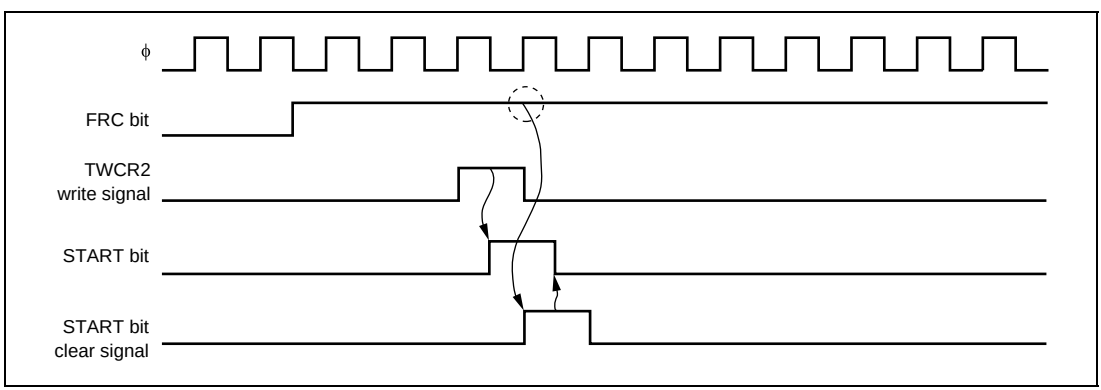


Figure 14.11 Write to START Bit during Free-Running Counter Operation

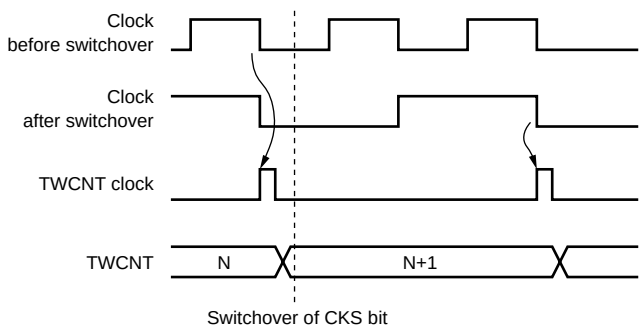
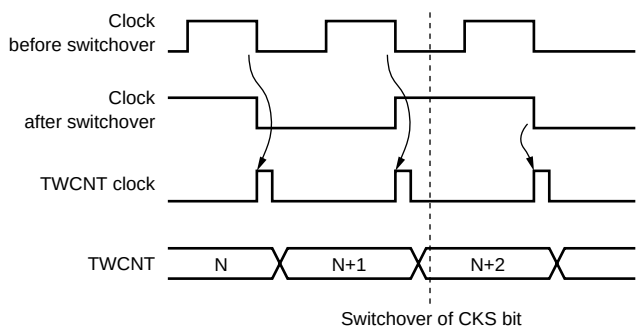
14.7.3 Switching of Internal Clock and TWCNT Operation

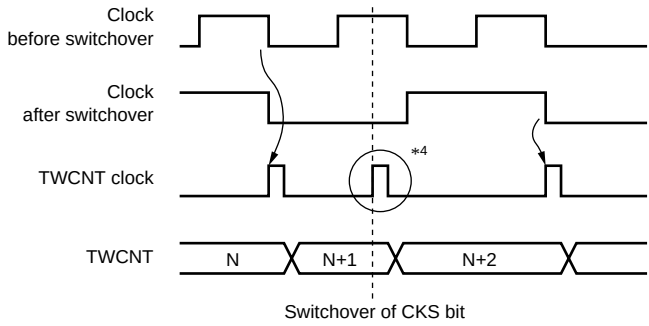
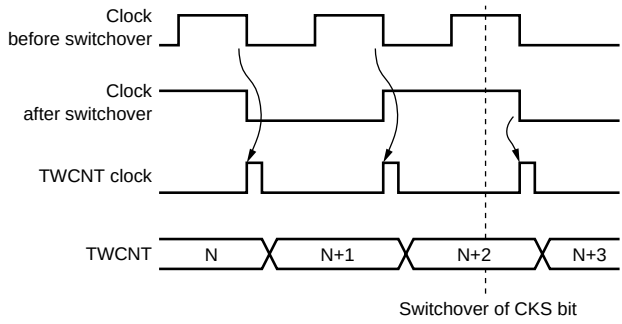
The changeover may cause TWCNT to be incremented depending on the timing at which the internal clock is switched (bits CKS2 to CKS0 are rewritten). Table 14.3 shows the relationship between the timing and TCNT operation.

The TWCNT clock is generated on detection of the falling edge of the internal clock. If the clock is changed when the old source is high and the new source is low, as in case no. 3 in table 14.3, the changeover is regarded as a falling edge that generates the TWCNT clock, and TWCNT is incremented.

Switching between an internal clock and external clock may also cause TWCNT to be incremented. To prevent incorrect operation, ensure that TWCNT should be stopped before an internal clock is changed.

Table 14.3 Switching of Internal Clock and TWCNT Operation

No.	Timing of Switchover by Means of CKS2 to CKS0 Bits	TWCNT Operation
1	Switching from low to low*[1]	
2	Switching from low to high*[2]	

No.	Timing of Switchover by Means of CKS2 to CKS0 Bits	TWCNT Operation
3	Switching from high to low*3	
4	Switching from high to high	

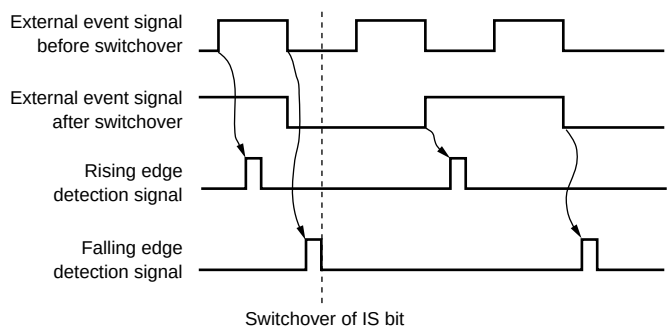
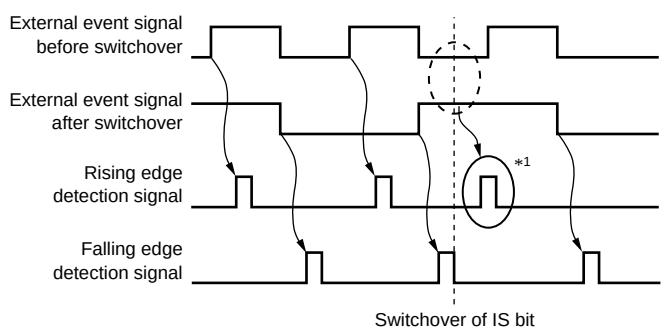
Notes: 1. Including switching from low to stop and from stop to low
2. Including switching from stop to high
3. Including switching from high to stop
4. Generated on the assumption that the switchover is a falling edge; TWCNT is incremented.

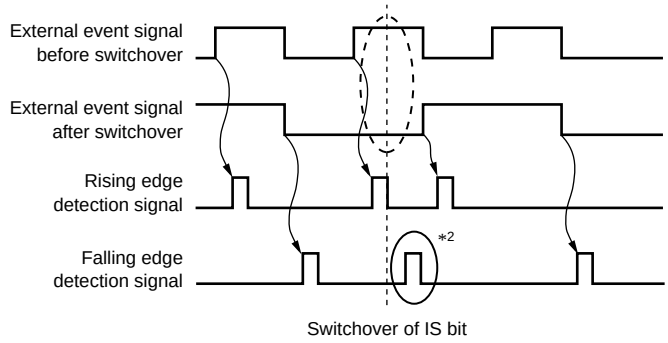
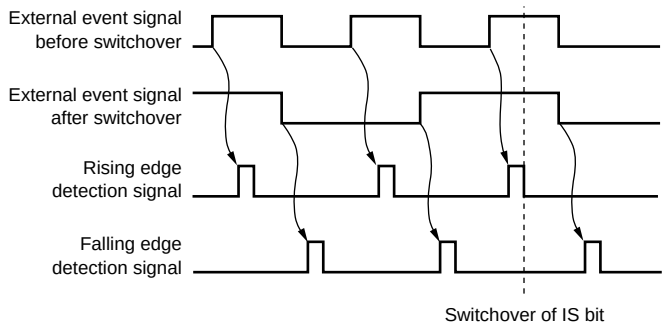
14.7.4 Switching of External Event Signal and Operation of Edge Detection Circuit

When the external event signal is changed, the edge detection circuit may regard it as a rising or falling edge. Table 14.4 shows the relationship between the timing at which the external event signal is switched (bits IS2 to IS0 are rewritten) and operation of the edge detection circuit.

Rising and falling edges of the external event signal are detected by the edge detection circuit. Thus, if the clock is changed when the old source is low and the new source is high, as in case no. 2 in table 14.4, the changeover is regarded as a rising edge and the TWCNT value is transferred to TWICR during duty measurement. If the clock is changed when the old source is high and the new source is low, as in case no. 3 in table 14.4, the changeover is regarded as a falling edge and duty measurement is started or ended. To prevent incorrect operation, ensure that the IS bit should not be rewritten during duty measurement.

Table 14.4 Switching of External Event Signal and Operation of Edge Detection Circuit

No.	Timing of Switchover by Means of IS2 to IS0 Bits	Operation of Edge Detection Circuit
1	Switching from low to low	
2	Switching from low to high	

No.	Timing of Switchover by Means of IS2 to IS0 Bits	Operation of Edge Detection Circuit
3	Switching from high to low	
4	Switching from high to high	

Notes: 1. The switchover timing is detected as a rising edge.
2. The switchover timing is detected as a falling edge.

Section 15 Watchdog Timer (WDT)

This LSI incorporates the watchdog timer (WDT). The WDT 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 timer function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is generated each time the counter overflows.

15.1 Features

- Selectable from eight 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.

Interval Timer Mode:

- If the counter overflows, an interval timer interrupt (WOVI) is generated.

A block diagram of the WDT is shown in figure 15.1.

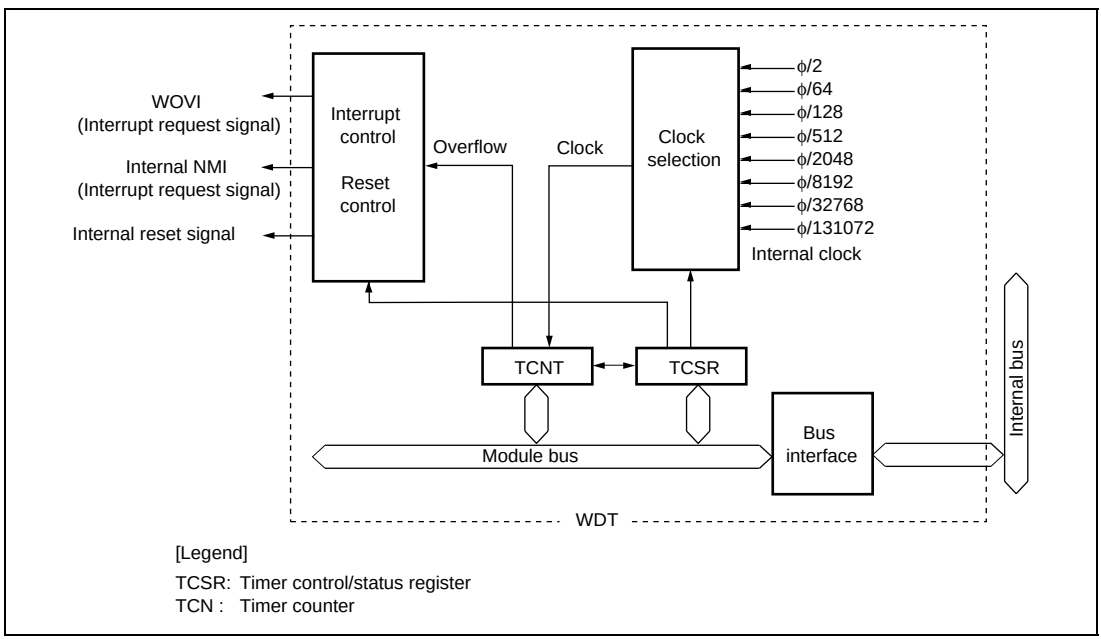


Figure 15.1 Block Diagram of WDT

15.2 Register Descriptions

The WDT has the following registers. To prevent accidental overwriting, TCNT and TCSR have to be written to in a method different from normal registers. For details, see section 15.5.1, 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)

15.2.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter.

TCNT is initialized to H'00 when the TME bit in the timer control/status register (TCSR) is cleared to 0.

15.2.2 Timer Control/Status Register (TCSR)

TCSR selects the clock source to be input to TCNT, and the timer mode.

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, this bit 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

Bit	Bit Name	Initial Value	R/W	Description
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/$\overline{\text{NMI}}$	0	R/W	Reset or NMI Selects whether an internal reset or an NMI interrupt is requested when TCNT has overflowed. 0: An NMI interrupt is requested 1: An internal reset is requested
2	CKS2	0	R/W	Clock Select 2 to 0 Select the clock source to be input to TCNT. The overflow cycle for ϕ = 20 MHz is enclosed in parentheses. 000: ϕ/2 (cycle: 25.6 μs) 001: ϕ/64 (cycle: 819.2 μs) 010: ϕ/128 (cycle: 1.6 ms) 011: ϕ/512 (cycle: 6.5 ms) 100: ϕ/2048 (cycle: 26.2 ms) 101: ϕ/8192 (cycle: 104.8 ms) 110: ϕ/32768 (cycle: 419.4 ms) 111: ϕ/131072 (cycle: 1.67 s)
1	CKS1	0	R/W	
0	CKS0	0	R/W	

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

15.3 Operation

15.3.1 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 writing H'00) before overflows occurs.

If the RST/$\overline{NMI}$ bit in 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 15.2. 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. A reset source can be identified by the state of the XRST bit 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.

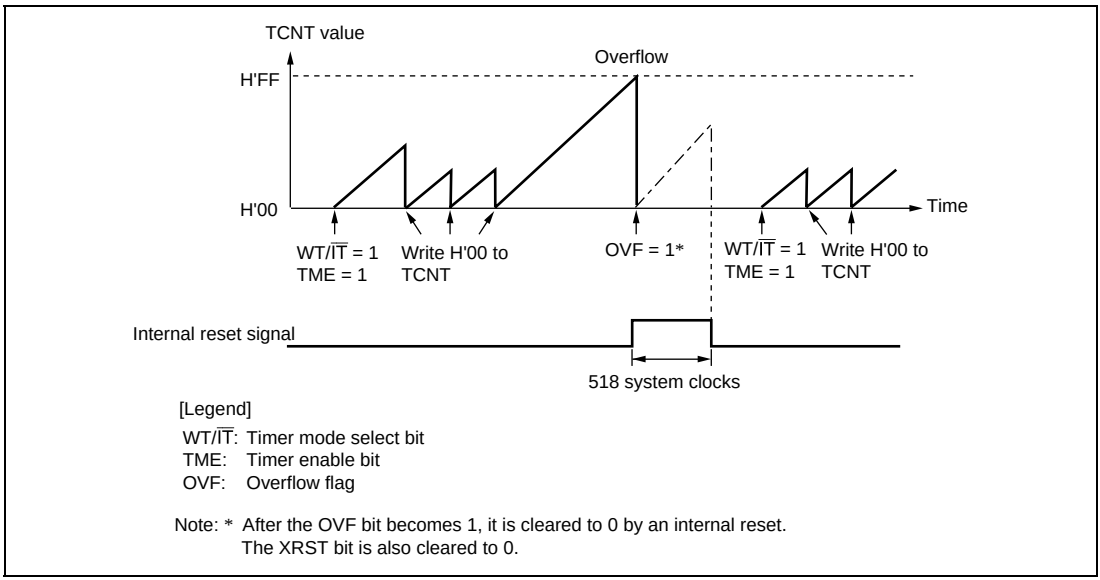


Figure 15.2 Watchdog Timer Mode (RST/$\overline{NMI}$ = 1) Operation

15.3.2 Interval Timer Mode

When the WDT is used as an interval timer, an interval timer interrupt (WOVI) is generated each time TCNT overflows, as shown in figure 15.3. Therefore, an interrupt can be generated at intervals.

When TCNT overflows in interval timer mode, the OVF bit in TCSR is set to 1 and at the same time an interval timer interrupt (WOVI) is requested. The timing is shown in figure 15.4.

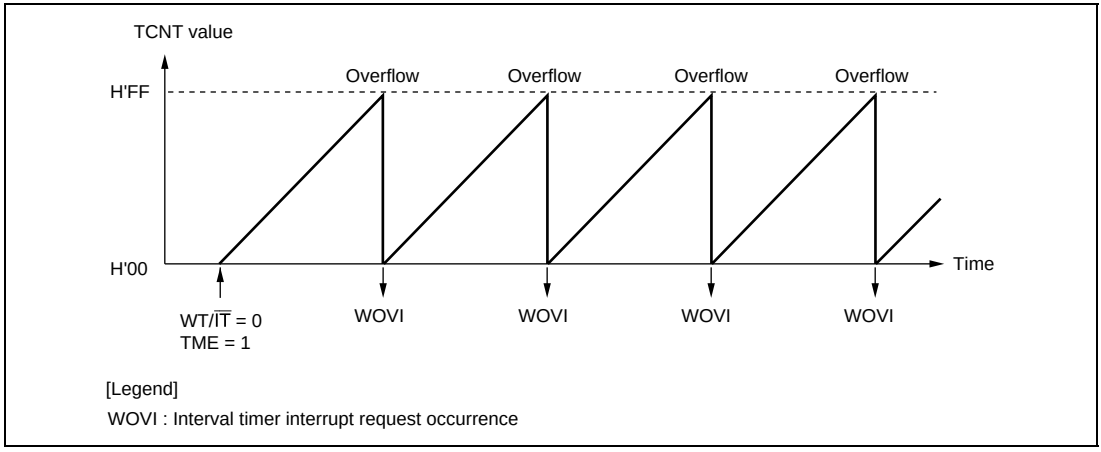


Figure 15.3 Interval Timer Mode Operation

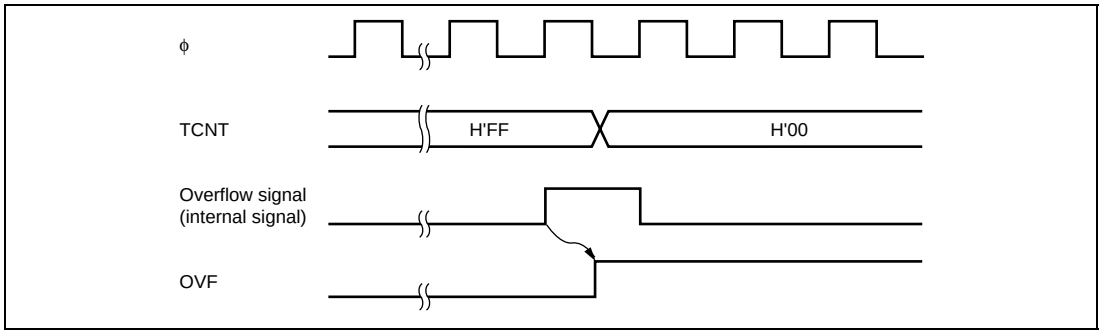


Figure 15.4 OVF Flag Set Timing

15.3.3 Internal Reset Signal Generation Timing

When TCNT overflows in watchdog timer mode, the OVF bit in TCSR is set to 1. When the RST/$\overline{\text{NMI}}$ bit is set to 1 here, the internal reset signal is generated for the entire LSI. The timing is shown in figure 15.5.

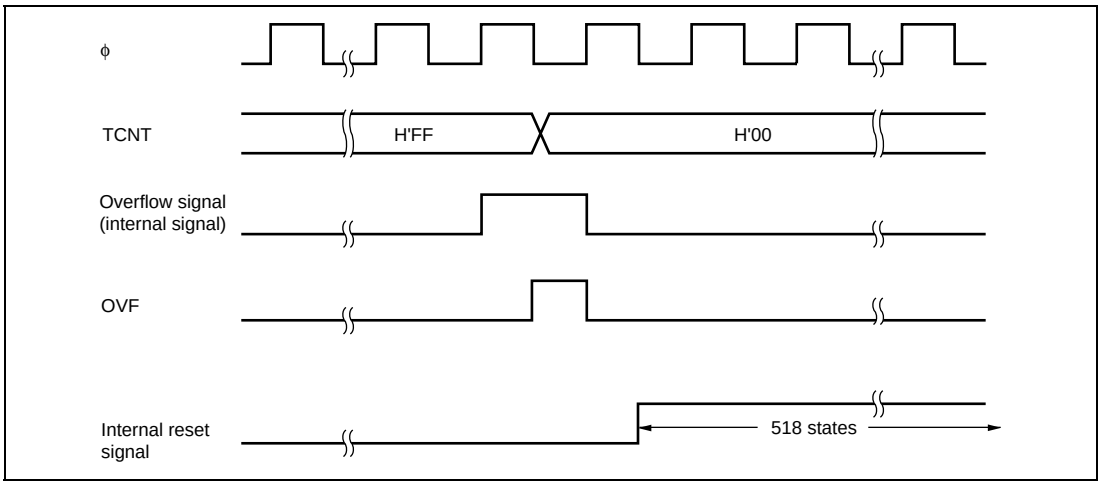


Figure 15.5 Internal Reset Signal Generation Timing

15.4 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. The OVF flag 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 15.1 Interrupt Source

Name	Interrupt Source	Interrupt Flag
WOVI	TCNT overflow	OVF

15.5 Usage Notes

15.5.1 Notes on Register Access

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

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 15.6 to write to TCNT or TCSR. To write to TCNT, the upper 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 upper bytes must contain the value H′A5 and the lower bytes must contain the write data before the transfer instruction execution.

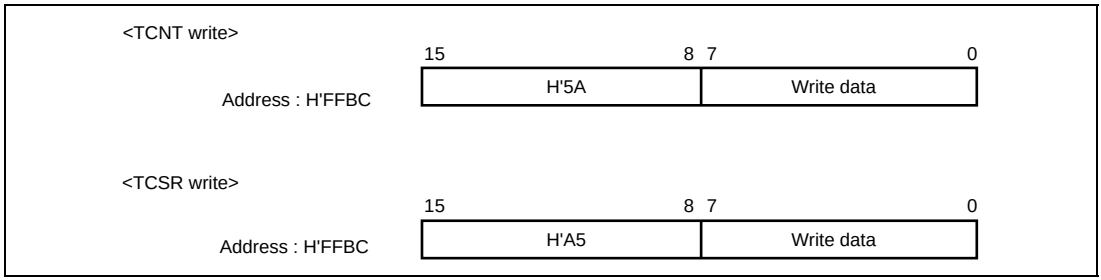


Figure 15.6 Writing to TCNT and TCSR

(2) Reading from TCNT and TCSR

These registers are read in the same way as other registers. The read address is H'FFBC for TCSR and H'FFBD for TCNT.

15.5.2 Conflict between Timer Counter (TCNT) Write and Increment

Even 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 15.7 shows this operation.

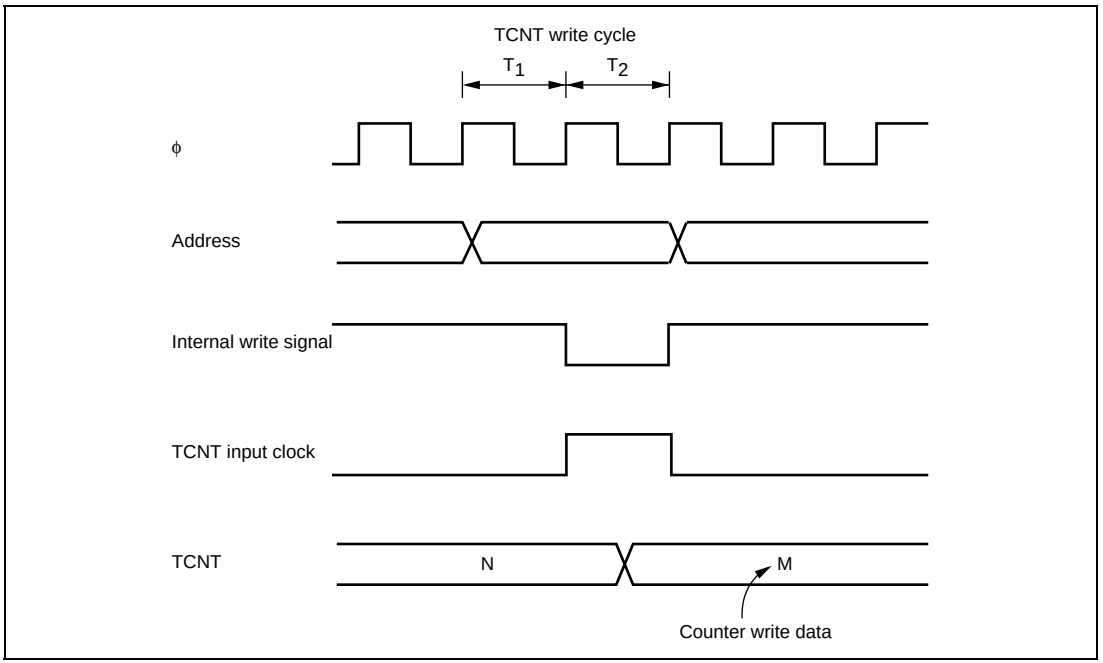


Figure 15.7 Conflict between TCNT Write and Increment

15.5.3 Changing Values of CKS2 to CKS0 Bits

If bits CKS2 to CKS0 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 bits CKS2 to CKS0.

15.5.4 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched between watchdog timer and interval timer, 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 switching the mode.

Section 16 Serial Communication Interface (SCI)

This LSI has five 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).

16.1 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.
- On-chip baud rate generator allows any bit rate to be selected

  The 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.
- Module stop mode availability

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

Figure 16.1 shows a block diagram of the SCI.

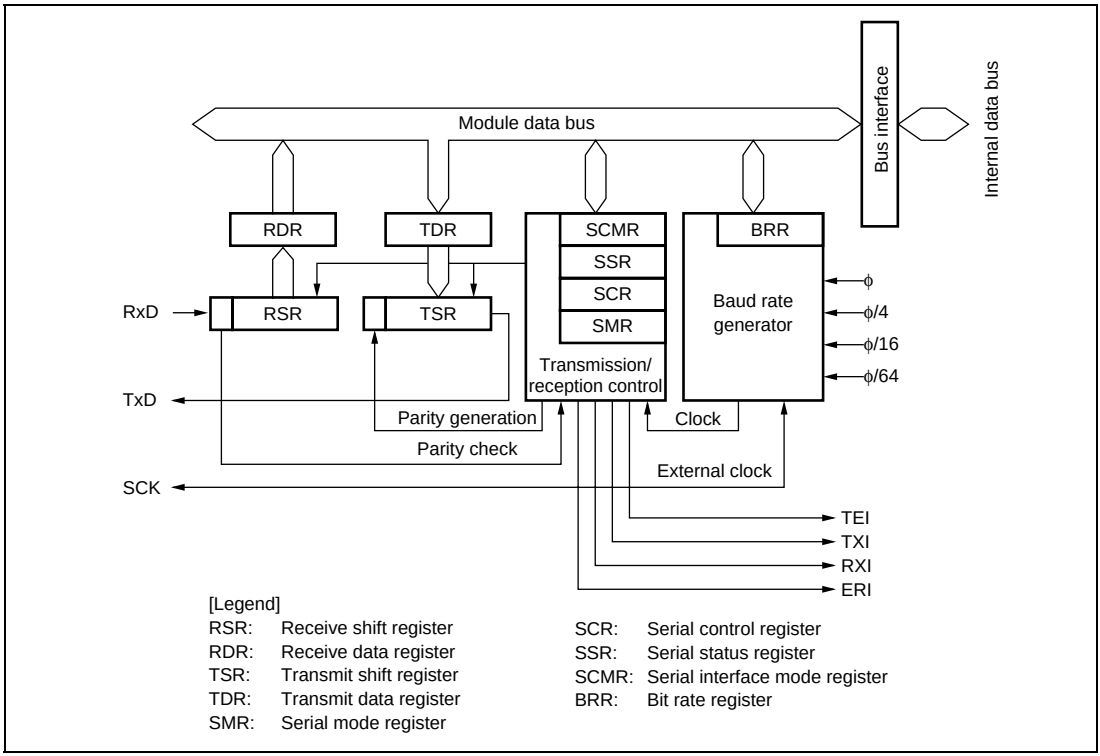


Figure 16.1 Block Diagram of SCI

16.2 Input/Output Pins

Table 16.1 shows the input/output pins for the SCI.

Table 16.1 Pin Configuration

Channel	Symbol*	I/O	Function
0	SCK0	I/O	Channel 0 clock input/output
	RxD0	Input	Channel 0 receive data input
	TxD0	Output	Channel 0 transmit data output
1	SCK1	I/O	Channel 1 clock input/output
	RxD1	Input	Channel 1 receive data input
	TxD1	Output	Channel 1 transmit data output
2	SCK2	I/O	Channel 2 clock input/output
	RxD2	Input	Channel 2 receive data input
	TxD2	Output	Channel 2 transmit data output
3	SCK3	I/O	Channel 3 clock input/output
	RxD3	Input	Channel 3 receive data input
	TxD3	Output	Channel 3 transmit data output
4	SCK4	I/O	Channel 4 clock input/output
	RxD4	Input	Channel 4 receive data input
	TxD4	Output	Channel 4 transmit data output

Note: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting the channel designation.

16.3 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)

16.3.1 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.

16.3.2 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 is 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. The initial value of RDR is H'00.

16.3.3 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 structure 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. The initial value of TDR is H'FF.

16.3.4 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.

16.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI's serial transfer format and select the clock source for the on-chip baud rate generator.

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.

Bit	Bit Name	Initial Value	R/W	Description
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 regardless of the STOP bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.
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 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 CKS bit settings and the baud rate, see section 16.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 16.3.9, Bit Rate Register (BRR)).

16.3.6 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. For details on interrupt requests, see section 16.7, 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.
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, see section 16.5, 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.

Bit	Bit Name	Initial Value	R/W	Description
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 to the SCK pin.) Clocked synchronous mode 0x: Internal clock (SCK pin functions as clock output.) 1x: External clock (SCK pin functions as clock input.)

Note: x: Don’t care.

16.3.7 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 data is written to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates whether 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 data is read from RDR The RDRF flag is not affected and retains its previous value even if the RE bit in SCR is cleared to 0.
5	ORER	0	R/(W)*	Overrun Error [Setting condition] • When the next serial reception is completed 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 data is written to TDR
1	MPB	0	R	Multiprocessor Bit Stores the multiprocessor bit in the receive frame. When the RE bit in SCR is cleared to 0, this bit is not changed.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Sets the multiprocessor bit to be added to the transmit frame.

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

16.3.8 Serial Interface Mode Register (SCMR)

SCMR is a register that selects the SCI functions.

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. Stores receive data as LSB first in RDR. 1: TDR contents are transmitted with MSB-first. Stores receive data 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	—	0	R/W	Reserved The initial value should not be changed.

16.3.9 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 16.2 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, and BRR can be read from or written to by the CPU at all times.

Table 16.2 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	$B = \frac{\phi \times 10^6}{8 \times 2^{2n-1} \times (N+1)}$	—
Smart card interface mode	$B = \frac{\phi \times 10^6}{S \times 2^{2n+1} \times (N+1)}$	$\text{Error (\%)} = \{\frac{\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1\} \times 100$

Notes: B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \le N \le 255$)
ϕ: Operating frequency (MHz)
n and S: 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

SMR Setting		
BCP1	**BCP0**	**S**
0	0	32
0	1	64
1	0	372
1	1	256

Table 16.3 shows sample N settings in BRR in normal asynchronous mode. Table 16.4 shows the maximum bit rate settable for each operating frequency. Table 16.6 shows sample N settings in BRR in clocked synchronous mode. Tables 16.5 and 16.7 show the maximum bit rates with external clock input.

Table 16.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1)

	Operating Frequency ϕ (MHz)											
	2			2.097152			2.4576			3		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	141	0.03	1	148	–0.04	1	174	–0.26	1	212	0.03
150	1	103	0.16	1	108	0.21	1	127	0.00	1	155	0.16
300	0	207	0.16	0	217	0.21	0	255	0.00	1	77	0.16
600	0	103	0.16	0	108	0.21	0	127	0.00	0	155	0.16
1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	77	0.16
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38	0.16
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	19	–2.34
9600	—	—	—	0	6	–2.48	0	7	0.00	0	9	–2.34
19200	—	—	—	—	—	—	0	3	0.00	0	4	–2.34
31250	0	1	0.00	—	—	—	—	—	—	0	2	0.00
38400	—	—	—	—	—	—	0	1	0.00	—	—	—

	Operating Frequency ϕ (MHz)											
	3.6864			4			4.9152			5		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88	–0.25
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64	0.16
300	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
19200	0	5	0.00	—	—	—	0	7	0.00	0	7	1.73
31250	—	—	—	0	3	0.00	0	4	–1.70	0	4	0.00
38400	0	2	0.00	—	—	—	0	3	0.00	0	3	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 16.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2)

	Operating Frequency ϕ (MHz)											
	6			6.144			7.3728			8		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	106	–0.44	2	108	0.08	2	130	–0.07	2	141	0.03
150	2	77	0.16	2	79	0.00	2	95	0.00	2	103	0.16
300	1	155	0.16	1	159	0.00	1	191	0.00	1	207	0.16
600	1	77	0.16	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	0.00	0	5	2.40	—	—	—	0	7	0.00
38400	0	4	–2.34	0	4	0.00	0	5	0.00	—	—	—

	Operating Frequency ϕ (MHz)											
	9.8304			10			12			12.288		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

[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 16.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (3)

	Operating Frequency ϕ (MHz)											
	14			14.7456			16			17.2032		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	64	0.70	3	70	0.03	3	75	0.48
150	2	181	0.16	2	191	0.00	2	207	0.16	2	223	0.00
300	2	90	0.16	2	95	0.00	2	103	0.16	2	111	0.00
600	1	181	0.16	1	191	0.00	1	207	0.16	1	223	0.00
1200	1	90	0.16	1	95	0.00	1	103	0.16	1	111	0.00
2400	0	181	0.16	0	191	0.00	0	207	0.16	0	223	0.00
4800	0	90	0.16	0	95	0.00	0	103	0.16	0	111	0.00
9600	0	45	–0.93	0	47	0.00	0	51	0.16	0	55	0.00
19200	0	22	–0.93	0	23	0.00	0	25	0.16	0	27	0.00
31250	0	13	0.00	0	14	–1.70	0	15	0.00	0	16	1.20
38400	—	—	—	0	11	0.00	0	12	0.16	0	16	0.00

	Operating Frequency ϕ (MHz)								
	18			19.6608			20		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	79	–0.12	3	86	0.31	3	88	–0.25
150	2	233	0.16	2	255	0.00	3	64	0.16
300	2	116	0.16	2	127	0.00	2	129	0.16
600	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	58	–0.69	0	63	0.00	0	64	0.16
19200	0	28	1.02	0	31	0.00	0	32	–1.36
31250	0	17	0.00	0	19	–1.70	0	19	0.00
38400	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 16.4 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode)

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N	ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
2	62500	0	0	9.8304	307200	0	0
2.097152	65536	0	0	10	312500	0	0
2.4576	76800	0	0	12	375000	0	0
3	93750	0	0	12.288	384000	0	0
3.6864	115200	0	0	14	437500	0	0
4	125000	0	0	14.7456	460800	0	0
4.9152	153600	0	0	16	500000	0	0
5	156250	0	0	17.2032	537600	0	0
6	187500	0	0	18	562500	0	0
6.144	192000	0	0	19.6608	614400	0	0
7.3728	230400	0	0	20	625000	0	0
8	250000	0	0				

Table 16.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
2	0.5000	31250	9.8304	2.4576	153600
2.097152	0.5243	32768	10	2.5000	156250
2.4576	0.6144	38400	12	3.0000	187500
3	0.7500	46875	12.288	3.0720	192000
3.6864	0.9216	57600	14	3.5000	218750
4	1.0000	62500	14.7456	3.6864	230400
4.9152	1.2288	76800	16	4.0000	250000
5	1.2500	78125	17.2032	4.3008	268800
6	1.5000	93750	18	4.5000	281250
6.144	1.5360	96000	19.6608	4.9152	307200
7.3728	1.8432	115200	20	5.0000	312500
8	2.0000	125000			

Table 16.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

Bit Rate (bit/s)	Operating Frequency φ (MHz)											
	2		4		8		10		16		20	
	n	N	n	N	n	N	n	N	n	N	n	N
110	3	70	—	—								
250	2	124	2	249	3	124	—	—	3	249		
500	1	249	2	124	2	249	—	—	3	124	—	—
1k	1	124	1	249	2	124	—	—	2	249	—	—
2.5k	0	199	1	99	1	199	1	249	2	99	2	124
5k	0	99	0	199	1	99	1	124	1	199	1	249
10k	0	49	0	99	0	199	0	249	1	99	1	124
25k	0	19	0	39	0	79	0	99	0	159	0	199
50k	0	9	0	19	0	39	0	49	0	79	0	99
100k	0	4	0	9	0	19	0	24	0	39	0	49
250k	0	1	0	3	0	7	0	9	0	15	0	19
500k	0	0*	0	1*	0	3	0	4	0	7	0	9
1M			0	0	0	1			0	3	0	4
2.5M							0	0*			0	1
5M											0	0*

[Legend]

Blank: Setting prohibited.

— : Can be set, but there will be a degree of error.

*: Continuous transmission or reception is not possible.

Table 16.7 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)
2	0.3333	333333.3	12	2.0000	2000000.0
4	0.6667	666666.7	14	2.3333	2333333.3
6	1.0000	1000000.0	16	2.6667	2666666.7
8	1.3333	1333333.3	18	3.0000	3000000.0
10	1.6667	1666666.7	20	3.3333	3333333.3

16.4 Operation in Asynchronous Mode

Figure 16.2 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 detects 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 from or written to during transmission or reception, enabling continuous data transmission and reception.

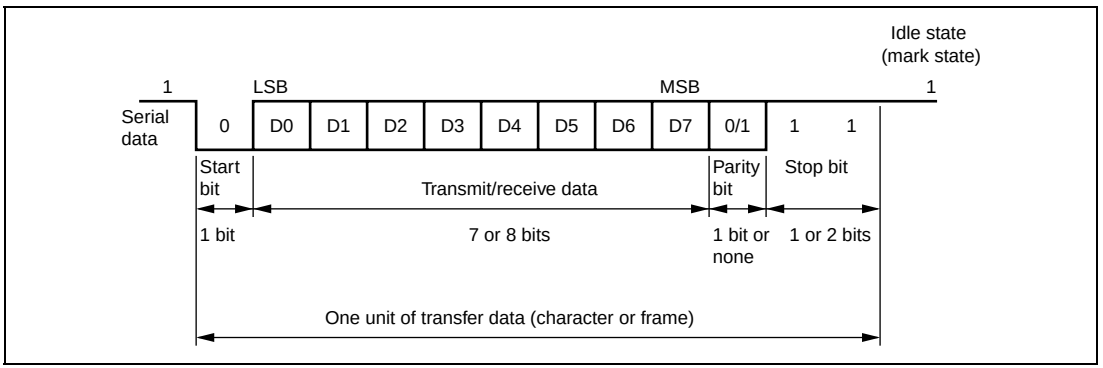


Figure 16.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

16.4.1 Data Transfer Format

Table 16.8 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, see section 16.5, Multiprocessor Communication Function.

Table 16.8 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

16.4.2 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. If receive data is sampled 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 16.3. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \{ (0.5 - \frac{1}{2N}) - \frac{D - 0.5}{N} - (L - 0.5) F \} \times 100 \quad [\%] \quad \cdots \quad \text{Formula (1)}$$

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.

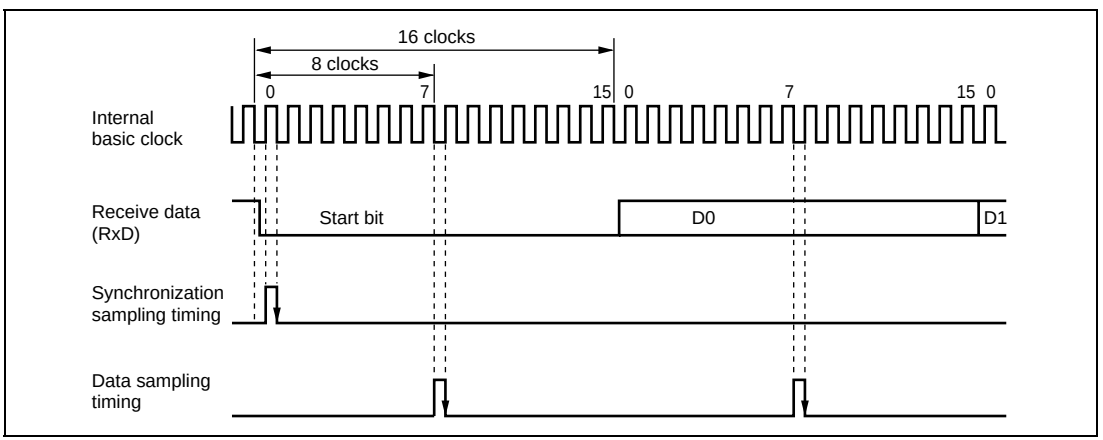


Figure 16.3 Receive Data Sampling Timing in Asynchronous Mode

16.4.3 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.

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 16.4.

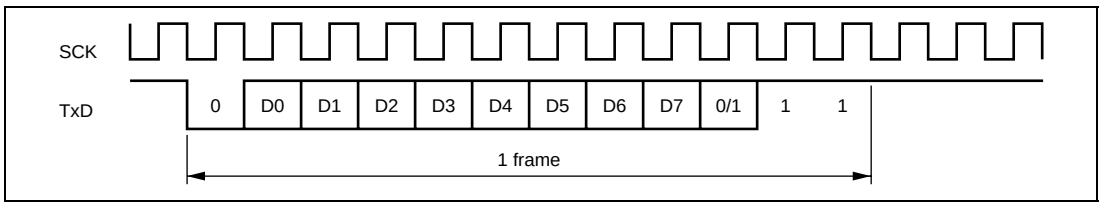
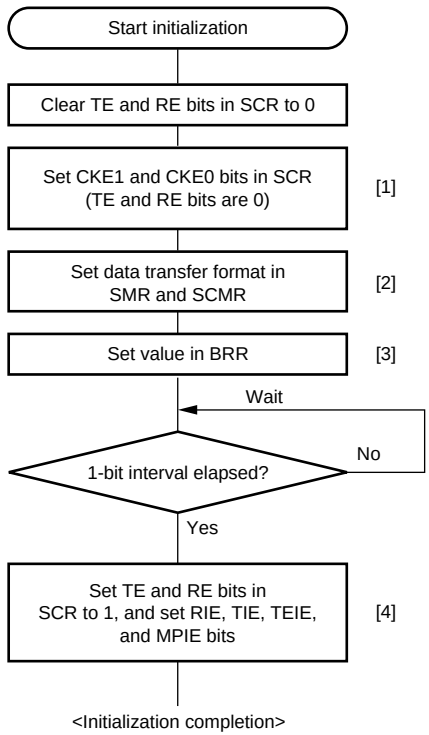


Figure 16.4 Relation between Output Clock and Transmit Data Phase (Asynchronous Mode)

16.4.4 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, first clear the TE and RE bits in SCR to 0, then initialize the SCI as shown in figure 16.5. 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 the external clock is used in asynchronous mode, the clock must be supplied even during initialization.



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

When the clock is selected in asynchronous mode, it is output immediately after SCR settings are made.

[2] Set the data transfer format in SMR and SCMR.

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

[4] Wait at least one bit interval, then set the TE bit or RE bit in 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.

Figure 16.5 Sample SCI Initialization Flowchart

16.4.5 Serial Data Transmission (Asynchronous Mode)

Figure 16.6 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 (TXI) interrupt request is generated. If 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 next transmit 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 16.7 shows a sample flowchart for transmission in asynchronous mode.

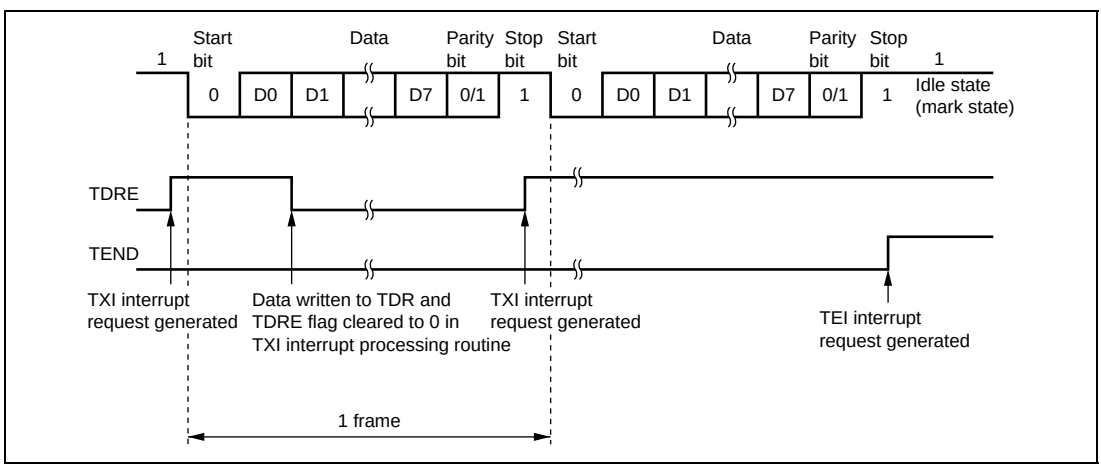
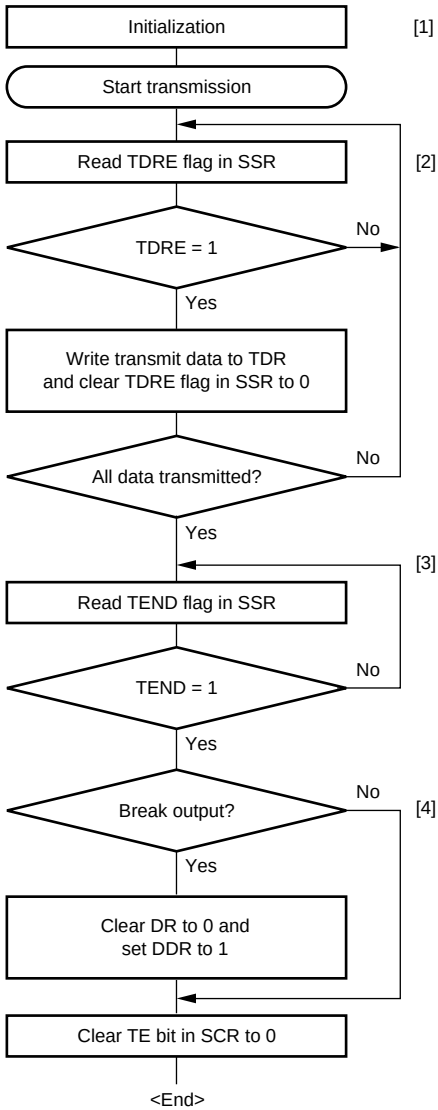


Figure 16.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)



[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.

[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.

Figure 16.7 Sample Serial Transmission Flowchart

16.4.6 Serial Data Reception (Asynchronous Mode)

Figure 16.8 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. If the RXI interrupt processing routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.

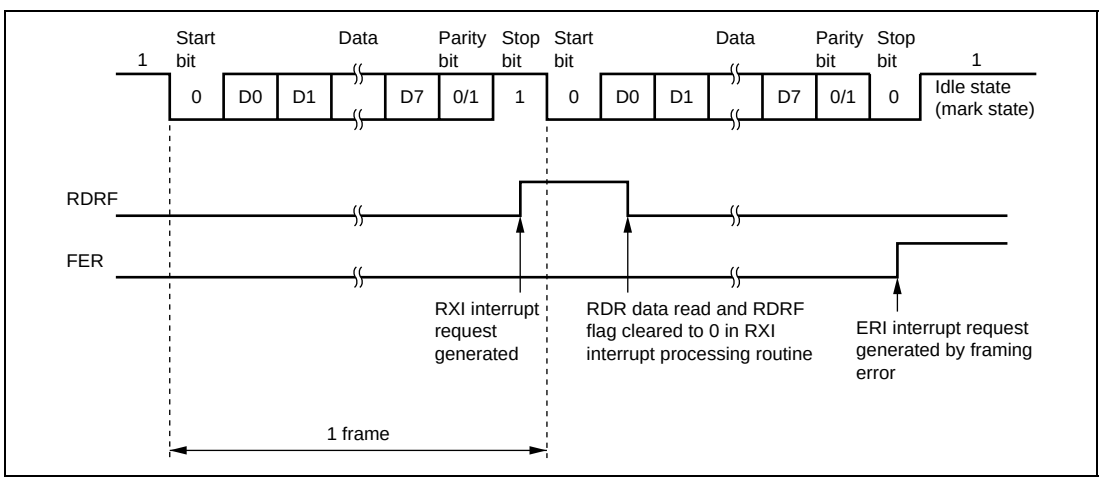


Figure 16.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

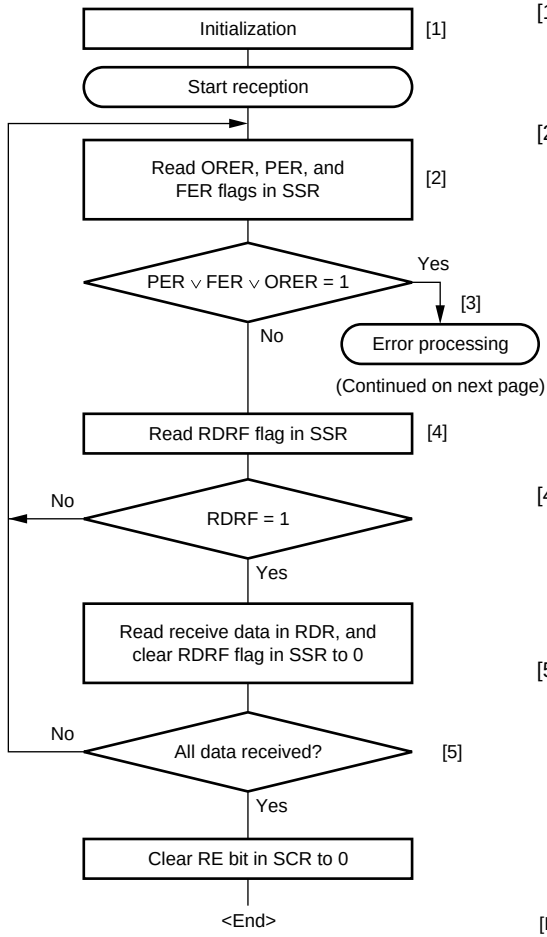
Table 16.9 shows the states of the SSR status flags and receive data handling when a receive error is detected. If a receive error is detected, the RDRF flag retains its state before receiving data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER,

FER, PER, and RDRF flags to 0 before resuming reception. Figure 16.9 shows a sample flowchart for serial data reception.

Table 16.9 SSR Status Flags and Receive Data Handling

SSR Status Flags					
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.

[Legend]
∨ : Logical add (OR)

Figure 16.9 Sample Serial Reception Flowchart (1)

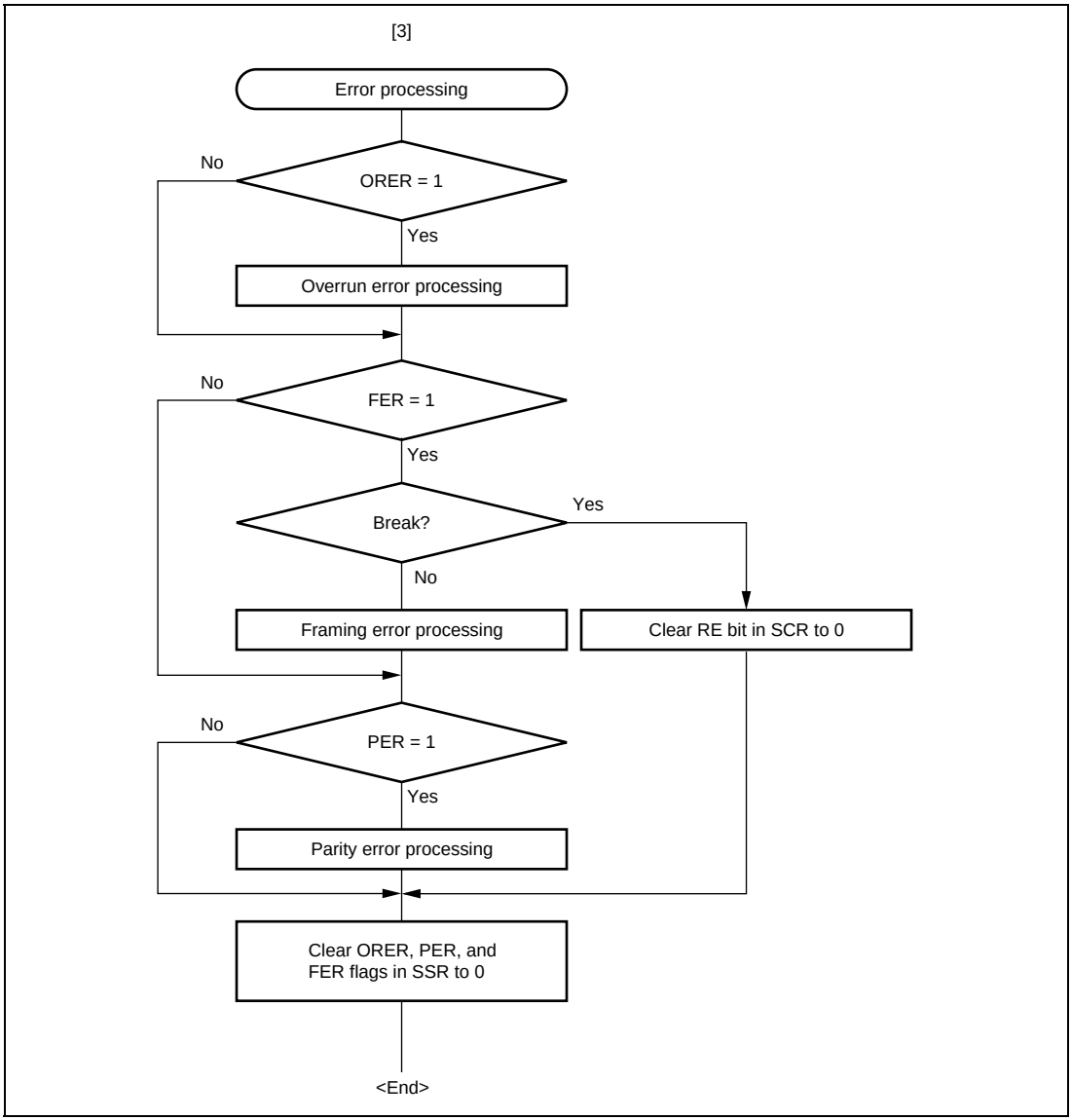


Figure 16.9 Sample Serial Reception Flowchart (2)

16.5 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 16.10 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends communication data with a 1 multiprocessor bit added to the ID code of the receiving station. 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.

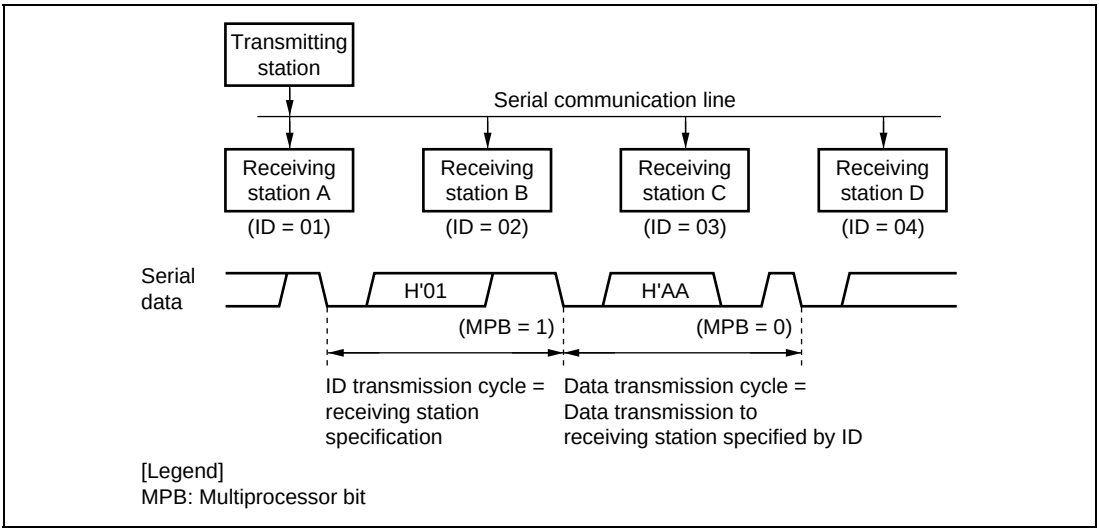


Figure 16.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

16.5.1 Multiprocessor Serial Data Transmission

Figure 16.11 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.

