REJ09B0427-0100

Everywhere you imagine. RENESAS

16

H8S/2607 Group

Hardware Manual

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer
H8S Family/H8S/2600 Series

H8S/2607	HD64F2607
	HD6432607
H8S/2606	HD6432606
H8S/2605	HD6432605

Rev.1.00
Revision Date: Jan. 29, 2008

Renesas Technology
www.renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

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

General Precautions on Handling of Product

1. Treatment of NC Pins

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

2. Treatment of Unused Input Pins

Note: Fix all unused input pins to high or low level.
Generally, the input pins of CMOS products are high-impedance input pins. If unused pins are in their open states, intermediate levels are induced by noise in the vicinity, a pass-through current flows internally, and a malfunction may occur.

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

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

Configuration of This Manual

This manual comprises the following items:

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

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

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

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

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

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

11. Index

Preface

The H8S/2607 Group single-chip microcomputer is made up of the high-speed H8S/2600 CPU as its core, and the peripheral functions required configuring a system. The H8S/2600 CPU has an instruction set that is compatible with the H8/300 and H8/300H CPUs.

Target Users: This manual was written for users who will be using the H8S/2607 Group in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8S/2607 Group to the target users.
Refer to the H8S/2600 Series, H8S/2000 Series Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions, and electrical characteristics.
- In order to understand the details of the CPU's functions
  Read the H8S/2600 Series, H8S/2000 Series Software Manual.
- In order to understand the details of a register when its name is known
  Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in section 22, List of Registers.

  Examples: Register name: The following notation is used for cases when the same or a similar function, e.g. 16-bit timer pulse unit or serial communication, 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.

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

H8S/2607 Group manuals:

Document Title	Document No.
H8S/2607 Group Hardware Manual	This manual
H8S/2600 Series, H8S/2000 Series Software Manual	REJ09B0139

User's manuals for development tools:

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

Contents

Figures

Section 11 Programmable Pulse Generator (PPG)

Section 12 Watchdog Timer (WDT)

Section 13 Serial Communication Interface (SCI)

Section 14 I^2C Bus Interface

Section 15 A/D Converter

Section 16 D/A Converter

Section 17 Motor Control PWM Timer (PWM)

Section 19 ROM

Section 20 Clock Pulse Generator

Section 21 Power-Down Modes

Section 23 Electrical Characteristics

Appendix

Tables

Section 1 Overview

1.1 Features

- High-speed H8S/2600 central processing unit with 16-bit architecture
  - Upward-compatible with H8/300 and H8/300H CPUs on an object level
  - Sixteen 16-bit general registers
  - Sixty-nine basic instructions
- Various peripheral functions
  - PC break controller (PBC)
  - Data transfer controller (DTC)
  - 16-bit timer pulse unit (TPU)
  - Programmable pulse generator (PPG)
  - Watchdog timer (WDT)
  - Asynchronous or clocked synchronous serial communication interface (SCI)
  - I^2C bus interface (IIC)
  - Motor control PWM timer
  - 10-bit A/D converter
  - 8-bit D/A converter
  - Clock pulse generator

- On-chip memory

ROM	Product Code	ROM	RAM	Remarks
Flash memory version	HD64F2607	384 kbytes	16 kbytes	
Masked ROM version	HD6432607	384 kbytes	16 kbytes	
	HD6432606	256 kbytes	16 kbytes	
	HD6432605	128 kbytes	16 kbytes	

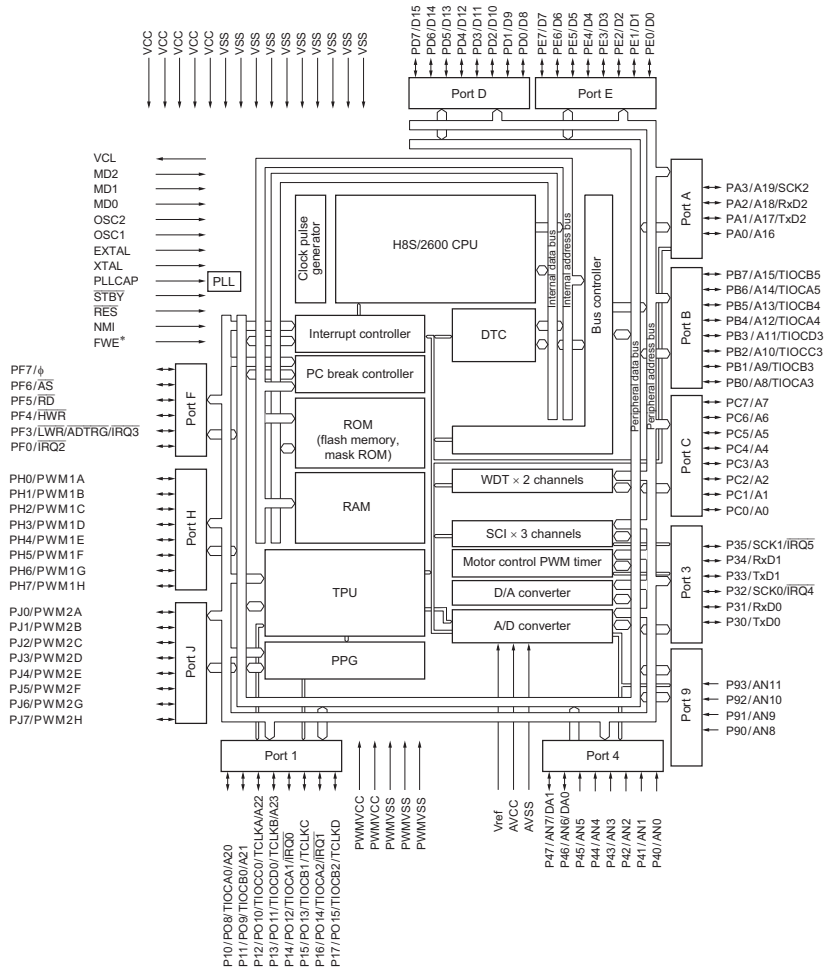
- General I/O ports
  - I/O pins: 72
  - Input pins: 12
- Supports various power-down modes

- Compact package

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

1.2 Block Diagram

Figure 1.1 shows a block diagram.



Notes: 1. Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask version. These functions cannot be used with the other versions. See section 20.6, Subclock Oscillator, for the method of fixing pins OSC1 and OSC2.

2. The FWE pin only applies to the flash memory version. The FWE pin is a NC pin in the mask ROM versions. In the mask ROM version, the FWE pin must be left open or be connected to Vss.

Figure 1.1 Block Diagram

1.3 Pin Assignments

1.3.1 Pin Assignments

Figure 1.2 shows pin assignments.

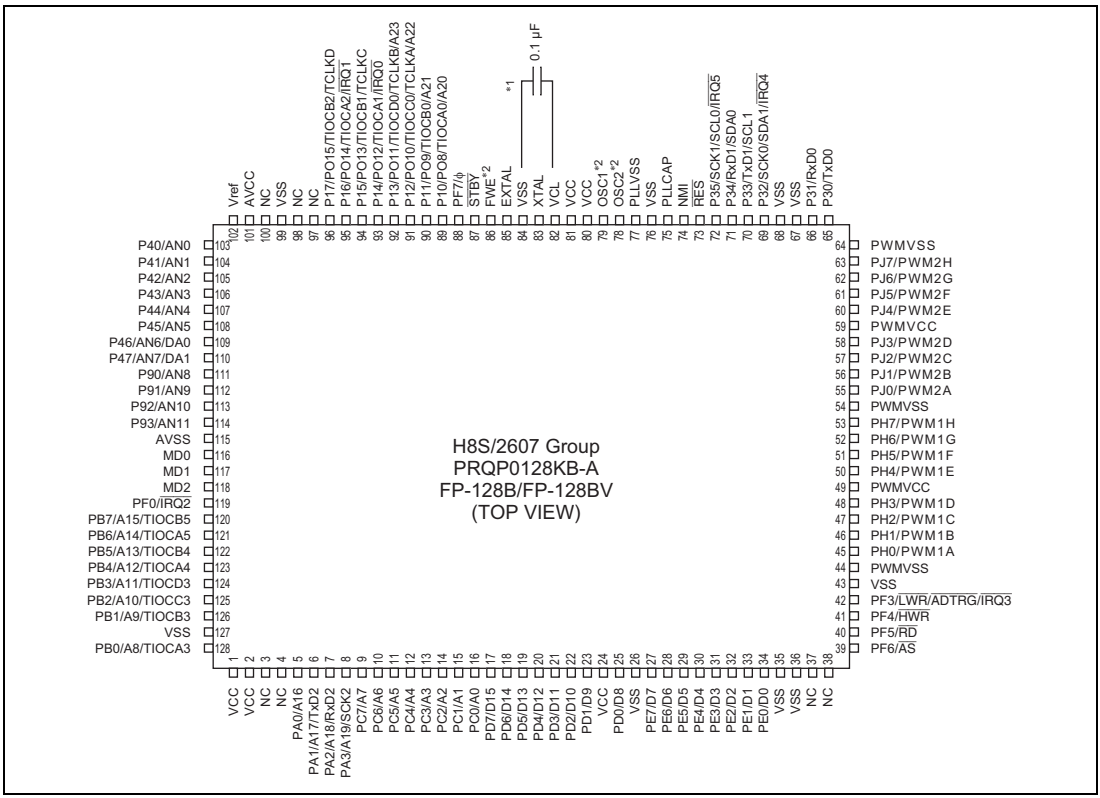


Figure 1.2 Pin Assignments

1.3.2 Pin Functions in Each Operating Mode

Table 1.1 shows the pin functions in each of the operating modes.

Table 1.1 Pin Functions in Each Operating Mode

	Pin Name			
Pin No.	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**
1	VCC	VCC	VCC	VCC
2	VCC	VCC	VCC	VCC
3	NC	NC	NC	NC
4	NC	NC	NC	NC
5	PA0/A16	PA0/A16	PA0/A16	PA0
6	PA1/A17/TxD2	PA1/A17/TxD2	PA1/A17/TxD2	PA1/TxD2
7	PA2/A18/RxD2	PA2/A18/RxD2	PA2/A18/RxD2	PA2/RxD2
8	PA3/A19/SCK2	PA3/A19/SCK2	PA3/A19/SCK2	PA3/SCK2
9	A7	A7	PC7/A7	PC7
10	A6	A6	PC6/A6	PC6
11	A5	A5	PC5/A5	PC5
12	A4	A4	PC4/A4	PC4
13	A3	A3	PC3/A3	PC3
14	A2	A2	PC2/A2	PC2
15	A1	A1	PC1/A1	PC1
16	A0	A0	PC0/A0	PC0
17	D15	D15	D15	PD7
18	D14	D14	D14	PD6
19	D13	D13	D13	PD5
20	D12	D12	D12	PD4
21	D11	D11	D11	PD3
22	D10	D10	D10	PD2
23	D9	D9	D9	PD1
24	VCC	VCC	VCC	VCC
25	D8	D8	D8	PD0
26	VSS	VSS	VSS	VSS

	Pin Name			
Pin No.	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**
27	PE7/D7	PE7/D7	PE7/D7	PE7
28	PE6/D6	PE6/D6	PE6/D6	PE6
29	PE5/D5	PE5/D5	PE5/D5	PE5
30	PE4/D4	PB0/A8	PB0/A8	PE4
31	PE3/D3	PE3/D3	PE3/D3	PE3
32	PE2/D2	PE2/D2	PE2/D2	PE2
33	PE1/D1	PE1/D1	PE1/D1	PE1
34	PE0/D0	PE0/D0	PE0/D0	PE0
35	VSS	VSS	VSS	VSS
36	VSS	VSS	VSS	VSS
37	NC	NC	NC	NC
38	NC	NC	NC	NC
39	$\overline{AS}$	$\overline{AS}$	$\overline{AS}$	PF6
40	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	PF5
41	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	$\overline{PF4}$
42	$\overline{LWR}$/ADTRG/ $\overline{IRQ3}$	PF3/$\overline{LWR}$/ADTRG/ $\overline{IRQ3}$	PF3/$\overline{LWR}$/ADTRG/ $\overline{IRQ3}$	PF3/ $\overline{ADTRG}$/ $\overline{IRQ3}$
43	VSS	VSS	VSS	VSS
44	PWMVSS	PWMVSS	PWMVSS	PWMVSS
45	PH0/PWM1A	PH0/PWM1A	PH0/PWM1A	PH0/PWM1A
46	PH1/PWM1B	PH1/PWM1B	PH1/PWM1B	PH1/PWM1B
47	PH2/PWM1C	PH2/PWM1C	PH2/PWM1C	PH2/PWM1C
48	PH3/PWM1D	PH3/PWM1D	PH3/PWM1D	PH3/PWM1D
49	PWMVCC	PWMVCC	PWMVCC	PWMVCC
50	PH4/PWM1E	PH4/PWM1E	PH4/PWM1E	PH4/PWM1E
51	PH5/PWM1F	PH5/PWM1F	PH5/PWM1F	PH5/PWM1F
52	PH6/PWM1G	PH6/PWM1G	PH6/PWM1G	PH6/PWM1G
53	PH7/PWM1H	PH7/PWM1H	PH7/PWM1H	PH7/PWM1H
54	PWMVSS	PWMVSS	PWMVSS	PWMVSS
55	PJ0/PWM2A	PJ0/PWM2A	PJ0/PWM2A	PJ0/PWM2A
56	PJ1/PWM2B	PJ1/PWM2B	PJ1/PWM2B	PJ1/PWM2B

	Pin Name			
Pin No.	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**
57	PJ2/PWM2C	PJ2/PWM2C	PJ2/PWM2C	PJ2/PWM2C
58	PJ3/PWM2D	PJ3/PWM2D	PJ3/PWM2D	PJ3/PWM2D
59	PWMVCC	PWMVCC	PWMVCC	PWMVCC
60	PJ4/PWM2E	PJ4/PWM2E	PJ4/PWM2E	PJ4/PWM2E
61	PJ5/PWM2F	PJ5/PWM2F	PJ5/PWM2F	PJ5/PWM2F
62	PJ6/PWM2G	PJ6/PWM2G	PJ6/PWM2G	PJ6/PWM2G
63	PJ7/PWM2H	PJ7/PWM2H	PJ7/PWM2H	PJ7/PWM2H
64	PJ2/PWM2C	PJ2/PWM2C	PJ2/PWM2C	PJ2/PWM2C
65	PJ3/PWM2D	PJ3/PWM2D	PJ3/PWM2D	PJ3/PWM2D
66	PWMVcc	PWMVcc	PWMVcc	PWMVcc
67	VSS	VSS	VSS	VSS
68	VSS	VSS	VSS	VSS
69	P32/SCK0/SDA1/$\overline{\text{IRQ4}}$	P32/SCK0/SDA1/$\overline{\text{IRQ4}}$	P32/SCK0/SDA1/$\overline{\text{IRQ4}}$	P32/SCK0/SDA1/$\overline{\text{IRQ4}}$
70	P33/TxD1/SCL1	P33/TxD1/SCL1	P33/TxD1/SCL1	P33/TxD1/SCL1
71	P34/RxD1/SDA0	P34/RxD1/SDA0	P34/RxD1/SDA0	P34/RxD1/SDA0
72	P35/SCK1/SCL0/$\overline{\text{IRQ5}}$	P35/SCK1/SCL0/$\overline{\text{IRQ5}}$	P35/SCK1/SCL0/$\overline{\text{IRQ5}}$	P35/SCK1/SCL0/$\overline{\text{IRQ5}}$
73	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
74	NMI	NMI	NMI	NMI
75	PLLCAP	PLLCAP	PLLCAP	PLLCAP
76	VSS	VSS	VSS	VSS
77	PLLVSS	PLLVSS	PLLVSS	PLLVSS
78	OSC2	OSC2	OSC2	OSC2
79	OSC1	OSC1	OSC1	OSC1
80	VCC	VCC	VCC	VCC
81	VCC	VCC	VCC	VCC
82	VCL	VCL	VCL	VCL
83	XTAL	XTAL	XTAL	XTAL
84	VSS	VSS	VSS	VSS

	Pin Name			
Pin No.	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**
85	XTAL	XTAL	XTAL	XTAL
86	FEW*	FEW*	FEW*	FEW*
87	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$
88	PF7/ϕ	PF7/ϕ	PF7/ϕ	PF7/ϕ
89	P10/PO8/TIOCA0/ A20	P10/PO8/TIOCA0/ A20	P10/PO8/TIOCA0/ A20	P10/PO8/TIOCA0
90	P11/PO9/TIOCB0/ A21	P11/PO9/TIOCB0/ A21	P11/PO9/TIOCB0/ A21	P11/PO9/TIOCB0
91	P12/PO10/TIOCC0/ TCLKA/A22	P12/PO10/TIOCC0/ TCLKA/A22	P12/PO10/TIOCC0/ TCLKA/A22	P12/PO10/TIOCC0/ TCLKA
92	P13/PO11/TIOCD0/ TCLKB/A23	P13/PO11/TIOCD0/ TCLKB/A23	P13/PO11/TIOCD0/ TCLKB/A23	P13/PO11/TIOCD0/ TCLKB
93	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$
94	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC
95	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$
96	P17/PO15/TIOCB2/ TCLKD	P17/PO15/TIOCB2/ TCLKD	P17/PO15/TIOCB2/ TCLKD	P17/PO15/TIOCB2/ TCLKD
97	NC	NC	NC	NC
98	NC	NC	NC	NC
99	VSS	VSS	VSS	VSS
100	NC	NC	NC	NC
101	AVCC	AVCC	AVCC	AVCC
102	Vref	Vref	Vref	Vref
103	P40/AN0	P40/AN0	P40/AN0	P40/AN0
104	P41/AN1	P41/AN1	P41/AN1	P41/AN1
105	P42/AN2	P42/AN2	P42/AN2	P42/AN2
106	P43/AN3	P43/AN3	P43/AN3	P43/AN3
107	P44/AN4	P44/AN4	P44/AN4	P44/AN4
108	P45/AN5	P45/AN5	P45/AN5	P45/AN5
109	P46/AN6/DA0	P46/AN6/DA0	P46/AN6/DA0	P46/AN6/DA0

	Pin Name			
Pin No.	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**
110	P47/AN7/DA1	P47/AN7/DA1	P47/AN7/DA1	P47/AN7/DA1
111	P90/AN8	P90/AN8	P90/AN8	P90/AN8
112	P91/AN9	P91/AN9	P91/AN9	P91/AN9
113	P92/AN10	P92/AN10	P92/AN10	P92/AN10
114	P93/AN11	P93/AN11	P93/AN11	P93/AN11
115	AVSS	AVSS	AVSS	AVSS
116	MD0	MD0	MD0	MD0
117	MD1	MD1	MD1	MD1
118	MD2	MD2	MD2	MD2
119	PF0/$\overline{\text{IRQ2}}$	PF0/$\overline{\text{IRQ2}}$	PF0/$\overline{\text{IRQ2}}$	PF0/$\overline{\text{IRQ2}}$
120	PB7/A15/TIOCB5	PB7/A15/TIOCB5	PB7/A15/TIOCB5	PB7/A15/TIOCB5
121	PB6/A14/TIOCA5	PB6/A14/TIOCA5	PB6/A14/TIOCA5	PB6/A14/TIOCA5
122	PB5/A13/TIOCB4	PB5/A13/TIOCB4	PB5/A13/TIOCB4	PB5/A13/TIOCB4
123	PB4/A12/TIOCA4	PB4/A12/TIOCA4	PB4/A12/TIOCA4	PB4/A12/TIOCA4
124	PB3/A11/TIOCD3	PB3/A11/TIOCD3	PB3/A11/TIOCD3	PB3/A11/TIOCD3
125	PB2/A10/TIOCC3	PB2/A10/TIOCC3	PB2/A10/TIOCC3	PB2/A10/TIOCC3
126	PB1/A9/TIOCB3	PB1/A9/TIOCB3	PB1/A9/TIOCB3	PB1/A9/TIOCB3
127	VSS	VSS	VSS	VSS
128	PB0/A8/TIOCA3	PB0/A8/TIOCA3	PB0/A8/TIOCA3	PB0/A8/TIOCA3

Notes: The NC pins should be connected to VSS, or left open.

* The FEW pin is only for Flash memory version, and functions as an NC pin for Mask ROM version. For Mask ROM version, this pin should be left open or connected to VSS

1.3.3 Pin Functions

Table 1.2 shows pin functions.

Table 1.2 Pin Functions

Type	Symbol	Pin No.	I/O	Name and Function
Power supply	Vcc	1, 2, 24, 81	Input	Power supply pins These pins should be connected to the system power supply.
	Vss	26, 35, 36, 43, 67, 68, 76, 84, 99, 127	Input	Ground pins These pins should be connected to the system power supply (0 V).
	VCL	82	Output	Connect to Vss via a 0.1 μF capacitor (which should be located near the pin). Do not connect this pin to an external power supply.
Clock	PLLVss	77	Input	Ground pin for on-chip PLL oscillator
	PLLCAP	75	Input	External capacitance pin for on-chip PLL oscillator
	XTAL	83	Input	Connects to a crystal oscillator. See section 20, Clock Pulse Generator, for typical connection diagrams for a crystal oscillator.
	EXTAL	85	Input	The EXTAL pin can also input an external clock.
	OSC1	79	Input	Connects to a 32.768-kHz crystal oscillator. See section 20, Clock Pulse Generator, for typical connection diagrams for a crystal oscillator.
	OSC2	78	Input	
	φ	88	Output	Supplies the system clock to an external device.

Type	Symbol	Pin No.	I/O	Name and Function
Operating mode control	MD2 to MD0	118 to 116	Input	These pins set the operating mode. These pins should not be changed while this LSI is operating.
System control	$\overline{\text{RES}}$	73	Input	Reset pin When this pin is driven low, the chip is reset.
	$\overline{\text{STBY}}$	87	Input	When this pin is driven low, a transition is made to hardware standby mode.
	FEW*	86	Input	Pin for flash memory
Interrupts	NMI	74	Input	Requests a nonmaskable interrupt. When this pin is not used, it should be fixed high.
	$\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$	72, 69, 42, 119, 95, 93	Input	These pins request a maskable interrupt.
Address bus	A23 to A0	92 to 89, 8 to 5, 120 to 126, 128, 9 to 16	Output	These pins output an address.
Data bus	D15 to D0	17 to 23, 25, 27 to 34	I/O	These pins constitute a bidirectional data bus.
Bus control	$\overline{\text{AS}}$	39	Output	When this pin is low, it indicates that address output on the address bus is enabled.
	$\overline{\text{RD}}$	40	Output	When this pin is low, it indicates that the external address space can be read.
	$\overline{\text{HWR}}$	41	Output	A strobe signal that writes to external space and indicates that the upper half (D15 to D8) of the data bus is enabled.
	$\overline{\text{LWR}}$	42	Output	A strobe signal that writes to external space and indicates that the lower half (D7 to D0) of the data bus is enabled.

Type	Symbol	Pin No.	I/O	Name and Function
16-bit timer pulse unit (TPU)	TCLKD to TCLKA	96, 94, 92, 91	Input	These pins input an external clock.
	TIOCA0, TIOCB0, TIOCC0, TIOCD0	89 90 91 92	I/O	TGR0A to TGR0D input capture input, output compare output, or output PWM.
	TIOCA1, TIOCB1	93 94	I/O	TGR1A and TGR1B input capture input, output compare output, or output PWM.
	TIOCA2, TIOCB2	95 96	I/O	TGR2A and TGR2B input capture input, output compare output, or output PWM.
	TIOCA3, TIOCB3, TIOCC3, TIOCD3	128 126 125 124	I/O	TGR3A to TGR3D input capture input, output compare output, or output PWM.
	TIOCA4, TIOCB4	123 122	I/O	TGR4A and TGR4B input capture input, output compare output, or output PWM.
	TIOCA5, TIOCB5	121 120	I/O	TGR5A and TGR5B input capture input, output compare output, or output PWM.
Programmable pulse generator (PPG)	PO15 to PO8	96 to 89	Output	Pulse output pins
Serial communication interface (SCI)/ Smart Card interface	TxD2, TxD1, TxD0	6 70 65	Output	Transmit data output pins
	RxD2, RxD1, RxD0	7 71 66	Input	Receive data input pins
	SCK2, SCK1, SCK0	8 72 69	I/O	Clock I/O pins

Type	Symbol	Pin No.	I/O	Name and Function
I^2C bus interface (IIC)	SCL0 SCL1	72 70	I/O	I^2C clock I/O pins that can drive the bus The output type of SCL0 is NMOS open-drain.
	SDA0 SDA1	71 69	I/O	I^2C data I/O pins that can drive the bus The output type of SDA0 is NMOS open-drain.
A/D converter	AN11 to AN0	114 to 103	Input	Analog input pins
	$\overline{\text{ADTRG}}$	42	Input	Pin for input of an external trigger to start A/D conversion
D/A converter	DA1 DA0	110 109	Output	D/A converter analog output pins
A/D converter/ D/A converter	AVcc	101	Input	Power supply pin for A/D converter or D/A converter If the A/D converter or D/A converter is not used, connect this pin to the system power supply (+5 V).
	AVss	115	Input	Ground pin for for A/D converter or D/A converter Connect this pin to the system power supply (0 V).
	Vref	102	Input	Reference voltage input pin for A/D converter or D/A converter If the A/D converter is not used, connect this pin to the system power supply (+5 V).
Motor control PWM timer	PWM1A to PWM1H	53 to 50, 48 to 45	Output	Motor control PWM timer channel 1 output pins
	PWM2A to PWM2H	63 to 60, 58 to 55	Output	Motor control PWM timer channel 2 output pins
	PWMVCC	49, 59	Input	Motor control PWM power supply pins If the motor control is not used, connect these pins to the system power supply (+5V).
	PWMVSS	44, 54, 64	Input	Motor control PWM ground pins Connect these pins to the system power supply (0V).

Type	Symbol	Pin No.	I/O	Name and Function
I/O ports	P17 to P10	96 to 89	I/O	8-bit I/O pins
	P35 to P30	72 to 69, 66, 65	I/O	6-bit I/O pins
	P47 to P40	110 to 103	Input	8-bit I/O pins
	P93 to P90	114 to 111	Input	4-bit input pins
	PA3 to PA0	8 to 5	I/O	4-bit I/O pins
	PB7 to PB0	120 to 126, 128	I/O	8-bit I/O pins
	PC7 to PC0	9 to 16	I/O	8-bit I/O pins
	PD7 to PD0	17 to 23, 25	I/O	8-bit I/O pins
	PE7 to PE0	27 to 34	I/O	8-bit I/O pins
	PF7 to PF2, PF0	88, 39 to 42, 119	I/O	6-bit I/O pins
	PH7 to PH0	53 to 50, 48 to 45	I/O	8-bit I/O pins
	PJ7 to PJ0	63 to 60, 58 to 55	I/O	8-bit I/O pins

Note: * The FEW pin is only for Flash memory version, and functions as an NC pin for Mask ROM version. For Mask ROM version, this pin should be left open or connected to VSS.

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 CPUs 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-and-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 the 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 shown below.

- Register configuration
  The MAC register is supported by the H8S/2600 CPU only.
- Basic instructions
  The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported by the H8S/2600 CPU only.
- 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, and 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 extended 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-and-accumulate instruction has been added.
  - Two-bit shift 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.

2.1.3 Differences from H8/300H CPU

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

- More control registers
  - 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-and-accumulate instruction has been added.
  - Two-bit shift 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 vector table and stack have the same structure as in the H8/300 CPU.

- Address Space
  Linear access to a 64-kbyte maximum address space is provided.
- 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 Vector Table and Memory Indirect Branch Addresses
  In normal mode the top area starting at H'0000 is allocated to the exception vector table. One branch address is stored per 16 bits. The exception vector table structure in normal mode is shown in figure 2.1. For details of the exception vector table, see section 4, Exception Handling.
  The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In normal mode the operand is a 16-bit word operand, providing a 16-bit branch address. Branch addresses can be stored in the area from H'0000 to H'00FF. Note that the first part of this range is also used for the exception 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.

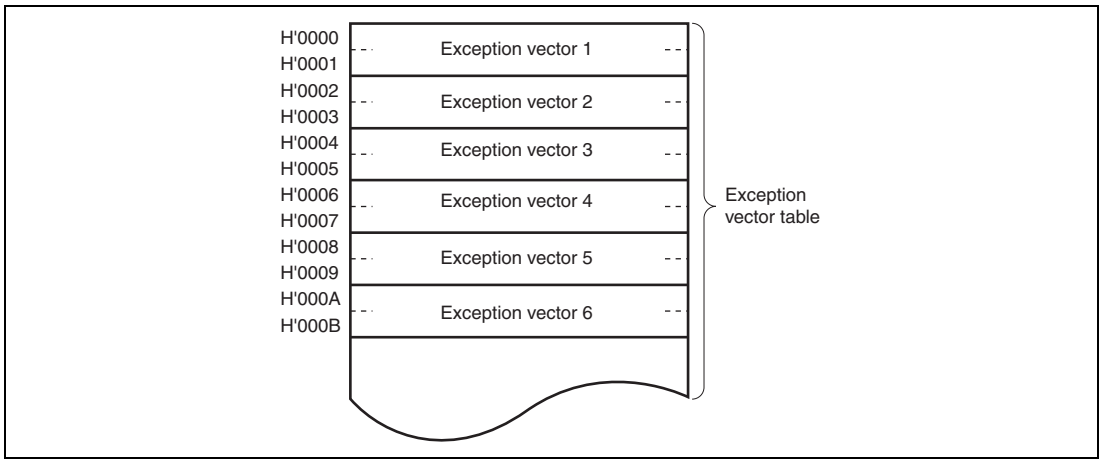


Figure 2.1 Exception Vector Table (Normal Mode)

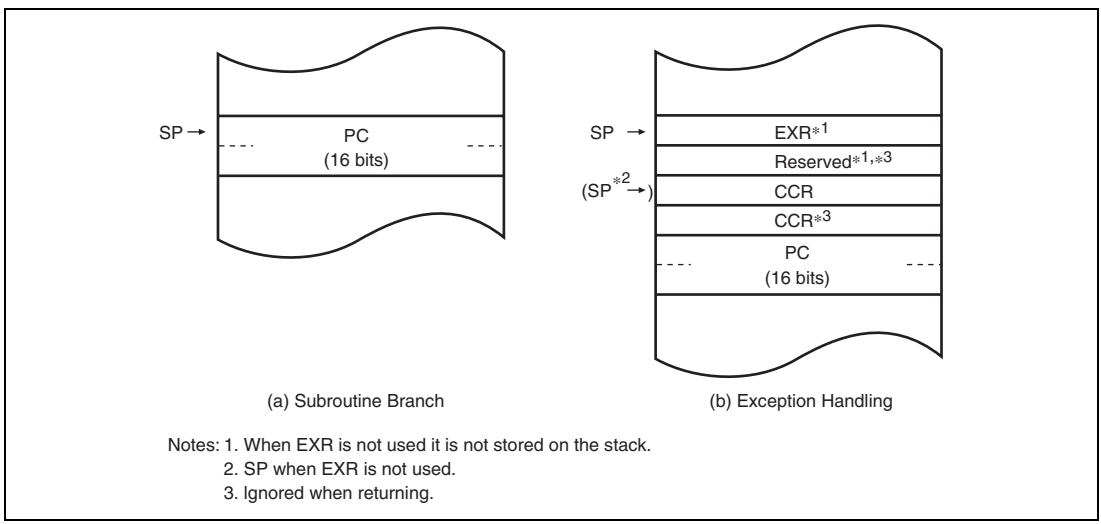


Figure 2.2 Stack Structure in Normal Mode

2.2.2 Advanced Mode

- Address Space

  Linear access to a 16-Mbyte maximum address space is provided.
- 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 Vector Table and Memory Indirect Branch Addresses

  In advanced mode, the top area starting at H'00000000 is allocated to the exception vector table in 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 vector table, see section 4, Exception Handling.

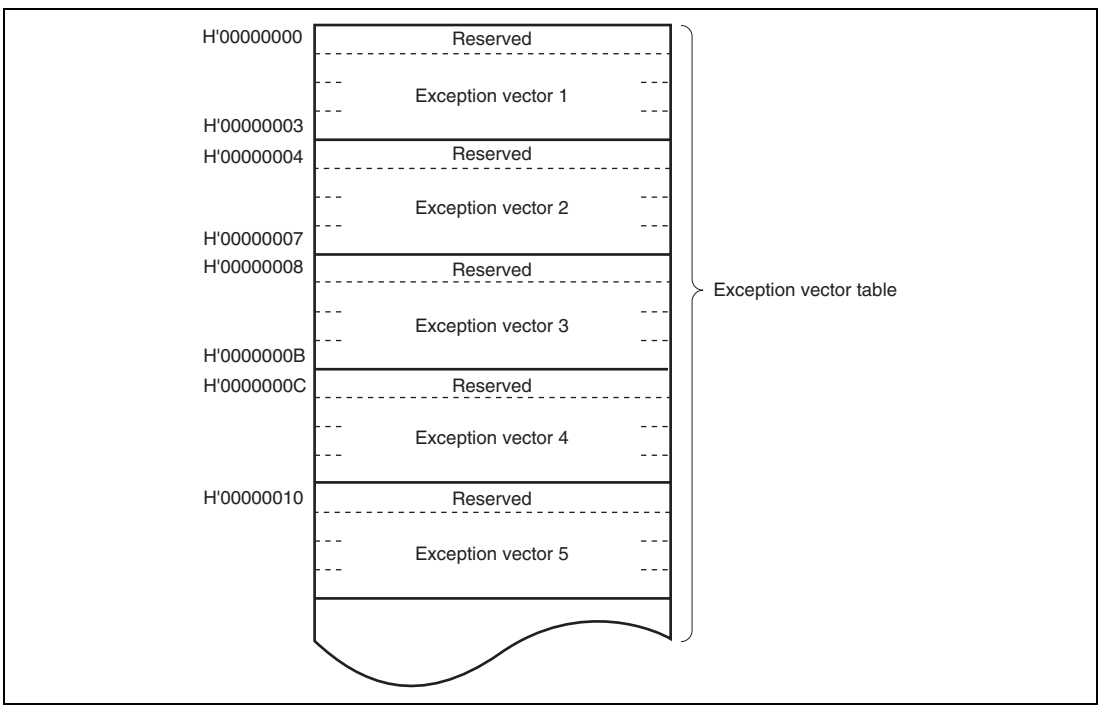
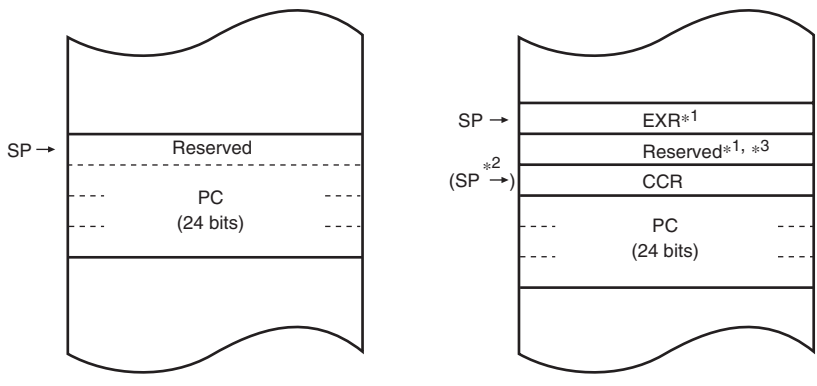


Figure 2.3 Exception Vector Table (Advanced Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In advanced mode the operand is a 32-bit longword operand, providing a 32-bit branch address. The upper 8 bits of these 32 bits is 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 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. When EXR is not pushed onto the stack in interrupt control mode 0. For details, see section 4, Exception Handling.



(a) Subroutine Branch (b) 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 for 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.

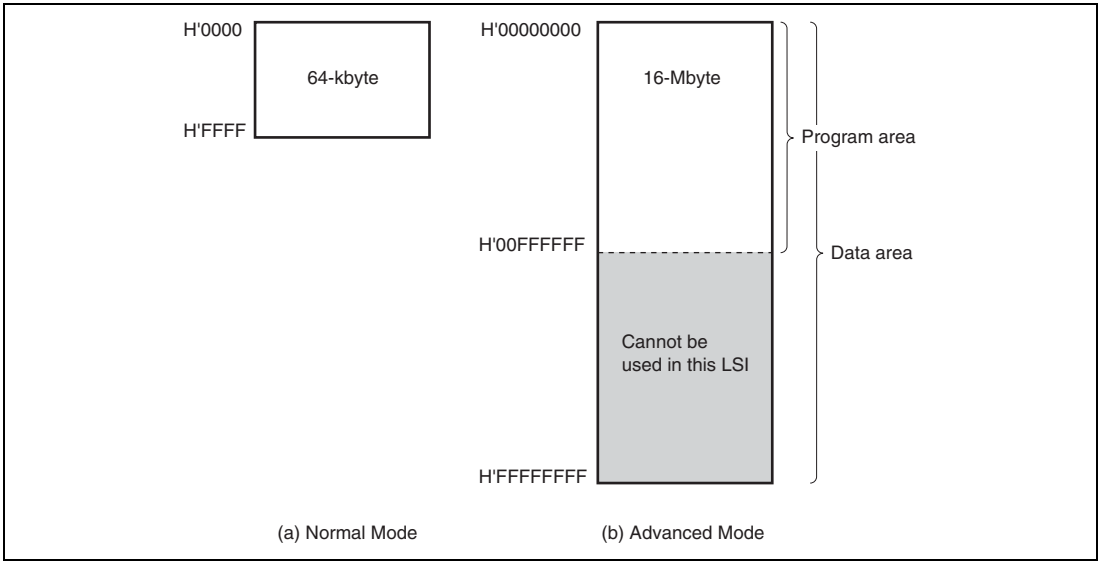


Figure 2.5 Memory Map

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. The control registers are a 24-bit program counter (PC), an 8-bit extended control register (EXR), an 8-bit condition code register (CCR), and a 64-bit multiply-accumulate register (MAC).

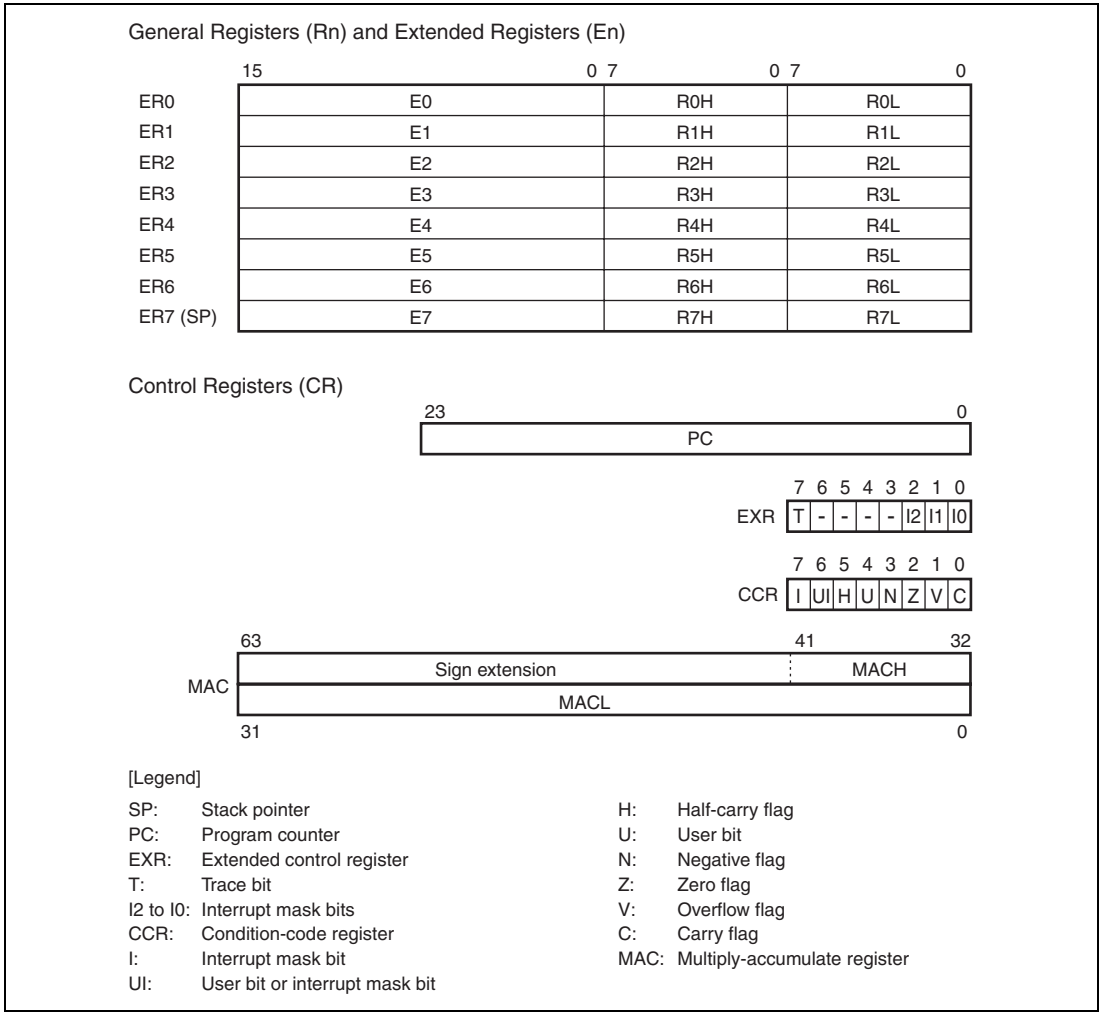


Figure 2.6 CPU Registers

2.4.1 General Registers

The H8S/2600 CPU has eight 32-bit general registers. These general registers are all functionally identical and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. Figure 2.7 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 of sixteen 16-bit registers. The E registers (E0 to E7) are also referred to as extended registers.

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

The usage of each register can be selected independently.

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.

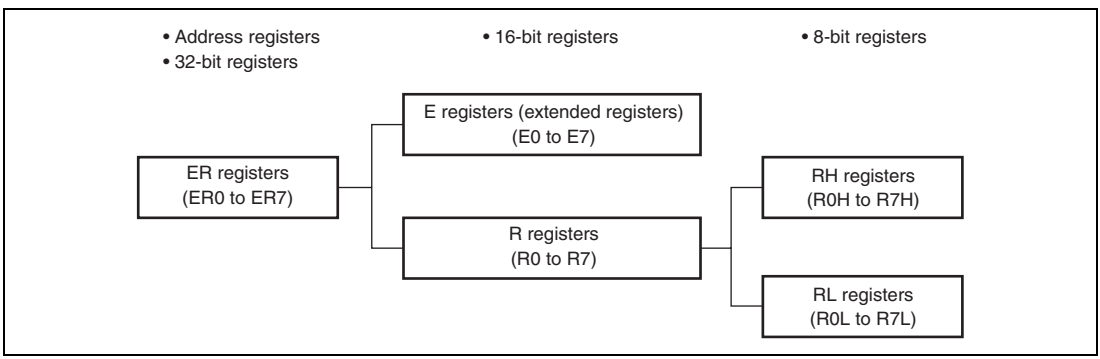


Figure 2.7 Usage of General Registers

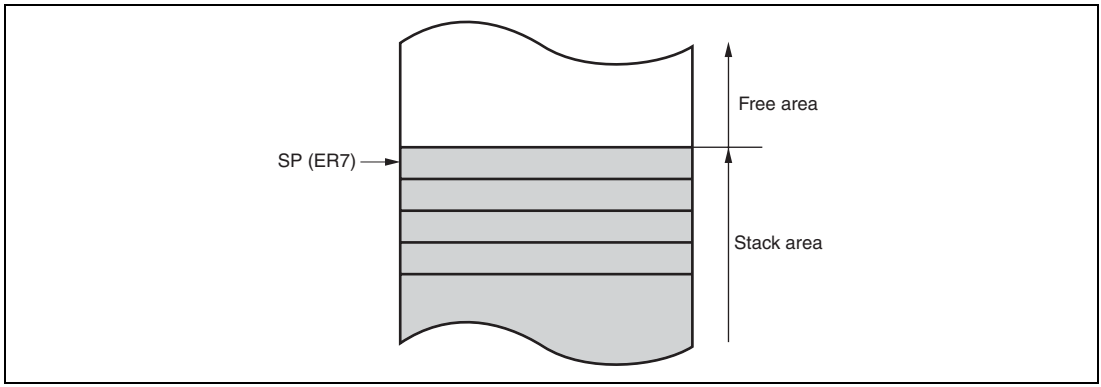


Figure 2.8 Stack

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 Control Register (EXR)

EXR is an 8-bit register that manipulates the LDC, STC, ANDC, ORC, and XORC instructions. When these instructions, except for the STC instruction, are 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 is generated 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 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 read or written 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 read or written 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 at other times.
0	C	Undefined	R/W	Carry Flag Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by: • Add instructions, to indicate a carry • Subtract instructions, to indicate a borrow • Shift and rotate instructions, to indicate a carry The carry flag is also used as a bit accumulator by bit manipulation instructions.

2.4.5 Multiply-Accumulate Register (MAC)

This 64-bit register stores the results of multiply-and-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 Registers

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

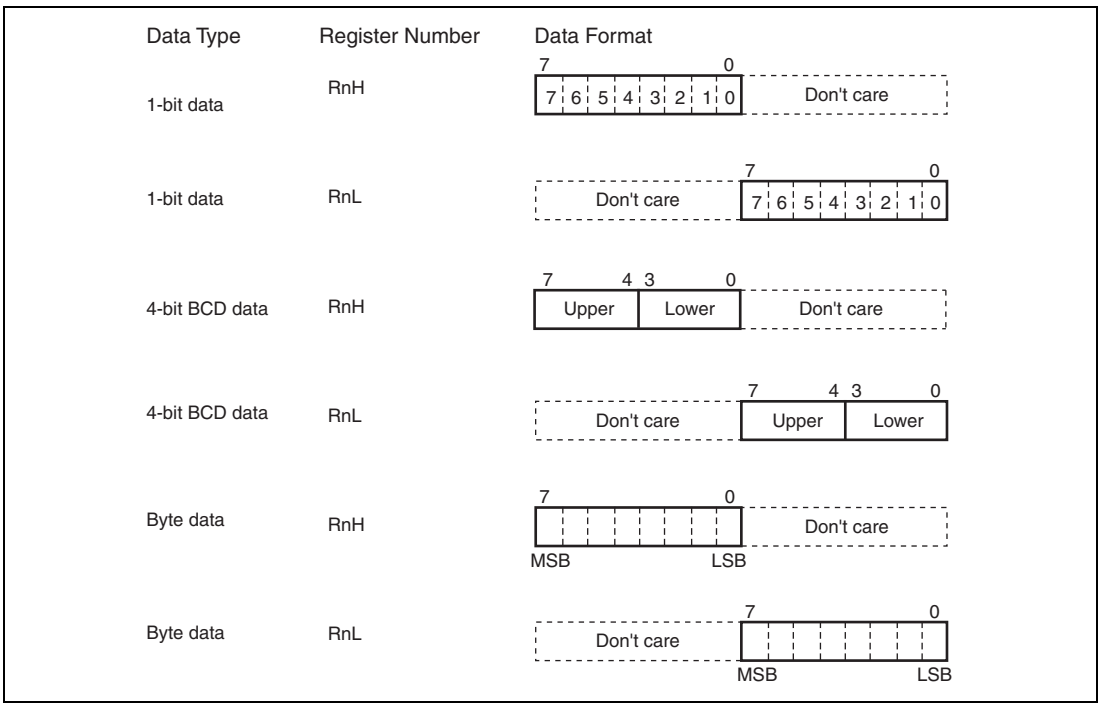


Figure 2.9 General Register Data Formats (1)

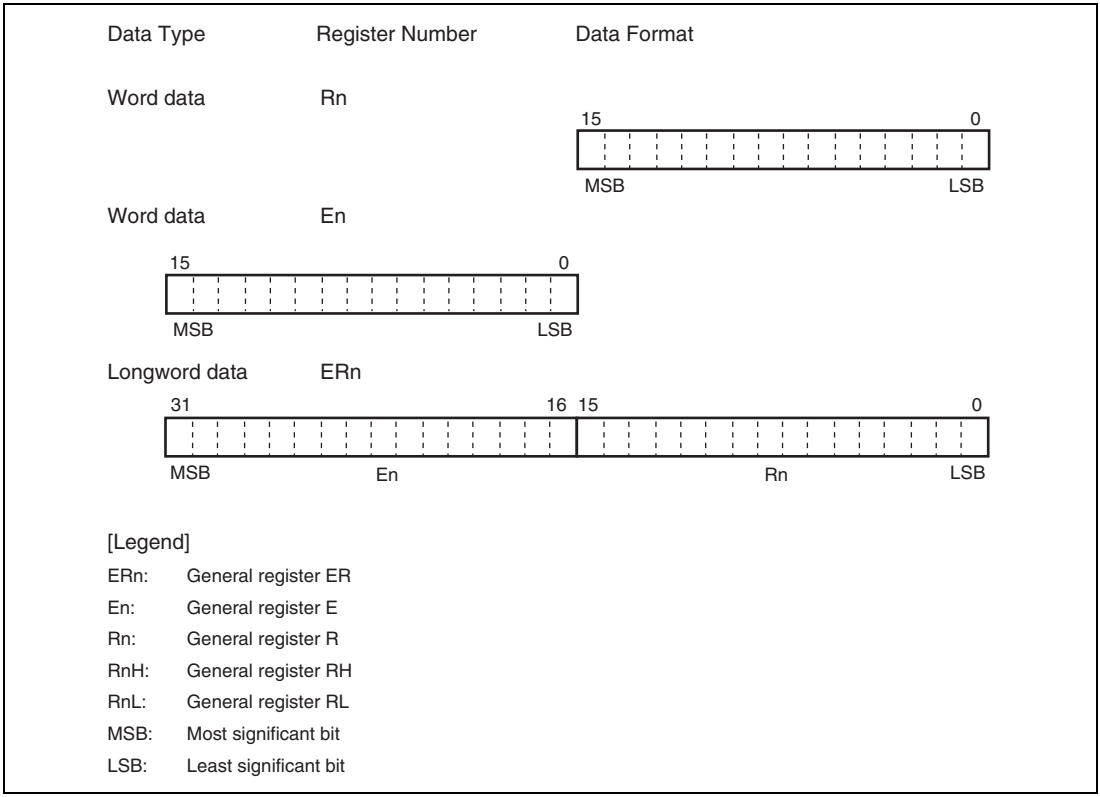


Figure 2.9 General Register Data Formats (2)

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, however word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, an address error does not occur, however the least significant bit of the address is regarded as 0, so access begins the preceding address. This also applies to instruction fetches.

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

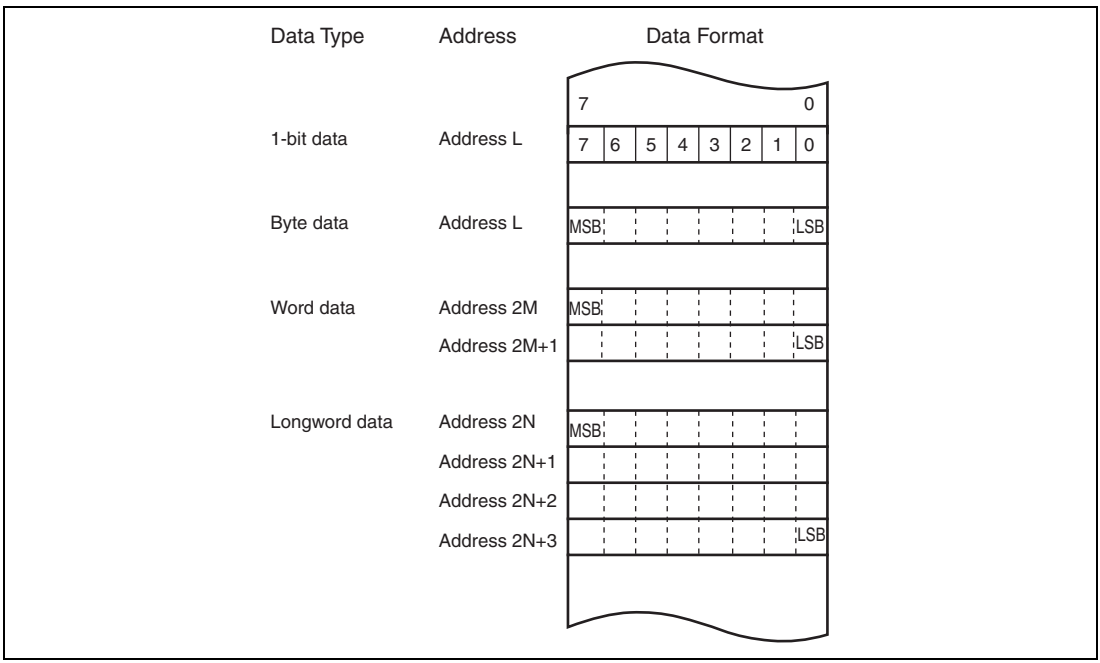


Figure 2.10 Memory Data Formats

2.6 Instruction Set

The H8S/2600 CPU has 69 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 operation	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; W-word; L-longword.

1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W Rn,@-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn,@-SP.
2. 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 control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical XOR
→	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) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
MOVFPE	B	Cannot be used in this LSI.
MOVTPE	B	Cannot be used in this LSI.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, ERn.
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @–SP. PUSH.L ERn is identical to MOV.L ERn, @–SP.
LDM	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: * 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 on byte data in two general registers, or on immediate data and data in a general register.
INC DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (Byte operands can be incremented or decremented by 1 only.)
ADDS SUBS	L	Rd ± 1 → Rd, Rd ± 2 → Rd, Rd ± 4 → Rd Adds or subtracts the value 1, 2, or 4 to or from data in a 32-bit register.
DAA DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction result in a general register by referring to the CCR to produce 4-bit BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
MULXS	B/W	Rd × Rs → Rd Performs signed multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
DIVXU	B/W	Rd ÷ Rs → Rd Performs unsigned division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.

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

Table 2.4 Arithmetic Operations Instructions (2)

Instruction	Size*	Function
DIVXS	B/W	Rd ÷ Rs → Rd Performs signed division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.
CMP	B/W/L	Rd – Rs, Rd – #IMM Compares data in a general register with data in another general register or with immediate data, and sets 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.

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

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: * 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 shifts are possible.
SHLL SHLR	B/W/L	Rd (shift) → Rd Performs a logical shift on general register contents. 1-bit or 2-bit shifts are possible.
ROTL ROTR	B/W/L	Rd (rotate) → Rd Rotates general register contents. 1-bit or 2-bit rotations are possible.
ROTXL ROTXR	B/W/L	Rd (rotate) → Rd Rotates general register contents through the carry flag. 1-bit or 2-bit rotations are possible.

Note: * 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: * 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 XORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIXOR	B	C ⊕ [~(<bit-No.> of <EAd>)] → C XORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BLD	B	(<bit-No.> of <EAd>) → C Transfers a specified bit in a general register or memory operand to the carry flag.
BILD	B	~(<bit-No.> of <EAd>) → C Transfers the inverse of a specified bit in a general register or memory operand to the carry flag. The bit number is specified by 3-bit immediate data.
BST	B	C → (<bit-No.> of <EAd>) Transfers the carry flag value to a specified bit in a general register or memory operand.
BIST	B	~C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data.

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

Table 2.8 Branch Instructions

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

Table 2.9 System Control Instructions

Instruction	Size*	Function
TRAPA	—	Starts trap-instruction exception handling.
RTE	—	Returns from an exception-handling routine.
SLEEP	—	Causes a transition to a power-down state.
LDC	B/W	(EAs) → CCR, (EAs) → EXR Moves general register or memory contents or immediate data to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 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 XORs the CCR or EXR contents with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

Note: * 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 CPU instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (op field), a register field (r field), an effective address extension (EA field), 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, and 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.

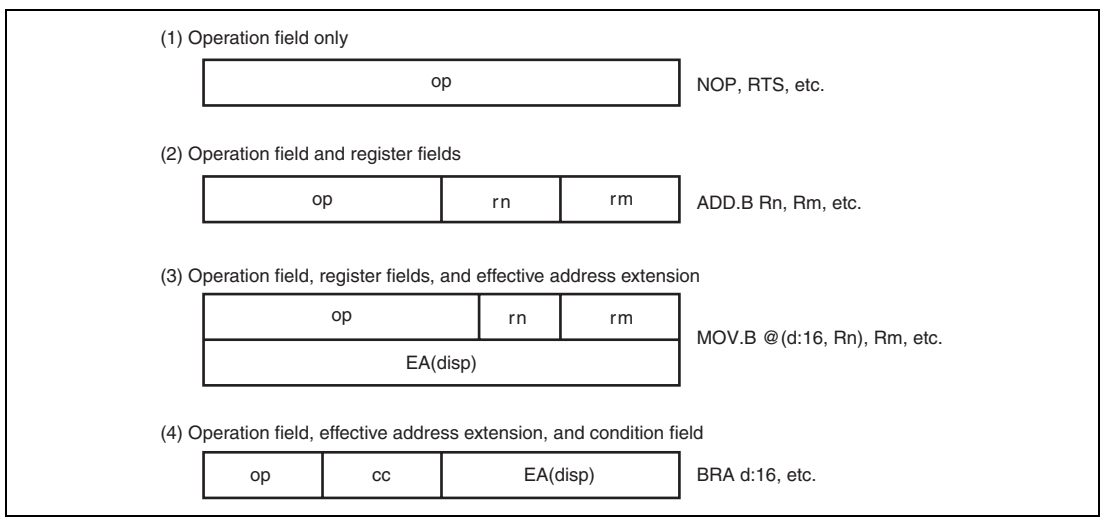


Figure 2.11 Instruction Formats (Examples)

2.7 Addressing Modes and Effective Address Calculation

The H8S/2600 CPU supports the eight addressing modes listed in table 2.11. Each instruction uses a subset of these addressing modes. Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions use register direct, register indirect, or the absolute addressing mode 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 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, 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 the word or longword transfer instructions, the register value should be even.

Register indirect with pre-decrement—@-ERn: The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the register field in the instruction code, and the result is the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word transfer instruction, or 4 for longword transfer instruction. For the word or longword transfer instructions, the register value should be even.

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

The instruction code contains the absolute address of a memory operand. The absolute address may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24), or 32 bits long (@aa:32). Table 2.12 indicates the accessible absolute address ranges.

To access data, the absolute address should be 8 bits (@aa:8), 16 bits (@aa:16), or 32 bits (@aa:32) long. For an 8-bit absolute address, the upper 24 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address the upper 16 bits are a sign extension. 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: Normal mode is not available in this LSI.

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

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

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some bit manipulation instructions contain 3-bit immediate data in the instruction code, specifying a bit number. 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 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 0 (H'00).

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

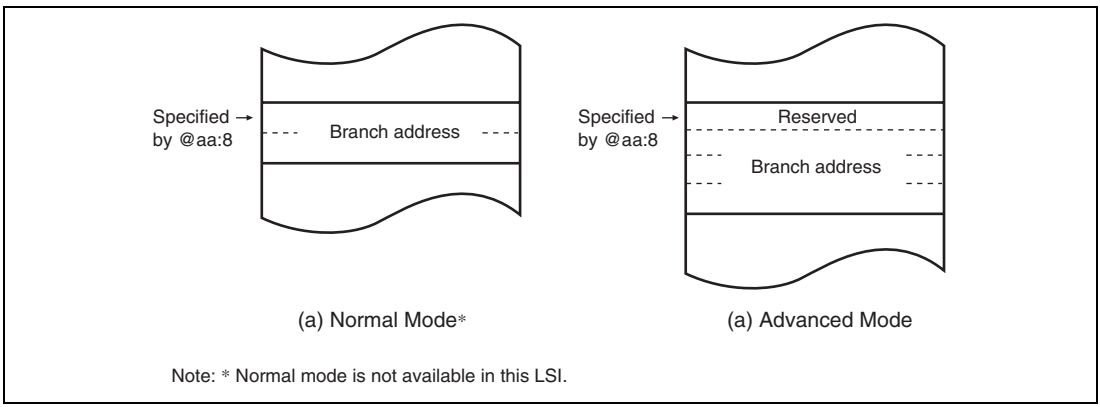


Figure 2.12 Branch Address Specification in Memory Indirect Mode

2.7.9 Effective Address Calculation

Table 2.13 indicates how effective addresses 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)

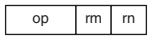
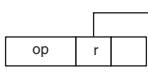
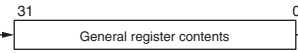
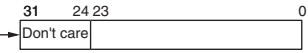
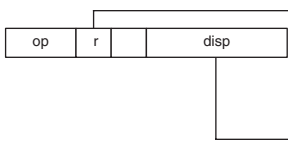
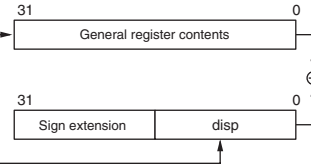
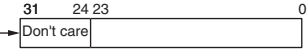
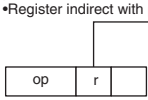
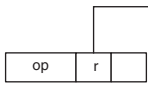
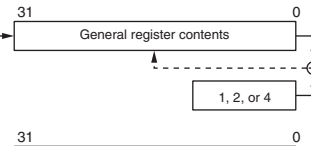
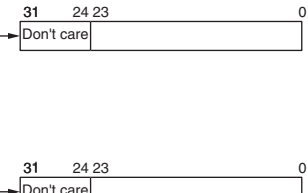
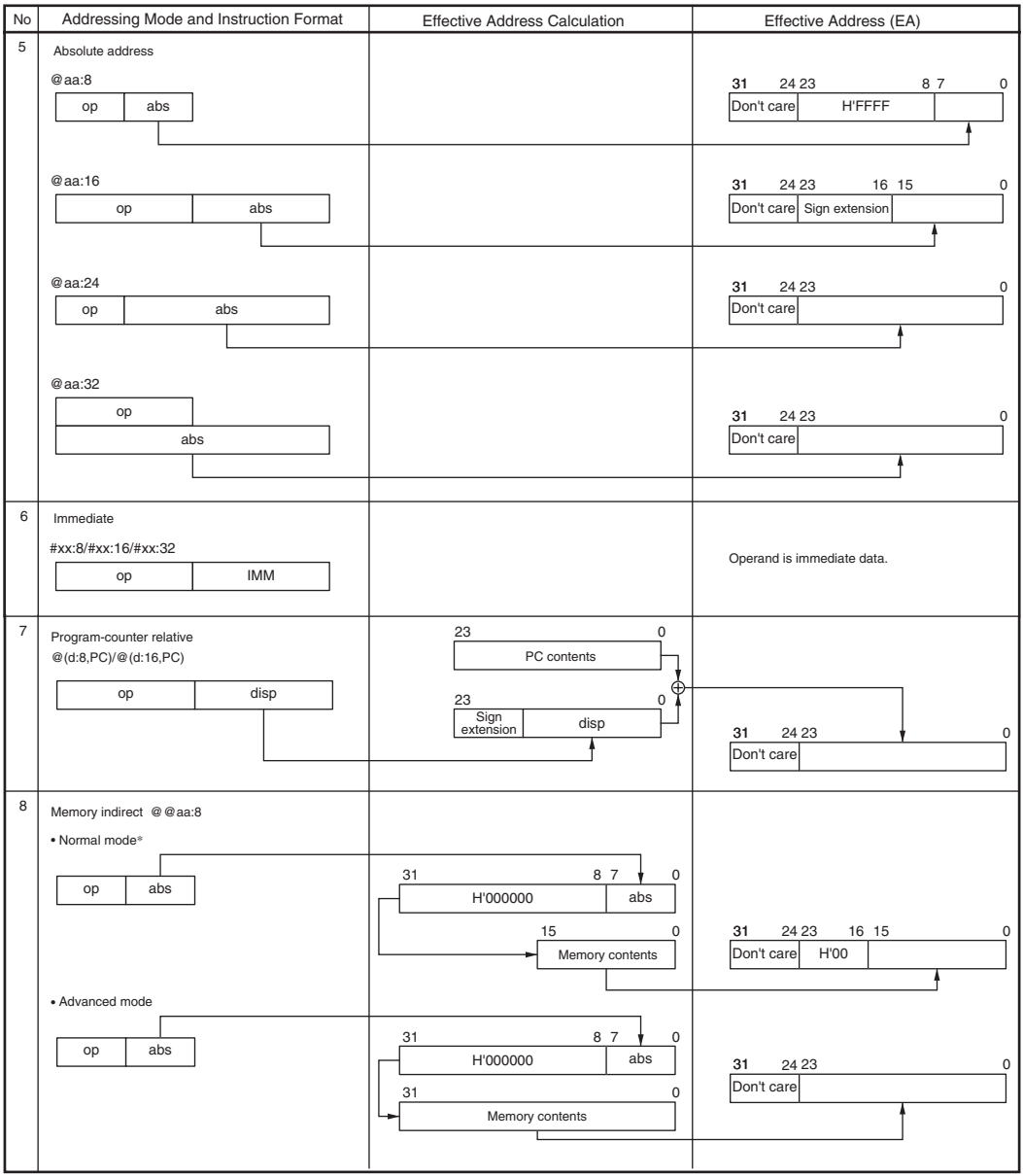
No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct(Rn) 		Operand is general register contents.
2	Register indirect(@ERn) 		
3	Register indirect with displacement @(d:16,ERn) or @(d:32,ERn) 		
4	Register indirect with post-increment or pre-decrement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



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 power-down state. Figure 2.13 indicates the state transitions.

- Reset State

  In this state, the CPU and all on-chip peripheral modules are initialized and not operating. 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.

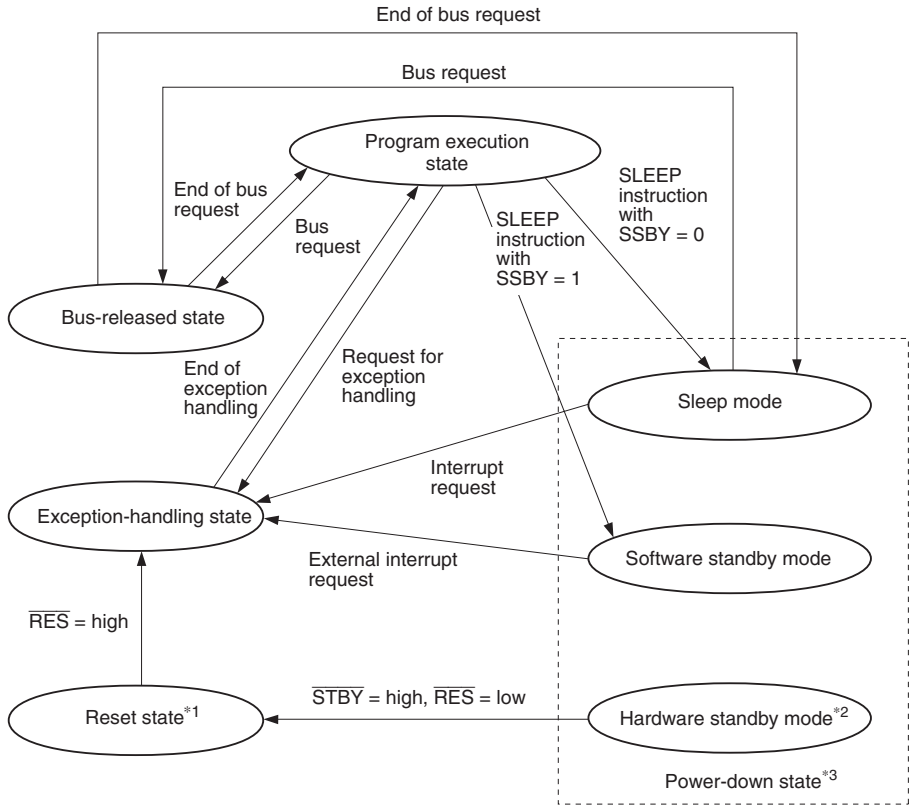
  The reset state can also be entered by a watchdog timer overflow.
- Exception-Handling State

  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 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

  The bus has been released in response to a bus request from a bus master other than the CPU. While the bus is released, the CPU halts operations.
- 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 21, Power-Down Modes.



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs whenever $\overline{RES}$ goes low. A transition can also be made to the reset state when the watchdog timer overflows.
2. From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ goes low.
3. For details, refer to section 21, Power-Down Modes.

Figure 2.13 State Transitions

2.9 Usage Note

2.9.1 Notes on Using the Bit Operation Instruction

Instructions BSET, BCLR, BNOT, BST, and BIST read data in byte units, and write data in byte units after bit operation. Therefore, attention must be paid when these instructions are used for ports or registers including write-only bits.

Instruction BCLR can be used to clear the flag in the internal I/O register to 0. If it is obvious that the flag has been set to 1 by the interrupt processing routine, it is unnecessary to read the flag beforehand.

Section 3 MCU Operating Modes

3.1 Operating Mode Selection

This LSI supports four operating modes (modes 4 to 7). These modes are determined by the mode pin (MD2 to MD0) setting. Do not change the mode pin settings during operation.

Table 3.1 MCU Operating Mode Selection

MCU Operating Mode	MD2	MD1	MD0	CPU Operating Mode	Description	On-Chip ROM
4	1	0	0	Advanced mode	Expanded mode with on-chip ROM disabled	Disabled
5	1	0	1	Advanced mode	Expanded mode with on-chip ROM disabled	Disabled
6	1	1	0	Advanced mode	Expanded mode with on-chip ROM enabled	Enabled
7	1	1	1	Advanced mode	Single-chip mode	Enabled

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)

Bit	Bit Name	Initial Value	R/W	Descriptions
7	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
6 to 3	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
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. These latches are canceled by a power-on reset.
1	MDS1	—*	R	
0	MDS0	—*	R	

Note: * Determined by pins MD2 to MD0.

3.2.2 System Control Register (SYSCR)

SYSCR selects saturating or non-saturating calculation for the MAC instruction, selects the interrupt control mode and the detected edge for NMI, and enables or disables on-chip RAM.

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 the MAC instruction 1: Saturating calculation for the MAC instruction
6	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
5	INTM1	0	R/W	These bits select the control mode of the interrupt controller. For details of the interrupt control modes, see section 5.6, Interrupt Control Modes and Interrupt Operation. 00: Interrupt control mode 0 01: Setting prohibited 10: Interrupt control mode 2 11: Setting prohibited
4	INTM0	0	R/W	
3	NMIEG	0	R/W	NMI Edge Select Selects the valid edge of the NMI interrupt input. 0: An interrupt is requested at the falling edge of NMI input 1: An interrupt is requested at the rising edge of NMI input
2, 1	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
0	RAME	1	R/W	RAM Enable Enables or disables 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 4

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled. Ports 1, A, B, and C function as an address bus, ports D and E function as a data bus, and part of port F carries bus control signals.

The bus mode immediately after a reset is 16 bits, with 16-bit access to all areas. However, note that if 8-bit access is designated by the bus controller for all areas, the bus mode switches to 8 bits.

3.3.2 Mode 5

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled. Ports1, A, B, and C function as an address bus, port D functions as a data bus, and part of port F carries bus control signals.

The bus mode immediately after a reset is 8 bits, with 8-bit access to all areas. However, note that if 16-bit access is designated by the bus controller for any area, the bus mode switches to 16 bits and port E becomes a data bus.

3.3.3 Mode 6

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled. Ports 1, A, B, and C function as input port pins immediately after a reset. Address output can be performed by setting the corresponding DDR (data direction register) bits to 1. Port D functions as a data bus, and part of port F carries bus control signals.

The bus mode immediately after a reset is 8 bits, with 8-bit access to all areas. However, note that if 16-bit access is designated by the bus controller for any area, the bus mode switches to 16 bits and port E becomes a data bus.

3.3.4 Mode 7

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled, but external addresses cannot be accessed. All I/O ports are available for use as input/output ports.

3.4 Pin Functions in Each Operating Mode

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

Table 3.2 Pin Functions in Each Mode

Port		Mode 4	Mode 5	Mode 6	Mode 7
Port 1	P11 to P13	P*/A	P*/A	P*/A	P
	P10	P/A*	P/A*	P*/A	P
Port A		P/A*	P/A*	P*/A	P
Port B		P/A*	P/A*	P*/A	P
Port C		A	A	P*/A	P
Port D		D	D	D	P
Port E		P/D*	P*/D	P*/D	P
Port F	PF7	P/C*	P/C*	P/C*	P*/C
	PF6 to PF4	C	C	C	P
	PF3	P/C*	P*/C	P*/C	
	PF0	P*/C	P*/C	P*/C	

[Legend]

P: I/O port

A: Address bus output

D: Data bus I/O

C: Control signals, clock I/O

*: After reset

3.5 Address Map

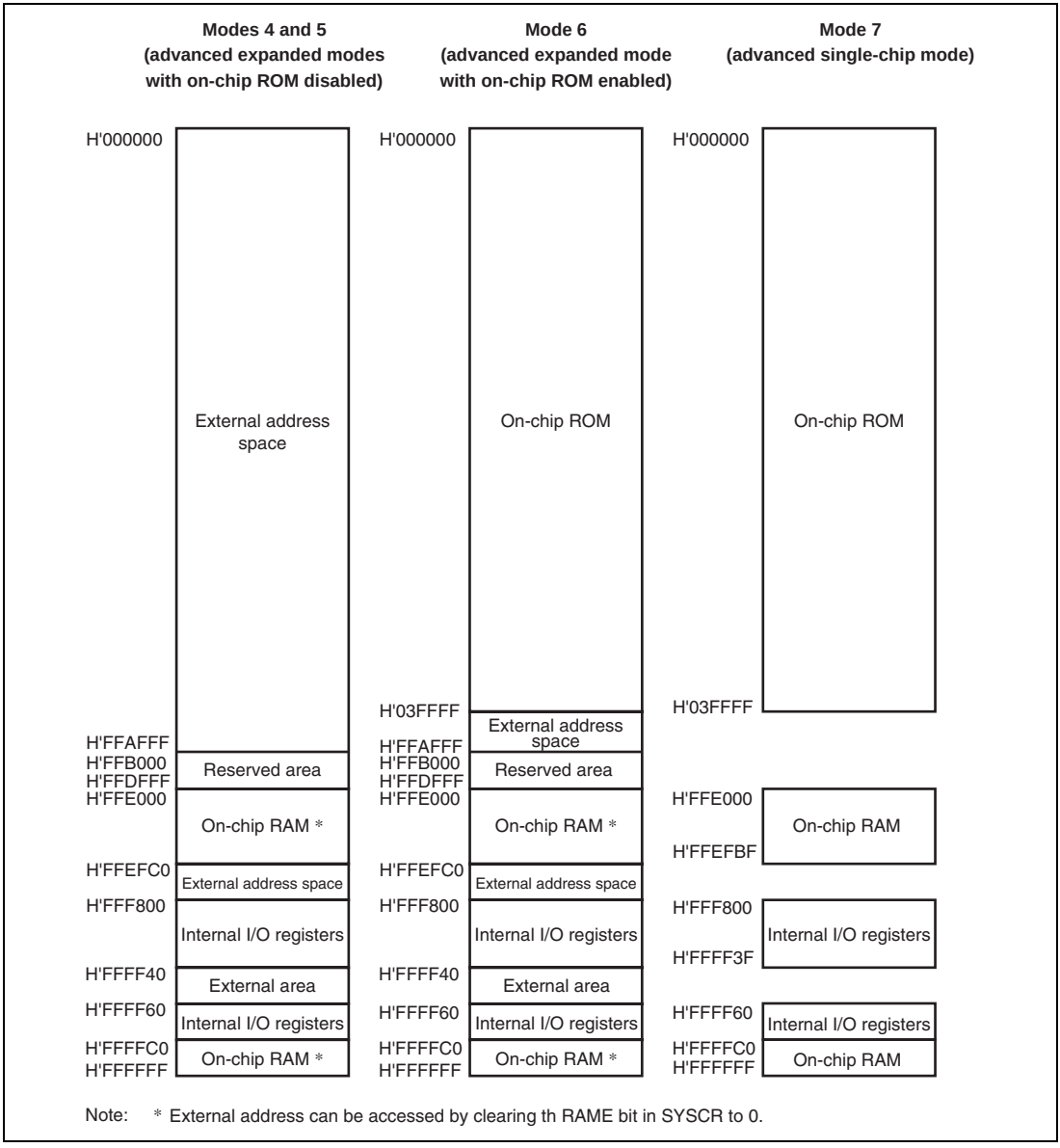
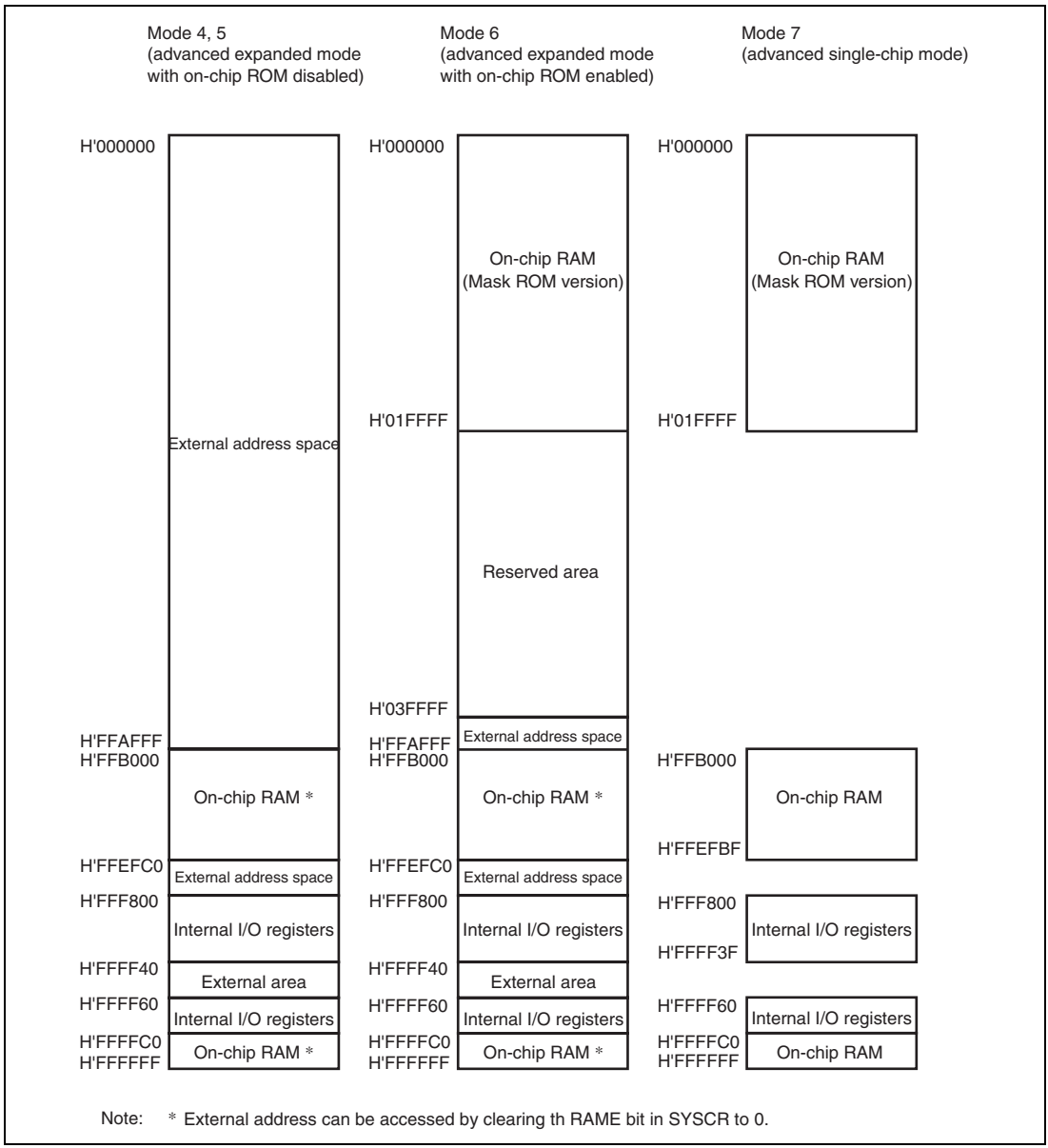


Figure 3.1 Address Map (H8S/2605)



Note: * External address can be accessed by clearing th RAME bit in SYSCR to 0.

Figure 3.2 Address Map (H8S/2606)

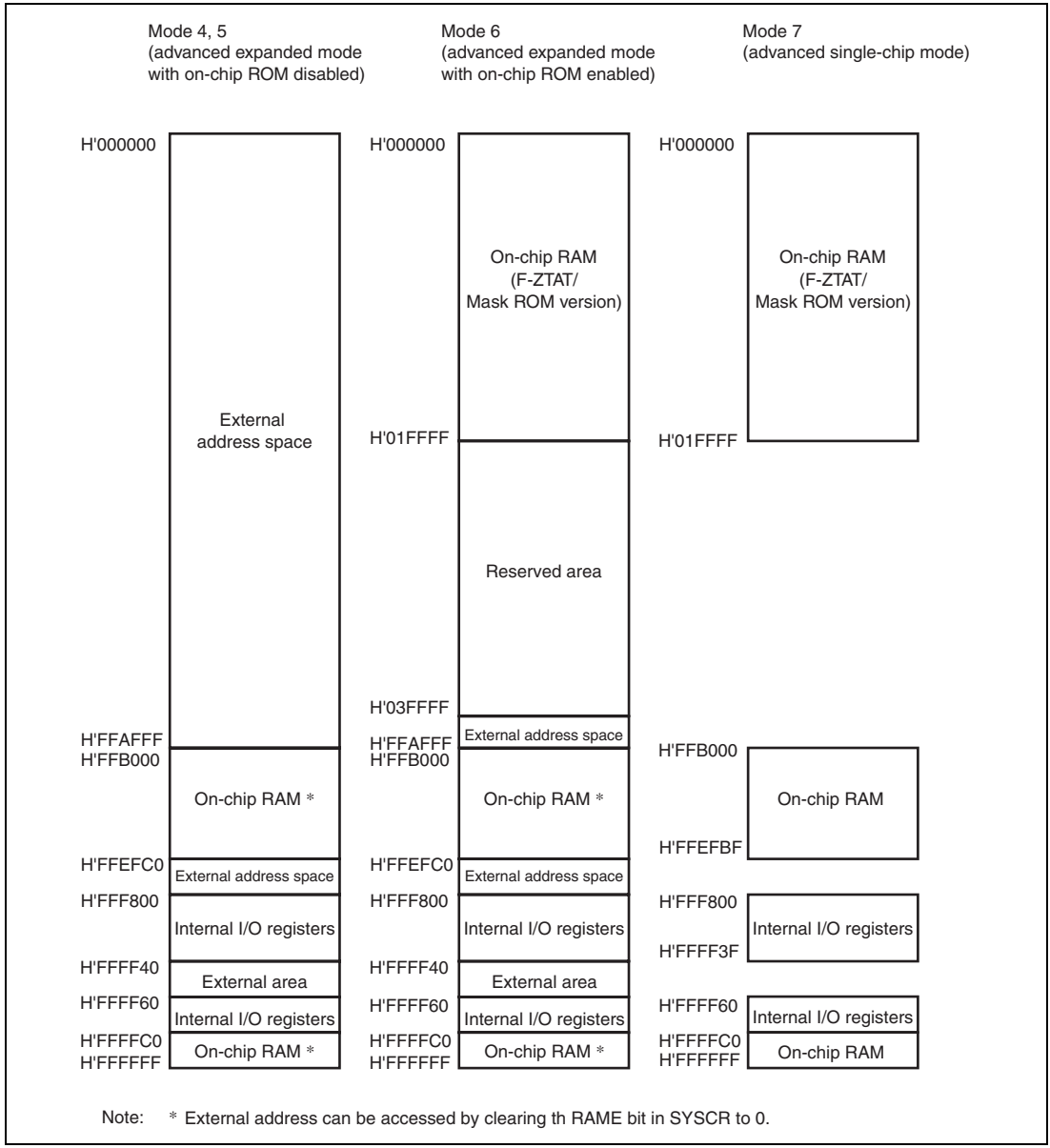


Figure 3.3 Address Map (H8S/2607)

Section 4 Exception Handling

4.1 Exception Handling Types and Priority

As shown in table 4.1, 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 EXR is set to 1.
	Direct transition	Starts when a direction transition occurs as the result of SLEEP instruction execution.
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued.*[2]
Low	Trap instruction *[3]	Started by execution of a trap instruction (TRAPA).

Notes: 1. Traces are enabled only in interrupt control mode 2. Trace exception handling is not executed after execution of an RTE instruction.
2. Interrupt detection is not performed on completion of ANDC, ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.
3. Trap instruction exception handling requests are accepted at all times in program execution state.

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 Handling Vector 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
Reserved for system use		1	H'0002 to H'0003	H'0004 to H'0007
		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'0019	H'0010 to H'0013
Trace		5	H'000A to H'000B	H'0014 to H'0017
Interrupt (direct transitions)*[4]		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
Reserved for system use		22	H'002C to H'002D	H'0058 to H'005B
		23	H'002E to H'002F	H'005C to H'005F

Exception Source	Vector Number	Vector Address*1	
		Normal Mode*2	Advanced Mode
Internal interrupt*3	24	H'0030 to H'0031	H'0060 to H'0063
	127	H'00FE to H'00FF	H'01FC to H'01FF

Notes: 1. Lower 16 bits of the address.
2. Not available in this LSI.
3. For details of internal interrupt vectors, see section 5.5, Interrupt Exception Handling Vector Table.
4. For direct transitions, see section 21.10, Direct Transitions.

4.3 Reset

A reset has the highest exception priority.

When the $\overline{\text{RES}}$ pin goes low, all processing halts and this LSI enters the reset state. To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-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 12, Watchdog Timer (WDT).

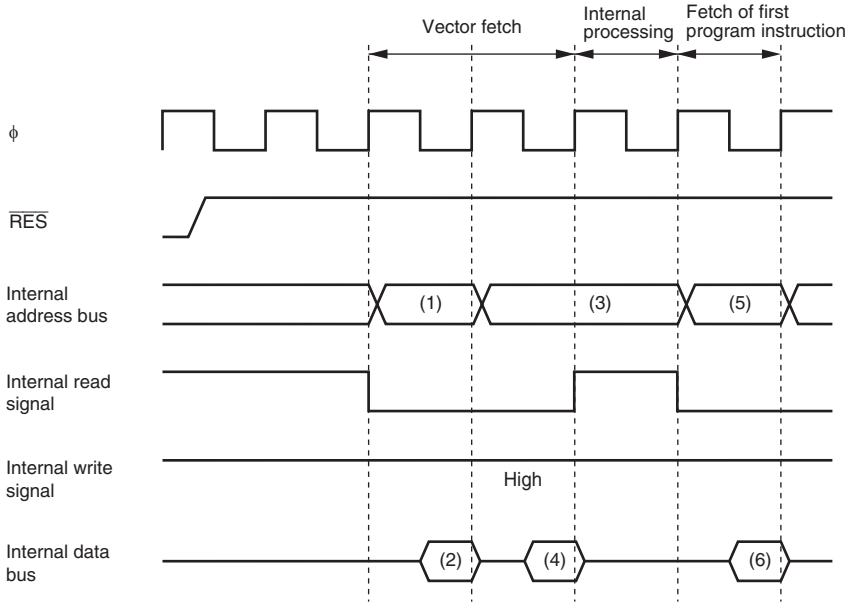
The interrupt control mode is 0 immediately after reset.

4.3.1 Reset Exception Handling

When the $\overline{\text{RES}}$ pin goes high after being held low for the necessary period, this LSI starts reset exception handling as follows:

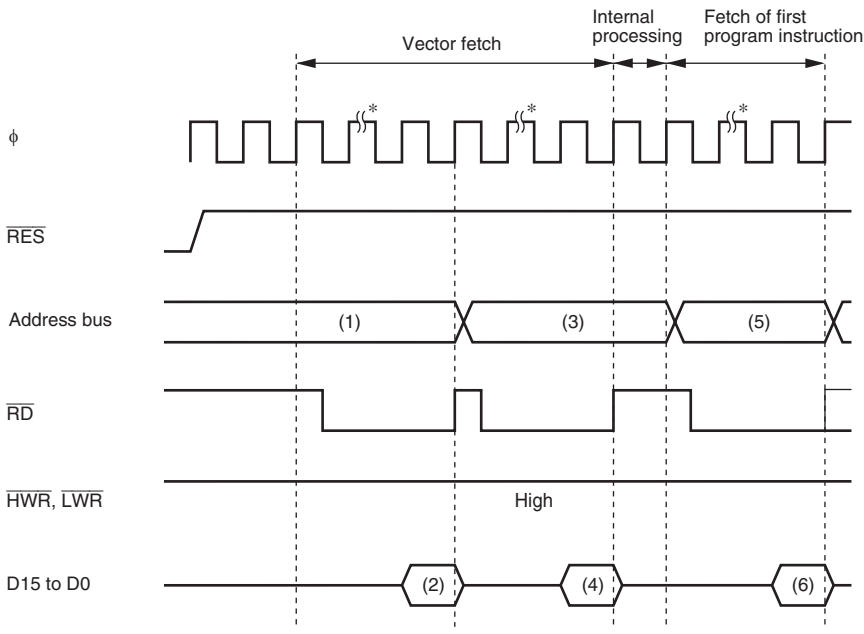
1. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized, the T bit in EXR is cleared to 0, and the I bit in EXR and CCR is set to 1.
2. The reset exception handling vector address is read and transferred to the PC, and program execution starts from the address indicated by the PC.

Figures 4.1 and 4.2 show examples 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 (Advanced Mode with On-chip ROM Enabled)



(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

Note:* Three program wait states are inserted.

Figure 4.2 Reset Sequence (Advanced Mode with On-chip ROM Disabled)

4.3.2 Interrupts after Reset

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

4.3.3 State of On-Chip Peripheral Modules after Reset Release

After reset release, MSTPCRA to MSTPCRA are initialized to H'3F, H'FF, and H'FF, and B'11xxxxxx respectively, and all modules except the DTC enter module stop mode. Consequently, on-chip peripheral module registers cannot be read or written to. Register reading and writing is enabled when the module stop mode is cancelled.

Note: The initial values of bits 5 to 0 in MSTPCRD are undefined.

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 of 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 mask bit in CCR. Table 4.3 shows the states of CCR and EXR after execution of trace exception handling. Trace mode is cancelled by clearing the T bit in EXR to 0 with the trace exception handling. 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 Statuses 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. The source to start interrupt exception handling and the vector address differ depending on the product. For details, refer to section 5, Interrupt Controller.

Interrupt exception handling is conducted as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended control register (EXR) are saved to 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 begins 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.

Trap instruction exception handling is conducted as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended control register (EXR) are saved to 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 statuses of CCR and EXR after execution of trap instruction exception handling.

Table 4.4 Statuses 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

Figures 4.3 shows the stack after completion of trap instruction exception handling and interrupt exception handling.

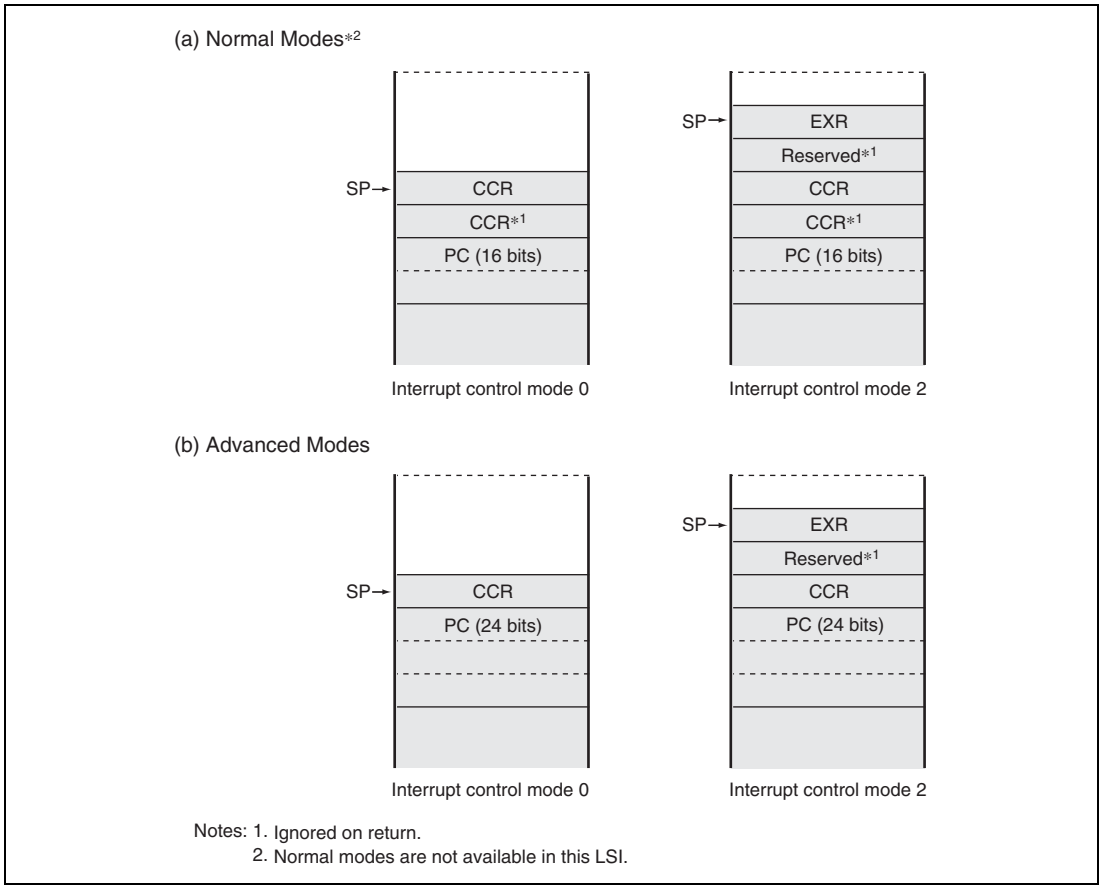


Figure 4.3 Stack Status after Exception Handling

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 transfer instruction or longword transfer instruction, 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 4.4 shows an example of what happens when the SP value is odd.

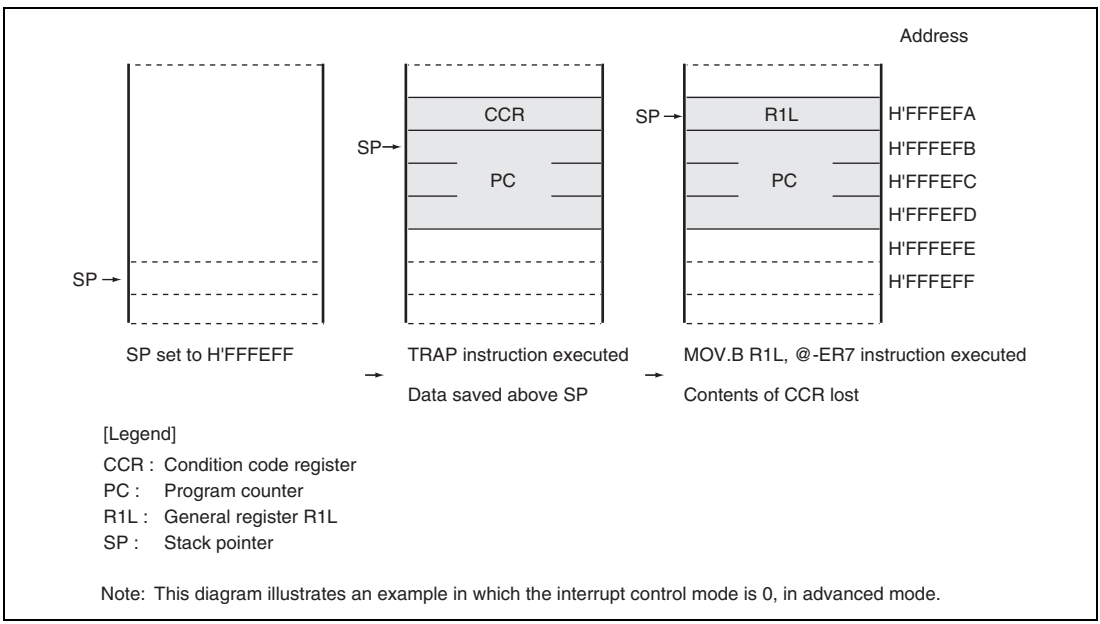


Figure 4.4 Operation when SP Value Is Odd

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 system control register (SYSCR).
- 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.
- Seven 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 $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$.
- DTC control
  - The DTC can be activated by an interrupt request.

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

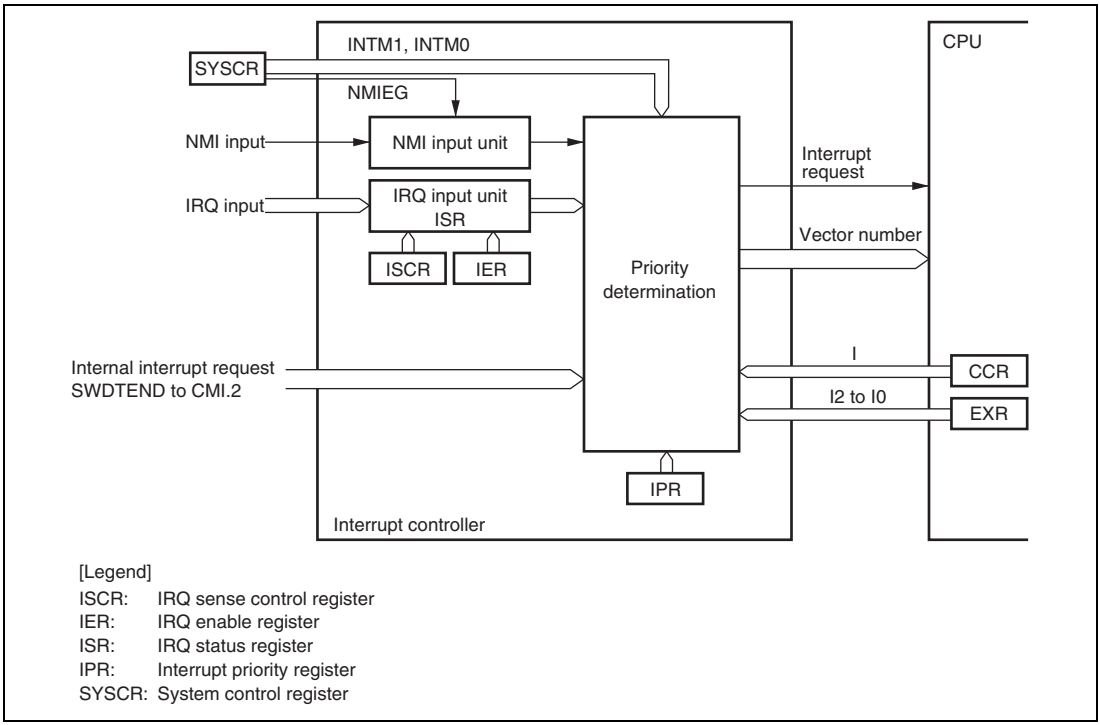


Figure 5.1 Block Diagram of Interrupt Controller

5.2 Input/Output Pins

Table 5.1 summarizes the pins 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
$\overline{\text{IRQ5}}$	Input	Maskable external interrupts
$\overline{\text{IRQ4}}$	Input	Rising, falling, or both edges, or level sensing, can be selected
$\overline{\text{IRQ3}}$	Input	
$\overline{\text{IRQ2}}$	Input	
$\overline{\text{IRQ1}}$	Input	
$\overline{\text{IRQ0}}$	Input	

5.3 Register Descriptions

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

- System control register (SYSCR)
- IRQ sense control register H (ISCRH)
- IRQ sense control register L (ISCRL)
- IRQ enable register (IER)
- IRQ status register (ISR)
- Interrupt priority register A (IPRA)
- Interrupt priority register B (IPRB)
- Interrupt priority register C (IPRC)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)
- Interrupt priority register H (IPRH)
- Interrupt priority register J (IPRJ)
- Interrupt priority register K (IPRK)
- Interrupt priority register L (IPRL)
- Interrupt priority register M (IPRM)

5.3.1 Interrupt Priority Registers A to H, J, K, L, and M (IPRA to IPRH, IPRJ, IPRK, IPRL, and IPRM)

The IPR registers 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 0 to 2 and 4 to 6 sets the priority of the corresponding interrupt.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved These bits are always read as 0.
6	IPR6	1	R/W	These bits set 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	IPR5	1	R/W	
4	IPR4	1	R/W	
3	—	0	—	Reserved These bits are always read as 0.
2	IPR2	1	R/W	These bits set 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)
1	IPR1	1	R/W	
0	IPR0	1	R/W	

5.3.2 IRQ Enable Register (IER)

IER is an 8-bit readable/writable register that controls the enabling and disabling of interrupt requests IRQ0 to IRQ5.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved The write value should always be 0.
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.

5.3.3 IRQ Sense Control Registers H and L (ISCRH, ISCRL)

The ISCR registers select the source that generates an interrupt request at pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$.

- ISCRH

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R/W	Reserved The write value should always be 0.
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 level low 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
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 level low 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

- ISCRL

Bit	Bit Name	Initial Value	R/W	Description
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 level low 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 level low 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
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 level low 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 level low 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

5.3.4 IRQ Status Register (ISR)

ISR indicates the status of IRQ0 to IRQ5 interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W*	Reserved These bits are always read as 0.
5	IRQ5F	0	R/W*	[Setting condition] When the interrupt source selected by the ISCR registers occurs [Clearing conditions] • Cleared by reading IRQnF flag when IRQnF = 1, then writing 0 to IRQnF flag • 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 when falling, rising, or both-edge detection is set • When the DTC is activated by an IRQn interrupt, and the DISEL bit in MRB of the DTC is cleared to 0
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.

5.4 Interrupt Sources

5.4.1 External Interrupts

There are seven external interrupts: NMI and IRQ0 to IRQ5. These interrupts can be used to restore this LSI from software standby mode.

NMI Interrupt: 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 SYSCR can be used to select whether an interrupt is requested at a rising edge or a falling edge on the NMI pin.

IRQ0 to IRQ5 Interrupts: Interrupts IRQ0 to IRQ5 are requested by an input signal at pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$. Interrupts IRQ0 to IRQ5 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 $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$.
- Enabling or disabling of interrupt requests IRQ0 to IRQ5 can be selected with IER.
- The interrupt priority level can be set with IPR.
- The status of interrupt requests IRQ0 to IRQ5 is indicated in ISR. ISR flags can be cleared to 0 by software.

The detection of IRQ0 to IRQ5 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 IRQ0 to IRQ5 is shown in figure 5.2.

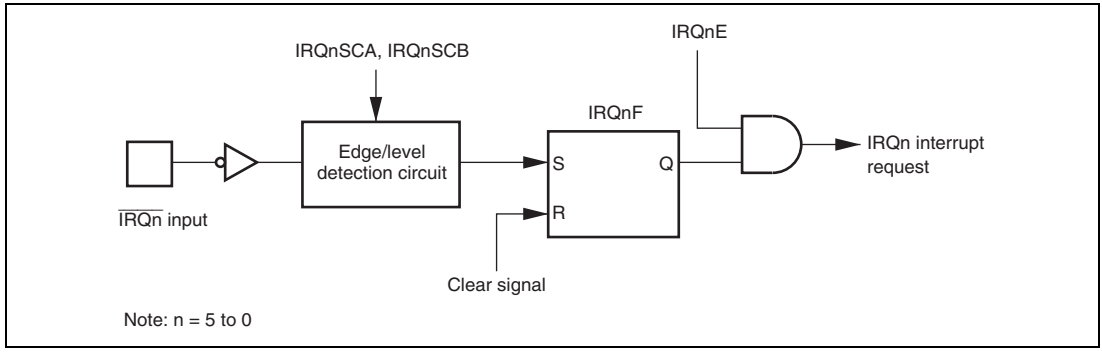


Figure 5.2 Block Diagram of Interrupts IRQ0 to IRQ5

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. If both of these are set to 1 for a particular interrupt source, an interrupt request is issued to the interrupt controller.
- The interrupt priority level can be set by means of IPR.
- The DTC can be activated by a TPU, SCI, or other interrupt request.
- When the DTC is activated by an interrupt request, it is not affected by the interrupt control mode or CPU interrupt mask bit.

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. Priorities among modules can be set by means of 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* Advanced Mode	IPR	Priority
External pin	NMI	7	H'001C	—	High
	IRQ0	16	H'0040	IPRA6 to IPRA4	
	IRQ1	17	H'0044	IPRA2 to IPRA0	
	IRQ2	18	H'0048	IPRB6 to IPRB4	
	IRQ3	19	H'004C		
	IRQ4	20	H'0050	IPRB2 to IPRB0	
	IRQ5	21	H'0054		
—	Reserved for system use	22	H'0058	—	
	Reserved for system use	23	H'005C	—	
DTC	SWDTEND	24	H'0060	IPRC2 to IPRC0	
WDT_0	WOVI0	25	H'0064	IPRD6 to IPRD4	
PC break control	PC break	27	H'006C	IPRE6 to IPRE4	
A/D	ADI	28	H'0070	IPRE2 to IPRE0	
WDT_1	WOVI1	29	H'0074		
TPU_0	TGIA_0	32	H'0080	IPRF6 to IPRF4	
	TGIB_0	33	H'0084		
	TGIC_0	34	H'0088		
	TGID_0	35	H'008C		
	TCIV_0	36	H'0090		
TPU_1	TGIA_1	40	H'00A0	IPRF2 to IPRF0	
	TGIB_1	41	H'00A4		
	TCIV_1	42	H'00A8		
	TCIU_1	43	H'00AC		
TPU_2	TGIA_2	44	H'00B0	IPRG6 to IPRG4	
	TGIB_2	45	H'00B4		
	TCIV_2	46	H'00B8		
	TCIU_2	47	H'00BC		Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority
TPU_3	TGIA_3	48	H'00C0	IPRG2 to IPRG0	High
	TGIB_3	49	H'00C4		
	TGIC_3	50	H'00C8		
	TGID_3	51	H'00CC		
	TCIV_3	52	H'00D0		
TPU_4	TGIA_4	56	H'00E0	IPRH6 to IPRH4	
	TGIB_4	57	H'00E4		
	TCIV_4	58	H'00E8		
	TCIU_4	59	H'00EC		
TPU_5	TGIA_5	60	H'00F0	IPRH2 to IPRH0	
	TGIB_5	61	H'00F4		
	TCIV_5	62	H'00F8		
	TCIU_5	63	H'00FC		
SCI_0	ERI_0	80	H'0140	IPRJ2 to IPRJ0	
	RXI_0	81	H'0144		
	TXI_0	82	H'0148		
	TEI_0	83	H'014C		
SCI_1	ERI_1	84	H'0150	IPRK6 to IPRK4	
	RXI_1	85	H'0154		
	TXI_1	86	H'0158		
	TEI_1	87	H'015C		
SCI_2	ERI_2	88	H'0160	IPRK2 to IPRK0	
	RXI_2	89	H'0164		
	TXI_2	90	H'0168		
	TEI_2	91	H'016C		
IIC-0	IIICI0	100	H'0190	IPRL2 to IPRLO	
	DDCSW1	101	H'0194		
IIC-1	IICI1	102	H'0198		
	Reserved for system use	103	H'019C		Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority
Motor control PWM timer	CMI_1	104	H'01A0	IPRM6 to IPRM4	High
	CMI_2	105	H'01A4		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 SYSCR. 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, are masked by the I bit.
2	IPR	I2 to I0	8 priority levels other than 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, interrupt requests other than for NMI are masked by the I bit in CCR in the CPU. 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 in CCR 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. When interrupt requests are sent to the interrupt controller, the interrupt with the highest priority according to the interrupt priority levels is selected 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.

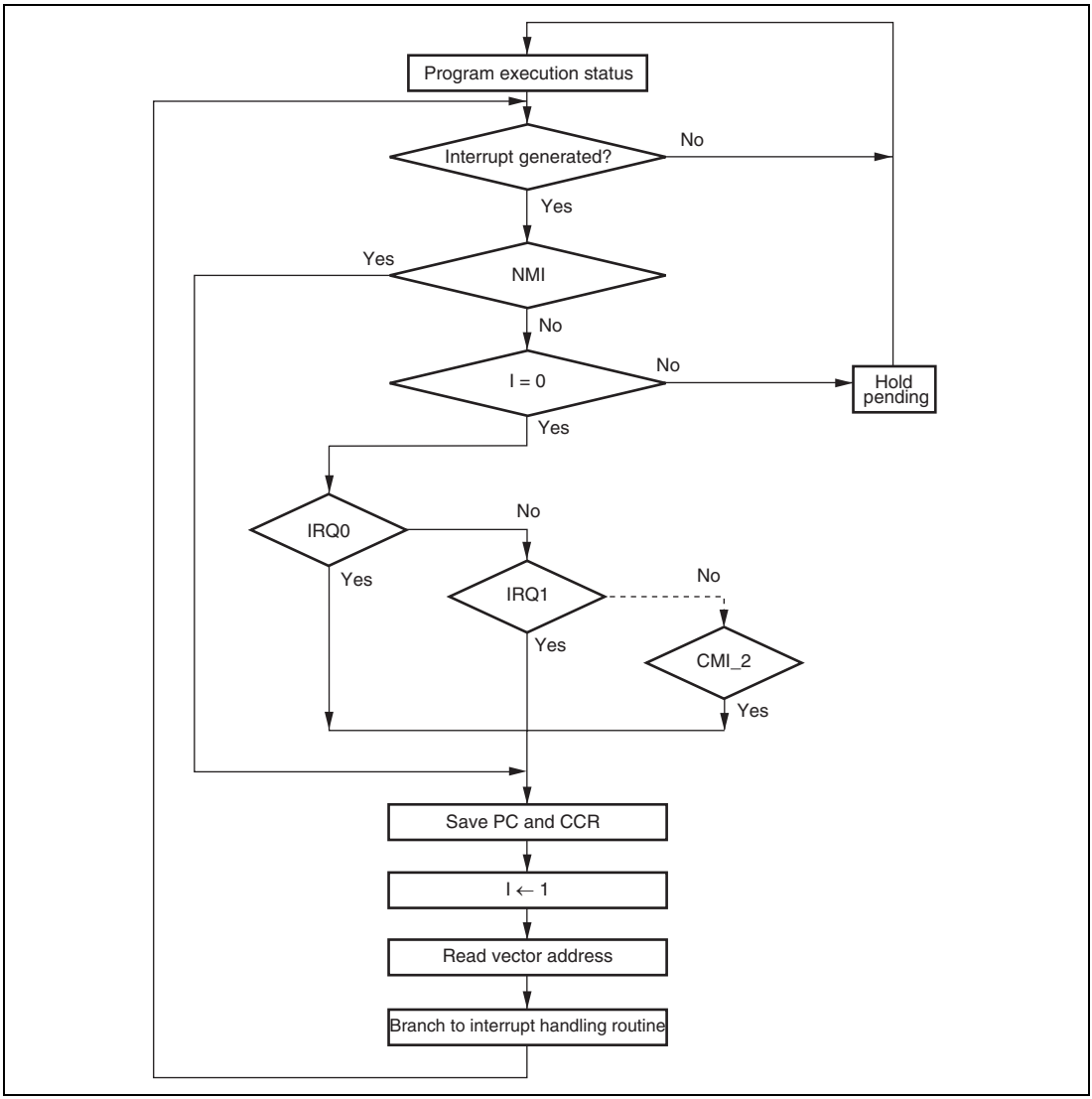


Figure 5.3 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 0

5.6.2 Interrupt Control Mode 2

In interrupt control mode 2, mask control is applied to eight levels for interrupt requests other than 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.

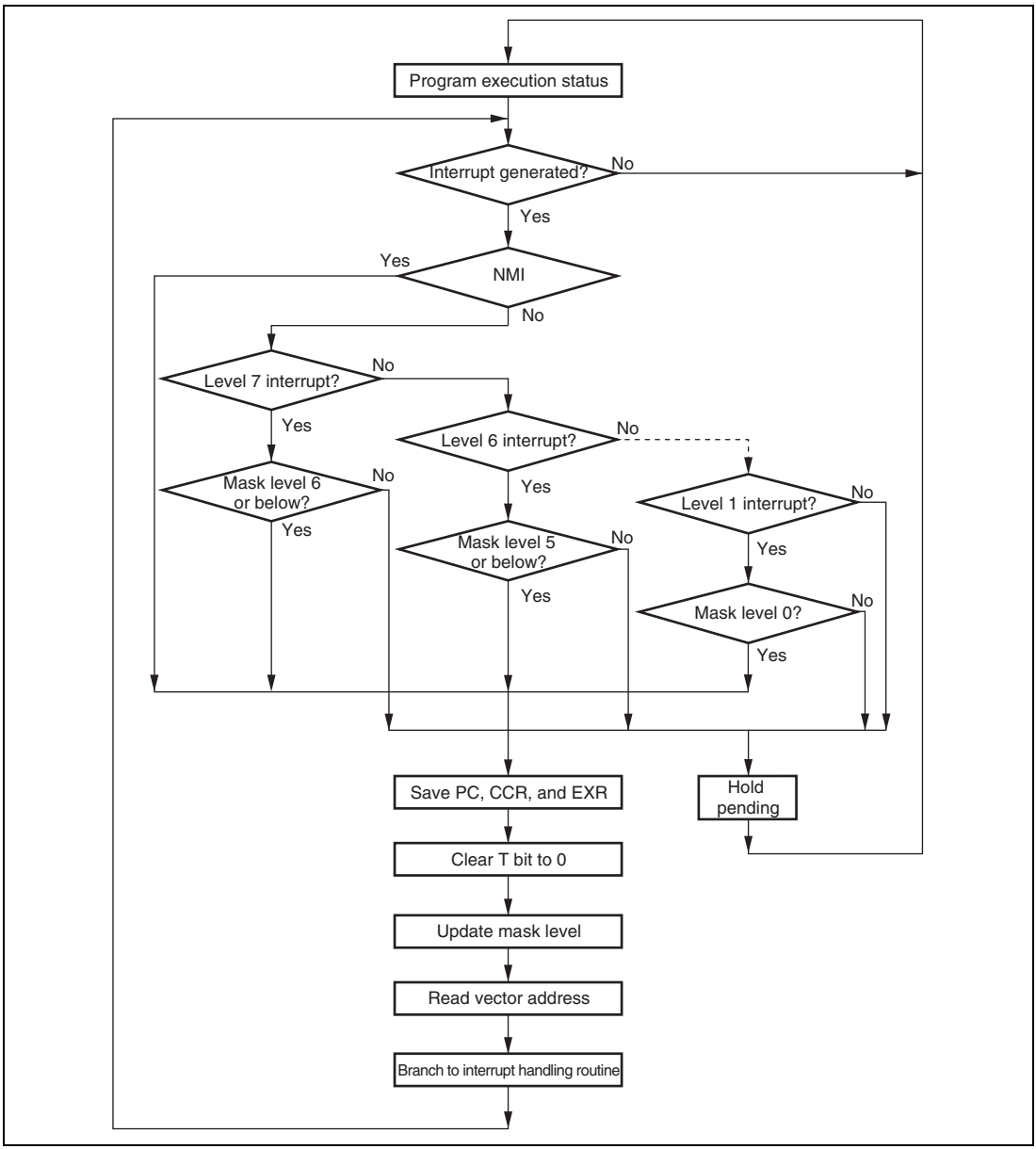


Figure 5.4 Flowchart of Procedure Up to Interrupt Acceptance in Control Mode 2

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.

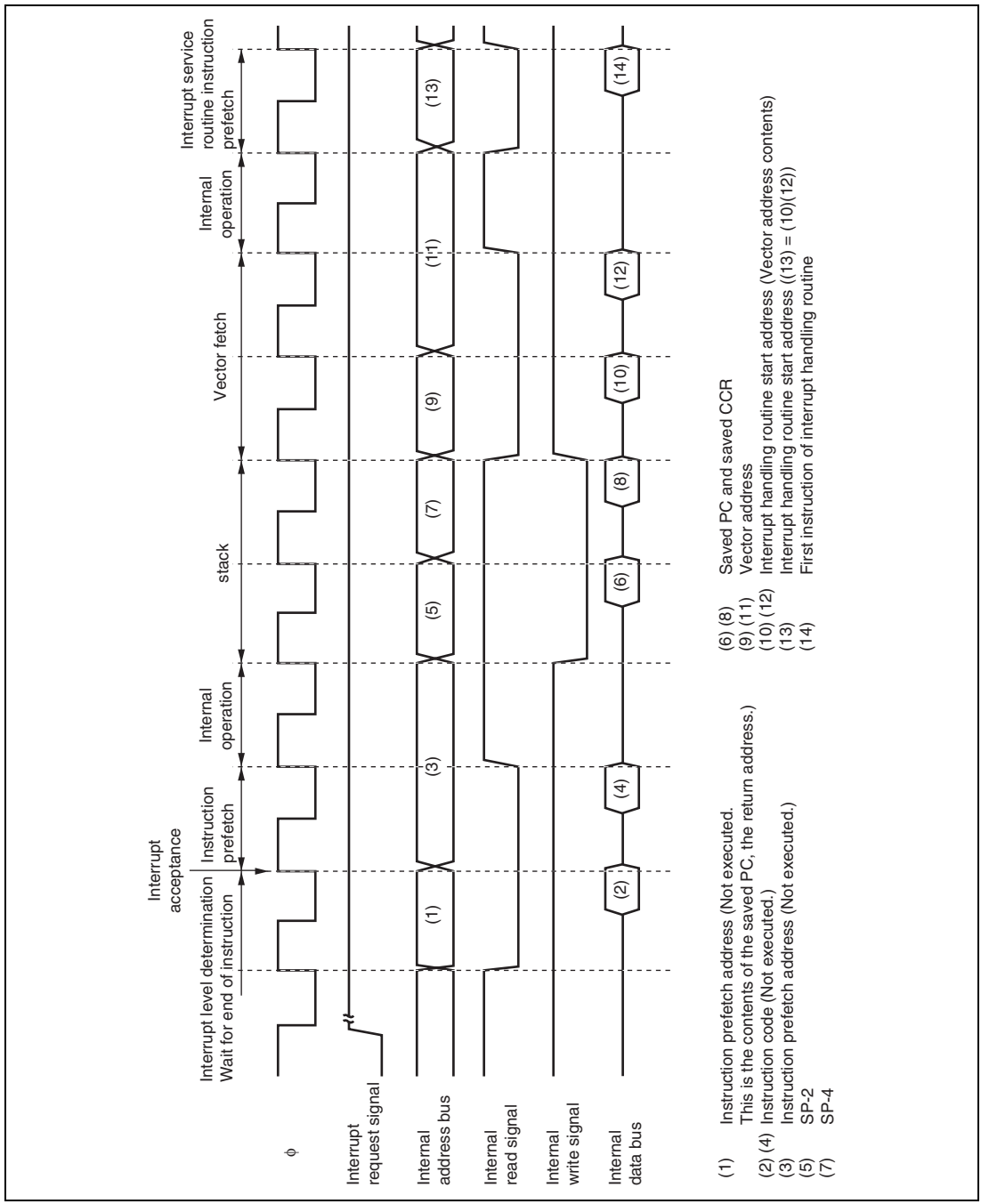


Figure 5.5 Interrupt Exception Handling

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, has 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 MULXS and 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 Status

		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.

5.6.5 DTC Activation by Interrupt

The DTC can be activated by an interrupt. For details, see section 8, Data Transfer Controller (DTC).

5.7 Usage Notes

5.7.1 Conflict between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to disable interrupts, the disabling becomes effective after execution of the instruction.

When an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, and if an interrupt is generated during execution of the instruction, the interrupt concerned will still be enabled on completion of the instruction, and so interrupt exception handling for that interrupt will be executed on completion of the instruction. However, if there is an interrupt request 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 TIER_0 of the TPU is cleared to 0.