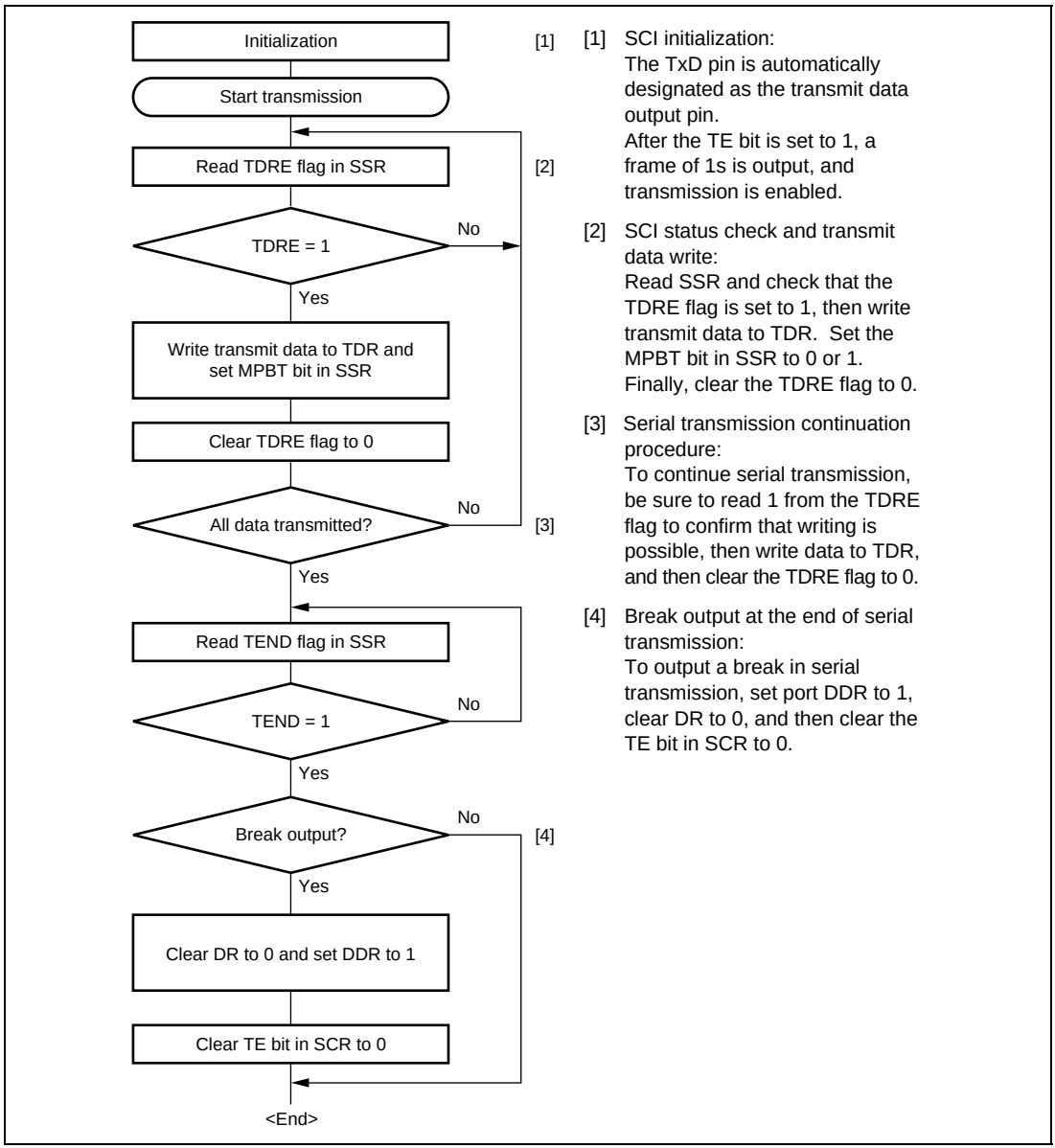


Figure 16.11 Sample Multiprocessor Serial Transmission Flowchart

16.5.2 Multiprocessor Serial Data Reception

Figure 16.13 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 those in asynchronous mode. Figure 16.12 shows an example of SCI operation for multiprocessor format reception.

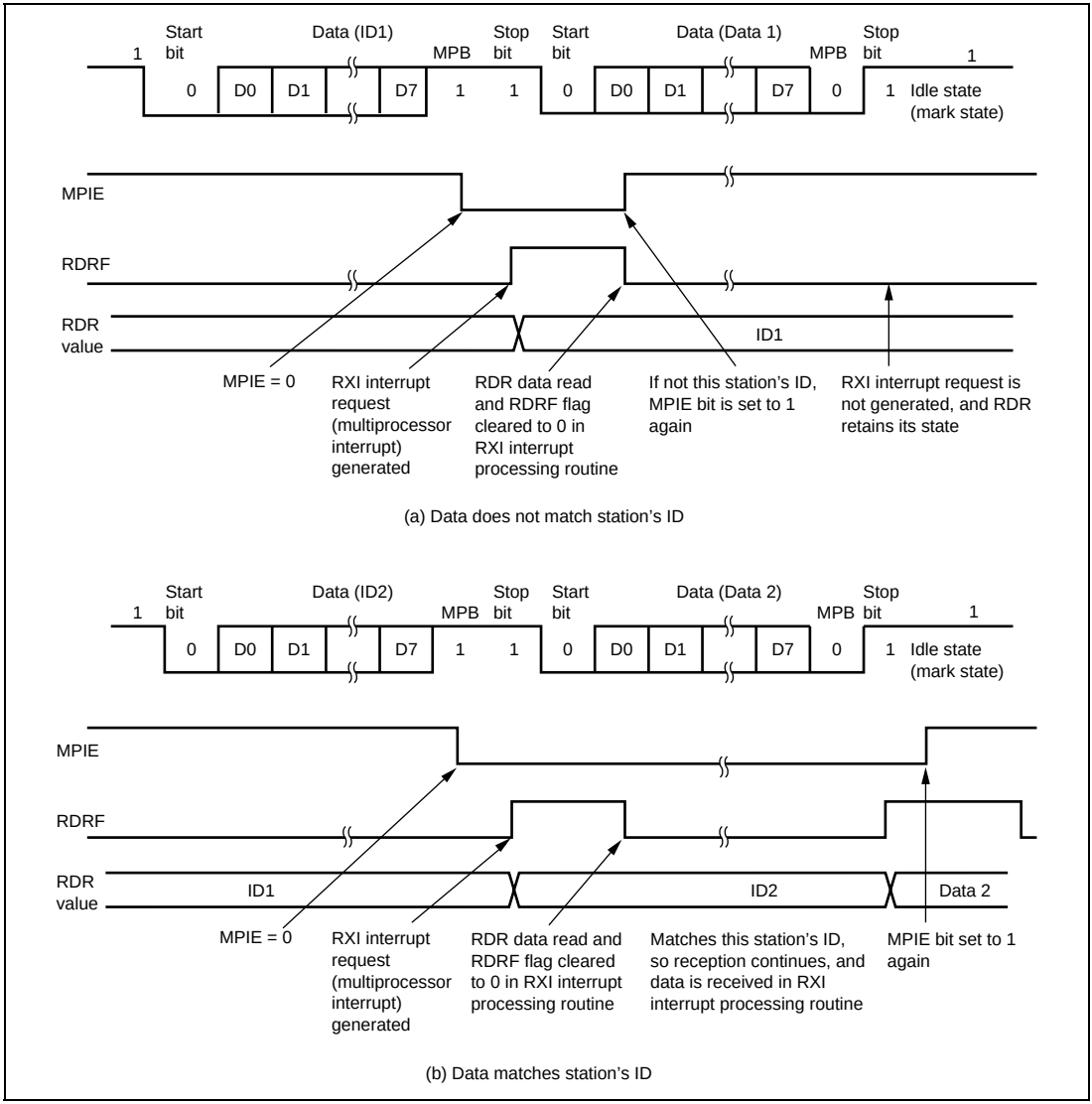
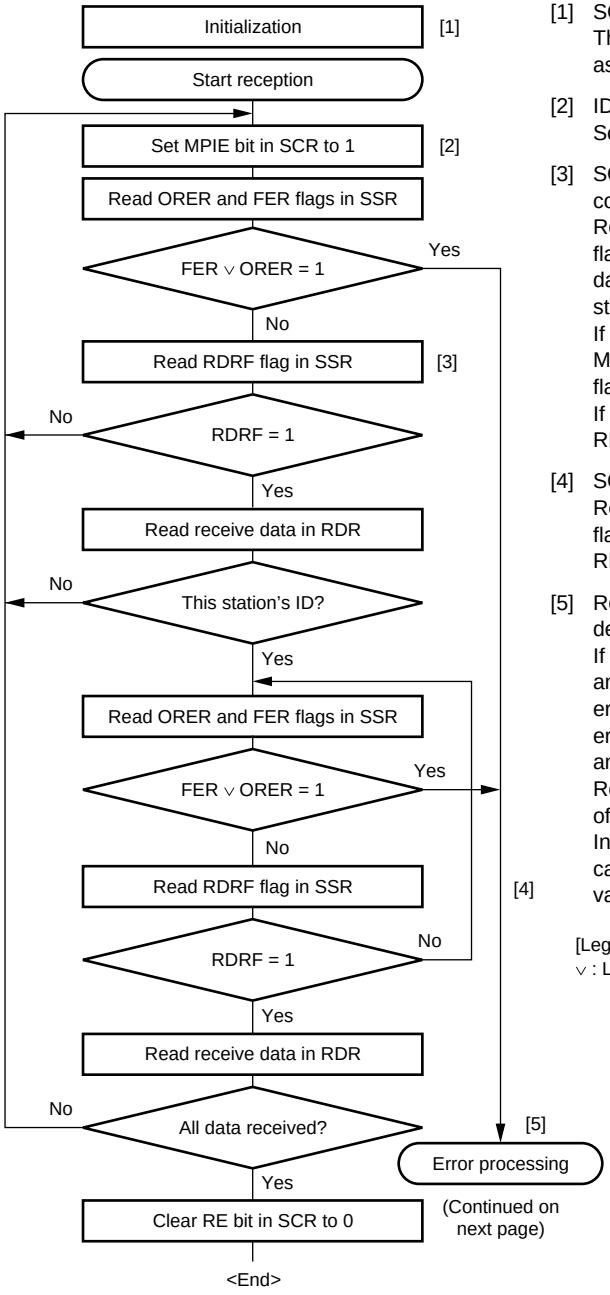


Figure 16.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



[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 add (OR)

Figure 16.13 Sample Multiprocessor Serial Reception Flowchart (1)

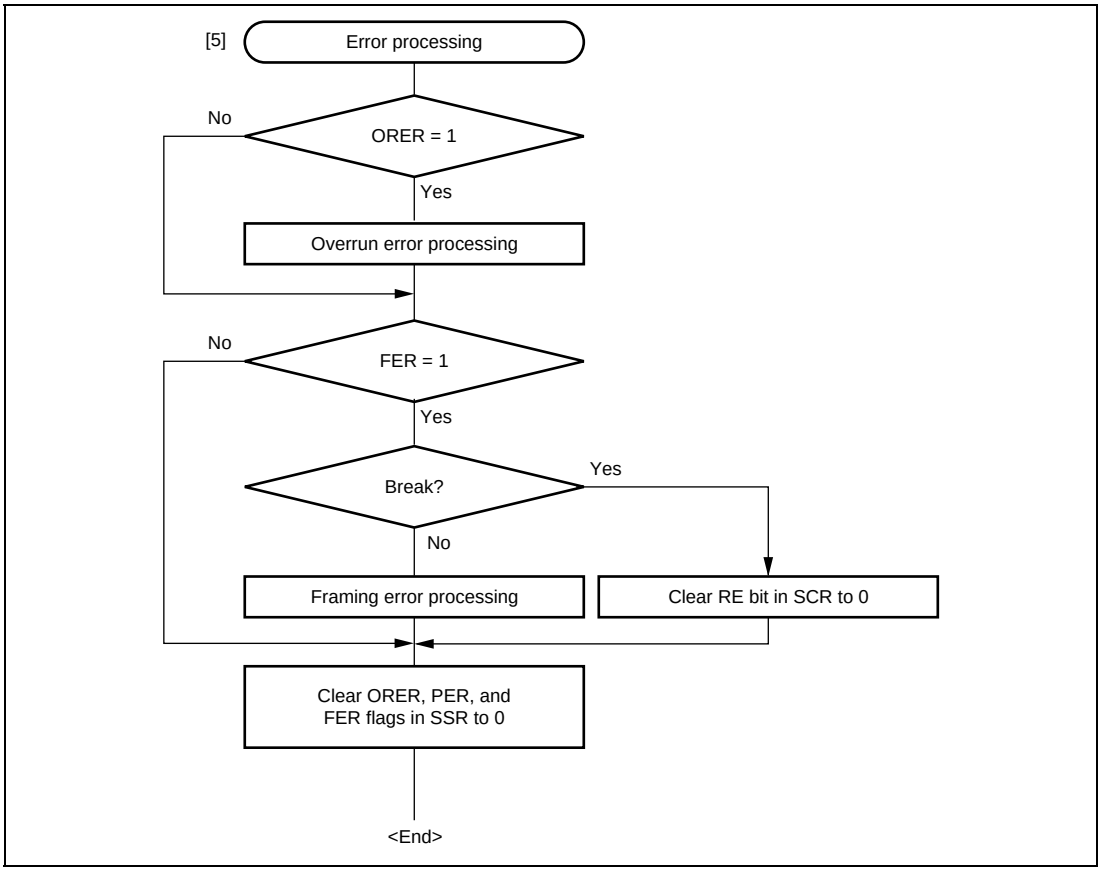


Figure 16.13 Sample Multiprocessor Serial Reception Flowchart (2)

16.6 Operation in Clocked Synchronous Mode

Figure 16.14 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 last-bit output 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.

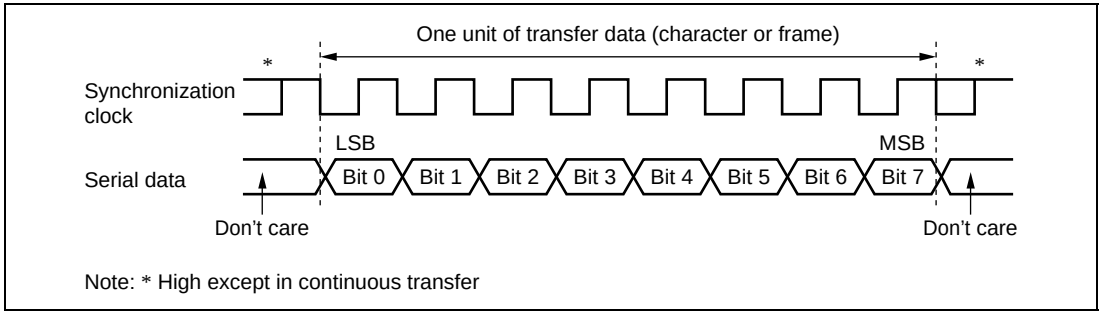


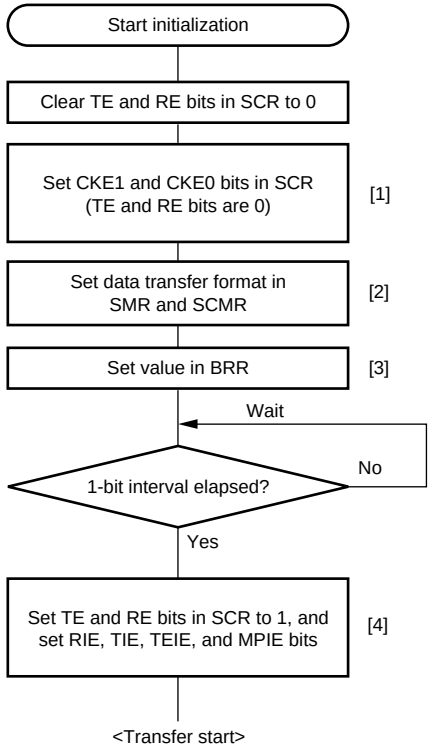
Figure 16.14 Data Format in Clocked Synchronous Communication (LSB-First)

16.6.1 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.

16.6.2 SCI Initialization (Clocked Synchronous Mode)

Before transmitting and receiving data, first clear the TE and RE bits in SCR to 0, then initialize the SCI as described in a sample flowchart in figure 16.15. 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, MPIE, TE, and RE to 0.

[2] Set the data transfer 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 16.15 Sample SCI Initialization Flowchart

16.6.3 Serial Data Transmission (Clocked Synchronous Mode)

Figure 16.16 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. If 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 output 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, the next transmit 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 SCI 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 16.17 shows a sample flowchart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (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.

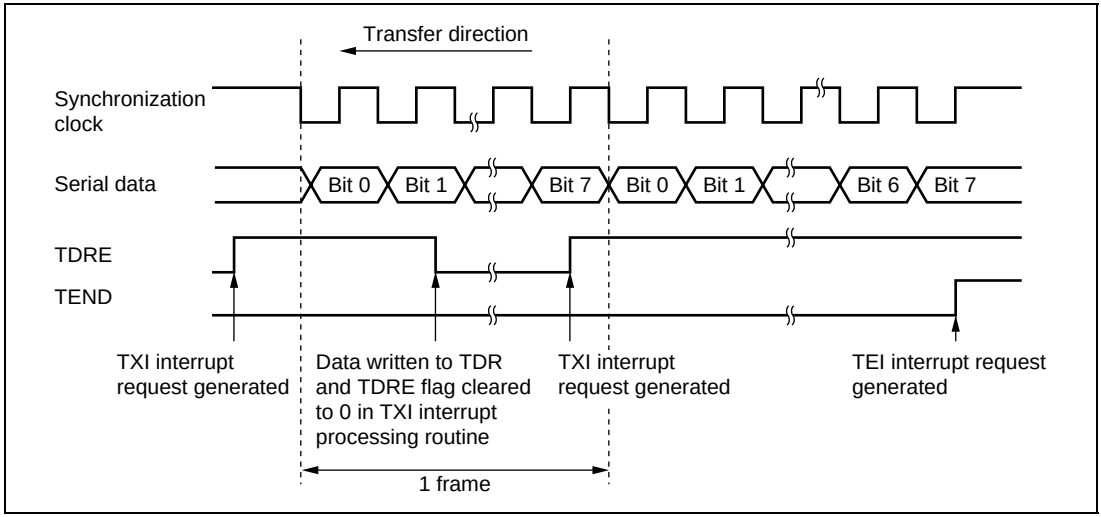
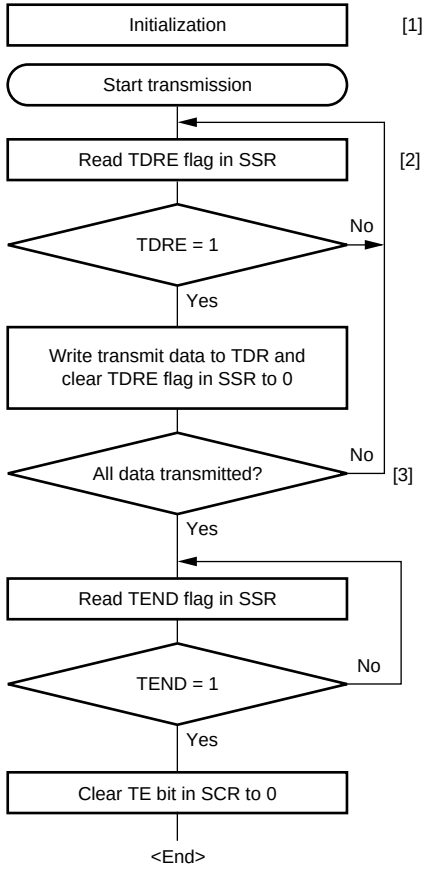


Figure 16.16 Sample SCI Transmission Operation in Clocked Synchronous Mode



[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.

Figure 16.17 Sample Serial Transmission Flowchart

16.6.4 Serial Data Reception (Clocked Synchronous Mode)

Figure 16.18 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 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 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. If the RXI interrupt processing routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.

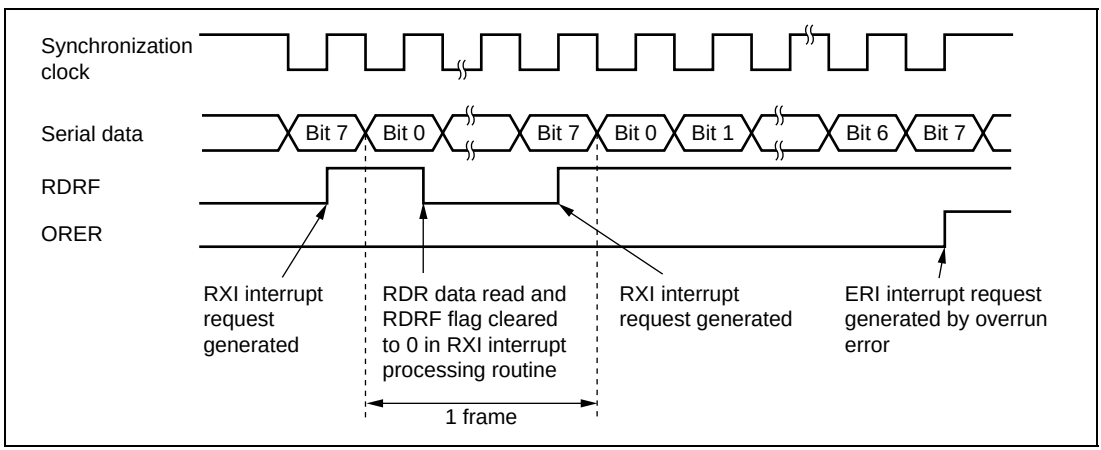
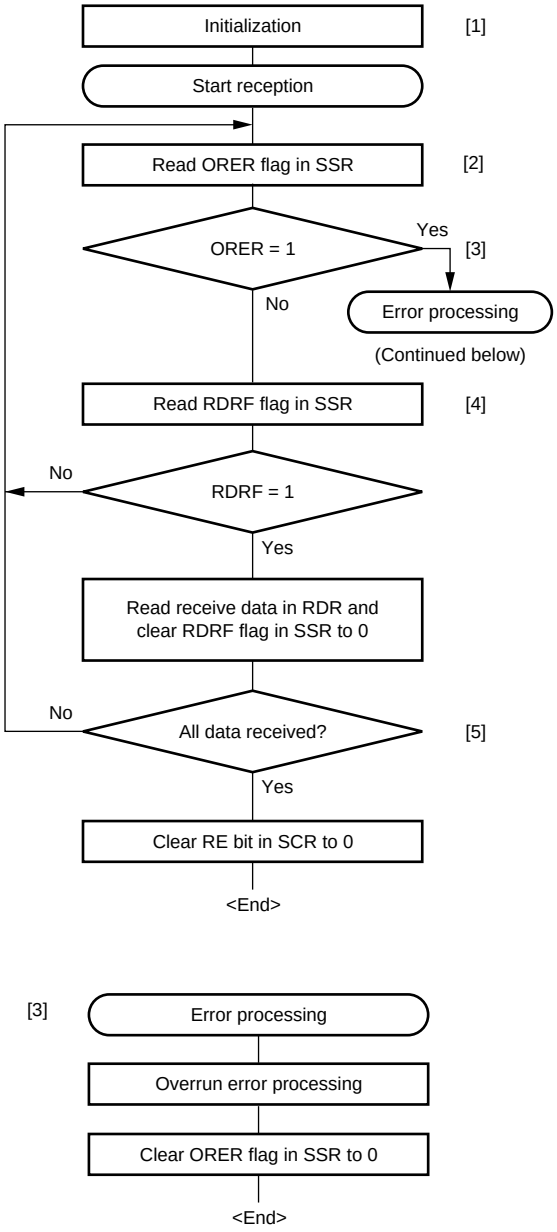


Figure 16.18 Example of SCI Receive Operation in Clocked Synchronous Mode

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF flags to 0 before resuming reception. Figure 16.19 shows a sample flowchart for serial data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [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. Transfer cannot be resumed if the ORER flag is set to 1.

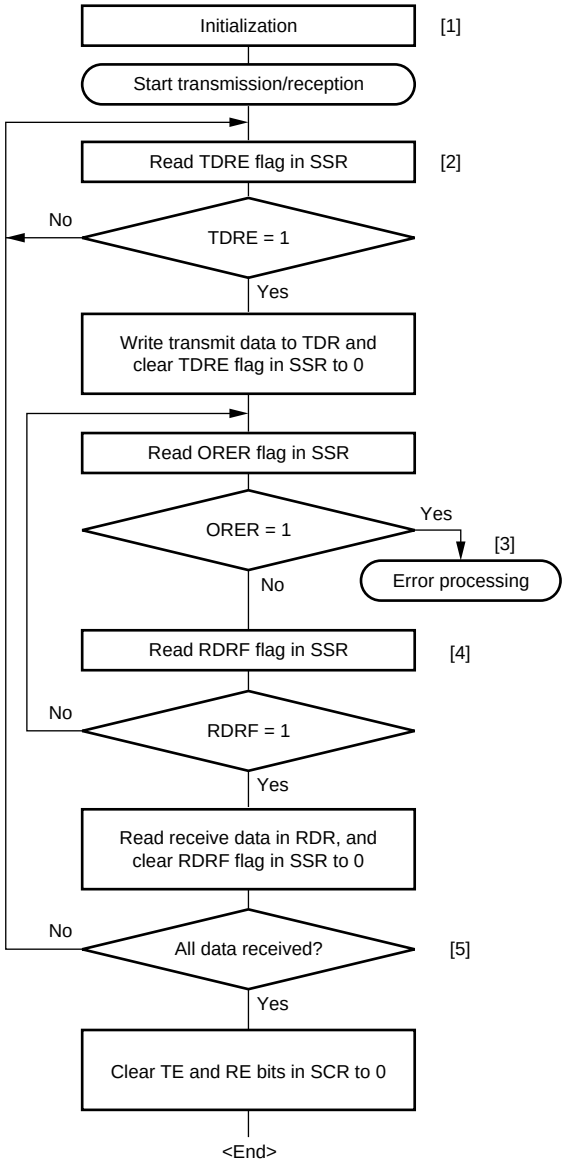
[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0.
Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0 should be finished.

Figure 16.19 Sample Serial Reception Flowchart

16.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode)

Figure 16.20 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, after checking 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. Then simultaneously set the TE and RE bits to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI has finished reception, 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, simultaneously set the TE and RE bits to 1 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.

[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.

Figure 16.20 Sample Flowchart of Simultaneous Serial Transmission and Reception

16.7 Interrupt Sources

Table 16.10 shows the interrupt sources in normal serial communication interface mode. 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.

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.

A TEI interrupt is generated 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 generated 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 16.10 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	Priority
0	ERI0	Receive error	ORER, FER, PER	High
	RXI0	Receive data full	RDRF	
	TXI0	Transmit data empty	TDRE	
	TEI0	Transmit end	TEND	
1	ERI1	Receive error	ORER, FER, PER	
	RXI1	Receive data full	RDRF	
	TXI1	Transmit data empty	TDRE	
	TEI1	Transmit end	TEND	
2	ERI2	Receive error	ORER, FER, PER	
	RXI2	Receive data full	RDRF	
	TXI2	Transmit data empty	TDRE	
	TEI2	Transmit end	TEND	
3	ERI3	Receive error	ORER, FER, PER	
	RXI3	Receive data full	RDRF	
	TXI3	Transmit data empty	TDRE	
	TEI3	Transmit end	TEND	
4	ERI4	Receive error	ORER, FER, PER	
	RXI4	Receive data full	RDRF	
	TXI4	Transmit data empty	TDRE	
	TEI4	Transmit end	TEND	Low

16.8 Usage Notes

16.8.1 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, see section 22, Power-Down Modes.

16.8.2 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.

16.8.3 Mark State and Break Sending

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 mark state (high level) or send a break during serial data transmission. To maintain the communication line at mark state until the TE bit 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.

16.8.4 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only)

Transmission cannot be started when a receive error flag (ORER, FER, or PER) in 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 that the receive error flags cannot be cleared to 0 even if the RE bit in SCR is cleared to 0.

16.8.5 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 while the TDRE flag is 0, the previous data in TDR is lost because the previous data has not been transferred to TSR yet. Be sure to write transmit data to TDR after confirming that the TDRE flag is set to 1.

16.8.6 SCI Operations during Mode Transitions

Transmission: Before making the transition to module stop, software standby, or subsleep mode, stop all 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 the 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 the TE bit to 1, read SSR, write to TDR, clear TDRE to 0 in this order, and then start transmission. To transmit data in a different transmission mode, initialize the SCI first.

Figure 16.21 shows a sample flowchart for mode transition during transmission. Figures 16.22 and 16.23 show the pin states during transmission.

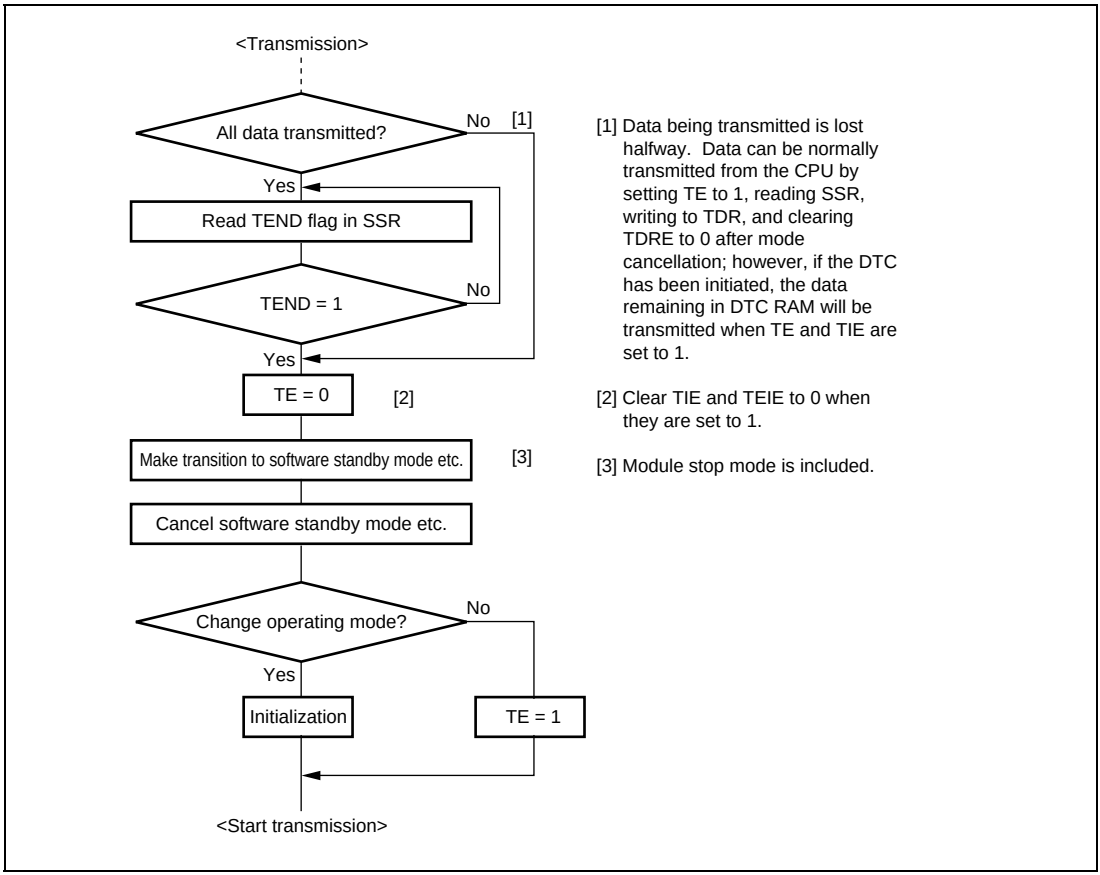


Figure 16.21 Sample Flowchart for Mode Transition during Transmission

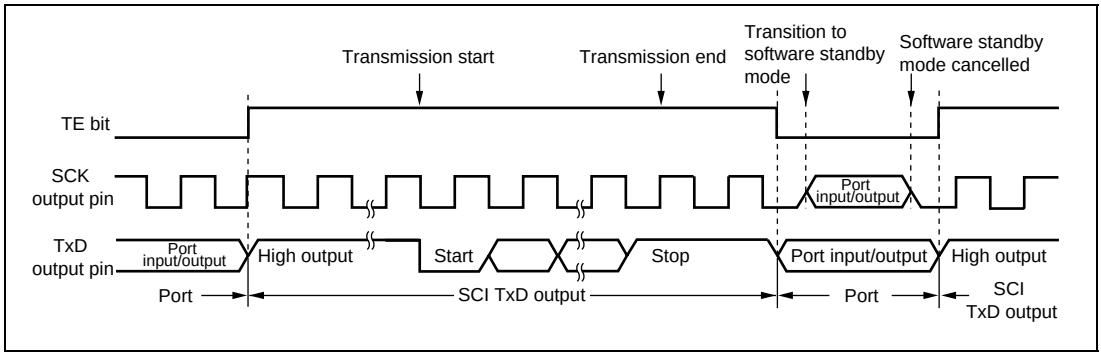


Figure 16.22 Pin States during Transmission in Asynchronous Mode (Internal Clock)

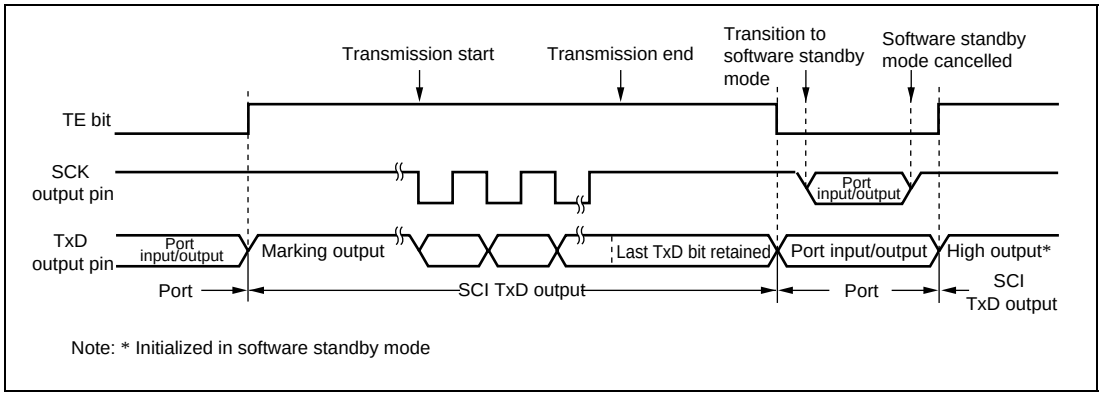


Figure 16.23 Pin States during Transmission in Clocked Synchronous Mode (Internal Clock)

Reception: Before making the transition to module stop, software standby, watch, subactive, or subsleep mode, stop reception (RE = 0). RSR, RDR, and SSR are reset. If 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 16.24 shows a sample flowchart for mode transition during reception.

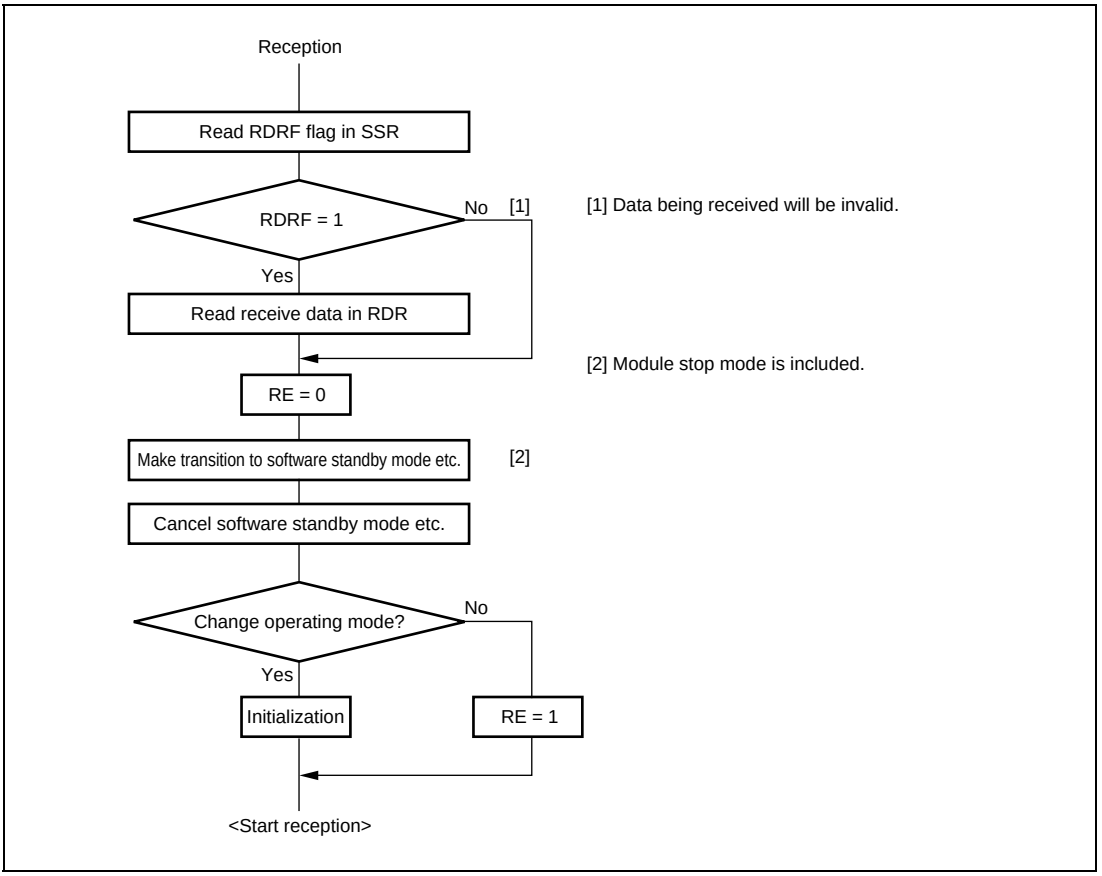


Figure 16.24 Sample Flowchart for Mode Transition during Reception

16.8.7 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 16.25.

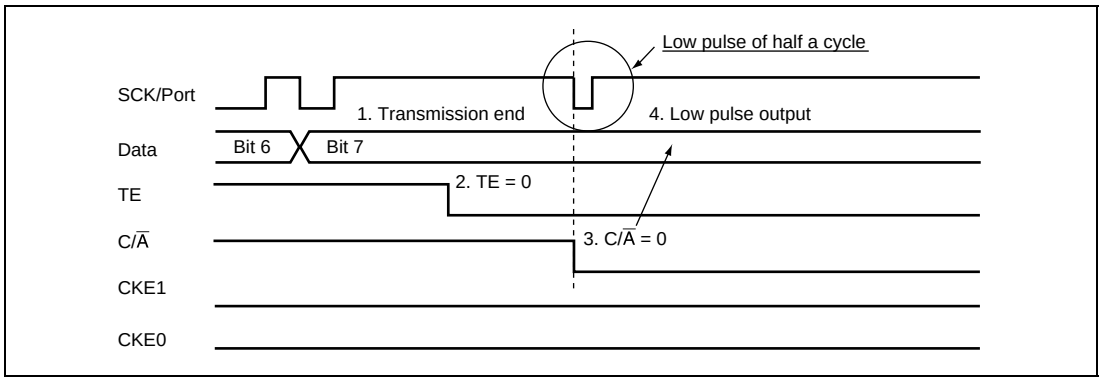


Figure 16.25 Switching from SCK Pins to Port Pins

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, CKE0 = 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

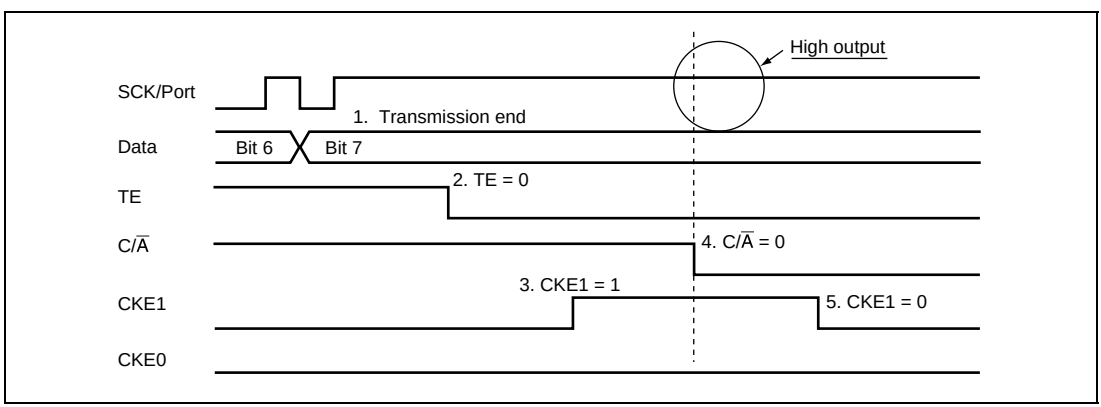


Figure 16.26 Prevention of Low Pulse Output at Switching from SCK Pins to Port Pins

Section 17 I^2C Bus Interface 3 (IIC3)

This LSI has a four-channel I^2C bus interface 3 (IIC3).

The I^2C bus interface conforms to and provides a subset of the Philips I^2C bus (inter-IC bus) interface functions. The register configuration that controls the I^2C bus differs partly from the Philips configuration, however.

Figure 17.1 shows a block diagram of the I^2C bus interface 3.

Figure 17.2 shows an example of I/O pin connections to external circuits.

17.1 Features

- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independent from each other, the continuous transmission/reception can be performed.
- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization/wait function

  In master mode, the state of the SCL is monitored per bit, and the timing is synchronized automatically

  If transfer is not ready, set the SCL to low until preparations are completed.
- Multiple slave addresses can be set

  Maximum three types of slave addresses can be set independently. Using the slave address mask register enables more slave addresses to be set.
- Six interrupt sources

  Transmit-data-empty (including slave-address match), transmit-end, receive-data-full (including slave-address match), arbitration lost, NACK detection, and stop condition detection
- Direct bus drive

  SCL and SDA pins function as NMOS open-drain outputs.

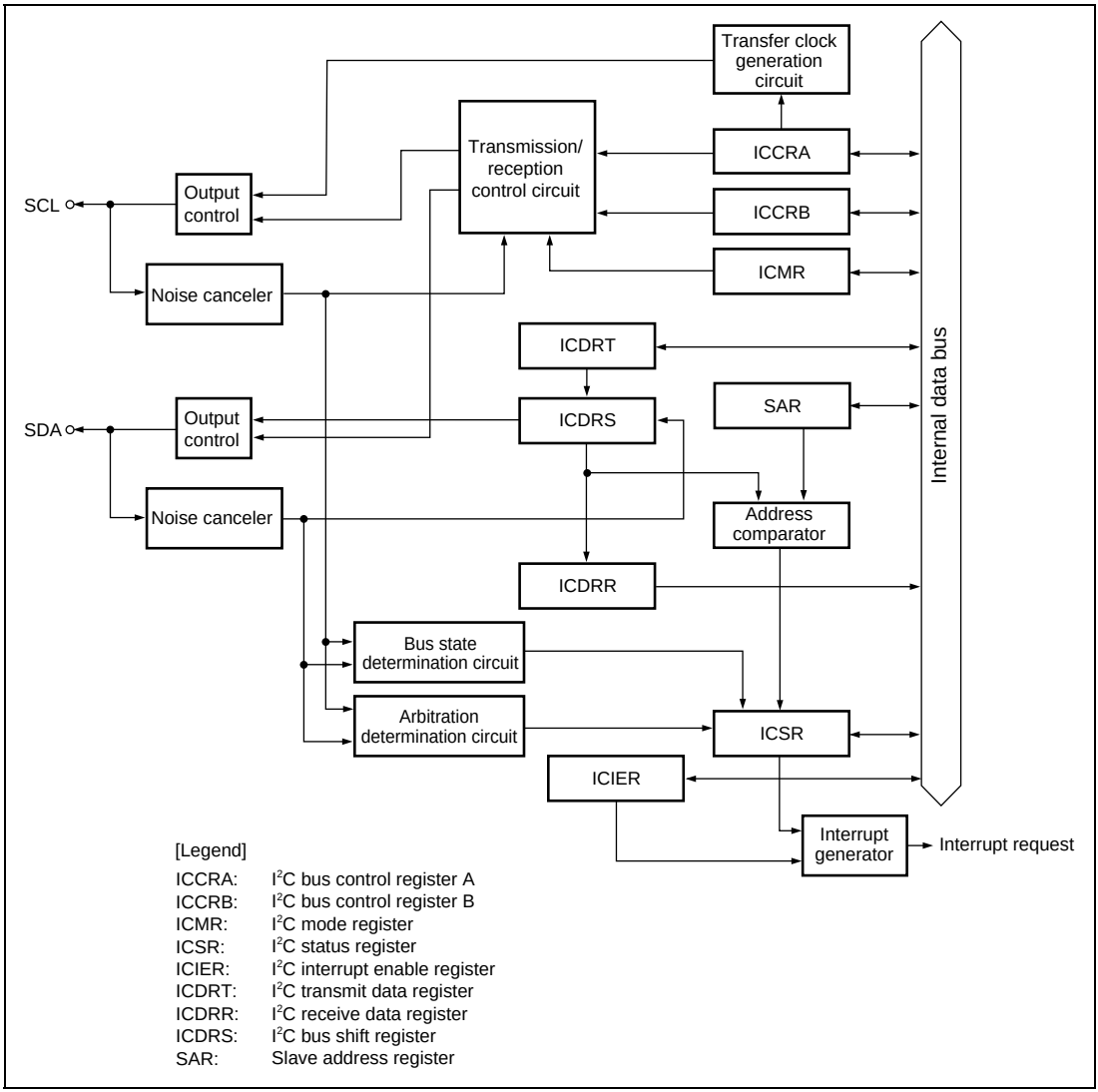


Figure 17.1 Block Diagram of I²C Bus Interface 3

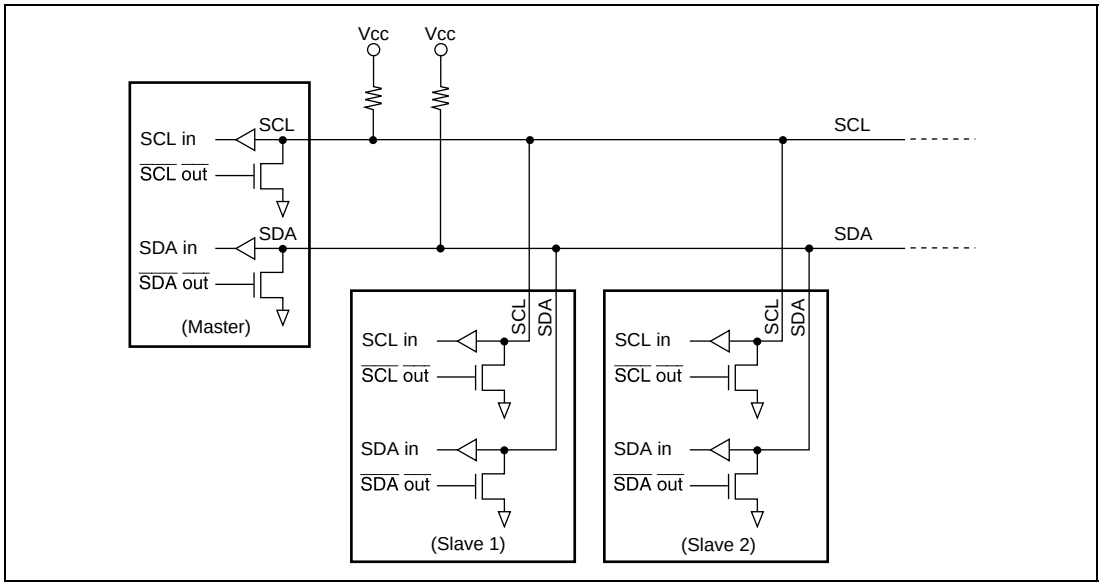


Figure 17.2 External Circuit Connections of I/O Pins

17.2 Input/Output Pins

Table 17.1 shows the pin configuration of the I²C bus interface 3.

Table 17.1 Pin Configuration

Name	Symbol	I/O	Function
Serial clock	SCL0	I/O	IIC3_0 serial clock input/output
Serial data	SDA0	I/O	IIC3_0 serial data input/output
Serial clock	SCL1	I/O	IIC3_1 serial clock input/output
Serial data	SDA1	I/O	IIC3_1 serial data input/output
Serial clock	SCL2	I/O	IIC3_2 serial clock input/output
Serial data	SDA2	I/O	IIC3_2 serial data input/output
Serial clock	SCL3	I/O	IIC3_3 serial clock input/output
Serial data	SDA3	I/O	IIC3_3 serial data input/output

Note: The pin symbols are represented as SCL and SDA; channel numbers are omitted in this manual.

17.3 Register Descriptions

The IIC3 has the following registers for each channel.

- I²C bus control register A (ICCRA)
- I²C bus control register B (ICCRB)
- I²C bus mode register (ICMR)
- I²C bus interrupt enable register (ICIER)
- I²C bus status register (ICSR)
- I²C bus status register A (ICSRA)
- Slave address register (SAR)
- Slave address register A (SARA)
- Slave address register B (SARB)
- Slave address mask register (SAMR)
- I²C bus transmit data register (ICDRT)
- I²C bus receive data register (ICDRR)
- I²C bus shift register (ICDRS)

17.3.1 I²C Bus Control Register A (ICCRA)

ICCRA enables or disables the I²C bus interface, controls transmission or reception, and selects master or slave mode, transmission or reception, and transfer clock frequency in master mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface Enable 0: This module is halted. 1: This module is enabled for transfer operations. (SCL and SDA pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable Enables or disables the next operation when TRS is 0 and ICDRR is read. 0: Enables next reception 1: Disables next reception
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select When arbitration is lost in master mode, MST and TRS are both reset by hardware, causing a transition to slave receive mode. Modification of the TRS bit should be made between transfer frames. Operating modes are described below according to MST and TRS combination. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode
3	CKS3	0	R/W	Transfer Clock Select 3 to 0 These bits are valid only in master mode and should be set according to the necessary transfer rate. For details on transfer rate, see table 17.2.
2	CKS2	0	R/W	
1	CKS1	0	R/W	
0	CKS0	0	R/W	

Table 17.2 Transfer Rate

Bit 3	Bit 2	Bit 1	Bit 0	Clock	Transfer Rate		
CKS3	CKS2	CKS1	CKS0		φ = 8 MHz	φ = 10 MHz	φ = 20 MHz
0	0	0	0	φ/28	286 kHz	357 kHz	714 kHz
			1	φ/40	200 kHz	250 kHz	500 kHz
		1	0	φ/48	167 kHz	208 kHz	417 kHz
			1	φ/64	125 kHz	156 kHz	313 kHz
	1	0	0	φ/168	47.6 kHz	59.5 kHz	119 kHz
			1	φ/100	80.0 kHz	100 kHz	200 kHz
		1	0	φ/112	71.4 kHz	89.3 kHz	179 kHz
			1	φ/128	62.5 kHz	78.1 kHz	156 kHz
1	0	0	0	φ/56	143 kHz	179 kHz	357 kHz
			1	φ/80	100 kHz	125 kHz	250 kHz
		1	0	φ/96	83.3 kHz	104 kHz	208 kHz
			1	φ/128	62.5 kHz	78.1 kHz	156 kHz
	1	0	0	φ/336	23.8 kHz	29.8 kHz	59.5 kHz
			1	φ/200	40.0 kHz	50.0 kHz	100 kHz
		1	0	φ/224	35.7 kHz	44.6 kHz	89.3 kHz
			1	φ/256	31.3 kHz	39.1 kHz	78.1 kHz

17.3.2 I²C Bus Control Register B (ICCRB)

ICCRB issues start/stop conditions, manipulates the SDA pin, monitors the SCL pin, and controls a reset in IIC control.

Bit	Bit Name	Initial Value	R/W	Description
7	BBSY	0	R/W	Bus Busy There are two functions: a flag function which indicates whether the I²C bus is occupied or released and a function which issues start and stop conditions in master mode. This bit is set to 1 when the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. Also follow this procedure when re-transmitting a start condition. Write 0 to BBSY and 0 to SCP to issue a stop condition. To issue a start/stop condition, use the MOV instruction.

Bit	Bit Name	Initial Value	R/W	Description
6	SCP	1	R/W	Start/Stop Condition Prohibit Controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 to BBSY and 0 to SCP. Also follow this procedure when retransmitting a start condition. To issue a stop condition, write 0 to BBSY and 0 to SCP. This bit is always read as 1. Even if 1 is written to this bit, the data is not stored.
5	SDAO	1	R	Monitors the SDA output level. When reading and the SDAO bit is 1, the SDA pin outputs high. When reading and the SDAO bit is 0, the SDA pin outputs low.
4	—	1	R/W	Reserved The write value should always be 1.
3	SCLO	1	R	Monitors the SCL output level. When reading and the SCLO bit is 1, the SCL pin outputs high. When reading and the SCLO bit is 0, the SCL pin outputs low.
2	—	1	—	Reserved This bit is always read as 1.
1	IICRST	0	R/W	IIC Control Part Reset Resets control parts except for I^2C registers. If this bit is set to 1 when a hang-up occurred because of communication failure during I^2C operation, I^2C control parts can be reset without setting ports and initializing registers.
0	—	1	—	Reserved This bit is always read as 1.

17.3.3 I²C Bus Mode Register (ICMR)

ICMR controls a wait in master mode and selects the transfer bit count.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The write value should always be 0.
6	WAIT	0	R/W	Wait Insertion Bit Selects whether to insert a wait after data transfer except for the acknowledge bit in master mode. When the WAIT bit is set to 1, after the fall of the clock for the last data bit, low period is extended for two transfer clocks. If the WAIT bit is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted. The setting of this bit is invalid in slave mode.
5	—	1	—	Reserved These bits are always read as 1.
4	—	1	—	
3	BCWP	1	R/W	BC Write Protect Controls the BC2 to BC0 modifications. When modifying the BC2 to BC0 bits, this bit should be cleared to 0 and use the MOV instruction. 0: When writing, values of BC2 to BC0 are set. 1: When reading, 1 is always read. When writing, settings of BC2 to BC0 are invalid.

Bit	Bit Name	Initial Value	R/W	Description
2	BC2	0	R/W	Bit Counter 2 to 0
1	BC1	0	R/W	Specify the number of bits to be transferred next. The data is transferred with one acknowledge bit added. 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 signal is low. The value automatically returns to B′000 at the end of a data transfer, including the acknowledge bit. 000: 9 bits 001: 2 bits 010: 3 bits 011: 4 bits 100: 5 bits 101: 6 bits 110: 7 bits 111: 8 bits
0	BC0	0	R/W	

17.3.4 I²C Bus Interrupt Enable Register (ICIER)

ICIER enables or disables interrupt sources and acknowledge bits, sets acknowledge bits to be transmitted, and confirms acknowledge bits to be received.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, this bit enables or disables the transmit data empty interrupt (TXI). 0: Transmit data empty interrupt request (TXI) is disabled. 1: Transmit data empty interrupt request (TXI) is enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. The TEI can be canceled by clearing the TEND bit or the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabled. 1: Transmit end interrupt request (TEI) is enabled.

Bit	Bit Name	Initial Value	R/W	Description
5	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive data full interrupt request (RXI) when receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. The RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) is disabled. 1: Receive data full interrupt request (RXI) is enabled.
4	NAKIE	0	R/W	NACK Receive Interrupt Enable Enables or disables the NACK receive interrupt request (NAKI) when the NACKF and AL bits in ICSR are set to 1. The NAKI can be canceled by clearing the NACKF, AL, or NAKIE bit to 0. 0: NACK receive interrupt request (NAKI) is disabled. 1: NACK receive interrupt request (NAKI) is enabled.
3	STIE	0	R/W	Stop Condition Detection Interrupt Enable 0: Stop condition detection interrupt request (STPI) is disabled. 1: Stop condition detection interrupt request (STPI) is enabled.
2	ACKE	0	R/W	Acknowledge Bit Determination Select 0: The value of the acknowledge bit is ignored, and continuous transfer is performed. 1: If the acknowledge bit is 1, continuous transfer is interrupted.
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the contents of the acknowledge bit that is returned by the receive device. This bit cannot be modified. 0: Receive acknowledge = 0 1: Receive acknowledge = 1
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, this bit specifies the bit to be sent at the acknowledge timing. 0: 0 is sent at the acknowledge timing. 1: 1 is sent at the acknowledge timing.

17.3.5 I²C Bus Status Register (ICSR)

ICSR confirms interrupt request flags and status.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	0	R/W	Transmit Data Empty [Setting condition] • When data is transferred from ICDRT to ICDRS and ICDRT becomes empty [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When data is written to ICDRT
6	TEND	0	R/W	Transmit end [Setting condition] • When the ninth clock of SCL rises while the TDRE flag is 1 [Clearing conditions] • When 0 is written to TEND after reading TEND = 1 • When data is written to ICDRT
5	RDRF	0	R/W	Receive Data Full [Setting condition] • When receive data is transferred from ICDRS to ICDRR [Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When data is read from ICDRR
4	NACKF	0	R/W	No Acknowledge Detection Flag [Setting condition] • When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1 [Clearing condition] • When 0 is written to NACKF after reading NACKF = 1

Bit	Bit Name	Initial Value	R/W	Description
3	STOP	0	R/W	Stop Condition Detection Flag [Setting condition] • When a stop condition is detected after frame transfer [Clearing condition] • When 0 is written to STOP after reading STOP = 1
2	AL	0	R/W	Arbitration Lost Flag Indicates that arbitration was lost in master mode. When two or more master devices attempt to seize the bus at nearly the same time, the I^2C bus interface monitors the SDA. If it detects data differing from the data it sent, it sets the AL flag to 1 to indicate that the bus has been taken by another master. [Setting conditions] • If the internal SDA and SDA pin does not match at the rise of SCL in master transmit mode • When the SDA pin goes high in master mode while a start condition is detected [Clearing condition] • When 0 is written to AL after reading AL = 1
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR. [Setting conditions] • When the slave address is detected in slave receive mode • When the general call address is detected in slave receive mode [Clearing condition] • When 0 is written to AAS after reading AAS = 1
0	ADZ	0	R/W	General Call Address Recognition Flag This bit is valid in slave receive mode. [Setting condition] • When the general call address is detected in slave receive mode [Clearing condition] • When 0 is written to ADZ after reading ADZ = 1

17.3.6 Slave Address Register (SAR)

SAR sets slave addresses. When the chip is in slave mode, if the upper 7 bits in SAR match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA6 to SVA0	All 0	R/W	Slave Address 6 to 0 Set a unique address in bits SVA6 to SVA0, differing from the addresses of other slave devices connected to the I^2C bus.
0	—	0	R/W	Reserved This bit is readable/writable. The write value should always be 0.

17.3.7 Slave Address Register A (SARA)

SARA sets slave addresses. When the chip is in slave mode, if the upper 7 bits in SARA match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA6 to SVA0	All 0	R/W	Slave Address 6 to 0 Set a unique address in bits SVA6 to SVA0, differing from the addresses of other slave devices connected to the I^2C bus.
0	SARE	0	R/W	Slave Address Enable Selects whether slave addresses in SARA are recognized or not. 0: Ignores slave addresses in SARA 1: Recognizes slave addresses in SARA

17.3.8 Slave Address Register B (SARB)

SARB sets slave addresses. When the chip is in slave mode, if the upper 7 bits in SARB match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA6 to SVA0	All 0	R/W	Slave Address 6 to 0 Set a unique address in bits SVA6 to SVA0, differing from the addresses of other slave devices connected to the I^2C bus.
0	SARE	0	R/W	Slave Address Enable Selects whether slave addresses in SARB are recognized or not. 0: Ignores slave addresses in SARB 1: Recognizes slave addresses in SARB

17.3.9 Slave Address Mask Register (SAMR)

SAMR masks slave addresses set in SAR and controls automatic switching of transmit modes in slave mode.

Bit	Bit Name	Initial Value	R/W	Description
7	MSA6	0	R/W	Slave Address Mask 6 to 0
6	MSA5	0	R/W	Correspond to the SVA6 to SVA0 bits in SAR and control comparison conditions for addresses set in SAR and addresses of upper 7 bits of the first frame received after a start condition in slave mode. 0: Compare addresses set in bits SVA6 to SVA0 in SAR and receive addresses 1: Operate assuming that receive addresses have matched bits SVA6 to SVA0 in SAR
5	MSA4	0	R/W	
4	MSA3	0	R/W	
3	MSA2	0	R/W	
2	MSA1	0	R/W	
1	MSA0	0	R/W	
0	MTRS	0	R/W	Transmit Mode Switch Mask Controls automatic switching of transmit modes by the eighth bit of the first frame in slave mode. 0: When the eighth bit of the first frame is 1, the TRS bit in ICCRA and TDRE bit in ICSR are automatically set to 1 and a transition is made to slave transmit mode. 1: The TRS bit in ICCRA and TDRE bit in ICSR are not automatically changed by the eighth bit of the first frame.

17.3.10 I²C Bus Status Register A (ICSRA)

ICSRA confirms slave address recognition flags.

Bit	Bit Name	Initial Value	R/W	Description
7	AASA	0	R/W	Slave Address Recognition Flag A In slave receive mode, this flag is set to 1 if the upper 7 bits in the first frame following a start condition match bits SVA6 to SVA0 in SARA. [Setting condition] • When the slave address is detected in slave receive mode [Clearing condition] • When 0 is written to AASA after reading AASA = 1
6	AASB	0	R/W	Slave Address Recognition Flag B In slave receive mode, this flag is set to 1 if the upper 7 bits in the first frame following a start condition match bits SVA6 to SVA0 in SARB. [Setting condition] • When the slave address is detected in slave receive mode [Clearing condition] • When 0 is written to AASB after reading AASB = 1
5 to 0	—	All 0	—	Reserved These bits are always read as 0.

17.3.11 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the I²C bus shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. If the next transfer data is written to ICDRT during transferring data in ICDRS, continuous transfer is possible.

17.3.12 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit register that stores the receive data. When one byte of data is received, ICDRR transfers the received data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register, therefore this register cannot be written to by the CPU.

17.3.13 I²C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred from ICDRS to ICDRR after one byte of data is received. This register cannot be read from the CPU.

17.4 Operation

17.4.1 I²C Bus Format

Figure 17.3 shows the I²C bus formats. Figure 17.4 shows the I²C bus timing. The first frame following a start condition always consists of 8 bits.

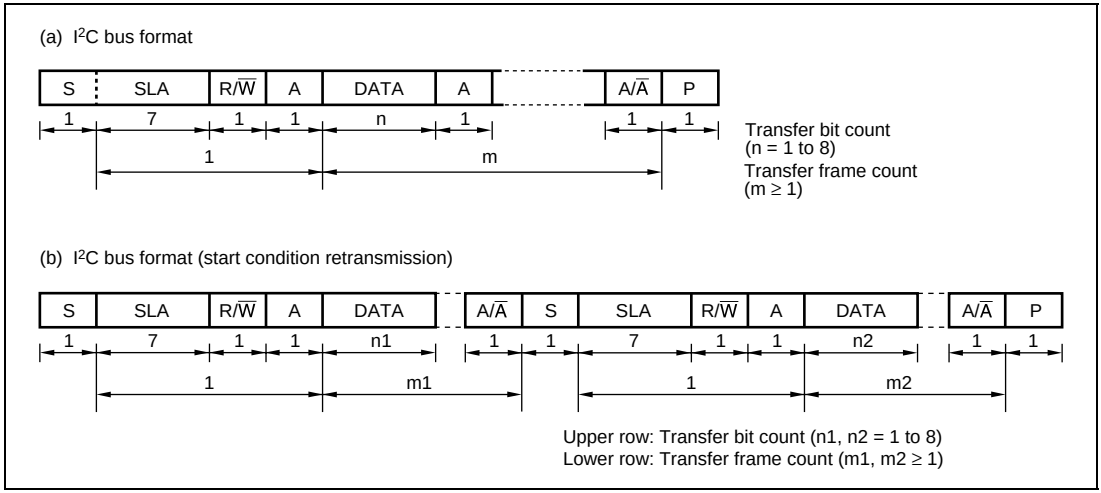


Figure 17.3 I²C Bus Formats

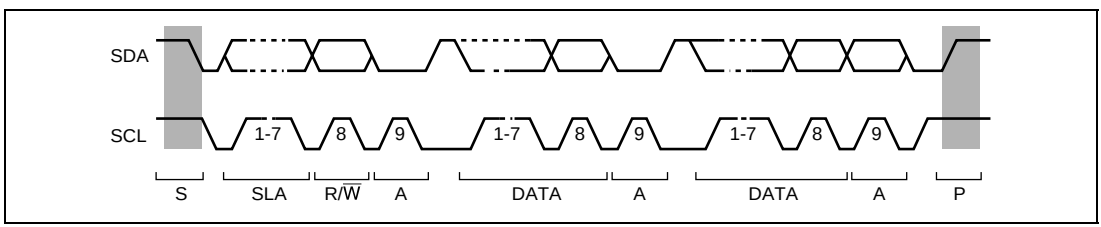


Figure 17.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

R/$\overline{W}$: Indicates the direction of data transfer: from the slave device to the master device when R/$\overline{W}$ is 1, or from the master device to the slave device when R/$\overline{W}$ is 0.

A: Acknowledge. The receive device drives SDA to low.

DATA: Transfer data

P: Stop condition. The master device drives SDA from low to high while SCL is high.

17.4.2 Master Transmit Operation

In master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. The operation timings in master transmit mode are shown in figures 17.5 and 17.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCRA to 1. Set the WAIT bit in ICMR and the CKS3 to CKS0 bits in ICCRA to 1 (initial setting).
2. Read the BBSY flag in ICCRB to confirm that the bus is free. Set the MST and TRS bits in ICCRA to select master transmit mode. Then, write 1 to BBSY and 0 to SCP using the MOV instruction. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first-byte data show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cleared to 0, then data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR is set to 1 at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confirm that the slave device has been selected. Then, write second byte data to ICDRT. When ACKBR is 1, the slave device has not been acknowledged, so issue the stop condition. To issue the stop condition, write 0 to BBSY and SCP using the MOV instruction. SCL is fixed low until the transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the end of last byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) from the receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TEND or NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to slave receive mode.

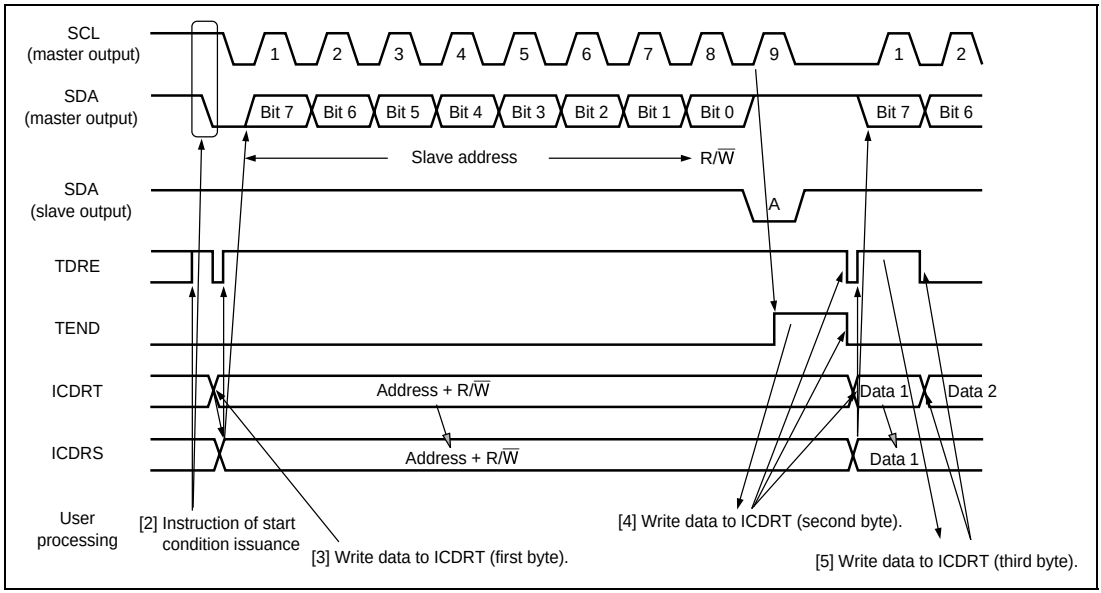


Figure 17.5 Operation Timing in Master Transmit Mode (1)

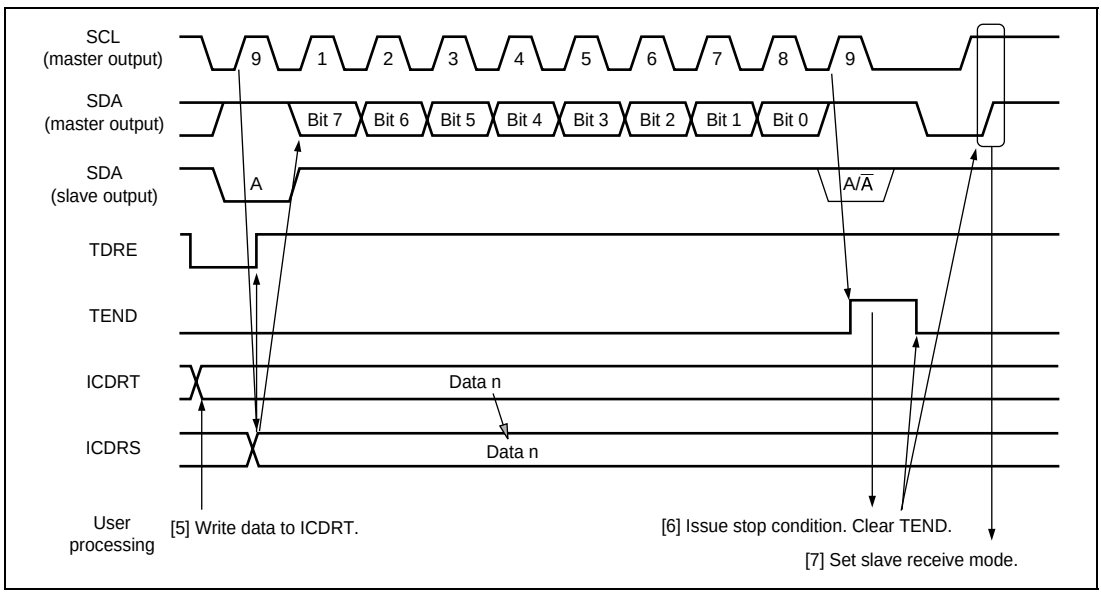


Figure 17.6 Operation Timing in Master Transmit Mode (2)

17.4.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data from the slave device, and returns an acknowledge signal. The operation timings in master receive mode are shown in figures 17.7 and 17.8. The reception procedure and operations in master receive mode are shown below.

1. Clear the TEND bit in ICSR to 0, then clear the TRS bit in ICCRA to 0 to switch from master transmit mode to master receive mode. Then, clear the TDRE bit to 0 and read ICDRR (dummy data read).
2. When ICDRR is read (dummy data read), reception is started, and the receive clock is output, and data is received, in synchronization with the internal clock. The master device outputs the level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.
3. After the reception of one frame data is completed, the RDRF bit in ICSR is set to 1 at the rise of 9th receive clock pulse. At this time, the received data can be read by reading ICDRR and at the same time the RDRF bit is cleared to 0.
4. The continuous reception is performed by reading ICDRR and clearing RDRF to 0 every time RDRF is set. If the 8th receive clock pulse falls after reading ICDRR by the other processing while RDRF is 1, SCL is fixed low until ICDRR is read.
5. If the next frame is the last receive data, set the RCVD bit in ICCRA to 1 before reading ICDRR. This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, issue the stop condition.
7. When the STOP bit in ICSR is set to 1, read ICDRR. Then clear the RCVD bit to 0.
8. The operation returns to slave receive mode.

Note: Operation described in step 1 should be executed continuously.

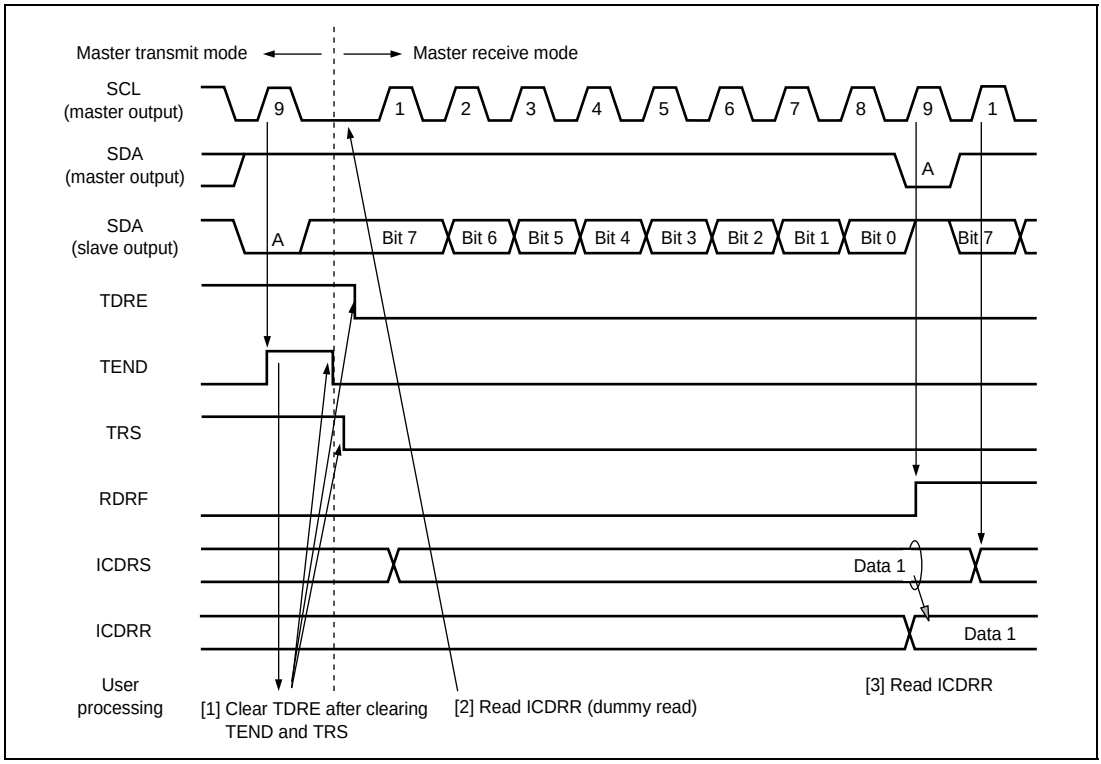


Figure 17.7 Operation Timing in Master Receive Mode (1)

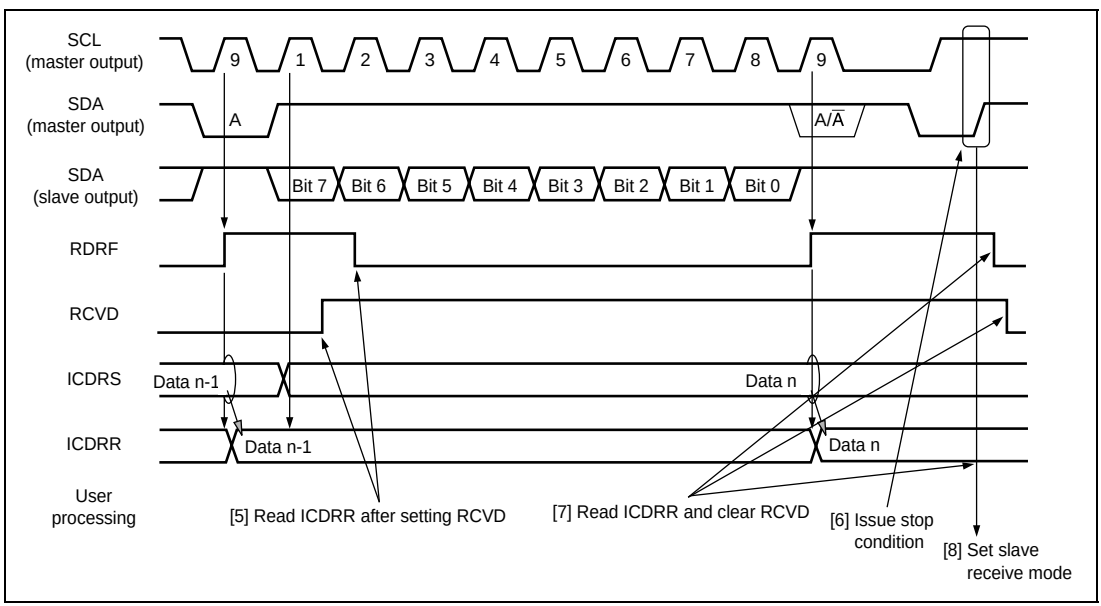


Figure 17.8 Operation Timing in Master Receive Mode (2)

17.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. The operation timings in slave transmit mode are shown in figures 17.9 and 17.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCRA to 1. Set the WAIT bit in ICMR and the CKS3 to CKS0 bits in ICCRA to 1 (initial setting). Set the MST and TRS bits in ICCRA to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At this time, if the 8th bit data (R/$\overline{W}$) is 1, the TRS in ICCRA and TDRE in ICSR are set to 1, and the mode changes to slave transmit mode automatically. The continuous transmission can be performed by writing transmit data to ICDRT every time TDRE is set.
3. If TDRE is set after writing the last transmit data to ICDRT, wait until TEND in ICSR is set to 1, with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for the end processing, and read ICDRR (dummy read). Then, SCL is free.
5. Clear TDRE.

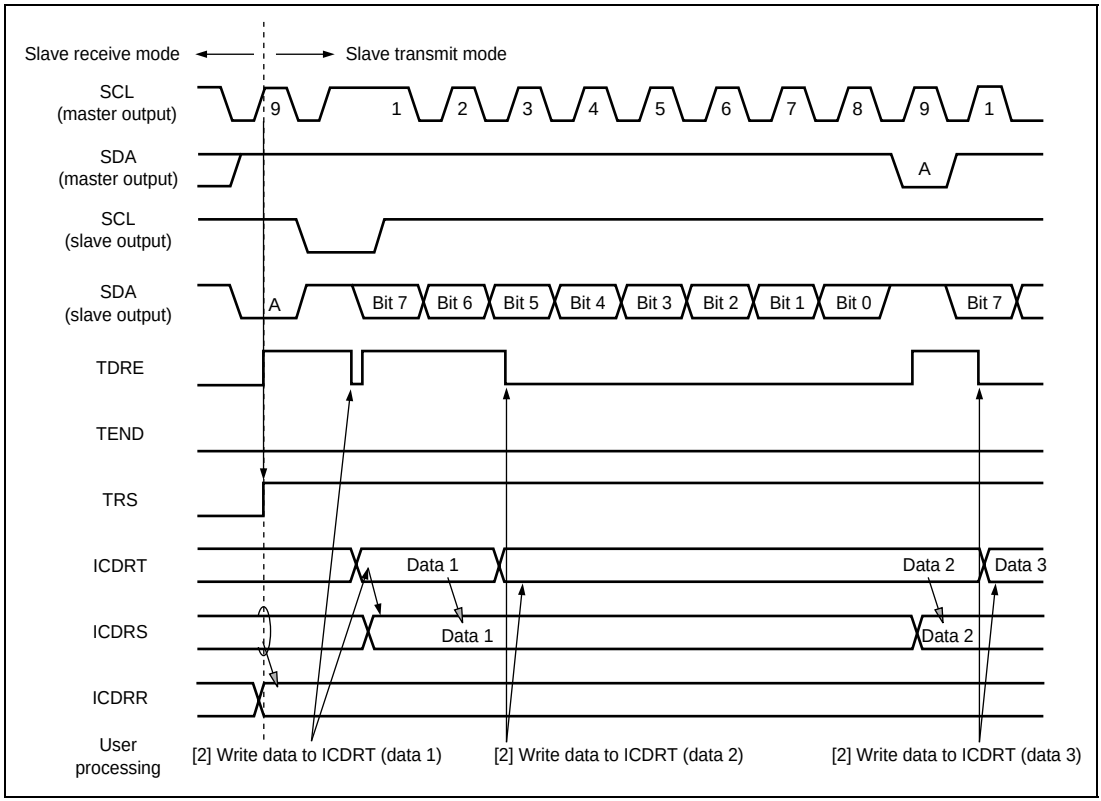


Figure 17.9 Operation Timing in Slave Transmit Mode (1)

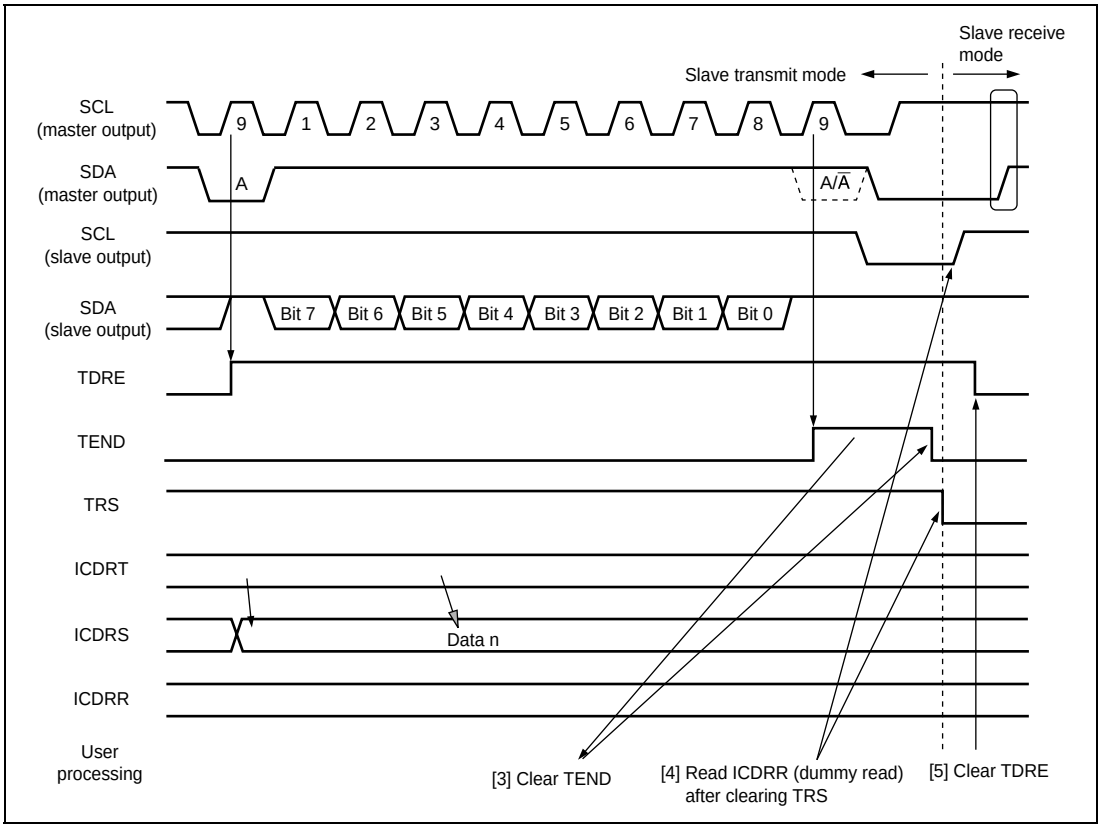


Figure 17.10 Operation Timing in Slave Transmit Mode (2)

17.4.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. The operation timings in slave receive mode are shown in figures 17.11 and 17.12. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCRA to 1. Set the WAIT bit in ICMR and the CKS3 to CKS0 bits in ICCRA to 1 (initial setting). Set the MST and TRS bits in ICCRA to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (Since the read data show the slave address and R/$\overline{W}$, it is not used.)
3. Read ICDRR every time RDRF is set. If the 8th receive clock pulse falls while RDRF is 1, SCL is fixed low until RDRF is cleared. The change of the acknowledge before clearing RDRF, to be returned to the master device, is reflected to the next transfer frame.

4. The last-byte data is read by reading ICDRR.

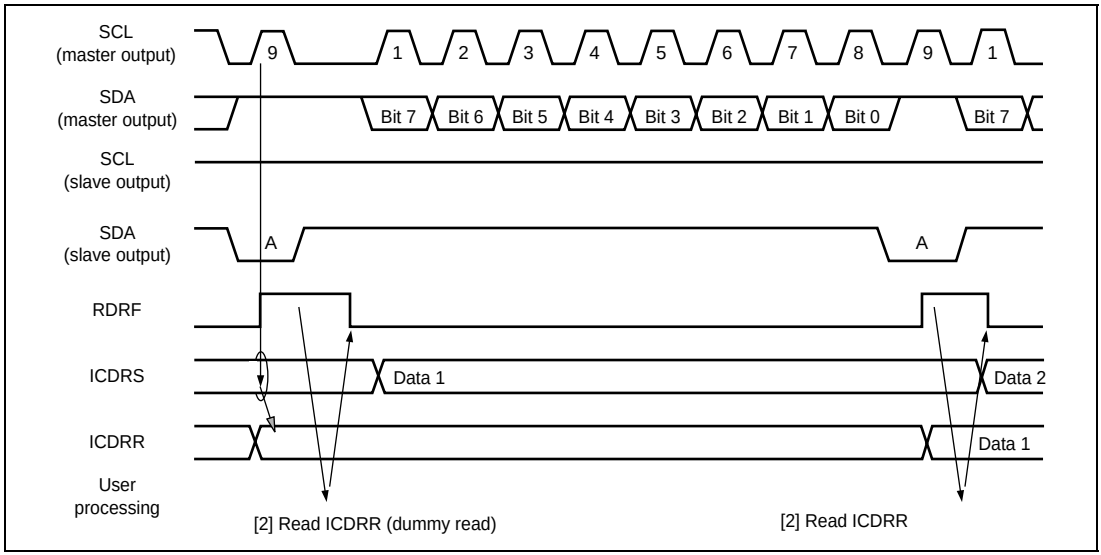


Figure 17.11 Operation Timing in Slave Receive Mode (1)

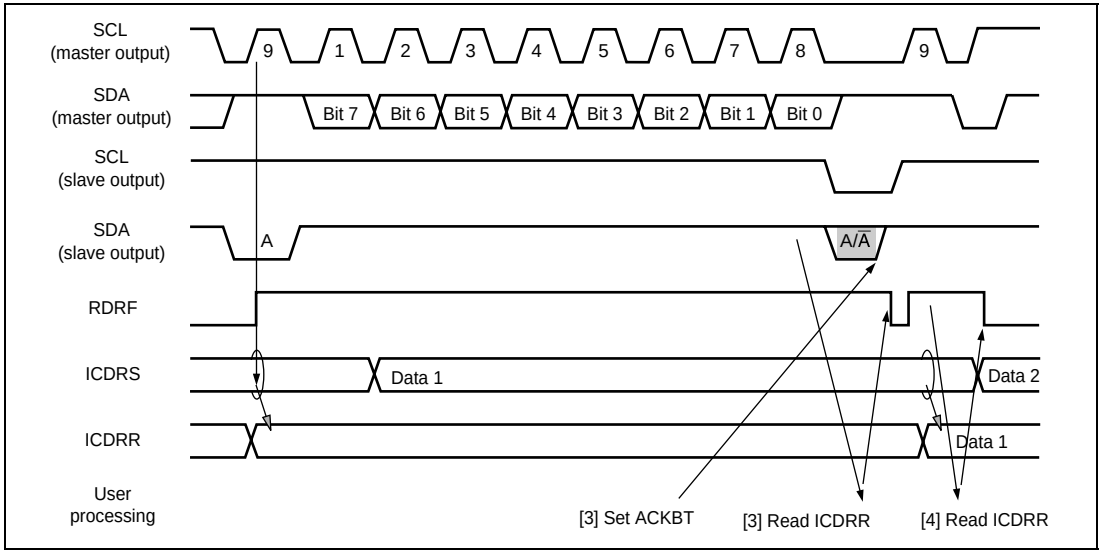


Figure 17.12 Operation Timing in Slave Receive Mode (2)

17.4.6 Noise Canceler

The logic levels at the SCL and SDA pins are latched internally via the noise canceler. Figure 17.13 shows a block diagram of the noise canceler.

The noise canceler consists of two cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the system clock, but is not passed forward to the next circuit unless the outputs of both latches match. If they do not match, the previous value is retained.

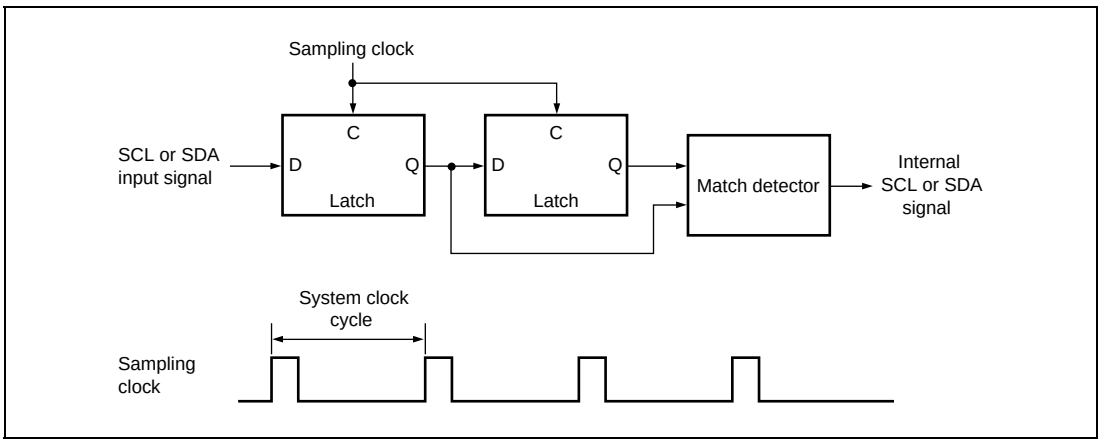


Figure 17.13 Block Diagram of Noise Canceler

17.4.7 Example of Use

Flowcharts in respective modes that use the I^2C bus interface are shown in figures 17.14 to 17.17.

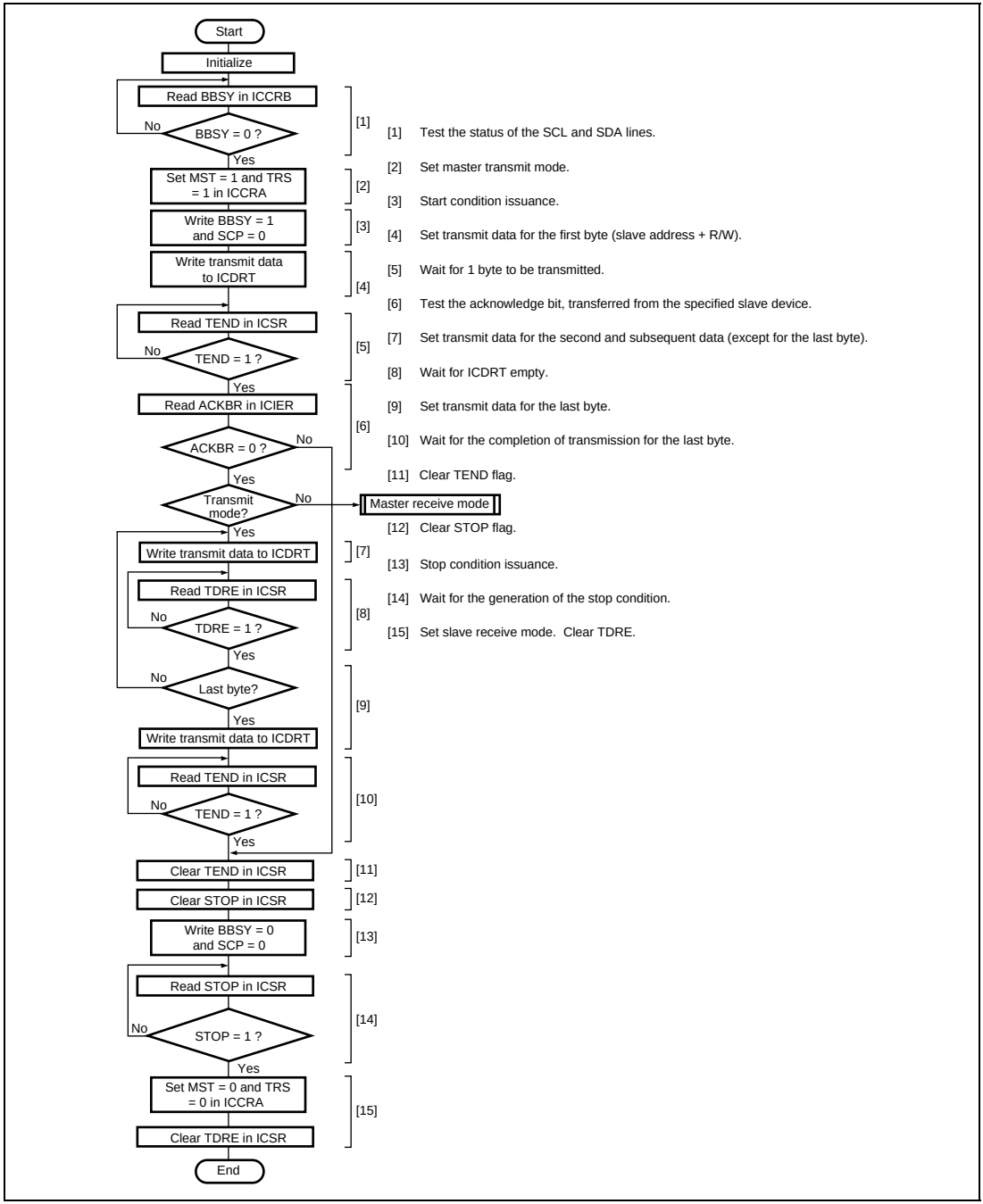
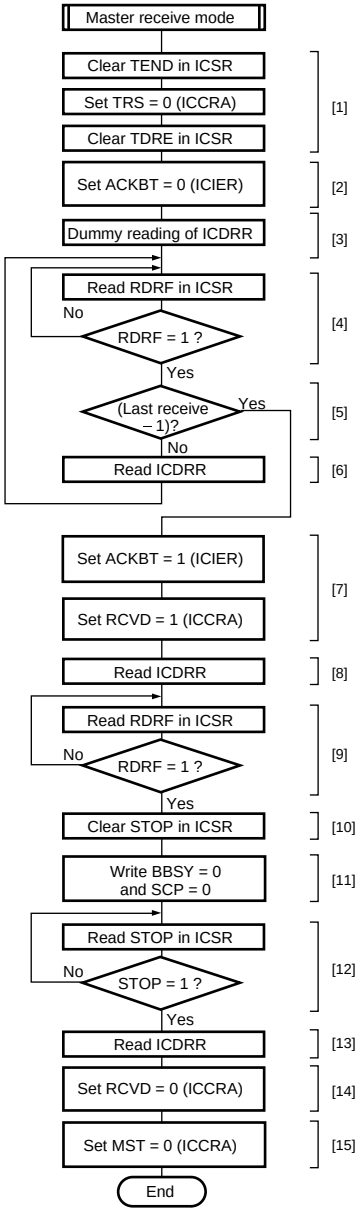


Figure 17.14 Sample Flowchart for Master Transmit Mode



[1] Clear TEND, set master receive mode, and then clear TDRE.*

[2] Set acknowledge to the transmit device.*

[3] Dummy reading of ICDDR*

[4] Wait for 1 byte to be received.

[5] Check if (last receive – 1).

[6] Read the receive data.

[7] Set acknowledge of the last byte. Disable continuous reception (RCVD = 1).

[8] Read receive data of (last byte – 1).

[9] Wait for the last byte to be received.

[10] Clear STOP flag.

[11] Stop condition issuance

[12] Wait for the generation of stop condition.

[13] Read the receive data of the last byte.

[14] Clear RCVD to 0.

[15] Set slave receive mode.

Note: * Steps 1 to 3 should be executed continously.

Figure 17.15 Sample Flowchart for Master Receive Mode

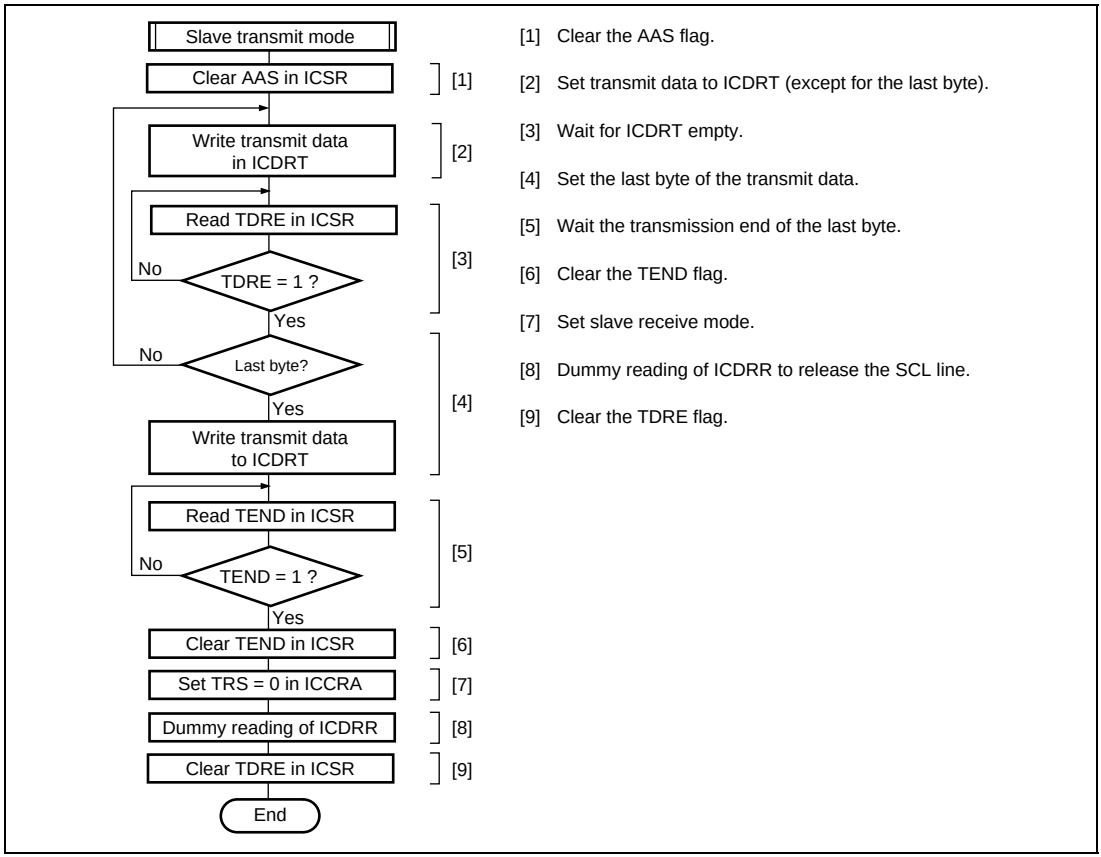


Figure 17.16 Sample Flowchart for Slave Transmit Mode

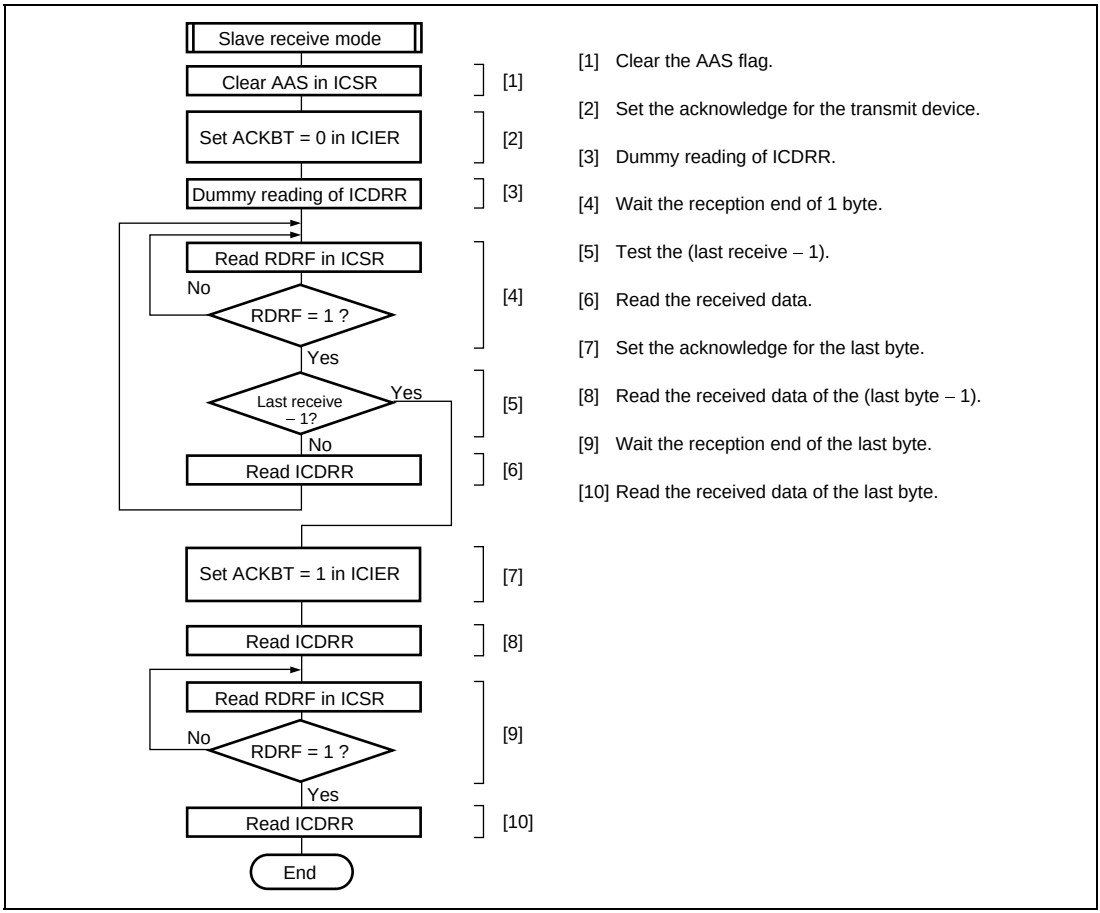


Figure 17.17 Sample Flowchart for Slave Receive Mode

17.5 Interrupt Requests

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK detection, STOP recognition, and arbitration lost. Table 17.3 shows the contents of each interrupt request.

Table 17.3 Interrupt Requests

Interrupt Request	Abbreviation	Interrupt Condition
Transmit data empty	TXI	(TDRE = 1) • (TIE = 1)
Transmit end	TEI	(TEND = 1) • (TEIE = 1)
Receive data full	RXI	(RDRF = 1) • (RIE = 1)
STOP recognition	STPI	(STOP = 1) • (STIE = 1)
NACK detection	NAKI	{(NACKF = 1) + (AL = 1)} • (NAKIE = 1)
Arbitration lost		

17.6 Bit Synchronous Circuit

In master mode,

- When the SCL is driven low by the slave device
- When the rising speed of the SCL is lower by the load of the SCL line (load capacitance or pull-up resistance)

This module has a possibility that the high level period may be short in the two states described above. Therefore it monitors the SCL and communicates by bits with synchronization.

Figure 17.18 shows the timing of the bit synchronous circuit and table 17.4 shows the time when SCL output changes from low to Hi-Z then the SCL is monitored.

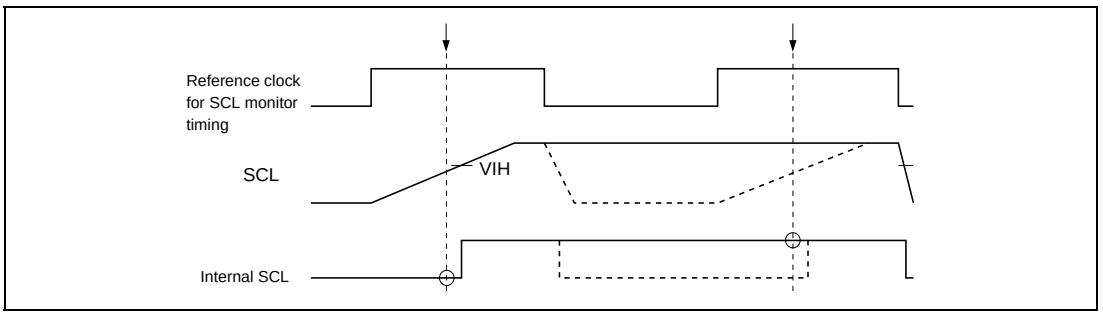


Figure 17.18 Timing of Bit Synchronous Circuit

Table 17.4 Time for Monitoring SCL

CKS3	CKS2	Time for Monitoring SCL
0	0	7.5 tcyc
	1	19.5 tcyc
1	0	17.5 tcyc
	1	41.5 tcyc

Section 18 A/D Converter

This LSI includes a successive-approximation-type 10-bit A/D converter that allows up to sixteen analog input channels to be selected. Figure 18.1 shows a block diagram of the A/D converter.

18.1 Features

- 10-bit resolution
- Sixteen input channels
- Conversion time: 8.38 µs per channel (min.)
- Two kinds of operating modes
  Single mode: Single-channel A/D conversion
  Scan mode: Continuous A/D conversion on 1 to 4 channels or 1 to 8 channels
- Eight data registers
  Conversion results are retained in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of conversion start
  Conversion can be started by software, conversion start trigger by 16-bit timer pulse unit (TPU) or 8-bit timer (TMR), or external trigger signal.
- Interrupt source
  A/D conversion end interrupt (ADI) request can be generated
- Module stop mode can be set

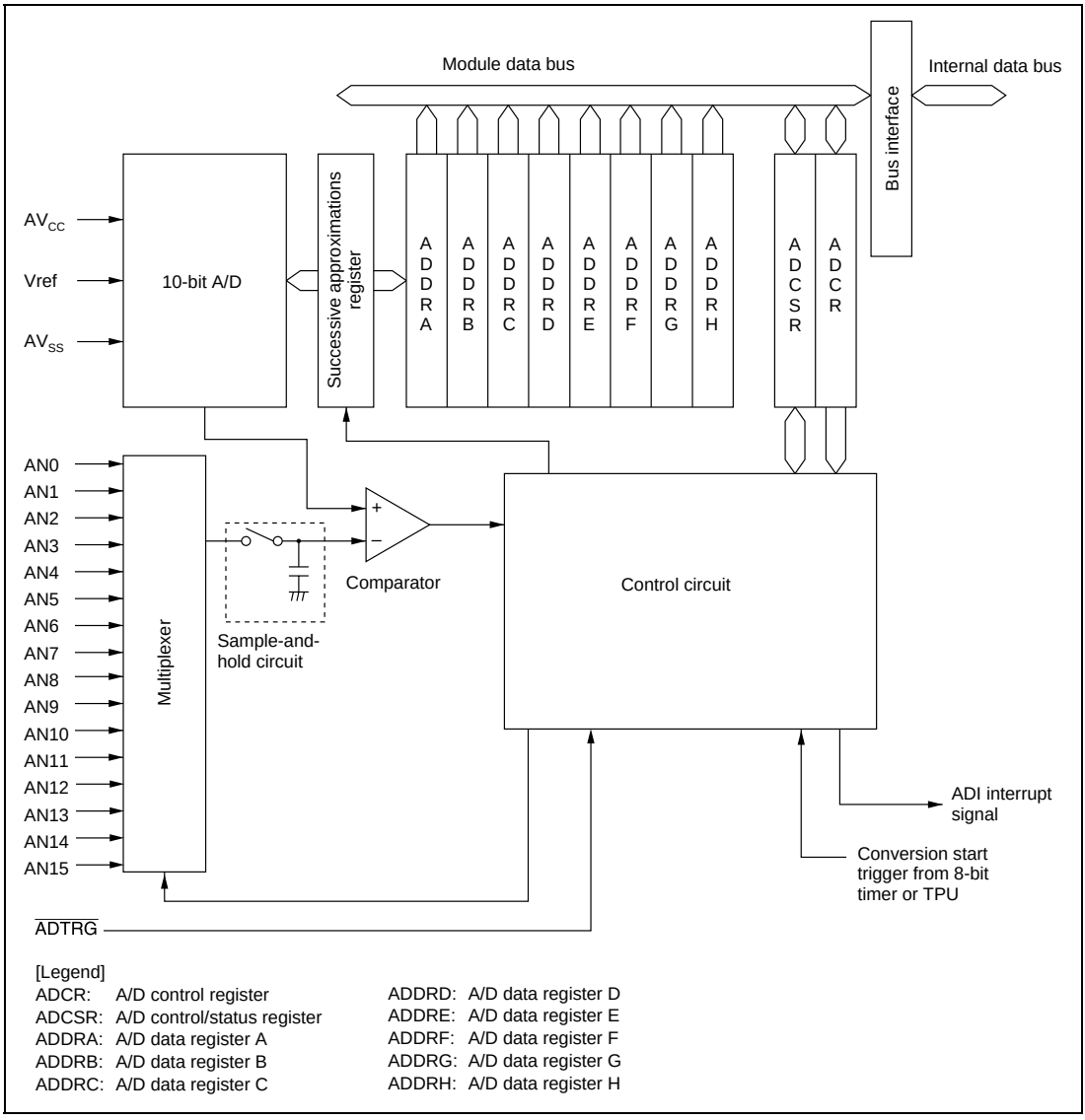


Figure 18.1 Block Diagram of A/D Converter

18.2 Input/Output Pins

Table 18.1 shows the pin configuration of the A/D converter.

The AV_{CC} and AV_{SS} pins are the power supply pins for the analog block in the A/D converter. The Vref pin is the reference voltage pin for the A/D conversion.

The sixteen analog input pins are divided into two channel sets: channel set 0 (AN0 to AN7) and channel set 1 (AN8 to AN15).

Table 18.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog block power supply
Analog ground pin	AV_{SS}	Input	Analog block ground
Reference voltage pin	Vref	Input	A/D conversion reference voltage
Analog input pin 0	AN0	Input	Analog inputs for channel set 0
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 8	AN8	Input	Analog inputs for channel set 1
Analog input pin 9	AN9	Input	
Analog input pin 10	AN10	Input	
Analog input pin 11	AN11	Input	
Analog input pin 12	AN12	Input	
Analog input pin 13	AN13	Input	
Analog input pin 14	AN14	Input	
Analog input pin 15	AN15	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

18.3 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 data register E (ADDRE)
- A/D data register F (ADDRF)
- A/D data register G (ADDRG)
- A/D data register H (ADDRH)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

18.3.1 A/D Data Registers A to H (ADDRA to ADDRH)

There are eight 16-bit read-only ADDR registers, ADDRA to ADDRH, used to store the results of A/D conversion. ADDR, which store a conversion result for each channel, are shown in table 18.2.

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

The data bus between the CPU and the A/D converter is 16-bit width. ADDR can be read directly from the CPU.

Table 18.2 Analog Input Channels and Corresponding ADDR

Analog Input Channel		A/D Data Register which Stores Conversion Result
Channel Set 0 (CH3 = 0)	**Channel Set 1 (CH3 = 1)**	
AN0	AN8	ADDRA
AN1	AN9	ADDRB
AN2	AN10	ADDRC
AN3	AN11	ADDRD
AN4	AN12	ADDRE
AN5	AN13	ADDRF
AN6	AN14	ADDRG
AN7	AN15	ADDRH

18.3.2 A/D Control/Status Register (ADCSR)

ADCSR controls A/D conversion operations.

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 specified channels in scan mode [Clearing condition] • When 0 is written after reading ADF = 1
6	ADIE	0	R/W	A/D Interrupt Enable Enables an ADI interrupt by the ADF bit when this bit is set to 1.
5	ADST	0	R/W	A/D Start Clearing this bit to 0 stops A/D conversion, and the A/D converter enters the wait state. When this bit is set to 1 by software, conversion start trigger by the TPU or TMR, or $\overline{\text{ADTRG}}$ pin, A/D conversion starts. This bit remains set to 1 during 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 a reset, a transition to hardware standby mode, or software.
4	—	0	—	Reserved This bit is always read as 0 and cannot be modified.

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

Bit	Bit Name	Initial Value	R/W	Description
3	CH3	0	R/W	Channel Select 3 to 0
2	CH2	0	R/W	Select analog input together with the SCANE and SCANS bits in ADCRS.
1	CH1	0	R/W	
0	CH0	0	R/W	Set the input channel when conversion is stopped (ADST = 0).
				When SCANE = 0 and SCANS = x
				0000: AN0 1000: AN8
				0001: AN1 1001: AN9
				0010: AN2 1010: AN10
				0011: AN3 1011: AN11
				0100: AN4 1100: AN12
				0101: AN5 1101: AN13
				0110: AN6 1110: AN14
				0111: AN7 1111: AN15
				When SCANE = 1 and SCANS = 0
				0000: AN0 1000: AN8
				0001: AN0 and AN1 1001: AN8 and AN9
				0010: AN0 to AN2 1010: AN8 to AN10
				0011: AN0 to AN3 1011: AN8 to AN11
				0100: AN4 1100: AN12
				0101: AN4 and AN5 1101: AN12 and AN13
				0110: AN4 to AN6 1110: AN12 to AN14
				0111: AN4 to AN7 1111: AN12 to AN15
				When SCANE = 1 and SCANS = 1
				0000: AN0 1000: AN8
				0001: AN0 and AN1 1001: AN8 and AN9
				0010: AN0 to AN2 1010: AN8 to AN10
				0011: AN0 to AN3 1011: AN8 to AN11
				0100: AN0 to AN4 1100: AN8 to AN12
				0101: AN0 to AN5 1101: AN8 to AN13
				0110: AN0 to AN6 1110: AN8 to AN14
				0111: AN0 to AN7 1111: AN8 to AN15

[Legend] x: Don’t care.

18.3.3 A/D Control Register (ADCR)

ADCR enables an A/D conversion start by an external trigger input.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 1 and 0
6	TRGS0	0	R/W	Select enabling or disabling of the start of A/D conversion by a trigger signal. 00: A/D conversion start by external trigger is disabled 01: A/D conversion start by conversion start trigger (TPU) is enabled 10: A/D conversion start by conversion start trigger (TMR) is enabled 11: A/D conversion start by $\overline{\text{ADTRG}}$ pin is enabled
5	SCANE	0	R/W	Scan Mode
4	SCANS	0	R/W	Select single mode or scan mode as the A/D conversion operating mode. 0x: Single mode 10: Scan mode. A/D conversion is performed continuously for channels 1 to 4 11: Scan mode. A/D conversion is performed continuously for channels 1 to 8.
3	CKS1	0	R/W	Clock Select 1 and 0
2	CKS0	0	R/W	Set the A/D conversion time. Only set bits CKS1 and CKS0 while conversion is stopped (ADST = 0). 00: A/D conversion time = 530 states (max.) 01: A/D conversion time = 266 states (max.) 10: A/D conversion time = 134 states (max.) 11: A/D conversion time = 68 states (max.)
1	—	0	R/W	Reserved
0	—	0	R/W	These bits are always read as 0 and cannot be modified.

[Legend] x: Don’t care.

18.4 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 as the operating mode or analog input channel is changed.

18.4.1 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 is started when the ADST bit in ADCSR is set to 1, according to software or external trigger input.
2. When A/D conversion is completed, the result is transferred to the corresponding A/D data register to the channel.
3. On completion of 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, and is automatically cleared to 0 when conversion ends. When the ADST bit is cleared to 0 during A/D conversion, A/D conversion stops and the A/D converter enters the wait state.

18.4.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the specified channels: maximum four channels or maximum eight channels. Operations are as follows.

1. When the ADST bit in ADCSR is set to 1 by software, TPU, or external trigger input, A/D conversion starts on the first channel in the specified channel set.

   The consecutive A/D conversion on maximum four channels (SCANE = 1 and SCANS = 0) or on maximum eight channels (SCANE = 1 and SCANS = 1) can be selected. When the consecutive A/D conversion is performed on the four channels, the A/D conversion starts on AN0 when CH3 = 0 and CH2 = 0, AN4 when CH3 = 0 and CH2 = 1, AN8 when CH3 = 1 and CH2 = 0, or AN12 when CH3 = 1 and CH2 = 1. When the consecutive A/D conversion is performed on the eight channels, the A/D conversion starts on AN0 when CH3 = 0 and CH2 = 0 and on AN8 when CH3 = 1 and CH2 = 0.
2. When A/D conversion for each channel is completed, the result is sequentially transferred to the corresponding A/D data register to each channel.
3. When conversion for 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 request is generated. Conversion for the first channel in the channel set starts again.

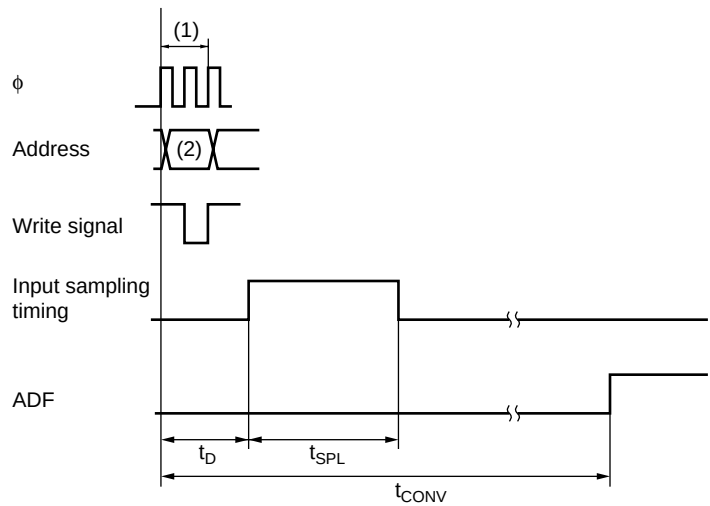
4. The ADST bit is not cleared automatically, and steps 2 and 3 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops and the A/D converter enters the wait state. Then if the ADST bit is set to 1, A/D conversion starts again for the first channel in the channel set.

18.4.3 Input Sampling and A/D Conversion Time

The A/D converter has an on-chip sample-and-hold circuit. The A/D converter samples the analog input when A/D conversion start delay time (t_D) passes after the ADST bit in ADCSR is set to 1, then starts conversion. Figure 18.2 shows the A/D conversion timing. Table 18.3 shows the A/D conversion time.

As shown in figure 18.2, 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 the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in tables 18.3.

In scan mode, the values given in tables 18.3 apply to the first conversion time. The values given in table 18.4 apply to the second and subsequent conversions.



[Legend]
(1): ADCSR write cycle
(2): ADCSR address
t_D: A/D conversion start delay time
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 18.2 A/D Conversion Timing

Table 18.3 A/D Conversion Time (Single Mode)

		CKS1 = 0						CKS1 = 1					
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS0 = 1		
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**
A/D conversion start delay time	t_D	18	—	33	10	—	17	6	—	9	4	—	5
Input sampling time	t_{SPL}	—	127	—	—	63	—	—	31	—	—	15	—
A/D conversion time	t_{CONV}	515	—	530	259	—	266	131	—	134	67	—	68

Note: Values in the table are the number of states.