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

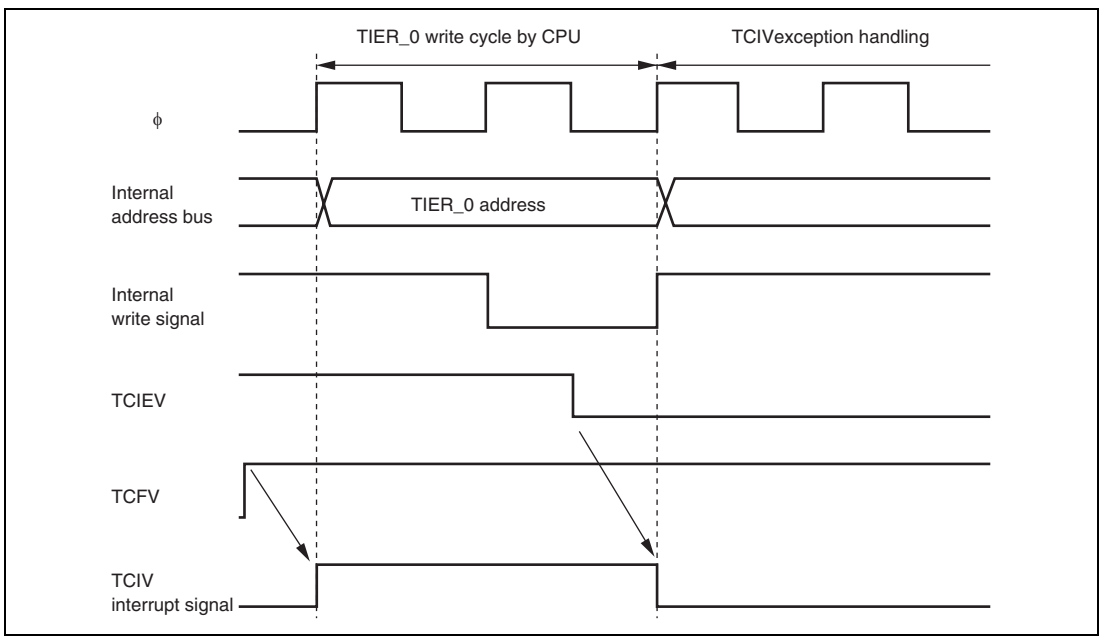


Figure 5.6 Conflict between Interrupt Generation and Disabling

5.7.2 Instructions that Disable Interrupts

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

5.7.3 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
```

Section 6 PC Break Controller (PBC)

The PC break controller (PBC) provides functions that simplify program debugging. Using these functions, it is easy to create a self-monitoring debugger, enabling programs to be debugged with the chip alone, without using an in-circuit emulator. A block diagram of the PC break controller is shown in figure 6.1.

6.1 Features

- Two break channels (A and B)
- 24-bit break address
  - Bit masking possible
- Four types of break compare conditions
  - Instruction fetch
  - data read
  - data write
  - data read/write
- Bus master
  - Either CPU or CPU/DTC can be selected
- The timing of PC break exception handling after the occurrence of a break condition is as follows:
  - Immediately before execution of the instruction fetched at the set address (instruction fetch)
  - Immediately after execution of the instruction that accesses data at the set address (data access)
- Module stop mode can be set

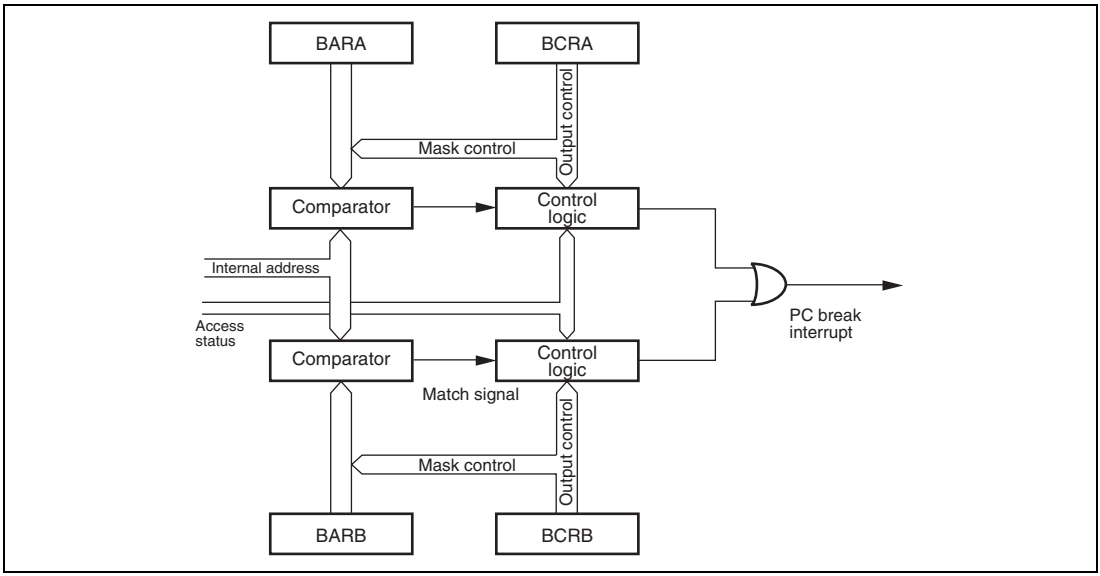


Figure 6.1 Block Diagram of PC Break Controller

6.2 Register Descriptions

The PC break controller has the following registers.

- Break address register A (BARA)
- Break address register B (BARB)
- Break control register A (BCRA)
- Break control register B (BCRB)

6.2.1 Break Address Register A (BARA)

BARA is a 32-bit readable/writable register that specifies the channel A break address.

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	Undefined	—	Reserved These bits are read as an undefined value and cannot be modified.
23 to 0	BAA23 to BAA0	H'000000	R/W	These bits set the channel A PC break address.

6.2.2 Break Address Register B (BARB)

BARB is the channel B break address register. The bit configuration is the same as for BARA.

6.2.3 Break Control Register A (BCRA)

BCRA controls channel A PC breaks. BCRA also contains a condition match flag.

Bit	Bit Name	Initial Value	R/W	Description
7	CMFA	0	R/W	Condition Match Flag A [Setting condition] When a condition set for channel A is satisfied [Clearing condition] When 0 is written to CMFA after reading CMFA = 1
6	CDA	0	R/W	CPU Cycle/DTC Cycle Select A Selects the channel A break condition bus master. 0: CPU 1: CPU or DTC
5	BAMRA2	0	R/W	Break Address Mask Register A2 to A0 These bits specify which bits of the break address set in BARA are to be masked. 000: BAA23 to BAA0 (All bits are unmasked) 001: BAA23 to BAA1 (Lowest bit is masked) 010: BAA23 to BAA2 (Lower 2 bits are masked) 011: BAA23 to BAA3 (Lower 3 bits are masked) 100: BAA23 to BAA4 (Lower 4 bits are masked) 101: BAA23 to BAA8 (Lower 8 bits are masked) 110: BAA23 to BAA12 (Lower 12 bits are masked) 111: BAA23 to BAA16 (Lower 16 bits are masked)
4	BAMRA1	0	R/W	
3	BAMRA0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
2	CSELA1	0	R/W	Break Condition Select A
1	CSELA0	0	R/W	Selects break condition of channel A. 00: Instruction fetch is used as break condition 01: Data read cycle is used as break condition 10: Data write cycle is used as break condition 11: Data read/write cycle is used as break condition
0	BIEA	0	R/W	Break Interrupt Enable A When this bit is 1, the PC break interrupt request of channel A is enabled.

6.2.4 Break Control Register B (BCRB)

BCRB is the channel B break control register. The bit configuration is the same as for BCRA.

6.3 Operation

The operation flow from break condition setting to PC break interrupt exception handling is shown in section 6.3.1, PC Break Interrupt Due to Instruction Fetch, and 6.3.2, PC Break Interrupt Due to Data Access, taking the example of channel A.

6.3.1 PC Break Interrupt Due to Instruction Fetch

1. Set the break address in BARA.
   For a PC break caused by an instruction fetch, set the address of the first instruction byte as the break address.
2. Set the break conditions in BCR.
   Set bit 6 (CDA) to 0 to select the CPU because the bus master must be the CPU for a PC break caused by an instruction fetch. Set the address bits to be masked to bits 3 to 5 (BAMA0 to BAMA2). Set bits 1 and 2 (CSELA0 and CSELA1) to 00 to specify an instruction fetch as the break condition. Set bit 0 (BIEA) to 1 to enable break interrupts.
3. When the instruction at the set address is fetched, a PC break request is generated immediately before execution of the fetched instruction, and the condition match flag (CMFA) is set.
4. After priority determination by the interrupt controller, PC break interrupt exception handling is started.

6.3.2 PC Break Interrupt Due to Data Access

1. Set the break address in BARA.

   For a PC break caused by a data access, set the target ROM, RAM, I/O, or external address space address as the break address. Stack operations and branch address reads are included in data accesses.
2. Set the break conditions in BCRA.

   Select the bus master with bit 6 (CDA). Set the address bits to be masked to bits 3 to 5 (BAMA BAMA0 to BAMA2). Set bits 1 and 2 (CSELA0 and CSELA1) to 01, 10, or 11 to specify data access as the break condition. Set bit 0 (BIEA) to 1 to enable break interrupts.
3. After execution of the instruction that performs a data access on the set address, a PC break request is generated and the condition match flag (CMFA) is set.
4. After priority determination by the interrupt controller, PC break interrupt exception handling is started.

6.3.3 PC Break Operation at Consecutive Data Transfer

- When a PC break interrupt is generated at the transfer address of an EEPMOV.B instruction

  PC break exception handling is executed after all data transfers have been completed and the EEPMOV.B instruction has ended.
- When a PC break interrupt is generated at a DTC transfer address

  PC break exception handling is executed after the DTC has completed the specified number of data transfers, or after data for which the DISEL bit is set to 1 has been transferred.

6.3.4 Operation in Transitions to Power-Down Modes

The operation when a PC break interrupt is set for an instruction fetch at the address after a SLEEP instruction is shown below.

1. When the SLEEP instruction causes a transition from high-speed (medium-speed) mode to sleep mode, or from subactive mode to subsleep mode:

   After execution of the SLEEP instruction, a transition is not made to sleep mode or subsleep mode, and PC break interrupt handling is executed. After execution of PC break interrupt handling, the instruction at the address after the SLEEP instruction is executed (figure 6-2 (A)).
2. When the SLEEP instruction causes a transition from high-speed (medium-speed) mode to subactive mode:

After execution of the SLEEP instruction, a transition is made to subactive mode via direct transition exception handling. After the transition, PC break interrupt handling is executed, then the instruction at the address after the SLEEP instruction is executed (figure 6-2 (B)).

3. When the SLEEP instruction causes a transition from subactive mode to high-speed (medium-speed) mode:

   After execution of the SLEEP instruction, and following the clock oscillation settling time, a transition is made to high-speed (medium-speed) mode via direct transition exception handling. After the transition, PC break interrupt handling is executed, then the instruction at the address after the SLEEP instruction is executed (figure 6.2 (C)).

4. When the SLEEP instruction causes a transition to software standby mode or watch mode:

   After execution of the SLEEP instruction, a transition is made to the respective mode, and PC break interrupt handling is not executed. However, the CMFA or CMFB flag is set (figure 6.2 (D)).

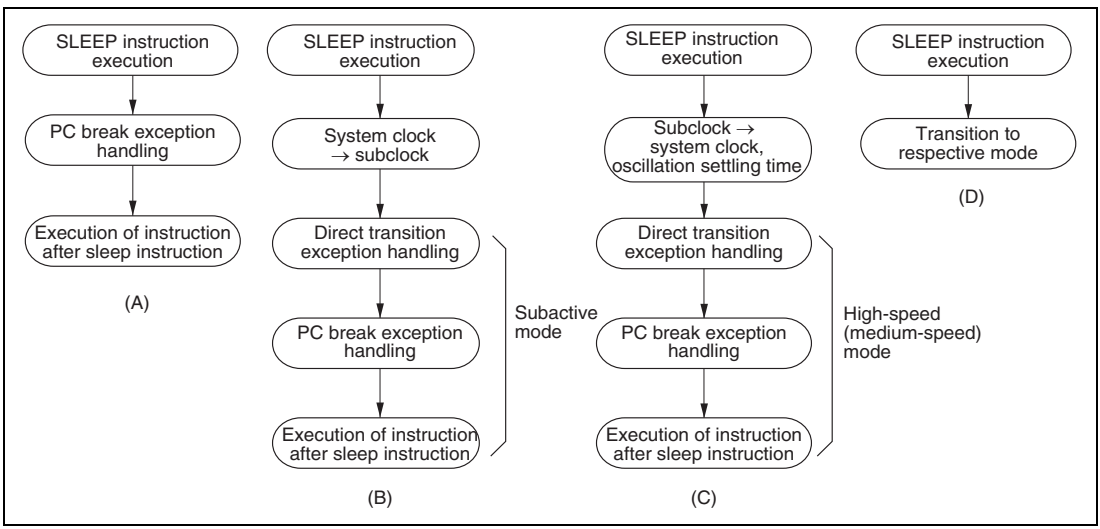


Figure 6.2 Operations in Power-Down Mode Transitions

6.3.5 When Instruction Execution Is Delayed by One State

While the break interrupt enable bit is set to 1, instruction execution is one state later than usual.

- For 1-word branch instructions (Bcc d:8, BSR, JSR, JMP, TRAPA, RTE, and RTS) in on-chip ROM or RAM.
- When break interrupt by instruction fetch is set, the set address indicates on-chip ROM or RAM space, and that address is used for data access, the instruction will be one state later than in normal operation.
- When break interrupt by instruction fetch is set and a break interrupt is generated, if the executing instruction immediately preceding the set instruction has one of the addressing modes shown below, and that address indicates on-chip ROM or RAM, the instruction will be one state later than in normal operation.

  Addressing modes: @ERn, @(d:16,ERn), @(d:32,ERn), @-ERn/ERn+, @aa:8, @aa:24, @aa:32, @(d:8,PC), @(d:16,PC), @@aa:8
- When break interrupt by instruction fetch is set and a break interrupt is generated, if the executing instruction immediately preceding the set instruction is NOP or SLEEP, or has #xx,Rn as its addressing mode, and that instruction is located in on-chip ROM or RAM, the instruction will be one state later than in normal operation.

6.4 Usage Notes

6.4.1 Module Stop Mode Setting

PBC operation can be disabled or enabled using the module stop control register. The initial setting is for PBC operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 21, Power-Down Modes.

6.4.2 PC Break Interrupts

The PC break interrupt is shared by channels A and B. The channel from which the request was issued must be determined by the interrupt handler.

6.4.3 CMFA and CMFB

The CMFA and CMFB flags are not automatically cleared to 0, so 0 must be written to CMFA or CMFB after first reading the flag while it is set to 1. If the flag is left set to 1, another interrupt will be requested after interrupt handling ends.

6.4.4 PC Break Interrupt when DTC Is Bus Master

A PC break interrupt generated when the DTC is the bus master is accepted after the bus mastership has been transferred to the CPU by the bus controller.

6.4.5 PC Break Set for Instruction Fetch at Address Following BSR, JSR, JMP, TRAPA, RTE, or RTS Instruction

Even if the instruction at the address following a BSR, JSR, JMP, TRAPA, RTE, or RTS instruction is fetched, it is not executed, and so a PC break interrupt is not generated by the instruction fetch at the next address.

6.4.6 I Bit Set by LDC, ANDC, ORC, or XORC Instruction

When the I bit is set by an LDC, ANDC, ORC, or XORC instruction, a PC break interrupt becomes valid two states after the end of the instruction execution. If a PC break interrupt is set for the instruction following one of these instructions, since interrupts, including NMI, are disabled for a 3-state period in the case of LDC, ANDC, ORC, and XOR, the next instruction is always executed. For details, see section 5, Interrupt Controller.

6.4.7 PC Break Set for Instruction Fetch at Address Following Bcc Instruction

A PC break interrupt is generated if the instruction at the next address is executed in accordance with the branch condition, and is not generated if the instruction at the next address is not executed.

6.4.8 PC Break Set for Instruction Fetch at Branch Destination Address of Bcc Instruction

A PC break interrupt is generated if the instruction at the branch destination is executed in accordance with the branch condition, and is not generated if the instruction at the branch destination is not executed.

Section 7 Bus Controller (BSC)

This LSI has an on-chip bus controller (BSC) that manages the external address space divided into eight areas.

The bus controller also has a bus arbitration function, and controls the operation of the bus masterships—the CPU and data transfer controller (DTC).

7.1 Features

- Manages external address space in area units
  Manages the external address space divided into eight areas of 2 Mbytes
  Bus specifications can be set independently for each area
  Burst ROM interface can be set
- Basic bus interface
  8-bit access or 16-bit access can be selected for each area
  2-state access or 3-state access can be selected for each area
  Program wait states can be inserted for each area
- Burst ROM interface
  Burst ROM interface can be set for area 0
  Choice of 1- or 2-state burst access
- Idle cycle insertion
  An idle cycle can be inserted between external read cycles for different areas
  An idle cycle can be inserted before an external write cycle immediately after an external read cycle
- Bus arbitration function
  Includes a bus arbiter that arbitrates bus mastership between the CPU and DTC

A block diagram of the bus controller is shown in figure 7.1.

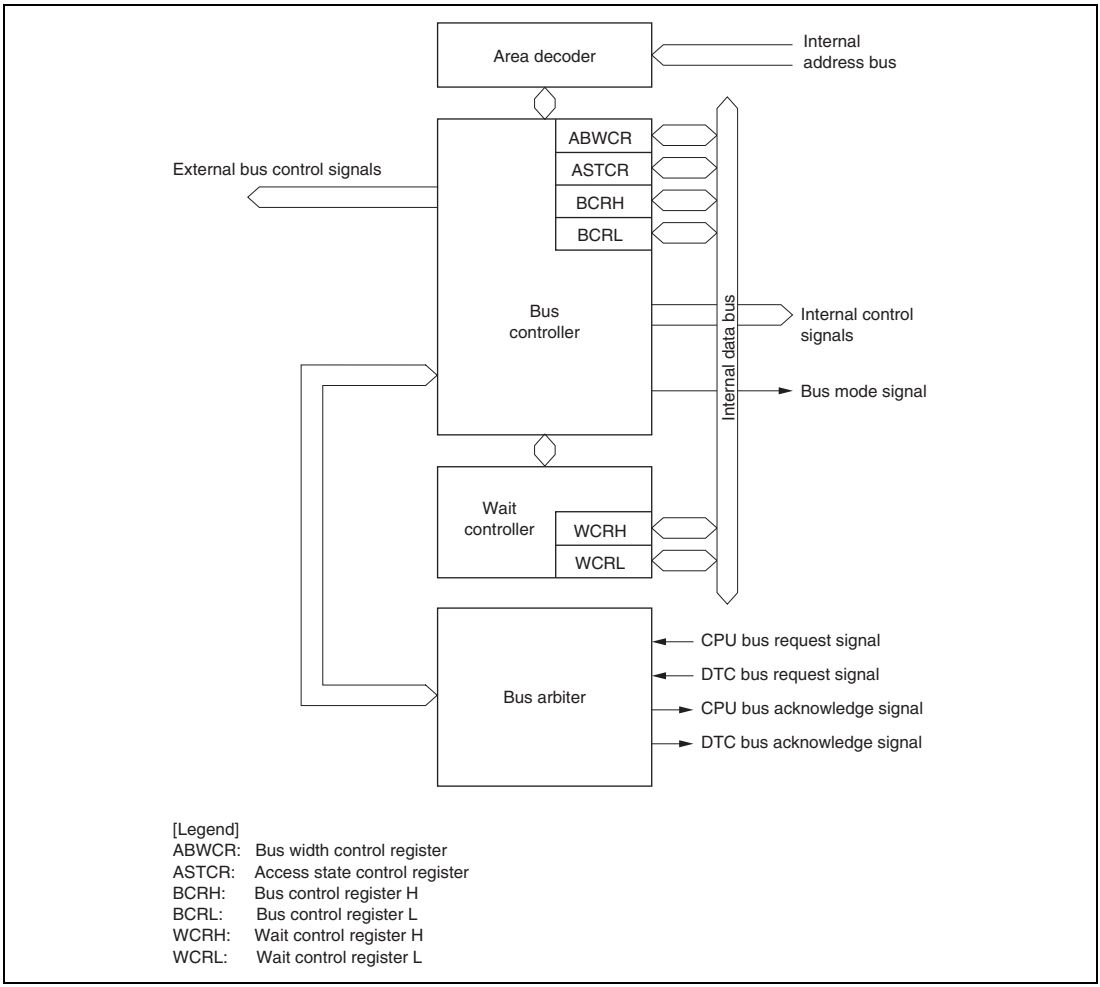


Figure 7.1 Block Diagram of Bus Controller

7.2 Input/Output Pins

Table 7.1 summarizes the pin configuration of the bus controller.

Table 7.1 Pin Configuration

Name	Symbol	I/O	Function
Address strobe	$\overline{AS}$	Output	Strobe signal indicating that an external address space is accessed and address output on address bus is enabled.
Read	$\overline{RD}$	Output	Strobe signal indicating that an external address space is being read.
High write	$\overline{HWR}$	Output	Strobe signal indicating that an external address space is written to, and upper half (D15 to D8) of data bus is enabled.
Low write	$\overline{LWR}$	Output	Strobe signal indicating that an external address space is written to, and lower half (D7 to D0) of data bus is enabled.

7.3 Register Descriptions

The bus controller has the following registers.

- Bus width control register (ABWCR)
- Access state control register (ASTCR)
- Wait control register H (WTCRH)
- Wait control register L (WTCRL)
- Bus control register H (BCRH)
- Bus control register L (BCRL)
- Pin function control register (PFCR)

7.3.1 Bus Width Control Register (ABWCR)

ABWCR designates each area in the external address space as either 8-bit access space or 16-bit access space.

Bit	Bit Name	Initial Value	R/W	Description
7	ABW7	1/0*	R/W	Area 7 to 0 Bus Width Control
6	ABW6	1/0*	R/W	These bits select whether the corresponding area is to be designated as 8-bit access space or 16-bit access space.
5	ABW5	1/0*	R/W	
4	ABW4	1/0*	R/W	0: Area n is designated as 16-bit access space
3	ABW3	1/0*	R/W	1: Area n is designated as 8-bit access space
2	ABW2	1/0*	R/W	(n = 7 to 0)
1	ABW1	1/0*	R/W	
0	ABW0	1/0*	R/W	

Note * Initial value is 0 in mode 4, and 1 in modes 5 to 7.

7.3.2 Access State Control Register (ASTCR)

ASTCR designates each area in the external address space as either 2-state access space or 3-state access space.

Bit	Bit Name	Initial Value	R/W	Description
7	AST7	1	R/W	Area 7 to 0 Access State Control
6	AST6	1	R/W	These bits select whether the corresponding area is to be designated as 2-state access space or 3-state access space. Wait state insertion is enabled or disabled at the same time.
5	AST5	1	R/W	
4	AST4	1	R/W	0: Area n is designated as 2-state access space Wait state insertion in area n access is disabled
3	AST3	1	R/W	
2	AST2	1	R/W	1: Area n is designated as 3-state access space Wait state insertion in area n access is enabled
1	AST1	1	R/W	
0	AST0	1	R/W	(n = 7 to 0)

7.3.3 Wait Control Registers H and L (WCRH and WCRL)

WCRH and WCRL select the number of program wait states for each area in the external address space.

- WCRH

Bit	Bit Name	Initial Value	R/W	Description
7	W71	1	R/W	Area 7 Wait Control 1 and 0
6	W70	1	R/W	These bits select the number of program wait states when accessing area 7 while AST7 bit in ASTCR = 1. 00: Program wait not inserted 01: 1 program wait state inserted 10: 2 program wait states inserted 11: 3 program wait states inserted
5	W61	1	R/W	Area 6 Wait Control 1 and 0
4	W60	1	R/W	These bits select the number of program wait states when accessing area 6 while AST6 bit in ASTCR = 1. 00: Program wait not inserted 01: 1 program wait state inserted 10: 2 program wait states inserted 11: 3 program wait states inserted
3	W51	1	R/W	Area 5 Wait Control 1 and 0
2	W50	1	R/W	These bits select the number of program wait states when accessing area 5 while AST5 bit in ASTCR = 1. 00: Program wait not inserted 01: 1 program wait state inserted 10: 2 program wait states inserted 11: 3 program wait states inserted
1	W41	1	R/W	Area 4 Wait Control 1 and 0
0	W40	1	R/W	These bits select the number of program wait states when accessing area 4 while AST4 bit in ASTCR = 1. 00: Program wait not inserted 01: 1 program wait state inserted 10: 2 program wait states inserted 11: 3 program wait states inserted

- WCRL

Bit	Bit Name	Initial Value	R/W	Description
7	W31	1	R/W	Area 3 Wait Control 1 and 0
6	W30	1	R/W	These bits select the number of program wait states when accessing area 3 while AST3 bit in ASTCR = 1. 00: Program wait not inserted 01: 1 program wait state inserted 10: 2 program wait states inserted 11: 3 program wait states inserted
5	W21	1	R/W	Area 2 Wait Control 1 and 0
4	W20	1	R/W	These bits select the number of program wait states when accessing area 2 while AST2 bit in ASTCR = 1. 00: Program wait not inserted 01: 1 program wait state inserted 10: 2 program wait states inserted 11: 3 program wait states inserted
3	W11	1	R/W	Area 1 Wait Control 1 and 0
2	W10	1	R/W	These bits select the number of program wait states when accessing area 1 while AST1 bit in ASTCR = 1. 00: Program wait not inserted 01: 1 program wait state inserted 10: 2 program wait states inserted 11: 3 program wait states inserted
1	W01	1	R/W	Area 0 Wait Control 1 and 0
0	W00	1	R/W	These bits select the number of program wait states when accessing area 0 while AST0 bit in ASTCR = 1. 00: Program wait not inserted 01: 1 program wait state inserted 10: 2 program wait states inserted 11: 3 program wait states inserted

7.3.4 Bus Control Register H (BCRH)

BCRH enables or disables idle cycle insertion and specifies the burst ROM interface.

Bit	Bit Name	Initial Value	R/W	Description
7	ICIS1	1	R/W	Idle Cycle Insert 1 When consecutive external read cycles are performed in different areas, an idle cycle can be inserted between the bus cycles. 0: Idle cycle not inserted 1: Idle cycle inserted
6	ICIS0	1	R/W	Idle Cycle Insert 0 When an external read cycle and an external write cycle are performed consecutively, an idle cycle can be inserted between the bus cycles. 0: Idle cycle not inserted 1: Idle cycle inserted
5	BRSTRM	0	R/W	Burst ROM Enable Selects whether the burst ROM interface is used for area 0. 0: Basic bus interface for area 0 1: Burst ROM interface for area 0
4	BRSTS1	1	R/W	Burst Cycle Select 1 Selects the number of burst cycles for the burst ROM interface. 0: 1 state for a burst cycle 1: 2 states for a burst cycle
3	BRSTS0	0	R/W	Burst Cycle Select 0 Selects the number of words that can be accessed in a burst access with the burst ROM interface. 0: Maximum four words in burst access 1: Maximum eight words in burst access
2 to 0	—	All 0	R/W	Reserved The write value should always be 0.

7.3.5 Bus Control Register L (BCRL)

BCRL selects the write data buffer function and enables or disables input to the $\overline{\text{WAIT}}$ pin.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved
6	—	0	R/W	The write value should always be 0.
5	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
4	—	0	R/W	Reserved The write value should always be 0.
3	—	1	R/W	Reserved The write value should always be 1.
2	—	0	R/W	Reserved The write value should always be 0.
1	WDBE	0	R/W	Write Data Buffer Enable Selects the write data buffer function for an external write cycle. 0: Write data buffer function not used 1: Write data buffer function used
0	—	0	R/W	Reserved This bit is always read as 0 and cannot be modified.

7.3.6 Pin Function Control Register (PFCR)

PFCR controls the address output in expanded mode with on-chip ROM enabled.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved The write value should always be 0.
3	AE3	0/1*	R/W	Address Output Enable 3 to 0 These bits enable or disable address outputs A8 to A23 in expanded mode with on-chip ROM. When a pin is enabled for address output, the address is output regardless of the corresponding DDR setting. For a pin disabled for address output, each becomes an output port when the corresponding DDR bit is set to 1. For pins A7 to A0, each becomes an address output when the corresponding DDR bit is set to 1. 0000: A8 to A23 output disabled 0001: A8 output enabled; A9 to A23 output disabled 0010: A8 and A9 output enabled; A10 to A23 output disabled 0011: A8 to A10 output enabled; A11 to A23 output disabled 0100: A8 to A11 output enabled; A12 to A23 output disabled 0101: A8 to A12 output enabled; A13 to A23 output disabled 0110: A8 to A13 output enabled; A14 to A23 output disabled 0111: A8 to A14 output enabled; A15 to A23 output disabled 1000: A8 to A15 output enabled; A16 to A23 output disabled 1001: A8 to A16 output enabled; A17 to A23 output disabled 1010: A8 to A17 output enabled; A18 to A23 output disabled 1011: A8 to A18 output enabled; A19 to A23 output disabled 1100: A8 to A19 output enabled; A20 to A23 output disabled 1101: A8 to A20 output enabled; A21 to A23 output disabled 1110: A8 to A21 output enabled; A22 and A23 output disabled 1111: A8 to A23 output enabled
2	AE2	0/1*	R/W	
1	AE1	0	R/W	
0	AE0	0/1*	R/W	

Note: * In expanded mode with on-chip ROM enabled, bits 3 to 0 are initialized to B'0000. In expanded mode with on-chip ROM disabled, bits 3 to 0 are initialized to B'1101.

7.4 Bus Control

7.4.1 Area Division

The bus controller divides the 16-Mbyte address space into eight areas, 0 to 7, in 2-Mbyte units, and performs bus control for external address space in area units. Figure 7.2 shows an outline of the memory map.

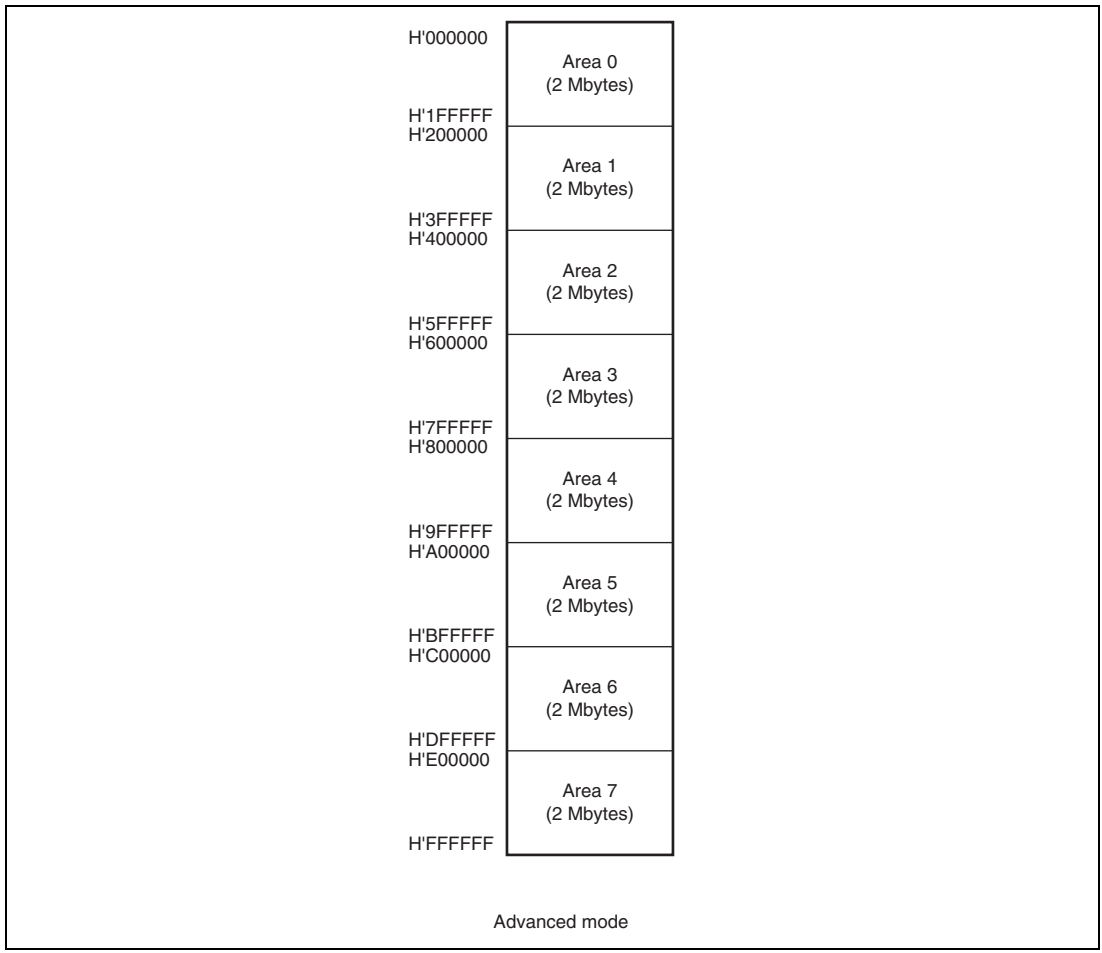


Figure 7.2 Area Divisions

7.4.2 Bus Specifications

The external address space bus specifications consist of three elements: bus width, number of access states, and number of 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.

Bus Width: A bus width of 8 or 16 bits can be selected with ABWCR. An area for which an 8-bit bus is selected functions as an 8-bit access space, and an area for which a 16-bit bus is selected functions as a 16-bit access space. If all areas are designated as 8-bit access space, 8-bit bus mode is set; if any area is designated as 16-bit access space, 16-bit bus mode is set.

Number of Access States: Two or three access states can be selected with ASTCR. An area for which 2-state access is selected functions as a 2-state access space, and an area for which 3-state access is selected functions as a 3-state access space.

When 2-state access space is designated, wait insertion is disabled. When 3-state access space is designated, it is possible to insert program waits by means of the WCRH and WCRL, and external waits by means of the $\overline{\text{WAIT}}$ pin.

Number of Program Wait States: When 3-state access space is designated by ASTCR, the number of program wait states to be inserted automatically is selected with WCRH and WCRL. From 0 to 3 program wait states can be selected.

Table 7.2 shows the bus specifications (bus width, number of access states, and program wait states) for each basic bus interface area.

Table 7.2 Bus Specifications for Each Area (Basic Bus Interface)

ABWCR	ASTCR	WCRH, WCRL		Bus Specifications (Basic Bus Interface)		
ABWn	ASTn	Wn1	Wn0	Bus Width	Access States	Program Wait States
0	0	—	—	16	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3
1	0	—	—	8	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3

(n = 0 to 7)

7.4.3 Memory Interfaces

The memory interfaces in this LSI allow direct connection of ROM, SRAM, and so on.

The initial state of each area is 3-state access space with the basic bus interface. The initial bus width is selected according to the operating mode.

Area 0: Area 0 includes on-chip ROM in expanded mode with on-chip ROM enabled and the space excluding on-chip ROM is external address space. In expanded mode with on-chip ROM disabled, all of area 0 is external address space.

Either basic bus interface or burst ROM interface can be selected for area 0.

Areas 1 to 6: In externally expanded mode, areas 1 to 6 are all external address space.

Only the basic bus interface can be used for areas 1 to 6.

Area 7: Area 7 includes the on-chip RAM and internal I/O registers. In externally expanded mode, the space excluding the on-chip RAM and internal I/O registers is external address space. The on-chip RAM is enabled when the RAME bit is set to 1 in the system control register (SYSCR); when the RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding addresses are in external address space.

Only the basic bus interface can be used for area 7.

7.5 Basic Bus Interface

The basic bus interface enables direct connection of ROM, SRAM, and so on.

7.5.1 Data Size and Data Alignment

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

8-Bit Access Space: Figure 7.3 illustrates data alignment control for the 8-bit access space. With the 8-bit access space, the upper data bus (D15 to D8) is always used for accesses. The amount of data that can be accessed at one time is one byte: a word access is performed as two byte accesses, and a longword access, as four byte accesses.

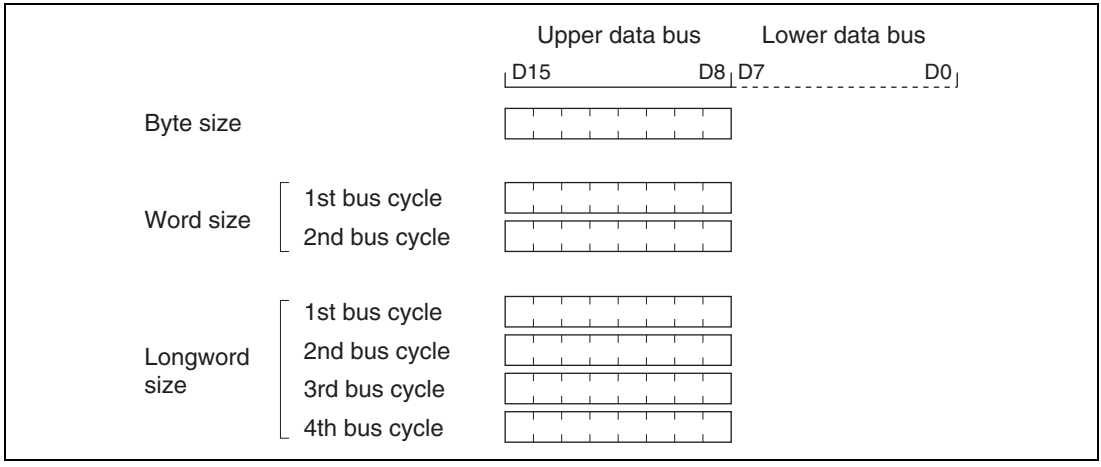


Figure 7.3 Access Sizes and Data Alignment Control (8-Bit Access Space)

16-Bit Access Space: Figure 7.4 illustrates data alignment control for the 16-bit access space. With the 16-bit access space, the upper data bus (D15 to D8) and lower data bus (D7 to D0) are used for accesses. The amount of data that can be accessed at one time is one byte or one word, and a longword access is executed as two word accesses.

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

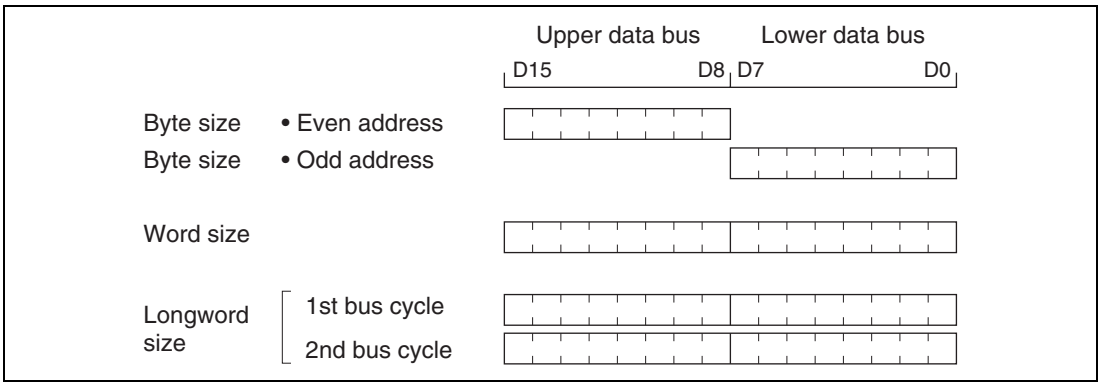


Figure 7.4 Access Sizes and Data Alignment Control (16-bit Access Space)

7.5.2 Valid Strobes

Table 7.3 shows the data buses used and valid strobes for the access spaces.

In a read, the $\overline{RD}$ signal is valid for both the upper and the lower half 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 7.3 Data Buses Used and Valid Strobes

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D15 to D8)	Lower Data Bus (D7 to D0)
8-bit access space	Byte	Read	—	$\overline{RD}$	Valid	Invalid
		Write	—	$\overline{HWR}$		Hi-Z
16-bit access space	Byte	Read	Even	$\overline{RD}$	Valid	Invalid
			Odd		Invalid	Valid
		Write	Even	$\overline{HWR}$	Valid	Hi-Z
			Odd	$\overline{LWR}$	Hi-Z	Valid
	Word	Read	—	$\overline{RD}$	Valid	Valid
		Write	—	$\overline{HWR}$, $\overline{LWR}$	Valid	Valid

Note: Hi-Z: High-impedance state
Invalid: Input state; input value is ignored.

7.5.3 Basic Timing

8-Bit, 2-State Access Space: Figure 7.5 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. The $\overline{\text{LWR}}$ pin is always fixed high. Wait states cannot be inserted.

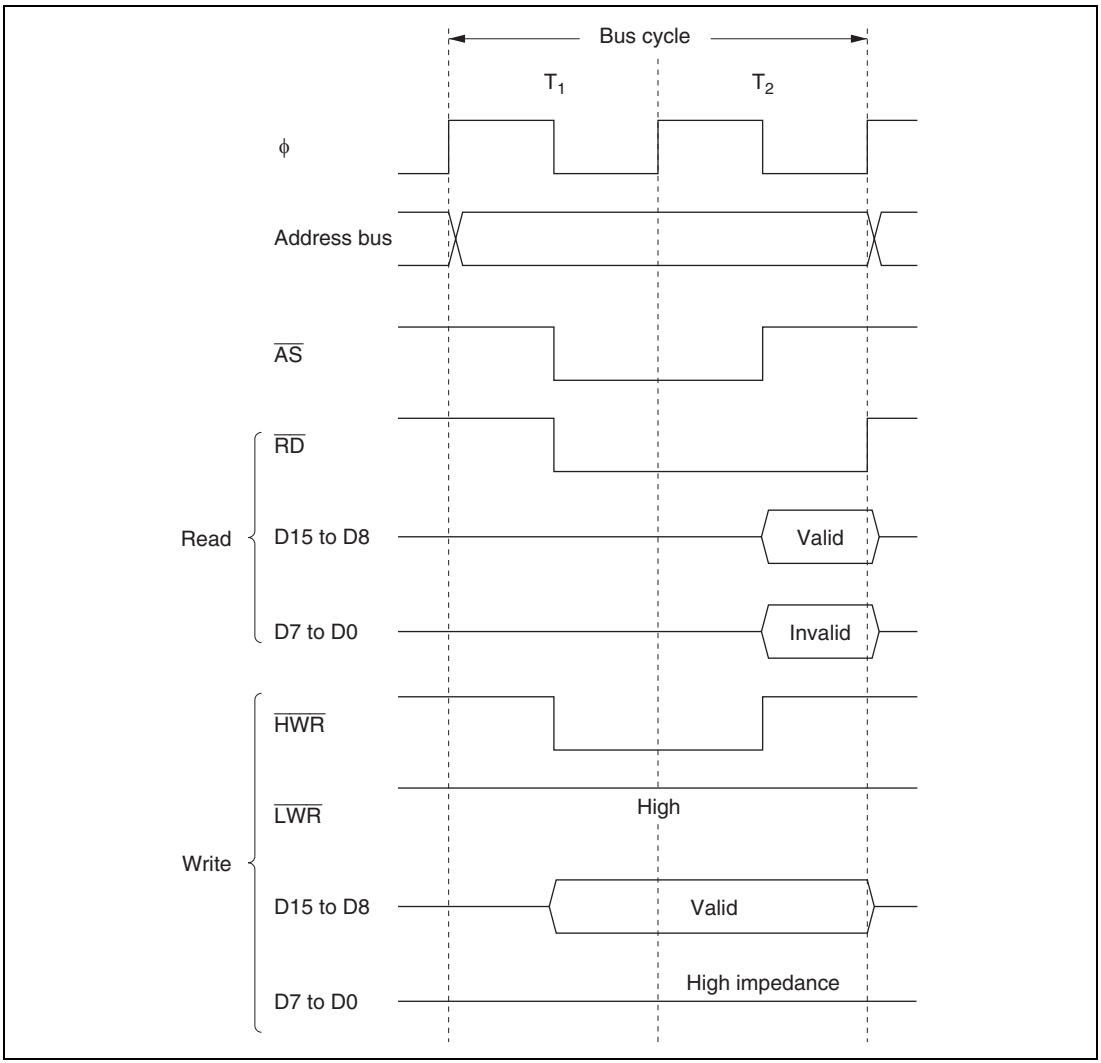


Figure 7.5 Bus Timing for 8-Bit, 2-State Access Space

8-Bit, 3-State Access Space: Figure 7.6 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. The $\overline{\text{LWR}}$ pin is always fixed high. Wait states can be inserted.

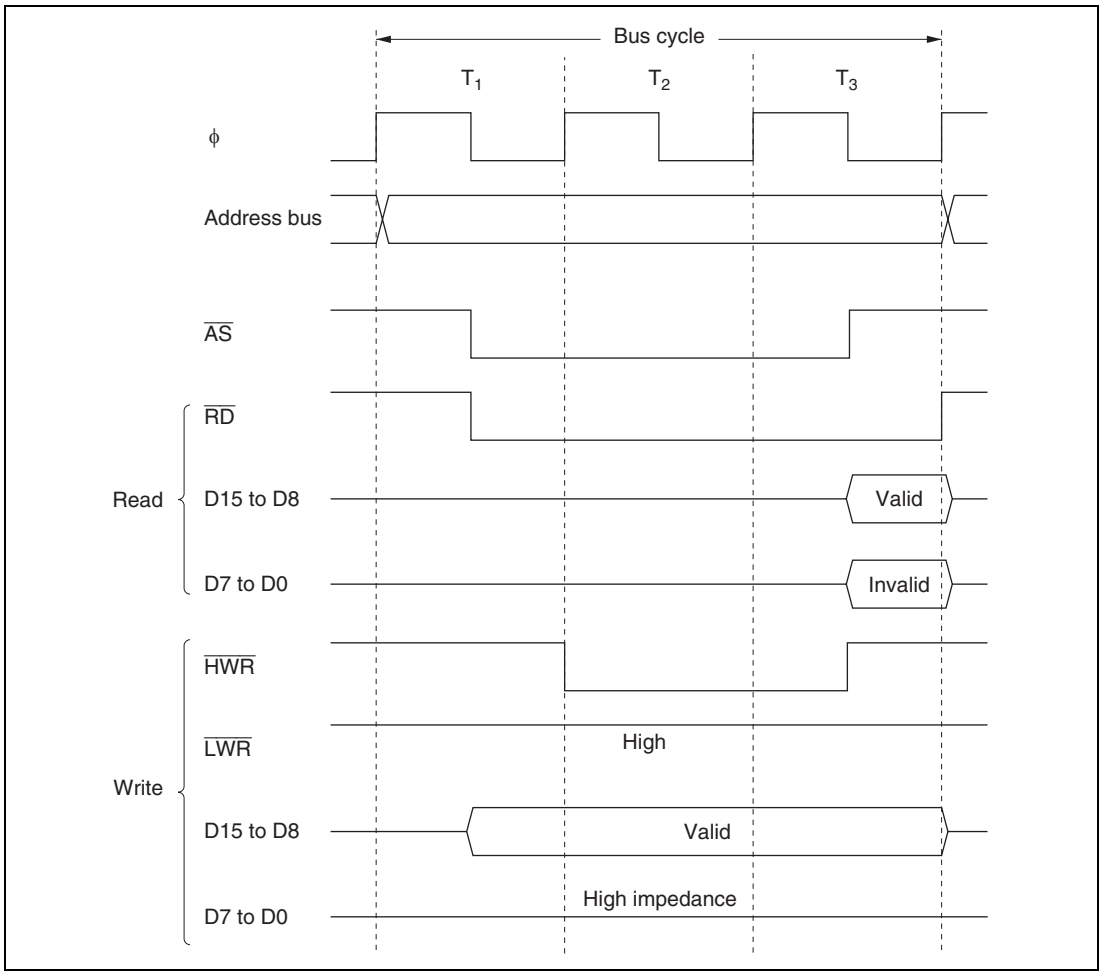


Figure 7.6 Bus Timing for 8-Bit, 3-State Access Space

16-Bit, 2-State Access Space: Figures 7.7 to 7.9 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 odd addresses, and the lower half (D7 to D0) for even addresses. Wait states cannot be inserted.

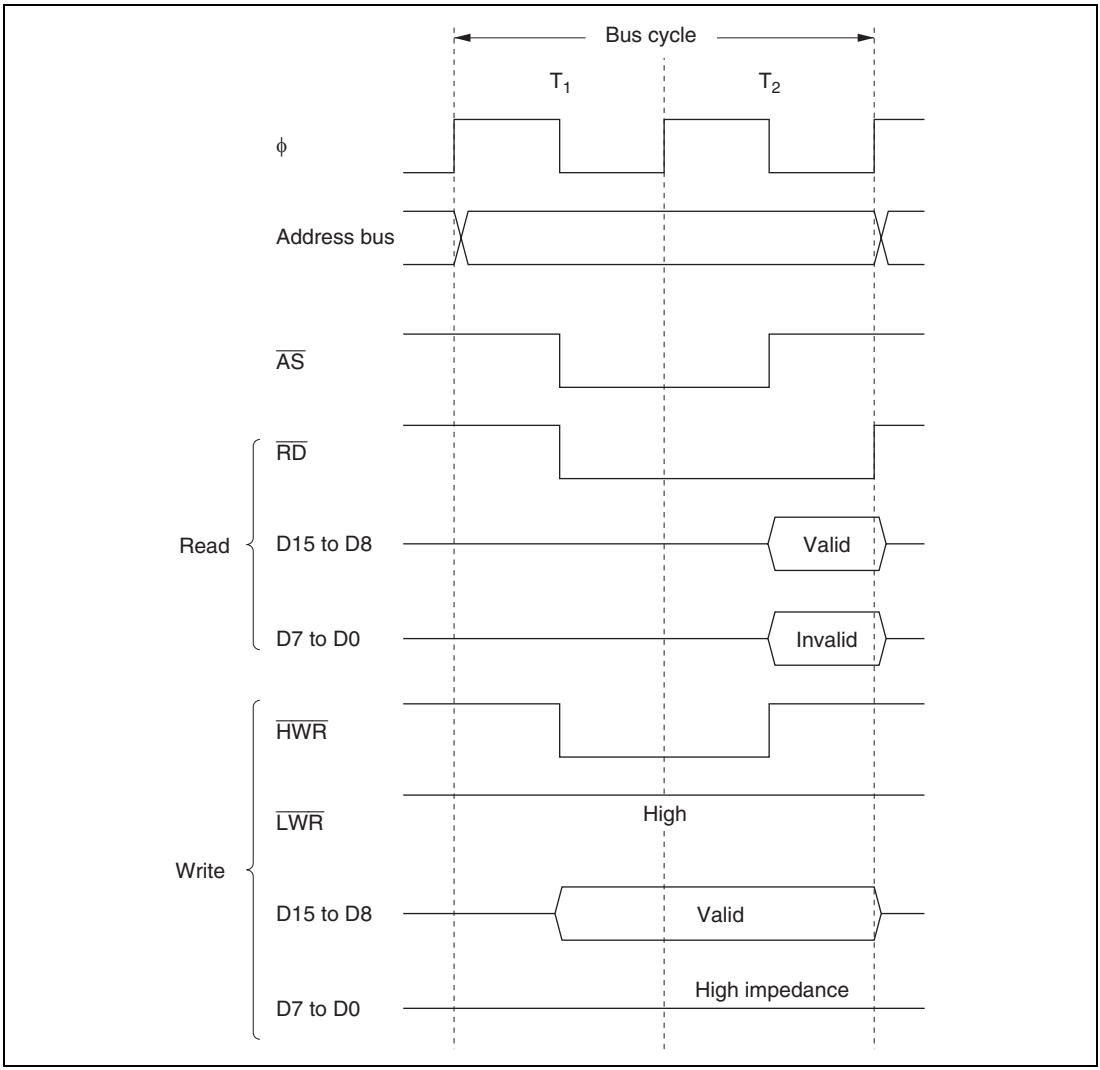


Figure 7.7 Bus Timing for 16-Bit, 2-State Access Space (Even Address Byte Access)

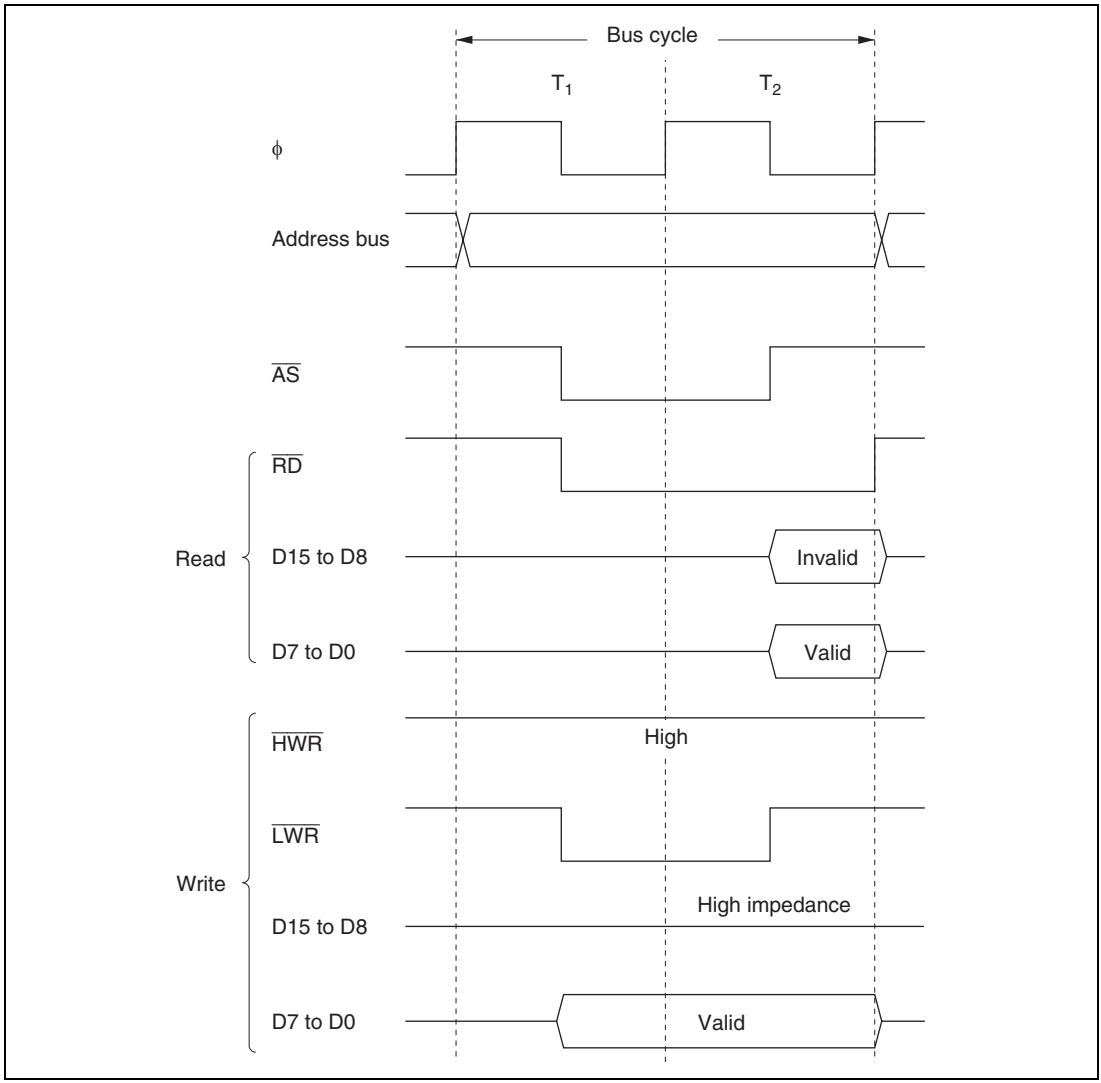


Figure 7.8 Bus Timing for 16-Bit, 2-State Access Space (Odd Address Byte Access)

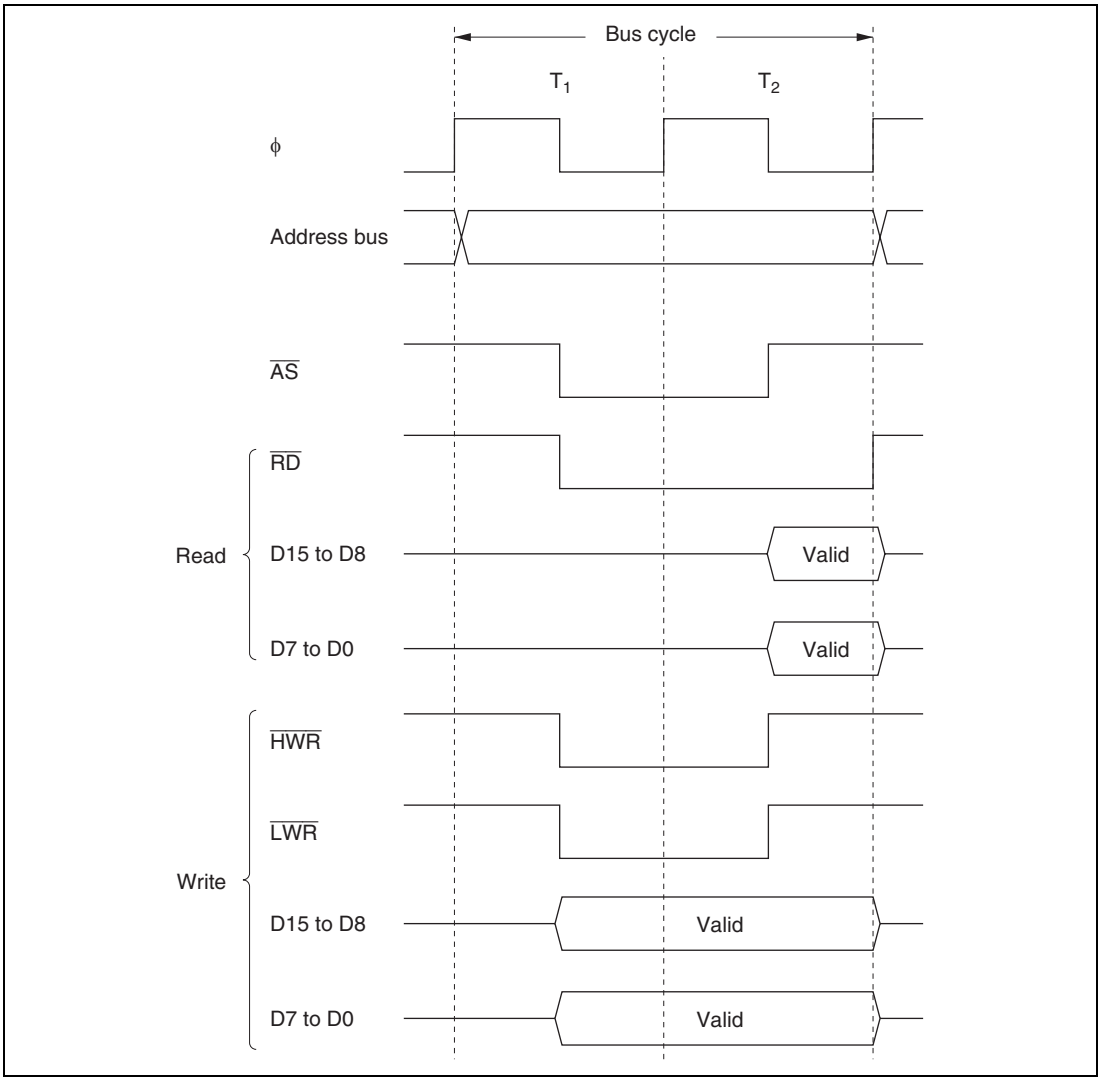


Figure 7.9 Bus Timing for 16-Bit, 2-State Access Space (Word Access)

16-Bit, 3-State Access Space: Figures 7.10 to 7.12 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 the even address, and the lower half (D7 to D0) for the odd address. Wait states can be inserted.

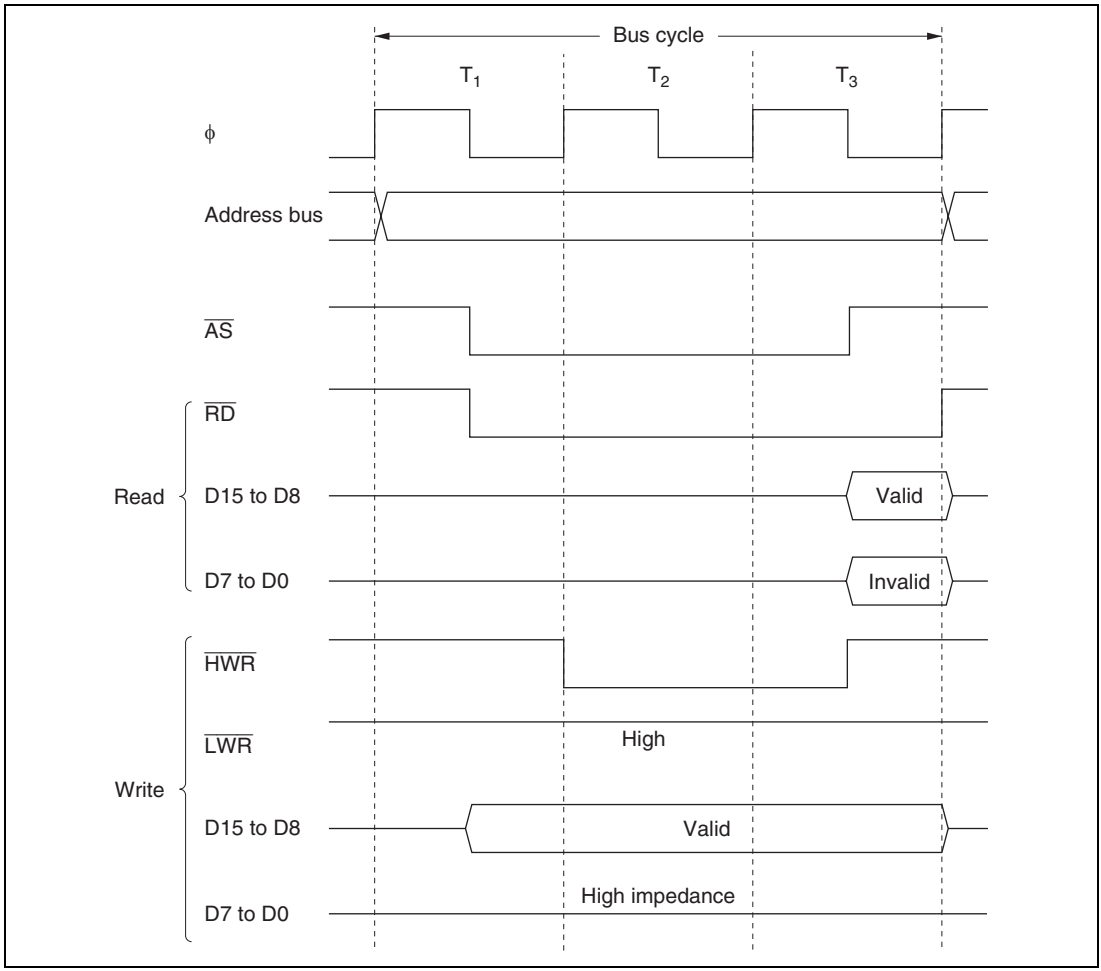


Figure 7.10 Bus Timing for 16-Bit, 3-State Access Space (Even Address Byte Access)

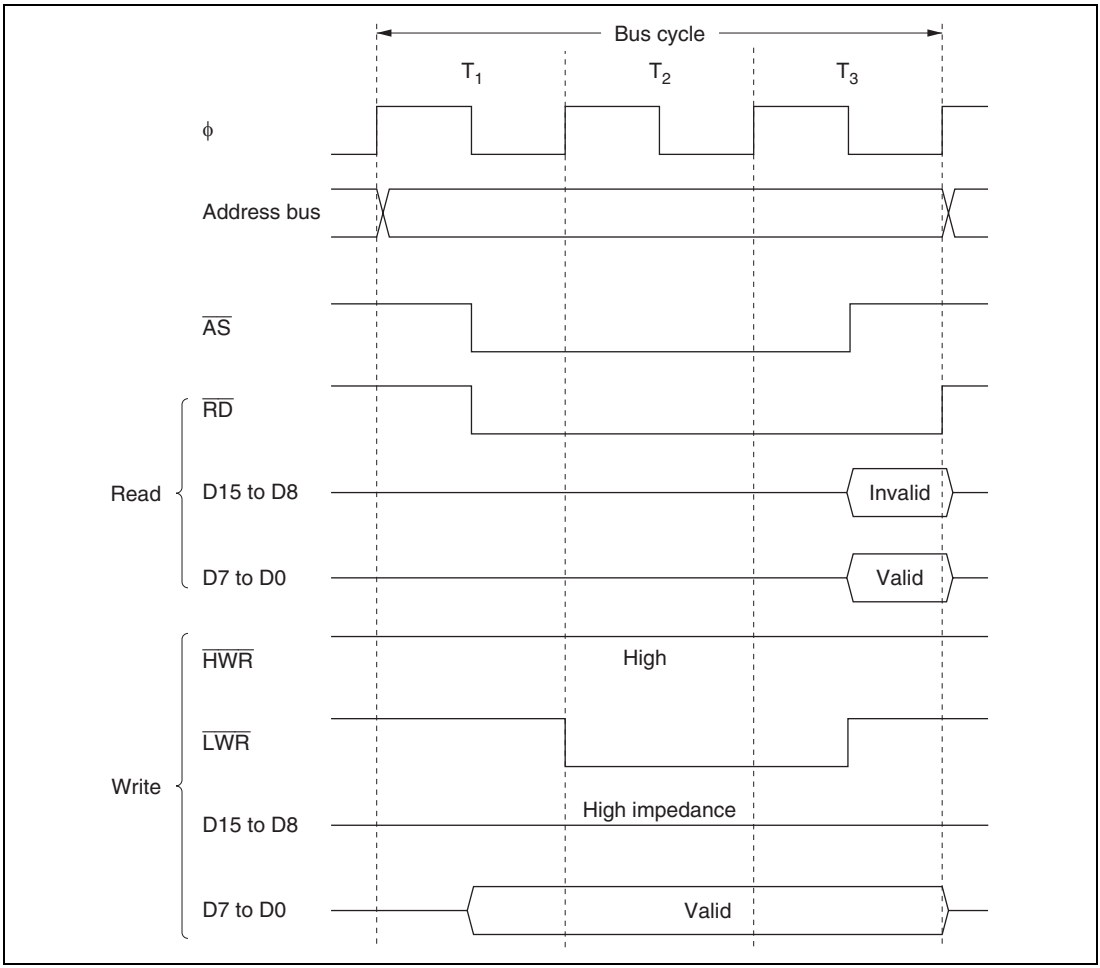


Figure 7.11 Bus Timing for 16-Bit, 3-State Access Space (Odd Address Byte Access)

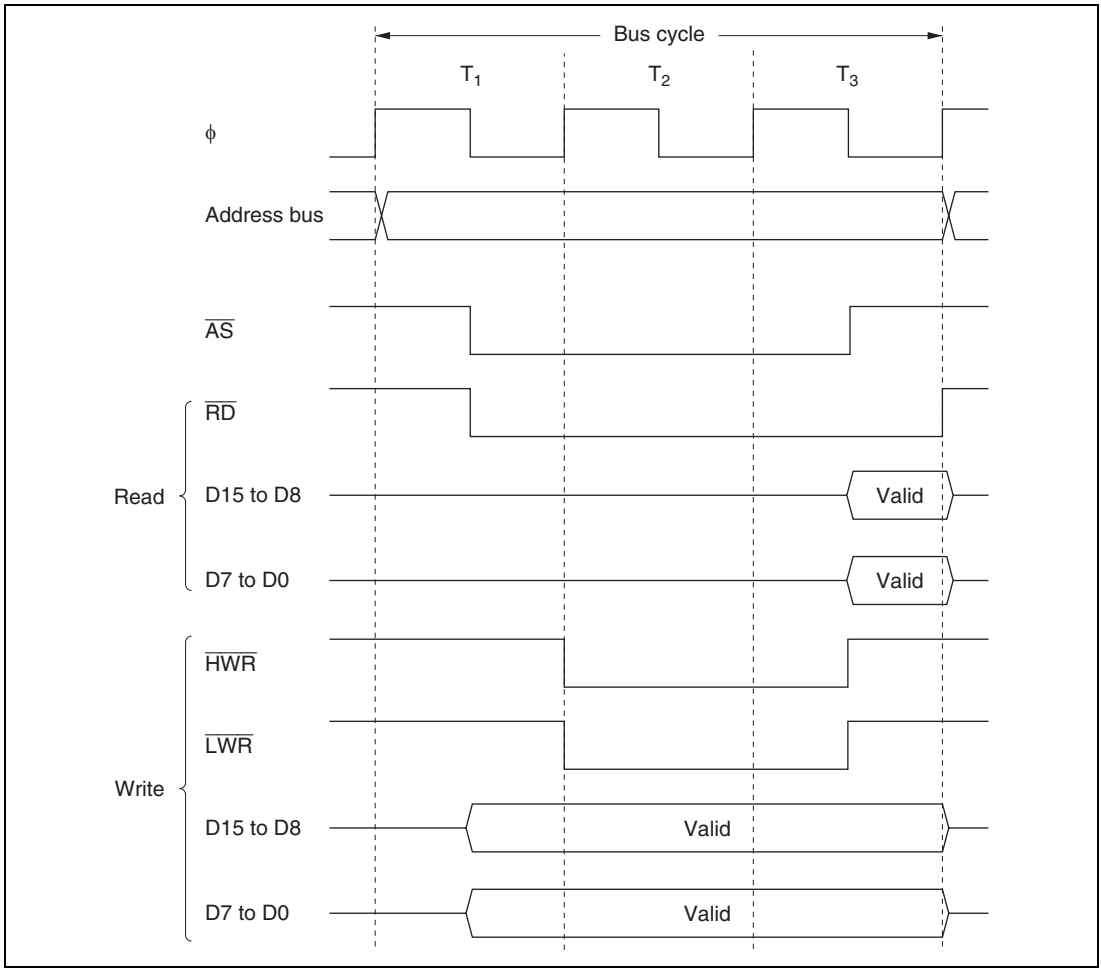


Figure 7.12 Bus Timing for 16-Bit, 3-State Access Space (Word Access)

7.5.4 Wait Control

When accessing external space, this LSI can extend the bus cycle by inserting one or more wait states (T_w). There is a way of inserting wait states: program wait insertion.

Program Wait Insertion: From 0 to 3 wait states can be inserted automatically between the T_2 state and T_3 state on an individual area basis in 3-state access space, according to the settings in WCRH and WCRL.

Figure 7.13 shows an example of wait state insertion timing.

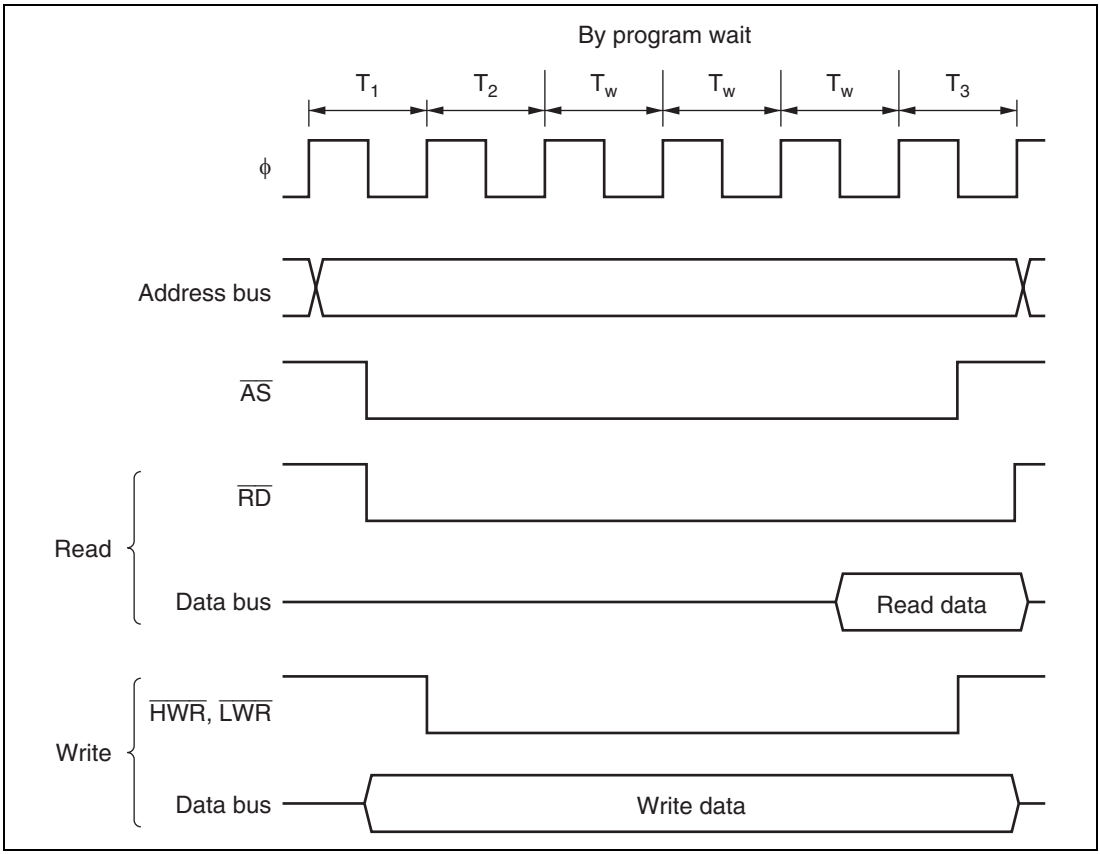


Figure 7.13 Example of Wait State Insertion Timing

After a power-on reset: 3-state access and insertion of three program wait states

7.6 Burst ROM Interface

In this LSI, external space area 0 can be designated as burst ROM space, and burst ROM interfacing performed. The burst ROM space interface enables ROM with burst access capability to be accessed at high speed.

Area 0 can be designated as burst ROM space by means of bit BSTRM in BCRH. Continuous burst accesses of four or eight words can be performed, according to the setting of the BRSTS0 bit in BCRH. One or two states can be selected for burst access.

In burst ROM interface space, burst access covers only CPU read accesses.

7.6.1 Basic Timing

The number of access states in the initial cycle (full access) with the burst ROM interface is determined by the AST0 setting in ASTCR. Wait states can be inserted when the AST0 bit is set to 1. One or two states can be selected for the burst cycle according to the BRSTS1 bit setting in BCRH. Wait states cannot be inserted. When area 0 is designated as burst ROM space, area 0 is a 16-bit access space regardless of the ABW0 bit setting in ABWCR.

When the BRSTS0 bit in BCRH is cleared to 0, burst access of up to four words is performed. When the BRSTS0 bit is set to 1, burst access of up to eight words is performed.

The basic access timing for burst ROM space is shown in figures 7.14 and 7.15.

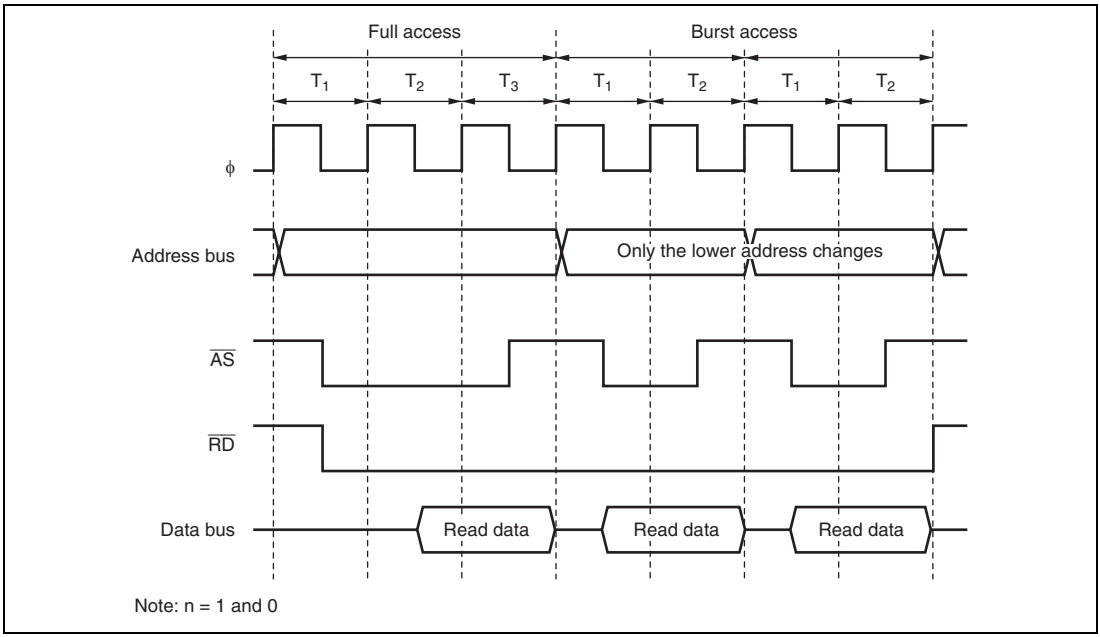


Figure 7.14 Example of Burst ROM Access Timing (AST0 = 1 and BRSTS0 = 1)

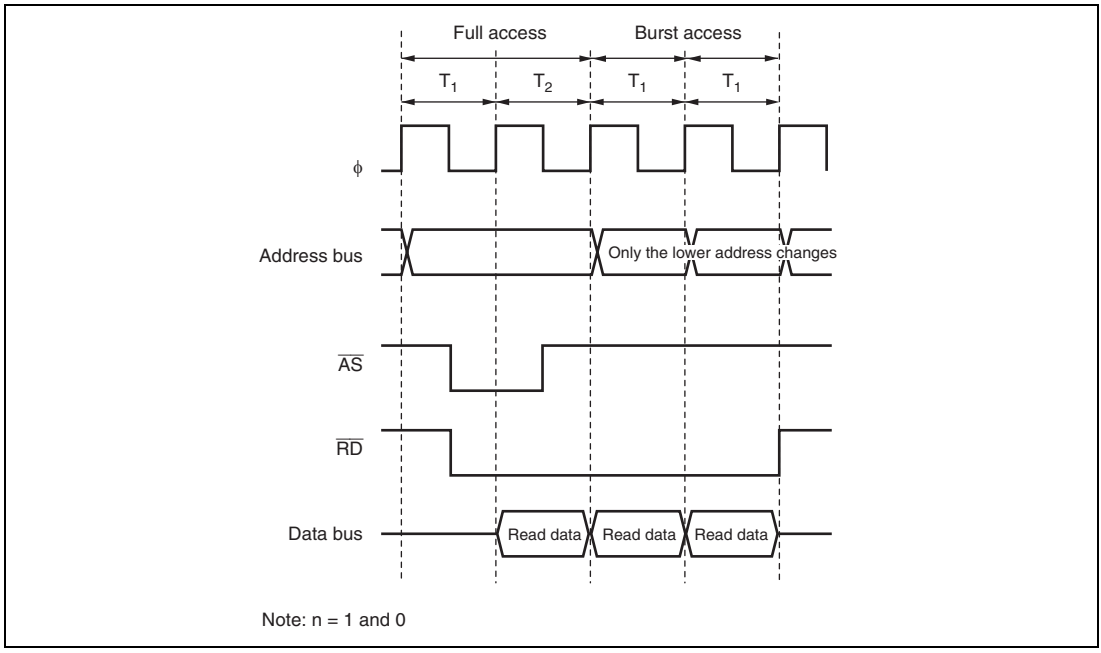


Figure 7.15 Example of Burst ROM Access Timing (AST0 = 0 and BRSTS1 = 0)

7.6.2 Wait Control

As with the basic bus interface, either program wait insertion or pin wait insertion using the $\overline{\text{WAIT}}$ pin can be used in the initial cycle (full access) on the burst ROM interface. See section 7.5.4, Wait Control. Wait states cannot be inserted in a burst cycle.

7.6.3 Write Access

When a write access to burst ROM interface space is executed, burst access is interrupted at that point and the write access is executed in line with the basic bus interface settings. Write accesses are not performed in burst mode even though burst ROM space is designated.

7.7 Idle Cycle

7.7.1 Operation

When this LSI accesses external address space, it can insert a 1-state idle cycle (T_i) between bus cycles in the following two cases: (1) when read accesses in different areas occur consecutively and (2) 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 memory with a long output floating time (such as ROM) and high-speed memory, I/O interfaces, and so on.

Consecutive Reads in Different Areas: If consecutive reads in different areas occur while the ICIS1 bit is set to 1 in BCRH, an idle cycle is inserted at the start of the second read cycle.

Figure 7.16 shows an example of the operation in this case. In this example, bus cycle A is a read cycle for ROM with a long output floating time, and bus cycle B is a read cycle for SRAM, each being located in a different area. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is inserted, and a data collision is prevented.

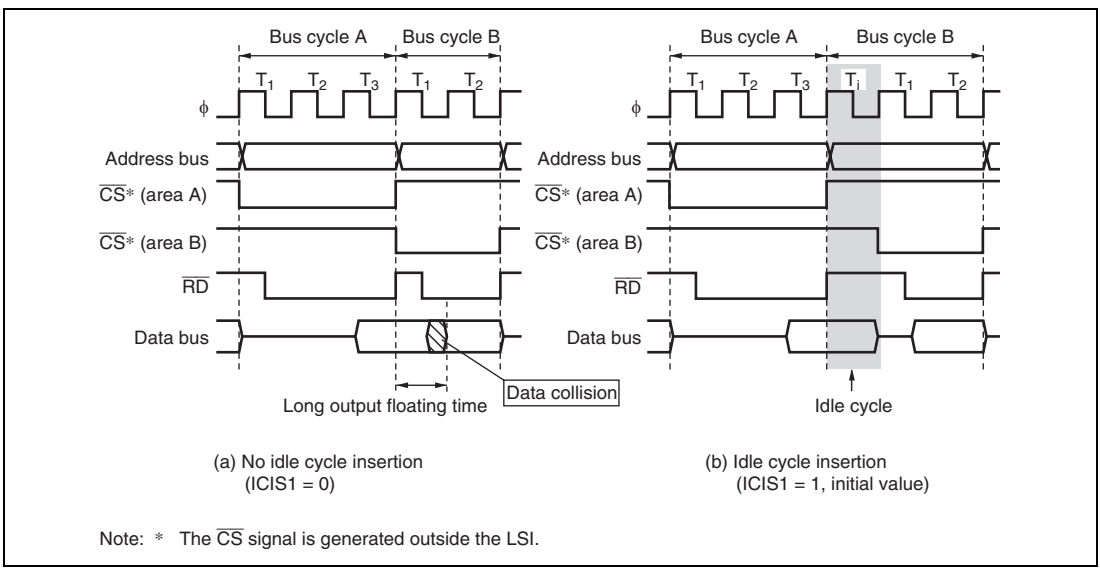


Figure 7.16 Example of Idle Cycle Operation (Consecutive Reads in Different Areas)

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

Figure 7.17 shows an example of the operation in this case. In this example, 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 (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the data read from ROM and the CPU write data. In (b), an idle cycle is inserted, and a data collision is prevented.

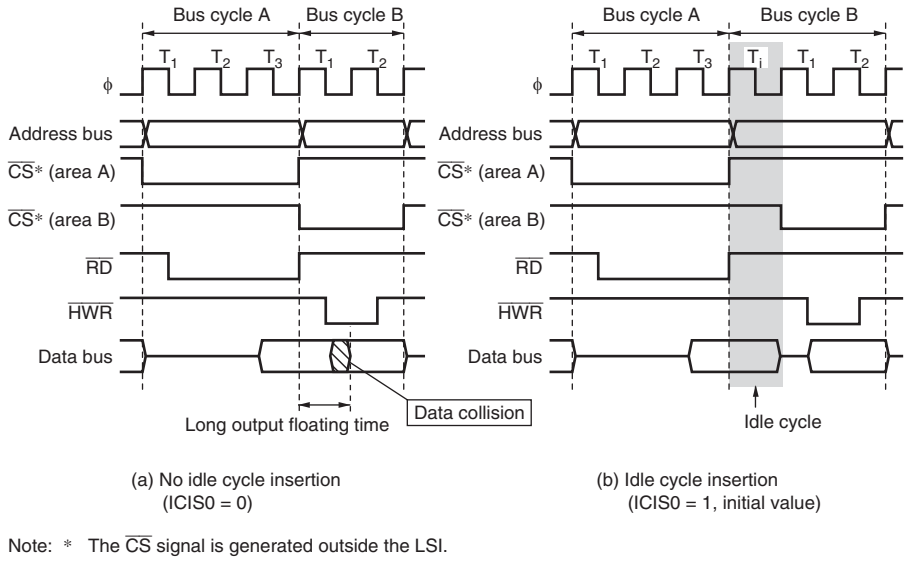


Figure 7.17 Example of Idle Cycle Operation (Write after Read)

Relationship between Chip Select ($\overline{CS}$) Signal and Read ($\overline{RD}$) Signal: Depending on the system's load conditions, the $\overline{RD}$ signal may lag behind the $\overline{CS}$ signal (generated outside the LSI). An example is shown in figure 7.18. In this case, with the setting for no idle cycle insertion (a), there may be a period of overlap between the bus cycle A $\overline{RD}$ signal and the bus cycle B $\overline{CS}$ signal. Setting idle cycle insertion, as in (b), however, will prevent any overlap between the $\overline{RD}$ and $\overline{CS}$ signals. In the initial state after reset release, idle cycle insertion (b) is set.

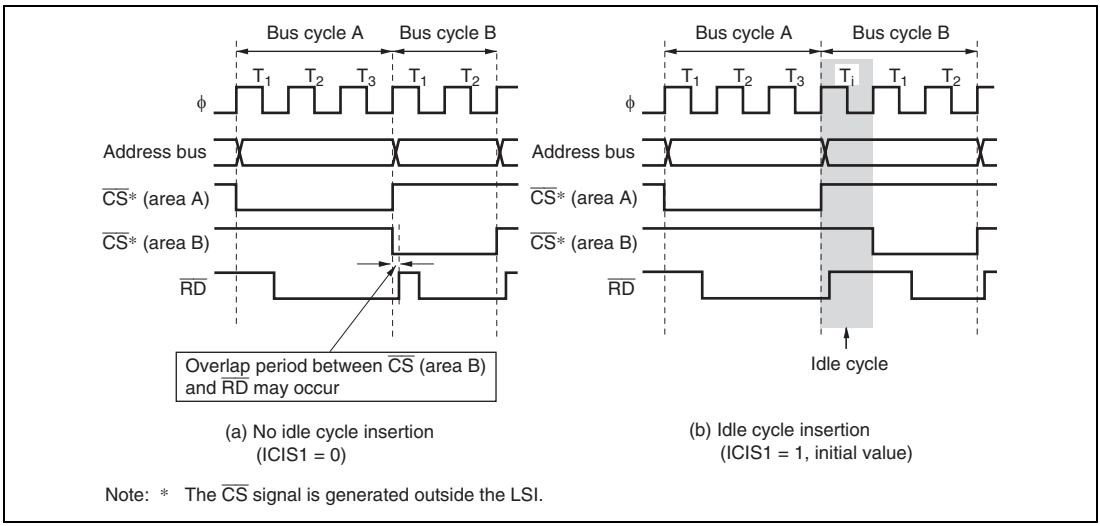


Figure 7.18 Relationship between Chip Select ($\overline{CS}$) and Read ($\overline{RD}$)

7.7.2 Pin States in Idle Cycle

Table 7.4 shows the pin states in an idle cycle.

Table 7.4 Pin States in Idle Cycle

Pins	Pin State
A23 to A0	Contents of following bus cycle
D15 to D0	High impedance
$\overline{AS}$	High
$\overline{RD}$	High
$\overline{HWR}$, $\overline{LWR}$	High

7.8 Write Data Buffer Function

This LSI has a write data buffer function for the external data bus. Using the write data buffer function enables external writes to be executed in parallel with internal accesses. The write data buffer function is made available by setting the WDBE bit to 1 in BCRL.

Figure 7.19 shows an example of the timing when the write data buffer function is used. When this function is used, if an external address space write continues for two states or longer, and there is an internal access next, an external write only is executed in the first state, but from the next state onward an internal access (on-chip memory or internal I/O register read/write) is executed in parallel with the external address space write rather than waiting until it ends.

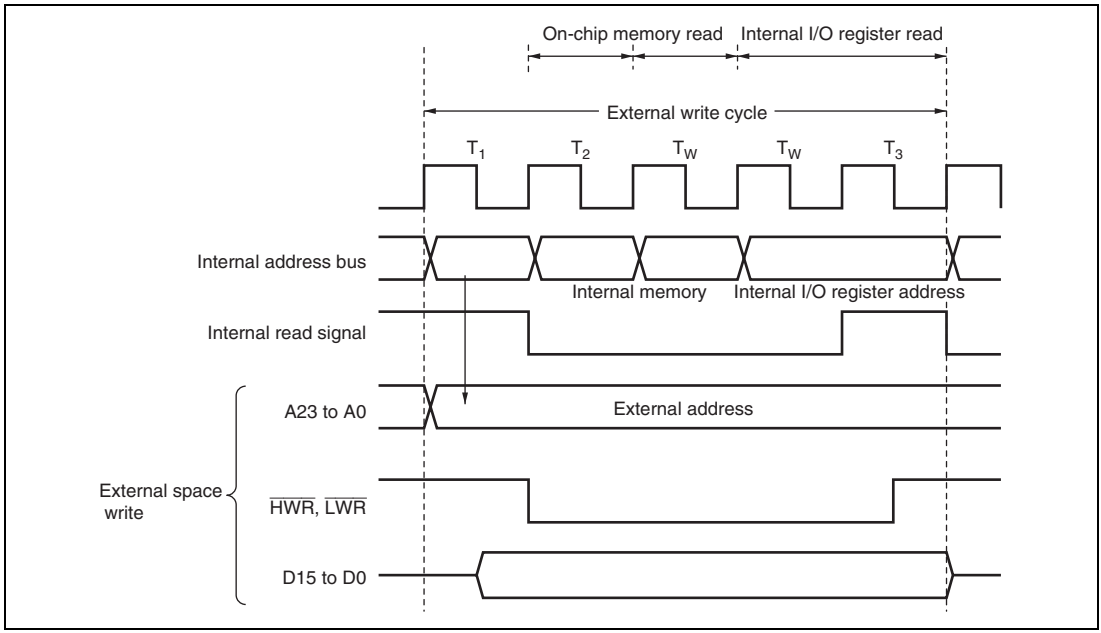


Figure 7.19 Example of Timing when Write Data Buffer Function is Used

7.9 Bus Arbitration

This LSI has a bus arbiter that arbitrates bus mastership operations (bus arbitration).

There are two bus masters—the CPU and DTC—that perform read/write operations when they have the bus mastership. Each bus master requests the bus by means of a bus request signal. The bus arbiter determines priorities at the prescribed timing, and permits use of the bus by means of a bus request acknowledge signal. The selected bus master then takes the bus mastership and begins its operation.

7.9.1 Operation

The bus arbiter detects the bus masters' bus request signals, and if the bus is requested, sends a bus request acknowledge signal to the bus master. If there are bus requests from more than one bus master, the bus request acknowledge signal is sent to the one with the highest priority. When a bus master receives the bus request acknowledge signal, it takes the bus mastership until that signal is canceled.

The order of priority of the bus masterships is as follows:

(High) DTC > CPU (Low)

7.9.2 Bus Transfer Timing

Even if a bus request is received from a bus master with a higher priority than that of the bus master that has acquired the bus and is currently operating, the bus is not necessarily transferred immediately. There are specific timings at which each bus master can relinquish the bus.

CPU: The CPU is the lowest-priority bus master, and if a bus request is received from the DTC, the bus arbiter transfers the bus to the bus master that issued the request. The timing for transfer of the bus is as follows:

- The bus is transferred at a break between bus cycles. However, if a bus cycle is executed in discrete operations, as in the case of a longword-size access, the bus is not transferred between the component operations.
- With bit manipulation instructions such as BSET and BCRL, the sequence of operations is: data read (read), relevant bit manipulation operation (modify), write-back (write). The bus is not transferred during this read-modify-write cycle, which is executed as a series of bus cycles.
- If the CPU is in sleep mode, the bus is transferred immediately.

DTC: The DTC sends the bus arbiter a request for the bus when an activation request is generated.

The DTC can release the bus after a vector read, a register information read (3 states), a single data transfer, or a register information write (3 states). It does not release the bus during a register information read (3 states), a single data transfer, or a register information write (3 states).

7.10 Bus Controller Operation in Reset

In a power-on reset, this LSI, including the bus controller, enters the reset state immediately, and any executing bus cycle is aborted.

Section 8 Data Transfer Controller (DTC)

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

Figure 8.1 shows a block diagram of the DTC.

The DTC's register information is stored in the on-chip RAM. When the DTC is used, the RAME bit in SYSCR must be set to 1. A 32-bit bus connects the DTC to the on-chip RAM (1 kbyte), enabling 32-bit/1-state reading and writing of the DTC register information.

8.1 Features

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

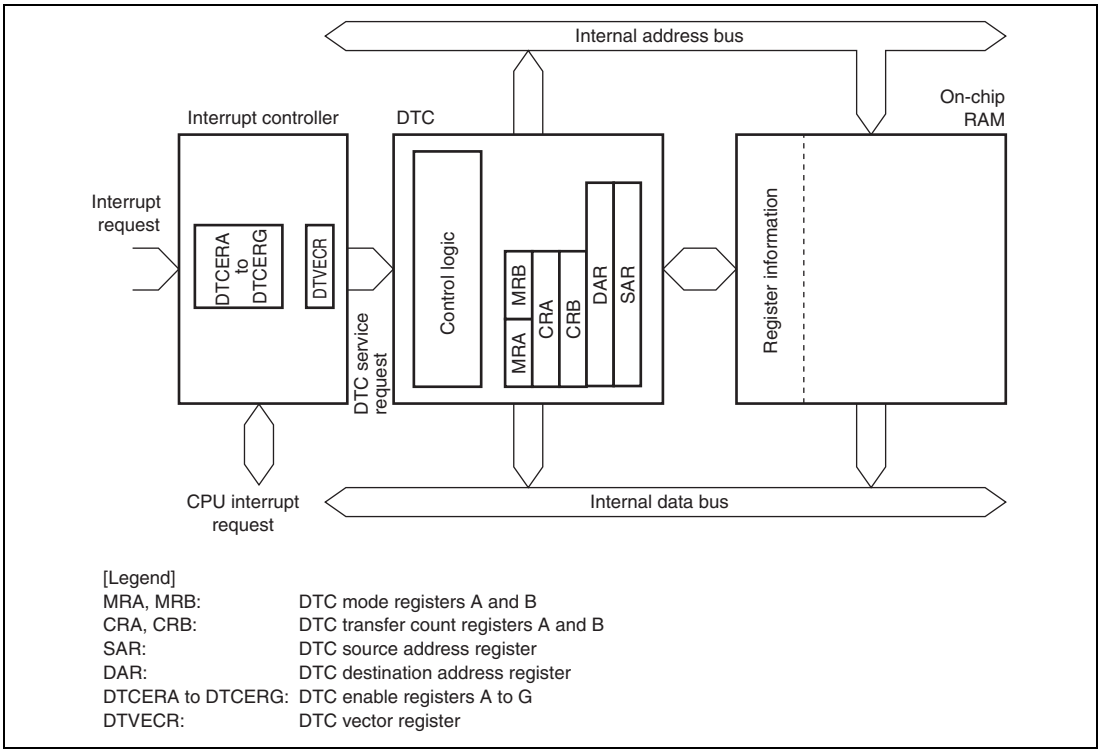


Figure 8.1 Block Diagram of DTC

8.2 Register Descriptions

The DTC has the following registers.

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

These six registers cannot be directly accessed from the CPU.

When activated, the DTC reads a set of register information that is stored in on-chip RAM to the corresponding DTC registers and transfers data. After the data transfer, it writes a set of updated register information back to the RAM.

- DTC enable registers A to G, and I (DTCERA to DTCERG, DTCERI)
- DTC vector register (DTVECR)

8.2.1 DTC Mode Register A (MRA)

MRA is an 8-bit register that selects the DTC operating mode.

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

Bit	Bit Name	Initial Value	R/W	Description
0	Sz	Undefined	—	DTC Data Transfer Size Specifies the size of data to be transferred. 0: Byte-size transfer 1: Word-size transfer

[Legend]
X: Don't care

8.2.2 DTC Mode Register B (MRB)

MRB is an 8-bit register that selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable When this bit is set to 1, a chain transfer will be performed. For details, refer to section 8.5.4, Chain Transfer. In data transfer with CHNE set to 1, determination of the end of the specified number of transfers, clearing of the interrupt source flag, and clearing of DTCER, are not performed.
6	DISEL	Undefined	—	DTC Interrupt Select When this bit is set to 1, a CPU interrupt request is generated every time after the end of a data transfer. When this bit is set to 0, a CPU interrupt request is generated at the time when the specified number of data transfer ends.
5 to 0	—	Undefined	—	Reserved These bits have no effect on DTC operation. Only 0 should be written to these bits.

8.2.3 DTC Source Address Register (SAR)

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

8.2.4 DTC Destination Address Register (DAR)

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

8.2.5 DTC Transfer Count Register A (CRA)

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

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

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

8.2.6 DTC Transfer Count Register B (CRB)

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

8.2.7 DTC Enable Registers A to G, and I (DTCERA to DTCERG)

DTCER is comprised of seven registers; DTCERA to DTCERG, and is a register that specifies DTC activation interrupt sources. The correspondence between interrupt sources and DTCE bits is shown in table 8.1. For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR for reading and writing. If all interrupts are masked, multiple activation sources can be set at one time (only at the initial setting) by writing data after executing a dummy read on the relevant register.

Bit	Bit Name	Initial Value	R/W	Description
7	DTCE7	0	R/W	DTC Activation Enable
6	DTCE6	0	R/W	Setting this bit to 1 specifies a relevant interrupt source as a DTC activation source. [Clearing conditions] • When the DISEL bit in MRB is 1 and the data transfer has ended • When the specified number of transfers have ended These bits are not cleared when the DISEL bit is 0 and the specified number of transfers have not been completed.
5	DTCE5	0	R/W	
4	DTCE4	0	R/W	
3	DTCE3	0	R/W	
2	DTCE2	0	R/W	
1	DTCE1	0	R/W	
0	DTCE0	0	R/W	

8.2.8 DTC Vector Registers A to G (DTVECRA to DTVECRG)

DTVECR is an 8-bit readable/writable register that enables or disables DTC activation by software, and sets a vector number for the software activation interrupt.

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

8.3 Activation Sources

The DTC operates when activated by an interrupt or by a write to DTVECR by software. An interrupt request can be directed to the CPU or DTC, as designated by the corresponding DTCER bit. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source or corresponding DTCER bit is cleared. The activation source flag, in the case of RXI_0, for example, is the RDRF flag of SCI_0.

When an interrupt has been designated a DTC activation source, the existing CPU mask level and interrupt controller priorities have no effect. If there is more than one activation source at the same time, the DTC operates in accordance with the default priorities.

Figure 8.2 shows a block diagram of DTC activation source control. For details, see section 5, Interrupt Controller.

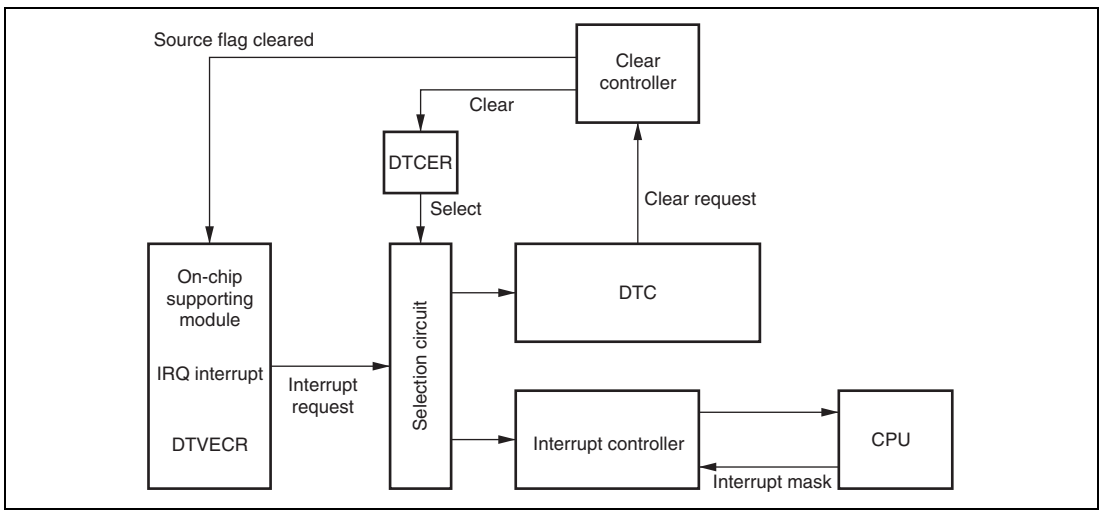


Figure 8.2 Block Diagram of DTC Activation Source Control

8.4 Location of Register Information and DTC Vector Table

Locate the register information in the on-chip RAM (addresses: H'FFEBC0 to H'FFEFBF). Register information should be located at an address that is a multiple of four within the range. Locating the register information in address space is shown in figure 8.3. Locate the MRA, SAR, MRB, DAR, CRA, and CRB registers, in that order, from the start address of the register information.

In the case of chain transfer, register information should be located in consecutive areas and the register information start address should be located at the vector address corresponding to the interrupt source as shown in figure 8.3. The DTC reads the start address of the register information from the vector address set for each activation source, and then reads the register information from that start address.

When the DTC is activated by software, the vector address is obtained from: H'0400 + (DTVECR[6:0] × 2). For example, if DTVECR is H'10, the vector address is H'0420. The configuration of the vector address is the same in both normal and advanced modes, a 2-byte unit being used in both cases. These two bytes specify the lower bits of the register information start address.

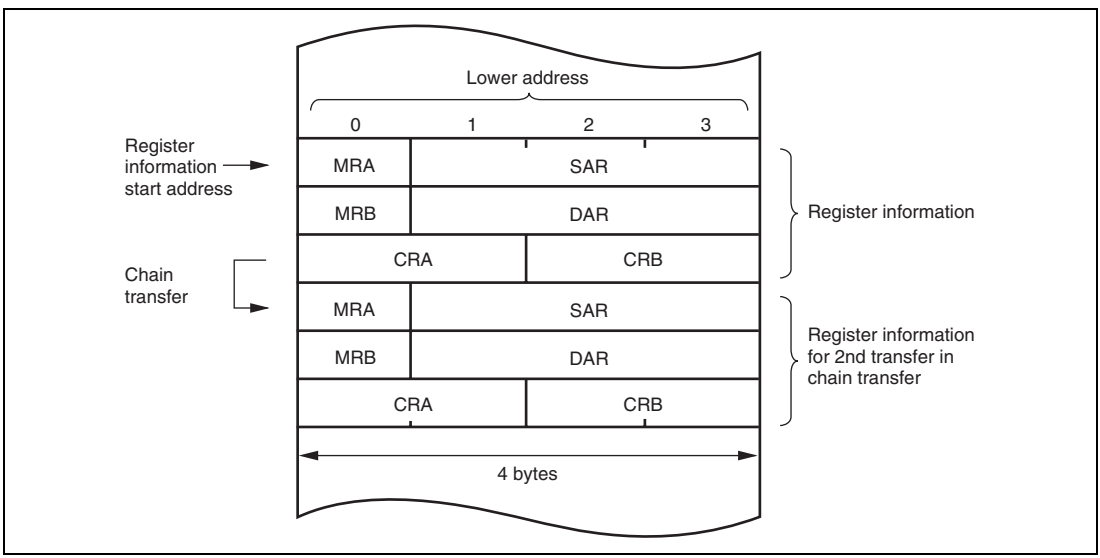


Figure 8.3 Location of DTC Register Information in Address Space

Table 8.1 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Interrupt Source	Origin of Interrupt Source	Vector Number	DTC Vector Address	DTCE*	Priority
Software	Write to DTVECR	DTVECR	H'0400 + (vector number × 2)	—	High
External pin	IRQ0	16	H'0420	DTCEA7	
	IRQ1	17	H'0422	DTCEA6	
	IRQ2	18	H'0424	DTCEA5	
	IRQ3	19	H'0426	DTCEA4	
	IRQ4	20	H'0428	DTCEA3	
	IRQ5	21	H'042A	DTCEA2	
—	Reserved for system use	22	H'042C	DTCEA1	
		23	H'042E	DTCEA0	
A/D counter	ADI (A/D conversion end)	28	H'0438	DTCEB6	
TPU_0	TGIA_0	32	H'0440	DTCEB5	
	TGIB_0	33	H'0442	DTCEB4	
	TGIC_0	34	H'0444	DTCEB3	
	TGID_0	35	H'0446	DTCEB2	
TPU_1	TGIA_1	40	H'0450	DTCEB1	
	TGIB_1	41	H'0452	DTCEB0	
TPU_2	TGIA_2	44	H'0458	DTCEC7	
	TGIB_2	45	H'045A	DTCEC6	
TPU_3	TGIA_3	48	H'0460	DTCEC5	
	TGIB_3	49	H'0462	DTCEC4	
	TGIC_3	50	H'0464	DTCEC3	
	TGID_3	51	H'0466	DTCEC2	
TPU_4	TGIA_4	56	H'0470	DTCEC1	
	TGIB_4	57	H'0472	DTCEC0	
TPU_5	TGIA_5	60	H'0478	DTCED5	
	TGIB_5	61	H'047A	DTCED4	Low

Interrupt Source	Origin of Interrupt Source	Vector Number	DTC Vector Address	DTCE*	Priority
—	Reserved for system use	64	H'0480	DTCED3	High
		65	H'0482	DTCED2	
		68	H'0488	DTCED1	
		69	H'048A	DTCED0	
—	Reserved for system use	72	H'0490	DTCEE7	
		73	H'0492	DTCEE6	
		74	H'0494	DTCEE5	
		75	H'0496	DTCEE4	
SCI_0	RXI_0	81	H'04A2	DTCEE3	
	TXI_0	82	H'04A4	DTCEE2	
SCI_1	RXI_1	85	H'04AA	DTCEE1	
	TXI_1	86	H'04AC	DTCEE0	
SCI_2	RXI_2	89	H'04B2	DTCEF7	
	TXI_2	90	H'04B4	DTCEF6	
—	Reserved for system use	92	H'04B8	DTCEF5	
		93	H'04BA	DTCEF4	
—	Reserved for system use	96	H'04C0	DTCEF3	
		97	H'04C2	DTCEF2	
IIC-0	IICI0	100	H'04C8	DTCEF1	
IIC-1	IICI1	102	H'04CC	DTCEF0	
Motor control PWM	CMI1	104	H'04D0	DTCEG7	
	CMI2	105	H'04D2	DTCEG6	
—	Reserved for system use	106	H'04D4	DTCEG5	
		107	H'04D6	DTCEG4	
—	Reserved for system use	108	H'04D8	DTCEG3	
		109	H'04DA	DTCEG2	Low

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

8.5 Operation

Register information is stored in on-chip RAM. When activated, the DTC reads register information in on-chip RAM and transfers data. After the data transfer, the DTC writes updated register information back to the on-chip RAM.

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

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

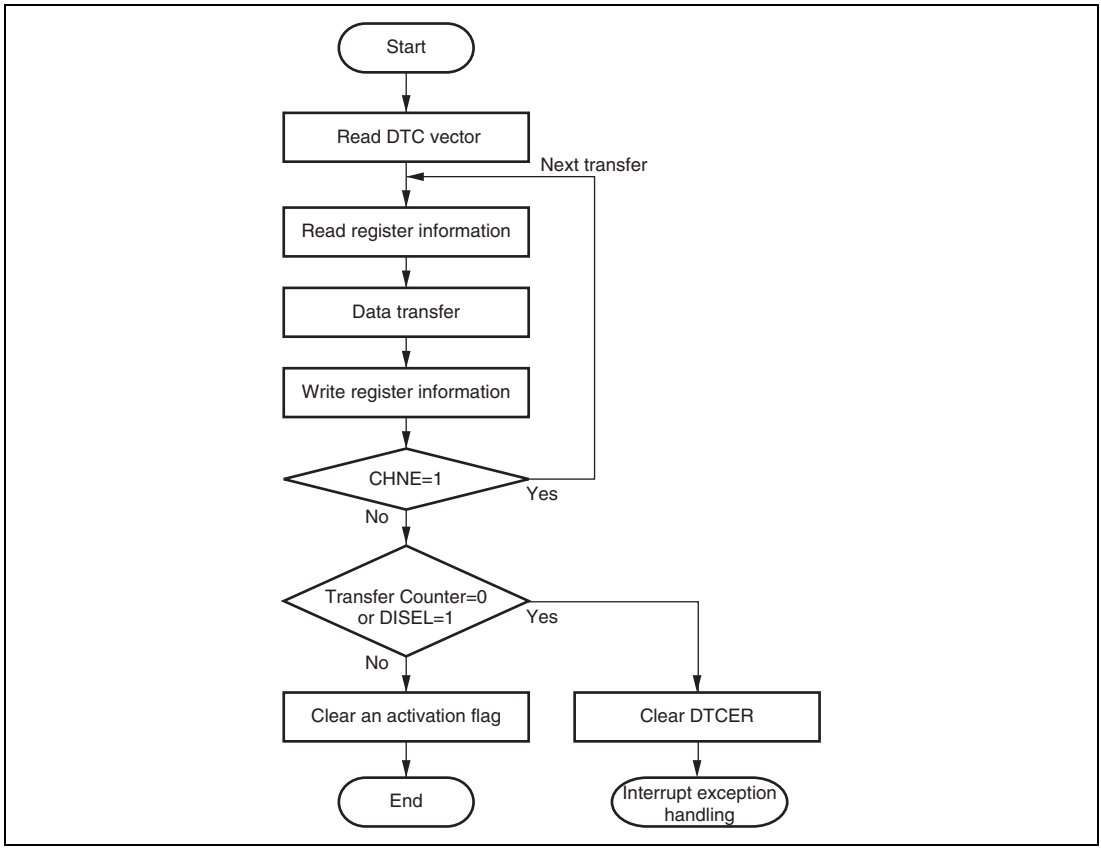


Figure 8.4 Flowchart of DTC Operation

8.5.1 Normal Mode

In normal mode, one operation transfers one byte or one word of data.

Table 8.2 lists the register information in normal mode.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have been completed, a CPU interrupt can be requested.

Table 8.2 Register Information in Normal Mode

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

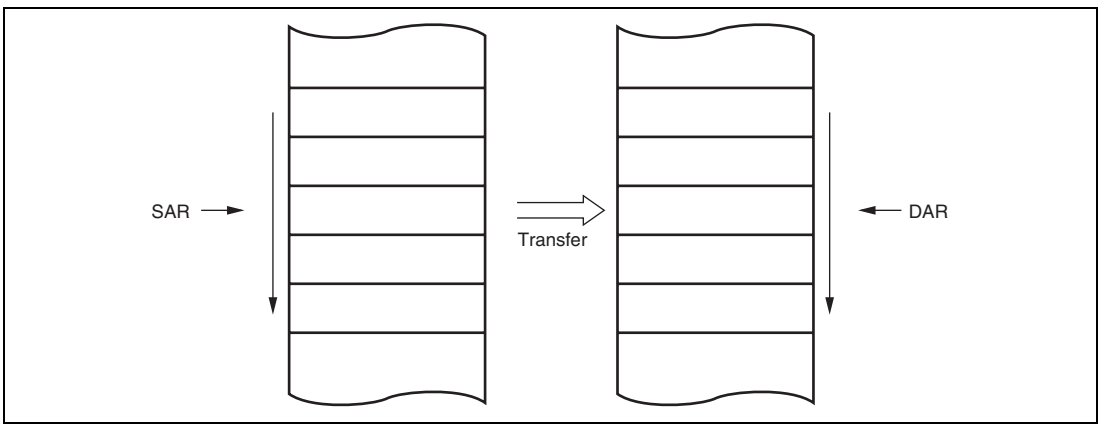


Figure 8.5 Memory Mapping in Normal Mode

8.5.2 Repeat Mode

In repeat mode, one operation transfers one byte or one word of data. Table 8.3 lists the register information in repeat mode.

From 1 to 256 transfers can be specified. Once the specified number of transfers have ended, the initial state of the transfer counter and the address register specified as the repeat area is restored, and transfer is repeated. In repeat mode the transfer counter value does not reach H′00, and therefore CPU interrupts cannot be requested when DISEL = 0.

Table 8.3 Register Information in Repeat Mode

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

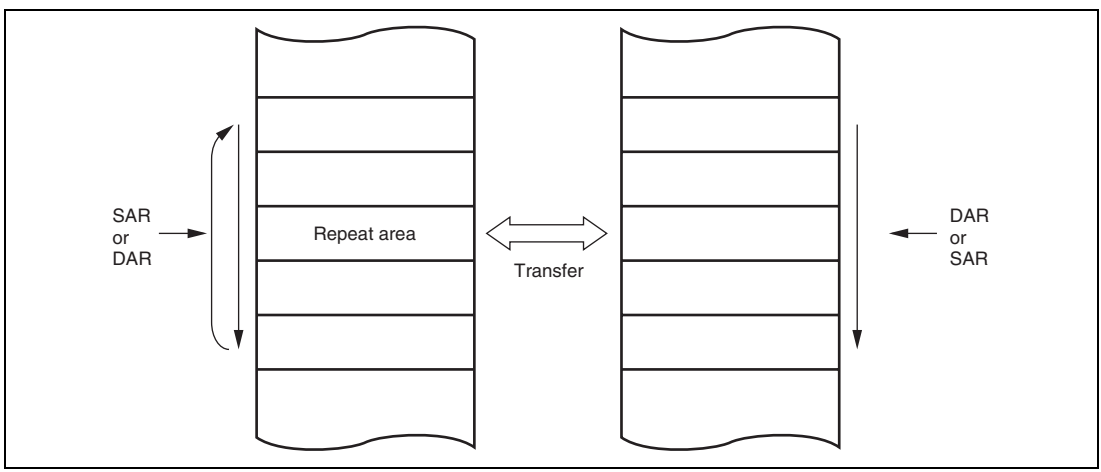


Figure 8.6 Memory Mapping in Repeat Mode

8.5.3 Block Transfer Mode

In block transfer mode, one operation transfers one block of data. Either the transfer source or the transfer destination is designated as a block area. Table 8.4 lists the register information in block transfer mode.

The block size can be between 1 and 256. When the transfer of one block ends, the initial state of the block size counter and the address register specified as the block area is restored. The other address register is then incremented, decremented, or left fixed.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have been completed, a CPU interrupt is requested.

Table 8.4 Register Information in Block Transfer Mode

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

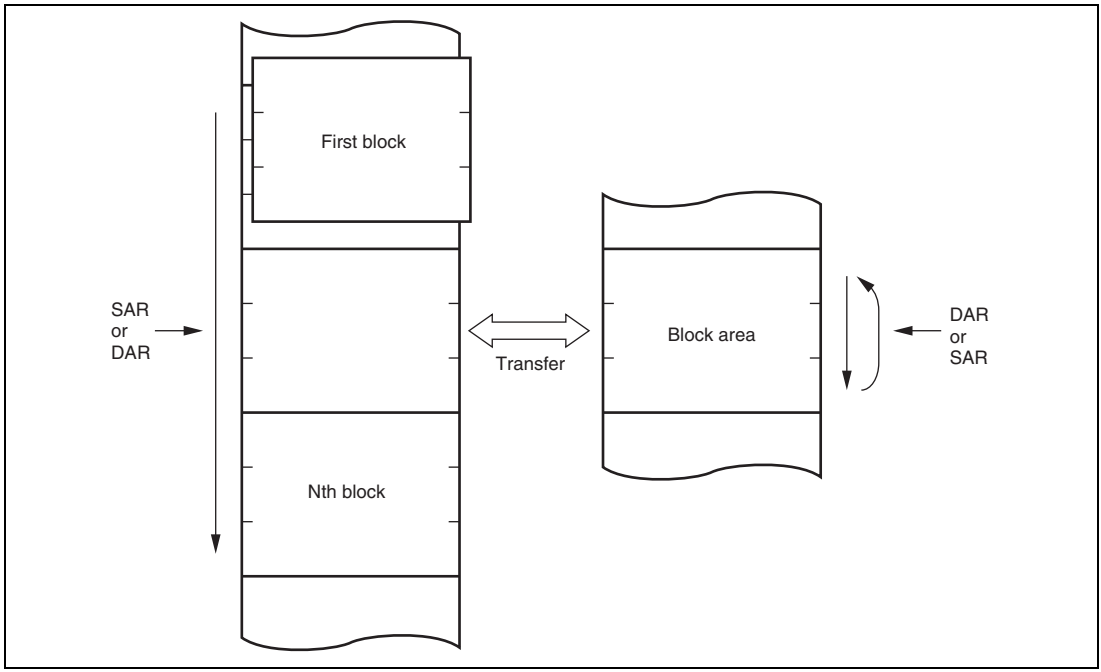


Figure 8.7 Memory Mapping in Block Transfer Mode

8.5.4 Chain Transfer

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

Figure 8.8 shows the outline of the chain transfer operation.

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

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

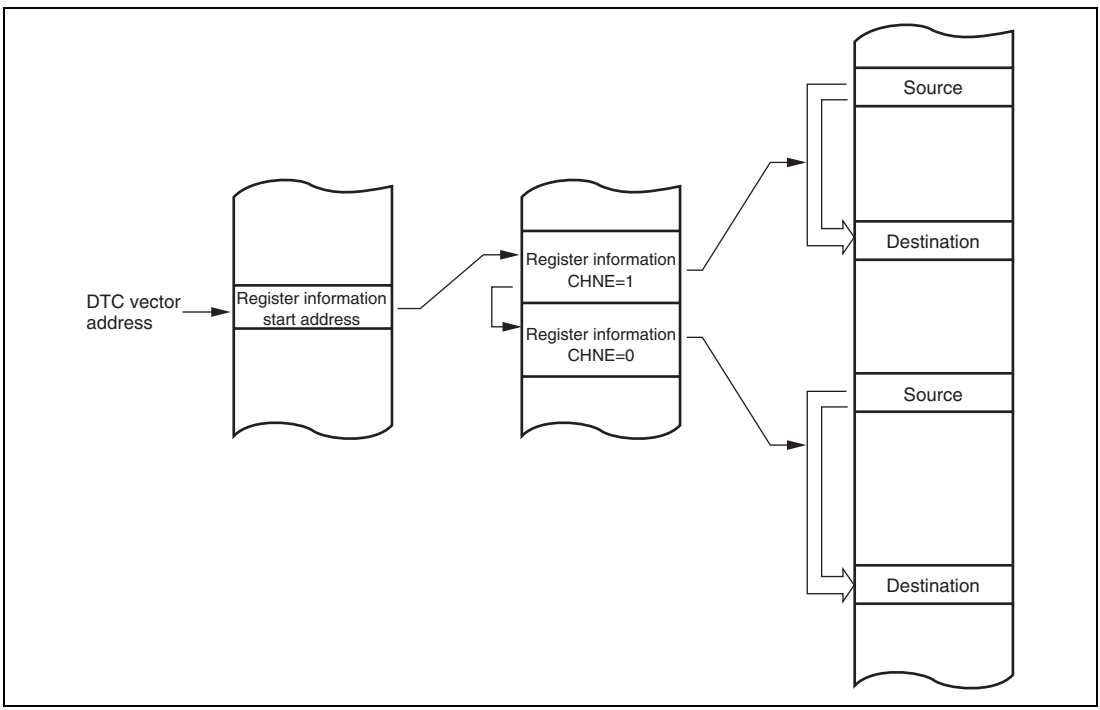


Figure 8.8 Chain Transfer Operation

8.5.5 Interrupts

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

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

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

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

8.5.6 Operation Timing

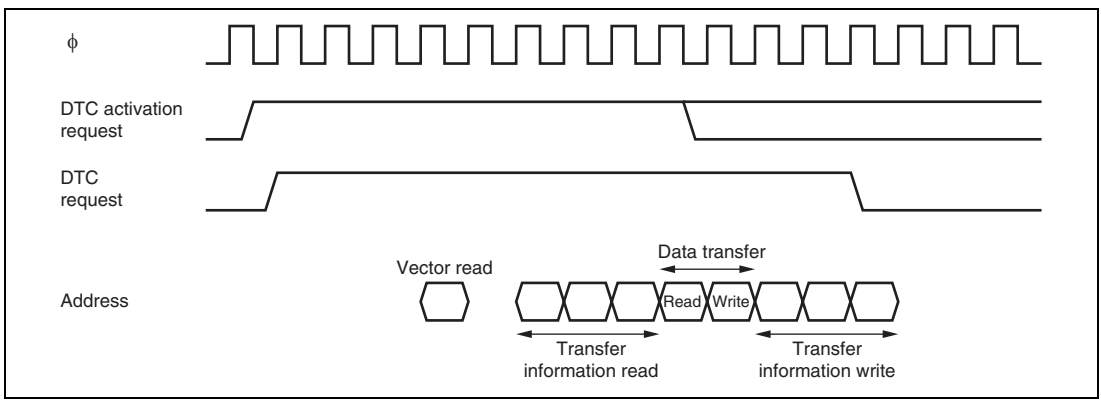


Figure 8.9 DTC Operation Timing (Example in Normal Mode or Repeat Mode)

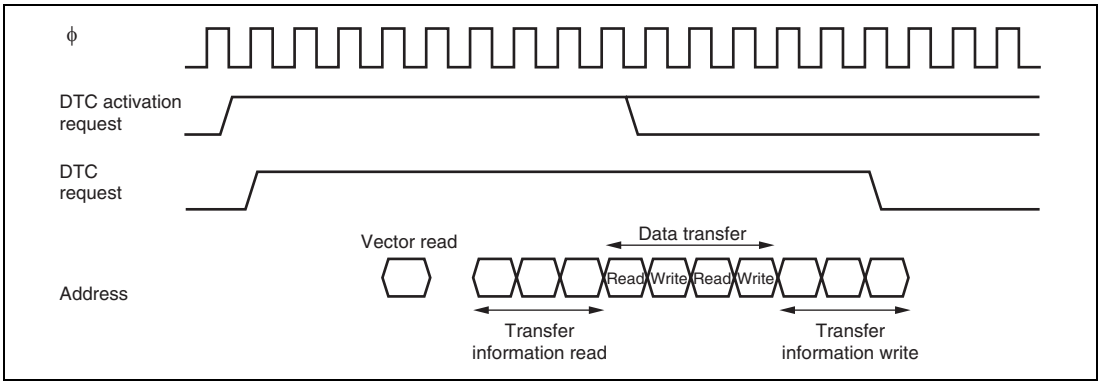


Figure 8.10 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)

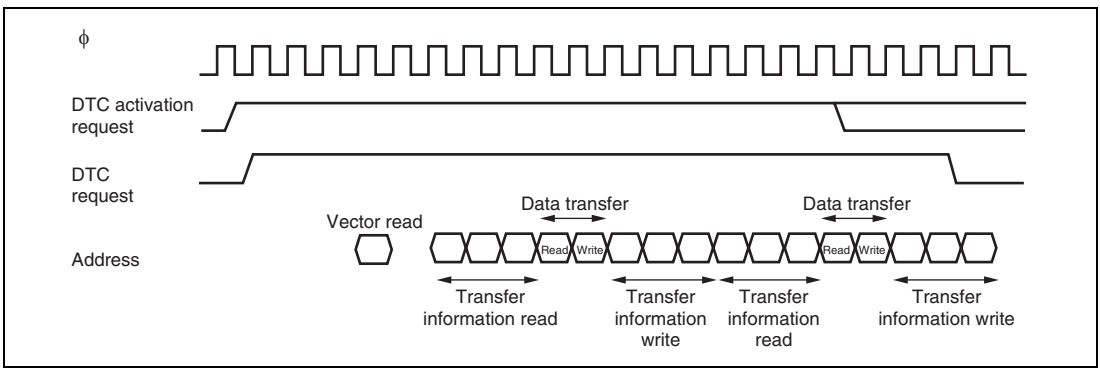


Figure 8.11 DTC Operation Timing (Example of Chain Transfer)

8.5.7 Number of DTC Execution States

Table 8.5 lists execution status for a single DTC data transfer, and table 8.6 shows the number of states required for each execution status.

Table 8.5 DTC Execution Status

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

[Legend]

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

Table 8.6 Number of States Required for Each Execution Status

Object to be Accessed		On-Chip RAM	On-Chip ROM	On-Chip I/O Registers		External Devices*			
Bus width		32	16	8	16	8		16	
Access states		1	1	2	2	2	3	2	3
Execution status	Vector read S_I	—	1	—	—	4	6+2m	2	3+m
	Register information read/write S_J	1	—	—	—	—	—	—	—
	Byte data read S_K	1	1	2	2	2	3+m	2	3+m
	Word data read S_K	1	1	4	2	4	6+2m	2	3+m
	Byte data write S_L	1	1	2	2	2	3+m	2	3+m
	Word data write S_L	1	1	4	2	4	6+2m	2	3+m
	Internal operation S_M	1							

Note: * Not available in this LSI.

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

$$\text{Number of execution states} = I \cdot (1 + S_I) + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$$

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

8.6 Procedures for Using DTC

8.6.1 Activation by Interrupt

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

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

8.6.2 Activation by Software

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

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

8.7 Examples of Use of the DTC

8.7.1 Normal Mode

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

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

8.7.2 Chain Transfer

An example of DTC chain transfer is shown in which pulse output is performed using the PPG. Chain transfer can be used to perform pulse output data transfer and PPG output trigger cycle updating. Repeat mode transfer to the PPG's NDR is performed in the first half of the chain transfer, and normal mode transfer to the TPU's TGR in the second half. This is because clearing of the activation source and interrupt generation at the end of the specified number of transfers are restricted to the second half of the chain transfer (transfer when CHNE = 0).

1. Perform settings for transfer to the PPG's NDR. Set MRA to incrementing source address (SM1 = 1, SM0 = 0), a fixed destination address (DM1 = DM0 = 0), repeat mode (MD1 = 0, MD0 = 1), and word size (Sz = 1). Set the source side as a repeat area (DTS = 1). Set MRB to chain mode (CHNE = 1, DISEL = 0). Set the data table start address in SAR, the NDRH address in DAR, and the data table size in CRAH and CRAL. CRB can be set to any value.
2. Perform settings for transfer to the TPU's TGR. Set MRA to incrementing source address (SM1 = 1, SM0 = 0), a fixed destination address (DM1 = DM0 = 0), normal mode (MD1 = MD0 = 0), and word size (Sz = 1). Set the data table start address in SAR, the TGRA address in DAR, and the data table size in CRA. CRB can be set to any value.
3. Locate the TPU transfer register information consecutively after the NDR transfer register information.
4. Set the start address of the NDR transfer register information to the DTC vector address.
5. Set the bit corresponding to TGIA in DTCER to 1.
6. Set TGRA as an output compare register (output disabled) with TIOR, and enable the TGIA interrupt with TIER.
7. Set the initial output value in PODR, and the next output value in NDR. Set bits in DDR and NDER for which output is to be performed to 1. Using PCR, select the TPU compare match to be used as the output trigger.
8. Set the CST bit in TSTR to 1, and start the TCNT count operation.
9. Each time a TGRA compare match occurs, the next output value is transferred to NDR and the set value of the next output trigger period is transferred to TGRA. The activation source TGFA flag is cleared.
10. When the specified number of transfers are completed (the TPU transfer CRA value is 0), the TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request is sent to the CPU. Termination processing should be performed in the interrupt handling routine.

8.7.3 Software Activation

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

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

8.8 Usage Notes

8.8.1 Module Stop Mode Setting

DTC operation can be disabled or enabled using the module stop control register. The initial setting is for DTC operation to be enabled. Register access is disabled by setting module stop mode. Note that module stop mode cannot be set during DTC being activated. For details, refer to section 21, Power-Down Modes.

8.8.2 On-Chip RAM

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

8.8.3 DTCE Bit Setting

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

Section 9 I/O Ports

This LSI has 10 I/O ports (ports 1 to 3, A to F, H, and J), and two input-only port (ports 4 and 9).

Table 9.1 shows the port functions. The pins of each port also have other functions.

Each I/O port includes a data direction register (DDR) that controls input/output, a data register (DR) that stores output data, and a port register (PORT) used to read the pin states. The input-only ports do not have a DR or DDR register.

Ports A to E have a built-in pull-up MOS function, and in addition to DR and DDR, have a MOS input pull-up control register (PCR) to control the on/off state of MOS input pull-up.

Ports 3, and A to C include an open-drain control register (ODR) that controls the on/off state of the output buffer PMOS.

When ports 10 to 13 and A to F are used as the output pins for expanded bus control signals, they can drive one TTL load plus a 90pF capacitance load. Ports other than those can drive one TTL load and a 30pF capacitance load. All I/O ports can drive Darlington transistors when set to output. Ports 1 and A to C can drive an LED (10 mA sink current).

Table 9.1 Port Functions

Port	Description	Mode 4	Mode 5	Mode 6	Mode 7	Input/Output Type
Port 1	General I/O port also functioning as TPU I/O pins, PPG and address output pins, and interrupt input pins	P17/PO15/TIOCB2/TCLKD			P17/PO15/TIOCB2/ TCLKD	
		P16/PO14/TIOCA2/$\overline{\text{IRQ1}}$			P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	
		P15/PO13/TIOCB1/TCLKC			P15/PO13/TIOCB1/ TCLKC	
		P14/PO12/TIOCA1/$\overline{\text{IRQ0}}$			P14/PO12/TIOCA1/$\overline{\text{IRQ0}}$	
		P13/PO11/TIOCD0/TCLKB/A23			P13/PO11/TIOCD0/ TCLKB	
		P12/PO10/TIOCC0/TCLKA/A22			P12/O10/TIOCC0/ TCLKA	
		P11/PO9/TIOCB0/A21			P11/PO9/TIOCB0	
		P10/PO8/TIOCA0/A20			P10/PO8/TIOCA0	
Port 3	General I/O port also functioning as SCI and IIC I/O pins, and interrupt input pins	P35/SCK1/SCL0/$\overline{\text{IRQ5}}$				Open-drain output
		P34/RxD1/SDA0				
		P33/TxD1/SCL1				
		P32/SCK0/SDA1/$\overline{\text{IRQ4}}$				
		P31/RxD0				
		P30/TxD0				
Port 4	General input port also functioning as A/D analog input pins and D/A analog output pins	P47/AN7/DA1				
		P46/AN6/DA0				
		P45/AN5				
		P44/AN4				
		P43/AN3				
		P42/AN2				
		P41/AN1				
		P40/AN0				

Port	Description	Mode 4	Mode 5	Mode 6	Mode 7	Input/Output Type
Port 9	General input port also functioning as analog input pins	P97/AN15				
		P96/AN14				
		P95/AN13				
		P94/AN12				
		P93/AN11				
		P92/AN10				
		P91/AN9				
		P90/AN8				
Port A	General I/O port also functioning as SCI I/O pins and address output pins	PA3/A19/SCK2			PA3/SCK2	Built-in pull-up MOS Open-drain output
		PA2/A18/RxD2			PA2/RxD2	
		PA1/A17/TxD2			PA1/TxD2	
		PA0/A16			PA0	
Port B	General I/O port also functioning as TPU I/O pins and address output pins	PB7/A15/TIOCB5			PB7/TIOCB5	Built-in pull-up MOS Open-drain output
		PB6/A14/TIOCA5			PB6/TIOCA5	
		PB5/A13/TIOCB4			PB5/TIOCB4	
		PB4/A12/TIOCA4			PB4/TIOCA4	
		PB3/A11/TIOCD3			PB3/TIOCD3	
		PB2/A10/TIOCC3			PB2/TIOCC3	
		PB1/A9/TIOCB3			PB1/TIOCB3	
		PB0/A8/TOICA3			PB0/TOICA3	
Port C	General I/O port also functioning as address output pins	PC7/A7			PC7	Built-in pull-up MOS Open-drain output
		PC6/A6			PC6	
		PC5/A5			PC5	
		PC4/A4			PC4	
		PC3/A3			PC3	
		PC2/A2			PC2	
		PC1/A1			PC1	
		PC0/A0			PC0	

Port	Description	Mode 4	Mode 5	Mode 6	Mode 7	Input/Output Type
Port D	General I/O port also functioning as data I/O pins	D15			PD7	Built-in pull-up MOS
		D14			PD6	
		D13			PD5	
		D12			PD4	
		D11			PD3	
		D10			PD2	
		D9			PD1	
		D8			PD0	
Port E	General I/O port also functioning as data I/O pins	PE7/D7			PE7	Built-in pull-up MOS
		PE6/D6			PE6	
		PE5/D5			PE5	
		PE4/D4			PE4	
		PE3/D3			PE3	
		PE2/D2			PE2	
		PE1/D1			PE1	
		PE0/D0			PE0	
Port F	General I/O port also functioning as ϕ output pin, bus control I/O pins, and interrupt input pins	PF7/ϕ			PF7/ϕ	
		PF6/$\overline{AS}$			PF6	
		PF5/$\overline{RD}$			PF5	
		PF4/$\overline{HWR}$			PF4	
		PF3/$\overline{LWR}$/$\overline{ADTRG}$/$\overline{IRQ3}$			PF3/$\overline{ADTRG}$/$\overline{IRQ3}$	
		PF0/$\overline{IRQ2}$			PF0/$\overline{IRQ2}$	

Port	Description	Mode 4	Mode 5	Mode 6	Mode 7	Input/Output Type
Port H	General I/O port also functioning as PWM output pins	PH7/PWM1H				
		PH6/PWM1G				
		PH5/PWM1F				
		PH4/PWM1E				
		PH3/PWM1D				
		PH2/PWM1C				
		PH1/PWM1B				
		PH0/PWM1A				
Port J	General I/O port also functioning as PWM output pins	PJ7/PWM2H				
		PJ6/PWM2G				
		PJ5/PWM2F				
		PJ4/PWM2E				
		PJ3/PWM2D				
		PJ2/PWM2C				
		PJ1/PWM2B				
		PJ0/PWM2A				

9.1 Port 1

Port 1 is an 8-bit I/O port. Port 1 pins also function as PPG output pins, TPU I/O pins, address output pins, and external interrupt pins.

- Port 1 data direction register (P1DDR)
- Port 1 data register (P1DR)
- Port 1 register (PORT1)

9.1.1 Port 1 Data Direction Register (P1DDR)

P1DDR is a write-only register, the individual bits of which specify input or output for the pins of port 1. P1DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	Setting a P1DDR bit to 1 makes the corresponding port 1 pin an output pin, while clearing the bit to 0 makes the pin an input pin.
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	

9.1.2 Port 1 Data Register (P1DR)

P1DR stores output data for the port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose output port.
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	

9.1.3 Port 1 Register (PORT1)

PORT1 shows the pin states. It cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P17	—*	R	If this register read is performed while P1DDR bits are set to 1, the P1DR values are read. If this register read is performed while P1DDR bits are cleared to 0, the pin states are read.
6	P16	—*	R	
5	P15	—*	R	
4	P14	—*	R	
3	P13	—*	R	
2	P12	—*	R	
1	P11	—*	R	
0	P10	—*	R	

Note: * Determined by state of pins P17 to P10.

9.1.4 Pin Functions

The correspondence between the register specification and the pin functions is shown below.

- P17/PO15/TIOCB2/TCLKD

The pin function is switched as shown below according to the combination of the TPU channel 2 setting (by bits MD3 to MD0 in TMDR_2, bits IOB3 to IOB0 in TIOR_2, and bits CCLR1 and CCLR0 in TCR_2), bits TPSC2 to TPSC0 in TCR_0 and TCR_5, bit NDER15 in NDERH, and bit P17DDR.

TPU Channel 2 Setting	Table below (1)	Table below (2)		
P17DDR	—	0	1	1
NDER15	—	—	0	1
Pin function	TIOCB2 output	P17 input	P17 output	PO15 output
		TIOCB2 input*1		
	TCLKD input*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. TIOCB2 input when MD3 to MD0 = B'0000 or B'01xx, and IOB3 = 1.

2. TCLKD input when the setting for either TCR_0 or TCR_5 is: TPSC2 to TPSC0 = B'111. TCLKD input when channels 2 and 4 are set to phase counting mode.

- P16/PO14/TIOCA2/$\overline{\text{IRQ1}}$

The pin function is switched as shown below according to the combination of the TPU channel 2 setting (by bits MD3 to MD0 in TMDR_2, bits IOA3 to IOA0 in TIOR_2, and bits CCLR1 and CCLR0 in TCR_2), bit NDER14 in NDERH, and bit P16DDR.

TPU channel 2 setting	Table below (1)	Table below (2)		
P16DDR	—	0	1	1
NDER14	—	—	0	1
Pin function	TIOCA2 output	P16 input	P16 output	PO14 output
		TIOCA2 input*1		
	$\overline{\text{IRQ1}}$ input			

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. TIOCA2 input when MD3 to MD0 = B'0000 or B'01xx, and IOA3 = 1.

2. TIOCB2 output is disabled.

- P15/PO13/TIOCB1/TCLKC

The pin function is switched as shown below according to the combination of the TPU channel 1 setting (by bits MD3 to MD0 in TMDR_1, bits IOB3 to IOB0 in TIOR_1, and bits CCLR1 and CCLR0 in TCR_1), bits TPSC2 to TPSC0 in TCR_0, TCR_2, TCR_4, and TCR_5, bit NDER13 in NDERH, and bit P15DDR.

TPU channel 1 setting	Table below (1)	Table below (2)		
P15DDR	—	0	1	1
NDER13	—	—	0	1
Pin function	TIOCB1 output	P15 input	P15 output	PO13 output
		TIOCB1 input*1		
	TCLKC input*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. TIOCB1 input when MD3 to MD0 = B'0000 or B'01xx, and IOB3 to IOB0 = B'10xx.

2. TCLKC input when the setting for either TCR_0 or TCR_2 is: TPSC2 to TPSC0 = B'110; or when the setting for either TCR_4 or TCR_5 is TPSC2 to TPSC0 = B'101. TCLKC input when channels 2 and 4 are set to phase counting mode.

- P14/PO12/TIOCA1/$\overline{\text{IRQ0}}$

The pin function is switched as shown below according to the combination of the TPU channel 1 setting (by bits MD3 to MD0 in TMDR_1, bits IOA3 to IOA0 in TIOR_1, and bits CCLR1 and CCLR0 in TCR_1), bit NDER12 in NDERH, and bit P14DDR.

TPU channel 1 setting	Table below (1)	Table below (2)		
P14DDR	—	0	1	1
NDER12	—	—	0	1
Pin function	TIOCA1 output	P14 input	P14 output	PO12 output
		TIOCA1 input*1		
	$\overline{\text{IRQ0}}$ input			

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. TIOCA1 input when MD3 to MD0 = B'0000 or B'01xx, and IOA3 to IOA0 = B'10xx.
2. TIOCB1 output is disabled.

- P13/PO11/TIOCD0/TCLKB

The pin function is switched as shown below according to the combination of the operating mode, bits MD3 to MD0 in TMDR0, bits IOD3 to IOD0 in TIORL_0, the TPU channel 0 setting (by bits CCLR2 to CCLR0 in TCR0), bits TPSC2 to TPSC0 in TCR_0 to TCR_2, bits AE3 to AE0 in PFCR, and bit P13DDR.

Operating mode	Modes 4 to 6				
AE3 to AE0	B'0000 to B'1110				B'1111
TPU channel 0 setting	Table below (1)	Table below (2)			—
P13DDR	—	0	1	1	—
NDER11	—	—	0	1	—
Pin function	TIOCD0 output	P13 input	P13 output	PO11 output	—
		TIOCD0 input*1			A23 input
	TCLKB input*2				

Operating mode	Modes 7			
AE3 to AE0	—			
TPU channel 0 setting	Table below (1)	Table below (2)		
P13DDR	—	0	1	1
NDER11	—	—	0	1
Pin function	TIOCD0 output	P13 input	P13 output	PO11 output
		TIOCD0 input*1		
	TCLKB input*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 toB'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. TIOCD0 input when MD3 to MD0 = B'0000, and IOD3 to IOD0 = B'10xx.

2. TCLKB input when the setting for TCR_0 to TCR_2 is: TPSC2 to TPSC0 = B'101. TCLKB input when channels 1 and 5 are set to phase counting mode.

- P12/PO10/TIOCC0/TCLKA/A22

The pin function is switched as shown below according to the combination of the operating mode, bits MD3 to MD0 in TMDR0, bits IOC3 to IOC0 in TIORL_0, the TPU channel 0 setting (by bits CCLR2 to CCLR0 in TCR_0), bits TPSC2 to TPSC0 in TCR0 to TCR5, bits AE3 to AE0 in PFCR, and bit P12DDR.

Operating mode	Modes 4 to 6				
AE3 toAE0	B'0000 to B'1110				B'1111
TPU channel 0 setting	Table below (1)	Table below (2)			—
P12DDR	—	0	1	1	—
NDER10	—	—	0	1	—
Pin function	TIOCC0 output	P12 input	P12 output	PO10 output	—
		TIOCC0 input*1			A22 output
	TCLKA input*2				

Operating mode	Modes 7			
AE3 to AE0	—			
TPU channel 0 setting	Table below (1)	Table below (2)		
P12DDR	—	0	1	1
NDER10	—	—	0	1
Pin function	TIOCC0 output	P12 input	P12 output	PO10 output
		TIOCC0 input*1		
	TCLKA input*2			

TPU Channel0 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		
CCLR2 to CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM mode 1 output*3	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. TIOCC0 input when MD3 to MD0 = B'0000, and IOC3 to IOC0 = B'10xx.

2. TCLKA input when the setting for TCR_0 to TCR_5 is: TPSC2 to TPSC0 = B'100.
   TCLKA input when channels 1 and 5 are set to phase counting mode.

3. TIOCD0 output is disabled.
   When BFA = 1 or BFB = 1 in TMDR_0, output is disabled and setting (2) applies.

- P11/PO9/TIOCB0/A21

The pin function is switched as shown below according to the combination of the operating mode, bits MD3 to MD0 in TMDR0, the TPU channel 0 setting (by bits IOB3 to IOB0 in TIORH_0, bits AE3 to AE0 in PFCR, bit NDER9 in NDERH and bit P11DDR.

Operating mode	Modes 4 to 6				
AE3 toAE0	B'0000 to B'1101				B'1111 to B'1111
TPU channel 0 setting	Table below (1)	Table below (2)			—
P11DDR	—	0	1	1	—
NDER9	—	—	0	1	—
Pin function	TIOCB0 output	P11 input	P11 output	PO9 output	A21 output
		TIOCB0 input*1			

Operating mode	Modes 7			
AE3 to AE0	—			
TPU channel 0 setting	Table below (1)	Table below (2)		
P11DDR	—	0	1	1
NDER9	—	—	0	1
Pin function	TIOCB0 output	P11 input	P11 output	PO9 output
		TIOCB0 input*1		

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 toB'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: * TIOCB0 input when MD3 to MD0 = B'0000, and IOB3 to IOB0 = B'10xx.

- P10/PO8/TIOCA0/A20