Table 18.4 A/D Conversion Time (Scan Mode)

CKS1	CKS0	Conversion Time (State)
0	0	512 (Fixed)
	1	256 (Fixed)
1	0	128 (Fixed)
	1	64 (Fixed)

18.4.4 External Trigger Input Timing

A/D conversion can be started by an external trigger input. When the TRGS1 and TRGS0 bits in ADCR are set to 11, an external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{ADTRG}}$ pin sets the ADST bit in ADCSR to 1, 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 18.3 shows the timing.

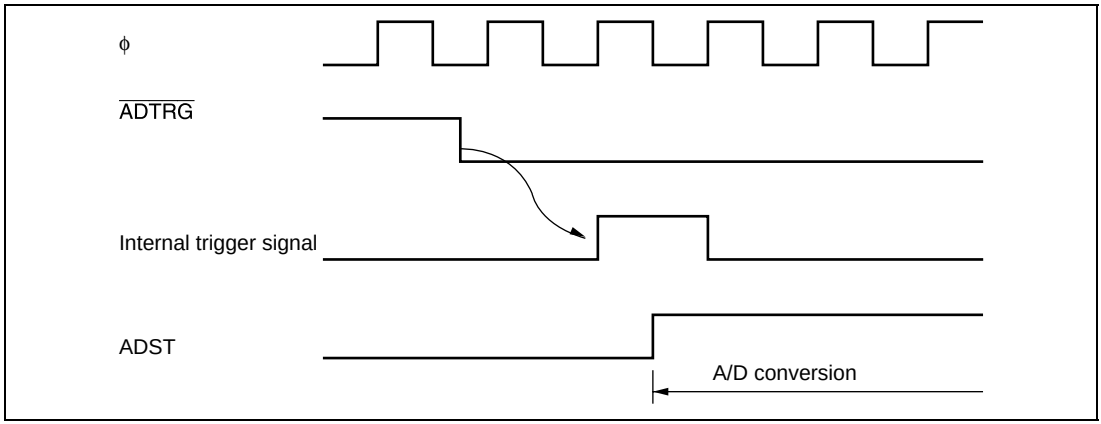


Figure 18.3 External Trigger Input Timing

18.5 Interrupt Source

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. Setting the ADIE bit to 1 enables an ADI interrupt request while the ADF bit in ADCSR is set to 1 after A/D conversion is completed.

Table 18.5 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Flag
ADI	End of A/D conversion	ADF

18.6 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 18.4).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value B'0000000000 (H'000) to B'0000000001 (H'001) (see figure 18.5).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from B'1111111110 (H'3FE) to B'1111111111 (H'3FF) (see figure 18.5).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error (see figure 18.5).
- 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.

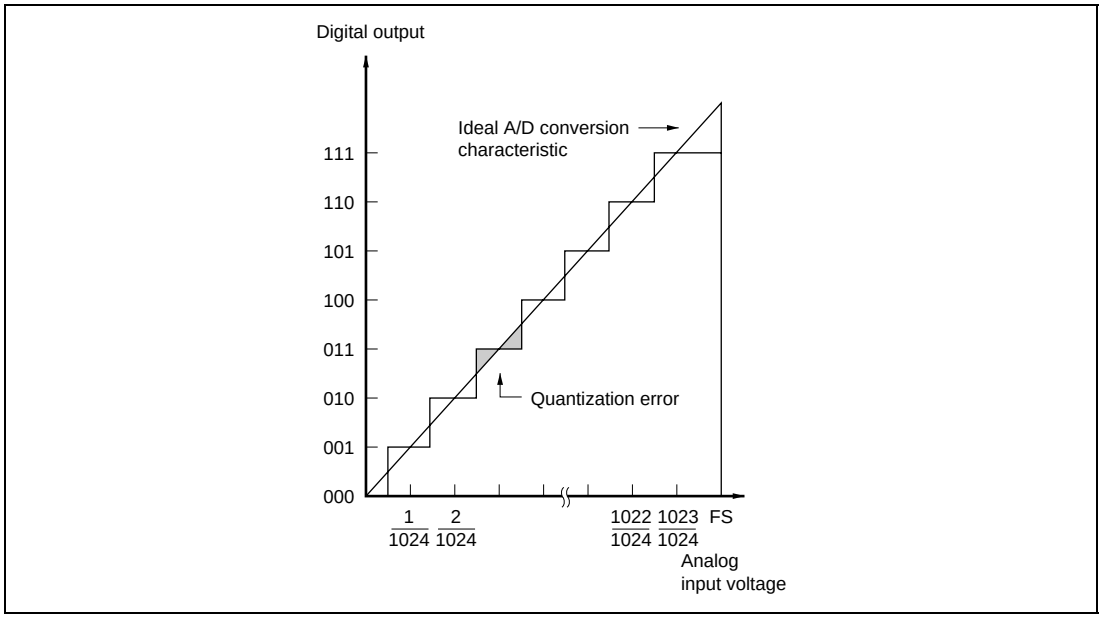


Figure 18.4 A/D Conversion Accuracy Definitions

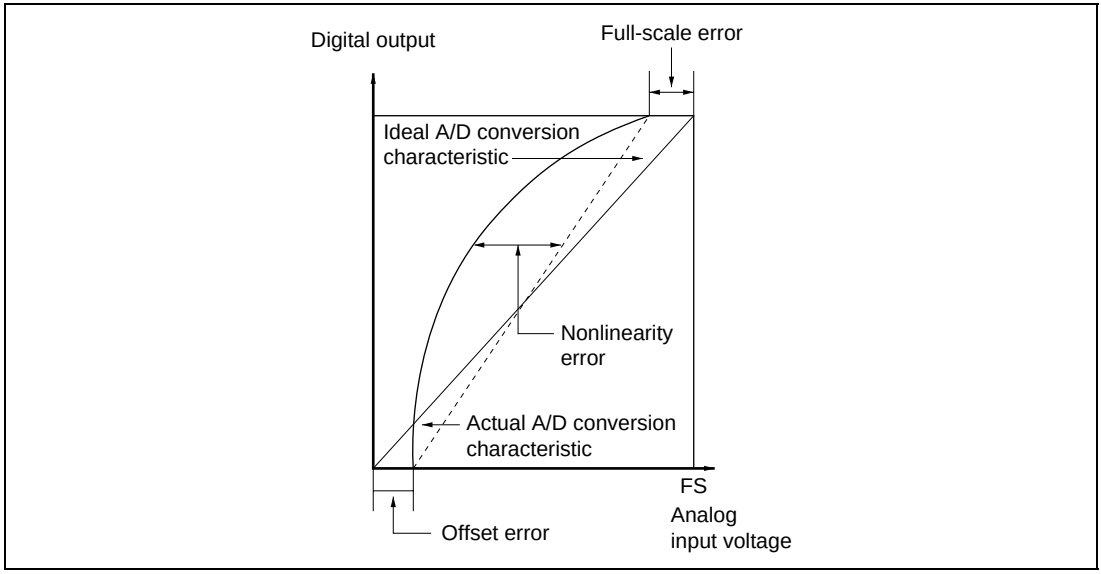


Figure 18.5 A/D Conversion Accuracy Definitions

18.7 Usage Notes

18.7.1 Module Stop Mode Setting

Operation of the A/D converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the A/D converter to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

18.7.2 Permissible Signal Source Impedance

This LSI's analog input is designed so that 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 input capacitance of the A/D converter's sample-and-hold circuit 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 for conversion 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., 5 mV/μs or greater) (see figure 18.6). When converting a high-speed analog signal or performing conversion in scan mode, a low-impedance buffer should be inserted.

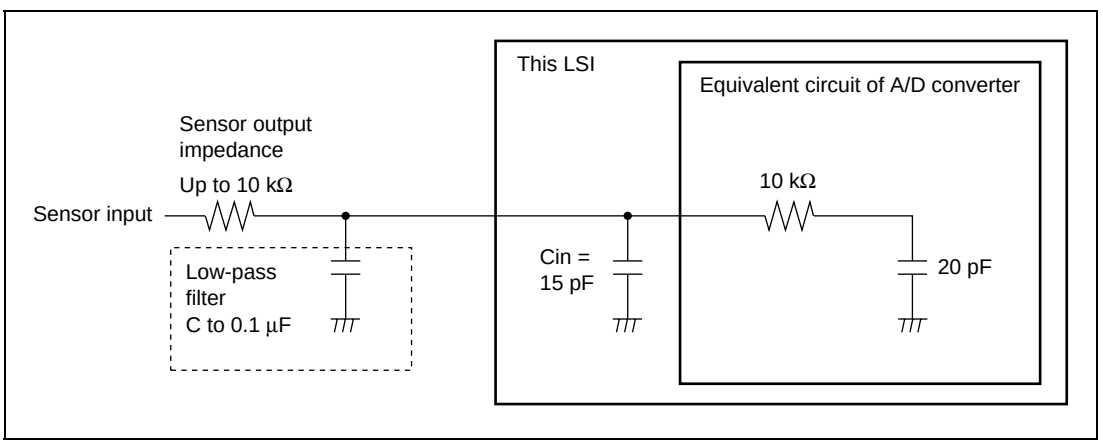


Figure 18.6 Example of Analog Input Circuit

18.7.3 Influences on Absolute Accuracy

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

18.7.4 Setting Range of Analog Power Supply and Other Pins

If conditions shown below are not met, the reliability of the 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 ≤ Vref.
- Relation between AVcc, AVss and Vcc, Vss
  For the relationship between AVcc, AVss and Vcc, Vss, set AVcc ≥ Vcc and AVss = Vss. If the A/D converter is not used, the AVcc and AVss pins must not be open.
- Vref setting range
  The reference voltage at the Vref pin should be set in the range Vref ≤ AVcc.

18.7.5 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 AN15), analog reference power supply (Vref), 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 digital ground (Vss) on the board.

18.7.6 Notes on Noise Countermeasures

A protection circuit connected to prevent damage due to an abnormal voltage such as an excessive surge at the analog input pins (AN0 to AN15) should be connected between AVcc and AVss as shown in figure 18.7. Also, the bypass capacitors connected to AVcc and the filter capacitor connected to AN0 to AN15 must be connected to AVss.

If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN15) 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.

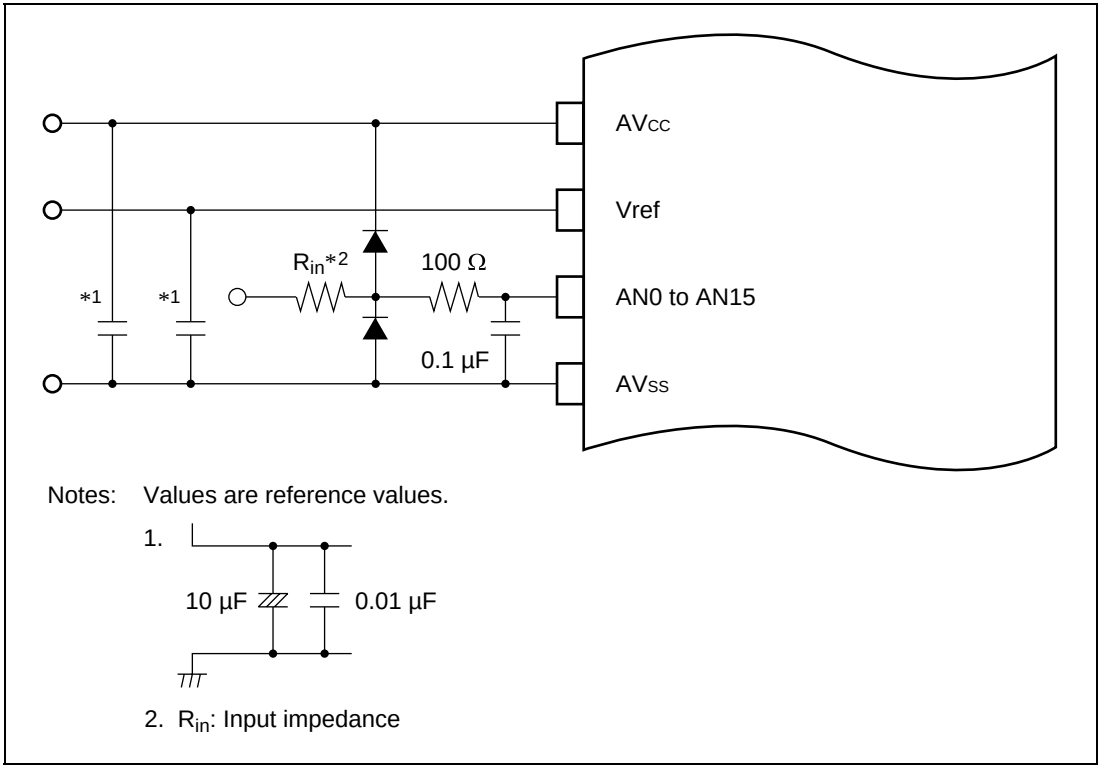


Figure 18.7 Example of Analog Input Protection Circuit

Table 18.6 Analog Pin Specifications

Item	Min.	Max.	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	5	kΩ

Section 19 RAM

This LSI has 16 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 to 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 20 Flash Memory (0.18-µm F-ZTAT Version)

The flash memory has the following features. Figure 20.1 shows a block diagram of the flash memory.

20.1 Features

- Size

Product Classification		ROM Size	ROM Address
H8S/2437	HD64F2437	256 kbytes	H'000000 to H'03FFFF

- 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 in the initiation determines which memory MAT is initiated first. The MAT can be switched by using the bank-switching method after initiation.

  The user memory MAT is initiated at a power-on reset in user mode: 256 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. This mode can automatically adjust the bit rate between the host and this LSI.

  User program mode:

  The user MAT can be programmed by using any interface.

  User boot mode:

  The user boot program of any interface can be created 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.

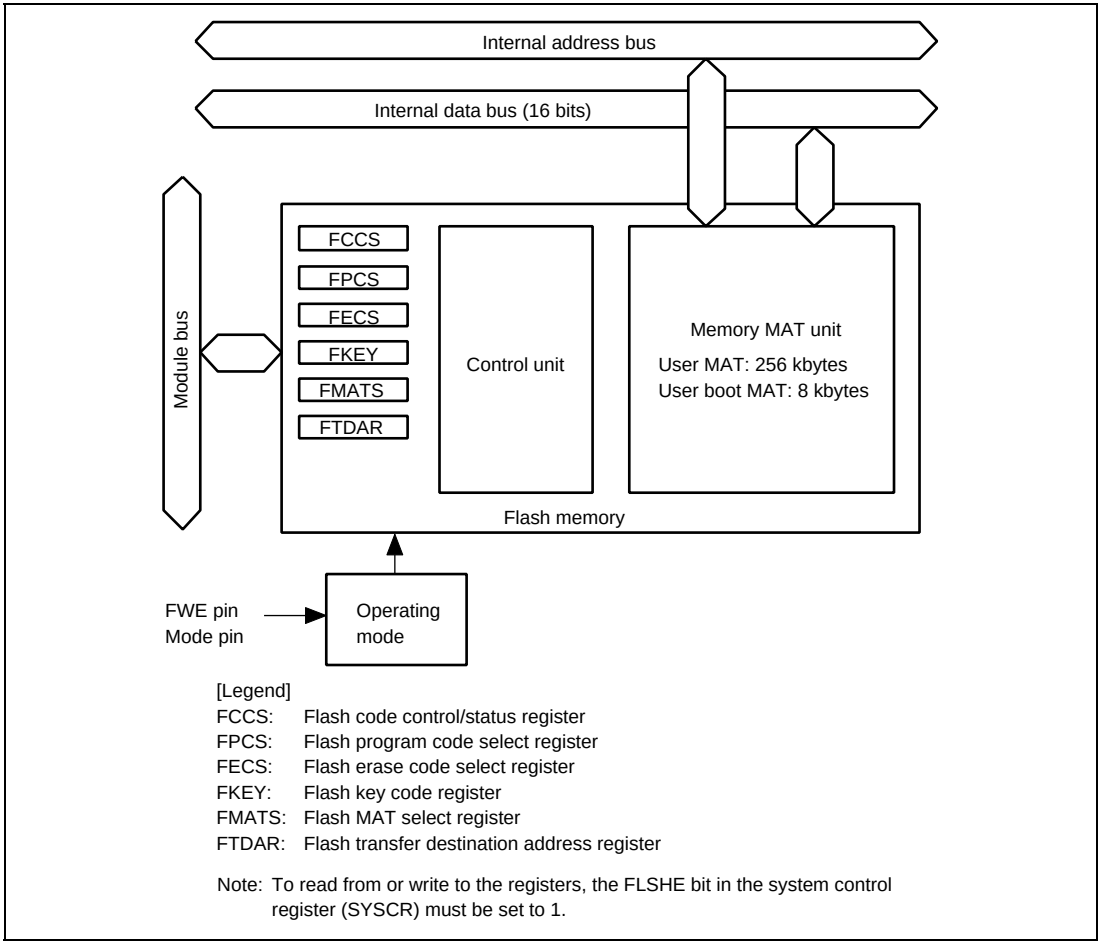


Figure 20.1 Block Diagram of Flash Memory

20.1.1 Mode Transition

When each mode pin and the FWE pin are set in the reset state and the reset is canceled, this LSI enters each operating mode as shown in figure 20.2.

- Flash memory can be read in user mode, but cannot be programmed or erased.
- Flash memory can be read, programmed, or erased on the board only in boot mode, user program mode, and user boot mode.
- Flash memory can be read, programmed, or erased by means of the PROM programmer in programmer mode.

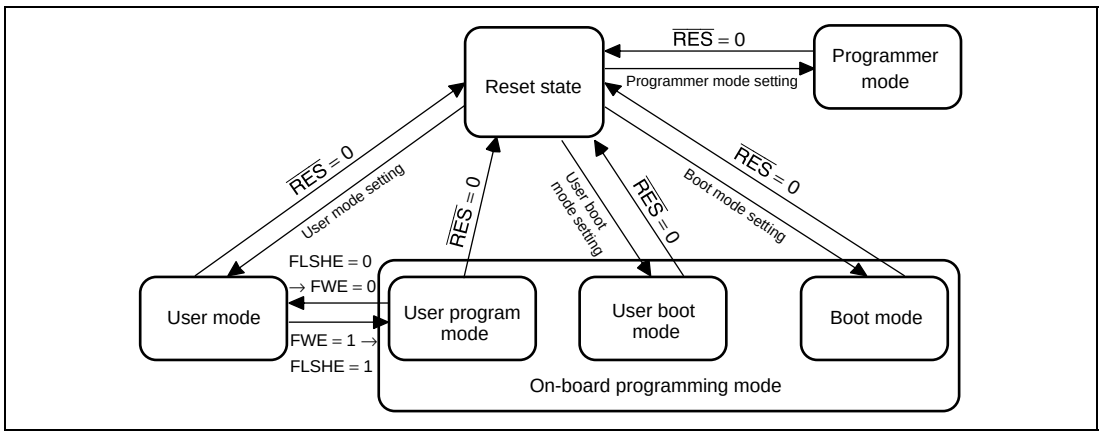


Figure 20.2 Mode Transition of Flash Memory

20.1.2 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 20.1.

Table 20.1 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	○ (Automatic)	○	○	○ (Automatic)
Block division erasure	○ *[1]	○	○	×
Program data transfer	From host via SCI	Via any device	Via any 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 and FWE pin	Changing mode setting and reset	—

Notes: 1. All-erasure is performed. After that, the specified block can be erased.
2. Firstly, 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.
- The user MAT and user boot MAT are erased in boot mode. Then, the user MAT and user boot MAT can be programmed by means of the command method. However, the contents of the MAT cannot be read until this state.
  Only user boot MAT is programmed and the user MAT is programmed in user boot mode or only user MAT is programmed because user boot mode is not used.
- The boot operation of any interface can be performed by the mode pin setting different from user program mode in user boot mode.

20.1.3 Flash Memory MAT Configuration

This LSI's flash memory consists of the 256-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 the 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.

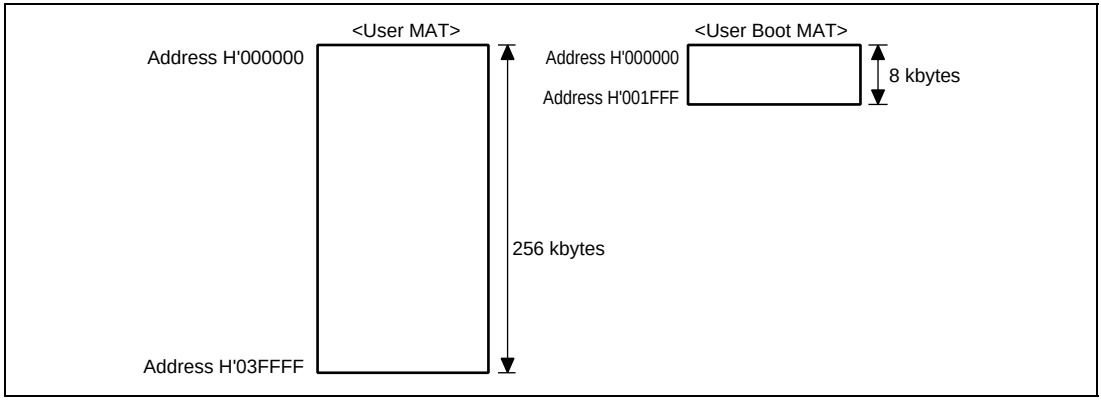


Figure 20.3 Flash Memory Configuration

The size of the user MAT is different from that of the user boot MAT. An address which exceeds the size of the 8-kbyte user boot MAT should not be accessed. If the attempt is made, an undefined value will be read.

20.1.4 Block Division

The user MAT is divided into 64 kbytes (three blocks), 32 kbytes (one block), and 4 kbytes (eight blocks) as shown in figure 20.4. The user MAT can be erased in this divided-block units and the erase-block number of EB0 to EB11 is specified when erasing.

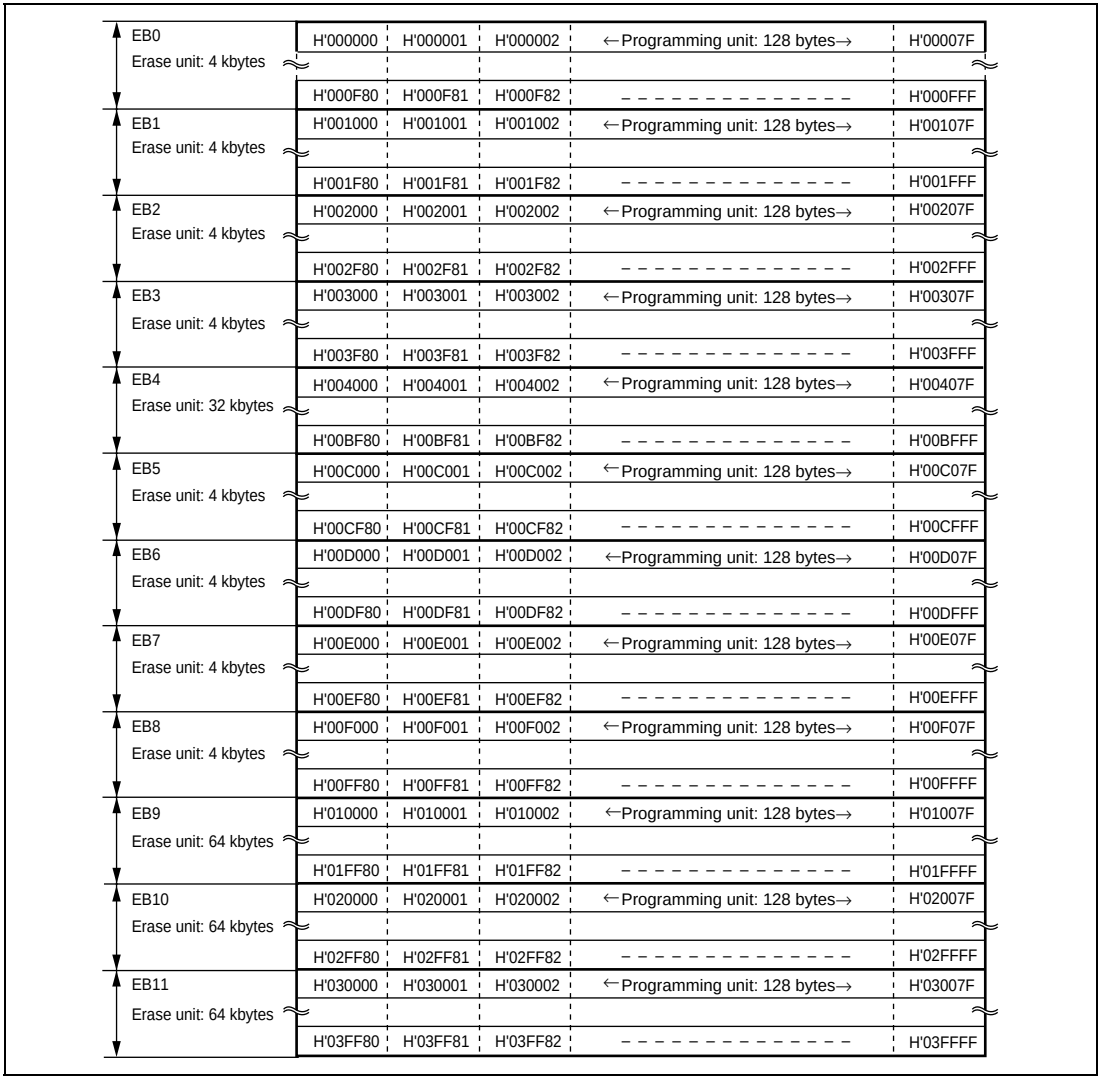


Figure 20.4 Block Division of User MAT

20.1.5 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 should be created by the user in user program mode and user boot mode. An overview of the procedure is given as follows. For details, see section 20.4.2, User Program Mode.

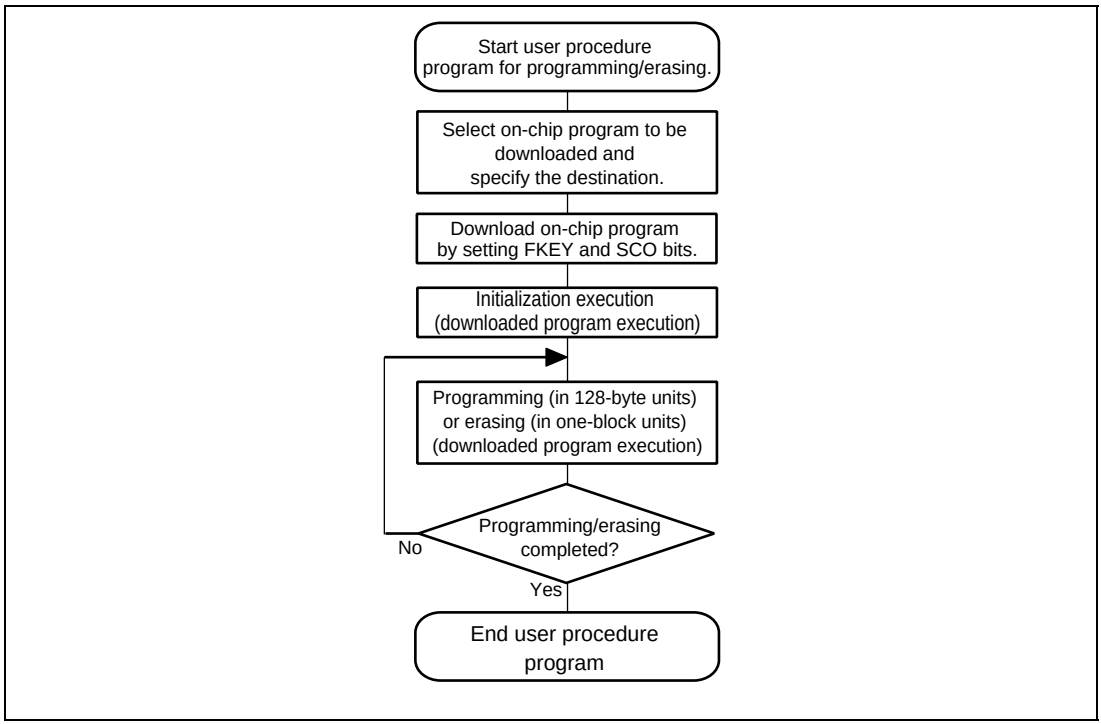


Figure 20.5 Overview of User Procedure Program

(1) Selection of On-chip Program to be Downloaded

For programming/erasing execution, the FLSHE bit in SYSCR must be set to 1 to make a transition to user program mode.

This LSI has programming/erasing programs which 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 can be 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 to the embedded program storage area during downloading. Since the flash memory MAT cannot be read when programming/erasing, the procedure program, which is working from download to completion of programming/erasing, must be executed in the space other than the flash memory (for example, on-chip RAM).

Since the result of download is returned to the programming/erasing interface parameter, whether the normal download is executed or not can be confirmed.

(3) Initialization of Programming/Erasing

The operating frequency is set before execution of programming/erasing. This setting is made by using the programming/erasing interface parameter.

(4) Programming/Erasing Execution

For programming/erasing execution, the FLSHE bit in SYSCR and the FWE pin must be set to 1 to make a transition to user program mode.

The program data/programming destination address is specified in 128-byte units when programming. The block to be erased is specified in erase-block units when erasing.

These specifications are set by using the programming/erasing interface parameter and the on-chip program is initiated. The on-chip program is executed by using the JSR or BSR instruction and performing 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) When Programming/Erasing is Executed Consecutively

When the processing is not ended by the 128-byte programming or one-block erasure, the program address/data and erase-block number must be updated to perform programming/erasing consecutively.

Since the downloaded on-chip program is left in the on-chip RAM after the processing, download and initialization are not required when the same processing is executed consecutively.

20.2 Input/Output Pins

Table 20.2 shows the flash memory pin configuration.

Table 20.2 Pin Configuration

Pin Name	I/O	Function
$\overline{\text{RES}}$	Input	Reset
FWE	Input	Flash memory programming/erasing enable pin
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)

20.3 Register Descriptions

The registers/parameters which control flash memory are shown below. To access these registers, the FLSHE bit in SYSCR must be set to 1. For details on SYSCR, see section 3.2.2, System Control Register (SYSCR).

- 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 program/erase 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 of operating modes and registers/parameters for use is shown in table 20.3.

Table 20.3 Registers/Parameters and Target Modes

		Download	Initialization	Program-ming	Erasure	Read
Programming/ Erasing Interface Registers	FCCS	○	—	—	—	—
	FPCS	○	—	—	—	—
	FECS	○	—	—	—	—
	FKEY	○	—	○	○	—
	FMATS	—	—	○ *1	○ *1	○ *2
	FTDAR	○	—	—	—	—
Programming/ Erasing Interface Parameters	DPFR	○	—	—	—	—
	FPFR	—	○	○	○	—
	FPEFEQ	—	○	—	—	—
	FMPAR	—	—	○	—	—
	FMPDR	—	—	○	—	—
	FEBS	—	—	—	○	—

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.

20.3.1 Programming/Erasing Interface Registers

The programming/erasing interface registers are described below. They are all 8-bit registers that can be accessed only in bytes. These registers are initialized by a reset or in hardware standby mode.

(1) Flash Code Control/Status Register (FCCS)

FCCS is configured by bits which request the monitor of the FWE pin state and error occurrence during programming or erasing flash memory and the download of the on-chip program.

Bit	Bit Name	Initial Value	R/W	Description
7	FWE	1/0	R	Flash Program Enable Monitors the signal level input to the FWE pin. 0: A low level signal is input to the FWE pin. (Hardware protection state) 1: A high level signal is input to the FWE pin.
6, 5	—	All 0	R/W	Reserved The initial value should not be changed.
4	FLER	0	R	Flash Memory Error Indicates an error occurs during programming and erasing flash memory. When this bit is set to 1, flash memory enters the error protection state. When 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 the reset period of 100 µs which is longer than normal. 0: Flash memory operates normally. Programming/erasing protection for flash memory (error protection) is invalid. [Clearing condition] • By a reset or in hardware standby mode 1: An error occurs during programming/erasing flash memory. Programming/erasing protection for flash memory (error protection) is valid. [Setting conditions] • When an interrupt, such as NMI, occurs during programming/erasing flash memory. • When the 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)

Bit	Bit Name	Initial Value	R/W	Description
3	WEINTE	0	R/W	Program/Erase Enable Modifies the space for the interrupt vector table, when interrupt vector data is not read successfully during programming/erasing flash memory or switching between a user MAT and a user boot MAT. When this bit is set to 1, interrupt vector data is read from address spaces H'FF6000 to H'FF607F (on-chip RAM space), instead of from address spaces H'000000 to H'00007F (up to vector number 31). Therefore, make sure to set the vector table in the on-chip RAM space before setting this bit to 1. The interrupt exception handling on and after vector number 32 should not be used because the correct vector is not read, resulting in the CPU runaway. 0: The space for the interrupt vector table is not modified. When interrupt vector data is not read successfully, the operation for the interrupt exception handling cannot be guaranteed. An occurrence of any interrupts should be masked. 1: The space for the interrupt vector table is modified. Even when interrupt vector data is not read successfully, the interrupt exception handling up to vector number 31 is enabled.
2, 1	—	All 0	R/W	Reserved The initial value should not be changed.
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. Four NOP instructions must be executed immediately after setting this bit to 1. Since this bit is cleared to 0 when download is completed, this bit cannot be read as 1. All interrupts must be disabled during download. This should be made in the user system. 0: Download of the on-chip programming/erasing program to the on-chip RAM is not executed.

Bit	Bit Name	Initial Value	R/W	Description
0	SCO	0	(R)/W*	[Clearing condition] • When download is completed 1: A request in which the on-chip programming/erasing program is downloaded to the on-chip RAM occurs. [Setting conditions] When all of the following conditions are satisfied and 1 is set to this bit • 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.

(2) 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.

(3) Flash Erase Code Select Register (FECS)

FECS selects download of the on-chip erasing program.

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.

(4) Flash Key Code Register (FKEY)

FKEY is a register for software protection that enables download of the on-chip program and programming/erasing of flash memory. Before setting the SCO bit to 1 in order to download the on-chip program or executing the downloaded programming/erasing program, these processing cannot be executed if the key code is not written.

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 the value other than H'A5 is written to FKEY, the SCO bit cannot be set to 1. Therefore downloading to the on-chip RAM cannot be executed.
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	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 the value other than H'5A is written to FKEY.
0	K0	0	R/W	
				H′A5: Writing to the SCO bit is enabled. (The SCO bit cannot be set by the value other than H'A5.)
				H′5A: Programming/erasing is enabled. (The software protection state is entered by the value other than H'5A.)
				H′00: Initial value

(5) 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	These bits are in the user-MAT selection state when the value other than H'AA is written and in the user-boot-MAT selection state when H′AA is written. The MAT is switched by writing the value to FMATS. When the MAT is switched, follow section 20.6, Switching between User MAT and User Boot MAT. (The user boot MAT cannot be programmed in user program mode if user boot MAT is selected by FMATS. The user boot MAT must be programmed in boot mode or in programmer mode.) H'AA: The user boot MAT is selected (in user-MAT selection state when the value of these bits are other than H′AA) Initial value when these bits are initiated in user boot mode. H′00: Initial value when these bits are initiated in a mode except for user boot mode (in user-MAT selection state) [Programmable condition] • During 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.

(6) Flash Transfer Destination Address Register (FTDAR)

FTDAR specifies the on-chip RAM address which is download destination of an on-chip program. This register must be set 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 to download an on-chip program, is over the range. Whether or not the range specified by bits TDA6 to TDA0 is within the range of H'00 to H'03 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 before setting the SCO bit to 1 and the value specified by bits TDA6 to TDA0 is within the range of H'00 to H'03. 0: The value specified by bits TDA6 to TDA0 is within the range. 1: The value specified by bits TDA6 to TDA0 is over the range (H'04 to H′FF) and the download is stopped.
6	TDA6	0	R/W	Transfer Destination Address Specify the start address to download an on-chip program. H'00 to H'03 can be specified as the start address in the on-chip RAM space. H′00: H'FF6000 is specified as a start address to download an on-chip program. H′01: H'FF7000 is specified as a start address to download an on-chip program. H′02: H'FF8000 is specified as a start address to download an on-chip program. H′03: H'FF9000 is specified as a start address to download an on-chip program. H′04 to H'FF: Setting prohibited. Specifying these values sets the TDER bit to 1 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	

20.3.2 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 exchange the processing result for the downloaded on-chip program. These parameters use the general registers of the CPU (ER0 and ER1) or the on-chip RAM area. The initial value is undefined by a reset or in hardware standby mode.

When download, initialization, or on-chip program is executed, registers of the CPU except for R0L are stored. The return value of the processing result is written to 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 stack area to be used is 128 bytes.)

The programming/erasing interface parameters are used in the following four items.

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 20.4.

A result of initialization, programming, or erasure processing is returned to the FPFR parameters. However, the meaning of bits in FPFR varies in each processing. For details, see descriptions of FPFR for each processing.

Table 20.4 Parameters and Target Modes

Name of Parameter	Abbreviation	Download	Initialization	Programming	Erasure	R/W	Initial Value	Allocation
Download pass/fail result	DPFR	○	—	—	—	R/W	Undefined	On-chip RAM*
Flash pass/fail result	FPFR	—	○	○	○	R/W	Undefined	R0L of CPU
Flash programming/ erasing frequency control	FPEFEQ	—	○	—	—	R/W	Undefined	ER0 of CPU
Flash multi-purpose address area	FMPAR	—	—	○	—	R/W	Undefined	ER1 of CPU
Flash multi-purpose data destination area	FMPDR	—	—	○	—	R/W	Undefined	ER0 of CPU
Flash erase block select	FEBS	—	—	—	○	R/W	Undefined	R0L of CPU

Note: * A single byte of the start address to download an on-chip program, which is 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 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 DPFR indicates the return value.

(a) Download Pass/Fail Result Parameter (DPFR: Single Byte of On-Chip RAM Start Address Specified by FTDAR)

DPFR indicates the return value of the download result. The value of this parameter can be used to determine if downloading is executed or not. Since the confirmation whether the SCO bit is set to 1 is difficult, the certain determination must be performed by setting the single byte of the start address specified by FTDAR to the value other than the return value of download (for example, H'FF) before the download start (before setting the SCO bit to 1).

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Unused Return 0.
2	SS	—	R/W	Source Select Error Detect Only one type of the on-chip program which can be downloaded can be specified. When more than two types of programs are selected, the program is not selected, or the program is selected without being mapped, an error occurs. 0: Download program can be selected normally 1: Download error occurred (multiple selection or program which is not mapped is selected)
1	FK	—	R/W	Flash Key Register Error Detect Returns the check result whether the value of FKEY is set to H'A5. 0: FKEY setting is normal (FKEY = H'A5) 1: Setting value of FKEY becomes error (FKEY = value other than H′A5)
0	SF	—	R/W	Success/Fail Returns the result whether download is ended normally or not. The determination result whether program that is downloaded to the on-chip RAM is read back and then transferred to the on-chip RAM is returned. 0: Downloading on-chip program is ended normally (no error) 1: Downloading on-chip program is ended abnormally (error occurs)

(2) Programming/Erasing Initialization

The on-chip programming/erasing program to be downloaded includes the initialization program.

The specified period pulse must be applied when programming or erasing. The specified pulse width is made by the method in which wait loop is configured by the CPU instruction. The operating frequency of the CPU must be set.

The initialization program is set as a parameter of the programming/erasing program which has downloaded these settings.

(a) Flash Program/Erase Frequency Parameter (FPEFEQ: General Register ER0 of CPU)

FPEFEQ sets the operating frequency of the CPU. The settable range of the operating frequency in this LSI is 5 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 Set the operating frequency of the CPU. With the PLL multiplication function, set the frequency multiplied. The setting value must be calculated as the following methods. • The operating frequency which is shown in MHz units must be rounded a number to three decimal places and be shown in a number of two decimal places. • The value multiplied by 100 is converted to the binary digit and is written to FPEFEQ (general register ER0). For example, when the operating frequency of the CPU is 20.000 MHz, the value is as follows. • The number to three decimal places of 20.000 is rounded and the value is thus 20.00. • The formula that 20.00 × 100 = 2000 is converted to the binary digit and B'0000, B'0111, B'1101, and B'0000 (H'07D0) are set to ER0.

(b) Flash Pass/Fail Parameter (FPFR: General Register R0L of CPU)

FPFR indicates the return value of the initialization result.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	—	—	Unused Return 0.
1	FQ	—	R/W	Frequency Error Detect Returns the check result whether the specified operating frequency of the CPU 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 is completed normally. 0: Initialization is ended normally (no error) 1: Initialization is ended abnormally (error occurs)

(3) Programming Execution

When flash memory is programmed, the programming destination address on the user MAT and program data must be transferred to the downloaded programming program.

1. The start address of the programming destination in the user MAT must be set to the general register ER1. This parameter is called the flash multipurpose address area parameter (FMPAR).

   Since program data is always in 128 bytes, the lower eight bits (A7 to A0) must be H'00 or H'80 as the boundary of the programming start address in the user MAT.
2. The program data for the user MAT must be prepared in the consecutive area. The program data must be in the consecutive space which can be accessed by using the MOV.B instruction of the CPU and in other than the flash memory space.

   When data to be programmed is less than 128 bytes, the 128-byte program data must be prepared by filling with the dummy code H'FF.

   The start address of the area in which the prepared program data is stored must be set to the general register ER0. This parameter is called the flash multipurpose data destination area parameter (FMPDR).

For details on the program processing procedure, see section 20.4.2, User Program Mode.

(a) Flash Multipurpose Address Area Parameter (FMPAR: General Register ER1 of CPU)

FMPAR sets the start address of the programming destination in the user MAT.

When the address in the 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 FPFR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOA31 to MOA0	—	R/W	Store the start address of the programming destination in the user MAT. The 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 bits MOA6 to MOA0 are always 0.

(b) Flash Multipurpose Data Destination Parameter (FMPDR: General Register ER0 of CPU):

FMPDR sets the start address in the area which stores the data to be programmed to 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 FPFR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOD31 to MOD0	—	R/W	Store the start address of the area which stores the program data for the user MAT. The consecutive 128-byte data is programmed to the user MAT starting from the specified start address.

(c) Flash Pass/Fail Parameter (FPFR: General Register R0L of CPU)

FPFR indicates the return value of the program processing result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused Returns 0.
6	MD	—	R/W	Error Detect for Programming Mode Related Setting Returns the check result that a high level signal is input to the FWE pin and the error protection state is not entered. When the low level signal is input to the FWE pin or the error protection state is entered, 1 is written to this bit. The state can be confirmed with the FWE and FLER bits in FCCS. For conditions to enter the error protection state, see section 20.5.3, Error Protection. 0: FWE and FLER settings are normal (FWE = 1, FLER = 0) 1: Programming cannot be performed (FWE = 0 or FLER = 1)
5	EE	—	R/W	Error Detect During Programming Execution 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 reprogrammed. In this case, after removing the error source, 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 reprogrammed. 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
4	FK	—	R/W	Error Detect for Flash Key Register Returns the check result of the value of FKEY before the start of the programming processing. 0: FKEY setting is normal (FKEY = H′5A) 1: FKEY setting error (FKEY = value other than H′5A)

Bit	Bit Name	Initial Value	R/W	Description
3	—	—	—	Unused Returns 0.
2	WD	—	R/W	Program Data Address Detect When the 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	Program Address Error Detect When the following items are specified as the start address of the programming destination, an error occurs. • When the programming destination address in the area other than flash memory is specified • When the specified address is not in a 128-byte boundary (The lower eight bits of the address are other than H'00 and 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 program processing is ended normally or not. 0: Programming is ended normally (no error) 1: Programming is ended abnormally (error occurs)

(4) Erasure Execution

When flash memory is erased, the erase-block number in the user MAT must be transferred to the erasing program which is downloaded. This is set to FEBS (general register ER0).

One block is specified from the block number 0 to 11.

For details on the erasing processing procedure, see section 20.4.2, User Program Mode.

(a) Flash Erase Block Select Parameter (FEBS: General Register ER0 of CPU)

FEBS specifies the erase-block number. The several block numbers cannot be specified.

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	—	—	Unused These bits should be cleared to 0.
11	EB11	—	R/W	Erase Block Set the erase-block number in the range from 0 to 11. 0 corresponds to the EB0 block and 11 corresponds to the EB11 block. An error occurs when the number other than 0 to 11 is set.
10	EB10	—	R/W	
9	EB9	—	R/W	
8	EB8	—	R/W	
7	EB7	—	R/W	
6	EB6	—	R/W	
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 Parameter (FPFR: General Register R0L of CPU)

FPFR indicates a return value of the erasing processing result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused Returns 0.
6	MD	—	R/W	Error Detect for Erasing Mode Related Setting Returns the check result that a high level signal is input to the FWE pin and the error protection state is not entered. When the low level signal is input to the FWE pin or the error protection state is entered, 1 is written to this bit. The state can be confirmed with the FWE and FLER bits in FCCS. For conditions to enter the error protection state, see section 20.5.3, Error Protection. 0: FWE and FLER settings are normal (FWE = 1, FLER = 0) 1: Erasing cannot be performed (FWE = 0 or 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 source, 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.
4	FK	—	R/W	Error Detect for Flash Key Register Returns the check result of FKEY value before start of the erasing processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting error (FKEY = value other than H′5A)
3	EB	—	R/W	Error Detect for Erase Block Select 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

Bit	Bit Name	Initial Value	R/W	Description
2, 1	—	—	—	Unused Return 0.
0	SF	—	R/W	Success/Fail Indicates whether the erasing processing is ended normally or not. 0: Erasure is ended normally (no error) 1: Erasure is ended abnormally (error occurs)

20.4 On-Board Programming Mode

When the pin is set in on-board programming mode and the reset start is executed, the on-board programming state that can program/erase the on-chip flash memory is entered. 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 20.5. For details on the state transition of each mode for flash memory, see figure 20.2.

Table 20.5 Setting On-Board Programming Mode

Mode Setting	FWE	MD2	MD1	MD0
Boot mode	1	0	0	1
User program mode	1*	1	1	1
User boot mode	1	1	0	1

Note: * Before downloading the programming/erasing programs, the FLSHE bit must be set to 1 to make a transition to user program mode.

20.4.1 Boot Mode

Boot mode executes programming/erasing the user MAT and user boot MAT by the method for transmitting control command and program data from the host using the on-chip SCI. The tool for transmitting the control command and program data must be prepared in the host. SCI communication mode is set to asynchronous mode. When a reset start is executed after this LSI's pin is set in boot mode, the boot program in the microcomputer is initiated. After the SCI bit rate is automatically adjusted, the communication with the host is executed by means of the control command method.

The system configuration diagram in boot mode is shown in figure 20.6. For details on the pin setting in boot mode, see table 20.5. The NMI and other interrupts are ignored in boot mode. However, the NMI and other interrupts should be disabled in the system.

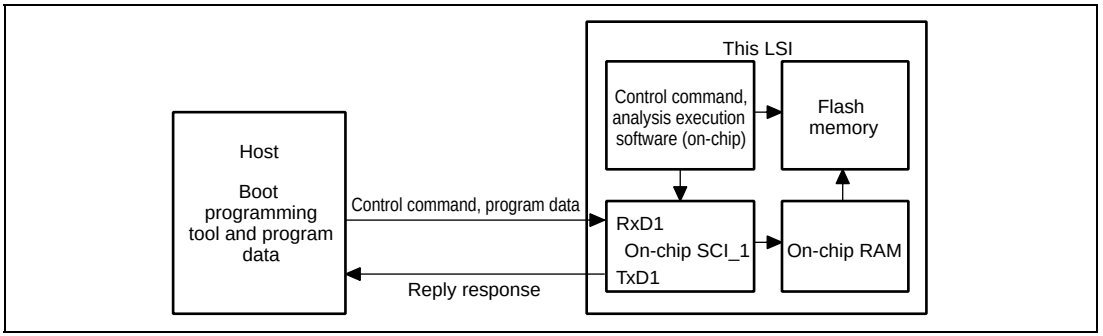


Figure 20.6 System Configuration in Boot Mode

(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 by the host. The SCI transmit/receive format should be set to 8-bit data, 1 stop bit, and no parity. This LSI calculates the bit rate which is transmitted by the host according to 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 transmit 1 byte of H'55 to this LSI. When reception is not performed normally, boot mode is initiated again (reset) and the operation described above must be executed. The bit rates of the host and this LSI do not match according to the bit rate transmitted by the host and 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 20.6. Boot mode must be initiated in the range of this system clock.

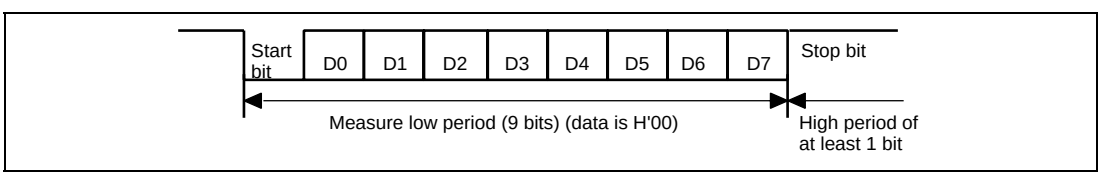


Figure 20.7 Automatic-Bit-Rate Adjustment Operation of SCI

Table 20.6 System Clock Frequency for Automatic-Bit-Rate Adjustment

Bit Rate of Host	System Clock Frequency which can Automatically Adjust Bit Rate of this LSI
4,800 bps	5 to 20 MHz
9,600 bps	5 to 20 MHz
19,200 bps	5 to 20 MHz

(2) State Transition Diagram

The overview of the state transition diagram after boot mode is initiated is shown in figure 20.8.

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 MAT and user boot MAT

   After inquiries have finished, all user MAT and user boot MAT are automatically erased.
4. Waiting for programming/erasing command

- When the program preparation notice is received, the state for waiting 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 for waiting program data is returned to the state for waiting programming/erasing command.
- When the erasure preparation notice is received, the state for waiting 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 for waiting erase-block data is returned to the state for waiting programming/erasing command. The erasure must be used when the specified block is programmed without a reset start after programming is executed in boot mode. When programming can be executed by only one operation, all blocks are erased before the state for waiting programming/erasing/other command is entered. Thus the erasing operation is not required.
- There are many commands other than programming/erasing: sum check, blank check (erasure check), and memory read of the user MAT/user boot MAT and acquisition of current status information.

Note that memory read of the user MAT/user boot MAT is only applied to data programmed after all user MAT/user boot MAT has been automatically erased.

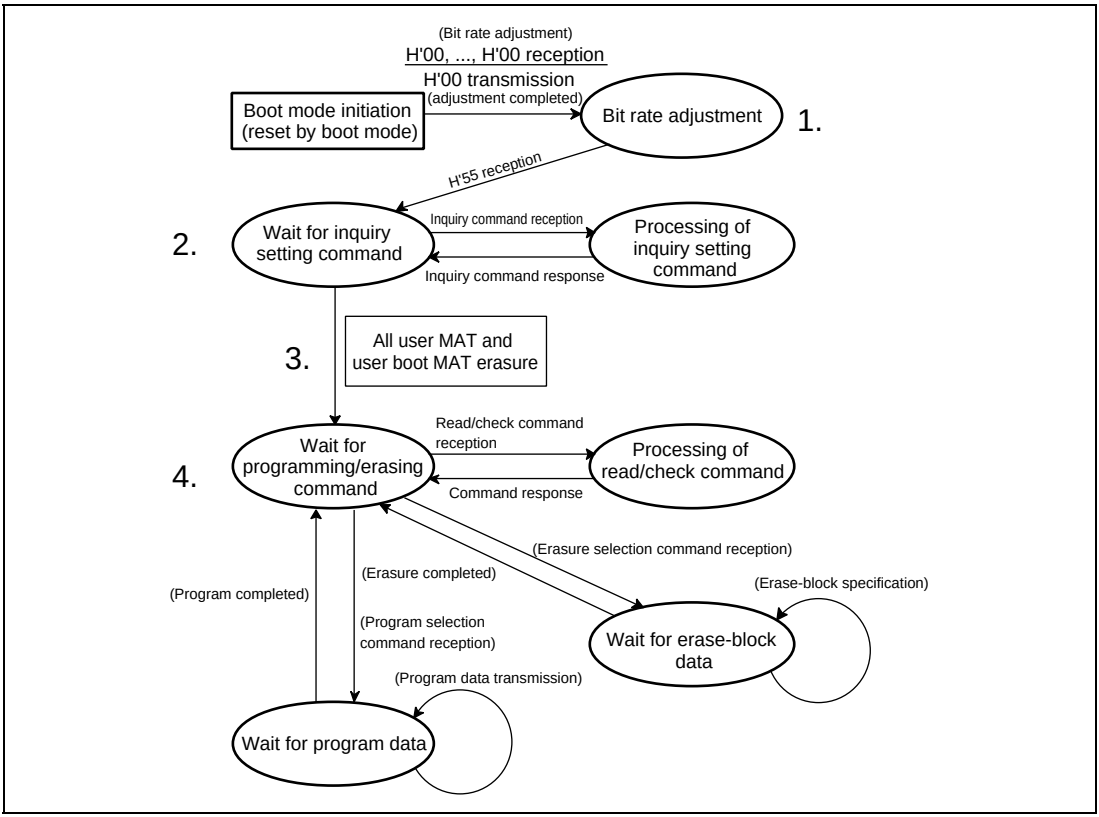


Figure 20.8 Overview of State Transition Diagram in Boot Mode

20.4.2 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 in the microcomputer.

The overview of programming/erasing flow is shown in figure 20.9.

High voltage is applied to the internal flash memory during programming/erasing processing. Therefore, a transition to a reset or hardware standby must not be made during programming/erasing processing. Doing so may damage or destroy flash memory. If a reset is executed accidentally, a reset must be released after the reset input period of 100 µs which is longer than normal.

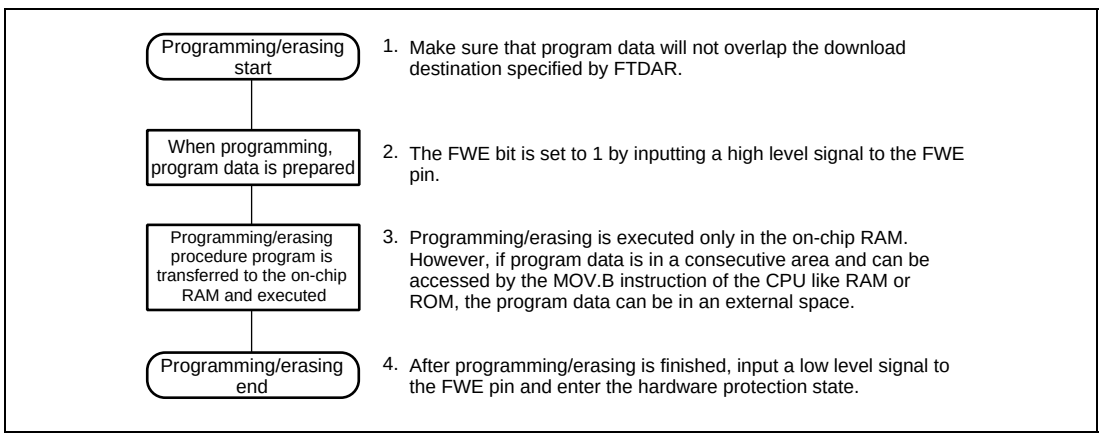


Figure 20.9 Overview of Programming/Erasing Flow

(1) On-chip RAM Address Map when Programming/Erasing is Executed

Some of the procedure programs that should be created by the user, such as a download request, programming/erasing procedure, and determination of the result, must be executed in the on-chip RAM. The on-chip programs to be downloaded are all in the on-chip RAM. Note that area in the on-chip RAM must be controlled so that these programs do not overlap.

Figure 20.10 shows the program area to be downloaded.

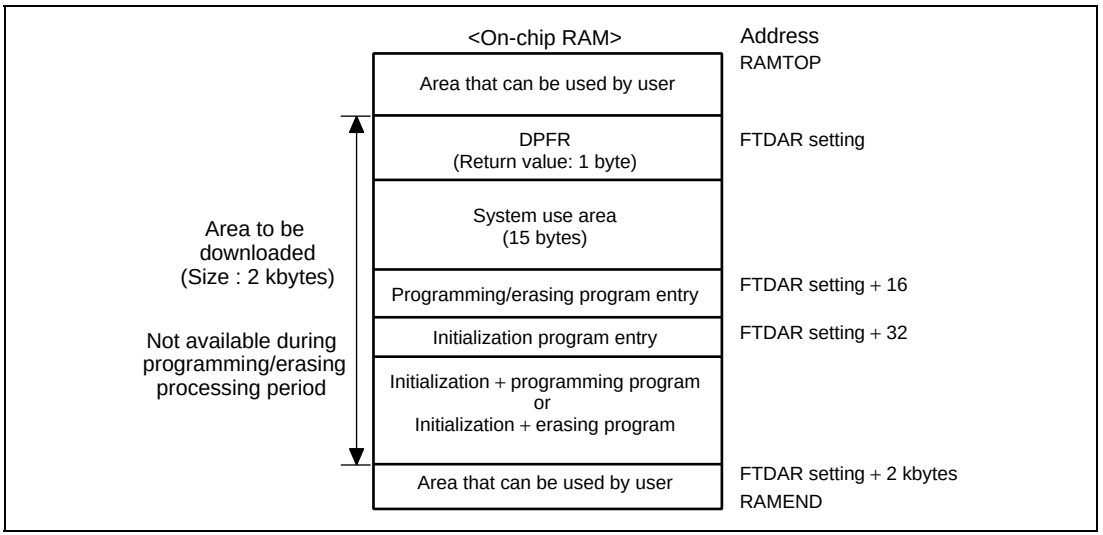


Figure 20.10 RAM Map when Programming/Erasing is Executed

(2) Programming Procedure in User Program Mode

The procedures for download, initialization, and programming are shown in figure 20.11.

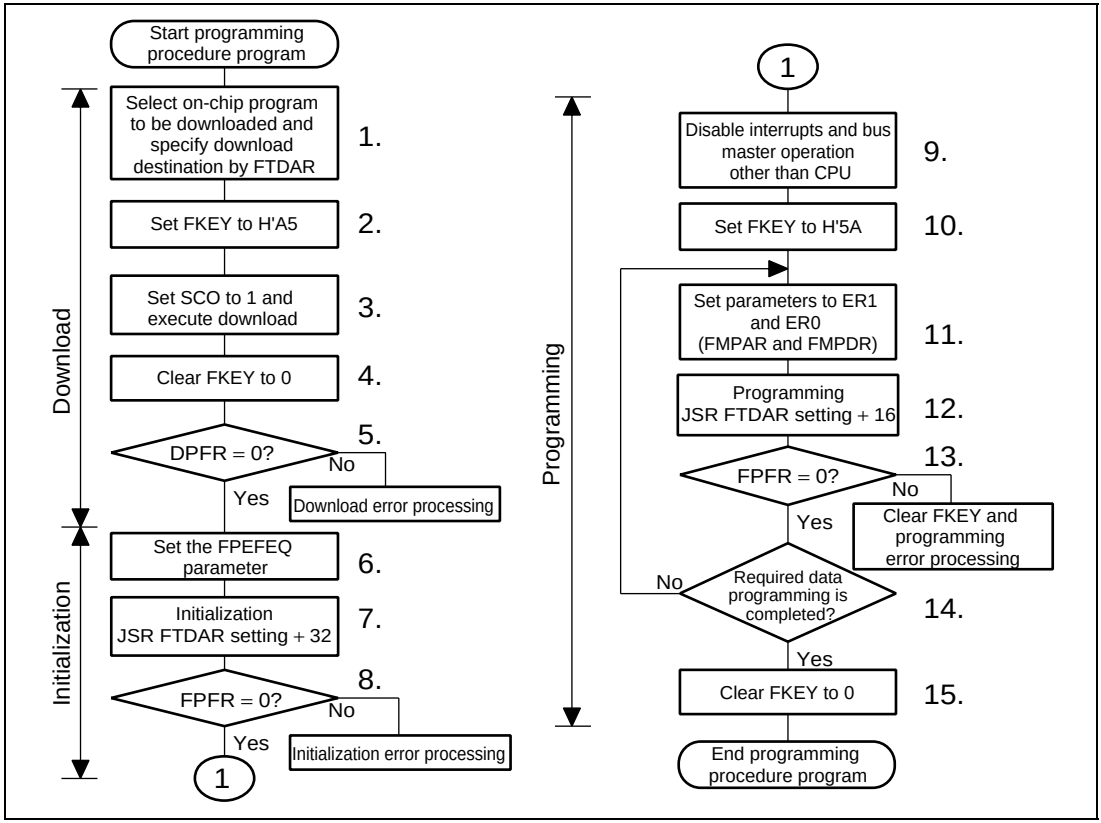


Figure 20.11 Programming Procedure

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 each step of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 20.4.4, Storable Area for Procedure Program and Program Data.

The following description assumes the area to be programmed in the user MAT is erased and program data is prepared in the consecutive area. When erasing is not executed, erasing should be executed before programming.

128-byte programming is performed in one program processing. When more than 128-byte programming is performed, programming destination address/program data parameter is updated in 128-byte units and programming is repeated.

When less than 128-byte programming is performed, data size must be128 bytes by adding the invalid data. If the invalid data to be added is set to H'FF, the program processing period can be shortened.

1. 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 a download destination is specified by FTDAR.

2. Write H'A5 to FKEY

   If H'A5 is not written to FKEY for protection, 1 cannot be set to the SCO bit for a download request.

3. Set 1 to the SCO bit in FCCS and then execute downloading

   To set 1 to the SCO bit, the following conditions must be satisfied.

   A H'A5 is written to FKEY.

   B The SCO bit writing is executed in the on-chip RAM.

   When the SCO bit is set to 1, download is started automatically. When the user procedure program is returned, the SCO bit is 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. Before the SCO bit is set to 1, incorrect determination must be prevented by setting the one byte of the start address specified by FTDAR (to be used as DPFR) to a value other than the return value (H'FF).
   When download is executed, particular interrupt processing, which is accompanied by the bank switch as described below, is performed as internal microcomputer processing. Four NOP instructions should be executed immediately after an instruction that sets the SCO bit to 1.

- The user-MAT space is switched to the on-chip program storage area.
- After the selection condition of the download program and the FTDAR setting are checked, the transfer processing to the on-chip RAM specified by FTDAR is executed.
- FPCS, FECS, and the SCO bit in FCCS are cleared to 0.
- The return value is set to DPFR.
- After the on-chip program storage area is returned to the user-MAT space, the user procedure program is returned.
- In download processing, the values of general registers of the CPU are retained.

- In download processing, all interrupts are not accepted. However, interrupt requests other than the NMI are retained. Therefore, when the user procedure program is returned, the interrupts occur.
- When the level-detection interrupt requests need to be retained, interrupts must be input until the download is ended.
- When hardware standby mode is entered during download processing, the normal download cannot be guaranteed in the on-chip RAM. Therefore, download must be executed again.
- Since a stack area of 128 bytes at the maximum is used, the area must be saved before setting the SCO bit to 1.

4. Clear FKEY to H'00 for protection

5. Check the value of DPFR to confirm the download result

- Check the value of DPFR (one 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 DPFR is the same as that 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 DPFR is different from that before downloading, check the SS and FK bits in DPFR to ensure that the download program selection and FKEY setting were normal, respectively.

6. Set the operating frequency to FPEFEQ for initialization

- The current frequency of the CPU clock is set to FPEFEQ (general register ER0).

  The settable range of FPEFEQ is 5 to 20 MHz. When the frequency is set to out of this range, an error is returned to FPFR of the initialization program and initialization is not performed. For details on the frequency setting, see section 20.3.2 (2) (a), Flash Program/Erase Frequency Parameter (FPEFEQ).

7. Execute initialization

   When the programming program is downloaded, the initialization program is also downloaded to the on-chip RAM. There is an entry point of the initialization program in the area from the start address specified by FTDAR + 32 bytes of the on-chip RAM. The subroutine should be called and initialization should be 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 retained in the initialization program.
- R0L is a return value of FPFR.

- Since the stack area is used in the initialization program, 128-byte stack area at the maximum must be saved in RAM.
- Interrupts can be accepted during the execution of the initialization program. The program storage area and stack area in the on-chip RAM and register values must not be destroyed.

8. Check the return value in the initialization program, FPFR (general register R0L)

9. Disable all interrupts and the use of a bus master other than the CPU

   The specified voltage is applied for the specified time when programming or erasing. If interrupts occur or the bus mastership is moved to other than the CPU during this time and the voltage is applied for more than the specified time, flash memory may be damaged. Therefore, interrupts and bus mastership moved to other than the CPU must be disabled.

   When interrupts are disabled, 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 CCR should be set to B′11 in interrupt control mode 2. Then interrupts other than the NMI are retained and are not executed.

   The NMI interrupt must be masked within the user system.

   The interrupts that are retained must be executed after all program processing.

10. Set FKEY to H'5A to enable the user MAT programming.

11. Set the parameter which is required for programming

    The start address of the programming destination of the user MAT (FMPAR) is set to the general register ER1. The start address of the program data area (FMPDR) is set to the general register ER0.

- Example of the 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 unit is 128 bytes, the lower eight bits of the address must be H'00 or H′80 as the boundary of 128 bytes.
- Example of the FMPDR setting

  When the storage destination of program data is flash memory, even if the program execution routine is executed, programming is not executed and an error is returned to FPFR. In this case, the program data must be transferred to the on-chip RAM and then programming must be executed.

12. Execute programming

    There is an entry point of the programming program in the area from the start address specified by FTDAR + 16 bytes of the on-chip RAM. The subroutine should be called and programming should be 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 retained in the programming program.
- R0L is a return value of FPFR.
- Since the stack area is used in the programming program, a stack area of 128 bytes at the maximum must be saved in RAM.

13. Check the return value in the programming program, FPFR (general register R0L)

14. Determine whether programming of the necessary data has finished

    If more than 128 bytes of data needs to be programmed, specify FMPAR and FMPDR in 128-byte units, and repeat steps 12 to 14. Increment the programming destination address by 128 bytes and update the programming data pointer correctly. If an address which has already been programmed is programmed again, not only will a programming error occur, but also flash memory will be damaged.

15. After programming is finished, clear FKEY and specify software protection

    If this LSI is restarted by a reset immediately after user MAT programming has finished, secure the 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 20.12.

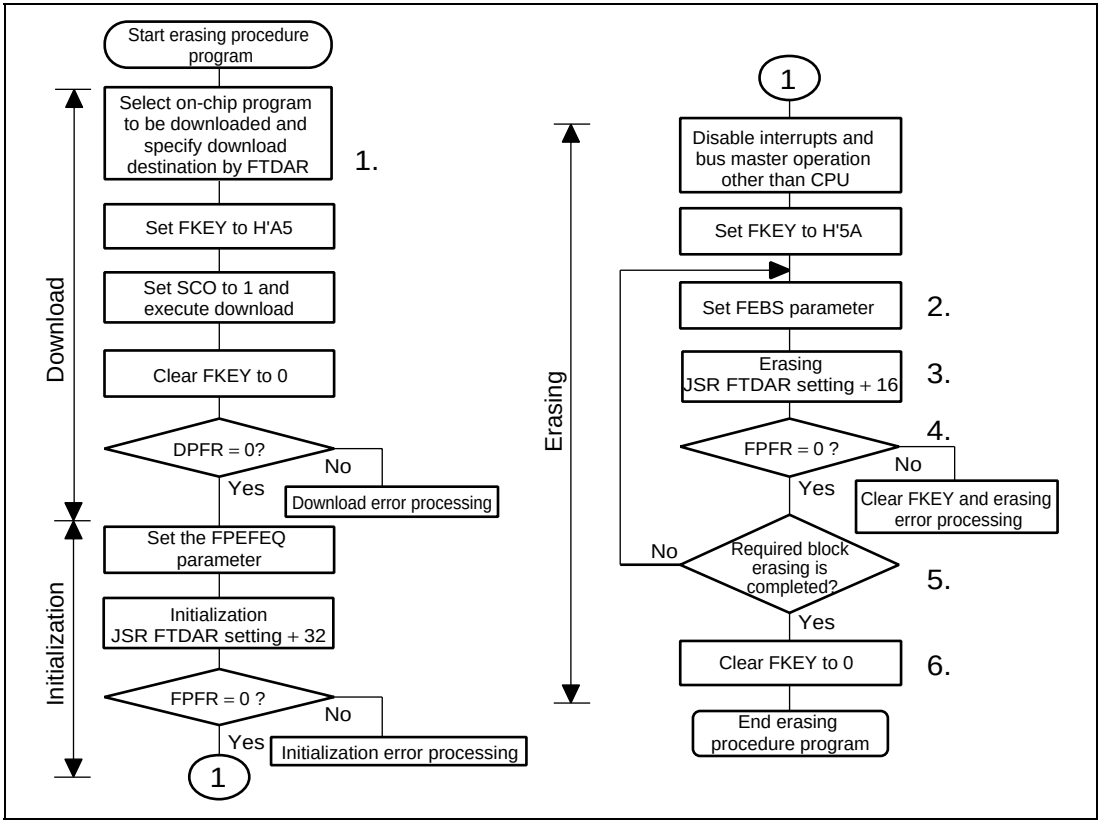


Figure 20.12 Erasing Procedure

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 each step of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 20.4.4, Storable Area for Procedure Program and Program Data.

For the area of the on-chip program to be downloaded, see the RAM map for programming/erasing in figure 20.10.

A single divided block is erased in one erasing processing. For block divisions, see figure 20.4. To erase two or more blocks, update the erase block number and perform the erasing processing for each block.

1. 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 returned to the SS bit in DPFR.

   Specify the start address of a 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 20.4.2 (2), Programming Procedure in User Program Mode.

   The procedures after setting parameters for erasing programs are as follows:

2. Set FEBS necessary for erasure

   Set the erase block number of the user MAT to 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.

3. Execute erasing

   Similar to as in programming, there is an entry point of the erasing program in the area from the start address of a download destination specified by FTDAR + 16 bytes of on-chip RAM. The subroutine should be called and erasing should be 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 retained in the erasing program.
   - R0L is a return value of FPFR.
   - Since the stack area is used in the erasing program, a stack area of 128 bytes at the maximum must be saved in RAM.

4. Check the return value in the erasing program, FPFR (general register R0L)

5. Determine whether erasure of the necessary blocks has completed

   If several blocks need to be erased, update FEBS and repeat steps 2 to 5. Blocks that have already been erased can be erased again.

6. 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 the reset period (period of $\overline{RES}$ = 0) of at least 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 20.13 shows a repeating procedure of erasing and programming.

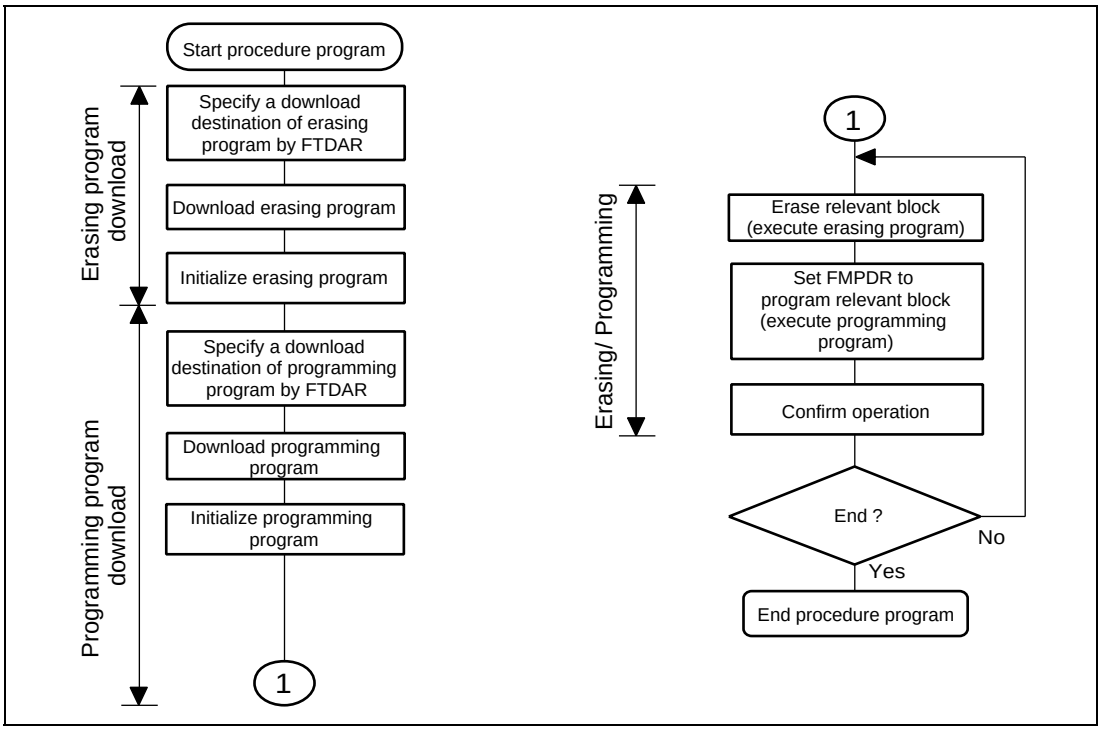


Figure 20.13 Repeating Procedure of Erasing and Programming

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 reserved in 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 FPEFEQ must be performed for both the erasing program and the 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.

20.4.3 User Boot Mode

This LSI has user boot mode which is initiated with different mode pin settings from 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 should be performed only in boot mode or programmer mode.

(1) User Boot Mode Initiation

For the mode pin settings to start up user boot mode, see table 20.5.

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 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 20.14 shows the procedure for programming the user MAT in user boot mode.

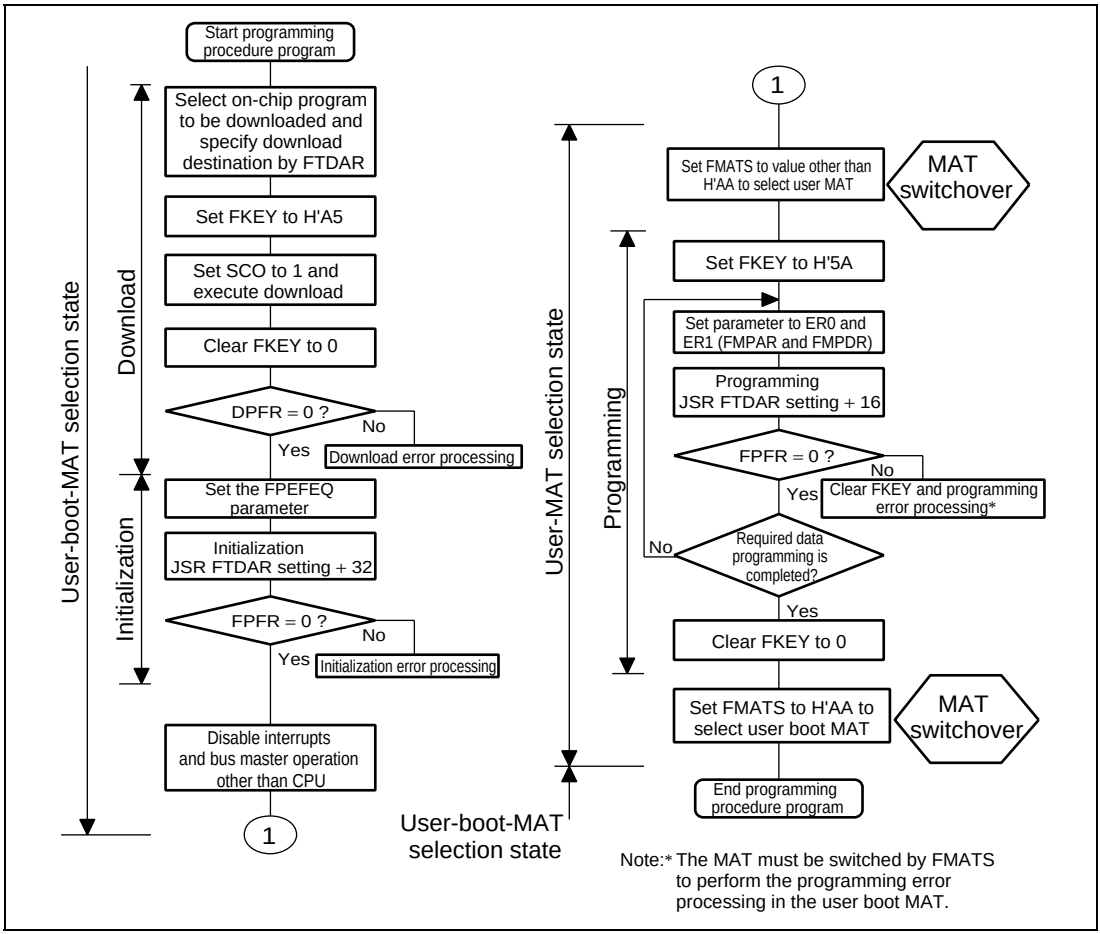


Figure 20.14 Procedure for Programming User MAT in User Boot Mode

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 20.14.

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 located in an area other than flash memory. After programming completes, switch the MATs again to return to the first state.

MAT switching is enabled by writing a specific value to FMATS. However, note 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 20.6, 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, user MAT, and external space) is shown in section 20.4.4, Storable Area 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 is 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 20.15 shows the procedure for erasing the user MAT in user boot mode.

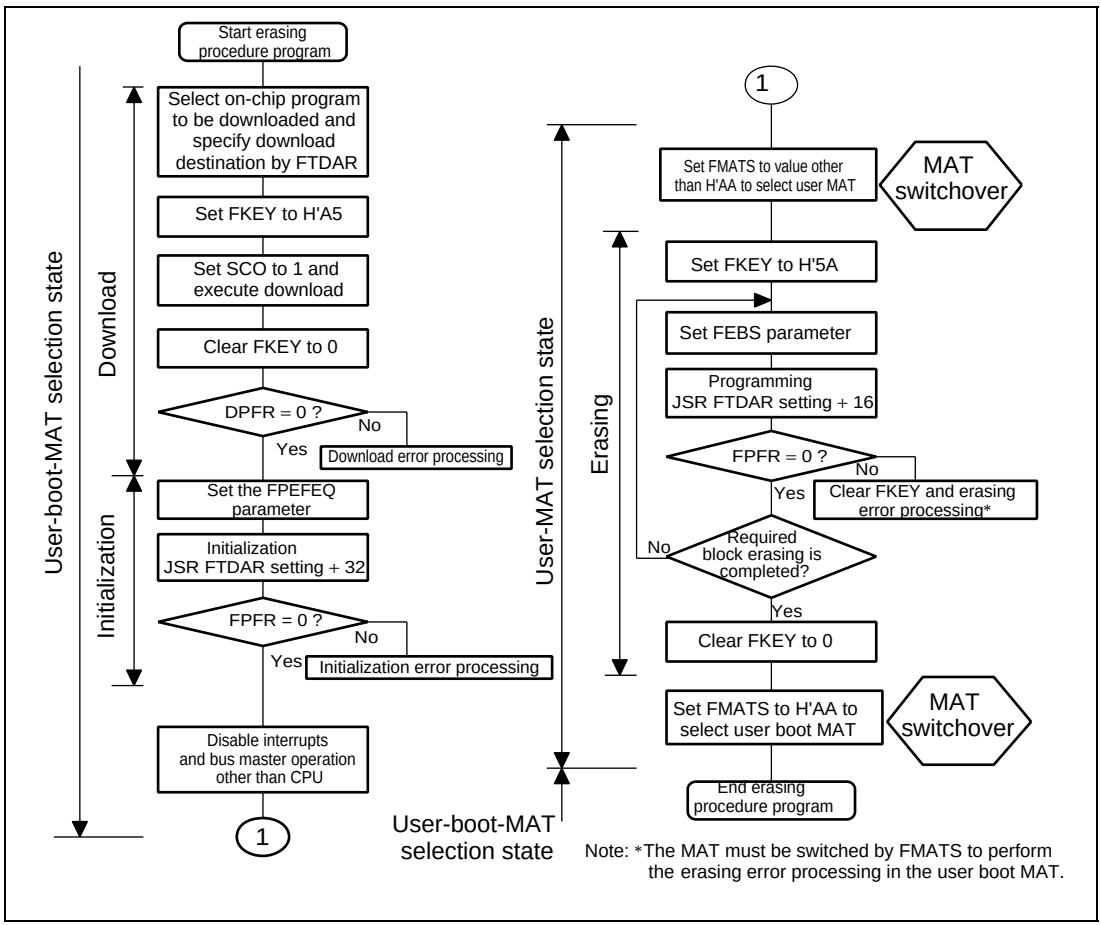


Figure 20.15 Procedure for Erasing User MAT in User Boot Mode

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 20.15.

MAT switching is enabled by writing a specific value to FMATS. However, note 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 20.6, 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, user MAT, and external space) is shown in section 20.4.4, Storable Area for Procedure Program and Program Data.

20.4.4 Storable Area for Procedure Program and Program Data

In the descriptions in the previous section, storable areas for the programming/erasing procedure programs and program data are assumed to be in the on-chip RAM. However, storable areas can be placed in other areas, such as part of flash memory which is not to be programmed or erased, or somewhere in the external address space by using the following conditions.

(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 secured.
3. Download by setting the SCO bit to 1 will lead to switching of the MAT. If, therefore, 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. When in a mode in which the external address space is not accessible, such as single-chip mode, the required procedure programs, NMI handling vector, and NMI handler should be transferred to the on-chip RAM before programming/erasing of the flash memory starts.
5. The flash memory is not accessible during programming/erasing operations, therefore, the operation program downloaded to the on-chip RAM is executed. The NMI-handling vector and processing programs such as that which activate the operation program, and NMI handler should thus be stored in on-chip RAM other than flash memory or the external bus space.
6. After programming/erasing, an access to the flash memory is prohibited until FKEY is cleared. The reset period ($\overline{RES}$ = 0) must be in place for more than 100 µs when the LSI mode is changed to reset on completion of a programming/erasing operation.

Transitions to the reset state, and hardware standby mode are prohibited during programming/erasing. When the reset signal is accidentally input to the chip, a longer period in the reset state than usual (100 µs) is needed before the reset signal is released.

7. Switching of the MATs by FMATS should be needed when programming/erasing of the user MAT is operated in user boot mode. The program which switches the MATs should be executed in the on-chip RAM. See section 20.6, Switching between User MAT and User Boot MAT. Please make sure you know which MAT is selected when switching between them.
8. When the data storable area indicated by programming parameter FMPDR is within the flash memory area, an error will occur even when the data stored is normal. Therefore, the data should be transferred to the on-chip RAM to place the address indicated by FMPDR in an area other than the flash memory.

In consideration of these conditions, there are three sources; operating mode, the bank structure of the user MAT, and operations.

The areas in which the programming data can be stored for execution are shown in tables.

Table 20.7 Executable MAT

	Initiated Mode	
Operation	**User Program Mode**	**User Boot Mode***
Programming	Table 20.8 (1)	Table 20.8 (3)
Erasing	Table 20.8 (2)	Table 20.8 (4)

Note: * Programming/Erasing is possible to user MATs.

Table 20.8 (1) Usable Area for Programming in User Program Mode

	Storable/Executable Area			Selected MAT	
Item	**On-chip RAM**	**User MAT**	**External Space (Extended Mode)**	**User MAT**	**Embedded Program Storage MAT**
Storage area for program data	○	×*	○	—	—
Operation for selection of on-chip program to be downloaded	○	○	○	○	
Operation for writing H'A5 to FKEY	○	○	○	○	
Execution of writing SCO = 1 in FCCS (download)	○	×	×		○
Operation for FKEY clear	○	○	○	○	
Determination of download result	○	○	○	○	
Operation for download error	○	○	○	○	
Operation for settings of initialization parameter	○	○	○	○	
Execution of initialization	○	×	×	○	
Determination of initialization result	○	○	○	○	
Operation for initialization error	○	○	○	○	
NMI handling routine	○	×	○	○	
Operation for disabling interrupt	○	○	○	○	
Operation for writing H'5A to FKEY	○	○	○	○	
Operation for settings of program parameter	○	×	○	○	

	Storable/Executable Area			Selected MAT	
Item	**On-chip RAM**	**User MAT**	**External Space (Extended Mode)**	**User MAT**	**Embedded Program Storage MAT**
Execution of programming	○	×	×	○	
Determination of program result	○	×	○	○	
Operation for program error	○	×	○	○	
Operation for FKEY clear	○	×	○	○	

Note: * Transferring the data to the on-chip RAM enables this area to be used.

Table 20.8 (2) Usable Area for Erasure in User Program Mode

	Storable/Executable Area			Selected MAT	
Item	**On-chip RAM**	**User MAT**	**External Space (Extended Mode)**	**User MAT**	**Embedded Program Storage MAT**
Operation for selection of on-chip program to be downloaded	○	○	○	○	
Operation for writing H'A5 to FKEY	○	○	○	○	
Execution of writing SCO = 1 in FCCS (download)	○	×	×		○
Operation for FKEY clear	○	○	○	○	
Determination of download result	○	○	○	○	
Operation for download error	○	○	○	○	
Operation for settings of initialization parameter	○	○	○	○	
Execution of initialization	○	×	×	○	
Determination of initialization result	○	○	○	○	
Operation for initialization error	○	○	○	○	
NMI handling routine	○	×	○	○	
Operation for disabling interrupt	○	○	○	○	
Operation for writing H′5A to FKEY	○	○	○	○	
Operation for settings of erasure parameter	○	×	○	○	
Execution of erasure	○	×	×	○	
Determination of erasure result	○	×	○	○	

	Storable/Executable Area			Selected MAT	
Item	On-chip RAM	User MAT	External Space (Extended Mode)	User MAT	Embedded Program Storage MAT
Operation for erasure error	○	×	○	○	
Operation for FKEY clear	○	×	○	○	

Table 20.8 (3) Usable Area for Programming in User Boot Mode

	Storable/Executable Area			Selected MAT		
Item	**On-chip RAM**	**User Boot MAT**	**External Space (Expanded Mode)**	**User MAT**	**User Boot MAT**	**Embedded Program Storage MAT**
Storage area for program data	○	×*[1]	○	—	—	—
Operation for selection of on-chip program to be downloaded	○	○	○		○	
Operation for writing H'A5 to FKEY	○	○	○		○	
Execution of writing SCO = 1 in FCCS (download)	○	×	×			○
Operation for FKEY clear	○	○	○		○	
Determination of download result	○	○	○		○	
Operation for download error	○	○	○		○	
Operation for settings of initialization parameter	○	○	○		○	
Execution of initialization	○	×	×		○	
Determination of initialization result	○	○	○		○	
Operation for initialization error	○	○	○		○	
NMI handling routine	○	×	○		○	
Operation for disabling interrupt	○	○	○		○	
Switching MATs by FMATS	○	×	×	○		
Operation for writing H'5A to FKEY	○	×	○	○		

Item	Storable/Executable Area			Selected MAT		
	On-chip RAM	User Boot MAT	External Space (Extended Mode)	User MAT	User Boot MAT	Embedded Program Storage MAT
Operation for settings of program parameter	○	×	○	○		
Execution of programming	○	×	×	○		
Determination of program result	○	×	○	○		
Operation for program error	○	×*2	○	○		
Operation for FKEY clear	○	×	○	○		
Switching MATs by FMATS	○	×	×		○	

Notes: 1. Transferring the data to the on-chip RAM enables this area to be used.
2. Switching FMATS in the on-chip RAM enables this area to be used.

Table 20.8 (4) Usable Area for Erasure in User Boot Mode

	Storable/Executable Area			Selected MAT		
Item	**On-chip RAM**	**User Boot MAT**	**External Space (Extended Mode)**	**User MAT**	**User Boot MAT**	**Embedded Program Storage MAT**
Operation for selection of on-chip program to be downloaded	○	○	○		○	
Operation for writing H'A5 to FKEY	○	○	○		○	
Execution of writing SCO = 1 in FCCS (download)	○	×	×			○
Operation for FKEY clear	○	○	○		○	
Determination of download result	○	○	○		○	
Operation for download error	○	○	○		○	
Operation for settings of initialization parameter	○	○	○		○	
Execution of initialization	○	×	×		○	
Determination of initialization result	○	○	○		○	
Operation for initialization error	○	○	○		○	
NMI handling routine	○	×	○		○	
Operation for disabling interrupt	○	○	○		○	
Switching MATs by FMATS	○	×	×		○	
Operation for writing H'5A to FKEY	○	×	○	○		
Operation for settings of erasure parameter	○	×	○	○		

Item	Storable/Executable Area			Selected MAT		
	On-chip RAM	User Boot MAT	External Space (Extended Mode)	User MAT	User Boot MAT	Embedded Program Storage MAT
Execution of erasure	○	×	×	○		
Determination of erasure result	○	×	○	○		
Operation for erasure error	○	×*	○	○		
Operation for FKEY clear	○	×	○	○		
Switching MATs by FMATS	○	×	×	○		

Note: * Switching FMATS in the on-chip RAM enables this area to be used.

20.5 Protection

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

20.5.1 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, an activated program for programming or erasure cannot program or erase the user MAT, and the error in programming/erasing is reported in the parameter FPFR.

Table 20.9 Hardware Protection

		Function to be Protected	
Item	**Description**	**Download**	**Program/Erase**
FWE pin protection	• When a low level signal is input to the FWE pin, the FWE bit in FCCS is cleared and the program/erase-protected state is entered.	—	○
Reset/standby protection	• The program/erase interface registers are initialized by a reset (including a reset by the WDT) and in hardware standby mode and the program/erase-protected 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 power is initially supplied. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the RES pulse width that is specified in the section on AC characteristics. If a reset is input during programming or erasure, data values in the flash memory are not guaranteed. In this case, execute erasure and then execute programming again.	○	○

20.5.2 Software Protection

Software protection is set up in any of two ways: by disabling the downloading of on-chip programs for programming and erasing and by means of a key code.

Table 20.10 Software Protection

Item	Description	Function to be Protected	
		Download	Program/Erase
Protection by SCO bit	• The program/erase-protected state is entered by clearing the SCO bit in FCCS to 0 which disables the downloading of the programming/erasing programs.	○	○
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 for programming/erasing.	○	○

20.5.3 Error Protection

Error protection is a mechanism for forcibly suspended 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 according to the specified procedures for programming/erasing. Forcibly suspending 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 suspends the programming or erasure.

The FLER bit is set 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

Error protection is canceled by a reset or in hardware standby mode. Note that the reset should be released after the 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.

Figure 20.16 shows state transitions to and from the error-protection state.

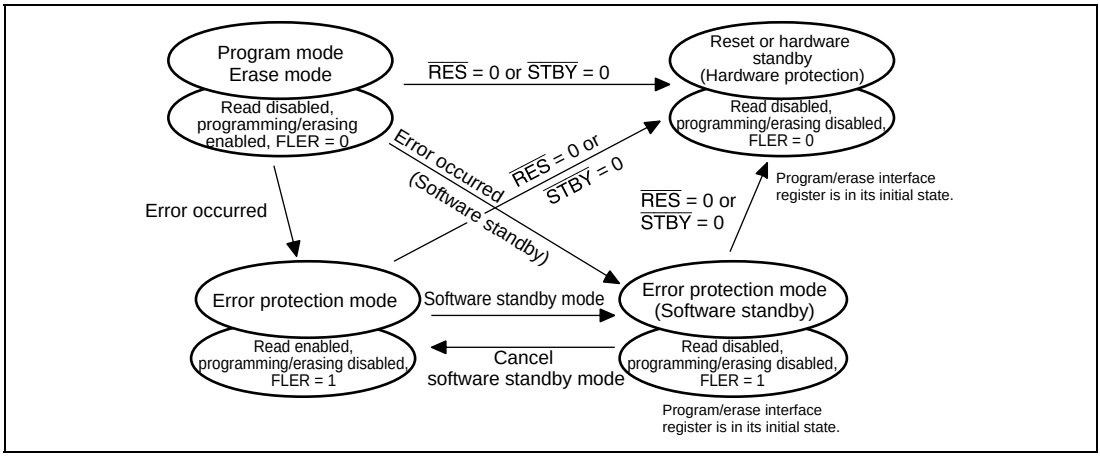


Figure 20.16 Transitions to Error Protection State

20.6 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 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 in the on-chip RAM.
2. To ensure that the MAT that has been switched to is accessible, execute four NOP instructions in the on-chip RAM immediately after writing to FMATS in the on-chip RAM (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 processing is to be the same before and after MAT switching, transfer the interrupt-processing 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. When accessing the user boot MAT, do not access addresses above the top of its 8-kbyte memory space. If an access is made beyond the 8-kbyte space, the undefined value will be read.

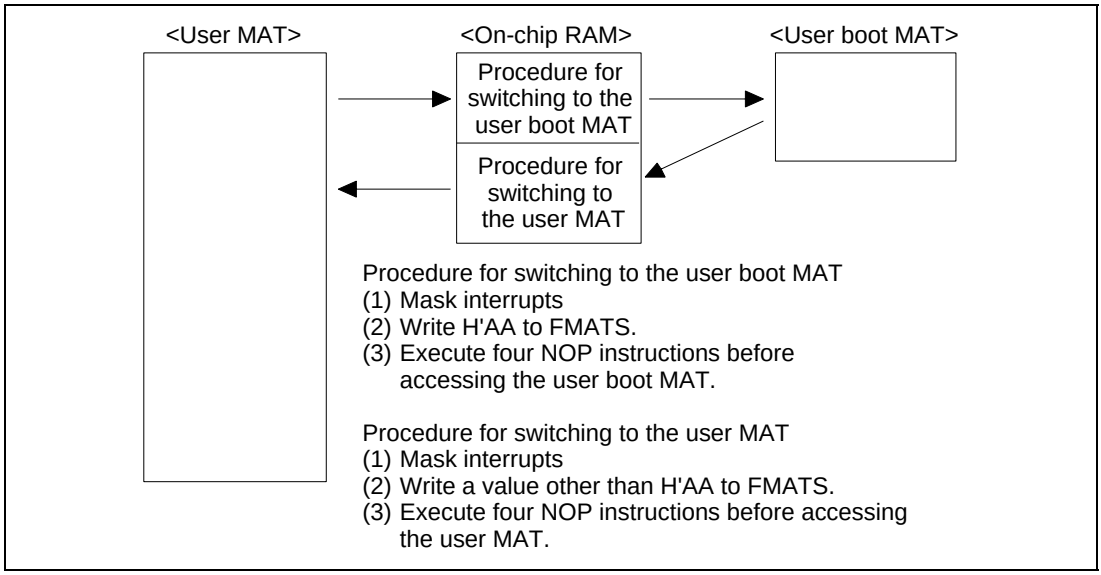


Figure 20.17 Switching between User MAT and User Boot MAT

20.7 Programmer Mode

Along with its on-board programming mode, this LSI also has programmer mode as a further mode for programming and erasing of programs and data. In programmer mode, a general-purpose PROM programmer can freely be used to program programs to the on-chip ROM. Program/erase is possible on the user MAT and user boot MAT. The PROM programmer must support Renesas Technology microcomputers with 256-kbyte flash memory as a device type*. Figure 20.18 shows a memory map in programmer mode.

A status-polling system is adopted for operation in automatic program, automatic erase, and status-read modes. In status-read mode, details of the system's internal signals are output after execution of automatic programming or automatic erasure. In programmer mode, provide a 12-MHz input-clock signal.

Note: * In this LSI, set the programming voltage of the PROM programmer to 3.3 V.

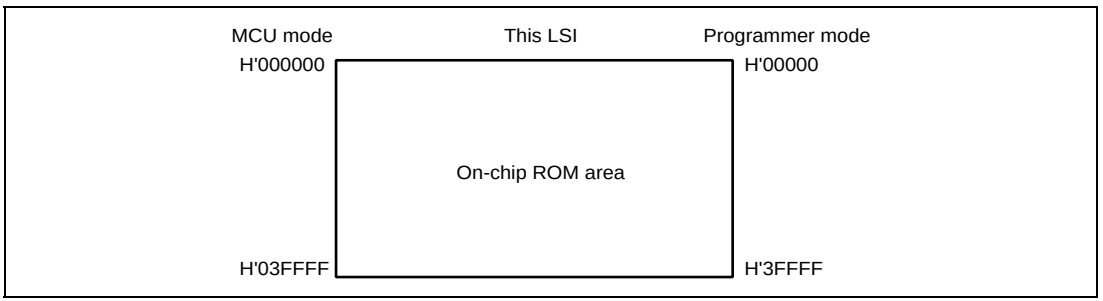


Figure 20.18 Memory Map in Programmer Mode

20.8 Serial Communication Interface Specification for Boot Mode

Initiating boot mode enables the boot program to communicate with the host by using the internal SCI. The serial communication interface specification is 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 entry to the bit-rate-adjustment state. The program receives the command from the host to adjust the bit rate. After adjusting the bit rate, the 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. After selection of these settings, the program is made to enter the programming/erasing state by the command for a transition to the programming/erasing state. The program transfers the libraries required for erasure to the on-chip RAM and erases the user MATs and user boot MATs before the transition.
3. Programming/Erasing State

   Programming and erasure by the boot program take place in this state. The boot program transfers the programming/erasing programs to the RAM by commands from the host to perform programming and erasing. Sum checks and blank checks are executed by sending these commands from the host.

These boot program states are shown in figure 20.19.

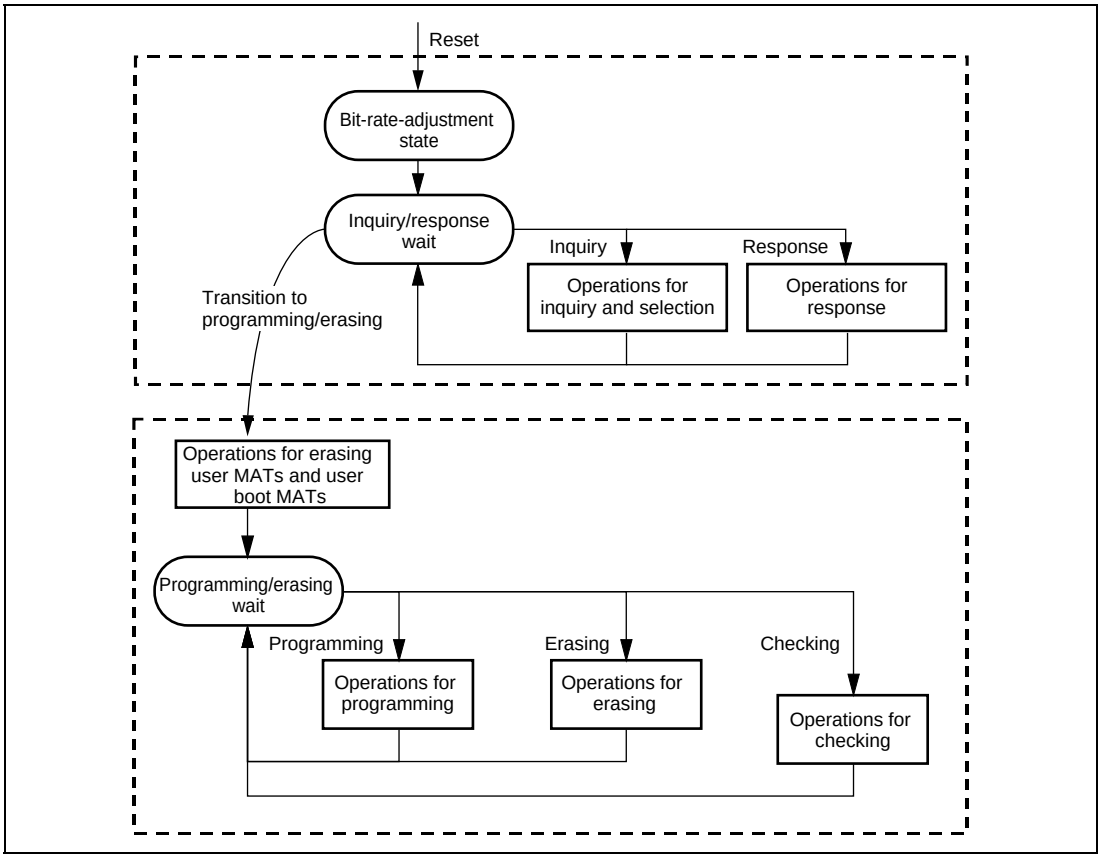


Figure 20.19 Boot Program States

(2) Bit-Rate-Adjustment State

The bit rate is calculated by measuring the period of transfer of a low-level byte (H'00) 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 20.20.

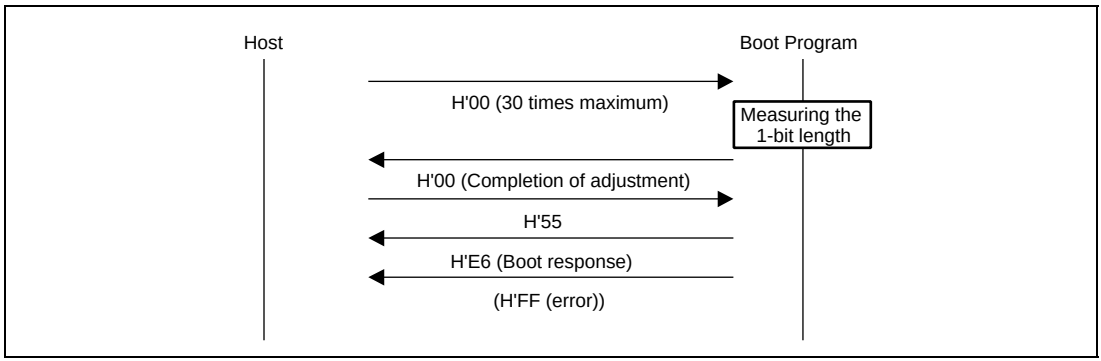


Figure 20.20 Bit-Rate-Adjustment Sequence

(3) Communication Protocol

After adjustment of the bit rate, the protocol for serial communications between the host and the boot program is as shown below.

1. One-Byte Commands and One-Byte Responses

   These commands and responses are comprised of a single byte. These are consists of 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. These are selections and responses to inquiries.

   The amount of programming data is not included under this heading because it is determined in another command.
3. Error Response

   The error response is a response to inquiries. It consists of two bytes: an error response and an error code.
4. Programming of 128 Bytes

   The size is not specified in commands. The size is indicated in response to the programming unit inquiry.
5. Memory Read Response

   This response consists of four bytes.

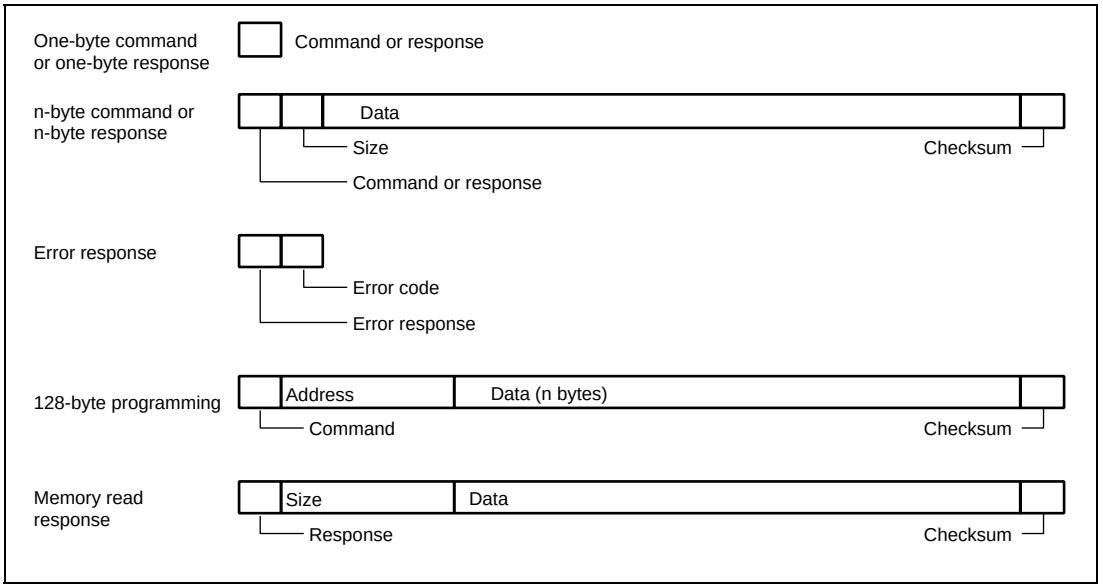


Figure 20.21 Communication Protocol Format

- Command (one byte): Commands including inquiries, selection, programming, erasing, and checking
- Response (one byte): Response to an inquiry
- Size (one byte): The amount of data for transfer excluding the command, amount of data, and checksum
- Checksum (one 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 (one byte): Error response to a command
- Error code (one byte): Type of the error
- Address (four bytes): Address for programming
- Data (n bytes): Data to be programmed (The size is indicated in the response to the programming unit inquiry.)
- Size (four bytes): Four-byte response to a memory read

(4) Inquiry/Selection States

The boot program returns information from the flash memory in response to the host's inquiry commands and selects the device code, clock mode, and bit rate in response to the host's selection command.

Inquiry/selection commands are listed below.

Table 20.11 Inquiry/Selection Commands

Command	Command Name	Description
H'20	Supported device inquiry	Inquiry regarding device codes and product names
H'10	Device selection	Selection of device code
H'21	Clock mode inquiry	Inquiry regarding the number of clock modes and values of each mode
H'11	Clock mode selection	Indication of the selected clock mode
H'22	Multiplication ratio inquiry	Inquiry regarding the number of frequency-multiplied clock types, the number of multiplication ratios, and the values of each multiple
H'23	Operating clock frequency inquiry	Inquiry regarding the maximum and minimum values of the main clock and peripheral clocks
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	Erasing 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 unit of programming data
H'3F	New bit rate selection	Selection of new bit rate
H'40	Transition to programming/erasing state	Erasing of user MAT and user boot MAT, and entry to programming/erasing state
H'4F	Boot program status inquiry	Inquiry regarding the operating state 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 will certainly be needed. When two or more selection commands are transmitted at once, the last command will be valid.

All of these commands, except for the boot program status inquiry command (H'4F), will be valid until the boot program receives the transition to programming/erasing state command (H'40). The host can select the needed commands out of the commands listed above and inquire them. The boot program status inquiry command (H'4F) is valid after the boot program has received the transition to programming/erasing state command (H'40).

(a) Supported Device Inquiry

The boot program will return the device codes of supported devices and the product name in response to the supported device inquiry.

Command

H'20

- Command, H'20, (one byte): Inquiry regarding supported devices

Response

H'30	Size	Number of devices	
Number of characters	Device code		Product name
…			
SUM			

- Response, H'30, (one byte): Response to the supported device inquiry
- Size (one byte): Number of bytes to be transferred, excluding the command, size, and checksum, that is, the amount of data consists of the number of devices, characters, device codes, and product names
- Number of devices (one byte): The number of device types supported by the boot program
- Number of characters (one byte): The number of characters in the device codes and boot program's name
- Device code (four bytes): ASCII code of the supported product name
- Product name (n bytes): Type name of the boot program in ASCII-coded characters
- SUM (one 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 supported device to the specified device code. The program will return the selected device code in response to the inquiry after this setting has been made.

Command

H'10	Size	Device code	SUM

- Command, H'10, (one byte): Device selection
- Size (one byte): Amount of device-code data
  This is fixed to 2

- Device code (four bytes): Device code (ASCII code) returned in response to the supported device inquiry
- SUM (one byte): Checksum

Response

H'06

- Response, H'06, (one byte): Response to the device selection command
  ACK will be returned when the device code matches.

Error response

H'90	ERROR

- Error response, H'90, (one byte): Error response to the device selection command
  ERROR : (one 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

H'21

- Command, H'21, (one byte): Inquiry regarding clock mode

Response

H'31	Size	Number of modes	Mode	…	SUM

- Response, H'31, (one byte): Response to the clock-mode inquiry
- Size (one byte): Amount of data that represents the number of modes and modes
- Number of clock modes (one byte): The number of supported clock modes
  H'00 indicates no clock mode or the device allows to read the clock mode.
- Mode (one byte): Values of the supported clock modes (i.e. H'01 means clock mode 1.)
- SUM (one byte): Checksum

(d) Clock Mode Selection

The boot program will set the specified clock mode. The program will return the selected clock-mode information in response to the inquiry after this setting has been made.

The clock-mode selection command should be transmitted after the device-selection commands.

Command

H'11	Size	Mode	SUM

- Command, H'11, (one byte): Selection of clock mode
- Size (one byte): Amount of data that represents the modes. This is fixed to 1.
- Mode (one byte): A clock mode returned in response to the supported clock mode inquiry.
- SUM (one byte): Checksum

Response

H'06

-

- Response, H'06, (one byte): Response to the clock mode selection command
  ACK will be returned when the clock mode matches.

Error Response	H'91	ERROR

- Error response, H'91, (one byte) : Error response to the clock mode selection command
- ERROR : (one byte): Error code
  H'11: Checksum error
  H'22: Clock mode error, that is, the clock mode does not match.

Even if the clock mode numbers are H'00 and H'01 by a clock mode inquiry, the clock mode must be selected using these respective values.

(e) Multiplication Ratio Inquiry

The boot program will return the supported multiplication and division ratios.

Command	H'22

- Command, H'22, (one byte): Inquiry regarding multiplication ratio

Response	H'32	Size	Number of types					
	Number of multiplication ratios	Multiplica-tion ratio	…					
	…							
	SUM							

- Response, H'32, (one byte): Response to the multiplication ratio inquiry
- Size (one byte): The amount of data that represents the number of clock sources and multiplication ratios and the multiplication ratios
- Number of types (one byte): The number of supported multiplied clock types
  (e.g. when there are two multiplied clock types, which are the main and peripheral clocks, the number of types will be H'02.)
- Number of multiplication ratios (one byte): The number of multiplication ratios for each type
  (e.g. the number of multiplication ratios to which the main clock can be set and the peripheral clock can be set.)
- Multiplication ratio (one byte)

  Multiplication ratio: The value of the multiplication ratio (e.g. when the clock-frequency multiplier is four, the value of multiplication ratio will be H'04.)

  Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the clock is divided by two, the value of division ratio will be H'FE. H'FE = D'−2)

  The number of multiplication ratios returned is the same as the number of multiplication ratios and as many groups of data are returned as there are types.
- SUM (one byte): Checksum

(f) Operating Clock Frequency Inquiry

The boot program will return the number of operating clock frequencies, and the maximum and minimum values.

Command

H'23

- Command, H'23, (one 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, (one byte): Response to operating clock frequency inquiry
- Size (one byte): The number of bytes that represents the minimum values, maximum values, and the number of frequencies.
- Number of operating clock frequencies (one byte): The number of supported operating clock frequency types
  (e.g. when there are two operating clock frequency types, which are the main and peripheral clocks, the number of types will be 2.)
- Minimum value of operating clock frequency (two bytes): The minimum value of the multiplied or divided clock frequency.

  The minimum and maximum values represent the values in MHz, valid to the hundredths place of MHz, and multiplied by 100. (e.g. when the value is 20.00 MHz, it will be 2000, which is H'07D0.)
- Maximum value (two bytes): Maximum value among the multiplied or divided clock frequencies

  There are as many pairs of minimum and maximum values as there are operating clock frequencies.
- SUM (one byte): Checksum

(g) User Boot MAT Information Inquiry

The boot program will return the number of user boot MATs and their addresses.

Command

H'24

- Command, H'24, (one byte): Inquiry regarding user boot MAT information

Response

H'34	Size	Number of areas	
Area-start address			Area-last address
…			
SUM			

- Response, H'34, (one byte): Response to user boot MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start addresses, and area-last address
- Number of Areas (one byte): The number of consecutive user boot MAT areas
  When user boot MAT areas are consecutive, the number of areas returned is H'01.
- Area-start address (four bytes): Start address of the area
- Area-last address (four bytes): Last address of the area
  There are as many groups of data representing the start and last addresses as there are areas.
- SUM (one byte): Checksum

(h) User MAT Information Inquiry

The boot program will return the number of user MATs and their addresses.

Command

H'25

- Command, H'25, (one byte): Inquiry regarding user MAT information

Response

H'35	Size	Number of areas	
Area-start address			Area-last address
…			
SUM			

- Response, H'35, (one byte): Response to the user MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start address and area-last address
- Number of areas (one byte): The number of consecutive user MAT areas
  When the user MAT areas are consecutive, the number of areas is H'01.
- Area-start address (four bytes): Start address of the area
- Area-last address (four bytes): Last address of the area
  There are as many groups of data representing the start and last addresses as there are areas.
- SUM (one byte): Checksum

(i) Erased Block Information Inquiry

The boot program will return the number of erased blocks and their addresses.

Command

H'26

- Command, H'26, (one byte): Inquiry regarding erased block information

Response

H'36	Size	Number of blocks	
Block-start address			Block-last address
…			
SUM			

- Response, H'36, (one byte): Response to the erased block information inquiry
- Size (two bytes): The number of bytes that represents the number of blocks, block-start addresses, and block-last addresses
- Number of blocks (one byte): The number of erased blocks
- Block start address (four bytes): Start address of a block
- Block last Address (four bytes): Last address of a block
  There are as many groups of data representing the start and last addresses as there are blocks.
- SUM (one byte): Checksum

(j) Programming Unit Inquiry

The boot program will return the programming unit used to program data.

Command

H'27

- Command, H'27, (one byte): Inquiry regarding programming unit

Response

H'37	Size	Programming unit	SUM

- Response, H'37, (one byte): Response to programming unit inquiry
- Size (one byte): The number of bytes that indicates the programming unit, which is fixed to 2
- Programming unit (two bytes): A unit for programming
  This is the unit for reception of program data.
- SUM (one byte): Checksum

(k) New Bit-Rate Selection

The boot program will set a new bit rate and return the new bit rate.

This command should be transmitted after the clock mode selection command.

Command

H'3F	Size	Bit rate	Input frequency
Number of multiplication ratios	Multiplication ratio 1	Multiplication ratio 2	
SUM			

- Command, H'3F, (one byte): Selection of new bit rate
- Size (one byte): The number of bytes that represents the bit rate, input frequency, number of multiplication ratios, and multiplication ratio
- Bit rate (two 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 (two bytes): Frequency of the clock input to the boot program
  This is valid to the hundredths place and represents the value in MHz multiplied by 100. (E.g. when the value is 20.00 MHz, it will be 2000, which is H'07D0.)
- Number of multiplication ratios (one byte): The number of multiplication ratios to which the device can be set. (e.g. when there are two operating clock frequency types, which are the main and peripheral clocks, the number of multiplication ratios will be 2.)
- Multiplication ratio 1 (one byte) : The value of multiplication or division ratios for the main operating frequency
  Multiplication ratio: The value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H'04.)
  Division ratio: The inverse of the division ratio, as a negative number (e.g. when the clock frequency is divided by two, the value of division ratio will be H'FE. H'FE = D'–2)
- Multiplication ratio 2 (one byte): The value of multiplication or division ratios for the peripheral frequency
  Multiplication ratio: The value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H′04.)
  Division ratio: The inverse of the division ratio, as a negative number (E.g. when the clock is divided by two, the value of division ratio will be H'FE. H'FE = D'–2)
- SUM (one byte): Checksum

Response

H'06

- Response, H'06, (one byte): Response to selection of a new bit rate
  When it is possible to set the bit rate, ACK will be returned.

Error Response

H'BF	ERROR

- Error response, H'BF, (one byte): Error response to selection of new bit rate
- ERROR: (one byte): Error code

H'11: Checksum error

H'24: Bit-rate selection error

The rate is not available.

H'25: Error in input frequency

This input frequency is not within the specified range.

H'26: Multiplication-ratio error

The ratio does not match an available ratio.

H'27: Operating frequency error

The frequency is not within the specified range.

(5) Received Data Check

The methods for checking of 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 minimum to maximum frequencies which matches the clock modes of the specified device. When the value is out of this range, an input-frequency error is generated.

2. Multiplication Ratio

   The received value of the multiplication ratio or division ratio is checked to ensure that it matches the clock modes of the specified device. When the value is out of this range, a multiplication-ratio error is generated.

3. Operating Frequency

   Operating frequency is calculated from the received value of the input frequency and the multiplication or division ratio. The input frequency is input to the LSI and the LSI is operated at the operating frequency. The expression is given below.

   Operating frequency = Input frequency × Multiplication ratio, or
   Operating frequency = Input frequency ÷ Division ratio

   The calculated operating frequency should be checked to ensure that it is within the range of minimum to maximum frequencies which is 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

   The value (n) of the clock select bit (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 error is generated. The error is calculated using the following expression:

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

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, (one byte): Confirmation of a new bit rate

Response	H'06

- Response, H'06, (one byte): Response to confirmation of a new bit rate

The sequence of new bit-rate selection is shown in figure 20.22.

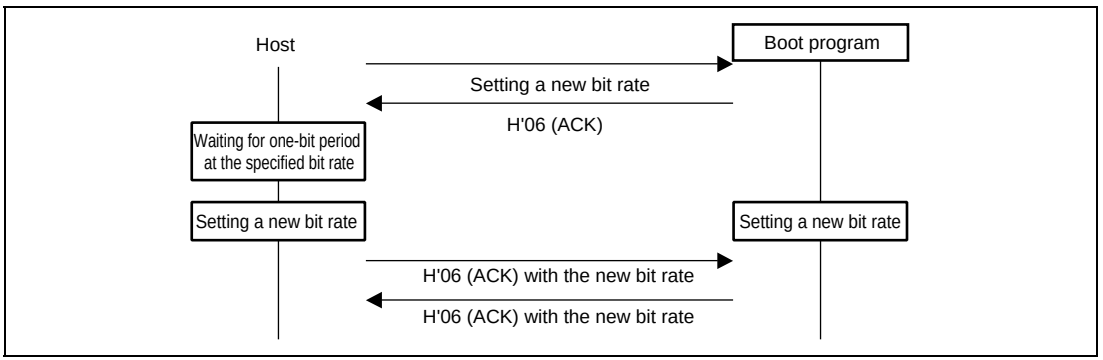


Figure 20.22 New Bit-Rate Selection Sequence

(6) Transition of Programming/Erasing State

The boot program will transfer the erasing program, and erase the user MATs and user boot MATs in that order. On completion of this erasure, ACK will be returned and the programming/erasing state will be entered.