The pin function is switched as shown below according to the combination of the operating mode, bits MD3 to MD0 in TMDR0, bits IOA3 to IOA0 in TIORH_0, the TPU channel 0 setting (by bits CCLR2 to CCLR0 in TCR0), bits AE3 to AE0 in PFCR, bit NDER8 in NDERH and bit P10DDR.

Operating mode	Modes 4 to 6				
AE3 toAE0	B'0000 to B'1100				B'1101 to B'1111
TPU channel 0 setting	Table below (1)	Table below (2)			—
P10DDR	—	0	1	1	—
NDER8	—	—	0	1	—
Pin function	TIOCA0 output	P10 input	P10 output	PO8 output	A20 output
		TIOCA0 input*[1]			

Operating mode	Mode 7			
AE3 to AE0	—			
TPU channel 0 setting	Table below (1)	Table below (2)		
P10DDR	—	0	1	1
NDER8	—	—	0	1
Pin function	TIOCA0 output	P10 input	P10 output	PO8 output
		TIOCA0 input*[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 mode 1 output*2	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. TIOCA0 input when MD3 to MD0 = B'0000, and IOA3 to IOA0 = B'10xx.

2. TIOCB0 output is disabled.

9.2 Port 3

Port 3 is a 6-bit I/O port. Port 3 pins also function as IIC I/O pins, SCI I/O pins and external interrupt input pins. All of the port 3 pin functions have the same operating mode.

- Port 3 data direction register (P3DDR)
- Port 3 data register (P3DR)
- Port 3 register (PORT3)
- Port 3 open-drain control register (PORT3)

9.2.1 Port 3 Data Direction Register (P3DDR)

P3DDR is a write-only register, the individual bits of which specify input or output for the pins of port 3. P3DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Setting a P3DDR bit to 1 makes the corresponding port 3 pin an output pin, while clearing the bit to 0 makes the pin an input pin.
6	—	Undefined	—	
5	P35DDR	0	W	
4	P34DDR	0	W	
3	P33DDR	0	W	
2	P32DDR	0	W	
1	P31DDR	0	W	
0	P30DDR	0	W	

9.2.2 Port 3 Data Register (P3DR)

P3DR stores output data for the port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	An output data for a pin is stored when the pin function is specified to a general purpose output port.
6	—	Undefined	—	
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	

9.2.3 Port 3 Register (PORT3)

PORT3 shows the pin states. It cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	If this register read is performed while P3DDR bits are set to 1, the P3DR values are read. If this register read is performed while P3DDR bits are cleared to 0, the pin states are read.
6	—	Undefined	—	
5	P35	—*	R	
4	P34	—*	R	
3	P33	—*	R	
2	P32	—*	R	
1	P31	—*	R	
0	P30	—*	R	

Note: * Determined by state of pins P35 to P30.

9.2.4 Port 3 Open-Drain Control Register (P3ODR)

P3ODR controls output of port 3.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	By setting P3ODR to 1, the port 3 pins become an NMOS open drain output, and when cleared to 0 they become CMOS output.
6	—	Undefined	—	
5	P35ODR	0	R	
4	P34ODR	0	R	
3	P33ODR	0	R	
2	P32ODR	0	R	
1	P31ODR	0	R	
0	P30ODR	0	R	

9.2.5 Pin Functions

The correspondence between the register specification and the pin functions is shown below.

- P35/SCK1/SCL0/$\overline{\text{IRQ5}}$

The pin function is switched as shown below according to the combination of the ICE bit in ICCR0 of IIC_0, the C/$\overline{\text{A}}$ bit in SMR of SCI_1, the CKE0 and CKE1 bits in SCR, and the P35DDR bit.

ICE	0					1
CKE1	0				1	0
C/$\overline{\text{A}}$	0			1	—	0
CKE0	0		1	—	—	0
P35DDR	0	1	—	—	—	—
Pin function	P35 input	P35 output	SCK1 output*	SCK1 output*	SCK1 input	SCLO I/On
	$\overline{\text{IRQ5}}$ input					

Note: * When P35ODR = 1, it becomes NMOS open drain output.

- P34/RxD1/SDA0

The pin function is switched as shown below according to the combination of the ICE bit in ICCR0 of ICC_0, the RE bit in SCR of SCI_1 and P34DDR bit.

ICE	0			1
RE	0		1	—
P34DDR	0	1	—	—
Pin function	P34 input	P34 output*	RxD1 input	SDA0 I/O

Note: * When P34ODR = 1, it becomes NMOS open drain output.

- P33/TxD1/SCL1

The pin function is switched as shown below according to the combination of the ICE bit in ICCR1 of ICC_1, the TE bit in SCR of SCI_1 and P33DDR bit.

ICE	0			1
TE	0		1	—
P33DDR	0	1	—	—
Pin function	P33 input	P33 output*	TxD1 output	SCL1 I/O

Note: * When P33ODR = 1, it becomes NMOS open drain output.

- P32/SCK0/SDA1/$\overline{\text{IRQ4}}$

The pin function is switched as shown below according to the combination of the ICE bit in ICCR1 of ICC_1, the C/$\overline{\text{A}}$ bit in SMR of SCI_0, the CKE0 and CKE1 bits in SCR, and the P32DDR bit.

ICE	0					1
CKE1	0				1	0
C/$\overline{\text{A}}$	0			1	—	0
CKE0	0		1	—	—	0
P32DDR	0	1	—	—	—	—
Pin function	P32 input	P32 output	SCK0 output*	SCK0 output pin*	SCK0 input	SDA1 I/O
	$\overline{\text{IRQ4}}$ input					

Note: * When P32ODR = 1, it becomes NMOS open drain output.

- P31/RxD0

The pin function is switched as shown below according to the combination of the RE bit in SCR of SCI_0 and P31DDR bit.

RE	0		1
P31DDR	0	1	—
Pin function	P31 input	P31 output*	RxD0 input

Note: * When P31ODR = 1, it becomes NMOS open drain output.

- P30/TxD0

The pin function is switched as shown below according to the combination of the TE bit in SCR of SCI_0 and P30DDR bit.

TE	0		1
P30DDR	0	1	—
Pin function	P30 input	P30 output*	TxD0 output pin

Note: * When P30ODR = 1, it becomes NMOS open drain output.

9.3 Port 4

Port 4 is an 8-bit input-only port. Port 4 pins also function as A/D converter analog input pins and D/A converter analog output pins. Port 4 pin functions are the same in all operating modes.

- Port 4 Register (PORT4)

9.3.1 Port 4 Register (PORT4)

PORT4 is a read-only register that shows the pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	P47	—*	R	The pin states are always read when PORT4 read is performed.
6	P46	—*	R	
5	P45	—*	R	
4	P44	—*	R	
3	P43	—*	R	
2	P42	—*	R	
1	P41	—*	R	
0	P40	—*	R	

Note: * Determined by state of pins P47 to P40.

9.3.2 Pin Functions

Port 4 also functions as A/D converter analog input pins and D/A converter analog output pins.

9.4 Port 9

Port 9 is a 4-bit input-only port. Port 9 pins also function as A/D converter analog input pins. Port 9 pin functions are the same in all operating modes.

- Port 9 Register (PORT9)

9.4.1 Port 9 Register (PORT9)

PORT9 is a read-only register that shows the pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—*	—	The pin states are always read when PORT9 read is performed.
6	—	—*	—	
5	—	—*	—	
4	—	—*	—	
3	P93	—*	R	
2	P92	—*	R	
1	P91	—*	R	
0	P90	—*	R	

Note: * Determined by state of pins P93 to P90.

9.4.2 Pin Functions

Port 9 also functions as A/D converter analog input pins.

9.5 Port A

Port A is a 4-bit I/O port. Port A also functions as address bus output pins and SCI-3 I/O pins. The pin functions change according to the operating mode.

Port A has a built-in MOS input pull-up function that can be controlled by software.

- Port A data direction register (PADDR)
- Port A data register (PADR)
- Port A register (PORTA)
- Port A pull-up MOS control register (PAPCR)
- Port A open-drain control register (PAODR)

9.5.1 Port A Data Direction Register (PADDR)

PADDR is a register, the individual bits of which specify input or output for the pins of port A. PADDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Mode 7:
6	—	Undefined	—	Setting a PADDR bit to 1 makes the corresponding port A pin an output port, and clearing a bit to 0 makes the corresponding pin an input port.
5	—	Undefined	—	
4	—	Undefined	—	
3	PA3DDR	0	W	Modes 4 to 6:
2	PA2DDR	0	W	The port A pins function as address outputs as specified by the setting of bits AE3 to AE0 of PFCR, regardless of the values of bits PA7DDR to PA0DDR. Also, when the pins are not used as address outputs, setting a PADDR bit to 1 makes the corresponding port A pin an output port, and clearing a bit to 0 makes the corresponding pin an input port.
1	PA1DDR	0	W	
0	PA0DDR	0	W	

9.5.2 Port A Data Register (PADR)

PADR stores output data for the port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	An output data for a pin is stored when the pin function is specified to a general purpose output port.
6	—	Undefined	—	
5	—	Undefined	—	
4	—	Undefined	—	
3	PA3DR	0	R/W	
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

9.5.3 Port A Register (PORTA)

PORTA shows the pin states. It cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	If this register read is performed while PADDR bits are set to 1, the PADR values are read. If this register read is performed while PADDR bits are cleared to 0, the pin states are read.
6	—	Undefined	—	
5	—	Undefined	—	
4	—	Undefined	—	
3	PA3	—*	R	
2	PA2	—*	R	
1	PA1	—*	R	
0	PA0	—*	R	

Note: * Determined by state of pins PA3 to PA0.

9.5.4 Port A Pull-Up MOS Control Register (PAPCR)

PAPCR controls the MOS input pull-up function incorporated into port A on an individual bit basis.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	In modes 4 to 6, if a pin is in the input state in accordance with the settings of SCMR, SMR, SCR, and PADDR in PFCR and SCI_2, setting the corresponding PAPCR bit to 1 turns on the MOS input pull-up for that pin.
6	—	Undefined	—	
5	—	Undefined	—	
4	—	Undefined	—	
3	PA3PCR	0	R/W	In mode 7, if a pin is in the input state in accordance with the setting of SCMR, SMR, SCR, and PADDR in SCI_2, setting the corresponding PAPCR bit to 1 turns on the MOS input pull-up for that pin.
2	PA2PCR	0	R/W	
1	PA1PCR	0	R/W	
0	PA0PCR	0	R/W	

9.5.5 Port A Open-Drain Control Register (PAODR)

PAODR controls output of port A.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	When pins are not address outputs in accordance with the setting of bits AE3 to AE0 in PFCR, setting a PAODR bit makes the corresponding port A pin an NMOS open-drain output, while clearing the bit to 0 makes the pin a CMOS output.
6	—	Undefined	—	
5	—	Undefined	—	
4	—	Undefined	—	
3	PA3ODR	0	R/W	
2	PA2ODR	0	R/W	
1	PA1ODR	0	R/W	
0	PA0ODR	0	R/W	

9.5.6 Pin Functions

Port A pins also function as SCI2 I/O pins and address bus output pins.

- PA3/A19/SCK2

The pin function is switched as shown below according to the combinations of the operating mode, bits AE3 to AE0 in PFCR, the C/$\overline{A}$ bit in SMR of SCI2, the CKE0 and CKE1 bits in SCR, and the PA3DDR bit.

Operating mode	Modes 4 to 6					
AE3 to AE0	B'0000 to B'1011					B'1100 to B'1111
CKE1	0				1	—
C/$\overline{A}$	0			1	—	—
CKE0	0		1	—	—	—
PA3DDR	0	1	—	—	—	—
Pin function	PA3 input	PA3 output	SCK2 output	SCK2 output	SCK2 input	A19 output

Operating mode	Mode 7				
CKE1	0				1
C/$\overline{A}$	0			1	—
CKE0	0		1	—	—
PA3DDR	0	1	—	—	—
Pin function	PA3 input	PA3 output	SCK2 output	SCK2 output	SCK2 input

- PA2/A18/RxD2

The pin function is switched as shown below according to the combinations of the operating mode, bits AE3 to AE0 in PFCR, the RE bit in SCR of SCI2 and the bit PA2DDR.

Operating mode	Modes 4 to 6			
AE3 to AE0	B'0000to B'1010			B'1011 to B'1111
RE	0		1	—
PA2DDR	0	1	—	—
Pin function	PA2 input	PA2 output	RxD2 input	A18 output

Operating mode	Mode 7		
RE	0		1
PA2DDR	0	1	—
Pin function	PA2 input	PA2 output	RxD2 input

- PA1/A17/TxD2

The pin function is switched as shown below according to the combinations of the operating mode, bits AE3 to AE0 in PFCR, the TE bit in SCR of SCI2 and the bit PA1DDR.

Operating mode	Modes 4 to 6			
AE3 to AE0	B'0000to B'1001			B'1010 to B'1111
TE	0		1	—
PA1DDR	0	1	—	—
Pin function	PA1 input	PA1 output	TxD2 input	A17 output

Operating mode	Mode 7		
TE	0		1
PA1DDR	0	1	—
Pin function	PA1 input	PA1 output	TxD2 input

- PA0/A16

The pin function is switched as shown below according to the combinations of the operating mode, bits AE3 to AE0 in PFCR and the bit PA0DDR.

Operating mode	Modes 4 to 6		
AE3 to AE0	B'0000 to B'1000		B'1001 to B'1111
PA0DDR	0	1	
Pin function	PA0 input	PA0 output	A16 output

Operating mode	Mode 7	
PA0DDR	0	1
Pin function	PA0 input	PA0 output

9.6 Port B

Port B is an 8-bit I/O port. Port B also functions as TPU I/O pins and address bus output pins. The pin functions are determined by the operating mode.

Port B has a built-in MOS input pull-up function that can be controlled by software.

- Port B data direction register (PBDDR)
- Port B data register (PBDR)
- Port B register (PORTB)
- Port B pull-up MOS control register (PBPCR)
- Port B open-drain control register (PBODR)

9.6.1 Port B Data Direction Register (PBDDR)

PBDDR is a write-only register, the individual bits of which specify input or output for the pins of port B. PBDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DDR	0	W	In modes 4 to 6, in accordance with the setting in the bits AE3 to AE0 in PFCR, corresponding port B pin is an output pin regardless of the PBDDR setting. When pins are not address outputs, setting a DDR bit to 1 makes the corresponding port B pin an output pin, while clearing the bit to 0 makes the pin a an input pin. In mode 7, setting a PBDDR bit makes the corresponding port B pin an output pin, while clearing the bit to 0 makes the pin an input pin.
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	

9.6.2 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	An output data for a pin is stored when the pin function is specified to a general purpose output port.
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	

9.6.3 Port B Register (PORTB)

PORTB shows the pin states. It cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7	—*	R	If this register read is performed while PBDDR bits are set to 1, the PBDR values are read. If this register read is performed while PBDDR bits are cleared to 0, the pin states are read.
6	PB6	—*	R	
5	PB5	—*	R	
4	PB4	—*	R	
3	PB3	—*	R	
2	PB2	—*	R	
1	PB1	—*	R	
0	PB0	—*	R	

Note: * Determined by state of pins PB7 to PB0.

9.6.4 Port B Pull-Up MOS Control Register (PBPCR)

PBPCR controls the MOS input pull-up function incorporated into port B on an individual bit basis.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7PCR	0	R/W	In modes 4 to 6, if a pin is in the input state in accordance with the settings in PFCR, TIOR of TPU, and in PBDDR, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for that pin. In mode 7, if a pin is in the input state in accordance with the setting in TIOR of TPU and in PBDDR, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for that pin.
6	PB6PCR	0	R/W	
5	PB5PCR	0	R/W	
4	PB4PCR	0	R/W	
3	PB3PCR	0	R/W	
2	PB2PCR	0	R/W	
1	PB1PCR	0	R/W	
0	PB0PCR	0	R/W	

9.6.5 Port B Open-Drain Control Register (PBODR)

PBODR controls output of port B.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7ODR	0	R/W	When pins are not address outputs in accordance with the setting of bits AE3 to AE0 in PFCR, setting a PBODR bit to 1 makes the corresponding port B pin an NMOS open-drain output, while clearing the bit to 0 makes the pin a CMOS output.
6	PB6ODR	0	R/W	
5	PB5ODR	0	R/W	
4	PB4ODR	0	R/W	
3	PB3ODR	0	R/W	
2	PB2ODR	0	R/W	
1	PB1ODR	0	R/W	
0	PB0ODR	0	R/W	

9.6.6 Pin Functions

The correspondence between the register specification and the pin functions is shown below.

- PB7/A15/TIOCB5

The pin function is switched as shown below according to the combination of the operating mode, bits AE3 to AE0 in PFCR, bits MD3 to MD0 in TMDR_5, bits IOB3 to IOB0 in TIOR_5, the TPU channel 5 setting (by bits CCLR1 and CCLR0 in TCR_5), and bit PB7DDR.

Operating mode	Modes 4 to 6			
AE3 to AE0	B'0000 to B'0111			B'1000 to B'1111
TPU channel 5 setting	Table below (1)	Table below (2)		—
PB7DDR	—	0	1	—
Pin function	TIIOCB5 output	PB7 input	PB7 output	A15 Output
		TIOCB5 input*		

Operating mode	Mode 7		
TPU channel 5 setting	Table below (1)	Table below (2)	
PB7DDR	—	0	1
Pin function	TIOCB5 output	PB7 input	PB7 output
		TIOCB5 input*	

TPU channel 5 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	B0001 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 input	—	—	PWM mode 2 output	—

[Legend] x: Don't care

Note: * TIOCB5 input when MD3 to MD0=B'0000, B'01xx, and IOB3=1.

- PB6/A14/TIOCA5

The pin function is switched as shown below according to the combination of the operating mode, bits AE3 to AE0 in PFCR, bits MD3 to MD0 in TMDR_5, bits IOA3 to IOA0 in TIOR_5, the TPU channel 5 setting (by bits CCLR1 and CCLR0 in TCR_5), and bit PB6DDR.

Operating mode	Modes 4 to 6			
AE3 to AE0	B'0000 to B'0110			B'0111 to B'1111
TPU channel 5 setting	Table below (1)	Table below (2)		—
PB6DDR	—	0	1	—
Pin function	TIOCA5 output	PB6 input	PB6 output	A14 Output
		TIOCA5 input*[1]		

Operating mode	Mode 7		
TPU channel 5 setting	Table below (1)	Table below (2)	
PB6DDR	—	0	1
Pin function	TIOCA5 output	PB6 input	PB6 output
		TIOCA5 input*[1]	

TPU channel 5 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. TIOCA5 input when MD3 to MD0=B'0000, B'01xx, and IOA3=1.

2. TIOCB5 is disabled.

- PB5/A13/TIOCB4

The pin function is switched as shown below according to the combination of the operating mode, bits AE3 to AE0 in PFCR, bits MD3 to MD0 in TMDR_4, bits IOB3 to IOB0 in TIOR_4, the TPU channel 4 setting (by bits CCLR1 and CCLR0 in TCR_4), and bit PB5DDR.

Operating mode	Modes 4 to 6			
AE3 to AE0	B'0000 to B'0101			B'0110 to B'1111
TPU channel 4 setting	Table below (1)	Table below (2)		—
PB5DDR	—	0	1	—
Pin function	TIOCB4 output	PB5 input	PB5 output	A13 Output
		TIOCB4 input*		

Operating mode	Mode 7		
TPU channel 4 setting	Table below (1)	Table below (2)	
PB5DDR	—	0	1
Pin function	TIOCB4 output	PB5 input	PB5 output
		TIOCB4 input*	

TPU channel 4 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

Note: * TIOCB4 input when MD3 to MD0=B'0000, B'01xx, and IOB3 to IOB0=B'10xx.

- PB4/A12/TIOCA4

The pin function is switched as shown below according to the combination of the operating mode, bits AE3 to AE0 in PFCR, bits MD3 to MD0 in TMDR_4, bits IOA3 to IOB0 in TIOR_4, the TPU channel 4 setting (by bits CCLR1 and CCLR0 in TCR_4), and bit PB4DDR.

Operating mode	Modes 4 to 6			
AE3 to AE0	B'0000 to B'0100			B'0101 to B'1111
TPU channel 4 setting	Table below (1)	Table below (2)		—
PB4DDR	—	0	1	—
Pin function	TIOCA4 output	PB4 input	PB4 output	A12 Output
		TIOCA4 input*[1]		

Operating mode	Mode 7		
TPU channel 4 setting	Table below (1)	Table below (2)	
PB4DDR	—	0	1
Pin function	TIOCA4 output	PB4 input	PB4 output
		TIOCA4 input*[1]	

TPU channel 4 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	—	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. TIOCA4 input when MD3 to MD0=B'0000, B'01xx, and IOA3 to IOA0=B'10xx.

2. TIOCB4 is disabled.

- PB3/A11/TIOCD3

The pin function is switched as shown below according to the combination of the operating mode, bits AE3 to AE0 in PFCR, bits MD3 to MD0 in TMDR_3, bits IOD3 to IOD0 in TIORL_3, the TPU channel 3 setting (by bits CCLR2 to CCLR0 in TCR_3), and bit PB3DDR.

Operating mode	Modes 4 to 6			
AE3 to AE0	B'0000 to B'0011			B'0100 to B'1111
TPU channel 3 setting	Table below (1)	Table below (2)		—
PB3DDR	—	0	1	—
Pin function	TIOCD3 output	PB3 input	PB3 output	A11 Output
		TIOCD3 input*		

Operating mode	Mode 7		
TPU channel 3 setting	Table below (1)	Table below (2)	
PB3DDR	—	0	1
Pin function	TIOCD3 output	PB3 input	PB3 output
		TIOCD3 input*	

TPU channel 3 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

Note: * TIOCD3 input when MD3 to MD0=B'0000, B'01xx, and IOD3 to IOD0=B'10xx.

- PB2/A10/TIOCC3

The pin function is switched as shown below according to the combination of the operating mode, bits AE3 to AE0 in PFCR, bits MD3 to MD0 in TMDR_3, bits IOC3 to IOC0 in TIORL_3, the TPU channel 3 setting (by bits CCLR2 to CCLR0 in TCR_3), and bit PB2DDR.

Operating mode	Modes 4 to 6			
AE3 to AE0	B'0000 to B'0010			B'0011 to B'1111
TPU channel 3 setting	Table below (1)	Table below (2)		—
PB2DDR	—	0	1	—
Pin function	TIOCC3 output	PB2 input	PB2 output	A10 Output
		TIOCC3 input*[1]		

Operating mode	Mode 7		
TPU channel 3 setting	Table below (1)	Table below (2)	
PB2DDR	—	0	1
Pin function	TIOCC3 output	PB2 input	PB2 output
		TIOCC3 input*[1]	

TPU channel 3 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 mode 1 output*[2]—	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. TIOCC3 input when MD3 to MD0=B'0000 and IOC3 to IOC0=B'10xx.

2. TIOCD3 is disabled.

- PB1/A9/TIOCB3

The pin function is switched as shown below according to the combination of the operating mode, bits AE3 to AE0 in PFCR, bits MD3 to MD0 in TMDR_3, bits IOB3 to IOB0 in TIORH_3, the TPU channel 3 setting (by bits CCLR2 to CCLR0 in TCR_3), and bit PB1DDR.

Operating mode	Modes 4 to 6			
AE3 to AE0	B'0000 to B'0001			B'0010 to B'1111
TPU channel 3 setting	Table below (1)	Table below (2)		—
PB1DDR	—	0	1	—
Pin function	TIOCB3 output	PB1 input	PB1 output	A9 Output
		TIOCB3 input*		

Operating mode	Mode 7		
TPU channel 3 setting	Table below (1)	Table below (2)	
PB1DDR	—	0	1
Pin function	TIOCB3 output	PB1 input	PB1 output
		TIOCB3 input*	

TPU channel 3 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'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

[Legend]

x: Don't care

Note: * TIOCB3 input when MD3 to MD0=B'0000, B'01xx, and IOB3 to IOB0=B'10xx.

- PB0/A8/TIOCA3

The pin function is switched as shown below according to the combination of the operating mode, bits AE3 to AE0 in PFCR, bits MD3 to MD0 in TMDR_3, bits IOA3 to IOA0 in TIORH_3, the TPU channel 3 setting (by bits CCLR2 to CCLR0 in TCR_3), and bit PB0DDR.

Operating mode	Modes 4 to 6			
AE3 to AE0	B'0000			B'0001 to B'1111
TPU channel 3 setting	Table below (1)	Table below (2)		—
PB0DDR	—	0	1	—
Pin function	TIOCA3 output	PB0 input	PB0 output	A8 Output
		TIOCA3 input*[1]		

Operating mode	Mode 7		
TPU channel 3 setting	Table below (1)	Table below (2)	
PB0DDR	—	0	1
Pin function	TIOCA3 output	PB0 input	PB0 output
		TIOCA3 input*[1]	

TPU channel 3 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'001	B'001
Output function	—	Output compare output	—	PWM mode 1 output*[2]—	PWM mode 2 output	—

[Legend]

x: Don't care

Notes: 1. TIOCA3 input when MD3 to MD0=B'0000 and IOA3 to IOA0=B'10xx.

2. TIOCB3 is disabled.

9.7 Port C

Port C is an 8-bit I/O port. Port C also functions as address bus output pins. The pin functions are determined by the operating mode.

Port C has a built-in MOS input pull-up function that can be controlled by software.

- Port C data direction register (PCDDR)
- Port C data register (PCDR)
- Port C register (PORTC)
- Port C pull-up MOS control register (PCPCR)
- Port C open-drain control register (PCODR)

9.7.1 Port C Data Direction Register (PCDDR)

PCDDR is a write-only register, the individual bits of which specify input or output for the pins of port C. PCDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DDR	0	W	In modes 4 and 5, regardless of the PCDDR setting, the corresponding port C pin is an output pin. In mode 6, setting a PCDDR bit to 1 makes the corresponding port C pin an address output pin, while clearing the bit to 0 makes the pin an input pin. In mode 7, setting a PCDDR bit to 1 makes the corresponding port C pin an output pin, while clearing the bit to 0 makes the pin an input pin.
6	PC6DDR	0	W	
5	PC5DDR	0	W	
4	PC4DDR	0	W	
3	PC3DDR	0	W	
2	PC2DDR	0	W	
1	PC1DDR	0	W	
0	PC0DDR	0	W	

9.7.2 Port C Data Register (PCDR)

PCDR stores output data for the port C pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose output port.
6	PC6DR	0	R/W	
5	PC5DR	0	R/W	
4	PC4DR	0	R/W	
3	PC3DR	0	R/W	
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

9.7.3 Port C Register (PORTC)

PORTC shows the pin states. It cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7	—*	R	If this register read is performed while PCDDR bits are set to 1, the PCDR values are read. If this register read is performed while PCDDR bits are cleared to 0, the pin states are read.
6	PC6	—*	R	
5	PC5	—*	R	
4	PC4	—*	R	
3	PC3	—*	R	
2	PC2	—*	R	
1	PC1	—*	R	
0	PC0	—*	R	

Note: * Determined by state of pins PC7 to PC0.

9.7.4 Port C Pull-Up MOS Control Register (PCPCR)

PCPCR controls the MOS input pull-up function incorporated into port C on an individual bit basis.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7PCR	0	R/W	In modes 4 to 6, if a pin is in the input state in accordance with the settings in PFCR and in PCDDR, setting the corresponding PCPCR bit to 1 turns on the MOS input pull-up for that pin. In mode 7, if a pin is in the input state in accordance with the settings in PCDDR, setting the corresponding PCPCR bit to 1 turns on the MOS input pull-up for that pin.
6	PC6PCR	0	R/W	
5	PC5PCR	0	R/W	
4	PC4PCR	0	R/W	
3	PC3PCR	0	R/W	
2	PC2PCR	0	R/W	
1	PC1PCR	0	R/W	
0	PC0PCR	0	R/W	

9.7.5 Port C Open-Drain Control Register (PCODR)

PCODR controls output of port C.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7ODR	0	R/W	If PCODR is set to 1 by setting AE3 to AE0 in PFCR in mode other than address output mode, port C pins function as NMOS open drain outputs and when the setting is cleared to 0, the pins function as CMOS outputs.
6	PC6ODR	0	R/W	
5	PC5ODR	0	R/W	
4	PC4ODR	0	R/W	
3	PC3ODR	0	R/W	
2	PC2ODR	0	R/W	
1	PC1ODR	0	R/W	
0	PC0ODR	0	R/W	

9.7.6 Pin Functions

The correspondence between the register specification and the pin functions is shown below.

- PC7/A7 to PC0/A0

The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PCnDDR bit in PCDDR.

Operating Mode	Modes 4 and 6			Mode 7	
Setting of AE3 to AE0	Address output enabled	Address output disabled		—	
PCnDDR	—	0	1	0	1
Pin function	A7 to A0 output	PC7 to PC0 input	PC7 to PC0 output	PC7 to PC0 input	PC7 to PC0 output

9.8 Port D

Port D is an 8-bit I/O port. Port D has a data bus I/O function, and the pin functions change according to the operating mode.

Port D has a built-in MOS input pull-up function that can be controlled by software.

- Port D data direction register (PDDDR)
- Port D data register (PDDR)
- Port D register (PORTD)
- Port D pull-up MOS control register (PDPCR)

9.8.1 Port D Data Direction Register (PDDDR)

PDDDR is a write-only register, the individual bits of which specify input or output for the pins of port D. PDDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DDR	0	W	Setting a PDDDR bit to 1 makes the corresponding port D pin an output pin, while clearing the bit to 0 makes the pin an input pin.
6	PD6DDR	0	W	
5	PD5DDR	0	W	
4	PD4DDR	0	W	
3	PD3DDR	0	W	
2	PD2DDR	0	W	
1	PD1DDR	0	W	
0	PD0DDR	0	W	

9.8.2 Port D Data Register (PDDR)

PDDR stores output data for the port D pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose output port.
6	PD6DR	0	R/W	
5	PD5DR	0	R/W	
4	PD4DR	0	R/W	
3	PD3DR	0	R/W	
2	PD2DR	0	R/W	
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

9.8.3 Port D Register (PORTD)

PORTD shows the pin states. It cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7	—*	R	If this register read is performed while PDDDR bits are set to 1, the PDDR values are read. If this register read is performed while PDDDR bits are cleared to 0, the pin states are read.
6	PD6	—*	R	
5	PD5	—*	R	
4	PD4	—*	R	
3	PD3	—*	R	
2	PD2	—*	R	
1	PD1	—*	R	
0	PD0	—*	R	

Note: * Determined by state of pins PD7 to PD0.

9.8.4 Port D Pull-Up MOS Control Register (PDPCR)

PDPCR controls the MOS input pull-up function incorporated into port D on an individual bit basis.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7PCR	0	R/W	In mode 7, if a pin is in the input state in accordance with the setting in PDDDR, setting the corresponding PDPCR bit to 1 turns on the MOS input pull-up for that pin.
6	PD6PCR	0	R/W	
5	PD5PCR	0	R/W	
4	PD4PCR	0	R/W	
3	PD3PCR	0	R/W	
2	PD2PCR	0	R/W	
1	PD1PCR	0	R/W	
0	PD0PCR	0	R/W	

9.8.5 Pin Functions

The correspondence between the register specification and the pin functions is shown below.

Operating Mode	Mode 4 to 6	Mode 7	
PDnDDR	—	0	1
Pin function	D15 to D8 I/O	PD7 to PD0 input	PD7 to PD0 output

(n = 7 to 0)

9.9 Port E

Port E is an 8-bit I/O port. Port E has a data bus I/O function, and the pin functions change according to the operating mode and whether 8-bit or 16-bit bus mode is selected.

Port E has a built-in MOS input pull-up function that can be controlled by software.

- Port E data direction register (PEDDR)
- Port E data register (PEDR)
- Port E register (PORTE)
- Port E pull-up MOS control register (PEPCR)

9.9.1 Port E Data Direction Register (PEDDR)

PEDDR is a write-only register, the individual bits of which specify input or output for the pins of port E. PEDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DDR	0	W	Setting a PEDDR bit to 1 makes the corresponding port E pin an output pin, while clearing the bit to 0 makes the pin an input pin.
6	PE6DDR	0	W	
5	PE5DDR	0	W	
4	PE4DDR	0	W	
3	PE3DDR	0	W	
2	PE2DDR	0	W	
1	PE1DDR	0	W	
0	PE0DDR	0	W	

9.9.2 Port E Data Register (PEDR)

PEDR stores output data for the port E pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose output port.
6	PE6DR	0	R/W	
5	PE5DR	0	R/W	
4	PE4DR	0	R/W	
3	PE3DR	0	R/W	
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

9.9.3 Port E Register (PORTE)

PORTE shows the pin states. It cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7	—*	R	If this register read is performed while PEDDR bits are set to 1, the PEDR values are read.
6	PE6	—*	R	
5	PE5	—*	R	
4	PE4	—*	R	
3	PE3	—*	R	
2	PE2	—*	R	
1	PE1	—*	R	
0	PE0	—*	R	

Note: * Determined by state of pins PE7 to PE0.

9.9.4 Port E Pull-Up MOS Control Register (PEPCR)

PEPCR controls the MOS input pull-up function incorporated into port E on an individual bit basis.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7PCR	0	R/W	In modes 4 to 6 with 8-bit-bus mode selected, or in mode 7, if a pin is in the input state in accordance with the setting in PEDDR, setting the corresponding PEPCR bit to 1 turns on the MOS input pull-up for that pin.
6	PE6PCR	0	R/W	
5	PE5PCR	0	R/W	
4	PE4PCR	0	R/W	
3	PE3PCR	0	R/W	
2	PE2PCR	0	R/W	
1	PE1PCR	0	R/W	
0	PE0PCR	0	R/W	

(n = 7 to 0)

9.9.5 Pin Functions

The correspondence between the register specification and the pin functions is shown below.

Operating Mode	Modes 4 to 6			Mode 7	
Bus width setting	16-bit mode	8-bit mode		—	
PEnDDR	—	0	1	0	1
Pin function	D7 to D0 I/O	PE7 to PE0 input	PE7 to PE0 output	PE7 to PE0 input	PE7 to PE0 output

(n = 7 to 0)

9.10 Port F

Port F is a 6-bit I/O port. Port F also functions as external interrupt input pins, the A/D trigger input pin, bus control signal I/O pins, and as the system clock output pin.

- Port F data direction register (PFDDR)
- Port F data register (PFDR)
- Port F register (PORTF)

9.10.1 Port F Data Direction Register (PFDDR)

PFDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port F. PFDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DDR	1*	W	Setting a PFDDR bit to 1 makes the PF7 pin a ϕ output pin, while clearing the bit to 0 makes the pin an input pin.
6	PF6DDR	0	W	Setting a PFDDR bit to 1 makes the corresponding port F pin an output port, while clearing the bit to 0 makes the pin an input port. Bit 2 and bit 1 are reserved bits.
5	PF5DDR	0	W	
4	PF4DDR	0	W	
3	PF3DDR	0	W	
2	—	Undefined	—	Reserved
1	—	Undefined	—	Reserved
0	PF0DDR	0	W	

Note: * The initial value is 0 in mode 7.

9.10.2 Port F Data Register (PFDR)

PFDR stores output data for the port F pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose output port. Bit 2 and bit 1 are reserved bits.
6	PF6DR	0	R/W	
5	PF5DR	0	R/W	
4	PF4DR	0	R/W	
3	PF3DR	0	R/W	
2	—	Undefined	—	Reserved
1	—	Undefined	—	Reserved
0	PF0DR	0	R/W	

9.10.3 Port F Register (PORTF)

PORTF shows the pin states. It cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7	Undefined *	R	If this register read is performed while PFDDR bits are set to 1, the PFDR values are read. If this register read is performed while PFDDR bits are cleared to 0, the pin states are read.
6	PF6	Undefined *	R	
5	PF5	Undefined *	R	
4	PF4	Undefined *	R	
3	PF3	Undefined *	R	
2	PF2	Undefined *	R	
1	PF1	Undefined *	R	
0	PF0	Undefined *	R	

Note: * Determined by state of pins PF7 to PF0.

9.10.4 Pin Functions

The correspondence between the register specification and the pin functions is shown below.

- PF7/ϕ

The pin function is switched as shown below according to bit PF7DDR.

PF7DDR	0	1
Pin function	PF7 input	ϕ output

- PF6/$\overline{AS}$

The pin function is switched as shown below according to the operating mode and the setting of PF6DDR bit.

Operating Mode	Modes 4 to 6	Mode 7	
PF6DDR	—	0	1
Pin function	$\overline{AS}$ output	PF6 input	PF6 output

- PF5/$\overline{RD}$

The pin function is switched as shown below according to the operating mode and the setting of PF5DDR bit.

Operating Mode	Modes 4 to 6	Mode 7	
PF5DDR	—	0	1
Pin function	$\overline{RD}$ output	PF5 input	PF5 output

- PF4/$\overline{HWR}$

The pin function is switched as shown below according to the operating mode and the setting of PF4DDR bit.

Operating Mode	Modes 4 to 6	Mode 7	
PF4DDR	—	0	1
Pin function	$\overline{HWR}$ output	PF4 input	PF4 output

- PF3/$\overline{\text{LWR}}$/$\overline{\text{ADTRG}}$/$\overline{\text{IRQ3}}$

The pin function is switched as shown below according to the operating mode and the setting of bits TRGS1 and TRGS0 bits in ADCR and PF3DDR bit.

Operating Mode	Modes 4 to 6			Mode 7	
Bus Mode	16	8			
PF3DDR	—	0	1	0	1
Pin function	$\overline{\text{LWR}}$ output	PF3 input	PF3 output	PF3 input	PF3 output
		$\overline{\text{ADTRG}}$ input*1			
		$\overline{\text{IRQ3}}$ input*2			

Notes: 1 $\overline{\text{ADTRG}}$ input when TRGS0=TRGS1=1.
2 When using as the external interrupt pins, refrain from using as I/O pins of other functions.

- PF0/$\overline{\text{IRQ2}}$

The pin function is switched as shown below according to the PF0DDR bit.

PF0DDR	0	1
Pin function	PF0 input	PF0 output
	$\overline{\text{IRQ2}}$ input	

9.11 Port H

Port H is an 8-bit I/O port. Port H pins also function as motor control PWM timer output pins.

- Port H data direction register (PHDDR)
- Port H data register (PHDR)
- Port H register (PORTH)

9.11.1 Port H Data Direction Register (PHDDR)

PHDDR is a write-only register, the individual bits of which specify input or output for the pins of port H. PHDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PH7DDR	0	W	Setting a PHDDR bit to 1 makes the corresponding port H pin an output pin, while clearing the bit to 0 makes the pin an input pin.
6	PH6DDR	0	W	
5	PH5DDR	0	W	
4	PH4DDR	0	W	
3	PH3DDR	0	W	
2	PH2DDR	0	W	
1	PH1DDR	0	W	
0	PH0DDR	0	W	

9.11.2 Port H Data Register (PHDR)

PHDR stores output data for the port H pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PH7DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose output port.
6	PH6DR	0	R/W	
5	PH5DR	0	R/W	
4	PH4DR	0	R/W	
3	PH3DR	0	R/W	
2	PH2DR	0	R/W	
1	PH1DR	0	R/W	
0	PH0DR	0	R/W	

9.11.3 Port H Register (PORTH)

PORTH shows the pin states. It cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PH7	—*	R	If this register read is performed while PHDDR bits are set to 1, the PHDR values are read. If this register read is performed while PHDDR bits are cleared to 0, the pin states are read.
6	PH6	—*	R	
5	PH5	—*	R	
4	PH4	—*	R	
3	PH3	—*	R	
2	PH2	—*	R	
1	PH1	—*	R	
0	PH0	—*	R	

Note: * Determined by state of pins PH7 to PH0.

9.11.4 Pin Functions

The correspondence between the register specification and the pin functions is shown below.

The pin function is switched as shown below according to the OE1A to OE1H bits in PWOCR_1 of the motor control PWM timer and the PHDDR values.

OE1A to OE1H	1	0	
PHDDR	—	0	1
Pin function	Motor control PWM timer output	PH7 to PH0 input	PH7 to PH0 output

9.12 Port J

Port J is an 8-bit I/O port. Port J pins also function as motor control PWM timer output pins.

- Port J data direction register (PJDDR)
- Port J data register (PJDR)
- Port J register (PORTJ)

9.12.1 Port J Data Direction Register (PJDDR)

PJDDR is a write-only register, the individual bits of which specify input or output for the pins of port J. PJDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PJ7DDR	0	W	Setting a PJDDR bit to 1 makes the corresponding port J pin an output pin, while clearing the bit to 0 makes the pin an input pin.
6	PJ6DDR	0	W	
5	PJ5DDR	0	W	
4	PJ4DDR	0	W	
3	PJ3DDR	0	W	
2	PJ2DDR	0	W	
1	PJ1DDR	0	W	
0	PJ0DDR	0	W	

9.12.2 Port J Data Register (PJDR)

PJDR stores output data for the port J pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PJ7DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose output port.
6	PJ6DR	0	R/W	
5	PJ5DR	0	R/W	
4	PJ4DR	0	R/W	
3	PJ3DR	0	R/W	
2	PJ2DR	0	R/W	
1	PJ1DR	0	R/W	
0	PJ0DR	0	R/W	

9.12.3 Port J Register (PORTJ)

PORTJ shows the pin states. It cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PJ7	—*	R	If this register read is performed while PJDDR bits are set to 1, the PJDR values are read. If this register read is performed while PJDDR bits are cleared to 0, the pin states are read.
6	PJ6	—*	R	
5	PJ5	—*	R	
4	PJ4	—*	R	
3	PJ3	—*	R	
2	PJ2	—*	R	
1	PJ1	—*	R	
0	PJ0	—*	R	

Note: * Determined by state of pins PJ7 to PJ0.

9.12.4 Pin Functions

The correspondence between the register specification and the pin functions is shown below.

The pin function is switched as shown below according to the OE2A to OE2H bits in PWOCR_2 of the motor control PWM timer and the PJDDR values.

OE2A to OE2H	1	0	
PJDDR	—	0	1
Pin function	Motor control PWM timer output	PJ7 to PJ0 input	PJ7 to PJ0 output

Section 10 16-Bit Timer Pulse Unit (TPU)

This LSI has an on-chip 16-bit timer pulse unit (TPU) comprised of six 16-bit timer channels.

The function list of the 16-bit timer unit and its block diagram are shown in table 10.1 and figure 10.1, respectively.

10.1 Features

- Maximum 16-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 is possible
    Register simultaneous input/output is possible by synchronous counter operation
  - A maximum 15-phase PWM output is possible in combination with synchronous operation
- Buffer operation settable for channels 0 and 3
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
- Cascaded operation
- Fast access via internal 16-bit bus
- 26 interrupt sources
- Automatic transfer of register data
- Programmable pulse generator (PPG) output trigger can be generated
- A/D converter conversion start trigger can be generated
- Module stop mode can be set

Table 10.1 TPU Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Count clock		φ/1 φ/4 φ/16 φ/64 TCLKA TCLKB TCLKC TCLKD	φ/1 φ/4 φ/16 φ/64 φ/256 TCLKA TCLKB	φ/1 φ/4 φ/16 φ/64 φ/1024 TCLKA TCLKB TCLKC	φ/1 φ/4 φ/16 φ/64 φ/256 φ/1024 φ/4096 TCLKA	φ/1 φ/4 φ/16 φ/64 φ/1024 TCLKA TCLKC	φ/1 φ/4 φ/16 φ/64 φ/256 TCLKA TCLKC TCLKD
General registers (TGR)		TGRA_0 TGRB_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2	TGRA_3 TGRB_3	TGRA_4 TGRB_4	TGRA_5 TGRB_5
General registers/ buffer registers		TGRC_0 TGRD_0	—	—	TGRC_3 TGRD_3	—	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TIOCA5 TIOCB5
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	○	○	○	○	○	○
	1 output	○	○	○	○	○	○
	Toggle output	○	○	○	○	○	○
Input capture function		○	○	○	○	○	○
Synchronous operation		○	○	○	○	○	○
PWM mode		○	○	○	○	○	○
Phase counting mode		—	○	○	—	○	○
Buffer operation		○	—	—	○	—	—

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
A/D converter trigger	TGRA_0 compare match or input capture	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture	TGRA_3 compare match or input capture	TGRA_4 compare match or input capture	TGRA_5 compare match or input capture
PPG trigger	TGRA_0/ TGRB_0 compare match or input capture	TGRA_1/ TGRB_1 compare match or input capture	TGRA_2/ TGRB_2 compare match or input capture	TGRA_3/ TGRB_3 compare match or input capture	—	—

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
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	5 sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	4 sources • Compare match or input capture 4A • Compare match or input capture 4B • Overflow • Underflow	4 sources • Compare match or input capture 5A • Compare match or input capture 5B • Overflow • Underflow

[Legend]

○: Possible

—: Not possible

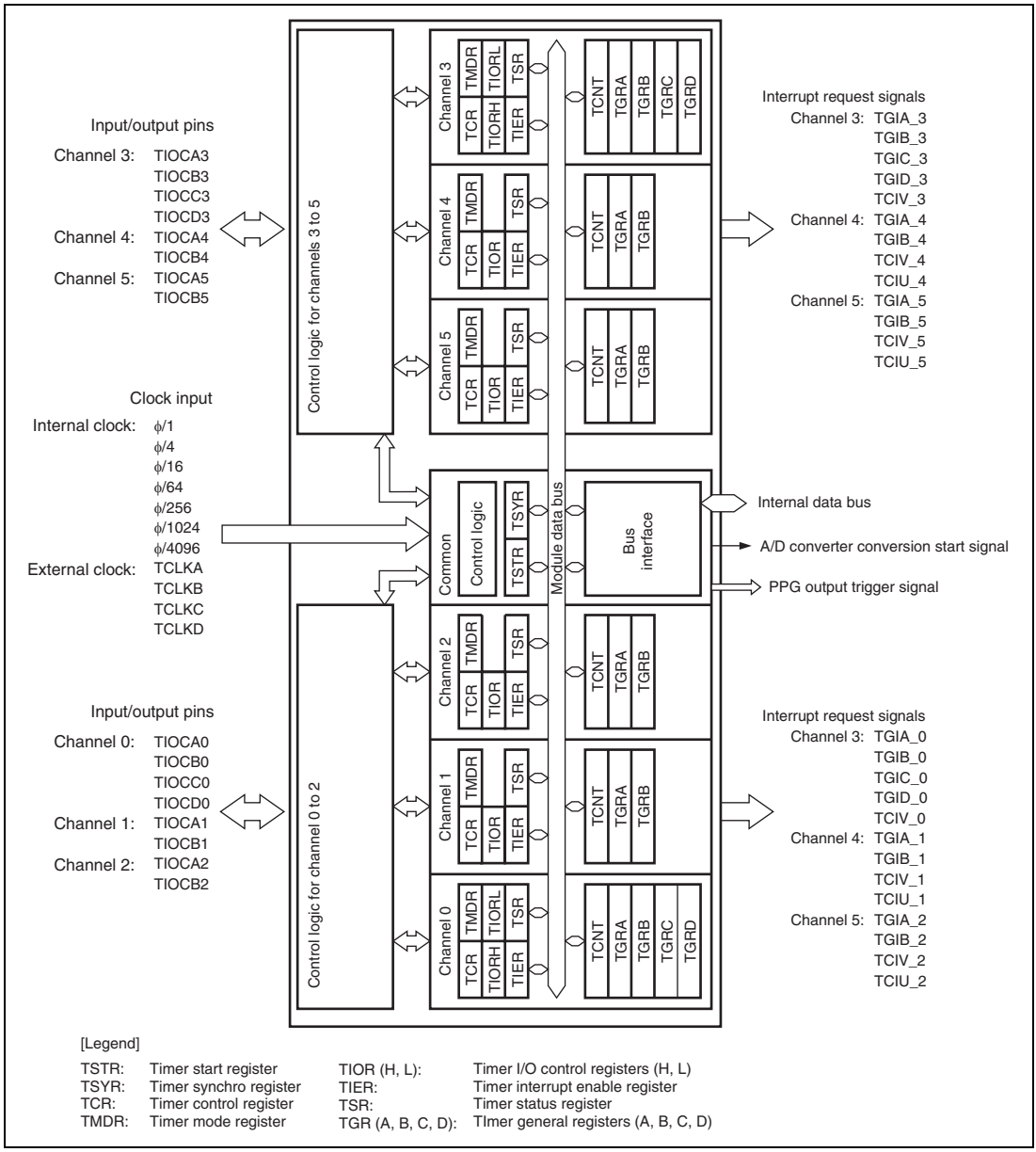


Figure 10.1 Block Diagram of TPU

10.2 Input/Output Pins

Table 10.2 TPU Pins

Channel	Symbol	I/O	Function
All	TCLKA	Input	External clock A input pin (Channel 1 and 5 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (Channel 1 and 5 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channel 2 and 4 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channel 2 and 4 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
3	TIOCA3	I/O	TGRA_3 input capture input/output compare output/PWM output pin
	TIOCB3	I/O	TGRB_3 input capture input/output compare output/PWM output pin
	TIOCC3	I/O	TGRC_3 input capture input/output compare output/PWM output pin
	TIOCD3	I/O	TGRD_3 input capture input/output compare output/PWM output pin
4	TIOCA4	I/O	TGRA_4 input capture input/output compare output/PWM output pin
	TIOCB4	I/O	TGRB_4 input capture input/output compare output/PWM output pin
5	TIOCA5	I/O	TGRA_5 input capture input/output compare output/PWM output pin
	TIOCB5	I/O	TGRB_5 input capture input/output compare output/PWM output pin

10.3 Register Descriptions

The TPU has the following registers. To distinguish registers in each channel, an underscore and the channel number are added as a suffix to the register name; TCR for channel 0 is expressed as TCR_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)
- 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)
- 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)
- Timer control register_3 (TCR_3)
- Timer mode register_3 (TMDR_3)

- Timer I/O control register H_3 (TIORH_3)
- Timer I/O control register L_3 (TIORL_3)
- Timer interrupt enable register_3 (TIER_3)
- Timer status register_3 (TSR_3)
- Timer counter_3 (TCNT_3)
- Timer general register A_3 (TGRA_3)
- Timer general register B_3 (TGRB_3)
- Timer general register C_3 (TGRC_3)
- Timer general register D_3 (TGRD_3)
- Timer control register_4 (TCR_4)
- Timer mode register_4 (TMDR_4)
- Timer I/O control register _4 (TIOR_4)
- Timer interrupt enable register_4 (TIER_4)
- Timer status register_4 (TSR_4)
- Timer counter_4 (TCNT_4)
- Timer general register A_4 (TGRA_4)
- Timer general register B_4 (TGRB_4)
- Timer control register_5 (TCR_5)
- Timer mode register_5 (TMDR_5)
- Timer I/O control register_5 (TIOR_5)
- Timer interrupt enable register_5 (TIER_5)
- Timer status register_5 (TSR_5)
- Timer counter_5 (TCNT_5)
- Timer general register A_5 (TGRA_5)
- Timer general register B_5 (TGRB_5)

Common Registers

- Timer start register (TSTR)
- Timer synchro register (TSYR)

10.3.1 Timer Control Register (TCR)

The TCR registers are 8-bit readable/writable registers that control the TCNT operation for each channel. The TPU has a total of six TCR registers, one for each channel (channel 0 to 5). TCR register settings should be conducted only when TCNT operation is stopped.

Bit	Bit Name	Initial value	R/W	Description
7	CCLR2	0	R/W	Counter Clear 0 to 2
6	CCLR1	0	R/W	These bits select the TCNT counter clearing source. See tables 10.3 and 10.4 for details.
5	CCLR0	0	R/W	
4	CKEG1	0	R/W	Clock Edge 0 and 1
3	CKEG0	0	R/W	These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. ϕ/4 both edges = ϕ/2 rising edge). If phase counting mode is used on channels 1, 2, 4, and 5, 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, or when overflow/underflow of another channel 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	Time Prescaler 0 to 2
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 10.5 to 10.10 for details.
0	TPSC0	0	R/W	

Table 10.3 CCLR0 to CCLR2 (Channels 0 and 3)

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0, 3	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 is set 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 does not occur.

Table 10.4 CCLR0 to CCLR2 (Channels 1, 2, 4, and 5)

Channel	Bit 7 Reserved*2	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5	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 is selected by setting the SYNC bit in TSYR to 1.
2. Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and cannot be modified.

Table 10.5 TPSC0 to TPSC2 (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 10.6 TPSC0 to TPSC2 (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 10.7 TPSC0 to TPSC2 (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.

Table 10.8 TPSC0 to TPSC2 (Channel 3)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3	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	Internal clock: counts on ϕ/1024
		1	0	Internal clock: counts on ϕ/256
			1	Internal clock: counts on ϕ/4096

Table 10.9 TPSC0 to TPSC2 (Channel 4)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
4	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 TCLKC pin input
		1	0	Internal clock: counts on ϕ/1024
			1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

Table 10.10 TPSC0 to TPSC2 (Channel 5)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	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 TCLKC pin input
		1	0	Internal clock: counts on ϕ/256
			1	External clock: counts on TCLKD pin input

Note: This setting is ignored when channel 5 is in phase counting mode.

10.3.2 Timer Mode Register (TMDR)

The TMDR registers are 8-bit readable/writable registers that are used to set the operating mode of each channel. The TPU has six TMDR registers, one for each channel. TMDR register settings should be changed 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, 2, 4, and 5, 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
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, 2, 4, and 5, 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 0 to 3 These bits are used to set the timer operating mode. MD3 is a reserved bit. In a write, it should always be written with 0. See table 10.11 for details.
2	MD2	0	R/W	
1	MD1	0	R/W	
0	MD0	0	R/W	

Table 10.11 MD0 to MD3

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	X	X	X	—

[Legend]

X: Don't care

Notes: 1. MD3 is a reserved bit. In a write, it should always be written with 0.

2. Phase counting mode cannot be set for channels 0 and 3. In this case, 0 should always be written to MD2.

10.3.3 Timer I/O Control Register (TIOR)

The TIOR registers are 8-bit readable/writable registers that control the TGR registers. The TPU has eight TIOR registers, two each for channels 0 and 3, and one each for channels 1, 2, 4, and 5.

Care is required as TIOR is affected by the TMDR setting. The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

- TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIOR_4, TIOR_5

Bit	Bit Name	Initial value	R/W	Description
7	IOB3	0	R/W	I/O Control B0 to B3
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 A0 to A3
2	IOA2	0	R/W	Specify the function of TGRA.
1	IOA1	0	R/W	
0	IOA0	0	R/W	

- TIORL_0, TIORL_3

Bit	Bit Name	Initial value	R/W	Description
7	IOD3	0	R/W	I/O Control D0 to D3
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 C0 to C3
2	IOC2	0	R/W	Specify the function of TGRC.
1	IOC1	0	R/W	
0	IOC0	0	R/W	

Table 10.12 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 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is the TIOCB0 pin Input capture at rising edge
			1		Capture input source is the TIOCB0 pin Input capture at falling edge
		1	X		Capture input source is the TIOCB0 pin Input capture at both edges.
	1	X	X		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 TPSC0 to TPSC2 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 10.13 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 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is the TIOCD0 pin Input capture at rising edge
			1		Capture input source is the TIOCD0 pin Input capture at falling edge
		1	X		Capture input source is the TIOCD0 pin Input capture at both edges
	1	X	X		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 TPSC0 to TPSC2 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 10.14 TIOR_1 (Channel 1)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_1 Function**	**TIOCB1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is the TIOCB1 pin Input capture at rising edge
			1		Capture input source is the TIOCB1 pin Input capture at falling edge
		1	X		Capture input source is the TIOCB1 pin Input capture at both edges
	1	X	X		TGRC_0 compare match/ input capture Input capture at generation of TGRC_0 compare match/input capture

[Legend]

X: Don't care

Table 10.15 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 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is the TIOCB2 pin Input capture at rising edge
			1		Capture input source is the TIOCB2 pin Input capture at falling edge
		1	X		Capture input source is the TIOCB2 pin Input capture at both edges

[Legend]

X: Don't care

Table 10.16 TIORH_3 (Channel 3)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_3 Function**	**TIOCB3 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is the TIOCB3 pin Input capture at rising edge
			1		Capture input source is the TIOCB3 pin Input capture at falling edge
		1	X		Capture input source is the TIOCB3 pin Input capture at both edges
	1	X	X		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down*

[Legend]

X: Don't care

Note: * When bits TPSC0 to TPSC2 in TCR_4 are set to B′000 and φ/1 is used as the TCNT_4 count clock, this setting is invalid and input capture is not generated.

Table 10.17 TIORL_3 (Channel 3)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_3 Function**	**TIOCD3 Pin Function**
0	0	0	0	Output compare register*2	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is the TIOCD3 pin Input capture at rising edge
			1		Capture input source is the TIOCD3 pin Input capture at falling edge
		1	X		Capture input source is the TIOCD3 pin Input capture at both edges
	1	X	X		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down*1

[Legend]

X: Don't care

Notes: 1. When bits TPSC0 to TPSC2 in TCR_4 are set to B'000 and ϕ/1 is used as the TCNT_4 count clock, this setting is invalid and input capture is not generated.

2. When the BFB bit in TMDR_3 is set to 1 and TGRD_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 10.18 TIOR_4 (Channel 4)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_4 Function**	**TIOCB4 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is the TIOCB4 pin Input capture at rising edge
			1		Capture input source is the TIOCB4 pin Input capture at falling edge
		1	X		Capture input source is the TIOCB4 pin Input capture at both edges
	1	X	X		Capture input source is TGRC_3 compare match/input capture Input capture at generation of TGRC_3 compare match/input capture

[Legend]

X: Don't care

Table 10.19 TIOR_5 (Channel 5)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_5 Function**	**TIOCB5 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is the TIOCB5 pin Input capture at rising edge
			1		Capture input source is the TIOCB5 pin Input capture at falling edge
		1	X		Capture input source is the TIOCB5 pin Input capture at both edges

[Legend]

X: Don't care

Table 10.20 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 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is the TIOCA0 pin Input capture at rising edge
			1		Capture input source is the TIOCA0 pin Input capture at falling edge
		1	X		Capture input source is the TIOCA0 pin Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Table 10.21 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 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is the TIOCC0 pin Input capture at rising edge
			1		Capture input source is the TIOCC0 pin Input capture at falling edge
		1	X		Capture input source is the TIOCC0 pin Input capture at both edges
	1	X	X		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 10.22 TIOR_1 (Channel 1)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_1 Function**	**TIOCA1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is the TIOCA1 pin Input capture at rising edge
			1		Capture input source is the TIOCA1 pin Input capture at falling edge
		1	X		Capture input source is the TIOCA1 pin Input capture at both edges
	1	X	X		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 10.23 TIOR_2 (Channel 2)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_2 Function**	**TIOCA2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is the TIOCA2 pin Input capture at rising edge
			1		Capture input source is the TIOCA2 pin Input capture at falling edge
		1	X		Capture input source is the TIOCA2 pin Input capture at both edges

[Legend]

X: Don't care

Table 10.24 TIORH_3 (Channel 3)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_3 Function**	**TIOCA3 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is the TIOCA3 pin Input capture at rising edge
			1		Capture input source is the TIOCA3 pin Input capture at falling edge
		1	X		Capture input source is the TIOCA3 pin Input capture at both edges
	1	X	X		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down

[Legend]

X: Don't care

Table 10.25 TIORL_3 (Channel 3)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_3 Function**	**TIOCC3 Pin Function**
0	0	0	0	Output compare register*	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is the TIOCC3 pin Input capture at rising edge
			1		Capture input source is the TIOCC3 pin Input capture at falling edge
		1	X		Capture input source is the TIOCC3 pin Input capture at both edges
	1	X	X		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down

[Legend]

X: Don't care

Note: * When the BFA bit in TMDR_3 is set to 1 and TGRC_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 10.26 TIOR_4 (Channel 4)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_4 Function**	**TIOCA4 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is the TIOCA4 pin Input capture at rising edge
			1		Capture input source is the TIOCA4 pin Input capture at falling edge
		1	X		Capture input source is the TIOCA4 pin Input capture at both edges
	1	X	X		Capture input source is TGRA_3 compare match/input capture Input capture at generation of TGRA_3 compare match/input capture

[Legend]

X: Don't care

Table 10.27 TIOR_5 (Channel 5)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_5 Function**	**TIOCA5 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is the TIOCA5 pin Input capture at rising edge
			1		Capture input source is the TIOCA5 pin Input capture at falling edge
		1	X		Capture input source is the TIOCA5 pin Input capture at both edges

[Legend]

X: Don't care

10.3.4 Timer Interrupt Enable Register (TIER)

The TIER registers are 8-bit readable/writable registers that control enabling or disabling of interrupt requests for each channel. The TPU has six 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	—	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, 2, 4, and 5. In channels 0 and 3, 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 channels 0 and 3. In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGID) by TGFD bit disabled 1: Interrupt requests (TGID) by TGFD bit enabled

Bit	Bit Name	Initial value	R/W	Description
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 channels 0 and 3. In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGIC) by TGFC bit disabled 1: Interrupt requests (TGIC) by TGFC bit enabled
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabled 1: Interrupt requests (TGIB) by TGFB bit enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled

10.3.5 Timer Status Register (TSR)

The TSR registers are 8-bit readable/writable registers that indicate the status of each channel. The TPU has six TSR registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TCNT counts in channels 1, 2, 4, and 5. In channels 0 and 3, bit 7 is reserved. It is always read as 1 and cannot be modified. 0: TCNT counts down 1: TCNT counts up
6	—	1	—	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, 2, 4, and 5 are set to phase counting mode. Only 0 can be written, for flag clearing. In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified. [Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF) [Clearing condition] When 0 is written to TCFU after reading TCFU = 1
4	TCFV	0	R/(W)	Overflow Flag Status flag that indicates that TCNT overflow has occurred. Only 0 can be written, for flag clearing. [Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000) [Clearing condition] When 0 is written to TCFV after reading TCFV = 1

Bit	Bit Name	Initial value	R/W	Description
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 channels 0 and 3. Only 0 can be written, for flag clearing. In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRD and TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal and TGRD is functioning as input capture register [Clearing conditions] • When DTC is activated by TGID interrupt and the DISEL bit of MRB in DTC is 0 • 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 channels 0 and 3. Only 0 can be written, for flag clearing. In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRC and TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal and TGRC is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIC interrupt and the DISEL bit of MRB in DTC is 0 • 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. Only 0 can be written, for flag clearing. [Setting conditions] • When TCNT = TGRB and TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal and TGRB is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIB interrupt and the DISEL bit of MRB in DTC is 0 • 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. Only 0 can be written, for flag clearing. [Setting conditions] • When TCNT = TGRA and TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal and TGRA is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIA interrupt and the DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after reading TGFA = 1

10.3.6 Timer Counter (TCNT)

The TCNT registers are 16-bit readable/writable counters. The TPU has six TCNT counters, one for each channel.

The TCNT counters are initialized to H'0000 by a reset, and in hardware standby mode.

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

10.3.7 Timer General Register (TGR)

The TGR registers are dual function 16-bit readable/writable registers, functioning as either output compare or input capture registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two each for channels 1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated for operation as buffer registers. The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit. TGR buffer register combinations are TGRA—TGRC and TGRB—TGRD.

10.3.8 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects operation/stoppage for channels 0 to 5. 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, 6	—	All 0	—	Reserved The write value should always be 0.
5	CST5	0	R/W	Counter Start 0 to 5 (CST0 to CST5) These bits select operation or stoppage for TCNT. If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_0 to TCNT_5 count operation is stopped 1: TCNT_0 to TCNT_5 performs count operation
4	CST4	0	R/W	
3	CST3	0	R/W	
2	CST2	0	R/W	
1	CST1	0	R/W	
0	CST0	0	R/W	

10.3.9 Timer Synchro Register (TSYR)

TSYR is an 8-bit readable/writable register that selects independent operation or synchronous operation for the channel 0 to 5 TCNT counters. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	R/W	Reserved The write value should always be 0.
5	SYNC5	0	R/W	Timer Synchro 0 to 5 These bits are used to select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by counter clearing on another channel, are possible. To set synchronous operation, the SYNC bits 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 CCLR0 to CCLR2 in TCR. 0: TCNT_0 to TCNT_5 operates independently (TCNT presetting /clearing is unrelated to other channels) 1: TCNT_0 to TCNT_5 performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible
4	SYNC4	0	R/W	
3	SYNC3	0	R/W	
2	SYNC2	0	R/W	
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

10.4 Operation

10.4.1 Basic Functions

Each channel has a TCNT and TGR register. 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 CST5 is set to 1 in TSTR, the TCNT counter for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

1. Example of count operation setting procedure
   Figure 10.2 shows an example of the count operation setting procedure.

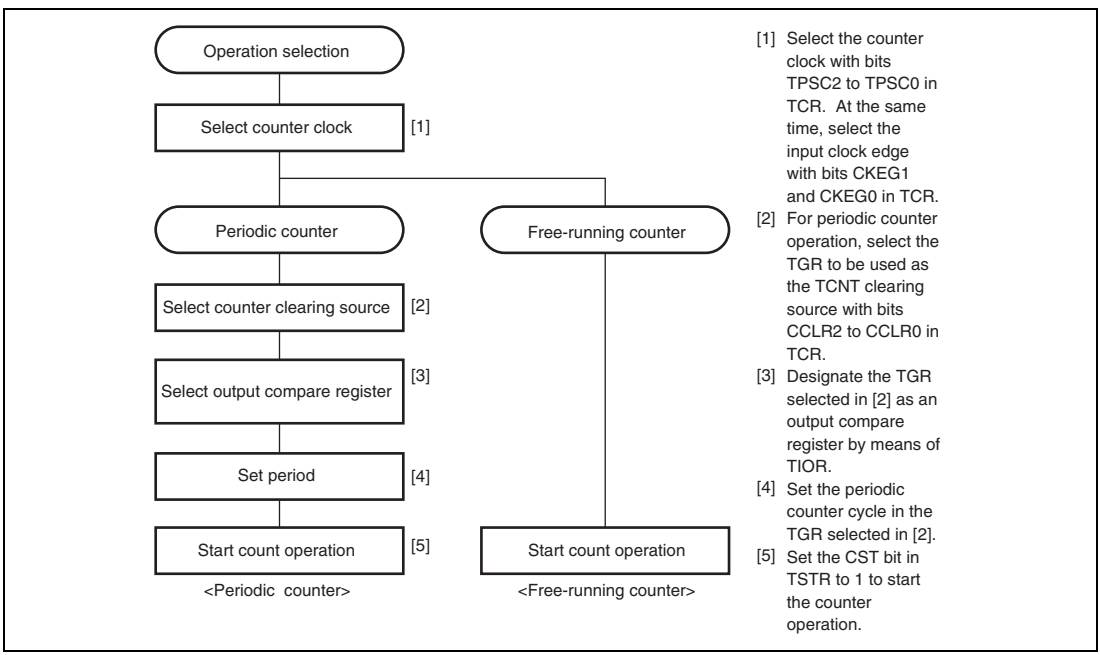


Figure 10.2 Example of Counter Operation Setting Procedure

2. 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 10.3 illustrates free-running counter operation.

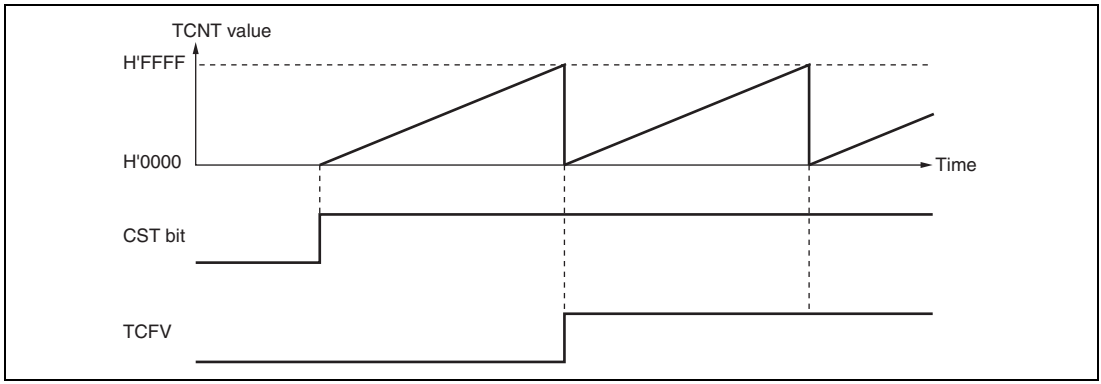


Figure 10.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR0 to CCLR2 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 10.4 illustrates periodic counter operation.

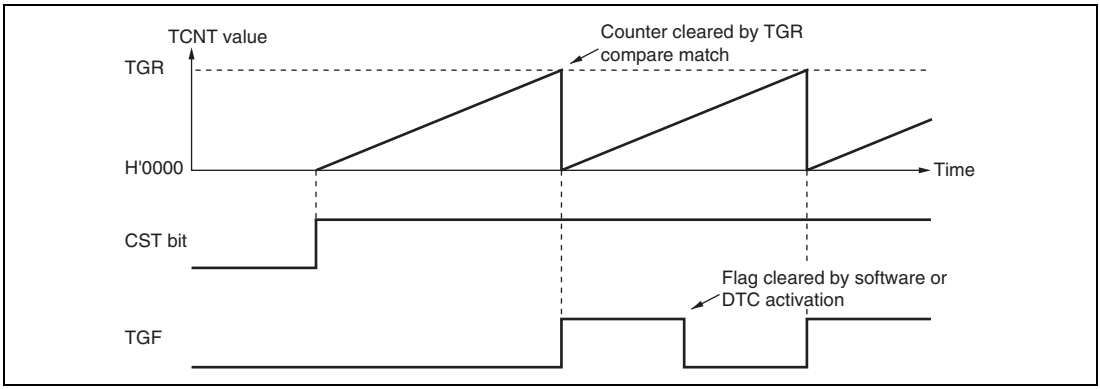


Figure 10.4 Periodic Counter Operation

Waveform Output by Compare Match: The TPU can perform 0, 1, or toggle output from the corresponding output pin using compare match.

1. Example of setting procedure for waveform output by compare match

   Figure 10.5 shows an example of the setting procedure for waveform output by compare match

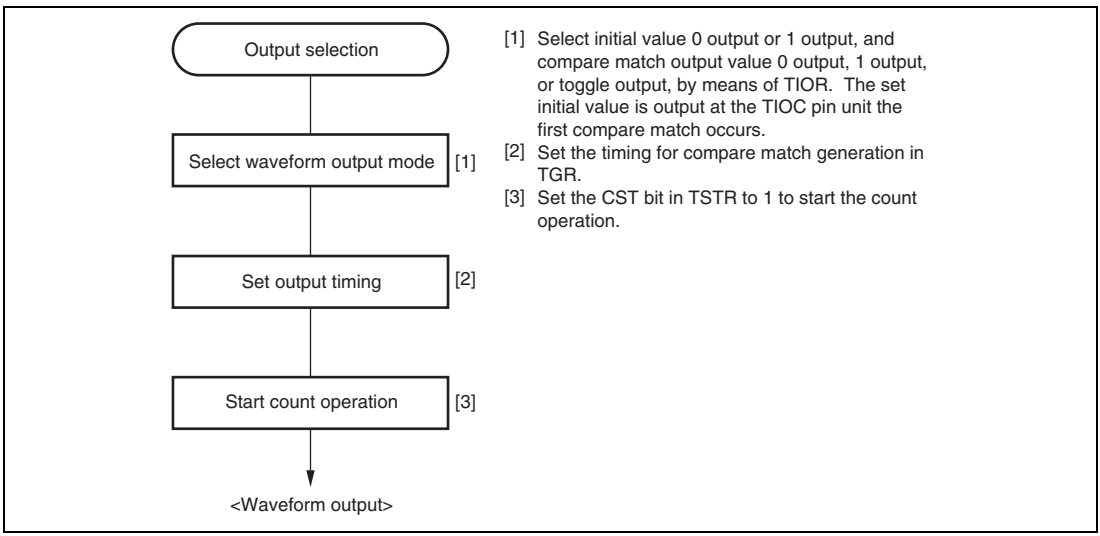


Figure 10.5 Example of Setting Procedure for Waveform Output by Compare Match

2. Examples of waveform output operation

Figure 10.6 shows an example of 0 output/1 output.

In this example TCNT has been designated as a free-running counter, and settings have been made such that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

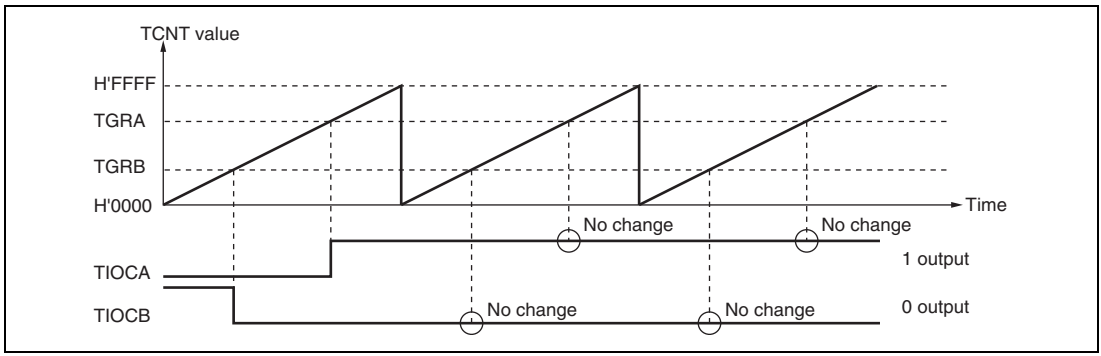


Figure 10.6 Example of 0 Output/1 Output Operation

Figure 10.7 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made such that the output is toggled by both compare match A and compare match B.

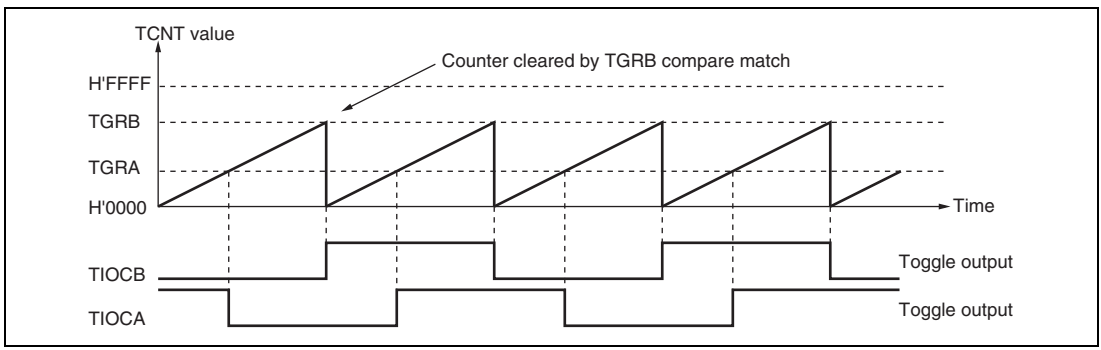


Figure 10.7 Example of Toggle Output Operation

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. For channels 0, 1, 3, and 4, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for channels 0 and 3, 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.

1. Example of input capture operation setting procedure
   Figure 10.8 shows an example of the input capture operation setting procedure.

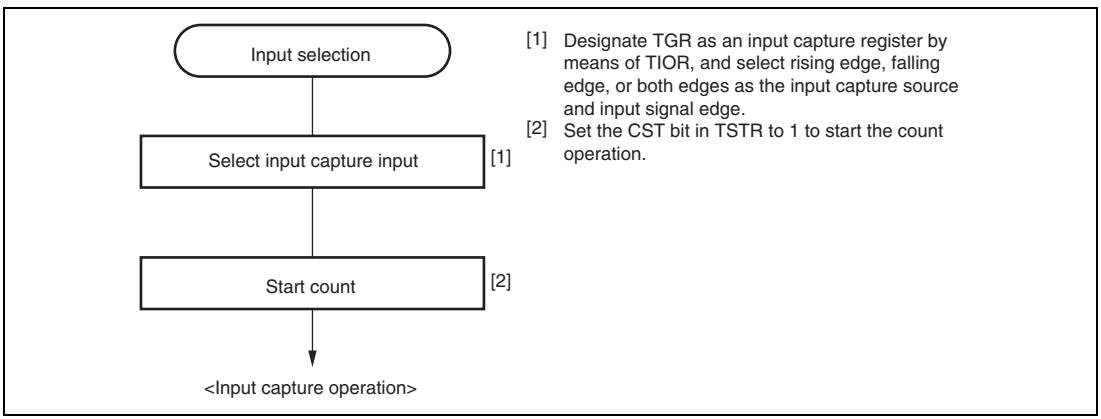


Figure 10.8 Example of Input Capture Operation Setting Procedure

2. Example of input capture operation

   Figure 10.9 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, the 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.

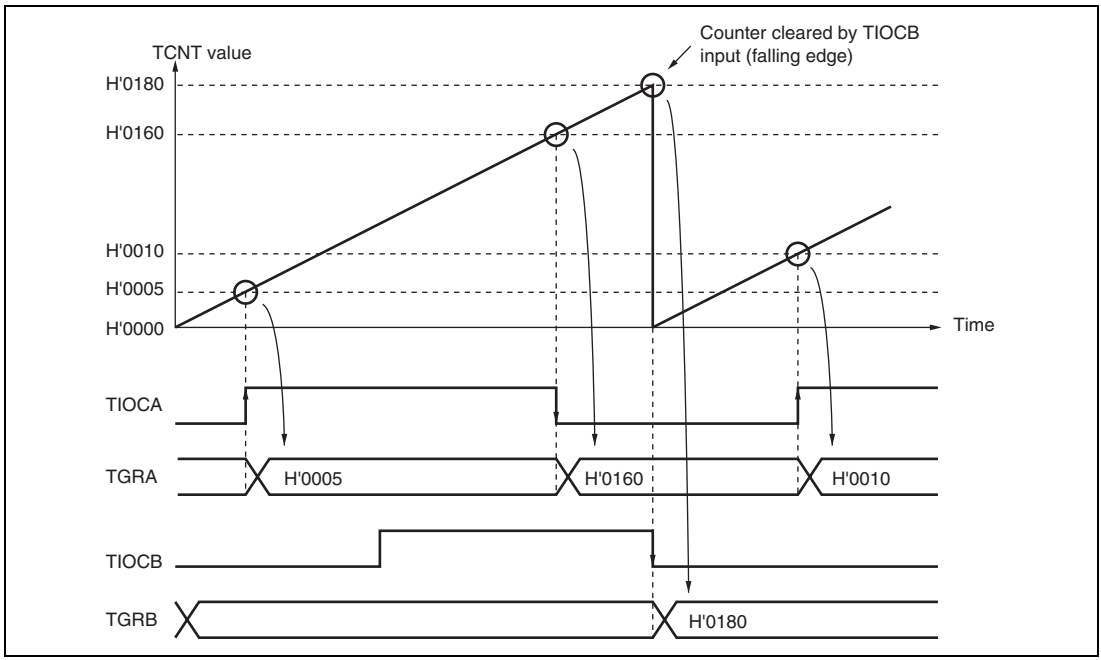


Figure 10.9 Example of Input Capture Operation

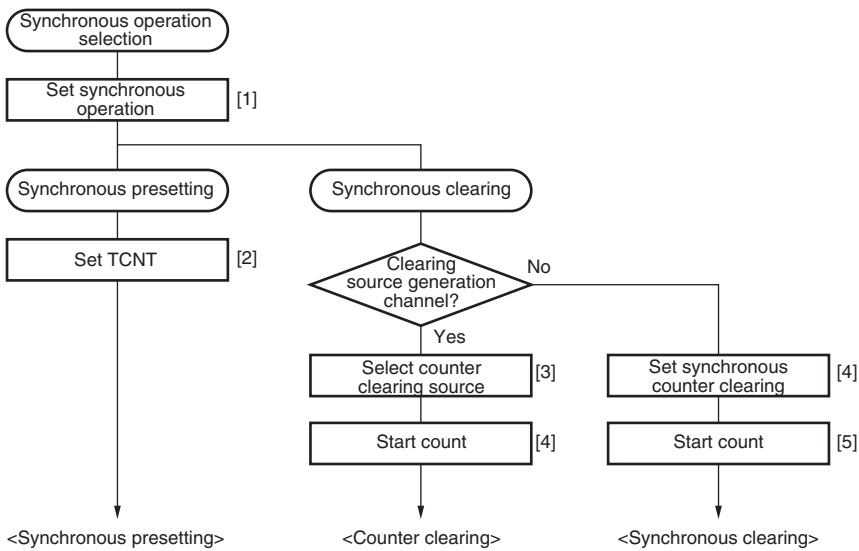
10.4.2 Synchronous Operation

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 5 can all be designated for synchronous operation.

Example of Synchronous Operation Setting Procedure: Figure 10.10 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.
[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
[3] Use bits 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 to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 10.10 Example of Synchronous Operation Setting Procedure

Example of Synchronous Operation: Figure 10.11 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 channel 0 to 2 TCNT counters, and the data set in TGRB_0 is used as the PWM cycle.

For details of PWM modes, see section 10.4.5, PWM Modes.

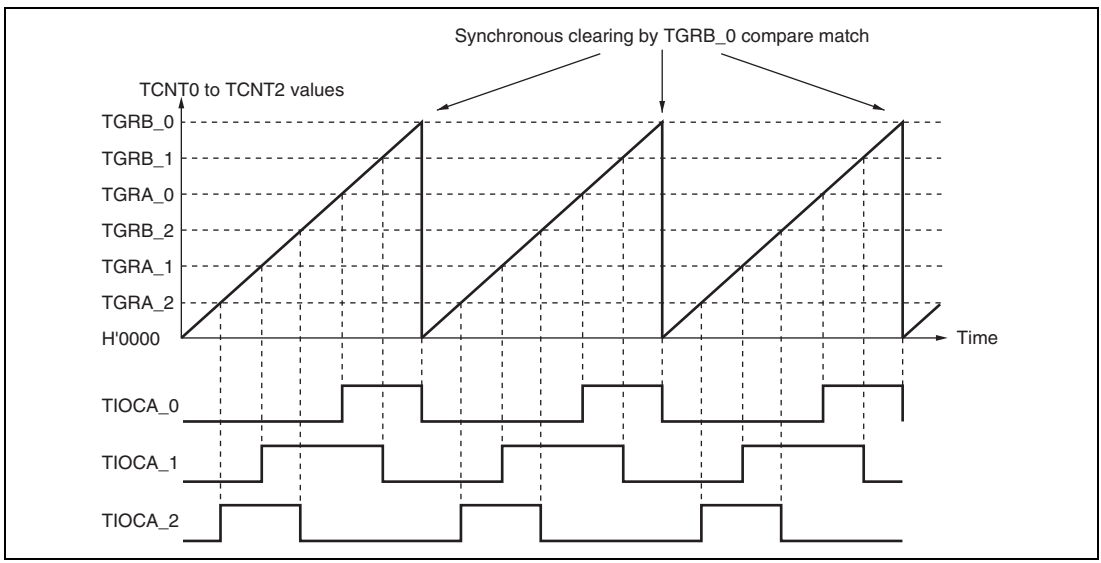


Figure 10.11 Example of Synchronous Operation

10.4.3 Buffer Operation

Buffer operation, provided for channels 0 and 3, 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 10.28 shows the register combinations used in buffer operation.

Table 10.28 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3

- 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 10.12.

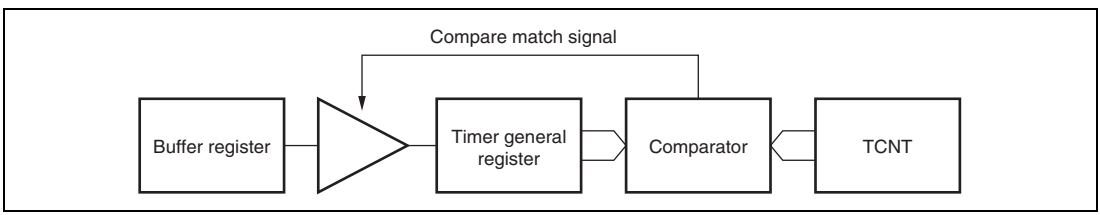


Figure 10.12 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.

  This operation is illustrated in figure 10.13.

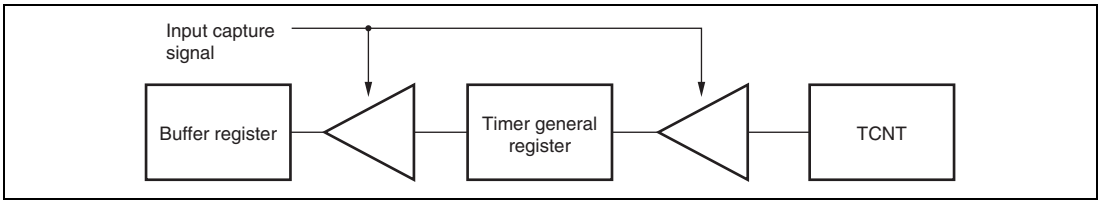


Figure 10.13 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 10.14 shows an example of the buffer operation setting procedure.

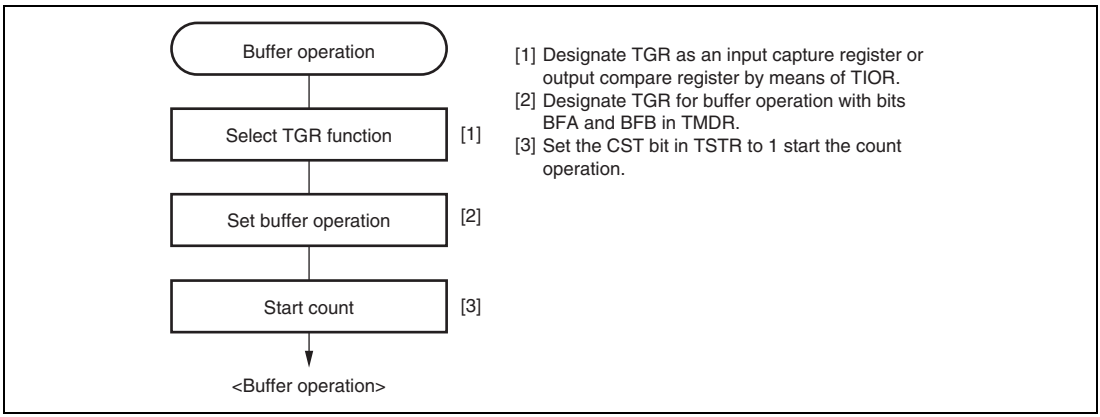


Figure 10.14 Example of Buffer Operation Setting Procedure

Examples of Buffer Operation:

1. When TGR is an output compare register

   Figure 10.15 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 that compare match A occurs.

   For details of PWM modes, see section 10.4.5, PWM Modes.

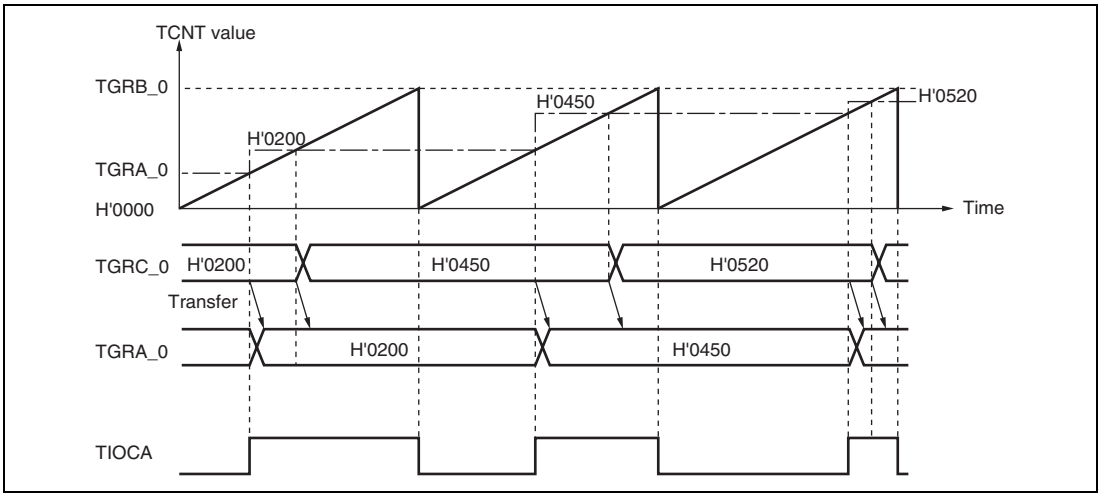


Figure 10.15 Example of Buffer Operation (1)

2. When TGR is an input capture register

   Figure 10.16 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 the occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

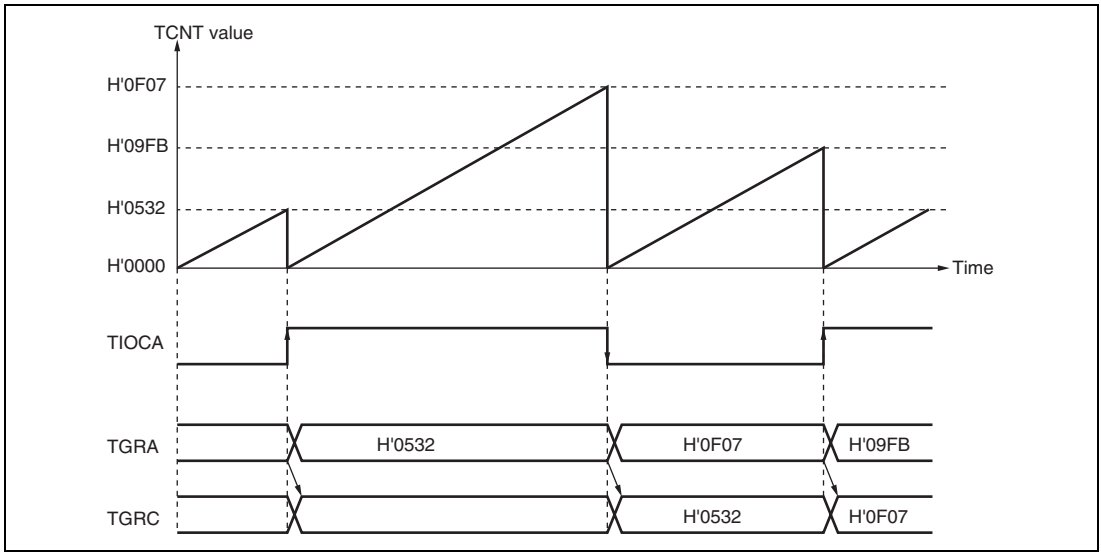


Figure 10.16 Example of Buffer Operation (2)

10.4.4 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 (channel 4) counter clock upon overflow/underflow of TCNT_2 (TCNT_5) as set in bits TPSC0 to TPSC2 in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase-counting mode.

Table 10.29 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1 or 4, the counter clock setting is invalid and the counters operates independently in phase counting mode.

Table 10.29 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2
Channels 4 and 5	TCNT_4	TCNT_5

Example of Cascaded Operation Setting Procedure: Figure 10.17 shows an example of the setting procedure for cascaded operation.

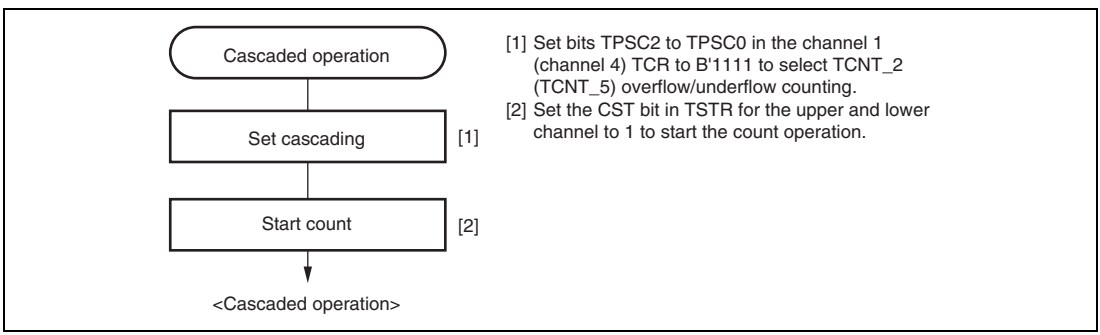


Figure 10.17 Cascaded Operation Setting Procedure

Examples of Cascaded Operation: Figure 10.18 illustrates the operation when TCNT_2 overflow/underflow counting has been set for TCNT_1, when TGRA_1 and TGRA_2 have been designated as input capture registers, and when 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.

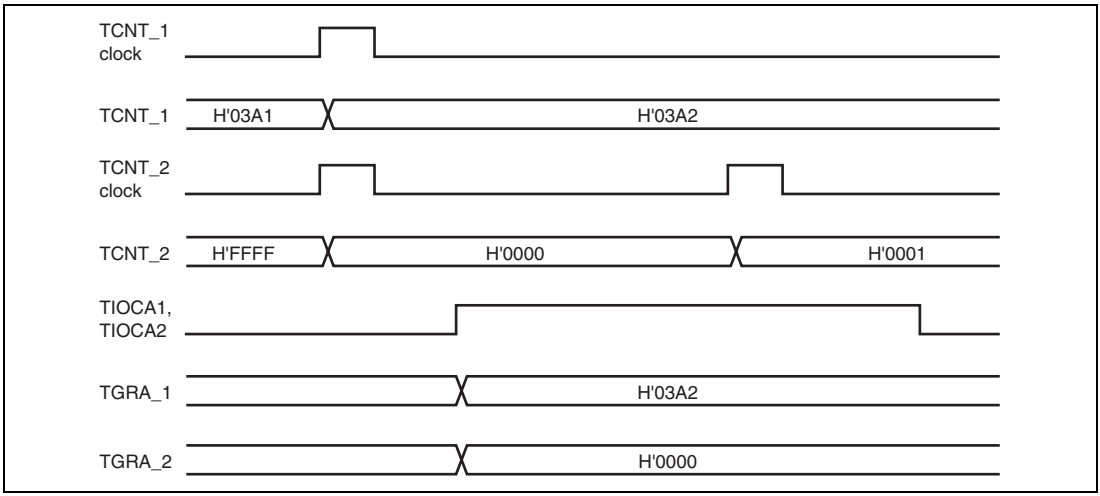


Figure 10.18 Example of Cascaded Operation (1)

Figure 10.19 illustrates the operation when TCNT_2 overflow/underflow counting 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.

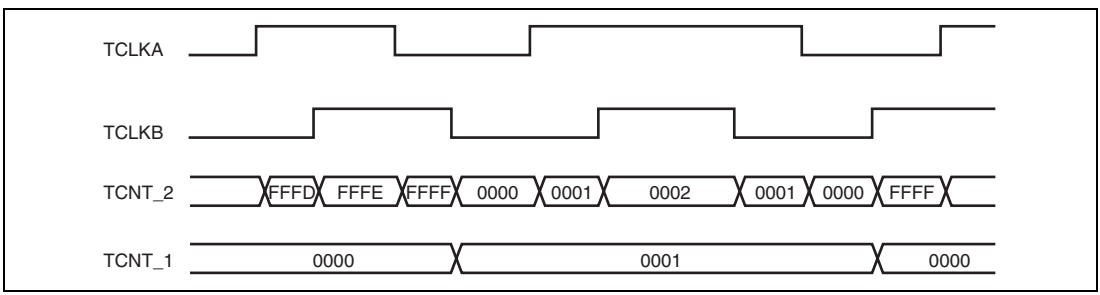


Figure 10.19 Example of Cascaded Operation (2)

10.4.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. The output level can be selected as 0, 1, or toggle output in response to a compare match of each TGR.

TGR registers settings can be used to output a PWM waveform in the range of 0% to 100% duty cycle.