The host should select the device code, clock mode, and new bit rate with device selection, clock-mode selection, and new bit-rate selection commands, and then send the command for the transition to programming/erasing state to the boot program. These procedures should be carried out before sending of the programming selection command or program data.

Command	H'40

- Command, H'40, (one byte): Transition to programming/erasing state

Response	H'06

- Response, H'06, (one byte): Response to transition to programming/erasing state
  ACK will be returned when the user MAT and user boot MAT have been erased by the transferred erasing program.

Error Response	H'C0	H'51

- Error response, H'C0, (one byte): Error response for user boot MAT blank check
- Error code, H'51, (one 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 or an inquiry command after the transition to programming/erasing state command, is an example.

Error Response	H'80	H'xx

- Error response, H'80, (one byte): Command error
- Command, H'xx, (one 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 multiplication-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 multiplication ratios and operating frequencies.
7. After selection of the device and clock mode, inquire the information of the user boot MAT and user MAT with the user boot MATs information inquiry (H'24), user MATs 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

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 below.

Table 20.12 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 the blank data of the user boot MAT
H'4D	User MAT blank check	Checks the blank data of the user MAT
H'4F	Boot program status inquiry	Inquires the boot program's status

- Programming

  Programming is executed by a programming-selection command and a 128-byte programming command.

  Firstly, the host should send the programming-selection command and select the programming method and programming MATs. There are two programming selection commands, and selection is 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 data programmed according to the method specified by the selection command. When more than 128-byte data is programmed, 128-byte commands should be executed repeatedly. Sending 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.

Where the sequence of programming operations that is executed includes programming with another method or of another MAT, the procedure must be repeated from the programming selection command.

The sequence for programming-selection and 128-byte programming commands is shown in figure 20.23.

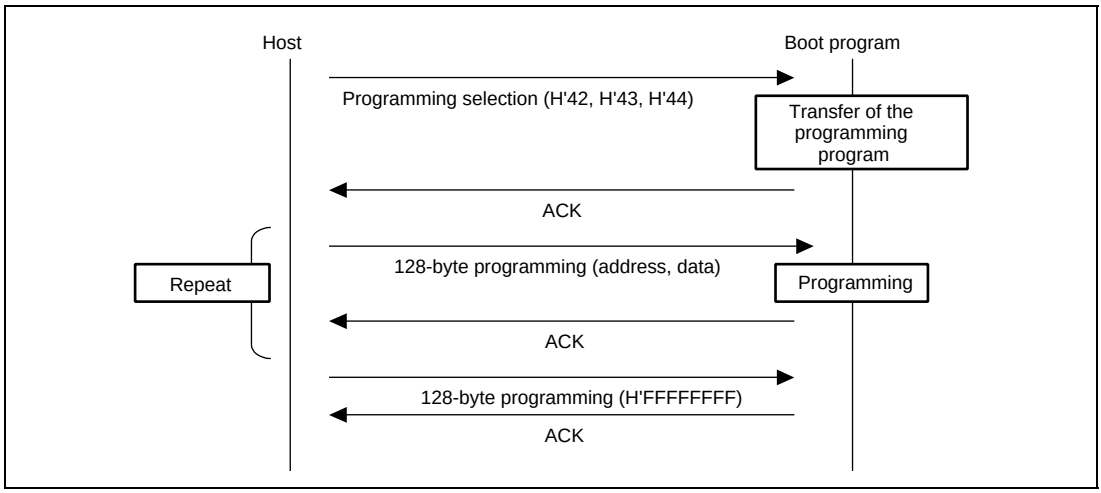


Figure 20.23 Programming Sequence

(a) User Boot MAT Programming Selection

The boot program will transfer a programming program. The data is programmed to the user boot MATs by the transferred programming program.

Command | H'42 |

- Command, H'42, (one byte): User boot MAT programming selection

Response | H'06 |

- Response, H'06, (one byte): Response to user boot MAT programming selection
  When the programming program has been transferred, ACK will be returned.

Error Response | H'C2 | ERROR |

- Error response: H'C2 (one byte): Error response to user boot MAT programming selection
- ERROR: (one 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. The data is programmed to the user MATs by the transferred programming program.

Command

H'43

- Command, H'43, (one byte): User MAT programming selection

Response

H'06

- Response, H'06, (one byte): Response to user MAT programming selection
  When the programming program has been transferred, ACK will be returned.

Error Response

H'C3	ERROR

- Error response: H'C3 (one byte): Error response to user MAT programming selection
- ERROR: (one 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 to program the user boot MATs or user MATs in response to 128-byte programming.

Command

H'50	Address						
Data	…						
…							
SUM							

- Command, H'50, (one byte): 128-byte programming
- Programming address (four bytes): Start address for programming
  Multiple of the size specified in response to the programming unit inquiry
  (i.e. H'00, H'01, H'00, H'00: H'010000)
- Programming data (128 bytes): Data to be programmed
  The size is specified in the response to the programming unit inquiry.
- SUM (one byte): Checksum

Response

H'06

- Response, H'06, (one byte): Response to 128-byte programming
  On completion of programming, ACK will be returned.

Error Response

H'D0	ERROR

- Error response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code
  H'11: Checksum Error
  H'53: Programming error
  A programming error has occurred and programming cannot be continued.

The specified address should match the unit for programming of data. For example, when the programming is in 128-byte units, the lower eight bits of the address should be H'00 or H'80.

When there are less than 128 bytes of data to be programmed, 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 the programming and wait for selection of programming or erasing.

Command	H'50	Address	SUM

- Command, H'50, (one byte): 128-byte programming
- Programming address (four bytes): End code is H'FF, H'FF, H'FF, H'FF.
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to 128-byte programming
  On completion of programming, ACK will be returned.

Error Response	H'D0	ERROR

- Error response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code
  H'11: Checksum 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.

Firstly, erasure is selected by the erasure selection command and the boot program then erases the specified block by a block-erasure command. The block-erasure 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 operating. On completion of erasing, the boot program will wait for selection of programming or erasing.

The sequences of erasure selection commands and block erasure commands are shown in figure 20.24.

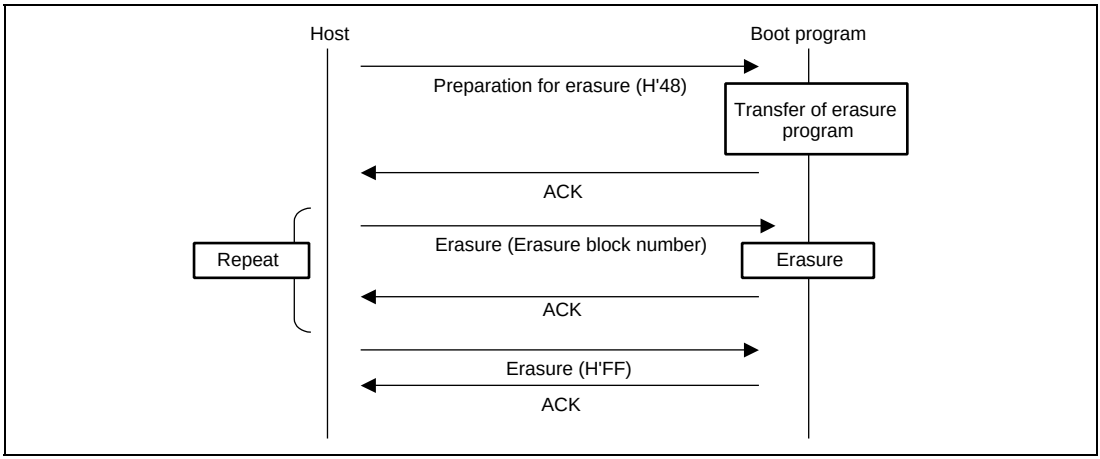


Figure 20.24 Erasure Sequence

(a) Erasure Selection

The boot program will transfer the erasure program. User MAT data is erased by the transferred erasure program.

Command	H'48

- Command, H'48, (one byte): Erasure selection

Response	H'06

- Response, H'06, (one byte): Response for erasure selection
  After the erasure program has been transferred, ACK will be returned.

Error Response	H'C8	ERROR

- Error response, H'C8, (one byte): Error response to erasure selection
- ERROR: (one 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.

Command	H'58	Size	Block number	SUM

- Command, H'58, (one byte): Erasure
- Size (one byte): The number of bytes that represents the erasure block number
  This is fixed to 1.
- Block number (one byte): Number of the block to be erased
- SUM (one byte): Checksum

Response

H'06

- Response, H'06, (one byte): Response to erasure
  After erasure has been completed, ACK will be returned.

Error Response

H'D8	ERROR

- Error response, H'D8, (one byte): Error response to erasure
- ERROR (one byte): Error code

  H'11: Checksum error
  H'29: Block number error
  Block number is incorrect.
  H'51: Erasure 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, (one byte): Erasure
- Size, (one byte): The number of bytes that represents the block number
  This is fixed to 1.
- Block number (one byte): H'FF
  Stop code for erasure
- SUM (one byte): Checksum

Response

H'06

- Response, H'06, (one byte): Response to end of erasure (ACK)

When erasure is to be performed after the block number H'FF has been specified, the procedure should be executed from the erasure selection command.

(11) Memory Read

The boot program will return the data in the specified address.

Command

H'52	Size	Area	Read address	
Read size				SUM

- Command: H'52 (one byte): Memory read
- Size (one byte): Amount of data that represents the area, read address, and read size (fixed to 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 (four bytes): Start address to be read from
- Read size (four bytes): Size of data to be read
- SUM (one byte): Checksum

Response

H'52	Read size						
Data	…						
SUM							

- Response: H'52 (one byte): Response to memory read
- Read size (four bytes): Size of data to be read
- Data (n bytes): Data for the read size from the read address
- SUM (one byte): Checksum

Error Response

H'D2	ERROR

- Error response: H'D2 (one byte): Error response to memory read
- ERROR: (one 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 byte-by-byte total of the contents of the bytes of the user boot MAT, as a four-byte value.

Command

H'4A

- Command, H'4A, (one byte): Sum check for user boot MAT

Response

H'5A	Size	Checksum of MATs	SUM

- Response, H'5A, (one byte): Response to the sum check of user boot MAT
- Size (one byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of MATs (four bytes): Checksum of user boot MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

(13) User MAT Sum Check

The boot program will return the byte-by-byte total of the contents of the bytes of the user MAT, as a four-byte value.

Command	H'4B

- Command, H'4B, (one byte): Sum check for user MAT

Response	H'5B	Size	Checksum of MATs	SUM

- Response, H'5B, (one byte): Response to the sum check of the user MAT
- Size (one byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of MATs (four bytes): Checksum of user MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

(14) User Boot MAT Blank Check

The boot program will check whether or not all user boot MATs are blank and return the result.

Command	H'4C

- Command, H'4C, (one byte): Blank check for user boot MAT

Response	H'06

- Response, H'06, (one byte): Response to the blank check of user boot MAT
  If all user boot MATs are blank (H′FF), ACK will be returned.

Error Response	H'CC	H′52

- Error response, H'CC, (one byte): Error response to blank check for user boot MAT
- Error code, H'52, (one byte): Erasure has not been completed.

(15) User MAT Blank Check

The boot program will check whether or not all user MATs are blank and return the result.

Command	H'4D

- Command, H'4D, (one byte): Blank check for user MATs

Response	H'06

- Response, H'06, (one byte): Response to the blank check for user MATs
  If all user MATs are blank (H'FF), ACK will be returned.

Error Response	H'CD	H'52

- Error Response, H'CD, (one byte): Error response to the blank check of user MATs.
- Error code, H'52, (one byte): Erasure has not been completed.

(16) Boot Program State Inquiry

The boot program will return indications of its present state and error condition. This inquiry can be made in the inquiry/selection state or the programming/erasing state.

Command	H'4F

- Command, H'4F, (one byte): Inquiry regarding boot program's state

Response	H'5F	Size	Status	ERROR	SUM

- Response, H'5F, (one byte): Response to boot program state inquiry
- Size (one byte): The number of bytes. This is fixed to 2.
- Status (one byte): State of the boot program
- ERROR (one byte): Error state

  ERROR = 0 indicates normal operation.
  ERROR = 1 indicates an error has occurred.

- SUM (one byte): Checksum

Table 20.13 Status Codes

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 State for Erasure
H'3F	Programming/Erasing Selection Wait (Erasure is completed)
H'4F	Programming Data Receive Wait (Programming is completed)
H'5F	Erasure Block Specification Wait (Erasure is completed)

Table 20.14 Error Codes

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	Multiplication Ratio Error
H'27	Operating Frequency Error
H'29	Block Number Error
H'2A	Address Error
H'2B	Data Length Error
H'51	Erasure 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

20.9 Usage Notes

1. The initial state of the Renesas Technology product at its shipment is in the erased state. For the product whose revision of erasing is undefined, we recommend to execute automatic erasure for checking the initial state (erased state) and compensating.
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 does not match the specifications, overcurrent 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 the Renesas Technology MCU device with 256-kbyte flash memory at 3.3 V. Do not set the programmer to HN28F101 or the programming voltage to 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 or destroy flash memory permanently. If reset is executed accidentally, reset must be released after the reset input period of 100 µs which is longer than normal.
6. The flash memory is not accessible until FKEY is cleared after programming/erasing completes. If this LSI is restarted by a reset immediately after programming/erasing has finished, secure the reset period (period of $\overline{RES}$ = 0) of more than 100 µs. Though a transition to the reset state or hardware standby state during programming/erasing is prohibited, if reset is executed accidentally, reset must be released after the reset input period of 100 µs which is longer than normal.
7. At powering 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/off timing must also be satisfied at a power-off and power-on caused by a power failure and other factors.
8. Program the area with 128-byte programming-unit blocks in on-board programming or programmer mode only once. Perform programming in the state where the programming-unit block is all erased.
9. When the chip is to be reprogrammed with the programmer after execution of programming or erasure in on-board programming mode, it is recommended that automatic programming be performed after execution of automatic erasure.
10. To program data to the flash memory, data or programs must be allocated to addresses higher than that of the external interrupt vector table (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 (four bytes) is written to the key code area (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 countermeasure is taken for the PROM programmer and its program version.

12. The programming program that includes the initialization routine and the erasing program that includes the initialization routine are each 2 kbytes or less. Accordingly, when the CPU clock frequency is 20 MHz, the download for each program takes approximately TBD μs at the maximum.
13. 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 LSI.
14. 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 the periodic timer interrupts) for the WDT with taking the programming/erasing time into consideration as required.

Section 21 Clock Pulse Generator

This LSI incorporates a clock pulse generator which generates the system clock (ϕ) and internal clock. The clock pulse generator consists of an oscillator, duty adjustment circuit, and divider. Figure 21.1 shows a block diagram of the clock pulse generator.

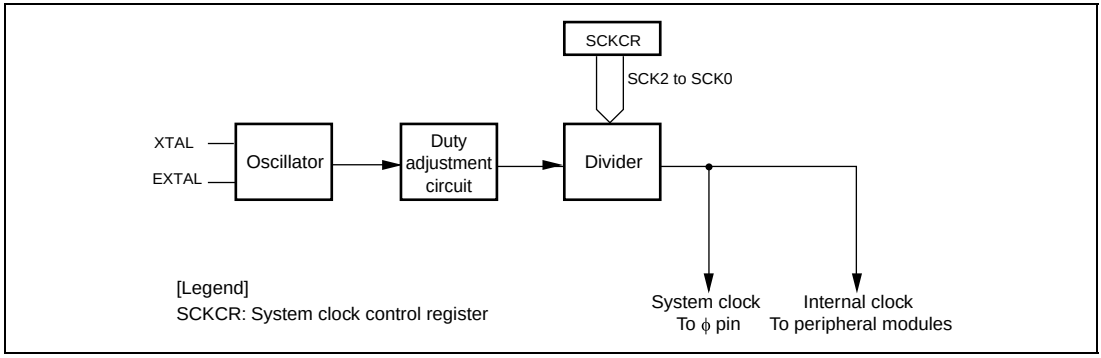


Figure 21.1 Block Diagram of Clock Pulse Generator

The internal frequency is changed by software according to the settings of the system clock control register (SCKCR).

21.1 Register Description

The clock pulse generator has the following register.

- System clock control register (SCKCR)

21.1.1 System Clock Control Register (SCKCR)

SCKCR controls φ output and selects the division ratio for the divider.

Bit	Bit Name	Initial Value	R/W	Description
7	PSTOP	0	R/W	φ Output Disabled Controls φ output. In normal operation: 0: φ output 1: Fixed to high In sleep mode: 0: φ output 1: Fixed to high In software standby mode: 0: Fixed to high 1: Fixed to high In hardware standby mode: 0: High impedance 1: High impedance
6 to 3	—	All 0	R/W	Reserved Although these bits are readable/writable, only 0 should be written here.

Bit	Bit Name	Initial Value	R/W	Description
2	SCK2	0	R/W	System Clock Select 2 to 0
1	SCK1	0	R/W	Select the division ratio.
0	SCK0	0	R/W	000: 1/1
				001: 1/2
				010: 1/4
				011: Setting prohibited
				100: Setting prohibited
				101: Setting prohibited
				11x: Setting prohibited

[Legend] x: Don't care.

21.2 Oscillator

Clock pulses can be supplied either by connecting a crystal resonator or by providing external clock input.

21.2.1 Connecting Crystal Resonator

Figure 21.2 shows a typical method of 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 the equivalent circuit of a crystal resonator. A crystal resonator having the characteristics given in table 21.2 should be used.

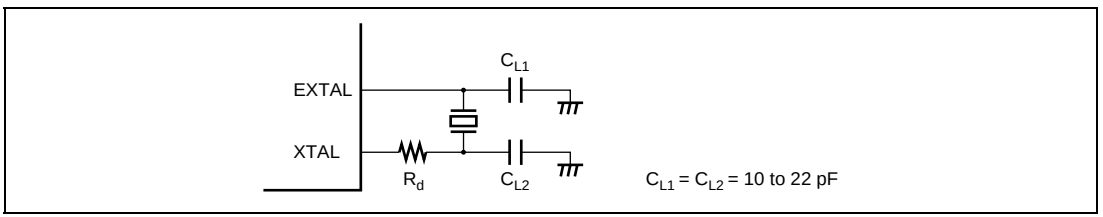


Figure 21.2 Typical Connection to Crystal Resonator

Table 21.1 Damping Resistor Values

Frequency (MHz)	10	12	16	20
R_d (Ω)	0	0	0	0

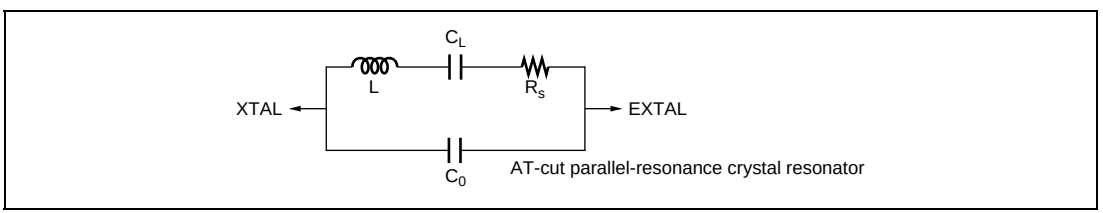


Figure 21.3 Equivalent Circuit of Crystal Resonator

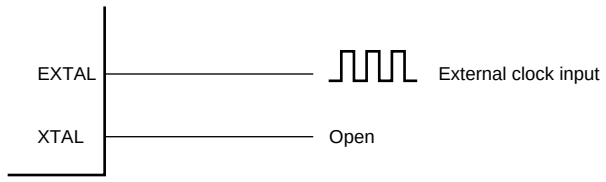
Table 21.2 Crystal Resonator Parameters

Frequency(MHz)	10	12	16	20
R_S (max.) (Ω)	70	60	50	40
C_0 (max.) (pF)	7	7	7	7

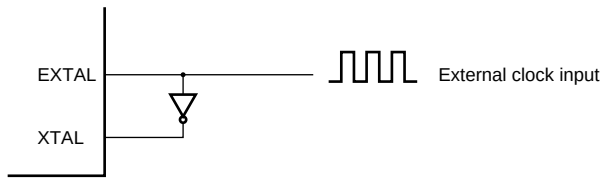
21.2.2 External Clock Input Method

Figure 21.4 shows a typical method of connecting an external clock signal. To leave the XTAL pin open, parasitic 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. External clock input conditions are shown in table 21.3.



(a) Example of external clock input when XTAL pin 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}	15	—	ns	Figure 21.5
External clock input pulse width high level	t_{EXH}	15	—	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 24.3
Clock pulse width high level	t_{CH}	0.4	0.6	t_{cyc}	

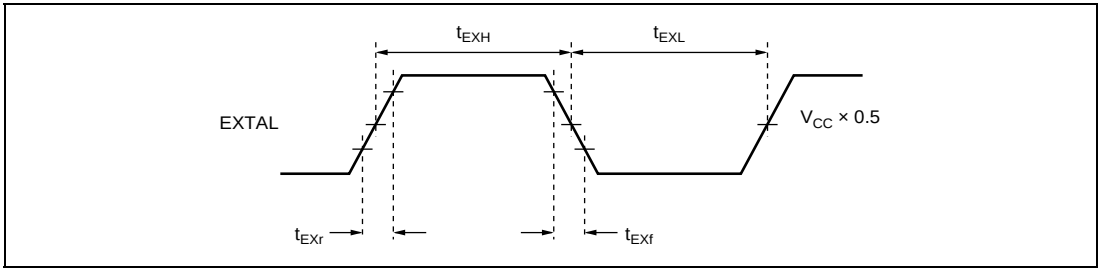


Figure 21.5 External Clock Input Timing

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 hold it in the reset state. Table 21.4 shows the output stabilization delay time for the external clock. Figure 21.6 shows the timing of the output stabilization delay time for the external clock.

Table 21.4 Output Stabilization Delay Time for External Clock

Condition: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{SS} = AV_{SS} = 0 V

Item	Symbol	Min.	Max.	Unit	Remarks
Output stabilization delay time for external clock	t_{DEXT}*	500	—	µs	Figure 21.6

Note: * t_{DEXT} includes a $\overline{RES}$ pulse width (t_{RESW}).

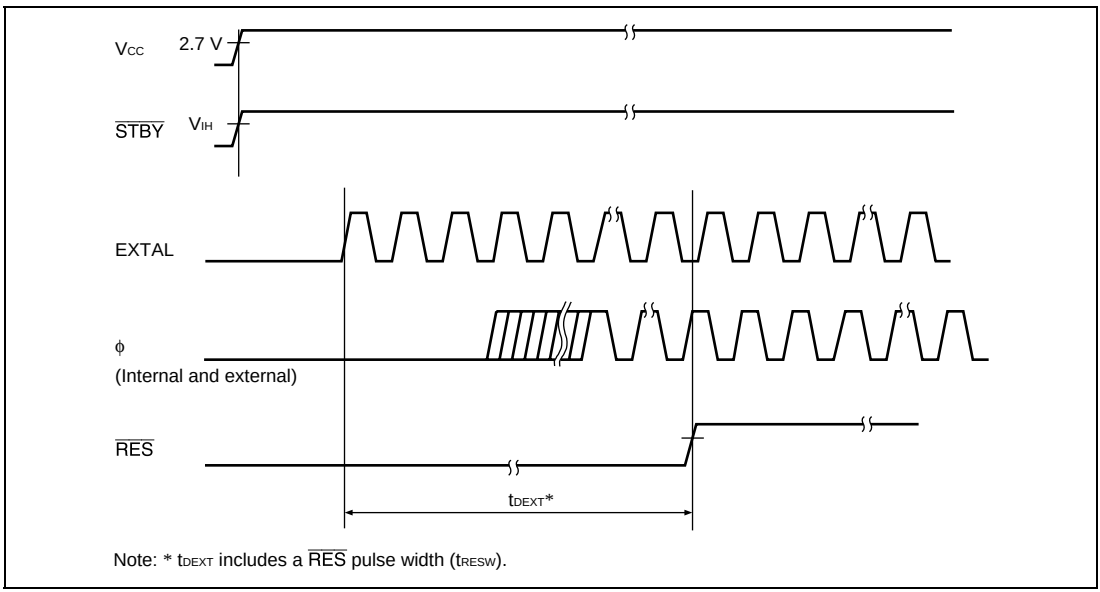


Figure 21.6 Timing of Output Stabilization Delay Time for External Clock

21.3 Duty Adjustment Circuit

The duty adjustment circuit is valid when the oscillation frequency is more than 5 MHz. This circuit adjusts the duty of the clock output by the oscillator and inputs it to the divider.

21.4 Divider

The divider divides the clock output by the duty adjustment circuit, and generates the system clock (ϕ) of 1/1, 1/2, and 1/4.

21.5 Usage Notes

21.5.1 Note 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 which 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.

21.5.2 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 EXTAL and XTAL pins. Other signal lines should be routed away from the oscillation circuit to prevent inductive interference with the correct oscillation as shown in figure 21.7.

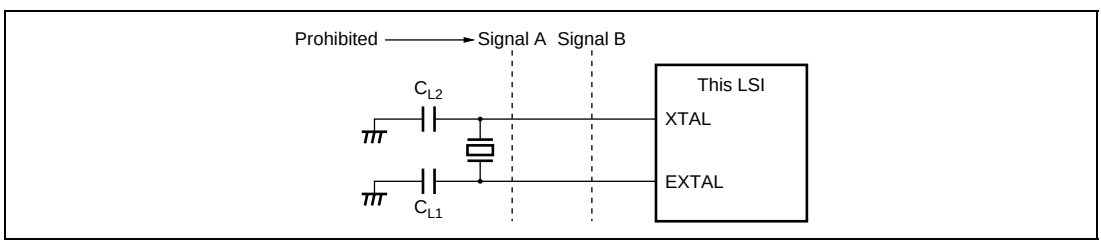


Figure 21.7 Note on Board Design of Oscillation Circuit Section

21.5.3 Notes on Operation Confirmation

Even if a crystal resonator is not connected to the EXTAL and XTAL pins or an external clock is not input, self-oscillation may occur at the several kHz frequency. Therefore, make sure that this LSI operates at the correct frequency.

Section 22 Power-Down Modes

In addition to the normal program execution state, this LSI has power-down modes in which operation of the CPU and oscillator is halted and power consumption is reduced. Low-power operation can be achieved by individually controlling the CPU, on-chip peripheral modules, and so on.

This LSI's operating modes are high-speed mode and five power down modes:

- Clock division mode
- Sleep mode
- Module stop mode
- Software standby mode
- Hardware standby mode

Sleep mode is a CPU state, clock division mode is CPU and on-chip peripheral function states, and module stop mode is an on-chip peripheral function state. A combination of these modes can be set.

After a reset, this LSI is in high-speed mode.

Table 22.1 shows the internal states of this LSI in each mode. Figure 22.1 shows the mode transition diagram.

Table 22.1 Operating Modes and Internal States of LSI

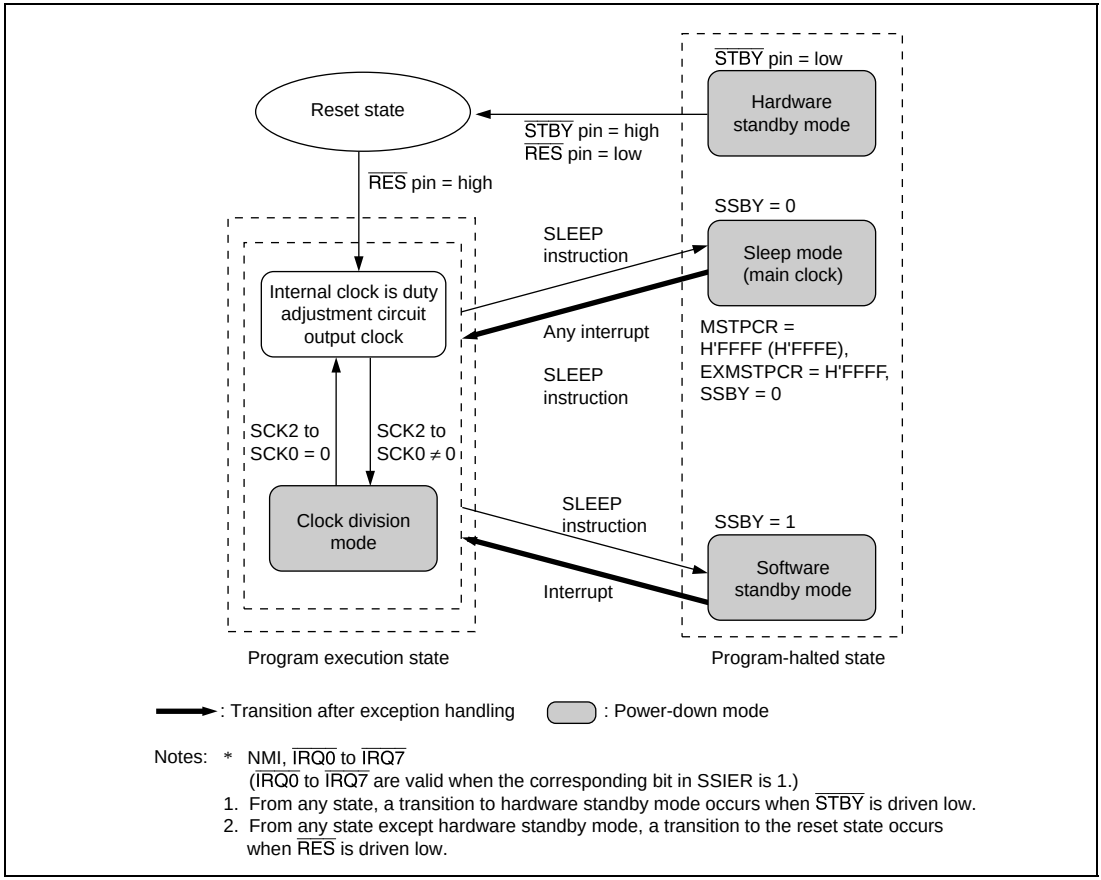
Operating State		High-Speed Mode	Clock Division Mode	Sleep Mode	Module Stop Mode	Software Standby Mode	Hardware Standby Mode
Clock pulse generator		Functions	Functions	Functions	Functions	Halted	Halted
CPU	Instruction execution	Functions	Functions	Halted	Functions	Halted	Halted
	Register			Retained		Retained	Undefined
External interrupts	NMI	Functions	Functions	Functions	Functions	Functions	Halted
	IRQ0 to IRQ7						
Peripheral functions	WDT	Functions	Functions	Functions	Functions	Halted (Retained)	Halted (Reset)
	TMR0, TMR1	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Reset)
	TMRX, TMRY	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Reset)
	FRT	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Reset)
	Timer connection	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Reset)
	Duty measurement circuit	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Reset)
	TPU	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Reset)
	A/D	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Reset)
	IIC3	Functions	Functions	Functions	Halted (Retained)*	Halted (Retained)*	Halted (Reset)
	PWM	Functions	Functions	Functions	Halted (Reset)	Halted (Reset)	Halted (Reset)
	PWMX	Functions	Functions	Functions	Halted (Reset)	Halted (Reset)	Halted (Reset)
	SCI	Functions	Functions	Functions	Halted (Partial reset)	Halted (Partial reset)	Halted (Reset)
	RAM	Functions	Functions	Functions	Functions	Retained	Retained
	I/O	Functions	Functions	Functions	Functions	Retained	High impedance

Notes: Halted (Retained) in the table means that internal register values are retained and internal operations are suspended.

Halted (Reset) in the table means that internal register values and internal states are initialized.

In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).

* The internal register values are retained and internal states are initialized.



Notes: * NMI, IRQ0 to IRQ7
(IRQ0 to IRQ7 are valid when the corresponding bit in SSIER is 1.)

1. From any state, a transition to hardware standby mode occurs when STBY is driven low.
2. From any state except hardware standby mode, a transition to the reset state occurs when RES is driven low.

Figure 22.1 Mode Transitions

22.1 Register Descriptions

The registers relating to the power-down mode are shown below. For details on the system clock control register (SCKCR), refer to section 21.1.1, System Clock Control Register (SCKCR).

- System clock control register (SCKCR)
- Standby control register (SBYCR)
- Module stop control registers H and L (MSTPCRH, MSTPCRL)
- Extension module stop control registers H and L (EXMSTPCRH, EXMSTPCRL)

22.1.1 Standby Control Register (SBYCR)

SBYCR performs software standby mode control.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby Specifies the transition mode after executing the SLEEP instruction. 0: Shifts to sleep mode after the SLEEP instruction is executed 1: Shifts to software standby mode after the SLEEP instruction is executed This bit does not change when clearing software standby mode by using external interrupts and shifting to normal operation. Write 0 to this bit when clearing.
6	OPE	1	R/W	Output Port Enable Specifies whether the states of the address bus and bus control signals ($\overline{CS1}$ to $\overline{CS3}$, $\overline{AS}$/$\overline{AH}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$) are retained or set to the high-impedance state in software standby mode. 0: In software standby mode, address bus and bus control signals are high-impedance 1: In software standby mode, address bus and bus control signals retain the previous states

Bit	Bit Name	Initial Value	R/W	Description
5 to 3	—	All 0	—	Reserved These bits are always read as 0. The initial value should not be changed.
2	STS2	1	R/W	Standby Timer Select 2 to 0 Select the time the MCU waits for the clock to stabilize when software standby mode is cleared. Make a selection according to the operating frequency so that the standby time is at least 8 ms (oscillation stabilization time). With an external clock, make a selection according to the operating frequency so that the standby time is at least 500 µs (output stabilization delay time for external clock). For relationship between setting values and the standby time, see table 22.2.
1	STS1	1	R/W	
0	STS0	1	R/W	

22.1.2 Module Stop Control Registers H and L (MSTPCRH, MSTPCRL)

MSTPCR performs module stop mode control.

Setting a bit to 1, the corresponding module enters module stop mode, while clearing the bit to 0 clears the module stop mode.

- MSTPCRH

Bit	Bit Name	Initial Value	R/W	Module
15	MSTP15	0	R/W	Reserved
14	MSTP14	0	R/W	The initial value should not be changed.
13	MSTP13	0	R/W	
12	MSTP12	0	R/W	
11	MSTP11	1	R/W	Duty measurement circuit (TWM)
10	MSTP10	1	R/W	16-bit timer pulse unit (TPU)
9	MSTP9	1	R/W	A/D converter
8	MSTP8	1	R/W	8-bit PWM timer (PWM), 14-bit PWM timer (PWMX)

- MSTPCRL

Bit	Bit Name	Initial Value	R/W	Module
7	MSTP7	1	R/W	Reserved
6	MSTP6	1	R/W	The initial value should not be changed.
5	MSTP5	1	R/W	16-bit free-running timer (FRT_1)
4	MSTP4	1	R/W	16-bit free-running timer (FRT_0)
3	MSTP3	1	R/W	8-bit timer (TMRX_1, TMRY_1), timer connection 1
2	MSTP2	1	R/W	8-bit timer (TMRX_0, TMRY_0), timer connection 0
1	MSTP1	1	R/W	8-bit timer (TMR0_1, TMR1_1)
0	MSTP0	1	R/W	8-bit timer (TMR0_0, TMR1_0)

22.1.3 Extension Module Stop Control Registers H and L (EXMSTPCRH, EXMSTPCRL)

EXMSTPCR performs module stop mode control.

Setting a bit to 1, the corresponding module enters module stop mode, while clearing the bit to 0 clears the module stop mode.

- EXMSTPCRH

Bit	Bit Name	Initial Value	R/W	Module
15	MSTP31	1	R/W	Reserved
14	MSTP30	1	R/W	The initial value should not be changed.
13	MSTP29	1	R/W	
12	MSTP28	1	R/W	
11	MSTP27	1	R/W	
10	MSTP26	1	R/W	
9	MSTP25	1	R/W	
8	MSTP24	1	R/W	Serial communication interface 4 (SCI_4)

- EXMSTPCRL

Bit	Bit Name	Initial Value	R/W	Module
7	MSTP23	1	R/W	Serial communication interface 3 (SCI_3)
6	MSTP22	1	R/W	Serial communication interface 2 (SCI_2)
5	MSTP21	1	R/W	Serial communication interface 1 (SCI_1)
4	MSTP20	1	R/W	Serial communication interface 0 (SCI_0)
3	MSTP19	1	R/W	I^2C bus interface 3_3 (IIC3_3)
2	MSTP18	1	R/W	I^2C bus interface 3_2 (IIC3_2)
1	MSTP17	1	R/W	I^2C bus interface 3_1 (IIC3_1)
0	MSTP16	1	R/W	I^2C bus interface 3_0 (IIC3_0)

22.2 Operation

22.2.1 Clock Division Mode

When the SCK2 to SCK0 bits in SCKCR are set to a value from B'001 to B'010, a transition is made to clock division mode. In clock division mode, the CPU and on-chip peripheral functions all operate on the operating clock (1/2 or 1/4) specified by bits SCK2 to SCK0.

Clock division mode is cleared by clearing all of bits SCK2 to SCK0 to 0. A transition is made to high-speed mode at the end of the bus cycle, and clock division mode is cleared.

If a SLEEP instruction is executed while the SSBY bit in SBYCR is cleared to 0, the chip enters sleep mode. When sleep mode is cleared by an interrupt, clock division mode is restored.

If a SLEEP instruction is executed while the SSBY bit in SBYCR is set to 1, the chip enters software standby mode. When software standby mode is cleared by an external interrupt, clock division mode is restored.

When the $\overline{\text{RES}}$ pin is driven low, the reset state is entered and clock division mode is cleared. The same applies to a reset caused by a watchdog timer overflow.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

22.2.2 Sleep Mode

Transition to Sleep Mode:

When the SLEEP instruction is executed when the SSBY bit is 0 in SBYCR, the CPU enters sleep mode. In sleep mode, CPU operation stops but the contents of the CPU's internal registers are retained. Other peripheral functions do not stop.

Exiting Sleep Mode:

Sleep mode is exited by any interrupt, or signals at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin.

- Exiting Sleep Mode by Interrupts
  When an interrupt occurs, sleep mode is exited and interrupt exception handling starts. Sleep mode is not exited if the interrupt is disabled, or interrupts other than NMI are masked by the CPU.
- Exiting Sleep Mode by $\overline{\text{RES}}$ pin
  Setting the $\overline{\text{RES}}$ pin level low selects the reset state. After the stipulated reset input duration, driving the $\overline{\text{RES}}$ pin high starts the CPU performing reset exception handling.

- Exiting Sleep Mode by $\overline{\text{STBY}}$ Pin
  When the $\overline{\text{STBY}}$ pin level is driven low, a transition is made to hardware standby mode.

22.2.3 Software Standby Mode

Transition to Software Standby Mode:

If a SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1, software standby mode is entered. In this mode, the CPU, on-chip peripheral functions, and oscillator all stop. However, the contents of the CPU's internal registers, on-chip RAM data, and the states of on-chip peripheral functions other than the PWM, PWMX, SCI, IIC3, and A/D converter, and I/O ports, are retained. Whether the address bus and bus control signals are placed in the high-impedance state or retain the output state can be specified by the OPE bit in SBYCR.

In this mode the oscillator stops, and therefore power consumption is significantly reduced.

Clearing Software Standby Mode:

Software standby mode is cleared by an external interrupt (NMI pin, or pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$), or by means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin. Setting the SSI bit in SSIER to 1 enables $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$ to be used as software standby mode clearing sources.

- Clearing with Interrupt
  When an NMI or IRQ0 to IRQ7 interrupt request signal is input, clock oscillation starts, and after the elapse of the time set in bits STS2 to STS0 in SBYCR, stable clocks are supplied to the entire LSI, 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 and ensure that no interrupt with a higher priority than interrupts IRQ0 to IRQ7 is generated. Software standby mode cannot be cleared if the interrupt has been masked on the CPU side.
- Clearing with $\overline{\text{RES}}$ Pin
  When the $\overline{\text{RES}}$ pin is driven low, clock oscillation is started. At the same time as clock oscillation starts, clocks are supplied to the entire LSI. Note that the $\overline{\text{RES}}$ pin must be held low until clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin goes high, the CPU starts reset exception handling.
- Clearing with $\overline{\text{STBY}}$ Pin
  When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Setting Oscillation Stabilization Time after Clearing Software Standby Mode:

Bits STS2 to STS0 in SBYCR should be set as described below.

- Using Crystal Resonator
  Set bits STS2 to STS0 so that the standby time is more than the oscillation stabilization time.
  Table 22.2 shows the standby times for operating frequencies and settings of bits STS2 to STS0.
- Using External Clock
  The desired value can be set.

Table 22.2 Oscillation Stabilization Time Settings

STS2	STS1	STS0	Standby Time	20 MHz	10 MHz	8 MHz	6 MHz	Unit
0	0	0	8192 states	0.4	0.8	1.0	1.3	ms
		1	16384 states	0.6	1.6	2.0	2.7	
	1	0	32768 states	1.6	3.3	4.1	5.5	
		1	65536 states	3.3	6.6	8.2	10.9	
1	0	0	131072 states	6.6	13.1	16.4	21.8	
		1	262144 states	13.1	26.2	32.8	43.7	
	1	0	Reserved	—	—	—	—	—
		1	16 states*	0.8	1.6	2.0	2.7	µs

: Recommended setting value

Note: * Setting prohibited.

Software Standby Mode Application Example:

Figure 22.2 shows an example in which a transition is made to software standby mode at the falling edge on the NMI pin, and software standby mode is cleared at the rising edge on the NMI pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in INTCR 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 on the NMI pin.

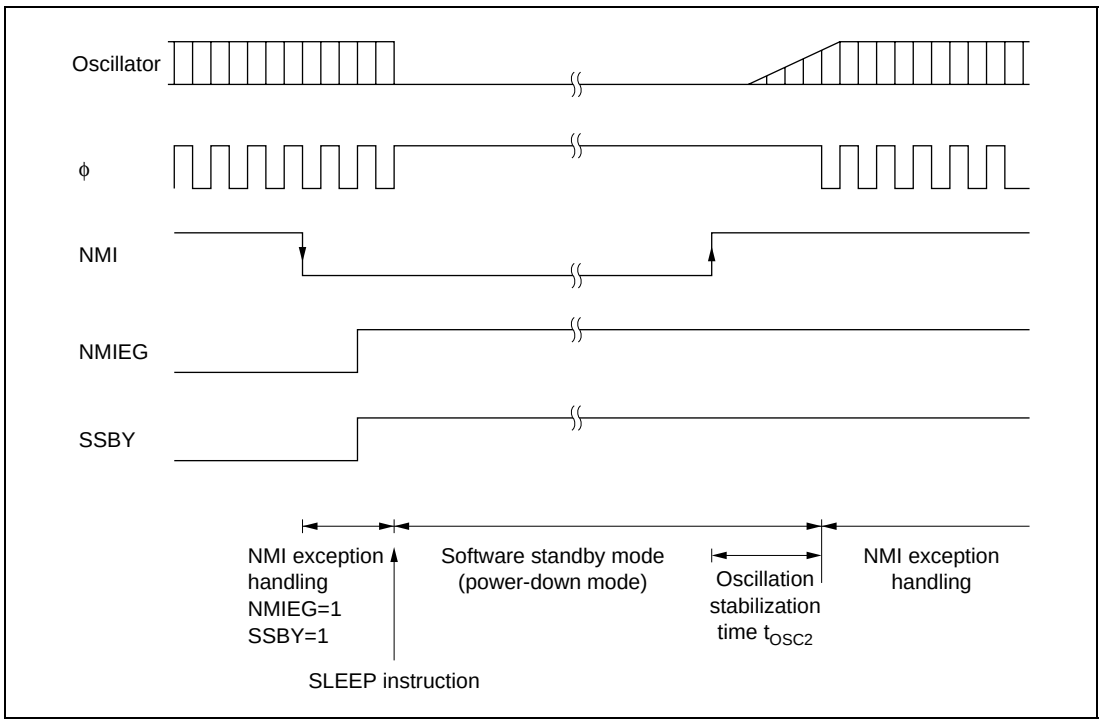


Figure 22.2 Software Standby Mode Application Example

22.2.4 Hardware Standby Mode

Transition to Hardware Standby Mode:

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode from any mode.

In hardware standby mode, all functions enter the reset state and stop operation, resulting in a significant reduction in power consumption. As long as the prescribed voltage is supplied, on-chip RAM data is retained. 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 to MD0) while this LSI is in hardware standby mode.

Note: Do not set hardware standby mode during a reset at power-on.

Clearing Hardware Standby Mode:

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When the $\overline{\text{STBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is set and clock oscillation is started. Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillator stabilizes (for details on the oscillation stabilization time, refer to table 22.2). When the $\overline{\text{RES}}$ pin is subsequently driven high, a transition is made to the program execution state via the reset exception handling state.

Hardware Standby Mode Timing:

Figure 22.3 shows an example of hardware standby mode timing.

When the $\overline{\text{STBY}}$ pin is driven low after the $\overline{\text{RES}}$ pin has been driven low, a transition is made to hardware standby mode. Hardware standby mode is cleared by driving the $\overline{\text{STBY}}$ pin high, waiting for the oscillation stabilization time, then changing the $\overline{\text{RES}}$ pin from low to high.

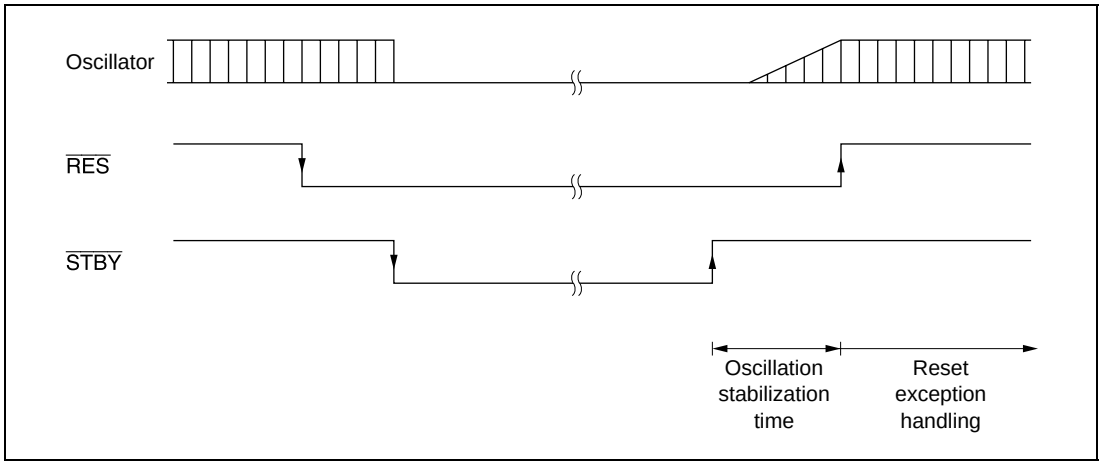


Figure 22.3 Hardware Standby Mode Timing

22.2.5 Module Stop Mode

Module stop mode can be set for individual on-chip peripheral modules.

When the corresponding MSTP bit in MSTPCR or EXMSTPCR is set to 1, module operation stops at the end of the bus cycle and a transition is made to module stop mode. The CPU continues operating independently.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the module starts operating at the end of the bus cycle. In module stop mode, the internal states of modules other than the PWM, PWMX, SCI, and IIC3 are retained.

After reset clearance, all modules are in module stop mode.

The module registers which are set in module stop mode cannot be read or written to.

22.3 φ Clock Output Control

Output of the φ clock can be controlled by means of the PSTOP bit in SCKCR, and DDR for the corresponding port. When the PSTOP bit is set to 1, the φ clock stops at the end of the bus cycle, and φ output goes high. φ clock output is enabled when the PSTOP bit is cleared to 0. When DDR for the corresponding port is cleared to 0, φ clock output is disabled and input port mode is set. Table 22.3 shows the state of the φ pin in each processing state.

Table 22.3 φ Pin State in Each Processing State

Register Setting		Normal Operating State	Sleep Mode	Software Standby Mode	Hardware Standby Mode
DDR	PSTOP				
0	X	High impedance	High impedance	High impedance	High impedance
1	0	φ output	φ output	Fixed high	High impedance
1	1	Fixed high	Fixed high	Fixed high	High impedance

22.4 Usage Notes

22.4.1 I/O Port State

In software standby mode, I/O port states are retained. Therefore, there is no reduction in current consumption for the output current when a high-level signal is output.

22.4.2 Current Consumption during Oscillation Stabilization Standby Period

Current consumption increases during the oscillation stabilization standby period.

22.4.3 On-Chip Peripheral Module Interrupts

Relevant interrupt operations cannot be performed in module stop mode. Consequently, if module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source. Interrupts should therefore be disabled before entering module stop mode.

22.4.4 Writing to MSTPCR, EXMSTPCR

MSTPCR and EXMSTPCR should only be written to by the CPU.

22.4.5 Notes on Clock Division Mode

The following points should be noted in clock division mode.

- Select the clock division ratio specified by the SCK2 to SCK0 bits so that the frequency of ϕ is within the operation guaranteed range of clock cycle time tcyc shown in the Electrical Characteristics. In other words, the range of ϕ must be specified to 5 MHz (min.); outside of this range ($\phi < 5$ MHz) must be prevented.
- All the on-chip peripheral modules operate on ϕ. Therefore, note that the time processing of modules such as a timer and SCI differs before and after changing the clock division ratio. In addition, wait time for clearing software standby mode differs by changing the clock division ratio.
- Note that the clock output of the ϕ pin will be changed by changing the clock division ratio.

Section 23 List of Registers

The register list gives information on the on-chip I/O register addresses, how the register bits are configured, and the register states in each operating mode. The information is given as shown below.

1. Register Addresses (Address Order)

- Registers are listed from the lower allocation addresses.
- The MSB-side address is indicated for 16-bit addresses.
- 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 (Address Order) above.
- 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, so 16-bit registers are shown as 2 lines.

3. Register States in Each Operating Mode

- Register states are described in the same order as the Register Addresses (Address Order) above.
- The register states described here are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, refer to the section on that on-chip peripheral module.