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 output specified by bits IOA0 to IOA3 and IOC0 to IOC3 in TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and the output specified by bits IOB0 to IOB3 and IOD0 to IOD3 in TIOR is output at compare matches B and D. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs.

  In PWM mode 1, a maximum 8-phase PWM output is possible.

- PWM mode 2

  PWM output is generated using one TGR as the cycle register and the others as duty cycle registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a synchronization register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the cycle and duty cycle registers are identical, the output value does not change when a compare match occurs.

  In PWM mode 2, a maximum 15-phase PWM output is possible in combination use with synchronous operation.

The correspondence between PWM output pins and registers is shown in table 10.30.

Table 10.30 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
3	TGRA_3	TIOCA3	TIOCA3
	TGRB_3		TIOCB3
	TGRC_3	TIOCC3	TIOCC3
	TGRD_3		TIOCD3
4	TGR4A_4	TIOCA4	TIOCA4
	TGR4B_4		TIOCB4
5	TGRA_5	TIOCA5	TIOCA5
	TGRB_5		TIOCB5

Note: * In PWM mode 2, PWM output is not possible for the TGR register in which the period is set.

Example of PWM Mode Setting Procedure: Figure 10.20 shows an example of the PWM mode setting procedure.

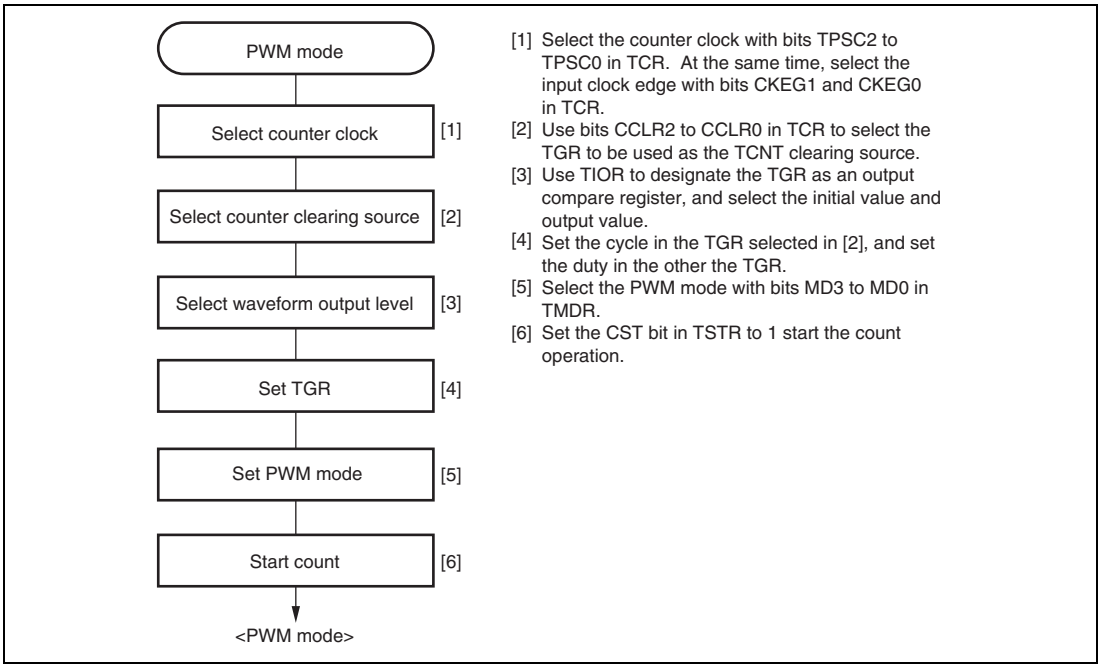


Figure 10.20 Example of PWM Mode Setting Procedure

Examples of PWM Mode Operation: Figure 10.21 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in the TGRB registers are used as the duty cycle levels.

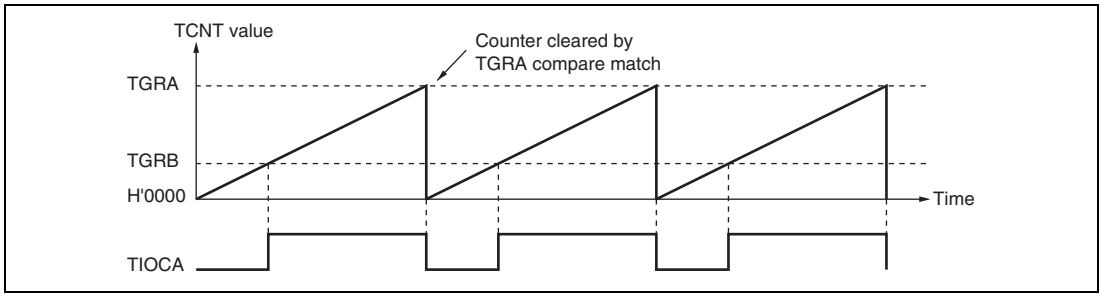


Figure 10.21 Example of PWM Mode Operation (1)

Figure 10.22 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), outputting a 5-phase PWM waveform.

In this case, the value set in TGRB_1 is used as the cycle, and the values set in the other TGRs are used as the duty cycle levels.

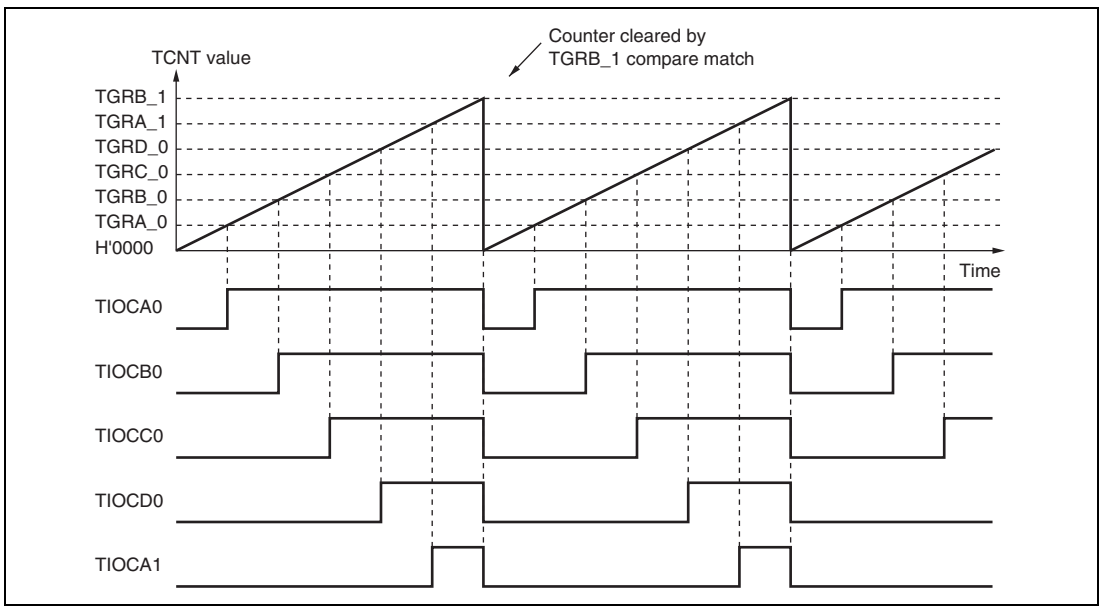


Figure 10.22 Example of PWM Mode Operation (2)

Figure 10.23 shows examples of PWM waveform output with 0% duty cycle and 100% duty cycle in PWM mode.

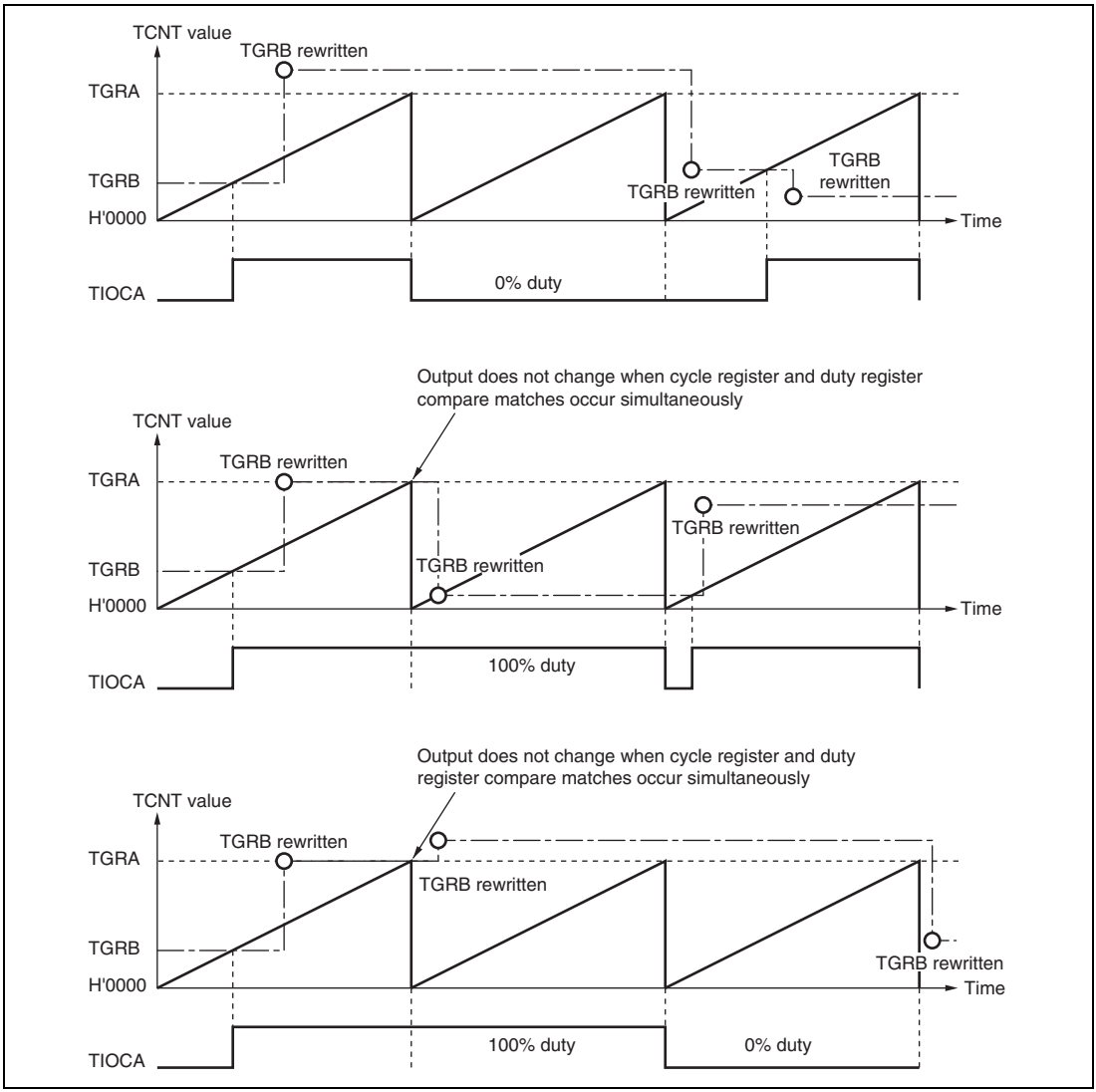


Figure 10.23 Example of PWM Mode Operation (3)

10.4.6 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, 2, 4, and 5.

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 setting of bits TPSC0 to TPSC2 and bits CKEG0 and CKEG1 in TCR. However, the functions of bits CCLR0 and CCLR1 in TCR, and of TIOR, TIER, and TGR, are valid, and input capture/compare match and interrupt functions can be used.

This can be used for two-phase encoder pulse input.

If overflow occurs when TCNT is counting up, the TCFV flag in TSR is set; if underflow occurs when TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag reveals whether TCNT is counting up or down.

Table 10.31 shows the correspondence between external clock pins and channels.

Table 10.31 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 or 5 is set to phase counting mode	TCLKA	TCLKB
When channel 2 or 4 is set to phase counting mode	TCLKC	TCLKD

Example of Phase Counting Mode Setting Procedure: Figure 10.24 shows an example of the phase counting mode setting procedure.

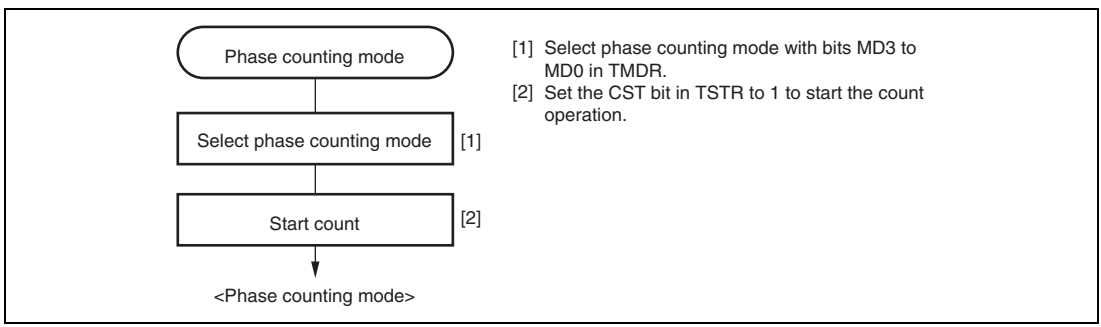


Figure 10.24 Example of Phase Counting Mode Setting Procedure

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.

1. Phase counting mode 1

   Figure 10.25 shows an example of phase counting mode 1 operation, and table 10.32 summarizes the TCNT up/down-count conditions.

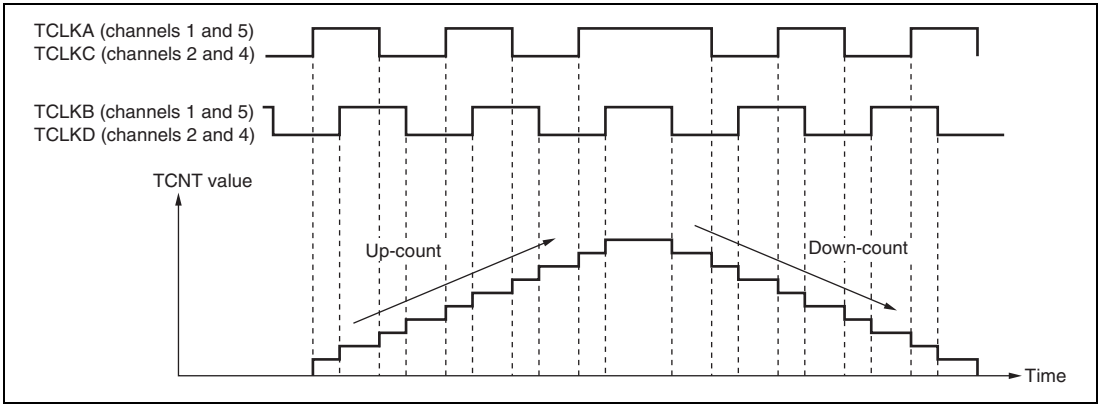


Figure 10.25 Example of Phase Counting Mode 1 Operation

Table 10.32 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	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

2. Phase counting mode 2

Figure 10.26 shows an example of phase counting mode 2 operation, and table 10.33 summarizes the TCNT up/down-count conditions.

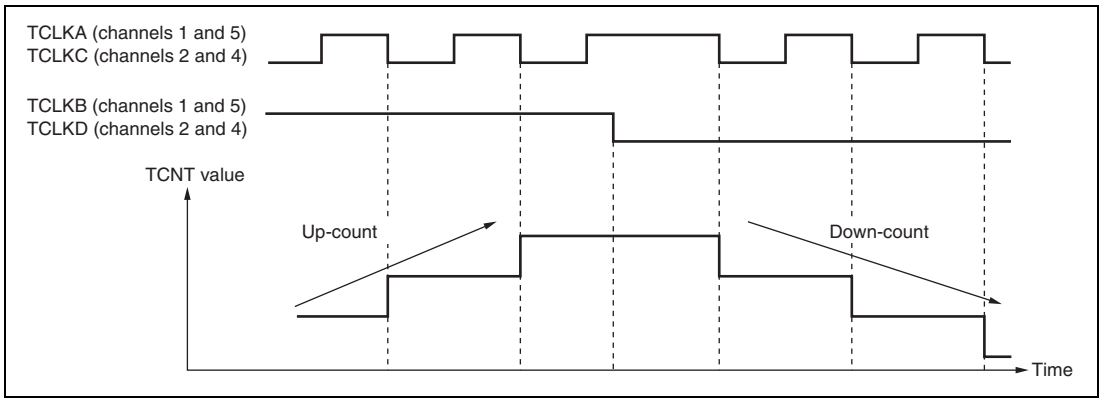


Figure 10.26 Example of Phase Counting Mode 2 Operation

Table 10.33 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	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

3. Phase counting mode 3

Figure 10.27 shows an example of phase counting mode 3 operation, and table 10.34 summarizes the TCNT up/down-count conditions.

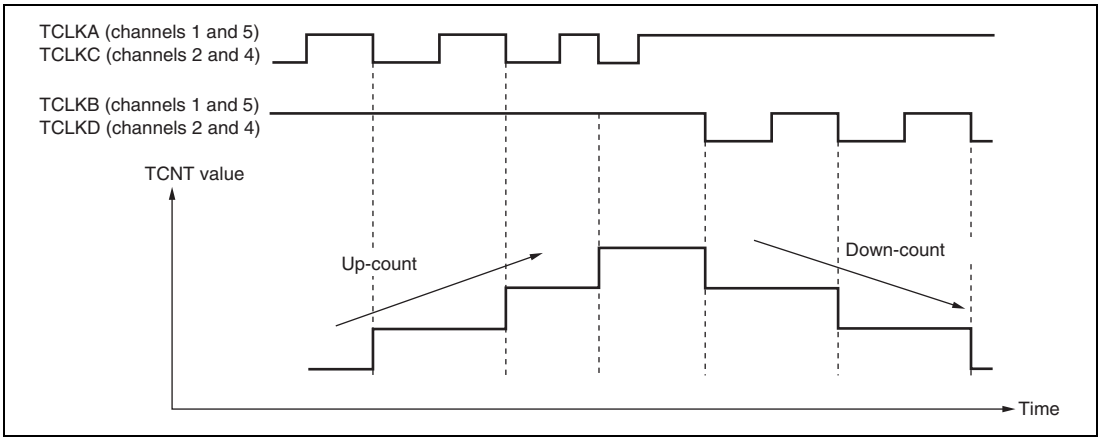


Figure 10.27 Example of Phase Counting Mode 3 Operation

Table 10.34 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	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

4. Phase counting mode 4

Figure 10.28 shows an example of phase counting mode 4 operation, and table 10.35 summarizes the TCNT up/down-count conditions.

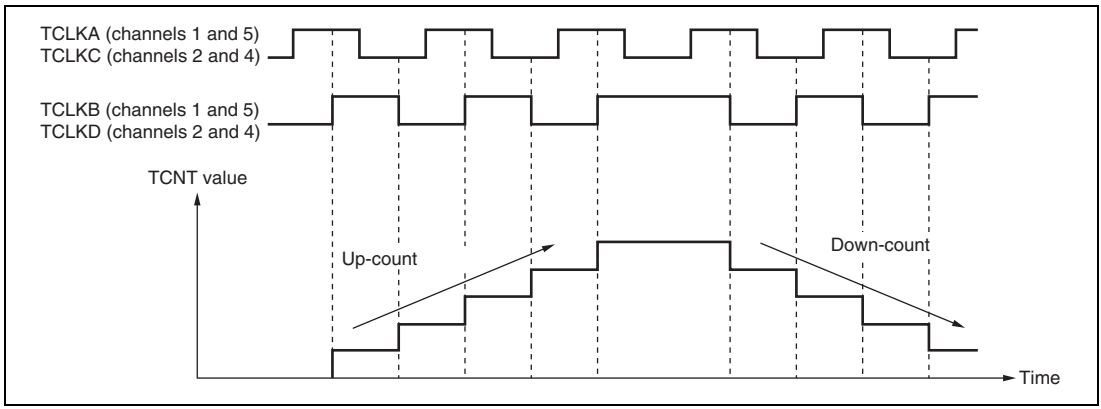


Figure 10.28 Example of Phase Counting Mode 4 Operation

Table 10.35 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	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

Phase Counting Mode Application Example: Figure 10.29 shows an example in which channel 1 is in phase counting mode, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect position or speed.

Channel 1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to TCLKA and TCLKB.

Channel 0 operates with TCNT counter clearing by TGRC_0 compare match; TGRA_0 and TGRC_0 are used for the compare match function and are set with the speed control period and position control period. TGRB_0 is used for input capture, with TGRB_0 and TGRD_0 operating in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input capture source, and the pulse widths of 2-phase encoder 4-multiplication pulses are detected.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, and channel 0 TGRA_0 and TGRC_0 compare matches are selected as the input capture source and store the up/down-counter values for the control periods.

This procedure enables the accurate detection of position and speed.

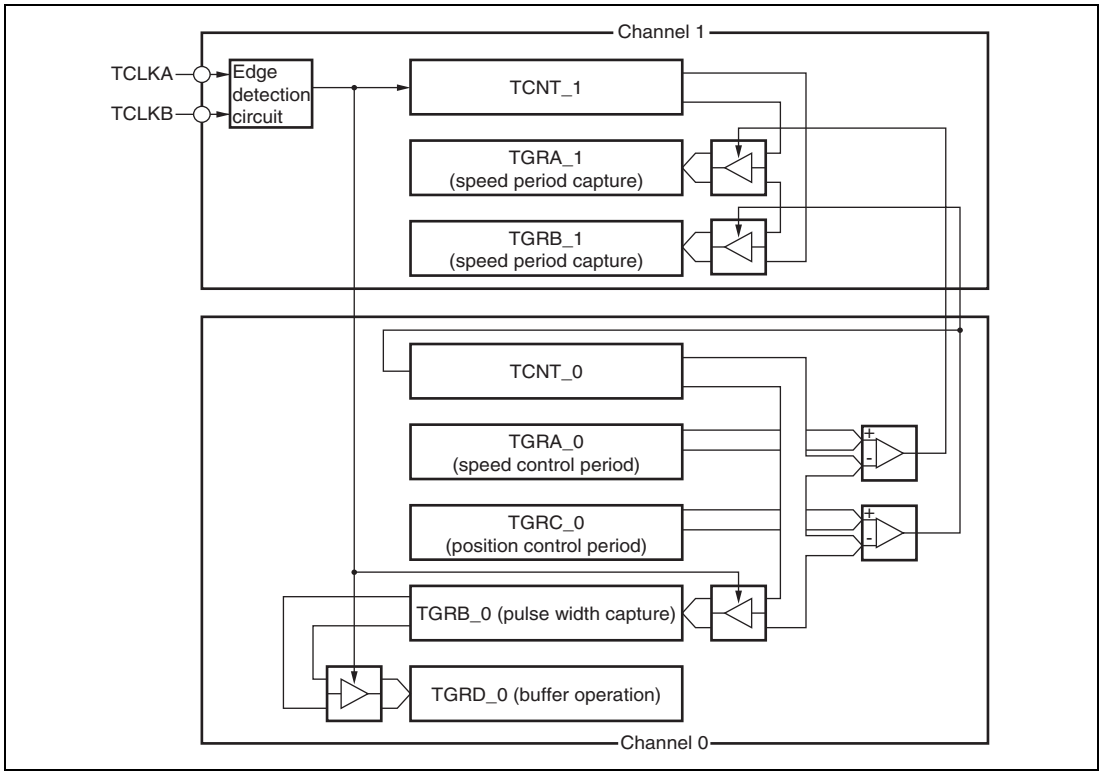


Figure 10.29 Phase Counting Mode Application Example

10.5 Interrupt Sources

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/disabled bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt request 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, however the priority order within a channel is fixed. For details, see section 5, Interrupt Controller.

Table 10.36 lists the TPU interrupt sources.

Table 10.36 TPU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation
0	TGIA_0	TGRA_0 input capture/compare match	TGFA_0	Possible
	TGIB_0	TGRB_0 input capture/compare match	TGFB_0	Possible
	TGIC_0	TGRC_0 input capture/compare match	TGFC_0	Possible
	TGID_0	TGRD_0 input capture/compare match	TGFD_0	Possible
	TCIV_0	TCNT_0 overflow	TCFV_0	Not possible
1	TGIA_1	TGRA_1 input capture/compare match	TGFA_1	Possible
	TGIB_1	TGRB_1 input capture/compare match	TGFB_1	Possible
	TCIV_1	TCNT_1 overflow	TCFV_1	Not possible
	TCIU_1	TCNT_1 underflow	TCFU_1	Not possible
2	TGIA_2	TGRA_2 input capture/compare match	TGFA_2	Possible
	TGIB_2	TGRB_2 input capture/compare match	TGFB_2	Possible
	TCIV_2	TCNT_2 overflow	TCFV_2	Not possible
	TCIU_2	TCNT_2 underflow	TCFU_2	Not possible
3	TGIA_3	TGRA_3 input capture/compare match	TGFA_3	Possible
	TGIB_3	TGRB_3 input capture/compare match	TGFB_3	Possible
	TGIC_3	TGRC_3 input capture/compare match	TGFC_3	Possible
	TGID_3	TGRD_3 input capture/compare match	TGFD_3	Possible
	TCIV_3	TCNT_3 overflow	TCFV_3	Not possible
4	TGIA_4	TGRA_4 input capture/compare match	TGFA_4	Possible
	TGIB_4	TGRB_4 input capture/compare match	TGFB_4	Possible
	TCIV_4	TCNT_4 overflow	TCFV_4	Not possible
	TCIU_4	TCNT_4 underflow	TCFU_4	Not possible
5	TGIA_5	TGRA_5 input capture/compare match	TGFA_5	Possible
	TGIB_5	TGRB_5 input capture/compare match	TGFB_5	Possible
	TCIV_5	TCNT_5 overflow	TCFV_5	Not possible
	TCIU_5	TCNT_5 underflow	TCFU_5	Not possible

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 16 input capture/compare match interrupts, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

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 six 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 four underflow interrupts, one each for channels 1, 2, 4, and 5.

10.6 DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt for a channel. For details, see section 8, Data Transfer Controller (DTC).

A total of 16 TPU input capture/compare match interrupts can be used as DTC activation sources, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

10.7 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 begin 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 begun.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

10.8 Operation Timing

10.8.1 Input/Output Timing

TCNT Count Timing: Figure 10.30 shows TCNT count timing in internal clock operation, and figure 10.31 shows TCNT count timing in external clock operation.

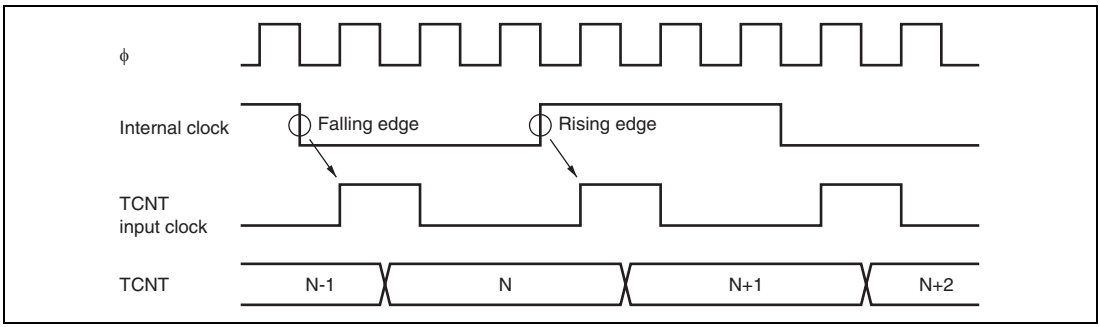


Figure 10.30 Count Timing in Internal Clock Operation

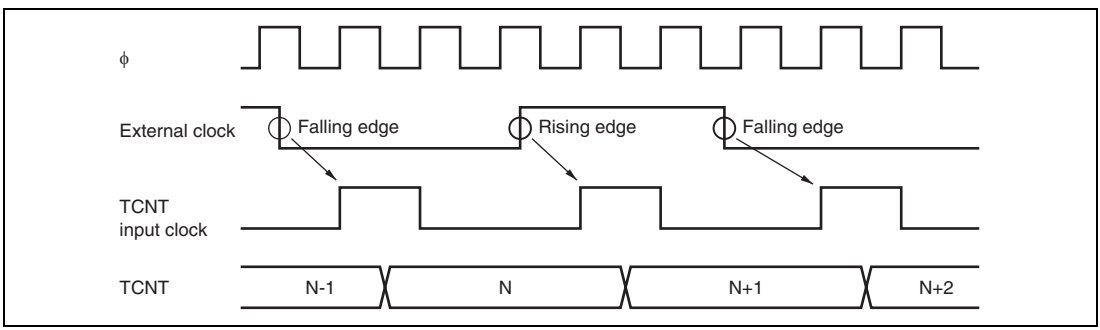


Figure 10.31 Count Timing in External Clock Operation

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 pin. After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 10.32 shows output compare output timing.

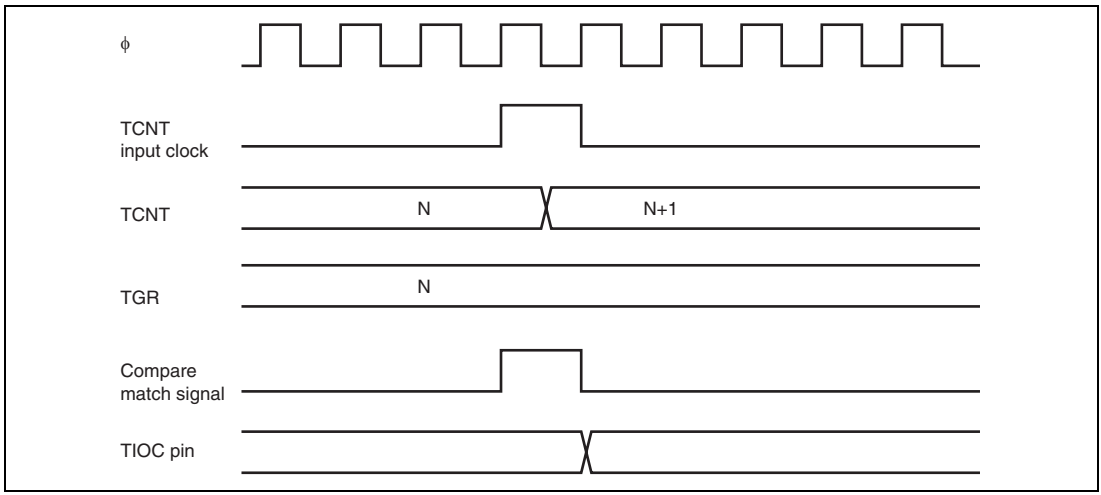


Figure 10.32 Output Compare Output Timing

Input Capture Signal Timing: Figure 10.33 shows input capture signal timing.

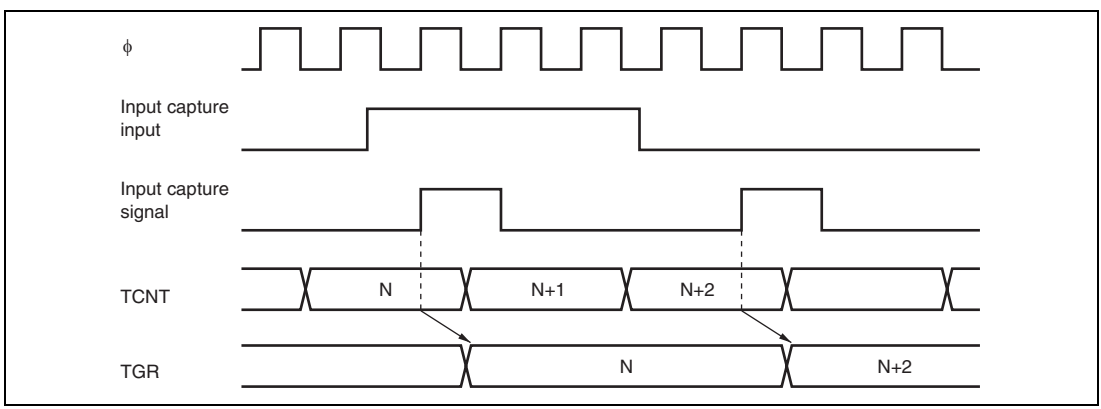


Figure 10.33 Input Capture Input Signal Timing

Timing for Counter Clearing by Compare Match/Input Capture: Figure 10.34 shows the timing when counter clearing on compare match is specified, and figure 10.35 shows the timing when counter clearing on input capture is specified.

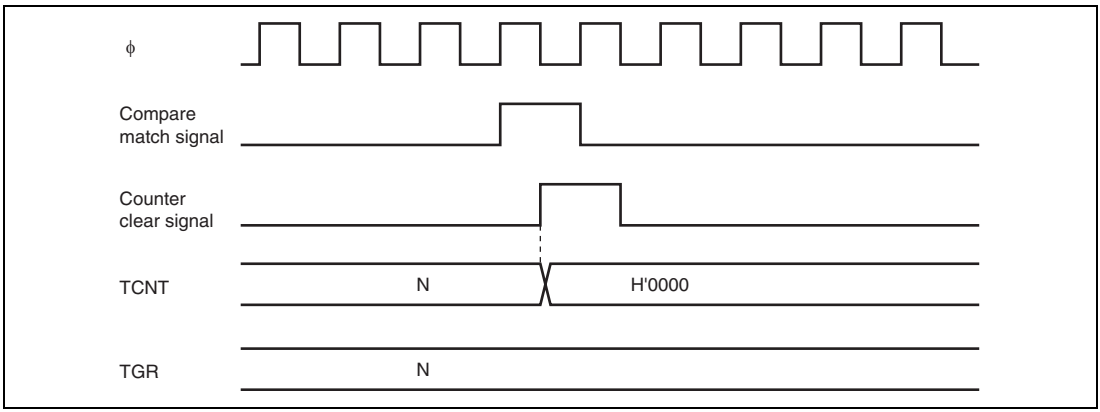


Figure 10.34 Counter Clear Timing (Compare Match)

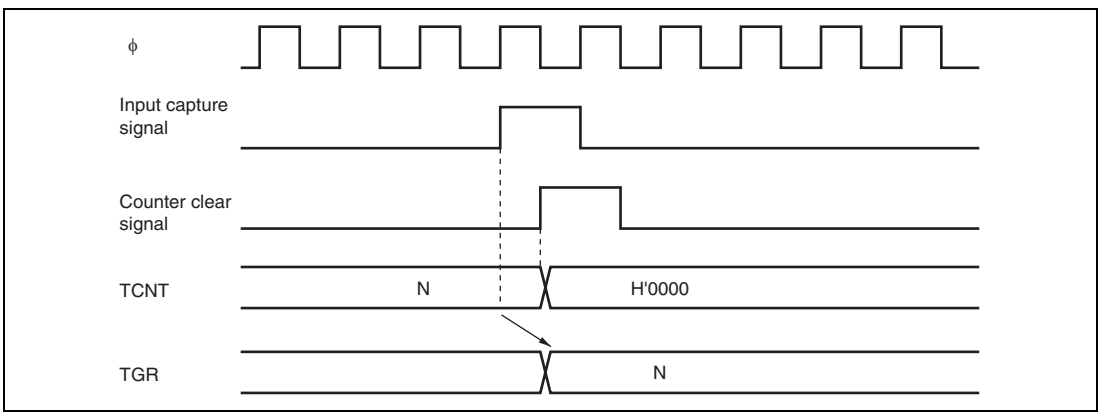


Figure 10.35 Counter Clear Timing (Input Capture)

Buffer Operation Timing: Figures 10.36 and 10.37 show the timing in buffer operation.

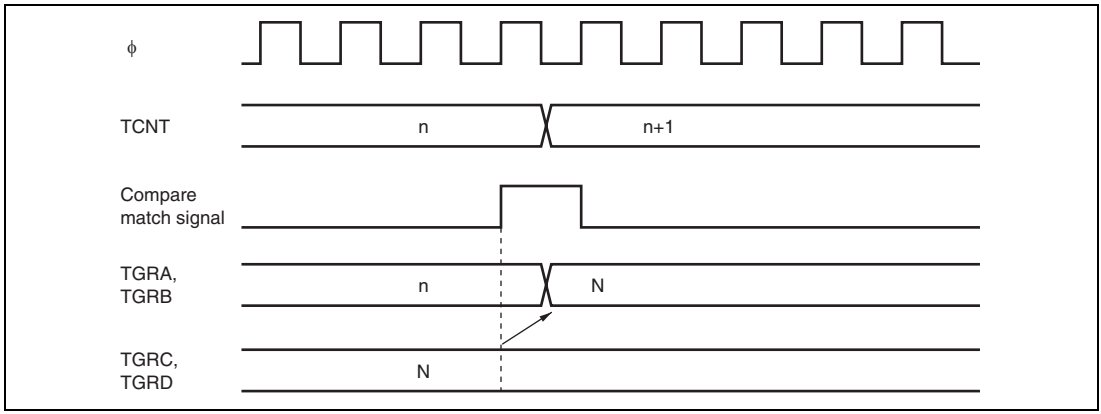


Figure 10.36 Buffer Operation Timing (Compare Match)

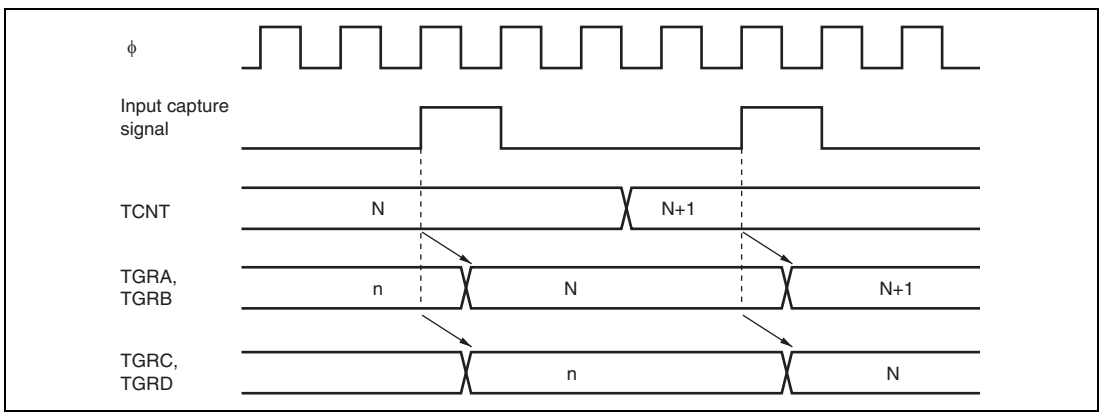


Figure 10.37 Buffer Operation Timing (Input Capture)

10.8.2 Interrupt Signal Timing

TGF Flag Setting Timing in Case of Compare Match: Figure 10.38 shows the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

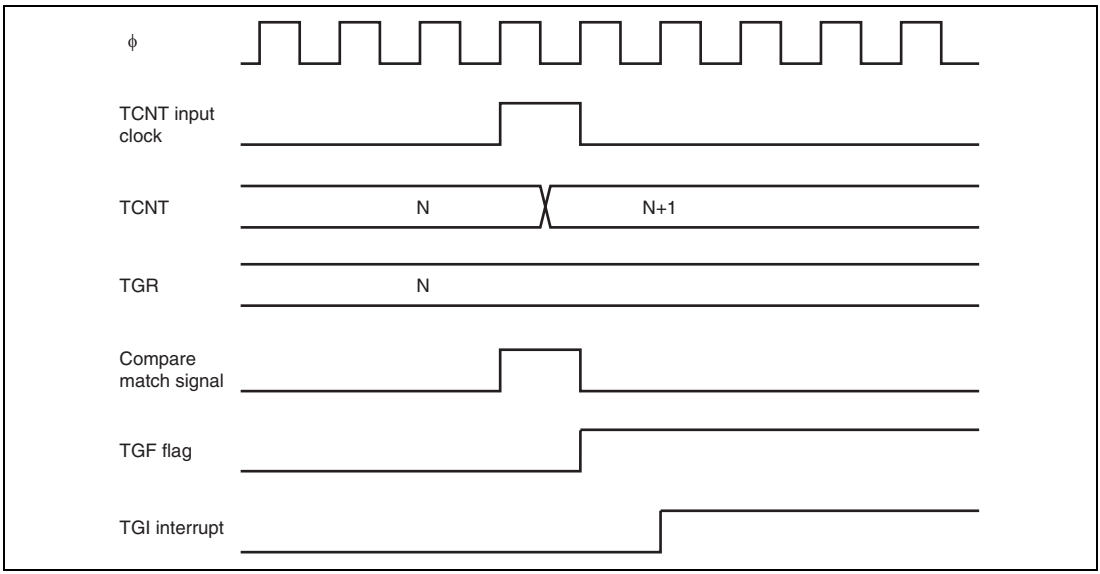


Figure 10.38 TGI Interrupt Timing (Compare Match)

TGF Flag Setting Timing in Case of Input Capture: Figure 10.39 shows the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

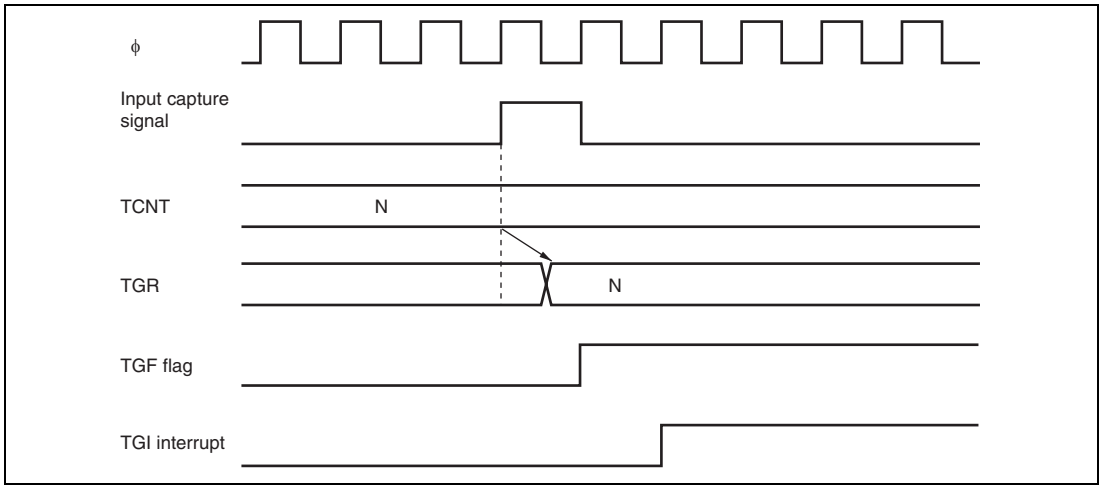


Figure 10.39 TGI Interrupt Timing (Input Capture)

TCFV Flag/TCFU Flag Setting Timing: Figure 10.40 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 10.41 shows the timing for setting of the TCFU flag in TSR on underflow, and TCIU interrupt request signal timing.

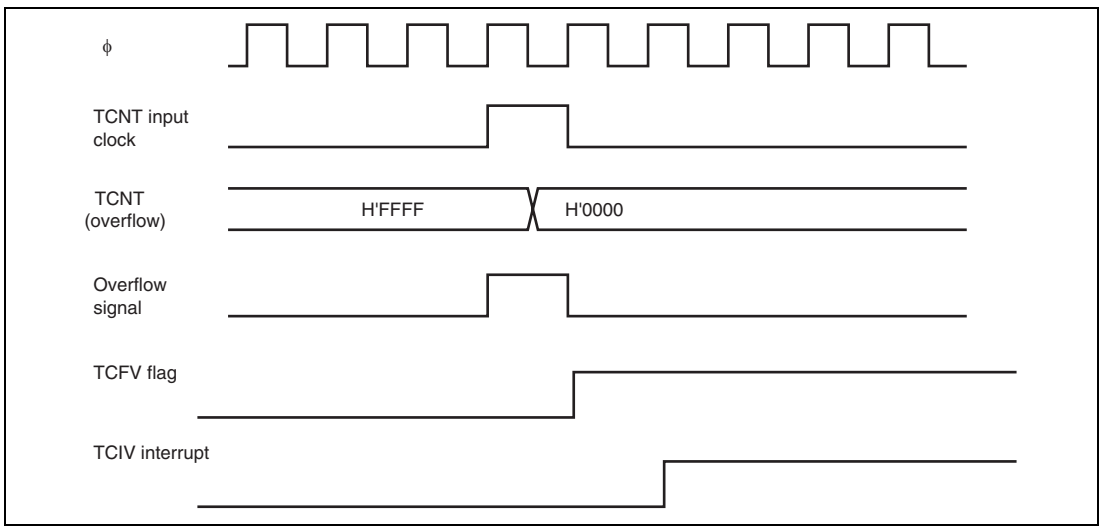


Figure 10.40 TCIV Interrupt Setting Timing

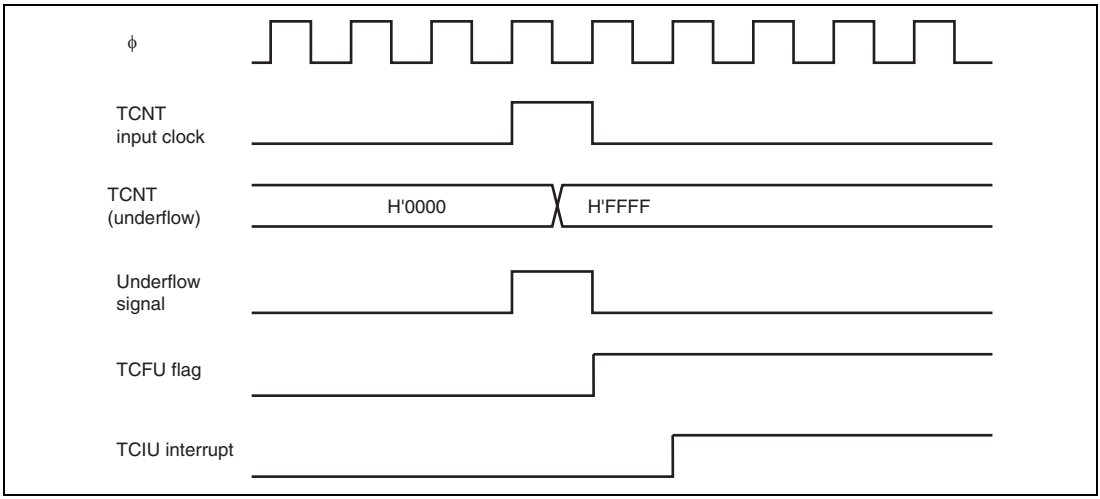


Figure 10.41 TCIU Interrupt Setting Timing

Status Flag Clearing Timing: After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DTC is activated, the flag is cleared automatically. Figure 10.42 shows the timing for status flag clearing by the CPU, and figure 10.43 shows the timing for status flag clearing by the DTC.

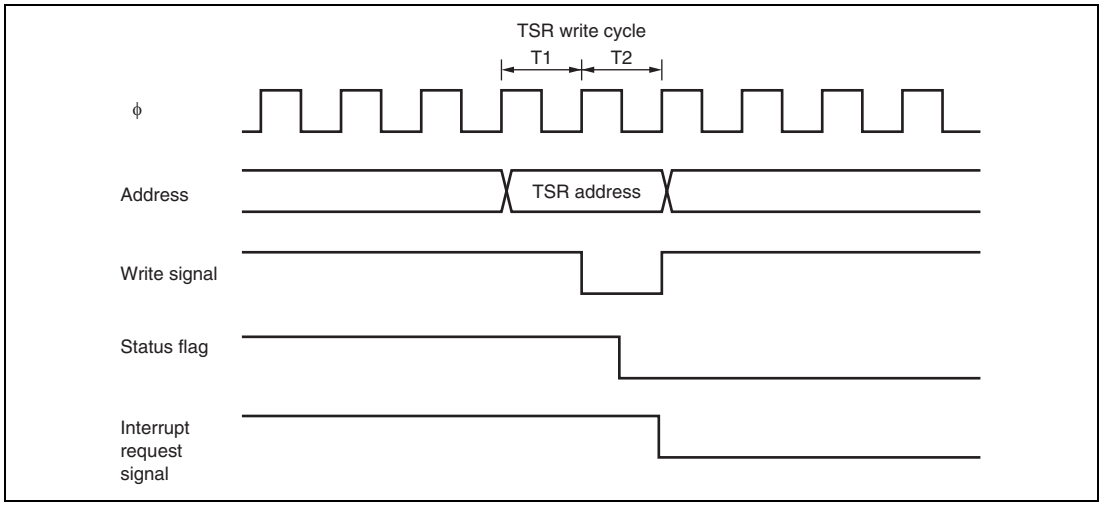


Figure 10.42 Timing for Status Flag Clearing by CPU

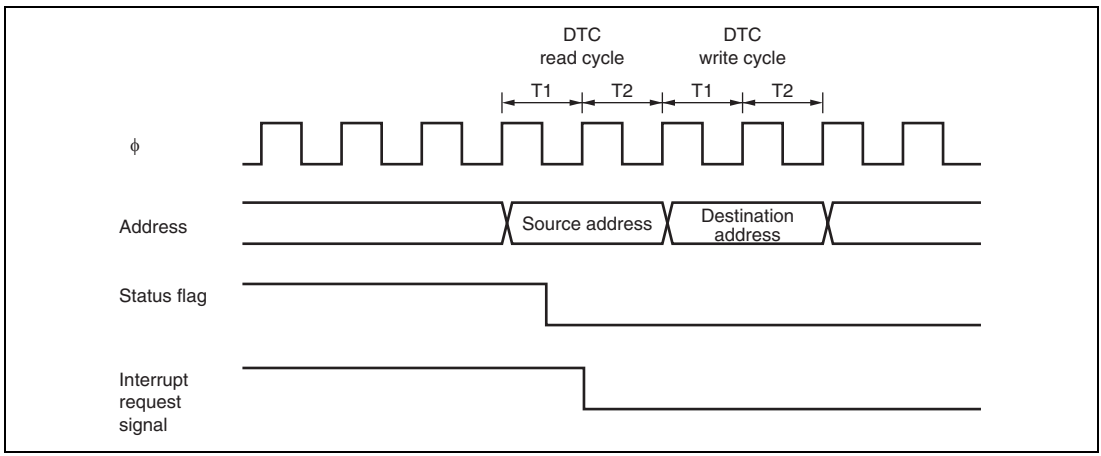


Figure 10.43 Timing for Status Flag Clearing by DTC Activation

10.9 Usage Notes

10.9.1 Module Stop Mode Setting

TPU operation can be disabled or enabled using the module stop control register. The initial setting is for TPU operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 21, Power-Down Modes.

10.9.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The TPU will not operate properly at narrower pulse widths.

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 10.44 shows the input clock conditions in phase counting mode.

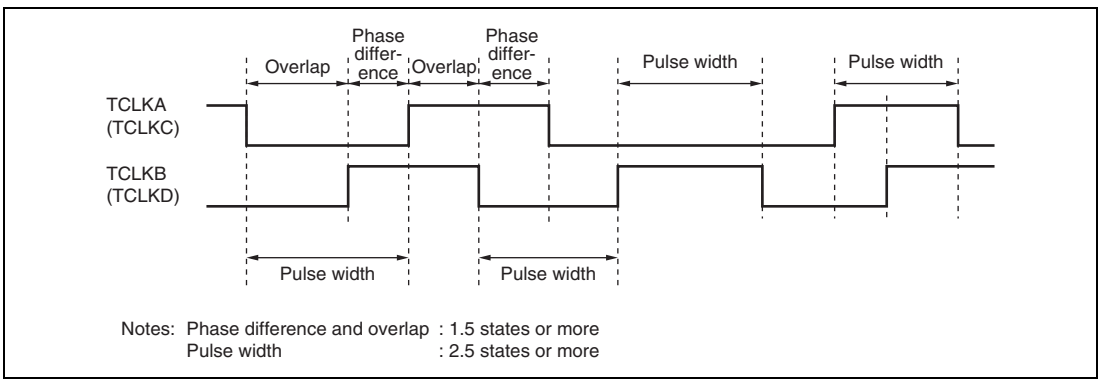


Figure 10.44 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

10.9.3 Caution on Period Setting

When counter clearing on 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

10.9.4 Conflict 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 precedence and the TCNT write is not performed.

Figure 10.45 shows the timing in this case.

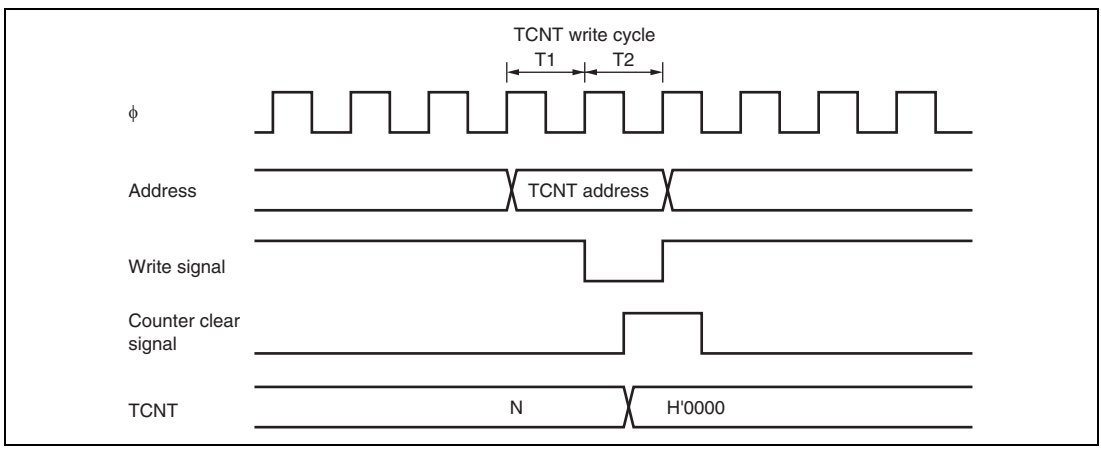


Figure 10.45 Conflict between TCNT Write and Clear Operations

10.9.5 Conflict between TCNT Write and Increment Operations

If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented.

Figure 10.46 shows the timing in this case.

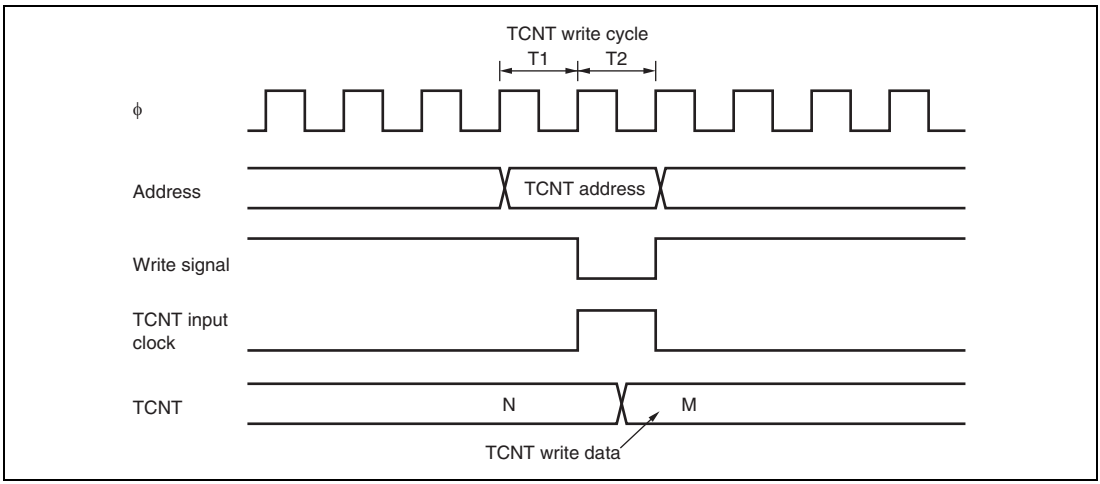


Figure 10.46 Conflict between TCNT Write and Increment Operations

10.9.6 Conflict between TGR Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the TGR write takes precedence and the compare match signal is inhibited. A compare match does not occur even if the previous value is written.

Figure 10.47 shows the timing in this case.

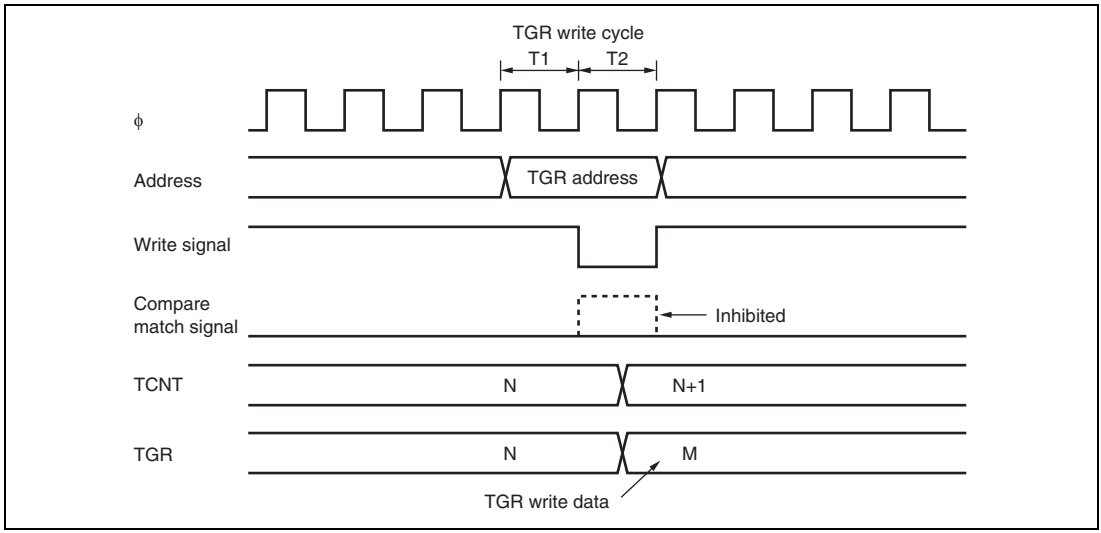


Figure 10.47 Conflict between TGR Write and Compare Match

10.9.7 Conflict between Buffer Register Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the data that is transferred to TGR by the buffer operation will be that in the buffer prior to the write.

Figure 10.48 shows the timing in this case.

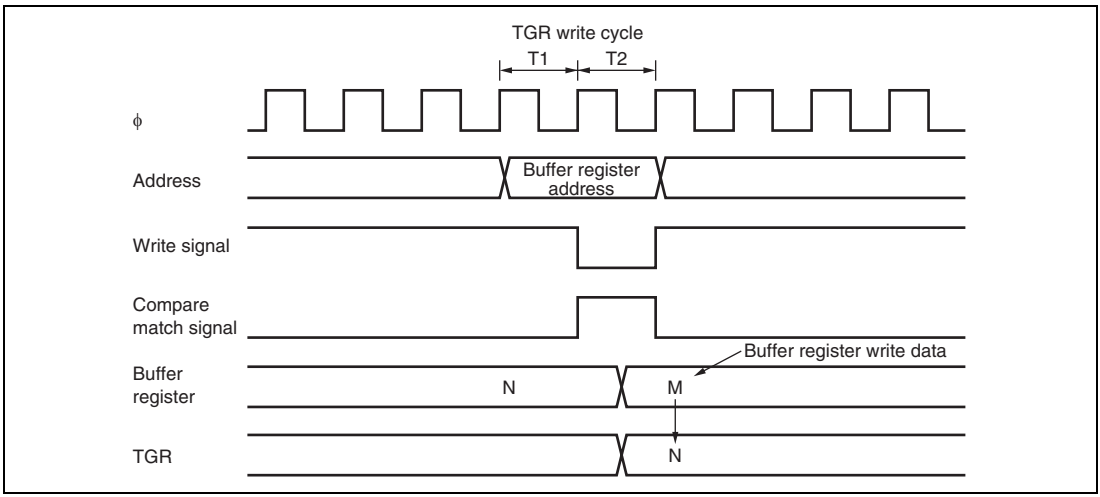


Figure 10.48 Conflict between Buffer Register Write and Compare Match

10.9.8 Conflict between TGR Read and Input Capture

If an input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be that in the buffer after input capture transfer.

Figure 10.49 shows the timing in this case.

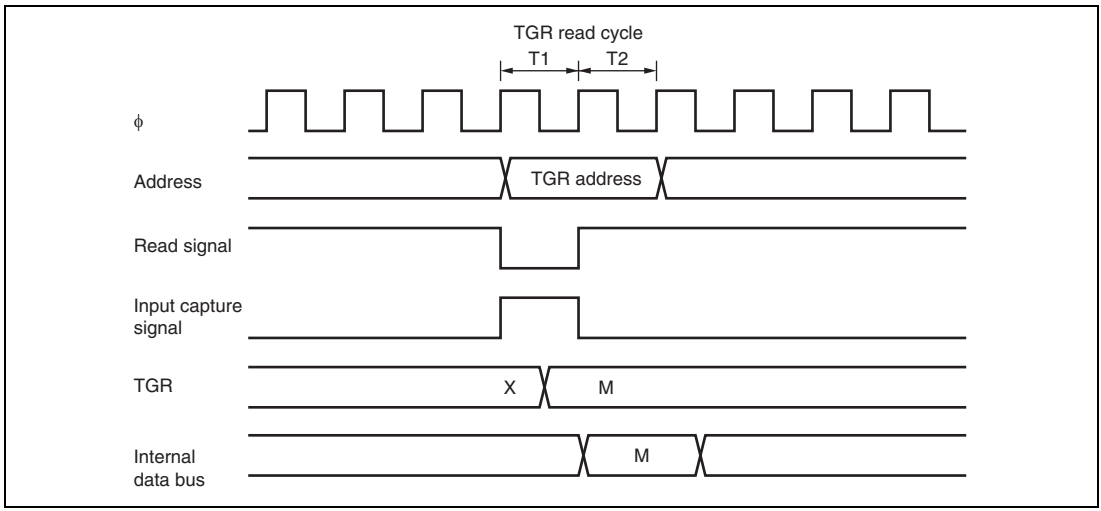


Figure 10.49 Conflict between TGR Read and Input Capture

10.9.9 Conflict between TGR Write and Input Capture

If an input capture signal is generated in the T2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed.

Figure 10.50 shows the timing in this case.

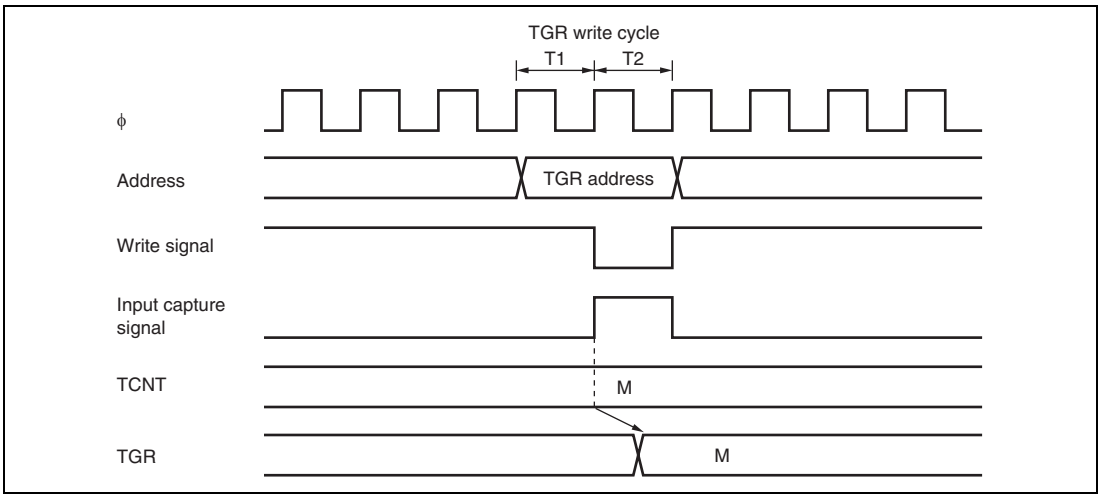


Figure 10.50 Conflict between TGR Write and Input Capture

10.9.10 Conflict between Buffer Register Write and Input Capture

If an input capture signal is generated in the T2 state of a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed.

Figure 10.51 shows the timing in this case.

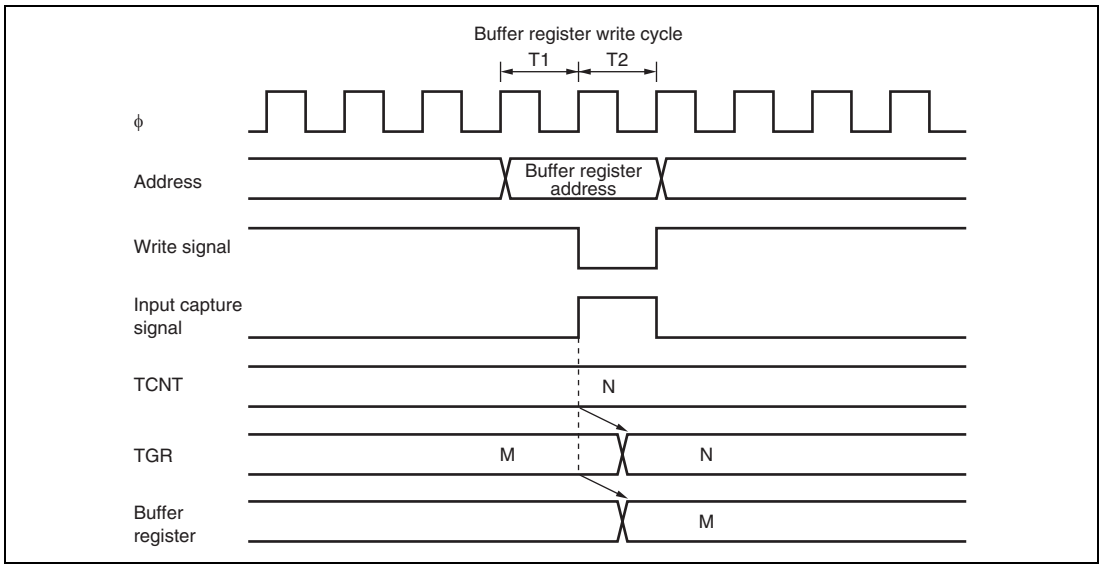


Figure 10.51 Conflict between Buffer Register Write and Input Capture

10.9.11 Conflict 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 precedence.

Figure 10.52 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

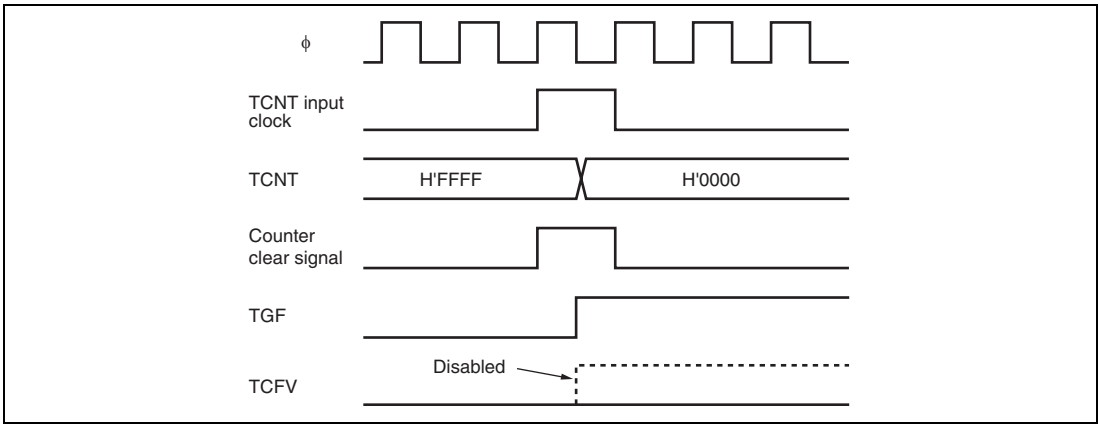


Figure 10.52 Conflict between Overflow and Counter Clearing

10.9.12 Conflict 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 precedence and the TCFV/TCFU flag in TSR is not set.

Figure 10.53 shows the operation timing when there is conflict between TCNT write and overflow.

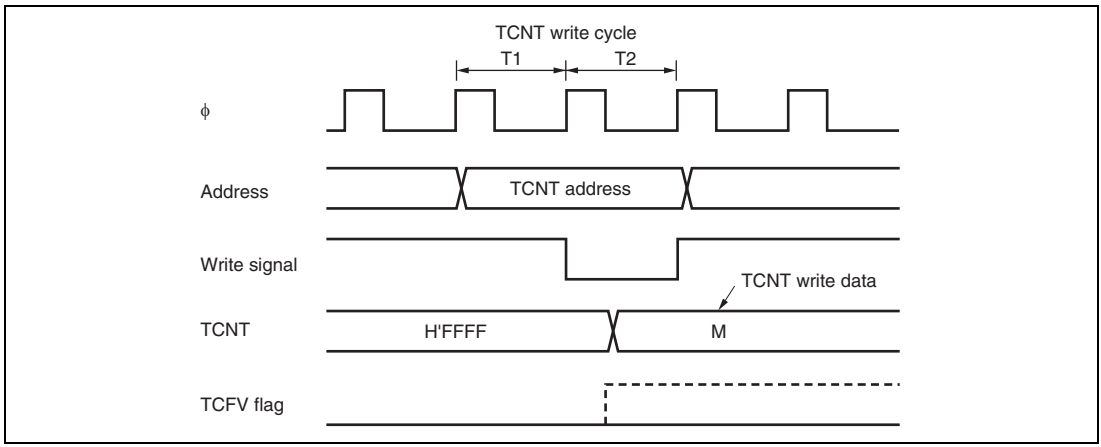


Figure 10.53 Conflict between TCNT Write and Overflow

10.9.13 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.

10.9.14 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 or the DTC activation source. Interrupts should therefore be disabled before entering module stop mode.

Section 11 Programmable Pulse Generator (PPG)

The programmable pulse generator provides pulse outputs using the 16-bit timer pulse unit (TPU) as a time base. The PPG pulse outputs are divided into 4-bit groups (group 2 and group 3) that can operate both simultaneously and independently. The block diagram of the PPG is shown in figure 11.1.

11.1 Features

- 8-bit output data
- Two output groups
- Selectable output trigger signals
- Non-overlap mode
- Can operate in tandem with the data transfer controller (DTC)
- Settable inverted output
- Module stop mode can be set

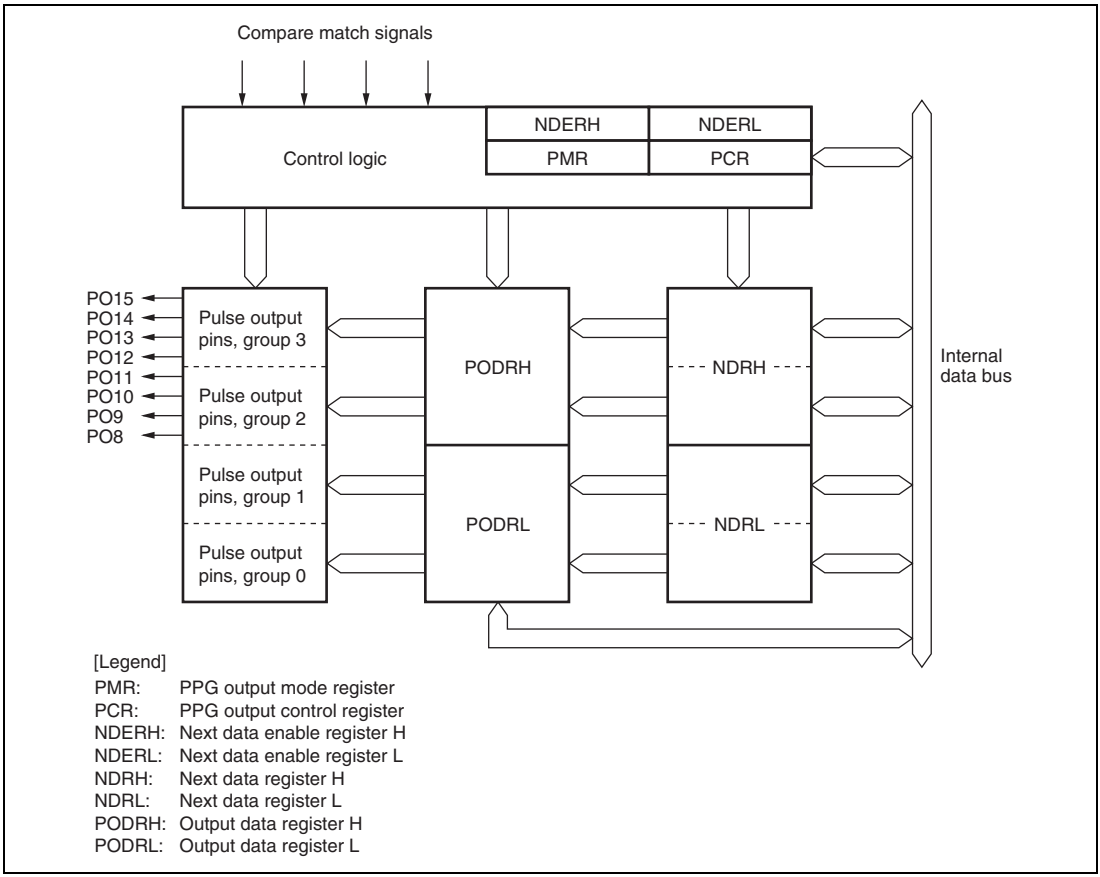


Figure 11.1 Block Diagram of PPG

11.2 Input/Output Pins

Table 11.1 summarizes the pin configuration of the PPG.

Table 11.1 Pin Configuration

Pin Name	I/O	Function
PO15	Output	Group 3 pulse output
PO14	Output	
PO13	Output	
PO12	Output	
PO11	Output	Group 2 pulse output
PO10	Output	
PO9	Output	
PO8	Output	

11.3 Register Descriptions

The PPG has the following registers.

- PPG output control register (PCR)
- PPG output mode register (PMR)
- Next data enable register H (NDERH)
- Next data enable register L (NDERL)
- Output data register H (PODRH)
- Output data register L (PODRL)
- Next data register H (NDRH)
- Next data register L (NDRL)

11.3.1 Next Data Enable Registers H, L (NDERH, NDERL)

NDERH and NDERL are 8-bit readable/writable registers that enable or disable pulse output on a bit-by-bit basis. The corresponding DDR also needs to be set to 1 in order to enable pulse output by the PPG.

- NDERH

Bit	Bit Name	Initial Value	R/W	Description
7	NDER15	0	R/W	Next Data Enable 8 to 15
6	NDER14	0	R/W	When a bit is set to 1 for pulse output by NDRH, the value in the corresponding NDRH bit is transferred to the PODRH bit by the selected output trigger. Values are not transferred from NDRH to PODRH for cleared bits.
5	NDER13	0	R/W	
4	NDER12	0	R/W	
3	NDER11	0	R/W	
2	NDER10	0	R/W	
1	NDER9	0	R/W	
0	NDER8	0	R/W	

- NDERL

Bit	Bit Name	Initial Value	R/W	Description
7	NDER7	0	R/W	Next Data Enable 0 to 7
6	NDER6	0	R/W	When a bit is set to 1 for pulse output by NDRL, the value in the corresponding NDRL bit is transferred to the PODRL bit by the selected output trigger. Values are not transferred from NDRL to PODRL for cleared bits.
5	NDER5	0	R/W	
4	NDER4	0	R/W	
3	NDER3	0	R/W	
2	NDER2	0	R/W	
1	NDER1	0	R/W	
0	NDER0	0	R/W	

11.3.2 Output Data Registers H, L (PODRH, PODRL)

PODRH and PODRL are 8-bit readable/writable registers that store output data for use in pulse output. A bit that has been set for pulse output by NDER is read-only and cannot be modified.

- PODRH

Bit	Bit Name	Initial Value	R/W	Description
7	POD15	0	R/W	Output Data Register 8 to 15
6	POD14	0	R/W	For bits that have been set to pulse output by NDERH, the output trigger transfers NDRH values to this register during PPG operation. While NDERH is set to 1, the CPU cannot write to this register. While NDERH is cleared, the initial output value of the pulse can be set.
5	POD13	0	R/W	
4	POD12	0	R/W	
3	POD11	0	R/W	
2	POD10	0	R/W	
1	POD9	0	R/W	
0	POD8	0	R/W	

- PODRL

Bit	Bit Name	Initial Value	R/W	Description
7	POD7	0	R/W	Output Data Register 0 to 7
6	POD6	0	R/W	For bits that have been set to pulse output by NDERL, the output trigger transfers NDRL values to this register during PPG operation. While NDERL is set to 1, the CPU cannot write to this register. While NDERL is cleared, the initial output value of the pulse can be set.
5	POD5	0	R/W	
4	POD4	0	R/W	
3	POD3	0	R/W	
2	POD2	0	R/W	
1	POD1	0	R/W	
0	POD0	0	R/W	

11.3.3 Next Data Registers H, L (NDRH, NDRL)

NDRH and NDRL are 8-bit readable/writable registers that store the data for the next pulse output. The NDR addresses differ depending on whether pulse output groups have the same output trigger or different output triggers.

- NDRH

If pulse output groups 2 and 3 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time, as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 8 to 15
6	NDR14	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3	NDR11	0	R/W	
2	NDR10	0	R/W	
1	NDR9	0	R/W	
0	NDR8	0	R/W	

If pulse output groups 2 and output pulse groups 3 have different output triggers, the upper 4 bits and the lower 4 bits are mapped to different addresses, as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 12 to 15
6	NDR14	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
3	NDR11	0	R/W	Next Data Register 8 to11 The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
2	NDR10	0	R/W	
1	NDR9	0	R/W	
0	NDR8	0	R/W	

- NDRL

If pulse output groups 0 and 1 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time, as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 0 to 7 The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
6	NDR6	0	R/W	
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3	NDR3	0	R/W	
2	NDR2	0	R/W	
1	NDR1	0	R/W	
0	NDR0	0	R/W	

If pulse output groups 0 and output pulse groups 1 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 4 to 7
6	NDR6	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
3	NDR3	0	R/W	Next Data Register 3 to 0
2	NDR2	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
1	NDR1	0	R/W	
0	NDR0	0	R/W	

11.3.4 PPG Output Control Register (PCR)

PCR is an 8-bit readable/writable register that selects output trigger signals on a group-by-group basis. For details on output trigger selection, see section 11.3.5, PPG Output Mode Register (PMR).

Bit	Bit Name	Initial Value	R/W	Description
7	G3CMS1	1	R/W	Group 3 Compare Match Select 0 and 1
6	G3CMS0	1	R/W	Select output trigger of pulse output group 3. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
5	G2CMS1	1	R/W	Group 2 Compare Match Select 0 and 1
4	G2CMS0	1	R/W	Select output trigger of pulse output group 2. 00: Compare match in TPC channel 0 01: Compare match in TPC channel 1 10: Compare match in TPC channel 2 11: Compare match in TPC channel 3
3	G1CMS1	1	R/W	Reserved
2	G1CMS0	1	R/W	
1	G0CMS1	1	R/W	Reserved
0	G0CMS0	1	R/W	

11.3.5 PPG Output Mode Register (PMR)

The PMR is an 8-bit readable/writable register that selects the pulse output mode of the PPG for each group. If inverted output is selected, a low-level pulse is output when PODRH is 1 and a high-level pulse is output when PODRH is 0. If non-overlapping operation is selected, PPG updates its output values on compare match A or B of the TPU that becomes the output trigger. For details, see section 11.4.5, Non-Overlapping Pulse Output.

Bit	Bit Name	Initial Value	R/W	Description
7	G3INV	1	R/W	Group 3 Inversion Selects direct output or inverted output for pulse output group 3. 0: Inverted output 1: Direct output
6	G2INV	1	R/W	Group 2 Inversion Selects direct output or inverted output for pulse output group 2. 0: Inverted output 1: Direct output
5, 4	—	All 1	R/W	Reserved
3	G3NOV	0	R/W	Group 3 Non-Overlap Selects normal or non-overlapping operation for pulse output group 3. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values at compare match A or B in the selected TPU channel)
2	G2NOV	0	R/W	Group 2 Non-Overlap Selects normal or non-overlapping operation for pulse output group 2. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values at compare match A or B in the selected TPU channel)
1, 0	—	All 0	R/W	Reserved

11.4 Operation

11.4.1 Overview

Figure 11.2 shows a block diagram of the PPG. PPG pulse output is enabled when the corresponding bits in P1DDR and NDER are set to 1. An initial output value is determined by its corresponding PODR initial setting. When the compare match event specified by PCR occurs, the corresponding NDR bit contents are transferred to PODR to update the output values.

The sequential output of up to 8 bits of data is possible by writing new output data to NDR before the next compare match.

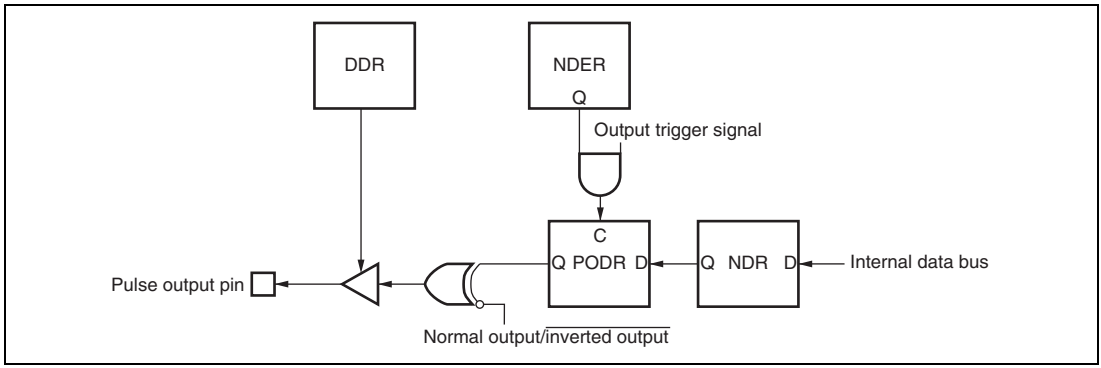


Figure 11.2 PPG Output Operation

11.4.2 Output Timing

If pulse output is enabled, the contents of NDR contents are transferred to PODR and output when the specified compare match event occurs. Figure 11.3 shows the timing of these operations for the case of normal output in groups 2 and 3, triggered by compare match A.

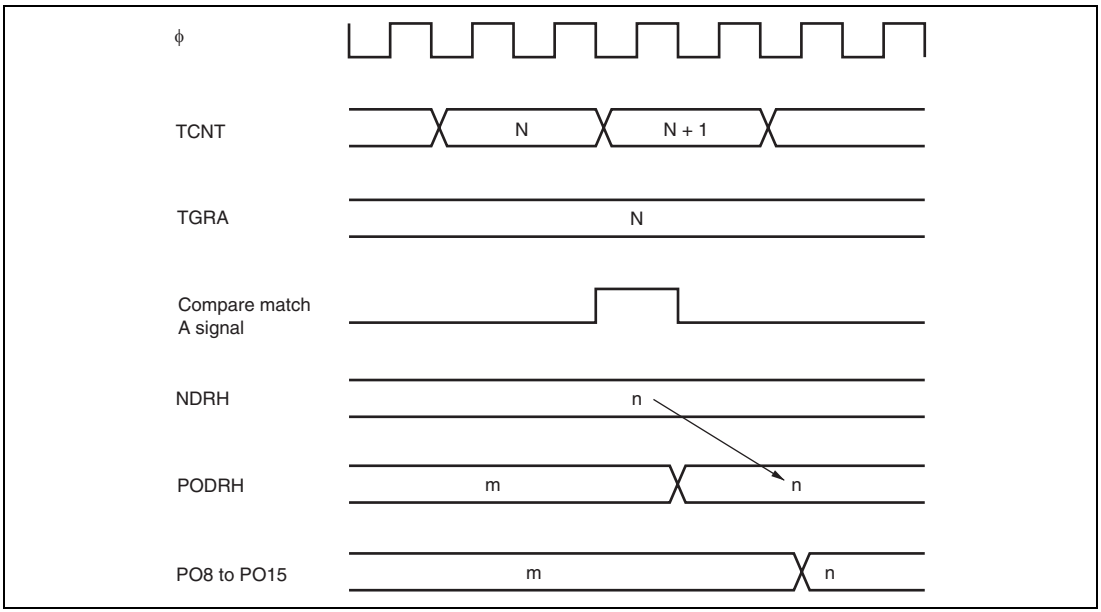


Figure 11.3 Timing of Transfer and Output of NDR Contents (Example)

11.4.3 Sample Setup Procedure for Normal Pulse Output

Figure 11.4 shows a sample procedure for setting up normal pulse output.

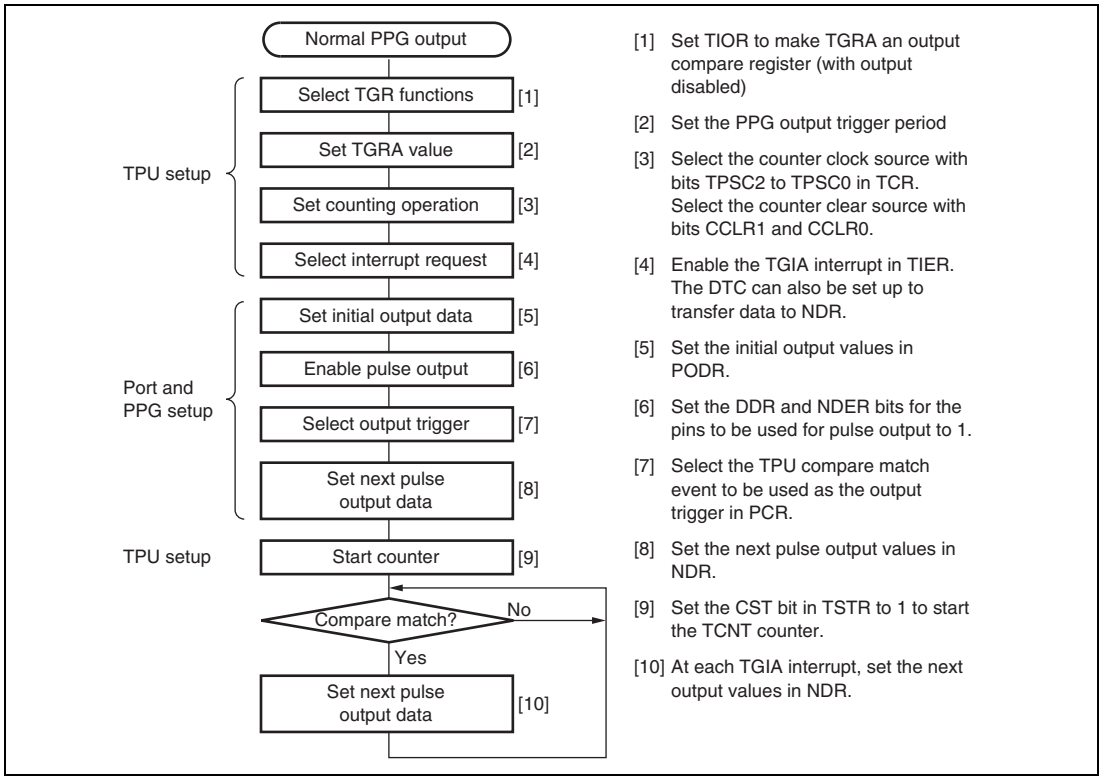


Figure 11.4 Setup Procedure for Normal Pulse Output (Example)

11.4.4 Example of Normal Pulse Output (Example of Five-Phase Pulse Output)

Figure 11.5 shows an example in which pulse output is used for cyclic five-phase pulse output.

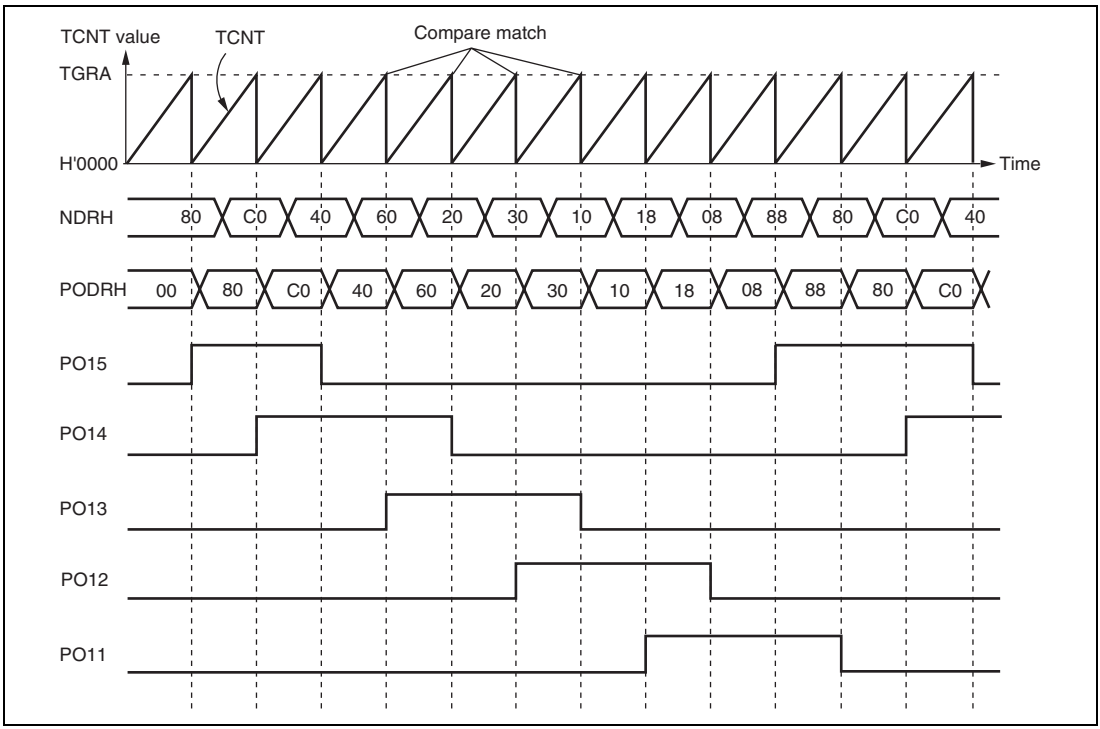


Figure 11.5 Normal Pulse Output Example (Five-Phase Pulse Output)

1. Set up TGRA of the TPU that is used as the output trigger to be an output compare register. Set a frequency in TGRA so the counter will be cleared on compare match A. Set the TGIEA bit of TIER to 1 to enable the compare match/input capture A (TGIA) interrupt.
2. Write H'F8 in P1DDR and NDERH, and set the G3CMS0, G3CMS1, G2CMS0, and G2CMS1 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Write output data H'80 in NDRH.
3. When compare match A occurs, the NDRH contents are transferred to PODRH and output. The TGIA interrupt handling routine writes the next output data (H'C0) in NDRH.
4. Five-phase overlapping pulse output (one or two phases active at a time) can be obtained subsequently by writing H'40, H'60, H'20, H'30. H'10, H'18, H'08, H'88... at successive TGIA interrupts. If the DTC is set for activation by this interrupt, pulse output can be obtained without imposing a load on the CPU.

11.4.5 Non-Overlapping Pulse Output

During non-overlapping operation, transfer from NDR to PODR is performed as follows:

- NDR bits are always transferred on PODR bits on compare match A.
- On compare match B, NDR bits are transferred only if their value is 0. Bits are not transferred if their value is 1.

Figure 11.6 illustrates the non-overlapping pulse output operation.

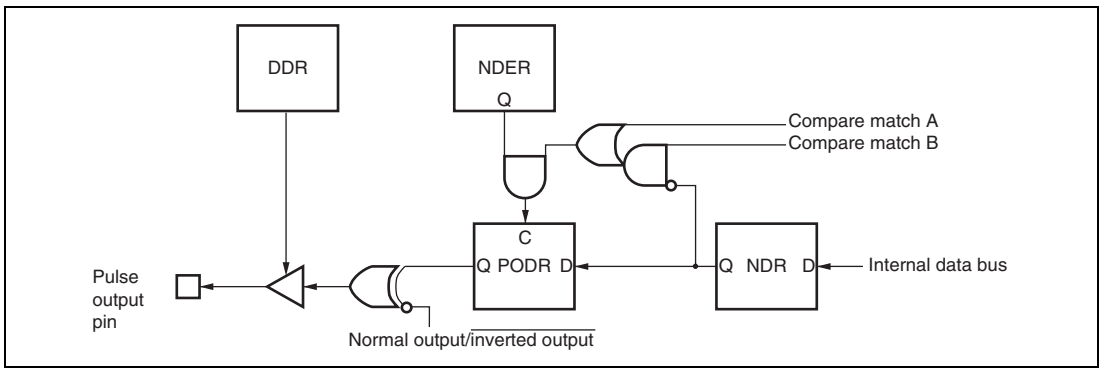


Figure 11.6 Non-Overlapping Pulse Output

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur before compare match A. The NDR contents should not be altered during the interval between compare match B and compare match A (the non-overlap margin).

This can be accomplished by having the TGIA interrupt handling routine write the next data in NDR, or by having the TGIA interrupt activate the DTC. Note, however, that the next data must be written before the next compare match B occurs.

Figure 11.7 shows the timing of this operation.

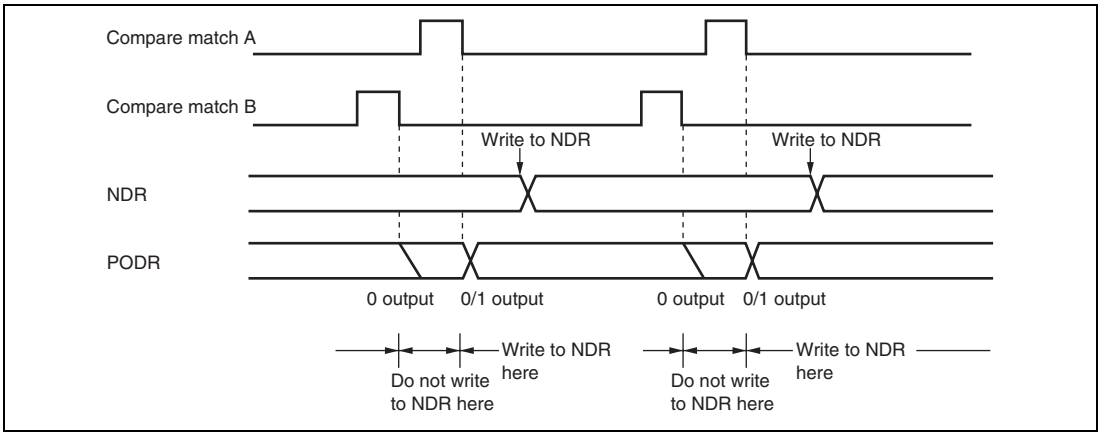


Figure 11.7 Non-Overlapping Operation and NDR Write Timing

11.4.6 Sample Setup Procedure for Non-Overlapping Pulse Output

Figure 11.8 shows a sample procedure for setting up non-overlapping pulse output.

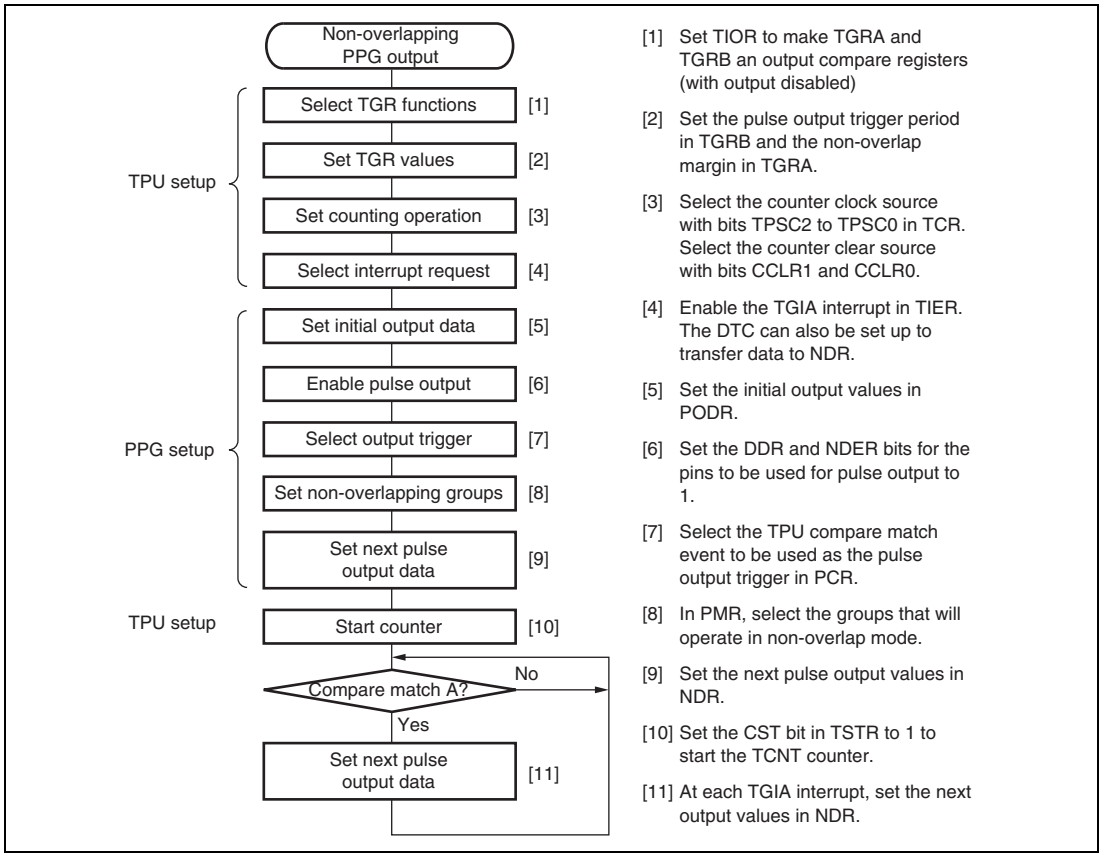


Figure 11.8 Setup Procedure for Non-Overlapping Pulse Output (Example)

11.4.7 Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)

Figure 11.9 shows an example in which pulse output is used for four-phase complementary non-overlapping pulse output.

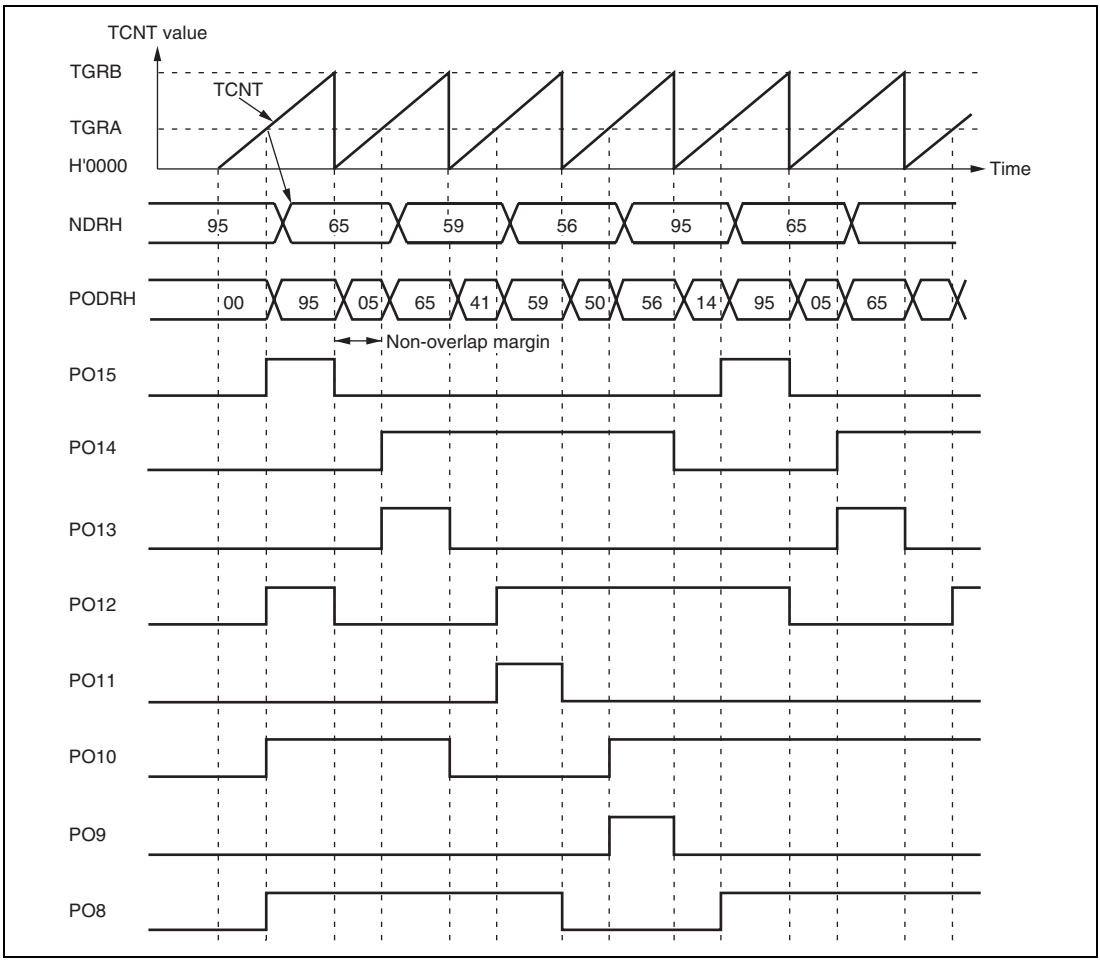


Figure 11.9 Non-Overlapping Pulse Output Example (Four-Phase Complementary)

1. Set up the TPU channel to be used as the output trigger channel such that TGRA and TGRB are output compare registers. Set the trigger period in TGRB and the non-overlap margin in TGRA, and set the counter to be cleared on compare match B. Set the TGIEA bit in TIER to 1 to enable the TGIA interrupt.
2. Write H'FF in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Set the G3NOV and G2NOV bits in PMR to 1 to select non-overlapping output. Write output data H'95 in NDRH.
3. The timer counter in the TPU channel starts. When a compare match with TGRB occurs, outputs change from 1 to 0. When a compare match with TGRA occurs, outputs change from 0 to 1 (the change from 0 to 1 is delayed by the value set in TGRA). The TGIA interrupt handling routine writes the next output data (H'65) in NDRH.
4. Four-phase complementary non-overlapping pulse output can be obtained subsequently by writing H'59, H'56, H'95... at successive TGIA interrupts. If the DTC is set for activation by this interrupt, pulse output can be obtained without imposing a load on the CPU.

11.4.8 Inverted Pulse Output

If the G3INV, G2INV, G1INV, and G0INV bits in PMR are cleared to 0, values that are the inverse of the PODR contents can be output.

Figure 11.10 shows the outputs when G3INV and G2INV are cleared to 0, in addition to the settings of figure 11.9.

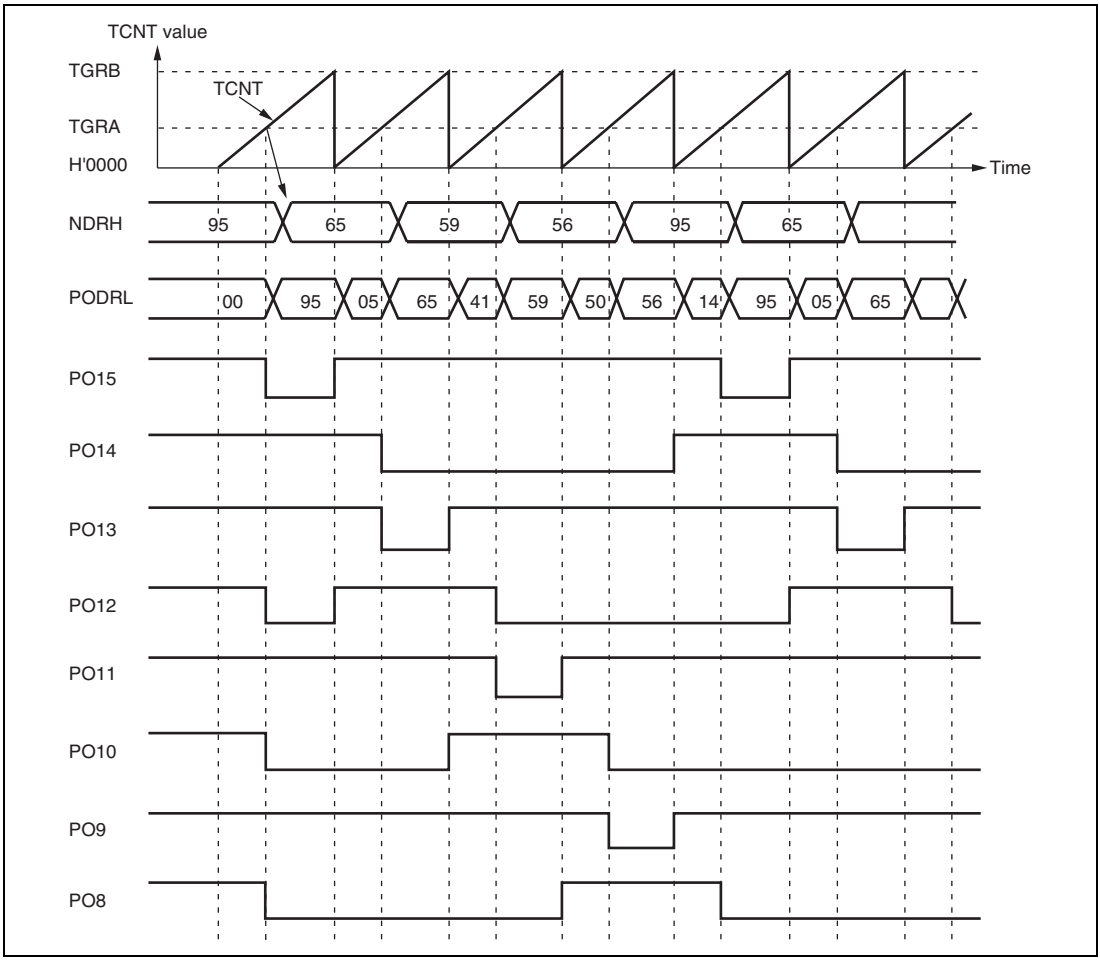


Figure 11.10 Inverted Pulse Output (Example)

11.4.9 Pulse Output Triggered by Input Capture

Pulse output can be triggered by TPU input capture as well as by compare match. If TGRA functions as an input capture register in the TPU channel selected by PCR, pulse output will be triggered by the input capture signal.

Figure 11.11 shows the timing of this output.

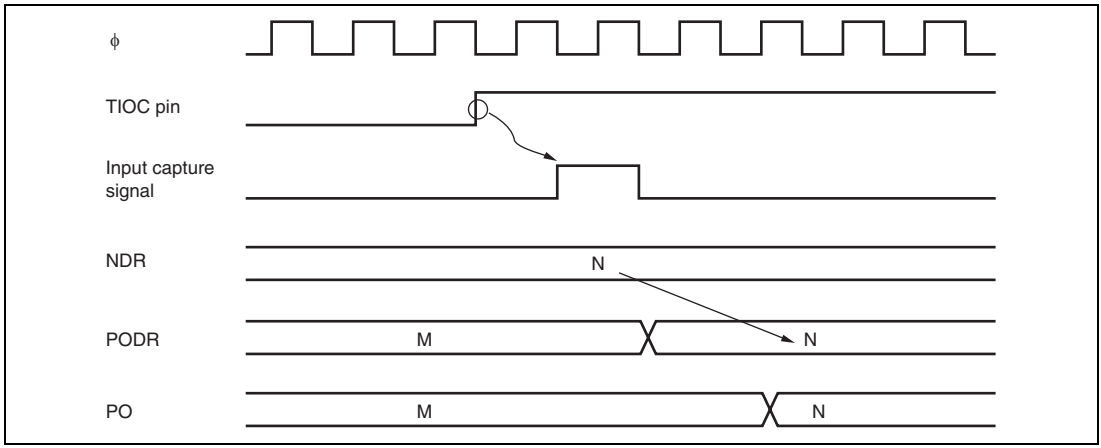


Figure 11.11 Pulse Output Triggered by Input Capture (Example)

11.5 Usage Notes

11.5.1 Module Stop Mode Setting

PPG operation can be disabled or enabled using the module stop control register. The initial setting is for PPG operation to be halted. Register access is enabled by clearing module stop mode. For details, see section 21, Power-Down Modes.

11.5.2 Operation of Pulse Output Pins

Pins PO8 to PO15 are also used for other peripheral functions such as the TPU. When output by another peripheral function is enabled, the corresponding pins cannot be used for pulse output. Note, however, that data transfer from NDR bits to PODR bits takes place, regardless of the usage of the pins.

Pin functions should be changed only under conditions in which the output trigger event will not occur.

Section 12 Watchdog Timer (WDT)

This LSI has a two-channel watchdog timer (WDT_0, WDT_1). WDT is an 8-bit timer that can generate an internal reset signal for this LSI if a system crash prevents the CPU from writing to the timer counter, thus allowing it to overflow.

When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is generated each time the counter overflows.

The block diagrams of the WDT_0 and WDT_1 are shown in figures 12.1 and 12.2, respectively.

12.1 Features

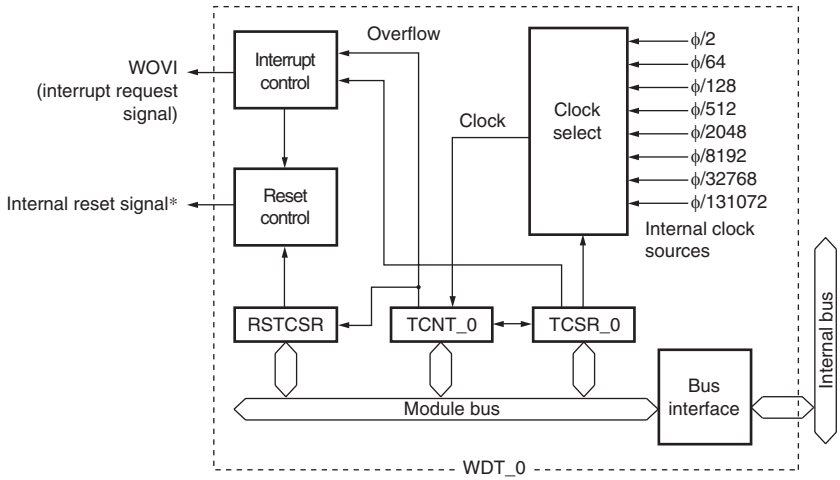
- Selectable from eight counter input clocks (WDT_0) or sixteen counter input clocks (WDT_1)
- Switchable between watchdog timer mode and interval timer mode

In watchdog timer mode:

- If the counter overflows, it is possible to select whether this LSI is internally reset or not or whether an internal NMI interrupt is generated or not.

In interval timer mode:

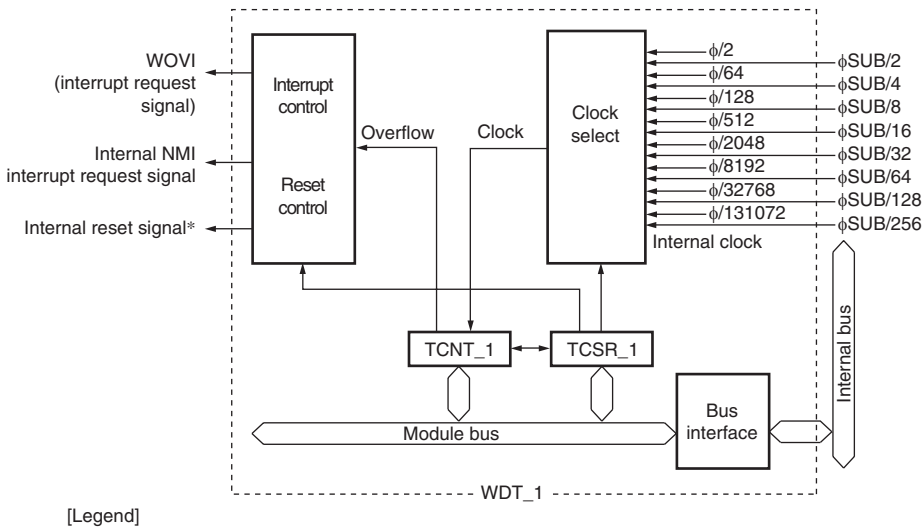
- If the counter overflows, the WDT generates an interval timer interrupt (WOVI).



[Legend]
TCSR_0: Timer control/status register_0
TCNT_0: Timer counter_0
RSTCSR: Reset control/status register

Note: * An internal reset signal can be generated by setting the register.

Figure 12.1 Block Diagram of WDT_0



[Legend]
TCSR_1: Timer control/status register_1
TCNT_1: Timer counter_1

Note: * An internal reset signal can be generated by setting the register.

Figure 12.2 Block Diagram of WDT_1

12.2 Register Descriptions

The WDT has the following registers. To prevent accidental overwriting, TCSR, TCNT, and RSTCSR have to be written to by a different method to normal registers. For details, see section 12.5.1, Notes on Register Access.

- Timer counter_0 (TCNT_0)
- Timer control/status register_0 (TCSR_0)
- Timer counter_1 (TCNT_1)
- Timer control/status register_1 (TCSR_1)
- Reset control/status register (RSTCSR)

12.2.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter. TCNT is initialized to H'00 by a reset, when the TME bit in TCSR is cleared to 0.

12.2.2 Timer Control/Status Register (TCSR)

TCSR selects the clock source to be input to TCNT and the timer mode.

- TCSR_0

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Overflow Flag Indicates that TCNT has overflowed. Only a write of 0 is permitted, to clear the flag. [Setting conditions] • When TCNT overflows (changes from H'FF to H'00) • When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset. [Clearing condition] • Cleared by reading TCSR when OVF = 1, then writing 0 to OVF

Bit	Bit Name	Initial Value	R/W	Description
6	WT/IT	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog timer or interval timer. 0: Interval timer mode 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting and is initialized to H'00.
4, 3	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
2	CKS2	0	R/W	Clock Select 2 to 0 These bits select the clock source to be input to TCNT. The overflow frequency for ϕ = 20 MHz is enclosed in parentheses. 000: Clock ϕ/2 (frequency: 25.6 μs) 001: Clock ϕ/64 (frequency: 819.2 μs) 010: Clock ϕ/128 (frequency: 1.6 ms) 011: Clock ϕ/512 (frequency: 6.6 ms) 100: Clock ϕ/2048 (frequency: 26.2 ms) 101: Clock ϕ/8192 (frequency: 104.9 ms) 110: Clock ϕ/32768 (frequency: 419.4 ms) 111: Clock ϕ/131072 (frequency: 1.68 s)
1	CKS1	0	R/W	
0	CKS0	0	R/W	

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

- TCSR_1

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Overflow Flag Indicates that TCNT has overflowed from H'FF to H'00. Only a write of 0 is permitted, to clear the flag. [Setting conditions] • When TCNT overflows (changes from H'FF to H'00) • When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset. [Clearing condition] • Cleared by reading TCSR when OVF = 1, then writing 0 to OVF
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog timer or interval timer. 0: Interval timer mode 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting and is initialized to H'00.
4	PSS	0	R/W	Prescaler Select Selects the clock source to be input to TCNT. 0: Counts the divided clock of ϕ–based prescaler (PSM) 1: Counts the divided clock of ϕSUB–based prescaler (PSS)
3	RST/$\overline{NMI}$	0	R/W	Reset or NMI Selects whether an internal reset request or an NMI interrupt request when the TCNT overflows during the watchdog timer mode. 0: NMI interrupt request 1: Internal reset request

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	Selects the clock source to be input to TCNT. The overflow cycle is the period from which TCNT starts incrementing at H'00 and until it overflows.
0	CKS0	0	R/W	
				When PSS = 0 (values in parentheses are for ϕ = 20 MHz):
				000: ϕ/2 (cycle: 25.6 μs)
				001: ϕ/64 (cycle: 819.2 ms)
				010: ϕ/128 (cycle: 1.6 ms)
				011: ϕ/512 (cycle: 6.6 ms)
				100: ϕ/2048 (cycle: 26.2 ms)
				101: ϕ/8192 (cycle: 104.9 ms)
				110: ϕ/32768 (cycle: 419.4 ms)
				111: ϕ/131072 (cycle: 1.68 s)
				When PSS = 1 (values in parentheses are for ϕSUB = 32.768 kHz):
				000: ϕSUB/2 (cycle: 15.6 ms)
				001: ϕSUB/4 (cycle: 31.3 ms)
				010: ϕSUB/8 (cycle: 62.5 ms)
				011: ϕSUB/16 (cycle: 125 ms)
				100: ϕSUB/32 (cycle: 250 ms)
				101: ϕSUB/64 (cycle: 500 ms)
				110: ϕSUB/128 (cycle: 1 s)
				111: ϕSUB/256 (cycle: 2 s)

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

12.2.3 Reset Control/Status Register (RSTCSR)

RSTCSR controls the generation of the internal reset signal when TCNT overflows, and selects the type of internal reset signal. RSTCSR is initialized to H'1F by a reset signal from the $\overline{\text{RES}}$ pin, and not by the WDT internal reset signal caused by overflows.

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)*	Watchdog Overflow Flag This bit is set when TCNT overflows in watchdog timer mode. This bit cannot be set in interval timer mode, and only 0 can be written. [Setting condition] • Set when TCNT overflows (changed from H'FF to H'00) in watchdog timer mode [Clearing condition] • Cleared by reading RSTCSR when WOVF = 1, and then writing 0 to WOVF
6	RSTE	0	R/W	Reset Enable Specifies whether or not a reset signal is generated in the chip if TCNT overflows during watchdog timer operation. 0: Reset signal is not generated even if TCNT overflows (Though this LSI is not reset, TCNT and TCSR in WDT are reset) 1: Reset signal is generated if TCNT overflows
5	RSTS	0	R/W	Reset Select Selects the type of internal reset generated if TCNT overflows during watchdog timer operation. 0: Power-on reset 1: Setting prohibited
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

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

12.3 Operation

12.3.1 Watchdog Timer Mode

To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ bit in TCSR and the TME bit to 1. TCNT does not overflow while the system is operating normally. Software must prevent TCNT overflows by rewriting the TCNT value (normally be writing H'00) before overflows occurs.

When the WDT is used as a watchdog timer, and if TCNT overflows without being rewritten because of a system malfunction or other error, an internal reset can be generated.

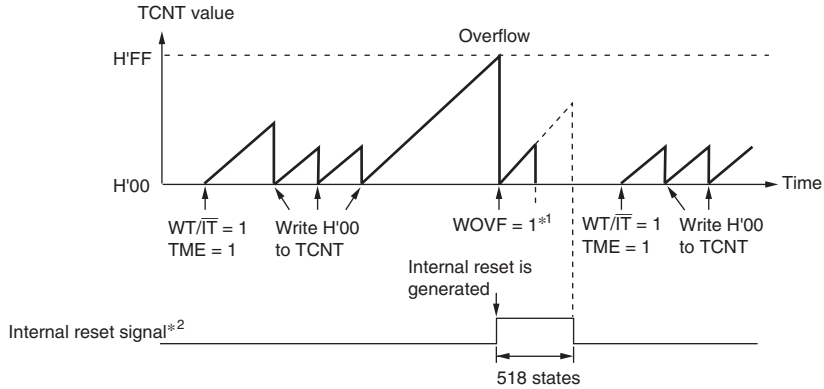
In the case of the WDT_0, when the TCNT overflows in watchdog timer mode, the WOVF bit in RSTCSR is set to 1. At this time, when the RSTE bit is set to 1, an internal reset signal for this LSI is output.

The internal reset signal is output for 518 states. This is illustrated in figure 12.3 (a).

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 WOVF bit in RSTCSR is cleared to 0.

In the case of the WDT_1, the LSI is reset, or an NMI interrupt request is generated, for 516 system clock periods (516 ϕ) (515 or 516 states when the clock source is ϕSUB (PSS = 1)). This is illustrated in figure 12.3 (b).

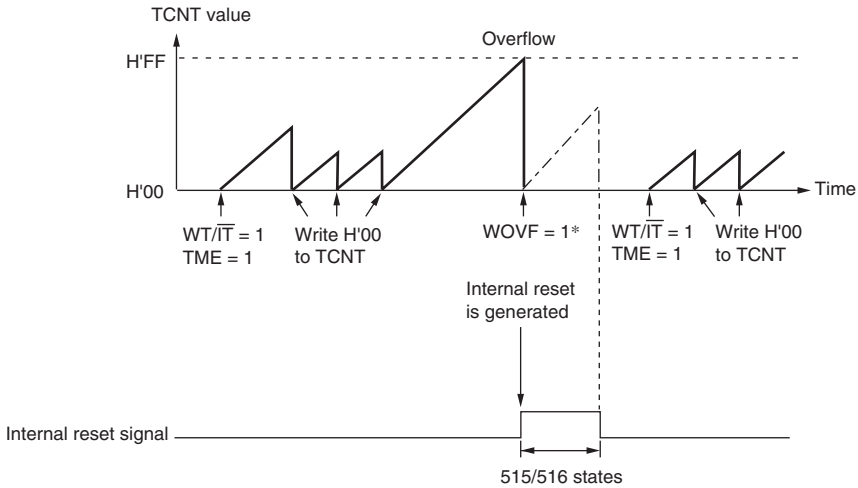
An NMI interrupt request from the watchdog timer and an interrupt request from the NMI pin are both treated as having the same vector. So, avoid handling an NMI interrupt request from the watchdog timer and an interrupt request from the NMI pin at the same time.



[Legend]
WT/$\overline{IT}$: Timer mode select bit
TME: Timer enable bit

Notes: 1. After the WOVF bit becomes 1, it is cleared to 0 by an internal reset.
2. The internal reset signal is generated only if the RSTE bit is set to 1.

Figure 12.3 (a) WDT_0 Operation in Watchdog Timer Mode



[Legend]
WT/$\overline{IT}$: Timer mode select bit
TME: Timer enable bit

Note: * After the WOVF bit becomes 1, it is cleared to 0 by an internal reset.

Figure 12.3 (b) WDT_1 Operation in Watchdog Timer Mode

12.3.2 Interval Timer Mode

When the WDT is used as an interval timer, an interval timer interrupt (WOVI) is generated each time the TCNT overflows. Therefore, an interrupt can be generated at intervals.

When the TCNT overflows in interval timer mode, an interval timer interrupt (WOVI) is requested at the time the OVF bit of the TCSR is set to 1.

12.4 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. OVF must be cleared to 0 in the interrupt handling routine.

If an NMI interrupt request has been selected in watchdog timer mode, an NMI interrupt request is generated when the TCNT overflows.

Table 12.1 WDT Interrupt Sources

Name	Interrupt Source	Interrupt Flag
WOVI	TCNT overflow (interval timer mode)	OVF
NMI	TCNT overflow (watchdog timer mode)	OVF

12.5 Usage Notes

12.5.1 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in being more difficult to write to. The procedures for writing to and reading these registers are given below.

Writing to TCNT, TCSR, and RSTCSR

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, the relative condition shown in figure 12.4 needs to be satisfied in order to write to TCNT or TCSR. The transfer instruction writes the lower byte data to TCNT or TCSR according to the satisfied condition.

To write to RSTCSR, execute a word transfer instruction for address H'FF76. A byte transfer instruction cannot write to RSTCSR.

The method of writing 0 to the WOVF bit differs from that of writing to the RSTE and RSTS bits. To write 0 to the WOVF bit, satisfy the condition shown in figure 12.4. If satisfied, the transfer instruction clears the WOVF bit to 0, but has no effect on the RSTE and RSTS bits. To write to the RSTE and RSTS bits, satisfy the condition shown in figure 12.4. If satisfied, the transfer instruction writes the values in bits 5 and 6 of the lower byte into the RSTE and RSTS bits, respectively, but has no effect on the WOVF bit.

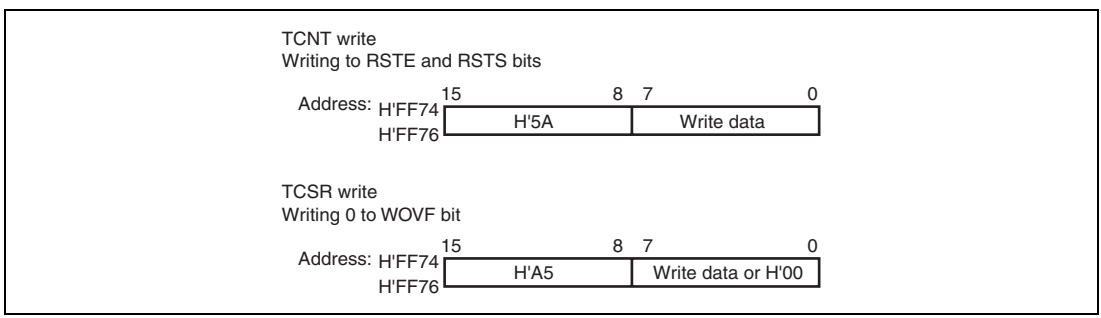


Figure 12.4 Writing to TCNT, TCSR, and RSTCSR (example for WDT0)

Reading TCNT, TCSR, and RSTCSR (WDT0)

These registers are read in the same way as other registers. The read addresses are H'FF74 for TCSR, H'FF75 for TCNT, and H'FF77 for RSTCSR.

12.5.2 Contention between Timer Counter (TCNT) Write and Increment

If a timer counter clock pulse is generated during the T2 state of a TCNT write cycle, the write takes priority and the timer counter is not incremented. Figure 12.5 shows this operation.

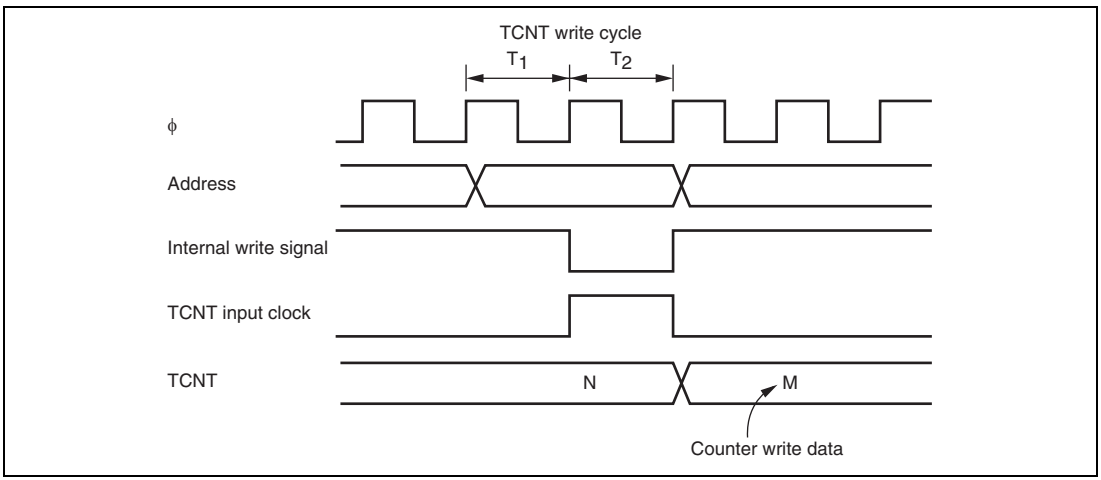


Figure 12.5 Contention between TCNT Write and Increment

12.5.3 Changing Value of CKS2 to CKS0

If bits CKS0 to CKS2 in TCSR are written to while the WDT is operating, errors could occur in the incrementation. Software must be used to stop the watchdog timer (by clearing the TME bit to 0) before changing the value of bits CKS0 to CKS2.

12.5.4 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from watchdog timer to interval timer while the WDT is operating, errors could occur in the incrementation. Software must be used to stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

12.5.5 Internal Reset in Watchdog Timer Mode

This LSI is not reset internally if TCNT overflows while the RSTE bit is cleared to 0 during watchdog timer operation, however TCNT and TCSR of the WDT are reset.

TCNT, TCSR, or RSTCR cannot be written to for 132 states following an overflow. During this period, any attempt to read the WOVF flag is not acknowledged. Accordingly, wait 132 states after overflow to write 0 to the WOVF flag for clearing.

12.5.6 OVF Flag Clearing in Interval Timer Mode

When the OVF flag setting conflicts with the OVF flag reading in interval timer mode, writing 0 to the OVF bit may not clear the flag even though the OVF bit has been read while it is 1. If there is a possibility that the OVF flag setting and reading will conflict, such as when the OVF flag is polled with the interval timer interrupt disabled, read the OVF bit while it is 1 at least twice before writing 0 to the OVF bit to clear the flag.

Section 13 Serial Communication Interface (SCI)

This LSI has four independent serial communication interface (SCI) channels. The SCI can handle both asynchronous and clocked synchronous serial communication. Serial data communication can be carried out using standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or an Asynchronous Communication Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function). The SCI also supports an IC card (smart card) interface conforming to ISO/IEC 7816-3 (Identification Card) as a serial communication interface extension function.

Figure 13.1 shows a block diagram of the SCI.

13.1 Features

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

  The transmitter and receiver are mutually independent, enabling transmission and 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

  External clock can be selected as a transfer clock source (except for in Smart card interface mode).
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode 7-bit data)
- Four interrupt sources

  Transmit-end, transmit-data-empty, receive-data-full, and receive error — that can issue requests.

  The transmit-data-empty interrupt and receive data full interrupts can be used to activate the data transfer controller (DTC).
- Module stop mode can be set

Asynchronous mode

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

- Break detection: Break can be detected by reading the RxD pin level directly in the case of a framing error

Clocked Synchronous mode

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

Smart Card Interface

- Automatic transmission of error signal (parity error) in receive mode
- Error signal detection and automatic data retransmission in transmit mode
- Direct convention and inverse convention both supported

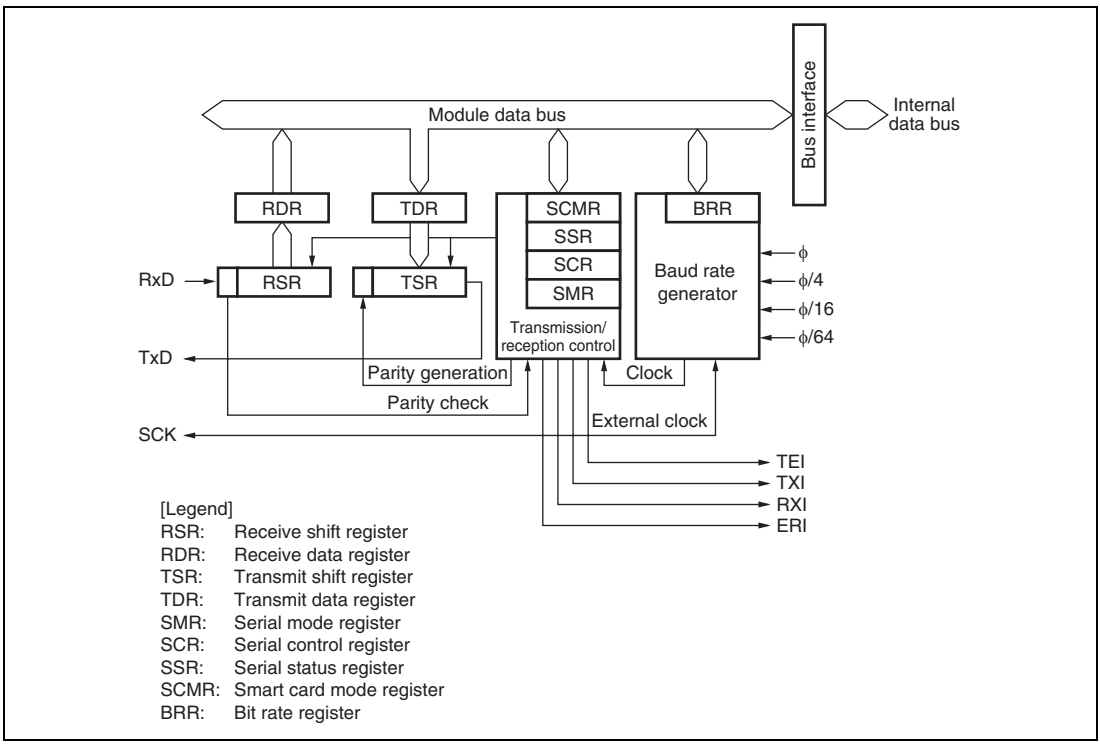


Figure 13.1 Block Diagram of SCI

13.2 Input/Output Pins

Table 13.1 shows the serial pins for each SCI channel.

Table 13.1 Pin Configuration

Channel	Pin Name*	I/O	Function
0	SCK0	I/O	SCI0 clock input/output
	RxD0	Input	SCI0 receive data input
	TxD0	Output	SCI0 transmit data output
1	SCK1	I/O	SCI1 clock input/output
	RxD1	Input	SCI1 receive data input
	TxD1	Output	SCI1 transmit data output
2	SCK2	I/O	SCI2 clock input/output
	RxD2	Input	SCI2 receive data input
	TxD2	Output	SCI2 transmit data output

Note: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting the channel designation.

13.3 Register Descriptions

The SCI has the following registers for each channel. The serial mode register (SMR), serial status register (SSR), and serial control register (SCR) are described separately for normal serial communication interface mode and smart card interface mode because their bit functions differ in part.

- 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)
- Smart card mode register (SCMR)
- Bit rate register (BRR)

13.3.1 Receive Shift Register (RSR)

RSR is a shift register that is used to receive serial data input to the RxD pin and convert it into parallel data. When one byte of data has been received, it is transferred to RDR automatically. RSR cannot be directly accessed by the CPU.

13.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores received data. When the SCI has received one byte of serial data, it transfers the received serial data from RSR to RDR, where it is stored. After this, RSR is receive-enabled. As RSR and RDR function as a double buffer in this way, continuous receive operations are possible. After confirming that the RDRF bit in SSR is set to 1, read RDR only once. RDR cannot be written to by the CPU.

13.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for transmission. 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 during serial transmission, the SCI transfers the written data to TSR to continue transmission. Although TDR can be read or written to by the CPU at all times, to achieve reliable serial transmission, write transmit data to TDR only once after confirming that the TDRE bit in SSR is set to 1.

13.3.4 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, and then sends the data to the TxD pin. TSR cannot be directly accessed by the CPU.

13.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI's serial transfer format and select the baud rate generator clock source.

Some bit functions of SMR differ between normal serial communication interface mode and smart card interface mode.

- Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

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.

Bit	Bit Name	Initial Value	R/W	Description
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. For a multiprocessor format, parity bit addition and checking are not performed regardless of the PE bit setting.
4	O/$\overline{E}$	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous mode) Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits In reception, only the first stop bit is checked. If the second stop bit is 0, it is treated as the start bit of the next transmit character.
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 0 and 1:
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 relationship between the bit rate register setting and the baud rate, see section 13.3.9, Bit Rate Register (BRR). n is the decimal representation of the value of n in BRR (see section 13.3.9, Bit Rate Register (BRR)).

- Smart Card Interface Mode (When SMIF in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	GM	0	R/W	GSM Mode When this bit is set to 1, the SCI operates in GSM mode. In GSM mode, the timing of the TEND setting is advanced by 11.0 etu (Elementary Time Unit: the time for transfer of one bit), and clock output control mode addition is performed. For details, see section 13.7.8, Clock Output Control.
6	BLK	0	R/W	When this bit is set to 1, the SCI operates in block transfer mode. For details on block transfer mode, see section 13.7.3, Block Transfer Mode.
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 in transmission, and the parity bit is checked in reception. In smart card interface mode, this bit must be set to 1.
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. For details on setting this bit in smart card interface mode, see section 13.7.2, Data Format (Except for Block Transfer Mode).
3	BCP1	0	R/W	Basic Clock Pulse 1 and 2
2	BCP0	0	R/W	These bits specify the number of basic clock periods in a 1-bit transfer interval on the smart card interface. 00: 32 clock (S = 32) 01: 64 clock (S = 64) 10: 372 clock (S = 372) 11: 256 clock (S = 256) For details, see section 13.7.4, Receive Data Sampling Timing and Reception Margin in smart card Interface Mode. S stands for the value of S in BRR (see section 13.3.9, Bit Rate Register (BRR)).

Bit	Bit Name	Initial Value	R/W	Description
1	CKS1	0	R/W	Clock Select 0 and 1
0	CKS0	0	R/W	These bits select the clock source for the 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 relationship between the bit rate register setting and the baud rate, see section 13.3.9, Bit Rate Register (BRR). n is the decimal representation of the value of n in BRR (see section 13.3.9, Bit Rate Register (BRR)).

13.3.6 Serial Control Register (SCR)

SCR is a register that enables or disables SCI transfer operations and interrupt requests, and is also used to selection of the transfer clock source. For details on interrupt requests, see section 13.8, Interrupt Sources. Some bit functions of SCR differ between normal serial communication interface mode and smart card interface mode.

- Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, the TXI interrupt 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 s set to 1, transmission is enabled.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.

Bit	Bit Name	Initial Value	R/W	Description
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode) When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped, and setting of the RDRF, FER, and ORER status flags in SSR is prohibited. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared and normal reception is resumed. For details, see section 13.5, Multiprocessor Communication Function.
2	TEIE	0	R/W	Transmit End Interrupt Enable This bit is set to 1, TEI interrupt request is enabled.
1	CKE1	0	R/W	Clock Enable 0 and 1
0	CKE0	0	R/W	Selects the clock source and SCK pin function. Asynchronous mode 00: Internal baud rate generator SCK pin functions as I/O port 01: Internal baud rate generator Outputs a clock of the same frequency as the bit rate from the SCK pin. 1X: External clock Inputs a clock with a frequency 16 times the bit rate from the SCK pin. Clocked synchronous mode 0X: Internal clock (SCK pin functions as clock output) 1X: External clock (SCK pin functions as clock input)

[Legend]

X: Don't care

- Smart Card Interface Mode (When SMIF in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, 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) Write 0 to this bit in smart card interface mode.
2	TEIE	0	R/W	Transmit End Interrupt Enable Write 0 to this bit in smart card interface mode.
1	CKE1	0	R/W	Clock Enable 0 and 1
0	CKE0	0	R/W	Enables or disables clock output from the SCK pin. The clock output can be dynamically switched in GSM mode. For details, see section 13.7.8, Clock Output Control. When the GM bit in SMR is 0: 00: Output disabled (SCK pin can be used as an I/O port pin) 01: Clock output 1X: Reserved When the GM bit in SMR is 1: 00: Output fixed low 01: Clock output 10: Output fixed high 11: Clock output

[Legend]

X: Don't care

13.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI and multiprocessor bits for transfer. 1 cannot be written to flags TDRE, RDRF, ORER, PER, and FER; they can only be cleared. Some bit functions of SSR differ between normal serial communication interface mode and smart card interface mode.

- Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/W	Transmit Data Register Empty Displays whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt request and writes data to TDR
6	RDRF	0	R/W	Receive Data Register Full Indicates that the received 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 the DTC is activated by an RXI interrupt and transferred data from RDR The RDRF flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
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
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 the DTC is activated by a TXI interrupt and writes data to TDR
1	MPB	0	R	Multiprocessor Bit MPB stores the multiprocessor bit in the receive data. When the RE bit in SCR is cleared to 0 its previous state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be added to the transmit data.

- Smart Card Interface Mode (When SMIF in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/W	Transmit Data Register Empty Displays whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt request and writes data to TDR
6	RDRF	0	R/W	Receive Data Register Full Indicates that the received 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 the DTC is activated by an RXI interrupt and transferred data from RDR The RDRF flag is not affected and retains their previous values when 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
4	ERS	0	R/W	Error Signal Status [Setting condition] • When the low level of the error signal is sampled [Clearing conditions] • When 0 is written to ERS after reading ERS = 1

Bit	Bit Name	Initial Value	R/W	Description
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 This bit is set to 1 when no error signal has been sent back from the receiving end and the next transmit data is ready to be transferred to TDR. [Setting conditions] • When the TE bit in SCR is 0 and the ERS bit is also 0 • When the ERS bit is 0 and the TDRE bit is 1 after the specified interval following transmission of 1-byte data. The timing of bit setting differs according to the register setting as follows: When GM = 0 and BLK = 0, 2.5 etu after transmission starts When GM = 0 and BLK = 1, 1.5 etu after transmission starts When GM = 1 and BLK = 0, 1.0 etu after transmission starts When GM = 1 and BLK = 1, 1.0 etu after transmission starts [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writes data to TDR
1	MPB	0	R	Multiprocessor Bit This bit is not used in smart card interface mode.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Write 0 to this bit in smart card interface mode.

13.3.8 Smart Card Mode Register (SCMR)

SCMR is a register that selects smart card interface mode and its format.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	SDIR	0	R/W	Smart Card Data Transfer Direction Selects the serial/parallel conversion format. 0: LSB-first in transfer 1: MSB-first in transfer The bit setting is valid only when the transfer data format is 8 bits. For 7-bit data, LSB-first is fixed.
2	SINV	0	R/W	Smart Card Data Invert Specifies inversion of the data logic level. The SINV bit does not affect the logic level of the parity bit. To invert the parity bit, 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	—	Reserved This bit is always read as 1.
0	SMIF	0	R/W	Smart Card Interface Mode Select This bit is set to 1 to make the SCI operate in smart card interface mode. 0: Normal asynchronous mode or clocked synchronous mode 1: Smart card interface mode

13.3.9 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 13.2 shows the relationships between the N setting in BRR and bit rate B for normal asynchronous mode, clocked synchronous mode, and smart card interface mode. The initial value of BRR is H'FF, and it can be read or written to by the CPU at all times.

Table 13.2 The Relationships between The N Setting in BRR and Bit Rate B

Mode	BRR Setting N	Error
Asynchronous Mode	$N = \frac{\phi \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \{ \frac{\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \} \times 100$
Clocked Synchronous Mode	$N = \frac{\phi \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart Card Interface Mode	$N = \frac{\phi \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$\text{Error (\%)} = \{ \frac{\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \} \times 100$

[Legend]

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 tables.

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 13.3 shows sample N settings in BRR in normal asynchronous mode. Table 13.4 shows the maximum bit rate for each frequency in normal asynchronous mode. Table 13.6 shows sample N settings in BRR in clocked synchronous mode. Table 13.8 shows sample N settings in BRR in smart card interface mode. In smart card interface mode, S (the number of basic clock periods in a 1-bit transfer interval) can be selected. For details, see section 13.7.4, Receive Data Sampling Timing and Reception Margin in smart card Interface Mode. Tables 13.5 and 13.7 show the maximum bit rates with external clock input.

Table 13.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (1)

	Operating Frequency ϕ (MHz)								
	4			4.9152			5		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	70	0.03	2	86	0.31	2	88	–0.25
150	1	207	0.16	1	255	0.00	2	64	0.16
300	1	103	0.16	1	127	0.00	1	129	0.16
600	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	25	0.16	0	31	0.00	0	32	–1.36
9600	0	12	0.16	0	15	0.00	0	15	1.73
19200	—	—	—	0	7	0.00	0	7	1.73
31250	0	3	0.00	0	4	–1.70	0	4	0.00
38400	—	—	—	0	3	0.00	0	3	1.73

	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	—	—	—

Table 13.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (2)

	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

	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	0.13	2	191	0.00	2	207	0.16	2	223	0.00
300	2	90	0.13	2	95	0.00	2	103	0.16	2	111	0.00
600	1	181	0.13	1	191	0.00	1	207	0.16	1	223	0.00
1200	1	90	0.13	1	95	0.00	1	103	0.16	1	111	0.00
2400	0	181	0.13	0	191	0.00	0	207	0.16	0	223	0.00
4800	0	90	0.13	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	13	1.20
38400	—	—	—	0	11	0.00	0	12	0.16	0	13	0.00

Table 13.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (3)

	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

Table 13.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N	ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
4	125000	0	0	12	375000	0	0
4.9152	153600	0	0	12.288	384000	0	0
5	156250	0	0	14	437500	0	0
6	187500	0	0	14.7456	460800	0	0
6.144	192000	0	0	16	500000	0	0
7.3728	230400	0	0	17.2032	537600	0	0
8	250000	0	0	18	562500	0	0
9.8304	307200	0	0	19.6608	614400	0	0
10	312500	0	0	20	625000	0	0

Table 13.5 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)
4	1.0000	62500	12	3.0000	187500
4.9152	1.2288	76800	12.288	3.0720	192000
5	1.2500	78125	14	3.5000	218750
6	1.5000	93750	14.7456	3.6864	230400
6.144	1.5360	96000	16	4.0000	250000
7.3728	1.8432	115200	17.2032	4.3008	268800
8	2.0000	125000	18	4.5000	281250
9.8304	2.4576	153600	19.6608	4.9152	307200
10	2.5000	156250	20	5.0000	312500

Table 13.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

Bit Rate (bit/s)	Operating Frequency ϕ (MHz)									
	4		8		10		16		20	
	n	N	n	N	n	N	n	N	n	N
110	—	—								
250	2	249	3	124	—	—	3	249		
500	2	124	2	249	—	—	3	124	—	—
1k	1	249	2	124	—	—	2	249	—	—
2.5k	1	99	1	199	1	249	2	99	2	124
5k	0	199	1	99	1	124	1	199	1	249
10k	0	99	0	199	0	249	1	99	1	124
25k	0	39	0	79	0	99	0	159	0	199
50k	0	19	0	39	0	49	0	79	0	99
100k	0	9	0	19	0	24	0	39	0	49
250k	0	3	0	7	0	9	0	15	0	19
500k	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 transfer is not possible.

Table 13.7 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)
4	0.6667	666666.7	14	2.3333	2333333.3
6	1.0000	1.000000.0	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
12	2.0000	2000000.0			

Table 13.8 Examples of Bit Rate for Various BRR Settings (Smart Card Interface Mode) (When n = 0 and S = 372)

	Operating Frequency ϕ (MHz)											
	7.1424			10.00			10.7136			13.00		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	0	0.00	0	1	30	0	1	25	0	1	8.99

	Operating Frequency ϕ (MHz)											
	14.2848			16.00			18.00			20.00		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	0.00	0	1	12.01	0	2	15.99	0	2	6.60

Table 13.9 Maximum Bit Rate at Various Frequencies (Smart Card Interface Mode) (when S = 372)

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N	ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
7.1424	9600	0	0	14.2848	19200	0	0
10.00	13441	0	0	16.00	21505	0	0
10.7136	14400	0	0	18.00	24194	0	0
13.00	17473	0	0	20.00	26882	0	0

13.4 Operation in Asynchronous Mode

Figure 13.2 shows the general format for asynchronous serial communication. One frame consists of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low level), and finally stop bits (high level). In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the transmission line. When the transmission line goes to the space state (low level), the SCI recognizes a start bit and starts serial communication. In asynchronous serial communication, the communication line is usually held in the mark state (high level). The SCI monitors the communication line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer.

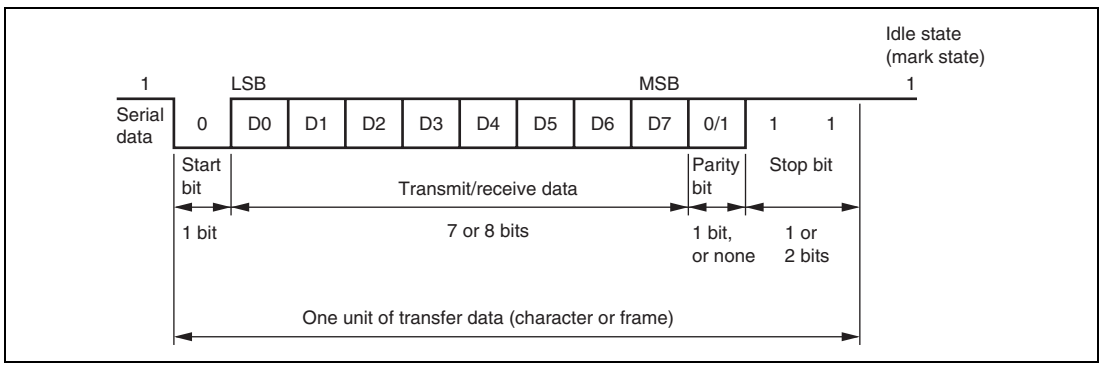


Figure 13.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

13.4.1 Data Transfer Format

Table 13.10 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 13.5, Multiprocessor Communication Function.

Table 13.10 Serial Transfer Formats (Asynchronous Mode)

SMR Settings				Serial Transfer Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	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

13.4.2 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 transfer rate. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 8th pulse of the basic clock as shown in figure 13.3. Thus, the reception margin in asynchronous mode is given by formula (1) below.

$$M = \{ (0.5 - \frac{1}{2N}) - \frac{D - 0.5}{N} - (L - 0.5) F\} \times 100 \ [\%]$$

... Formula (1)

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

Assuming values of F (absolute value of clock rate deviation) = 0 and D (clock duty cycle) = 0.5 in formula (1), the reception margin can be given by the formula.

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

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

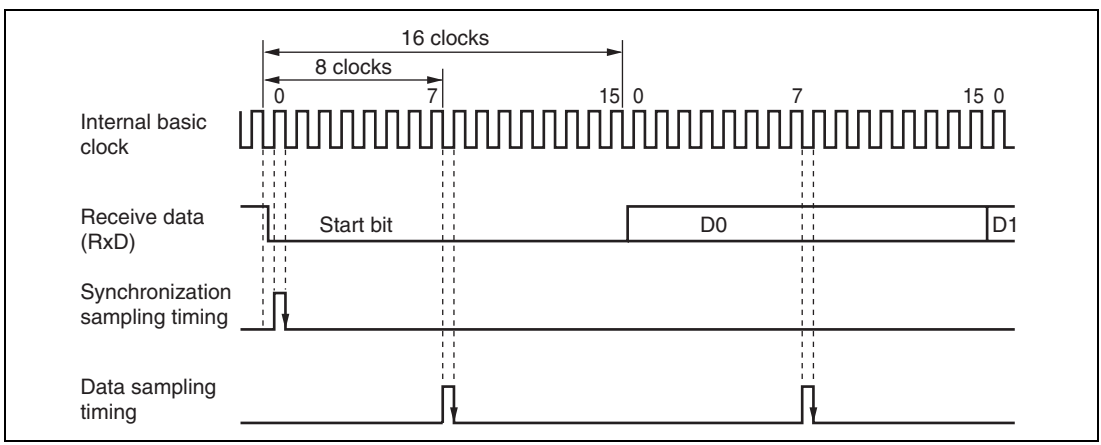


Figure 13.3 Receive Data Sampling Timing in Asynchronous Mode

13.4.3 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 serial clock, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE0 and CKE1 bits in SCR. When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 13.4.

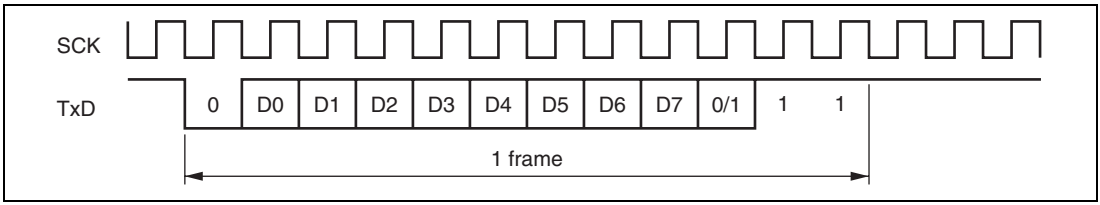


Figure 13.4 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode)

13.4.4 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 described below. When the operating mode, or transfer format, is changed for example, 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 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, or the contents of RDR. When the external clock is used in asynchronous mode, the clock must be supplied even during initialization.

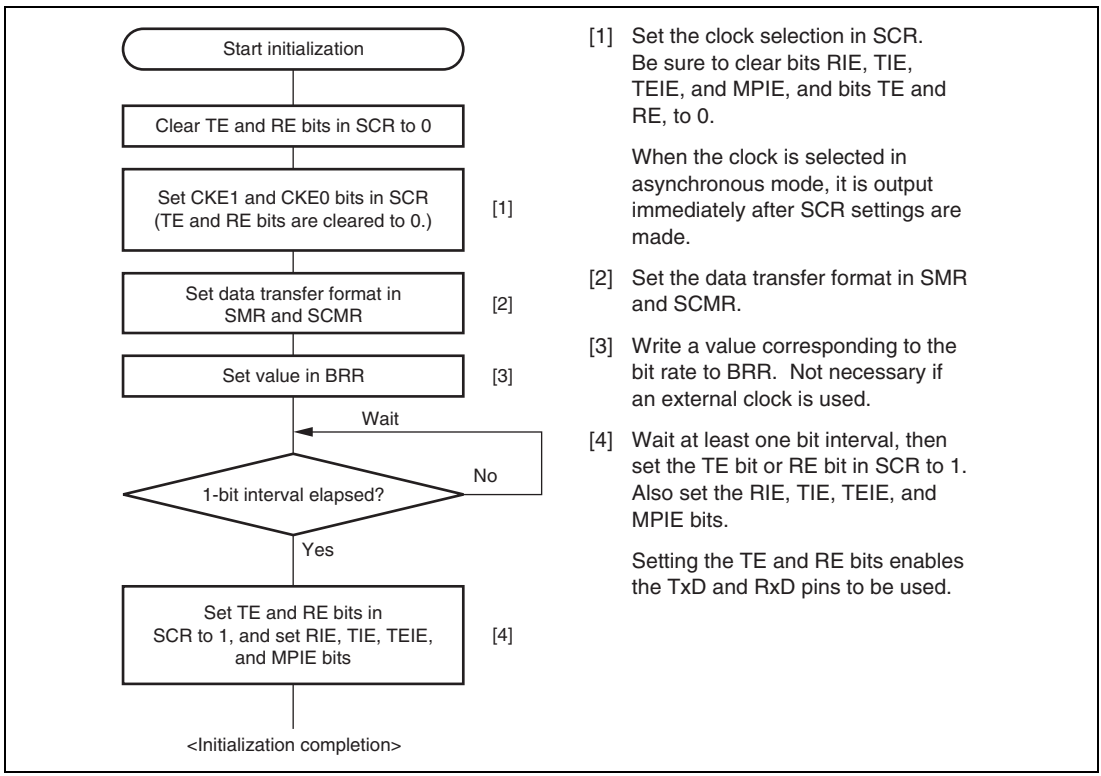


Figure 13.5 Sample SCI Initialization Flowchart

13.4.5 Data Transmission (Asynchronous Mode)

Figure 13.6 shows an example of operation for transmission in asynchronous mode. In transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR. If the flag is cleared to 0, the SCI 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 is set to 1 at this time, a transmit data empty interrupt request (TXI) is generated. Continuous transmission is possible because the TXI interrupt routine writes next transmit data to TDR before transmission of the current transmit data has been completed.
3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit or multiprocessor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks the TDRE flag at the timing for sending the stop bit.
5. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.
6. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the "mark state" is entered, in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 13.7 shows a sample flowchart for transmission in asynchronous mode.

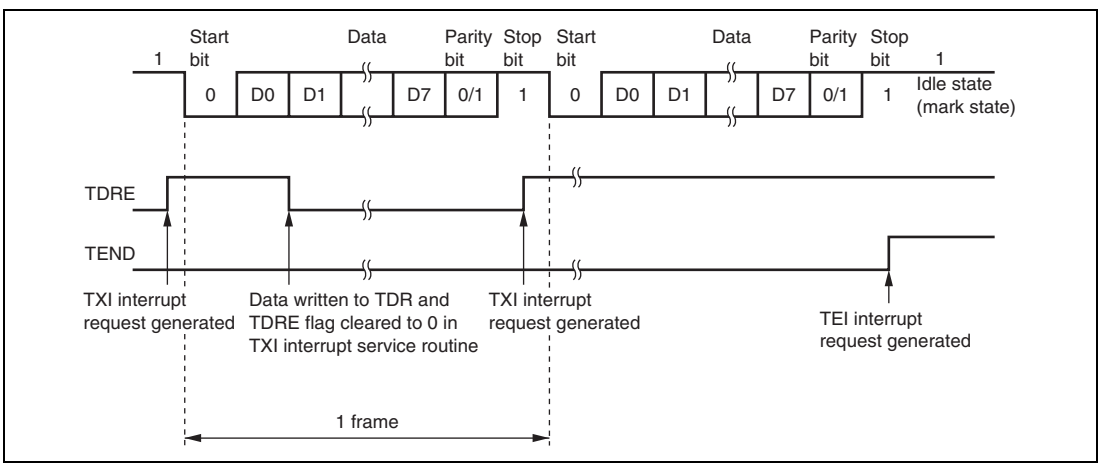
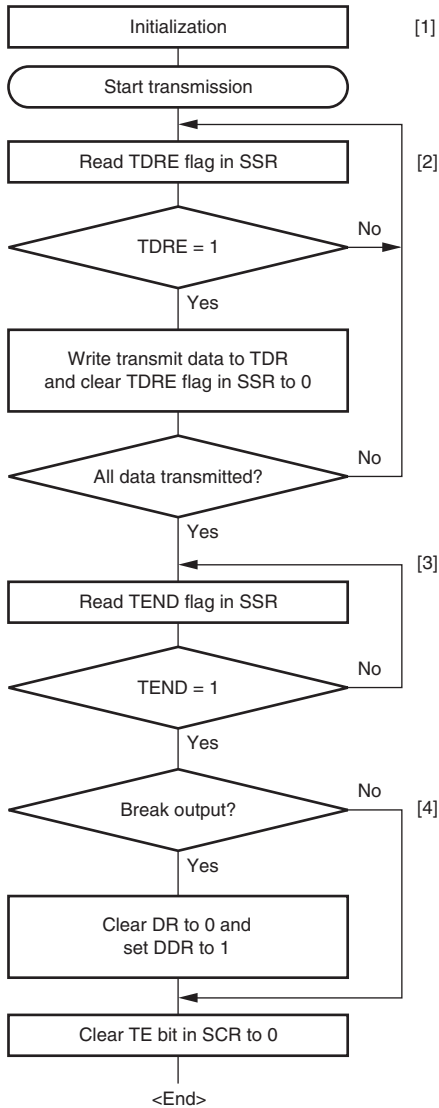


Figure 13.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)



[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 then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request, and data is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 13.7 Sample Serial Transmission Flowchart

13.4.6 Serial Data Reception (Asynchronous Mode)

Figure 13.8 shows an example of operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

1. The SCI monitors the communication line. If a start bit is detected, the SCI performs internal synchronization, receives receive data in RSR, and checks the parity bit and stop bit.
2. If an overrun error occurs (when reception of the next data is completed while the RDRF flag is still set to 1), 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 is detected (when the stop bit is 0), the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
5. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Continuous reception is possible because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has been completed.

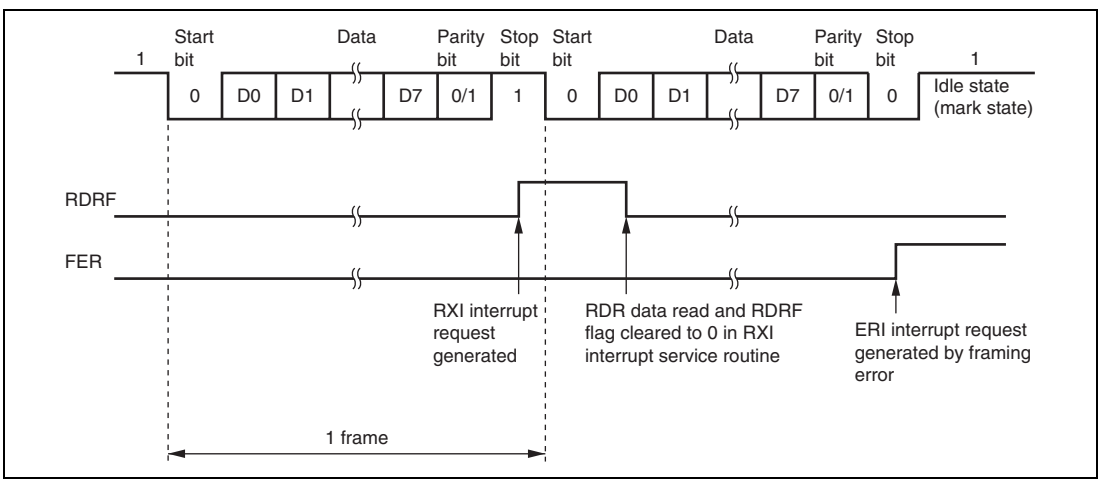


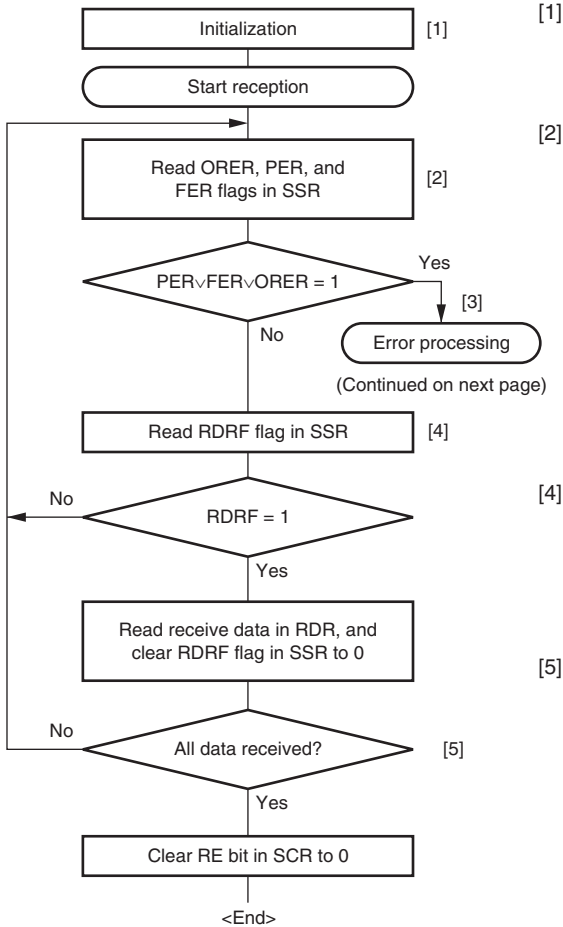
Figure 13.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

Table 13.11 shows the states of the SSR status flags and receive data handling when a receive error is detected. If a receive error is detected, the RDRF flag retains its state before receiving data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 13.9 shows a sample flow chart for serial data reception.

Table 13.11 SSR Status Flags and Receive Data Handling

SSR Status Flag					
RDRF*	**ORER**	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	1	0	0	Lost	Overrun error
0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + framing error
1	1	0	1	Lost	Overrun error + parity error
0	0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	1	Lost	Overrun error + framing error + parity error

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



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error processing and break detection:
If a receive error occurs, read the ORER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxD pin.