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 Bus Width	Number of Access States
Interrupt priority register A	IPRA	16	H'FD80	Interrupts	16	2
Interrupt priority register B	IPRB	16	H'FD82	Interrupts	16	2
Interrupt priority register C	IPRC	16	H'FD84	Interrupts	16	2
Interrupt priority register D	IPRD	16	H'FD86	Interrupts	16	2
Interrupt priority register E	IPRE	16	H'FD88	Interrupts	16	2
Interrupt priority register F	IPRF	16	H'FD8A	Interrupts	16	2
Interrupt priority register G	IPRG	16	H'FD8C	Interrupts	16	2
Interrupt priority register H	IPRH	16	H'FD8E	Interrupts	16	2
Interrupt priority register I	IPRI	16	H'FD90	Interrupts	16	2
Interrupt priority register J	IPRJ	16	H'FD92	Interrupts	16	2
Interrupt priority register K	IPRK	16	H'FD94	Interrupts	16	2
IRQ sense control register	ISCR	16	H'FD96	Interrupts	16	2
Software standby release IRQ enable register	SSIER	8	H'FD98	Interrupts	16	2
Interrupt control register	INTCR	8	H'FD99	Interrupts	16	2
IRQ enable register	IER	8	H'FD9A	Interrupts	16	2
IRQ status register	ISR	8	H'FD9B	Interrupts	16	2
Serial mode register_0	SMR_0	8	H'FDB0	SCI_0	8	2
Bit rate register_0	BRR_0	8	H'FDB1	SCI_0	8	2
Serial control register_0	SCR_0	8	H'FDB2	SCI_0	8	2
Transmit data register_0	TDR_0	8	H'FDB3	SCI_0	8	2
Serial status register_0	SSR_0	8	H'FDB4	SCI_0	8	2
Receive data register_0	RDR_0	8	H'FDB5	SCI_0	8	2
Serial interface mode register_0	SCMR_0	8	H'FDB6	SCI_0	8	2
Serial mode register_1	SMR_1	8	H'FDB8	SCI_1	8	2
Bit rate register_1	BRR_1	8	H'FDB9	SCI_1	8	2
Serial control register_1	SCR_1	8	H'FDBA	SCI_1	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Bus Width	Number of Access States
Transmit data register_1	TDR_1	8	H'FDBB	SCI_1	8	2
Serial status register_1	SSR_1	8	H'FDBC	SCI_1	8	2
Receive data register_1	RDR_1	8	H'FDBD	SCI_1	8	2
Serial interface mode register_1	SCMR_1	8	H'FDBE	SCI_1	8	2
Serial mode register_2	SMR_2	8	H'FDC0	SCI_2	8	2
Bit rate register_2	BRR_2	8	H'FDC1	SCI_2	8	2
Serial control register_2	SCR_2	8	H'FDC2	SCI_2	8	2
Transmit data register_2	TDR_2	8	H'FDC3	SCI_2	8	2
Serial status register_2	SSR_2	8	H'FDC4	SCI_2	8	2
Receive data register_2	RDR_2	8	H'FDC5	SCI_2	8	2
Serial interface mode register_2	SCMR_2	8	H'FDC6	SCI_2	8	2
Serial mode register_3	SMR_3	8	H'FDC8	SCI_3	8	2
Bit rate register_3	BRR_3	8	H'FDC9	SCI_3	8	2
Serial control register_3	SCR_3	8	H'FDCA	SCI_3	8	2
Transmit data register_3	TDR_3	8	H'FDCB	SCI_3	8	2
Serial status register_3	SSR_3	8	H'FDCC	SCI_3	8	2
Receive data register_3	RDR_3	8	H'FDCD	SCI_3	8	2
Serial interface mode register_3	SCMR_3	8	H'FDCE	SCI_3	8	2
Serial mode register_4	SMR_4	8	H'FDD0	SCI_4	8	2
Bit rate register_4	BRR_4	8	H'FDD1	SCI_4	8	2
Serial control register_4	SCR_4	8	H'FDD2	SCI_4	8	2
Transmit data register_4	TDR_4	8	H'FDD3	SCI_4	8	2
Serial status register_4	SSR_4	8	H'FDD4	SCI_4	8	2
Receive data register_4	RDR_4	8	H'FDD5	SCI_4	8	2
Serial interface mode register_4	SCMR_4	8	H'FDD6	SCI_4	8	2
A/D data register A	ADDRA	16	H'FDE0	A/D	16	2
A/D data register B	ADDRB	16	H'FDE2	A/D	16	2
A/D data register C	ADDRC	16	H'FDE4	A/D	16	2
A/D data register D	ADDRD	16	H'FDE6	A/D	16	2
A/D data register E	ADDRE	16	H'FDE8	A/D	16	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Bus Width	Number of Access States
A/D data register F	ADDRF	16	H'FDEA	A/D	16	2
A/D data register G	ADDRG	16	H'FDEC	A/D	16	2
A/D data register H	ADDRH	16	H'FDEE	A/D	16	2
A/D control/status register	ADCSR	8	H'FDF0	A/D	16	2
A/D control register	ADCR	8	H'FDF1	A/D	16	2
Input capture register	TWICR	8	H'FDF8	Duty measure-ment circuit	16	2
Free-running counter	TWCNT	8	H'FDF9	Duty measure-ment circuit	16	2
Duty measurement control register 1	TWCR1	8	H'FDFA	Duty measure-ment circuit	16	2
Duty measurement control register 2	TWCR2	8	H'FDFB	Duty measure-ment circuit	16	2
Timer interrupt enable register_0	FR_TIER_0*	8	H'FE00	FRT_0	16	2
Timer control/status register_0	TCSR_0	8	H'FE01	FRT_0	16	2
Free-running counter_0	FRC_0	16	H'FE02	FRT_0	16	2
Output compare register A_0	OCRA_0	16	H'FE04	FRT_0	16	2
Output compare register B_0	OCRB_0	16	H'FE04	FRT_0	16	2
Timer control register_0	FR_TCR_0*	8	H'FE06	FRT_0	16	2
Timer output compare control register_0	TOCR_0	8	H'FE07	FRT_0	16	2
Input capture register A_0	ICRA_0	16	H'FE08	FRT_0	16	2
Output compare register AR_0	OCRAR_0	16	H'FE08	FRT_0	16	2
Input capture register B_0	ICRB_0	16	H'FE0A	FRT_0	16	2
Output compare register AF_0	OCRAF_0	16	H'FE0A	FRT_0	16	2
Input capture register C_0	ICRC_0	16	H'FE0C	FRT_0	16	2
Output compare register DM_0	OCRDM_0	16	H'FE0C	FRT_0	16	2
Input capture register D_0	ICRD_0	16	H'FE0E	FRT_0	16	2
Timer control register X_0	TCRX_0	8	H'FE10	TMRX_0	8	2
Timer control/status register X_0	TCSRX_0	8	H'FE11	TMRX_0	8	2
Input capture register R_0	TICRR_0	8	H'FE12	TMRX_0	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Bus Width	Number of Access States
Input capture register F_0	TICRF_0	8	H'FE13	TMRX_0	8	2
Timer counter X_0	TCNTX_0	8	H'FE14	TMRX_0	8	2
Time constant register C_0	TCORC_0	8	H'FE15	TMRX_0	8	2
Timer constant register AX_0	TCORAX_0	8	H'FE16	TMRX_0	8	2
Timer constant register BX_0	TCORBX_0	8	H'FE17	TMRX_0	8	2
Timer control register 0_0	TCR0_0	8	H'FE18	TMR01_0	16	2
Timer control register 1_0	TCR1_0	8	H'FE19	TMR01_0	16	2
Timer control/status register 0_0	TCSR0_0	8	H'FE1A	TMR01_0	16	2
Timer control/status register 1_0	TCSR1_0	8	H'FE1B	TMR01_0	16	2
Time constant register A0_0	TCORA0_0	8	H'FE1C	TMR01_0	16	2
Time constant register A1_0	TCORA1_0	8	H'FE1D	TMR01_0	16	2
Time constant register B0_0	TCORB0_0	8	H'FE1E	TMR01_0	16	2
Time constant register B1_0	TCORB1_0	8	H'FE1F	TMR01_0	16	2
Timer counter 0_0	TCNT0_0	8	H'FE20	TMR01_0	16	2
Timer counter 1_0	TCNT1_0	8	H'FE21	TMR01_0	16	2
Timer connection register I_0	TCONRI_0	8	H'FE24	Timer connection_0	8	2
Timer connection register O_0	TCONRO_0	8	H'FE25	Timer connection_0	8	2
Timer connection register S_0	TCONRS_0	8	H'FE26	Timer connection_0	8	2
Edge sense register_0	SEDGR_0	8	H'FE27	Timer connection_0	8	2
Timer control register Y_0	TCRY_0	8	H'FE28	TMRY_0	8	2
Timer control/status register Y_0	TCSRY_0	8	H'FE29	TMRY_0	8	3
Time constant register AY_0	TCORAY_0	8	H'FE2A	TMRY_0	8	2
Time constant register BY_0	TCORBY_0	8	H'FE2B	TMRY_0	8	2
Timer counter Y_0	TCNTY_0	8	H'FE2C	TMRY_0	8	2
Timer input select register_0	TISR_0	8	H'FE2D	TMRY_0	8	2
Timer interrupt enable register_1	FR_TIER_1*	8	H'FE30	FRT_1	16	2
Timer control/status register_1	TCSR_1	8	H'FE31	FRT_1	16	2
Free-running counter_1	FRC_1	16	H'FE32	FRT_1	16	2
Output compare register A_1	OCRA_1	16	H'FE34	FRT_1	16	2
Output compare register B_1	OCRB_1	16	H'FE34	FRT_1	16	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Bus Width	Number of Access States
Timer control register_1	FR_TCR_1*	8	H'FE36	FRT_1	16	2
Timer output compare control register_1	TOCR_1	8	H'FE37	FRT_1	16	2
Input capture register A_1	ICRA_1	16	H'FE38	FRT_1	16	2
Output compare register AR_1	OCRAR_1	16	H'FE38	FRT_1	16	2
Input capture register B_1	ICRB_1	16	H'FE3A	FRT_1	16	2
Output compare register AF_1	OCRAF_1	16	H'FE3A	FRT_1	16	2
Input capture register C_1	ICRC_1	16	H'FE3C	FRT_1	16	2
Output compare register DM_1	OCRDM_1	16	H'FE3C	FRT_1	16	2
Input capture register D_1	ICRD_1	16	H'FE3E	FRT_1	16	2
Timer control register X_1	TCRX_1	8	H'FE40	TMRX_1	8	2
Timer control/status register X_1	TCSRX_1	8	H'FE41	TMRX_1	8	2
Input capture register R_1	TICRR_1	8	H'FE42	TMRX_1	8	2
Input capture register F_1	TICRF_1	8	H'FE43	TMRX_1	8	2
Timer counter X_1	TCNTX_1	8	H'FE44	TMRX_1	8	2
Time constant register C_1	TCORC_1	8	H'FE45	TMRX_1	8	2
Time constant register AX_1	TCORAX_1	8	H'FE46	TMRX_1	8	2
Time constant register BX_1	TCORBX_1	8	H'FE47	TMRX_1	8	2
Timer control register 0_1	TCR0_1	8	H'FE48	TMR01_1	16	2
Timer control register 1_1	TCR1_1	8	H'FE49	TMR01_1	16	4
Timer control/status register 0_1	TCSR0_1	8	H'FE4A	TMR01_1	16	4
Timer control/status register 1_1	TCSR1_1	8	H'FE4B	TMR01_1	16	2
Time constant register A0_1	TCORA0_1	8	H'FE4C	TMR01_1	16	2
Time constant register A1_1	TCORA1_1	8	H'FE4D	TMR01_1	16	2
Time constant register B0_1	TCORB0_1	8	H'FE4E	TMR01_1	16	2
Time constant register B1_1	TCORB1_1	8	H'FE4F	TMR01_1	16	2
Timer counter 0_1	TCNT0_1	8	H'FE50	TMR01_1	16	2
Timer counter 1_1	TCNT1_1	8	H'FE51	TMR01_1	16	2
Timer connection register I_1	TCONRI_1	8	H'FE54	Timer connection_1	8	2
Timer connection register O_1	TCONRO_1	8	H'FE55	Timer connection_1	8	2
Timer connection register S_1	TCONRS_1	8	H'FE56	Timer connection_1	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Bus Width	Number of Access States
Edge sense register_1	SEDGR_1	8	H'FE57	Timer connection_1	8	2
Timer control register Y_1	TCRY_1	8	H'FE58	TMRY_1	8	2
Timer control/status register Y_1	TCSRY_1	8	H'FE59	TMRY_1	8	2
Time constant register AY_1	TCORAY_1	8	H'FE5A	TMRY_1	8	2
Time constant register BY_1	TCORBY_1	8	H'FE5B	TMRY_1	8	2
Timer counter Y_1	TCNTY_1	8	H'FE5C	TMRY_1	8	2
Timer input select register_1	TISR_1	8	H'FE5D	TMRY_1	8	2
Flash code control/status register	FCCS	8	H'FD9B	Flash memory	8	2
Flash program code select register	FPCS	8	H'FE91	Flash memory	8	2
Flash erase code select register	FECS	8	H'FE92	Flash memory	8	2
Flash key code register	FKEY	8	H'FE94	Flash memory	8	2
Flash MAT select register	FMATS	8	H'FE95	Flash memory	8	2
Flash transfer destination address register	FTDAR	8	H'FE96	Flash memory	8	2
Mode control register	MDCR	8	H'FEB0	SYSTEM	8	2
System control register	SYSCR	8	H'FEB1	SYSTEM	8	2
Standby control register	SBYCR	8	H'FEB2	SYSTEM	8	2
System clock control register	SCKCR	8	H'FEB3	SYSTEM	8	2
Module stop control register H	MSTPCRH	8	H'FEB4	SYSTEM	8	2
Module stop control register L	MSTPCRL	8	H'FEB5	SYSTEM	8	2
Extension module stop control register H	EXMSTPCRH	8	H'FEB6	SYSTEM	8	2
Extension module stop control register L	EXMSTPCRL	8	H'FEB7	SYSTEM	8	2
Timer extended control register	TECR	8	H'FEB8	SYSTEM	8	2
Port register 0	PORT0	8	H'FEC0	PORT	8	2
Port register 1	PORT1	8	H'FEC1	PORT	8	2
Port register 2	PORT2	8	H'FEC2	PORT	8	2
Port register 3	PORT3	8	H'FEC3	PORT	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Bus Width	Number of Access States
Port register 4	PORT4	8	H'FEC4	PORT	8	2
Port register 5	PORT5	8	H'FEC5	PORT	8	2
Port register 6	PORT6	8	H'FEC6	PORT	8	2
Port register 7	PORT7	8	H'FEC7	PORT	8	2
Port register 8	PORT8	8	H'FEC8	PORT	8	2
Port register 9	PORT9	8	H'FEC9	PORT	8	2
Port register A	PORTA	8	H'FECA	PORT	8	2
Port register B	PORTB	8	H'FECB	PORT	8	2
Port register C	PORTC	8	H'FECC	PORT	8	2
Port 1 data register	P1DR	8	H'FED1	PORT	8	2
Port 2 data register	P2DR	8	H'FED2	PORT	8	2
Port 3 data register	P3DR	8	H'FED3	PORT	8	2
Port 4 data register	P4DR	8	H'FED4	PORT	8	2
Port 5 data register	P5DR	8	H'FED5	PORT	8	2
Port 6 data register	P6DR	8	H'FED6	PORT	8	2
Port 8 data register	P8DR	8	H'FED8	PORT	8	2
Port 9 data register	P9DR	8	H'FED9	PORT	8	2
Port A data register	PADR	8	H'FEDA	PORT	8	2
Port B data register	PBDR	8	H'FEDB	PORT	8	2
Port C data register	PCDR	8	H'FEDC	PORT	8	2
Port 1 data direction register	P1DDR	8	H'FEE1	PORT	8	2
Port 2 data direction register	P2DDR	8	H'FEE2	PORT	8	2
Port 3 data direction register	P3DDR	8	H'FEE3	PORT	8	2
Port 4 data direction register	P4DDR	8	H'FEE4	PORT	8	2
Port 5 data direction register	P5DDR	8	H'FEE5	PORT	8	2
Port 6 data direction register	P6DDR	8	H'FEE6	PORT	8	2
Port 8 data direction register	P8DDR	8	H'FEE8	PORT	8	2
Port 9 data direction register	P9DDR	8	H'FEE9	PORT	8	2
Port A data direction register	PADDR	8	H'FEEA	PORT	8	2
Port B data direction register	PBDDR	8	H'FEEB	PORT	8	2
Port C data direction register	PCDDR	8	H'FEEC	PORT	8	2
Port 1 pull-up MOS control register	P1PCR	8	H'FEF0	PORT	8	2
Port 2 pull-up MOS control register	P2PCR	8	H'FEF1	PORT	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Bus Width	Number of Access States
Port 3 pull-up MOS control register	P3PCR	8	H'FEF2	PORT	8	2
Port 6 pull-up MOS control register	P6PCR	8	H'FEF3	PORT	8	2
Port 6 open-drain control register	P6ODR	8	H'FEF4	PORT	8	2
Port function control register	PFCR	8	H'FEF8	PORT	8	2
Port control register 0	PTCNT0	8	H'FEFA	PORT	8	2
Port control register 1	PTCNT1	8	H'FEFB	PORT	8	2
Port control register 2	PTCNT2	8	H'FEFC	PORT	8	2
I^2C bus control register A_0	ICCRA_0	8	H'FF80	IIC3_0	8	2
I^2C bus control register B_0	ICCRB_0	8	H'FF81	IIC3_0	8	2
I^2C bus mode register_0	ICMR_0	8	H'FF82	IIC3_0	8	2
I^2C bus interrupt enable register_0	ICIER_0	8	H'FF83	IIC3_0	8	2
I^2C bus status register_0	ICSR_0	8	H'FF84	IIC3_0	8	2
Slave address register_0	SAR_0	8	H'FF85	IIC3_0	8	2
I^2C transmit data register_0	ICDRT_0	8	H'FF86	IIC3_0	8	2
I^2C receive data register_0	ICDRR_0	8	H'FF87	IIC3_0	8	2
I^2C bus control register A_1	ICCRA_1	8	H'FF88	IIC3_1	8	2
I^2C bus control register B_1	ICCRB_1	8	H'FF89	IIC3_1	8	2
I^2C bus mode register_1	ICMR_1	8	H'FF8A	IIC3_1	8	2
I^2C bus interrupt enable register_1	ICIER_1	8	H'FF8B	IIC3_1	8	2
I^2C bus status register_1	ICSR_1	8	H'FF8C	IIC3_1	8	2
Slave address register_1	SAR_1	8	H'FF8D	IIC3_1	8	2
I^2C transmit data register_1	ICDRT_1	8	H'FF8E	IIC3_1	8	2
I^2C receive data register_1	ICDRR_1	8	H'FF8F	IIC3_1	8	2
I^2C status register A_0	ICSRA_0	8	H'FF90	IIC3_0	8	2
Slave address register A_0	SARA_0	8	H'FF91	IIC3_0	8	2
Slave address register B_0	SARB_0	8	H'FF92	IIC3_0	8	2
Slave address mask register_0	SAMR_0	8	H'FF93	IIC3_0	8	2
I^2C status register A_1	ICSRA_1	8	H'FF94	IIC3_1	8	2
Slave address register A_1	SARA_1	8	H'FF95	IIC3_1	8	2
Slave address register B_1	SARB_1	8	H'FF96	IIC3_1	8	2
Slave address mask register_1	SAMR_1	8	H'FF97	IIC3_1	8	2
I^2C bus control register A_2	ICCRA_2	8	H'FF98	IIC3_2	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Bus Width	Number of Access States
I²C bus control register B_2	ICCRB_2	8	H'FF99	IIC3_2	8	2
I²C bus mode register_2	ICMR_2	8	H'FF9A	IIC3_2	8	2
I²C bus interrupt enable register_2	ICIER_2	8	H'FF9B	IIC3_2	8	2
I²C bus status register_2	ICSR_2	8	H'FF9C	IIC3_2	8	2
Slave address register_2	SAR_2	8	H'FF9D	IIC3_2	8	2
I²C transmit data register_2	ICDRT_2	8	H'FF9E	IIC3_2	8	2
I²C receive data register_2	ICDRR_2	8	H'FF9F	IIC3_2	8	2
I²C bus control register A_3	ICCRA_3	8	H'FFA0	IIC3_3	8	2
I²C bus control register B_3	ICCRB_3	8	H'FFA1	IIC3_3	8	2
I²C bus mode register_3	ICMR_3	8	H'FFA2	IIC3_3	8	2
I²C bus interrupt enable register_3	ICIER_3	8	H'FFA3	IIC3_3	8	2
I²C bus status register_3	ICSR_3	8	H'FFA4	IIC3_3	8	2
Slave address register_3	SAR_3	8	H'FFA5	IIC3_3	8	2
I²C transmit data register_3	ICDRT_3	8	H'FFA6	IIC3_3	8	2
I²C receive data register_3	ICDRR_3	8	H'FFA7	IIC3_3	8	2
I²C status register A_2	ICSRA_2	8	H'FFA8	IIC3_2	8	2
Slave address register A_2	SARA_2	8	H'FFA9	IIC3_2	8	2
Slave address register B_2	SARB_2	8	H'FFAA	IIC3_2	8	2
Slave address mask register_2	SAMR_2	8	H'FFAB	IIC3_2	8	2
I²C status register A_3	ICSRA_3	8	H'FFAC	IIC3_3	8	2
Slave address register A_3	SARA_3	8	H'FFAD	IIC3_3	8	2
Slave address register B_3	SARB_3	8	H'FFAE	IIC3_3	8	2
Slave address mask register_3	SAMR_3	8	H'FFAF	IIC3_3	8	2
Bus control register	BCR	8	H'FFB0	BSC	8	2
Area control register A1	BCRA1	8	H'FFB1	BSC	8	2
Area control register A2	BCRA2	8	H'FFB2	BSC	8	2
Area control register A3	BCRA3	8	H'FFB3	BSC	8	2
Timer counter	TCNT	16	H'FFBC (Write)	WDT	16	2
Timer counter	TCNT	8	H'FFBD (Read)	WDT	16	2
Timer control/status register	TCSR	16	H'FFBC (Write)	WDT	16	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Bus Width	Number of Access States
Timer control/status register	TCSR	8	H'FFBC (Read)	WDT	16	2
Peripheral clock select register	PCSR	8	H'FFC0	PWM	8	2
PWM output enable register	PWOER	8	H'FFC3	PWM	8	2
PWM data polarity register	PWDPR	8	H'FFC5	PWM	8	2
PWM register select	PWSL	8	H'FFC6	PWM	8	2
PWM data registers 7 to 0	PWDR7 to PWDR0	8	H'FFC7	PWM	8	2
PWMX (D/A) data register A	DADRA	16	H'FFC8	PWMX	8	4
PWMX (D/A) control register	DACR	8	H'FFC8	PWMX	8	2
PWMX (D/A) data register B	DADRB	16	H'FFCA	PWMX	8	4
PWMX (D/A) counter H	DACNTH	8	H'FFCA	PWMX	8	4
PWMX (D/A) counter L	DACNTL	8	H'FFCA	PWMX	8	4
Timer start register	TSTR	8	H'FFCC	TPU common	16	2
Timer synchro register	TSYR	8	H'FFCD	TPU common	16	2
Timer control register_0	TCR_0	8	H'FFD0	TPU_0	16	2
Timer mode register_0	TMDR_0	8	H'FFD1	TPU_0	16	2
Timer I/O control register H_0	TIORH_0	8	H'FFD2	TPU_0	16	2
Timer I/O control register L_0	TIORL_0	8	H'FFD3	TPU_0	16	2
Timer interrupt enable register_0	TIER_0	8	H'FFD4	TPU_0	16	2
Timer status register_0	TSR_0	8	H'FFD5	TPU_0	16	2
Timer counter_0	TCNT_0	16	H'FFD6	TPU_0	16	2
Timer general register A_0	TGRA_0	16	H'FFD8	TPU_0	16	2
Timer general register B_0	TGRB_0	16	H'FFDA	TPU_0	16	2
Timer general register C_0	TGRC_0	16	H'FFDC	TPU_0	16	2
Timer general register D_0	TGRD_0	16	H'FFDE	TPU_0	16	2
Timer control register_1	TCR_1	8	H'FFE0	TPU_1	16	2
Timer mode register_1	TMDR_1	8	H'FFE1	TPU_1	16	2
Timer I/O control register_1	TIOR_1	8	H'FFE2	TPU_1	16	2
Timer interrupt enable register_1	TIER_1	8	H'FFE4	TPU_1	16	2
Timer status register_1	TSR_1	8	H'FFE5	TPU_1	16	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Bus Width	Number of Access States
Timer counter_1	TCNT_1	16	H'FFE6	TPU_1	16	2
Timer general register A_1	TGRA_1	16	H'FFE8	TPU_1	16	2
Timer general register B_1	TGRB_1	16	H'FFEA	TPU_1	16	2
Timer control register_2	TCR_2	8	H'FFF0	TPU_2	16	2
Timer mode register_2	TMDR_2	8	H'FFF1	TPU_2	16	2
Timer I/O control register_2	TIOR_2	8	H'FFF2	TPU_2	16	2
Timer interrupt enable register_2	TIER_2	8	H'FFF4	TPU_2	16	2
Timer status register_2	TSR_2	8	H'FFF5	TPU_2	16	2
Timer counter_2	TCNT_2	16	H'FFF6	TPU_2	16	2
Timer general register A_2	TGRA_2	16	H'FFF8	TPU_2	16	2
Timer general register B_2	TGRB_2	16	H'FFFA	TPU_2	16	2

Note: * To classify the same name registers of the other timers, “FR” are added to the abbreviations.

23.2 Register Bits

Register addresses and bit names of the on-chip peripheral modules are described below.

Each line covers eight bits, so 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
IPRA	—	IPRA14	IPRA13	IPRA12	—	IPRA10	IPRA9	IPRA8	Interrupt
	—	IPRA6	IPRA5	IPRA4	—	IPRA2	IPRA1	IPRA0	
IPRB	—	IPRB14	IPRB13	IPRB12	—	IPRB10	IPRB9	IPRB8	
	—	IPRB6	IPRB5	IPRB4	—	IPRB2	IPRB1	IPRB0	
IPRC	—	IPRC14	IPRC13	IPRC12	—	IPRC10	IPRC9	IPRC8	
	—	IPRC6	IPRC5	IPRC4	—	IPRC2	IPRC1	IPRC0	
IPRD	—	IPRD14	IPRD13	IPRD12	—	IPRD10	IPRD9	IPRD8	
	—	IPRD6	IPRD5	IPRD4	—	IPRD2	IPRD1	IPRD0	
IPRE	—	IPRE14	IPRE13	IPRE12	—	IPRE10	IPRE9	IPRE8	
	—	IPRE6	IPRE5	IPRE4	—	IPRE2	IPRE1	IPRE0	
IPRF	—	IPRF14	IPRF13	IPRF12	—	IPRF10	IPRF9	IPRF8	
	—	IPRF6	IPRF5	IPRF4	—	IPRF2	IPRF1	IPRF0	
IPRG	—	IPRG14	IPRG13	IPRG12	—	IPRG10	IPRG9	IPRG8	
	—	IPRG6	IPRG5	IPRG4	—	IPRG2	IPRG1	IPRG0	
IPRH	—	IPRH14	IPRH13	IPRH12	—	IPRH10	IPRH9	IPRH8	
	—	IPRH6	IPRH5	IPRH4	—	IPRH2	IPRH1	IPRH0	
IPRI	—	IPRI14	IPRI13	IPRI12	—	IPRI10	IPRI9	IPRI8	
	—	IPRI6	IPRI5	IPRI4	—	IPRI2	IPRI1	IPRI0	
IPRJ	—	IPRJ14	IPRJ13	IPRJ12	—	IPRJ10	IPRJ9	IPRJ8	
	—	IPRJ6	IPRJ5	IPRJ4	—	IPRJ2	IPRJ1	IPRJ0	
IPRK	—	IPRK14	IPRK13	IPRK12	—	IPRK10	IPRK9	IPRK8	
	—	IPRK6	IPRK5	IPRK4	—	IPRK2	IPRK1	IPRK0	
ISCR	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	
	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA	
SSIER	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1	SSI0	
INTCR	—	—	INTM1	INTM0	NMIEG	—	—	—	
IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
ISR	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SMR_0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_0
BRR_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SSR_0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
RDR_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SCMR_0	—	—	—	—	SDIR	SINV	—	—	
SMR_1	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_1
BRR_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SSR_1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
RDR_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SCMR_1	—	—	—	—	SDIR	SINV	—	—	
SMR_2	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_2
BRR_2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SSR_2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
RDR_2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SCMR_2	—	—	—	—	SDIR	SINV	—	—	
SMR_3	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_3
BRR_3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SCR_3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SSR_3	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
RDR_3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SCMR_3	—	—	—	—	SDIR	SINV	—	—	
SMR_4	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_4
BRR_4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SCR_4	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SSR_4	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
RDR_4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	SCI_4
SCMR_4	—	—	—	—	SDIR	SINV	—	—	
ADDRA	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D converter
	AD1	AD0	—	—	—	—	—	—	
ADDRB	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRC	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRD	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRE	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRF	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRG	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRG	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0	
ADCR	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	—	—	
TWICR	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Duty measurement circuit
TWCNT	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWCR1	FRC	—	CKS2	CKS1	CKS0	IS2	IS1	IS0	
TWCR2	ENDIE	OVIE	ENDF	OVF	—	—	—	START	
FR_TIER_0*	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—	FRT_0
TCSR_0	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA	
FRC_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
OCRA_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
OCRB_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	FRT_0
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FR_TCR_0*	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0	
TOCR_0	ICRDMS	OCRAMS	ICRS	OCRS	OEA	OEB	OLVLA	OLVLB	
ICRA_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
OCRAR_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ICRB_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
OCRAF_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ICRC_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
OCRDM_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ICRD_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCRX_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMRX_0
TCSRX_0	CMFB	CMFA	OVF	ICF	OS3	OS2	OS1	OS0	
TICRR_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TICRF_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCNTX_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORC_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORAX_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORBX_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCR0_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR01_0
TCR1_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	
TCSR0_0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0	
TCSR1_0	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
TCORA0_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORA1_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORB0_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCORB1_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	TMR01_0
TCNT0_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCNT1_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCONRI_0	SIMOD1	SIMOD0	SCONE	ICST	HFINV	VFINV	HIINV	VIINV	Timer connection_0
TCONRO_0	HOE	VOE	CLOE	CBOE	HOINV	VOINV	CLOINV	CBOINV	
TCONRS_0	—	ISGENE	HOMOD1	HOMOD0	VOMOD1	VOMOD0	CLMOD1	CLMOD0	
SEDGR_0	VEDG	HEDG	CEDG	HFEDG	VFEDG	PREQF	IHI	IVI	
TCRY_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMRY_0
TCSRY_0	CMFB	CMFA	OVF	ICIE	OS3	OS2	OS1	OS0	
TCORAY_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORBY_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCNTY_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TISR_0	—	—	—	—	—	—	—	IS	
FR_TIER_1*	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—	FRT_1
TCSR_1	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA	
FRC_1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
OCRA_1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
OCRB_1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FR_TCR_1*	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0	
TOCR_1	ICRDMS	OCRAMS	ICRS	OCRS	OEA	OEB	OLVLA	OLVLB	
ICRA_1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
OCRAR_1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ICRB_1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
OCRAF_1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ICRC_1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
OCRDM_1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	FRT_1
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ICRD_1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCRX_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMRX_1
TCSRX_1	CMFB	CMFA	OVF	ICF	OS3	OS2	OS1	OS0	
TICRR_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TICRF_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCNTX_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORC_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORAX_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORBX_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCR0_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR01_1
TCR1_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	
TCSR0_1	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0	
TCSR1_1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
TCORA0_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORA1_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	TMR01_1
TCORB0_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORB1_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCNT0_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCNT1_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCONRI_1	SIMOD1	SIMOD0	SCONE	ICST	—	—	HIINV	VIINV	Timer connection_1
TCONRO_1	—	—	—	—	HOINV	VOINV	—	—	
TCONRS_1	TMRX/Y	ISGENE	HOMOD1	HOMOD0	VOMOD1	VOMOD0	CLMOD1	CLMOD0	
SEDGR_1	VEDG	HEDG	CEDG	—	—	PREDG	IHI	IVI	
TCRY_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMRY_1
TCSRY_1	CMFB	CMFA	OVF	ICIE	OS3	OS2	OS1	OS0	
TCORAY_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORBY_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCNTY_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TISR_1	—	—	—	—	—	—	—	IS	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
FCCS	FWE	—	—	FLER	WEINTE	—	—	SCO	Flash memory
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	
MDCR	EXPE	—	—	—	—	MDS2	MDS1	MDS0	System
SYSCR	MACS	—	—	—	XRST	FLASHE	—	RAME	
SBYCR	SSBY	OPE	—	—	—	STS2	STS1	STS0	
SCKCR	PSTOP	—	—	—	—	SCK2	SCK1	SCK0	
MSTPCRH	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	
MSTPCRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0	
EXMSTPCRH	MSTP31	MSTP30	MSTP29	MSTP28	MSTP27	MSTP26	MSTP25	MSTP24	
EXMSTPCRL	MSTP23	MSTP22	MSTP21	MSTP20	MSTP19	MSTP18	MSTP17	MSTP16	
TECR	VS0	HS2	HS1	HS0	ICKS1_1	ICKS0_1	ICKS1_0	ICKS0_0	
PORT0	P07	P06	P05	P04	P03	P02	P01	P00	Port
PORT1	P17	P16	P15	P14	P13	P12	P11	P10	
PORT2	P27	P26	P25	P24	P23	P22	P21	P20	
PORT3	P37	P36	P35	P34	P33	P32	P31	P30	
PORT4	P47	P46	P45	P44	P43	P42	P41	P40	
PORT5	P57	P56	P55	P54	P53	P52	P51	P50	
PORT6	P67	P66	P65	P64	P63	P62	P61	P60	
PORT7	P77	P76	P75	P74	P73	P72	P71	P70	
PORT8	P87	P86	P85	P84	P83	P82	P81	P80	
PORT9	P97	P96	P95	P94	P93	P92	P91	P90	
PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	Port
P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR	
P3DR	P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR	
P4DR	P47DR	P46DR	P45DR	P44DR	P43DR	P42DR	P41DR	P40DR	
P5DR	P57DR	P56DR	P55DR	P54DR	P53DR	P52DR	P51DR	P50DR	
P6DR	P67DR	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR	
P8DR	P87DR	P86DR	P85DR	P84DR	P83DR	P82DR	P81DR	P80DR	
P9DR	P97DR	P96DR	P95DR	P94DR	P93DR	P92DR	P91DR	P90DR	
PADR	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	
PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
PCDR	—	—	—	—	PC3DR	PC2DR	PC1DR	PC0DR	
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	
P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR	
P3DDR	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR	
P4DDR	P47DDR	P46DDR	P45DDR	P44DDR	P43DDR	P42DDR	P41DDR	P40DDR	
P5DDR	P57DDR	P56DDR	P55DDR	P54DDR	P53DDR	P52DDR	P51DDR	P50DDR	
P6DDR	P67DDR	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR	
P8DDR	P87DDR	P86DDR	P85DDR	P84DDR	P83DDR	P82DDR	P81DDR	P80DDR	
P9DDR	P97DDR	P96DDR	P95DDR	P94DDR	P93DDR	P92DDR	P91DDR	P90DDR	
PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR	
PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR	
PCDDR	—	—	—	—	PC3DDR	PC2DDR	PC1DDR	PC0DDR	
P1PCR	P17PCR	P16PCR	P15PCR	P14PCR	P13PCR	P12PCR	P11PCR	P10PCR	
P2PCR	P27PCR	P26PCR	P25PCR	P24PCR	P23PCR	P22PCR	P21PCR	P20PCR	
P3PCR	P37PCR	P36PCR	P35PCR	P34PCR	P33PCR	P32PCR	P31PCR	P30PCR	
P6PCR	P67PCR	P66PCR	P65PCR	P64PCR	P63PCR	P62PCR	P61PCR	P60PCR	
P6ODR	P67ODR	P66ODR	P65ODR	P64ODR	P63ODR	P62ODR	P61ODR	P60ODR	
PFCR	—	—	—	CS3E	CS2E	CS1E	LWROE	ASOE	
PTCNT0	PW7S	PW6S	PW5S	PW4S	PW3S	PW2S	PW1S	PW0S	
PTCNT1	IRQ7S	IRQ6S	IRQ5S	IRQ4S	IRQ3S	IRQ2S	IRQ1S	IRQ0S	
PTCNT2	TIOCB2/ TCLKDS	TIOCA2S	TIOCB1/ TCLKCS	TIOCA1S	TIOCD0/ TCLKBS	TIOCC0/ TCLKAS	TIOCB0S	TIOCA0S	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ICCRA_0	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC3_0
ICCRB_0	BBSY	SCP	SDAO	—	SCLO	—	IICRST	—	
ICMR_0	—	WAIT	—	—	BCWP	BC2	BC1	BC0	
ICIER_0	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR_0	TDRE	TEND	RDRF	NACKF	STOP	AL	AAS	ADZ	
SAR_0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA2	SVA1	—	
ICDRT_0	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR_0	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	
ICCRA_1	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC3_1
ICCRB_1	BBSY	SCP	SDAO	—	SCLO	—	IICRST	—	
ICMR_1	—	WAIT	—	—	BCWP	BC2	BC1	BC0	
ICIER_1	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR_1	TDRE	TEND	RDRF	NACKF	STOP	AL	AAS	ADZ	
SAR_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA2	SVA1	—	
ICDRT_1	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR_1	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	
ICSRA_0	AASA	AASB	—	—	—	—	—	—	IIC3_0
SARA_0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	SARE	
SARB_0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	SARE	
SAMR_0	MSA6	MSA5	MSA4	MSA3	MSA2	MSA1	MSA0	MTRS	
ICSRA_1	AASA	AASB	—	—	—	—	—	—	IIC3_1
SARA_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	SARE	
SARB_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	SARE	
SAMR_1	MSA6	MSA5	MSA4	MSA3	MSA2	MSA1	MSA0	MTRS	
ICCRA_2	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC3_2
ICCRB_2	BBSY	SCP	SDAO	—	SCLO	—	IICRST	—	
ICMR_2	—	WAIT	—	—	BCWP	BC2	BC1	BC0	
ICIER_2	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR_2	TDRE	TEND	RDRF	NACKF	STOP	AL	AAS	ADZ	
SAR_2	SVA6	SVA5	SVA4	SVA3	SVA2	SVA2	SVA1	—	
ICDRT_2	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR_2	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ICCRA_3	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC3_3
ICCRB_3	BBSY	SCP	SDAO	—	SCLO	—	IICRST	—	
ICMR_3	—	WAIT	—	—	BCWP	BC2	BC1	BC0	
ICIER_3	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR_3	TDRE	TEND	RDRF	NACKF	STOP	AL	AAS	ADZ	
SAR_3	SVA6	SVA5	SVA4	SVA3	SVA2	SVA2	SVA1	—	
ICDRT_3	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR_3	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	
ICSRA_2	AASA	AASB	—	—	—	—	—	—	IIC3_2
SARA_2	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	SARE	
SARB_2	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	SARE	
SAMR_2	MSA6	MSA5	MSA4	MSA3	MSA2	MSA1	MSA0	MTRS	
ICSRA_3	AASA	AASB	—	—	—	—	—	—	IIC3_3
SARA_3	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	SARE	
SARB_3	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	SARE	
SAMR_3	MSA6	MSA5	MSA4	MSA3	MSA2	MSA1	MSA0	MTRS	
BCR	—	ICIS	—	—	—	—	PNCASH	ADMXE	BSC
BCRA1	ABW1	AST1	PNCCS1	AW1	WMS11	WMS10	WC11	WC10	
BCRA2	ABW2	AST2	PNCCS2	AW2	WMS21	WMS20	WC21	WC20	
BCRA3	ABW3	AST3	PNCCS3	AW3	WMS31	WMS30	WC31	WC30	
TCNT	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	WDT
TCSR	OVF	WT/$\overline{IT}$	TME	—	RST/$\overline{NMI}$	CKS2	CKS1	CKS0	
PCSR	PWCKXC	PWCKXB	PWCKXA	—	—	—	PWCKB	PWCKA	PWM
PWOER	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0	
PWDPR	OS7	OS6	OS5	OS4	OS3	OS2	OS1	OS0	
PWSL	PWCKE	PWCKS	—	—	—	RS2	RS1	RS0	
PWDR7 to PWDR0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
DADRA	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	PWMX
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—	
DACR	—	PWME	—	—	OEB	OEA	OS	CKS	
DADRB	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	REGS	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
DACNTH	UC7	UC6	UC5	UC4	UC3	UC2	UC1	UC0	PWMX
DACNTL	UC8	UC9	UC10	UC11	UC12	UC13	—	REGS	
TSTR	—	—	—	—	—	CST2	CST1	CST0	TPU common
TSYR	—	—	—	—	—	SYNC2	SYNC1	SYNC0	
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_0
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRA_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRB_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRC_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRD_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_1
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRA_1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRB_1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_2
TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_2	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRA_2	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRB_2	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

Note: * To classify the same name registers of the other timers, “FR” are added to the abbreviations.

23.3 Register States in Each Operating Mode

Register Abbreviation	Reset	High-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
IPRA	Initialized	—	—	—	—	Initialized	Interrupt
IPRB	Initialized	—	—	—	—	Initialized	
IPRC	Initialized	—	—	—	—	Initialized	
IPRD	Initialized	—	—	—	—	Initialized	
IPRE	Initialized	—	—	—	—	Initialized	
IPRF	Initialized	—	—	—	—	Initialized	
IPRG	Initialized	—	—	—	—	Initialized	
IPRH	Initialized	—	—	—	—	Initialized	
IPRI	Initialized	—	—	—	—	Initialized	
IPRJ	Initialized	—	—	—	—	Initialized	
IPRK	Initialized	—	—	—	—	Initialized	
ISCR	Initialized	—	—	—	—	Initialized	
SSIER	Initialized	—	—	—	—	Initialized	
INTCR	Initialized	—	—	—	—	Initialized	
IER	Initialized	—	—	—	—	Initialized	
ISR	Initialized	—	—	—	—	Initialized	
SMR_0	Initialized	—	—	—	—	Initialized	SCI_0
BRR_0	Initialized	—	—	—	—	Initialized	
SCR_0	Initialized	—	—	—	—	Initialized	
TDR_0	Initialized	—	—	Initialized	Initialized	Initialized	
SSR_0	Initialized	—	—	Initialized	Initialized	Initialized	
RDR_0	Initialized	—	—	Initialized	Initialized	Initialized	
SCMR_0	Initialized	—	—	—	—	Initialized	
SMR_1	Initialized	—	—	—	—	Initialized	SCI_1
BRR_1	Initialized	—	—	—	—	Initialized	
SCR_1	Initialized	—	—	—	—	Initialized	
TDR_1	Initialized	—	—	Initialized	Initialized	Initialized	
SSR_1	Initialized	—	—	Initialized	Initialized	Initialized	
RDR_1	Initialized	—	—	Initialized	Initialized	Initialized	
SCMR_1	Initialized	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
SMR_2	Initialized	—	—	—	—	Initialized	SCI_2
BRR_2	Initialized	—	—	—	—	Initialized	
SCR_2	Initialized	—	—	—	—	Initialized	
TDR_2	Initialized	—	—	Initialized	Initialized	Initialized	
SSR_2	Initialized	—	—	Initialized	Initialized	Initialized	
RDR_2	Initialized	—	—	Initialized	Initialized	Initialized	
SCMR_2	Initialized	—	—	—	—	Initialized	
SMR_3	Initialized	—	—	—	—	Initialized	SCI_3
BRR_3	Initialized	—	—	—	—	Initialized	
SCR_3	Initialized	—	—	—	—	Initialized	
TDR_3	Initialized	—	—	Initialized	Initialized	Initialized	
SSR_3	Initialized	—	—	Initialized	Initialized	Initialized	
RDR_3	Initialized	—	—	Initialized	Initialized	Initialized	
SCMR_3	Initialized	—	—	—	—	Initialized	
SMR_4	Initialized	—	—	—	—	Initialized	SCI_4
BRR_4	Initialized	—	—	—	—	Initialized	
SCR_4	Initialized	—	—	—	—	Initialized	
TDR_4	Initialized	—	—	Initialized	Initialized	Initialized	
SSR_4	Initialized	—	—	Initialized	Initialized	Initialized	
RDR_4	Initialized	—	—	Initialized	Initialized	Initialized	
SCMR_4	Initialized	—	—	—	—	Initialized	
ADDRA	Initialized	—	—	—	—	Initialized	A/D converter
ADDRB	Initialized	—	—	—	—	Initialized	
ADDRC	Initialized	—	—	—	—	Initialized	
ADDRD	Initialized	—	—	—	—	Initialized	
ADDRE	Initialized	—	—	—	—	Initialized	
ADDRF	Initialized	—	—	—	—	Initialized	
ADDRG	Initialized	—	—	—	—	Initialized	
ADDRH	Initialized	—	—	—	—	Initialized	
ADCSR	Initialized	—	—	—	—	Initialized	
ADCR	Initialized	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
TWICR	Initialized	—	—	—	—	Initialized	Duty measurement circuit
TWCNT	Initialized	—	—	—	—	Initialized	
TWCR1	Initialized	—	—	—	—	Initialized	
TWCR2	Initialized	—	—	—	—	Initialized	
FR_TIER_0*	Initialized	—	—	—	—	Initialized	FRT_0
TCSR_0	Initialized	—	—	—	—	Initialized	
FRC_0	Initialized	—	—	—	—	Initialized	
OCRA_0	Initialized	—	—	—	—	Initialized	
OCRB_0	Initialized	—	—	—	—	Initialized	
FR_TCR_0*	Initialized	—	—	—	—	Initialized	
TOCR_0	Initialized	—	—	—	—	Initialized	
ICRA_0	Initialized	—	—	—	—	Initialized	
OCRAR_0	Initialized	—	—	—	—	Initialized	
ICRB_0	Initialized	—	—	—	—	Initialized	
OCRAF_0	Initialized	—	—	—	—	Initialized	
ICRC_0	Initialized	—	—	—	—	Initialized	
OCRDM_0	Initialized	—	—	—	—	Initialized	
ICRD_0	Initialized	—	—	—	—	Initialized	
TCRX_0	Initialized	—	—	—	—	Initialized	TMRX_0
TCSRX_0	Initialized	—	—	—	—	Initialized	
TICRR_0	Initialized	—	—	—	—	Initialized	
TICRF_0	Initialized	—	—	—	—	Initialized	
TCNTX_0	Initialized	—	—	—	—	Initialized	
TCORC_0	Initialized	—	—	—	—	Initialized	
TCORAX_0	Initialized	—	—	—	—	Initialized	
TCORBX_0	Initialized	—	—	—	—	Initialized	
TISR_0	Initialized	—	—	—	—	Initialized	TMR01_0
TCR0_0	Initialized	—	—	—	—	Initialized	
TCR1_0	Initialized	—	—	—	—	Initialized	
TCSR0_0	Initialized	—	—	—	—	Initialized	
TCSR1_0	Initialized	—	—	—	—	Initialized	
TCORA0_0	Initialized	—	—	—	—	Initialized	
TCORA1_0	Initialized	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
TCORB0_0	Initialized	—	—	—	—	Initialized	TMR01_0
TCORB1_0	Initialized	—	—	—	—	Initialized	
TCNT0_0	Initialized	—	—	—	—	Initialized	
TCNT1_0	Initialized	—	—	—	—	Initialized	
TCONRI_0	Initialized	—	—	—	—	Initialized	Timer connection_0
TCONRO_0	Initialized	—	—	—	—	Initialized	
TCONRS_0	Initialized	—	—	—	—	Initialized	
SEDGR_0	Initialized	—	—	—	—	Initialized	
TCRY_0	Initialized	—	—	—	—	Initialized	TMRY_0
TCSRY_0	Initialized	—	—	—	—	Initialized	
TCORAY_0	Initialized	—	—	—	—	Initialized	
TCORBY_0	Initialized	—	—	—	—	Initialized	
TCNTY_0	Initialized	—	—	—	—	Initialized	
FR_TIER_1*	Initialized	—	—	—	—	Initialized	FRT_1
TCSR_1	Initialized	—	—	—	—	Initialized	
FRC_1	Initialized	—	—	—	—	Initialized	
OCRA_1	Initialized	—	—	—	—	Initialized	
OCRB_1	Initialized	—	—	—	—	Initialized	
FR_TCR_1*	Initialized	—	—	—	—	Initialized	
TOCR_1	Initialized	—	—	—	—	Initialized	
ICRA_1	Initialized	—	—	—	—	Initialized	
OCRAR_1	Initialized	—	—	—	—	Initialized	
ICRB_1	Initialized	—	—	—	—	Initialized	
OCRAF_1	Initialized	—	—	—	—	Initialized	
ICRC_1	Initialized	—	—	—	—	Initialized	
OCRDM_1	Initialized	—	—	—	—	Initialized	
ICRD_1	Initialized	—	—	—	—	Initialized	
TCRX_1	Initialized	—	—	—	—	Initialized	TMRX_1
TCSRX_1	Initialized	—	—	—	—	Initialized	
TICRR_1	Initialized	—	—	—	—	Initialized	
TICRF_1	Initialized	—	—	—	—	Initialized	
TCNTX_1	Initialized	—	—	—	—	Initialized	
TCORC_1	Initialized	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
TCORAX_1	Initialized	—	—	—	—	Initialized	TMRX_1
TCORBX_1	Initialized	—	—	—	—	Initialized	
TCR0_1	Initialized	—	—	—	—	Initialized	TMR01_1
TCR1_1	Initialized	—	—	—	—	Initialized	
TCSR0_1	Initialized	—	—	—	—	Initialized	
TCSR1_1	Initialized	—	—	—	—	Initialized	
TCORA0_1	Initialized	—	—	—	—	Initialized	
TCORA1_1	Initialized	—	—	—	—	Initialized	
TCORB0_1	Initialized	—	—	—	—	Initialized	
TCORB1_1	Initialized	—	—	—	—	Initialized	
TCNT0_1	Initialized	—	—	—	—	Initialized	
TCNT1_1	Initialized	—	—	—	—	Initialized	
TCONRI_1	Initialized	—	—	—	—	Initialized	Timer connection_1
TCONRO_1	Initialized	—	—	—	—	Initialized	
TCONRS_1	Initialized	—	—	—	—	Initialized	
SEDGR_1	Initialized	—	—	—	—	Initialized	
TCRY_1	Initialized	—	—	—	—	Initialized	TMRY_1
TCSRY_1	Initialized	—	—	—	—	Initialized	
TCORAY_1	Initialized	—	—	—	—	Initialized	
TCORBY_1	Initialized	—	—	—	—	Initialized	
TCNTY_1	Initialized	—	—	—	—	Initialized	
TISR_1	Initialized	—	—	—	—	Initialized	
FCCS	Initialized	—	—	—	—	Initialized	Flash memory
FPCS	Initialized	—	—	—	—	Initialized	
FECS	Initialized	—	—	—	—	Initialized	
FKEY	Initialized	—	—	—	—	Initialized	
FMATS	Initialized	—	—	—	—	Initialized	
FTDAR	Initialized	—	—	—	—	Initialized	
MDCR	Initialized	—	—	—	—	Initialized	System
SYSCR	Initialized	—	—	—	—	Initialized	
SBYCR	Initialized	—	—	—	—	Initialized	
SCKCR	Initialized	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
MSTPCRH	Initialized	—	—	—	—	Initialized	System
MSTPCRL	Initialized	—	—	—	—	Initialized	
EXMSTPCRH	Initialized	—	—	—	—	Initialized	
EXMSTPCRL	Initialized	—	—	—	—	Initialized	
TECR	Initialized	—	—	—	—	Initialized	
PORT0	—	—	—	—	—	—	Port
PORT1	—	—	—	—	—	—	
PORT2	—	—	—	—	—	—	
PORT3	—	—	—	—	—	—	
PORT4	—	—	—	—	—	—	
PORT5	—	—	—	—	—	—	
PORT6	—	—	—	—	—	—	
PORT7	—	—	—	—	—	—	
PORT8	—	—	—	—	—	—	
PORT9	—	—	—	—	—	—	
PORTA	—	—	—	—	—	—	
PORTB	—	—	—	—	—	—	
PORTC	—	—	—	—	—	—	
P1DR	Initialized	—	—	—	—	Initialized	
P2DR	Initialized	—	—	—	—	Initialized	
P3DR	Initialized	—	—	—	—	Initialized	
P4DR	Initialized	—	—	—	—	Initialized	
P5DR	Initialized	—	—	—	—	Initialized	
P6DR	Initialized	—	—	—	—	Initialized	
P8DR	Initialized	—	—	—	—	Initialized	
P9DR	Initialized	—	—	—	—	Initialized	
PADR	Initialized	—	—	—	—	Initialized	
PBDR	Initialized	—	—	—	—	Initialized	
PCDR	Initialized	—	—	—	—	Initialized	
P1DDR	Initialized	—	—	—	—	Initialized	
P2DDR	Initialized	—	—	—	—	Initialized	
P3DDR	Initialized	—	—	—	—	Initialized	
P4DDR	Initialized	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
P5DDR	Initialized	—	—	—	—	Initialized	Port
P6DDR	Initialized	—	—	—	—	Initialized	
P8DDR	Initialized	—	—	—	—	Initialized	
P9DDR	Initialized	—	—	—	—	Initialized	
PADDR	Initialized	—	—	—	—	Initialized	
PBDDR	Initialized	—	—	—	—	Initialized	
PCDDR	Initialized	—	—	—	—	Initialized	
P1PCR	Initialized	—	—	—	—	Initialized	
P2PCR	Initialized	—	—	—	—	Initialized	
P3PCR	Initialized	—	—	—	—	Initialized	
P6PCR	Initialized	—	—	—	—	Initialized	
P6ODR	Initialized	—	—	—	—	Initialized	
PFCR	Initialized	—	—	—	—	Initialized	
PTCNT0	Initialized	—	—	—	—	Initialized	
PTCNT1	Initialized	—	—	—	—	Initialized	
PTCNT2	Initialized	—	—	—	—	Initialized	
ICCRA_0	Initialized	—	—	—	—	Initialized	IIC3_0
ICCRB_0	Initialized	—	—	—	—	Initialized	
ICMR_0	Initialized	—	—	—	—	Initialized	
ICIER_0	Initialized	—	—	—	—	Initialized	
ICSR_0	Initialized	—	—	—	—	Initialized	
SAR_0	Initialized	—	—	—	—	Initialized	
ICDRT_0	Initialized	—	—	—	—	Initialized	
ICDRR_0	Initialized	—	—	—	—	Initialized	
ICCRA_1	Initialized	—	—	—	—	Initialized	IIC3_1
ICCRB_1	Initialized	—	—	—	—	Initialized	
ICMR_1	Initialized	—	—	—	—	Initialized	
ICIER_1	Initialized	—	—	—	—	Initialized	
ICSR_1	Initialized	—	—	—	—	Initialized	
SAR_1	Initialized	—	—	—	—	Initialized	
ICDRT_1	Initialized	—	—	—	—	Initialized	
ICDRR_1	Initialized	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
ICSRA_0	Initialized	—	—	—	—	Initialized	IIC3_0
SARA_0	Initialized	—	—	—	—	Initialized	
SARB_0	Initialized	—	—	—	—	Initialized	
SAMR_0	Initialized	—	—	—	—	Initialized	
ICSRA_1	Initialized	—	—	—	—	Initialized	IIC3_1
SARA_1	Initialized	—	—	—	—	Initialized	
SARB_1	Initialized	—	—	—	—	Initialized	
SAMR_1	Initialized	—	—	—	—	Initialized	
ICCRA_2	Initialized	—	—	—	—	Initialized	IIC3_2
ICCRB_2	Initialized	—	—	—	—	Initialized	
ICMR_2	Initialized	—	—	—	—	Initialized	
ICIER_2	Initialized	—	—	—	—	Initialized	
ICSR_2	Initialized	—	—	—	—	Initialized	
SAR_2	Initialized	—	—	—	—	Initialized	
ICDRT_2	Initialized	—	—	—	—	Initialized	
ICDRR_2	Initialized	—	—	—	—	Initialized	
ICCRA_3	Initialized	—	—	—	—	Initialized	IIC3_3
ICCRB_3	Initialized	—	—	—	—	Initialized	
ICMR_3	Initialized	—	—	—	—	Initialized	
ICIER_3	Initialized	—	—	—	—	Initialized	
ICSR_3	Initialized	—	—	—	—	Initialized	
SAR_3	Initialized	—	—	—	—	Initialized	
ICDRT_3	Initialized	—	—	—	—	Initialized	
ICDRR_3	Initialized	—	—	—	—	Initialized	
ICSRA_2	Initialized	—	—	—	—	Initialized	IIC3_2
SARA_2	Initialized	—	—	—	—	Initialized	
SARB_2	Initialized	—	—	—	—	Initialized	
SAMR_2	Initialized	—	—	—	—	Initialized	
ICSRA_3	Initialized	—	—	—	—	Initialized	IIC3_3
SARA_3	Initialized	—	—	—	—	Initialized	
SARB_3	Initialized	—	—	—	—	Initialized	
SAMR_3	Initialized	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
BCR	Initialized	—	—	—	—	Initialized	BSC
BCRA1	Initialized	—	—	—	—	Initialized	
BCRA2	Initialized	—	—	—	—	Initialized	
BCRA3	Initialized	—	—	—	—	Initialized	
TCNT	Initialized	—	—	—	—	Initialized	WDT
TCSR	Initialized	—	—	—	—	Initialized	
PCSR	Initialized	—	—	—	—	Initialized	PWM
POWER	Initialized	—	—	—	—	Initialized	
PWDPR	Initialized	—	—	—	—	Initialized	
PWSL	Initialized	—	—	Initialized	Initialized	Initialized	
PWDR7 to PWDR0	Initialized	—	—	Initialized	Initialized	Initialized	
DADRA	Initialized	—	—	Initialized	Initialized	Initialized	PWMX
DACR	Initialized	—	—	Initialized	Initialized	Initialized	
DADRB	Initialized	—	—	Initialized	Initialized	Initialized	
DACNTH	Initialized	—	—	Initialized	Initialized	Initialized	
DACNTL	Initialized	—	—	Initialized	Initialized	Initialized	
TSTR	Initialized	—	—	—	—	Initialized	TPU common
TSYR	Initialized	—	—	—	—	Initialized	
TCR_0	Initialized	—	—	—	—	Initialized	TPU_0
TMDR_0	Initialized	—	—	—	—	Initialized	
TIORH_0	Initialized	—	—	—	—	Initialized	
TIORL_0	Initialized	—	—	—	—	Initialized	
TIER_0	Initialized	—	—	—	—	Initialized	
TSR_0	Initialized	—	—	—	—	Initialized	
TCNT_0	Initialized	—	—	—	—	Initialized	
TGRA_0	Initialized	—	—	—	—	Initialized	
TGRB_0	Initialized	—	—	—	—	Initialized	
TGRC_0	Initialized	—	—	—	—	Initialized	
TGRD_0	Initialized	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
TCR_1	Initialized	—	—	—	—	Initialized	TPU_1
TMDR_1	Initialized	—	—	—	—	Initialized	
TIOR_1	Initialized	—	—	—	—	Initialized	
TIER_1	Initialized	—	—	—	—	Initialized	
TSR_1	Initialized	—	—	—	—	Initialized	
TCNT_1	Initialized	—	—	—	—	Initialized	
TGRA_1	Initialized	—	—	—	—	Initialized	
TGRB_1	Initialized	—	—	—	—	Initialized	
TCR_2	Initialized	—	—	—	—	Initialized	TPU_2
TMDR_2	Initialized	—	—	—	—	Initialized	
TIOR_2	Initialized	—	—	—	—	Initialized	
TIER_2	Initialized	—	—	—	—	Initialized	
TSR_2	Initialized	—	—	—	—	Initialized	
TCNT_2	Initialized	—	—	—	—	Initialized	
TGRA_2	Initialized	—	—	—	—	Initialized	
TGRB_2	Initialized	—	—	—	—	Initialized	

Note: * To classify the same name registers of the other timers, “FR” are added to the abbreviations.

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
Power supply voltage (VCL pin)	V_{CL}	–0.3 to +4.3	
Input voltage (except ports 0 and 7)	V_{in}	–0.3 to V_{CC} +0.3	
Input voltage (ports 0 and 7)	V_{in}	–0.3 to AV_{CC} +0.3	
Reference power supply voltage	AV_{ref}	–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}	0 to +75	
Storage temperature	T_{stg}	–55 to +125	

Caution: Permanent damage to this LSI may result if absolute maximum ratings are exceeded.
Note that the applied voltage should not exceed 4.3 V.

Note: * Voltage applied to the VCC pin.

24.2 DC Characteristics

Table 24.2 lists the DC characteristics. Table 24.3 lists the permissible output currents.

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,
AV_{ref}*[1] = 3.0 V to AV_{CC}, V_{SS} = AV_{SS}*[1] = 0 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	$\overline{IRQ0}$ to $\overline{IRQ7}$, ExIRQ0 to ExIRQ7	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, FWE, MD0 to MD2, SCK0 to SCK4, RxD0 to RxD4, TMI0_0, TMI1_0, TMIX_0, TMIY_0, TMI0_1, TMI1_1, TMIX_1, TMIY_1, FTCI_0, FTIA_0, FTIB_0, FTIC_0, FTID_0, FTCI_1, FTIA_1, FTIB_1, FTIC_1, FTID_1, TCLKA, TCLKB, TCLKC, TCLKD, TIOCA0, TIOCB0, TIOCC0, TIOCD0, TIOCA1, TIOCB1, TIOCA2, TIOCB2, ExTCLKA, ExTCLKB, ExTCLKC, ExTCLKD, ExTIOCA0, ExTIOCB0, ExTIOCC0, ExTIOCD0, ExTIOCA1, ExTIOCB1, ExTIOCA2, ExTIOCB2	V_{IH}	V_{CC} × 0.9	—	V_{CC} + 0.3		
	SCL3, SCL2, SCL1, SCL0, SDA3, SDA2, SDA1, SDA0, P80 to P83, PC0 to PC4		V_{CC} × 0.7	—	5.5		
	Other than above		V_{CC} × 0.7	—	V_{CC} + 0.3		
Input low voltage	$\overline{RES}$, $\overline{STBY}$, FWE, MD0 to MD2	V_{IL}	–0.3	—	V_{CC} × 0.1		
	Other than above		–0.3	—	V_{CC} × 0.2		

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	All output pins (except for P80 to P83, PC0 to PC3)	V_{OH}	V_{CC} – 0.5	—	—		I_{OH} = –200 µA
			V_{CC} – 1.0	—	—		I_{OH} = –1 mA
	P80 to P83, PC0 to PC3*[2]		0.5	—	—		I_{OH} = –200 µA
Output low voltage	SCL3, SCL2, SCL1, SCL0, SDA3, SDA2, SDA1, SDA0	V_{OL}	—	—	0.4		I_{OL} = 3 mA
	Output pins other than above		—	—	0.4		I_{OL} = 1.6 mA
Input leakage current	$\overline{STBY}$, NMI, $\overline{RES}$, MD0 to MD2, FWE	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
	Ports 0 and 7		—	—	1.0		V_{in} = 0.5 to AV_{CC} – 0.5 V
Three-state leakage current (off state)	Ports 1 to 6, ports 84 to 87, ports 9, A to D	$\lvert I_{TSI} \rvert$	—	—	1.0		V_{in} = 0.5 to V_{CC} – 0.5 V
	Ports 80 to 83, ports C0 to C3		—	—	1.0		V_{in} = 0.5 to 5.5 V
Input pull-up MOS current	Ports 1 to 3, 6	$-I_P$	5	—	150		V_{in} = 0 V
Input capacitance	All input pins (except for P80 toP83, PC0 to PC3)	C_{in}	—	—	10	pF	V_{in} = 0 V f = 1 MHz Ta = 25°C
	P80 to P83, PC0 to PC3		—	—	10		
Current consumption*[3]	Normal operation*[5]	I_{CC}	—	23	35	mA	V_{CC} = 3.0 V to 3.6 V f = 20 MHz, when all module stop cleared, high-speed mode
	Sleep mode		—	15	25		V_{CC} = 3.0 V to 3.6 V f = 20 MHz
	Standby mode*[4]		—	—	90	µA	Ta ≤ 50°C
			—	—	120		50°C < Ta
Analog power supply current	During A/D conversion	AI_{cc}	—	—	2.0	mA	
	A/D conversion standby		—	—	5.0	µA	AV_{CC} = 2.0 V to 3.6 V
Reference power supply current	During A/D conversion	AI_{ref}	—	—	1.0	mA	
	A/D conversion standby		—	—	5.0	µA	AV_{ref} = 2.0 V to 3.6 V
RAM standby voltage*[4]		V_{RAM}	2.0	—	—	V	

Notes: 1. Do not leave the AVCC, AVref, and AVSS pins open even if the A/D converter is not used.
Even if the A/D converter is not used, apply a value in the range from 3.0 V to 3.6 V to the AVCC and AVref pins by connection to the power supply (VCC). The relationship between these two pins should be AVref ≤ AVCC.

2. P80/SCL0 to P83/SDA1 and PC0/SCL2 to PC3/SDA3 are NMOS push-pull outputs.
An external pull-up resistor is necessary to provide high-level output from SCL0 to SCL3 and SDA0 to SDA3 (ICE bit in ICCRA is 1).
P80 to P83 and PC0 to PC3 (ICE bit in ICCRA is 0) high levels 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.

3. Current consumption values are for V_{IH} min. = VCC – 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.

4. The values are for $V_{RAM} \leq VCC < 3.0$ V, V_{IH} min. = VCC – 0.2 V, and V_{IL} max. = 0.2 V.

5. Except for flash memory programming/erasing.

Table 24.3 Permissible Output Currents

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	SCL0 to SCL3, SDA0 to SDA3	I_{OL}	—	—	10	mA
	All output pins		—	—	2	
Permissible output low current (total)	Total of all output pins	$\sum I_{OL}$	—	—	60	
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2	
Permissible output high current (total)	Total of all output pins	$\sum -I_{OH}$	—	—	30	

Notes: 1. To protect LSI reliability, do not exceed the output current values in table 24.3.

2. When driving a Darlington transistor directly, always insert a current-limiting resistor in the output line, as shown in figure 24.1.

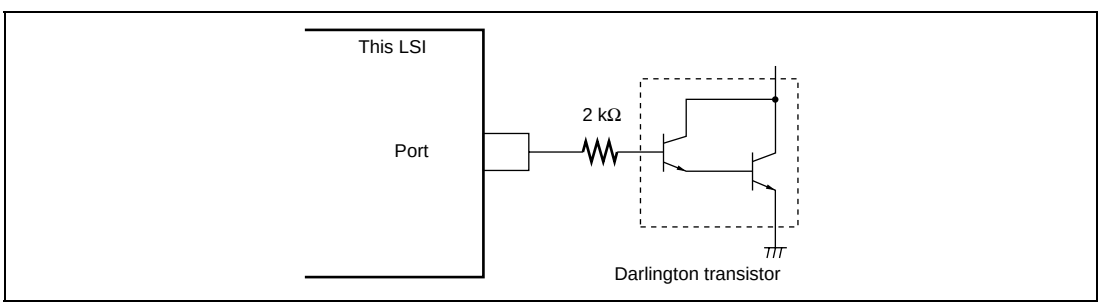
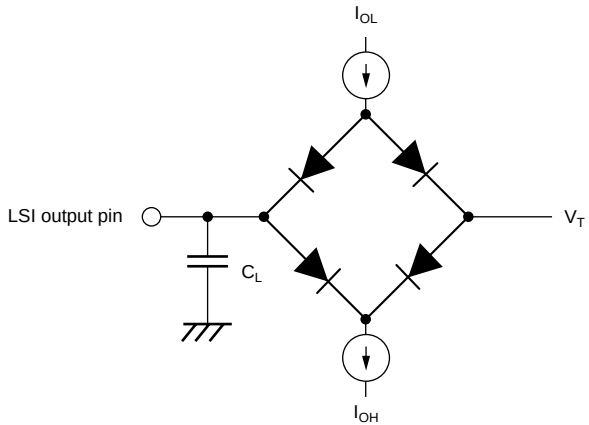


Figure 24.1 Darlington Transistor Drive Circuit (Example)

24.3 AC Characteristics

Figure 24.2 shows the test conditions for the AC characteristics.



IOL = 1.6 μA, IOH = 200 μA I/O reference level: 1.5 V
VT = 1.5 V
CL = 30 pF (The CL value includes capacitance of measuring jigs.)

Figure 24.2 Output Load Circuit

24.3.1 Clock Timing

Table 24.4 shows the clock timing. The clock timing specified here covers oscillation stabilization times for clock output (ϕ), clock pulse generator (crystal), and external clock input (EXTAL pin). For details on external clock input (EXTAL pin and EXCL pin) timing, see section 21, Clock Pulse Generator.

Table 24.4 Clock Timing

Condition: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 5 MHz to 20 MHz

Item	Symbol	Min.	Max.	Unit	Reference
Clock cycle time	t_{cyc}	50	200	ns	Figure 24.3
Clock high pulse width	t_{CH}	10	—		
Clock low pulse width	t_{CL}	10	—		
Clock rise time	t_{Cr}	—	8		
Clock fall time	t_{Cf}	—	8		
Reset oscillation stabilization (crystal)	t_{OSC1}	10	—	ms	Figure 24.4
Software standby oscillation stabilization time (crystal)	t_{OSC2}	8	—		Figure 24.5
External clock output stabilization delay time	t_{DEXT}	500	—	µs	Figure 24.4

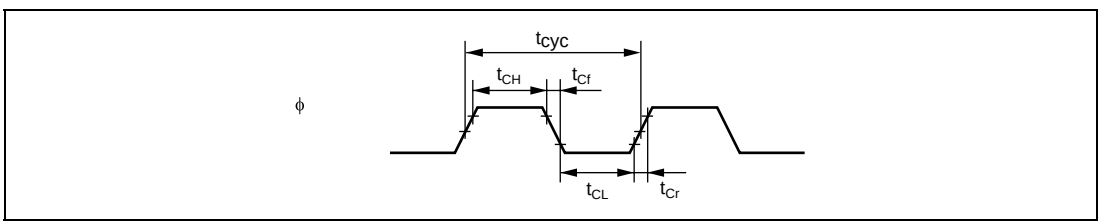


Figure 24.3 System Clock Timing

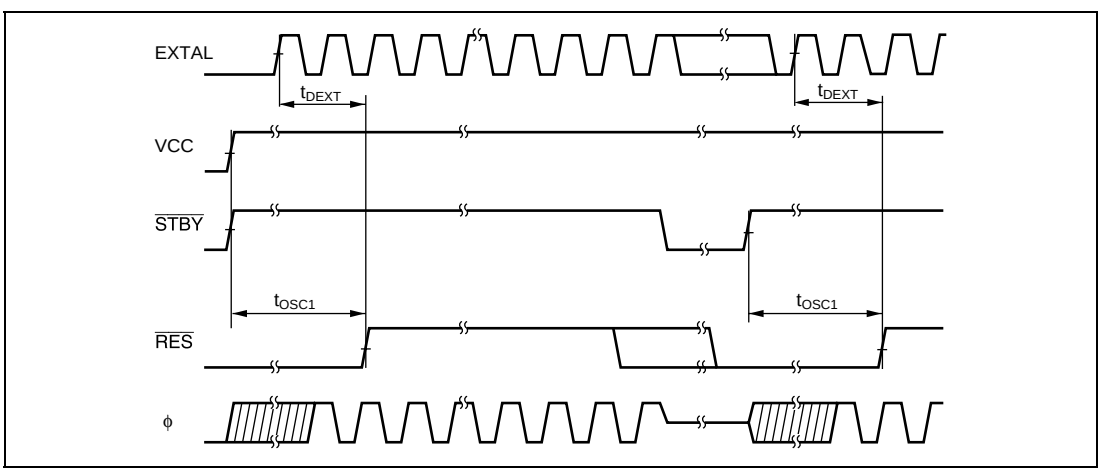


Figure 24.4 Oscillation Stabilization Timing

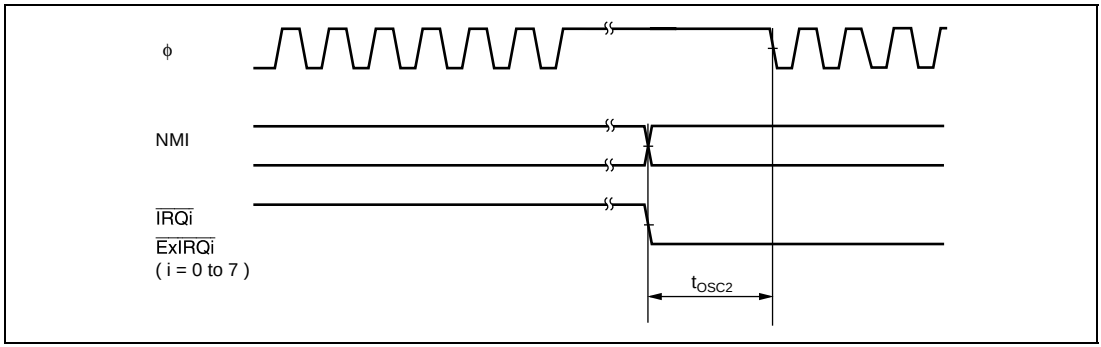


Figure 24.5 Oscillation Stabilization Timing (Exiting Software Standby Mode)

24.3.2 Control Signal Timing

Table 24.5 shows the control signal timing.

Table 24.5 Control Signal Timing

Condition: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, φ = 5 MHz to 20 MHz

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figure 24.6
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 24.7
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—		
IRQ setup time ($\overline{IRQ0}$ to $\overline{IRQ7}$, $\overline{ExIRQ0}$ to $\overline{ExIRQ7}$)	t_{IRQS}	150	—		
IRQ hold time ($\overline{IRQ0}$ to $\overline{IRQ7}$, $\overline{ExIRQ0}$ to $\overline{ExIRQ7}$)	t_{IRQH}	10	—		
IRQ pulse width ($\overline{IRQ0}$ to $\overline{IRQ7}$, $\overline{ExIRQ0}$ to $\overline{ExIRQ7}$) (exiting software standby mode)	t_{IRQW}	200	—		

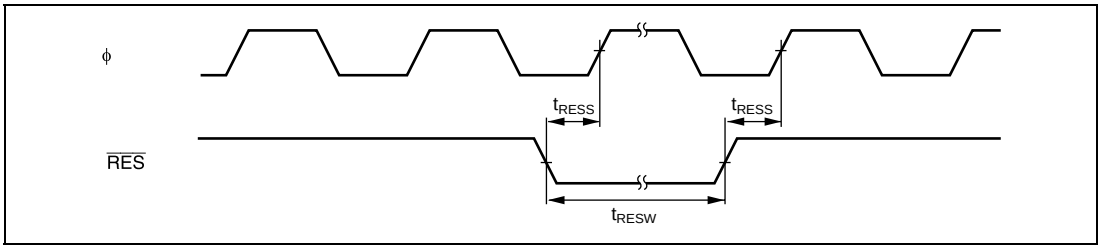


Figure 24.6 Reset Input Timing

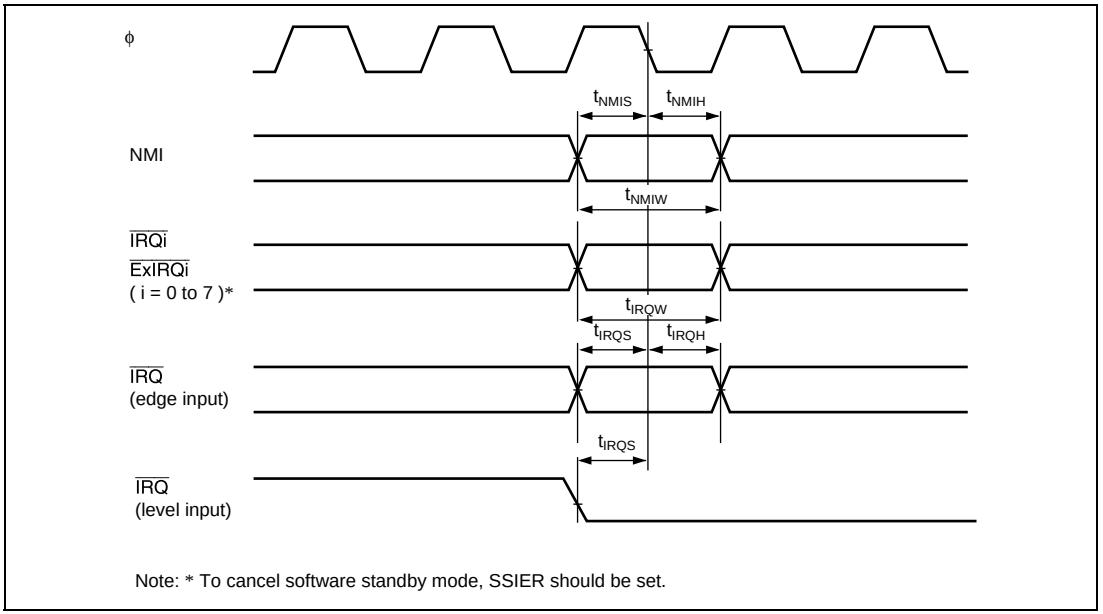


Figure 24.7 Interrupt Input Timing

24.3.3 Bus Timing

Table 24.6 shows the bus timing.

Table 24.6 Bus Timing (Normal Extension)

Condition: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 5 MHz to 20 MHz

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time	t_{AD}	—	20	ns	Figures 24.8 to 24.10
Address setup time	t_{AS}	0.5 x t_{cyc} –15	—		
Address hold time	t_{AH}	0.5 x t_{cyc} – 10	—		
$\overline{CS}$ delay time	t_{CSD}	—	20		
$\overline{AS}$ delay time	t_{ASD}	—	20		
$\overline{RD}$ delay time 1	t_{RSD1}	—	20		
$\overline{RD}$ delay time 2	t_{RSD2}	—	20		
Read data setup time	t_{RDS}	15	—		
Read data hold time	t_{RDH}	0	—		
Read data access time 2	t_{ACC2}	—	1.5 x t_{cyc} – 25		
Read data access time 3	t_{ACC3}	—	2.0 x t_{cyc} – 30		
Read data access time 4	t_{ACC4}	—	2.5 x t_{cyc} – 25		
Read data access time 5	t_{ACC5}	—	3.0 x t_{cyc} – 30		
$\overline{WR}$ delay time 1	t_{WRD1}	—	20		
$\overline{WR}$ delay time 2	t_{WRD2}	—	20		
$\overline{WR}$ pulse width 1	t_{WSW1}	1.0 x t_{cyc} –20	—		
$\overline{WR}$ pulse width 2	t_{WSW2}	1.5 x t_{cyc} –20	—		
Write data delay time	t_{WDD}	—	30		
Write data setup time	t_{WDS}	0	—		
Write data hold time	t_{WDH}	10	—		
$\overline{WAIT}$ setup time	t_{WTS}	30	—		
$\overline{WAIT}$ hold time	t_{WTH}	5	—		

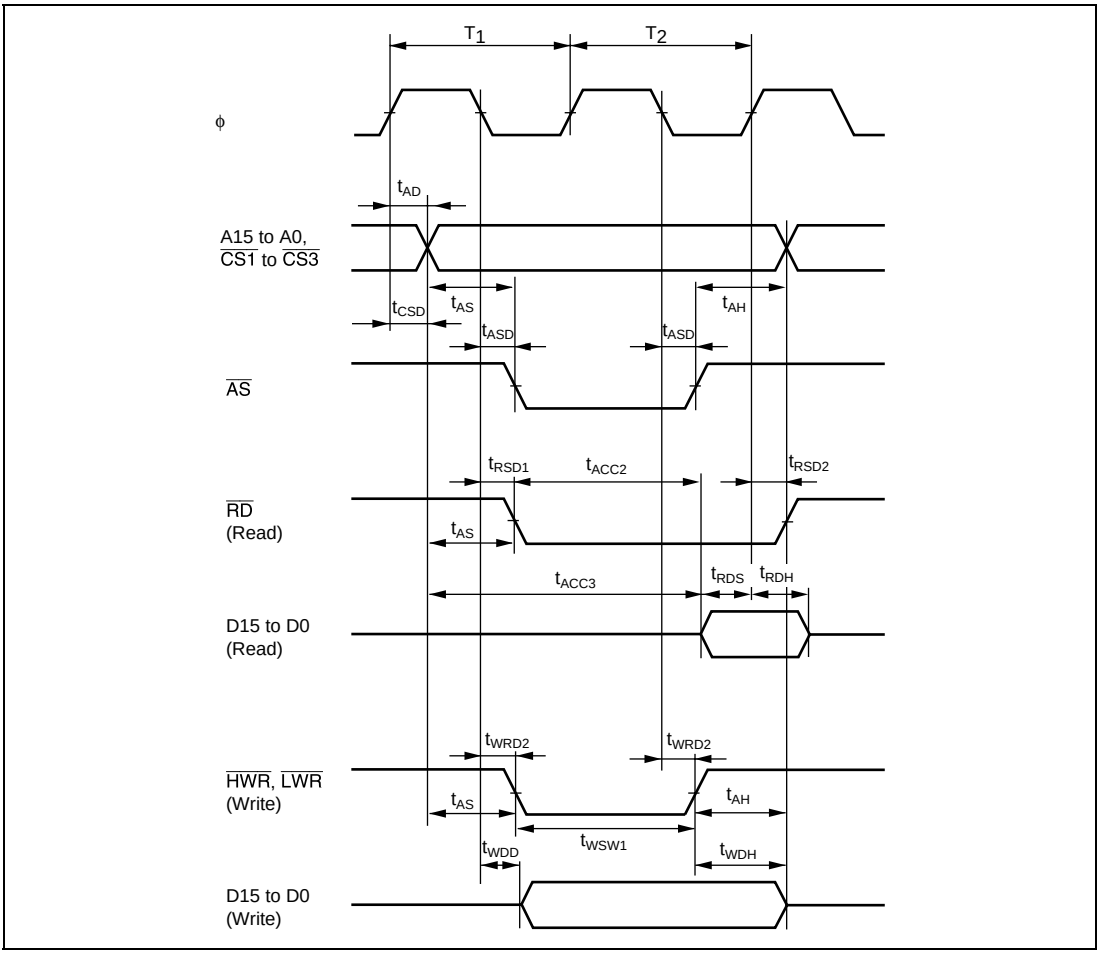


Figure 24.8 Basic Bus Timing/2-State Access

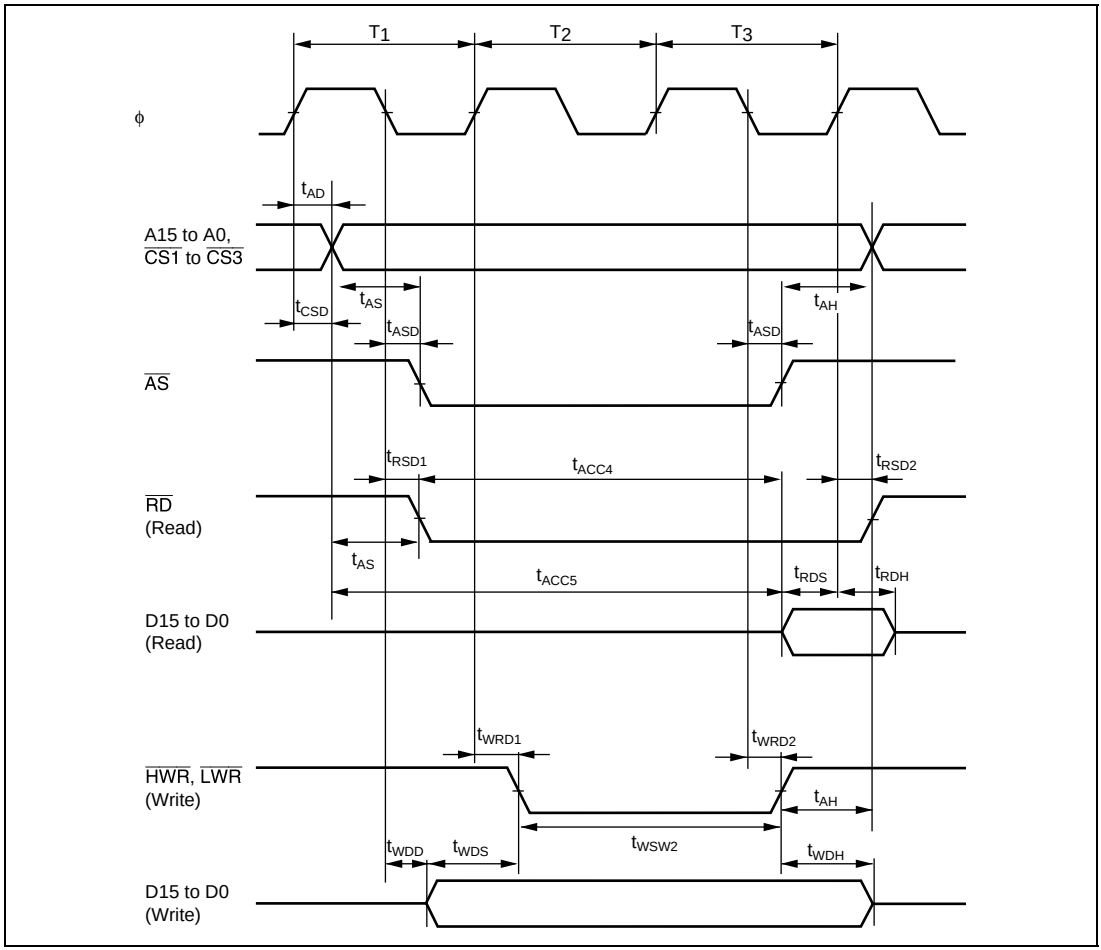


Figure 24.9 Basic Bus Timing/3-State Access

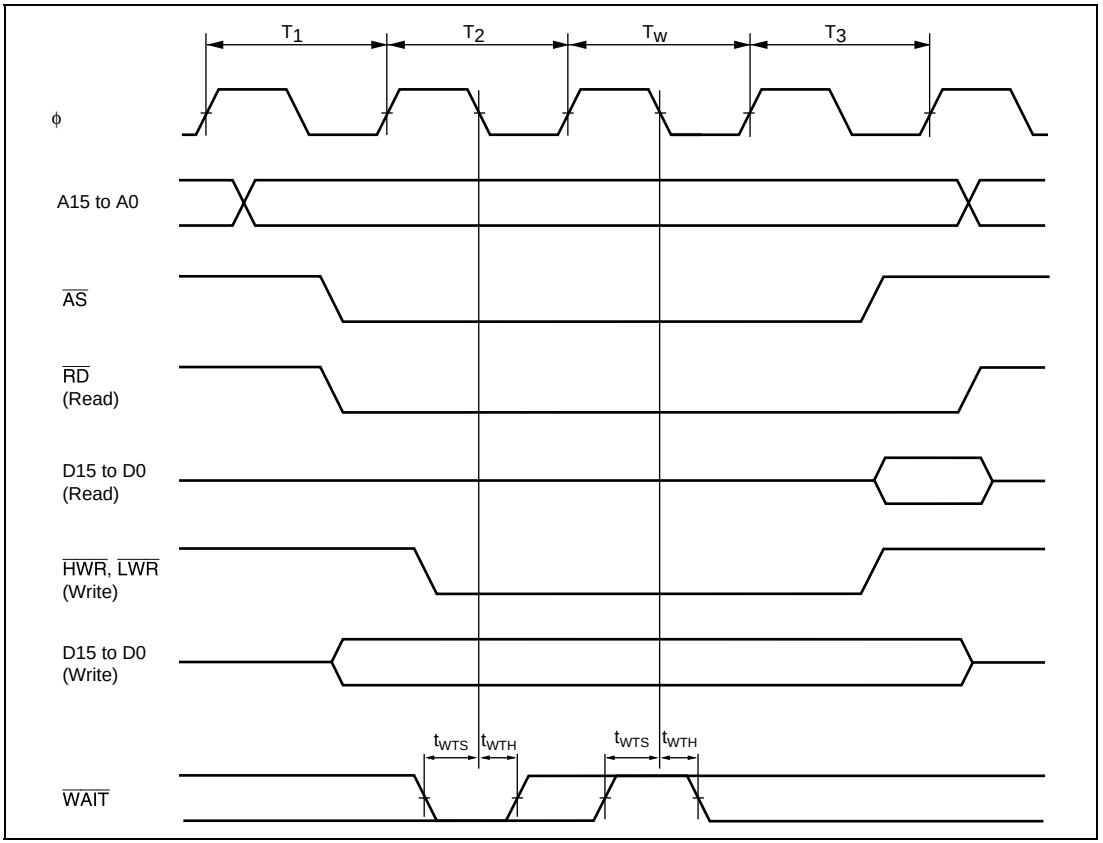


Figure 24.10 Basic Bus Timing/3-State Access with One Wait State

Table 24.7 Bus Timing (Multiplex Extension)

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time	t_{AD}	—	20	ns	Figures 24.11 to 24.13
Address setup time 2	t_{AS2}	0.5 x t_{cyc} –15	—		
Address hold time 2	t_{AH2}	1.0 x t_{cyc} – 10	—		
$\overline{CS}$ delay time ($\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$)	t_{CSD}	—	20		
$\overline{AH}$ delay time	t_{AHD}	—	20		
$\overline{RD}$ delay time 1	t_{RSD1}	—	20		
$\overline{RD}$ delay time 2	t_{RSD2}	—	20		
Read data setup time	t_{RDS}	15	—		
Read data hold time	t_{RDH}	0	—		
Read data access time 2	t_{ACC2}	—	1.5 x t_{cyc} – 25		
Read data access time 4	t_{ACC4}	—	2.5 x t_{cyc} – 30		
Read data access time 6	t_{ACC6}	—	3.5 x t_{cyc} – 25		
Read data access time 7	t_{ACC7}	—	4.5 x t_{cyc} – 30		
$\overline{WR}$ delay time 1	t_{WRD1}	—	20		
$\overline{WR}$ delay time 2	t_{WRD2}	—	20		
$\overline{WR}$ pulse width 1	t_{WSW1}	1.0 x t_{cyc} –20	—		
$\overline{WR}$ pulse width 2	t_{WSW2}	1.5 x t_{cyc} –20	—		
Write data delay time	t_{WDD}	—	30		
Write data setup time	t_{WDS}	0	—		
Write data hold time	t_{WDH}	10	—		
$\overline{WAIT}$ setup time	t_{WTS}	30	—		
$\overline{WAIT}$ hold time	t_{WTH}	5	—		

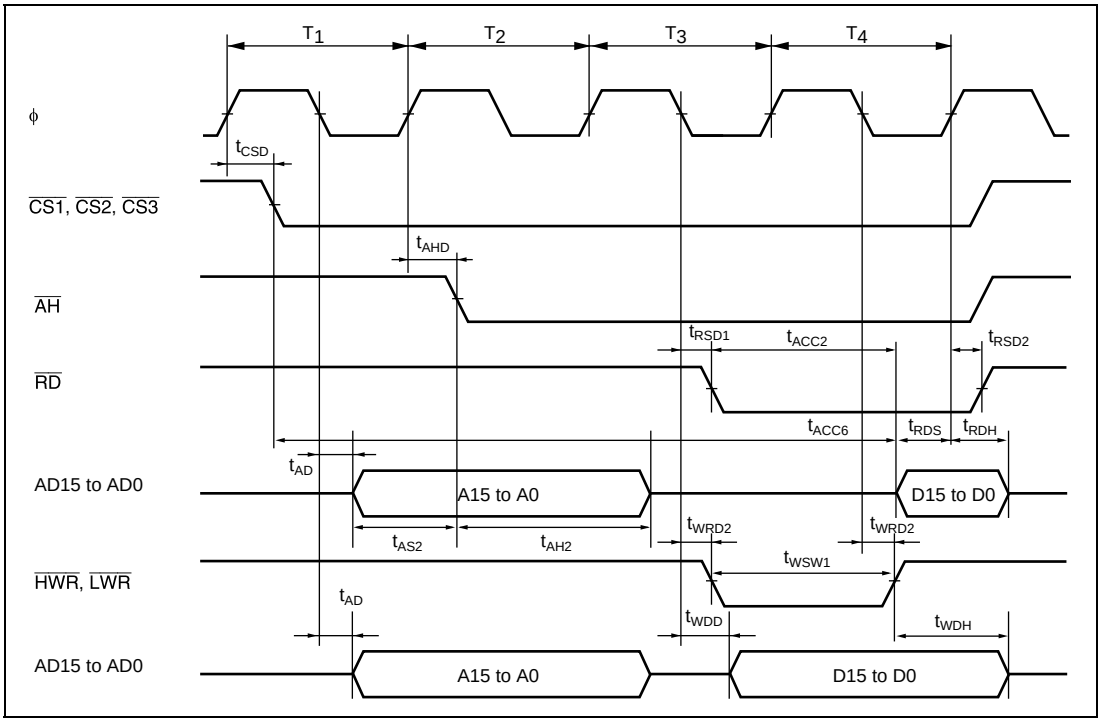


Figure 24.11 Muliplex Bus Timing/2-State Access

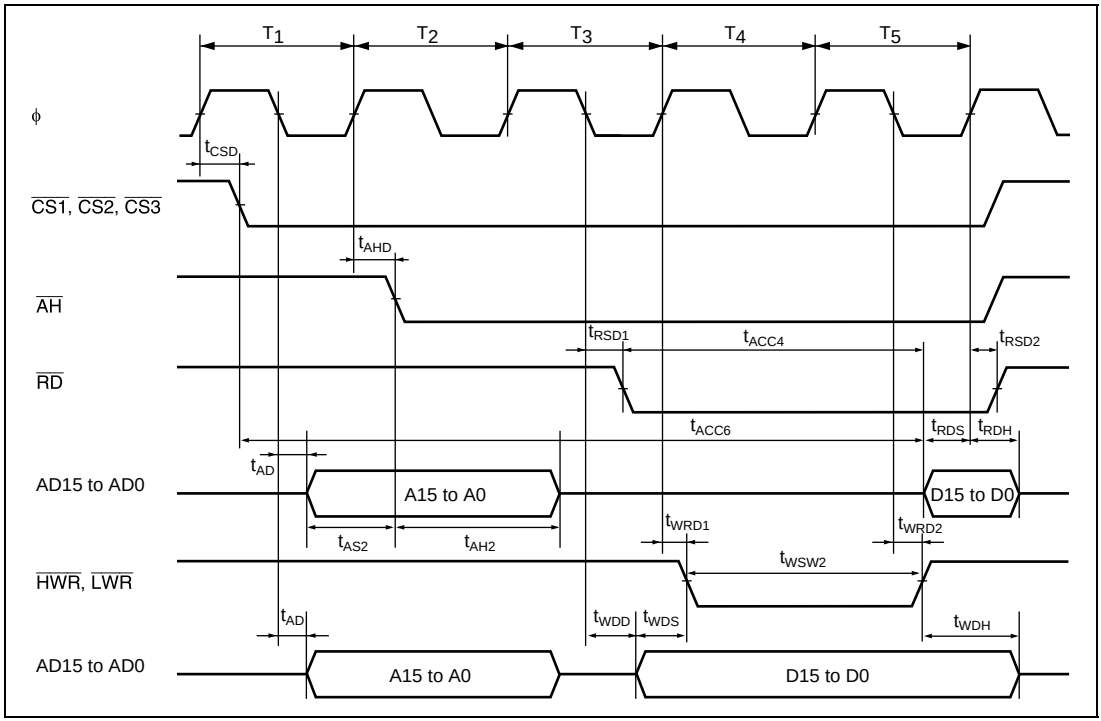


Figure 24.12 Multiplex Bus Timing/3-State Access

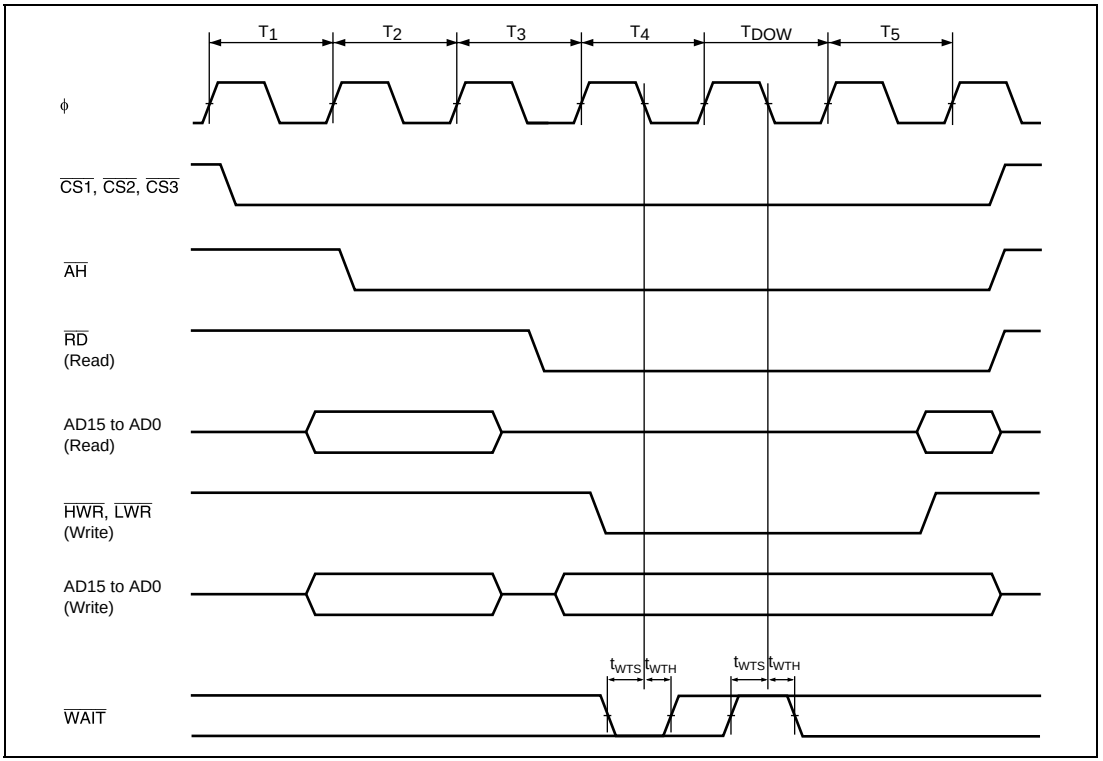


Figure 24.13 Multiplex Bus Timing/3-State Access with One Wait State

24.3.4 Timing of On-Chip Peripheral Modules

Tables 24.8 to 24.10 show the on-chip peripheral module timing.

Table 24.8 Timing of On-Chip Peripheral Modules

Condition: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 5 MHz to 20 MHz

Item			Symbol	Min.	Max.	Unit	Test Conditions
I/O ports	Output data delay time		t_{PWD}	—	40	ns	Figure 24.14
	Input data setup time		t_{PRS}	20	—		
	Input data hold time		t_{PRH}	20	—		
FRT	Timer output delay time		t_{FTOD}	—	40		Figure 24.15
	Timer input setup time		t_{FTIS}	20	—		
	Timer clock input setup time		t_{FTCS}	20	—		Figure 24.16
	Timer clock pulse width	Single edge	t_{FTCWH}	1.5	—	t_{cyc}	
		Both edges	t_{FTCWL}	2.5	—		
TPU	Timer output delay time		t_{TOCD}	—	40	ns	Figure 24.17
	Timer input setup time		t_{TICS}	25	—		
	Timer clock input setup time		t_{TCKS}	25	—		Figure 24.18
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—		
TMR	Timer output delay time		t_{TMOD}	—	40	ns	Figure 24.19
	Timer reset input setup time		t_{TMRS}	25	—		Figure 24.21
	Timer clock input setup time		t_{TMCS}	25	—		Figure 24.20
	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	t_{cyc}	
		Both edges	t_{TMCWL}	2.5	—		
PWM, PWMX	Pulse output delay time		t_{PWOD}	—	40	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}	—	40	ns	Figure 24.24
	Receive data setup time (synchronous)		t_{RXS}	40	—		
	Receive data hold time (synchronous)		t_{RXH}	40	—		
A/D converter	Trigger input setup time		t_{TRGS}	30	—		Figure 24.25

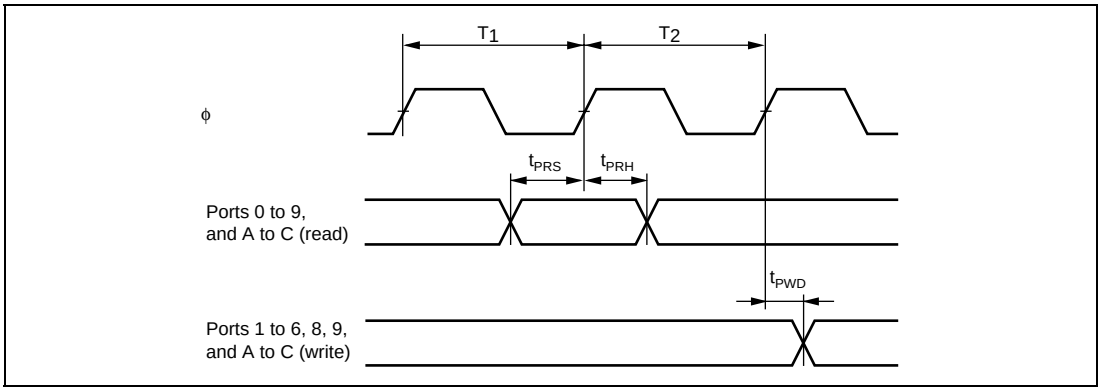


Figure 24.14 I/O Port Input/Output Timing

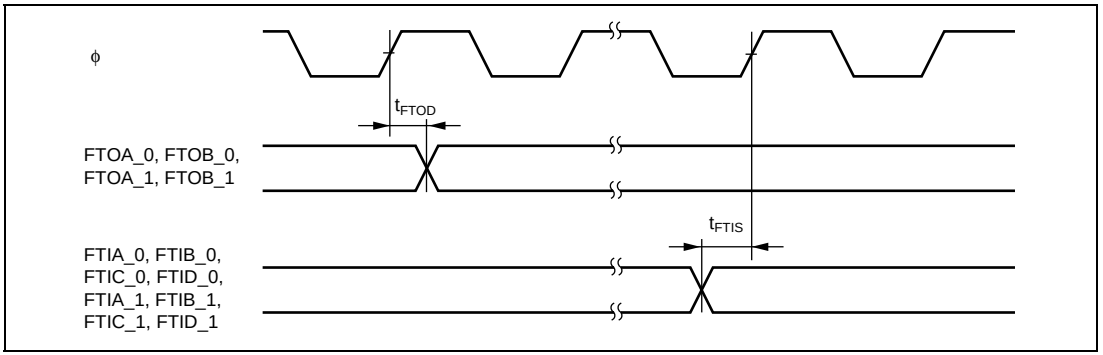


Figure 24.15 FRT Input/Output Timing

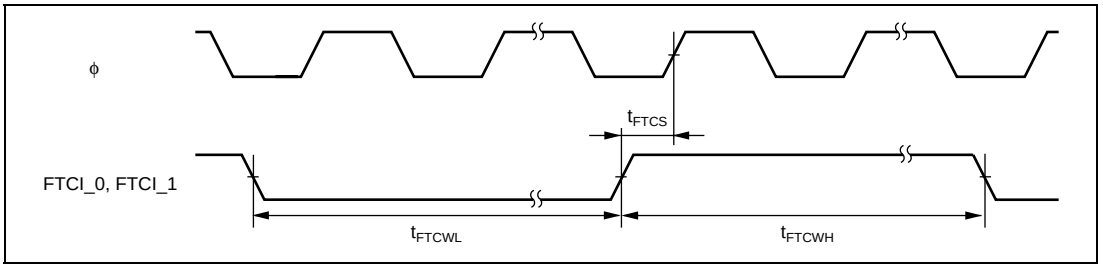


Figure 24.16 FRT Clock Input Timing

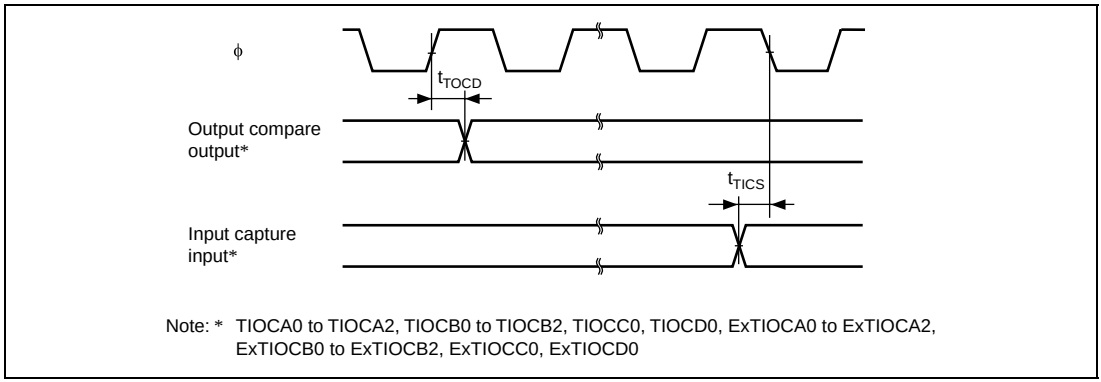


Figure 24.17 TPU Input/Output Timing

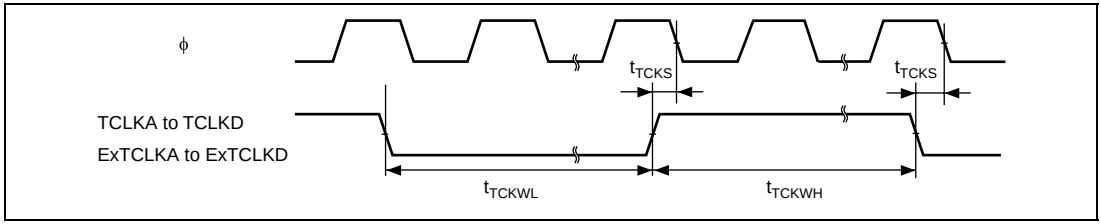


Figure 24.18 TPU Clock Input Timing

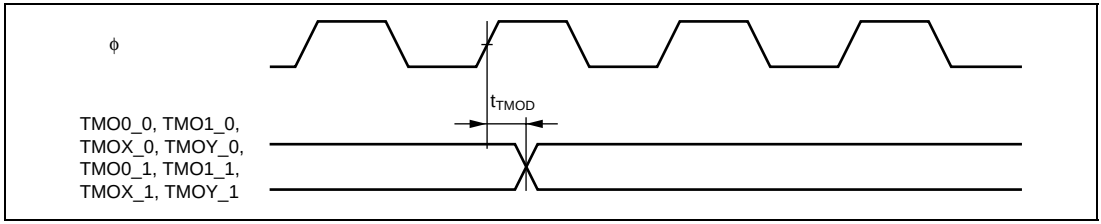


Figure 24.19 8-Bit Timer Output Timing

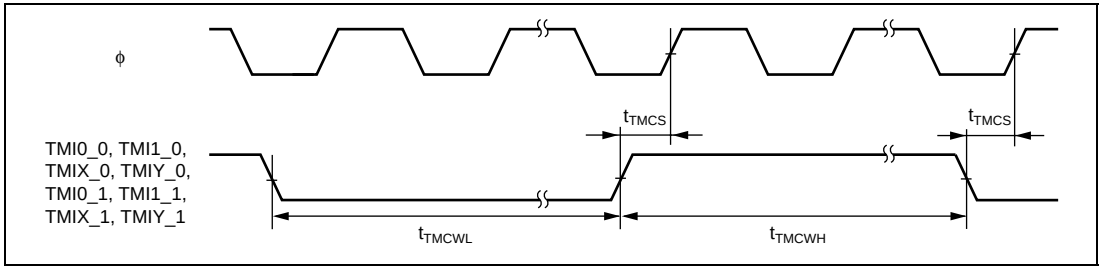


Figure 24.20 8-Bit Timer Clock Input Timing

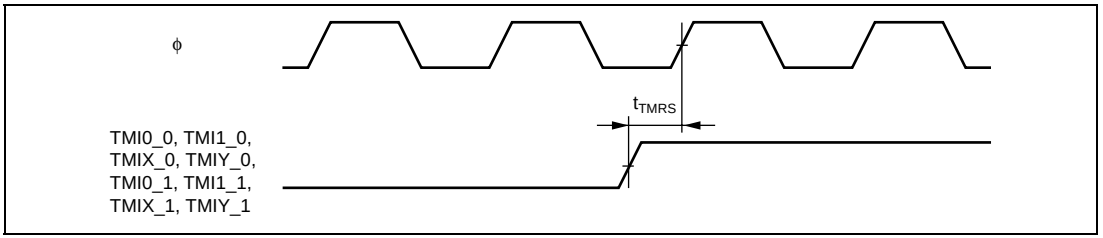


Figure 24.21 8-Bit Timer Reset Input Timing

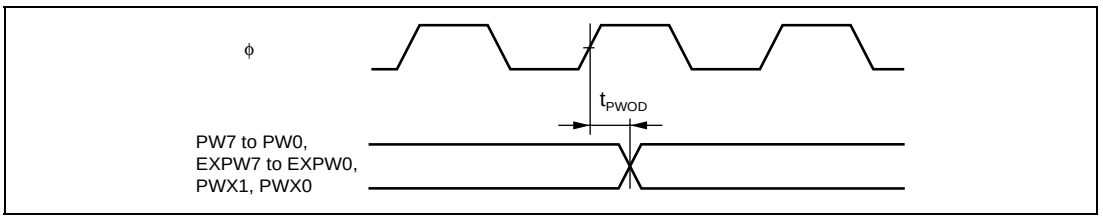


Figure 24.22 PWM, PWMX Output Timing

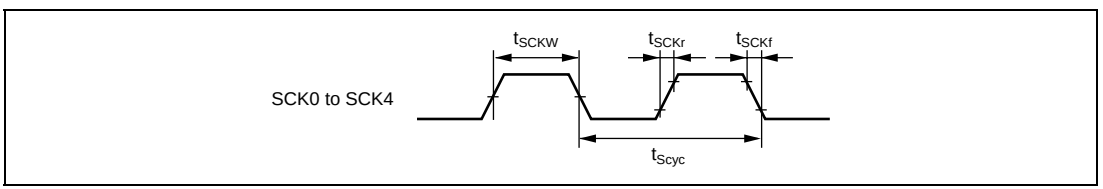


Figure 24.23 SCK Clock Input Timing

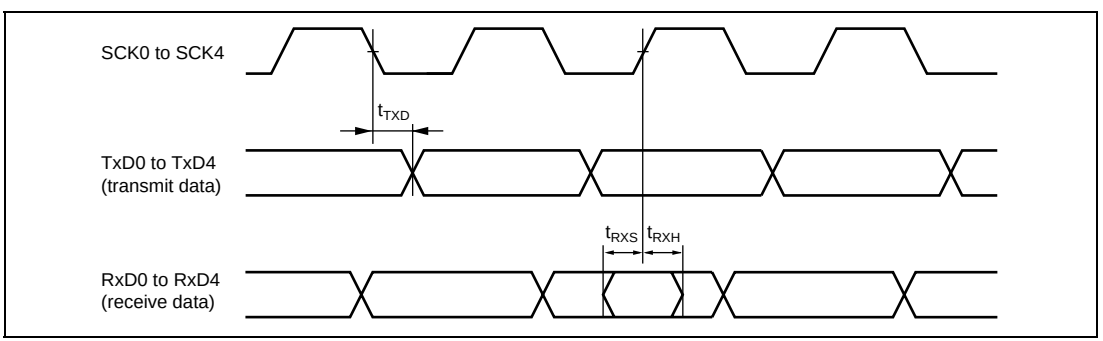


Figure 24.24 SCI Input/Output Timing (Clock Synchronous Mode)

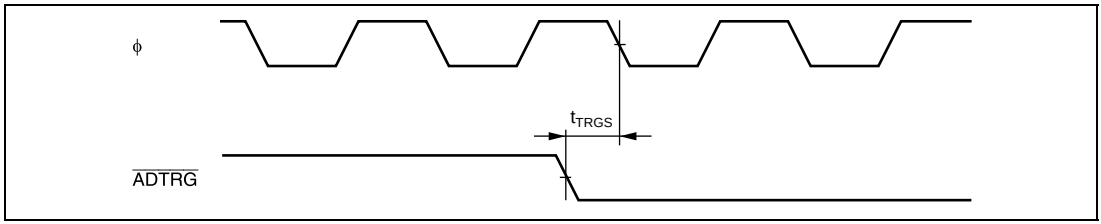
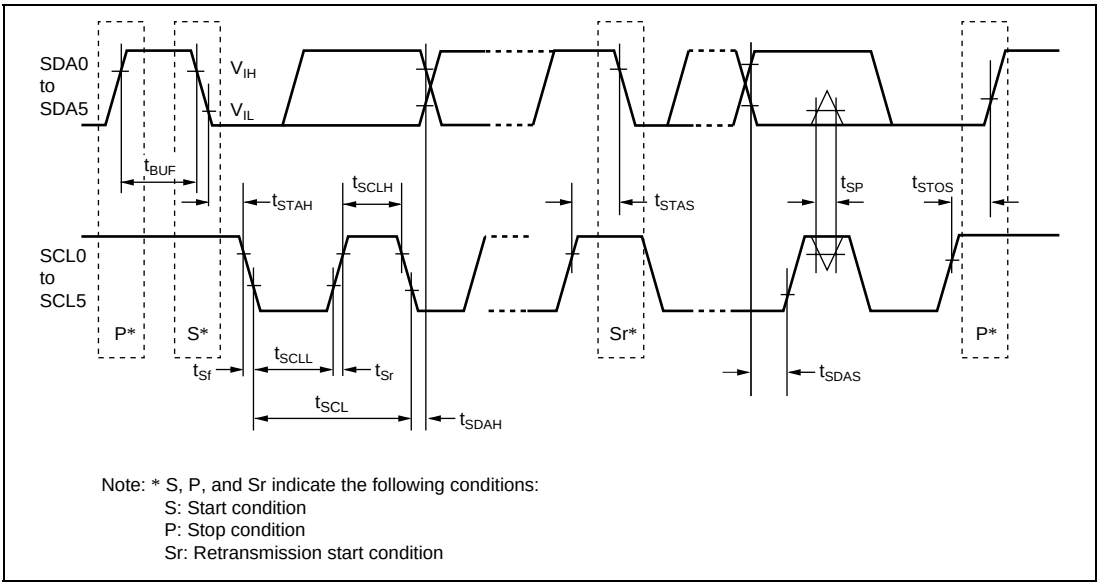


Figure 24.25 A/D Converter External Trigger Input Timing

Table 24.9 I²C Bus Interface Timing

V_{CC} = 3.0 to 3.6 V, V_{SS} = 0.0 V, T_a = –20 to +75°C, unless otherwise indicated.

Item	Symbol	Test Condition	Values			Unit	Reference Figure
			Min.	Typ.	Max.		
SCL input cycle time	t_{SCL}		$12t_{cyc}$ + 600	—	—	ns	Figure 24.26
SCL input high width	t_{SCLH}		$3t_{cyc}$ + 300	—	—	ns	
SCL input low width	t_{SCLL}		$5t_{cyc}$ + 300	—	—	ns	
SCL and SDA input fall time	t_{Sf}		—	—	300	ns	
SCL and SDA input spike pulse removal time	t_{SP}		—	—	$1t_{cyc}$	ns	
SDA input bus-free time	t_{BUF}		$5t_{cyc}$	—	—	ns	
Start condition input hold time	t_{STAH}		$3t_{cyc}$	—	—	ns	
Retransmission start condition input setup time	t_{STAS}		$3t_{cyc}$	—	—	ns	
Setup time for stop condition input	t_{STOS}		$3t_{cyc}$	—	—	ns	
Data-input setup time	t_{SDAS}		$1t_{cyc}$+20	—	—	ns	
Data-input hold time	t_{SDAH}		0	—	—	ns	
Capacitive load of SCL and SDA	c_b		0	—	400	pF	
SCL and SDA output fall time	t_{Sf}		—	—	300	ns	



Note: * S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Retransmission start condition

Figure 24.26 Input/Output Timing of I^2C Bus Interface 3

24.4 A/D Conversion Characteristics

Table 24.10 lists the A/D conversion characteristics.

Table 24.10 A/D Conversion Characteristics (AN15 to AN0 Input: 134/266-State Conversion)

Condition: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, AV_{ref} = 3.0 V to AVCC, V_{SS} = AV_{SS} = 0 V, ϕ = 5 MHz to 20 MHz

	Condition			
Item	**Min.**	**Typ.**	**Max.**	**Unit**
Resolution		10		Bits
Conversion time	6.7*	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5.0	kΩ
Nonlinearity error	—	—	±5.5	LSB
Offset error	—	—	±5.5	
Full-scale error	—	—	±5.5	
Quantization error	—	—	±0.5	
Absolute accuracy	—	—	±6.0	

Note: * Value when using the maximum operating frequency in single mode of 134 states.

24.5 Flash Memory Characteristics

Table 24.11 lists the flash memory characteristics.

Table 24.11 Flash Memory Characteristics

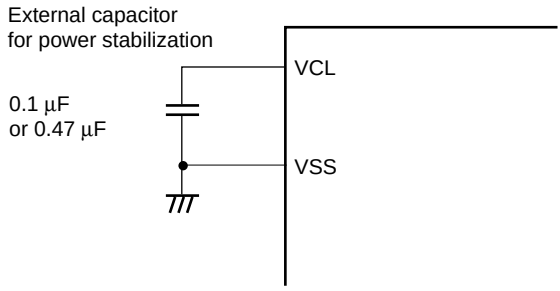
Condition: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, AVref = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = 0°C to +75°C (operating temperature range for programming/erasing)

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 kbytes	
		—	500	5000	ms/32 kbytes	
		—	1000	10000	ms/64 kbytes	
Programming time (total)*1*2*4	Σt_P	—	5	15	s/256 kbytes	Ta = 25°C
Erase time (total)*1*2*4	Σt_E	—	5	15	s/256 kbytes	Ta = 25°C
Programming and erase time (total)*1*2*4	Σt_{PE}	—	10	30	s/256 kbytes	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 does not include data transfer time.
3. This value indicates the minimum number of which the flash memory is 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 capacitor between the VCL pin and VSS pin for stable internal voltage. An example of connection is shown in figure 24.27.



Do not connect Vcc power supply to the VCL pin.
Always connect a capacitor for power stabilization.
Use a ceramic multilayer capacitor (0.1 µF or 0.47 µF)
and place it near the pin.

Figure 24.27 Connection of VCL Capacitor

A. I/O Port States in Each Pin State

Table A.1 I/O Port States in Each Pin State

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Sleep Mode	Program Execution State
Port 0	(EXPE = 1)	T	T	Kept	Kept	Input port
	(EXPE = 0)					
Port 1	(EXPE = 1)	T	T	Kept*	Kept*	Address output/Input port
A7 to A0	(EXPE = 0)					I/O port
Port 2	(EXPE = 1)	T	T	Kept*	Kept*	Address output/I/O port
A15 to A8	(EXPE = 0)					I/O port
Port 3	(EXPE = 1)	T	T	T	T	D15 to D8
D15 to D8	(EXPE = 0)			Kept	Kept	I/O port
Port 4	(EXPE = 1)	T	T	Kept	Kept	I/O port
	(EXPE = 0)					
Port 5	(EXPE = 1)	T	T	Kept	Kept	I/O port
	(EXPE = 0)					
Port 6	(EXPE = 1)	T	T	Kept	Kept	D7 to D0/I/O port
D7 to D0	(EXPE = 0)					I/O port
Port 7	(EXPE = 1)	T	T	T	T	Input port
	(EXPE = 0)					
Port 8	(EXPE = 1)	T	T	Kept	Kept	I/O port
	(EXPE = 0)					
Port 97	(EXPE = 1)	T	T	T/Kept	T/Kept	$\overline{\text{WAIT}}$/I/O port
$\overline{\text{WAIT}}$	(EXPE = 0)			Kept	Kept	I/O port
Port 96	(EXPE = 1)	T	T	[DDR = 1] H	[DDR = 1] Clock output	Clock output/Input port
ϕ	(EXPE = 0)			[DDR = 0] T	[DDR = 0] T	

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Sleep Mode	Program Execution State
Ports 95 to 93	(EXPE = 1)	T	T	H	H	$\overline{AS}$/$\overline{AH}$, $\overline{HWR}$/$\overline{RD}$
$\overline{AS}$, $\overline{AH}$, $\overline{HWR}$, $\overline{RD}$	(EXPE = 0)			Kept	Kept	I/O port
Ports 92 and 91	(EXPE = 1)	T	T	Kept	Kept	I/O port
	(EXPE = 0)					I/O port
Port 90	(EXPE = 1)	T	T	H/Kept	H/Kept	LWR/I/O port
$\overline{LWR}$	(EXPE = 0)			Kept	Kept	I/O port
Port A	(EXPE = 1)	T	T	Kept	Kept	I/O port
	(EXPE = 0)					I/O port
Port B	(EXPE = 1)	T	T	Kept	Kept	I/O port
	(EXPE = 0)					
Port C	(EXPE = 1)	T	T	Kept	Kept	I/O port
	(EXPE = 0)					(PC7 to PC4 are input ports)

[Legend]

H : High level

L : Low level

T : High impedance

Kept : Input ports are in the high-impedance state (when DDR = 0 and PCR = 1, the input pull-up MOS remains on).

Output ports retain their previous state.

Depending on the pins, the on-chip peripheral modules may be initialized and the pins may function as I/O ports determined by DDR and DR.

DDR : Data direction register

Note: * In the case of address output, the last address accessed is retained.

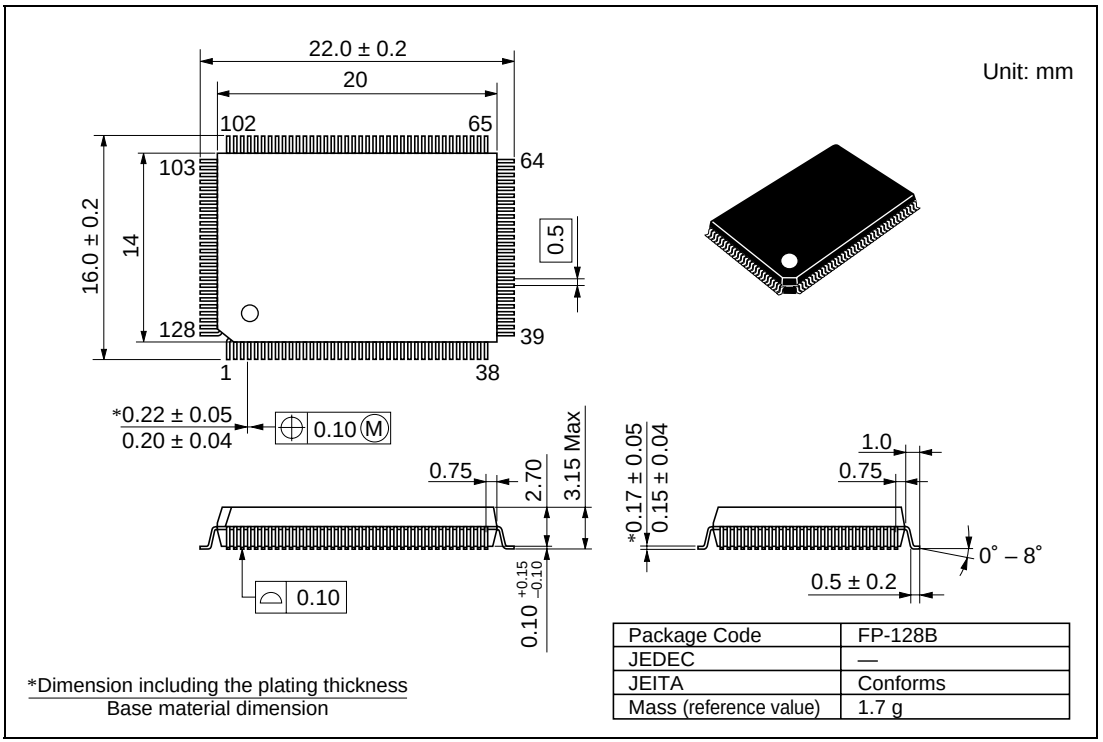
B. Product Lineup

Product Type		Type Code	Mark Code	Package (Code)
H8S/2437	F-ZTAT version	HD64F2437F	DF2437F	128-pin QFP (FP-128B)
H8S/2437	F-ZTAT version	HD64F2437FV*	DF2437FV	128-pin QFP (FP-128B)

Note: * Pb-free version

C. Package Dimensions

For package dimensions, dimensions described in Renesas Semiconductor Packages have priority.



Package Code	FP-128B
JEDEC	—
JEITA	Conforms
Mass (reference value)	1.7 g

Figure C.1 Package Dimensions (FP-128B)

Index

H8S/2437 Group Hardware Manual

Publication Date: Rev.1.00, September 19, 2003
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Technical Documentation & Information Department
Renesas Kodaira Semiconductor Co., Ltd.

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

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

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

Renesas Technology Europe GmbH
Dornacher Str. 3, D-85622 Feldkirchen, Germany
Tel: <49> (89) 380 70 0, Fax: <49> (89) 929 30 11

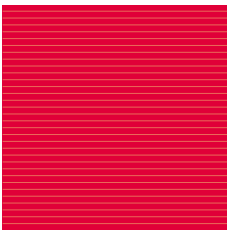
Renesas Technology Hong Kong Ltd.
7/F., North Tower, World Finance Centre, Harbour City, Canton Road, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2375-6836

Renesas Technology Taiwan Co., Ltd.
FL 10, #99, Fu-Hsing N. Rd., Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

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

Renesas Technology Singapore Pte. Ltd.
1, Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

H8S/2437 Group
Hardware Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan

REJ09B0059-0100Z