[4] SCI status check and receive data read:
Read SSR and check that RDRF = 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0. The RDRF flag is cleared automatically when DTC is activated by an RXI interrupt and the RDR value is read.

Figure 13.9 Sample Serial Reception Data Flowchart (1)

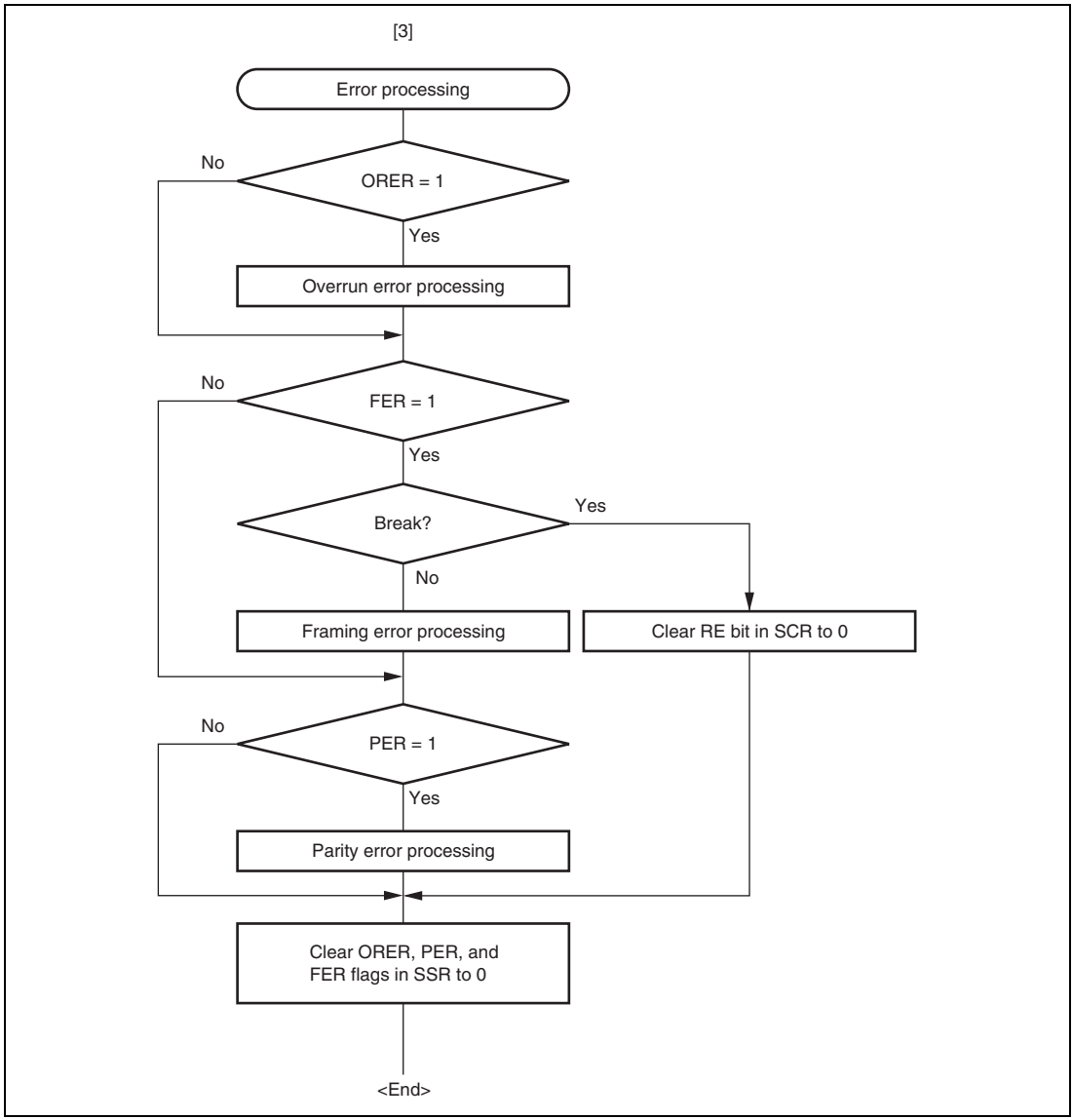


Figure 13.9 Sample Serial Reception Data Flowchart (2)

13.5 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer between a number of processors sharing communication lines by asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is performed, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles; an ID transmission cycle that specifies the receiving station, and a data transmission cycle. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle; if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 13.10 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends the ID code of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose IDs do not match continue to skip data until data with a 1 multiprocessor bit is again received.

The 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 to 1, are inhibited until data with a 1 multiprocessor bit is received. On reception of a receive character with a 1 multiprocessor bit, the 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 rendered invalid. All other bit settings are the same as those in normal asynchronous mode. The clock used for multiprocessor communication is the same as that in normal asynchronous mode.

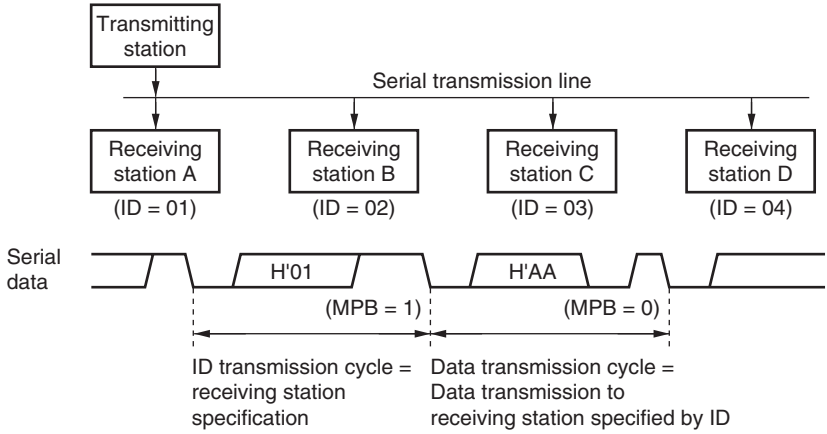
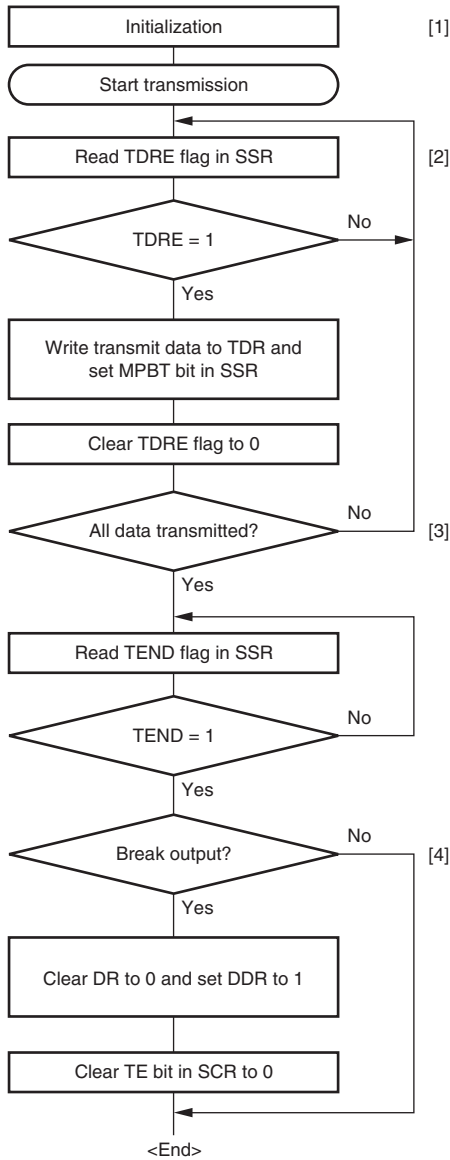


Figure 13.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

13.5.1 Multiprocessor Serial Data Transmission

Figure 13.11 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmission cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI operations are the same as those in asynchronous mode.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1. Finally, clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request, and data is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the port DDR to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 13.11 Sample Multiprocessor Serial Transmission Flowchart

13.5.2 Multiprocessor Serial Data Reception

Figure 13.13 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCR is set to 1, data is skipped until data with a 1 multiprocessor bit is sent. On receiving data with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI interrupt request is generated at this time. All other SCI operations are the same as in asynchronous mode. Figure 13.12 shows an example of SCI operation for multiprocessor format reception.

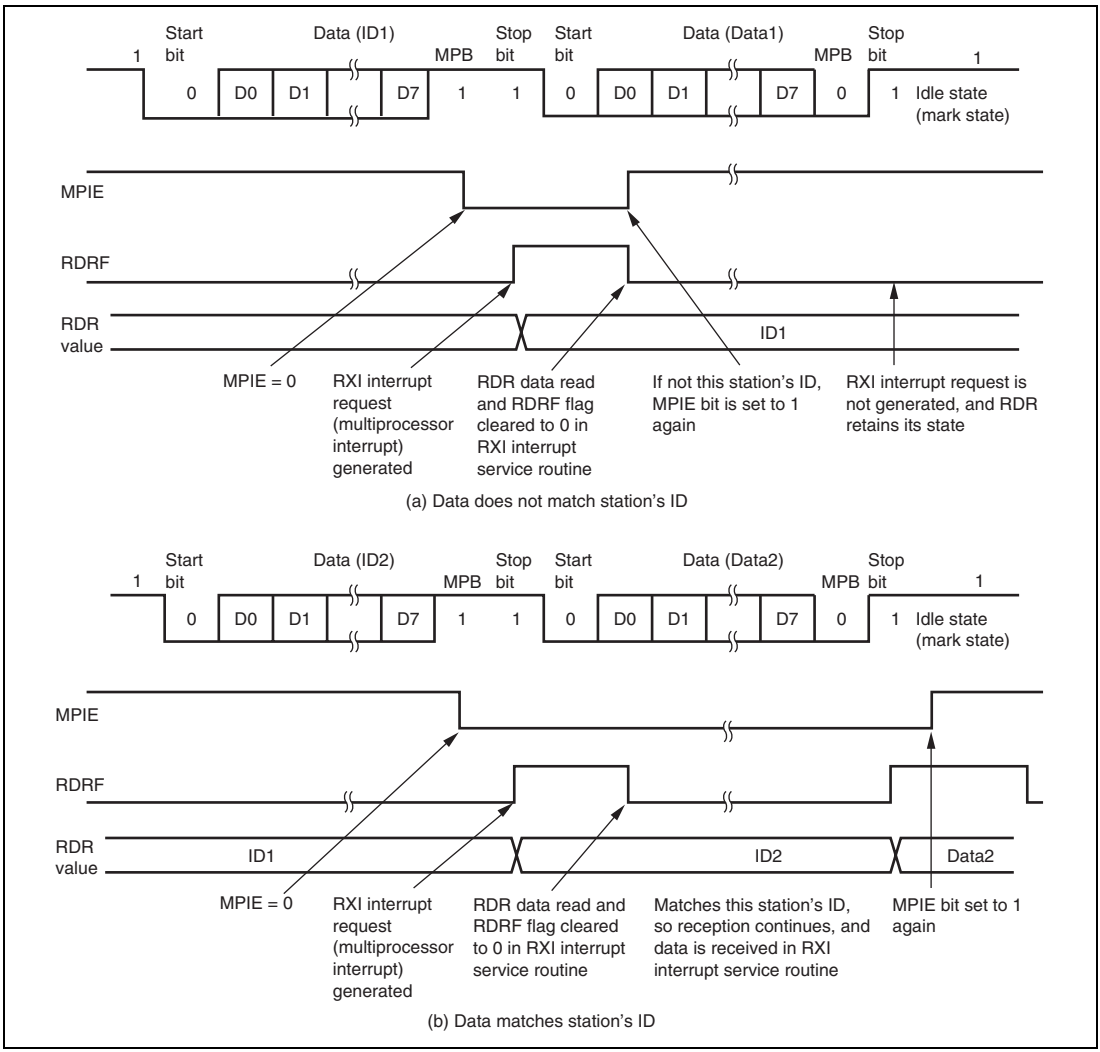
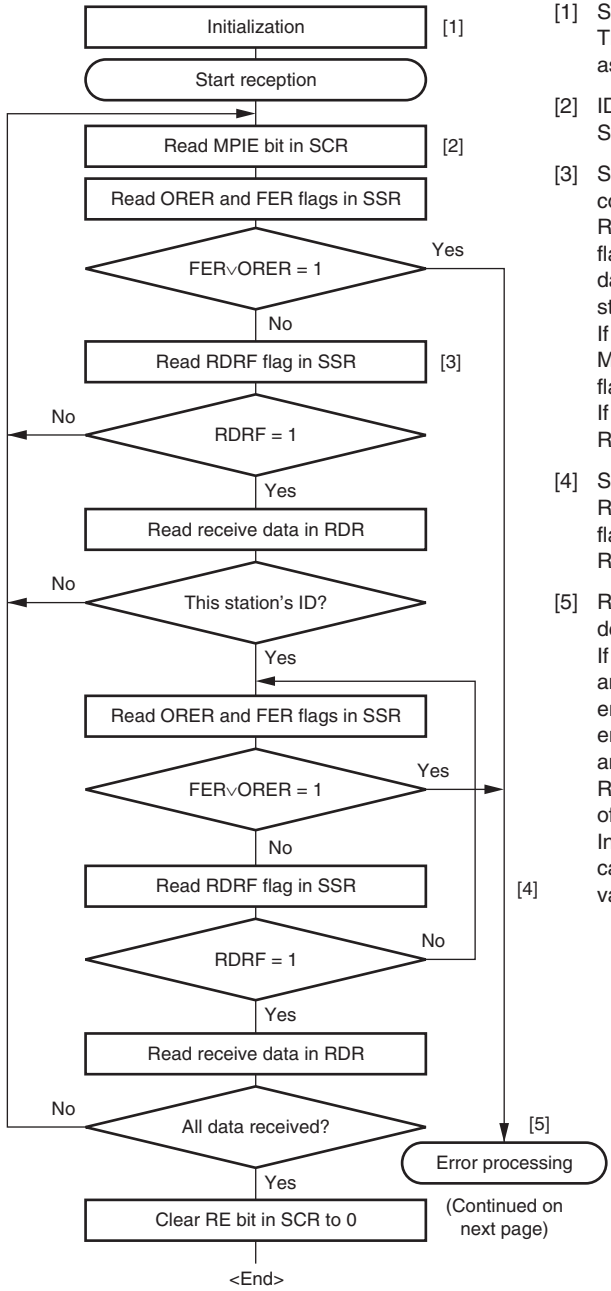


Figure 13.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



[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.

Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (1)

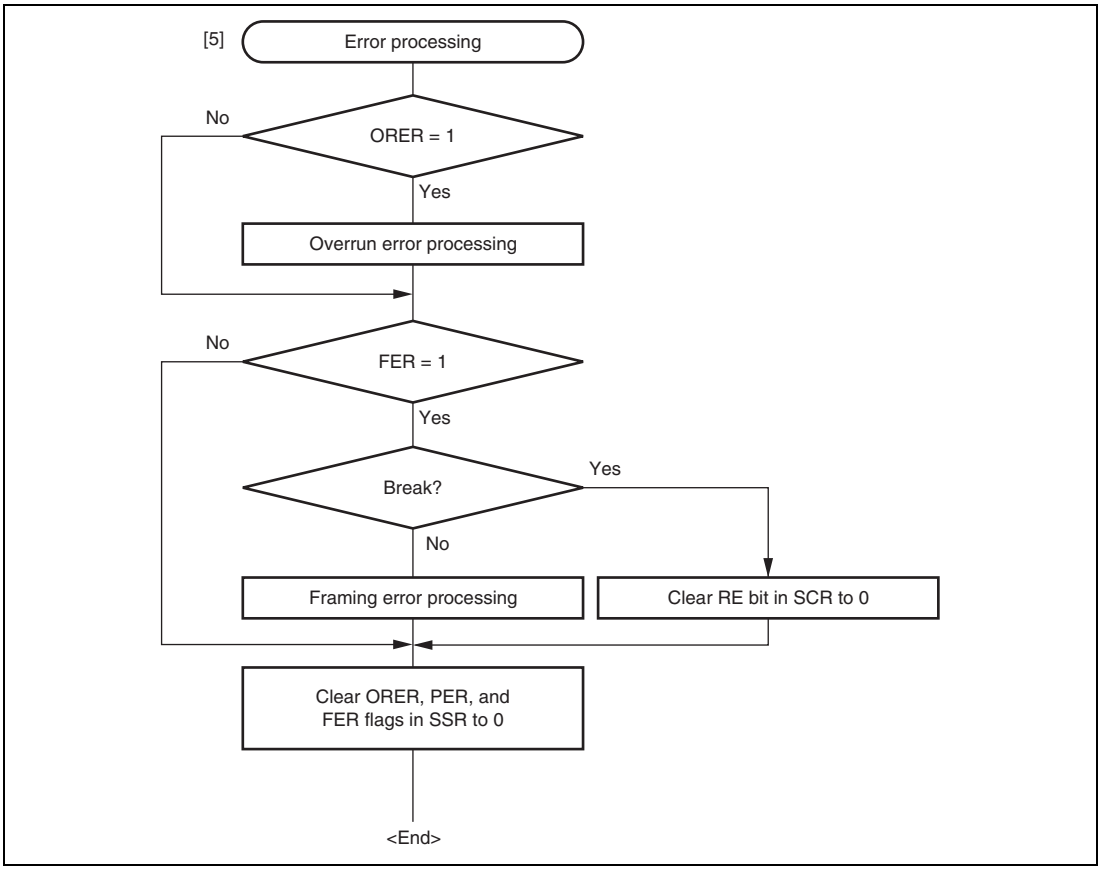


Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (2)

13.6 Operation in Clocked Synchronous Mode

Figure 13.14 shows the general format for clocked synchronous communication. In clocked synchronous mode, data is transmitted or received synchronous with clock pulses. In clocked synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. In clocked synchronous mode, the SCI receives data in synchronous with the rising edge of the serial clock. After 8-bit data is output, the transmission line holds the MSB state. In clocked synchronous mode, no parity or multiprocessor bit is added. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication through the use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer.

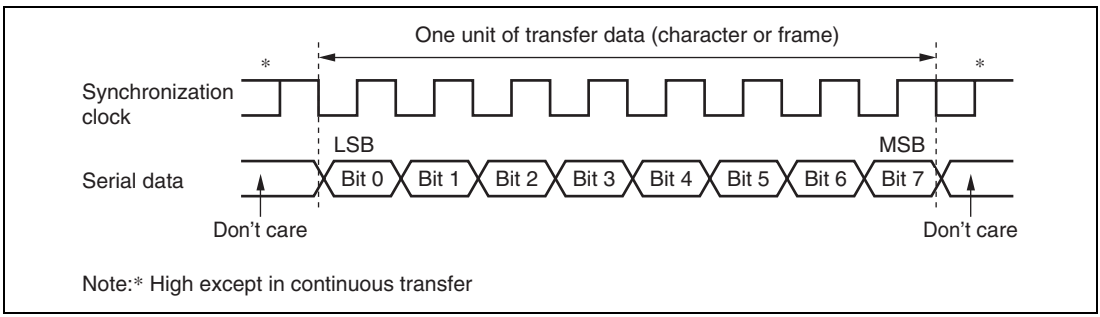


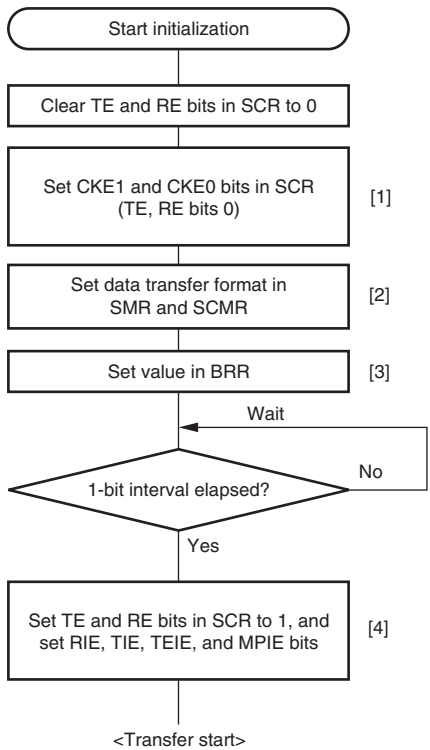
Figure 13.14 Data Format in Synchronous Communication (For LSB-First)

13.6.1 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 CKE0 and CKE1 bits in SCR. When the SCI is operated on an internal clock, the serial clock is output from the SCK pin. Eight serial clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high.

13.6.2 SCI Initialization (Clocked Synchronous Mode)

Before transmitting and receiving data, the TE and RE bits in SCR should be cleared to 0, then the SCI should be initialized as described in a sample flowchart in figure 13.15. When the operating mode, or transfer format, is changed for example, 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 is set to 1. Note that clearing the RE bit to 0 does not change the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR.



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

[2] Set the data transfer 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.

Note:* In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 13.15 Sample SCI Initialization Flowchart

13.6.3 Serial Data Transmission (Clocked Synchronous Mode)

Figure 13.16 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 the flag is 0, the SCI recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a transmit data empty interrupt (TXI) is generated. Continuous transmission is possible because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has been completed.
3. 8-bit data is sent from the TxD pin synchronized with the output clock when output clock mode has been specified, and synchronized with the input clock when use of an external clock has been specified.
4. The SCI checks the TDRE flag at the timing for sending the MSB (bit 7).
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TDRE flag maintains the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated. The SCK pin is fixed high.

Figure 13.17 shows a sample flow chart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) is set to 1. Make sure that the receive error flags are cleared to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.

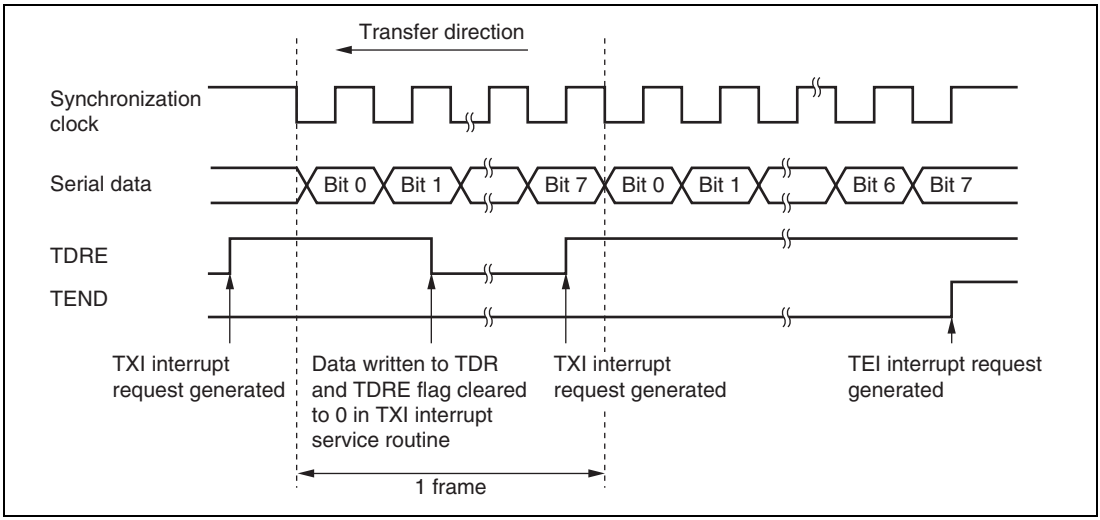
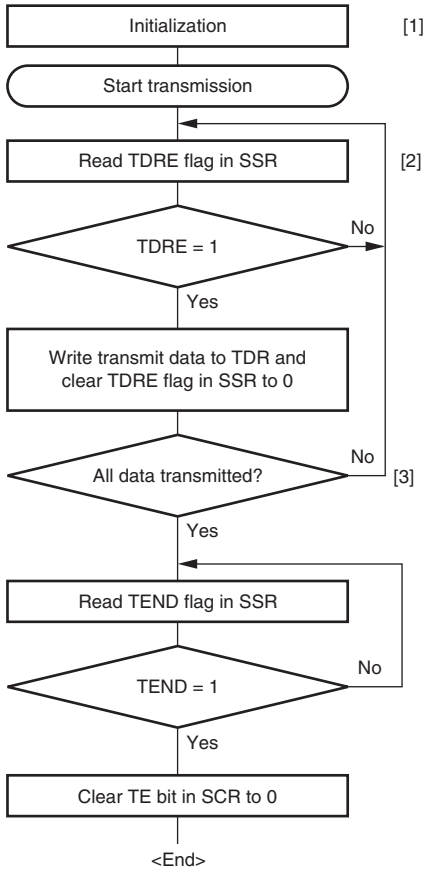


Figure 13.16 Sample SCI Transmission Operation in Clocked Synchronous Mode



[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.
Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request and data is written to TDR.

Figure 13.17 Sample Serial Transmission Flowchart

13.6.4 Serial Data Reception (Clocked Synchronous Mode)

Figure 13.18 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 synchronous with a synchronous clock input or output, starts receiving data, and stores the received data in RSR.
2. If an overrun error occurs (when reception of the next data is completed while the RDRF flag in SSR is still set to 1), the 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, and the RDRF flag remains to be set to 1.
3. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Continuous reception is possible because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished.

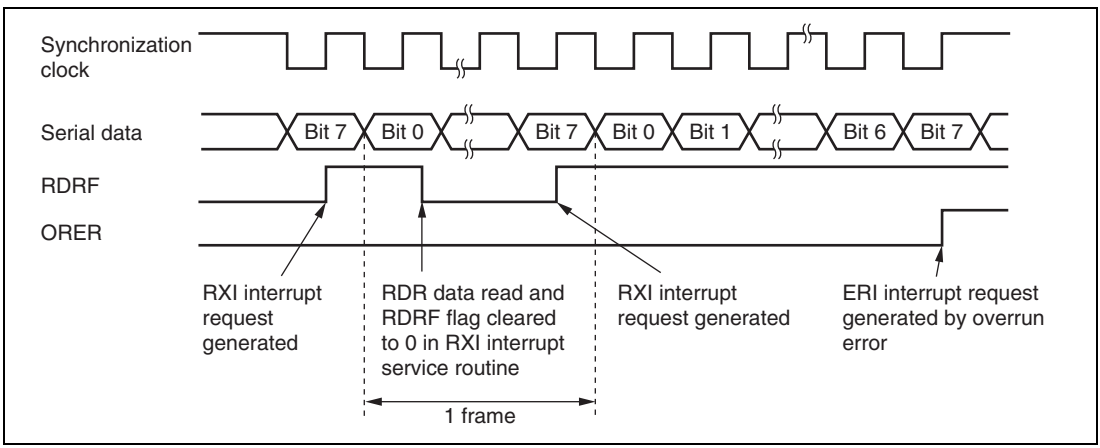
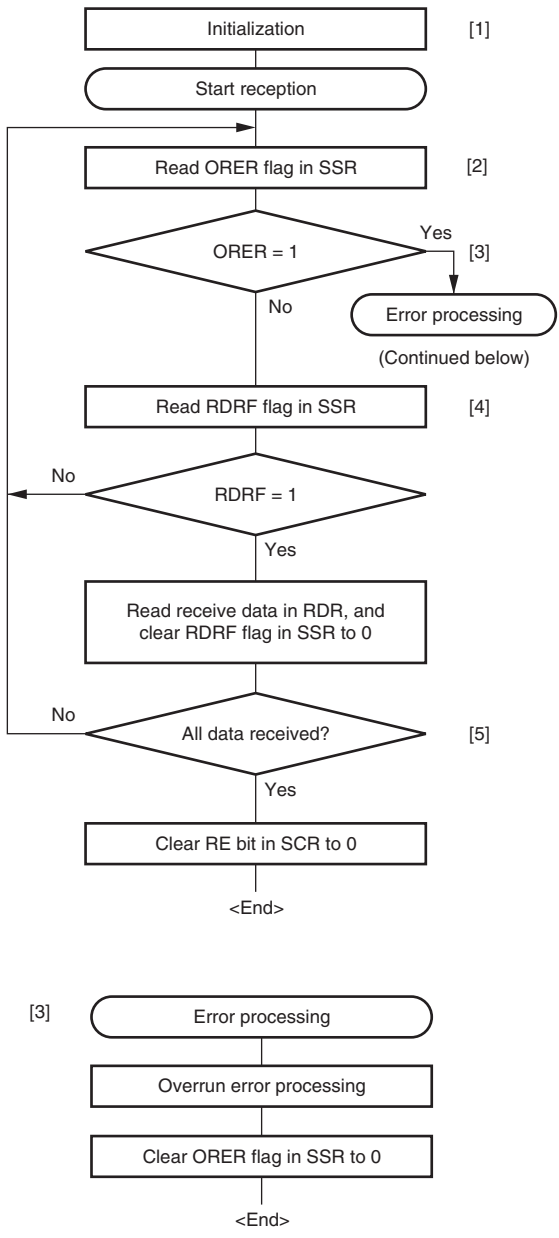


Figure 13.18 Example of SCI Operation in Reception

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 13.19 shows a sample flow chart 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.

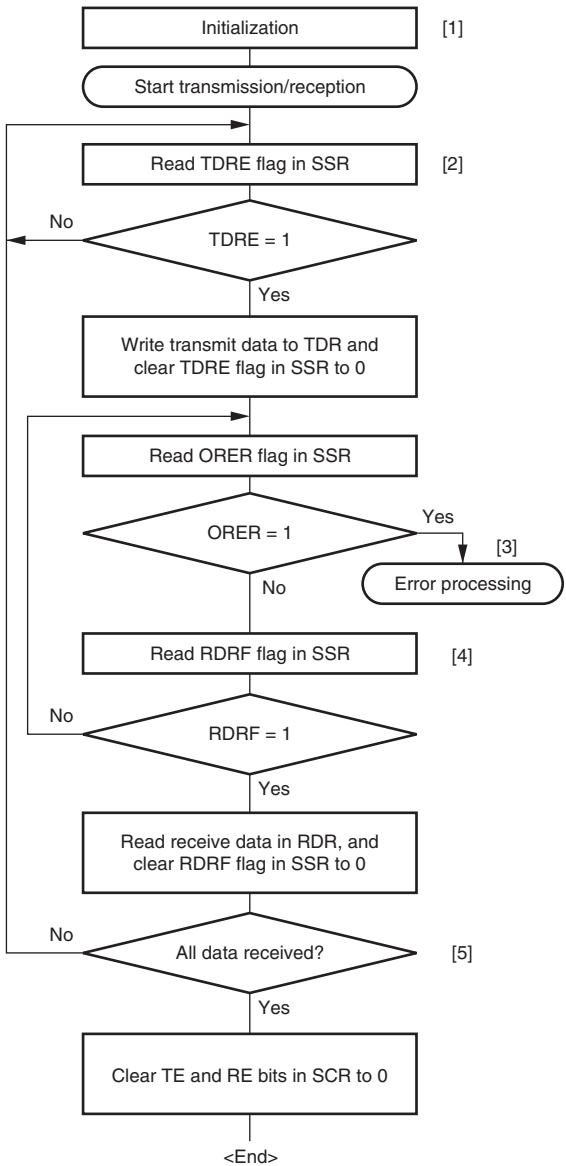
[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0.
Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[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. The RDRF flag is cleared automatically when the DTC is activated by a receive data full interrupt (RXI) request and the RDR value is read.

Figure 13.19 Sample Serial Reception Flowchart

13.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode)

Figure 13.20 shows a sample flowchart for simultaneous serial transmit and receive operations. The following procedure should be used for simultaneous serial data transmit and receive operations after initializing the SCI. 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 are set to 1, clear TE to 0. Then simultaneously set TE and RE to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (ORER, FER, and PER) are cleared to 0, simultaneously set TE and RE 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. Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request and data is written to TDR. Also, the RDRF flag is cleared automatically when the DTC is activated by a receive data full interrupt (RXI) request and the RDR value is read.

Figure 13.20 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations

13.7 Operation in Smart Card Interface

The SCI supports an IC card (smart card) interface that conforms to ISO/IEC 7816-3 (Identification Card) as a serial communication interface extension function. Switching between the normal serial communication interface and the smart card interface mode is carried out by means of a register setting.

13.7.1 Pin Connection Example

Figure 13.21 shows an example of connection with the smart card. In communication with an IC card, as both transmission and reception are carried out on a single data transmission line, the TxD pin and RxD pin should be connected to the LSI pin. The data transmission line should be pulled up to the V_{CC} power supply with a resistor. If an IC card is not connected, and the TE and RE bits are both set to 1, closed transmission/reception is possible, enabling self-diagnosis to be carried out. When the clock generated on the smart card interface is used by an IC card, the SCK pin output is input to the CLK pin of the IC card. This LSI port output is used as the reset signal.

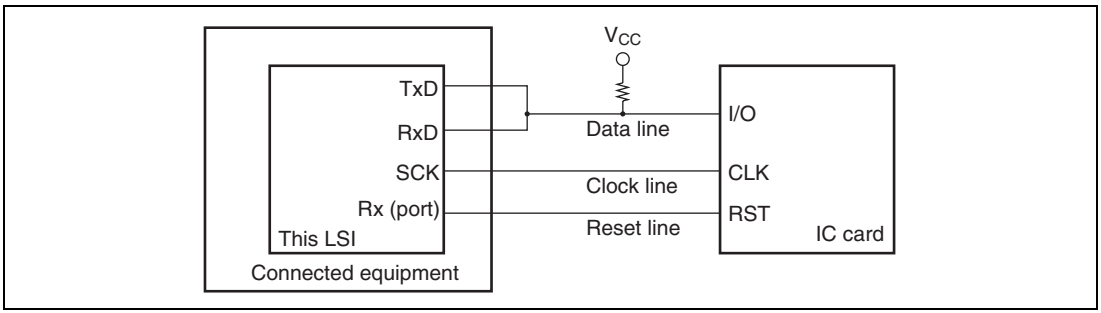


Figure 13.21 Schematic Diagram of Smart Card Interface Pin Connections

13.7.2 Data Format (Except for Block Transfer Mode)

Figure 13.22 shows the transfer data format in smart card interface mode.

- One frame consists of 8-bit data plus a parity bit in asynchronous mode.
- In transmission, a guard time of at least 2 etu (Elementary Time Unit: the time for transfer of one bit) is left between the end of the parity bit and the start of the next frame.
- If a parity error is detected during reception, a low error signal level is output for one etu period, 10.5 etu after the start bit.
- If an error signal is sampled during transmission, the same data is retransmitted automatically after a delay of 2 etu or longer.

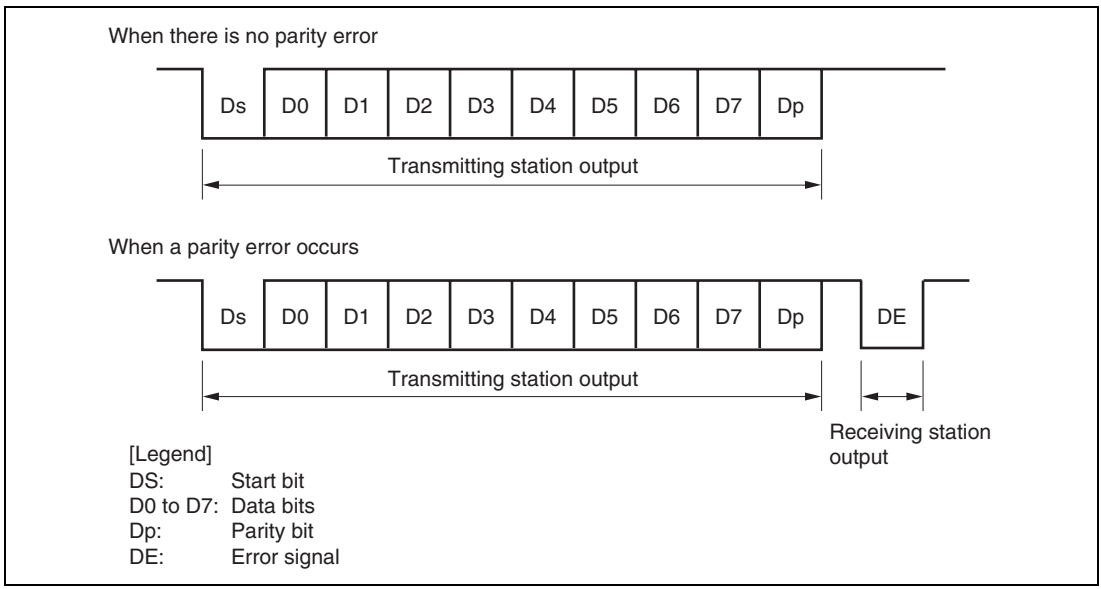


Figure 13.22 Normal Smart Card Interface Data Format

Data transfer with other types of IC cards (direct convention and inverse convention) are performed as described in the following.

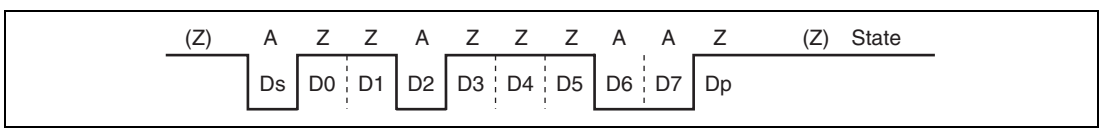


Figure 13.23 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0)

With the direction convention type IC and the above sample start character, the logic 1 level corresponds to state Z and the logic 0 level to state A, and transfer is performed in LSB-first order. The start character data above is H'3B. For the direct convention type, clear the SDIR and SINV bits in SCMR to 0. According to smart card regulations, clear the O/$\overline{E}$ bit in SMR to 0 to select even parity mode.

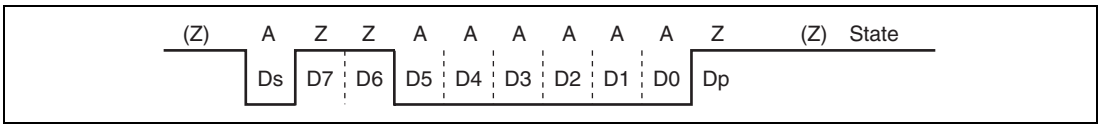


Figure 13.24 Inverse Convention (SDIR = SINV = O/$\overline{E}$ = 1)

With the inverse convention type, the logic 1 level corresponds to state A and the logic 0 level to state Z, and transfer is performed in MSB-first order. The start character data for the above is H'3F. For the inverse convention type, set the SDIR and SINV bits in SCMR to 1. According to smart card regulations, even parity mode is the logic 0 level of the parity bit, and corresponds to state Z. In this LSI, the SINV bit inverts only data bits D0 to D7. Therefore, set the O/$\overline{E}$ bit in SMR to 1 to invert the parity bit for both transmission and reception.

13.7.3 Block Transfer Mode

Operation in block transfer mode is the same as that in SCI asynchronous mode, except for the following points.

- In reception, though the parity check is performed, no error signal is output even if an error is detected. However, the PER bit in SSR is set to 1 and must be cleared before receiving the parity bit of the next frame.
- In transmission, a guard time of at least 1 etu is left between the end of the parity bit and the start of the next frame.
- In transmission, because retransmission is not performed, the TEND flag is set to 1, 11.5 etu after transmission start.
- As with the normal smart card interface, the ERS flag indicates the error signal status, but since error signal transfer is not performed, this flag is always cleared to 0.

13.7.4 Receive Data Sampling Timing and Reception Margin in Smart Card Interface Mode

In smart card interface mode, the SCI operates on a basic clock with a frequency of 32, 64, 372, or 256 times the transfer rate (fixed at 16 times in normal asynchronous mode) as determined by bits BCP1 and BCP0. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. As shown in figure 13.25, by sampling receive data at the rising-edge of the 16th, 32nd, 186th, or 128th pulse of the basic clock, data can be latched at the middle of the bit. The reception margin is given by the following formula.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|\,D - 0.5\,|}{N}\,(1 + F) \right| \times 100\%$$

Where M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, and 256)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5 and N = 372 in the above formula, the reception margin formula is as follows.

$$M = (0.5 - 1/2 \times 372) \times 100\%$$
$$= 49.866\%$$

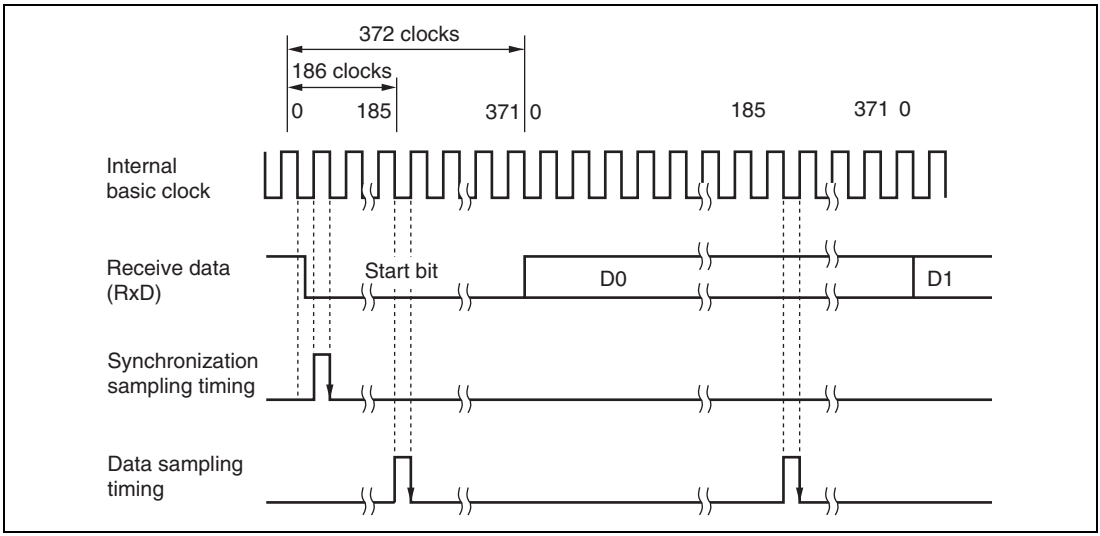


Figure 13.25 Receive Data Sampling Timing in Smart Card Mode (Using Clock of 372 Times the Transfer Rate)

13.7.5 Initialization

Before transmitting and receiving data, initialize the SCI as described below. Initialization is also necessary when switching from transmit mode to receive mode, or vice versa.

1. Clear the TE and RE bits in SCR to 0.
2. Clear the error flags ERS, PER, and ORER in SSR to 0.
3. Set the GM, BLK, O/$\overline{E}$, BCP0, BCP1, CKS0, and CKS1 bits in SMR. Set the PE bit to 1.
4. Set the SMIF, SDIR, and SINV bits in SCMR.
   When the SMIF bit is set to 1, the TxD and RxD pins are both switched from ports to SCI pins, and are placed in the high-impedance state.
5. Set the value corresponding to the bit rate in BRR.
6. Set the CKE0 and CKE1 bits in SCR. Clear the TIE, RIE, TE, RE, MPIE, and TEIE bits to 0.
   If the CKE0 bit is set to 1, the clock is output from the SCK pin.
7. Wait at least one bit interval, then set the TIE, RIE, TE, and RE bits in SCR. Do not set the TE bit and RE bit at the same time, except for self-diagnosis.

To switch from receive mode to transmit mode, after checking that the SCI has finished reception, initialize the SCI, and set RE to 0 and TE to 1. Whether SCI has finished reception or not can be checked with the RDRF, PER, or ORER flags. To switch from transmit mode to receive mode,

after checking that the SCI has finished transmission, initialize the SCI, and set TE to 0 and RE to 1. Whether SCI has finished transmission or not can be checked with the TEND flag.

13.7.6 Data Transmission (Except for Block Transfer Mode)

As data transmission in smart card interface mode involves error signal sampling and retransmission processing, the operations are different from those in normal serial communication interface mode (except for block transfer mode). Figure 13.26 illustrates the retransfer operation when the SCI is in transmit mode.

1. If an error signal is sent back from the receiving end after transmission of one frame is complete, the ERS bit in SSR is set to 1. If the RIE bit in SCR is enabled at this time, an ERI interrupt request is generated. The ERS bit in SSR should be kept cleared to 0 until the next parity bit is sampled.
2. The TEND bit in SSR is not set for a frame in which an error signal indicating an abnormality is received. Data is retransferred from TDR to TSR, and retransmitted automatically.
3. If an error signal is not sent back from the receiving end, the ERS bit in SSR is not set. Transmission of one frame, including a retransfer, is judged to have been completed, and the TEND bit in SSR is set to 1. If the TIE bit in SCR is enabled at this time, a TXI interrupt request is generated. Writing transmit data to TDR transfers the next transmit data.

Figure 13.28 shows a flowchart for transmission. The sequence of transmit operations can be performed automatically by specifying the DTC to be activated with a TXI interrupt source. In a transmit operation, the TDRE flag is set to 1 at the same time as the TEND flag in SSR is set, and a TXI interrupt will be generated if the TIE bit in SCR has been set to 1. If the TXI request is designated beforehand as a DTC activation source, the DTC will be activated by the TXI request, and transfer of the transmit data will be carried out. The TDRE and TEND flags are automatically cleared to 0 when data is transferred by the DTC. In the event of an error, the SCI retransmits the same data automatically. During this period, the TEND flag remains cleared to 0 and the DTC is not activated. Therefore, the SCI and DTC will automatically transmit the specified number of bytes in the event of an error, including retransmission. However, the ERS flag is not cleared automatically when an error occurs, and so the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the event of an error, and the ERS flag will be cleared.

When performing transfer using the DTC, it is essential to set and enable the DTC before carrying out SCI setting. For details of the DTC setting procedures, see section 8, Data Transfer Controller (DTC).

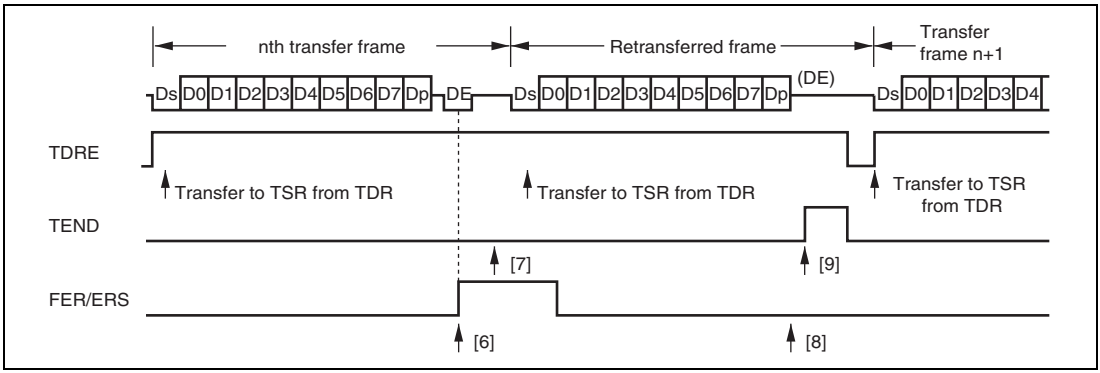


Figure 13.26 Retransfer Operation in SCI Transmit Mode

The timing for setting the TEND flag depends on the value of the GM bit in SMR. The TEND flag set timing is shown in figure 13.27.

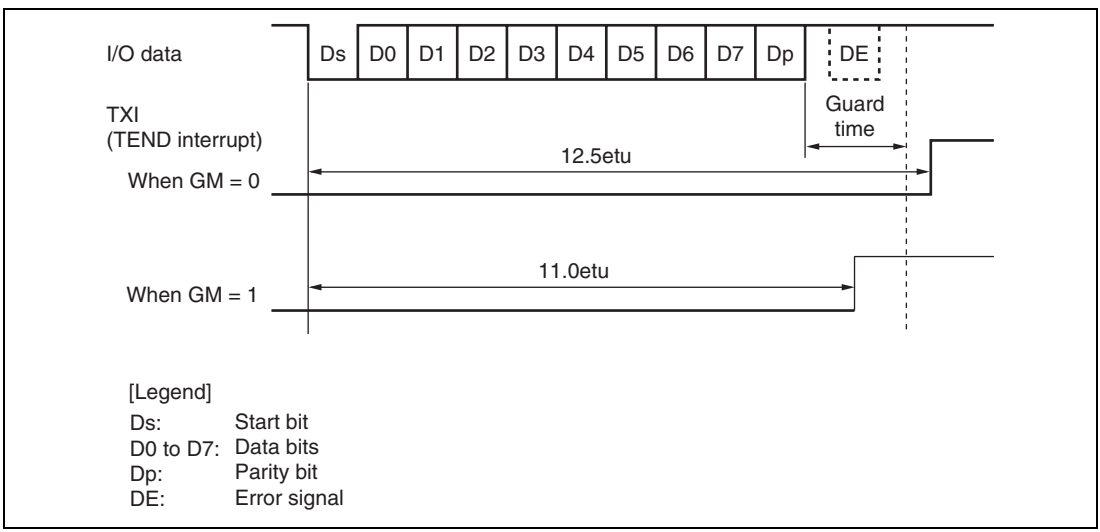


Figure 13.27 TEND Flag Generation Timing in Transmission Operation

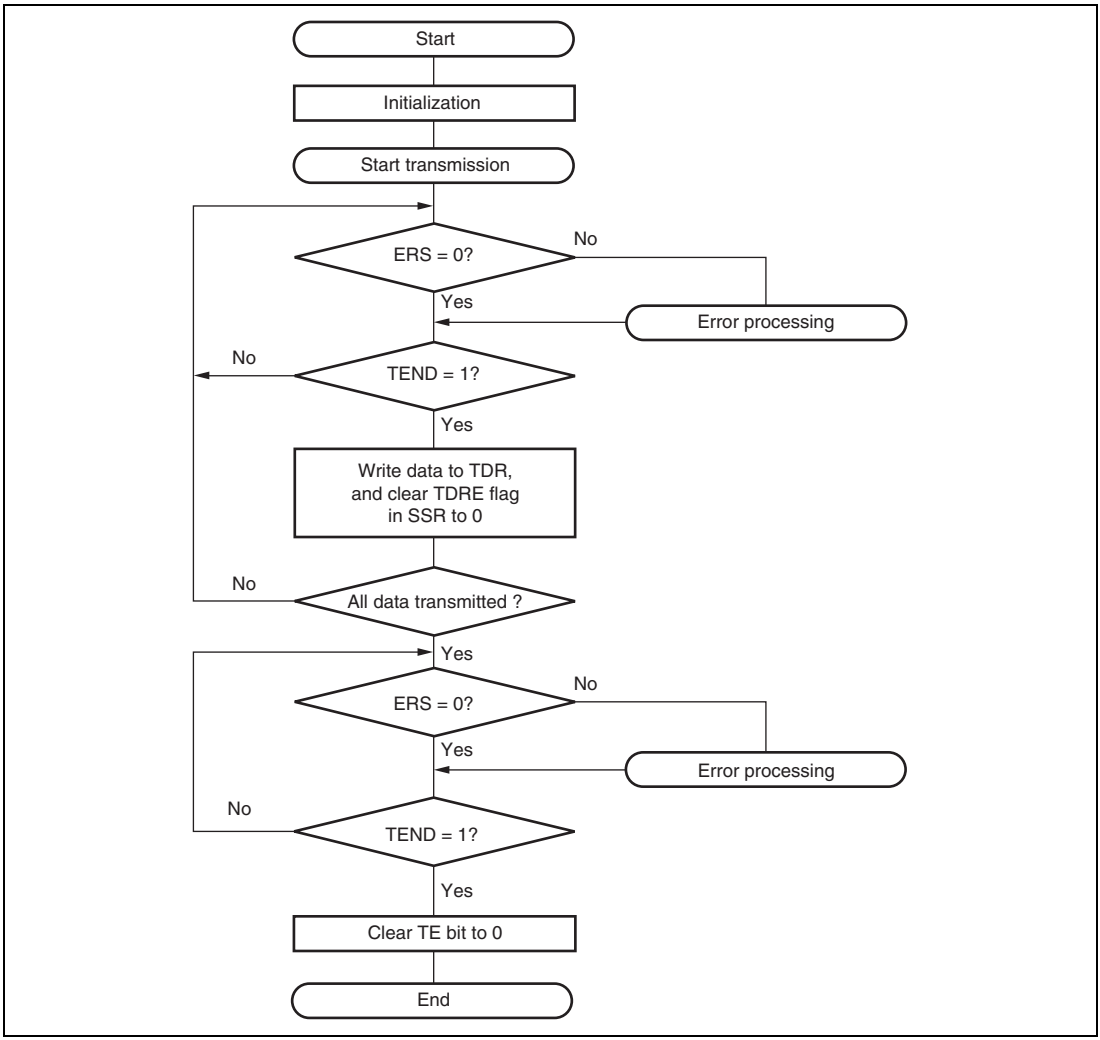


Figure 13.28 Example of Transmission Processing Flow

13.7.7 Serial Data Reception (Except for Block Transfer Mode)

Data reception in smart card interface mode uses the same operation procedure as for normal serial communication interface mode. Figure 13.29 illustrates the retransfer operation when the SCI is in receive mode.

1. If an error is found when the received parity bit is checked, the PER bit in SSR is automatically set to 1. If the RIE bit in SCR is set at this time, an ERI interrupt request is generated. The PER bit in SSR should be kept cleared to 0 until the next parity bit is sampled.
2. The RDRF bit in SSR is not set for a frame in which an error has occurred.
3. If no error is found when the received parity bit is checked, the PER bit in SSR is not set to 1, the receive operation is judged to have been completed normally, and the RDRF flag in SSR is automatically set to 1. If the RIE bit in SCR is enabled at this time, an RXI interrupt request is generated.

Figure 13.30 shows a flowchart for reception. A sequence of receive operations can be performed automatically by specifying the DTC to be activated using an RXI interrupt source. In a receive operation, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If the RXI request is designated beforehand as a DTC activation source, the DTC will be activated by the RXI request, and the receive data will be transferred. The RDRF flag is cleared to 0 automatically when data is transferred by the DTC. If an error occurs in receive mode and the ORER or PER flag is set to 1, a transfer error interrupt (ERI) request will be generated. Hence, so the error flag must be cleared to 0. In the event of an error, the DTC is not activated and receive data is skipped. Therefore, receive data is transferred for only the specified number of bytes in the event of an error. Even when a parity error occurs in receive mode and the PER flag is set to 1, the data that has been received is transferred to RDR and can be read from there.

Note: For details on receive operations in block transfer mode, see section 13.4, Operation in Asynchronous Mode.

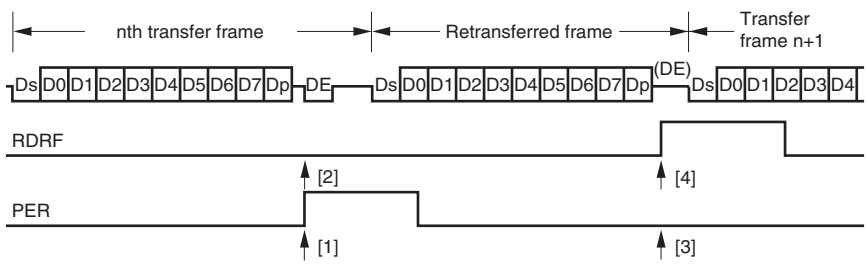


Figure 13.29 Retransfer Operation in SCI Receive Mode

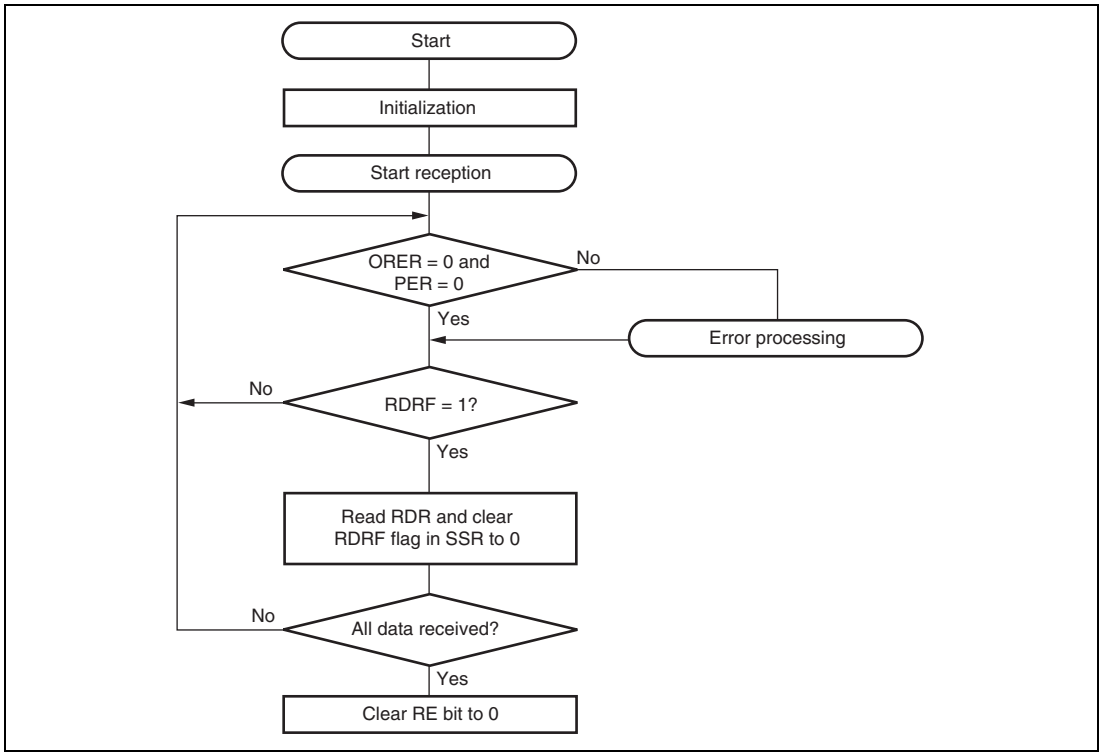


Figure 13.30 Example of Reception Processing Flow

13.7.8 Clock Output Control

When the GM bit in SMR is set to 1, the clock output level can be fixed with bits CKE0 and CKE1 in SCR. At this time, the minimum clock pulse width can be made the specified width. Figure 13.31 shows the timing for fixing the clock output level. In this example, GM is set to 1, CKE1 is cleared to 0, and the CKE0 bit is controlled.

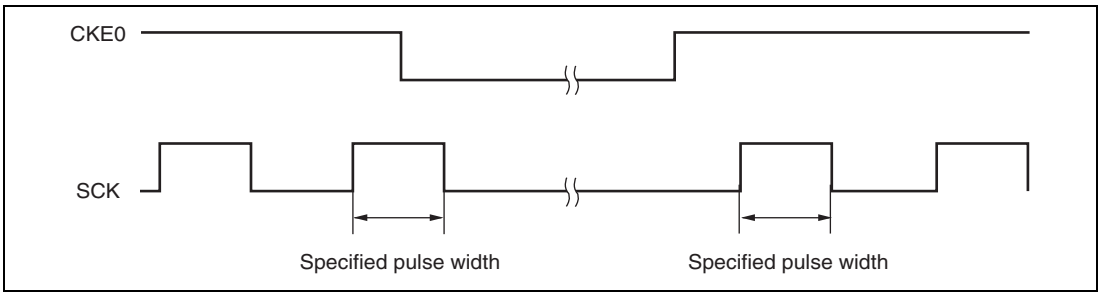


Figure 13.31 Timing for Fixing Clock Output Level

When turning on the power or switching between smart card interface mode and software standby mode, the following procedures should be followed in order to maintain the clock duty cycle.

Powering On: To secure clock duty cycle from power-on, the following switching procedure should be followed.

1. The initial state is port input and high impedance. Use a pull-up resistor or pull-down resistor to fix the potential.
2. Fix the SCK pin to the specified output level with the CKE1 bit in SCR.
3. Set SMR and SCMR, and switch to smart card mode operation.
4. Set the CKE0 bit in SCR to 1 to start clock output.

When changing from smart card interface mode to software standby mode:

1. Set the data register (DR) and data direction register (DDR) corresponding to the SCK pin to the value for the fixed output state in software standby mode.
2. Write 0 to the TE bit and RE bit in the serial control register (SCR) to halt transmit/receive operation. At the same time, set the CKE1 bit to the value for the fixed output state in software standby mode.
3. Write 0 to the CKE0 bit in SCR to halt the clock.
4. Wait for one serial clock period.

During this interval, clock output is fixed at the specified level, with the duty cycle preserved.

5. Make the transition to the software standby state.

When returning to smart card interface mode from software standby mode:

1. Exit the software standby state.
2. Write 1 to the CKE0 bit in SCR and output the clock. Signal generation is started with the normal duty cycle.

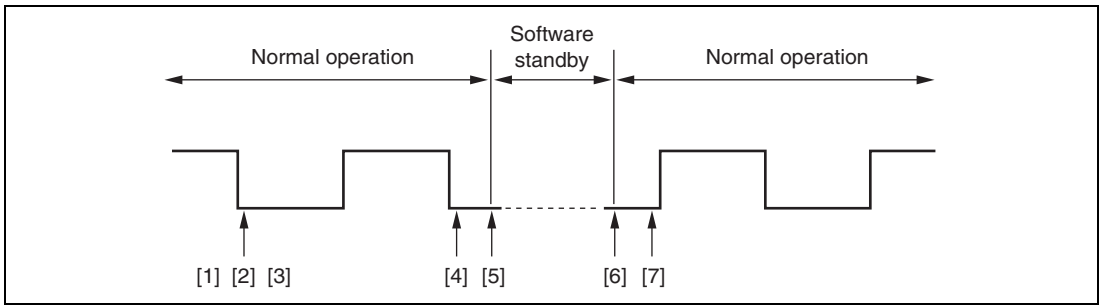


Figure 13.32 Clock Halt and Restart Procedure

13.8 Interrupt Sources

13.8.1 Interrupts in Normal Serial Communication Interface Mode

Table 13.12 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. A TXI interrupt can activate the DTC to perform data transfer. The TDRE flag is cleared to 0 automatically when data is transferred by the DTC.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the ORER, PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interrupt request can activate the DTC to transfer data. The RDRF flag is cleared to 0 automatically when data is transferred by the DTC.

A TEI interrupt is requested when the TEND flag is set to 1 and the TEIE bit is set to 1. If a TEI interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt has priority for acceptance. However, 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 13.12 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation
0	ERI_0	Receive Error	ORER, FER, PER	Not possible
	RXI_0	Receive Data Full	RDRF	Possible
	TXI_0	Transmit Data Empty	TDRE	Possible
	TEI_0	Transmission End	TEND	Not possible
1	ERI_1	Receive Error	ORER, FER, PER	Not possible
	RXI_1	Receive Data Full	RDRF	Possible
	TXI_1	Transmit Data Empty	TDRE	Possible
	TEI_1	Transmission End	TEND	Not possible
2	ERI_2	Receive Error	ORER, FER, PER	Not possible
	RXI_2	Receive Data Full	RDRF	Possible
	TXI_2	Transmit Data Empty	TDRE	Possible
	TEI_2	Transmission End	TEND	Not possible

13.8.2 Interrupts in Smart Card Interface Mode

Table 13.13 shows the interrupt sources in smart card interface mode. The transmit end interrupt (TEI) request cannot be used in this mode.

Table 13.13 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation
0	ERI_0	Receive Error, error signal detection	ORER, PER, ERS	Not possible
	RXI_0	Receive Data Full	RDRF	Possible
	TXI_0	Transmit Data Empty	TEND	Possible
1	ERI_0	Receive Error, error signal detection	ORER, PER, ERS	Not possible
	RXI_0	Receive Data Full	RDRF	Possible
	TXI_0	Transmit Data Empty	TEND	Possible
2	ERI_2	Receive Error, error signal detection	ORER, PER, ERS	Not possible
	RXI_2	Receive Data Full	RDRF	Possible
	TXI_2	Transmit Data Empty	TEND	Possible

In smart card interface mode, as in normal serial communication interface mode, transfer can be carried out using the DTC. In transmit operations, the TDRE flag is also set to 1 at the same time as the TEND flag in SSR is set, and a TXI interrupt is generated. If the TXI request is designated beforehand as a DTC activation source, the DTC will be activated by the TXI request, and transmit data will be transferred. The TDRE and TEND flags are automatically cleared to 0 when data is transferred by the DTC. In the event of an error, the SCI retransmits the same data automatically. During this period, the TEND flag remains cleared to 0 and the DTC is not activated. Therefore, the SCI and DTC will automatically transmit the specified number of bytes in the event of an error, including retransmission. However, the ERS flag is not cleared automatically when an error occurs. Hence, the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the event of an error, and the ERS flag will be cleared.

When transferring using the DTC, it is essential to set and enable the DTC before carrying out SCI setting. For details of the DTC setting procedures, see section 8, Data Transfer Controller (DTC).

In receive operations, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If the RXI request is designated beforehand as a DTC activation source, the DTC will be activated by the RXI request, and the receive data will be transferred. The RDRF flag is cleared to

0 automatically when data is transferred by the DTC. If an error occurs, an error flag is set but the RDRF flag is not. Consequently, the DTC is not activated, instead, an ERI interrupt request is sent to the CPU. Therefore, the error flag should be cleared.

13.9 Usage Notes

13.9.1 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 21, Power-Down Modes.

13.9.2 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, setting the FER flag, and possibly the PER flag. Note that as the SCI continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

13.9.3 Mark State and Break Detection

When TE 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. 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 TE is set to 1, set both DDR and DR to 1. As TE 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 TE to 0. When TE 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.

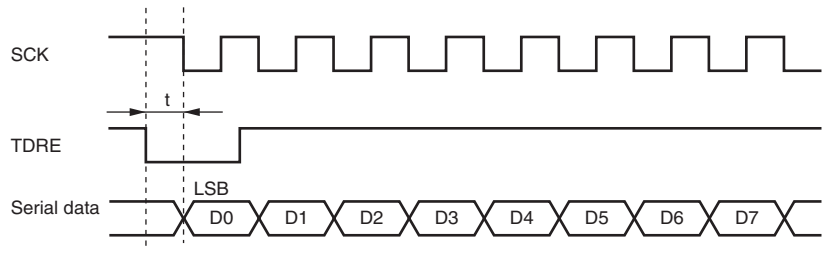
13.9.4 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only)

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

13.9.5 Restrictions on Using DTC

When the external clock source is used as a synchronization clock, update TDR by the DTC and wait for at least five ϕ clock cycles before allowing the transmit clock to be input. If the transmit clock is input within four clock cycles after TDR modification, the SCI may malfunction (figure 13.33).

When using the DTC to read RDR, be sure to set the receive end interrupt source (RXI) as a DTC activation source.



Note: When external clock is supplied, t must be more than four clock cycles.

Figure 13.33 Sample Transmission using DTC in Clocked Synchronous Mode

13.9.6 SCI Operations during Mode Transitions

Transmission: Before making the transition to module stop, software standby, watch, sub-active, or sub-sleep mode, stop all transmit operations (TE = TIE = TEIE = 0). TSR, TDR, and SSR are reset. The states of the output pins during each mode depend on the port settings, and the pins output a high-level signal after mode is cancelled and then the TE is set to 1 again. 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 TE to 1, read SSR, write to TDR, clear TDRE in this order, and then start transmission. To transmit data in a different transmission mode, initialize the SCI first.

Figure 13.34 shows a sample flowchart for mode transition during transmission. Figures 13.35 and 13.36 show the pin states during transmission.

Before making the transition from the transmission mode using DTC transfer to module stop, software standby, watch, sub-active, or sub-sleep mode, stop all transmit operations (TE = TIE = TEIE = 0). Setting TE and TIE to 1 after mode cancellation generates a TXI interrupt request to start transmission using the DTC.

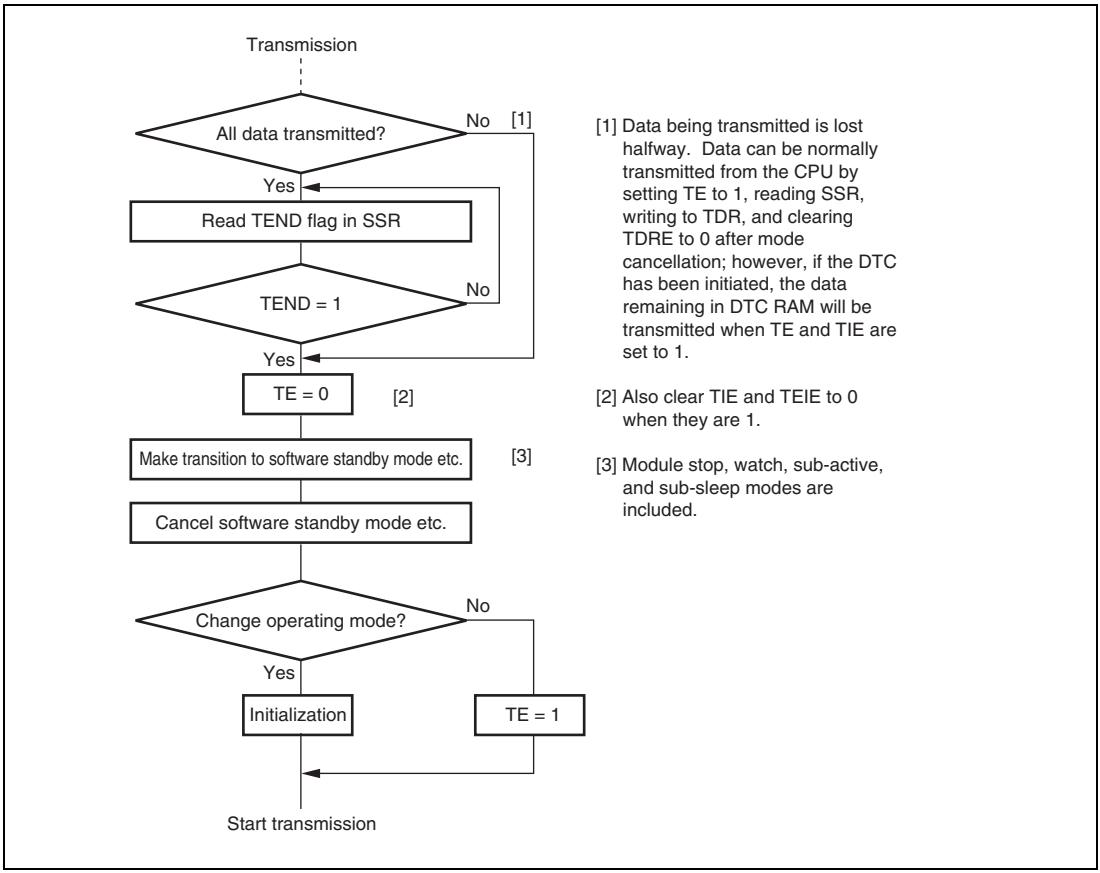


Figure 13.34 Sample Flowchart for Mode Transition during Transmission

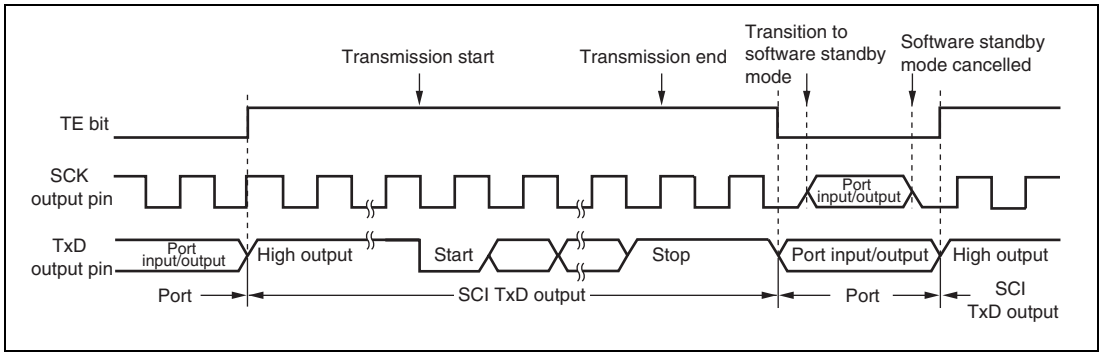


Figure 13.35 Pin States during Transmission in Asynchronous Mode (Internal Clock)

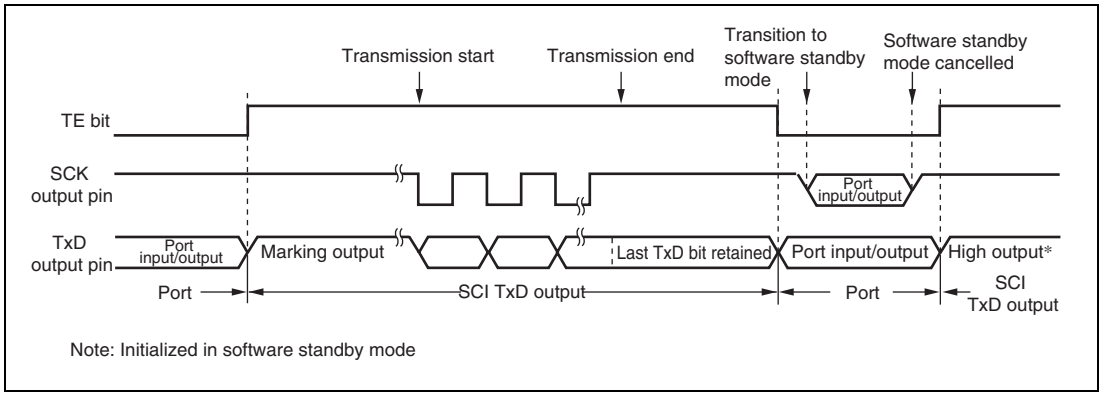


Figure 13.36 Pin States during Transmission in Clocked Synchronous Mode (Internal Clock)

Reception: Before making the transition to module stop, software standby, watch, sub-active, or sub-sleep 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 13.37 shows a sample flowchart for mode transition during reception.

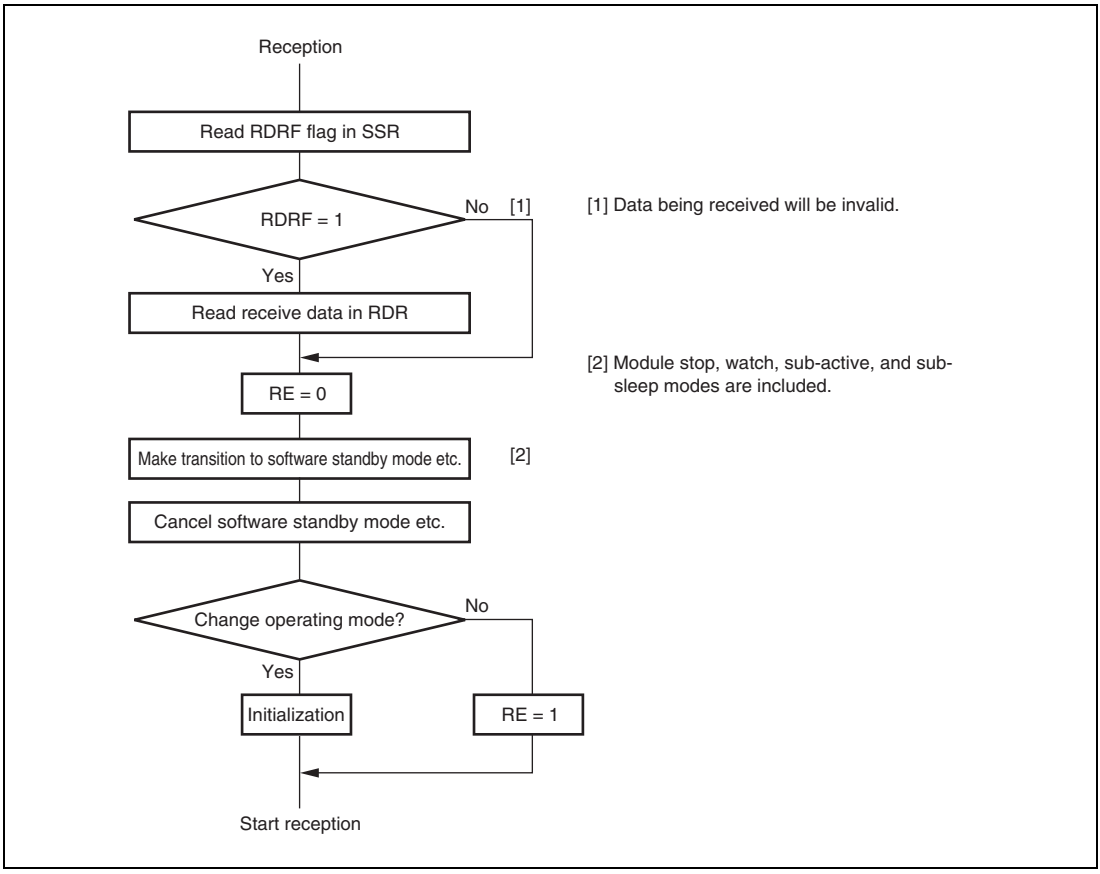


Figure 13.37 Sample Flowchart for Mode Transition during Reception

13.9.7 Notes when Switching from SCK Pin to Port Pin

- Problem in Operation: When DDR and DR are set to 1, SCI clock output is used in clocked synchronous mode, and the SCK pin is changed to the port pin while transmission is ended, port output is enabled after low-level output occurs for one half-cycle.

  When switching the SCK pin to the port pin by making the following settings while DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1, low-level output occurs for one half-cycle.

1. End of serial data transmission
2. TE bit = 0
3. C/$\overline{A}$ bit = 0 ... switchover to port output
4. Occurrence of low-level output (see figure 13.38)

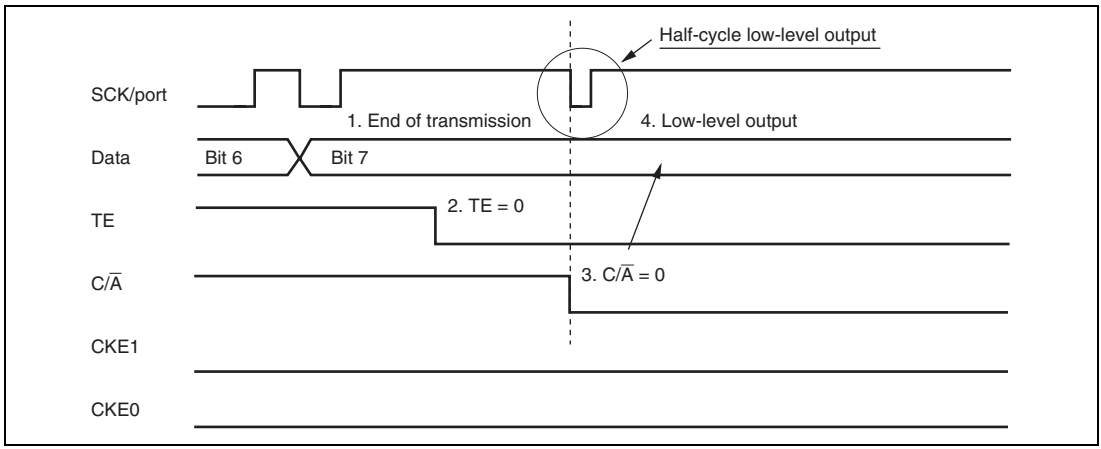


Figure 13.38 Operation when Switching from SCK Pin to Port Pin

- Usage Note: To prevent low-level output occurred when switching the SCK pin to port pin, follow the procedure described below.

  As this sample procedure temporarily places the SCK pin in the input state, the SCK/port pin should be pulled up beforehand with an external circuit.

  With DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1, make the following settings in the order shown.

1. End of serial data transmission
2. TE bit = 0
3. CKE1 bit = 1
4. C/$\overline{A}$ bit = 0 ... switchover to port output
5. CKE1 bit = 0

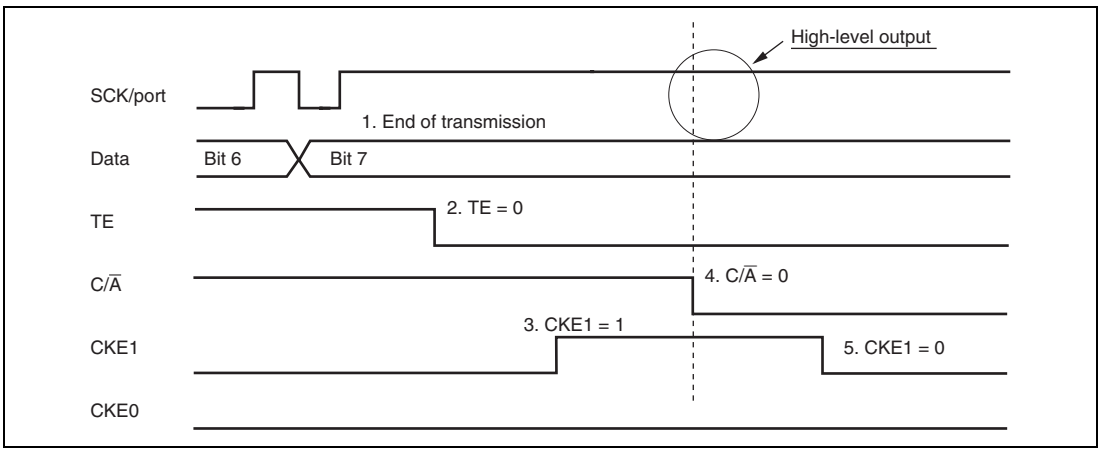


Figure 13.39 Operation when Switching from SCK Pin to Port Pin (Example of Preventing Low-Level Output)

Section 14 I²C Bus Interface

This LSI has a two-channel I²C bus interface.

The conforms to and provides a subset of the Philips I²C bus (inter-IC bus) interface functions. The register configuration that controls the I²C bus differs partly from the Philips configuration, however.

Each I²C bus interface channel uses only one data line (SDA) and one clock line (SCL) to transfer data, saving board and connector space.

14.1 Features

- Selection of addressing format or non-addressing format
  - I²C bus format: addressing format with acknowledge bit, for master/slave operation
  - Serial format: non-addressing format without acknowledge bit, for master operation only
- Conforms to Philips I²C bus interface (I²C bus format)
- Two ways of setting slave address (I²C bus format)
- Start and stop conditions generated automatically in master mode (I²C bus format)
- Selection of acknowledge output levels when receiving (I²C bus format)
- Automatic loading of acknowledge bit when transmitting (I²C bus format)
- Wait function in master mode (I²C bus format)
  - A wait can be inserted by driving the SCL pin low after data transfer, excluding acknowledgement. The wait can be cleared by clearing the interrupt flag.
- Wait function in slave mode (I²C bus format)
  - A wait request can be generated by driving the SCL pin low after data transfer, excluding acknowledgement. The wait request is cleared when the next transfer becomes possible.
- Three interrupt sources
  - Data transfer end (including transmission mode transition with I²C bus format and address reception after loss of master arbitration)
  - Address match: when any slave address matches or the general call address is received in slave receive mode (I²C bus format)
  - Stop condition detection
- Selection of 16 internal clocks (in master mode)
- Direct bus drive (with SCL and SDA pins)
  - Two pins—P35/SCL0 and P34/SDA0—(normally NMOS push-pull outputs) function as NMOS open-drain outputs when the bus drive function is selected.

— Two pins—P33/SCL1 and P32/SDA1—(normally CMOS pins) function as NMOS-only outputs when the bus drive function is selected.

Figure 14.1 shows a block diagram of the I²C bus interface.

Figure 14.2 shows an example of I/O pin connections to external circuits. Channel 0 I/O pins are NMOS open drains, and it is possible to apply voltages in excess of the power supply (V_{CC}) voltage for this LSI. Set the upper limit of voltage applied to the power supply (V_{CC}) power supply range + 0.3 V, i.e. 5.8 V. Channel 1 I/O pins are driven solely by NMOS, so in terms of appearance they carry out the same operations as an NMOS open drain. However, the voltage which can be applied to the I/O pins depends on the voltage of the power supply (V_{CC}) of this LSI.

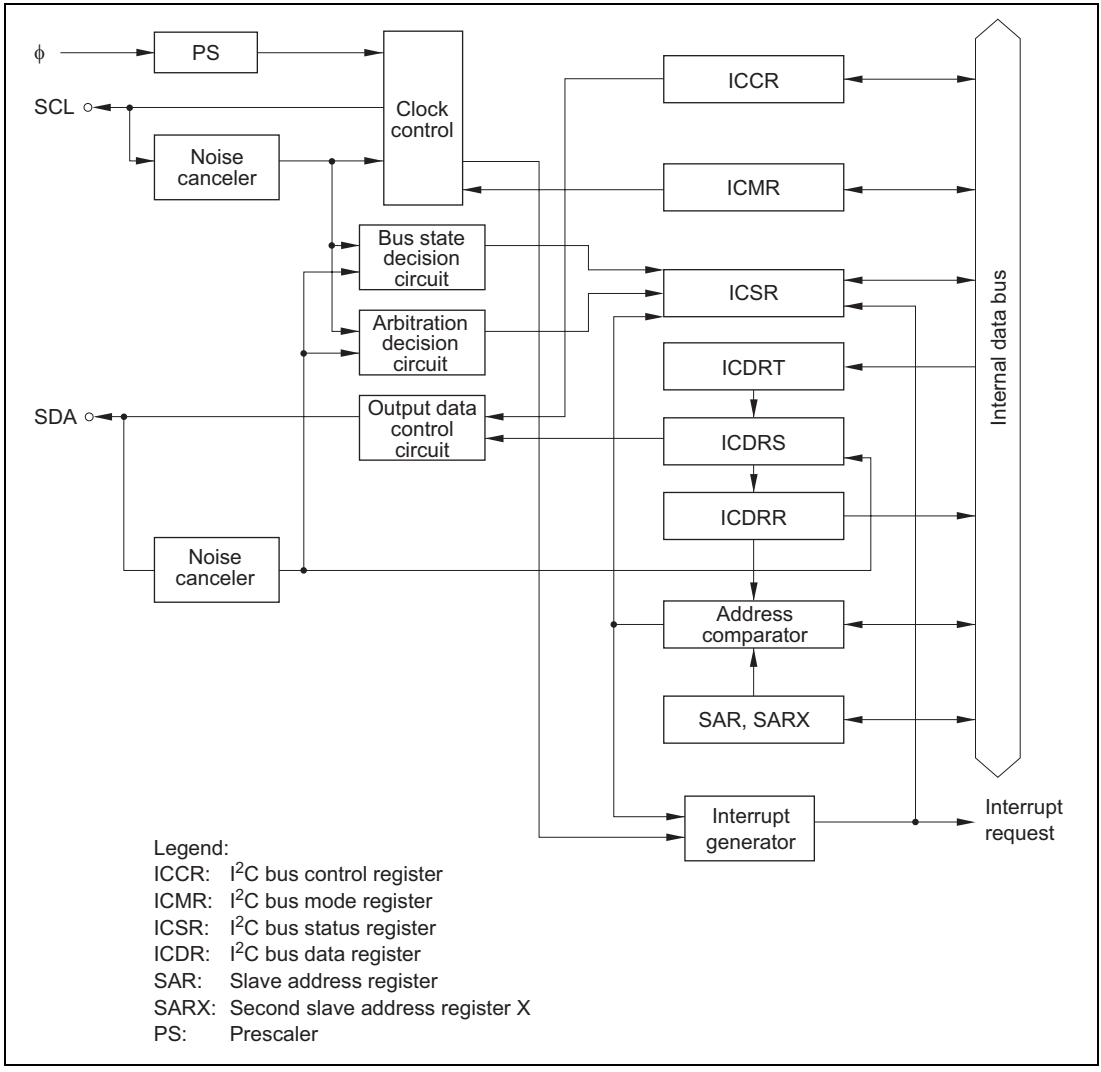


Figure 14.1 Block Diagram of I²C Bus Interface

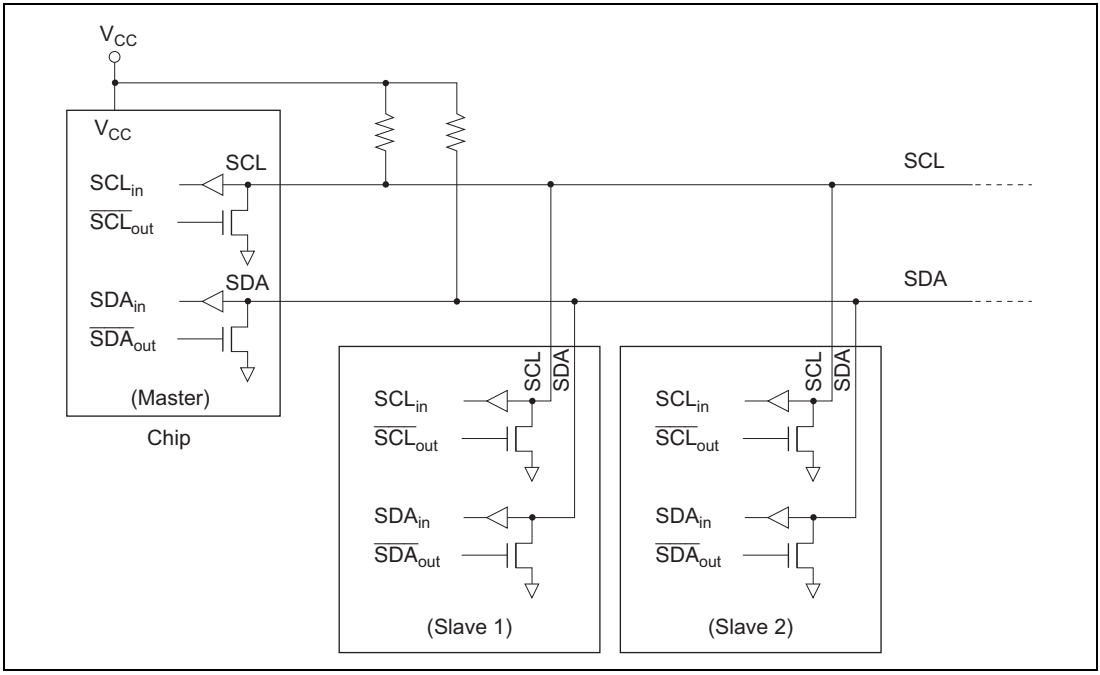


Figure 14.2 I²C Bus Interface Connections (Example: The Chip as Master)

14.2 Input/Output Pins

Table 14.1 summarizes the input/output pins used by the I²C bus interface.

Table 14.1 I²C Bus Interface Pins

Channel	Name	Abbreviation	I/O	Function
0	Serial clock	SCL0	I/O	IIC_0 serial clock input/output
	Serial data	SDA0	I/O	IIC_0 serial data input/output
1	Serial clock	SCL1	I/O	IIC_1 serial clock input/output
	Serial data	SDA1	I/O	IIC_1 serial data input/output

Note: In the text, the channel subscript is omitted, and only SCL and SDA are used.

14.3 Register Configuration

The I²C bus interface consists of the following registers per channel.

ICDR, SARX, and ICMR are assigned to the same address. Register selection is performed by means of the IICE bit in the serial control register X (SCRX).

- I²C bus control register (ICCR)
- I²C bus status register (ICSR)
- I²C bus data register (ICDR)
- I²C bus mode register (ICMR)
- Slave address register (SAR)
- Second slave address register (SARX)
- Serial control register X (SCRX)
- DDC switch register (DDCSWR)

14.3.1 I²C Bus Data Register (ICDR)

ICDR is an 8-bit readable/writable register that functions as a transmit data register when transmitting and a receive data register when receiving. ICDR is divided internally into a shift register (ICDRS), receive buffer (ICDRR), and transmit buffer (ICDRT). ICDRS cannot be read or written by the CPU, ICDRR is read-only, and ICDRT is write-only. Data transfers among the three registers are performed automatically in coordination with changes in the bus state, and affect the status of internal flags such as TDRE and RDRF.

If IIC is in transmit mode and the next data is in ICDRT (the TDRE flag is 0) following transmission/reception of one frame of data using ICDRS, data is transferred automatically from ICDRT to ICDRS. If IIC is in receive mode and no previous data remains in ICDRR (the RDRF flag is 0) following transmission/reception of one frame of data using ICDRS, data is transferred automatically from ICDRS to ICDRR.

If the number of bits in a frame, excluding the acknowledge bit, is less than 8, transmit data and receive data are stored differently. Transmit data should be written justified toward the MSB side when MLS = 0, and toward the LSB side when MLS = 1. Receive data bits read from the LSB side should be treated as valid when MLS = 0, and bits read from the MSB side when MLS = 1.

ICDR is assigned to the same address as SARX, and can be written and read only when the ICE bit is set to 1 in ICCR.

The value of ICDR is undefined after a reset.

The TDRE and RDRF flags are set and cleared under the conditions shown below. Setting the TDRE and RDRF flags affects the status of the interrupt flags.

TDRE	Description
0	The next transmit data is in ICDR (ICDRT), or transmission cannot be started (Initial value) [Clearing conditions] • When transmit data is written in ICDR (ICDRT) in transmission mode (TRS = 1) • When a stop condition is detected in the bus line state after a stop condition is issued with the I²C bus format or serial format selected • When a stop condition is detected with the I²C bus format selected • In receive mode (TRS = 0) (A 0 write to TRS during transfer is valid after reception of a frame containing an acknowledge bit)
1	The next transmit data can be written in ICDR (ICDRT) [Setting conditions] • In transmit mode (TRS = 1), when a start condition is detected in the bus line state after a start condition is issued in master mode with the I²C bus format or serial format selected • When data is transferred from ICDRT to ICDRS (Data transfer from ICDRT to ICDRS when TRS = 1 and TDRE = 0, and ICDRS is empty) • In receive mode (TRS = 0), when a switch is made from slave receive mode (TRS =0) to transmit mode (TRS = 1) after detection of a start condition (first time only)

RDRF	Description
0	The data in ICDR (ICDRR) is invalid (Initial value) [Clearing condition] • When ICDR (ICDRR) receive data is read in receive mode
1	The ICDR (ICDRR) receive data can be read [Setting condition] • When data is transferred from ICDRS to ICDRR (Data transfer from ICDRS to ICDRR in case of normal termination with TRS = 0 and RDRF = 0)

14.3.2 Slave Address Register (SAR)

SAR stores the slave address and selects the communication format. When the chip is in slave mode (and the addressing format is selected), if the upper 7 bits of SAR match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device specified by the master device. SAR is assigned to the same address as ICMR, and can be accessed when the ICE bit is cleared to 0 in ICCR.

SAR is initialized to H'00 by a reset and in hardware standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	SVA6	0	R/W	Slave address 6 to 0
6	SVA5	0	R/W	Specifies the slave address
5	SVA4	0	R/W	
4	SVA3	0	R/W	
3	SVA2	0	R/W	
2	SVA1	0	R/W	
1	SVA0	0	R/W	
0	FS	0	R/W	Format select: Used together with the FSX bit in SARX to select the communication format. For the detail, see table 14.2.

14.3.3 Second Slave Address Register (SARX)

SARX stores the second slave address and selects the communication format. When the chip is in slave mode (and the addressing format is selected), if the upper 7 bits of SARX match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device specified by the master device. SARX is assigned to the same address as ICDR, and can be accessed only when the ICE bit is cleared to 0 in ICCR.

Bit	Bit Name	Initial Value	R/W	Description
7	SVA6	0	R/W	Slave address 6 to 0
6	SVA5	0	R/W	Specifies the slave address
5	SVA4	0	R/W	
4	SVA3	0	R/W	
3	SVA2	0	R/W	
2	SVA1	0	R/W	
1	SVA0	0	R/W	
0	FS	0	R/W	Format select: Used together with the FS bit in SAR to select the communication format. For the detail, see table 14.2.

- I²C bus format: addressing format with acknowledge bit
- Synchronous serial format: non-addressing format without acknowledge bit, for master mode only

Table 14.2 Transfer Format

SAR Bit 0	SARX Bit 0	Operating Mode
FS FSX	FS FSX	
0	0	I²C bus format • Enables the slave addresses in SAR and SARX
	1	I²C bus format • Enables the slave address in SAR • Disables the slave address in SARX
1	0	I²C bus format • Enables the slave address in SARX • Disables the slave address in SAR
	1	Clock- synchronous serial format • Disables the slave addresses in SAR and SARX

14.3.4 I²C Bus Mode Register (ICMR)

ICMR sets the communication format and transfer rate. It can only be accessed when the ICE bit in ICCR is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I²C bus format is used.
6	WAIT	0	R/W	Wait Insertion Bit This bit is valid only in master mode with the I²C bus format. The setting of this bit is invalid in slave mode. 0: Data and the acknowledge bit are transferred consecutively with no wait inserted. 1: After the fall of the clock for the final data bit (8th clock), the IRIC flag is set to 1 in ICCR, and a wait state begins (with SCL at the low level). When the IRIC flag is cleared to 0 in ICCR, the wait ends and the acknowledge bit is transferred. For details, refer to section 14.4.7, IRIC Setting Timing and SCL Control.
5	CKS2	0	R/W	Transfer Clock Select 2 to 0 These bits are used only in master mode. These bits select the required transfer rate, together with the IICX1 (IIC_1) and IICX0 (IIC_0) bits in SCRX. Refer to table 14.3.
4	CKS1	0	R/W	
3	CKS0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
2	BC2	0	R/W	Bit Counter 2 to 0
1	BC1	0	R/W	These bits specify the number of bits to be transferred next. Bit BC2 to BC0 settings should be made during an interval between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL line is low.
0	BC0	0	R/W	The bit counter is initialized to 000 when a start condition is detected. The value returns to 000 at the end of a data transfer.

I²C Bus Format	Clocked Synchronous Serial Mode
000: 9 bits	000: 8 bits
001: 2 bits	001: 1 bits
010: 3 bits	010: 2 bits
011: 4 bits	011: 3 bits
100: 5 bits	100: 4 bits
101: 6 bits	101: 5 bits
110: 7 bits	110: 6 bits
111: 8 bits	111: 7 bits

Table 14.3 I²C Transfer Rate

STCR	ICMR				Transfer Rate				
Bits 5 and 6	Bit 5	Bit 4	Bit 3						
IICX	CKS2	CKS1	CKS0	Clock	φ = 5 MHz	φ = 8 MHz	φ = 10 MHz	φ = 16 MHz	φ = 20 MHz
0	0	0	0	φ/28	179 kHz	286 kHz	357 kHz	517 kHz*	714 kHz*
0	0	0	1	φ/40	125 kHz	200 kHz	250 kHz	400 kHz	500 kHz*
0	0	1	0	φ/48	104 kHz	167 kHz	208 kHz	333 kHz	417 kHz*
0	0	1	1	φ/64	78.1 kHz	125 kHz	156 kHz	250 kHz	313 kHz
0	1	0	0	φ/80	62.5 kHz	100 kHz	125 kHz	200 kHz	250 kHz
0	1	0	1	φ/100	50.0 kHz	80.0 kHz	100 kHz	160 kHz	200 kHz
0	1	1	0	φ/112	44.6 kHz	71.4 kHz	89.3 kHz	143 kHz	179 kHz
0	1	1	1	φ/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz
1	0	0	0	φ/56	89.3 kHz	143 kHz	179 kHz	286 kHz	357 kHz
1	0	0	1	φ/80	62.5 kHz	100 kHz	125 kHz	200 kHz	250 kHz
1	0	1	0	φ/96	52.1 kHz	83.3 kHz	104 kHz	167 kHz	208 kHz
1	0	1	1	φ/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz
1	1	0	0	φ/160	31.3 kHz	50.0 kHz	62.5 kHz	100 kHz	125 kHz
1	1	0	1	φ/200	25.0 kHz	40.0 kHz	50.0 kHz	80.0 kHz	100 kHz
1	1	1	0	φ/224	22.3 kHz	35.7 kHz	44.6 kHz	71.4 kHz	89.3 kHz
1	1	1	1	φ/256	19.5 kHz	31.3 kHz	39.1 kHz	62.5 kHz	78.1 kHz

Note: * Outside the I²C bus interface specifications (standard mode: max. 100 kHz; high-speed mode: max. 400 kHz)

14.3.5 I²C Bus Control Register (ICCR)

ICCR controls the I²C bus interface and performs interrupt flag confirmation.

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	Bus Interface Enable 0: I²C bus interface modules are stopped and I²C bus interface module internal state is initialized. SAR and SARX can be accessed. 1: I²C bus interface modules can perform transfer operation, and the ports function as the SCL and SDA input/output pins. ICMR and ICDR can be accessed.
6	IEIC	0	R/W	I²C Bus Interface Interrupt Enable 0: Disables interrupts from the I²C bus interface to the CPU 1: Enables interrupts from the I²C bus interface to the CPU.

Bit	Bit Name	Initial Value	R/W	Description
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode Both these bits will be cleared by hardware when they lose in a bus contention in master mode with the I²C bus format. In slave receive mode with I²C bus format, the R/$\overline{W}$ bit in the first frame immediately after the start condition sets these bits in receive mode or transmit mode automatically by hardware. Modification of the TRS bit during transfer is deferred until transfer is completed, and the changeover is made after completion of the transfer. [MST clearing conditions] 1. When 0 is written by software 2. When lost in bus contention in I²C bus format master mode [MST setting conditions] 1. When 1 is written by software (for MST clearing condition 1) 2. When 1 is written in MST after reading MST = 0 (for MST clearing condition 2) [TRS clearing conditions] 1. When 0 is written by software (except for TRS setting condition 3) 2. When 0 is written in TRS after reading TRS = 1 (for TRS setting condition 3) 3 When the master device in the I²C bus format starts transmission and then fails because of a bus conflict. [TRS setting conditions] 1. When 1 is written by software (except for TRS clearing conditions 3 and 4) 2. When 1 is written in TRS after reading TRS = 0 (for TRS clearing conditions 3 and 4) 3. When 1 is received as the R/$\overline{W}$ bit after the first frame address matching in I²C bus format slave mode

Bit	Bit Name	Initial Value	R/W	Description
3	ACKE	0	R/W	Specifies whether the value of the acknowledge bit returned from the receiving device when using the I²C bus format is to be ignored and continuous transfer is performed, or transfer is to be aborted and error handling, etc., performed if the acknowledge bit is 1. When the ACKE bit is 0, the value of the received acknowledge bit is not indicated by the ACKB bit, which is always 0. In this LSI, the DTC can be used to perform continuous transfer. The DTC is activated when the IRTR interrupt flag is set to 1 (IRTR is one of two interrupt flags, the other being IRIC). When the ACKE bit is 0, the TDRE, IRIC, and IRTR flags are set on completion of data transmission, regardless of the value of the acknowledge bit. When the ACKE bit is 1, the TDRE, IRIC, and IRTR flags are set on completion of data transmission when the acknowledge bit is 0, and the IRIC flag alone is set on completion of data transmission when the acknowledge bit is 1. When the DTC is activated, the TDRE, IRIC, and IRTR flags are cleared to 0 after the specified number of data transfers have been executed. Consequently, interrupts are not generated during continuous data transfer, but if data transmission is completed with a 1 acknowledge bit when the ACKE bit is set to 1, the DTC is not activated and an interrupt is generated, if enabled. Depending on the receiving device, the acknowledge bit may be significant, in indicating completion of processing of the received data, for instance, or may be fixed at 1 and have no significance.

Bit	Bit Name	Initial Value	R/W	Description
2	BBSY	0	R/W	Bus Busy
0	SCP	1	W	Start Condition/Stop Condition Prohibit In master mode: • Writing 0 in BBSY and 0 in SCP: A stop condition is issued • Writing 1 in BBSY and 0 in SCP: A start condition and a restart condition are issued In slave mode: • Writing to the BBSY flag is disabled. [BBSY setting condition] When the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued. [BBSY clearing condition] When the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued. To issue a start/stop condition, use the MOV instruction. The I²C bus interface must be set in master transmit mode before the issue of a start condition. Set MST to 1 and TRS to 1 before writing 1 in BBSY and 0 in SCP. The BBSY flag can be read to check whether the I²C bus (SCL, SDA) is busy or free. The SCP bit is always read as 1. If 1 is written, the data is not stored.

Bit	Bit Name	Initial Value	R/W	Description
1	IRIC	0	R/(W)*	I²C Bus Interface Interrupt Request Flag Indicates that the I²C bus interface has issued an interrupt request to the CPU. IRIC is set at different times depending on the FS bit in SAR, the FSX bit in SARX, and the WAIT bit in ICMR. See section 14.4.7, IRIC Setting Timing and SCL Control. The conditions under which IRIC is set also differ depending on the setting of the ACKE bit in ICCR. [Setting conditions] I²C bus format master mode:· • When a start condition is detected in the bus line state after a start condition is issued (when the ICDRE flag is set to 1 because of first frame transmission)· • When a wait is inserted between the data and acknowledge bit when the WAIT bit is 1 (fall of the 8th transmit/receive clock) • At the end of data transfer (rise of the 9th transmit/receive clock or fall of the 8th transmit/receive clock while a wait is inserted)· • When a slave address is received after bus arbitration is lost (when the AL flag is set to 1) • If 1 is received as the acknowledge bit (when the ACKB bit in ICSR is set to 1) when the ACKE bit is 1· I²C bus format slave mode:· • When the slave address (SVA or SVAX) matches (when the AAS or AASX flag in ICSR is set to 1) and at the end of data transfer up to the subsequent retransmission start condition or stop condition detection (when the TDRE or RDRF flag is set to 1) • When the general call address is detected (when FS=0 and the ADZ flag is set to 1) and at the end of data reception up to the subsequent retransmission start condition or stop condition detection (when the TDRE or RDRF flag is set to 1)

Bit	Bit Name	Initial Value	R/W	Description
1	IRIC	0	R/(W)*	• If 1 is received as the acknowledge bit (when the ACKB bit in ICSR is set to 1) while the ACKE bit is 1·
				• When a stop condition is detected (when the STOP or ESTP flag in ICSR is set to 1)
				• Clocked synchronous serial format and formatless modes:
				• At the end of data transfer (when TDRE or RDRF flag is set to 1)
				• When a start condition is detected with serial format selected
				• When the TDRE or RDRF flag is set to 1 occurs other than the above conditions
				[Clearing conditions]·
				• When 0 is written in IRIC after reading IRIC = 1·
				• When ICDR is read from or written to by the DTC (This may not function as a clearing condition depending on the situation. For details, see the description of the DTC operation given below.)

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

When the DTC is used, IRIC is cleared automatically and transfer can be performed continuously without CPU intervention.

When, with the I²C bus format selected, IRIC is set to 1 and an interrupt is generated, other flags must be checked in order to identify the source that set IRIC to 1. Although each source has a corresponding flag, caution is needed at the end of a transfer.

When the TDRE or RDRF flag is set, the IRTR flag may or may not be set. The IRTR flag (the DTC start request flag) is not set at the end of a data transfer up to detection of a retransmission start condition or stop condition after a slave address (SVA) or general call address match in I²C bus format slave mode.

Even when the IRIC flag and IRTR flag are set, the TDRE or RDRF flag may not be set. The IRIC and IRTR flags are not cleared at the end of the specified number of transfers in continuous transfer using the DTC. The TDRE or RDRF flag is cleared, however, since the specified number of ICDR reads or writes have been completed.

Table 14.4 shows the relationship between the flags and the transfer states.

Table 14.4 Flags and Transfer States

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	State
1/0	1/0	0	0	0	0	0	0	0	0	0	Idle state (flag clearing required)
1	1	0	0	0	0	0	0	0	0	0	Start condition issuance
1	1	1	0	0	1	0	0	0	0	0	Start condition established
1	1/0	1	0	0	0	0	0	0	0	0/1	Master mode wait
1	1/0	1	0	0	1	0	0	0	0	0/1	Master mode transmit/receive end
0	0	1	0	0	0	1/0	1	1/0	1/0	0	Arbitration lost
0	0	1	0	0	0	0	0	1	0	0	SAR match by first frame in slave mode
0	0	1	0	0	0	0	0	1	1	0	General call address match
0	0	1	0	0	0	1	0	0	0	0	SARX match
0	1/0	1	0	0	0	0	0	0	0	0/1	Slave mode transmit/receive end (except after SARX match)
0	1/0	1	0	0	1	1	0	0	0	0	Slave mode transmit/receive end (after SARX match)
0	1	1	0	0	0	1	0	0	0	1	
0	1/0	0	1/0	1/0	0	0	0	0	0	0/1	Stop condition detected

14.3.6 I²C Bus Status Register (ICSR)

ICSR consists of status flags.

Bit	Bit Name	Initial Value	R/W	Description
7	ESTP	0	R/(W)*	Error Stop Condition Detection Flag This bit is valid in I²C bus format slave mode. [Setting condition] When a stop condition is detected during frame transfer. [Clearing conditions] • When 0 is written in ESTP after reading ESTP = 1 • When the IRIC flag in ICCR is cleared to 0
6	STOP	0	R/(W)*	Normal Stop Condition Detection Flag This bit is valid in I²C bus format slave mode. [Setting condition] When a stop condition is detected after frame transfer completion. [Clearing conditions] • When 0 is written in STOP after reading STOP = 1 When the IRIC flag is cleared to 0
5	IRTR	0	R/(W)*	I²C Bus Interface Continuous Transfer Interrupt Request Flag Indicates that the I²C bus interface has issued an interrupt request to the CPU, and the source is completion of reception/transmission of one frame in continuous transmission/reception for which DTC activation is possible. When the IRTR flag is set to 1, the IRIC flag is also set to 1 at the same time. The IRTR flag is set while the TDRE or RDRF flag is set to 1. The IRTR flag is cleared by reading an existing 1 from and then writing a 0 to the flag. The IRTR flag is automatically cleared when the IRIC flag is cleared. [Setting conditions] • I²C bus format slave mode: When the TDRE or RDRF flag is set to 1 with AASX = 1. • Other than I²C bus format slave mode: When the TDRE or RDRF flag is set to 1 [Clearing conditions] • When 0 is written after reading IRTR = 1 • When the IRIC flag is cleared to 0 while ICE is 1

Bit	Bit Name	Initial Value	R/W	Description
4	AASX	0	R/(W)*	Second Slave Address Recognition Flag In I²C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVAX6 to SVAX0 in SARX. [Setting condition] • When the second slave address is detected in slave receive mode and FSX = 0 in SARX [Clearing conditions] • When 0 is written in AASX after reading AASX = 1 • When a start condition is detected • In master mode
3	AL	0	R/(W)*	Arbitration Lost Flag Indicates that arbitration was lost in master mode. [Setting conditions] • If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode • If the internal SCL line is high at the fall of SCL in master transmit mode [Clearing conditions] • When ICDR is written to (transmit mode) or read from (receive mode) • When 0 is written in AL after reading AL = 1

Bit	Bit Name	Initial Value	R/W	Description
2	AAS	0	R/(W)*	Slave Address Recognition Flag In I²C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR, or if the general call address (H'00) is detected. [Setting condition] • When the slave address or general call address (one frame including a R/$\overline{W}$ bit is H'00) is detected in slave receive mode and FS = 0 in SAR [Clearing conditions] • When ICDR is written to (transmit mode) or read from (receive mode) • When 0 is written in AAS after reading AAS = 1 • In master mode
1	ADZ	0	R/(W)*	General Call Address Recognition Flag In I²C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition is the general call address (H'00). To clear the ADZ flag, read a 1 from and then write a 0 to it. This flag is automatically reset when ICDR is written to (during transmission) or read from (during reception). [Setting condition] • When the general call address (one frame including a R/W bit is H'00) is detected in slave receive mode and FS = 0 [Clearing conditions] • When ICDR is written to (transmit mode) or read from (receive mode) • When 0 is written in ADZ after reading ADZ = 1· In master mode

Bit	Bit Name	Initial Value	R/W	Description
0	ACKB	0	R/W	Acknowledge Bit Stores acknowledge data. Transmit mode: [Setting condition] • When 1 is received as the acknowledge bit when ACKE=1 in transmit mode • [Clearing conditions] • When 0 is received as the acknowledge bit when ACKE=1 in transmit mode • When 0 is written to the ACKE bit Receive mode: 0: Returns 0 as acknowledge data after data reception 1: Returns 1 as acknowledge data after data reception When this bit is read, the value loaded from the bus line (returned by the receiving device) is read in transmission (when TRS = 1). In reception (when TRS = 0), the value set by internal software is read. When this bit is written, acknowledge data that is returned after receiving is rewritten regardless of the TRS value. The value loaded from the receiving device is retained. Therefore, care must be taken when rewriting the register using bit-manipulation instructions.

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

14.3.7 Serial Control Register X (SCRX)

SCRX controls register access, the I²C interface operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	These bits are reserved and always return 0 when read, and should only be written with 0.
6	IICX1	0	R/W	I²C transfer-rate select 1 and 0
5	IICX0	0	R/W	Along with bits CKS2 to CKS0 of ICMR, this bit selects the transfer rate in the master mode. For details on setting the transfer rate, refer to table 14.3
4	IICE	0	R/W	I²C master enable This bit controls access by the CPU to the data register and control registers (ICCR, ICSR, ICDR/SARX, and ICMR/SAR) of the I²C bus interface. 0: Disables CPU access to the data register and control registers of the I²C bus interface. 1: Enables CPU access to the data register and control registers of the I²C bus interface.
3	—	1	R	Reserved This bit is always read as 1.
2 to 0	—	All 0	R/W	Reserved This bit should not be written with 0

14.3.8 DDC Switch Register (DDCSWR)

DDCSWR controls IIC internal latch clearance.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4		All 0		Reserved. Only 0 can be written.
3	CLR3	1	W*2	IIC Clear 3 to 0
2	CLR2	1	W*2	Controls initialization of the internal state of IIC_0 and
1	CLR1	1	W*2	IIC_1.
0	CLR0	1	W*2	00--: Setting prohibited 0100: Setting prohibited 0101: IIC_0 internal latch cleared 0110: IIC_1 internal latch cleared 0111: IIC_0 and IIC_1 internal latches cleared 1---: Invalid setting When a write operation is performed on these bits, a clear signal is generated for the internal latch circuit of the corresponding module, and the internal state of the IIC module is initialized. These bits can only be written to; they are always read as 1. Write data to this bit is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR. When clearing is required again, all the bits must be written to in accordance with the setting. If the function of these bits is not used, set all of the CLR3 to CLR0 bits to 1 when writing to DDCSWR.

Notes: 1. Only 0 can be written, to clear the flag.
2. This bit is always read as 1.

14.4 Operation

14.4.1 I²C Bus Data Format

The I²C bus interface has serial and I²C bus formats.

The I²C bus formats are addressing formats with an acknowledge bit. These are shown in figures 15-3 (a) and (b). The first frame following a start condition always consists of 8 bits.

The serial format is a non-addressing format with no acknowledge bit. Although start and stop conditions must be issued, this format can be used as a synchronous serial format. This is shown in figure 14.4.

Figure 14.5 shows the I²C bus timing.

The symbols used in figures 15-3 to 15-5 are explained in table 14.5.

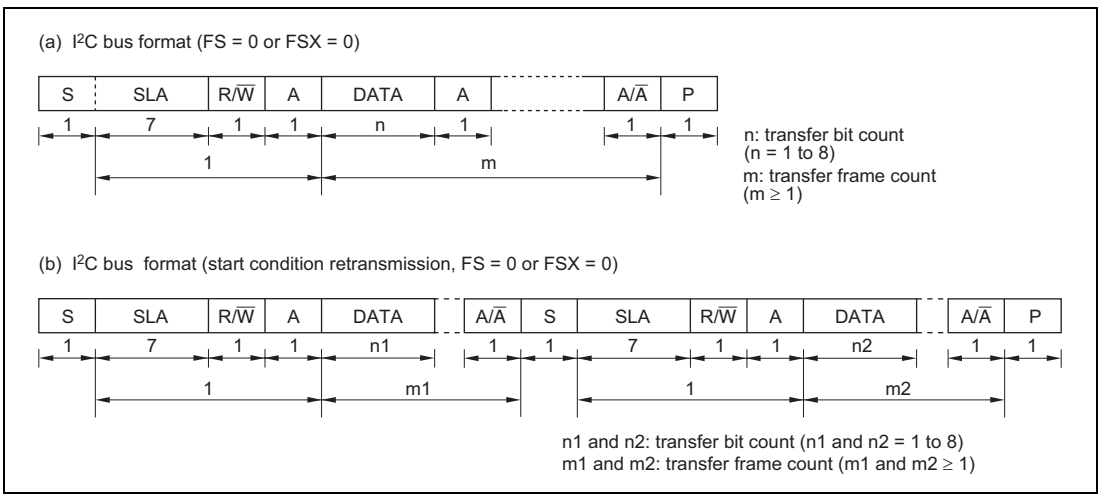


Figure 14.3 I²C Bus Data Formats (I²C Bus Formats)

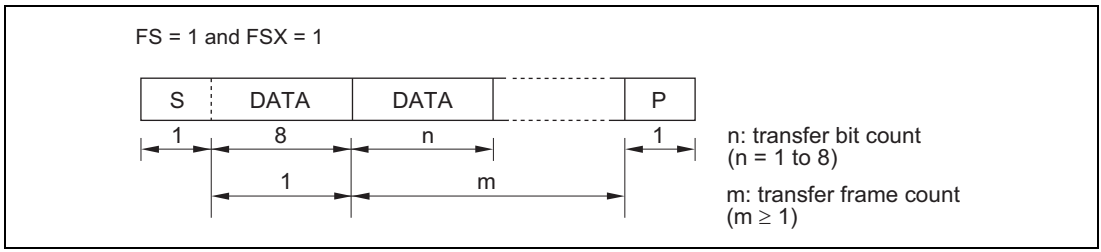


Figure 14.4 I²C Bus Data Format (Serial Format)

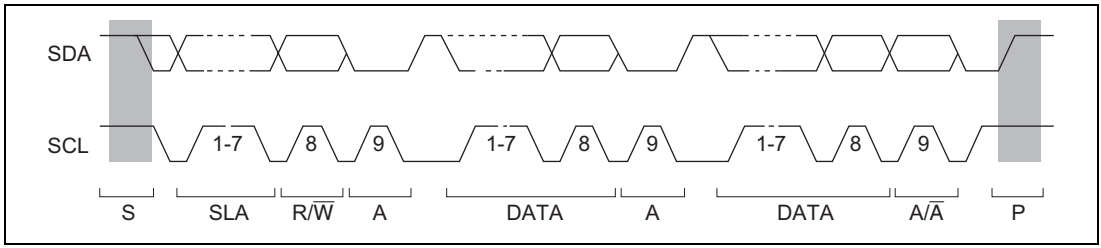


Figure 14.5 I²C Bus Timing

Table 14.5 I²C Bus Data Format Symbols

Legend	
S	Start condition. The master device drives SDA from high to low while SCL is high
SLA	Slave address, by which the master device selects a slave device
R/$\overline{W}$	Indicates the direction of data transfer: from the slave device to the master device when R/$\overline{W}$ is 1, or from the master device to the slave device when R/$\overline{W}$ is 0
A	Acknowledge. The receiving device (the slave in master transmit mode, or the master in master receive mode) drives SDA low to acknowledge a transfer
DATA	Transferred data. The bit length is set by bits BC2 to BC0 in ICMR. The MSB-first or LSB-first format is selected by bit MLS in ICMR
P	Stop condition. The master device drives SDA from low to high while SCL is high

14.4.2 Initial Setting

At startup the following procedure is used to initialize the IIC.

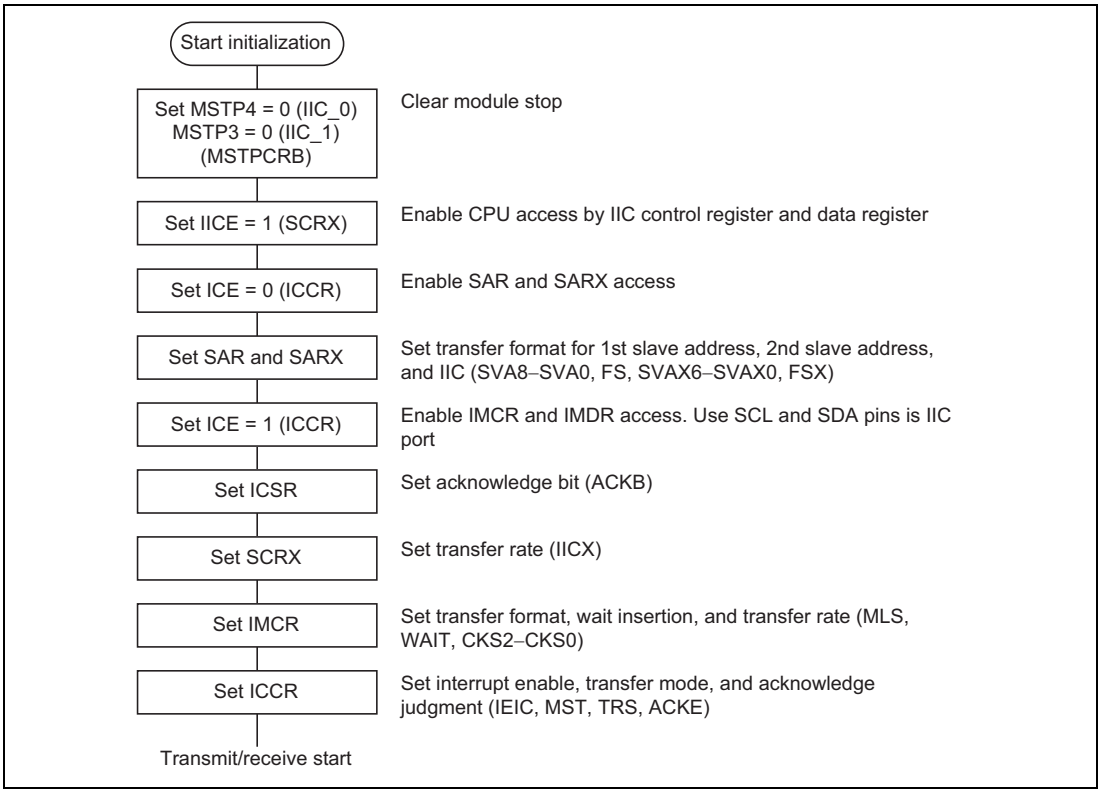


Figure 14.6 Flowchart for IIC Initialization (Example)

Note: The ICMR register should be written to only after transmit or receive operations have completed.
Writing to the ICMR register while a transmit or receive operation is in progress could cause an erroneous value to be written to bit counter bits BC2 to BC0. This could result in improper operation.

14.4.3 Master Transmit Operation

In I²C bus format master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal.

Figure 14.7 is a flowchart showing an example of the master transmit mode.

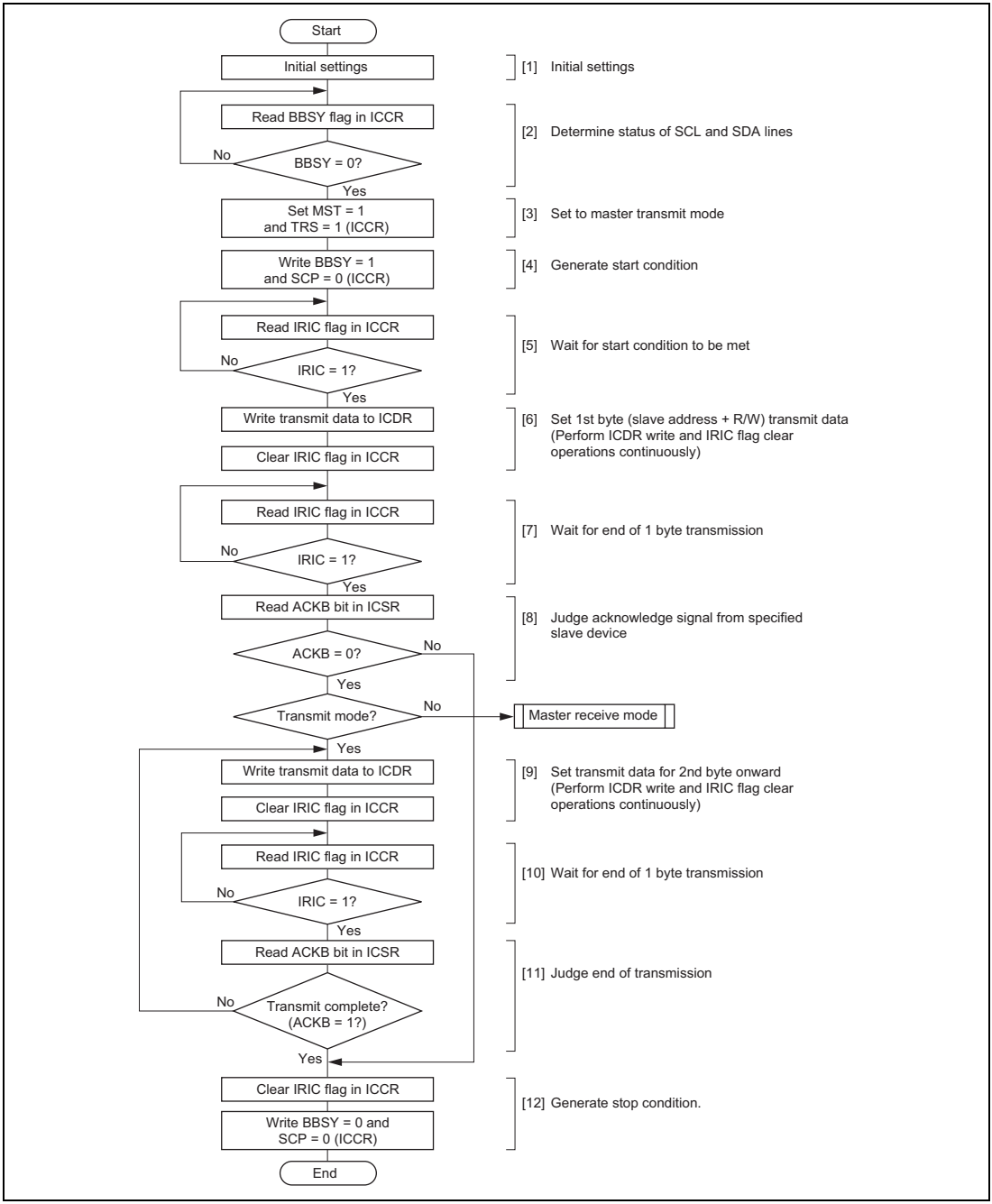


Figure 14.7 Flowchart for Master Transmit Mode (Example)

The procedure for transmitting data sequentially, synchronized with ICDR (ICDRT) write operations, is described below.

1. Perform initial settings as described in section 14.4.2, Initial Setting.
2. Read the BBSY flag in ICCR to confirm that the bus is free.
3. Set bits MST and TSR in ICCR to 1 to switch to the master transmit mode.
4. Write 1 to BBSY and 0 to SCP in ICCR. This changes SDA from high to low when SCL is high, and generates the start condition.
5. The IRIC and IRTR flags are set to 1 when the start condition is generated. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU.
6. After the start condition is detected, write the data (slave address + R/$\overline{W}$) to ICDR. With the I²C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the first frame data following the start condition indicates the 7-bit slave address and transmit/receive direction (R/W). Next, clear the IRIC flag to 0 to indicate the end of the transfer. Continue successively writing to ICDR and clearing the IRIC flag to ensure that processing of other interrupts does not intervene. If the time required to transmit one byte of data elapses by the time the IRIC flag is cleared, it will not be possible to determine the end of the transmission. The master device sequentially sends the transmit clock and the data written to ICDR. The selected slave device (i.e., the slave device with the matching slave address) drives SDA low at the 9th transmit clock pulse and returns an acknowledge signal.
7. When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of the 9th transmit clock pulse. After one frame has been transmitted, SCL is automatically fixed low in synchronization with the internal clock until the next transmit data is written.
8. Read the ACKB bit in ICSR to confirm that its value is 0. If the slave device has not returned an acknowledge signal and the value of ACKB is 1, perform the transmit end processing described in step.12 and then recommence the transmit operation from the beginning.
9. Write the transmit data to ICDR. Next, clear the IRIC flag to 0 to indicate the end of the transfer. Then continue successively writing to ICDR and clearing the IRIC flag as described in step.6. Transmission of the next frame is synchronized with the internal clock.
10. When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of the 9th transmit clock pulse. After one frame has been transmitted, SCL is automatically fixed low in synchronization with the internal clock until the next transmit data is written.
11. Read the ACKB bit in ICSR to confirm that the slave device has returned an acknowledge signal and the value of ACKB is 0. If the slave device has not returned an acknowledge signal and the value of ACKB is 1, perform the transmit end processing described in step.12.
12. Clear the IRIC flag to 0. Write 0 to the ACKE bit in ICCR and clear the received ACKB bit to 0.

Write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.

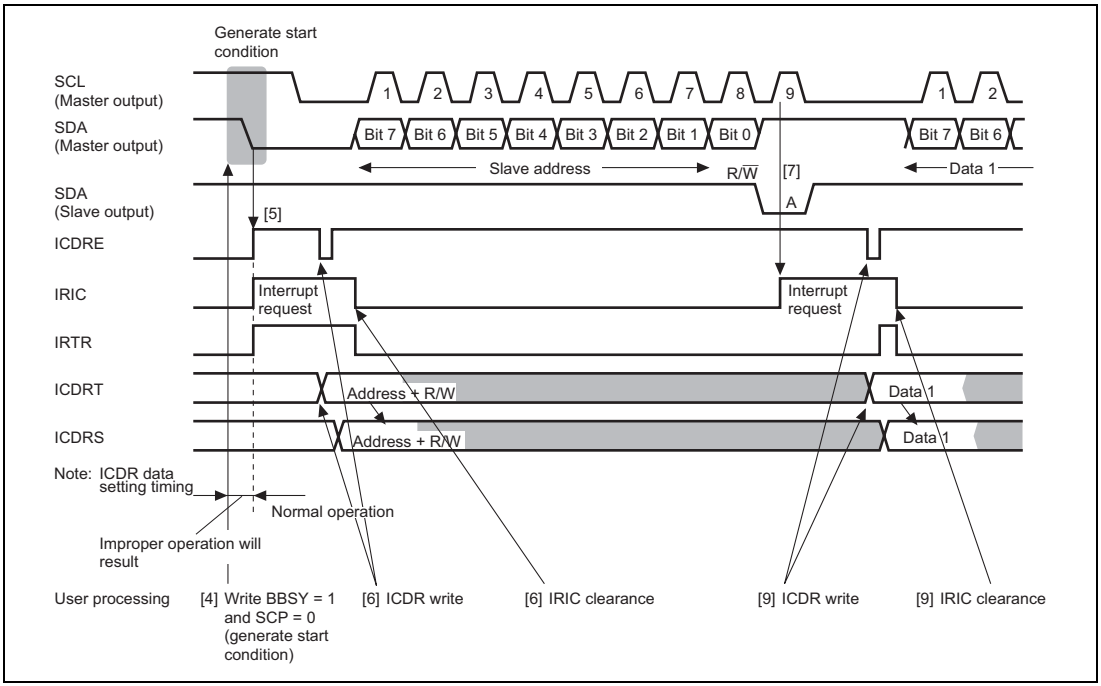


Figure 14.8 (1) Example of Master Transmit Mode Operation Timing (MLS = WAIT = 0)

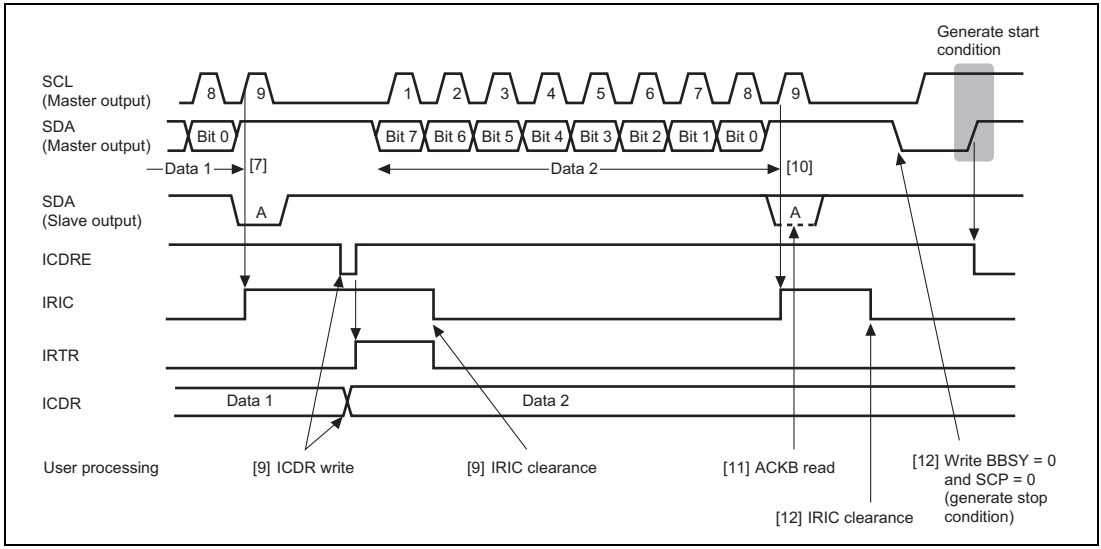


Figure 14.8 (2) Example of Master Transmit Mode Stop Condition Generation Timing (MLS = WAIT = 0)

14.4.4 Master Receive Operation

In I²C bus format master receive mode, the master device outputs the receive clock, receives data, and returns an acknowledge signal. The slave device transmits data.

The master device transmits the data containing the slave address + R/W (0: read) in the 1st frame after a start condition is generated in the master transmit mode. After the slave device is selected the switch to receive operation takes place.

(1) Receive Operation Using Wait States

Figures 14-9 (1) and 14-9 (2) are flowcharts showing examples of the master receive mode (WAIT = 1).

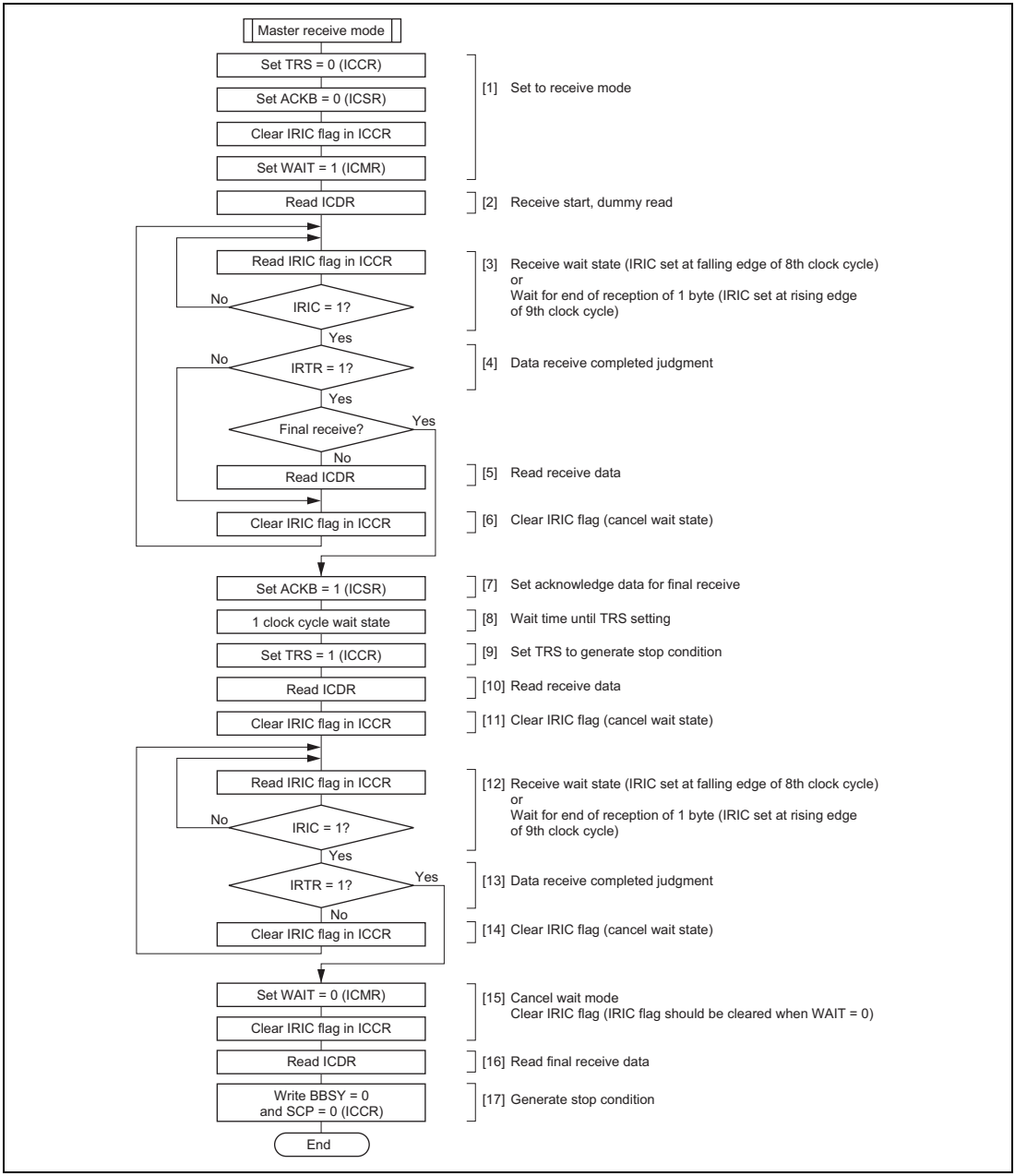


Figure 14.9 (1) Flowchart for Master Receive Mode (Receiving Multiple Bytes) (WAIT = 1) (Example)

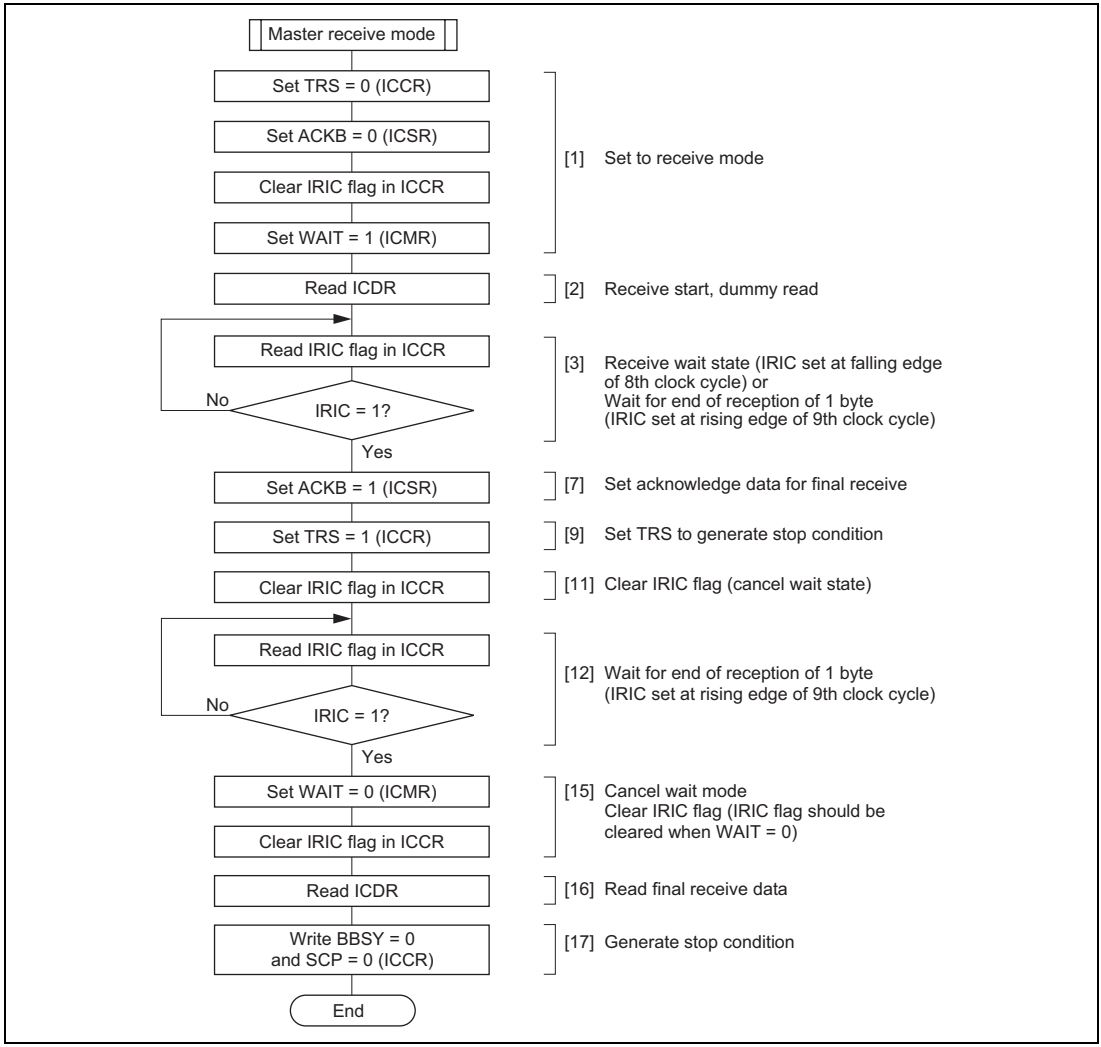


Figure 14.9 (2) Flowchart for Master Receive Mode (Receiving 1 Byte) (WAIT = 1) (Example)

The procedure for receiving data sequentially, using the wait states (WAIT bit) for synchronization with ICDR (ICDRR) read operations, is described below.

The procedure below describes the operation for receiving multiple bytes. Note that some of the steps are omitted when receiving only 1 byte. Refer to figure 14.9 (2) for details.

1. Clear the TRS bit in ICCR to 0 to switch from transmit mode to receive mode. Clear the ACKB bit in ICSR to 0 (acknowledge data setting). Clear the IRIC flag to 0, then set the WAIT bit in ICMR to 1.
2. When ICDR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock.
3. The IRIC flag is set to 1 by the following two conditions. At that point, an interrupt request is issued to the CPU if the IEIC bit in ICCR is set to 1.
   - The flag is set at the falling edge of the 8th clock cycle of the receive clock for 1 frame. SCL is automatically held low, in synchronization with the internal clock, until the IRIC flag is cleared.
   - The flag is set at the rising edge of the 9th clock cycle of the receive clock for 1 frame. The IRIC flag and ICDRF flag are set to 1, indicating that reception of 1 frame of data has ended. The master device continues to output the receive clock for the receive data.
4. Read the IRTR flag in ICSR. If the IRTR flag value is 0, the wait state is cancelled by clearing the IRIC flag as described in step.6 below. If the IRTR flag value is 1 and the next receive data is the final receive data, perform the end processing described in step.7 below.
5. If the IRTR flag value is 1, read the ICDR receive data.
6. Clear the IRTR flag to 0. If condition described in step.3-1 is true, the master device drives SDA to low level and returns an acknowledge signal when the receive clock outputs the 9th clock cycle.

   Further data can be received by repeating steps.3 through 6.
7. Set the ACKB bit in ICSR to 1 to set the acknowledge data for the final receive.
8. Wait for at least 1 clock cycle after the IRIC flag is set to 1 and then wait for the rising edge of the 1st clock cycle of the next receive data.
9. Set the TSR bit in ICCR to 1 to switch from the receive mode to the transmit mode. The TSR bit setting value at this point becomes valid when the rising edge of the next 9th clock cycle is input.
10. Read the ICDR receive data.
11. Clear the IRTR flag to 0.
12. The IRIC flag is set to 1 by the following two conditions.
    - The flag is set at the falling edge of the 8th clock cycle of the receive clock for 1 frame. SCL is automatically held low, in synchronization with the internal clock, until the IRIC flag is cleared.
    - The flag is set at the rising edge of the 9th clock cycle of the receive clock for 1 frame. The IRIC flag and ICDRF flag are set to 1, indicating that reception of 1 frame of data has ended. The master device continues to output the receive clock for the receive data.

13. Read the IRTR flag in ICSR. If the IRTR flag value is 0, the wait state is cancelled by clearing the IRIC flag as described in step.14 below. If the IRTR flag value is 1 and the receive operation has finished, perform the issue stop condition processing described in step.15 below.
14. If the IRTR flag value is 0, clear the IRIC flag to 0 to cancel the wait state. Return to reading the IRIC flag, as described in step.12, to detect the end of the receive operation.
15. Clear the WAIT bit in ICMR to 0 to cancel the wait mode. Then clear the IRIC flag to 0. The IRIC flag should be cleared when the value of WAIT is 0 (The stop condition may not be output properly when the issue stop condition instruction is executed if the WAIT bit was cleared to 0 after the IRIC flag is cleared to 0).
16. Read the final receive data in ICDR.
17. Write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.

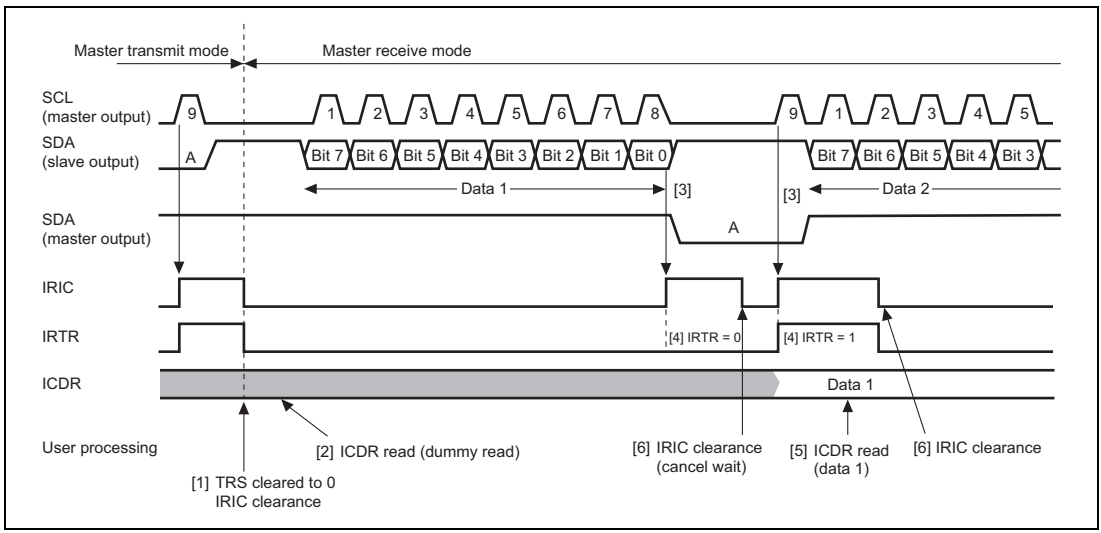


Figure 14.10 (1) Example of Master Receive Mode Operation Timing (MLS = ACKB = 0, WAIT = 1)

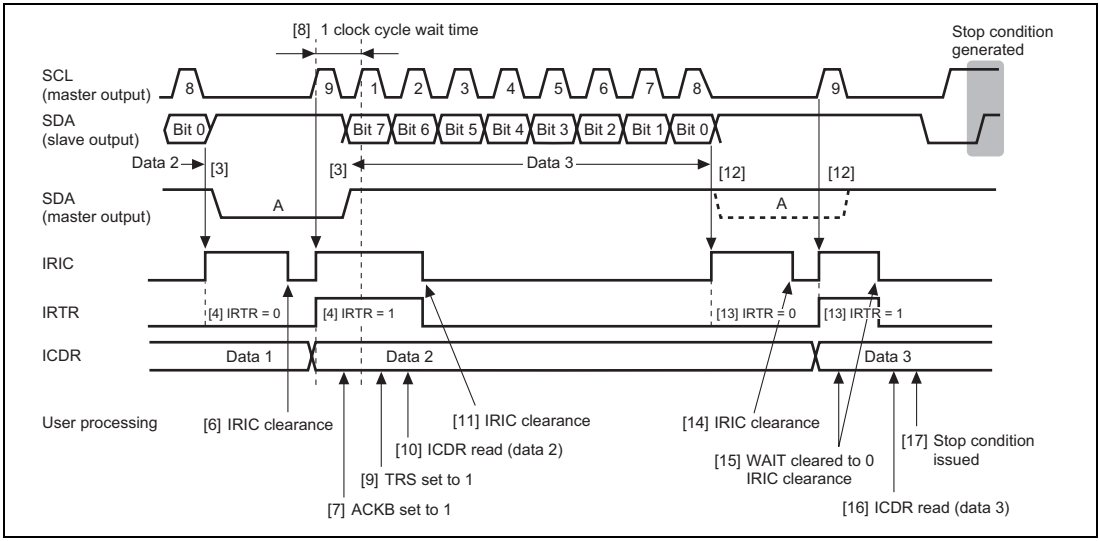


Figure 14.10 (2) Example of Master Receive Mode Stop Condition Generation Timing (MLS = ACKB = 0, WAIT = 1)

14.4.5 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 slave device compares its own address with the slave address in the first frame following the establishment of the start condition issued by the master device. If the addresses match, the slave device operates as the slave device designated by the master device.

Figure 14.14 is a flowchart showing an example of slave receive mode operation.

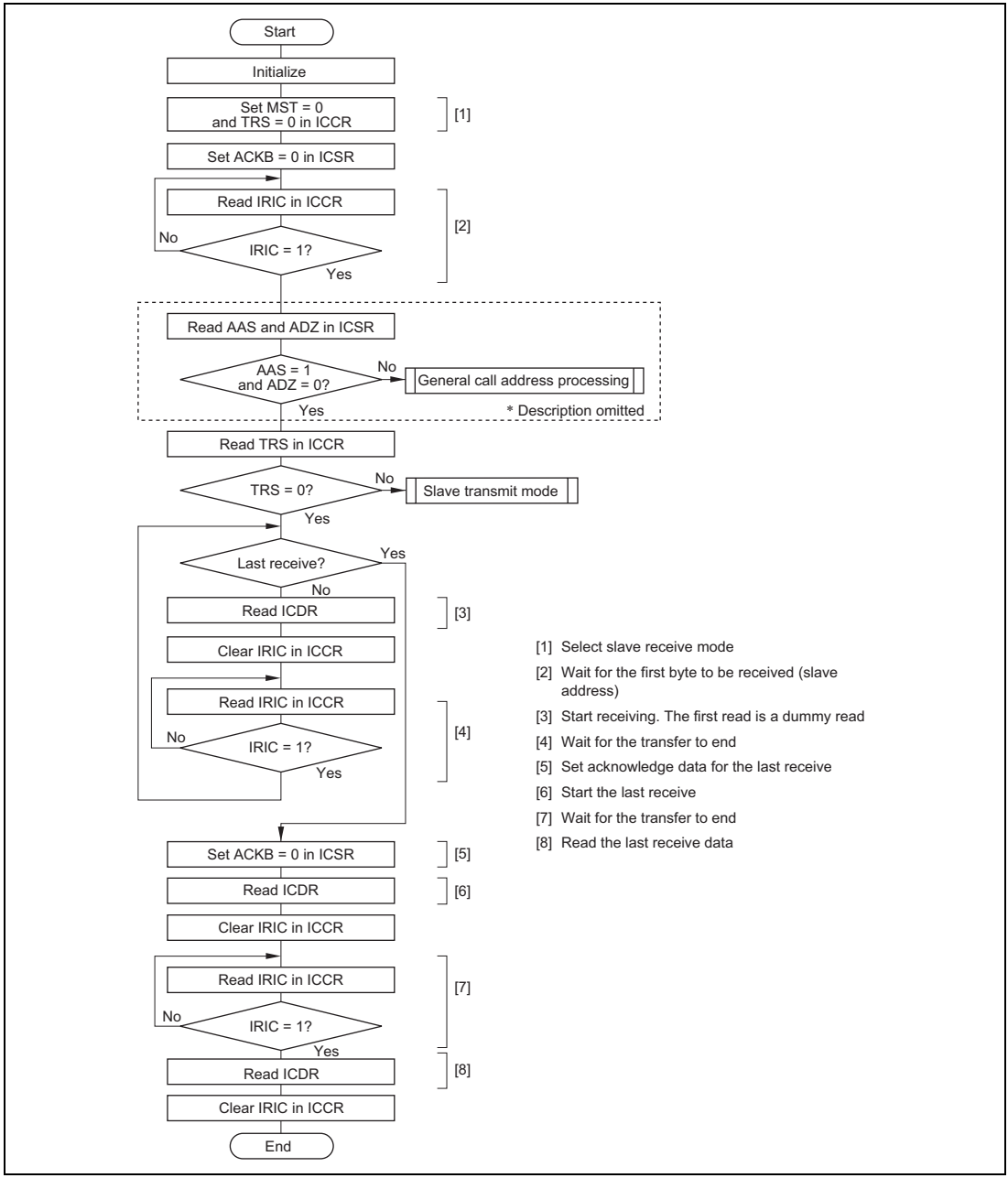


Figure 14.11 Flowcharts for Slave Transmit Mode (Example)

The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.
2. When the start condition output by the master device is detected, the BBSY flag in ICCR is set to 1.
3. When the slave address matches in the first frame following the start condition, the device operates as the slave device specified by the master device. If the 8th data bit (R/$\overline{W}$) is 0, the TRS bit in ICCR remains cleared to 0, and slave receive operation is performed.
4. At the 9th clock pulse of the receive frame, the slave device drives SDA low and returns an acknowledge signal. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. If the RDRF internal flag has been cleared to 0, it is set to 1, and the receive operation continues. If the RDRF internal flag has been set to 1, the slave device drives SCL low from the fall of the receive clock until data is read into ICDR.
5. Read ICDR and clear the IRIC flag in ICCR to 0. The RDRF flag is cleared to 0.

Receive operations can be performed continuously by repeating steps 4 and 5. When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

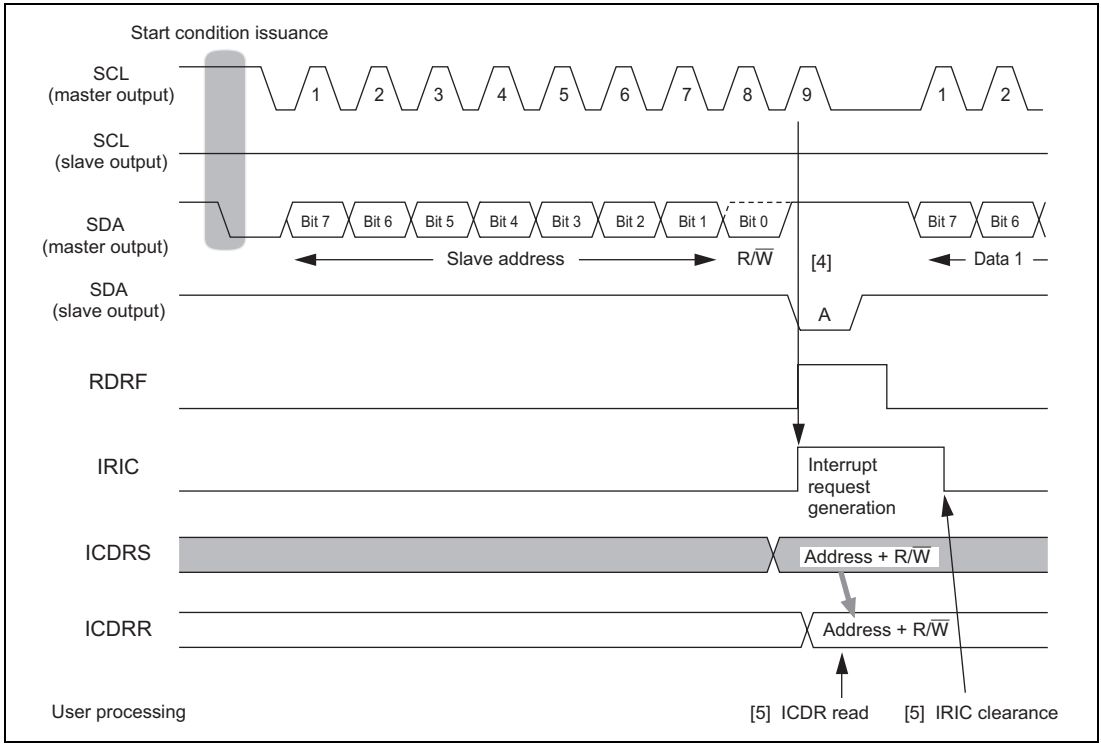


Figure 14.12 Example of Slave Receive Mode Operation Timing (1)
(MLS = ACKB = 0)

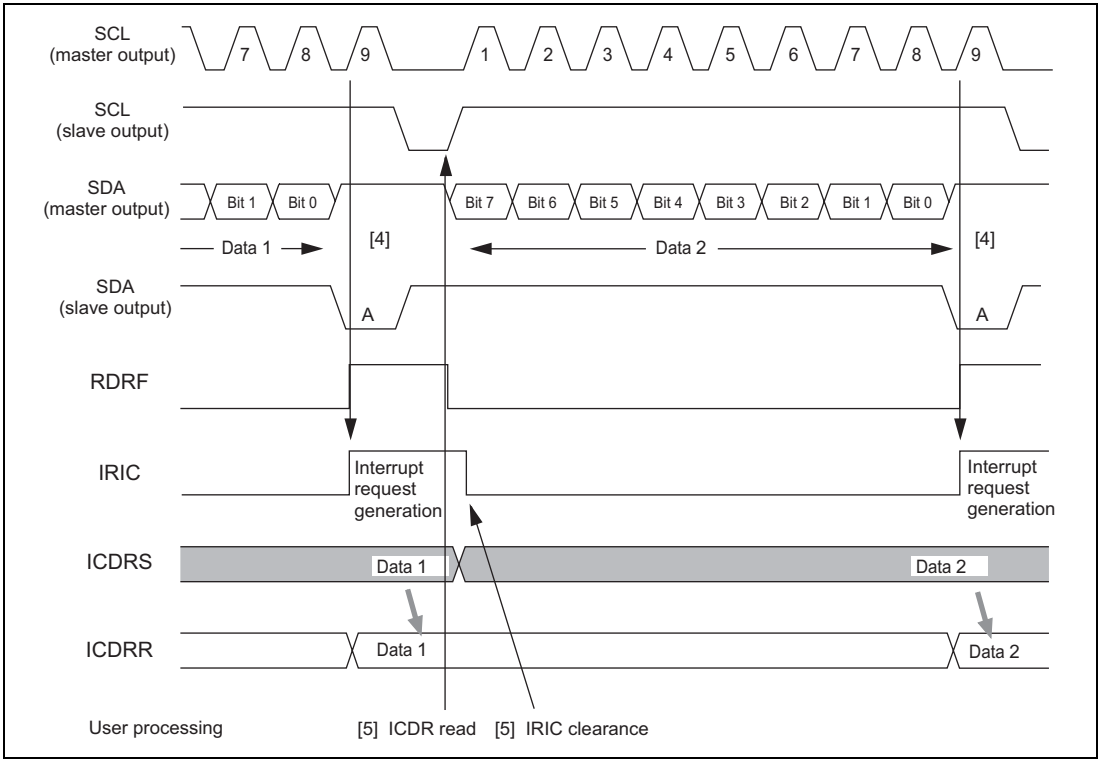


Figure 14.13 Example of Slave Receive Mode Operation Timing (2)
(MLS = ACKB = 0)

14.4.6 Slave Transmit Operation

In slave transmit operation, the slave device compares its own address with the slave address transmitted by the master device in the first frame (address receive frame) following detection of the start condition. If the addresses match and the 8th bit (R/W) is set to 1 (read), the TRS bit in ICCR is automatically set to 1 and slave transmit mode is activated.

Figure 14.17 is a flowchart showing an example of slave transmit mode operation.

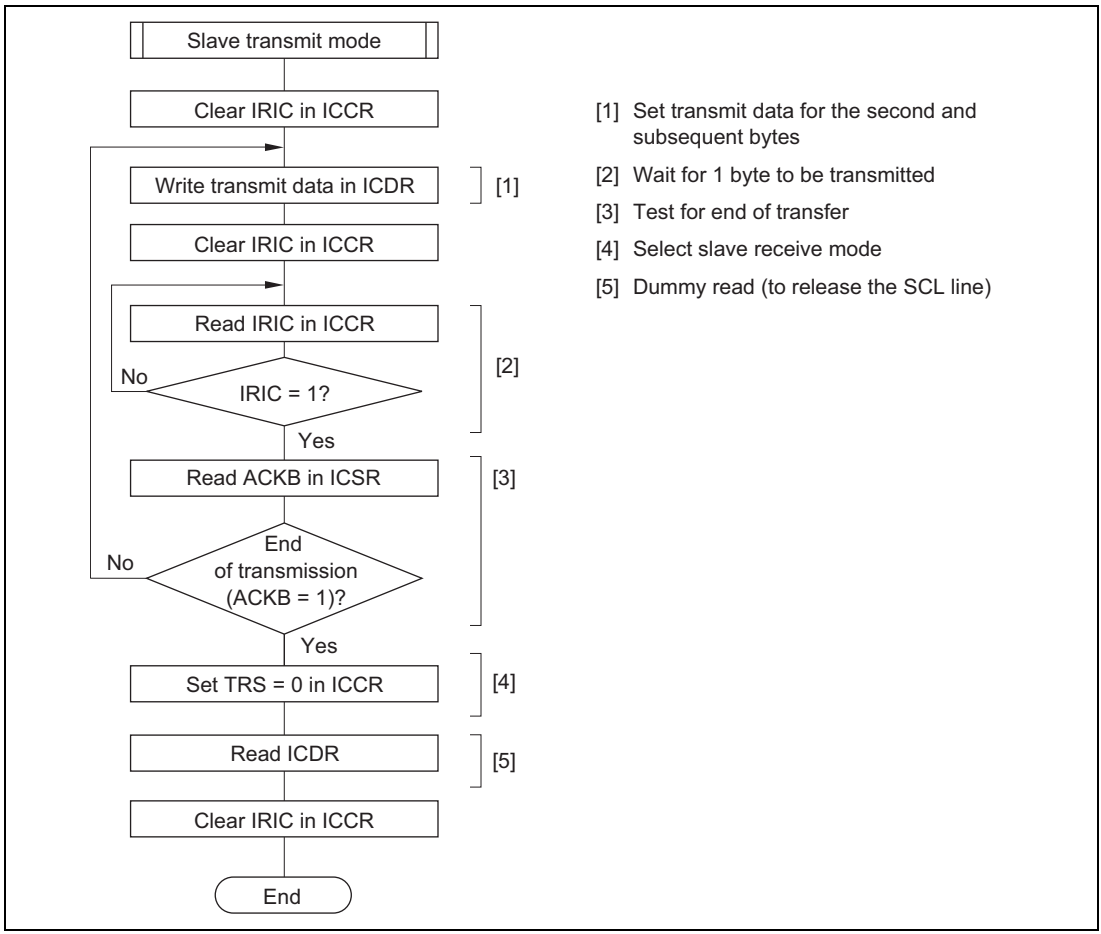


Figure 14.14 Flowcharts for Slave Receive Mode (Example)

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.
2. When the slave address matches in the first frame following detection of the start condition, the slave device drives SDA low at the 9th clock pulse and returns an acknowledge signal. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. If the 8th data bit (R/$\overline{W}$) is 1, the TRS bit in ICCR is set to

1, and the mode changes to slave transmit mode automatically. The TDRF flag is set to 1. The slave device drives SCL low from the fall of the transmit clock until ICDR data is written.

3. After clearing the IRIC flag to 0, write data to ICDR. The TDRE internal flag is cleared to 0. The written data is transferred to ICDRS, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again. After clearing the IRIC flag to 0, write the next data to ICDR. The slave device sequentially sends the data written into ICDR in accordance with the clock output by the master device at the timing shown in figure 14.18.
4. When one frame of data has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit clock pulse. If the TDRE internal flag has been set to 1, this slave device drives SCL low from the fall of the transmit clock until data is written to ICDR. The master device drives SDA low at the 9th clock pulse, and returns an acknowledge signal. As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used to determine whether the transfer operation was performed normally. When the TDRE internal flag is 0, the data written into ICDR is transferred to ICDRS, transmission is started, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again.
5. To continue transmission, clear the IRIC flag to 0, then write the next data to be transmitted into ICDR. The TDRE flag is cleared to 0.

Transmit operations can be performed continuously by repeating steps 4 and 5. To end transmission, write H'FF to ICDR to release SDA on the slave side. When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

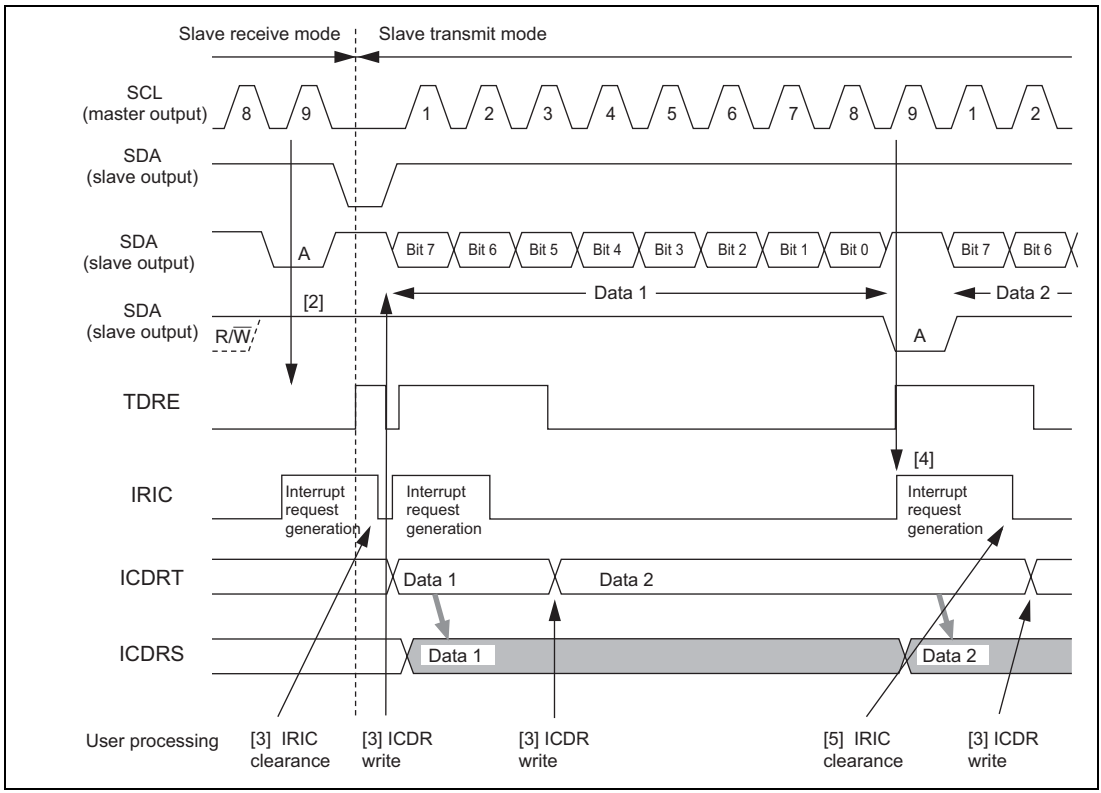


Figure 14.15 Example of Slave Transmit Mode Operation Timing (MLS = 0)

14.4.7 IRIC Setting Timing and SCL Control

The interrupt request flag (IRIC) is set at different times depending on the WAIT bit in ICMR, the FS bit in SAR, and the FSX bit in SARX. If the TDRE or RDRF internal flag is set to 1, SCL is automatically held low after one frame has been transferred; this timing is synchronized with the internal clock. Figure 14.19 shows the IRIC set timing and SCL control.

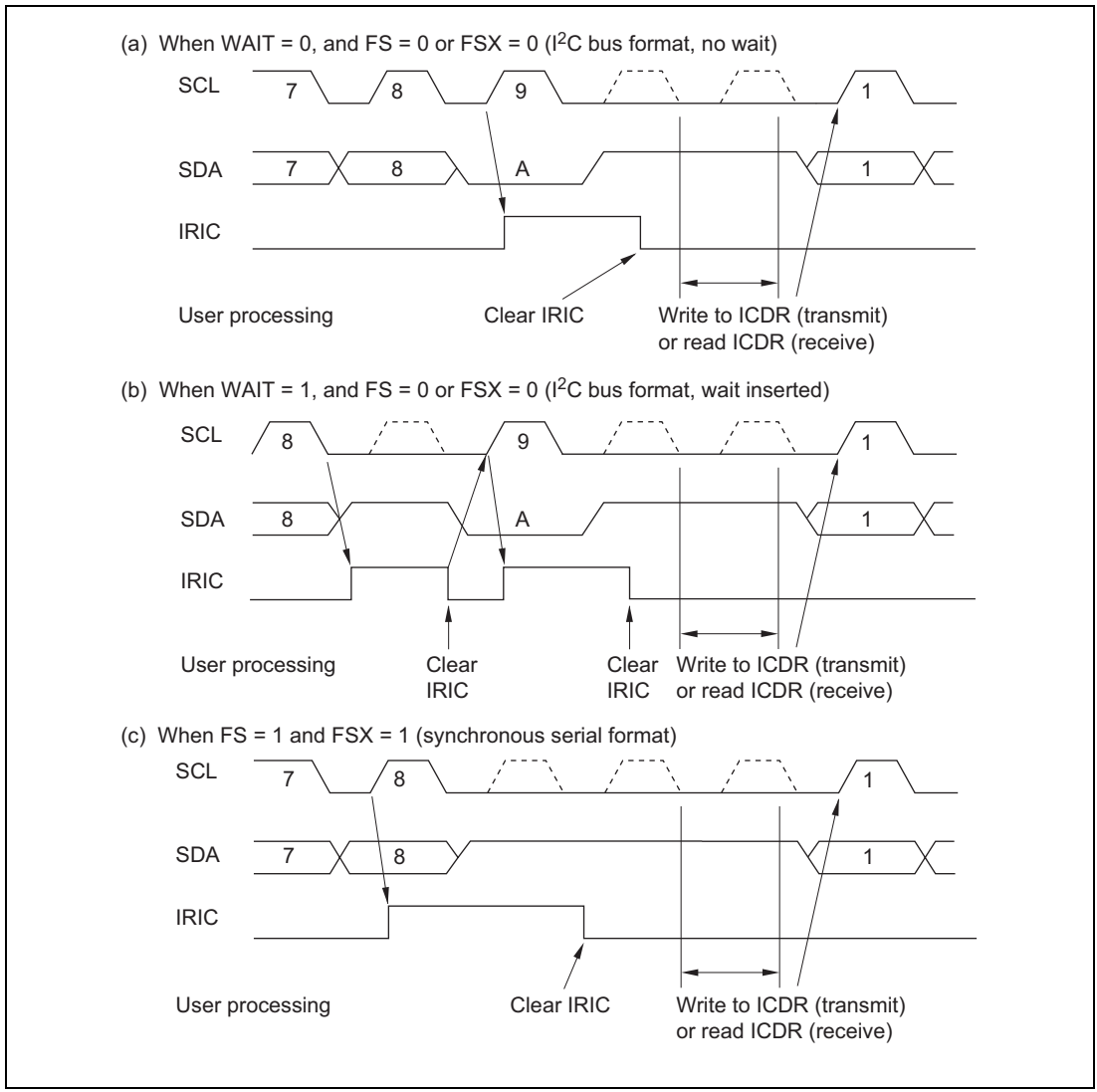


Figure 14.16 IRIC Setting Timing and SCL Control

14.4.8 Operation Using the DTC

The I²C bus format provides for selection of the slave device and transfer direction by means of the slave address and the R/$\overline{W}$ bit, confirmation of reception with the acknowledge bit, indication of the last frame, and so on. Therefore, continuous data transfer using the DTC must be carried out in conjunction with CPU processing by means of interrupts.

Table 14.5 shows some examples of processing using the DTC. These examples assume that the number of transfer data bytes is known in slave mode.

Table 14.6 Examples of Operation Using the DTC

Item	Master Transmit Mode	Master Receive Mode	Slave Transmit Mode	Slave Receive Mode
Slave address + R/$\overline{W}$ bit transmission/ reception	Transmission by DTC (ICDR write)	Transmission by CPU (ICDR write)	Reception by CPU (ICDR read)	Reception by CPU (ICDR read)
Dummy data read	—	Processing by CPU (ICDR read)	—	—
Actual data transmission/ reception	Transmission by DTC (ICDR write)	Reception by DTC (ICDR read)	Transmission by DTC (ICDR write)	Reception by DTC (ICDR read)
Dummy data (H'FF) write	—	—	Processing by DTC (ICDR write)	—
Last frame processing	Not necessary	Reception by CPU (ICDR read)	Not necessary	Reception by CPU (ICDR read)
Transfer request processing after last frame processing	1st time: Clearing by CPU 2nd time: End condition issuance by CPU	Not necessary	Automatic clearing on detection of end condition during transmission of dummy data (H'FF)	Not necessary
Setting of number of DTC transfer data frames	Transmission: Actual data count + 1 (+1 equivalent to slave address + R/$\overline{W}$ bits)	Reception: Actual data count	Transmission: Actual data count + 1 (+1 equivalent to dummy data (H'FF))	Reception: Actual data count

14.4.9 Noise Canceler

The logic levels at the SCL and SDA pins are routed through noise cancelers before being latched internally. Figure 14.20 shows a block diagram of the noise canceler circuit.

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 agree. If they do not agree, the previous value is held.

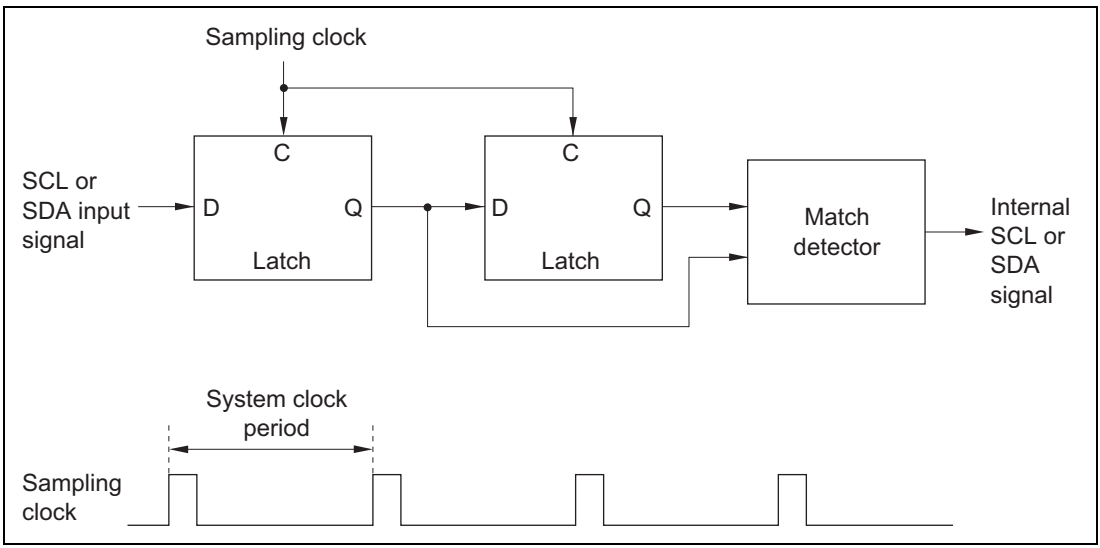


Figure 14.17 Block Diagram of Noise Canceler

14.4.10 Initialization of Internal State

The IIC has a function for forcible initialization of its internal state if a deadlock occurs during communication.

Initialization is executed by (1) setting bits CLR3 to CLR0 in the DDCSWR register or (2) clearing the ICE bit. For details of settings for bits CLR3 to CLR0, see section 14.3.8, DDC Switch Register (DDCSWR).

Scope of Initialization: The initialization executed by this function covers the following items:

- TDRE and RDRF internal flags
- Transmit/receive sequencer and internal operating clock counter

- Internal latches for retaining the output state of the SCL and SDA pins (wait, clock, data output, etc.)

The following items are not initialized:

- Actual register values (ICDR, SAR, SARX, ICMR, ICCR, ICSR, DDCSWR, and STCR)
- Internal latches used to retain register read information for setting/clearing flags in the ICMR, ICCR, ICSR, and DDCSWR registers
- The value of the ICMR register bit counter (BC2 to BC0)
- Generated interrupt sources (interrupt sources transferred to the interrupt controller)

Notes on Initialization:

- Interrupt flags and interrupt sources are not cleared, and so flag clearing measures must be taken as necessary.
- Basically, other register flags are not cleared either, and so flag clearing measures must be taken as necessary.
- When initialization is performed by means of the DDCSWR register, the write data for bits CLR3 to CLR0 is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR. Similarly, when clearing is required again, all the bits must be written to simultaneously in accordance with the setting.
- If a flag clearing setting is made during transmission/reception, the IIC module will stop transmitting/receiving at that point and the SCL and SDA pins will be released. When transmission/reception is started again, register initialization, etc., must be carried out as necessary to enable correct communication as a system.

The value of the BBSY bit cannot be modified directly by this module clear function, but since the stop condition pin waveform is generated according to the state and release timing of the SCL and SDA pins, the BBSY bit may be cleared as a result. Similarly, state switching of other bits and flags may also have an effect.

To prevent problems caused by these factors, the following procedure should be used when initializing the IIC state.

1. Execute initialization of the internal state according to the setting of bits CLR3 to CLR0, or according to the ICE bit.
2. Execute a stop condition issuance instruction (write 0 to BBSY and SCP) to clear the BBST bit to 0, and wait for two transfer rate clock cycles.

3. Re-execute initialization of the internal state according to the setting of bits CLR3 to CLR0, or according to the ICE bit.
4. Initialize (re-set) the IIC registers.

14.5 Usage Notes

1. In master mode, if an instruction to generate a start condition is immediately followed by an instruction to generate a stop condition, neither condition will be output correctly. To output consecutive start and stop conditions, after issuing the instruction that generates the start condition, read the relevant ports, check that SCL and SDA are both low, then issue the instruction that generates the stop condition. Note that SCL may not yet have gone low when BBSY is cleared to 0.
2. Either of the following two conditions will start the next transfer. Pay attention to these conditions when reading or writing to ICDR.
   — Write access to ICDR when ICE = 1 and TRS = 1 (including automatic transfer from ICDRT to ICDRS)
   — Read access to ICDR when ICE = 1 and TRS = 0 (including automatic transfer from ICDRS to ICDRR)
3. Table 14.7 shows the timing of SCL and SDA output in synchronization with the internal clock. Timings on the bus are determined by the rise and fall times of signals affected by the bus load capacitance, series resistance, and parallel resistance.

Table 14.7 I²C Bus Timing (SCL and SDA Output)

Item	Symbol	Output Timing	Unit	Notes
SCL output cycle time	t_{SCLO}	28 t_{cyc} to 256 t_{cyc}	ns	Figure 23-20 (reference)
SCL output high pulse width	t_{SCLHO}	0.5 t_{SCLO}	ns	
SCL output low pulse width	t_{SCLLO}	0.5 t_{SCLO}	ns	
SDA output bus free time	t_{BUFO}	0.5 t_{SCLO} – 1 t_{cyc}	ns	
Start condition output hold time	t_{STAHO}	0.5 t_{SCLO} – 1 t_{cyc}	ns	
Retransmission start condition output setup time	t_{STASO}	1 t_{SCLO}	ns	
Stop condition output setup time	t_{STOSO}	0.5 t_{SCLO} + 2 t_{cyc}	ns	
Data output setup time (master)	t_{SDASO}	1 t_{SCLLO} – 3 t_{cyc}	ns	
Data output setup time (slave)		1 t_{SCLL} – 3 t_{cyc}		
Data output hold time	t_{SDAHO}	3 t_{cyc}	ns	

4. SCL and SDA input is sampled in synchronization with the internal clock. The AC timing therefore depends on the system clock cycle t_{cyc}, as shown in table 23-9 in section 23, Electrical Characteristics. Note that the I²C bus interface AC timing specifications will not be met with a system clock frequency of less than 5 MHz.
5. The I²C bus interface specification for the SCL rise time t_{sr} is under 1000 ns (300 ns for high-speed mode). In master mode, the I²C bus interface monitors the SCL line and synchronizes one bit at a time during communication. If t_{sr} (the time for SCL to go from low to V_{IH}) exceeds the time determined by the input clock of the I²C bus interface, the high period of SCL is extended. The SCL rise time is determined by the pull-up resistance and load capacitance of the SCL line. To insure proper operation at the set transfer rate, adjust the pull-up resistance and load capacitance so that the SCL rise time does not exceed the values given in the table 14.8.

Table 14.8 Permissible SCL Rise Time (t_{Sr}) Values

			Time Indication					
IICX	t_{cyc} Indication		I²C Bus Specification (Max.)	φ = 5 MHz	φ = 8 MHz	φ = 10 MHz	φ = 16 MHz	φ = 20 MHz
0	7.5 t_{cyc}	Standard mode	1000 ns	1000 ns	937 ns	750 ns	468 ns	375 ns
		High-speed mode	300 ns	300 ns	300 ns	300 ns	300 ns	300 ns
1	17.5 t_{cyc}	Standard mode	1000 ns	1000 ns	1000 ns	1000 ns	1000 ns	875 ns
		High-speed mode	300 ns	300 ns	300 ns	300 ns	300 ns	300 ns

6. The I²C bus interface specifications for the SCL and SDA rise and fall times are under 1000 ns and 300 ns. The I²C bus interface SCL and SDA output timing is prescribed by t_{Scyc} and t_{cyc}, as shown in table 14.7. However, because of the rise and fall times, the I²C bus interface specifications may not be satisfied at the maximum transfer rate. Table 14.9 shows output timing calculations for different operating frequencies, including the worst-case influence of rise and fall times.

   t_{BUFO} fails to meet the I²C bus interface specifications at any frequency. The solution is either (a) to provide coding to secure the necessary interval (approximately 1 µs) between issuance of a stop condition and issuance of a start condition, or (b) to select devices whose input timing permits this output timing for use as slave devices connected to the I²C bus.

   t_{SCLLO} in high-speed mode and t_{STASO} in standard mode fail to satisfy the I²C bus interface specifications for worst-case calculations of t_{Sr}/t_{Sf}. Possible solutions that should be investigated include (a) adjusting the rise and fall times by means of a pull-up resistor and capacitive load, (b) reducing the transfer rate to meet the specifications, or (c) selecting devices whose input timing permits this output timing for use as slave devices connected to the I²C bus.

Table 14.9 I²C Bus Timing (with Maximum Influence of t_{Sr}/t_{Sf})

Item	t_{cyc} Indication		Time Indication (at Maximum Transfer Rate) [ns]						
			t_{Sr}/t_{Sf} Influence (Max.)	I²C Bus Specification (Min.)	ϕ = 5 MHz	ϕ = 8 MHz	ϕ = 10 MHz	ϕ = 16 MHz	ϕ = 20 MHz
t_{SCLHO}	0.5 t_{SCLO} ($-t_{Sr}$)	Standard mode	–1000	4000	4000	4000	4000	4000	4000
		High-speed mode	–300	600	950	950	950	950	950
t_{SCLLO}	0.5 t_{SCLO} ($-t_{Sf}$)	Standard mode	–250	4700	4750	4750	4750	4750	4750
		High-speed mode	–250	1300	1000*1	1000*1	1000*1	1000*1	1000*1
t_{BUFO}	0.5 t_{SCLO} – 1 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	4700	3800*1	3875*1	3900*1	3938*1	3950*1
		High-speed mode	–300	1300	750*1	825*1	850*1	888*1	900*1
t_{STAHO}	0.5 t_{SCLO} – 1 t_{cyc} ($-t_{Sf}$)	Standard mode	–250	4000	4550	4625	4650	4688	4700
		High-speed mode	–250	600	800	875	900	938	950
t_{STASO}	1 t_{SCLO} ($-t_{Sr}$)	Standard mode	–1000	4700	9000	9000	9000	9000	9000
		High-speed mode	–300	600	2200	2200	2200	2200	2200
t_{STOSO}	0.5 t_{SCLO} + 2 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	4000	4400	4250	4200	4125	4100
		High-speed mode	–300	600	1350	1200	1150	1075	1050
t_{SDASO} (master)	1 t_{SCLLO}*2 –3 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	250	3100	3325	3400	3513	3550
		High-speed mode	–300	100	400	625	700	813	850
t_{SDASO} (slave)	1 t_{SCLL}*2 – 3 t_{cyc}*2 ($-t_{Sr}$)	Standard mode	–1000	250	3100	3325	3400	3513	3550
		High-speed mode	–300	100	400	625	700	813	850

Item	t_{cyc} Indication		Time Indication (at Maximum Transfer Rate) [ns] t_{sr}/t_{sf} Influence (Max.)	I²C Bus Specification (Min.)	ϕ = 5 MHz	ϕ = 8 MHz	ϕ = 10 MHz	ϕ = 16 MHz	ϕ = 20 MHz
t_{SDAHO}	3 t_{cyc}	Standard mode	0	0	600	375	300	188	150
		High-speed mode	0	0	600	375	300	188	150

Notes: 1. Does not meet the I²C bus interface specification. Remedial action such as the following is necessary: (a) secure a start/stop condition issuance interval; (b) adjust the rise and fall times by means of a pull-up resistor and capacitive load; (c) reduce the transfer rate; (d) select slave devices whose input timing permits this output timing.

The values in the above table will vary depending on the settings of the IICX bit and bits CKS0 to CKS2. Depending on the frequency it may not be possible to achieve the maximum transfer rate; therefore, whether or not the I²C bus interface specifications are met must be determined in accordance with the actual setting conditions.

2. Calculated using the I²C bus specification values (standard mode: 4700 ns min.; high-speed mode: 1300 ns min.).

7. Note on ICDR Read at End of Master Reception

To halt reception at the end of a receive operation in master receive mode, set the TRS bit to 1 and write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition. After this, receive data can be read by means of an ICDR read, but if data remains in the buffer the ICDRS receive data will not be transferred to ICDR, and so it will not be possible to read the second byte of data.

If it is necessary to read the second byte of data, issue the stop condition in master receive mode (i.e. with the TRS bit cleared to 0). When reading the receive data, first confirm that the BBSY bit in the ICCR register is cleared to 0, the stop condition has been generated, and the bus has been released, then read the ICDR register with TRS cleared to 0.

Note that if the receive data (ICDR data) is read in the interval between execution of the instruction for issuance of the stop condition (writing of 0 to BBSY and SCP in ICCR) and the actual generation of the stop condition, the clock may not be output correctly in subsequent master transmission.

Clearing of the MST bit after completion of master transmission/reception, or other modifications of IIC control bits to change the transmit/receive operating mode or settings, must be carried out during interval (a) in figure 14.18 (after confirming that the BBSY bit has been cleared to 0 in the ICCR register).

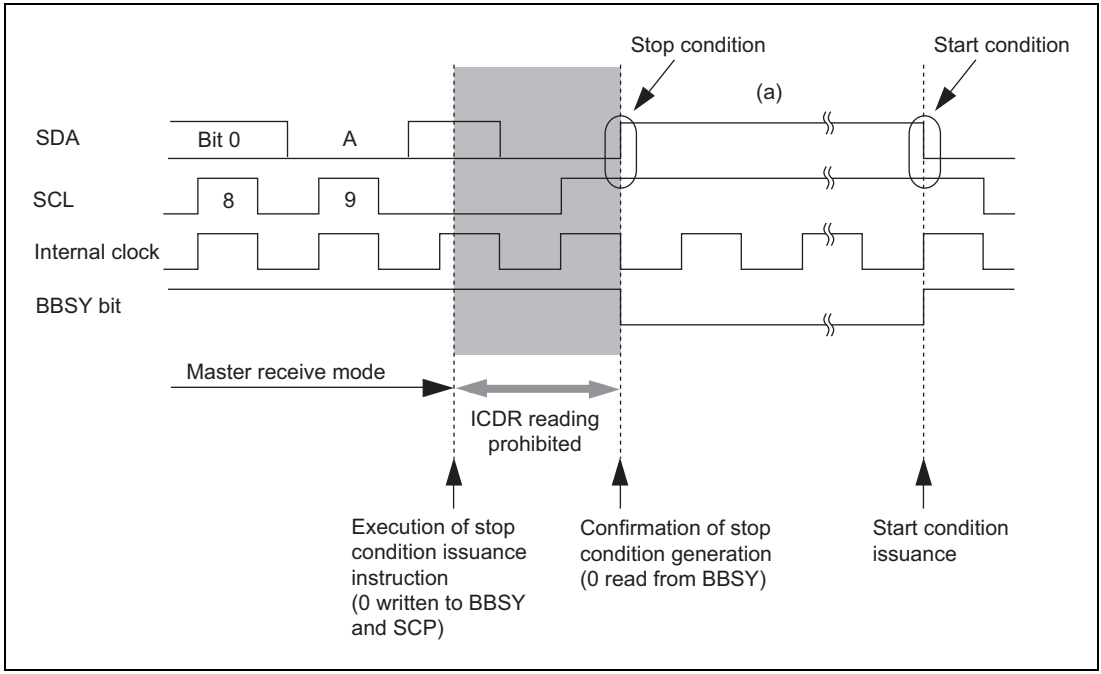


Figure 14.18 Points for Attention Concerning Reading of Master Receive Data

8. Notes on Start Condition Issuance for Retransmission

   Figure 14.19 shows the timing of start condition issuance for retransmission, and the timing for subsequently writing data to ICDR, together with the corresponding flowchart.

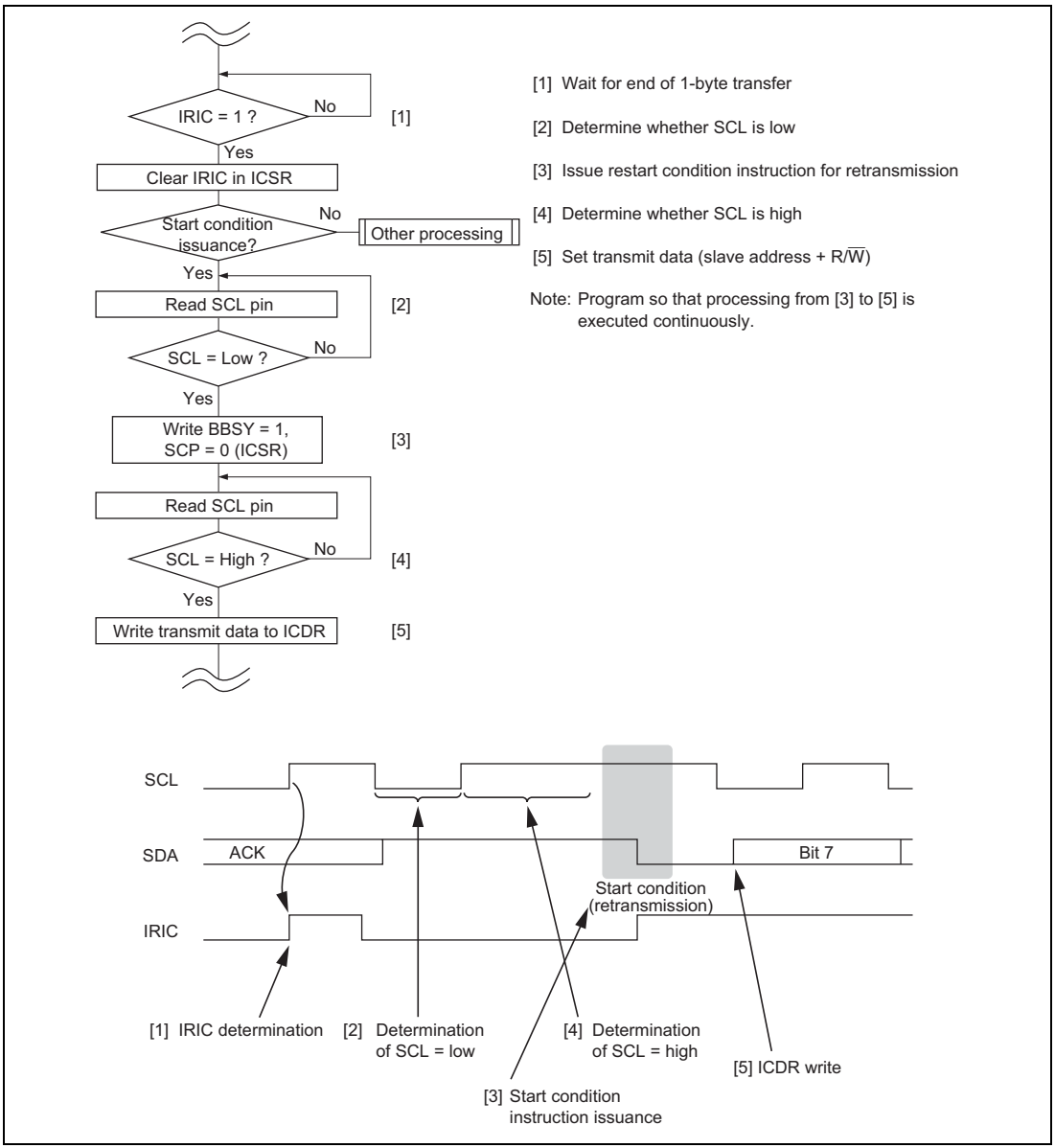


Figure 14.19 Flowcharts and Timing of Start Condition Instruction Issuance for Retransmission

9. Notes on I²C Bus Interface Stop Condition Instruction Issuance

If the rise time of the 9th SCL acknowledge exceeds the specification because the bus load capacitance is large, or if there is a slave device of the type that drives SCL low to effect a wait, issue the stop condition instruction after reading SCL and determining it to be low, as shown below.

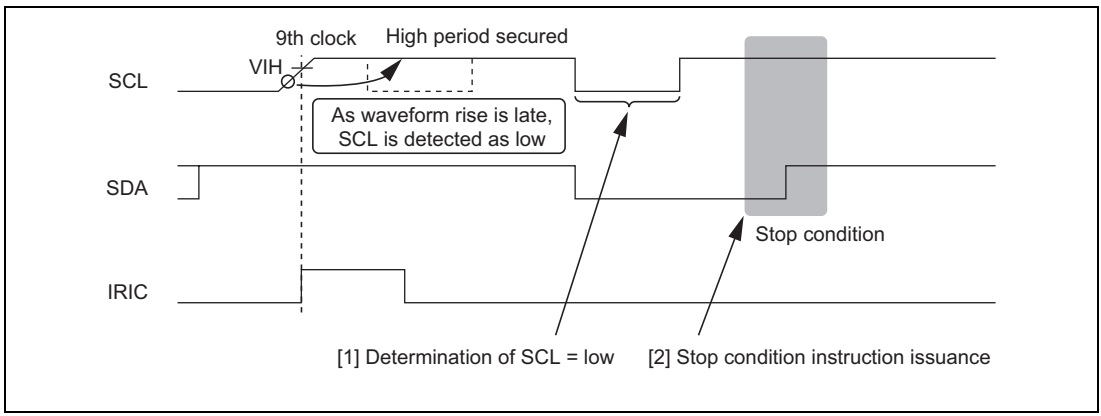


Figure 14.20 Timing of Stop Condition Issuance

10. Notes on IRIC Flag Clearance when Using Wait Function

If the SCL rise time exceeds the designated duration or if the slave device is of the type that keeps SCL low and applies a wait state when the wait function is used in the master mode of the I²C bus interface, read SCL and clear the IRIC flag after determining that SCL has gone low, as shown below.

Clearing the IRIC flag to 0 when WAIT is set to 1 and SCL is being held at high level can cause the SDA value to change before SCL goes low, resulting in a start condition or stop condition being generated erroneously.

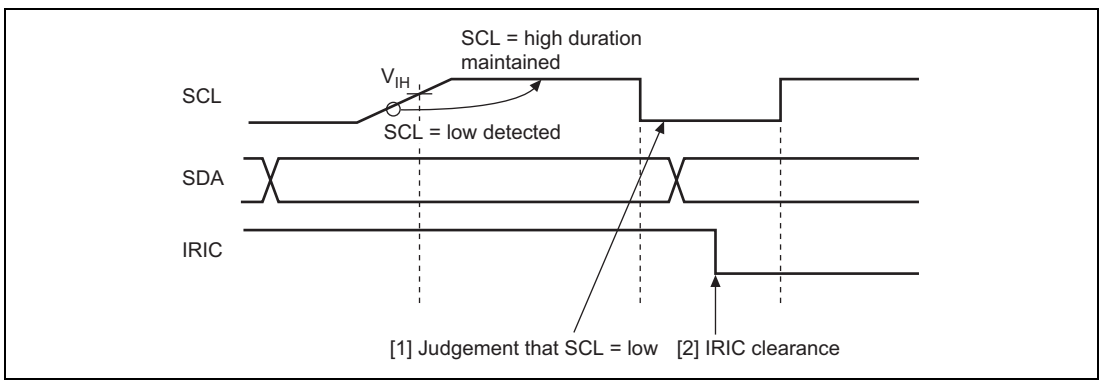


Figure 14.21 IRIC Flag Clearance in WAIT = 1 Status

11. Notes on ICDR Reads and ICCR Access in Slave Transmit Mode

In a transmit operation in the slave mode of the I²C bus interface, do not read the ICDR register or read or write to the ICCR register during the period indicated by the shaded portion in figure 14.22.

Normally, when interrupt processing is triggered in synchronization with the rising edge of the 9th clock cycle, the period in question has already elapsed when the transition to interrupt processing takes place, so there is no problem with reading the ICDR register or reading or writing to the ICCR register.

To ensure that the interrupt processing is performed properly, one of the following two conditions should be applied.

(a) Make sure that reading received data from the ICDR register, or reading or writing to the ICCR register, is completed before the next slave address receive operation starts.

(b) Monitor the BC2 to BC0 counter in the ICMR register and, when the value of BC2 to BC0 is 000 (8th or 9th clock cycle), allow a waiting time of at least 2 transfer clock cycles in order to involve the problem period in question before reading from the ICDR register, or reading or writing to the ICCR register.

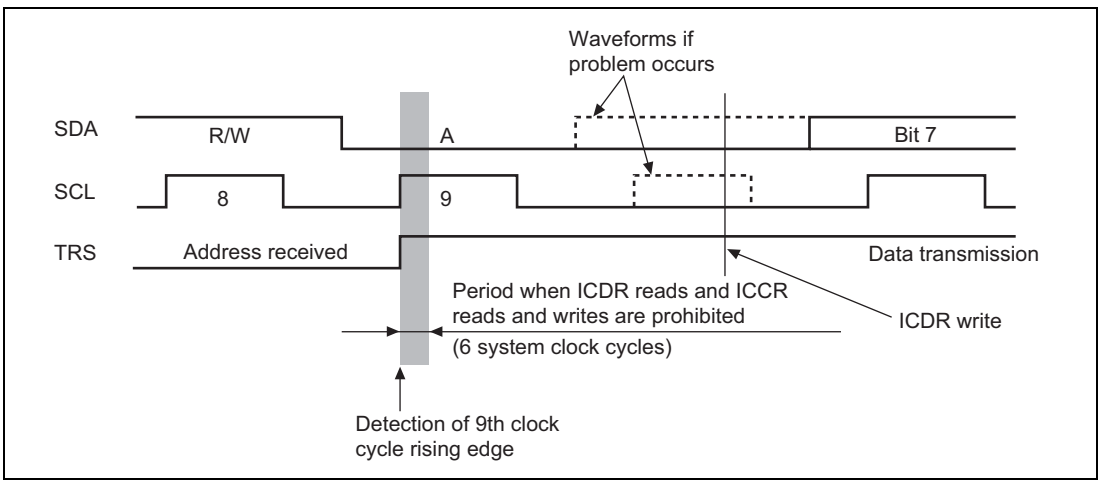


Figure 14.22 ICDR Read and ICCR Access Timing in Slave Transmit Mode

12. Notes on TRS Bit Setting in Slave Mode

From the detection of the rising edge of the 9th clock cycle or of a stop condition to when the rising edge of the next SCL pin signal is detected (the period indicated as (a) in figure 14.23) in the slave mode of the I²C bus interface, the value set in the TRS bit in the ICCR register is effective immediately.

However, at other times (indicated as (b) in figure 14.23) the value set in the TRS bit is put on hold until the next rising edge of the 9th clock cycle or stop condition is detected, rather than taking effect immediately.

This results in the actual internal value of the TRS bit remaining 1 (transmit mode) and no acknowledge bit being sent at the 9th clock cycle address receive completion in the case of an address receive operation following a restart condition input with no stop condition intervening.

When receiving an address in the slave mode, clear the TRS bit to 0 during the period indicated as (a) in figure 14.23.

To cancel the holding of the SCL bit low by the wait function in the slave mode, clear the TRS bit to 0 and then perform a dummy read of the ICDR register.

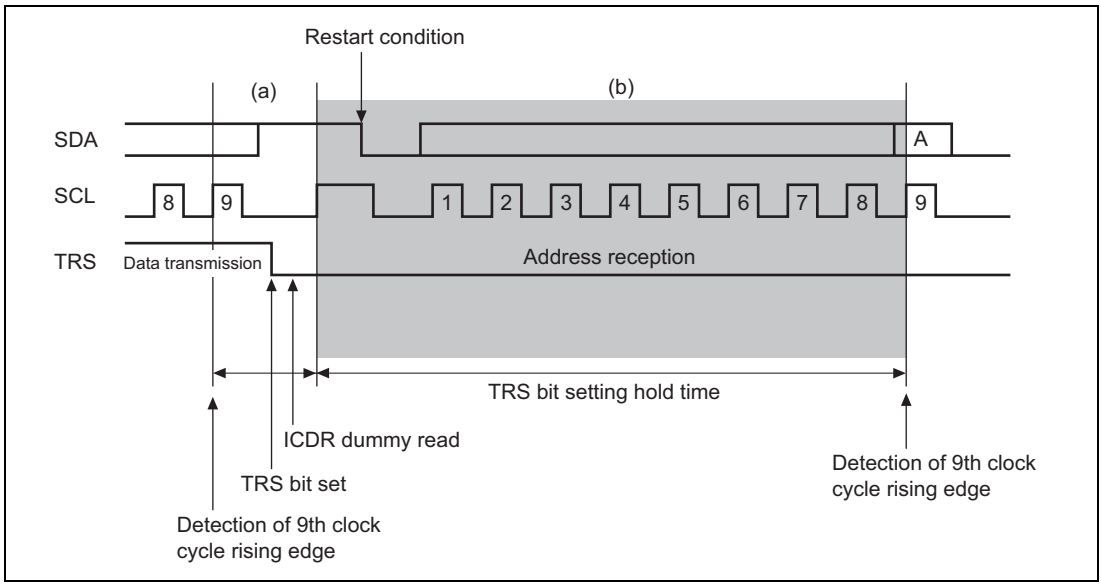


Figure 14.23 TRS Bit Setting Timing in Slave Mode

13. Notes on ICDR Reads in Transmit Mode and ICDR Writes in Receive Mode

When attempting to read ICDR in the transmit mode (TRS = 1) or write to ICDR in the receive mode (TRS = 0) under certain conditions, the SCL pin may not be held low after the completion of the transmit or receive operation and a clock may not be output to the SCL bus line before the ICDR register access operation can take place properly.

When accessing ICDR, always change the setting to the transmit mode before performing a read operation, and always change the setting to the receive mode before performing a write operation.

14. Notes on ACKE Bit and TRS Bit in Slave Mode

When using the I²C bus interface, if an address is received in the slave mode immediately after 1 is received as an acknowledge bit (ACKB = 1) in the transmit mode (TRS = 1), an interrupt may be generated at the rising edge of the 9th clock cycle if the address does not match.

When performing slave mode operations using the IIC bus interface module, make sure to do the following.

(a) When a 1 is received as an acknowledge bit for the final transmit data after completing a series of transmit operations, clear the ACKE bit in the ICCR register to 0 to initialize the ACKB bit to 0.

(b) In the slave mode, change the setting to the receive mode (TRS = 0) before the start condition is input. To ensure that the switch from the slave transmit mode to the slave receive mode is accomplished properly, end the transmission as described in figure 14.14.

15. Notes on Arbitration Lost in Master Mode

The I²C bus interface recognizes the data in transmit/receive frame as an address when arbitration is lost in master mode and a transition to slave receive mode is automatically carried out.

When arbitration is lost not in the first frame but in the second frame or subsequent frame, transmit/receive data that is not an address is compared with the value set in the SAR or SARX register as an address. If the receive data matches with the address in the SAR or SARX register, the I²C bus interface erroneously recognizes that the address call has occurred. (See figure 14.24.)

In multi-master mode, a bus conflict could happen. When The I²C bus interface is operated in master mode, check the state of the AL bit in the ICSR register every time after one frame of data has been transmitted or received.

When arbitration is lost during transmitting the second frame or subsequent frame, take avoidance measures.

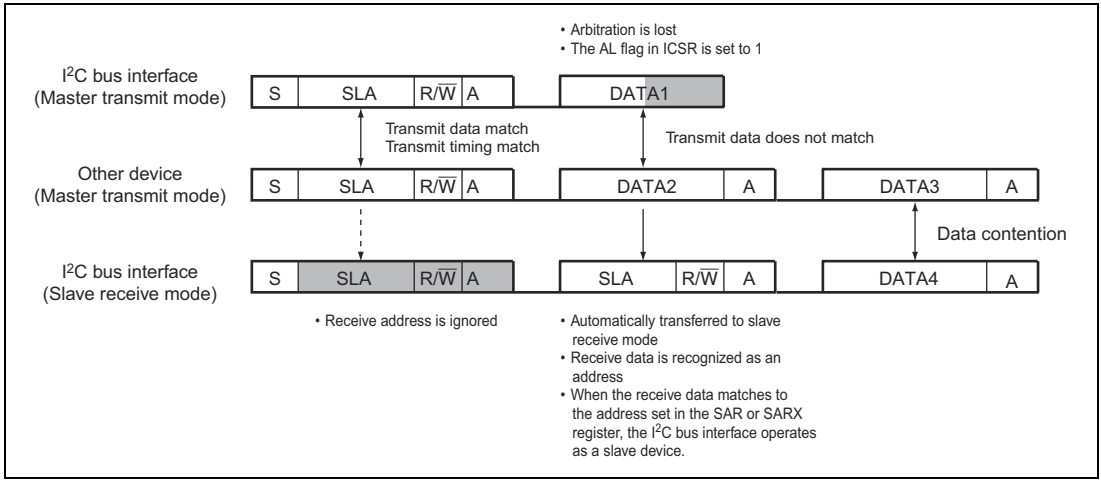


Figure 14.24 Diagram of Erroneous Operation when Arbitration is Lost

Though it is prohibited in the normal I²C protocol, the same problem may occur when the MST bit is erroneously set to 1 and a transition to master mode is occurred during data transmission or reception in slave mode. In multi-master mode, pay attention to the setting of the MST bit when a bus conflict may occur. In this case, the MST bit in the ICCR register should be set to 1 according to the order below.

(a) Make sure that the BBSY flag in the ICCR register is 0 and the bus is free before setting the MST bit.

(b) Set the MST bit to 1.

(c) To confirm that the bus was not entered to the busy state while the MST bit is being set, check that the BBSY flag in the ICCR register is 0 immediately after the MST bit has been set.

16. Notes on Wait Function in Master Mode

While the WAIT bit in ICMR is set to 1 and WAIT in master mode, if the interrupt flag of the IRIC bit is cleared from 1 to 0 between the falling edge of the 7th clock and the falling edge of the 8th clock, the clock pulse of the 9th clock may be output continuously due to the failure to insert a wait after the falling edge of the 8th clock.

When the wait function is used in master mode, clear the IRIC flag after the IRIC flag is set to 1 on the falling edge of the 9th clock and before the rising edge of the 7th clock (the counter value of BC2 to BC0 should be 2 or greater).

If the clearing of the IRIC flag is delayed due to the interrupt or other processes and the value of the RC counter is changed to 1 or 0, confirm that the SCL pins are in the L state after the counter values of BC2 to BC0 are cleared to 0, and then clear the IRIC flag (see figure 14.25).

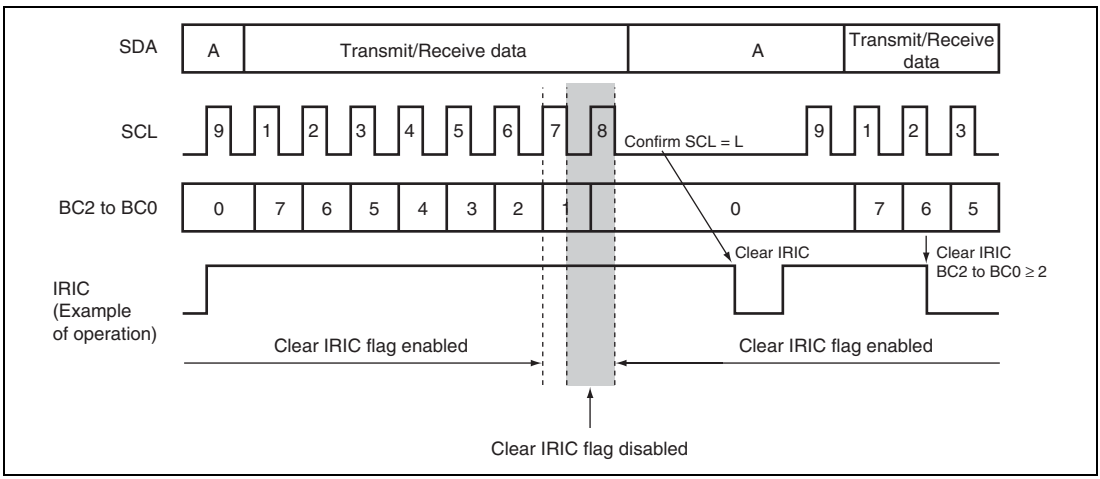


Figure 14.25 IRIC Flag Clear Timing when the Wait Function is Used

17. Module Stop Mode Setting

The IIC operation can be enabled or disabled using the module stop control register. The initial setting is for the IIC operation to be halted. Register access is enabled by canceling module stop mode. For details, refer to section 21, Power-Down Modes.

Section 15 A/D Converter

This LSI includes a successive approximation type 10-bit A/D converter that allows up to twelve analog input channels to be selected. The Block diagram of the A/D converter is shown in figure 15.1.

15.1 Features

- 10-bit resolution
- Twelve input channels
- Conversion time: 13.3 µs per channel (at 20-MHz operation)
- Two operating modes
  - — Single mode: Single-channel A/D conversion
  - — Scan mode: Continuous A/D conversion on 1 to 4 channels
- Four data registers
  - — Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three methods conversion start
  - — Software
  - — 16-bit timer pulse unit (TPU) conversion start trigger
  - — External trigger signal
- Interrupt request
  - — An A/D conversion end interrupt request (ADI) can be generated
- Module stop mode can be set

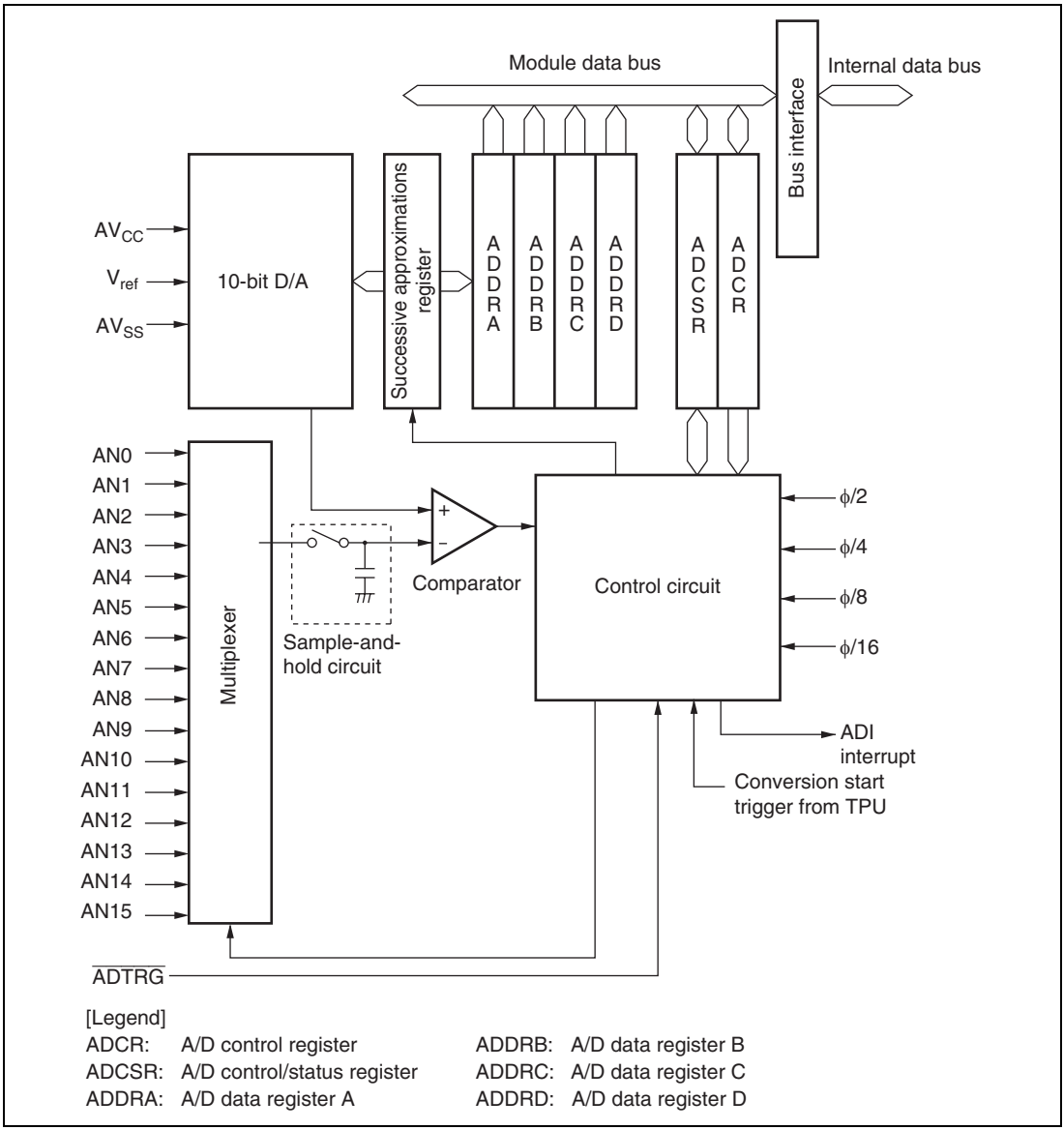


Figure 15.1 Block Diagram of A/D Converter

15.2 Input/Output Pins

Table 15.1 summarizes the input pins used by the A/D converter. The 12 analog input pins are divided into four channel sets and three groups; analog input pins 0 to 3 (AN0 to AN3) comprising group 0, analog input pins 4 to 7 (AN4 to AN7) comprising group 1, and analog input pins 8 to 11 (AN8 to AN11) comprising group 2. The AVcc and AVss pins are the power supply pins for the analog block in the A/D converter.

Table 15.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog block power supply and reference voltage
Analog ground pin	AV_{SS}	Input	Analog block ground and reference voltage
Reference voltage pin	Vref	Input	A/D conversion reference voltage
Analog input pin 0	AN0	Input	Group 0 analog input pins
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	Group 1 analog input pins
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 8	AN8	Input	Group 2 analog input pins
Analog input pin 9	AN9	Input	
Analog input pin 10	AN10	Input	
Analog input pin 11	AN11	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input pin for starting A/D conversion

15.3 Register Description

The A/D converter has the following registers. The MSTPA1 bit in the module stop control

- A/D data register A (ADDRA)
- A/D data register B (ADDRB)
- A/D data register C (ADDRC)
- A/D data register D (ADDRD)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

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

There are four 16-bit read-only ADDR registers; ADDRA to ADDRD, used to store the results of A/D conversion. The ADDR registers, which store a conversion result for each channel, are shown in table 15.2.

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

The data bus between the CPU and the A/D converter is 8 bits wide. The upper byte can be read directly from the CPU, however the lower byte should be read via a temporary register. The temporary register contents are transferred from the ADDR when the upper byte data is read. When reading the ADDR, read the upper byte before the lower byte, or read in word unit. When only the lower byte is read, the contents are not guaranteed.

Table 15.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel				
CH3 = 0		CH3 = 1		A/D Data Register to Be Stored the Results of A/D Conversion
Group 0 (CH2 = 0)	Group 1 (CH2 = 1)	Group 2 (CH2 = 0)	— (CH2 = 1)	
AN0	AN4	AN8	Setting prohibited	ADDRA
AN1	AN5	AN9	Setting prohibited	ADDRB
AN2	AN6	AN10	Setting prohibited	ADDRC
AN3	AN7	AN11	Setting prohibited	ADDRD

15.3.2 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 • When A/D conversion ends on all specified channels [Clearing conditions] • When 0 is written after reading ADF = 1 • When the DTC is activated by an ADI interrupt and ADDR is read
6	ADIE	0	R/W	A/D Interrupt Enable A/D conversion end interrupt (ADI) request enabled when 1 is set

Bit	Bit Name	Initial Value	R/W	Description	
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.	
				Setting this bit to 1 starts A/D conversion. In single mode, this bits is cleared to 0 automatically when conversion on the specified channel is complete. In scan mode, conversion continues sequentially on the specified channels until this bit is cleared to 0 by software, a reset, or a transition to software standby mode, hardware standby mode or module stop mode.	
4	SCAN	0	R/W	Scan Mode	
				Selects single mode or scan mode as the A/D conversion operating mode.	
				0: Single mode	
				1: Scan mode	
3	CH3	0	R/W	Channel Select 0 to 3	
2	CH2	0	R/W	Select analog input channels.	
1	CH1	0	R/W	When SCAN = 0	When SCAN = 1
0	CH0	0	R/W	0000: AN0	0000: AN0
				0001: AN1	0001: AN0 and AN1
				0010: AN2	0010: AN0 to AN2
				0011: AN3	0011: AN0 to AN3
				0100: AN4	0100: AN4
				0101: AN5	0101: AN4 and AN5
				0110: AN6	0110: AN4 to AN6
				0111: AN7	0111: AN4 to AN7
				1000: AN8	1000: AN8
				1001: AN9	1001: AN8 and AN9
				1010: AN10	1010: AN8 to AN10
				1011: AN11	1011: AN8 to AN11
				1100: Setting prohibited	1100: Setting prohibited
				1101: Setting prohibited	1101: Setting prohibited
				1110: Setting prohibited	1110: Setting prohibited
				1111: Setting prohibited	1111: Setting prohibited

15.3.3 A/D Control Register (ADCR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 0 and 1
6	TRGS0	0	R/W	Enables the start of A/D conversion by a trigger signal. Only set bits TRGS0 and TRGS1 while conversion is stopped (ADST = 0). 00: A/D conversion start by software is enabled 01: A/D conversion start by TPU conversion start trigger is enabled 10: Setting prohibited 11: A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is enabled
5, 4	—	All 1	—	Reserved These bits are always read as 1.
3	CKS1	0	R/W	Clock Select 0 and 1
2	CKS0	0	R/W	These bits specify the A/D conversion time. The conversion time should be changed only when ADST = 0. Specify a setting that gives a value within the range shown in table 20.7. 00: Conversion time = 530 states (max.) 01: Conversion time = 266 states (max.) 10: Conversion time = 134 states (max.) 11: Conversion time = 68 states (max.)
1, 0	—	All 1	—	Reserved These bits are always read as 1.

15.4 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, in order to prevent incorrect operation, first clear the bit ADST to 0 in ADCSR. The ADST bit can be set at the same time as the operating mode or analog input channel is changed.

15.4.1 Single Mode

In single mode, A/D conversion is to be performed only once on the specified single channel. The operations are as follows.

1. A/D conversion is started when the ADST bit 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. When A/D converion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters the wait state.

15.4.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the specified channels (four channels maximum). The operations are as follows.

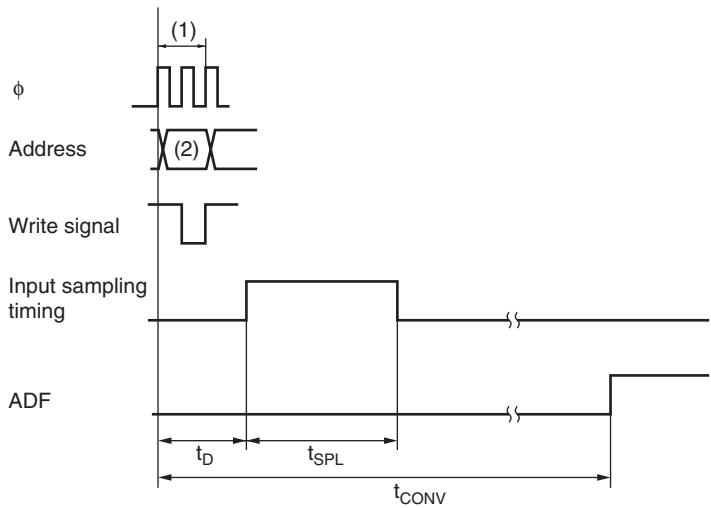
1. When the ADST bit is set to 1 by software, TPU or external trigger input, A/D conversion starts on the first channel in the group (AN0 when CH3 and CH2 = 00, AN4 when CH3 and CH2 = 01, or AN8 when CH3 and CH2 = 10).
2. When A/D conversion for each channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF flag is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested after A/D conversion ends. Conversion of the first channel in the group starts again.
4. Steps 2 to 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.

15.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input when the A/D conversion start delay time (t_D) has passed after the ADST bit is set to 1, then starts conversion. Figure 15.2 shows the A/D conversion timing. Table 15.3 shows the A/D conversion time.

As indicated in figure 15.2, 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 table 15.3.

In scan mode, the values given in table 15.3 apply to the first conversion time. The values given in table 15.4 apply to the second and subsequent conversions. In both cases, set bits CKS1 and CKS0 in ADCR to give an A/D conversion time within the range shown in table 23.10.



Legend:
(1): ADCSR write cycle
(2) ADCSR address
t_D: A/D conversion start delay
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 15.2 A/D Conversion Timing

Table 15.3 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	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: All values represent the number of states.

Table 15.4 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)

15.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS0 and TRGS1 bits are set to 11 in ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{ADTRG}}$ pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single and scan modes, are the same as when the bit ADST has been set to 1 by software. Figure 15.3 shows the timing.

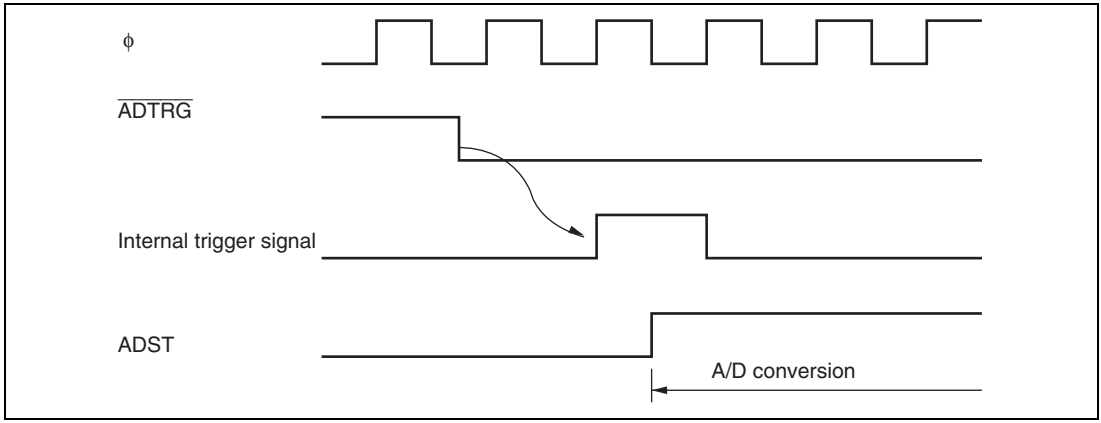


Figure 15.3 External Trigger Input Timing

15.5 Interrupts

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 ADI interrupt requests while the bit ADF in ADCSR is set to 1 after A/D conversion is completed. The DTC can be activated by an ADI interrupt. Having the converted data read by the DTC in response to an ADI interrupt enables continuous conversion without imposing a load on software.

Table 15.5 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Source Flag	DTC Activation
ADI	A/D conversion completed	ADF	Possible

15.6 A/D Conversion Precision Definitions

This LSI's A/D conversion precision 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 15.4).
- 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'00) to B'0000000001 (H'01) (see figure 15.5).
- 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'3E) to B'1111111111 (H'3F) (see figure 15.5).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between zero voltage and full-scale voltage. Does not include offset error, full-scale error, or quantization error (see figure 15.5).
- Absolute precision

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

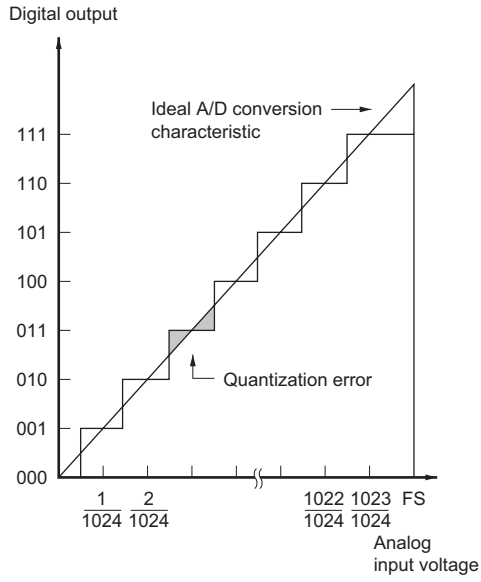


Figure 15.4 A/D Conversion Precision Definitions

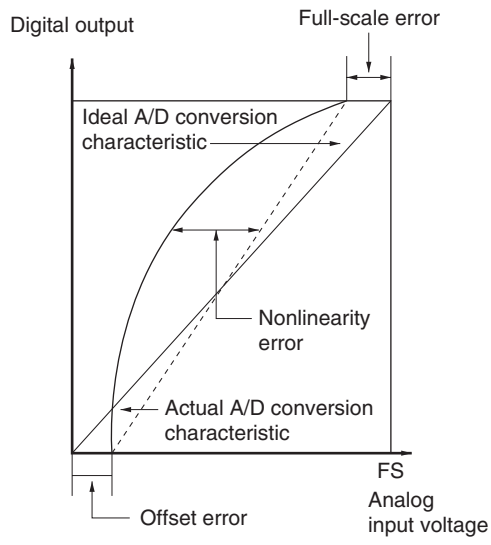


Figure 15.5 A/D Conversion Precision Definitions

15.7 Usage Notes

15.7.1 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 21, Power-Down Modes.

15.7.2 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion precision is guaranteed for an input signal for which the signal source impedance is 5 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 5 kΩ, charging may be insufficient and it may not be possible to guarantee A/D conversion precision. However, for A/D conversion in single mode with a large capacitance provided externally, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored. However, as a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/µs or greater) (see figure 15.6). When converting a high-speed analog signal, a low-impedance buffer should be inserted.

15.7.3 Influences on Absolute Precision

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. 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 communicate with digital signals on the mounting board (i.e, acting as antennas).

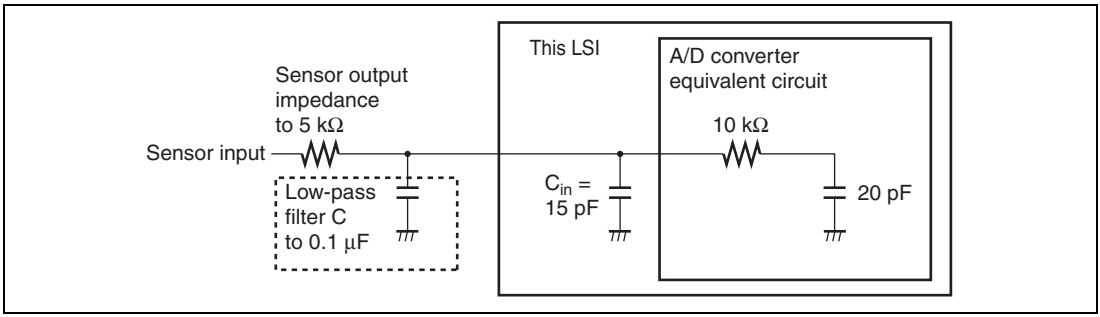


Figure 15.6 Example of Analog Input Circuit

15.7.4 Range of Analog Power Supply and Other Pin Settings

If the conditions below are not met, the reliability of the device 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 ≤ V_{AN} ≤ AVcc.
- Relationship between AVcc, AVss and Vcc, Vss
  Set AVss = Vss as the relationship between AVcc, AVss and Vcc, Vss. If the A/D converter is not used, the AVcc and AVss pins must not be left open.
- Range of Vref pin setting
  Vref pin reference voltage setting should be in the range Vref ≤ AVCC.

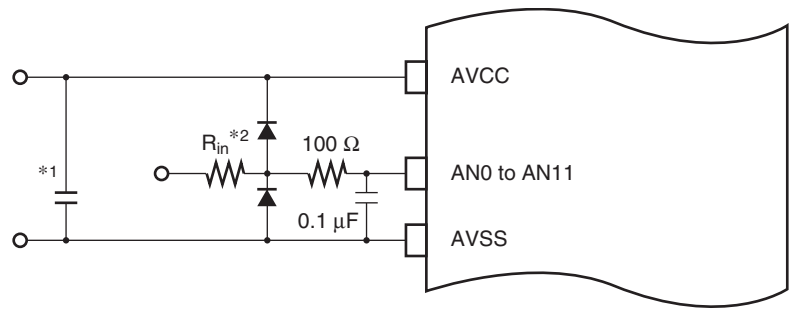
15.7.5 Notes on Board Design

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 signals (AN0 to AN11), and analog power supply (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.

15.7.6 Notes on Noise Countermeasures

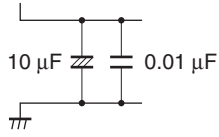
A protection circuit should be connected in order to prevent damage due to abnormal voltage, such as an excessive surge at the analog input pins (AN0 to AN11), between AVcc and AVss, as shown in figure 15.7. Also, the bypass capacitors connected to AVcc and the filter capacitor connected to AN0 to AN11 must be connected to AVss.

If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN11) 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 circuit constants.



Notes: Values are reference values.

1.

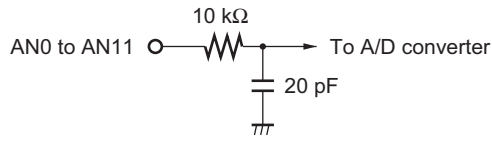


2. R_{in}: Input impedance

Figure 15.7 Example of Analog Input Protection Circuit

Table 15.6 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	5	kΩ



Note: Values are reference values.

Figure 15.8 Analog Input Pin Equivalent Circuit

Section 16 D/A Converter

16.1 Features

- 8-bit resolution
- Two output channels
- Conversion time: Max. 10 µs (when load capacitance is 20 pF)
- Output voltage: 0 V to Vref
- D/A output retaining function in software standby mode

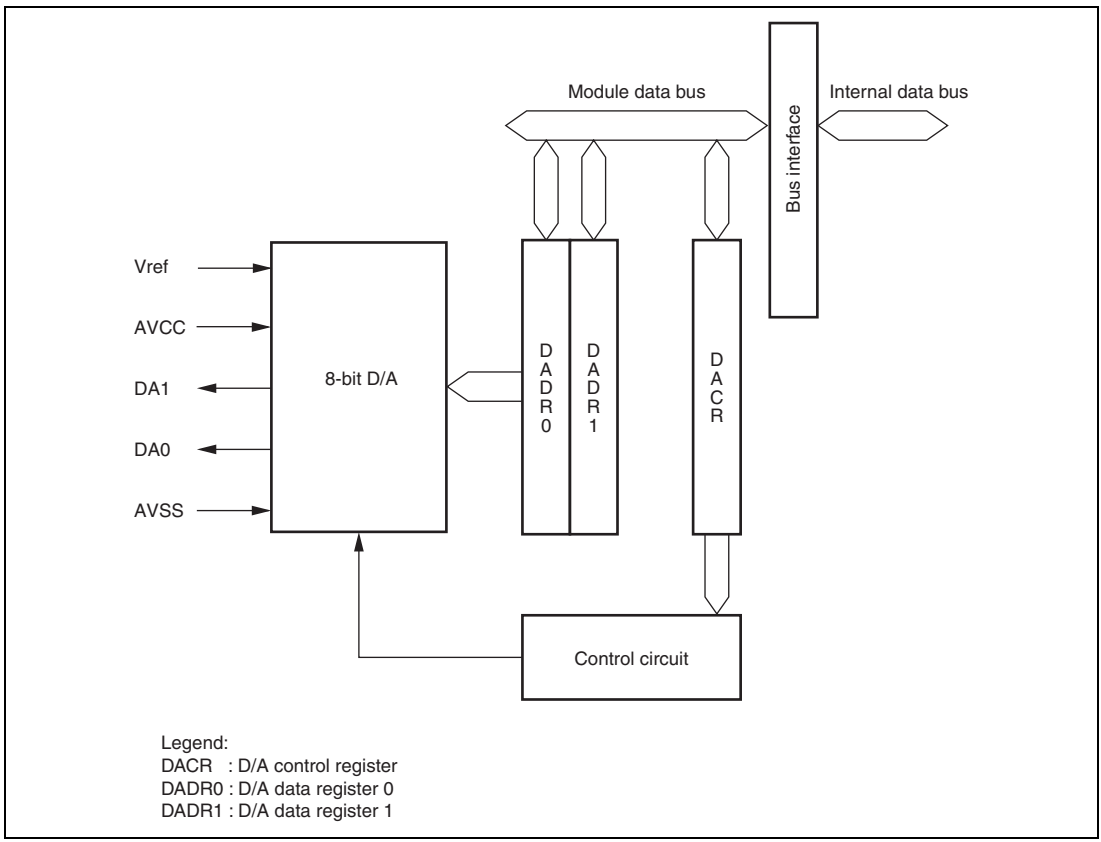


Figure 16.1 Block Diagram of D/A Converter

16.2 Input/Output Pins

Table 16.1 summarizes the input/output pins used by the D/A converter.

Table 16.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVCC	Input	Analog block power supply
Analog ground pin	AVSS	Input	Analog block ground and reference voltage
Analog output pin 0	DA0	Output	Channel 0 analog output
Analog output pin 1	DA1	Output	Channel 1 analog output
Reference power supply pin	Vref	Input	Analog block reference voltage

16.3 Register Descriptions

The D/A converter has the following registers.

- D/A data register 0 (DADR0)
- D/A data register 1 (DADR1)
- D/A control register (DACR)

16.3.1 D/A Data Registers 0 and 1 (DADR0, DADR1)

DADR0 and DADR1 are 8-bit readable/writable registers that store data for D/A conversion. When analog output is permitted, DADR contents are converted and output to analog output pins. DADR0 and DADR1 are initialized to H'00.

16.3.2 D/A Control Register (DACR)

DACR controls D/A converter operation.

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE1	0	R/W	D/A Output Enable 1 Controls D/A conversion and analog output. 0: Analog output DA1 is disabled 1: D/A conversion for channel 1 and analog output DA1 are enabled
6	DAOE0	0	R/W	D/A Output Enable 0 Controls D/A conversion and analog output. 0: Analog output DA0 is disabled 1: D/A conversion for channel 0 and analog output DA0 are enabled
5	DAE	0	R/W	D/A Enable Controls D/A conversion in conjunction with the DAOE0 and DAOE1 bits. When the DAE bit is cleared to 0, D/A conversion for channels 0 and 1 is controlled individually. When the DAE bit is set to 1, D/A conversion for channels 0 and 1 are controlled as one. Conversion result output is controlled by the DAOE0 and DAOE1 bits. For details, see table 16.2 below.
4 to 0	—	All 1	R	Reserved These bits are always read as 1 and cannot be modified.

Table 16.2 D/A Channel Enable

Bit 7	Bit 6	Bit 5	
DAOE1	**DAOE0**	**DAE**	**Description**
0	0	—	Disables D/A conversion
	1	0	Enables D/A conversion for channel 0
			Disables D/A conversion for channel 1
		1	Enables D/A conversion for channels 0 and 1
1	0	0	Disables D/A conversion for channel 0
			Enables D/A conversion for channel 1
		1	Enables D/A conversion for channels 0 and 1
	1	—	Enables D/A conversion for channels 0 and 1

16.4 Operation

The D/A converter incorporates two channels of the D/A circuits and can be converted individually.

When the DAOE bit in DACR is set to 1, D/A conversion is enabled and conversion results are output.

An example of D/A conversion of channel 0 is shown below. The operation timing is shown in figure 16.2.

1. Write conversion data to DADR0.
2. When the DAOE0 bit in DACR is set to 1, D/A conversion starts. After the interval of t_{DCONV}, conversion results are output from the analog output pin DA0. The conversion results are output continuously until DADR0 is modified or the DAOE0 bit is cleared to 0. The output value is calculated by the following formula:

   DADR contents/256 × Vref

3. Conversion starts immediately after DADR0 is modified. After the interval of t_{DCONV}, conversion results are output.
4. When the DAOE0 bit is cleared to 0, analog output is disabled.

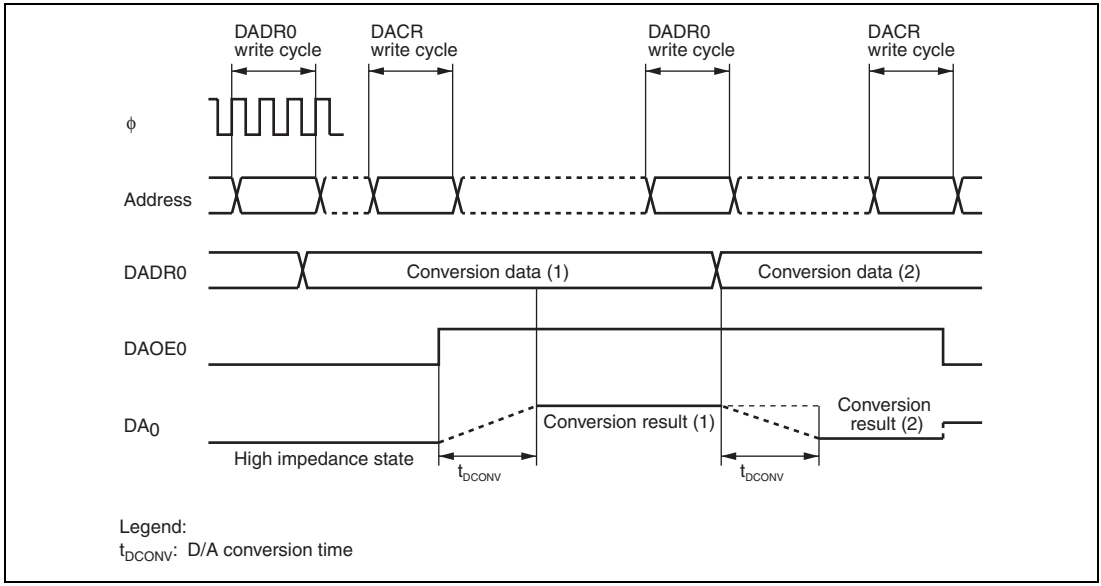


Figure 16.2 D/A Converter Operation Example

16.5 Usage Note

When this LSI enters software standby mode with D/A conversion enabled, the D/A output is retained, and the analog power supply current is equal to as during D/A conversion. If the analog power supply current needs to be reduced in software standby mode, clear the DAOE1, DAOE0, and DAE bits all to 0 to disable D/A output.

16.5.1 Module Stop Mode Setting

D/A converter operation can be enabled or disabled using the module stop control register. The initial setting is for D/A converter operation to be halted. Register access is enabled by canceling module stop mode. For details, refer to section 21, Power-Down Modes.

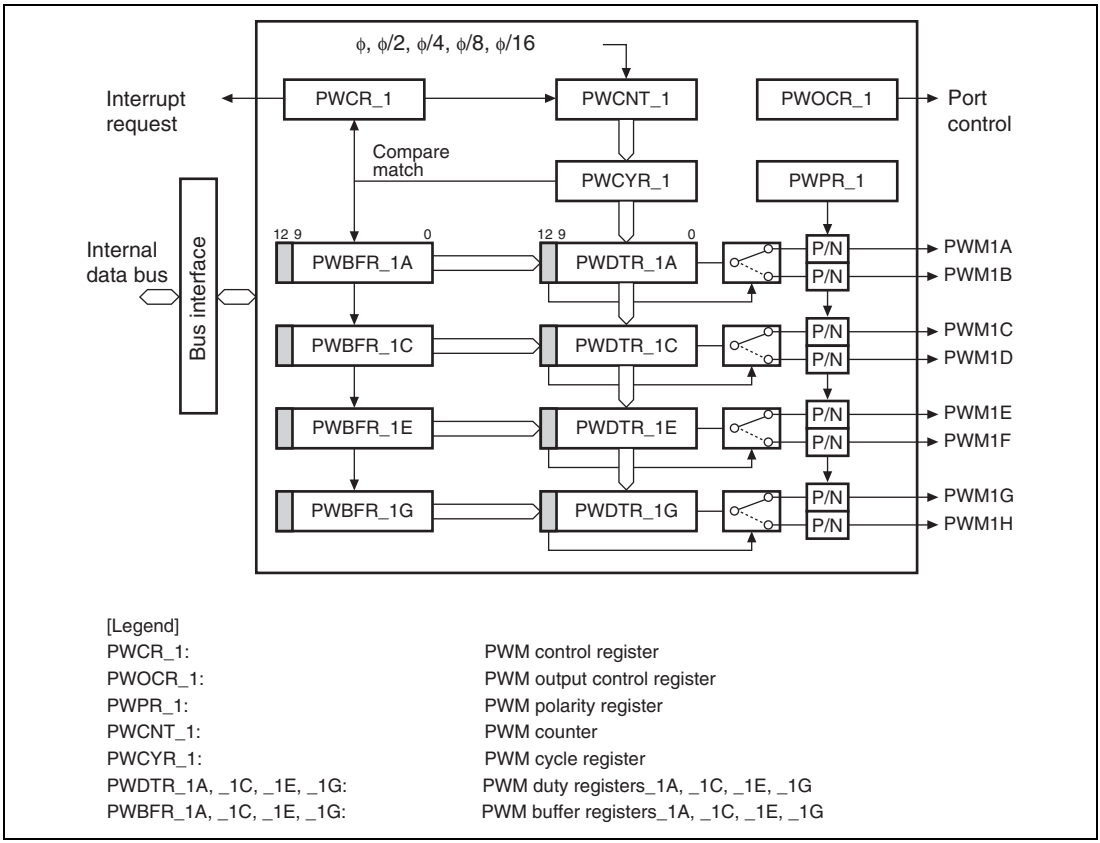
Section 17 Motor Control PWM Timer (PWM)

This LSI has on-chip motor control PWM (pulse width modulator) with a maximum capability of 16 pulse outputs in total.

17.1 Features

- Maximum of 16 pulse outputs
  - Two 10-bit PWM channels, each with eight outputs.
  - 10-bit counter (PWCNT) and cycle register (PWCYR).
  - Duty and output polarity can be set for each output.
- Automatic data transfer in every cycle
  - Each of four duty registers (PWDTR) is provided with buffer registers (PWBFR), with data transferred automatically every cycle.
  - Channel 1 is provided with four duty registers and four buffer registers.
  - Channel 2 is provided with eight duty registers and four buffer registers.
- Duty settings selectable
  - A duty cycle of 0% to 100% can be selected by means of a duty register setting.
- Operating clock selectable
  - There is a choice of five operating clocks (ϕ, ϕ/2, ϕ/4, ϕ/8, ϕ/16).
- High-speed access via internal 16-bit bus
- Two interrupt sources
  - An interrupt can be requested independently for each channel by a cycle register compare match.
- Automatic transfer of register data
  - Block transfer and one-word data transfer are available by activating the data transfer controller (DTC).
- Module stop mode can be set

Figure 17.1 shows a block diagram of PWM_1.

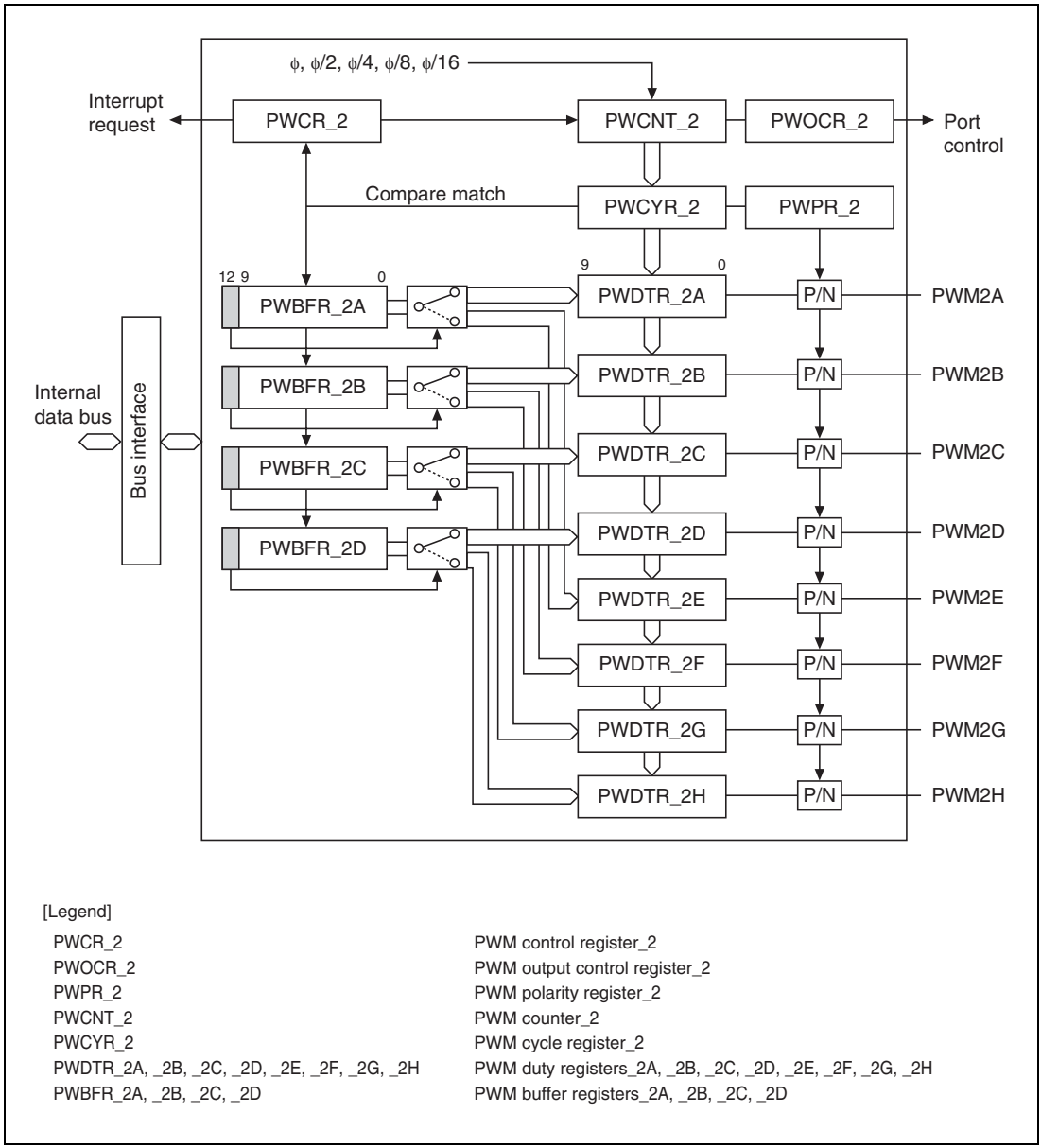


[Legend]

PWCR_1:	PWM control register
PWOCR_1:	PWM output control register
PWPR_1:	PWM polarity register
PWCNT_1:	PWM counter
PWCYR_1:	PWM cycle register
PWDTR_1A, _1C, _1E, _1G:	PWM duty registers_1A, _1C, _1E, _1G
PWBFR_1A, _1C, _1E, _1G:	PWM buffer registers_1A, _1C, _1E, _1G

Figure 17.1 Block Diagram of PWM_1

Figure 17.2 shows a block diagram of PWM_2.



[Legend]

PWCR_2	PWM control register_2
PWOCR_2	PWM output control register_2
PWPR_2	PWM polarity register_2
PWCNT_2	PWM counter_2
PWCYR_2	PWM cycle register_2
PWDTR_2A, _2B, _2C, _2D, _2E, _2F, _2G, _2H	PWM duty registers_2A, _2B, _2C, _2D, _2E, _2F, _2G, _2H
PWBFR_2A, _2B, _2C, _2D	PWM buffer registers_2A, _2B, _2C, _2D

Figure 17.2 Block Diagram of PWM_2

17.2 Input/Output Pins

Table 17.1 shows the PWM pin configuration.

Table 17.1 Pin Configuration

Channel	Name	Abbrev.	I/O	Function
1	PWM output pin 1A	PWM1A	Output	Channel 1A PWM output
	PWM output pin 1B	PWM1B	Output	Channel 1B PWM output
	PWM output pin 1C	PWM1C	Output	Channel 1C PWM output
	PWM output pin 1D	PWM1D	Output	Channel 1D PWM output
	PWM output pin 1E	PWM1E	Output	Channel 1E PWM output
	PWM output pin 1F	PWM1F	Output	Channel 1F PWM output
	PWM output pin 1G	PWM1G	Output	Channel 1G PWM output
	PWM output pin 1H	PWM1H	Output	Channel 1H PWM output
2	PWM output pin 2A	PWM2A	Output	Channel 2A PWM output
	PWM output pin 2B	PWM2B	Output	Channel 2B PWM output
	PWM output pin 2C	PWM2C	Output	Channel 2C PWM output
	PWM output pin 2D	PWM2D	Output	Channel 2D PWM output
	PWM output pin 2E	PWM2E	Output	Channel 2E PWM output
	PWM output pin 2F	PWM2F	Output	Channel 2F PWM output
	PWM output pin 2G	PWM2G	Output	Channel 2G PWM output
	PWM output pin 2H	PWM2H	Output	Channel 2H PWM output

17.3 Register Descriptions

The PWM has the following registers for each channel.

- PWM control register (PWCR)
- PWM output control register (PWOCR)
- PWM polarity register (PWPR)
- PWM counter (PWCNT)
- PWM cycle register (PWCYR)
- PWM duty register (PWDTR)
- PWM buffer register (PWBFR)

17.3.1 PWM Control Register (PWCR)

PWCR performs interrupt control, starting/stopping of the counter, and counter clock selection. It also contains a flag that indicates a compare match with PWCYR.

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	IE	0	R/W	Interrupt Enable Enables or disables an interrupt request in the event of a compare match with PWCYR of the corresponding channel. 0: Interrupt disabled 1: Interrupt enabled
4	CMF	0	R/(W)*	Compare Match Flag Indicates the occurrence of a compare match with PWCYR of the corresponding channel. [Setting condition] When PWCNT = PWCYR [Clearing condition] • When 0 is written to CMF after reading CMF = 1 • When the DTC is activated by a compare match interrupt, and the DISEL bit in MRB of the DTC is 0
3	CST	0	R/W	Counter Start Selects starting or stopping of PWCNT of the corresponding channel. 0: PWCNT is stopped 1: PWCNT is started

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select
1	CKS1	0	R/W	These bits select the operating clock for PWCNT of the corresponding channel.
0	CKS0	0	R/W	000: Counts on ϕ/1 001: Counts on ϕ/2 010: Counts on ϕ/4 011: Counts on ϕ/8 1xx: Counts on ϕ/16

[Legend]

x: Don't care

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

17.3.2 PWM Output Control Register (PWOCR)

PWOCR enables or disables PWM output.

Bit	Bit Name	Initial Value	R/W	Description
7	OEnH	0	R/W	Output Enable
6	OEnG	0	R/W	Each of these bits enables or disables the corresponding PWM output. 0: PWM output disabled 1: PWM output enabled
5	OEnF	0	R/W	
4	OEnE	0	R/W	
3	OEnD	0	R/W	
2	OEnC	0	R/W	
1	OEnB	0	R/W	
0	OEnA	0	R/W	

(n = 1, 2)

17.3.3 PWM Polarity Register (PWPR)

PWPR selects the PWM output polarity.

Bit	Bit Name	Initial Value	R/W	Description
7	OPSnH	0	R/W	Output Polarity Select
6	OPSnG	0	R/W	Each of these bits selects the PWM output polarity.
5	OPSnF	0	R/W	0: PWM direct output
4	OPSnE	0	R/W	1: PWM inverse output
3	OPSnD	0	R/W	
2	OPSnC	0	R/W	
1	OPSnB	0	R/W	
0	OPSnA	0	R/W	

(n = 1, 2)

17.3.4 PWM Counter (PWCNT)

PWCNT is a 10-bit up-counter incremented by the input clock. The input clock is selected by clock select bits CKS2 to CKS0 in PWCR.

PWCNT can not be directly accessed by the CPU. PWCNT is initialized to H'FC00, when CST bit in PWCR is 0.

17.3.5 PWM Cycle Register (PWCYR)

PWCYR is a 16-bit readable/writable register that sets the PWM conversion cycle. When a PWCYR compare match occurs, PWCNT is cleared and data is transferred from the buffer register (PWBFR) to the duty register (PWDTR).

PWCYR should be written to only while PWCNT is stopped. A value of H'FC00 must not be set. PWCYR is initialized to H'FFFF.

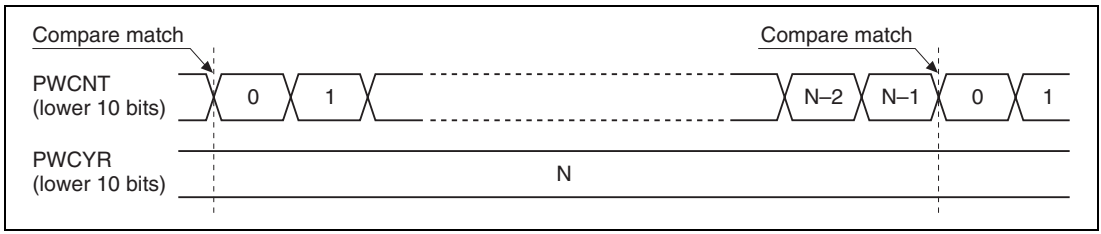


Figure 17.3 Cycle Register Compare Match

17.3.6 PWM Duty Registers (PWDTR)

- PWM_1

PWDTR_1 has four PWDTR registers (PWDTR_A, PWDTR_C, PWDTR_E, and PWDTR_G). The PWDTR_1A is used for outputs PWM1A and PWM1B, PWDTR_1C for outputs PWM1C and PWM1D, PWDTR_1E for outputs PWM1E and PWM1F, and PWDTR_1G for outputs PWM1G and PWM1H.

PWDTR_1 cannot be directly accessed by the CPU. When a PWCYR_1 compare match occurs, data is transferred from the buffer register (PWBFR) to the duty register (PWDTR).

PWDTR_1 is initialized to H'EC00 when the CST bit is 0.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 1	—	Reserved
12	OTS	0	—	Output Terminal Select Bit 12 selects the pin used for PWM output according to the value in bit 12 in the buffer registers that is transferred by a PWCYR_1 compare match. Unselected pins output a low level (or a high level when the corresponding bit in PWPR_1 is set to 1). For details, see table 17.2.
11, 10	—	1	—	Reserved
9	DT9	0	—	Duty These bits specify the PWM output duty. A high level (or a low level when the corresponding bit in PWPR is set to 1) is output from the time PWCNT_1 is cleared by a PWCYR_1 compare match until a PWDTR_1 compare match occurs. When all of the bits are 0, there is no high-level (or low-level when the corresponding bit in PWPR is set to 1) output period.
8	DT8	0	—	
7	DT7	0	—	
6	DT6	0	—	
5	DT5	0	—	
4	DT4	0	—	
3	DT3	0	—	
2	DT2	0	—	
1	DT1	0	—	
0	DT0	0	—	

Table 17.2 Output Selection by OTS Bit

	Bit 12	
Register	**OTS**	**Description**
PWDTR_1A	0	PWM1A output selected
	1	PWM1B output selected
PWDTR_1C	0	PWM1C output selected
	1	PWM1D output selected
PWDTR_1E	0	PWM1E output selected
	1	PWM1F output selected
PWDTR_1G	0	PWM1G output selected
	1	PWM1H output selected

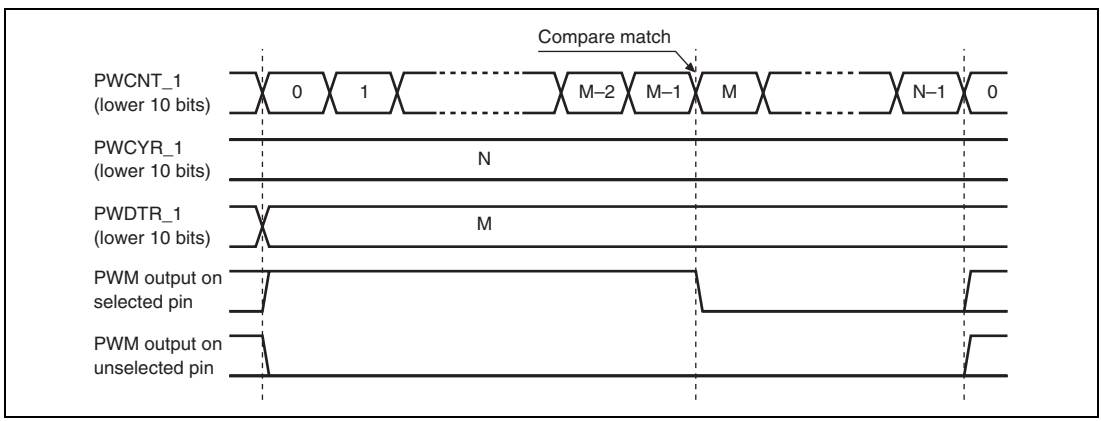


Figure 17.4 Duty Register Compare Match (OPS = 0 in PWPR_1)

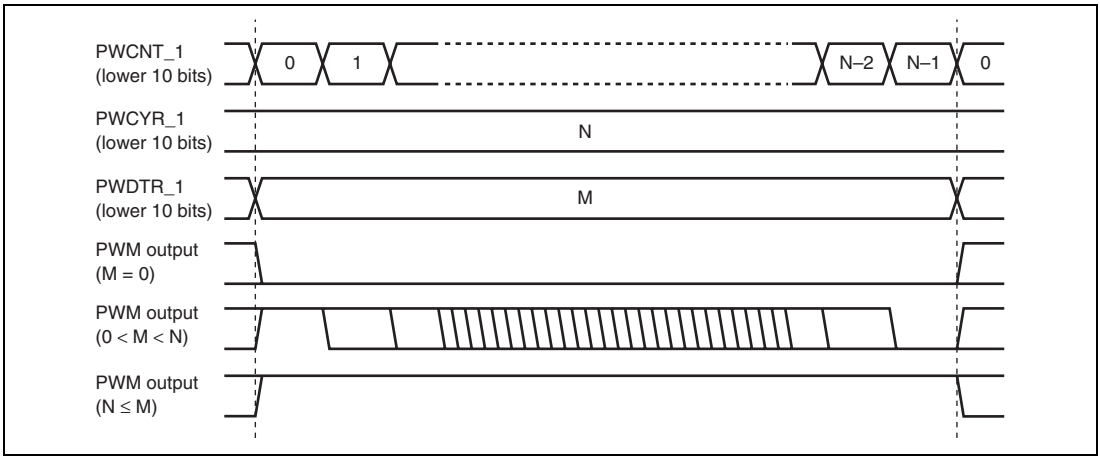


Figure 17.5 Differences in PWM Output According to Duty Register Set Value (OPS = 0 in PWPR_1)

- PWM_2

PWDTR_2 has eight registers (PWDTR2A to 2H).
The PWDTR_2A is used for outputs PWM2A, PWDTR_2B for outputs PWM2B, PWDTR_2C for outputs PWM2C, PWDTR_2D for outputs PWM2D, PWDTR_2E for outputs PWM2E, PWDTR_2F for outputs PWM2F, PWDTR_2G for outputs PWM2G, PWDTR_2H for outputs PWM2H.

PWDTR_2 cannot be directly read or written. When a PWCYR_2 compare match occurs, data is transferred from the buffer register_2 (PWBFR_2) to the PWDTR_2.

PWDTR_2 is initialized to H'EC00 when the Counter Start bit (CST) in PWCR_2 is 0.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	—All 1	—	Reserved
12	—	0	—	Reserved
11	—	1	—	Reserved
10	—	1	—	
9	DT9	0	—	Duty
8	DT8	0	—	Bits 9 to 0 set PWM output duty according to the values in bits 9 to 0 in the buffer register that is transferred by a PWCYR_2 compare match.
7	DT7	0	—	
6	DT6	0	—	A high level (or a low level when the corresponding bit in PWPR is set to 1) is output from the time PWCNT_2 is cleared by a PWCYR_2 compare match until a PWDTR compare match occurs. When all of the bits are 0, there is no high-level (or low-level when the corresponding bit in PWPR is set to 1) output period.
5	DT5	0	—	
4	DT4	0	—	
3	DT3	0	—	
2	DT2	0	—	
1	DT1	0	—	
0	DT0	0	—	

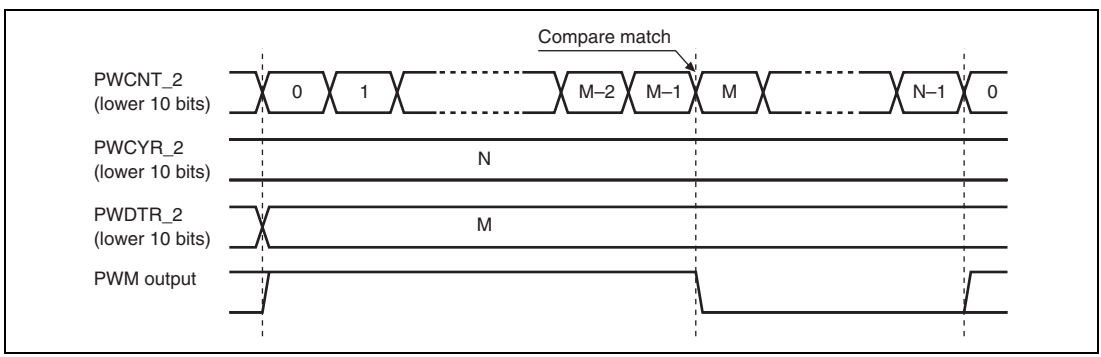


Figure 17.6 Duty Register Compare Match (OPS = 0 in PWPR_2)

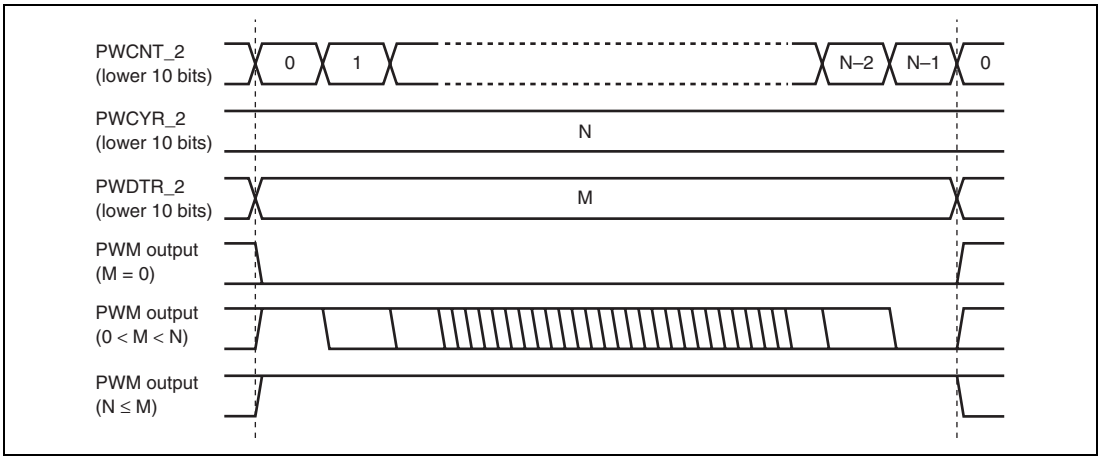


Figure 17.7 Differences in PWM Output According to Duty Register Set Value (OPS = 0 in PWPR_2)

17.3.7 PWM Buffer Register (PWBFR)

- PWM_1

There are four PWBFR_1 registers (PWBFR_A, PWBFR_C, PWBFR_E, and PWBFR_G). When a PWCYR_1 compare match occurs, data is transferred from the buffer register (PWBFR) to the duty register (PWDTR).

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
12	OTS	0	R/W	Output Terminal Select Holds the data to be sent to bit 12 in PWDTR1.
11	—	1	—	Reserved
10	—	1	—	These bits are always read as 1 and cannot be modified.
9	DT9	0	R/W	Duty These bits hold the data to be sent to bits 9 to 0 in PWDTR_1.
8	DT8	0	R/W	
7	DT7	0	R/W	
6	DT6	0	R/W	
5	DT5	0	R/W	
4	DT4	0	R/W	
3	DT3	0	R/W	
2	DT2	0	R/W	
1	DT1	0	R/W	
0	DT0	0	R/W	

- PWM_2

There are four 16 bit read/write PWBFR_2 registers (PWBFR2A to PWBFR2D). When a PWCYR_2 compare match occurs, data is transferred from PWBFR_2A to PWDTR_2A or PWDTR_2E, from PWBFR_2B to PWDTR_2B or PWDTR_2F, from PWBFR_2C to PWDTR_2C or PWDTR_2G, and from PWBFR_2D to PWDTR_2D or PWDTR_2H. The transfer determination is determined by the value of the TDS bit.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
12	TDS	0	R/W	Transfer Destination Select (TDS) Bit 12 selects the PWDTR2 register to which data is to be transferred.
11	—	1	—	Reserved
10	—	1	—	These bits are always read as 1 and cannot be modified.
9	DT9	0	R/W	Duty
8	DT8	0	R/W	The data is transferred to bits 9 to 0 in PWDTR_2.
7	DT7	0	R/W	
6	DT6	0	R/W	
5	DT5	0	R/W	
4	DT4	0	R/W	
3	DT3	0	R/W	
2	DT2	0	R/W	
1	DT1	0	R/W	
0	DT0	0	R/W	

Table 17.3 Output Selection by TDS Bit

	Bit 12	
Register	**TDS**	**Description**
PWBFR_2A/	0	PWDTR_2A selected
	1	PWDTR_2E selected
PWBFR_2B	0	PWDTR_2B selected
	1	PWDTR_2F selected
PWBFR_2C	0	PWDTR_2C selected
	1	PWDTR_2G selected
PWBFR_2D	0	PWDTR_2D selected
	1	PWDTR_2H selected

17.4 Bus Master Interface

17.4.1 16-Bit Data Registers

PWCYR and PWBFR are 16-bit registers. These registers are linked to the bus master by a 16-bit data bus, and can be read or written in 16-bit units. They cannot be read or written by 8-bit access; 16-bit access must always be used.

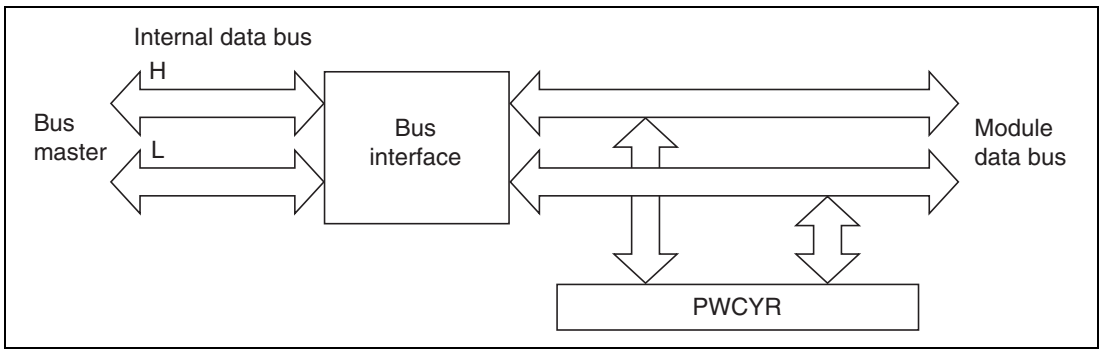


Figure 17.8 16-Bit Register Access Operation (Bus Master ↔ PWCYR (16 Bits))

17.4.2 8-Bit Data Registers

PWCR, PWOCR, and PWPR are 8-bit registers that can be read and written to in 8-bit units. These registers are linked to the bus master by a 16-bit data bus, and can be read or written by 16-bit access; in this case, the lower eight bits are read as an undefined value.

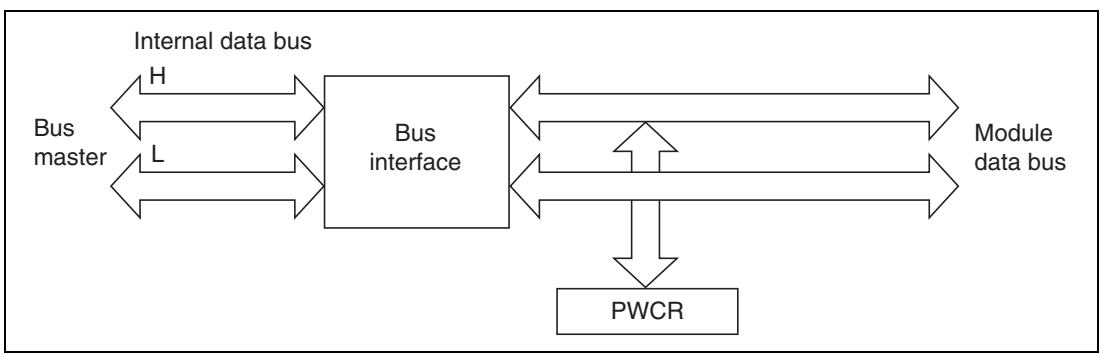


Figure 17.9 8-Bit Register Access Operation (Bus Master ↔ PWCR (Upper Eight Bits))

17.5 Operation

17.5.1 PWM_1 Operation

PWM waveforms are output from pins PWM1A to PWM1H and PWM2A to PWM2H as shown in figure 17.10.

Initial Settings: Set the PWM output polarity in PWPR; set the OEn bit in PWOCR to 1 to enable PWM output from the corresponding pin; select the clock to be input to PWCNT with the CKS2 to CKS0 bits in PWCR; set the PWM conversion cycle in PWCYR; and set the first frame of data in PWBFRA, PWBFRC, PWBFRE, and PWBFRG.

Activation: Setting the CST bit in PWCR to 1 starts counting by PWCNT. When a compare match between PWCNT and PWCYR occurs, data is transferred from the buffer register to the duty register and the CMF bit in PWCR is set to 1. If the IE bit in PWCR has been set to 1 at this time, an interrupt can be requested or the DTC can be activated.

Waveform Output: The PWM outputs selected by the OTS bits in PWDTRA, PWDTRC, PWDTRE, and PWDTRG go high when a compare match occurs between PWCNT and PWCYR. The PWM outputs not selected by the OTS bit are low. When a compare match occurs between PWCNT and PWDTRA, PWDTRC, PWDTRE, or PWDTRG, the corresponding PWM output goes low. If the corresponding bit in PWPR is set to 1, the output is inverted.

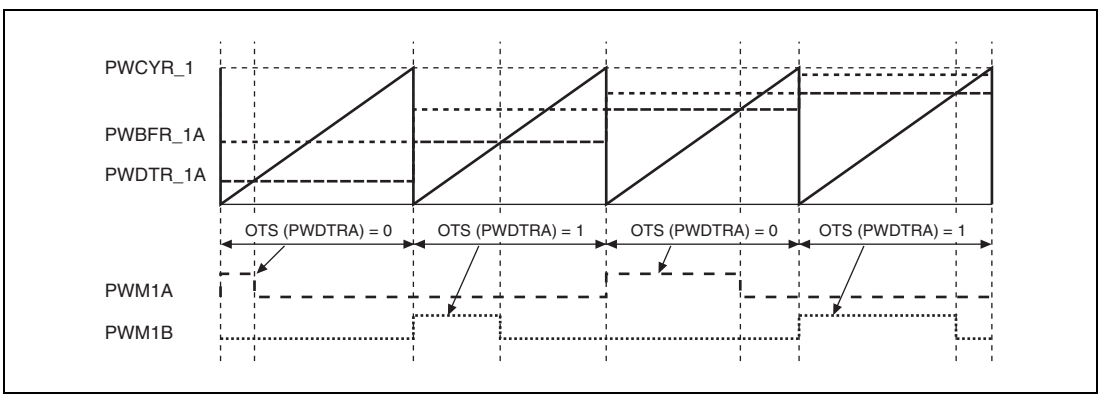


Figure 17.10 PWM_1 Operation

Next Frame: When a compare match occurs between PWCNT and PWCYR, data is transferred from the buffer register to the duty register. PWCNT is reset and starts counting up from H'000. The CMF bit in PWCR is set, and if the IE bit in PWCR1 or PWCR2 has been set, an interrupt can be requested or the DTC can be activated.

Stopping: When the CST bit in PWCR is cleared to 0, PWCNT is reset and stops. All PWM outputs go low (or high if the corresponding bit in PWPR is set to 1).

17.5.2 PWM_2 Operation

PWM waveforms are output from pins PWM2A to PWM2H as shown in Figure 17.11.

Initial Settings: Set the PWM output polarity in PWPR2; enable the pins for PWM output with PWOCR2; select the clock to be input to PWCNT2 with bits CKS2 to CKS0 in PWCR2; set the PWM conversion cycle in PWCYR2; and set the first frame of data in PWBFR2A, PWBFR2B, PWBFR2C, and PWBFR2D.

Activation: When the CST bit in PWCR2 is set to 1, a compare match between PWCNT2 and PWCYR2 is generated. Data is transferred from PWBFR2A to PWDTR2A or PWDTR2E, from PWBFR2B to PWDTR2B or PWDTR2F, from PWBFR2C to PWDTR2C or PWDTR2G, and from PWBFR2D to PWDTR2D or PWDTR2H, according to the value of the TDS bit. PWCNT2 starts counting up. At the same time the CMF bit in PWCR2 is set, so that, if the IE bit in PWCR2 has been set, an interrupt can be requested or the DTC can be activated.

Waveform Output: The PWM outputs go high when a compare match occurs between PWCNT2 and PWCYR2. When a compare match occurs between PWCNT2 and PWDTR2A to PWDTR2H, the corresponding PWM output goes low. If the corresponding bit in PWPR2 is set to 1, the output is inverted.

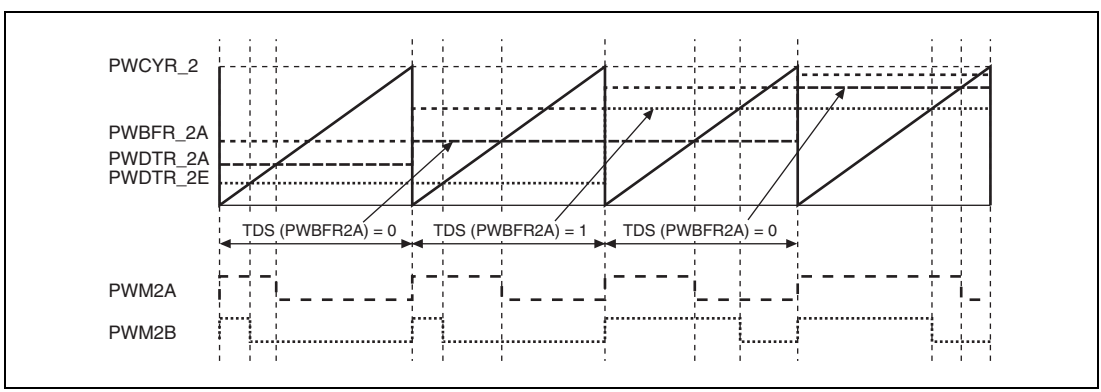


Figure 17.11 PWM_2 Operation

Next Frame: When a compare match occurs between PWCNT2 and PWCYR2, data is transferred from PWBFR2A to PWDTR2A or PWDTR2E, from PWBFR2B to PWDTR2B or PWDTR2F, from PWBFR2C to PWDTR2C or PWDTR2G, and from PWBFR2D to PWDTR2D or PWDTR2H, according to the value of the TDS bit. PWCNT2 is reset and starts counting up from

H'000. The CMF bit in PWCR2 is set, and if the IE bit in PWCR2 has been set, an interrupt can be requested or the DTC can be activated.

Stopping: When the CST bit in PWCR2 is cleared to 0, PWCNT2 is reset and stops. PWDTR2A to PWDTR2H are reset. All PWM outputs go low (or high if the corresponding bit in PWPR2 is set to 1).

17.6 Usage Note

17.6.1 Conflict between Buffer Register Write and Compare Match

If a PWBFR write is performed in the state immediately after a cycle register compare match, the buffer register and duty register are both modified. PWM output changed by the cycle register compare match is not changed by modification of the duty register due to conflict. This may result in unanticipated duty output.

Buffer register modification must be completed before automatic transfer by the DTC, exception handling due to a compare match interrupt, or the occurrence of a cycle register compare match on detection of the rise of CMF (compare match flag) in PWCR.

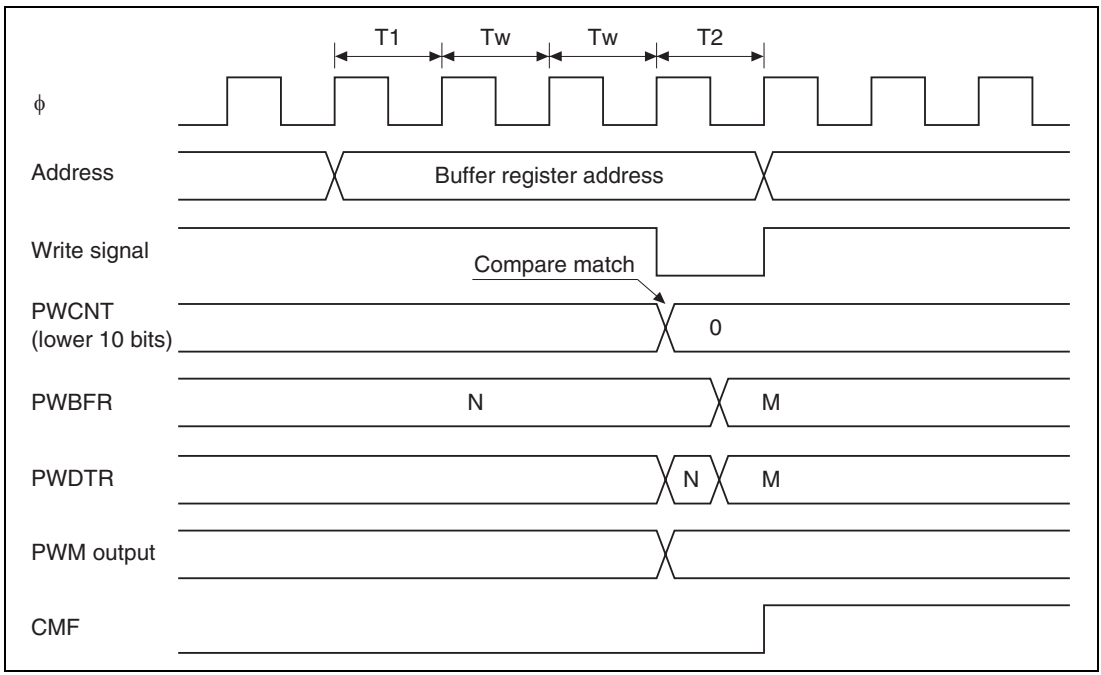


Figure 17.12 Conflict between Buffer Register Write and Compare Match

Section 18 RAM

This LSI has an 16-kbyte 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 19 ROM

The features of the flash memory are summarized below.

The block diagram of the flash memory is shown in figure 19.1.

19.1 Features

- Size: 384 kbytes
- Programming/erase methods

  The flash memory is programmed in 128-byte units at a time. Erase is performed in single-block units. The flash memory is configured as follows: 64 kbytes × 5 blocks, 32 kbytes × 1 block, and 4 kbytes × 8 blocks. To erase the entire flash memory, each block must be erased in turn.
- Reprogramming capability

  The flash memory can be reprogrammed for 100 times.
- Two on-board programming modes

  Boot mode

  User program mode

  On-board programming/erasing can be done in boot mode, in which the boot program built into the chip is started to erase or program the entire flash memory. In normal user program mode, individual blocks can be erased or programmed.
- Programmer mode

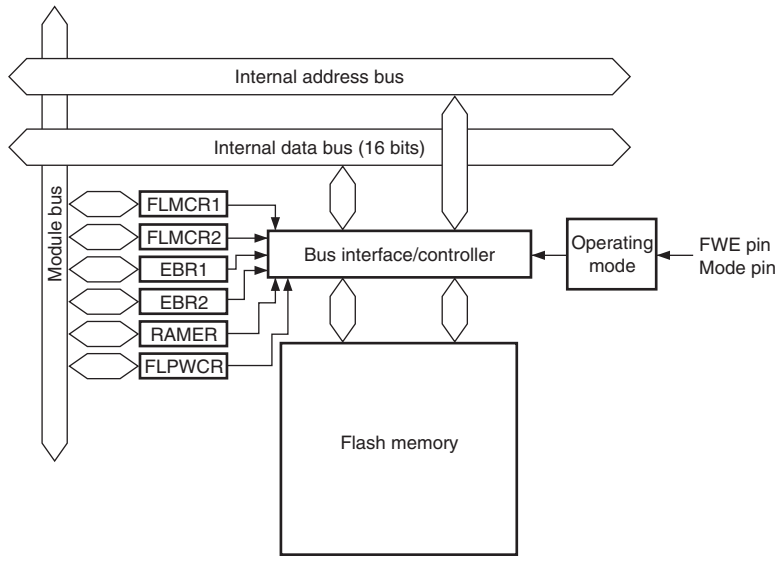
  Flash memory can be programmed/erased in programmer mode using a PROM programmer, as well as in on-board programming mode.
- Automatic bit rate adjustment

  For data transfer in boot mode, this LSI's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Programming/erasing protection

  There are three protect modes, hardware, software, and error protect, which allow protected status to be designated for flash memory program/erase operations.

- Emulation function for flash memory in RAM

  The real-time emulation for programming of flash memory is possible by overlapping the flash memory to a part of RAM.



[Legend]
FLMCR1: Flash memory control register 1
FLMCR2: Flash memory control register 2
EBR1: Erase block register 1
EBR2: Erase block register 2
RAMER: RAM emulation register
FLPWCR: Flash memory power control register

Figure 19.1 Block Diagram of Flash Memory

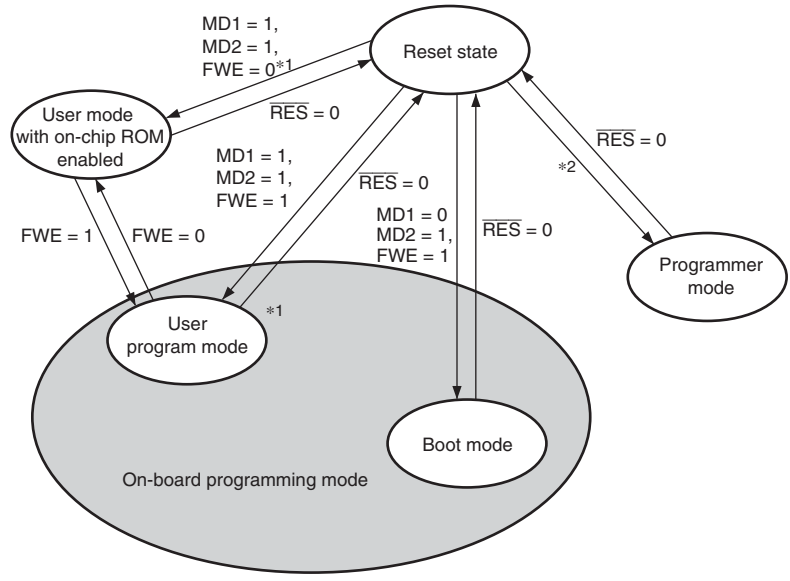
19.2 Mode Transitions

When the mode pins and the FWE pin are set in the reset state and a reset-start is executed, this LSI enters an operating mode as shown in figure 19.2. In user mode, flash memory can be read but not programmed or erased.

The boot, user program, and programmer modes are provided as modes to program and erase the flash memory.

The differences between boot mode and user program mode are shown in table 19.1.

Figure 19.3 shows the operation flow for boot mode and figure 19.4 shows that for user program mode.



Notes: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.

1. RAM emulation possible
2. MD0 = 0, MD1 = 0, MD2 = 0, FWE = 1, P14 = 0, P16 = 0, PF0 = 1

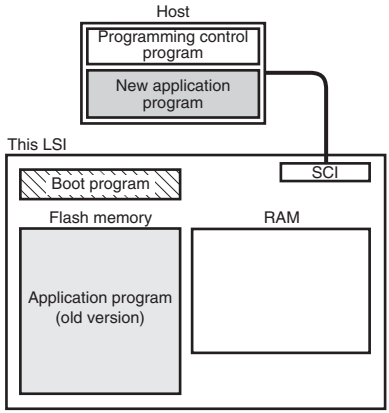
Figure 19.2 Flash Memory State Transitions

Table 19.1 Differences between Boot Mode and User Program Mode

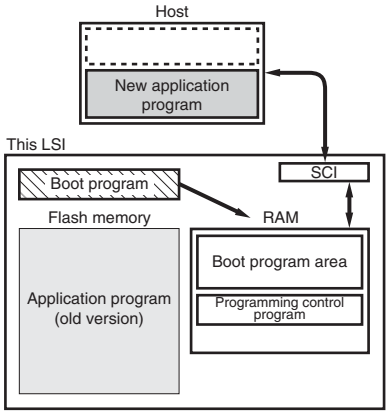
	Boot Mode	User Program Mode
Entire erase	Yes	Yes
Block erase	No	Yes
Programming control program*	Program/program-verify	Program/program-verify/erase/ erase-verify/emulation

Note: * To be provided by the user, in accordance with the recommended algorithm.

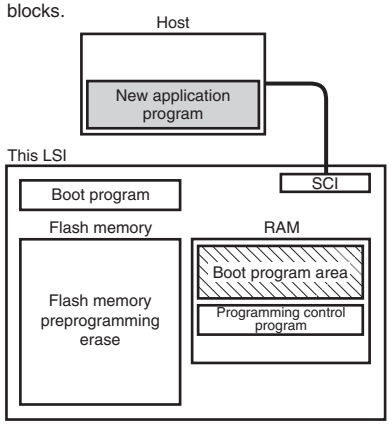
1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



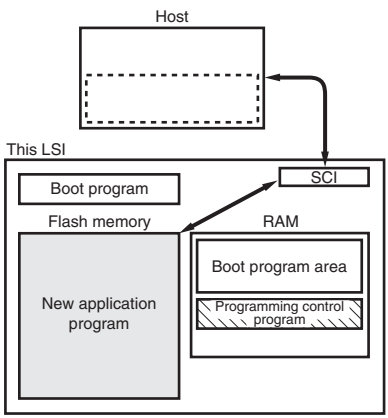
2. Programming control program transfer
   When boot mode is entered, the boot program in this LSI (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot program area.



3. Flash memory initialization
   The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, total flash memory erasure is performed, without regard to blocks.



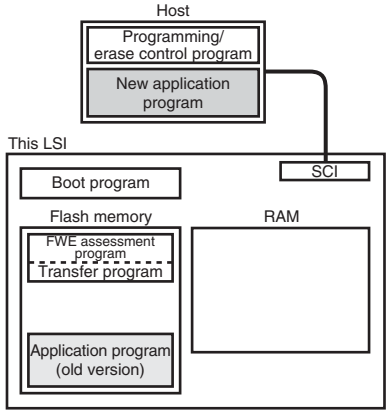
4. Writing new application program
   The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.



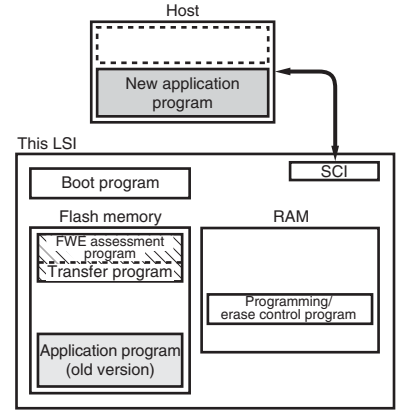
Program execution state

Figure 19.3 Boot Mode

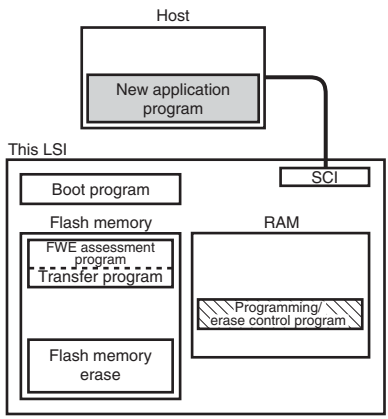
1. Initial state
   The FWE assessment program that confirms that user program mode has been entered, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control program should be prepared in the host or in the flash memory.



2. Programming/erase control program transfer
   When user program mode is entered, user software confirms this fact, executes transfer program in the flash memory, and transfers the programming/erase control program to RAM.



3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.



4. Writing new application program
   Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.

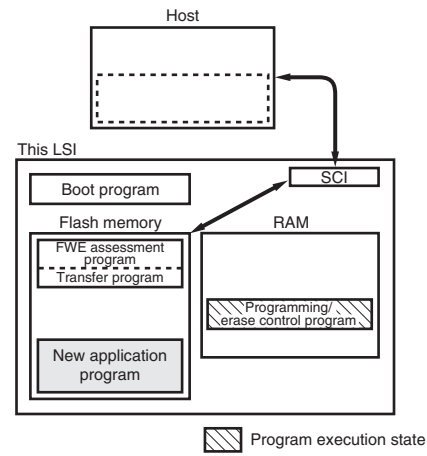


Figure 19.4 User Program Mode (Example)

19.3 Block Configuration

Figure 19.5 shows the block configuration of 384-kbyte flash memory. The thick lines indicate erasing units, the narrow lines indicate programming units, and the values are addresses. The flash memory is divided into 4 kbytes (8 blocks), 32 kbytes (1 block), and 64 kbytes (5 blocks). Erasing is performed in these units. Programming is performed in 128-byte units starting from an address with lower eight bits H'00 or H'80.

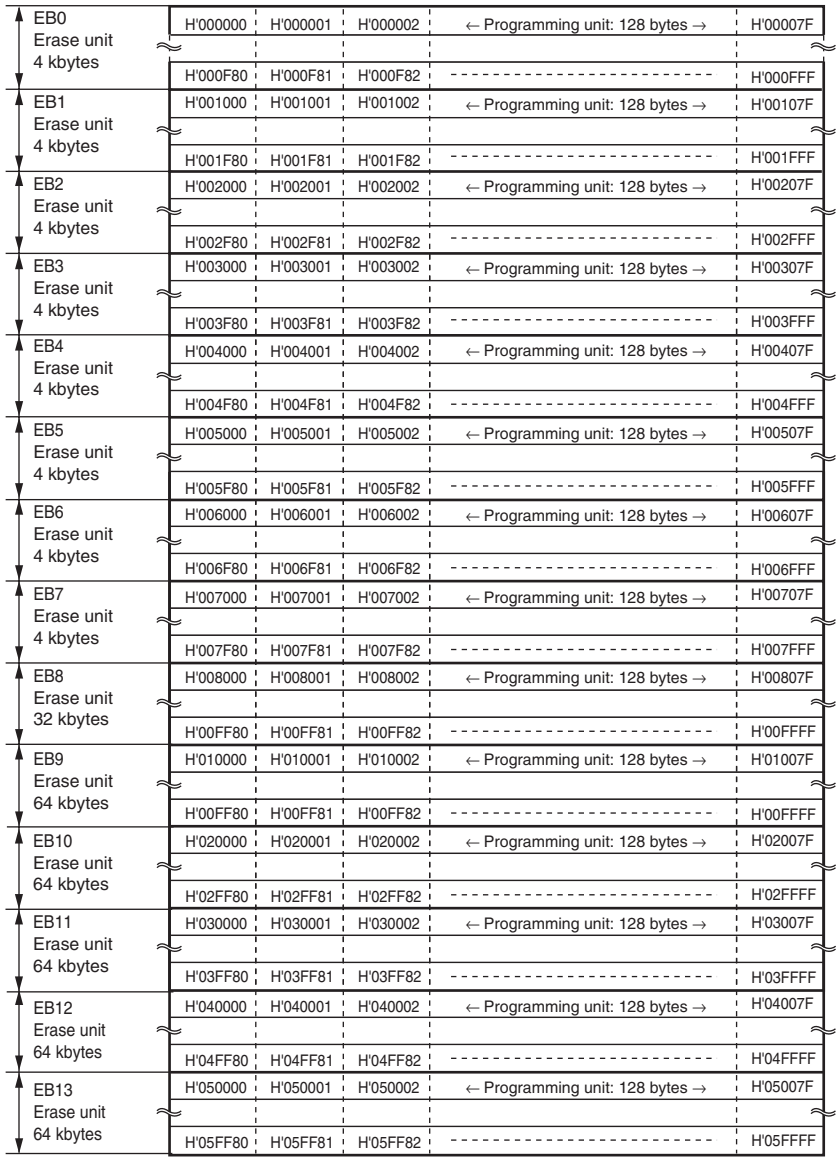


Figure 19.5 Flash Memory Block Configuration

19.4 Input/Output Pins

The flash memory is controlled by means of the pins shown in table 19.2.

Table 19.2 Pin Configuration

Pin Name	I/O	Function
$\overline{\text{RES}}$	Input	Reset
FWE	Input	Flash program/erase protection by hardware
MD2	Input	Sets this LSI's operating mode
MD1	Input	Sets this LSI's operating mode
MD0	Input	Sets this LSI's operating mode
PF0	Input	Sets MCU operating mode in programmer mode
P16	Input	Sets MCU operating mode in programmer mode
P14	Input	Sets MCU operating mode in programmer mode
TxD1	Output	Serial transmit data output
RxD1	Input	Serial receive data input

19.5 Register Descriptions

The flash memory has the following registers.

- Flash memory control register 1 (FLMCR1)
- Flash memory control register 2 (FLMCR2)
- Erase block register 1 (EBR1)
- Erase block register 2 (EBR2)
- RAM emulation register (RAMER)
- Flash memory power control register (FLPWCR)

19.5.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is a register that makes the flash memory change to program mode, program-verify mode, erase mode, or erase-verify mode. For details on register setting, see section 19.8, Flash Memory Programming/Erasing.

Bit	Bit Name	Initial Value	R/W	Description
7	FWE	—	R	Flash Write Enable Bit Reflects the input level at the FWE pin. It is set to 1 when a low level is input to the FWE pin, and cleared to 0 when a high level is input. When this bit is cleared to 0, the flash memory changes to hardware protect mode.
6	SWE	0	R/W	Software Write Enable Bit When this bit is set to 1 while FWE = 1, flash memory programming/erasing is enabled. When this bit is cleared to 0, bits 5 to 0 in FLMCR1 and all EBR1 and EBR2 bits cannot be set.
5	ESU	0	R/W	Erase Setup Bit When this bit is set to 1, the flash memory changes to the erase setup state. When it is cleared to 0, the erase setup state is cancelled. Set this bit to 1 before setting the E bit in FLMCR1.
4	PSU	0	R/W	Program Setup Bit When this bit is set to 1, the flash memory changes to the program setup state. When it is cleared to 0, the program setup state is cancelled. Set this bit to 1 before setting the P bit in FLMCR1.
3	EV	0	R/W	Erase-Verify When this bit is set to 1, the flash memory changes to erase-verify mode. When it is cleared to 0, erase-verify mode is cancelled.
2	PV	0	R/W	Program-Verify When this bit is set to 1, the flash memory changes to program-verify mode. When it is cleared to 0, program-verify mode is cancelled.

Bit	Bit Name	Initial Value	R/W	Description
1	E	0	R/W	Erase When this bit is set to 1, the flash memory changes to erase mode. When it is cleared to 0, erase mode is cancelled.
0	P	0	R/W	Program When this bit is set to 1, and while the SWE and PSU bits are 1, the flash memory changes to program mode. When it is cleared to 0, program mode is cancelled.

19.5.2 Flash Memory Control Register 2 (FLMCR2)

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

Bit	Bit Name	Initial Value	R/W	Description
7	FLER	0	R	Flash Memory Error Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state. See section 19.9.3, Error Protection, for details.
6 to 0	—	All 0	R	Reserved These bits are always read as 0.

19.5.3 Erase Block Register 1 (EBR1)

EBR1 specifies the flash memory erase area block. EBR1 is initialized to H'00 when the SWE bit in FLMCR1 is 0. Do not set more than one bit at a time, as this will cause all the bits in EBR1 and EBR2 to be automatically cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7	EB7	0	R/W	When this bit is set to 1, 4 kbytes of EB7 (H'007000 to H'007FFF) will be erased.
6	EB6	0	R/W	When this bit is set to 1, 4 kbytes of EB6 (H'006000 to H'006FFF) will be erased.
5	EB5	0	R/W	When this bit is set to 1, 4 kbytes of EB5 (H'005000 to H'005FFF) will be erased.
4	EB4	0	R/W	When this bit is set to 1, 4 kbytes of EB4 (H'004000 to H'004FFF) will be erased.
3	EB3	0	R/W	When this bit is set to 1, 4 kbytes of EB3 (H'003000 to H'003FFF) will be erased.
2	EB2	0	R/W	When this bit is set to 1, 4 kbytes of EB2 (H'002000 to H'002FFF) will be erased.
1	EB1	0	R/W	When this bit is set to 1, 4 kbytes of EB1 (H'001000 to H'001FFF) will be erased.
0	EB0	0	R/W	When this bit is set to 1, 4 kbytes of EB0 (H'000000 to H'000FFF) will be erased.

19.5.4 Erase Block Register 2 (EBR2)

EBR2 specifies the flash memory erase area block. EBR2 is initialized to H'00 when the SWE bit in FLMCR1 is 0. Do not set more than one bit at a time, as this will cause all the bits in EBR1 and EBR2 to be automatically cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved
5	EB13	0	R/W	When this bit is set to 1, 64 kbytes of EB13 (H'050000 to H'05FFFF) will be erased.
4	EB12	0	R/W	When this bit is set to 1, 64 kbytes of EB12 (H'040000 to H'04FFFF) will be erased.
3	EB11	0	R/W	When this bit is set to 1, 64 kbytes of EB11 (H'030000 to H'03FFFF) will be erased.
2	EB10	0	R/W	When this bit is set to 1, 64 kbytes of EB10 (H'020000 to H'02FFFF) will be erased.
1	EB9	0	R/W	When this bit is set to 1, 64 kbytes of EB9 (H'010000 to H'01FFFF) will be erased.
0	EB8	0	R/W	When this bit is set to 1, 32 kbytes of EB8 (H'008000 to H'00FFFF) will be erased.

19.5.5 RAM Emulation Register (RAMER)

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER settings should be made in user mode or user program mode. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0.
5, 4	—	0	R/W	Reserved Always write 0 before read.
3	RAMS	0	R/W	RAM Select Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, the flash memory is overlapped with part of RAM, and all flash memory block are program/erase-protected.
2	RAM2	0	R/W	Flash Memory Area Selection
1	RAM1	0	R/W	When the RAMS bit is set to 1, one of the following flash memory areas is selected to overlap the RAM area. The areas correspond with 4-kbyte erase blocks.
0	RAM0	0	R/W	
				000: H'000000 to H'000FFF (EB0)
				001: H'001000 to H'001FFF (EB1)
				010: H'002000 to H'002FFF (EB2)
				011: H'003000 to H'003FFF (EB3)
				100: H'004000 to H'004FFF (EB4)
				101: H'005000 to H'005FFF (EB5)
				110: H'006000 to H'006FFF (EB6)
				111: H'007000 to H'007FFF (EB7)

19.5.6 Flash Memory Power Control Register (FLPWCR)

FLPWCR enables/disables transition to power-down modes for the flash memory when this LSI enters sub-active mode.

Bit	Bit Name	Initial Value	R/W	Description
7	PDWND	0	R/W	Power Down Disable Enables/disables transition to power-down modes for the flash memory when this LSI enters sub-active mode. 0: Transition to power-down modes for the flash memory enabled. 1: Transition to power-down modes for the flash memory disabled.
6 to 0	—	All 0	R	Reserved These bits are always read as 0.

19.6 On-Board Programming Modes

When pins are set to on-board programming mode, program/erase/verify operations can be performed on the on-chip flash memory. There are two on-board programming modes: boot mode and user program mode. The pin settings for transition to each of these modes are shown in table 19.3. For a diagram of the transitions to the various flash memory modes, see figure 19.2.

Table 19.3 Setting On-Board Programming Modes

Mode Setting		FWE	MD2	MD1	MD0
Boot mode	Expanded mode	1	0	1	0
	Single-chip mode		0	1	1
User program mode	Expanded mode	1	1	1	0
	Single-chip mode		1	1	1

19.6.1 Boot Mode

Table 19.4 shows the boot mode operations between reset end and branching to the programming control program.

1. When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. Prepare a programming control program in accordance with the description in section 19.8, Flash Memory Programming/Erasing.

   In boot mode, if any data has been programmed into the flash memory (if all data is not 1), all flash memory blocks are erased. Boot mode is for use when user program mode is unavailable, such as the first time on-board programming is performed, or if the program activated in user program mode is accidentally erased.
2. SCI_1 should be set to asynchronous mode, and the transfer format as follows: 8-bit data, 1 stop bit, and no parity.
3. When the boot program is initiated, the chip measures the low-level period of asynchronous SCI communication data (H'00) transmitted continuously from the host. The chip then calculates the bit rate of transmission from the host, and adjusts the SCI_1 bit rate to match that of the host. The reset should end with the RxD pin high. The RxD and TxD pins should be pulled up on the board if necessary. After the reset is complete, it takes approximately 100 states before the chip is ready to measure the low-level period.

4. After matching the bit rates, the chip transmits one H'00 byte to the host to indicate the completion of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the chip. If reception could not be performed normally, initiate boot mode again by a reset. Depending on the host's transfer bit rate and system clock frequency of this LSI, there will be a discrepancy between the bit rates of the host and the chip. To operate the SCI properly, set the host's transfer bit rate and system clock frequency of this LSI within the ranges listed in table 19.5.
5. In boot mode, a part of the on-chip RAM area is used by the boot program. The area H'FFE800 to H'FFEFBF is the area to which the programming control program is transferred from the host. The boot program area cannot be used until the execution state in boot mode switches to the programming control program.
6. Before branching to the programming control program, the chip terminates transfer operations by SCI_1 (by clearing the RE and TE bits in SCR to 0), however the adjusted bit rate value remains set in BRR. Therefore, the programming control program can still use it for transfer of write data or verify data with the host. The TxD pin is high. The contents of the CPU general registers are undefined immediately after branching to the programming control program. These registers must be initialized at the beginning of the programming control program, as the stack pointer (SP), in particular, is used implicitly in subroutine calls, etc.
7. Boot mode can be cleared by driving the reset pin low, waiting at least 20 states, then setting the FWE pin and mode pins, and executing reset release. Boot mode is also cleared when a WDT overflow occurs.
8. All interrupts are disabled during programming or erasing of the flash memory.

Table 19.4 Boot Mode Operation

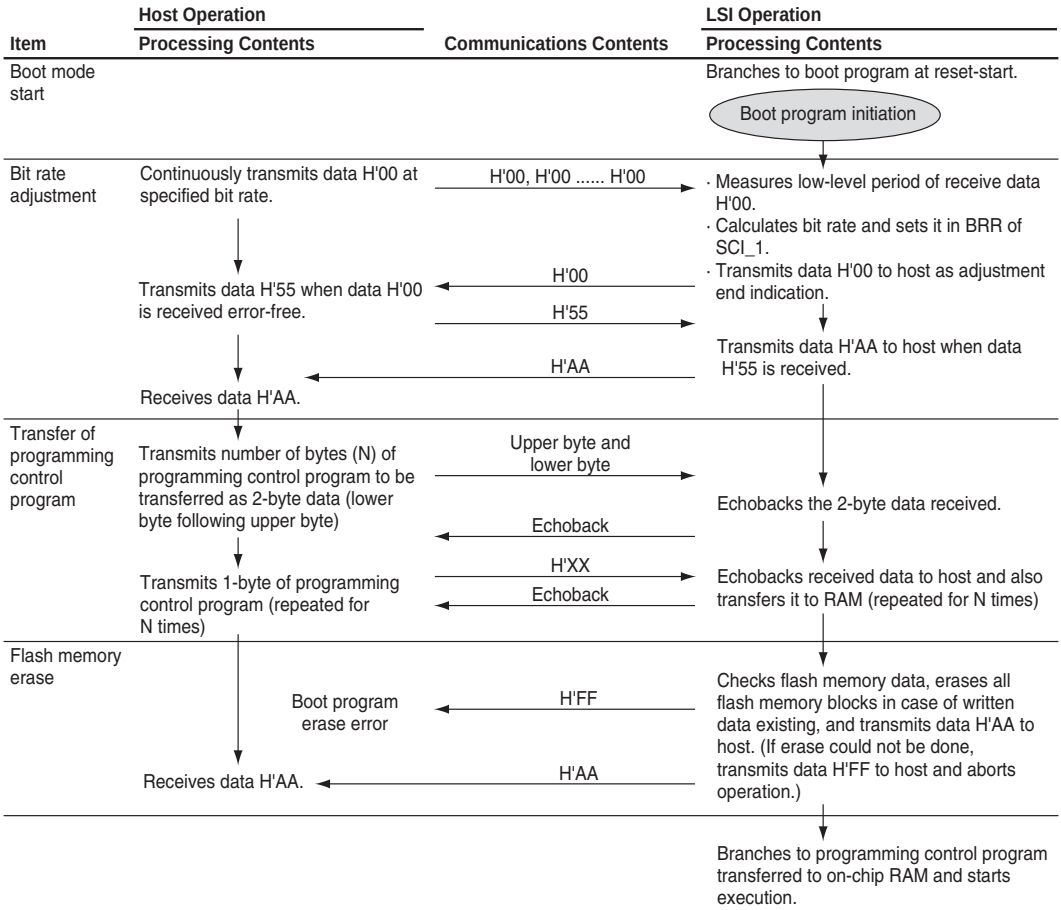


Table 19.5 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible

Host Bit Rate	System Clock Frequency Range of this LSI
19,200 bps	20 MHz
9,600 bps	8 to 20 MHz
4,800 bps	4 to 20 MHz

19.6.2 Programming/Erasing in User Program Mode

On-board programming/erasing of an individual flash memory block can also be performed in user program mode by branching to a user program/erase control program. The user must prepare on-board means for controlling FWE, on-board means of supplying programming data, and branching conditions. The flash memory must contain the user program/erase control program or a program that provides the user program/erase control program from external memory. As the flash memory itself cannot be read during programming/erasing, transfer the user program/erase control program to on-chip RAM, as in boot mode. Figure 19.6 shows a sample procedure for programming/erasing in user program mode. Prepare a user program/erase control program in accordance with the description in section 19.8, Flash Memory Programming/Erasing.

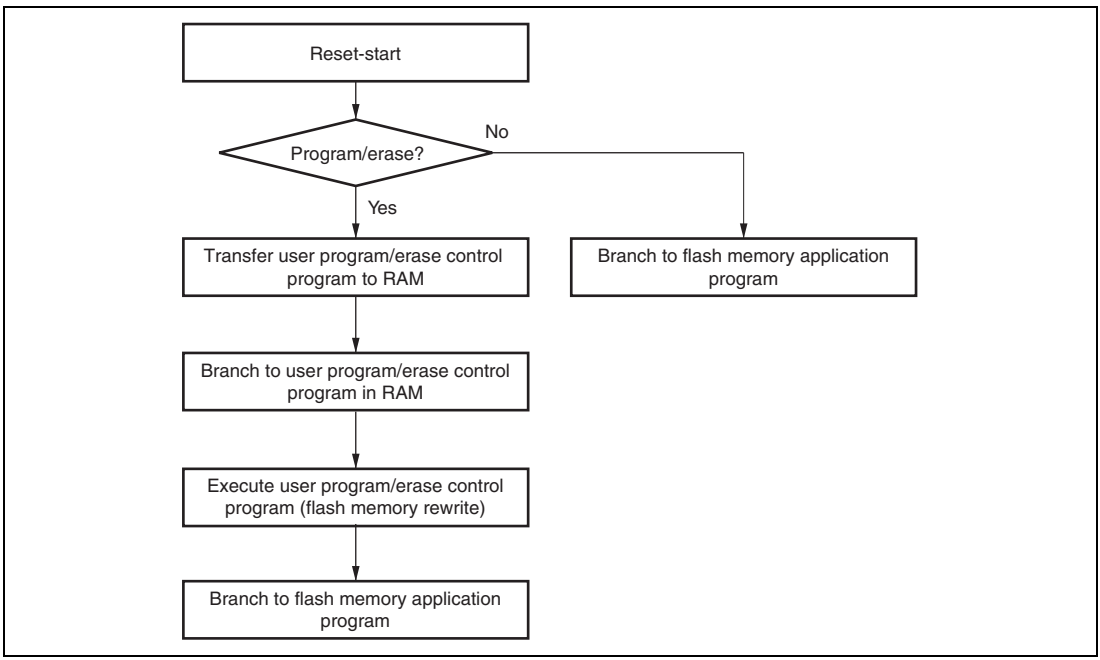


Figure 19.6 Programming/Erasing Flowchart Example in User Program Mode

19.7 Flash Memory Emulation in RAM

A setting in the RAM emulation register (RAMER) enables part of RAM to be overlapped onto the flash memory area so that data to be written to flash memory can be emulated in RAM in real time. Emulation can be performed in user mode or user program mode. Figure 19.7 shows an example of emulation of real-time flash memory programming.

1. Set RAMER to overlap part of RAM onto the area for which real-time programming is required.
2. Emulation is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, thus releasing the RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space.

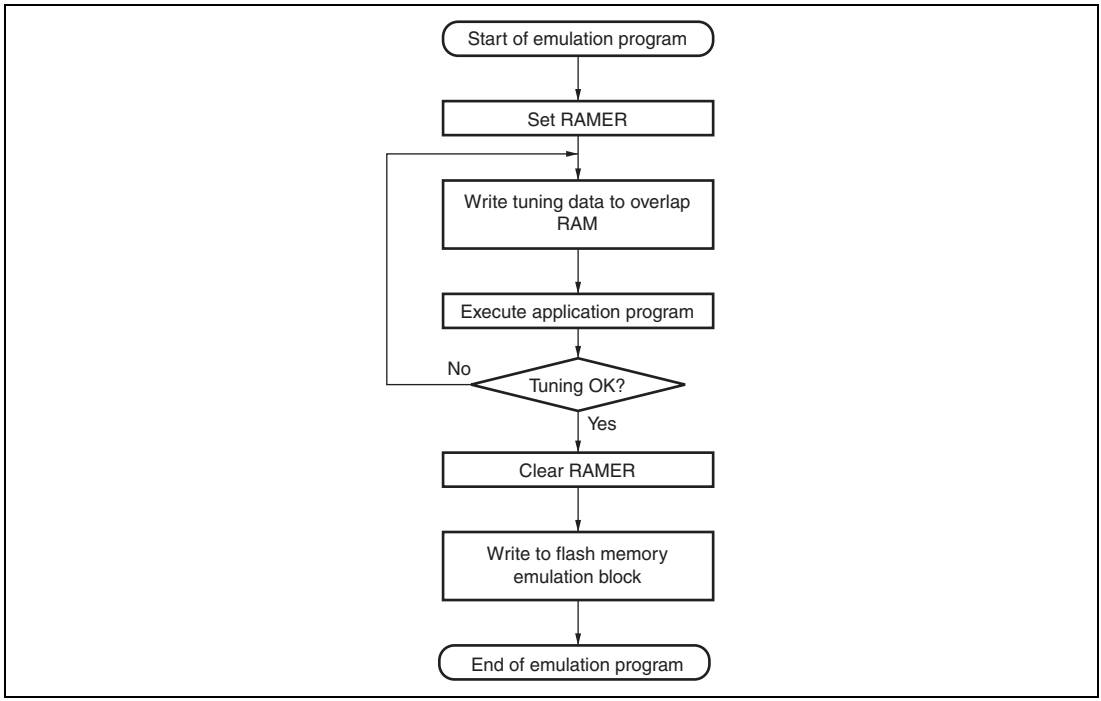


Figure 19.7 Flowchart for Flash Memory Emulation in RAM

An example in which flash memory block area EB0 is overlapped is shown in figure 19.8.

1. The RAM area to be overlapped is fixed at a 4-kbyte area in the range H'FFD800 to H'FFE7FF.
2. The flash memory area to be overlapped is selected by RAMER from a 4-kbyte area of the EB0 to EB7 blocks.
3. The overlapped RAM area can be accessed from both the flash memory addresses and RAM addresses.
4. When the RAMS bit in RAMER is set to 1, program/erase protection is enabled for all flash memory blocks (emulation protection). In this state, setting the P or E bit in FLMCR1 to 1 does not cause a transition to program mode or erase mode.
5. A RAM area cannot be erased by execution of software in accordance with the erase algorithm.
6. Block area EB0 contains the vector table. When performing RAM emulation, the vector table is needed in the overlap RAM.

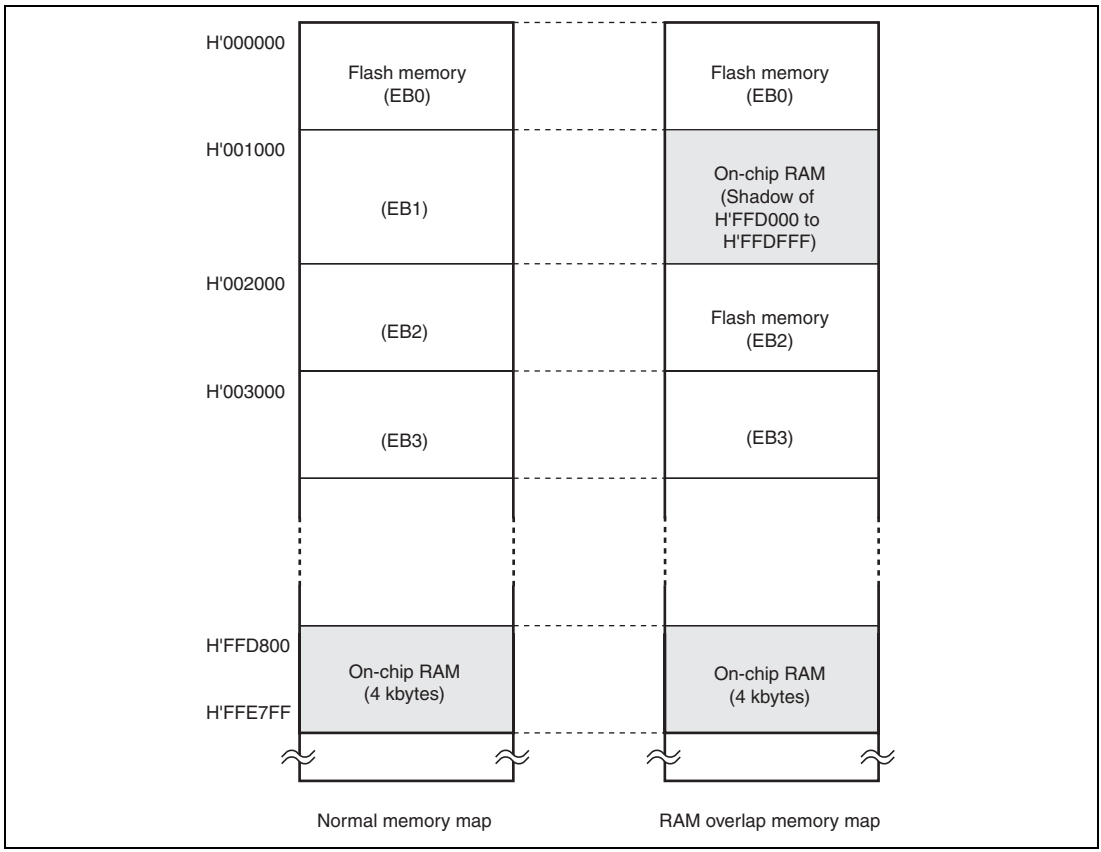


Figure 19.8 Example of RAM Overlap Operation

19.8 Flash Memory Programming/Erasing

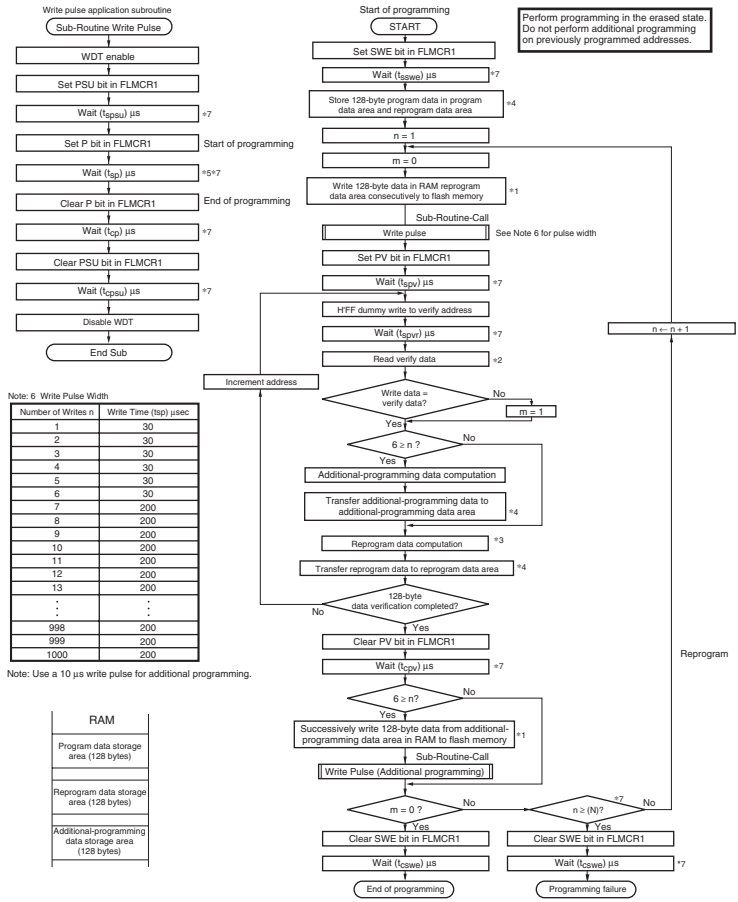
A software method using the CPU is employed to program and erase flash memory in the on-board programming modes. Depending on the FLMCR1 setting, the flash memory operates in one of the following four modes: Program mode, program-verify mode, erase mode, and erase-verify mode. The programming control program in boot mode and the user program/erase control program in user program mode use these operating modes in combination to perform programming/erasing. Flash memory programming and erasing should be performed in accordance with the descriptions in section 19.8.1, Program/Program-Verify and section 19.8.2, Erase/Erase-Verify, respectively.

19.8.1 Program/Program-Verify

When writing data or programs to the flash memory, the program/program-verify flowchart shown in figure 19.9 should be followed. Performing programming operations according to this flowchart will enable data or programs to be written to the flash memory without subjecting the chip to voltage stress or sacrificing program data reliability.

1. Programming must be done to an empty address. Do not reprogram an address to which programming has already been performed.
2. Programming should be carried out 128 bytes at a time. A 128-byte data transfer must be performed even if writing fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. Prepare the following data storage areas in RAM: A 128-byte programming data area, a 128-byte reprogramming data area, and a 128-byte additional-programming data area. Perform reprogramming data computation and additional programming data computation according to figure 19.10.
4. Consecutively transfer 128 bytes of data in byte units from the reprogramming data area or additional-programming data area to the flash memory. The program address and 128-byte data are latched in the flash memory. The lower 8 bits of the start address in the flash memory destination area must be H'00 or H'80.
5. The time during which the P bit is set to 1 is the programming time. Figure 19.10 shows the allowable programming times.
6. The watchdog timer (WDT) is set to prevent over-programming due to program runaway, etc. Set a value greater than (tspsu + tsp200 + tcp + tcpsu) μs as the WDT overflow period.
7. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower 2 bits are B'00. Verify data can be read in words from the address to which a dummy write was performed.

8. The maximum number of repetitions of the program/program-verify sequence of the same bit is (N).



Notes: 1. Data transfer is performed by byte transfer. The lower 8 bits of the first address written to must be H'00 or H'80.
A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.
2. Verify data is read in 16-bit (word) units.
3. Reprogram data is determined by the operation shown in the table below (comparison between the data stored in the program data area and the verify data). Bits for which the reprogram data is 0 are programmed in the next reprogramming loop. Therefore, even bits for which programming has been completed will be subjected to programming once again if the result of the subsequent verify operation is NG.
4. A 128-byte area for storing program data, a 128-byte area for storing reprogram data, and a 128-byte area for storing additional data must be provided in RAM. The contents of the reprogram data area and additional data area are modified as programming proceeds.
5. A write pulse of 30 µs or 200 µs is applied according to the progress of the programming operation. See Note 6 for details of the pulse widths. When writing of additional-programming data is executed, a 10 µs write pulse should be applied. Reprogram data X' means reprogram data when the write pulse is applied.
7. The wait times and value of N are shown in section 23.7, Flash Memory Characteristics.

Reprogram Data Computation Table

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming completed
0	1	0	Programming incomplete; reprogram
1	0	1	
1	1	1	Still in erased state; no action

Additional-Programming Data Computation Table

Reprogram Data (X')	Verify Data (V)	Additional-Programming Data (Y)	Comments
0	0	0	Additional programming to be executed
0	1	1	Additional programming not to be executed
1	0	1	Additional programming to be executed
1	1	1	Additional programming not to be executed

Figure 19.9 Program/Program-Verify Flowchart

19.8.2 Erase/Erase-Verify

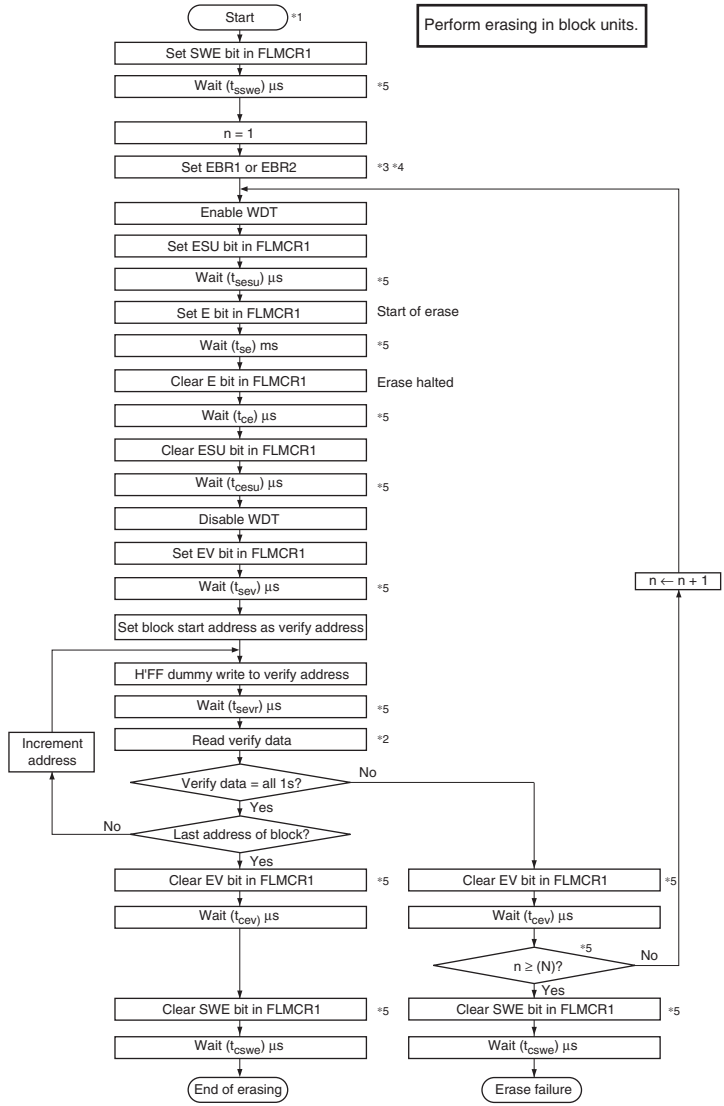
When erasing flash memory, the erase/erase-verify flowchart shown in figure 19.10 should be followed.

1. Prewriting (setting erase block data to all 0) is not necessary.
2. Erasing is performed in block units. Make only a single-bit specification in the erase block register 1 and 2 (EBR1 and EBR2). To erase multiple blocks, each block must be erased in turn.
3. The time during which the E bit is set to 1 is the flash memory erase time.
4. The watchdog timer (WDT) is set to prevent over-programming due to program runaway, etc. Set a value greater than ($t_{sesu} + t_{se} + t_{ce} + t_{cesu}$) ms as the WDT overflow period.
5. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower two bits are B'00. Verify data can be read in words from the address to which a dummy write was performed.
6. If the read data is not erased successfully, set erase mode again, and repeat the erase/erase-verify sequence as before. The maximum number of repetitions of the erase/erase-verify sequence is (N).

19.8.3 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including the $\overline{\text{NMI}}$ interrupt, are disabled while flash memory is being programmed or erased, or while the boot program is executing, for the following three reasons:

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



Notes: 1. Prewriting (setting erase block data to all 0s) is not necessary.
2. Verify data is read in 16-bit (word) units.
3. Make only a single-bit specification in the erase block registers (EBR1 and EBR2). Two or more bits must not be set simultaneously.
4. Erasing is performed in block units. To erase multiple blocks, each block must be erased in turn.
5. The wait times and the value of N are shown in section 23.7, Flash Memory Characteristics.

Figure 19.10 Erase/Erase-Verify Flowchart

19.9 Program/Erase Protection

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

19.9.1 Hardware Protection

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

19.9.2 Software Protection

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

19.9.3 Error Protection

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

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

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

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

19.10 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including NMI input, are disabled when flash memory is being programmed or erased (when the P or E bit is set in FLMCR1), and while the boot program is executing in boot mode*[1], to give priority to the program or erase operation. There are three reasons for this:

1. Interrupt during programming or erasing might cause a violation of the programming or erasing algorithm, with the result that normal operation could not be assured.
2. In the interrupt exception handling sequence during programming or erasing, the vector would not be read correctly*[2], possibly resulting in CPU runaway.
3. If an interrupt occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.

Notes: 1. Interrupt requests must be disabled inside and outside the CPU until the programming control program has completed programming.

2. The vector may not be read correctly in this case for the following two reasons:
   - If flash memory is read while being programmed or erased (while the P or E bit is set in FLMCR1), correct read data will not be obtained (undetermined values will be returned).
   - If the interrupt entry in the vector table has not been programmed yet, interrupt exception handling will not be executed correctly.

19.11 Programmer Mode

In programmer mode, a PROM programmer can be used to perform programming/erasing via a socket adapter, just as for a discrete flash memory. Use a PROM programmer that supports the Renesas 256-kbyte flash memory on-chip microcomputer device type (FZTAT256V5A).

19.12 Power-Down States for Flash Memory

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

- Normal operating mode
  The flash memory can be read and written to at high speed.
- Power-down state
  The flash memory can be read when part of the power circuit is halted and the LSI operates by subclocks.
- Standby mode
  All flash memory circuits are halted.

Table 19.6 shows the correspondence between the operating modes of this LSI and the flash memory. When the flash memory returns to its normal operating state from standby mode, a period to stabilize the power supply circuits that were stopped is needed. When the flash memory returns to its normal operating state, bits STS2 to STS0 in SBYCR must be set to provide a wait time of at least 100 μs, even when the external clock is being used.

Table 19.6 Flash Memory Operating States

LSI Operating State	Flash Memory Operating State
Active mode	Normal operating mode
Sleep mode	Normal operating mode
Watch mode Standby mode	Standby mode
Sub-active mode Sub-sleep mode	PDWND = 0: Power-down mode (read only) PDWND = 1: Normal operating mode (read only)

19.13 Usage Notes

Use the specified voltages and timing for programming and erasing: Applied voltages in excess of the rating can permanently damage the device.

Only use the specified socket adapter. Failure to observe these points may result in damage to the device.

Powering on and off: Do not apply a high level to the FWE pin until VCC has stabilized. Also, drive the FWE pin low before turning off VCC.

When applying or disconnecting VCC power, fix the FWE pin low and place the flash memory in the hardware protection state.

The power-on and power-off timing requirements should also be satisfied in the event of a power failure and subsequent recovery.

FWE application/disconnection: FWE application should be carried out when MCU operation is in a stable condition. If MCU operation is not stable, fix the FWE pin low and set the protection state.

The following points must be observed concerning FWE application and disconnection to prevent unintentional programming or erasing of flash memory:

- Apply FWE when the VCC voltage has stabilized within its rated voltage range.
- In boot mode, apply and disconnect FWE during a reset.
- In user program mode, FWE can be switched between high and low level regardless of the reset state. FWE input can also be switched during execution of a program in flash memory.
- Do not apply FWE if program runaway has occurred.
- Disconnect FWE only when the SWE, ESU, PSU, EV, PV, P, and E bits in FLMCR1 are cleared.

  Make sure that the SWE, ESU, PSU, EV, PV, P, and E bits are not set by mistake when applying or disconnecting FWE.

Do not apply a constant high level to the FWE pin: Apply a high level to the FWE pin only when programming or erasing flash memory. A system configuration in which a high level is constantly applied to the FWE pin should be avoided. Also, while a high level is applied to the FWE pin, the watchdog timer should be activated to prevent over-programming or over-erasing due to program runaway, etc.

Use the recommended algorithm when programming and erasing flash memory: The recommended algorithm enables programming and erasing to be carried out without subjecting the device to voltage stress or sacrificing program data reliability. When setting the P or E bit in FLMCR1, the watchdog timer should be set beforehand as a precaution against program runaway, etc.

Do not set or clear the SWE bit during execution of a program in flash memory: Wait for at least 100 µs after clearing the SWE bit before executing a program or reading data in flash memory.

When the SWE bit is set, data in flash memory can be rewritten. Access flash memory only for verify operations (verification during programming/erasing). Also, do not clear the SWE bit during programming, erasing, or verifying. Similarly, when using the RAM emulation function while a high level is being input to the FWE pin, the SWE bit must be cleared before executing a program or reading data in flash memory.

However, the RAM area overlapping flash memory space can be read and written to regardless of whether the SWE bit is set or cleared.

Do not use interrupts while flash memory is being programmed or erased: All interrupt requests, including NMI, should be disabled during FWE application to give priority to program/erase operations.

Do not perform additional programming. Erase the memory before reprogramming: In on-board programming, perform only one programming operation on a 128-byte programming unit block. In programmer mode, too, perform only one programming operation on a 128-byte programming unit block. Programming should be carried out with the entire programming unit block erased.

Before programming, check that the chip is correctly mounted in the PROM programmer: Overcurrent damage to the device can result if the index marks on the PROM programmer socket, socket adapter, and chip are not correctly aligned.

Do not touch the socket adapter or chip during programming: Touching either of these can cause contact faults and write errors.

Reset the flash memory before turning on the power: To reset the flash memory during oscillation stabilization period, the reset signal must be input for at least 100 µs.

Apply the reset signal while SWE is low to reset the flash memory during its operation: The reset signal is applied at least 100 µs after the SWE bit has been cleared.

Note on Switching from F-ZTAT Version to Mask ROM Version: The mask ROM version does not have the internal registers for flash memory control that are provided in the F-ZTAT version. Table 19.7 lists the registers that are present in the F-ZTAT version but not in the mask ROM version. If a register listed in table 19.7 is read in the mask ROM version, an undefined value will be returned. Therefore, if application software developed on the F-ZTAT version is switched to a mask ROM version product, it must be modified to ensure that the registers in table 19.7 have no effect.

Table 19.7 Registers Present in F-ZTAT Version but Absent in Mask ROM Version

Register	Abbreviation	Address
Flash memory control register 1	FLMCR1	H'FFA8
Flash memory control register 2	FLMCR2	H'FFA9
Erase block register 1	EBR1	H'FFAA
RAM emulation register	RAMER	FFAB
Flash memory power control register	FLPWCR	FEDB

Section 20 Clock Pulse Generator

This LSI has an on-chip clock pulse generator that generates the system clock (φ), the bus master clock, and internal clocks. The clock pulse generator consists of an oscillator, PLL circuit, clock selection circuit, medium-speed clock divider, and bus master clock selection circuit. A block diagram of the clock pulse generator is shown in figure 20.1.

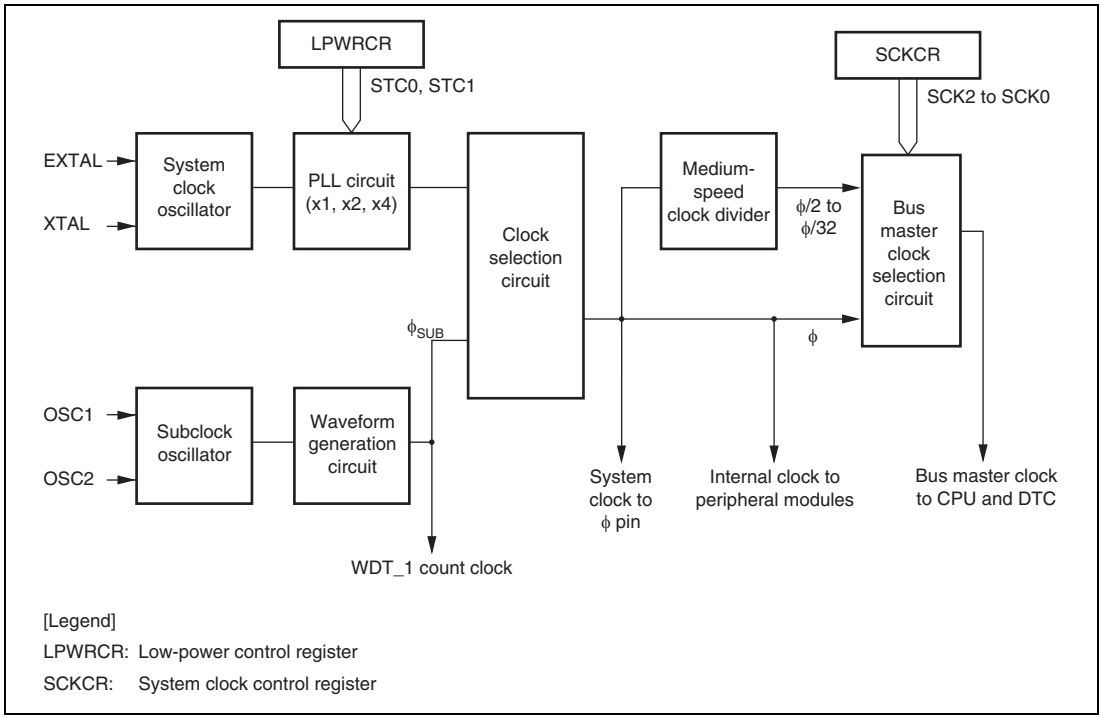


Figure 20.1 Block Diagram of Clock Pulse Generator

The frequency can be changed by means of the PLL circuit. Frequency changes are performed by software by settings in the low-power control register (LPWRCR) and system clock control register (SCKCR).

20.1 Register Descriptions

The on-chip clock pulse generator has the following registers. For details on LPWRCR, see section 21.1.2, Low-Power Control Register (LPWRCR).

- System clock control register (SCKCR)
- Low-power control register (LPWRCR)

20.1.1 System Clock Control Register (SCKCR)

SCKCR performs φ clock output control, selection of operation when the PLL circuit frequency multiplication factor is changed, and medium-speed mode control.

Bit	Bit Name	Initial Value	R/W	Description
7	PSTOP	0	R/W	φ Clock Output Disable Controls φ output. High-speed Mode, Medium-Speed Mode 0: φ output 1: Fixed high Sleep Mode 0: φ output 1: Fixed high Software Standby Mode 0: Fixed high 1: Fixed high Hardware Standby Mode 0: High impedance 1: High impedance
6 to 4	—	All 0	—	Reserved These bits are always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
3	STCS	0	R/W	Frequency Multiplication Factor Switching Mode Select Selects the operation when the PLL circuit frequency multiplication factor is changed. 0: Specified multiplication factor is valid after transition to software standby mode 1: Specified multiplication factor is valid immediately after STC1 bit and STC0 bit are rewritten
2	SCK2	0	R/W	System Clock Select 0 to 2 These bits select the bus master clock. 000: High-speed mode 001: Medium-speed clock is ϕ/2 010: Medium-speed clock is ϕ/4 011: Medium-speed clock is ϕ/8 100: Medium-speed clock is ϕ/16 101: Medium-speed clock is ϕ/32 11X: Setting prohibited
1	SCK1	0	R/W	
0	SCK0	0	R/W	

[Legend]
X: Don't care

20.2 Oscillator

Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock. In either case, the input clock should not exceed 20 MHz.

20.2.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as shown in the example in figure 20.2. Select the damping resistance R_d according to table 20.1. An AT-cut parallel-resonance crystal should be used.

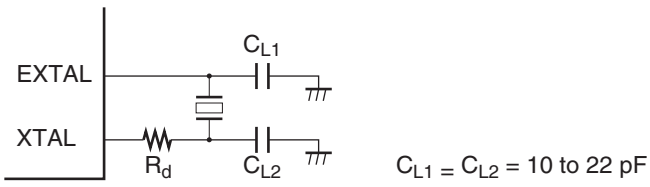


Figure 20.2 Connection of Crystal Resonator (Example)

Table 20.1 Damping Resistance Value

Frequency (MHz)	4	8	10	12	16	20
R_d (Ω)	500	200	0	0	0	0

Figure 20.3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 20.2.

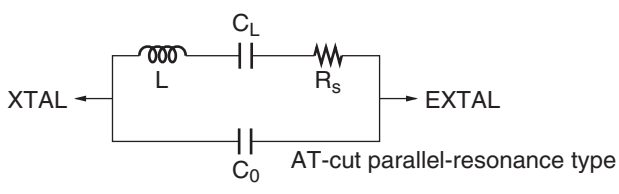


Figure 20.3 Crystal Resonator Equivalent Circuit

Table 20.2 Crystal Resonator Characteristics

Frequency (MHz)	4	8	10	12	16	20
R_S max (Ω)	120	80	70	60	50	40
C_0 max (pF)	7	7	7	7	7	7

20.2.2 External Clock Input

Circuit Configuration: An external clock signal can be input as shown in the examples in figure 20.4. If the XTAL pin is left open, ensure that stray capacitance does not exceed 10 pF. When complementary clock is input to the XTAL pin, the external clock input should be fixed high in standby mode.

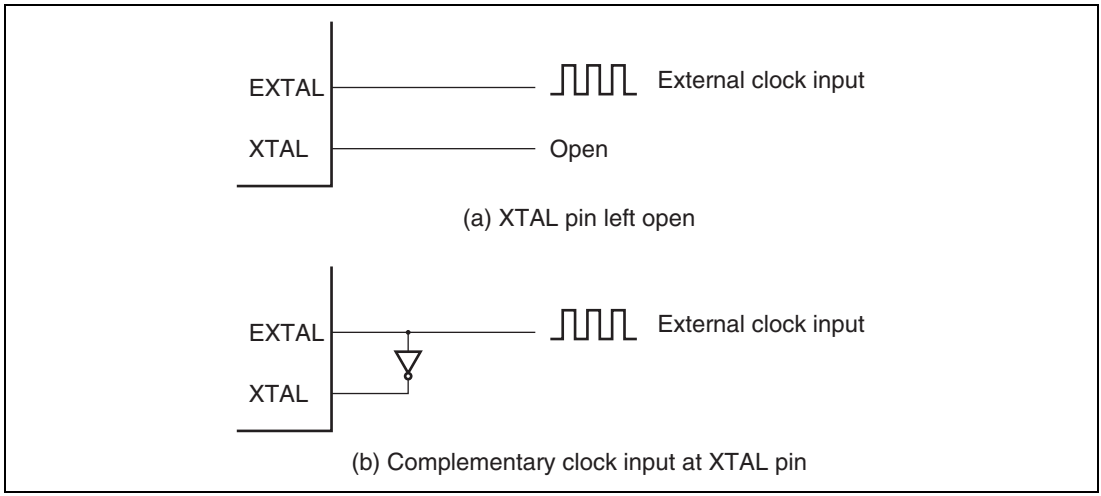


Figure 20.4 External Clock Input (Examples)

Table 20.3 shows the input conditions for the external clock.

Table 20.3 External Clock Input Conditions

		V_{cc} = 5.0 V ± 10%			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Test Conditions**
External clock input low pulse width	t_{EXL}	15	—	ns	Figure 20.5
External clock input high pulse width	t_{EXH}	15	—	ns	
External clock rise time	t_{EXr}	—	5	ns	
External clock fall time	t_{EXf}	—	5	ns	

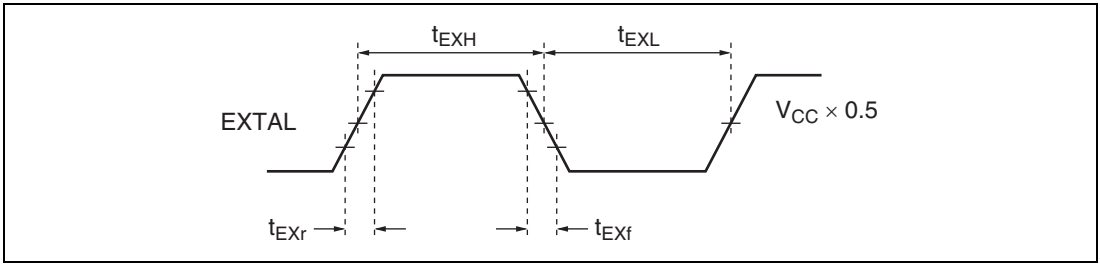


Figure 20.5 External Clock Input Timing

20.3 PLL Circuit

The PLL circuit multiplies the frequency of the clock from the oscillator by a factor of 1, 2, or 4. The multiplication factor is set by the STC0 bit and the STC1 bit in LPWRCR. The phase of the rising edge of the internal clock is controlled so as to match that at the EXTAL pin.

When the multiplication factor of the PLL circuit is changed, the operation varies according to the setting of the STCS bit in SCKCR.

When STCS = 0, the setting becomes valid after a transition to software standby mode. The transition time count is performed in accordance with the setting of bits STS0 to STS2 in SBYCR. For details on SBYCR, see section 21.1.1, Standby Control Register (SBYCR).

1. The initial PLL circuit multiplication factor is 1.
2. STS0 to STS2 are set to give the specified transition time.
3. The target value is set in STC0 and STC1, and a transition is made to software standby mode.
4. The clock pulse generator stops and the value set in STC0 and STC1 becomes valid.
5. Software standby mode is cleared, and a transition time is secured in accordance with the setting in STS0 to STS2.
6. After the set transition time has elapsed, this LSI resumes operation using the target multiplication factor.

If a PC break is set for the SLEEP instruction, software standby mode is entered and break exception handling is executed after the oscillation settling time. In this case, the instruction following the SLEEP instruction is executed after execution of the RTE instruction. When STCS = 1, this LSI operates on the changed multiplication factor immediately after bits STC0 and STC1 are rewritten.

20.4 Medium-Speed Clock Divider

The medium-speed clock divider divides the system clock to generate $\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, and $\phi/32$.

20.5 Bus Master Clock Selection Circuit

The bus master clock selection circuit selects the clock supplied to the bus master by setting the bits SCK 2 to 0 in SCKCR. The bus master clock can be selected from high-speed mode, or medium-speed clocks ($\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, $\phi/32$).

20.6 Subclock Oscillator

20.6.1 Connecting 32.768-kHz Crystal Resonator

To supply a clock to the subclock divider, connect a 32.768-kHz crystal resonator, as shown in figure 20.6. Concerning the note on connection, see section 20.8.2, Note on Board Design.

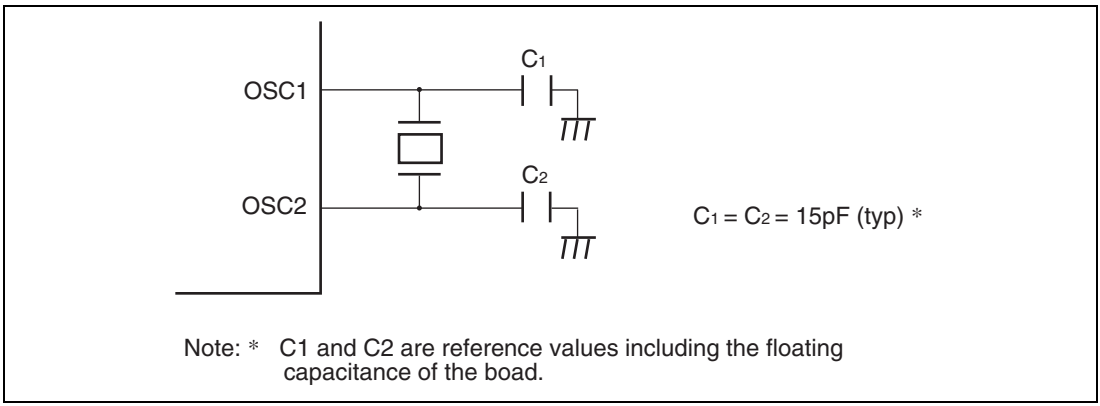


Figure 20.6 Connection Example of 32.768-kHz Crystal Resonator

Figure 20.7 shows the equivalent circuit for a 32.768-kHz crystal resonator.

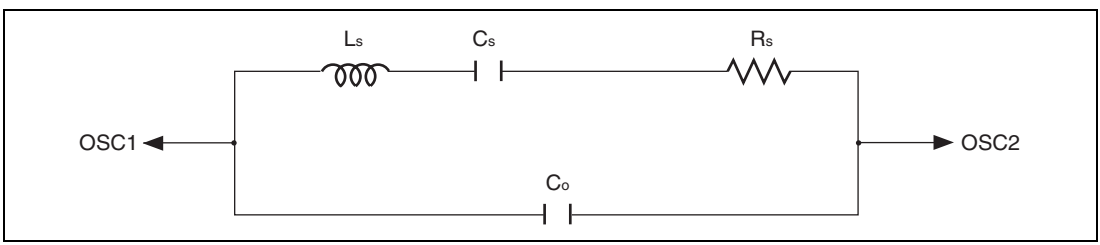


Figure 20.7 Equivalent Circuit for 32.768-kHz Crystal Resonator

20.6.2 Handling Pins when Subclock is not Used

If no subclock is required, connect the OSC1 pin to Vss and leave the OSC2 pin open, as shown in figure 20.8.

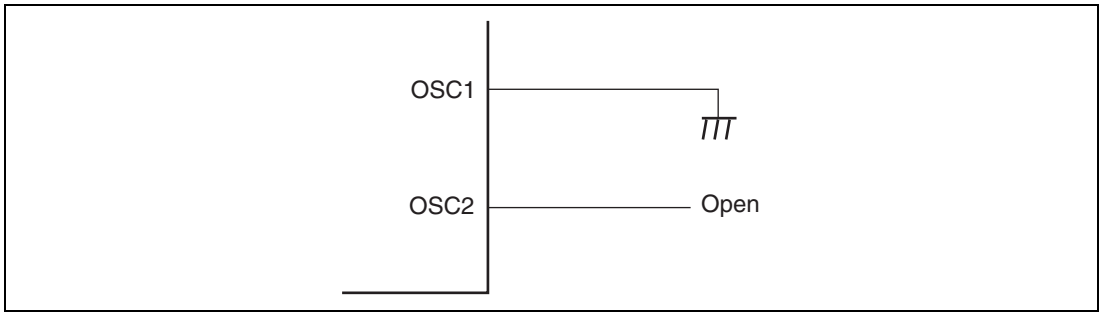


Figure 20.8 Pin Handling when Subclock is not Used

20.7 Subclock Waveform Generation Circuit

To eliminate noise from the subclock input from the OSC1 pin, the subclock is sampled using the dividing clock ϕ. The sampling frequency is set using the NESEL bit in LPWRCR. For details, see section 21.1.2, Low-Power Control Register (LPWRCR).

No sampling is performed in subactive, subsleep, and watch mode.

20.8 Usage Notes

20.8.1 Note on Crystal Resonator

As various characteristics related to the crystal resonator are closely linked to the user's board design, thorough evaluation is necessary on the user's part, using the resonator connection examples shown in this section as a guide. As the resonator circuit ratings will depend on the floating capacitance of the resonator and the mounting circuit, the ratings should be determined in consultation with the resonator manufacturer. The design must ensure that a voltage exceeding the maximum rating is not applied to the oscillator pin.

20.8.2 Note on Board Design

When designing the board, place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins. Other signal lines should be routed away from the oscillator circuit, as shown in figure 20.9. This is to prevent induction from interfering with correct oscillation.

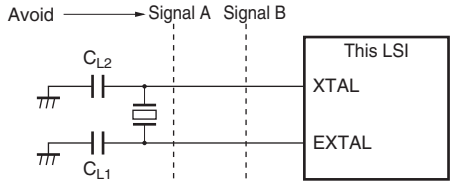
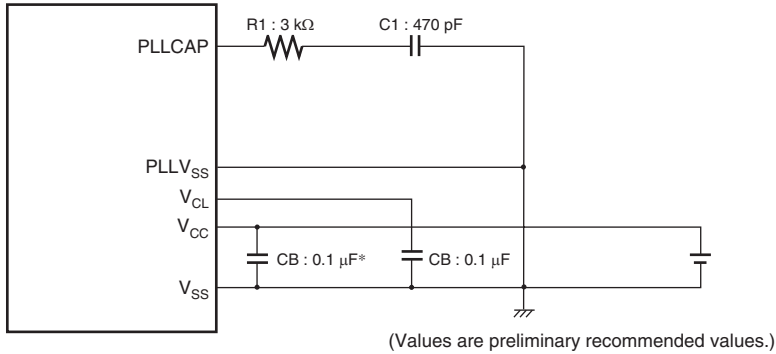


Figure 20.9 Note on Board Design of Oscillator Circuit

Figure 20.10 shows external circuitry recommended to be provided around the PLL circuit. Place oscillation settling capacitor C1 and resistor R1 close to the PLLCAP pin, and ensure that no other signal lines cross this line. Separate PLLVss from the other Vcc and Vss lines at the board power supply source, and be sure to insert bypass capacitors CB close to the pins.



Note: * CB is laminated ceramic.

Figure 20.10 External Circuitry Recommended for PLL Circuit

Section 21 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 as follows:

1. High-speed mode
2. Medium-speed mode
3. Subactive mode
4. Sleep mode
5. Subsleep mode
6. Watch mode
7. Module stop mode
8. Software standby mode
9. Hardware standby mode

Modes 2. to 9. are power-down modes. Sleep mode and subsleep mode are CPU states, medium-speed mode is a CPU and bus master state, subactive mode is a CPU, bus master, and on-chip peripheral function state, and module stop mode is an on-chip peripheral function (including bus masters other than the CPU) state. Some of these states can be combined.

After a reset, the LSI operates in high-speed mode with on-chip peripheral modules other than the DTC in module stop mode.

Table 21.1 shows the internal state of the LSI in each mode. Table 21.2 shows the conditions for shifting between the power-down modes. Figure 21.1 shows the mode transition diagram.

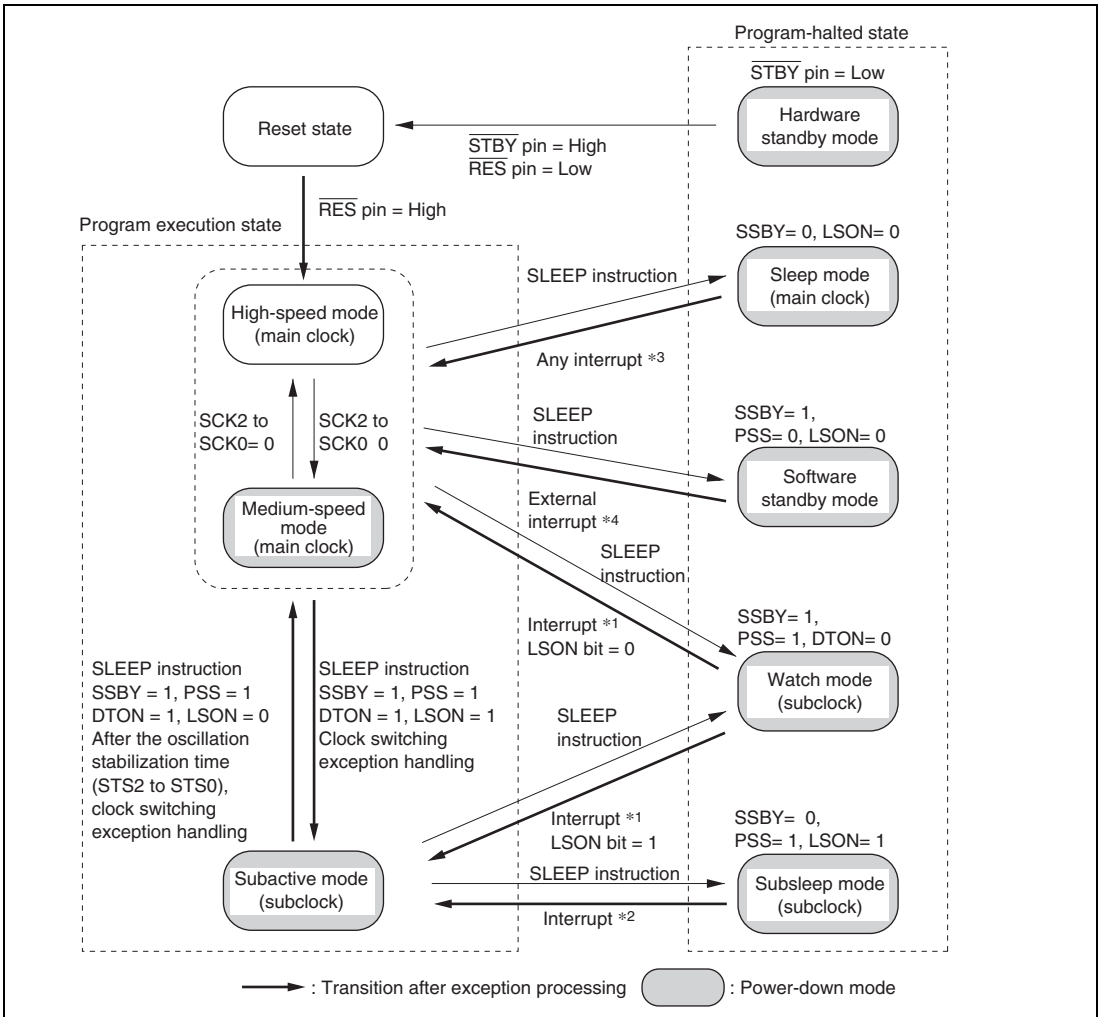
Table 21.1 LSI Internal States in Each Mode

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby
System clock pulse generator		Functioning	Functioning	Functioning	Functioning	Halted	Halted	Halted	Halted	Halted
Subclock generator		Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Halted	Halted
CPU	Instructions Registers	Functioning	Medium-speed operation	Halted (retained)	High/ medium-speed operation	Halted (retained)	Subclock operation	Halted (retained)	Halted (retained)	Halted (undefined)
External interrupts	NMI	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Halted
	IRQ0 to IRQ5									
Peripheral functions	WDT0	Functioning	Functioning	Functioning	—	Halted (retained)	Subclock operation	Subclock operation	Halted (retained)	Halted (reset)
	WDT1	Functioning	Functioning	Functioning	—	Subclock operation	Subclock operation	Subclock operation	Halted (retained)	Halted (reset)
	DTC	Functioning	Medium-speed operation	Functioning	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	TPU	Functioning	Functioning	Functioning	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	PPG									
	IIC_0									
	IIC_1									
	DA_1,0									
	PBC	Functioning	Functioning	Functioning	Halted (retained)	Halted (retained)	Subclock operation	Halted (retained)	Halted (retained)	Halted (reset)
	SCI0 to SCI2	Functioning	Functioning	Functioning	Halted (retained/ reset)	Halted (retained/ reset)	Halted (retained/ reset)	Halted (retained/ reset)	Halted (reset)	Halted (reset)
	SCI4									
	PWM	Functioning	Functioning	Functioning	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)
	A/D									
	RAM	Functioning	Functioning	Functioning (DTC)	Functioning	Retained	Functioning	Retained	Retained	Retained
	I/O	Functioning	Functioning	Functioning	Functioning	Retained	Functioning	Retained	Retained	High impedance

Notes: "Halted (retained)" means that internal register values are retained. The internal state is "operation suspended. "

"Halted (reset)" 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).



Notes: 1. NMI, IRQ0 to IRQ5, and WDT1 interrupts
2. NMI, IRQ0 to IRQ5, IWDT0 interrupts, and WDT1 interrupts.
3. All interrupts
4. NMI and IRQ0 to IRQ5

- When a transition is made between modes by means of an interrupt, the transition cannot be made on interrupt source generation alone. Ensure that interrupt handling is performed after accepting the interrupt request.
- From any state except hardware standby mode, a transition to the reset state occurs when $\overline{RES}$ is driven low.
- From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ is driven low.
- Always select high-speed mode before making a transition to watch mode or subactive mode.

Figure 21.1 Mode Transition Diagram

Table 21.2 Power-Down Mode Transition Conditions

Pre-Transition State	Status of Control Bit at Transition				State after Transition Invoked by SLEEP Instruction	State after Transition back from Power-Down Mode Invoked by Interrupt
	SSBY	PSS	LSON	DTON		
High-speed/ medium-speed	0	X	0	X	Sleep	High-speed/medium-speed
	0	X	1	X	—	—
	1	0	0	X	Software standby	High-speed/medium-speed
	1	0	1	X	—	—
	1	1	0	0	Watch	High-speed
	1	1	1	0	Watch	Subactive
	1	1	0	1	—	—
	1	1	1	1	Subactive	—
Subactive	0	0	X	X	—	—
	0	1	0	X	—	—
	0	1	1	X	Subsleep	Subactive
	1	0	X	X	—	—
	1	1	0	0	Watch	High-speed
	1	1	1	0	Watch	Subactive
	1	1	0	1	High-speed	—
	1	1	1	1	—	—

[Legend]

X: Don't care

— : Setting prohibited

21.1 Register Descriptions

Registers related to power-down modes are shown below. For details on SCKCR, see section 20.1.1, System Clock Control Register (SCKCR). For details on TCSR, see section 12.2.2, Timer Control/Status Register (TCSR).

- Standby control register (SBYCR)
- System clock control register (SCKCR)
- Low-power control register (LPWRCR)
- Timer control/status register (TCSR)
- Module stop control register A (MSTPCRA)
- Module stop control register B (MSTPCRB)
- Module stop control register C (MSTPCRC)
- Module stop control register D (MSTPCRD)

21.1.1 Standby Control Register (SBYCR)

SBYCR controls power-down modes.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby This bit in combination with other control bits specifies the operating mode after executing the SLEEP instruction. For details, see table 21.2. 0: Shifts to sleep mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode, or to subsleep mode when the SLEEP instruction is executed in subactive mode 1: Shifts to software standby mode, subactive mode, or watch mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode, or to watch mode or high-speed mode when the SLEEP instruction is executed in subactive mode This bit does not change when shifting between modes by using interrupts. 0 should be written to this bit to clear it.

Bit	Bit Name	Initial Value	R/W	Description
6	STS2	0	R/W	Standby Timer Select
5	STS1	0	R/W	These bits select the MCU standby time for clock stabilization when software standby mode, watch mode, or subactive mode is canceled by an interrupt or an instruction to shift to high-speed mode or medium-speed mode. With a crystal resonator (table 21.3), select a standby time of 8 ms (oscillation stabilization time) or more, depending on the operating frequency. With an external clock, any standby time can be selected. 000: Standby time = 8192 states 001: Standby time = 16384 states 010: Standby time = 32768 states 011: Standby time = 65536 states 100: Standby time = 131072 states 101: Standby time = 262144 states 110: Reserved 111: Standby time = 16 states
4	STS0	0	R/W	
3	OPE	1	R/W	Output Port Enable Specifies whether the outputs of the address bus and bus control signals ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$) are retained or set to high-impedance state in software standby mode or watch mode, or when making a direct transition.
2 to 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

- With Crystal Resonator

  The STS2 to STS0 bits must be specified appropriately so that the standby time (oscillation stabilization time) is at least 8 ms. Table 21.3 shows the standby time determined by a combination of the operating frequency and the STS2 to STS0 bit setting.

Table 21.3 Standby Time Settings

				Operating Frequency (MHz)							Unit
STS2	**STS1**	**STS0**	**Standby Time**	**20**	**16**	**12**	**10**	**8**	**6**	**4**	
0	0	0	8192 states	0.41	0.51	0.65	0.8	1.0	1.3	2.0	ms
0	0	1	16384 states	0.82	1.0	1.3	1.6	2.0	2.7	4.1	
0	1	0	32768 states	1.6	2.0	2.7	3.3	4.1	5.5	8.2	
0	1	1	65536 states	3.3	4.1	5.5	6.6	8.2	10.9	16.4	
1	0	0	131072 states	6.6	8.2	10.9	13.1	16.4	21.8	32.8	
1	0	1	262144 states	13.1	16.4	21.8	26.2	32.8	43.6	65.6	
1	1	0	Reserved	—	—	—	—	—	—	—	
1	1	1	16 states*	0.8	1.0	1.3	1.6	2.0	1.7	4.0	µs

Notes: : Recommended standby time

* Setting prohibited

- With External Clock

  The PLL circuit stabilization time must be ensured. Specify 2-ms or more standby time.

21.1.2 Low-Power Control Register (LPWRCR)

LPWRCR controls power-down modes.

Bit	Bit Name	Initial Value	R/W	Description
7	DTON	0	R/W	Direct Transition ON Flag This bit in combination with the SSBY bit in SBYCR, the LSON bit in LPWRCR, and the PSS bit in TCSR specifies the operating mode after executing the SLEEP instruction. For details, see table 21.2.
6	LSON	0	R/W	Low-Speed ON Flag This bit in combination with the SSBY bit in SBYCR, the DTON bit in LPWRCR, and the PSS bit in TCSR specifies the operating mode after executing the SLEEP instruction. For details, see table 21.2.
5	NESEL	0	R/W	Noise Elimination Sampling Frequency Select Selects the frequency of the clock (ϕ) generated by the system clock oscillator to be used to sample the subclock (ϕ_{SUB}) generated by the subclock oscillator. Set this bit to 0 when ϕ is 5 MHz or larger. 0: Sampled at ϕ/32 1: Sampled at ϕ/4
4	SUBSTP	0	R/W	Subclock Enable Enables or disables subclock generation 0: Enables subclock generation 1: Disables subclock generation
3	RFCUT	0	R/W	Oscillation Circuit Feedback Resistance Control Turns the internal feedback resistance of the main clock oscillator on or off. 0: When the main clock is oscillating, sets the feedback resistance on. When the main clock is stopped, sets the feedback resistance off. 1: Sets the feedback resistance off.

Bit	Bit Name	Initial Value	R/W	Description
2	—	0	R/W	Reserved The write value should always be 0.
1	STC1	0	R/W	Frequency Multiplication Factor
0	STC0	0	R/W	These bits specify the frequency multiplication factor of the PLL circuit. 00: ×1 01: ×2 10: ×4 11: Setting prohibited The peripheral clock frequency after the frequency multiplication should not be set beyond the maximum operating frequency of this LSI. Lowering the external clock peripheral frequency with this PLLx4, the current consumption and noise can be eliminated.

21.1.3 Module Stop Control Registers A to D (MSTPCRA to MSTPCRD)

MSTPCR controls module stop mode. Setting a bit to 1 causes the corresponding module to enter module stop mode. Clearing the bit to 0 clears the module stop mode.

- MSTPCRA

Bit	Bit Name	Initial Value	R/W	Module
7	MSTPA7*[1]	0	R/W	
6	MSTPA6	0	R/W	Data transfer controller (DTC)
5	MSTPA5	1	R/W	16-bit timer pulse unit (TPU)
4	MSTPA4*[2]	1	R/W	
3	MSTPA3	1	R/W	Programmable pulse generator (PPG)
2	MSTPA2	1	R/W	D/A converter
1	MSTPA1	1	R/W	A/D converter
0	MSTPA0*[2]	1	R/W	

- MSTPCRB

Bit	Bit Name	Initial Value	R/W	Module
7	MSTPB7	1	R/W	Serial communication interface_0 (SCI_0)
6	MSTPB6	1	R/W	Serial communication interface_1 (SCI_1)
5	MSTPB5	1	R/W	Serial communication interface_2 (SCI_2)
4	MSTPB4	1	R/W	I^2C bus interface_0 (IIC_0)
3	MSTPB3	1	R/W	I^2C bus interface_1 (IIC_1)
2	MSTPB2*[2]	1	R/W	
1	MSTPB1*[2]	1	R/W	
0	MSTPB0*[2]	1	R/W	

- MSTPCRC

Bit	Bit Name	Initial Value	R/W	Module
7	MSTPC7*[2]	1	R/W	
6	MSTPC6*[2]	1	R/W	
5	MSTPC5*[2]	1	R/W	
4	MSTPC4	1	R/W	PC break controller (PBC)
3	MSTPC3*[2]	1	R/W	
2	MSTPC2*[2]	1	R/W	
1	MSTPC1*[2]	1	R/W	
0	MSTPC0*[2]	1	R/W	

- MSTPCRD

Bit	Bit Name	Initial Value	R/W	Module
7	MSTPD7	1	R/W	Motor control PWM timer (PWM)
6	MSTPD6*2	1	R/W	
5	MSTPD5*3	Undefined	—	
4	MSTPD4*2	1	R/W	
3	MSTPD3*3	Undefined	—	
2	MSTPD2*2	1	R/W	
1	MSTPD1*3	Undefined	—	
0	MSTPD0*3	Undefined	—	

Notes: 1. Although this bit is readable/writable, only 0 should be written to.
2. Although this bit is readable/writable, only 1 should be written to.
3. This bit is read as undefined value and cannot be modified.

21.2 Medium-Speed Mode

When the SCK2 to SCK0 bits in SCKCR are set to 1 in high-speed mode, the operating mode changes to medium-speed mode as soon as the current bus cycle ends. In medium-speed mode, the CPU operates on the operating clock (ϕ/2, ϕ/4, ϕ/8, ϕ/16, or ϕ/32) specified by the SCK2 to SCK0 bits. The bus master (DTC) other than the CPU operates in medium-speed mode.

On-chip peripheral modules other than the bus masters always operate on the high-speed clock (ϕ).

In medium-speed mode, a bus access is executed in the specified number of states with respect to the bus master operating clock. For example, if ϕ/4 is selected as the operating clock, on-chip memory is accessed in four states, and internal I/O registers in eight states.

Medium-speed mode is canceled by clearing all of bits SCK2 to SCK0 to 0. A transition is made to high-speed mode and medium-speed mode is canceled at the end of the current bus cycle.

If the SLEEP instruction is executed when the SSBY bit in SBYCR = 0 and LSON bit in LPWRCR = 0, a transition is made to sleep mode. When sleep mode is canceled by an interrupt, medium-speed mode is restored.

When the SLEEP instruction is executed with the SSBY bit = 1, LSON bit in LPWRCR = 0, and PSS bit in TCSR (WDT1) = 0, operation shifts to software standby mode. When software standby mode is canceled by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is driven low and medium-speed mode is canceled, operation shifts to the reset state. The same applies to a reset caused by an overflow of the watchdog timer.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 21.2 shows the timing for transition to and clearance of medium-speed mode.

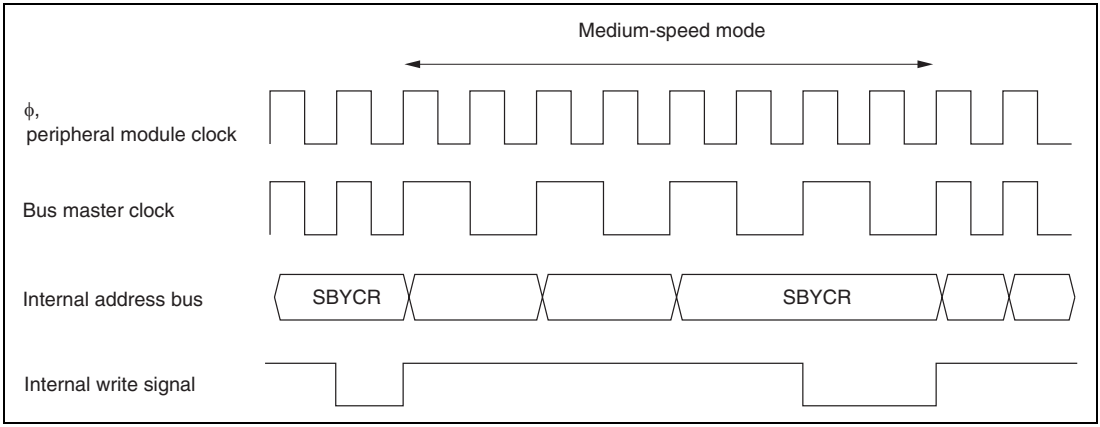


Figure 21.2 Medium-Speed Mode Transition and Clearance Timing

21.3 Sleep Mode

When the SLEEP instruction is executed when the SSBY bit in SBYCR = 0 and the LSON bit in LPWRCR = 0, the CPU enters the sleep mode. In sleep mode, CPU operation stops but the contents of the CPU's internal registers are retained. Other peripheral modules do not stop.

Sleep mode is canceled by any interrupt, or signals at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin.

When an interrupt occurs, sleep mode is canceled and interrupt exception handling starts. Sleep mode is not canceled if the interrupt is disabled, or if interrupts other than NMI are masked by the CPU.

When the $\overline{\text{RES}}$ pin is driven low, operation shifts to the reset state. After the stipulated reset input duration, driving the $\overline{\text{RES}}$ pin high makes the CPU start reset exception handling.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

21.4 Software Standby Mode

When the SLEEP instruction is executed when the SSBY bit in SBYCR = 1, the LSON bit in LPWRCR = 0, and the PSS bit in TCSR (WDT1) = 0, a transition is made to software standby mode. In this mode, the CPU, on-chip peripheral modules, and oscillator all stop. However, the contents of the CPU internal registers and on-chip RAM data, the states of on-chip peripheral modules other than the SCI, A/D converter, and motor control PWM, and the states of I/O ports are retained. The address bus and bus control signals are placed in the high-impedance state.

Software standby mode is canceled by an external interrupt (NMI and $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$ pins), or signals at the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

When an NMI or IRQ5 to IRQ0 interrupt request signal is input, clock oscillation starts, and after the time set in bits STS2 to STS0 in SBYCR has elapsed, stable clocks are supplied to the entire chip, software standby mode is canceled, and interrupt exception handling is started.

When canceling software standby mode with an IRQ5 to IRQ0 interrupt, set the corresponding enable bit to 1 and ensure that no interrupt with a higher priority than interrupts IRQ5 to IRQ0 is generated. Software standby mode cannot be canceled if the interrupt has been masked on the CPU side or has been designated as a DTC activation source.

When the $\overline{\text{RES}}$ pin is driven low, clock oscillation starts. At the same time as clock oscillation starts, the clock is supplied to the entire chip. Note that the $\overline{\text{RES}}$ pin must be held low until clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin is driven high, the CPU begins reset exception handling.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 21.3 shows an example in which a transition is made to software standby mode at a falling edge of the NMI pin, and software standby mode is canceled at a rising edge of the NMI pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in SYSCR cleared to 0 (falling edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY bit is set to 1, and a SLEEP instruction is executed, causing a transition to software standby mode.

Software standby mode is then canceled at the rising edge of the NMI pin.

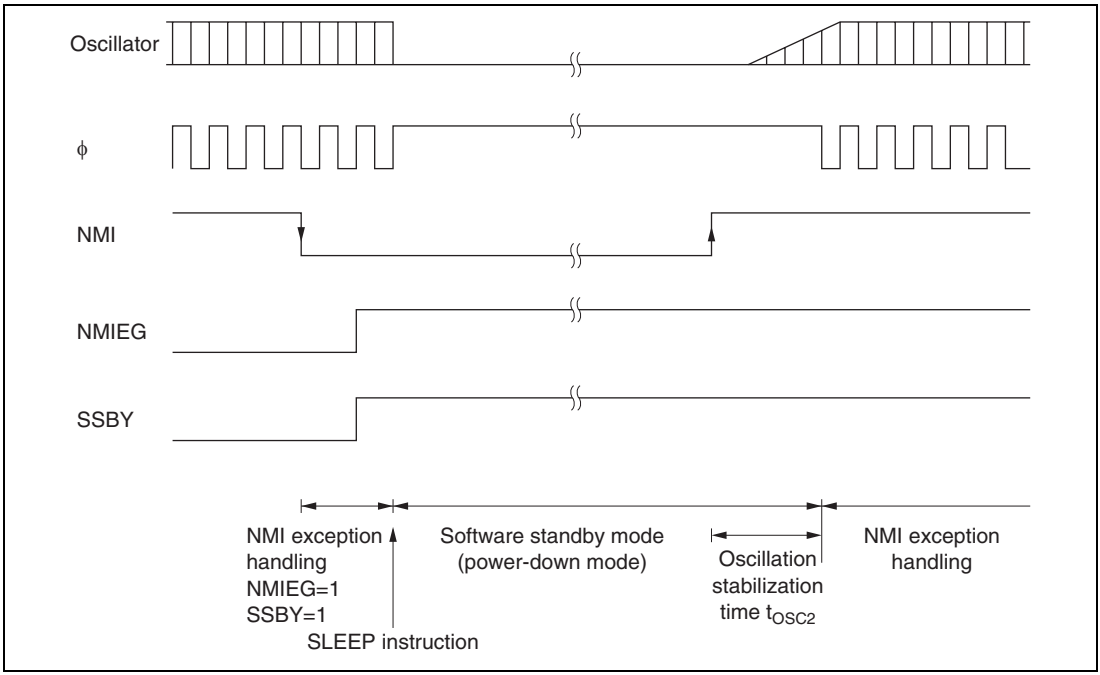


Figure 21.3 Software Standby Mode Application Example

21.5 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. The I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 before driving the $\overline{\text{STBY}}$ pin low.

Do not change the state of the mode pins (MD2 and MD0) while this LSI is in hardware standby mode.

Hardware standby mode is canceled by signals at 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 entered and clock oscillation is started. Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillator stabilizes (at least 8 ms (oscillation stabilization time) when using a crystal oscillator). 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.

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 canceled by driving the $\overline{\text{STBY}}$ pin high, waiting for the oscillation stabilization time, then switching the $\overline{\text{RES}}$ pin from low to high.

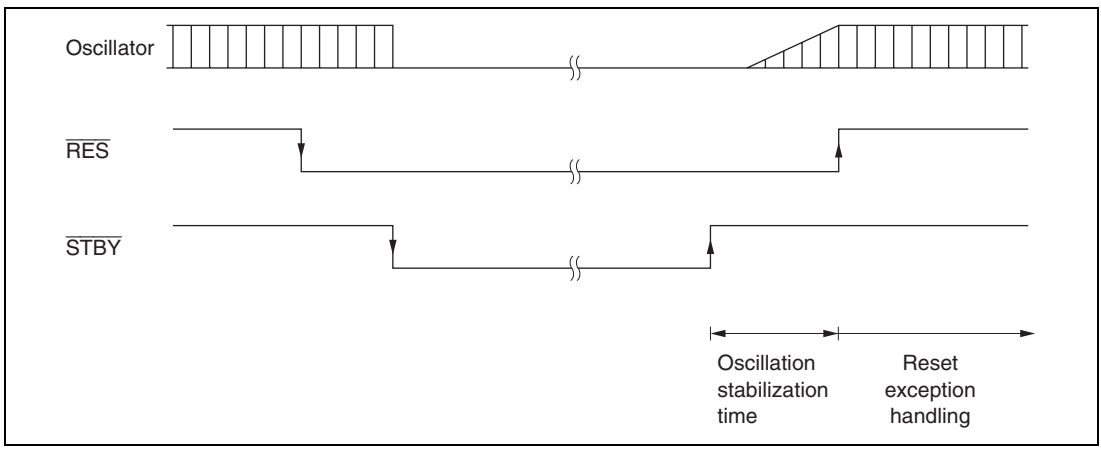


Figure 21.4 Hardware Standby Mode Timing

21.6 Watch Mode

When the SLEEP instruction is executed in high-speed mode or subactive mode with the SSBY bit in SBYCR = 1, the DTON bit in LPWRCR = 0, and the PSS bit in TCSR (WDT1) = 1, CPU operation shifts to watch mode.

In watch mode, the CPU stops and peripheral modules other than WDT1 also stop. The contents of the CPU internal registers and on-chip RAM data, the states of on-chip peripheral modules other than the SCI, A/D converter, and motor control PWM, and the states of I/O ports are retained.

Watch mode is canceled by any interrupt (WOVI1 interrupt, NMI pin, or $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$ pins), or signals at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin.

When an interrupt occurs, watch mode is canceled and a transition is made to high-speed mode or medium-speed mode when the LSON bit in LPWRCR = 0 or to subactive mode when the LSON bit = 1. When a transition is made to high-speed mode, a stable clock is supplied to the entire LSI and interrupt exception handling starts after the time set in the STS2 to STS0 bits of SBYCR has elapsed. For an IRQ5 to IRQ0 interrupt, watch mode is not canceled if the corresponding enable bit has been cleared to 0. For an interrupt from an on-chip peripheral module, if the interrupt enable register has been set to disable the reception of that interrupt or is masked by the CPU, watch mode is not canceled.

For the setting of the oscillation stabilization time when making a transition from watch mode to high-speed mode, see table 21.3.

For canceling watch mode by the $\overline{\text{RES}}$ pin, see section 21.4, Software Standby Mode.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

21.7 Subsleep Mode

When the SLEEP instruction is executed in subactive mode with the SSBY bit in SBYCR = 0, the LSON bit in LPWRCR = 1, and the PSS bit in TCSR (WDT1) = 1, CPU operation shifts to subsleep mode.

In subsleep mode, the CPU stops and peripheral modules other than WDT0 and WDT1 also stop. The contents of the CPU internal registers and on-chip RAM data, and the states of on-chip peripheral modules other than the SCI, A/D converter, and motor control PWM, and the states of I/O ports are retained.

Subsleep mode is canceled by any interrupt (interrupts from on-chip peripheral modules, NMI pin, or $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$ pins), or signals at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin.

When an interrupt occurs, subsleep mode is canceled and interrupt exception handling starts.

For an IRQ5 to IRQ0 interrupt, subsleep mode is not canceled if the corresponding enable bit has been cleared to 0. For an interrupt from an on-chip peripheral module, if the interrupt enable register has been set to disable the reception of that interrupt or is masked by the CPU, subsleep mode is not canceled.

For canceling subsleep mode by the $\overline{\text{RES}}$ pin, see section 21.4, Software Standby Mode.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

21.8 Subactive Mode

CPU operation shifts to subactive mode when the SLEEP instruction is executed in high-speed mode with the SSBY bit in SBYCR = 1, the DTON bit in LPWRCR = 1, the LSON bit = 1, and the PSS bit in TCSR (WDT1) = 1. When an interrupt occurs in watch mode, and if the LSON bit in LPWRCR is 1, a transition is made to subactive mode. If an interrupt occurs in subsleep mode, a transition is made to subactive mode.

In subactive mode, the CPU operates at low speed on the subclock, and the program is executed one after another. Peripheral modules other than WDT0 and WDT1 are also stopped.

When operating the CPU in subactive mode, the SCK2 to SCK0 bits in SCKCR must be set to 0.

Subactive mode is canceled by the SLEEP instruction or signals at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin.

When the SLEEP instruction is executed with the SSBY bit in SBYCR = 1, the DTON bit in LPWRCR = 0, and the PSS bit in TCSR (WDT1) = 1, subactive mode is canceled and a transition is made to watch mode. When the SLEEP instruction is executed with the SSBY bit in SBYCR = 0, the LSON bit in LPWRCR = 1, and the PSS bit in TCSR (WDT1) = 1, a transition is made to subsleep mode. When the SLEEP instruction is executed with the SSBY bit in SBYCR = 1, the DTON bit in LPWRCR = 1, the LSON bit = 0, and the PSS bit in TCSR (WDT1) = 1, a direct transition is made to high-speed mode (SCK0 to SCK2 are all 0).

For details on direct transitions, see section 21.10, Direct Transitions.

For canceling subactive mode by the RES pin, see section 21.4, Software Standby Mode.

When the STBY pin is driven low, a transition is made to hardware standby mode.

21.9 Module Stop Mode

Module stop mode can be set for individual on-chip peripheral modules.

When the corresponding MSTP bit in MSTPCR is set to 1, module operation stops at the end of the bus cycle and a transition is made to module stop mode. The CPU continues operating independently.

Table 21.4 shows MSTP bits and the corresponding on-chip supporting modules.

When the corresponding MSTP bit is cleared to 0, module stop mode is canceled and the module starts operating at the end of the bus cycle. In module stop mode, the internal states of modules other than the SCI*, motor control PWM, and A/D converter are retained.

After reset clearance, all modules other than DTC are in module stop mode.

When an on-chip peripheral module is in module stop mode, read/write access to its registers is disabled.

Note * Internal states in some part of registers in SCI are retained.

21.10 Direct Transitions

21.10.1 Overview of Direct Transitions

There are three modes, high-speed, medium-speed, and subactive, in which the CPU executes programs. When a direct transition is made, there is no interruption of program execution in shifting between high-speed and subactive modes. Direct transitions are enabled by setting the DTON bit in LPWRCR to 1, then executing the SLEEP instruction. After a transition, direct transition interrupt exception handling starts.

Direct Transition from High-Speed Mode to Subactive Mode: Execute the SLEEP instruction in high-speed mode with the SSBY bit in SBYCR = 1, the LSON bit in LPWRCR = 1, the DTON bit = 1, and the PSS bit in TCSR (WDT1) = 1, to make a direct transition to subactive mode.

Direct Transition from Subactive Mode to High-Speed Mode: Execute the SLEEP instruction in subactive mode with the SSBY bit in SBYCR = 1, the LSON bit in LPWRCR = 0, the DTON bit = 1, and the PSS bit in TCSR (WDT_1) = 1, to make a direct transition to high-speed mode after the time set in the STS2 to STS0 bits in SBYCR has elapsed.

21.11 φ Clock Output Control

The 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 21.4 shows the state of the φ pin in each processing state.

PLL circuit of this LSI is effective to eliminate the EMI* with lowering the resonator frequency or disabling the φ clock output. This should be well taken into consideration for the users' system board design

Note * EMI (Electromagnetic Interference)

Table 21.4 φ Pin State in Each Processing State

DDR	0	1	
PSTOP	–	0	1
Hardware Standby Mode	High impedance		
Software Standby Mode, Watch Mode, Direct Transitions	High impedance	Fixed high	
Sleep Mode, Subsleep Mode	High impedance	φ output	Fixed high
High-Speed Mode, Medium-Speed Mode,	High impedance	φ output	Fixed high
Subactive Mode	High impedance	φ sub output	Fixed high

21.12 Usage Notes

21.12.1 I/O Port Status

The status of the I/O ports is retained in software standby mode and watch mode. When the OPE bit is set to 1, the address bus and bus control signal outputs are also retained. Therefore, when a high level is output, the current consumption is not diminished by the amount of current to support the high level output.

21.12.2 Current Consumption during Oscillation Stabilization Wait Period

The current consumption increases during the oscillation stabilization wait period.

21.12.3 DTC Module Stop Setting

The MSTPA6 bits cannot be set to 1 depending on the DTC operating status. Module stop mode for the DTC must be specified while the DTC is stopped.

For details, see section 8, Data Transfer Controller (DTC).

21.12.4 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 activation source. Interrupts should therefore be disabled before entering module stop mode.

21.12.5 Writing to MSTPCR

MSTPCR should only be written to by the CPU.

21.12.6 Transition to Subactive Mode

1. When making a transition to subactive mode or watch mode, set the DTC to enter module stop mode (write 1 to the relevant bits in MSTPCR), and then read the relevant bits to confirm that they are set to 1 before mode transition. Do not clear module stop mode (write 0 to the relevant bits in MSTPCR) until a transition from subactive mode to high-speed mode or medium-speed mode has been performed.

   If a DTC activation source occurs in subactive mode, the DTC will be activated only after module stop mode has been cleared and high-speed mode or medium-speed mode has been entered.
2. The on-chip peripheral modules (DTC and TPU) which halt operation in subactive mode cannot clear an interrupt in subactive mode. Therefore, if a transition is made to subactive mode while an interrupt is requested, the CPU interrupt source cannot be cleared. Disable the interrupts of each on-chip peripheral module before executing a SLEEP instruction to enter subactive mode or watch mode.
3. A 1 is always returned when an attempt is made to read the pin status of I/O ports 1, 4, 9, or F during operation in subactive mode. (In the case of port 1, pins 13 to 10 are readable.) In addition, the ports may be used as output ports (except for ports 4 and 9). The procedure for determining the pin status during operation in subactive mode is as follows.

   [1] Use ports 3, A, B, C, D, E, H, and J as input ports.

   [2] Use external interrupt inputs (IRQ0 to IRQ5). (If the level sense setting has been selected for the IRQ pins, an interrupt request is generated by a low-level input.)
4. Operation cannot be guaranteed if a transition is made to the subactive mode, subsleep mode, or watch mode when the SUBSTP bit in LPWRCR is set to 1 (subclock generation prohibited). To prevent problems, it should be confirmed that the SUBSTP bit has been cleared to 0 before transitioning to the subactive mode, subsleep mode, or watch mode.

Section 22 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 in the order of ascending addresses.
- For 16-bit registers, the addresses of MSB are shown.
- Registers are classified according to functional modules.
- The access size is indicated.

2. Register bits

- Bit configurations of the registers are listed in the same order as the register addresses.
- Reserved bits are indicated by "—" in the bit name columns.
- Registers for which bit numbers are shown are those operate as counters or hold data.
- For 16-bit registers, bits in MSB are shown in the upper line and bits in LSB in the lower line.

3. Register states in each operating mode

- Register states are listed in the same order as the register addresses.
- The register states shown here are for the basic operating modes. If an on-chip module has its own reset state, refer to the section on that on-chip module.

22.1 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
PWM control register_1	PWCR_1	8	H'FC00	PWM_1	16	4
PWM output control register_1	PWOCR_1	8	H'FC02	PWM_1	16	4
PWM polarity register_1	PWPR_1	8	H'FC04	PWM_1	16	4
PWM cycle register_1	PWCYR_1	16	H'FC06	PWM_1	16	4
PWM buffer register_1A	PWBFR_1A	16	H'FC08	PWM_1	16	4
PWM buffer register_1C	PWBFR_1C	16	H'FC0A	PWM_1	16	4
PWM buffer register_1E	PWBFR_1E	16	H'FC0C	PWM_1	16	4
PWM buffer register_1G	PWBFR_1G	16	H'FC0E	PWM_1	16	4
PWM control register_2	PWCR_2	8	H'FC10	PWM_2	16	4
PWM output control register_2	PWOCR_2	8	H'FC12	PWM_2	16	4
PWM polarity register_2	PWPR_2	8	H'FC14	PWM_2	16	4
PWM cycle register_2	PWCYR_2	16	H'FC16	PWM_2	16	4
PWM buffer register_2A	PWBFR_2A	16	H'FC18	PWM_2	16	4
PWM buffer register_2B	PWBFR_2B	16	H'FC1A	PWM_2	16	4
PWM buffer register_2C	PWBFR_2C	16	H'FC1C	PWM_2	16	4
PWM buffer register_2D	PWBFR_2D	16	H'FC1E	PWM_2	16	4
Port H data direction register	PHDDR	8	H'FC20	PORT	16	4
Port J data direction register	PJDDR	8	H'FC21	PORT	16	4
Port H data register	PHDR	8	H'FC24	PORT	16	4
Port J data register	PJDR	8	H'FC25	PORT	16	4
Port H register	PORTH	8	H'FC28	PORT	16	4
Port J register	PORTJ	8	H'FC29	PORT	16	4
Module stop control register D	MSTPCRD	8	H'FC60	SYSTEM	8	4

Register Name	Abbreviation	Number of Bits	Address*	Module	Data Bus Width	Number of Access States
Serial Control Register X	SCRX	8	H'FDB4	IIC	8	2
DDC Switch Register	DDCSWR	8	H'FDB5	IIC	8	2
Standby control register	SBYCR	8	H'FDE4	SYSTEM	8	2
System control register	SYSCR	8	H'FDE5	SYSTEM	8	2
System clock control register	SCKCR	8	H'FDE6	SYSTEM	8	2
Mode control register	MDCR	8	H'FDE7	SYSTEM	8	2
Module stop control register A	MSTPCRA	8	H'FDE8	SYSTEM	8	2
Module stop control register B	MSTPCRB	8	H'FDE9	SYSTEM	8	2
Module stop control register C	MSTPCRC	8	H'FDEA	SYSTEM	8	2
Pin function control register	PFCR	8	H'FDEB	BSC	8	2
Low power control register	LPWRCR	8	H'FDEC	SYSTEM	8	2
Break address register A	BARA	32	H'FE00	PBC	32	2
Break address register B	BARB	32	H'FE04	PBC	32	2
Break control register A	BCRA	8	H'FE08	PBC	8	2
Break control register B	BCRB	8	H'FE09	PBC	8	2
IRQ sense control register H	ISCRH	8	H'FE12	INT	8	2
IRQ sense control register L	ISCRL	8	H'FE13	INT	8	2
IRQ enable register	IER	8	H'FE14	INT	8	2
IRQ status register	ISR	8	H'FE15	INT	8	2
DTC enable register A	DTCERA	8	H'FE16	DTC	8	2
DTC enable register B	DTCERB	8	H'FE17	DTC	8	2
DTC enable register C	DTCERC	8	H'FE18	DTC	8	2
DTC enable register D	DTCERD	8	H'FE19	DTC	8	2
DTC enable register E	DTCERE	8	H'FE1A	DTC	8	2
DTC enable register F	DTCERF	8	H'FE1B	DTC	8	2
DTC enable register G	DTCERG	8	H'FE1C	DTC	8	2
DTC vector register	DTVECR	8	H'FE1F	DTC	8	2
PPG output control register	PCR	8	H'FE26	PPG	8	2
PPG output mode register	PMR	8	H'FE27	PPG	8	2
Next data enable register H	NDERH	8	H'FE28	PPG	8	2

Register Name	Abbreviation	Number of Bits	Address*	Module	Data Bus Width	Number of Access States
Next data enable register L	NDERL	8	H'FE29	PPG	8	2
Output data register H	PODRH	8	H'FE2A	PPG	8	2
Output data register L	PODRL	8	H'FE2B	PPG	8	2
Next data register H	NDRH	8	H'FE2C	PPG	8	2
Next data register L	NDRL	8	H'FE2D	PPG	8	2
Next data register H	NDRH	8	H'FE2E	PPG	8	2
Next data register L	NDRL	8	H'FE2F	PPG	8	2
Port 1 data direction register	P1DDR	8	H'FE30	PORT	8	2
Port 3 data direction register	P3DDR	8	H'FE32	PORT	8	2
Port A data direction register	PADDR	8	H'FE39	PORT	8	2
Port B data direction register	PBDDR	8	H'FE3A	PORT	8	2
Port C data direction register	PCDDR	8	H'FE3B	PORT	8	2
Port D data direction register	PDDDR	8	H'FE3C	PORT	8	2
Port E data direction register	PEDDR	8	H'FE3D	PORT	8	2
Port F data direction register	PFDDR	8	H'FE3E	PORT	8	2
Port A pull-up MOS control register	PAPCR	8	H'FE40	PORT	8	2
Port B pull-up MOS control register	PBPCR	8	H'FE41	PORT	8	2
Port C pull-up MOS control register	PCPCR	8	H'FE42	PORT	8	2
Port D pull-up MOS control register	PDPCR	8	H'FE43	PORT	8	2
Port E pull-up MOS control register	PEPCR	8	H'FE44	PORT	8	2
Port 3 open-drain control register	P3ODR	8	H'FE46	PORT	8	2
Port A open-drain control register	PAODR	8	H'FE47	PORT	8	2
Port B open-drain control register	PBODR	8	H'FE48	PORT	8	2
Port C open-drain control register	PCODR	8	H'FE49	PORT	8	2
Timer control register_3	TCR_3	8	H'FE80	TPU_3	16	2
Timer mode register_3	TMDR_3	8	H'FE81	TPU_3	16	2
Timer I/O control register H_3	TIORH_3	8	H'FE82	TPU_3	16	2

Register Name	Abbreviation	Number of Bits	Address*	Module	Data Bus Width	Number of Access States
Timer I/O control register L_3	TIORL_3	8	H'FE83	TPU_3	16	2
Timer interrupt enable register_3	TIER_3	8	H'FE84	TPU_3	16	2
Timer status register_3	TSR_3	8	H'FE85	TPU_3	16	2
Timer counter H_3	TCNTH_3	8	H'FE86	TPU_3	16	2
Timer counter L_3	TCNTL_3	8	H'FE87	TPU_3	16	2
Timer general register AH_3	TGRAH_3	8	H'FE88	TPU_3	16	2
Timer general register AL_3	TGRAL_3	8	H'FE89	TPU_3	16	2
Timer general register BH_3	TGRBH_3	8	H'FE8A	TPU_3	16	2
Timer general register BL_3	TGRBL_3	8	H'FE8B	TPU_3	16	2
Timer general register CH_3	TGRCH_3	8	H'FE8C	TPU_3	16	2
Timer general register CL_3	TGRCL_3	8	H'FE8D	TPU_3	16	2
Timer general register DH_3	TGRDH_3	8	H'FE8E	TPU_3	16	2
Timer general register DL_3	TGRDL_3	8	H'FE8F	TPU_3	16	2
Timer control register_4	TCR_4	8	H'FE90	TPU_4	16	2
Timer mode register_4	TMDR_4	8	H'FE91	TPU_4	16	2
Timer I/O control register _4	TIOR_4	8	H'FE92	TPU_4	16	2
Timer interrupt enable register_4	TIER_4	8	H'FE94	TPU_4	16	2
Timer status register_4	TSR_4	8	H'FE95	TPU_4	16	2
Timer counter H_4	TCNTH_4	8	H'FE96	TPU_4	16	2
Timer counter L_4	TCNTL_4	8	H'FE97	TPU_4	16	2
Timer general register AH_4	TGRAH_4	8	H'FE98	TPU_4	16	2
Timer general register AL_4	TGRAL_4	8	H'FE99	TPU_4	16	2
Timer general register BH_4	TGRBH_4	8	H'FE9A	TPU_4	16	2
Timer general register BL_4	TGRBL_4	8	H'FE9B	TPU_4	16	2
Timer control register_5	TCR_5	8	H'FEA0	TPU_5	16	2
Timer mode register_5	TMDR_5	8	H'FEA1	TPU_5	16	2
Timer I/O control register _5	TIOR_5	8	H'FEA2	TPU_5	16	2
Timer interrupt enable register_5	TIER_5	8	H'FEA4	TPU_5	16	2
Timer status register_5	TSR_5	8	H'FEA5	TPU_5	16	2
Timer counter H_5	TCNTH_5	8	H'FEA6	TPU_5	16	2

Register Name	Abbreviation	Number of Bits	Address*	Module	Data Bus Width	Number of Access States
Timer counter L_5	TCNTL_5	8	H'FEA7	TPU_5	16	2
Timer general register AH_5	TGRAH_5	8	H'FEA8	TPU_5	16	2
Timer general register AL_5	TGRAL_5	8	H'FEA9	TPU_5	16	2
Timer general register BH_5	TGRBH_5	8	H'FEAA	TPU_5	16	2
Timer general register BL_5	TGRBL_5	8	H'FEAB	TPU_5	16	2
Timer start register	TSTR	8	H'FEB0	TPU common	16	2
Timer synchro register	TSYR	8	H'FEB1	TPU common	16	2
Interrupt priority register A	IPRA	8	H'FEC0	INT	8	2
Interrupt priority register B	IPRB	8	H'FEC1	INT	8	2
Interrupt priority register C	IPRC	8	H'FEC2	INT	8	2
Interrupt priority register D	IPRD	8	H'FEC3	INT	8	2
Interrupt priority register E	IPRE	8	H'FEC4	INT	8	2
Interrupt priority register F	IPRF	8	H'FEC5	INT	8	2
Interrupt priority register G	IPRG	8	H'FEC6	INT	8	2
Interrupt priority register H	IPRH	8	H'FEC7	INT	8	2
Interrupt priority register J	IPRJ	8	H'FEC9	INT	8	2
Interrupt priority register K	IPRK	8	H'FECA	INT	8	2
Interrupt priority register L	IPRL	8	H'FECB	INT	8	2
Interrupt priority register M	IPRM	8	H'FECC	INT	8	2
Bus width control register	ABWCR	8	H'FED0	BSC	8	2
Access state control register	ASTCR	8	H'FED1	BSC	8	2
Wait control register H	WCRH	8	H'FED2	BSC	8	2
Wait control register L	WCRL	8	H'FED3	BSC	8	2
Bus control register H	BCRH	8	H'FED4	BSC	8	2
Bus control register L	BCRL	8	H'FED5	BSC	8	2
RAM emulation register	RAMER	8	H'FEDB	FLASH (F-ZTAT)	8	2
Port 1 data register	P1DR	8	H'FF00	PORT	8	2
Port 3 data register	P3DR	8	H'FF02	PORT	8	2

Register Name	Abbreviation	Number of Bits	Address*	Module	Data Bus Width	Number of Access States
Port A data register	PADR	8	H'FF09	PORT	8	2
Port B data register	PBDR	8	H'FF0A	PORT	8	2
Port C data register	PCDR	8	H'FF0B	PORT	8	2
Port D data register	PDDR	8	H'FF0C	PORT	8	2
Port E data register	PEDR	8	H'FF0D	PORT	8	2
Port F data register	PFDR	8	H'FF0E	PORT	8	2
Timer control register_0	TCR_0	8	H'FF10	TPU_0	16	2
Timer mode register_0	TMDR_0	8	H'FF11	TPU_0	16	2
Timer I/O control register H_0	TIORH_0	8	H'FF12	TPU_0	16	2
Timer I/O control register L_0	TIORL_0	8	H'FF13	TPU_0	16	2
Timer interrupt enable register_0	TIER_0	8	H'FF14	TPU_0	16	2
Timer status register_0	TSR_0	8	H'FF15	TPU_0	16	2
Timer counter H_0	TCNTH_0	8	H'FF16	TPU_0	16	2
Timer counter L_0	TCNTL_0	8	H'FF17	TPU_0	16	2
Timer general register AH_0	TGRAH_0	8	H'FF18	TPU_0	16	2
Timer general register AL_0	TGRAL_0	8	H'FF19	TPU_0	16	2
Timer general register BH_0	TGRBH_0	8	H'FF1A	TPU_0	16	2
Timer general register BL_0	TGRBL_0	8	H'FF1B	TPU_0	16	2
Timer general register CH_0	TGRCH_0	8	H'FF1C	TPU_0	16	2
Timer general register CL_0	TGRCL_0	8	H'FF1D	TPU_0	16	2
Timer general register DH_0	TGRDH_0	8	H'FF1E	TPU_0	16	2
Timer general register DL_0	TGRDL_0	8	H'FF1F	TPU_0	16	2
Timer control register_1	TCR_1	8	H'FF20	TPU_1	16	2
Timer mode register_1	TMDR_1	8	H'FF21	TPU_1	16	2
Timer I/O control register _1	TIOR_1	8	H'FF22	TPU_1	16	2
Timer interrupt enable register_1	TIER_1	8	H'FF24	TPU_1	16	2
Timer status register_1	TSR_1	8	H'FF25	TPU_1	16	2
Timer counter H_1	TCNTH_1	8	H'FF26	TPU_1	16	2
Timer counter L_1	TCNTL_1	8	H'FF27	TPU_1	16	2

Register Name	Abbreviation	Number of Bits	Address*	Module	Data Bus Width	Number of Access States
Timer general register AH_1	TGRAH_1	8	H'FF28	TPU_1	16	2
Timer general register AL_1	TGRAL_1	8	H'FF29	TPU_1	16	2
Timer general register BH_1	TGRBH_1	8	H'FF2A	TPU_1	16	2
Timer general register BL_1	TGRBL_1	8	H'FF2B	TPU_1	16	2
Timer control register_2	TCR_2	8	H'FF30	TPU_2	16	2
Timer mode register_2	TMDR_2	8	H'FF31	TPU_2	16	2
Timer I/O control register_2	TIOR_2	8	H'FF32	TPU_2	16	2
Timer interrupt enable register_2	TIER_2	8	H'FF34	TPU_2	16	2
Timer status register_2	TSR_2	8	H'FF35	TPU_2	16	2
Timer counter H_2	TCNTH_2	8	H'FF36	TPU_2	16	2
Timer counter L_2	TCNTL_2	8	H'FF37	TPU_2	16	2
Timer general register AH_2	TGRAH_2	8	H'FF38	TPU_2	16	2
Timer general register AL_2	TGRAL_2	8	H'FF39	TPU_2	16	2
Timer general register BH_2	TGRBH_2	8	H'FF3A	TPU_2	16	2
Timer general register BL_2	TGRBL_2	8	H'FF3B	TPU_2	16	2
Timer control/status register_0	TCSR_0	8	H'FF74	WDT_0	16	2
Timer counter_0	TCNT_0	8	H'FF75	WDT_0	16	2
Reset control/status register	RSTCSR	8	H'FF77	WDT_0	16	2
Serial mode register_0	SMR_0	8	H'FF78	SCI_0	8	2
I^2C Bus Control Register	ICCR_0	8	H'FF78	IIC_0	8	2
Bit rate register_0	BRR_0	8	H'FF79	SCI_0	8	2
I^2C Bus Status Register	ICSR0	8	H'FF79	IIC_0	8	2
Serial control register_0	SCR_0	8	H'FF7A	SCI_0	8	2
Transmit data register_0	TDR_0	8	H'FF7B	SCI_0	8	2
Serial status register_0	SSR_0	8	H'FF7C	SCI_0	8	2
Receive data register_0	RDR_0	8	H'FF7D	SCI_0	8	2
Smart card mode register_0	SCMR_0	8	H'FF7E	SCI_0	8	2
I^2C Bus Data Register	ICDR_0	8	H'FF7E	IIC_0	8	2
Serial mode register_1	SMR_1	8	H'FF80	SCI_1	8	2
I^2C Bus Control Register	ICCR_1	8	H'FF80	IIC_0	8	2

Register Name	Abbreviation	Number of Bits	Address*	Module	Data Bus Width	Number of Access States
Bit rate register_1	BRR_1	8	H'FF81	SCI_1	8	2
I^2C Bus Status Register	ICSR_1	8	H'FF81	IIC_1	8	2
Serial control register_1	SCR_1	8	H'FF82	SCI_1	8	2
Transmit data register_1	TDR_1	8	H'FF83	SCI_1	8	2
Serial status register_1	SSR_1	8	H'FF84	SCI_1	8	2
Receive data register_1	RDR_1	8	H'FF85	SCI_1	8	2
Smart card mode register_1	SCMR_1	8	H'FF86	SCI_1	8	2
Serial mode register_2	SMR_2	8	H'FF88	SCI_2	8	2
Bit rate register_2	BRR_2	8	H'FF89	SCI_2	8	2
Serial control register_2	SCR_2	8	H'FF8A	SCI_2	8	2
Transmit data register_2	TDR_2	8	H'FF8B	SCI_2	8	2
Serial status register_2	SSR_2	8	H'FF8C	SCI_2	8	2
Receive data register_2	RDR_2	8	H'FF8D	SCI_2	8	2
Smart card mode register_2	SCMR_2	8	H'FF8E	SCI_2	8	2
A/D data register AH	ADDRAH	8	H'FF90	A/D	8	2
A/D data register AL	ADDRAL	8	H'FF91	A/D	8	2
A/D data register BH	ADDRBH	8	H'FF92	A/D	8	2
A/D data register BL	ADDRBL	8	H'FF93	A/D	8	2
A/D data register CH	ADDRCH	8	H'FF94	A/D	8	2
A/D data register CL	ADDRCL	8	H'FF95	A/D	8	2
A/D data register DH	ADDRDH	8	H'FF96	A/D	8	2
A/D data register DL	ADDRDL	8	H'FF97	A/D	8	2
A/D control/status register	ADCSR	8	H'FF98	A/D	8	2
A/D control register	ADCR	8	H'FF99	A/D	8	2
Timer control/status register_1	TCSR_1	8	H'FFA2	WDT_1	16	2
Timer counter_1	TCNT_1	8	H'FFA3	WDT_1	16	2
D/A Data Register 0	DADR0	8	H'FFA4	D/A	8	2
D/A Data Register 1	DADR1	8	H'FFA5	D/A	8	2
D/A Control Register	DACR	8	H'FFA6	D/A	8	2
Flash memory control register 1	FLMCR1	8	H'FFA8	FLASH (F-ZTAT)	8	2

Register Name	Abbreviation	Number of Bits	Address*	Module	Data Bus Width	Number of Access States
Flash memory control register 2	FLMCR2	8	H'FFA9	FLASH (F-ZTAT)	8	2
Erase block register 1	EBR1	8	H'FFAA	FLASH (F-ZTAT)	8	2
Erase block register 2	EBR2	8	H'FFAB	FLASH (F-ZTAT)	8	2
Flash memory power control register	FLPWCR	8	H'FFAC	FLASH (F-ZTAT)	8	2
Port 1 register	PORT1	8	H'FFB0	PORT	8	2
Port 2 register	PORT2	8	H'FFB1	PORT	8	2
Port 3 register	PORT3	8	H'FFB2	PORT	8	2
Port 4 register	PORT4	8	H'FFB3	PORT	8	2
Port 5 register	PORT5	8	H'FFB4	PORT	8	2
Port 9 register	PORT9	8	H'FFB8	PORT	8	2
Port A register	PORTA	8	H'FFB9	PORT	8	2
Port B register	PORTB	8	H'FFBA	PORT	8	2
Port C register	PORTC	8	H'FFBB	PORT	8	2
Port D register	PORTD	8	H'FFBC	PORT	8	2
Port F register	PORTF	8	H'FFBE	PORT	8	2

Note: * The lower 16 bits are indicated.

22.2 Register Bits

Register addresses and bit names of the on-chip peripheral modules are described below.

Each line covers eight bits, and 16-bit registers are shown as 2 lines, respectively.

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PWCR_1	—	—	IE	CMF	CST	CKS2	CKS1	CKS0	PWM_1
PWOCR_1	OE1H	OE1G	OE1F	OE1E	OE1D	OE1C	OE1B	OE1A	
PWPR_1	OPS1H	OPS1G	OPS1F	OPS1E	OPS1D	OPS1C	OPS1B	OPS1A	
PWCYR_1	—	—	—	—	—	—	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
PWBFR_1A	—	—	—	OTS	—	—	DT9	DT8	
	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
PWBFR_1B	—	—	—	OTS	—	—	DT9	DT8	
	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
PWBFR_1C	—	—	—	OTS	—	—	DT9	DT8	
	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
PWBFR_1D	—	—	—	OTS	—	—	DT9	DT8	
	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
PWCR_2	—	—	IE	CMF	CST	CKS2	CKS1	CKS0	PWM_2
PWOCR_2	OE2H	OE2G	OE2F	OE2E	OE2D	OE2C	OE2B	OE2A	
PWPR_2	OPS2H	OPS2G	OPS2F	OPS2E	OPS2D	OPS2C	OPS2B	OPS2A	
PWCYR_2	—	—	—	—	—	—	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
PWBFR_2A	—	—	—	TDS	—	—	DT9	DT8	
	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
PWBFR_2B	—	—	—	TDS	—	—	DT9	DT8	
	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
PWBFR_2C	—	—	—	TDS	—	—	DT9	DT8	
	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
PWBFR_2D	—	—	—	TDS	—	—	DT9	DT8	
	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PHDDR	PH7DDR	PH6DDR	PH5DDR	PH4DDR	PH3DDR	PH2DDR	PH1DDR	PH0DDR	PORT
PJDDR	PJ7DDR	PJ6DDR	PJ5DDR	PJ4DDR	PJ3DDR	PJ2DDR	PJ1DDR	PJ0DDR	
PHDR	PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR	
PJDR	PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR	
PORTH	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	
PORTJ	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	
MSTPCRD	MSTPD7	MSTPD6	MSTPD5	MSTPD4	MSTPD3	MSTPD2	MSTPD1	MSTPD0	SYSTEM
SCRX	—	IICX1	IICX0	IICE	—	—	—	—	IIC
DOCSWR	SWE	SW	IE	IF	CLR3	CLR2	CLR1	CLR0	
SBYCR	SSBY	STS2	STS1	STS0	OPE	—	—	—	SYSTEM
SYSCR	MACS	—	INTM1	INTM0	NMIEG	—	—	RAME	
SCKCR	PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0	
MDCR	—	—	—	—	—	MDS2	MDS1	MDS0	
MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0	
MSTPCRB	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0	
MSTPCRC	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0	
PFCR	—	—	—	—	AE3	AE2	AE1	AE0	BSC
LPWRCR	DTON	LSON	NESEL	SUBSTP	RFCUT	—	STC1	STC0	SYSTEM
BARA	—	—	—	—	—	—	—	—	PBC
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	
BARB	—	—	—	—	—	—	—	—	
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16	
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0	
BCRA	CMFA	CDA	BAMRA2	BAMRA1	BAMRA0	CSELA1	CSELA0	BIEA	
BCRB	CMFB	CDB	BAMRB2	BAMRB1	BAMRB0	CSELB1	CSELB0	BIEB	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ISCRH	—	—	—	—	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	INT
ISCRL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA	
IER	—	—	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
ISR	—	—	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0	DTC
DTCERB	DTCEB7	DTCEB6	DTCEB5	DTCEB4	DTCEB3	DTCEB2	DTCEB1	DTCEB0	
DTCERC	DTCEC7	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0	
DTCERD	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0	
DTCERE	DTCEE7	DTCEE6	DTCEE5	DTCEE4	DTCEE3	DTCEE2	DTCEE1	DTCEE0	
DTCERF	DTCEF7	DTCEF6	DTCEF5	DTCEF4	DTCEF3	DTCEF2	DTCEF1	DTCEF0	
DTCERG	DTCEG7	DTCEG6	DTCEG5	DTCEG4	DTCEG3	DTCEG2	DTCEG1	DTCEG0	
DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	
PCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0	PPG
PMR	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV	
NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8	
NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0	
PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8	
PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0	
NDRH	NDR15	NDR14	NDR13	NDR12	—	—	—	—	
NDRL	NDR7	NDR6	NDR5	NDR4	—	—	—	—	
NDRH	—	—	—	—	NDR11	NDR10	NDR9	NDR8	
NDRL	—	—	—	—	NDR3	NDR2	NDR1	NDR0	
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	PORT
P3DDR	—	—	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR	
PADDR	—	—	—	—	PA3DDR	PA2DDR	PA1DDR	PA0DDR	
PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR	
PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR	
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR	
PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR	
PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	—	—	PF0DDR	
PAPCR	—	—	—	—	PA3PCR	PA2PCR	PA1PCR	PA0PCR	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR	PORT
PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR	
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR	
PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR	
P3ODR	—	—	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR	
PAODR	—	—	—	—	PA3ODR	PA2ODR	PA1ODR	PA0ODR	
PBODR	PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR	
PCODR	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR	
TCR_3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_3
TMDR_3	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_3	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_3	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNTH_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TCNTL_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRAH_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TGRAL_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRBH_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TGRBL_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRCH_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TGRCL_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRDH_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TGRDL_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_4
TMDR_4	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNTH_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TCNTL_4	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
TGRAH_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	TPU_4
TGRAL_4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRBH_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TGRBL_4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_5
TCNTH_5	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TCNTL_5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRAH_5	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TGRAL_5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRBH_5	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TGRBL_5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0	TPU common
TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	
IPRA	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	INT
IPRB	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRC	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRD	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRE	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRF	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRG	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRH	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRJ	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRK	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRL	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRM	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	BSC
ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0	
WCRH	W71	W70	W61	W60	W51	W50	W41	W40	
WCRL	W31	W30	W21	W20	W11	W10	W01	W00	
BCRH	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	—	—	
BCRL	—	—	—	—	—	—	WDBE	—	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
RAMER	—	—	—	—	RAMS	RAM2	RAM1	RAM0	FLASH (F-ZTAT)
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	PORT
P3DR	—	—	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR	
PADR	—	—	—	—	PA3DR	PA2DR	PA1DR	PA0DR	
PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
PCDR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	—	—	PF0DR	
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	
TCNTH_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TCNTL_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRAH_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TGRAL_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRBH_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TGRBL_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRCH_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TGRCL_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRDH_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TGRDL_0	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	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCNTH_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	TPU_1
TCNTL_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRAH_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TGRAL_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRBH_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TGRBL_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
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	
TCNTH_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TCNTL_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRAH_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TGRAL_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRBH_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
TGRBL_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCSR_0	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0	WDT_0
TCNT_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
RSTCSR	WOVF	RSTE	RSTS	—	—	—	—	—	
SMR_0*	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_0
	(GM)	(BLK)	(PE)	(O/$\overline{E}$)	(BCP1)	(BCP0)	(CKS1)	(CKS0)	
ICCR_0	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	IIC_0
BRR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SCI_0
ICSR_0	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	IIC_0
SCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	SCI_0
TDR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_0*	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
	(TDRE)	(RDRF)	(ORER)	(ERS)	(PER)	(TEND)	(MPB)	(MPBT)	
RDR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_0	—	—	—	—	SDIR	SINV	—	SMIF	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ICDR_0	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0	IIC_0
SARX_0	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX	
ICMR_0	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0	
SAR_0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
SMR_1*	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_1
	(GM)	(BLK)	(PE)	(O/$\overline{E}$)	(BCP1)	(BCP0)	(CKS1)	(CKS0)	
ICCR_1	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	IIC_1
BRR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SCI_1
ICSR_1	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	IIC_1
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	SCI_1
TDR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_1*	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
	(TDRE)	(RDRF)	(ORER)	(ERS)	(PER)	(TEND)	(MPB)	(MPBT)	
RDR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF	
ICDR_1	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0	IIC_1
SARX_1	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX	
ICMR_1	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0	
SAR_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
SMR_2*	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_2
	(GM)	(BLK)	(PE)	(O/$\overline{E}$)	(BCP1)	(BCP0)	(CKS1)	(CKS0)	
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	
	(TDRE)	(RDRF)	(ORER)	(ERS)	(PER)	(TEND)	(MPB)	(MPBT)	
RDR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_2	—	—	—	—	SDIR	SINV	—	SMIF	
ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
ADDRAL	AD1	AD0	—	—	—	—	—	—	
ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ADDRBL	AD1	AD0	—	—	—	—	—	—	A/D
ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRCL	AD1	AD0	—	—	—	—	—	—	
ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRDL	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	SCAN	CH3	CH2	CH1	CH0	
ADCR	TRGS1	TRGS0	—	—	CKS1	CKS0	—	—	
TCSR_1	OVF	WT/$\overline{IT}$	TME	PSS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0	WDT_1
TCNT_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
DADR0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	DA
DADR1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
DACR	DAOE1	DA0E0	DAE	—	—	—	—	—	
FLMCR1	FWE	SWE	ESU	PSU	EV	PV	E	P	FLASH (F-ZTAT)
FLMCR2	FLER	—	—	—	—	—	—	—	
EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
EBR2	—	—	EB13	EB12	EB11	EB10	EB9	EB8	
FLPWCR	PDWND	—	—	—	—	—	—	—	
PORT1	P17	P16	P15	P14	P13	P12	P11	P10	PORT
PORT3	—	—	P35	P34	P33	P32	P31	P30	
PORT4	P47	P46	P45	P44	P43	P42	P41	P40	
PORT9	—	—	—	—	P93	P92	P91	P90	
PORTA	—	—	—	—	PA3	PA2	PA1	PA0	
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	
PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	
PORTF	PF7	PF6	PF5	PF4	PF3	—	—	PF1	

Note: * Parts of the bit functions differ in normal mode and the smart card interface mode. The bit function in smart card interface mode is enclosed in parentheses.

22.3 Register States in Each Operating Mode

Register Abbreviation	Reset	High speed	Medium speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
PWCR_1	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	PWM_1
PWOCR_1	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
PWPR_1	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
PWCYR_1	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
PWBFR_1A	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
PWBFR_1C	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
PWBFR_1E	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
PWBFR_1G	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
PWCR_2	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	PWM_2
PWOCR_2	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
PWPR_2	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
PWCYR_2	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
PWBFR_2A	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
PWBFR_2B	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
PWBFR_2C	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
PWBFR_2D	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
PHDDR	Initialized	–	–	–	–	–	–	–	–	Initialized	PORT
PJDDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PHDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PJDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PORTH	Initialized	–	–	–	–	–	–	–	–	Initialized	
PORTJ	Initialized	–	–	–	–	–	–	–	–	Initialized	
MSTPCRD	Initialized	–	–	–	–	–	–	–	–	Initialized	SYSTEM
SCRX	Initialized	–	–	–	–	–	–	–	–	Initialized	IIC
DDCSWR	Initialized	–	–	–	–	–	–	–	–	Initialized	

Register Abbreviation	Reset	High speed	Medium speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
SBYCR	Initialized	–	–	–	–	–	–	–	–	Initialized	SYSTEM
SYSCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
SCKCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
MDCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
MSTPCRA	Initialized	–	–	–	–	–	–	–	–	Initialized	
MSTPCRB	Initialized	–	–	–	–	–	–	–	–	Initialized	
MSTPCRC	Initialized	–	–	–	–	–	–	–	–	Initialized	
PFCR	Initialized	–	–	–	–	–	–	–	–	Initialized	BSC
LPWRCR	Initialized	–	–	–	–	–	–	–	–	Initialized	SYSTEM
BARA	Initialized	–	–	–	–	–	–	–	–	Initialized	PBC
BARB	Initialized	–	–	–	–	–	–	–	–	Initialized	
BCRA	Initialized	–	–	–	–	–	–	–	–	Initialized	
BCRB	Initialized	–	–	–	–	–	–	–	–	Initialized	
ISCRH	Initialized	–	–	–	–	–	–	–	–	Initialized	INT
ISCRL	Initialized	–	–	–	–	–	–	–	–	Initialized	
IER	Initialized	–	–	–	–	–	–	–	–	Initialized	
ISR	Initialized	–	–	–	–	–	–	–	–	Initialized	
DTCERA	Initialized	–	–	–	–	–	–	–	–	Initialized	DTC
DTCERB	Initialized	–	–	–	–	–	–	–	–	Initialized	
DTCERC	Initialized	–	–	–	–	–	–	–	–	Initialized	
DTCERD	Initialized	–	–	–	–	–	–	–	–	Initialized	
DTCERE	Initialized	–	–	–	–	–	–	–	–	Initialized	
DTCERF	Initialized	–	–	–	–	–	–	–	–	Initialized	
DTCERG	Initialized	–	–	–	–	–	–	–	–	Initialized	
DTVECR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PCR	Initialized	–	–	–	–	–	–	–	–	Initialized	PPG
PMR	Initialized	–	–	–	–	–	–	–	–	Initialized	

Register Abbreviation	Reset	High speed	Medium speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
NDERH	Initialized	–	–	–	–	–	–	–	–	Initialized	PPG
NDERL	Initialized	–	–	–	–	–	–	–	–	Initialized	
PODRH	Initialized	–	–	–	–	–	–	–	–	Initialized	
PODRL	Initialized	–	–	–	–	–	–	–	–	Initialized	
NDRH	Initialized	–	–	–	–	–	–	–	–	Initialized	
NDRL	Initialized	–	–	–	–	–	–	–	–	Initialized	
NDRH	Initialized	–	–	–	–	–	–	–	–	Initialized	
NDRL	Initialized	–	–	–	–	–	–	–	–	Initialized	
P1DDR	Initialized	–	–	–	–	–	–	–	–	Initialized	PORT
P3DDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PADDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PBDDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PCDDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PDDDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PEDDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PFDDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PAPCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PBPCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PCPCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PDPCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PEPCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
P3ODR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PAODR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PBODR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PCODR	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCR_3	Initialized	–	–	–	–	–	–	–	–	Initialized	TPU_3
TMDR_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIORH_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIORL_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIER_3	Initialized	–	–	–	–	–	–	–	–	Initialized	

Register Abbreviation	Reset	High speed	Medium speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
TSR_3	Initialized	–	–	–	–	–	–	–	–	Initialized	TPU_3
TCNTH_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCNTL_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRAH_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRAL_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRBH_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRBL_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRCH_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRCL_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRDH_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRDL_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCR_4	Initialized	–	–	–	–	–	–	–	–	Initialized	TPU_4
TMDR_4	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIOR_4	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIER_4	Initialized	–	–	–	–	–	–	–	–	Initialized	
TSR_4	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCNTH_4	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCNTL_4	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRAH_4	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRAL_4	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRBH_4	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRBL_4	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCR_5	Initialized	–	–	–	–	–	–	–	–	Initialized	TPU_5
TMDR_5	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIOR_5	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIER_5	Initialized	–	–	–	–	–	–	–	–	Initialized	
TSR_5	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCNTH_5	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCNTL_5	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRAH_5	Initialized	–	–	–	–	–	–	–	–	Initialized	

Register Abbreviation	Reset	High speed	Medium speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
TGRAL_5	Initialized	–	–	–	–	–	–	–	–	Initialized	TPU_5
TGRBH_5	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRBL_5	Initialized	–	–	–	–	–	–	–	–	Initialized	
TSTR	Initialized	–	–	–	–	–	–	–	–	Initialized	TPU common
TSYR	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRA	Initialized	–	–	–	–	–	–	–	–	Initialized	INT
IPRB	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRC	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRD	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRE	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRF	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRG	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRH	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRJ	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRK	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRL	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRM	Initialized	–	–	–	–	–	–	–	–	Initialized	
ABWCR	Initialized	–	–	–	–	–	–	–	–	Initialized	BSC
ASTCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
WCRH	Initialized	–	–	–	–	–	–	–	–	Initialized	
WCRL	Initialized	–	–	–	–	–	–	–	–	Initialized	
BCRH	Initialized	–	–	–	–	–	–	–	–	Initialized	
BCRL	Initialized	–	–	–	–	–	–	–	–	Initialized	
RAMER	Initialized	–	–	–	–	–	–	–	–	Initialized	FLASH (F-ZTAT)
P1DR	Initialized	–	–	–	–	–	–	–	–	Initialized	PORT
P3DR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PADR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PBDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PCDR	Initialized	–	–	–	–	–	–	–	–	Initialized	

Register Abbreviation	Reset	High speed	Medium speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
PDDR	Initialized	–	–	–	–	–	–	–	–	Initialized	PORT
PEDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PFDR	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	
TCNTH_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCNTL_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRAH_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRAL_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRBH_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRBL_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRCH_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRCL_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRDH_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRDL_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	TPU_1
TMDR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIOR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIER_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TSR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCNTH_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCNTL_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRAH_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRAL_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRBH_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRBL_1	Initialized	–	–	–	–	–	–	–	–	Initialized	

Register Abbreviation	Reset	High speed	Medium speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
TCR_2	Initialized	–	–	–	–	–	–	–	–	Initialized	TPU_2
TMDR_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIOR_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIER_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TSR_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCNTH_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCNTL_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRAH_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRAL_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRBH_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRBL_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCSR_0	Initialized	–	–	–	–	–	–	–	–	Initialized	WDT_0
TCNT_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
RSTCSR	Initialized	–	–	–	–	–	–	–	–	Initialized	
SMR_0	Initialized	–	–	–	–	–	–	–	–	Initialized	SCI_0
ICCR_0	Initialized	–	–	–	–	–	–	–	–	Initialized	IIC_0
BRR_0	Initialized	–	–	–	–	–	–	–	–	Initialized	SCI_0
ICSR_0	Initialized	–	–	–	–	–	–	–	–	Initialized	IIC_0
SCR_0	Initialized	–	–	–	–	–	–	–	–	Initialized	SCI_0
TDR_0	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_0	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_0	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
ICDR_0	–	–	–	–	–	–	–	–	–	–	IIC_0
SARX_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
ICMR_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
SAR_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
SMR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	SCI_1
ICCR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	IIC_1
BRR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	SCI_1

Register Abbreviation	Reset	High speed	Medium speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
ICSR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	IIC_1
SCR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	SCI_1
TDR_1	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_1	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_1	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
ICDR_1	–	–	–	–	–	–	–	–	–	–	IIC_1
SARX_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
ICMR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
SAR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
SMR_2	Initialized	–	–	–	–	–	–	–	–	Initialized	SCI_2
BRR_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
SCR_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TDR_2	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_2	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_2	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
ADDRAH	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	A/D
ADDRAL	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRBH	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRBL	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRCH	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRCL	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRDH	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRDL	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADCSR	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADCR	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
TCSR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	WDT_1
TCNT_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
DADR0	Initialized	–	–	–	–	–	–	–	–	Initialized	D/A

Register Abbreviation	Reset	High speed	Medium speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
DADR1	Initialized	–	–	–	–	–	–	–	–	Initialized	D/A
DACR	Initialized	–	–	–	–	–	–	–	–	Initialized	
FLMCR1	Initialized	–	–	–	–	–	–	–	–	Initialized	FLASH (F-ZTAT)
FLMCR2	Initialized	–	–	–	–	–	–	–	–	Initialized	
EBR1	Initialized	–	–	–	–	–	–	–	–	Initialized	
EBR2	Initialized	–	–	–	–	–	–	–	–	Initialized	
FLPWCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PORT1	–	–	–	–	–	–	–	–	–	–	PORT
PORT3	–	–	–	–	–	–	–	–	–	–	
PORT4	–	–	–	–	–	–	–	–	–	–	
PORT9	–	–	–	–	–	–	–	–	–	–	
PORTA	–	–	–	–	–	–	–	–	–	–	
PORTB	–	–	–	–	–	–	–	–	–	–	
PORTC	–	–	–	–	–	–	–	–	–	–	
PORTD	–	–	–	–	–	–	–	–	–	–	
PORTF	–	–	–	–	–	–	–	–	–	–	

Note: — is not initialized.

Section 23 Electrical Characteristics

23.1 Absolute Maximum Ratings

Table 23.1 lists the absolute maximum ratings.

Table 23.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	–0.3 to +7.0	V
Input voltage (OSC1, OSC2)	V_{in}	–0.3 +4.3	V
Input voltage (XTAL, EXTAL)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (ports 4 and 9)	V_{in}	–0.3 to AV_{CC} +0.3	V
Input voltage (ports H and J)	V_{in}	–0.3 to $PWMV_{CC}$ +0.3	V
Input voltage (except XTAL, EXTAL, OSC1, OSC2, ports 4, 9, H and J)	V_{in}	–0.3 to V_{CC} +0.3	V
Reference voltage	V_{ref}	–0.3 to AV_{CC} +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +7.0	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to +75	°C
		Wide-range specifications: –40 to +85	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum rating are exceeded.

23.2 DC Characteristics

Table 23.2 lists the DC characteristics. Table 23.3 lists the permissible output currents.

Table 23.2 DC Characteristics

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)*1 *5

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	IRQ0 to IRQ5	V_T^-	1.0	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$		
		$V_T^+ - V_T^-$	0.4	—	—		
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, FWE, MD2 to MD0	V_{IH}	$V_{CC} - 0.7$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$		
	Ports 1, 3, F		2.2	—	$V_{CC} + 0.3$		
	Ports A to E		$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$		
	Ports H, J		$PWMV_{CC} \times 0.8$	—	$PWMV_{CC} + 0.3$		
	Ports 4 and 9		$AV_{CC} \times 0.7$	—	$AV_{CC} + 0.3$		
Input low voltage	$\overline{RES}$, $\overline{STBY}$, NMI, FWE, MD2 to MD0	V_{IL}	–0.3	—	0.5	V	
	EXTAL		–0.3	—	0.8		
	Ports 1, 3, F		–0.3	—	0.8		
	Ports A to E		–0.3	—	$V_{CC} \times 0.2$		
	Ports H, J		–0.3	—	$PWMV_{CC} \times 0.2$		
	Ports 4, 9		–0.3	—	$AV_{CC} \times 0.2$		

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	Ports 1, 3, A to F, H,J (excluding P34 and P35)	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
	P34, P35		V_{CC} – 2.5	—	—		I_{OH} = –100 µA
	Ports 1, 3, A to F, H, J (excluding P34 and P35)		3.5	—	—		I_{OH} = –1 mA
	PWM1A to PWM1H, PWM2A to PWM2H		$PWMV_{CC}$ – 0.5	—	—		I_{OH} = –15 mA
Output low voltage	All output pins except PWM1A to PWM1H, PWM2A to PWM2H	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	PWM1A to PWM1H, PWM2A to PWM2H		—	—	0.5	V	I_{OL} = 15 mA
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
	$\overline{STBY}$, $\overline{NMI}$, MD2 to MD0		—	—	1.0		
	Ports 4, 9		—	—	1.0		V_{in} = 0.5 V to AV_{CC} – 0.5 V
Three-state leakage current (off state)	Ports 1, 3, A to F, H, J	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
MOS input pull-up current	Ports A to E	$-I_P$	50	—	300	µA	V_{in} = 0 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V
	NMI		—	—	30		f = 1 MHz
	All input pins except $\overline{RES}$ and NMI		—	—	15		T_a = 25°C
Current dissipation*2	Normal operation	I_{CC}*4	—	75	90	mA	f = 20 MHz
	Sleep mode		—	65	80		
	All modules stopped		—	57	—	mA	f = 20 MHz (reference value)
	Medium-speed mode (φ/32)		—	49	—		
	Subactive mode		—	130	220	µA	Using 32.768 kHz crystal resonator
	Subsleep mode		—	80	160		
	Watch mode		—	30	60		
	Standby mode		—	2.0	5.0	µA	$T_a \leq 50°C$
			—	—	20		$50°C < T_a$
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	1.0	2.0	mA	AV_{CC} = 5.0 V
	Idle		—	0.1	5.0	µA	
Reference current	During A/D and D/A conversion	AI_{CC}	—	4.0	5.0	mA	V_{ref} = 5.0 V
	Idle		—	0.1	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open. Apply a voltage between 4.5 V and 5.5 V to the AV_{CC} and V_{ref} pins by connecting them to V_{CC}, for instance. Set $V_{ref} \leq AV_{CC}$.

2. Current dissipation values are for V_{IH} (min.) = V_{CC} – 0.5 V, V_{IL} (max.) = 0.5 V with all output pins unloaded and the on-chip pull-up resistors in the off state.
3. The values are for $V_{RAM} \leq V_{CC} < 3.0$ V, V_{IH} (min.) = $V_{CC} \times 0.9$, and V_{IL} (max.) = 0.3 V.
4. I_{CC} depends on V_{CC} and f as follows:
   I_{CC} (max.) = 30 (mA) + 0.54 (mA/(MHz × V)) × V_{CC} × f (normal operation)

I_{CC} (max.) = 30 (mA) + 0.45 (mA/(MHz × V)) × V_{CC} × f (sleep mode)

5. If the motor-control PWM timer is not used, do not leave the $PMWV_{CC}$, or $PMWV_{SS}$ pins open. If the motor-control PWM timer is not used, apply a voltage of between 4.5 and 5.5 V to the $PWMV_{CC}$ pin, for instance, by connecting it to V_{CC}.

Table 23.3 Permissible Output Currents

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test condition
Permissible output low current (per pin)	All output pins except PWM1A to PWM1H, PWM2A to PWM2H	I_{OL}	—	—	10	mA	
	PWM1A to PWM1H, PWM2A to PWM2H	I_{OL}	—	—	25	mA	T_a = 85°C
			—	—	30	mA	T_a = 25°C
			—	—	40	mA	T_a = –40°C
Permissible output low current (total)	Total of all output pins excepting PWM1A to PWM1H, and PWM2A to PWM2H	$\sum I_{OL}$	—	—	80	mA	
	Total of PWM1A to PWM1H, and PWM2A to PWM2H	$\sum I_{OL}$	—	—	150	mA	T_a = 85°C
			—	—	180	mA	T_a = 25°C
			—	—	220	mA	T_a = –40°C
Permissible output high current (per pin)	All output pins except PWM1A to PWM1H, PWM2A to PWM2H	$-I_{OH}$	—	—	2.0	mA	
	PWM1A to PWM1H, PWM2A to PWM2H	$-I_{OH}$	—	—	25	mA	T_a = 85°C
			—	—	30	mA	T_a = 25°C
			—	—	40	mA	T_a = –40°C
Permissible output high current (total)	Total of all output pins excepting PWM1A to PWM1H, and PWM2A to PWM2H	$-\sum I_{OH}$	—	—	40	mA	
	Total of PWM1A to PWM1H, and PWM2A to PWM2H	$-\sum I_{OH}$	—	—	150	mA	T_a = 85°C
			—	—	180	mA	T_a = 25°C
			—	—	220	mA	T_a = –40°C

Note: To protect chip reliability, do not exceed the output current values in table 23.3.

Table 23.4 Bus Drive Characteristics

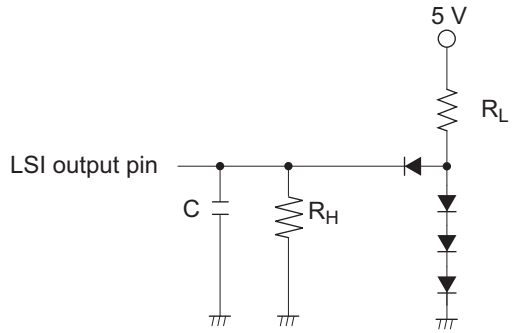
Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Applicable Pins: SCL1-0, SDA1-0

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	V_T^-	1.0	—	—	V	
	V_T^+	—	—	$V_{CC} \times 0.7$		
	$V_T^+ - V_T^-$	0.4	—	—		V_{CC} = 4.5 V to 5.5 V
Input high voltage	V_{IH}	$V_{CC} \times 0.7$	—	V_{CC} + 0.5	V	
Input low voltage	V_{IL}	– 0.5	—	$V_{CC} \times 0.3$	V	
Output low voltage	V_{OL}	—	—	0.7	V	I_{OL} = 8 mA, V_{CC} = 4.5 V to 5.5 V
		—	—	0.4		I_{OL} = 3 mA, V_{CC} = 4.5 V to 5.5 V
		—	—	0.4		I_{OL} = 1.6 mA, V_{CC} = 3.3 V to 5.5 V
Input capacitance	C_{in}	—	—	20	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25 °C
Three-state leakage current (off state)	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 5.5 V
SCL, SDA, output fall time	t_{of}	20 + 0.1Cb	—	250	ns	

23.3 AC Characteristics

Figure 23.1 show, the test conditions for the AC characteristics.



C = 50 pF: Ports 10 to 13, A to F
(In case of expansion bus control signal output pin setting)
C = 30 pF: All ports
R_L = 2.4 kΩ
R_H = 12 kΩ
Input/output timing measurement levels
- Low level: 0.8 V
- High level: 2.0 V

Figure 23.1 Output Load Circuit

23.3.1 Clock Timing

Table 23.5 lists the clock timing

Table 23.5 Clock Timing

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V,
V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

		Condition			
		20MHz			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Test Conditions**
Clock cycle time	t_{cyc}	50	250	ns	Figure 23.2
Clock high pulse width	t_{CH}	15	—	ns	
Clock low pulse width	t_{CL}	15	—	ns	
Clock rise time	t_{Cr}	—	10	ns	
Clock fall time	t_{Cf}	—	10	ns	
Clock oscillator settling time at reset (crystal)	t_{OSC1}	20	—	ms	Figure 23.3
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	8	—	ms	Figure 20.3
External clock output stabilization delay time	t_{DEXT}	2	—	ms	Figure 23.3
32-kHz clock oscillation settling time	t_{OSC3}	—	2	s	
Subclock oscillator frequency	f_{SUB}	32.768		kHz	
Subclock (ϕ_{SUB}) cycle time	t_{SUB}	30.5		µs	

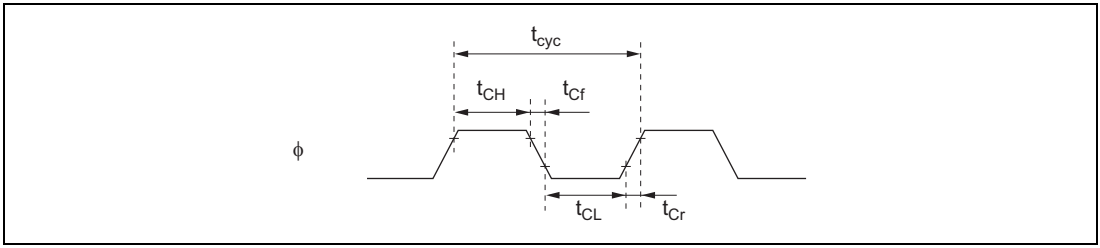


Figure 23.2 System Clock Timing

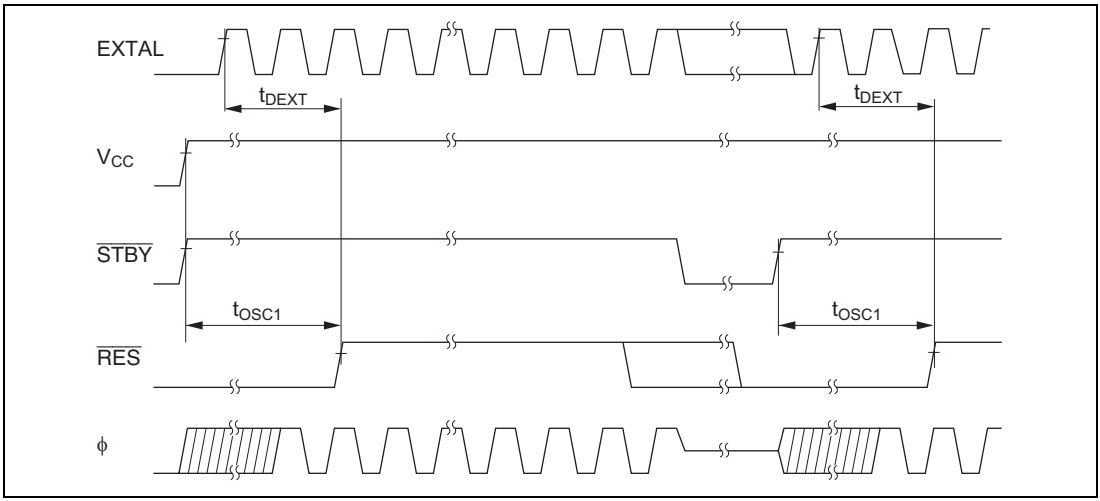


Figure 23.3 Oscillator Settling Timing

23.3.2 Control Signal Timing

Table 23.6 lists the control signal timing.

Table 23.6 Control Signal Timing

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V,
V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

		Condition			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Test Conditions**
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figure 23.4
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 23.5
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	150	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	ns	
$\overline{IRQ}$ pulse width (exiting software standby mode)	t_{IRQW}	200	—	ns	

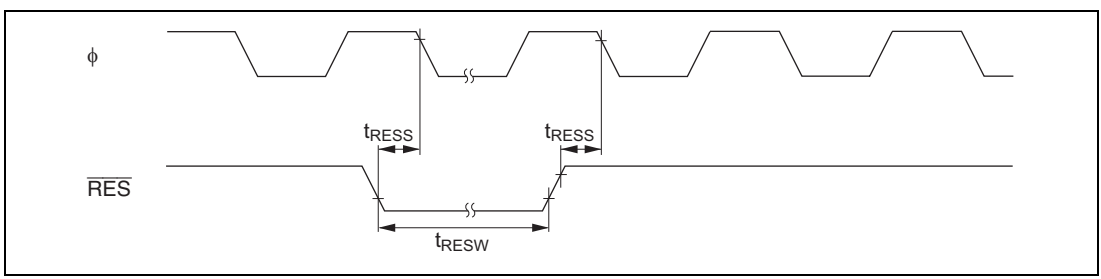


Figure 23.4 Reset Input Timing

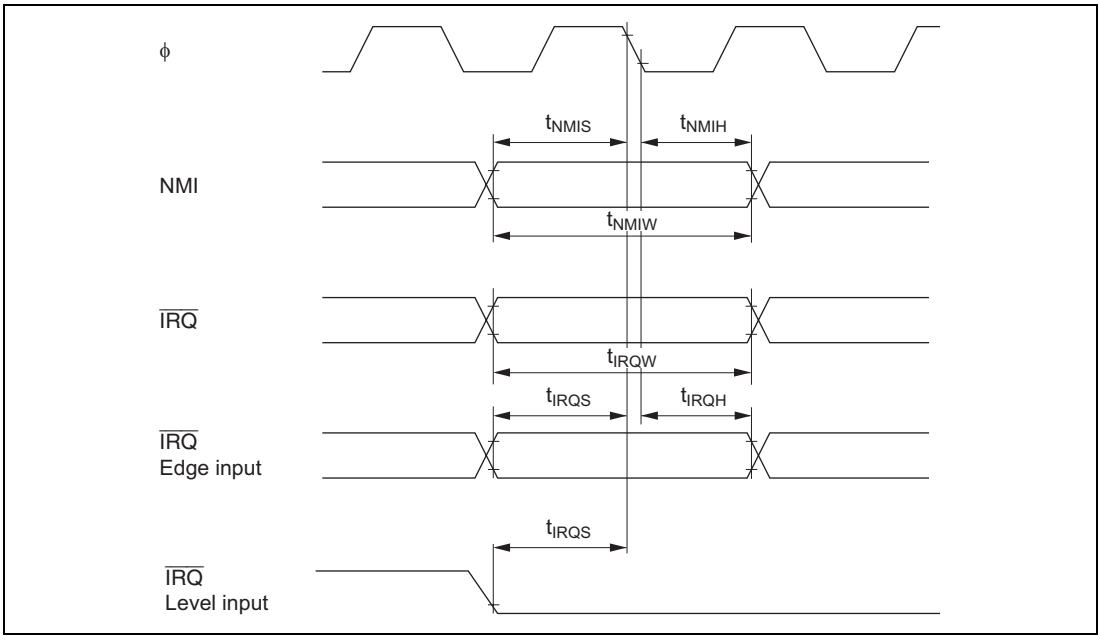


Figure 23.5 Interrupt Input Timing

23.3.3 Bus Timing

Table 23.7 lists the bus timing.

Table 23.7 Bus Timing

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V,
V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Condition		Unit	Test Conditions
		Min.	Max.		
Address delay time	t_{AD}	—	35	ns	Figure 23.6 to Figure 23.10
Address setup time	t_{AS}	$0.5 \times t_{cyc} - 20$	—	ns	
Address hold time	t_{AH}	$0.5 \times t_{cyc} - 15$	—	ns	
$\overline{AS}$ delay time	t_{ASD}	—	20	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	20	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	20	ns	
Read data setup time	t_{RDS}	20	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Read data access time1	t_{ACC1}	—	$1.0 \times t_{cyc} - 48$	ns	
Read data access time2	t_{ACC2}	—	$1.5 \times t_{cyc} - 45$	ns	
Read data access time3	t_{ACC3}	—	$2.0 \times t_{cyc} - 45$	ns	
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 45$	ns	
Read data access time 5	t_{ACC5}	—	$3.0 \times t_{cyc} - 50$	ns	
$\overline{WR}$ delay time 1	t_{WRD1}	—	20	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	20	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 20$	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 20$	—	ns	
Write data delay time	t_{WDD}	—	30	ns	
Write data setup time	t_{WDS}	$0.5 \times t_{cyc} - 20$	—	ns	
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 10$	—	ns	

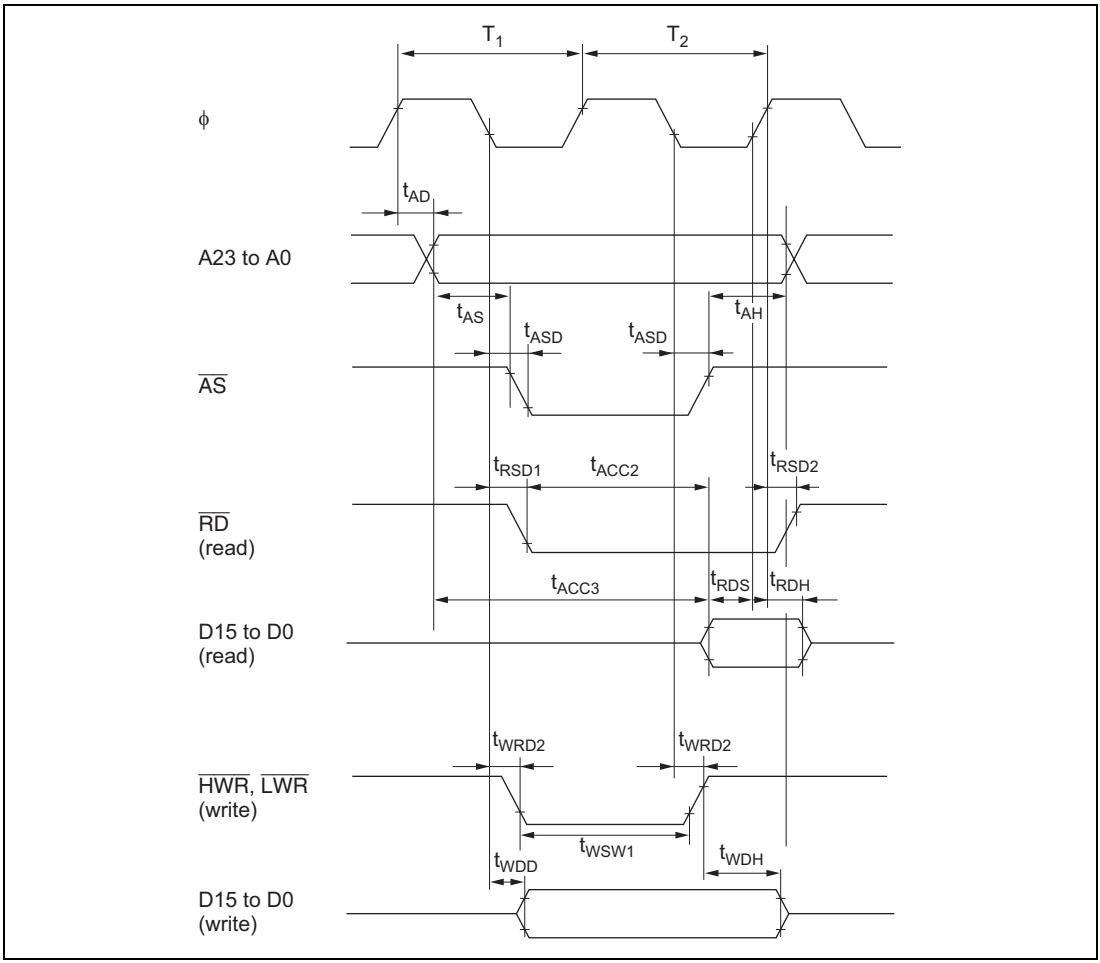


Figure 23.6 Basic Bus Timing (Two-State Access)

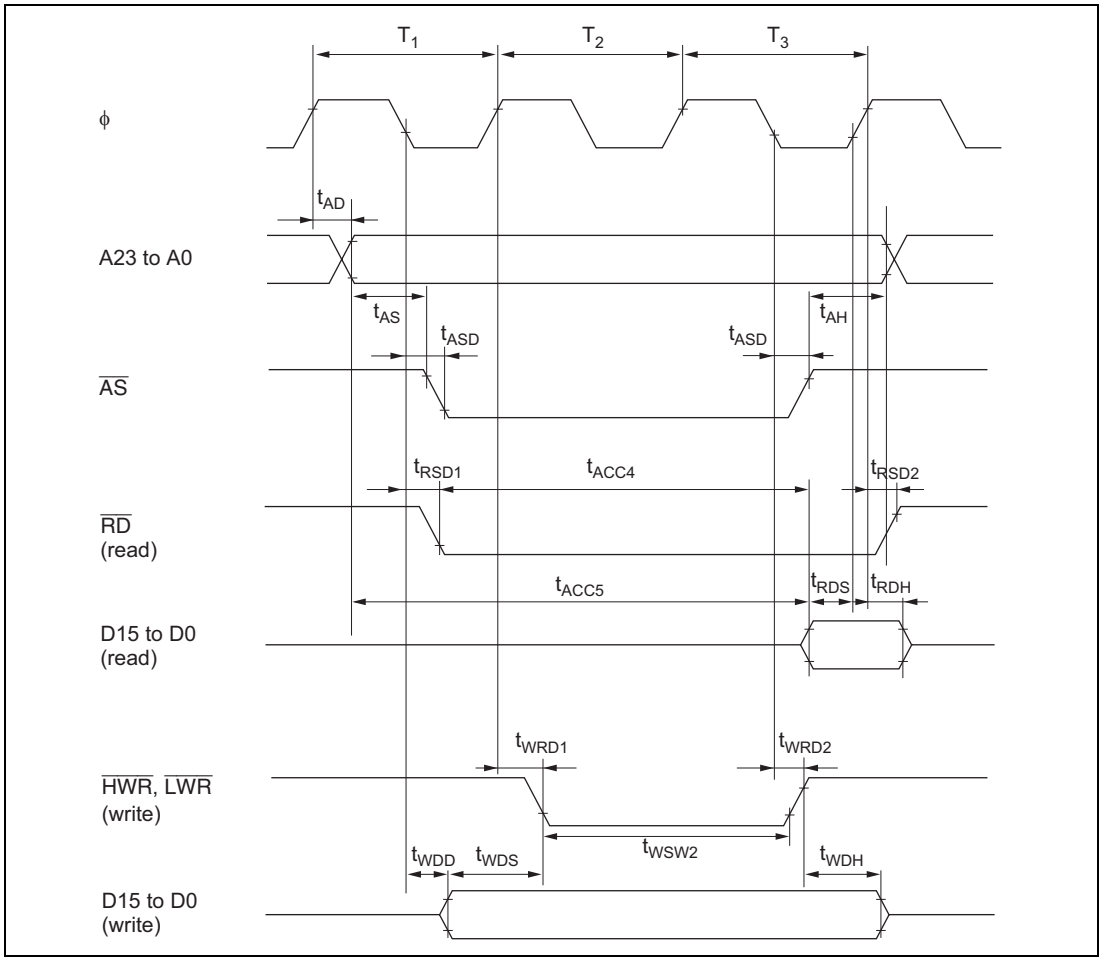


Figure 23.7 Basic Bus Timing (Three-State Access)

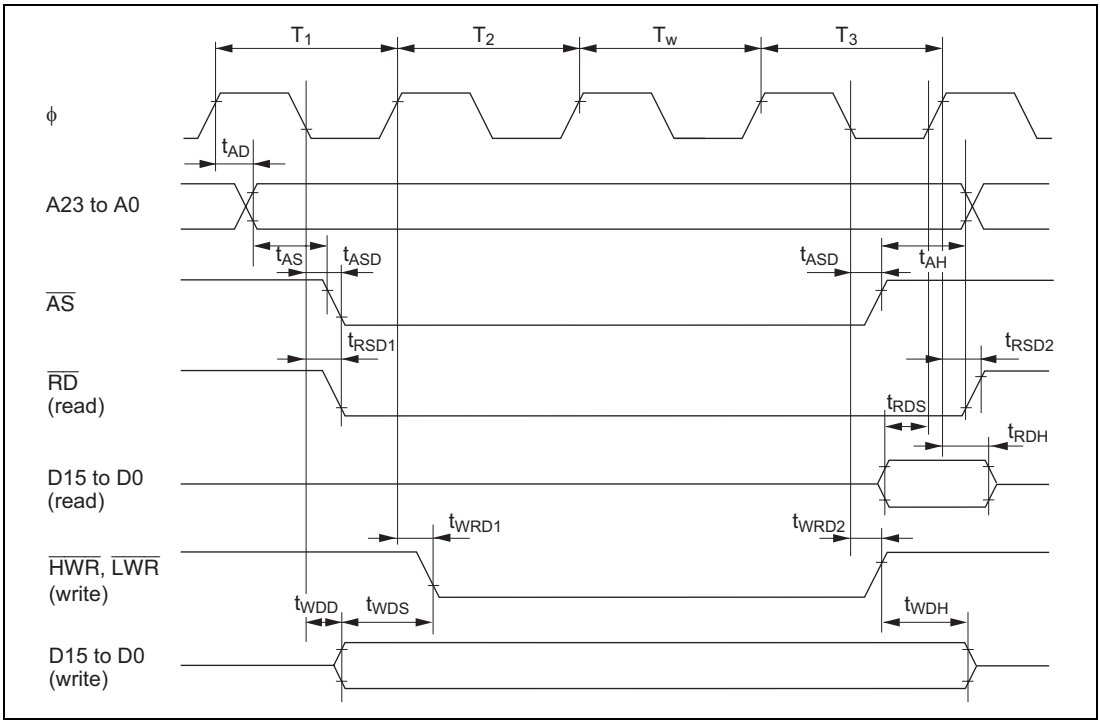


Figure 23.8 Basic Bus Timing (Three-State Access with One Wait State)

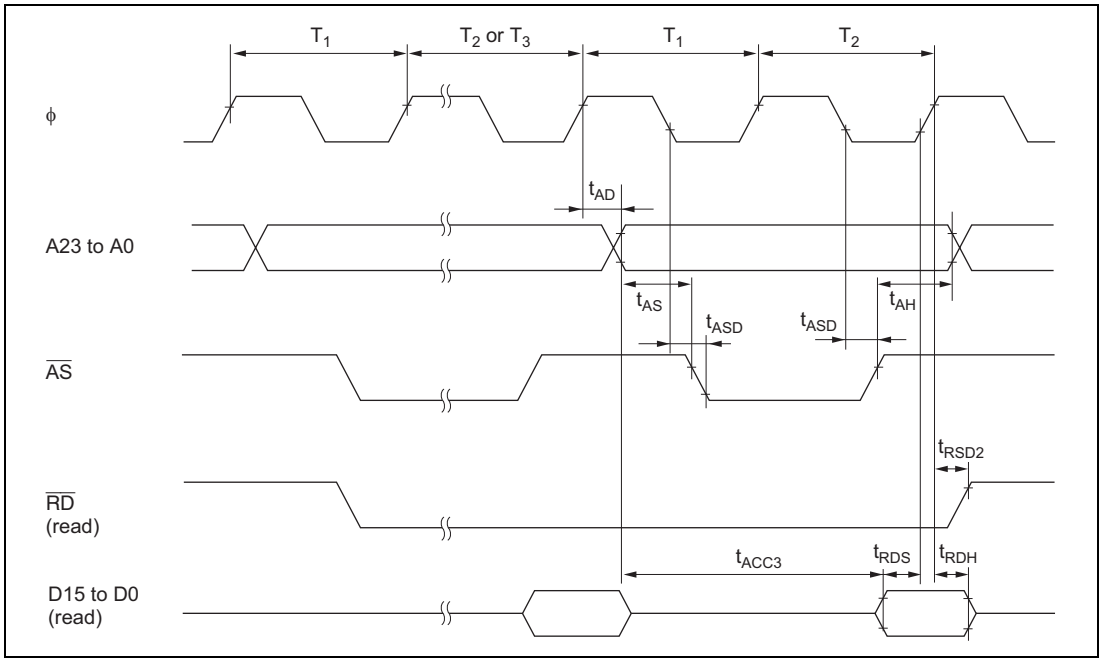


Figure 23.9 Burst ROM Access Timing (Two-State Access)

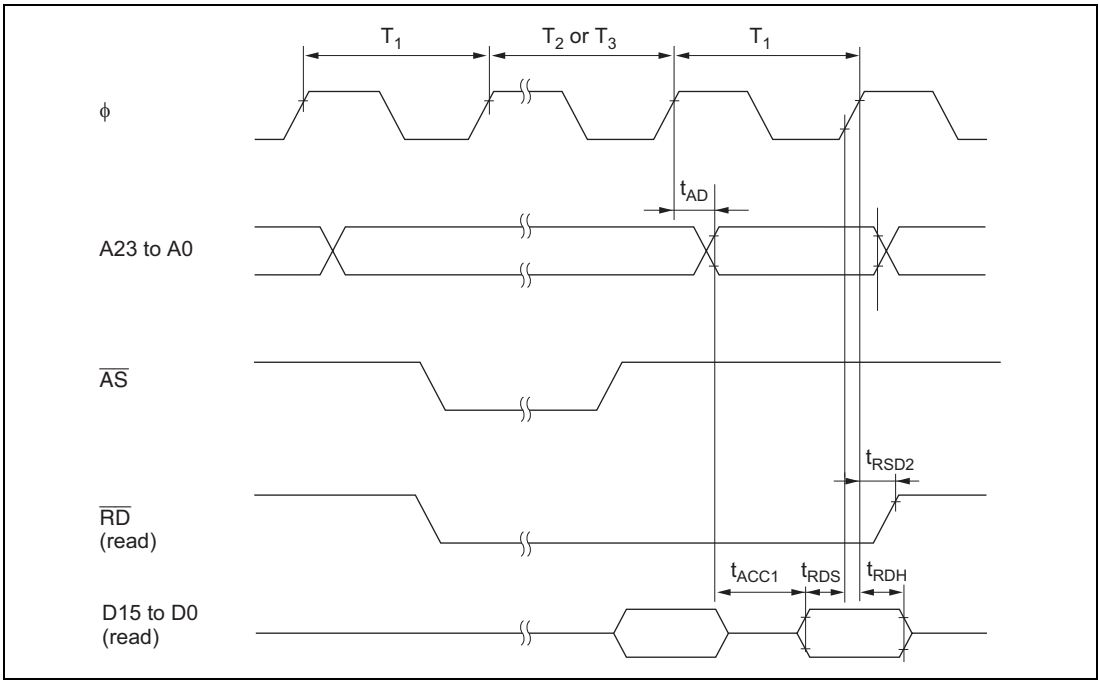


Figure 23.10 Burst ROM Access Timing (One-State Access)

23.3.4 Timing of On-Chip Supporting Modules

Table 23.8 lists the timing of on-chip supporting modules.

Table 23.8 Timing of On-Chip Supporting Modules

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V,
V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item			Symbol	Condition Min.	Condition Max.	Unit	Test Conditions
I/O port	Output data delay time		t_{PWD}	—	50	ns	Figure 23.11 Figure 23.12
	Output data delay time 2		t_{PWD2}	—	50		
	Input data setup time		t_{PRS}	30	—		
	Input data hold time		t_{PRH}	30	—		
PPG	Pulse output delay time		t_{POD}	—	50	ns	Figure 23.13
TPU	Timer output delay time		t_{TOCD}	—	50	ns	Figure 23.14
	Timer input setup time		t_{TICS}	30	—		
	Timer clock input setup time		t_{TCKS}	30	—	ns	Figure 23.15
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—		
PWM	Pulse output delay time		t_{MPWMOD}	—	50	ns	Figure 23.16
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 23.17
		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		t_{TXD}	—	50	ns	Figure 23.18
	Receive data setup time (synchronous)		t_{RXS}	50	—		
	Receive data hold time (synchronous)		t_{RXH}	50	—		
A/D converter	Trigger input setup time		t_{TRGS}	50	—	ns	Figure 23.19

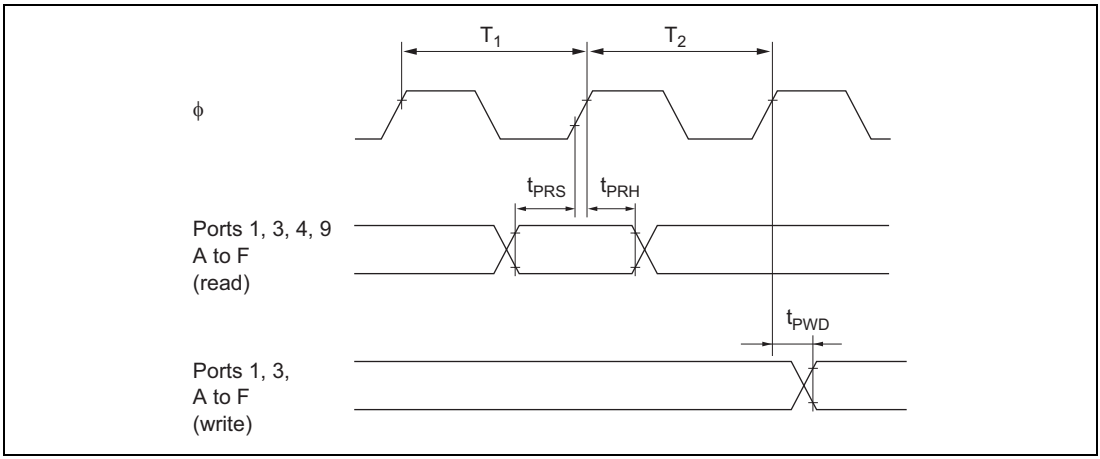


Figure 23.11 I/O Port Input/Output Timing (Ports 1, 3, 4, 9, A to F)

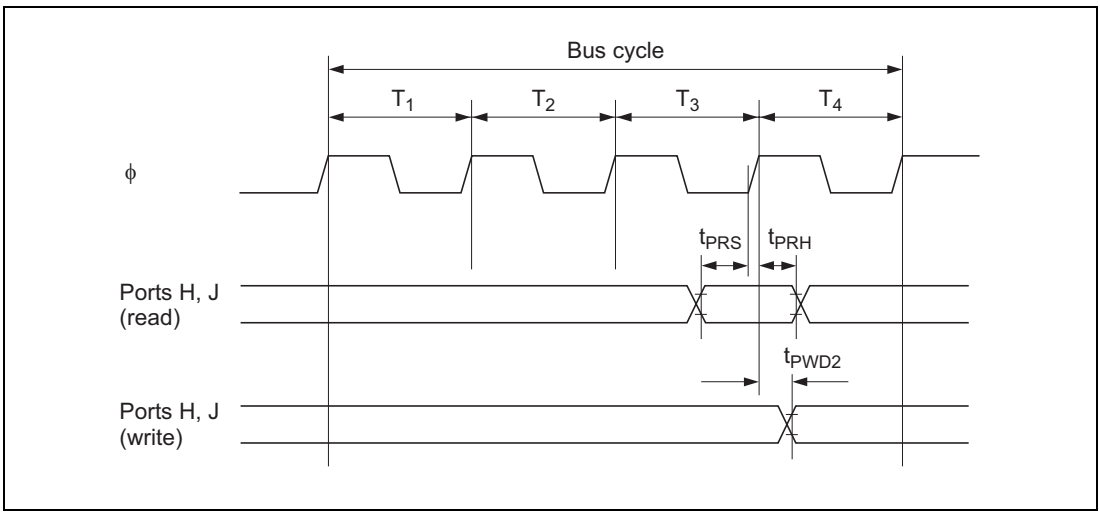


Figure 23.12 I/O Port (Ports H and J) Input/Output Timing

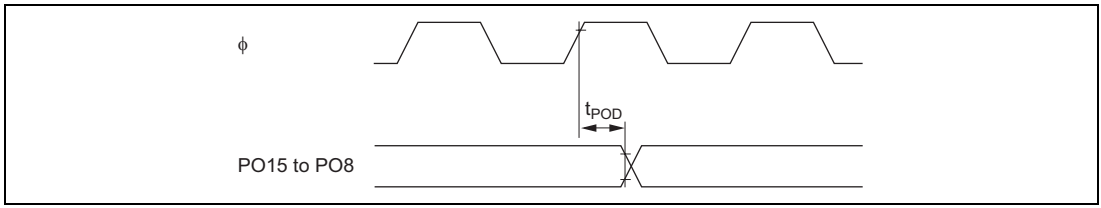


Figure 23.13 PPG Output Timing*

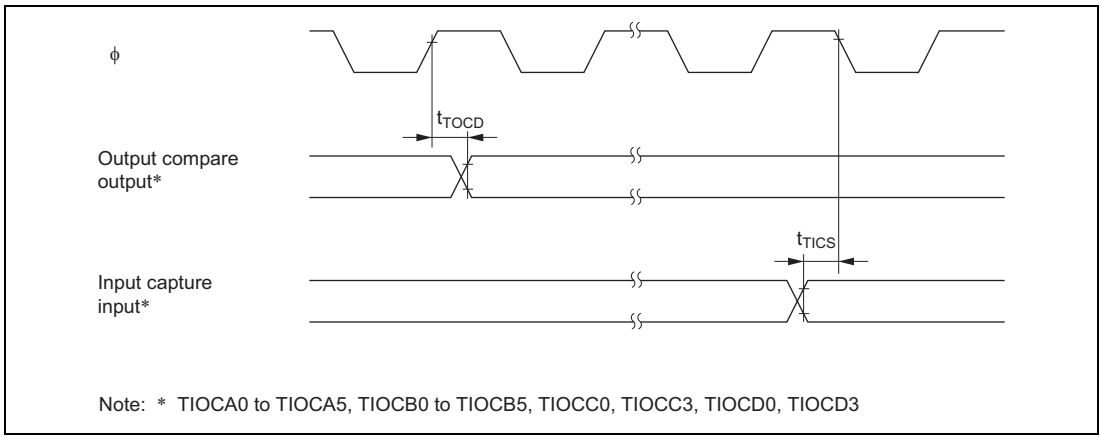


Figure 23.14 TPU Input/Output Timing

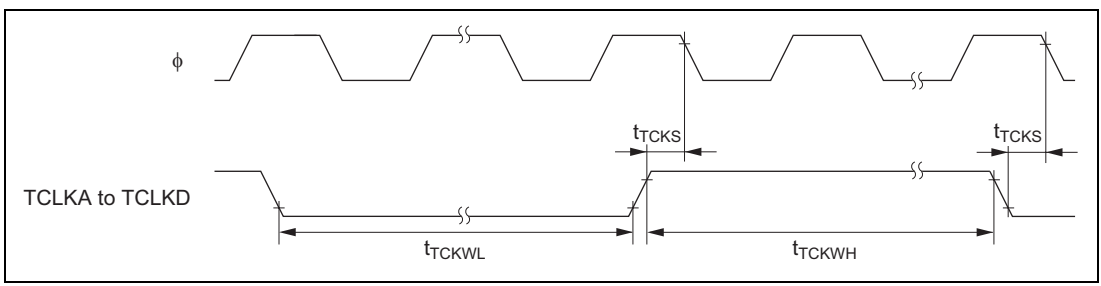


Figure 23.15 TPU Clock Input Timing

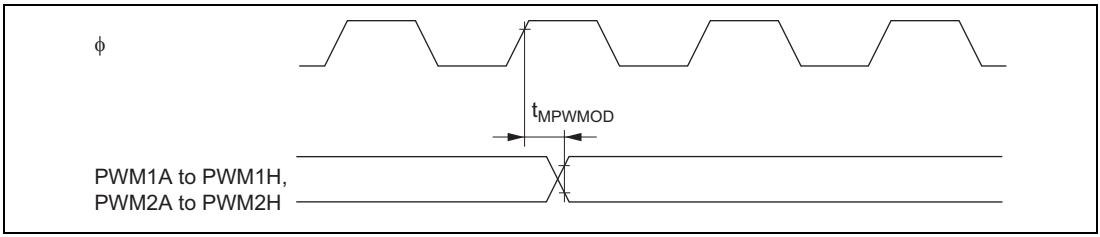


Figure 23.16 Motor Control PWM Output Timing

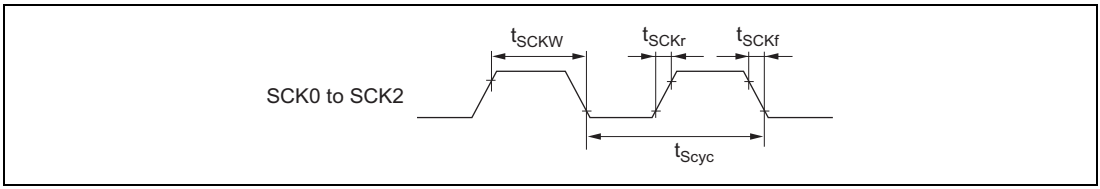


Figure 23.17 SCK Clock Input Timing

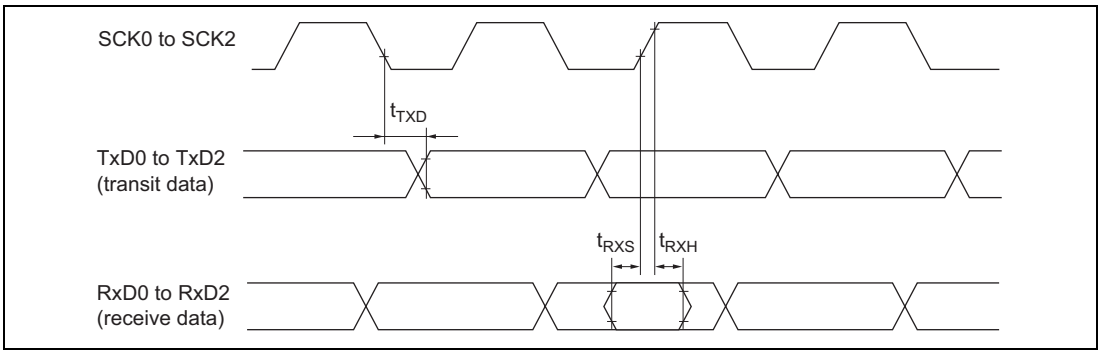


Figure 23.18 SCI Input/Output Timing (Clock Synchronous Mode)

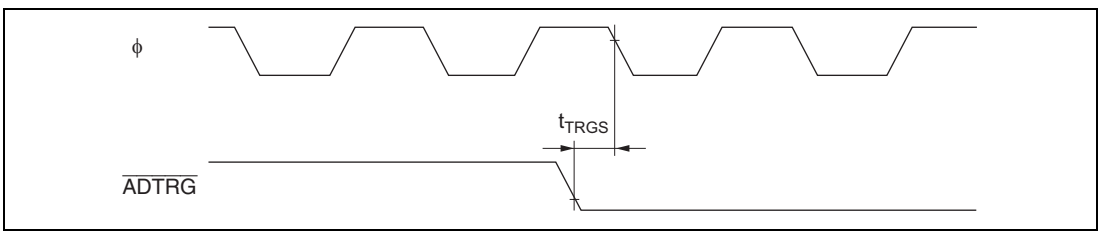


Figure 23.19 A/D Converter External Trigger Input Timing

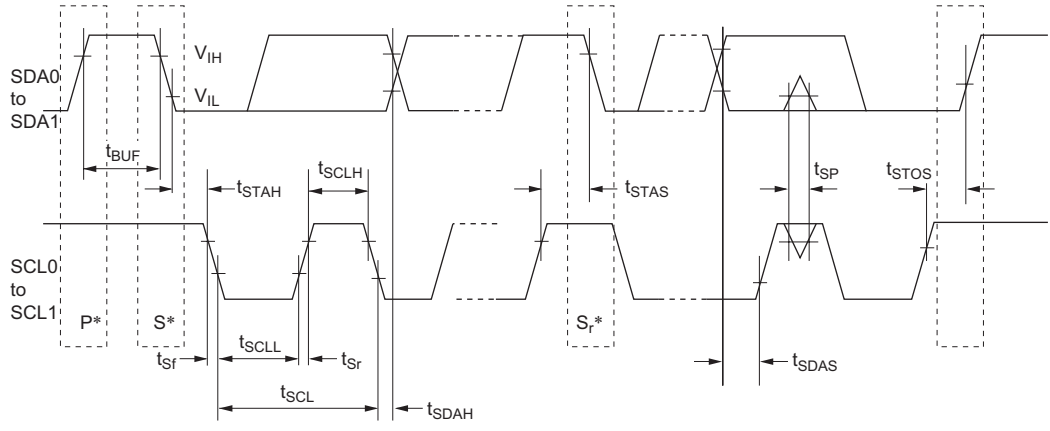
23.4 I²C Bus Interface Timing

Table 23.9 I²C Bus Interface Timing

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, ϕ = 5 MHz to maximum operating frequency, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition				
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Unit**	**Notes**
SCL input cycle time	t_{SCL}	$12t_{cyc}$	—	—	ns	Figure 23.20
SCL input high pulse width	t_{SCLH}	$3t_{cyc}$	—	—	ns	
SCL input low pulse width	t_{SCLL}	$5t_{cyc}$	—	—	ns	
SCL, SDA input rise time	t_{Sr}	—	—	$7.5t_{cyc}$ *	ns	
SCL, SDA input fall time	t_{Sf}	—	—	300	ns	
SCL, SDA input spike pulse elimination 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	
Stop condition input setup time	t_{STOS}	$3t_{cyc}$	—	—	ns	
Data input setup time	t_{SDAS}	$0.5t_{cyc}$	—	—	ns	
Data input hold time	t_{SDAH}	0	—	—	ns	
SCL, SDA capacitive load	C_b	—	—	400	pF	

Note: $17.5t_{cyc}$ can be set according to the clock selected for use by the I²C module. For details, see section 14.5, Usage Notes.



Note: * S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Retransmission start condition

Figure 23.20 I^2C Bus Interface Input/Output Timing

23.5 A/D Conversion Characteristics

Table 23.10 lists the A/D conversion characteristics.

Table 23.10 A/D Conversion Characteristics

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V,
V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

	Condition			
Item	**Min.**	**Typ.**	**Max.**	**Unit**
Resolution	10	10	10	bits
Conversion time	10	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±3.5	LSB
Offset error	—	—	±3.5	LSB
Full-scale error	—	—	±3.5	LSB
Quantization	—	±0.5	—	LSB
Absolute accuracy	—	—	±4.0	LSB

23.6 D/A Conversion Characteristics

Table 23.11 shows the D/A conversion characteristics.

Table 23.11 D/A Conversion Characteristics

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V,
V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

	Condition				
Item	**Min.**	**Typ.**	**Max.**	**Unit**	**Test Conditions**
Resolution	8	8	8	bits	
Conversion time	—	—	10	µs	20-pF capacitive load
Absolute accuracy	—	±1.5	±2.0	LSB	2-MΩ resistive load
	—	—	±1.5	LSB	4-MΩ resistive load

23.7 Flash Memory Characteristics

Table 23.12 shows the flash memory characteristics.

Table 23.12 Flash Memory Characteristics

Conditions: V_{CC} =4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V,
V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$, AV_{SS} = 0 V
T_a = 0 to +75°C (Programming/erasing operating temperature range: regular specification)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Condition
Programming time*1 *2 *4		t_P	—	10	200	ms/ 128 bytes	
Erase time*1 *3 *5		t_E	—	100	1200	ms/block	
Reprogramming count		N_{WEC}	—	—	100	Times	
Programming	Wait time after SWE bit setting*1	t_{sswe}	1	1	—	µs	
	Wait time after PSU bit setting*1	t_{spsu}	50	50	—	µs	
	Wait time after P bit setting*1 *4	t_{sp30}	28	30	32	µs	Programming time wait
		t_{sp200}	198	200	202	µs	Programming time wait
		t_{sp10}	8	10	12	µs	Additional-programming time wait
	Wait time after P bit clear*1	t_{cp}	5	5	—	µs	
	Wait time after PSU bit clear*1	t_{cpsu}	5	5	—	µs	
	Wait time after PV bit setting*1	t_{spv}	4	4	—	µs	
	Wait time after H'FF dummy write*1	t_{spvr}	2	2	—	µs	
	Wait time after PV bit clear*1	t_{cpv}	2	2	—	µs	
	Wait time after SWE bit clear*1	t_{cswe}	100	100	—	µs	
	Maximum programming count*1 *4	N	—	—	1000	Times	
Erase	Wait time after SWE bit setting*1	t_{sswe}	1	1	—	µs	
	Wait time after ESU bit setting*1	t_{sesu}	100	100	—	µs	
	Wait time after E bit setting*1 *5	t_{se}	10	10	100	ms	Erase time wait
	Wait time after EV bit setting*1	t_{sev}	20	20	—	µs	

Item		Symbol	Min.	Typ.	Max.	Unit	Test Condition
Erase	Wait time after E bit clear*1	t_{ce}	10	10	—	µs	
	Wait time after ESU bit clear*1	t_{cesu}	10	10	—	µs	
	Wait time after H'FF dummy write*1	t_{sevr}	2	2	—	µs	
	Wait time after EV bit clear*1	t_{cev}	4	4	—	µs	
	Wait time after SWE bit clear*1	t_{cswe}	100	100	—	µs	
	Maximum erase count*1 *5	N	12	—	120	Times	

Notes: 1. Make each time setting in accordance with the program/program-verify flowchart or erase/erase-verify flowchart.

2. Programming time per 128 bytes (Shows the total period for which the P-bit in the flash memory control register (FLMCR1) is set. It does not include the programming verification time)
3. Block erase time (Shows the total period for which the E-bit in FLMCR1 is set. It does not include the erase verification time)
4. To specify the maximum programming time value (t_p (max.)) in the 128-byte programming algorithm, set the max. value (1000) for the maximum programming count (N).

   The wait time after P bit setting should be changed as follows according to the value of the programming counter (n).

   Programming counter (n) = 1 to 6: t_{sp30} = 30 µs

   Programming counter (n) = 7 to 1000: t_{sp200} = 200 µs

   [In additional programming]

   Programming counter (n)= 1 to 6: t_{sp10} = 10 µs
5. For the maximum erase time (t_E (max.)), the following relationship applies between the wait time after E bit setting (t_{se}) and the maximum erase count (N):

   t_E (max.) = Wait time after E bit setting (t_{se}) x maximum erase count (N)

   To set the maximum erase time, the values of (t_{se}) and (N) should be set so as to satisfy the above formula.

   Examples: When t_{se} = 100 [ms], N = 12 times

   When t_{se} = 10 [ms], N = 120 times

23.8 Usage Note

Although both the F-ZTAT and mask ROM versions fully meet the electrical specifications listed in this manual, there may be differences in the actual values of the electrical characteristics, operating margins, noise margins, and so forth, due to differences in the fabrication process, the on-chip ROM, and the layout patterns.

Therefore, if a system is evaluated using the F-ZTAT version, a similar evaluation should also be performed using the mask ROM version.

Appendix

A. I/O Port States in Each Operating State

Port Name	MCU Operating Mode	Power-on Reset	Hardware Standby Mode	Software Standby Mode	Program Execution State, Sleep Mode
Port 1	4 and 5	T	T	P10 to P13	P10 to P13
	6			[Address output, OPE = 0]	[Address output]
				T	A20 to A23
				[Address output, OPE = 1]	[Other than the above]
				Keep	I/O port
				[Other than the above]	P14 to P17
				Keep	I/O port
				P14 to P17	
				Keep	
	7			Keep	P10 to P17
					I/O port
Port 3	4 to 7	T	T	Keep	I/O port
Port 4	4 to 7	T	T	T	Input port
Port 9	4 to 7	T	T	T	Input port
Port A	4 and 5	L	T	[Address output, OPE = 0]	[Address output]
	6	T		T	A19 to A17
				[Address output, OPE = 1]	[Other than the above]
				Keep	I/O port
				[Other than the above]	
				Keep	
	7	T	T	Keep	I/O port

Port Name	MCU Operating Mode		Power-on Reset	Hardware Standby Mode	Software Standby Mode	Program Execution State, Sleep Mode
Port B	4 and 5		L	T	[Address output, OPE = 0]	[Address output]
	6		T		T	A15 to A8
					[Address output, OPE = 1]	[Other than the above]
					Keep	I/O port
					[Other than the above]	
					Keep	
	7		T	T	Keep	I/O port
Port C	4 and 5		L	T	[OPE = 0]	A7 to A0
					T	
					[OPE = 1]	
					Keep	
	6		T	T	[DDR = 1, OPE = 0]	[DDR = 1]
					T	A7 to A0
					[DDR = 1, OPE = 1]	[DDR = 0]
					Keep	Input port
					[DDR = 0]	
					Keep	
	7		T	T	Keep	I/O port
Port D	4 to 6		T	T	T	Data bus
	7		T	T	Keep	I/O port
Port E	4 to 6	8-bit bus	T	T	Keep	I/O port
		16-bit bus	T	T	T	Data bus
	7		T	T	Keep	I/O port
PF7/ϕ	4 to 6		Clock output	T	[DDR = 0]	[DDR = 0]
					T	T
	7		T		[DDR = 1]	[DDR = 1]
					H	Clock output

Port Name	MCU Operating Mode	Power-on Reset	Hardware Standby Mode	Software Standby Mode	Program Execution State, Sleep Mode
PF6/$\overline{AS}$	4 to 6	H	T	[OPE = 0] T [OPE = 1] H	AS
	7	T	T	Keep	I/O port
PF5/$\overline{RD}$, PF4/$\overline{HWR}$	4 to 6	H	T	[OPE = 0] T [OPE = 1] H	RD, $\overline{HWR}$
	7	T	T	Keep	I/O port
PF3/$\overline{LWR}$	4 to 6	H	T	[OPE = 0] T [OPE = 1] H	LWR
	7	T	T	Keep	I/O port
PF0	4 to 7	T	T	Keep	I/O port
Port H	4 to 7	T	T	Keep	I/O port
Port J	4 to 7	T	T	Keep	I/O port

[Legend]

H: High level

L: Low level

T: High impedance

Keep: Input port becomes high-impedance, and output port retains state

Port: Depends on the port setting (input becomes high-impedance)

DDR: Data direction register

OPE: Output port enable

WAITE: Wait input enable

B. Product Code Lineup

Product Type		Product Code	Mark Code	Package (Renesas Package Code)
H8S/2607 group	F-ZTAT version	HD64F2607	HD64F2607WF20 (Normal spec)	128-pin QFP PRQP0128KB-A (FP-128B/FP-128BV)
			HD64F2607WF20W (Wide Temperature Range spec)	
			HD64F2607WF20V (Normal spec)	
			HD64F2607WF20WV (Wide Temperature Range spec)	
	Masked ROM version	HD6432607	HD6432607W(***)F (Normal spec)	
			HD6432607W(***)FW (Wide Temperature Range spec)	
			HD6432607W(***)FV (Normal spec)	
			HD6432607W(***)FWV (Wide Temperature Range spec)	
		HD6432606	HD6432606W(***)F (Normal spec)	
			HD6432606W(***)FW (Wide Temperature Range spec)	
			HD6432606W(***)FV (Normal spec)	
			HD6432606W(***)FWV (Wide Temperature Range spec)	
		HD6432605	HD6432605W(***)F (Normal spec)	
			HD6432605W(***)FW (Wide Temperature Range spec)	
			HD6432605W(***)FV (Normal spec)	
			HD6432605W(***)FWV (Normal spec)	

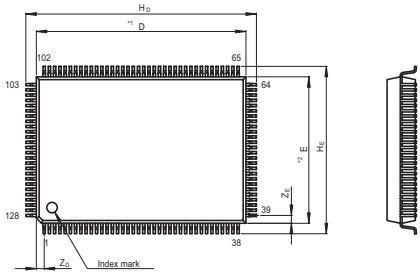
[Legend]

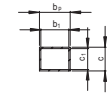
(***): ROM code

C. Package Dimensions

The package dimensions that are shown in the Renesas Semiconductor Package Data Book have priority.

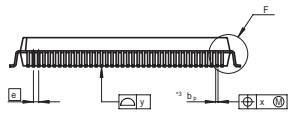
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-QFP128-14x20-0.50	PRQP0128KB-A	FP-128B/FP-128BV	1.7g

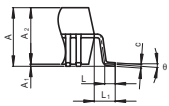




Terminal cross section

NOTE)
1. DIMENSIONS"*1"AND"*2" DO NOT INCLUDE MOLD FLASH
2. DIMENSION"*3"DOES NOT INCLUDE TRIM OFFSET.





Detail F

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	20	—
E	—	14	—
A_2	—	2.70	—
H_D	21.8	22.0	22.2
H_E	15.8	16.0	16.2
A	—	—	3.15
A_1	0.00	0.10	0.25
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.12	0.17	0.22
c_1	—	0.15	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.10
y	—	—	0.10
Z_D	—	0.75	—
Z_E	—	0.75	—
L	0.3	0.5	0.7
L_1	—	1.0	—

Figure C.1 FP-128B Package Dimensions

Index

S

T

U

V

W

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2607 Group

Publication Date:	Rev.1.00, Jan. 29, 2008
Published by:	Sales Strategic Planning Div. Renesas Technology Corp.
Edited by:	Customer Support Department Global Strategic Communication Div. Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

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

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

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

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

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

H8S/2607 Group
Hardware Manual

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan