To all our customers

Information regarding change of names mentioned within this document, to Renesas Technology Corp.

On April 1st 2003 the following semiconductor operations were transferred to Renesas Technology Corporation: operations covering microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.).
Accordingly, although Hitachi, Hitachi, Ltd., Hitachi Semiconductors, and other Hitachi brand names are mentioned in the document, these names have all been changed to Renesas Technology Corporation.
Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.
Thank you for your understanding.

Renesas Technology Home Page: www.renesas.com

Renesas Technology Corp.
April 1, 2003

RENESAS
Renesas Technology Corp.

Hitachi 16-Bit Single-Chip Microcomputer

H8S/2615 Series

H8S/2615
HD64F2615
HD6432615

Hardware Manual

— Preliminary —

HITACHI

ADE-602-309

Rev. 0.5
03/10/03
Hitachi, Ltd.

Shaping intelligence
HITACHI SEMICONDUCTOR

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

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/2615 Series are single-chip microcomputers made up of the high-speed H8S/2600 CPU as its core, and the peripheral functions required to configure a system. The H8S/2600 CPU has an instruction set that is compatible with the H8/300 and H8/300H CPUs.

This LSI is equipped with ROM and RAM memory, a 16-bit timer pulse unit (TPU), a watchdog timer (WDT), a serial communication interface (SCI), a Hitachi controller area network (HCAN), an A/D converter, and I/O ports as on-chip peripheral modules required for system configuration. This LSI is suitable for use as an embedded microcomputer for high-level control systems. A single-power flash memory (F-ZTAT™) version is available for this LSI's ROM. This provides flexibility as it can be reprogrammed in no time to cope with all situations from the early stages of mass production to full-scale mass production. This is particularly applicable to application devices with specifications that will most probably change.

Note: * F-ZTAT™ is a trademark of Hitachi, Ltd.

Target Users: This manual was written for users who will be using the H8S/2615 Series 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/2615 Series to the target users.
Refer to the H8S/2600 Series, H8S/2000 Series Programming Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip

  Read 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 Programming 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 17, 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.hitachisemiconductor.com/

H8S/2615 Series manuals:

Manual Title	ADE No.
H8S/2615 Series Hardware Manual	This manual
H8S/2600 Series, H8S/2000 Series Programming Manual	ADE-602-083

User's manuals for development tools:

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

Contents

Section 3 MCU Operating Modes 45

Section 4 Exception Handling 51

Section 5 Interrupt Controller 61

Figures

Section 1 Overview

Section 2 CPU

Section 3 MCU Operating Modes

Section 4 Exception Handling

Section 5 Interrupt Controller

Section 6 Bus Controller

Section 8 16-Bit Timer Pulse Unit (TPU)

Section 9 Watchdog Timer (WDT)

Section 10 Serial Communication Interface (SCI)

Section 11 Hitachi Controller Area Network (HCAN)

Section 12 A/D Converter

Section 14 ROM

Section 15 Clock Pulse Generator

Section 16 Power-Down Modes

Section 18 Electrical Characteristics

Appendix

Tables

Section 2 CPU

Section 3 MCU Operating Modes

Section 4 Exception Handling

Section 5 Interrupt Controller

Section 7 I/O Ports

Section 8 16-Bit Timer Pulse Unit (TPU)

Section 9 Watchdog Timer (WDT)

Section 10 Serial Communication Interface (SCI)

Section 11 Hitachi Controller Area Network (HCAN)

Section 12 A/D Converter

Section 14 ROM

Section 15 Clock Pulse Generator

Section 16 Power-Down Modes

Section 18 Electrical Characteristics

Section 1 Overview

1.1 Features

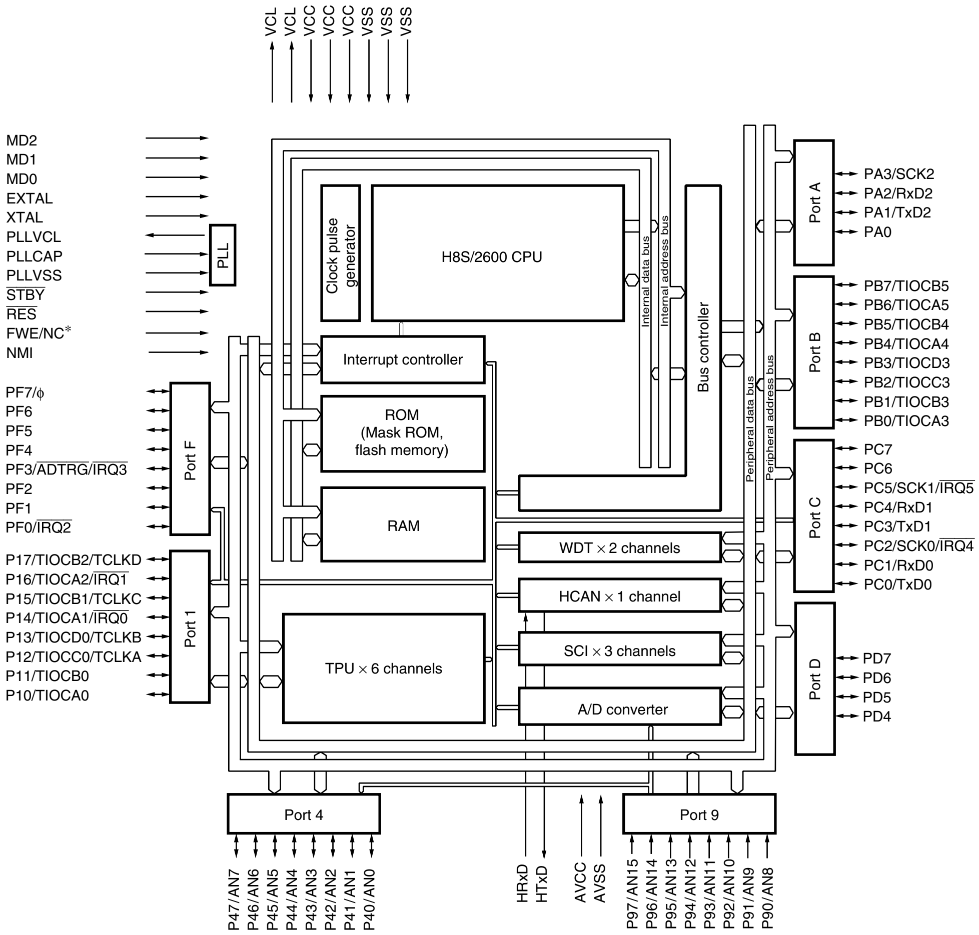
- High-speed H8S/2600 central processing unit with an internal 16-bit architecture
  Upward-compatible with H8/300 and H8/300H CPUs on an object level
  Sixteen 16-bit general registers
  69 basic instructions
- Various peripheral functions
  16-bit timer pulse unit (TPU)
  Watchdog timer
  Asynchronous or clocked synchronous serial communication interface (SCI)
  Hitachi controller area network (HCAN)
  10-bit A/D converter
  Clock pulse generator
- On-chip memory

ROM	Model	ROM	RAM	Remarks
F-ZTAT Version	HD64F2615	64 kbytes	4 kbytes	
Masked ROM version	HD6432615	64 kbytes	4 kbytes	In planning

- General I/O ports
  I/O pins: 39
  Input-only pins: 17
- Supports various power-down modes
- Compact package

Package	Code	Body Size	Pin Pitch
QFP-80Q	FP-80Q	14.0 × 14.0 mm	0.65 mm

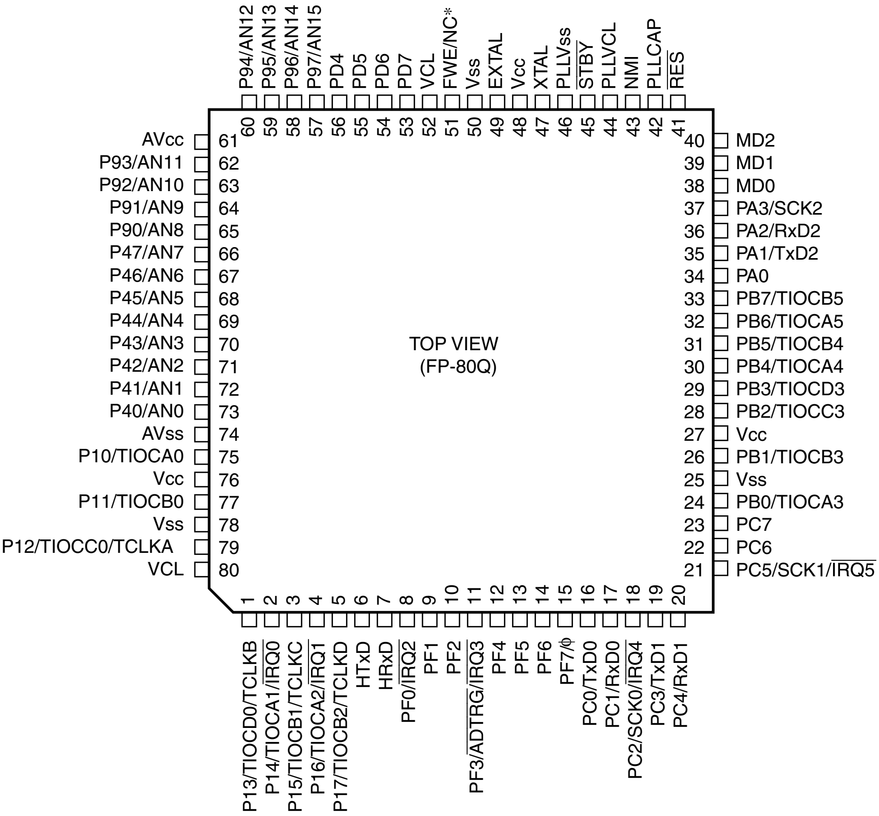
1.2 Internal Block Diagram



Note: * The FWE pin is available only in the flash memory version.
The NC pin is available only in the masked ROM version.

Figure 1.1 Internal Block Diagram

1.3 Pin Arrangement



Note: * The FWE pin is available only in the flash memory version.
The NC pin is available only in the masked ROM version.

Figure 1.2 Pin Arrangement

1.4 Pin Functions

Type	Symbol	Pin No.	I/O	Function
Power supply	VCC	27 48 76	Input	Power supply pins. Connect all these pins to the system power supply.
	VSS	25 50 78	Input	Ground pins. Connect all these pins to the system power supply (0 V).
	VCL	52 80	Output	External capacitance pin for internal step-down power supply. Connect these pins to VSS via a 0.1-μF capacitor (placed close to the pins).
Clock	PLLVCL	44	Output	External capacitance pin for internal step-down power supply for an on-chip PLL oscillator. Connect this pin to PLLVSS via a 0.1-μF capacitor (placed close to the pins).
	PLLVSS	46	Input	On-chip PLL oscillator ground pin.
	PLLCAP	42	Output	External capacitance pin for an on-chip PLL oscillator.
	XTAL	47	Input	For connection to a crystal resonator. For examples of crystal resonator connection and external clock input, see section 15, Clock Pulse Generator.
	EXTAL	49	Input	For connection to a crystal resonator. An external clock can be input to the EXTAL pin. For examples of crystal resonator connection and external clock input, see section 15, Clock Pulse Generator.
	ϕ	15	Output	Supplies the system clock to external devices.
Operating mode control	MD2 MD1 MD0	40 39 38	Input	Set the operating mode. Inputs at these pins should not be changed during operation.
System control	$\overline{\text{RES}}$	41	Input	Reset input pin. When this pin is low, the chip is reset.
	$\overline{\text{STBY}}$	45	Input	When this pin is low, a transition is made to hardware standby mode.
	FWE	51	Input	Pin for use by flash memory. This pin is only available in the flash memory version.

Type	Symbol	Pin No.	I/O	Function
Interrupts	NMI	43	Input	Nonmaskable interrupt request pin. If this pin is not used, it should be fixed high.
	$\overline{\text{IRQ5}}$ $\overline{\text{IRQ4}}$ $\overline{\text{IRQ3}}$ $\overline{\text{IRQ2}}$ $\overline{\text{IRQ1}}$ $\overline{\text{IRQ0}}$	21 18 11 8 4 2	Input	These pins request a maskable interrupt.
16-bit timer pulse unit	TCLKA TCLKB TCLKC TCLKD	79 1 3 5	Input	These pins input an external clock.
	TIOCA0 TIOCB0 TIOCC0 TIOCD0	75 77 79 1	Input/ output	TGRA_0 to TGRD_0 input capture input/output compare output/PWM output pins.
	TIOCA1 TIOCB1	2 3	Input/ output	TGRA_1 and TGRB_1 input capture input/output compare output/PWM output pins.
	TIOCA2 TIOCB2	4 5	Input/ output	TGRA_2 and TGRB_2 input capture input/output compare output/PWM output pins.
	TIOCA3 TIOCB3 TIOCC3 TIOCD3	24 26 28 29	Input/ output	TGRA_3 to TGRD_3 input capture input/output compare output/PWM output pins.
	TIOCA4 TIOCB4	30 31	Input/ output	TGRA_4 and TGRB_4 input capture input/output compare output/PWM output pins.
	TIOCA5 TIOCB5	32 33	Input/ output	TGRA_5 and TGRB_5 input capture input/output compare output/PWM output pins.
Serial communication interface (SCI)/ smart card interface	TxD2 TxD1 TxD0	35 19 16	Output	Data output pins
	RxD2 RxD1 RxD0	36 20 17	Input	Data input pins
	SCK2 SCK1 SCK0	37 21 18	Input/ Output	Clock input/output pins
HCAN	HTxD	6	Output	CAN bus transmission pin
	HRxD	7	Input	CAN bus reception pin

Type	Symbol	Pin No.	I/O	Function
A/D converter	AN15	57	Input	Analog input pins
	AN14	58		
	AN13	59		
	AN12	60		
	AN11	62		
	AN10	63		
	AN9	64		
	AN8	65		
	AN7	66		
	AN6	67		
	AN5	68		
	AN4	69		
	AN3	70		
	AN2	71		
	AN1	72		
	AN0	73		
	$\overline{\text{ADTRG}}$	11	Input	Pin for input of an external trigger to start A/D conversion
	AVCC	61	Input	Power supply pin for the A/D converter. When the A/D converter is not used, connect this pin to the system power supply (+5 V).
	AVSS	74	Input	The ground pin for the A/D converter. Connect this pin to the system power supply (0 V).
I/O ports	P17	5	Input/ output	8-bit input/output pins
	P16	4		
	P15	3		
	P14	2		
	P13	1		
	P12	79		
	P11	77		
	P10	75		
	P47	66	Input	8-bit input pins
	P46	67		
	P45	68		
	P44	69		
	P43	70		
	P42	71		
	P41	72		
	P40	73		

Type	Symbol	Pin No.	I/O	Function
I/O ports	P97 P96 P95 P94 P93 P92 P91 P90	57 58 59 60 62 63 64 65	Input	8-bit input pins
	PA3 PA2 PA1 PA0	37 36 35 34	Input/ output	4-bit input/output pins
	PB7 PB6 PB5 PB4 PB3 PB2 PB1 PB0	33 32 31 30 29 28 26 24	Input/ output	8-bit input/output pins
	PC7 PC6 PC5 PC4 PC3 PC2 PC1 PC0	23 22 21 20 19 18 17 16	Input/ output	8-bit input/output pins
	PD7 PD6 PD5 PD4	53 54 55 56	Input/ output	4-bit input/output pins
	PF7 PF6 PF5 PF4 PF3 PF2 PF1 PF0	15 14 13 12 11 10 9 8	Input/ output	8-bit input/output pins

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

- Additional control register
  - One 8-bit and two 32-bit control registers have been added.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - A multiply-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

  A maximum address space of 64 kbytes can be accessed.

- 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 differs depending on the microcontroller. For details on the exception vector table, see section 4, Exception Handling.

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

- Stack Structure

  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.

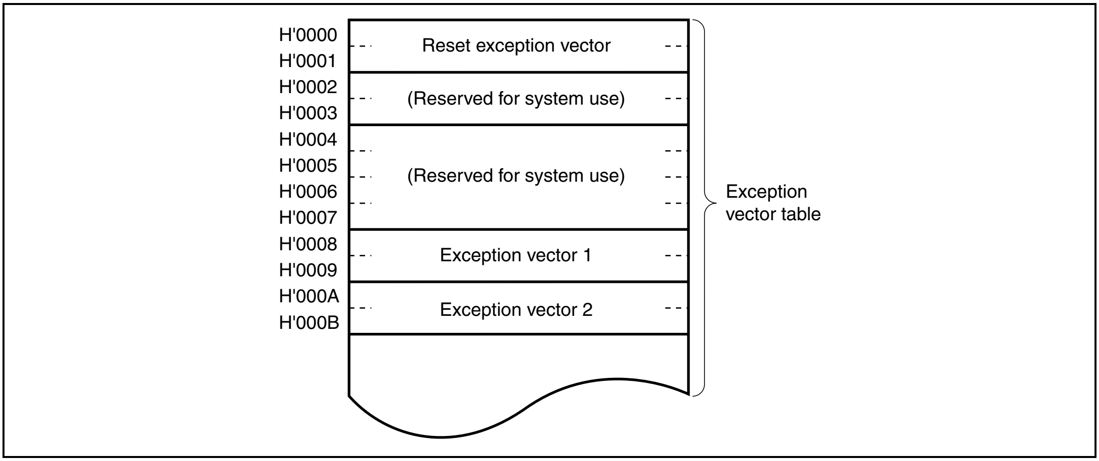
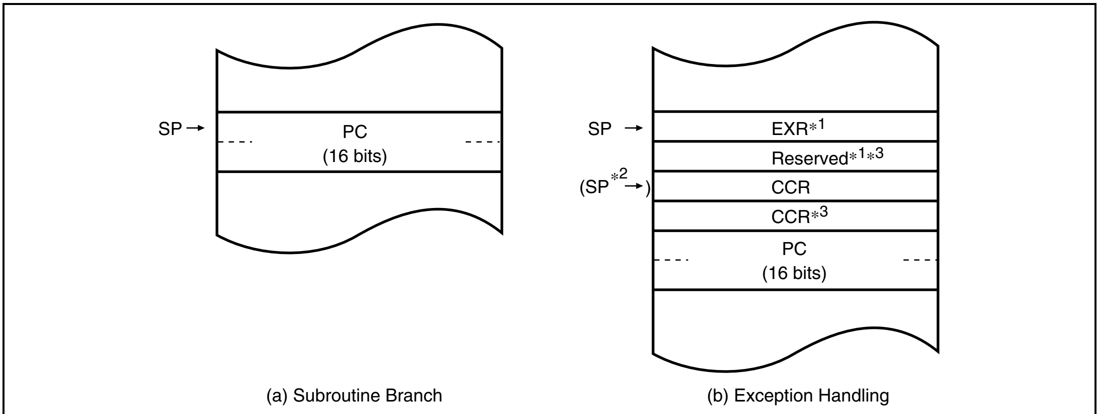


Figure 2.1 Exception Vector Table (Normal Mode)



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

Figure 2.2 Stack Structure in Normal Mode

2.2.2 Advanced Mode

- Address Space

  Linear access is provided to a 16-Mbyte maximum address space 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 on the exception vector table, see section 4, Exception Handling.

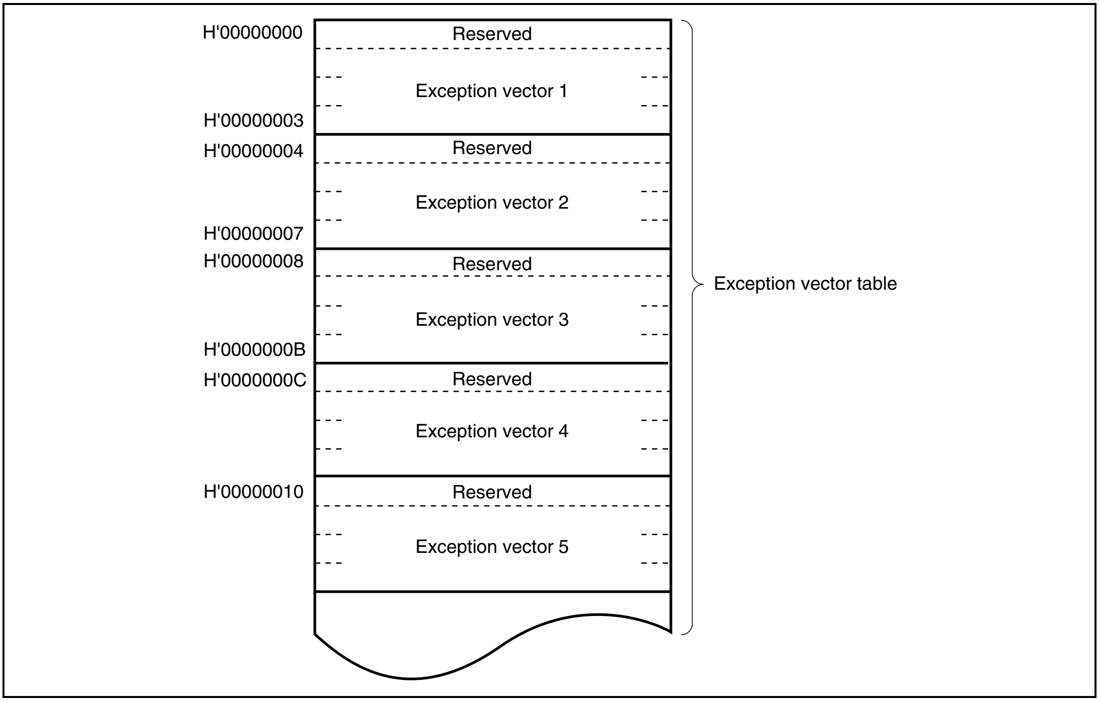
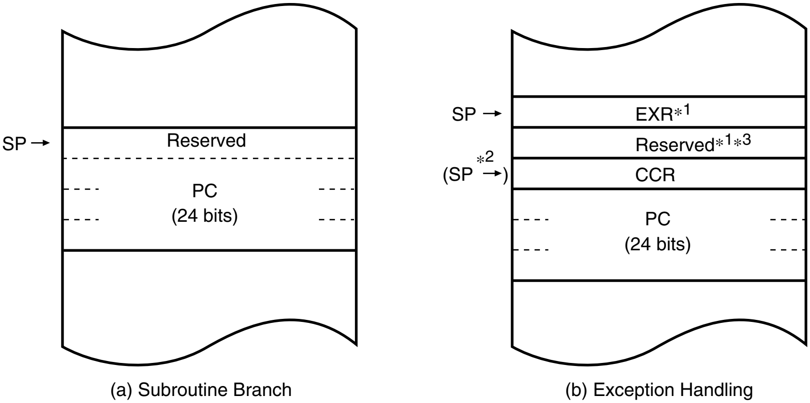


Figure 2.3 Exception Vector Table (Advanced Mode)

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 are 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 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 invalid, it is not pushed onto the stack. For details, see section 4, Exception Handling.



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

Figure 2.4 Stack Structure in Advanced Mode

2.3 Address Space

Figure 2.5 shows a memory map 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.

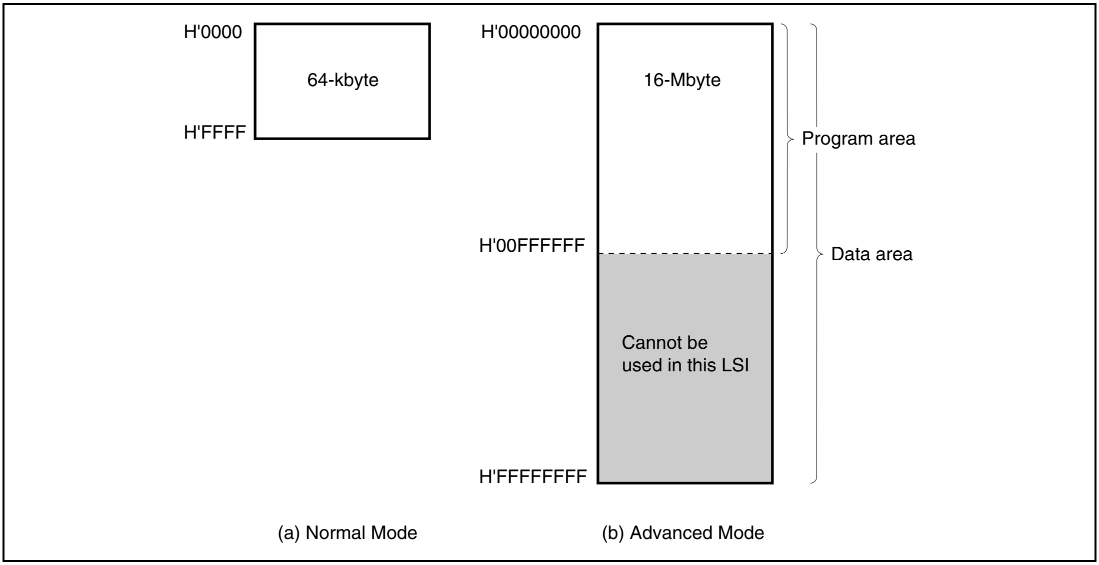
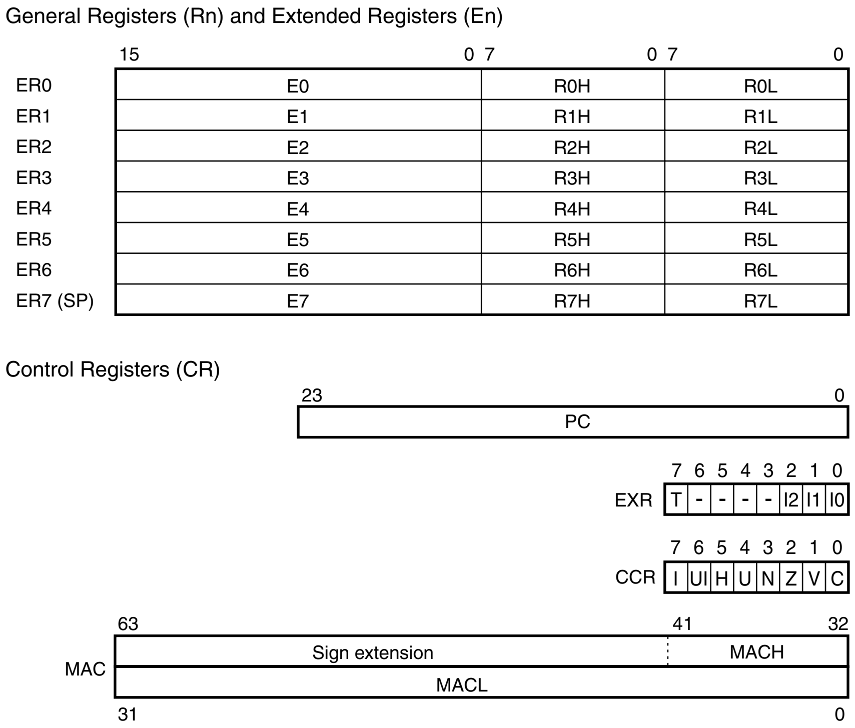


Figure 2.5 Memory Map

2.4 Register Configuration

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).



Legend

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

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.

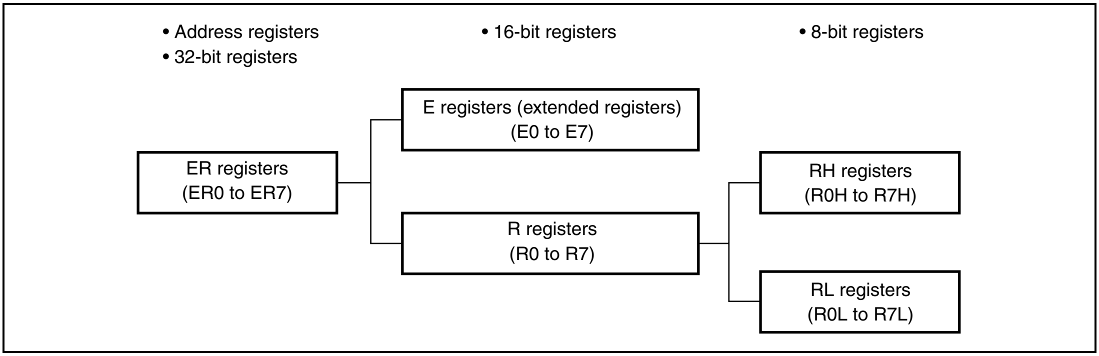


Figure 2.7 Usage of General Registers

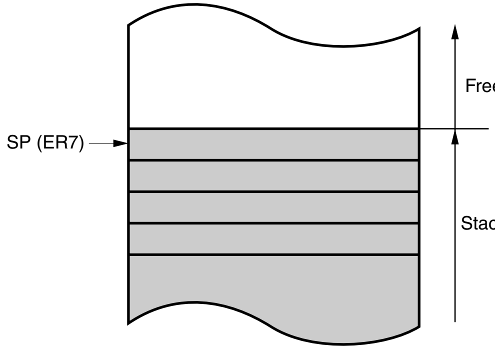


Figure 2.8 Stack

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

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

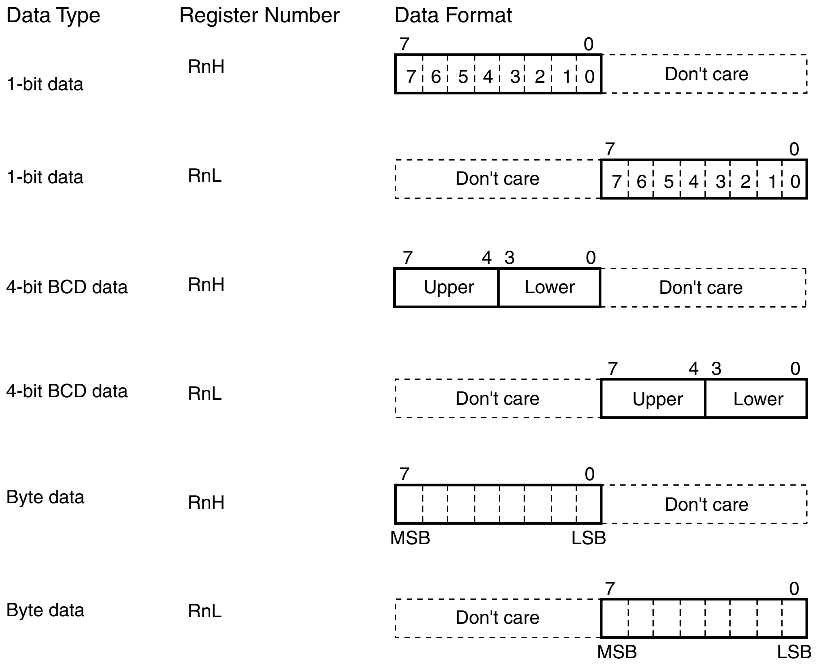


Figure 2.9 General Register Data Formats (1)

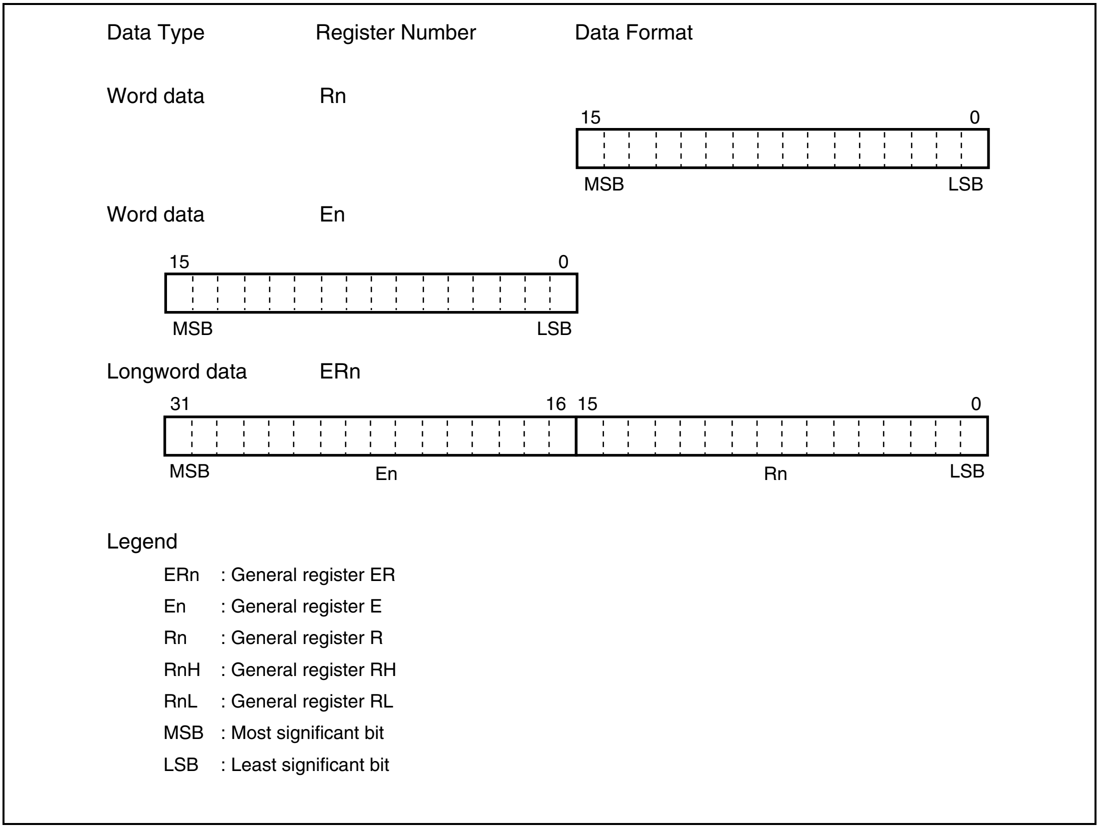


Figure 2.9 General Register Data Formats (2)

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.

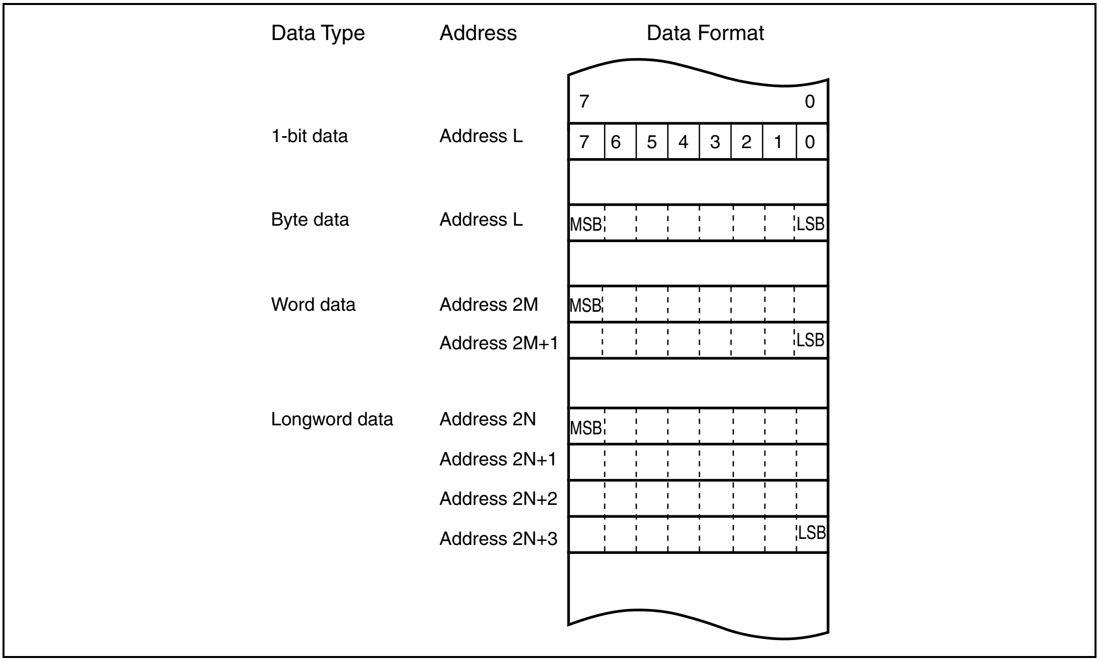


Figure 2.10 Memory Data Formats

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 operations	ADD, SUB, CMP, NEG	B/W/L	23
	ADDX, SUBX, DAA, DAS	B	
	INC, DEC	B/W/L	
	ADDS, SUBS	L	
	MULXU, DIVXU, MULXS, DIVXS	B/W	
	EXTU, EXTS	W/L	
	TAS*[4]	B	
	MAC, LDMAC, STMAC, CLRMAC	—	
Logic operations	AND, OR, XOR, NOT	B/W/L	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	B/W/L	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B	14
Branch	Bcc*[2], JMP, BSR, JSR, RTS	—	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—	9
Block data transfer	EEPMOV	—	1

Total: 69

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

Note: 1. Refers to the operand size.

B: Byte

W: Word

L: Longword

2. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

Table 2.5 Logic Operations Instructions

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

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

Table 2.9 System Control Instructions

Instruction	Size*	Function
TRAPA	—	Starts trap-instruction exception handling.
RTE	—	Returns from an exception-handling routine.
SLEEP	—	Causes a transition to a power-down state.
LDC	B/W	(EAs) → CCR, (EAs) → EXR Moves the source operand contents or immediate data to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 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
L: Longword

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

This LSI 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.

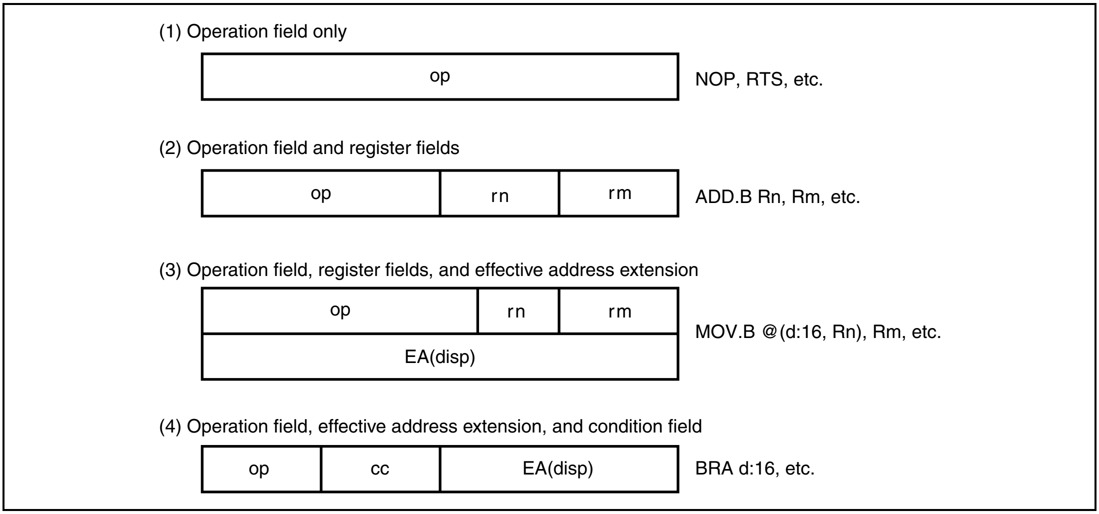

(1) Operation field only
op
NOP, RTS, etc.
(2) Operation field and register fields
op
rn
rm
ADD.B Rn, Rm, etc.
(3) Operation field, register fields, and effective address extension
op
rn
rm
EA(disp)
MOV.B @(d:16, Rn), Rm, etc.
(4) Operation field, effective address extension, and condition field
op
cc
EA(disp)
BRA d:16, etc.

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.

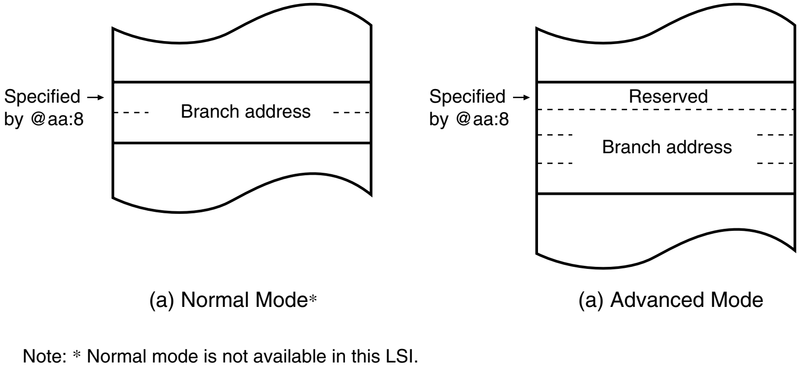


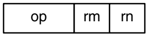
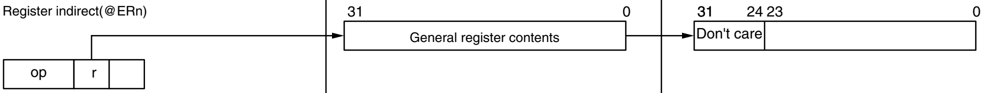
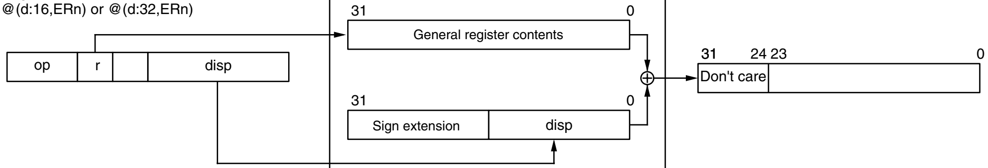
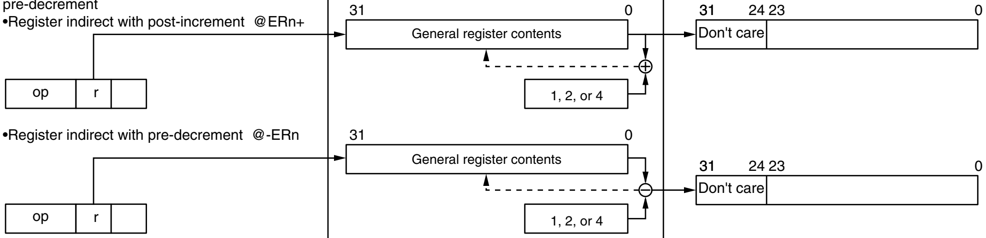
Figure 2.12 Branch Address Specification in Memory Indirect Mode

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.

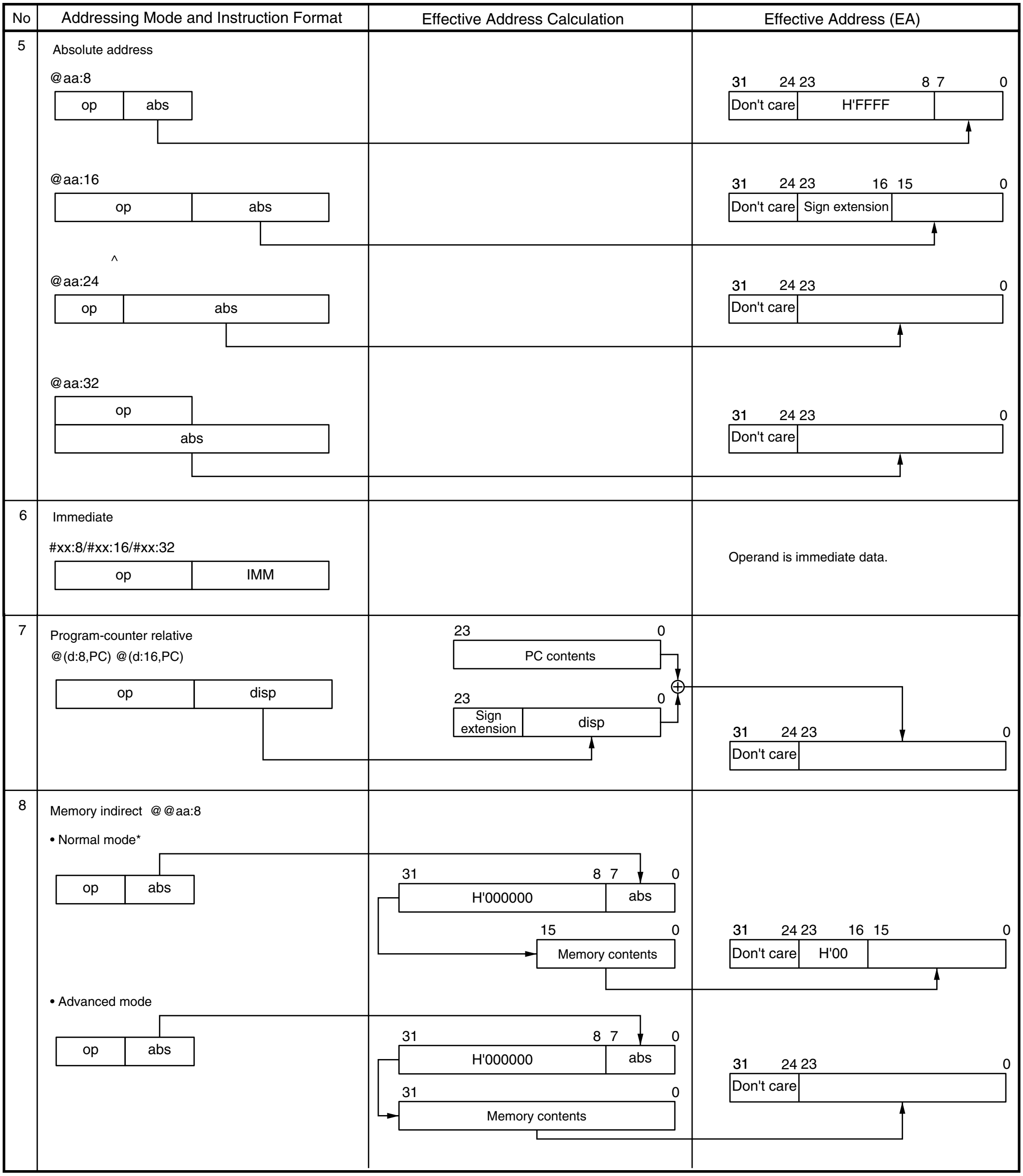
Note: Normal mode is not available in this LSI.

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)	 op r 31 0 General register contents 31 24 23 0 Don't care	
3	Register indirect with displacement @(d:16,ERn) or @(d:32,ERn)	 op r disp 31 0 General register contents 31 0 Sign extension disp 31 24 23 0 Don't care	
4	Register indirect with post-increment or pre-decrement •Register indirect with post-increment @ERn+ •Register indirect with pre-decrement @-ERn	 op r 31 0 General register contents 1, 2, or 4 31 24 23 0 Don't care op r 31 0 General register contents 1, 2, or 4 31 24 23 0 Don't care	

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 2.14 shows a diagram of the processing states. 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.

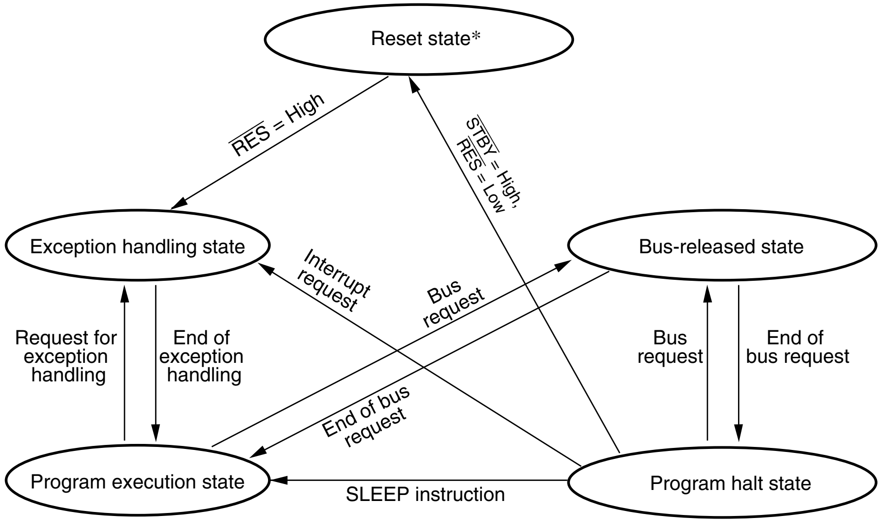
- Bus-Released State

  In a product which has a bus master other than the CPU, such as a data transfer controller (DTC), the bus-released state occurs when the bus has been released in response to a bus request from a bus 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 16, Power-Down Modes.



Notes: From any state, a transition to hardware standby mode occurs when $\overline{\text{STBY}}$ goes low.
* From any state except hardware standby mode, a transition to the reset state occurs whenever $\overline{\text{RES}}$ goes low. A transition can also be made to the reset state when the watchdog timer overflows.

Figure 2.13 State Transitions

2.9 Usage Notes

2.9.1 Usage Notes on Bit Manipulation Instructions

The BSET, BCLR, BNOT, BST, and BIST instructions are used to read data in bytes, then, after bit manipulation, they write data in bytes again. Therefore, special care is necessary to use these instructions for the registers and the ports that include write-only bit.

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

Section 3 MCU Operating Modes

3.1 Operating Mode Selection

This LSI supports only operating mode 7, that is, the advanced single-chip mode. The operating mode is determined by the setting of the mode pins (MD2 to MD0). Only mode 7 can be used in this LSI. Therefore, all mode pins must be fixed high, as shown in table 3.1. 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	External Data Bus	
							Initial Width	**Max. Width**
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)

MDCR monitors the current operating mode of this LSI.

Bit	Bit Name	Initial Value	R/W	Descriptions
7	—	1	R/W	Reserved Only 1 should be written to this bit.
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
1	MDS1	—*	R	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 reset.
0	MDS0	—*	R	

Note: * The initial values are determined according to the settings of the MD2 to MD0 pins.

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	—	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

Bit	Bit Name	Initial Value	R/W	Descriptions
6	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
5 4	INTM1 INTM0	0 0	R/W 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
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	— —	0 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 Pin Functions in Each Operating Mode

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

3.3.1 Pin Functions

Table 3.2 shows their functions in mode 7.

Table 3.2 Pin Functions in Each Operating Mode

	Port	Mode 7
Port 1	P10	P
	P11 to P13	P
Port A	PA3 to PA0	P
Port B		P
Port C		P
Port D		P
Port F	PF7	P*/C
	PF6 to PF4	P
	PF3	
	PF2 to PF0	

Legend

P:	I/O port
A:	Address bus output
D:	Data bus I/O
C:	Control signals, clock I/O
*:	After reset

3.4 Address Map

Figure 3.1 shows the address map in each operating mode.

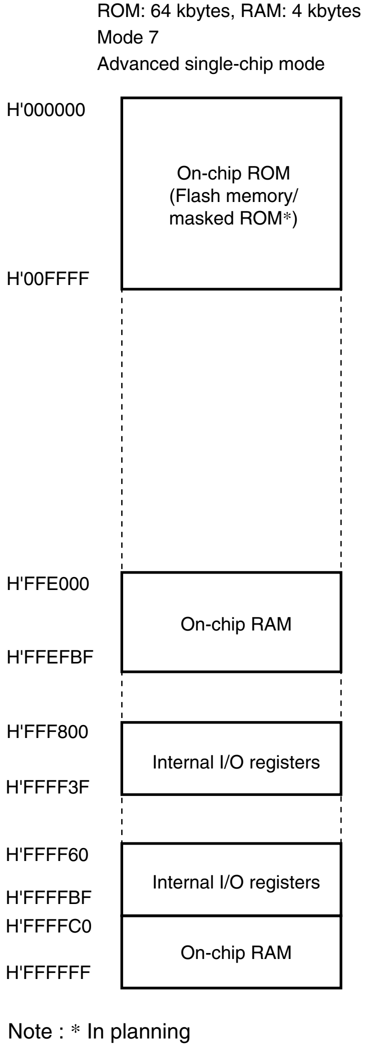


Figure 3.1 Address Map

Section 4 Exception Handling

4.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, trap instruction, or interrupt. Exception handling is prioritized as shown in table 4.1. If two or more exceptions occur simultaneously, they are accepted and processed in order of priority. Exception sources, the stack structure, and operation of the CPU vary depending on the interrupt control mode. For details on the interrupt control mode, refer to section 5, Interrupt Controller.

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High ↑	Reset	Starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin, or when the watchdog timer overflows. The CPU enters the reset state when the $\overline{\text{RES}}$ pin is low.
	Trace*[1]	Starts when execution of the current instruction or exception handling ends, if the trace (T) bit in the EXR is set to 1
	Direct transition	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**	**Advanced Mode**
Power-on reset		0	H'0000 to H'0001	H'0000 to H'0003
Manual reset *[2]		1	H'0002 to H'0003	H'0004 to H'0007
Reserved for system use		2	H'0004 to H'0005	H'0008 to H'000B
		3	H'0006 to H'0007	H'000C to H'000F
		4	H'0008 to H'0019	H'0010 to H'0013
Trace		5	H'000A to H'000B	H'0014 to H'0017
Interrupt (direct transitions)		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
Internal interrupt*[3]		24 | 127	H'0030 to H'0031 | H'00FE to H'00FF	H'0060 to H'0063 | H'01FC to H'01FF

Notes: 1. Lower 16 bits of the address.

2. Not available in this LSI.

3. For details of internal interrupt vectors, see section 5.5, Interrupt Exception Handling Vector Table.

4.3 Reset

A reset has the highest exception priority.

When the $\overline{\text{RES}}$ pin goes low, all processing halts and this LSI enters the reset. To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-up. To reset the chip during operation, hold the $\overline{\text{RES}}$ pin low for at least 20 states. A reset initializes the internal state of the CPU and the registers of on-chip peripheral modules.

The chip can also be reset by overflow of the watchdog timer. For details, see section 9, Watchdog Timer (WDT).

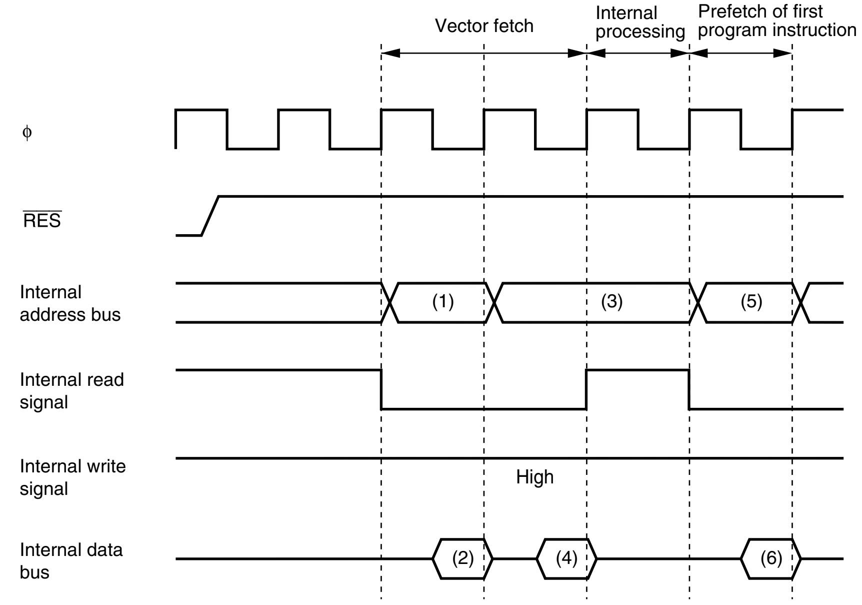
The interrupt control mode is 0 immediately after reset.

4.3.1 Reset Exception Handling

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

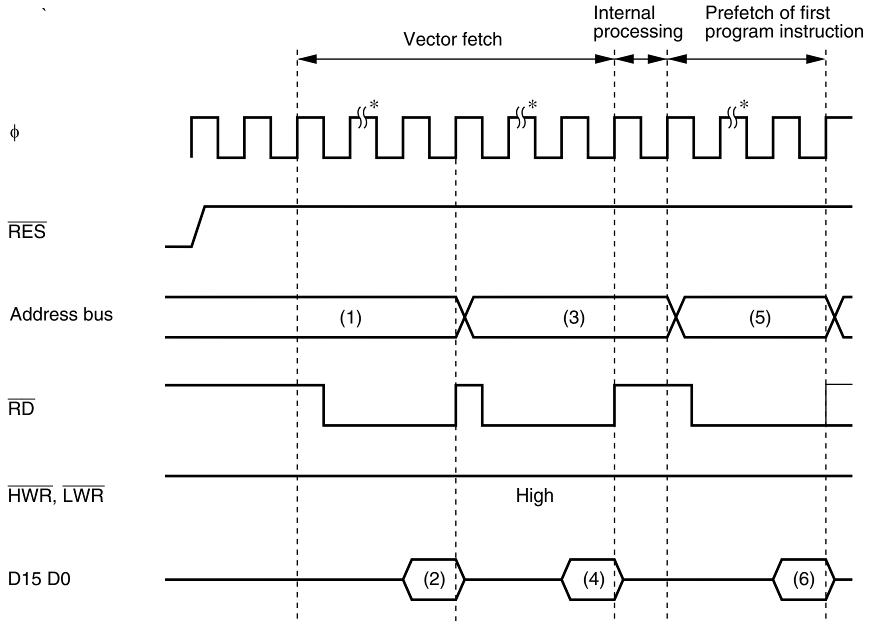
1. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized, the T bit is cleared to 0 in EXR, and the I bit is set to 1 in EXR and CCR.
2. The reset exception handling vector address is read and transferred to the PC, and program execution starts from the address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

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: Cannot be Used in this LSI)

4.3.2 Interrupts after Reset

If an interrupt is accepted after a reset and before the stack pointer (SP) is initialized, the PC and CCR will not be saved correctly, leading to a program crash. To prevent this, all interrupt requests, including NMI, are disabled immediately after a reset. Since the first instruction of a program is always executed immediately after the reset state ends, 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 MSTPCRC are initialized to H'3F, H'FF, and H'FF, respectively, and all modules enter module stop mode. Consequently, on-chip peripheral module registers cannot be read or written to. Register reading and writing is enabled when the module stop mode is exited.

4.4 Traces

Traces are enabled in interrupt control mode 2. Trace mode is not activated in interrupt control mode 0, irrespective of the state of the T bit. For details 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 masking. Table 4.3 shows the state of CCR and EXR after execution of trace exception handling. Trace mode is canceled by clearing the T bit in EXR to 0. The T bit saved on the stack retains its value of 1, and when control is returned from the trace exception handling routine by the RTE instruction, trace mode resumes. Trace exception handling is not carried out after execution of the RTE instruction.

Interrupts are accepted even within the trace exception handling routine.

Table 4.3 Status of CCR and EXR after Trace Exception Handling

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

Legend

1:	Set to 1
0:	Cleared to 0
—:	Retains value prior to execution

4.5 Interrupts

Interrupts are controlled by the interrupt controller. The interrupt controller has two interrupt control modes and can assign interrupts other than NMI to eight priority/mask levels to enable multiplexed interrupt control. 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.
3. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution starts from that address.

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

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

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

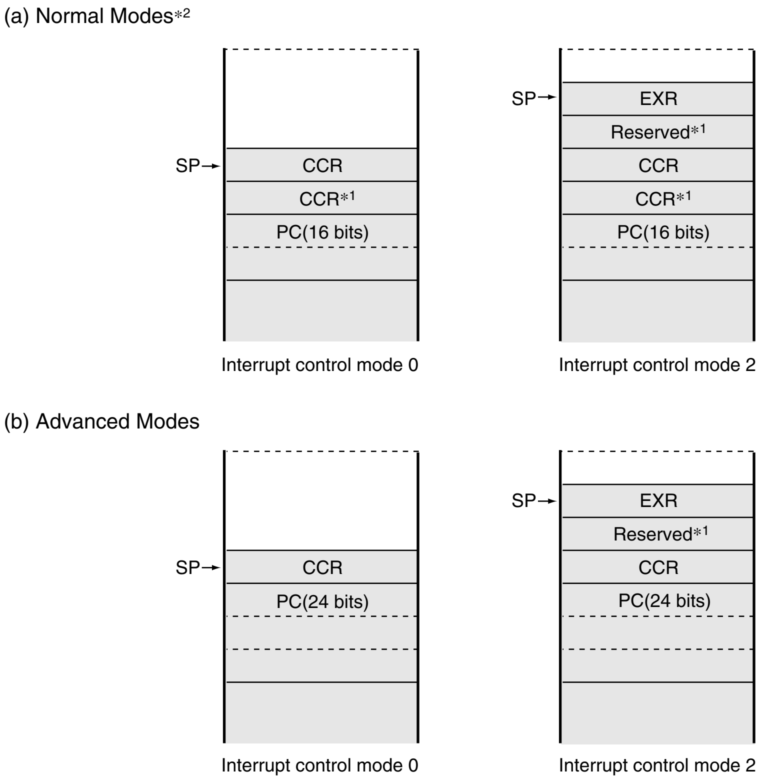
	CCR		EXR	
Interrupt Control Mode	**I**	**UI**	**I2 to I0**	**T**
0	1	—	—	—
2	1	—	—	0

Legend

1: Set to 1
0: Cleared to 0
—: Retains value prior to execution

4.7 Stack Status after Exception Handling

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



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

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.

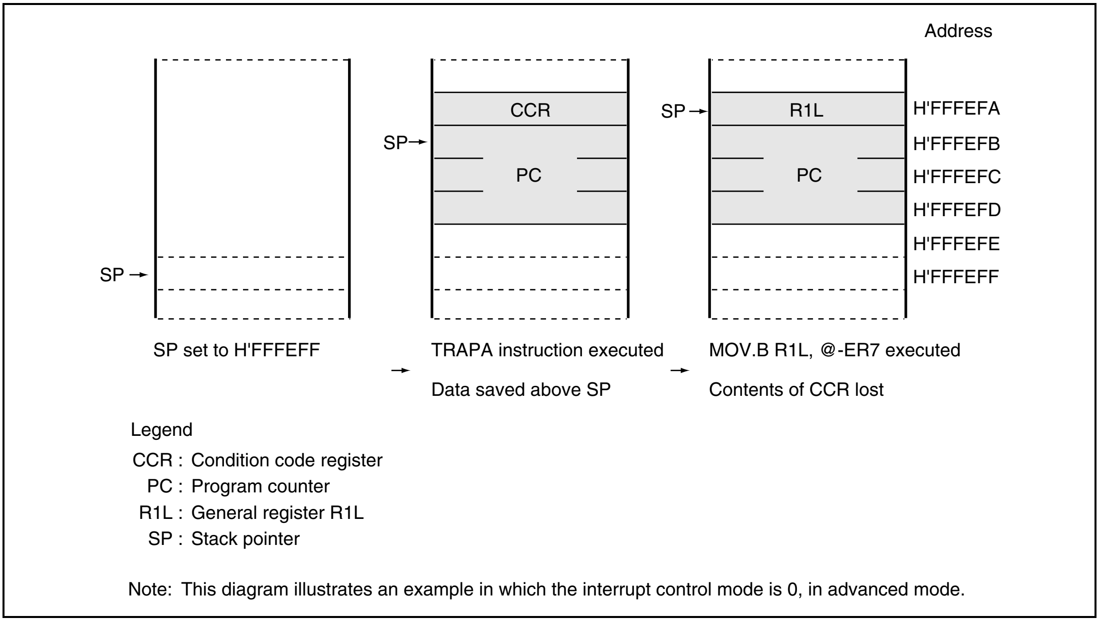

Address
SP
CCR
PC
SP
R1L
PC
SP
H'FFFEFA
H'FFFEFB
H'FFFEFC
H'FFFEFD
H'FFFEFE
H'FFFEFF
SP set to H'FFFEFF
TRAPA instruction executed
Data saved above SP
MOV.B R1L, @-ER7 executed
Contents of CCR lost
Legend
CCR : Condition code register
PC : Program counter
R1L : General register R1L
SP : Stack pointer
Note: This diagram illustrates an example in which the interrupt control mode is 0, in advanced mode.

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 IRQ0 to IRQ5.

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

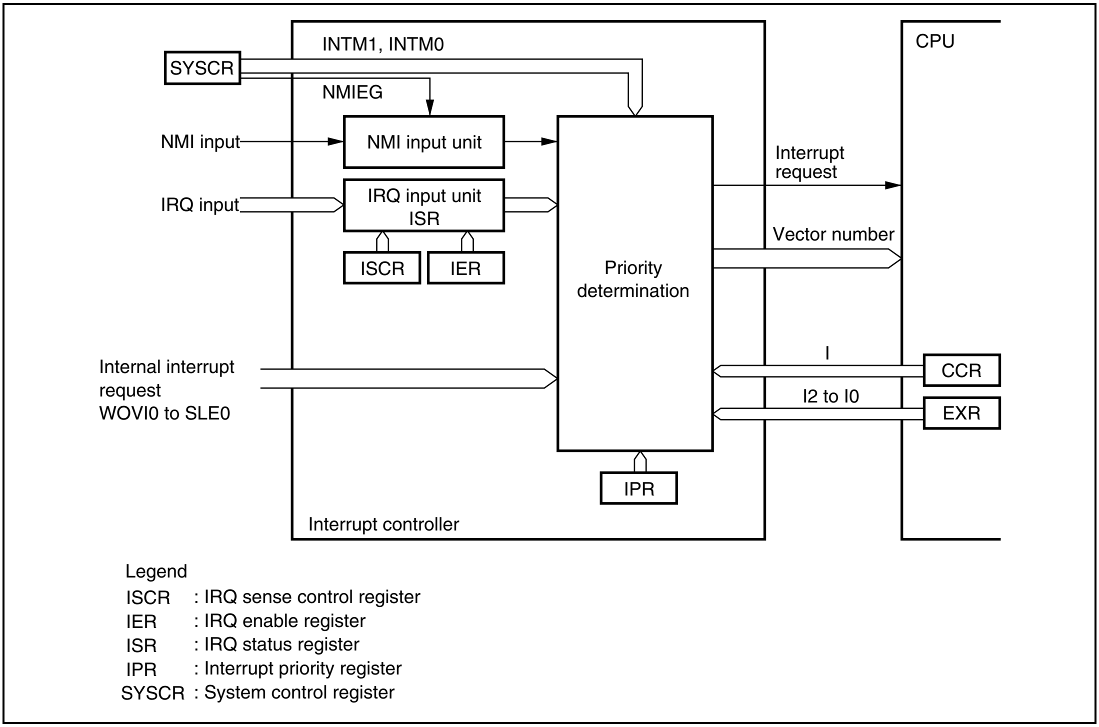


Figure 5.1 Block Diagram of Interrupt Controller

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 details on 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 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 M (IPRM)

5.3.1 Interrupt Priority Registers A, B, D to H, J, K, M (IPRA, IPRB, IPRD to IPRH, IPRJ, IPRK, IPRM)

IPR are ten 8-bit readable/writable registers that set priorities (levels 7 to 0) for interrupts other than NMI.

The correspondence between interrupt sources and IPR settings is shown in table 5.2 (Interrupt Sources, Vector Addresses, and Interrupt Priorities). 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	Sets the priority of the corresponding interrupt source.
5	IPR5	1	R/W	000: Priority level 0 (Lowest)
4	IPR4	1	R/W	001: Priority level 1
				010: Priority level 2
				011: Priority level 3
				100: Priority level 4
				101: Priority level 5
				110: Priority level 6
				111: Priority level 7 (Highest)
3	—	0	—	Reserved These bits are always read as 0.
2	IPR2	1	R/W	Sets the priority of the corresponding interrupt source.
1	IPR1	1	R/W	000: Priority level 0 (Lowest)
0	IPR0	1	R/W	001: Priority level 1
				010: Priority level 2
				011: Priority level 3
				100: Priority level 4
				101: Priority level 5
				110: Priority level 6
				111: Priority level 7 (Highest)

5.3.2 IRQ Enable Register (IER)

IER controls the enabling and disabling of interrupt requests IRQ0 to IRQ5.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved
6	—	0	R/W	Only 0 should be written to these bits.
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)

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R/W	Reserved Only 0 should be written to these bits.
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
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

Bit	Bit Name	Initial Value	R/W	Description
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	—	0	R/W	Reserved
6	—	0	R/W	Only 0 should be written to these bits.
5	IRQ5F	0	R/W	[Setting conditions] 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 (n=5 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	

5.4 Interrupt

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.

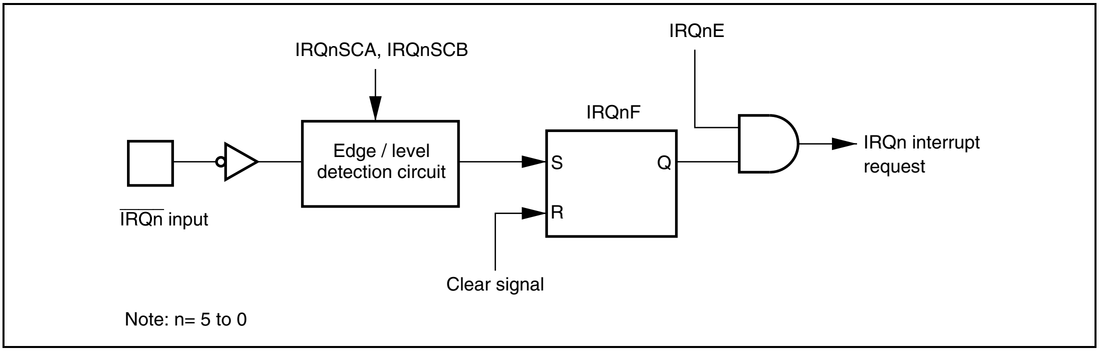


Figure 5.2 Block Diagram of Interrupts IRQ0 to IRQ5

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.

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 the IPR. Modules set at the same priority will conform to their default priorities. Priorities within a module are fixed.

Table 5.2 Interrupt Sources, Vector Addresses, and Interrupt Priorities

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*	IPR	Priority
			Advanced Mode		
External pin	NMI	7	H'001C		High
	IRQ0	16	H'0040	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		
Watchdog timer 0	WOVI0	25	H'0064	IPRD6 to IPRD4	
A/D	ADI	28	H'0070	IPRE2 to IPRE0	
Watchdog timer 1	WOVI1	29	H'0074	IPRE2 to IPRE0	
TPU channel 0	TGI0A	32	H'0080	IPRF6 to IPRF4	
	TGI0B	33	H'0084		
	TGI0C	34	H'0088		
	TGI0D	35	H'008C		
	TCI0V	36	H'0090		
TPU channel 1	TGI1A	40	H'00A0	IPRF2 to IPRF0	
	TGI1B	41	H'00A4		
	TCI1V	42	H'00A8		
	TCI1U	43	H'00AC		
TPU channel 2	TGI2A	44	H'00B0	IPRG6 to IPRG4	
	TGI2B	45	H'00B4		
	TCI2V	46	H'00B8		
	TCI2U	47	H'00BC		Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*	IPR	Priority
			Advanced Mode		
TPU channel 3	TGI3A	48	H'00C0	IPRG2 to IPRG0	High
	TGI3B	49	H'00C4		
	TGI3C	50	H'00C8		
	TGI3D	51	H'00CC		
	TCI3V	52	H'00D0		
TPU channel 4	TGI4A	56	H'00E0	IPRH6 to IPRH4	
	TGI4B	57	H'00E4		
	TCI4V	58	H'00E8		
	TCI4U	59	H'00EC		
TPU channel 5	TGI5A	60	H'00F0	IPRH2 to IPRH0	
	TGI5B	61	H'00F4		
	TCI5V	62	H'00F8		
	TCI5U	63	H'00FC		
SCI channel 0	ERI0	80	H'0140	IPRJ2 to IPRJ0	
	RXI0	81	H'0144		
	TXI0	82	H'0148		
	TEI0	83	H'014C		
SCI channel 1	ERI1	84	H'0150	IPRK6 to IPRK4	
	RXI1	85	H'0154		
	TXI1	86	H'0158		
	TEI1	87	H'015C		
SCI channel 2	ERI2	88	H'0160	IPRK2 to IPRK0	
	RXI2	89	H'0164		
	TXI2	90	H'0168		
	TEI2	91	H'016C		
HCAN	ERS0, OVR0	104	H'01A0	IPRM6 to IPRM4	
	RM0	105	H'01A4		
	RM1	106	H'01A8		
	SLE0	107	H'01AC		
—	Reserved for system use	111	H'01BC	IPRM2 to IPRM0	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 of the 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 is set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending. If the I bit is cleared, an interrupt request is accepted.
3. Interrupt requests are sent to the interrupt controller, the highest-ranked interrupt according to the priority system is accepted, and other interrupt requests are held pending.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except NMI.
7. The CPU generates a vector address for the accepted interrupt and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

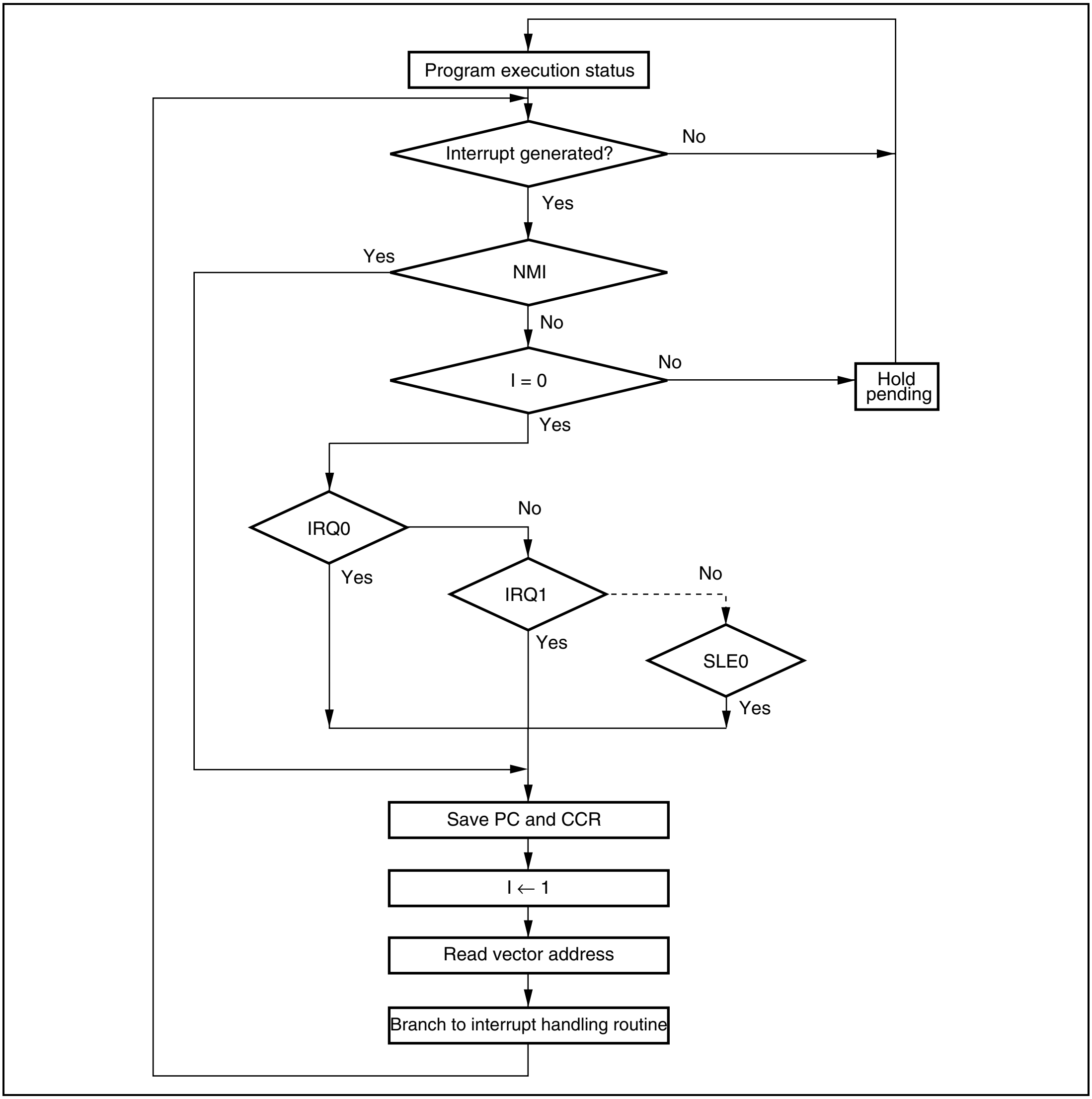


Figure 5.3 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 0

5.6.2 Interrupt Control Mode 2

In interrupt control mode 2, mask control is 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.

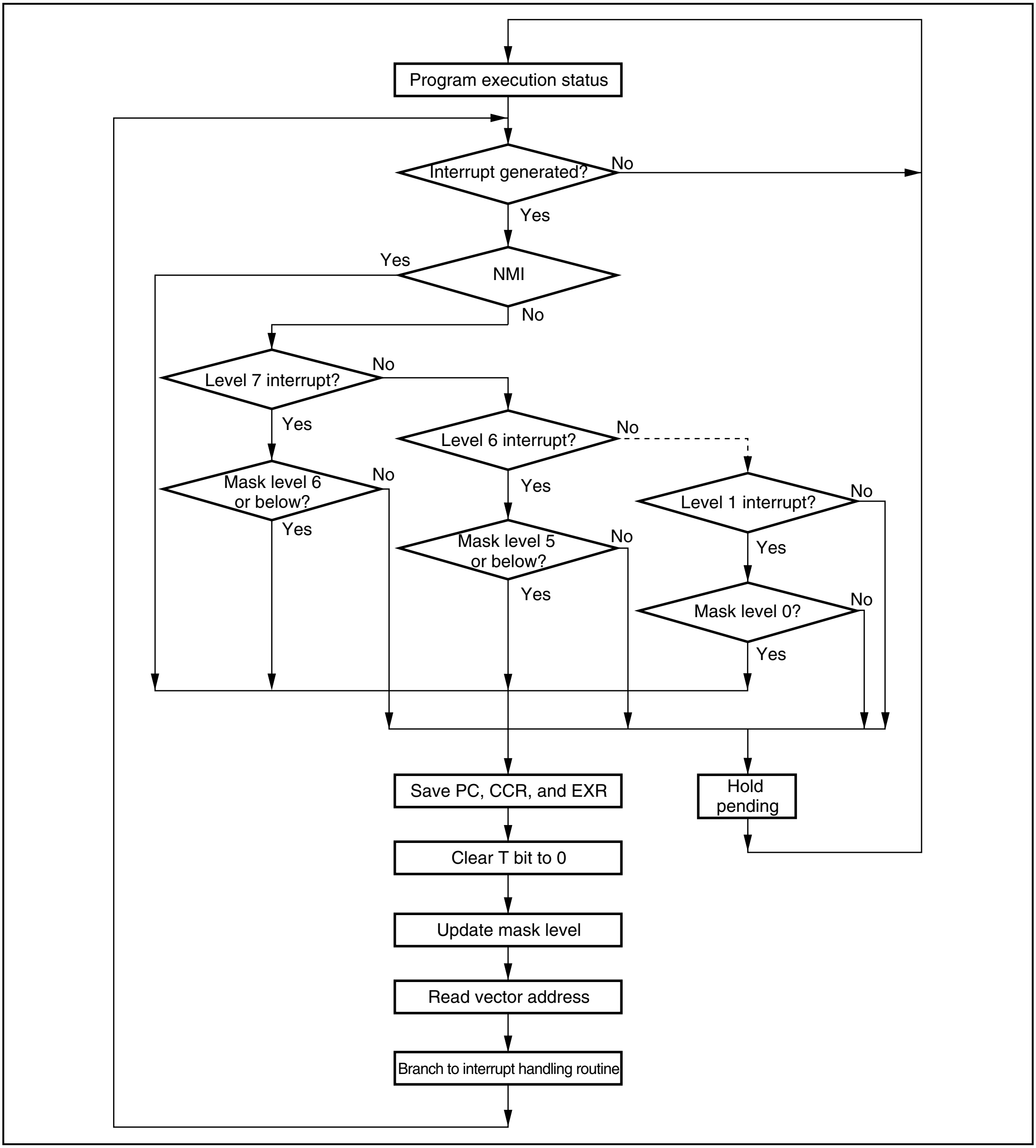
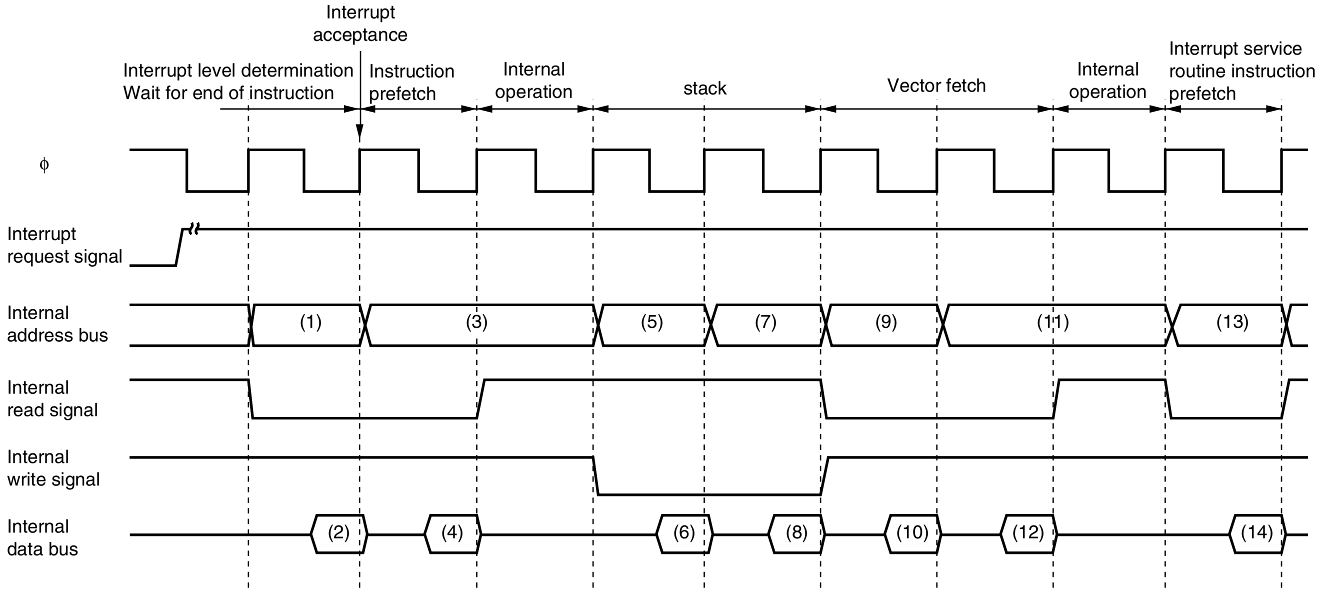


Figure 5.4 Flowchart of Procedure Up to Interrupt Acceptance in Control Mode 2

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.



(1) Instruction prefetch address (Not executed.
This is the contents of the saved PC, the return address.)
(2) (4) Instruction code (Not executed.)
(3) Instruction prefetch address (Not executed.)
(5) SP-2
(7) SP-4

(6) (8) Saved PC and saved CCR
(9) (11) Vector address
(10) (12) Interrupt handling routine start address (Vector address contents)
(13) Interrupt handling routine start address ((13) = (10)(12))
(14) First instruction of interrupt handling routine

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	SI	1	4	6+2m	2	3+m
Branch address read	SJ					
Stack manipulation	SK					

Legend

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

Note:* Cannot be used in this LSI.

5.7 Usage Notes

5.7.1 Contention between Interrupt Generation and Disabling

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

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

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

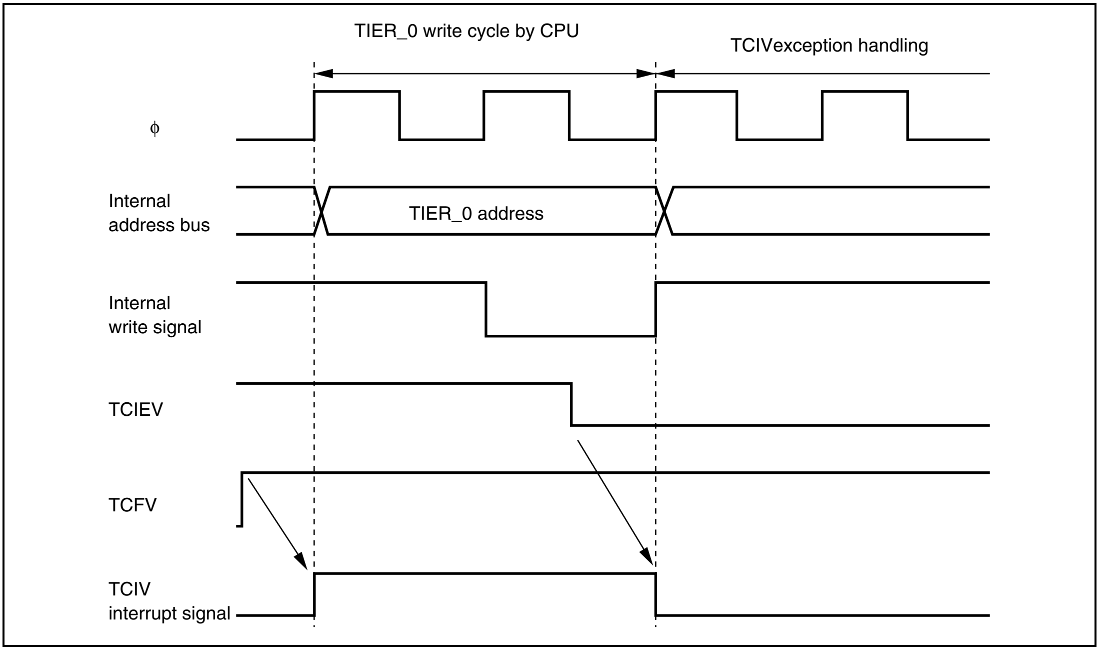


Figure 5.6 Contention between Interrupt Generation and Disabling

5.7.2 Instructions that Disable Interrupts

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 move 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 Bus Controller

The H8S/2600 CPU is driven by a system clock, denoted by the symbol ϕ.

The bus controller controls a memory cycle and a bus cycle. Different methods are used to access on-chip memory and on-chip peripheral modules.

6.1 Basic Timing

The period from one rising edge of ϕ to the next is referred to as a "state." The memory cycle or bus cycle consists of one, two, three, or four states. Different methods are used to access on-chip memory, on-chip support modules, and the external address space.

6.1.1 On-Chip Memory Access Timing (ROM, RAM)

On-chip memory is accessed in one state. The data bus is 16 bits wide, permitting both byte and word transfer instruction. Figure 6.1 shows the on-chip memory access cycle.

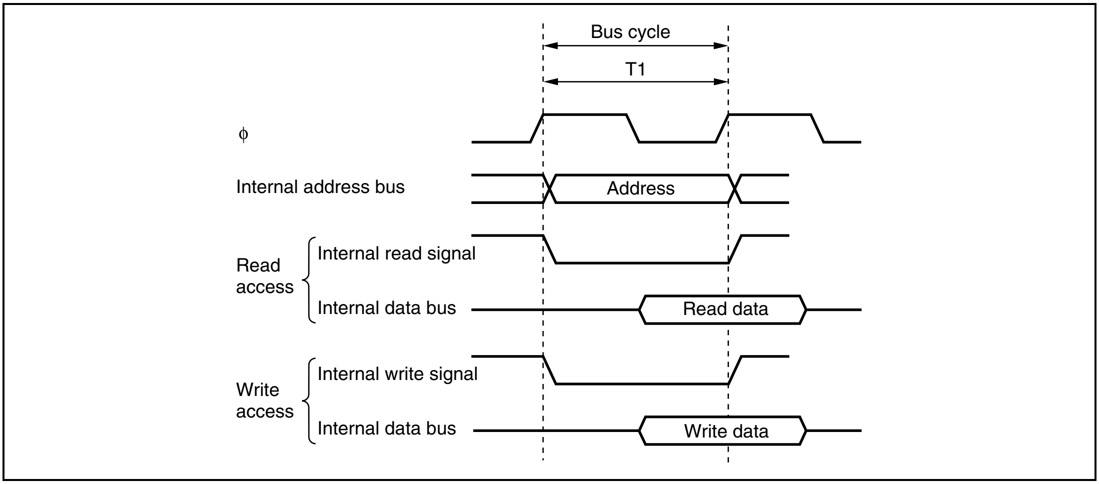


Figure 6.1 On-Chip Memory Access Cycle

6.1.2 On-Chip Peripheral Module Access Timing

The on-chip peripheral modules, except for HCAN are accessed in two states. The data bus is either 8 bits or 16 bits wide, depending on the particular internal I/O register being accessed. For details, refer to section 17, List of Registers. Figure 6.2 shows access timing for the on-chip peripheral modules.

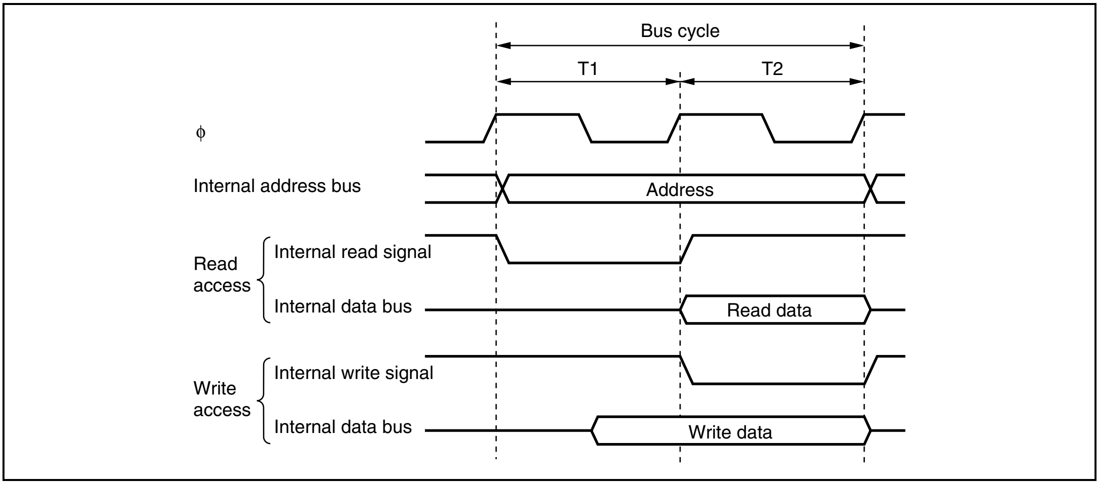


Figure 6.2 On-Chip Support Module Access Cycle

6.1.3 On-Chip HCAN Module Access Timing

On-chip HCAN module access is performed in four states. The data bus width is 16 bits. Wait states can be inserted by means of a wait request from the HCAN. On-chip HCAN module access timing is shown in figure 6.3.

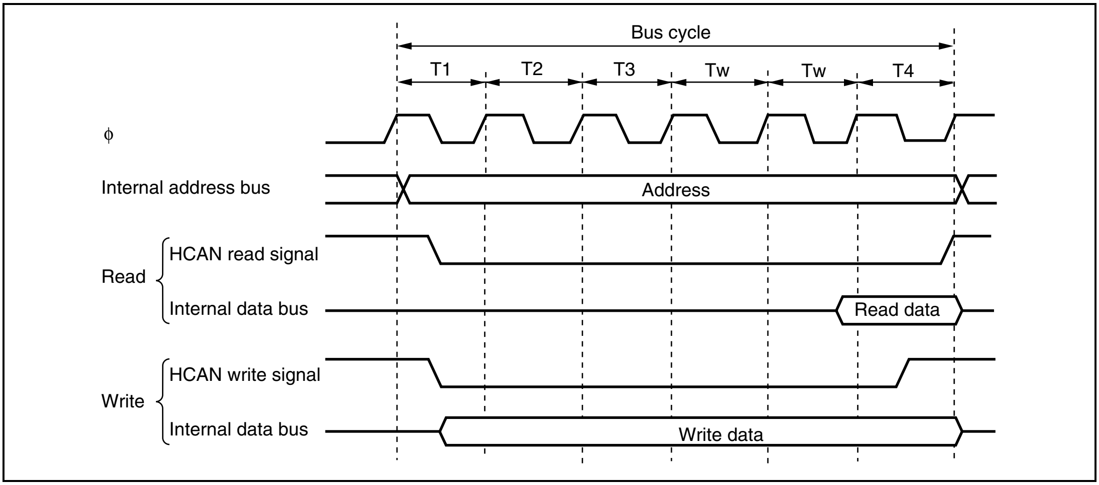


Figure 6.3 On-Chip HCAN Module Access Cycle (Wait States Inserted)

Section 7 I/O Ports

Table 7.1 summarizes the port functions. The pins of each port also have other functions such as input/output or interrupt input pins of on-chip peripheral modules.

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 D have a built-in input pull-up MOS function and an input pull-up MOS control register (PCR) to control the on/off state of the input pull-up MOS.

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

All the I/O ports can drive a single TTL load and a 30 pF capacitive load.

Table 7.1 Port Functions

Port	Description	Port and Other Functions Name	Input/Output and Output Type
Port 1	General I/O port also functioning as TPU I/O pins and interrupt input pins	P17/TIOCB2/TCLKD	
		P16/TIOCA2/$\overline{\text{IRQ1}}$	
		P15/TIOCB1/TCLKC	
		P14/TIOCA1/$\overline{\text{IRQ0}}$	
		P13/TIOCD0/TCLKB	
		P12/TIOCC0/TCLKA	
		P11/TIOCB0	
		P10/TIOCA0	
Port 4	General input port also functioning as A/D converter analog inputs	P47/AN7	
		P46/AN6	
		P45/AN5	
		P44/AN4	
		P43/AN3	
		P42/AN2	
		P41/AN1	
		P40/AN0	
Port 9	General input port also functioning as A/D converter analog inputs	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_2 I/O pins	PA3/SCK2	Built-in input pull-up MOS Push-pull or open-drain output selectable
		PA2/RxD2	
		PA1/TxD2	
		PA0	

Port	Description	Port and Other Functions Name	Input/Output and Output Type
Port B	General I/O port also functioning as TPU_5, TPU_4, and TPU_3 I/O pins	PB7/TIOCB5	Built-in input pull-up MOS Push-pull or open-drain output selectable
		PB6/TIOCA5	
		PB5/TIOCB4	
		PB4/TIOCA4	
		PB3/TIOCD3	
		PB2/TIOCC3	
		PB1/TIOCB3	
		PB0/TIOCA3	
Port C	General I/O port also functioning as SCI_1 and SCI_0 I/O pins, and interrupt input pins	PC7	Built-in input pull-up MOS Push-pull or open-drain output selectable
		PC6	
		PC5/SCK1/$\overline{\text{IRQ5}}$	
		PC4/RxD1	
		PC3/TxD1	
		PC2/SCK0/$\overline{\text{IRQ4}}$	
		PC1/RxD0	
		PC0/TxD0	
Port D	General I/O port	PD7	Built-in input pull-up MOS
		PD6	
		PD5	
		PD4	
Port F	General I/O port also functioning as interrupt input pins, an A/D converter start trigger input pin, and a system clock output pin (ϕ)	PF7/ ϕ	
		PF6	
		PF5	
		PF4	
		PF3/$\overline{\text{ADTRG}}$/$\overline{\text{IRQ3}}$	
		PF2	
		PF1	
		PF0/$\overline{\text{IRQ2}}$	

7.1 Port 1

Port 1 is an 8-bit I/O port that also has other functions. Port 1 has the following registers.

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

7.1.1 Port 1 Data Direction Register (P1DDR)

P1DDR specifies the 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	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port 1 pin an output pin. Clearing this 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	

7.1.2 Port 1 Data Register (P1DR)

P1DR stores output data for port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DR	0	R/W	Output data for a pin is stored when the pin is specified as 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	

7.1.3 Port 1 Register (PORT1)

PORT1 shows the pin states of the port 1. PORT1 cannot be modified.

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

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

7.1.4 Pin Functions

Port 1 pins also function as TPU I/O pins and interrupt input pins. The correspondence between the register specification and the pin functions is shown below.

Table 7.2 P17 Pin Function

TPU Channel 2 Setting*	Output	Input or Initial Value	
P17DDR	—	0	1
Pin function	TIOCB2 output	P17 input	P17 output
		TIOCB2 input	
		TCLKD input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

Table 7.3 P16 Pin Function

TPU Channel 2 Setting*	Output	Input or Initial Value	
P16DDR	—	0	1
Pin function	TIOCA2 output	P16 input	P16 output
		TIOCA2 input	
		$\overline{\text{IRQ1}}$ input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

Table 7.4 P15 Pin Function

TPU Channel 1 Setting*	Output	Input or Initial Value	
P15DDR	—	0	1
Pin function	TIOCB1 output	P15 input	P15 output
		TIOCB1 input	
		TCLKC input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

Table 7.5 P14 Pin Function

TPU Channel 1 Setting*	Output	Input or Initial Value	
P14DDR	—	0	1
Pin function	TIOCA1 output	P14 input	P14 output
		TIOCA1 input	
		$\overline{\text{IRQ0}}$ input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

Table 7.6 P13 Pin Function

TPU Channel 0 Setting*	Output	Input or Initial Value	
P13DDR	—	0	1
Pin function	TIOCD0 output	P13 input	P13 output
		TIOCD0 input	
		TCLKB input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

Table 7.7 P12 Pin Function

TPU Channel 0 Setting*	Output	Input or Initial Value	
P12DDR	—	0	1
Pin function	TIOCC0 output	P12 input	P12 output
		TIOCC0 input	
		TCLKA input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

Table 7.8 P11 Pin Function

TPU Channel 0 Setting*	Output	Input or Initial Value	
P11DDR	—	0	1
Pin function	TIOCB0 output	P11 input	P11 output
		TIOCB0 input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

Table 7.9 P10 Pin Function

TPU Channel 0 Setting*	Output	Input or Initial Value	
P10DDR	—	0	1
Pin function	TIOCA0 output	P10 input	P10 output
		TIOCA0 input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

7.2 Port 4

Port 4 is an 8-bit input-only port. Port 4 pins also function as A/D converter analog input pins. Port 4 has the following register.

- Port 4 register (PORT4)

7.2.1 Port 4 Register (PORT4)

PORT4 shows port 4 pin states. PORT4 cannot be modified.

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

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

7.3 Port 9

Port 9 is an 8-bit input-only port. Port 9 pins also function as A/D converter analog input pins. Port 9 has the following register.

- Port 9 register (PORT9)

7.3.1 Port 9 Register (PORT9)

PORT9 shows port 9 pin states. PORT9 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P97	Undefined*	R	The pin states are always read when a port 9 read is performed.
6	P96	Undefined*	R	
5	P95	Undefined*	R	
4	P94	Undefined*	R	
3	P93	Undefined*	R	
2	P92	Undefined*	R	
1	P91	Undefined*	R	
0	P90	Undefined*	R	

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

7.4 Port A

Port A is a 4-bit I/O port that also has other functions. Port A has the following registers.

- 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)

7.4.1 Port A Data Direction Register (PADDR)

PADDR specifies whether the pins of port A are used for input or output. PADDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved
3	PA3DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port A pin an output pin. Clearing this bit to 0 makes the pin an input pin.
2	PA2DDR	0	W	
1	PA1DDR	0	W	
0	PA0DDR	0	W	

7.4.2 Port A Data Register (PADR)

PADR stores output data for port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved The read value is undefined.
3	PA3DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose output port.
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

7.4.3 Port A Register (PORTA)

PORTA shows port A pin states. PORTA cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved The read value is undefined.
3	PA3	Undefined*	R	If a port A read is performed while PADDR bits are set to 1, the PADR values are read. If a port A read is performed while PADDR bits are cleared to 0, the pin states are read.
2	PA2	Undefined*	R	
1	PA1	Undefined*	R	
0	PA0	Undefined*	R	

Note: * Determined by the states of pins PA3 to PA0.

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

PAPCR controls the on/off state of the input pull-up MOS of port A.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved The read value is undefined.
3	PA3PCR	0	R/W	When a pin is specified as an input port, setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
2	PA2PCR	0	R/W	
1	PA1PCR	0	R/W	
0	PA0PCR	0	R/W	

7.4.5 Port A Open-Drain Control Register (PAODR)

PAODR specifies the output type of port A.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved The read value is undefined.
3	PA3ODR	0	R/W	When a pin is specified as an output port, setting the corresponding bit to 1 specifies pin output to open-drain and the PMOS to the off state. Clearing this bit to 0 specifies that to push-pull output.
2	PA2ODR	0	R/W	
1	PA1ODR	0	R/W	
0	PA0ODR	0	R/W	

7.4.6 Pin Functions

Port A pins also function as SCI_2 I/O pins. The correspondence between the register specification and the pin functions is shown below.

Table 7.10 PA3 Pin Function

CKE1	0				1
C/A	0			1	—
CKE0	0		1	—	—
PA3DDR	0	1	—	—	—
Pin function	PA3 input	PA3 output	SCK2 output	SCK2 output	SCK2 input

Table 7.11 PA2 Pin Function

RE	0		1
PA2DDR	0	1	—
Pin function	PA2 input	PA2 output	RxD2 input

Table 7.12 PA1 Pin Function

TE	0		1
PA1DDR	0	1	—
Pin function	PA1 input	PA1 output	TxD2 output

Table 7.13 PA0 Pin Function

PA0DDR	0	1
Pin function	PA0 input	PA0 output

7.5 Port B

Port B is an 8-bit I/O port that also has other functions. Port B has the following registers.

- 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)

7.5.1 Port B Data Direction Register (PBDDR)

PBDDR specifies whether the pins of port B are used for input or output. 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	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port 1 pin an output pin. Clearing this 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	

7.5.2 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	Output data for a pin is stored when the pin is specified as 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	

7.5.3 Port B Register (PORTB)

PORTB shows port B pin states. PORTB cannot be modified.

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

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

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

PBPCR controls the on/off state of the input pull-up MOS of port B.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7PCR	0	R/W	When a pin is specified as an input port, setting the corresponding bit to 1 turns on the input pull-up MOS 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	

7.5.5 Port B Open-Drain Control Register (PBODR)

PBODR specifies the output type of port B.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7ODR	0	R/W	When a pin function is specified as an output port, setting the corresponding bit to 1 specifies pin output as open-drain and the PMOS to the off state. Clearing this bit to 0 specifies push-pull 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	

7.5.6 Pin Functions

Port B pins also function as TPU I/O pins. The correspondence between the register specification and the pin functions is shown below.

Table 7.14 PB7 Pin Function

TPU channel 5 setting*	Output	Input or Initial Value	
PB7DDR	—	0	1
Pin function	TIOCB5 output	PB7 input	PB7 output
		TIOCB5 input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

Table 7.15 PB6 Pin Function

TPU channel 5 setting*	Output	Input or Initial Value	
PB6DDR	–	0	1
Pin function	TIOCA5 output	PB6 input	PB6 output
		TIOCA5 input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

Table 7.16 PB5 Pin Function

TPU channel 4 setting*	Output	Input or Initial Value	
PB5DDR	—	0	1
Pin function	TIOCB4 output	PB5 input	PB5 output
		TIOCB4 input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

Table 7.17 PB4 Pin Function

TPU channel 4 setting*	Output	Input or Initial Value	
PB4DDR	—	0	1
Pin function	TIOCA4 output	PB4 input	PB4 output
		TIOCA4 input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

Table 7.18 PB3 Pin Function

TPU channel 3 setting*	Output	Input or Initial Value	
PB3DDR	—	0	1
Pin function	TIOCD3 output	PB3 input	PB3 output
		TIOCD3 input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

Table 7.19 PB2 Pin Function

TPU channel 3 setting*	Output	Input or Initial Value	
PB2DDR	—	0	1
Pin function	TIOCC3 output	PB2 input	PB2 output
		TIOCC3 input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

Table 7.20 PB1 Pin Function

TPU channel 3 setting*	Output	Input or Initial Value	
PB1DDR	—	0	1
Pin function	TIOCB3 output	PB1 input	PB1 output
		TIOCB3 input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

Table 7.21 PB0 Pin Function

TPU channel 3 setting*	Output	Input or Initial Value	
PB0DDR	—	0	1
Pin function	TIOCA3 output	PB0 input	PB0 output
		TIOCA3 input	

Note: * For details on the TPU channel specification, refer to section 8, 16-Bit Timer Pulse Unit (TPU).

7.6 Port C

Port C is an 8-bit I/O port that also has other functions. Port C has the following registers.

- 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)

7.6.1 Port C Data Direction Register (PCDDR)

PCDDR specifies whether the pins of port C are used for input or output. 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	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port 1 pin an output pin. Clearing this 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	

7.6.2 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	Output data for a pin is stored when the pin is specified as 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	

7.6.3 Port C Register (PORTC)

PORTC shows port C pin states. PORTC cannot be modified.

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

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

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

PCPCR controls the on/off state of the input pull-up MOS of port C.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7PCR	0	R/W	When a pin is specified as an input port, setting the corresponding bit to 1 turns on the input pull-up MOS 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	

7.6.5 Port C Open-Drain Control Register (PCODR)

PCODR specifies an output type of port C.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7ODR	0	R/W	When a pin is specified as an output port, setting the corresponding bit to 1 specifies pin output as open-drain and the PMOS to the off state. Clearing this bit to 0 specifies push-pull output.
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	

7.6.6 Pin Functions

Port C pins also function as SCI_1 and SCI_0 I/O and interrupt input. The correspondence between the register specification and the pin functions is shown below.

Table 7.22 PC7 Pin Function

PC7DDR	0	1
Pin function	PC7 input	PC7 output

Table 7.23 PC6 Pin Function

PC6DDR	0	1
Pin function	PC6 input	PC6 output

Table 7.24 PC5 Pin Function

CKE1	0				1
C/$\overline{A}$	0			1	—
CKE0	0		1	—	—
PC5DDR	0	1	—	—	—
Pin function	PC5 input $\overline{IRQ5}$ input	PC5 output	SCK1 output	SCK1 output	SCK1 input

Table 7.25 PC4 Pin Function

RE	0		1
PC4DDR	0	1	—
Pin function	PC4 input	PC4 output	RxD1 input

Table 7.26 PC3 Pin Function

TE	0		1
PC3DDR	0	1	—
Pin function	PC3 input	PC3 output	TxD1 output

Table 7.27 PC2 Pin Function

CKE1	0				1
C/$\overline{\text{A}}$	0			1	—
CKE0	0		1	—	—
PC2DDR	0	1	—	—	—
Pin function	PC2 input $\overline{\text{IRQ4}}$ input	PC2 output	SCK0 output	SCK0 output	SCK0 input

Table 7.28 PC1 Pin Function

RE	0		1
PC1DDR	0	1	—
Pin function	PC1 input	PC1 output	RxD0 input

Table 7.29 PC0 Pin Function

TE	0		1
PC0DDR	0	1	—
Pin function	PC0 input	PC0 output	TxD0 output

7.7 Port D

Port D is a 4-bit I/O port that also has other functions. Port D has the following registers.

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

7.7.1 Port D Data Direction Register (PDDDR)

PDDDR specifies whether the pins of port D are used for input or output. 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	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port 1 pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	PD6DDR	0	W	
5	PD5DDR	0	W	
4	PD4DDR	0	W	
3 to 0	—	Undefined	—	Reserved The write value should always be 0.

7.7.2 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	Output data for a pin is stored when the pin is specified as a general purpose output port.
6	PD6DR	0	R/W	
5	PD5DR	0	R/W	
4	PD4DR	0	R/W	
3 to 0	—	Undefined	—	Reserved The read value is undefined.

7.7.3 Port D Register (PORTD)

PORTD shows port D pin states. PORTD cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7	Undefined*	R	If a port D read is performed while PDDDR bits are set to 1, the PDDR values are read. If a port D read is performed while PDDDR bits are cleared to 0, the pin states are read.
6	PD6	Undefined*	R	
5	PD5	Undefined*	R	
4	PD4	Undefined*	R	
3 to 0	—	Undefined	—	Reserved The read value is undefined.

Note: * Determined by the states of pins PD7 to PD4.

7.7.4 Port D Pull-up MOS Control Register (PDPCR)

PDPCR controls on/off states of the input pull-up MOS of port D.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7PCR	0	R/W	When the pin is in its input state, the input pull-up MOS of the input pin is on when the corresponding bit is set to 1.
6	PD6PCR	0	R/W	
5	PD5PCR	0	R/W	
4	PD4PCR	0	R/W	
3 to 0	—	Undefined	—	Reserved The write value should always be 0.

7.7.5 Pin Function

Port D is a 4-bit I/O port.

Table 7.30 PDn Pin Function

PDnDDR	0	1
Pin function	PDn input	PDn output

Legend: n = 7 to 4

7.8 Port F

Port F is an 8-bit I/O port that also has other functions. Port F has the following registers.

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

7.8.1 Port F Data Direction Register (PFDDR)

PFDDR specifies whether the pins of port F are used for input or output. PFDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the PF7 pin a ϕ output pin. Clearing this bit to 0 makes the pin an input pin.
6	PF6DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port F pin an output pin. Clearing this bit to 0 makes the pin an input pin.
5	PF5DDR	0	W	
4	PF4DDR	0	W	
3	PF3DDR	0	W	
2	PF2DDR	0	W	
1	PF1DDR	0	W	
0	PF0DDR	0	W	

7.8.2 Port F Data Register (PFDR)

PFDR stores output data for the port F pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The write value should always be 0.
6	PF6DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose output port.
5	PF5DR	0	R/W	
4	PF4DR	0	R/W	
3	PF3DR	0	R/W	
2	PF2DR	0	R/W	
1	PF1DR	0	R/W	
0	PF0DR	0	R/W	

7.8.3 Port F Register (PORTF)

PORTF shows port F pin states. PORTF cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7	Undefined*	R	If a port F read is performed while PFDDR bits are set to 1, the PFDR values are read. If a port F 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 the states of pins PF7 to PF0.

7.8.4 Pin Functions

Port F pins also function as external interrupt input ($\overline{\text{IRQ2}}$ and $\overline{\text{IRQ3}}$), A/D trigger input ($\overline{\text{ADTRG}}$), and system clock output (ϕ).

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

Table 7.31 PF7 Pin Function

PF7DDR	0	1
Pin function	PF7 input	ϕ output

Table 7.32 PF6 Pin Function

PF6DDR	0	1
Pin function	PF6 input	PF6 output

Table 7.33 PF5 Pin Function

PF5DDR	0	1
Pin function	PF5 input	PF5 output

Table 7.34 PF4 Pin Function

PF4DDR	0	1
Pin function	PF4 input	PF4 output

Table 7.35 PF3 Pin Function

PF3DDR	0	1
Pin function	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 used as an external interrupt input pin, do not use as an I/O pin for another function.

Table 7.36 PF2 Pin Function

PF2DDR	0	1
Pin function	PF2 input	PF2 output

Table 7.37 PF1 Pin Function

PF1DDR	0	1
Pin function	PF1 input	PF1 output

Table 7.38 PF0 Pin Function

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

Section 8 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 8.1 and figure 8.1, respectively.

8.1 Features

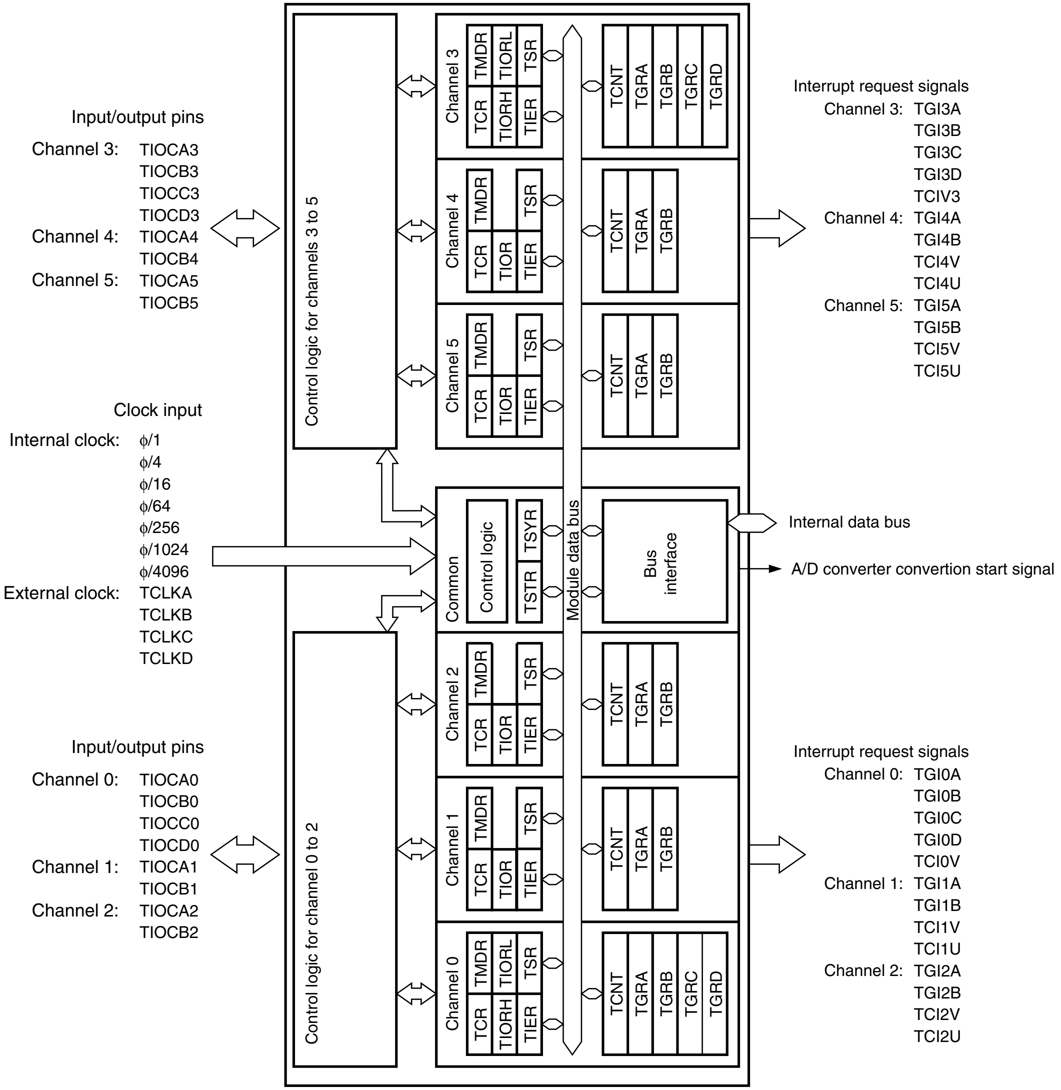
- Maximum 16-pulse input/output
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  Waveform output at compare match
  Input capture function
  Counter clear operation
  Synchronous operation:
  Multiple timer counters (TCNT) can be written 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
- A/D converter conversion start trigger can be generated
- Module stop mode can be set

Table 8.1 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		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
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
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


Input/output pins
Channel 3: TIOCA3 TIOCB3 TIOCC3 TIOCD3
Channel 4: TIOCA4 TIOCB4
Channel 5: TIOCA5 TIOCB5
Clock input
Internal clock: ϕ/1 ϕ/4 ϕ/16 ϕ/64 ϕ/256 ϕ/1024 ϕ/4096
External clock: TCLKA TCLKB TCLKC TCLKD
Input/output pins
Channel 0: TIOCA0 TIOCB0 TIOCC0 TIOCD0
Channel 1: TIOCA1 TIOCB1
Channel 2: TIOCA2 TIOCB2
Control logic for channels 3 to 5
Control logic for channel 0 to 2
Channel 3
TMDR TIORL TSR TCR TIORH TIER
TCNT TGRA TGRB TGRC TGRD
Channel 4
TMDR TSR TCR TIOR TIER
TCNT TGRA TGRB
Channel 5
TMDR TSR TCR TIOR TIER
TCNT TGRA TGRB
Common
Control logic
TSYR TSTR
Module data bus
Bus interface
Channel 2
TMDR TSR TCR TIOR TIER
TCNT TGRA TGRB
Channel 1
TMDR TSR TCR TIOR TIER
TCNT TGRA TGRB
Channel 0
TMDR TIORL TSR TCR TIORH TIER
TCNT TGRA TGRB TGRC TGRD
Interrupt request signals
Channel 3: TGI3A TGI3B TGI3C TGI3D TCIV3
Channel 4: TGI4A TGI4B TCI4V TCI4U
Channel 5: TGI5A TGI5B TCI5V TCI5U
Internal data bus
A/D converter convertion start signal
Interrupt request signals
Channel 0: TGI0A TGI0B TGI0C TGI0D TCI0V
Channel 1: TGI1A TGI1B TCI1V TCI1U
Channel 2: TGI2A TGI2B TCI2V TCI2U

Legend

TSTR:	Timer start register	TIOR(H, L)	Timer I/O control registers (H, L)
TSYR:	Timer synchro register	TIER:	Timer interrupt enable register
TCR:	Timer control register	TSR:	Timer status register
TMDR:	Timer mode register	TGR(A, B, C, D):	TImer general registers (A, B, C, D)

Figure 8.1 Block Diagram of TPU

8.2 Input/Output Pins

Table 8.2 Pin Configuration

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

8.3 Register Descriptions

The TPU has the following registers for each channel.

- 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)

8.3.1 Timer Control Register (TCR)

TCR controls 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 8.3 and 8.4 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 8.5 to 8.10 for details.
0	TPSC0	0	R/W	

Table 8.3 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 8.4 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 8.5 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 8.6 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 8.7 TPSC0 to TPSC2 (channels 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 8.8 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 8.9 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 8.10 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.

8.3.2 Timer Mode Register (TMDR)

TMDR sets 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	—	1	—	Reserved
6	—	1	—	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
2	MD2	0	R/W	These bits are used to set the timer operating mode.
1	MD1	0	R/W	MD3 is a reserved bit. In a write, it should always be written with 0. See table 8.11 for details.
0	MD0	0	R/W	

Table 8.11 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.

8.3.3 Timer I/O Control Register (TIOR)

TIOR controls TGR. 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 8.12 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 TIOCB0 pin Input capture at rising edge
			1		Capture input source is TIOCB0 pin Input capture at falling edge
		1	X		Capture input source is 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 8.13 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 TIOCD0 pin Input capture at rising edge
			1		Capture input source is TIOCD0 pin Input capture at falling edge
		1	X		Capture input source is 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 8.14 TIOR_1 (Channel 1)

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

Legend

X: Don’t care

Table 8.15 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 TIOCB2 pin Input capture at rising edge
			1		Capture input source is TIOCB2 pin Input capture at falling edge
		1	X		Capture input source is TIOCB2 pin Input capture at both edges

Legend

X: Don't care

Table 8.16 TIORH_3 (Channel 3)

Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description 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 TIOCB3 pin Input capture at rising edge
			1		Capture input source is TIOCB3 pin Input capture at falling edge
		1	X		Capture input source is 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 8.17 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 TIOCD3 pin Input capture at rising edge
			1		Capture input source is TIOCD3 pin Input capture at falling edge
		1	X		Capture input source is 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 8.18 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 TIOCB4 pin Input capture at rising edge
			1		Capture input source is TIOCB4 pin Input capture at falling edge
		1	X		Capture input source is 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 8.19 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 TIOCB5 pin Input capture at rising edge
			1		Capture input source is TIOCB5 pin Input capture at falling edge
		1	X		Capture input source is TIOCB5 pin Input capture at both edges

Legend
X: Don't care

Table 8.20 TIORH_0 (Channel 0)

Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description 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 TIOCA0 pin Input capture at rising edge
			1		Capture input source is TIOCA0 pin Input capture at falling edge
		1	X		Capture input source is 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

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 8.21 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 TIOCC0 pin Input capture at rising edge
			1		Capture input source is TIOCC0 pin Input capture at falling edge
		1	X		Capture input source is 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 8.22 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 TIOCA1 pin Input capture at rising edge
			1		Capture input source is TIOCA1 pin Input capture at falling edge
		1	X		Capture input source is TIOCA1 pin Input capture at both edges
	1	X	X		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 8.23 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 TIOCA2 pin Input capture at rising edge
			1		Capture input source is TIOCA2 pin Input capture at falling edge
		1	X		Capture input source is TIOCA2 pin Input capture at both edges

Legend

X: Don’t care

Table 8.24 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 TIOCA3 pin Input capture at rising edge
			1		Capture input source is TIOCA3 pin Input capture at falling edge
		1	X		Capture input source is 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

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 8.25 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*[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 TIOCC3 pin Input capture at rising edge
			1		Capture input source is TIOCC3 pin Input capture at falling edge
		1	X		Capture input source is 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*[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 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 8.26 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 TIOCA4 pin Input capture at rising edge
			1		Capture input source is TIOCA4 pin Input capture at falling edge
		1	X		Capture input source is 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 8.27 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 TIOCA5 pin Input capture at rising edge
			1		Capture input source is TIOCA5 pin Input capture at falling edge
		1	X		Capture input source is TIOCA5 pin Input capture at both edges

Legend
X: Don’t care

8.3.4 Timer Interrupt Enable Register (TIER)

TIER controls 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

8.3.5 Timer Status Register (TSR)

TSR indicates 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.

Bit	Bit Name	Initial value	R/W	Description
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. 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. [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
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. 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 condition] • When 0 is written to TGFD after reading TGFD = 1

Bit	Bit Name	Initial value	R/W	Description
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. 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 condition] • When 0 is written to TGFC after reading TGFC = 1
1	TGFB	0	R/(W)*	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGRB input capture or compare match. [Setting conditions] • When TCNT = TGRB 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 condition] • When 0 is written to TGFB after reading TGFB = 1
0	TGFA	0	R/(W)*	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TGRA input capture or compare match. [Setting conditions] • When TCNT = TGRA 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 condition] • When 0 is written to TGFA after reading TGFA = 1

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

8.3.6 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.

8.3.7 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.

8.3.8 Timer Start Register (TSTR)

TSTR selects the TCNT 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	—	0	—	Reserved
6	—	0	—	Only 0 should be written to these bits.
5	CST5	0	R/W	Counter Start 0 to 5 (CST0 to CST5)
4	CST4	0	R/W	These bits select operation or stoppage for TCNT.
3	CST3	0	R/W	If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the 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.
2	CST2	0	R/W	0: TCNT_0 to TCNT_5 count operation is stopped
1	CST1	0	R/W	1: TCNT_0 to TCNT_5 performs count operation
0	CST0	0	R/W	

8.3.9 Timer Synchro Register (TSYR)

TSYR selects independent operation or synchronous operation for channels 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	—	0	R/W	Reserved
6	—	0	R/W	Only 0 should be written to these bits.
5	SYNC5	0	R/W	Timer Synchro 0 to 5
4	SYNC4	0	R/W	These bits are used to select whether operation is independent of or synchronized with other channels.
3	SYNC3	0	R/W	
2	SYNC2	0	R/W	When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by counter clearing on another channel, are possible.
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	
				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

8.4 Operation

8.4.1 Basic Functions

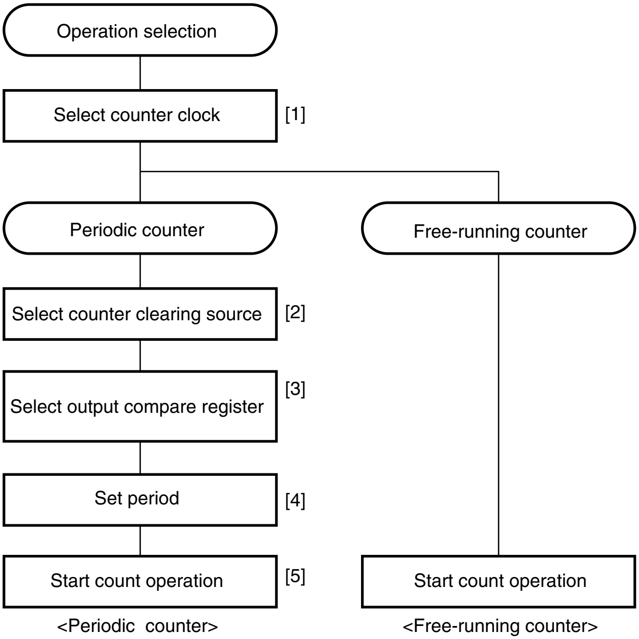
Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, synchronous counting, and external event counting.

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 8.2 shows an example of the count operation setting procedure.



[1] Select the counter clock with bits TPSC2 to TPSC0 in TCR. At the same time, select the input clock edge with bits CKEG1 and CKEG0 in TCR.

[2] For periodic counter operation, select the TGR to be used as the TCNT clearing source with bits CCLR2 to CCLR0 in TCR.

[3] Designate the TGR selected in [2] as an output compare register by means of TIOR.

[4] Set the periodic counter cycle in the TGR selected in [2].

[5] Set the CST bit in TSTR to 1 to start the counter operation.

Figure 8.2 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 8.3 illustrates free-running counter operation.

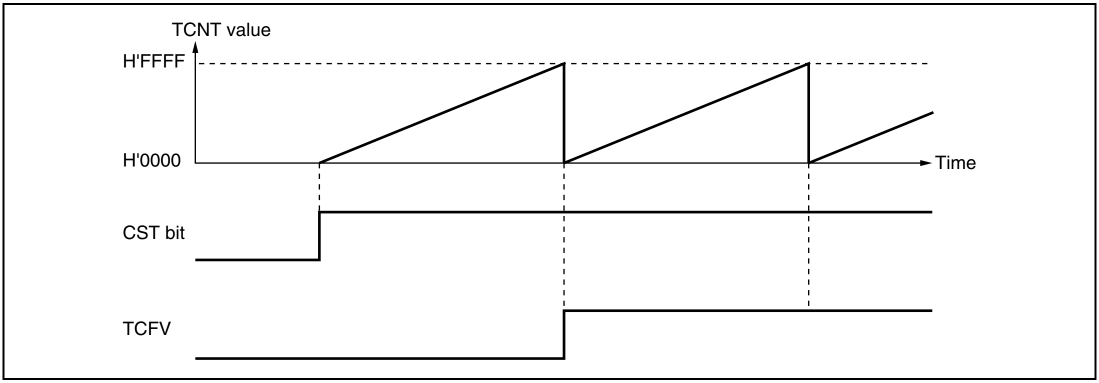


Figure 8.3 Free-Running Counter Operation

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 8.4 illustrates periodic counter operation.

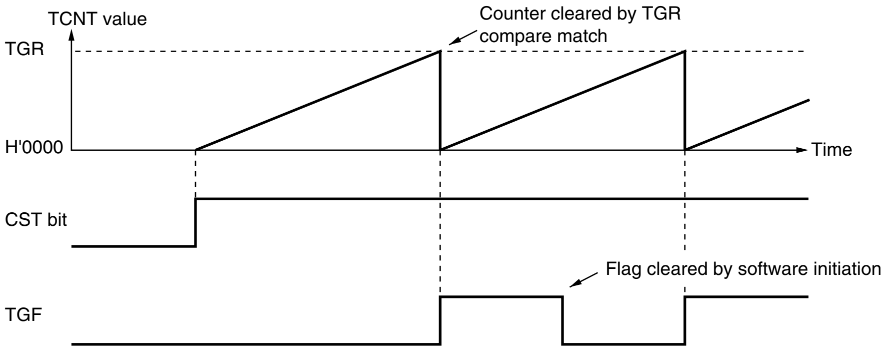
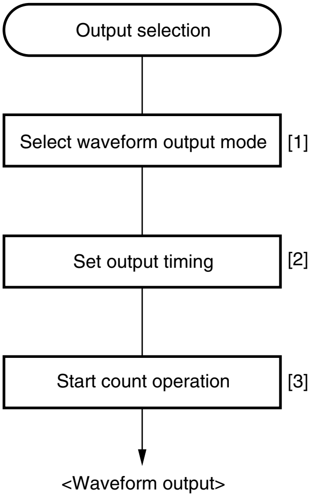


Figure 8.4 Periodic Counter Operation

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



[1] Select initial value 0 output or 1 output, and compare match output value 0 output, 1 output, or toggle output, by means of TIOR. The set initial value is output at the TIOC pin unit the first compare match occurs.

[2] Set the timing for compare match generation in TGR.

[3] Set the CST bit in TSTR to 1 to start the count operation.

Figure 8.5 Example of Setting Procedure for Waveform Output by Compare Match

2. Examples of waveform output operation

Figure 8.6 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.

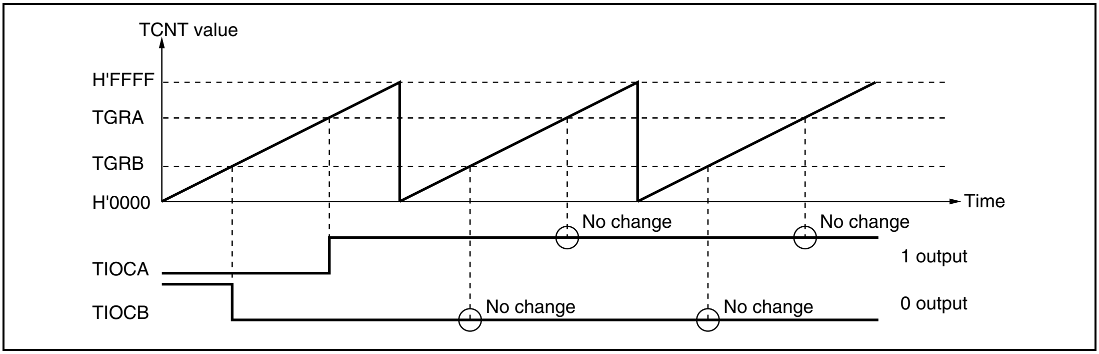


Figure 8.6   Example of 0 Output/1 Output Operation

Figure 8.7 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.

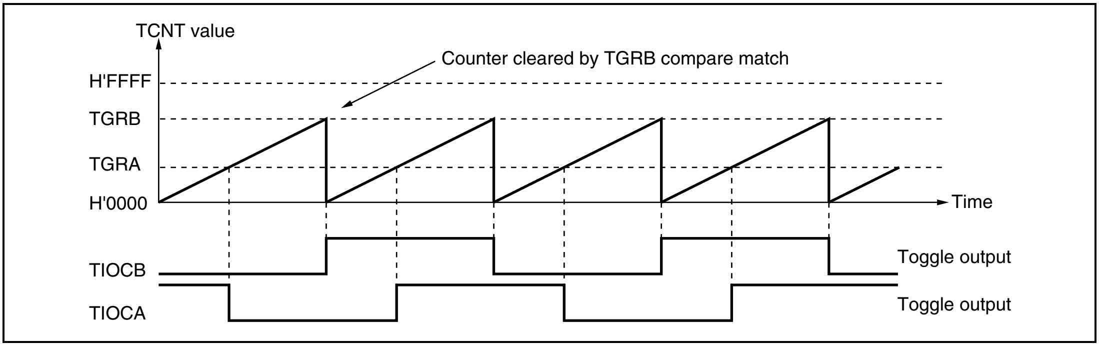


Figure 8.7   Example of Toggle Output Operation

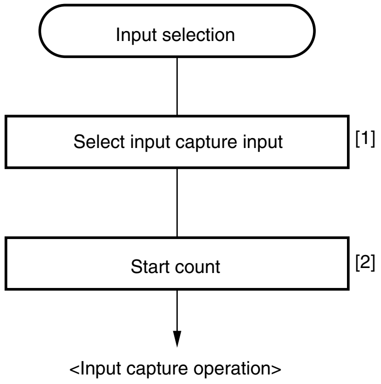
Input Capture Function: The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

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 8.8 shows an example of the input capture operation setting procedure.



[1] Designate TGR as an input capture register by means of TIOR, and select rising edge, falling edge, or both edges as the input capture source and input signal edge.

[2] Set the CST bit in TSTR to 1 to start the count operation.

Figure 8.8 Example of Input Capture Operation Setting Procedure

2. Example of input capture operation

Figure 8.9 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.

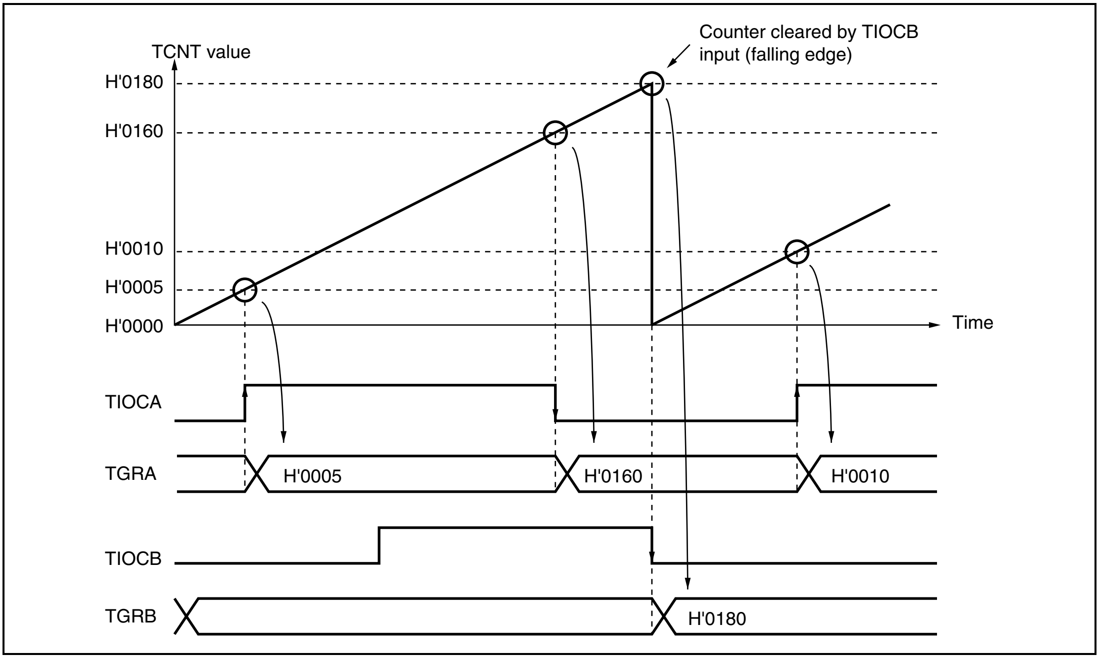


Figure 8.9 Example of Input Capture Operation

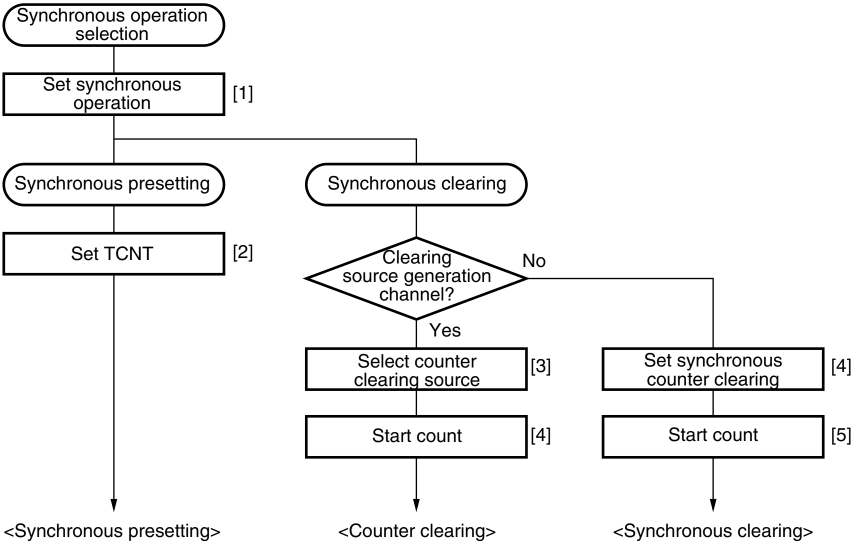
8.4.2 Synchronous Operation

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 8.10 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 8.10 Example of Synchronous Operation Setting Procedure

Example of Synchronous Operation: Figure 8.11 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 TIOC0A, TIOC1A, and TIOC2A. 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 8.4.5, PWM Modes.

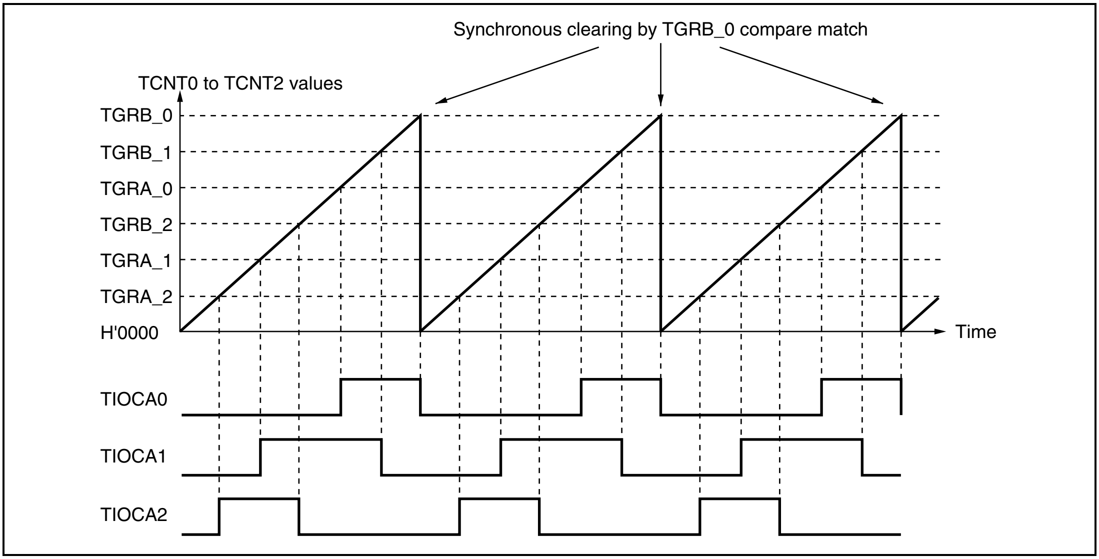


Figure 8.11 Example of Synchronous Operation

8.4.3 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 8.28 shows the register combinations used in buffer operation.

Table 8.28 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 8.12.

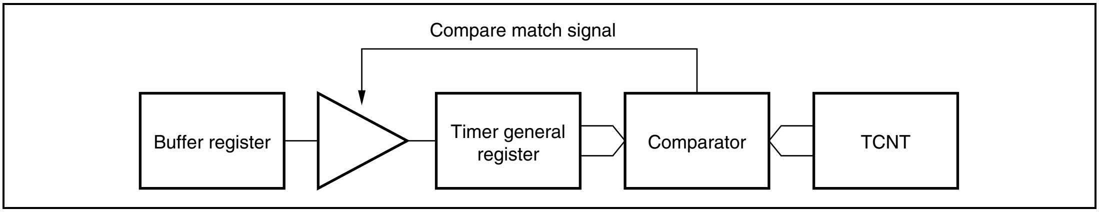


Figure 8.12 Compare Match Buffer Operation

- 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 8.13.

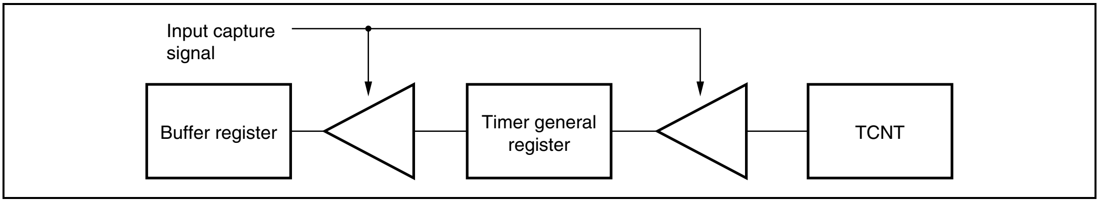
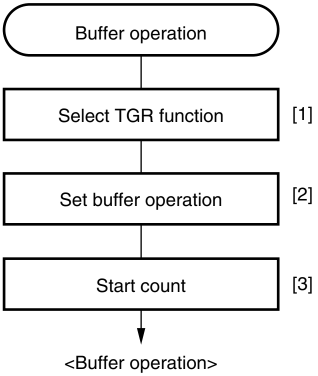


Figure 8.13 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 8.14 shows an example of the buffer operation setting procedure.



[1] Designate TGR as an input capture register or output compare register by means of TIOR.

[2] Designate TGR for buffer operation with bits BFA and BFB in TMDR.

[3] Set the CST bit in TSTR to 1 start the count operation.

Figure 8.14 Example of Buffer Operation Setting Procedure

Examples of Buffer Operation

1. When TGR is an output compare register

   Figure 8.15 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 8.4.5, PWM Modes.

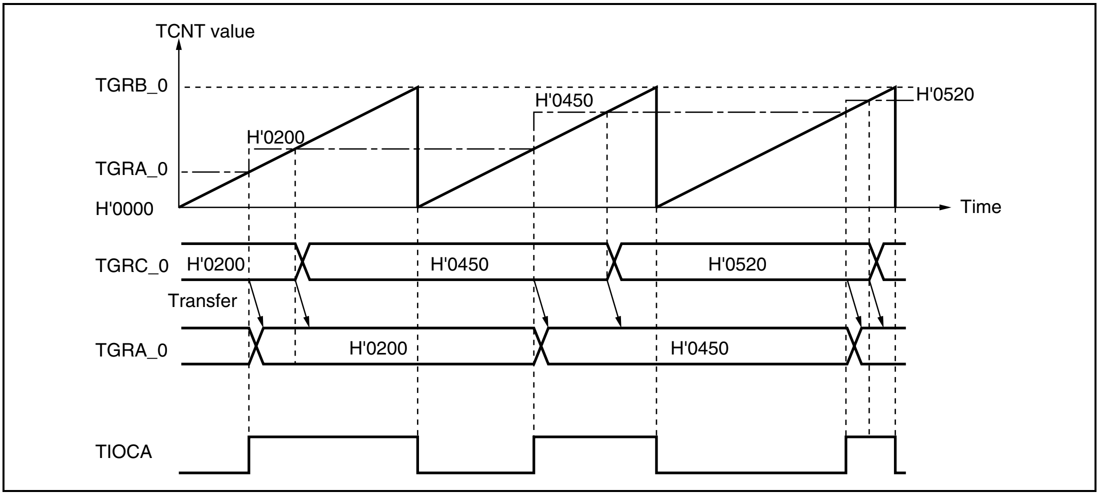


Figure 8.15 Example of Buffer Operation (1)

2. When TGR is an input capture register

   Figure 8.16 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.

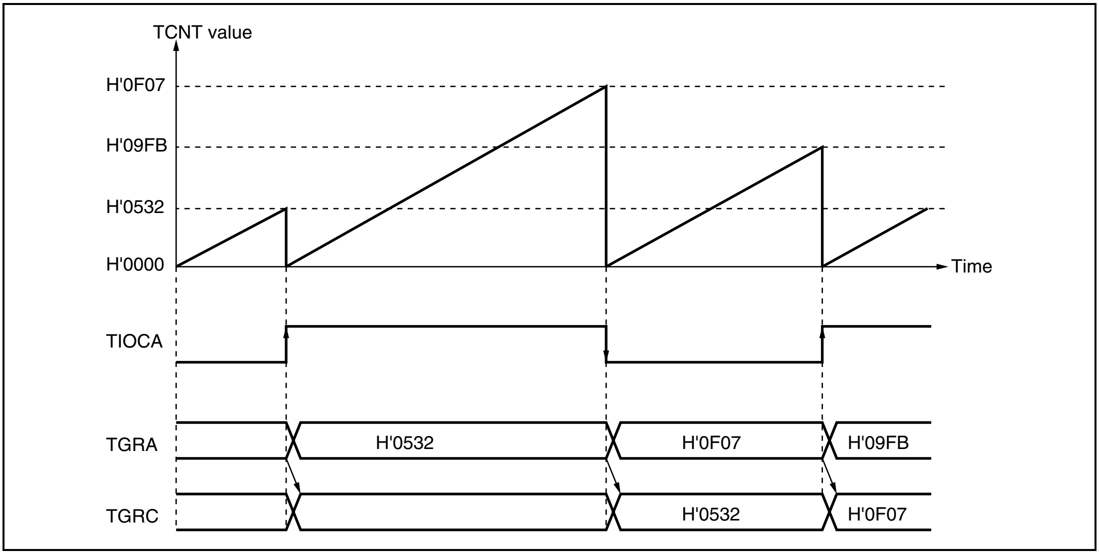


Figure 8.16 Example of Buffer Operation (2)

8.4.4 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.

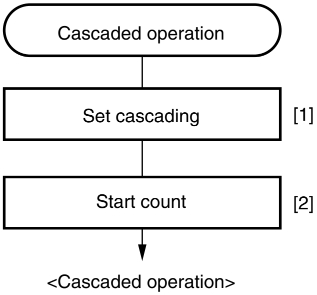
Table 8.29 shows the register combinations used in cascaded operation.

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 8.29 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 8.17 shows an example of the setting procedure for cascaded operation.



[1] Set bits TPSC2 to TPSC0 in the channel 1 (channel 4) TCR to B'1111 to select TCNT_2 (TCNT_5) overflow/underflow counting.

[2] Set the CST bit in TSTR for the upper and lower channel to 1 to start the count operation.

Figure 8.17 Cascaded Operation Setting Procedure

Examples of Cascaded Operation: Figure 8.18 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.

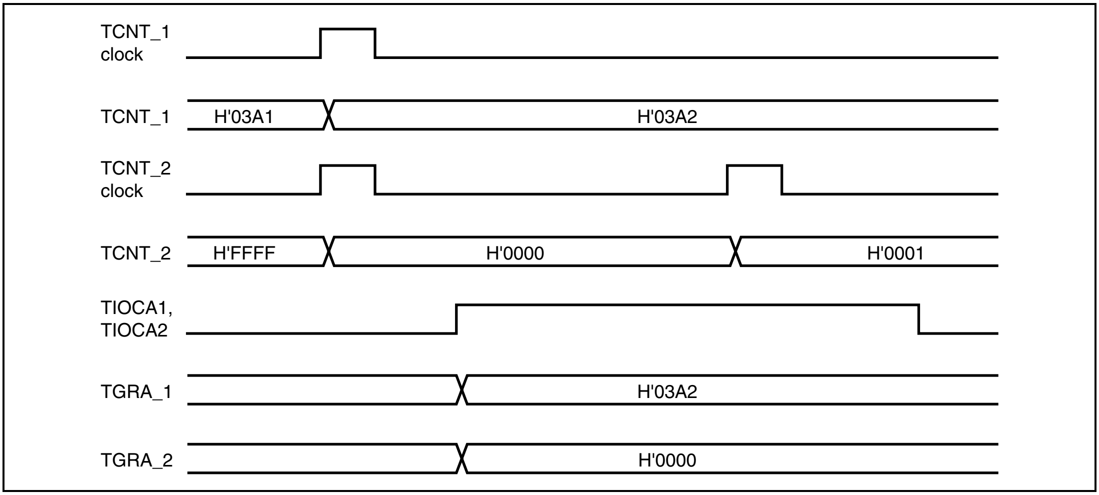


Figure 8.18 Example of Cascaded Operation (1)

Figure 8.19 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.

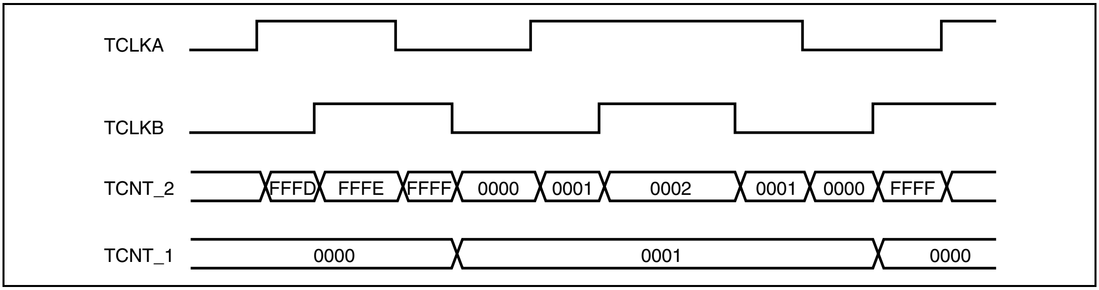


Figure 8.19 Example of Cascaded Operation (2)

8.4.5 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.

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

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

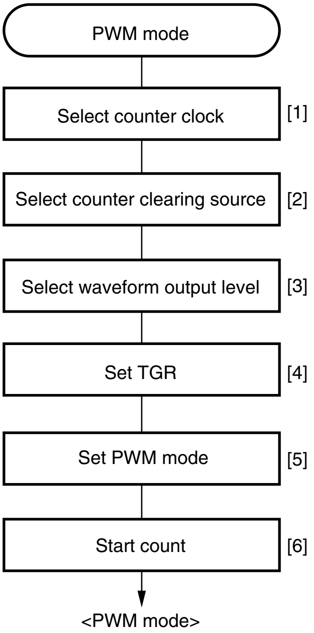
The correspondence between PWM output pins and registers is shown in table 8.30.

Table 8.30 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 8.20 shows an example of the PWM mode setting procedure.



[1] Select the counter clock with bits TPSC2 to TPSC0 in TCR. At the same time, select the input clock edge with bits CKEG1 and CKEG0 in TCR.

[2] Use bits CCLR2 to CCLR0 in TCR to select the TGR to be used as the TCNT clearing source.

[3] Use TIOR to designate the TGR as an output compare register, and select the initial value and output value.

[4] Set the cycle in the TGR selected in [2], and set the duty in the other the TGR.

[5] Select the PWM mode with bits MD3 to MD0 in TMDR.

[6] Set the CST bit in TSTR to 1 start the count operation.

Figure 8.20 Example of PWM Mode Setting Procedure

Examples of PWM Mode Operation: Figure 8.21 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 levels.

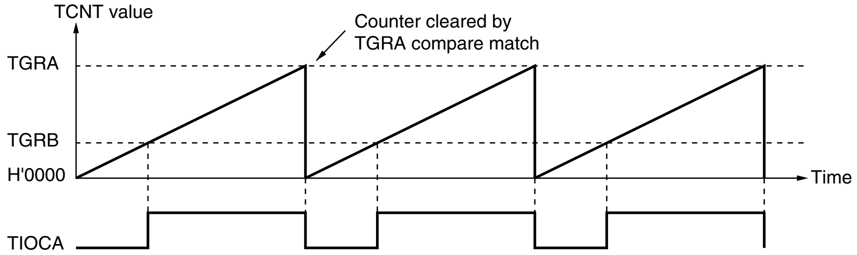


Figure 8.21 Example of PWM Mode Operation (1)

Figure 8.22 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 levels.

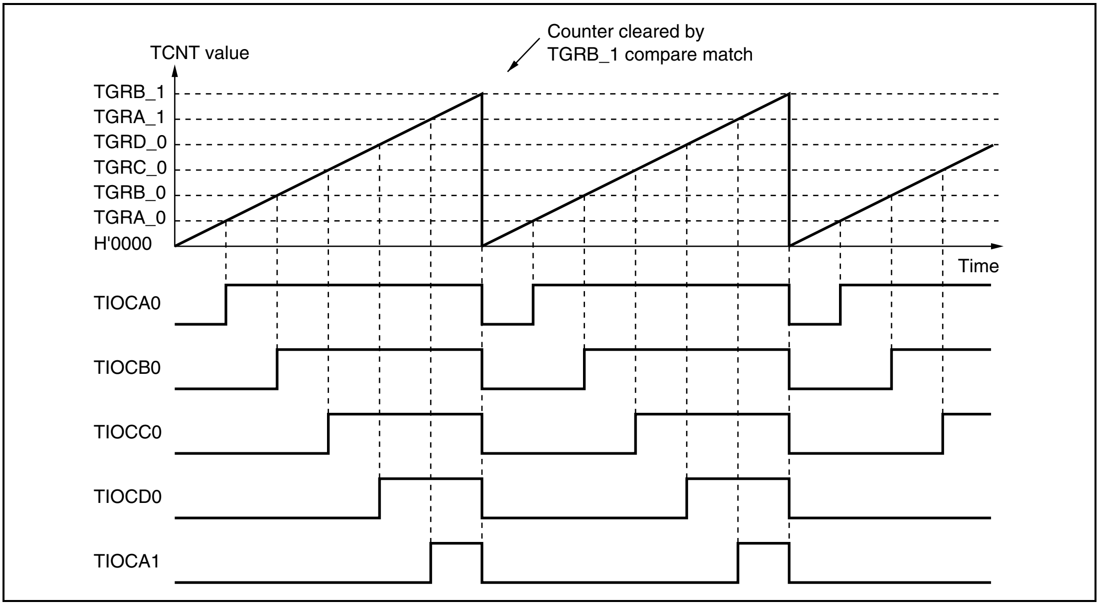


Figure 8.22 Example of PWM Mode Operation (2)

Figure 8.23 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

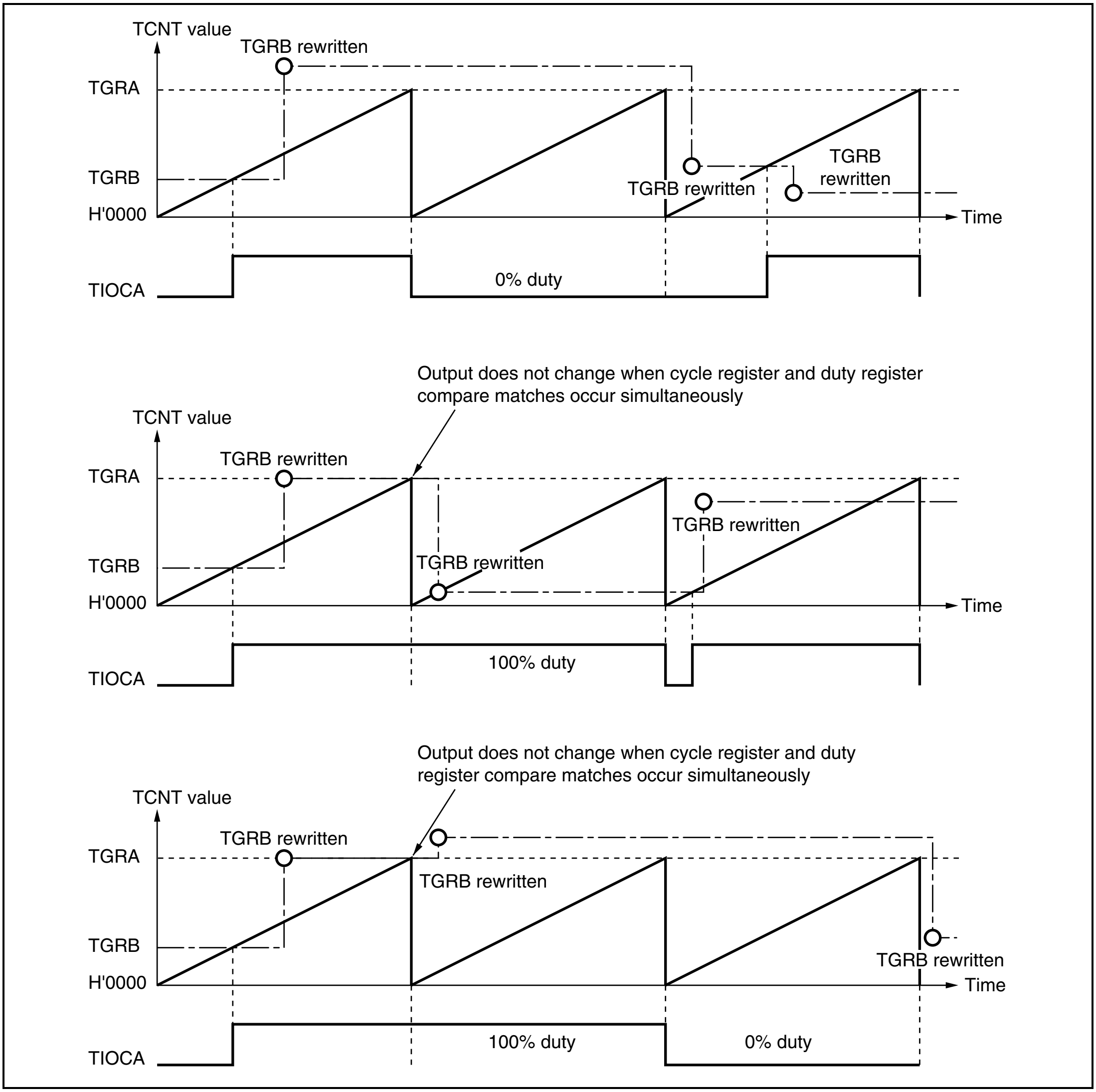


Figure 8.23 Example of PWM Mode Operation (3)

8.4.6 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.

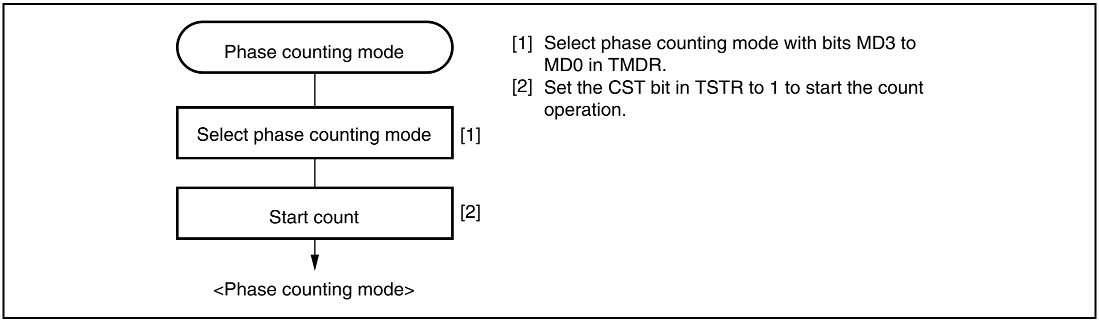
The TCFD bit in TSR is the count direction flag. Reading the TCFD flag reveals whether TCNT is counting up or down.

Table 8.31 shows the correspondence between external clock pins and channels.

Table 8.31 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 8.24 shows an example of the phase counting mode setting procedure.



[1] Select phase counting mode with bits MD3 to MD0 in TMDR.

[2] Set the CST bit in TSTR to 1 to start the count operation.

Figure 8.24 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 8.25 shows an example of phase counting mode 1 operation, and table 8.32 summarizes the TCNT up/down-count conditions.

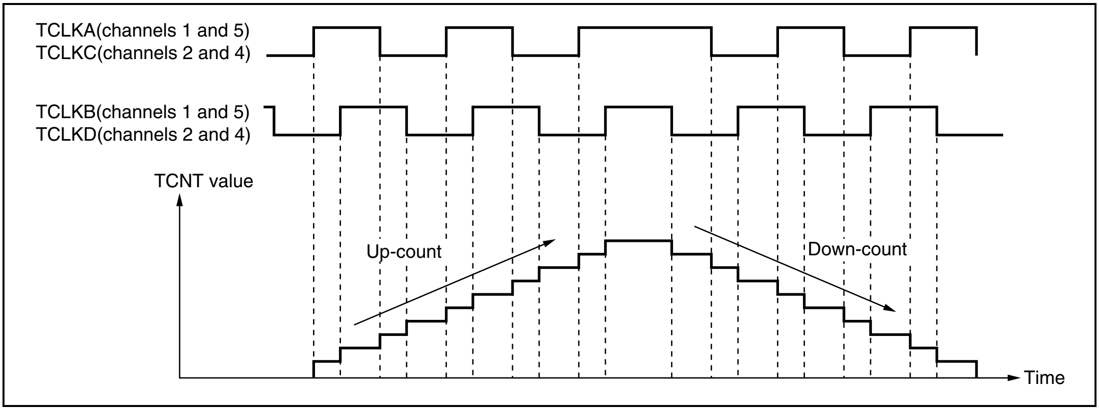


Figure 8.25 Example of Phase Counting Mode 1 Operation

Table 8.32 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 8.26 shows an example of phase counting mode 2 operation, and table 8.33 summarizes the TCNT up/down-count conditions.

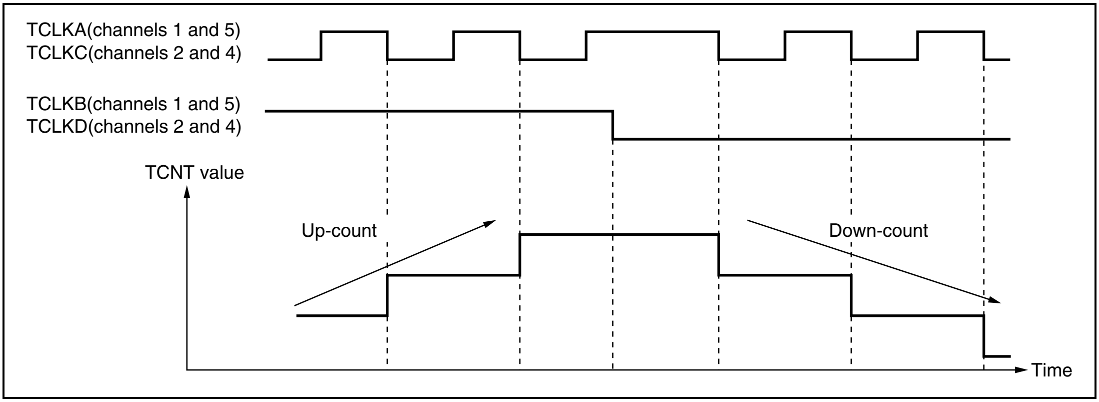


Figure 8.26 Example of Phase Counting Mode 2 Operation

Table 8.33 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 8.27 shows an example of phase counting mode 3 operation, and table 8.34 summarizes the TCNT up/down-count conditions.

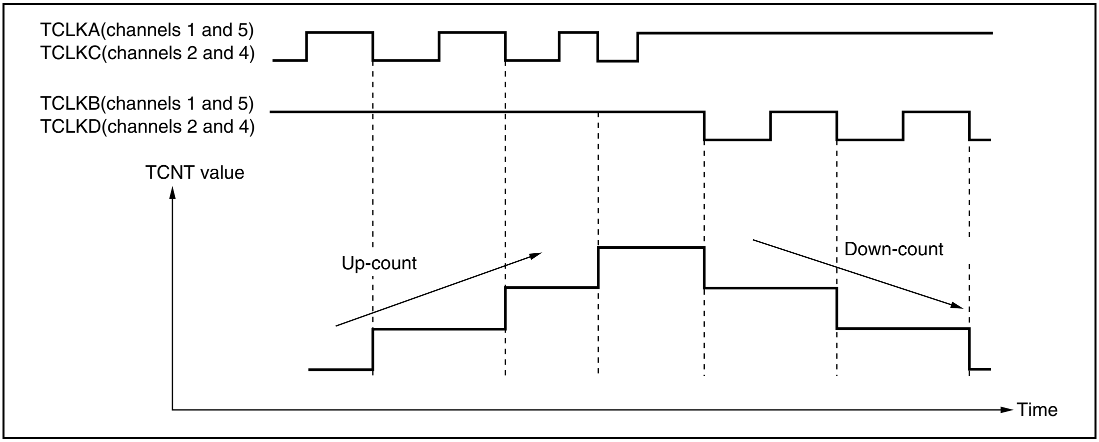


Figure 8.27 Example of Phase Counting Mode 3 Operation

Table 8.34 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 8.28 shows an example of phase counting mode 4 operation, and table 8.35 summarizes the TCNT up/down-count conditions.

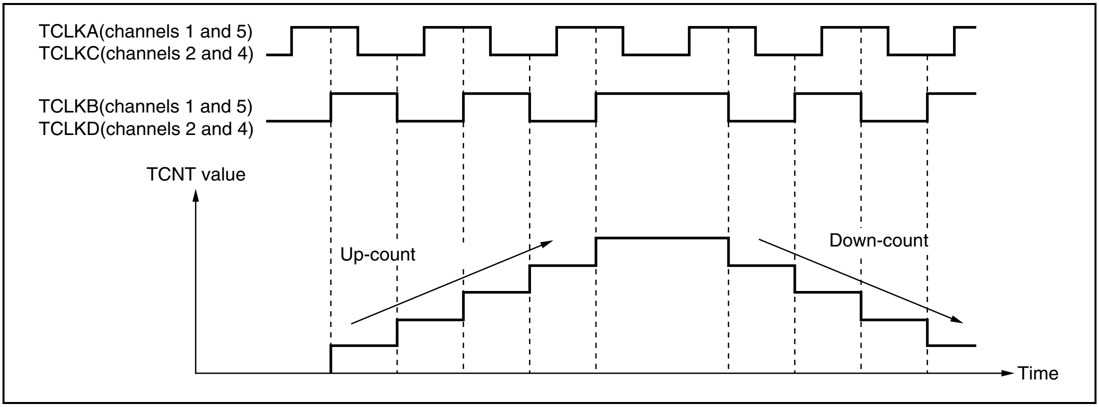


Figure 8.28 Example of Phase Counting Mode 4 Operation

Table 8.35 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 8.29 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.

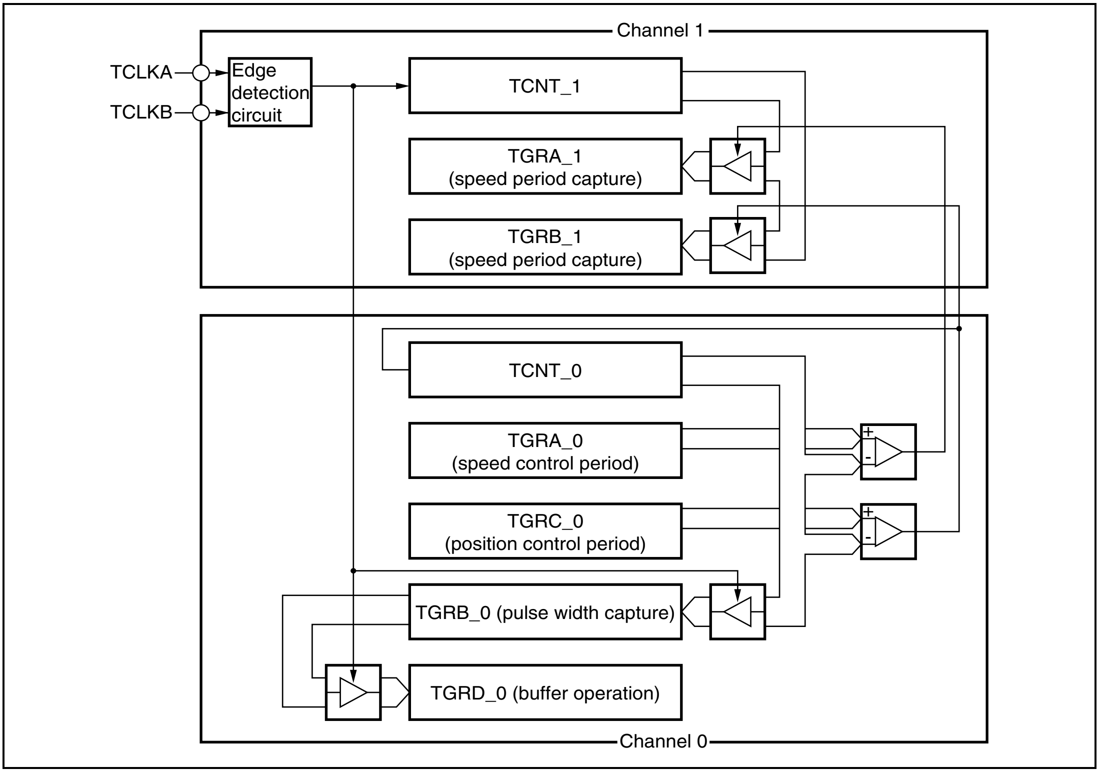

Channel 1
TCLKA
TCLKB
Edge detection circuit
TCNT_1
TGRA_1 (speed period capture)
TGRB_1 (speed period capture)
TCNT_0
TGRA_0 (speed control period)
TGRC_0 (position control period)
TGRB_0 (pulse width capture)
TGRD_0 (buffer operation)
+
-
+
-
Channel 0

Figure 8.29 Phase Counting Mode Application Example

8.5 Interrupts

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 8.36 lists the TPU interrupt sources.

Table 8.36 TPU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag
0	TGI0A	TGRA_0 input capture/compare match	TGFA_0
	TGI0B	TGRB_0 input capture/compare match	TGFB_0
	TGI0C	TGRC_0 input capture/compare match	TGFC_0
	TGI0D	TGRD_0 input capture/compare match	TGFD_0
	TCI0V	TCNT_0 overflow	TCFV_0
1	TGI1A	TGRA_1 input capture/compare match	TGFA_1
	TGI1B	TGRB_1 input capture/compare match	TGFB_1
	TCI1V	TCNT_1 overflow	TCFV_1
	TCI1U	TCNT_1 underflow	TCFU_1
2	TGI2A	TGRA_2 input capture/compare match	TGFA_2
	TGI2B	TGRB_2 input capture/compare match	TGFB_2
	TCI2V	TCNT_2 overflow	TCFV_2
	TCI2U	TCNT_2 underflow	TCFU_2
3	TGI3A	TGRA_3 input capture/compare match	TGFA_3
	TGI3B	TGRB_3 input capture/compare match	TGFB_3
	TGI3C	TGRC_3 input capture/compare match	TGFC_3
	TGI3D	TGRD_3 input capture/compare match	TGFD_3
	TCI3V	TCNT_3 overflow	TCFV_3
4	TGI4A	TGRA_4 input capture/compare match	TGFA_4
	TGI4B	TGRB_4 input capture/compare match	TGFB_4
	TCI4V	TCNT_4 overflow	TCFV_4
	TCI4U	TCNT_4 underflow	TCFU_4
5	TGI5A	TGRA_5 input capture/compare match	TGFA_5
	TGI5B	TGRB_5 input capture/compare match	TGFB_5
	TCI5V	TCNT_5 overflow	TCFV_5
	TCI5U	TCNT_5 underflow	TCFU_5

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Input Capture/Compare Match Interrupt: An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has 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.

8.6 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.

8.7 Operation Timing

8.7.1 Input/Output Timing

TCNT Count Timing: Figure 8.30 shows TCNT count timing in internal clock operation, and figure 8.31 shows TCNT count timing in external clock operation.

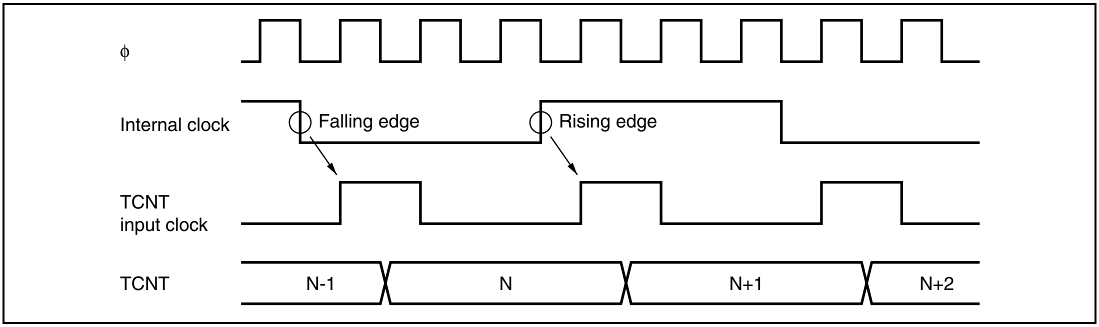


Figure 8.30 Count Timing in Internal Clock Operation

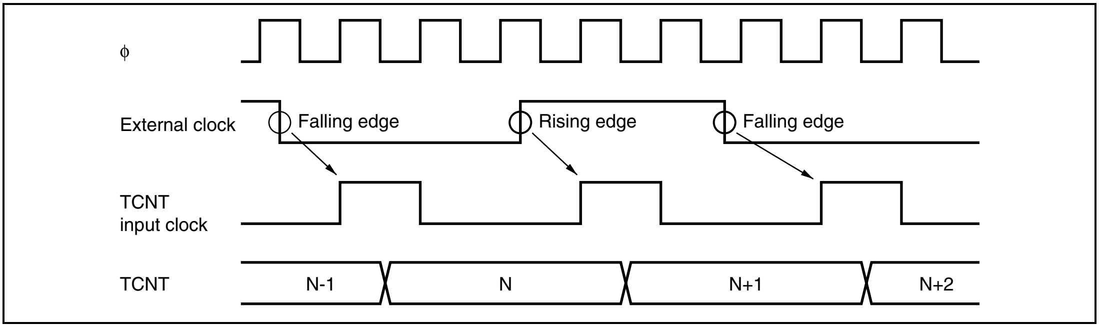


Figure 8.31 Count Timing in External Clock Operation

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 8.32 shows output compare output timing.

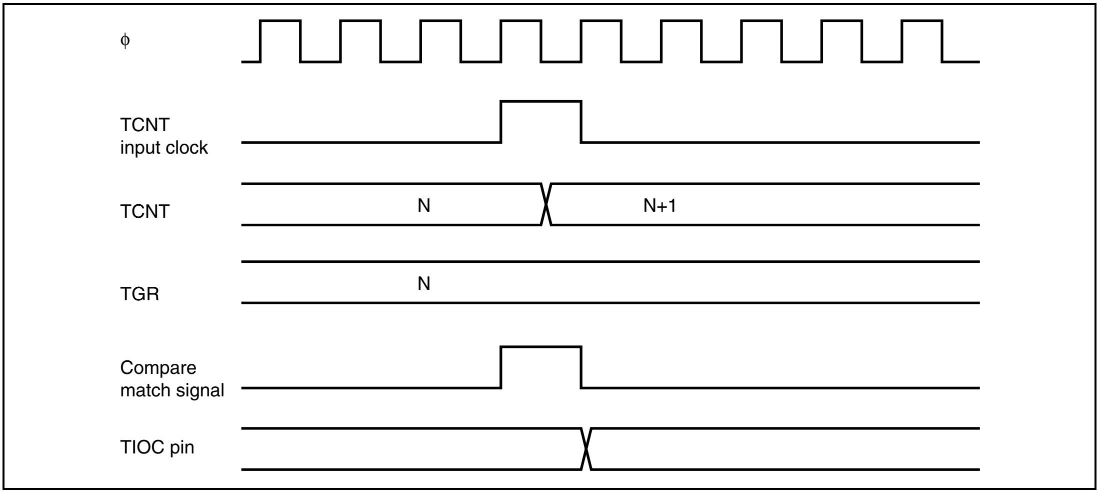


Figure 8.32 Output Compare Output Timing

Input Capture Signal Timing: Figure 8.33 shows input capture signal timing.

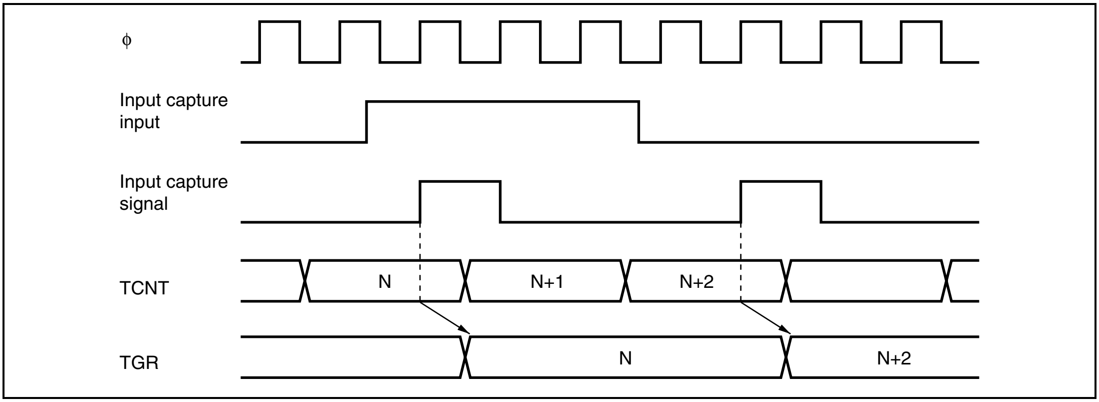


Figure 8.33 Input Capture Input Signal Timing

Timing for Counter Clearing by Compare Match/Input Capture: Figure 8.34 shows the timing when counter clearing on compare match is specified, and figure 8.35 shows the timing when counter clearing on input capture is specified.

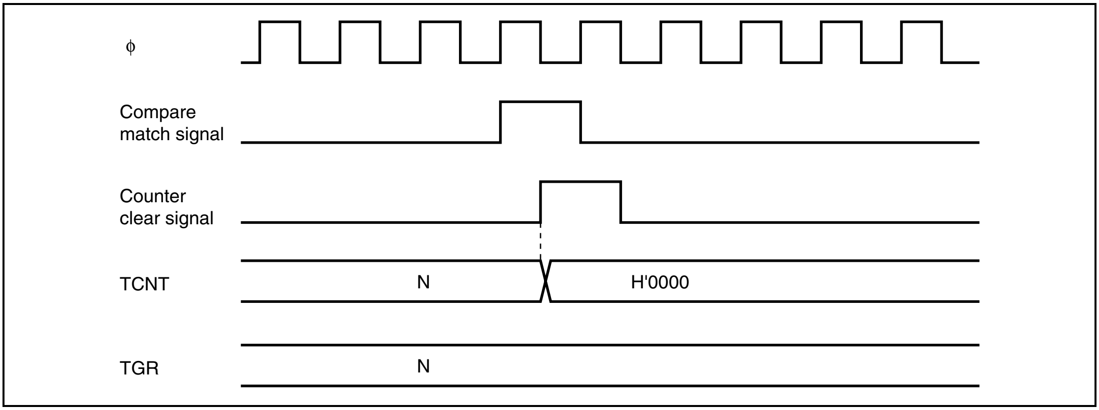


Figure 8.34 Counter Clear Timing (Compare Match)

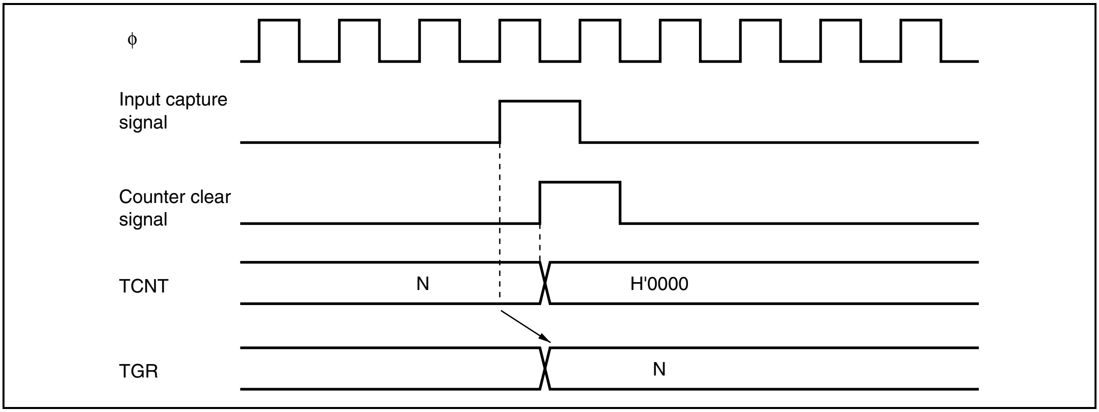


Figure 8.35 Counter Clear Timing (Input Capture)

Buffer Operation Timing: Figures 8.36 and 8.37 show the timing in buffer operation.

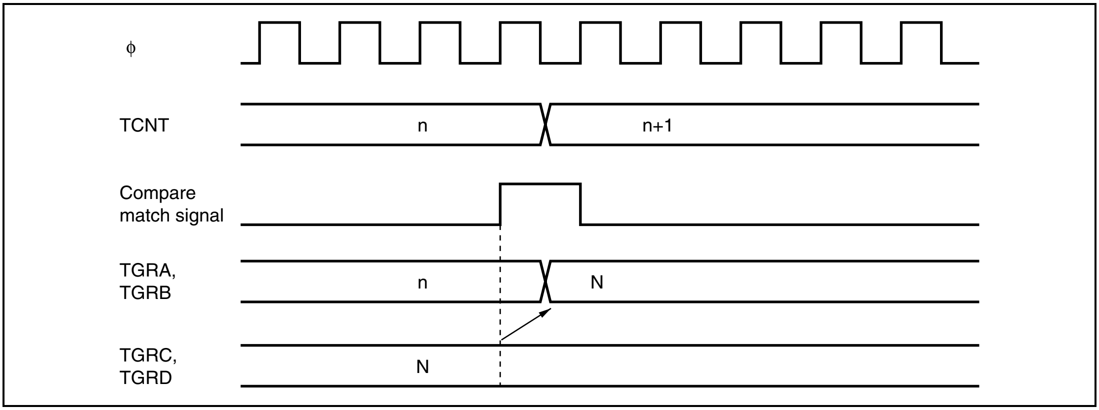


Figure 8.36 Buffer Operation Timing (Compare Match)

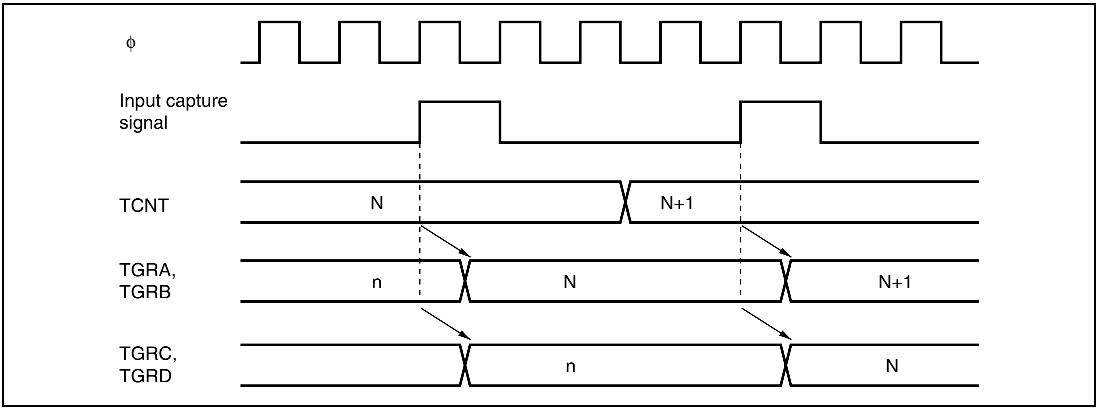


Figure 8.37 Buffer Operation Timing (Input Capture)

8.7.2 Interrupt Signal Timing

TGF Flag Setting Timing in Case of Compare Match: Figure 8.38 shows the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

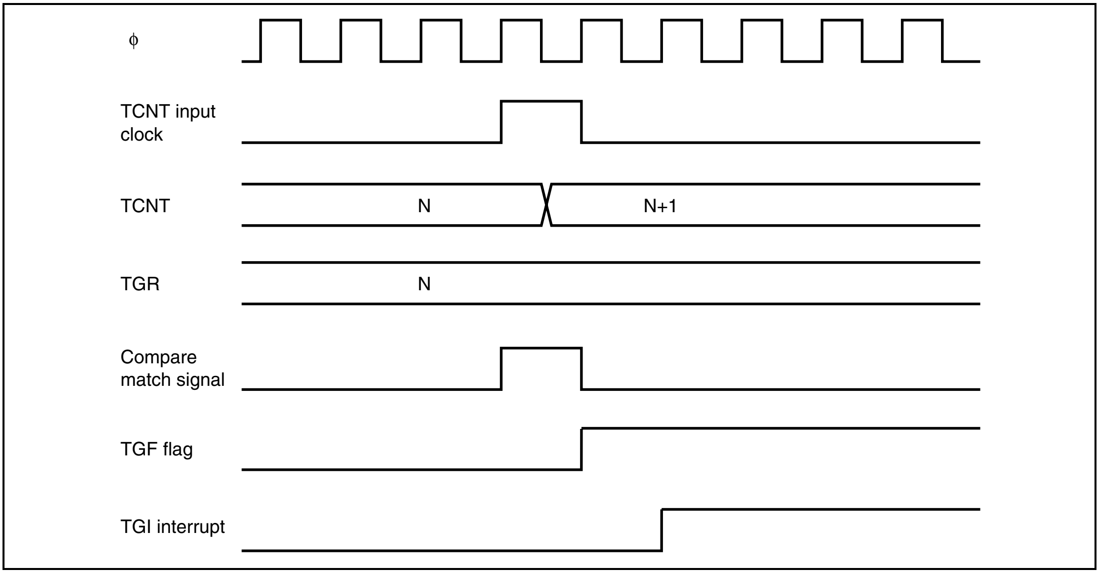


Figure 8.38 TGI Interrupt Timing (Compare Match)

TGF Flag Setting Timing in Case of Input Capture: Figure 8.39 shows the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

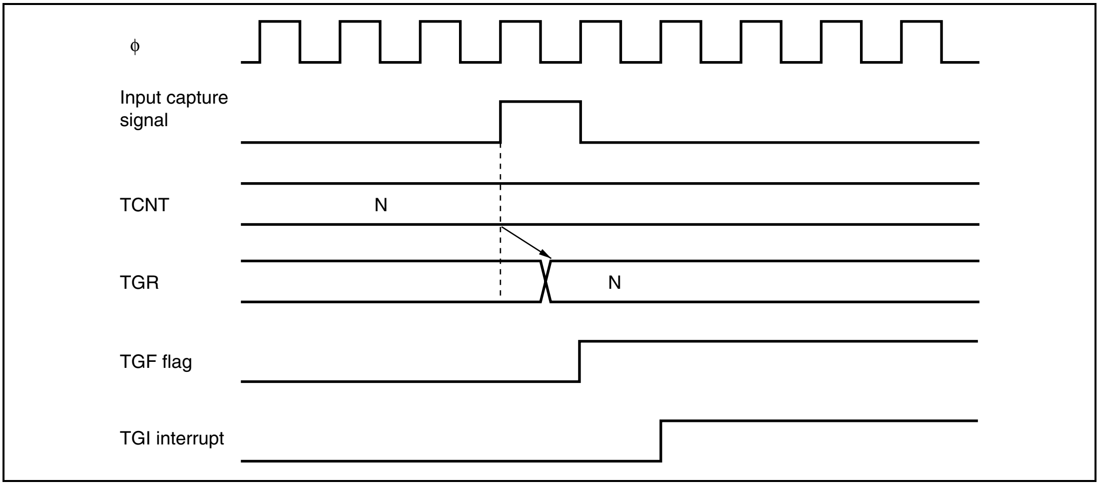


Figure 8.39 TGI Interrupt Timing (Input Capture)

TCFV Flag/TCFU Flag Setting Timing: Figure 8.40 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 8.41 shows the timing for setting of the TCFU flag in TSR on underflow, and TCIU interrupt request signal timing.

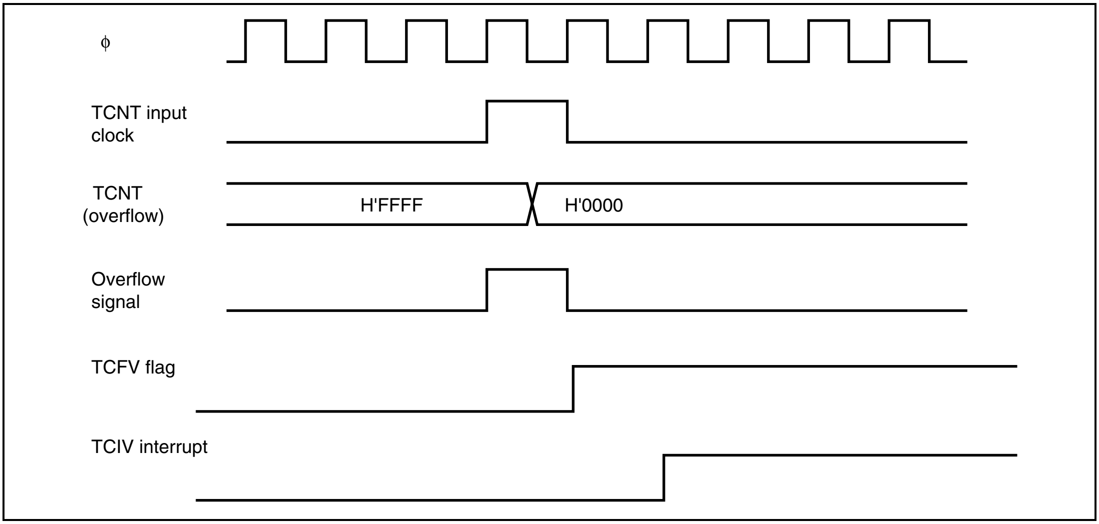


Figure 8.40 TCIV Interrupt Setting Timing

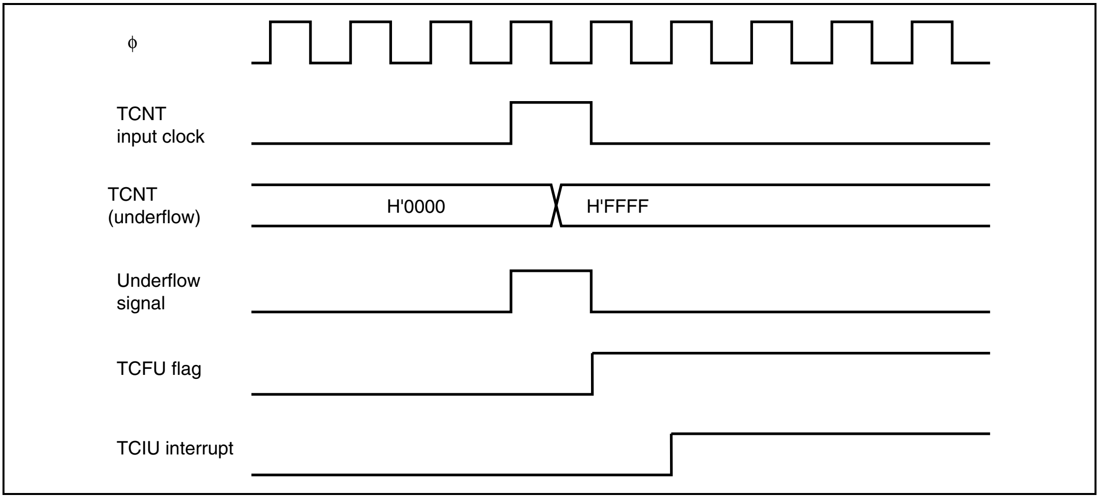


Figure 8.41 TCIU Interrupt Setting Timing

Status Flag Clearing Timing: After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. Figure 8.42 shows the timing for status flag clearing by the CPU.

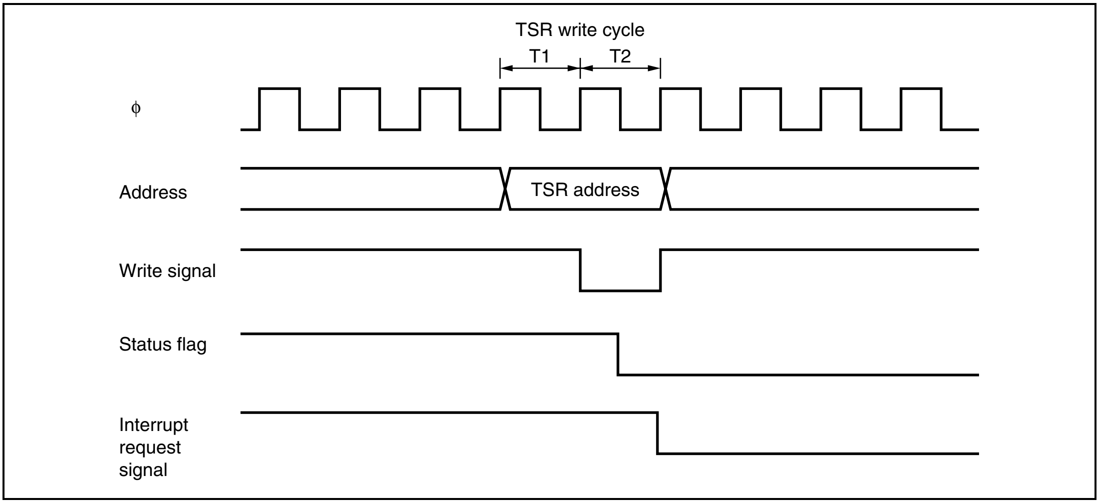


Figure 8.42 Timing for Status Flag Clearing by CPU

8.8 Usage Notes

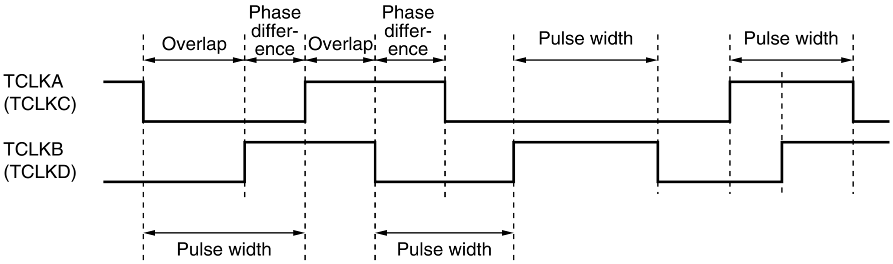
8.8.1 Module Stop Mode Setting

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 16, Power-Down Modes.

8.8.2 Input Clock Restrictions

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

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 8.43 shows the input clock conditions in phase counting mode.



Notes: Phase difference and overlap : 1.5 states or more
Pulse width : 2.5 states or more

Figure 8.43 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

8.8.3 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

8.8.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed.

Figure 8.44 shows the timing in this case.

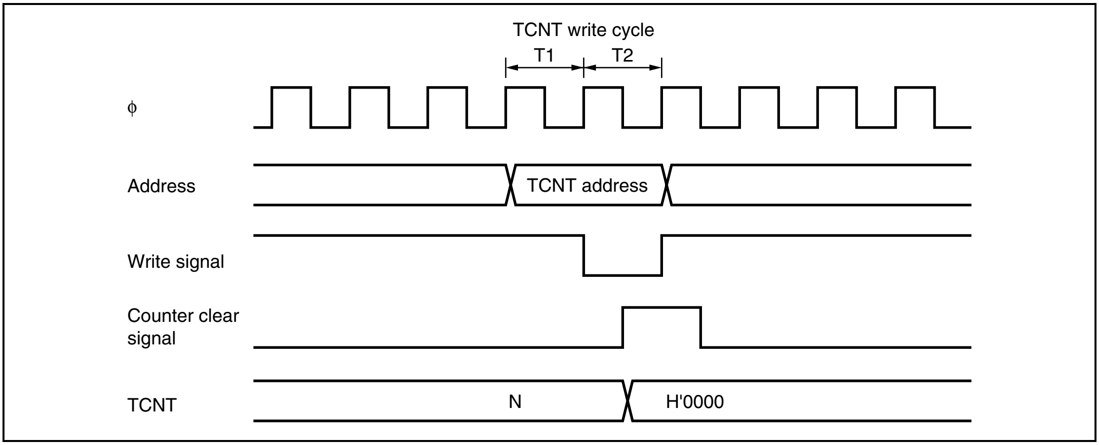


Figure 8.44 Contention between TCNT Write and Clear Operations

8.8.5 Contention 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 8.45 shows the timing in this case.

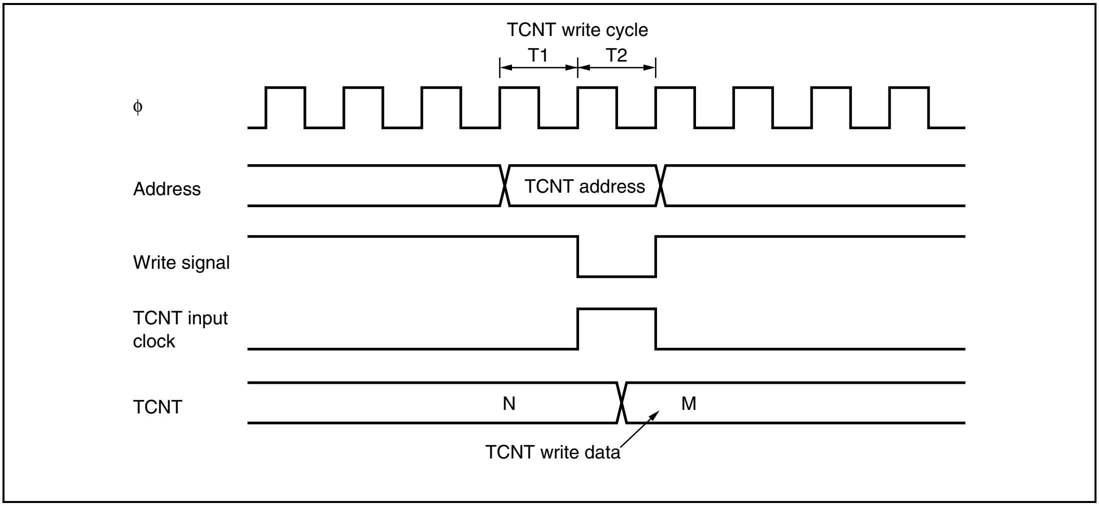


Figure 8.45 Contention between TCNT Write and Increment Operations

8.8.6 Contention between TGR Write and Compare Match

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

Figure 8.46 shows the timing in this case.

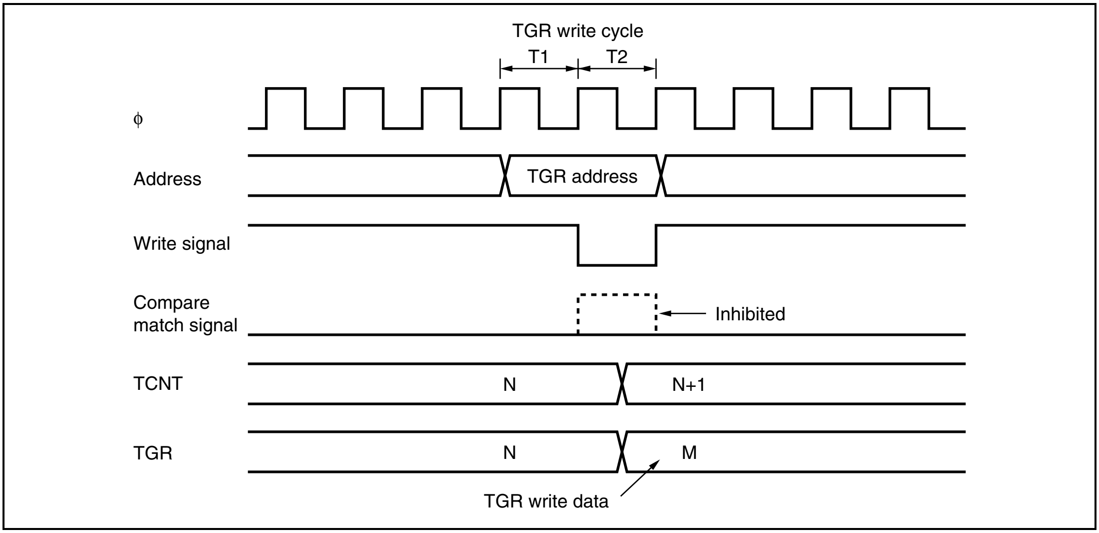


Figure 8.46 Contention between TGR Write and Compare Match

8.8.7 Contention 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 8.47 shows the timing in this case.

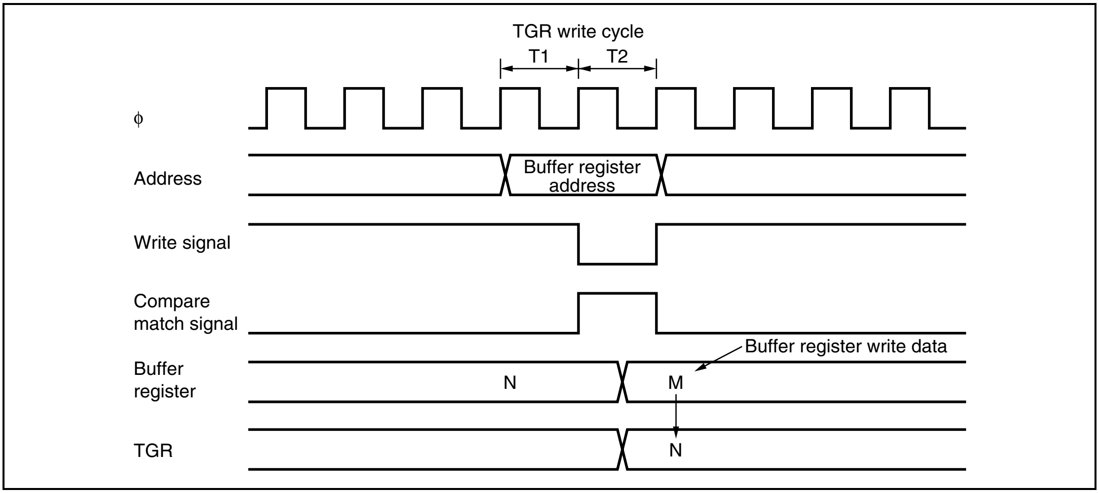


Figure 8.47 Contention between Buffer Register Write and Compare Match

8.8.8 Contention 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 8.48 shows the timing in this case.

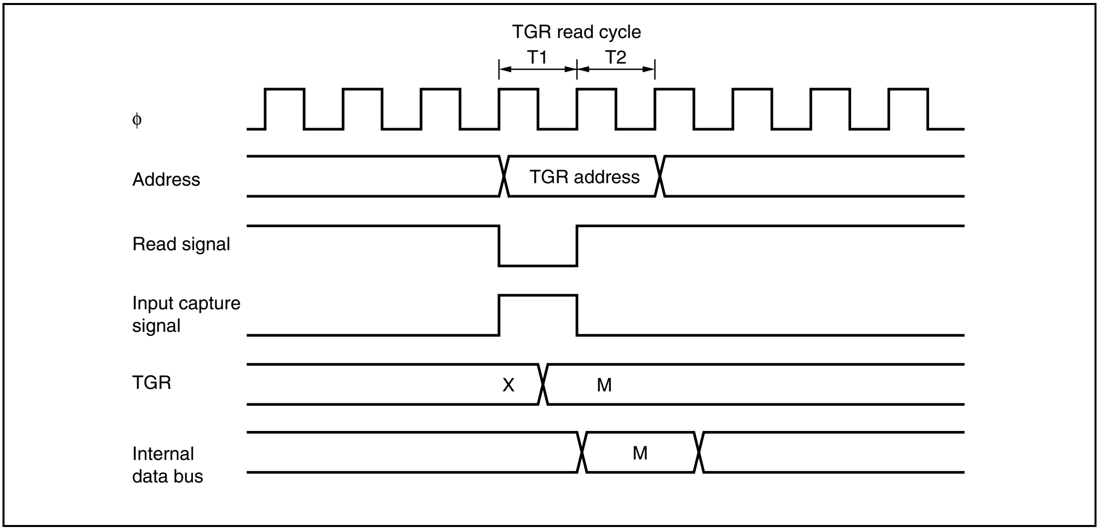


Figure 8.48 Contention between TGR Read and Input Capture

8.8.9 Contention between TGR Write and Input Capture

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

Figure 8.49 shows the timing in this case.

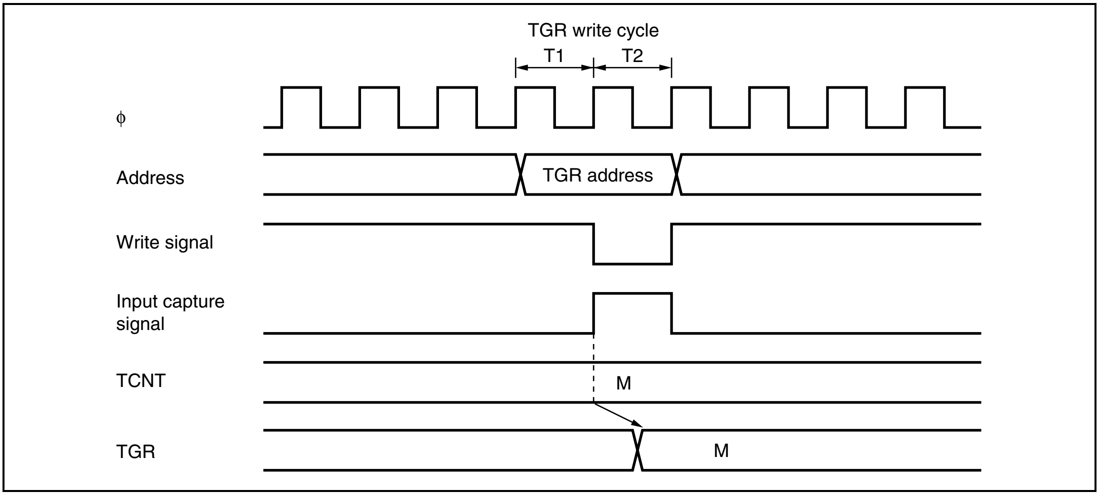


Figure 8.49 Contention between TGR Write and Input Capture

8.8.10 Contention 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 8.50 shows the timing in this case.

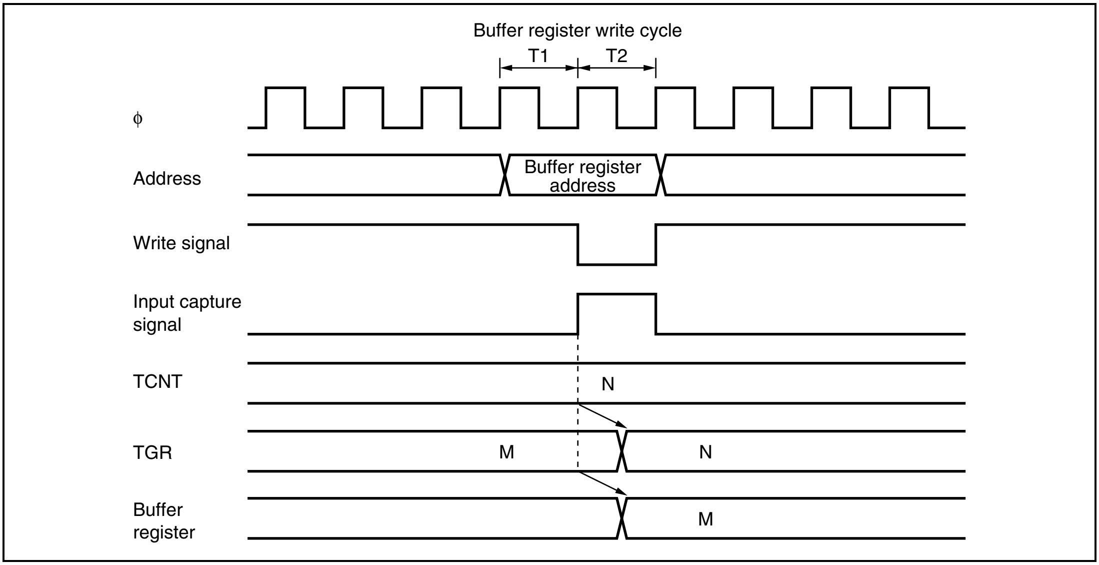


Figure 8.50 Contention between Buffer Register Write and Input Capture

8.8.11 Contention between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

Figure 8.51 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

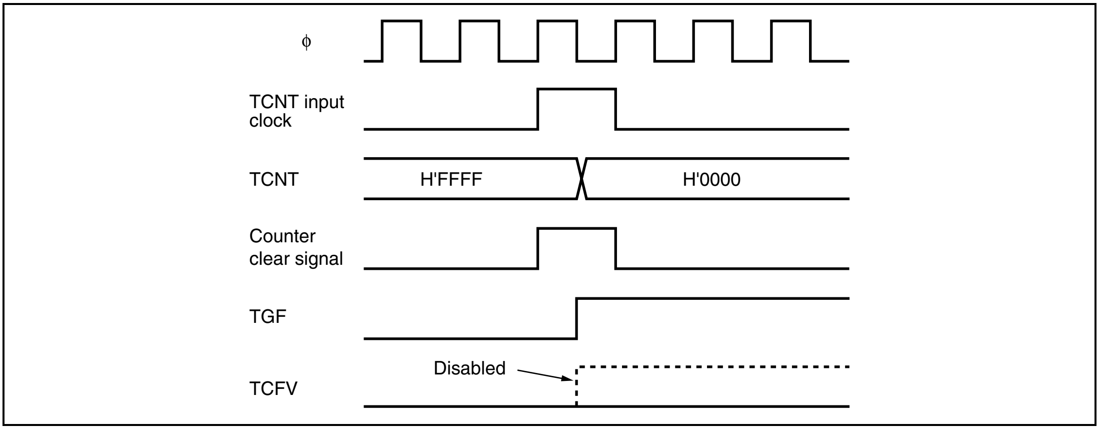


Figure 8.51 Contention between Overflow and Counter Clearing

8.8.12 Contention between TCNT Write and Overflow/Underflow

If there is an up-count or down-count in the T2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 8.52 shows the operation timing when there is contention between TCNT write and overflow.

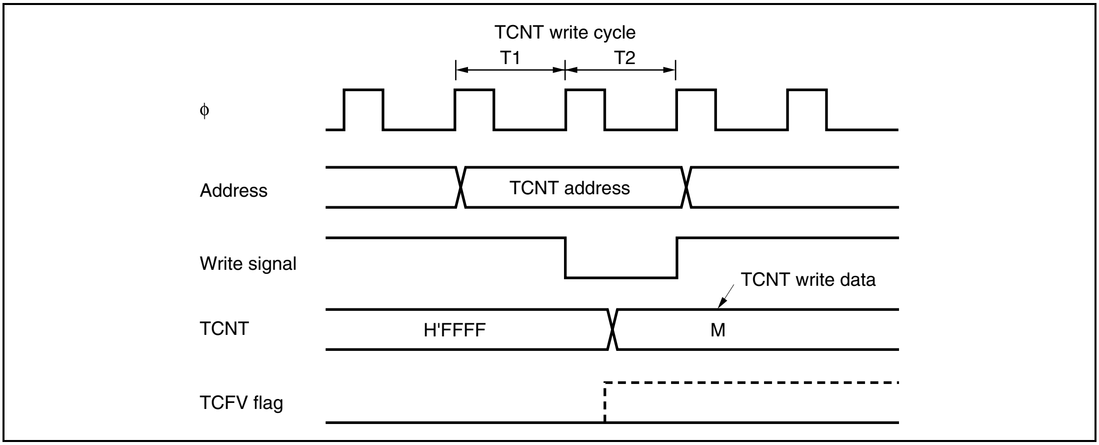


Figure 8.52 Contention between TCNT Write and Overflow

8.8.13 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.

8.8.14 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. Interrupts should therefore be disabled before entering module stop mode.

Section 9 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 9.1 and 9.2, respectively.

9.1 Features

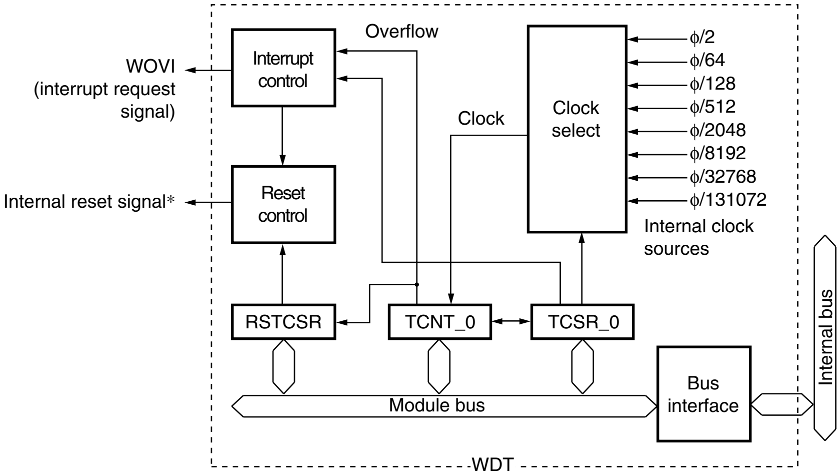
- Selectable from eight counter input clocks (WDT_0) or sixteen counter input clocks (WDT_1)
- Switchable between watchdog timer mode and interval timer mode

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

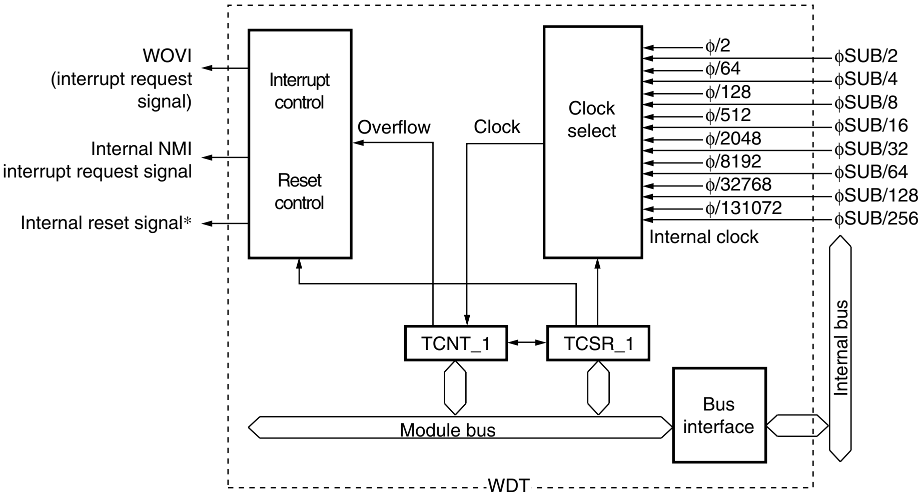
- If the counter overflows, the WDT generates an interval timer interrupt (WOVI).



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 9.1 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 9.2 Block Diagram of WDT_1

9.2 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, refer to section 9.5.1, 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)

9.2.1 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.

9.2.2 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 condition] When TCNT overflows (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset. [Clearing conditions] Cleared by reading TCSR when OVF = 1, then writing 0 to OVF
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

Bit	Bit Name	Initial Value	R/W	Description
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting and is initialized to H'00.
4	—	1	—	Reserved
3	—	1	—	These bits are always read as 1 and cannot be modified.
2	CKS2	0	R/W	Clock Select 0 to 2
1	CKS1	0	R/W	Selects the clock source to be input to TCNT. The overflow frequency for φ = 20 MHz is enclosed in parentheses.
0	CKS0	0	R/W	000: 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)

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 condition] When TCNT overflows (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset. [Clearing condition] Cleared by reading TCSR when OVF = 1, then writing 0 to OVF
6	WT/$\overline{\text{IT}}$	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/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 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. 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: 13.1 ms) 001: φSUB/4 (cycle: 26.2 ms) 010: φSUB/8 (cycle: 52.4 ms) 011: φSUB/16 (cycle: 104.9 ms) 100: φSUB/32 (cycle: 209.7 ms) 101: φSUB/64 (cycle: 419.4 ms) 110: φSUB/128 (cycle: 838.9 ms) 111: φSUB/256 (cycle: 1.6777 s)
1	CKS1	0	R/W	
0	CKS0	0	R/W	

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

9.2.3 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.

9.3 Operation

9.3.1 Watchdog Timer Mode

To use the WDT as a watchdog timer, set the WT/$\overline{\text{IT}}$ 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, a WDTOVF signal is output when using the WDT_0.

In watchdog timer mode, the WDT can internally reset this LSI with a WDTOVF signal.

When the RSTE bit of the RSTCSR is set to 1, and if the TCNT overflows, an internal reset signal for this LSI is issued at the same time as a WDTOVF signal. In this case, select power-on reset by setting the RSTS bit in RSTCSR to 0.

If a reset caused by a signal input to the $\overline{\text{RES}}$ pin occurs at the same time as a reset caused by a WDT overflow, the $\overline{\text{RES}}$ pin reset has priority and the WOVF bit in RSTCSR is cleared to 0.

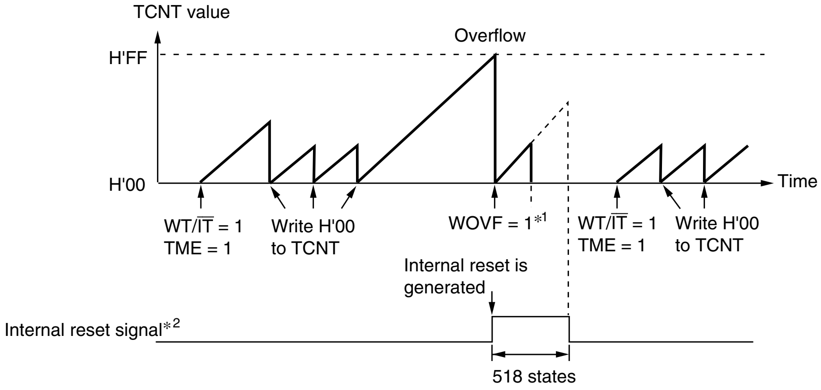
The WDTOVF signal is output for 132 states when the RSTE bit = 1 in RSTCSR, and for 130 states when the RSTE bit = 0.

The internal reset signal is output for 518 states. This is illustrated in figure 9.3 (a).

When the TCNT overflows in watchdog timer mode, the WOVF bit in RSTCSR is set to 1. If the RSTE bit in RSTCSR has been set to 1, an internal reset signal for the entire LSI is generated at TCNT overflow.

In the case of the WDT_1, the chip 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 9.3 (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.



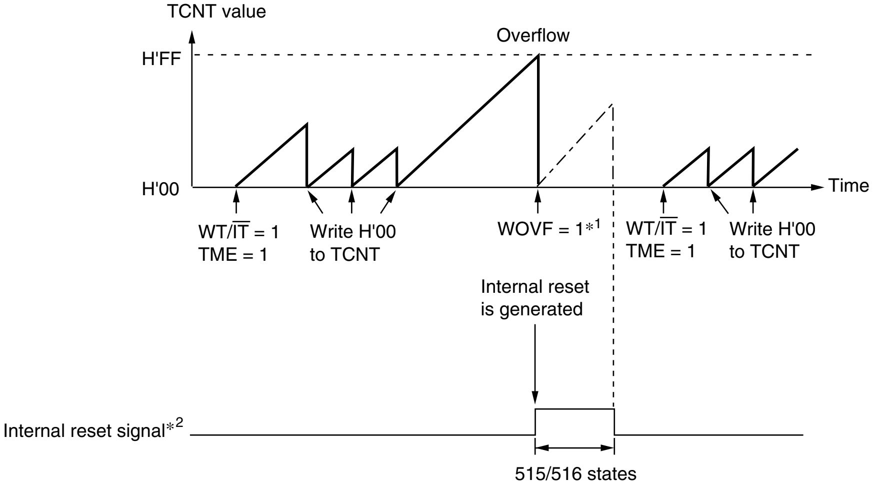
Legend

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 9.3 (a) WDT_0 Operation in Watchdog Timer Mode



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 9.3 (b) WDT_1 Operation in Watchdog Timer Mode

9.3.2 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.

9.4 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 9.1 WDT Interrupt Sources

Name	Interrupt Source	Interrupt Flag
WOVI	TCNT overflow (interval timer mode)	OVF
NMI	TCNT overflow (watchdog timer mode)	OVF

9.5 Usage Notes

9.5.1 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 9.3 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 9.4. 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 9.4. 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.

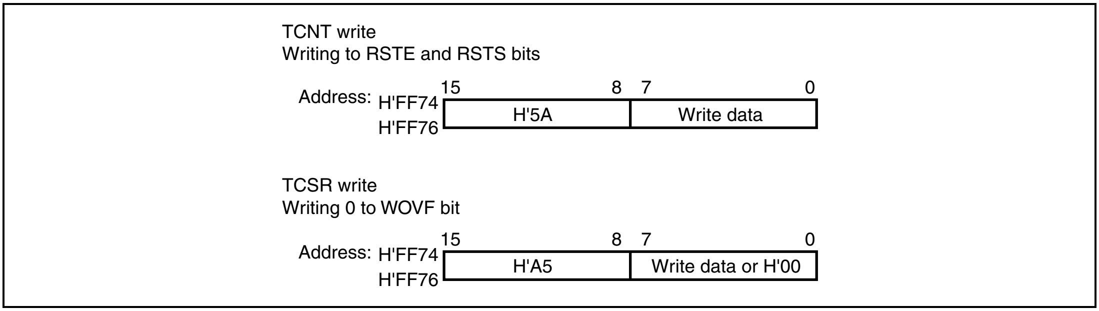


Figure 9.4   Writing to TCNT, TCSR, and RSTCSR (example for WDT0)

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.

9.5.2 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 9.5 shows this operation.

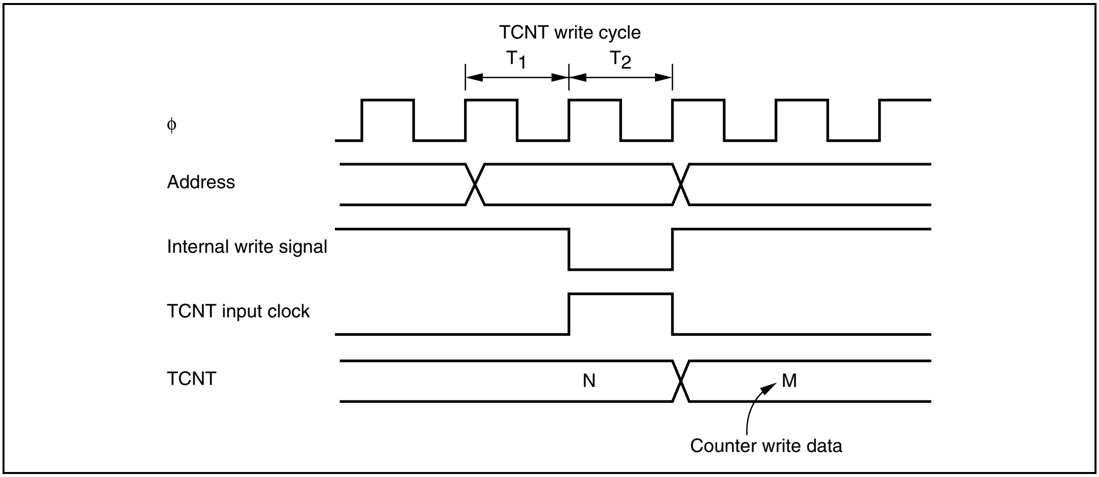


Figure 9.5   Contention between TCNT Write and Increment

9.5.3 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.

9.5.4 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.

9.5.5 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.

9.5.6 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 10 Serial Communication Interface (SCI)

This LSI has three 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 10.1 shows a block diagram of the SCI.

10.1 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.
- 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

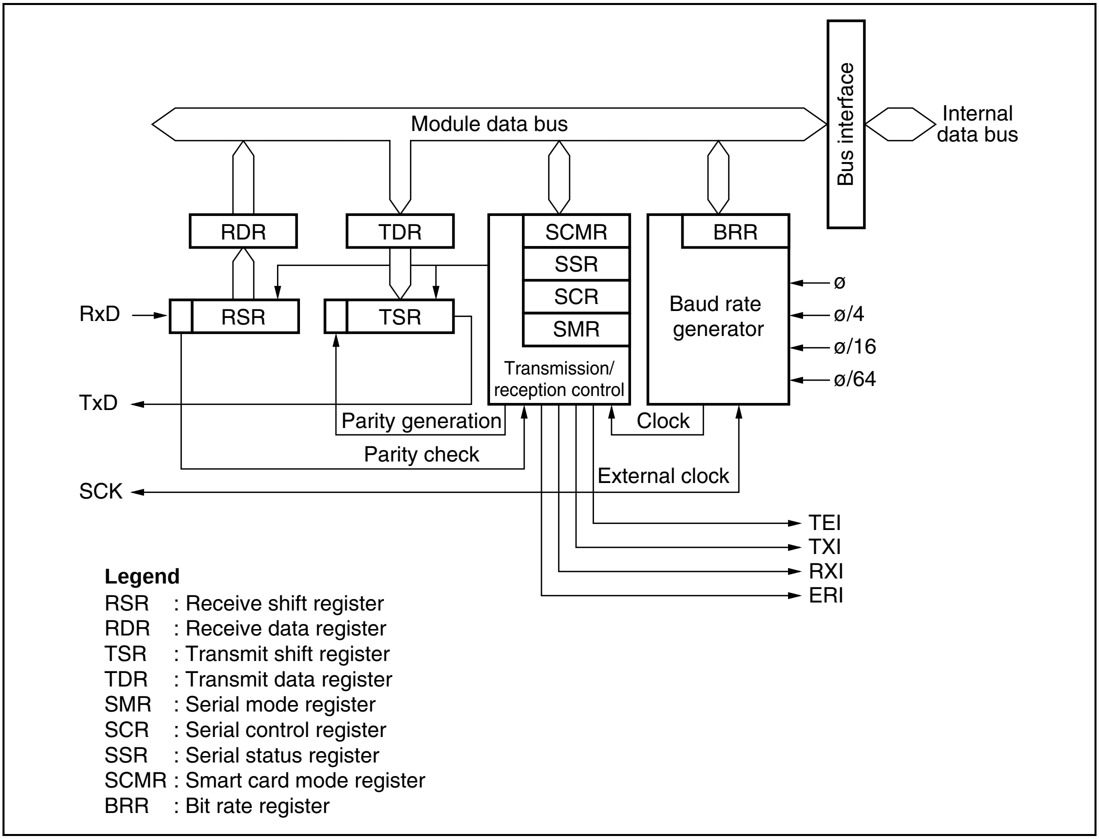

Module data bus
Bus interface
Internal data bus
RDR
TDR
SCMR
SSR
SCR
SMR
BRR
Baud rate generator
ø
ø/4
ø/16
ø/64
RxD
RSR
TSR
TxD
Transmission/ reception control
Parity generation
Parity check
Clock
External clock
SCK
TEI
TXI
RXI
ERI
Legend
RSR : Receive shift register
RDR : Receive data register
TSR : Transmit shift register
TDR : Transmit data register
SMR : Serial mode register
SCR : Serial control register
SSR : Serial status register
SCMR : Smart card mode register
BRR : Bit rate register

Figure 10.1 Block Diagram of SCI

10.2 Input/Output Pins

Table 10.1 shows the serial pins for each SCI channel.

Table 10.1 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.

10.3 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)

10.3.1 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.

10.3.2 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.

10.3.3 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.

10.3.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first transfers transmit data from TDR to TSR, then sends the data to the TxD pin. TSR cannot be directly accessed by the CPU.

10.3.5 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/$\overline{\text{A}}$	0	R/W	Communication Mode 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous mode) 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first is fixed and the MSB of TDR is not transmitted in transmission. In clocked synchronous mode, a fixed data length of 8 bits is used.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. For a multiprocessor format, parity bit addition and checking are not performed regardless of the PE bit setting.
4	O/$\overline{\text{E}}$	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.

Bit	Bit Name	Initial Value	R/W	Description
2	MP	0	R/W	Multiprocessor Mode (enabled only in asynchronous mode) When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and O/$\overline{E}$ bit settings are invalid in multiprocessor mode.
1	CKS1	0	R/W	Clock Select 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 10.3.9, Bit Rate Register (BRR). n is the decimal representation of the value of n in BRR (see section 10.3.9, 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, refer to section 10.7.8, 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, refer to section 10.7.3, 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.

Bit	Bit Name	Initial Value	R/W	Description
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, refer to section 10.7.2, 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, refer to section 10.7.4, Receive Data Sampling Timing and Reception Margin in Smart Card Interface Mode. S stands for the value of S in BRR (see section 10.3.9, Bit Rate Register (BRR)).
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 10.3.9, Bit Rate Register (BRR). n is the decimal representation of the value of n in BRR (see section 10.3.9, Bit Rate Register (BRR)).

10.3.6 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, refer to section 10.8, 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.
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, refer to section 10.5, Multiprocessor Communication Function.
2	TEIE	0	R/W	Transmit End Interrupt Enable This bit is set to 1, TEI interrupt request is enabled.

Bit	Bit Name	Initial Value	R/W	Description
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 clock SCK pin functions as I/O port 01: Internal clock Outputs a clock of the same frequency as the bit rate from the SCK pin. 1X: External clock Inputs a clock with a frequency 16 times the bit rate from the SCK pin. Clocked synchronous mode 0X: Internal clock (SCK pin functions as clock output) 1X: External clock (SCK pin functions as clock input)

Legend

X: Don't care

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.

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	Clock Enable 0 and 1
0	CKE0	0		Enables or disables clock output from the SCK pin. The clock output can be dynamically switched in GSM mode. For details, refer to section 10.7.8, 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

10.3.7 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 condition] • When 0 is written to TDRE after reading TDRE = 1

Bit	Bit Name	Initial Value	R/W	Description
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 condition] • When 0 is written to RDRF after reading RDRF = 1 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	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

Bit	Bit Name	Initial Value	R/W	Description
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 condition] • When 0 is written to TDRE after reading TDRE = 1
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 condition] • When 0 is written to TDRE after reading TDRE = 1

Bit	Bit Name	Initial Value	R/W	Description
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 condition] • When 0 is written to RDRF after reading RDRF = 1 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 condition] • When 0 is written to ERS after reading ERS = 1
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

Bit	Bit Name	Initial Value	R/W	Description
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 ESR 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 condition] • When 0 is written to TDRE after reading TDRE = 1
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.

10.3.8 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

10.3.9 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 10.2 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 10.2 Relationships between N Setting in BRR and Bit Rate B

Mode	Bit Rate	Error
Asynchronous Mode	$B = \dfrac{\phi \times 10^6}{64 \times 2^{2n-1} \times (N+1)}$	$\text{Error (\%)} = \{ \dfrac{\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \} \times 100$
Clocked Synchronous Mode	$B = \dfrac{\phi \times 10^6}{8 \times 2^{2n-1} \times (N+1)}$	—————
Smart Card Interface Mode	$B = \dfrac{\phi \times 10^6}{S \times 2^{2n-1} \times (N+1)}$	$\text{Error (\%)} = \{ \dfrac{\phi \times 10^6}{B \times S \times 2^{2n-1} \times (N+1)} - 1 \} \times 100$

Note: B: Bit rate (bit/s)

N: BRR setting for baud rate generator (0 ≤ N ≤ 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 10.3 shows sample N settings in BRR in normal asynchronous mode. Table 10.4 shows the maximum bit rate for each frequency in normal asynchronous mode. Table 10.6 shows sample N settings in BRR in clocked synchronous mode. Table 10.8 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, refer to section 10.7.4, Receive Data Sampling Timing and Reception Margin. Tables 10.5 and 10.7 show the maximum bit rates with external clock input.

Table 10.3 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 10.3 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	0.13	2	223	0.00
300	2	90	0.13	2	95	0.00	2	103	0.13	2	111	0.00
600	1	181	0.13	1	191	0.00	1	207	0.13	1	223	0.00
1200	1	90	0.13	1	95	0.00	1	103	0.13	1	111	0.00
2400	0	181	0.13	0	191	0.00	0	207	0.13	0	223	0.00
4800	0	90	0.13	0	95	0.00	0	103	0.13	0	111	0.00
9600	0	45	–0.93	0	47	0.00	0	51	0.13	0	55	0.00
19200	0	22	–0.93	0	23	0.00	0	25	0.13	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	0.13	0	13	0.00

Table 10.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (3)

	Operating Frequency φ (MHz)											
	18			19.6608			20			24		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	79	–0.12	3	86	0.31	3	88	–0.25	3	106	–0.44
150	2	233	0.16	2	255	0.00	3	64	0.16	3	77	0.16
300	2	116	0.16	2	127	0.00	2	129	0.16	2	155	0.16
600	1	233	0.16	1	255	0.00	2	64	0.16	2	77	0.16
1200	1	116	0.16	1	127	0.00	1	129	0.16	1	155	0.16
2400	0	233	0.16	0	255	0.00	1	64	0.16	1	77	0.16
4800	0	116	0.16	0	127	0.00	0	129	0.16	0	155	0.16
9600	0	58	–0.69	0	63	0.00	0	64	0.16	0	77	0.16
19200	0	28	1.02	0	31	0.00	0	32	–1.36	0	38	0.16
31250	0	17	0.00	0	19	–1.70	0	19	0.00	0	23	0.00
38400	0	14	–2.34	0	15	0.00	0	15	1.73	0	19	–2.34

Table 10.4 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	12.288	384000	0	0
4.9152	153600	0	0	14	437500	0	0
5	156250	0	0	14.7456	460800	0	0
6	187500	0	0	16	500000	0	0
6.144	192000	0	0	17.2032	537600	0	0
7.3728	230400	0	0	18	562500	0	0
8	250000	0	0	19.6608	614400	0	0
9.8304	307200	0	0	20	625000	0	0
10	312500	0	0	24	750000	0	0
12	375000	0	0				

Table 10.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

φ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
4	1.0000	62500
4.9152	1.2288	76800
5	1.2500	78125
6	1.5000	93750
6.144	1.5360	96000
7.3728	1.8432	115200
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
19.6608	4.9152	307200
20	5.0000	312500
24	6.0000	375000

Table 10.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

Bit Rate (bit/s)	Operating Frequency ϕ (MHz)											
	4		8		10		16		20		24	
	n	N	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	2	149
5k	0	199	1	99	1	124	1	199	1	249	2	74
10k	0	99	0	199	0	249	1	99	1	124	1	149
25k	0	39	0	79	0	99	0	159	0	199	1	59
50k	0	19	0	39	0	49	0	79	0	99	1	29
100k	0	9	0	19	0	24	0	39	0	49	0	59
250k	0	3	0	7	0	9	0	15	0	19	0	23
500k	0	1	0	3	0	4	0	7	0	9	0	11
1M	0	0*	0	1			0	3	0	4	0	5
2.5M					0	0*			0	1	—	—
5M									0	0*	—	—

Legend

Blank	: Cannot be set.
—	: Can be set, but there will be a degree of error.
*	: Continuous transfer is not possible.

Table 10.7 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	1000000.0	16	2.6667	2666666.7
8	1.3333	1333333.3	18	3.0000	3000000.0
10	1.6667	1666666.7	20	3.3333	3333333.3
12	2.0000	2000000.0	24	4.0000	4000000.0

Table 10.8 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			24.00		
Bit Rate (bit/s)	n	N	Error (%)	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	0	2	12.01

Table 10.9 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	16.00	21505	0	0
10.00	13441	0	0	18.00	24194	0	0
10.7136	14400	0	0	20.00	26882	0	0
13.00	17473	0	0	24.00	32258	0	0
14.2848	19200	0	0				

10.4 Operation in Asynchronous Mode

Figure 10.2 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.

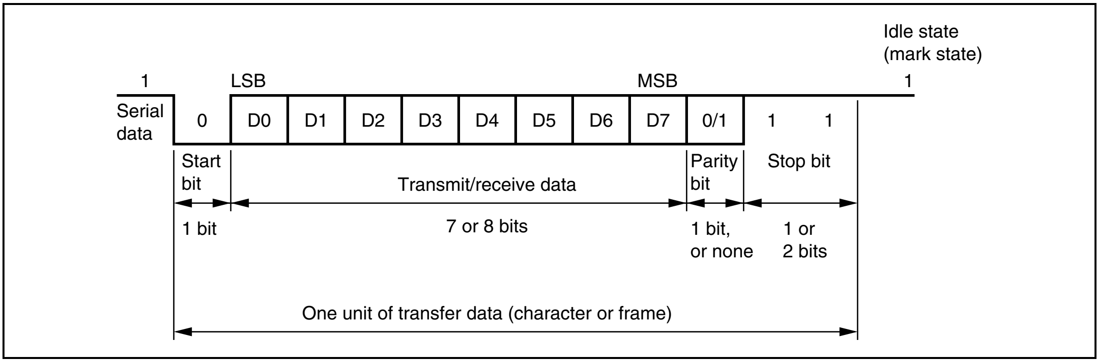


Figure 10.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

10.4.1 Data Transfer Format

Table 10.10 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, refer to section 10.5, Multiprocessor Communication Function.

Table 10.10 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

10.4.2 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 10.3. Thus, the reception margin in asynchronous mode is given by formula (1) below.

$$M = \left\{ \left(0.5 - \frac{1}{2N}\right) - \frac{D - 0.5}{N} - (L - 0.5) F \right\} \times 100\ [\%]$$

... Formula (1)

Where M : Reception margin

N : Ratio of bit rate to clock (N = 16)

D : Clock duty (D = 0.5 to 1.0)

L : Frame length (L = 9 to 12)

F : Absolute value of clock rate deviation

Assuming values of F (absolute value of clock rate deviation) = 0 and D (clock duty) = 0.5 in formula (1), the reception margin can be given by the formula.

$M = \{0.5 - 1/(2 \times 16)\} \times 100\ [\%] = 46.875\%$

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

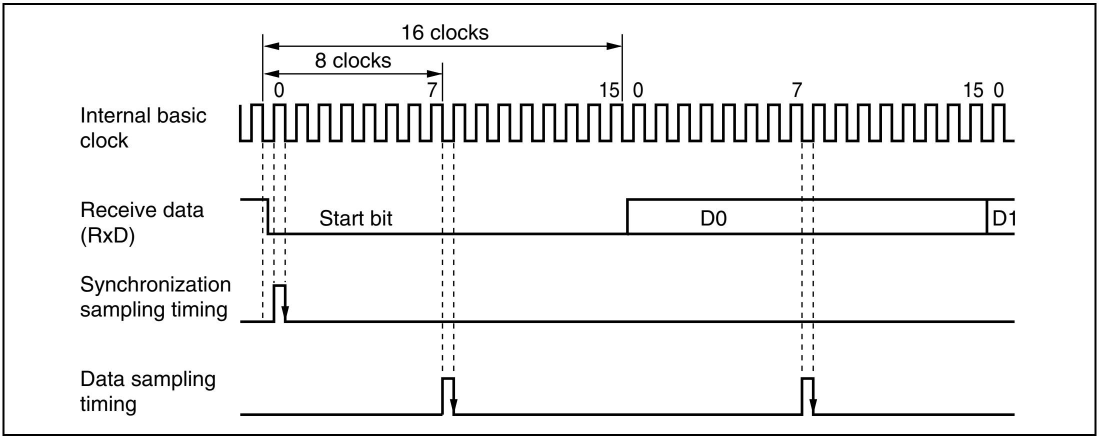


Figure 10.3 Receive Data Sampling Timing in Asynchronous Mode

10.4.3 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/$\overline{\text{A}}$ 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 10.4.

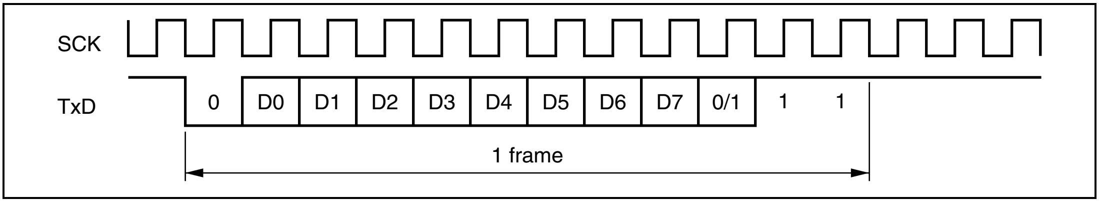
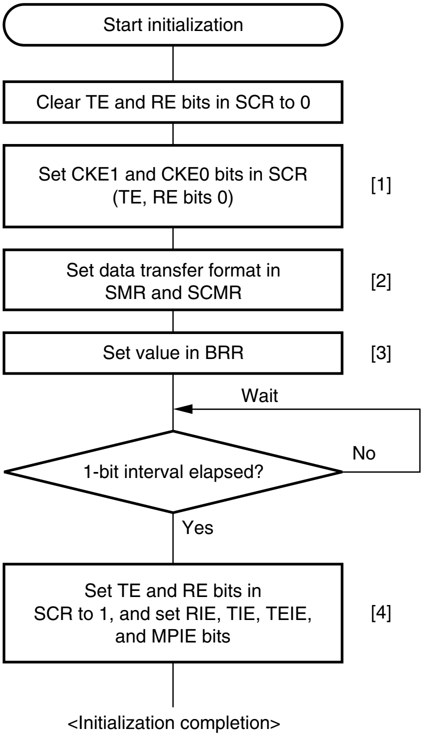


Figure 10.4 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode)

10.4.4 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as 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.


Start initialization
Clear TE and RE bits in SCR to 0
Set CKE1 and CKE0 bits in SCR (TE, RE bits 0)
[1]
Set data transfer format in SMR and SCMR
[2]
Set value in BRR
[3]
Wait
No
1-bit interval elapsed?
Yes
Set TE and RE bits in SCR to 1, and set RIE, TIE, TEIE, and MPIE bits
[4]
<Initialization completion>

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

When the clock is selected in asynchronous mode, it is output immediately after SCR settings are made.

[2] Set the data transfer format in SMR and SCMR.

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

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.

Setting the TE and RE bits enables the TxD and RxD pins to be used.

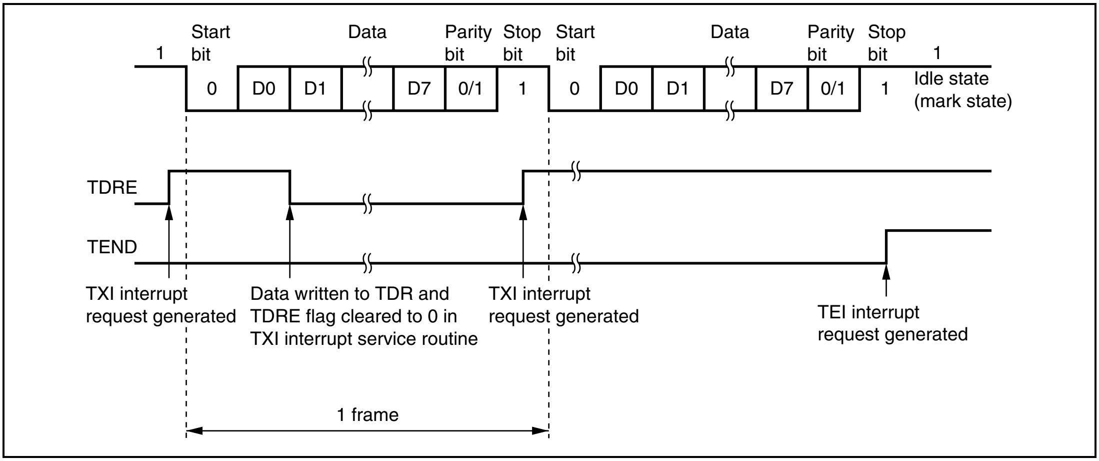
Figure 10.5 Sample SCI Initialization Flowchart

10.4.5 Data Transmission (Asynchronous Mode)

Figure 10.6 shows an example of operation for transmission in asynchronous mode. In transmission, the SCI operates as described below.

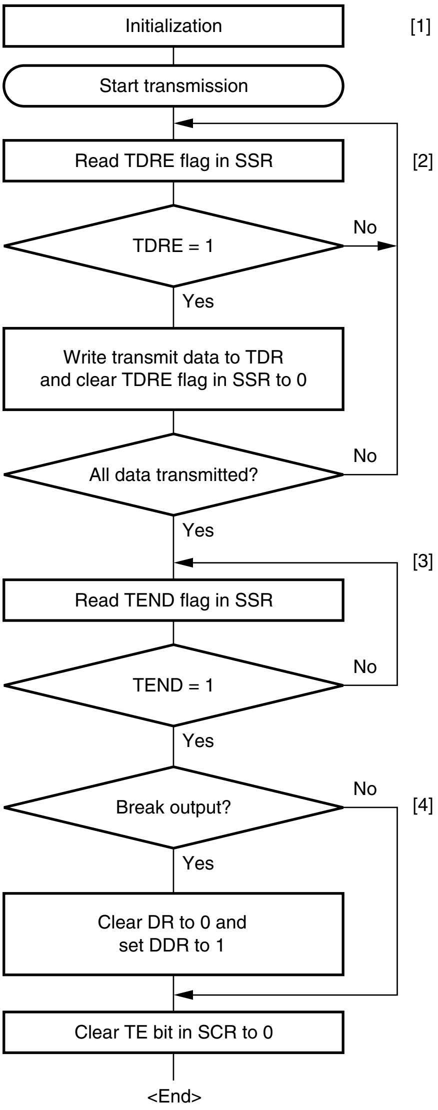
1. The SCI monitors the TDRE flag in SSR. If the flag is cleared to 0, the SCI recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit is set to 1 at this 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 10.7 shows a sample flowchart for transmission in asynchronous mode.



1
Start bit
Data
Parity bit
Stop bit
Start bit
Data
Parity bit
Stop bit
1
0
D0
D1
D7
0/1
1
0
D0
D1
D7
0/1
1
Idle state (mark state)
TDRE
TEND
TXI interrupt request generated
Data written to TDR and TDRE flag cleared to 0 in TXI interrupt service routine
TXI interrupt request generated
TEI interrupt request generated
1 frame

Figure 10.6 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.

[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 10.7 Sample Serial Transmission Flowchart

10.4.6 Serial Data Reception (Asynchronous Mode)

Figure 10.8 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.

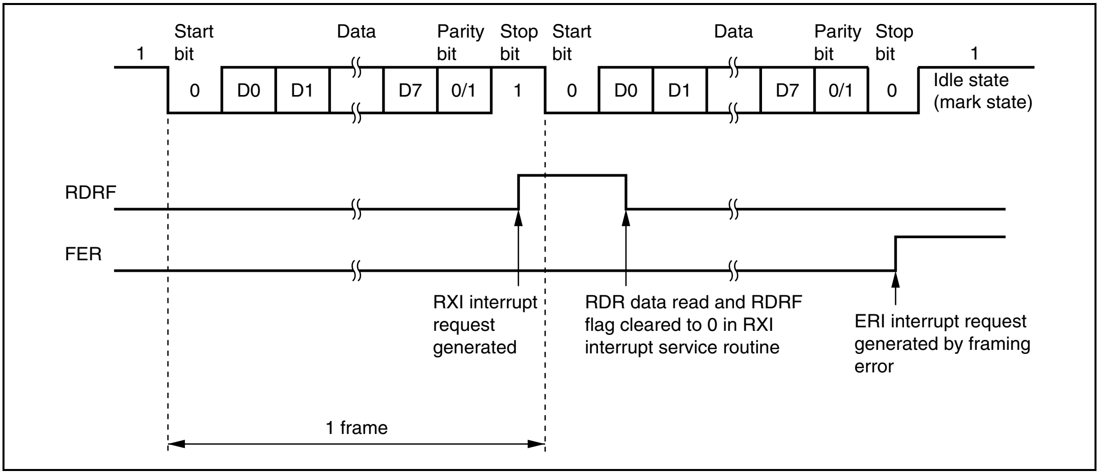

1
Start bit
0
D0
D1
Data
D7
Parity bit
0/1
Stop bit
1
Start bit
0
D0
D1
Data
D7
Parity bit
0/1
Stop bit
0
1
Idle state (mark state)
RDRF
FER
RXI interrupt request generated
RDR data read and RDRF flag cleared to 0 in RXI interrupt service routine
ERI interrupt request generated by framing error
1 frame

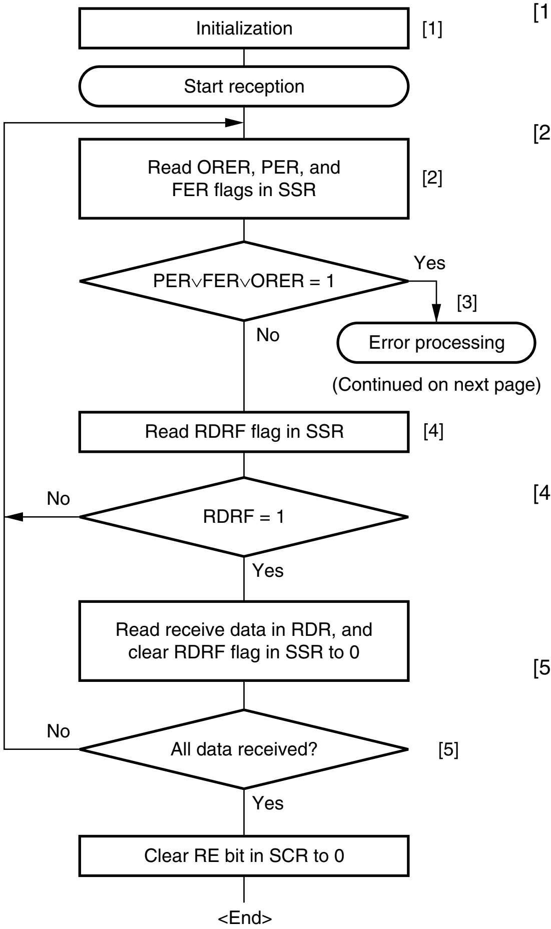
Figure 10.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

Table 10.11 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 10.9 shows a sample flow chart for serial data reception.

Table 10.11 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.

Figure 10.9 Sample Serial Reception Data Flowchart (1)

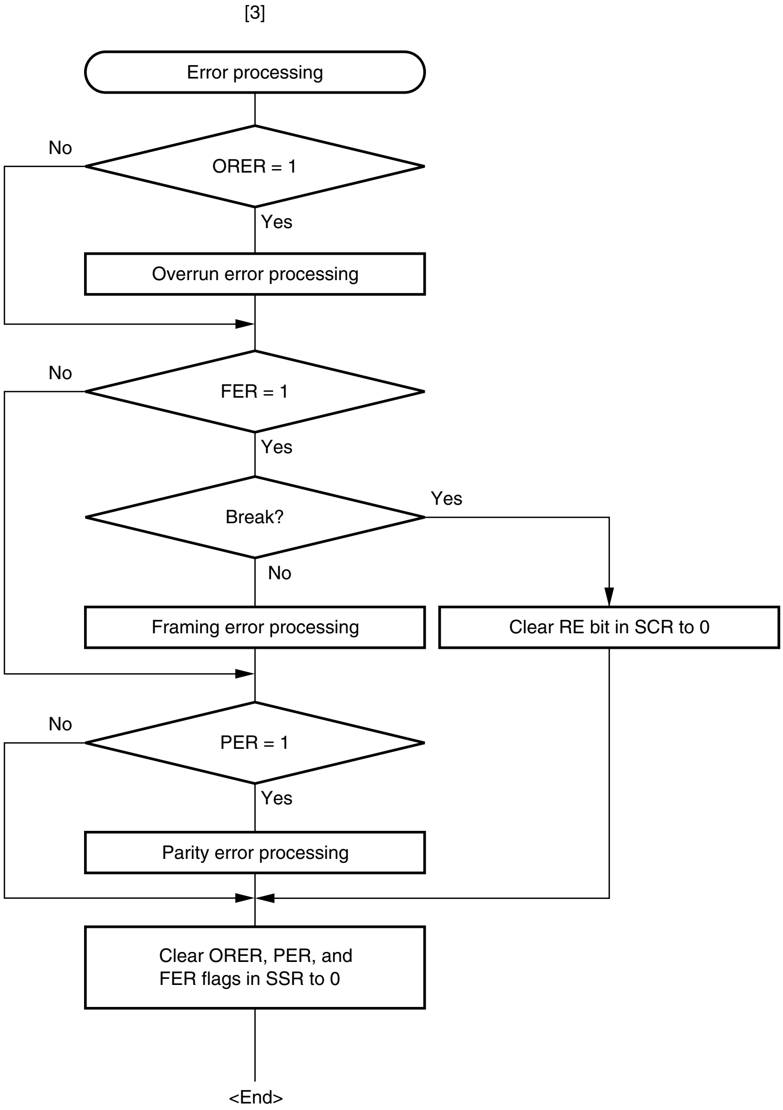


Figure 10.9 Sample Serial Reception Data Flowchart (2)

10.5 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 10.10 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.

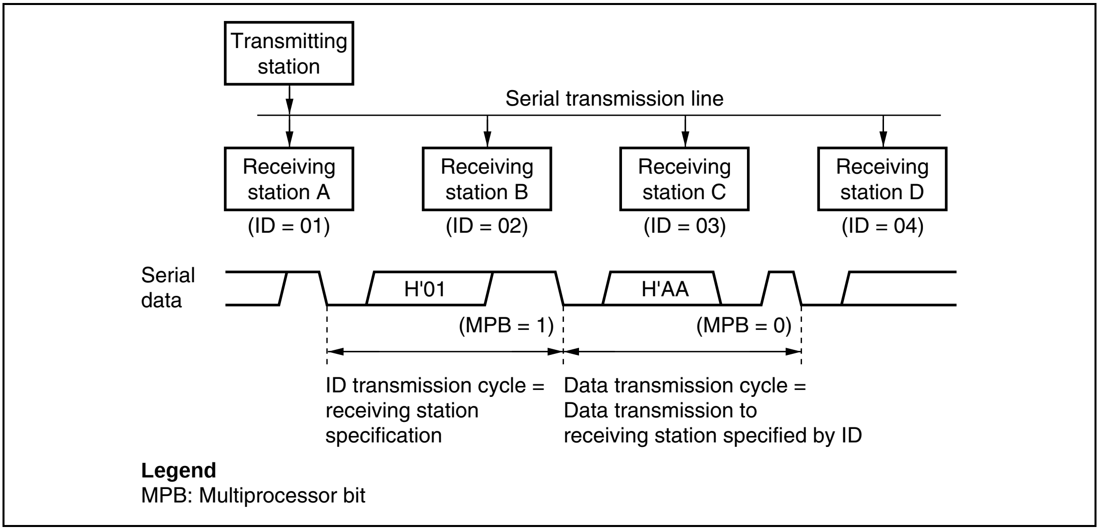
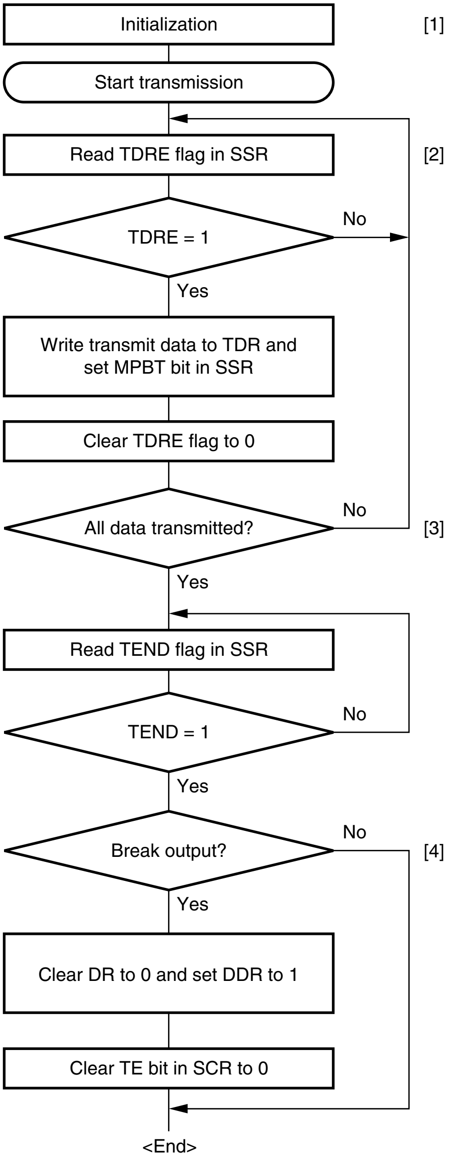


Figure 10.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

10.5.1 Multiprocessor Serial Data Transmission

Figure 10.11 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:

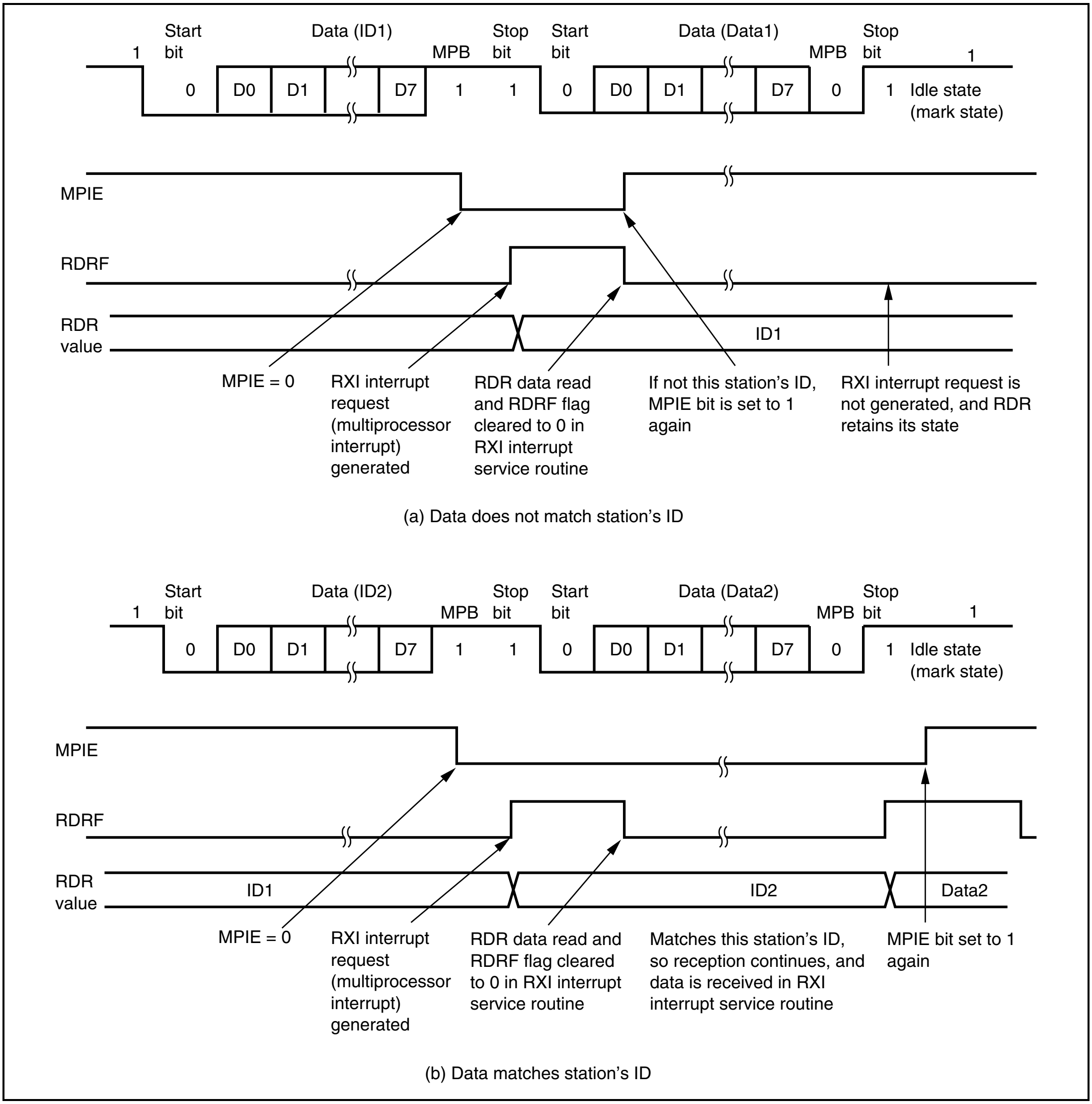
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.

[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 10.11 Sample Multiprocessor Serial Transmission Flowchart

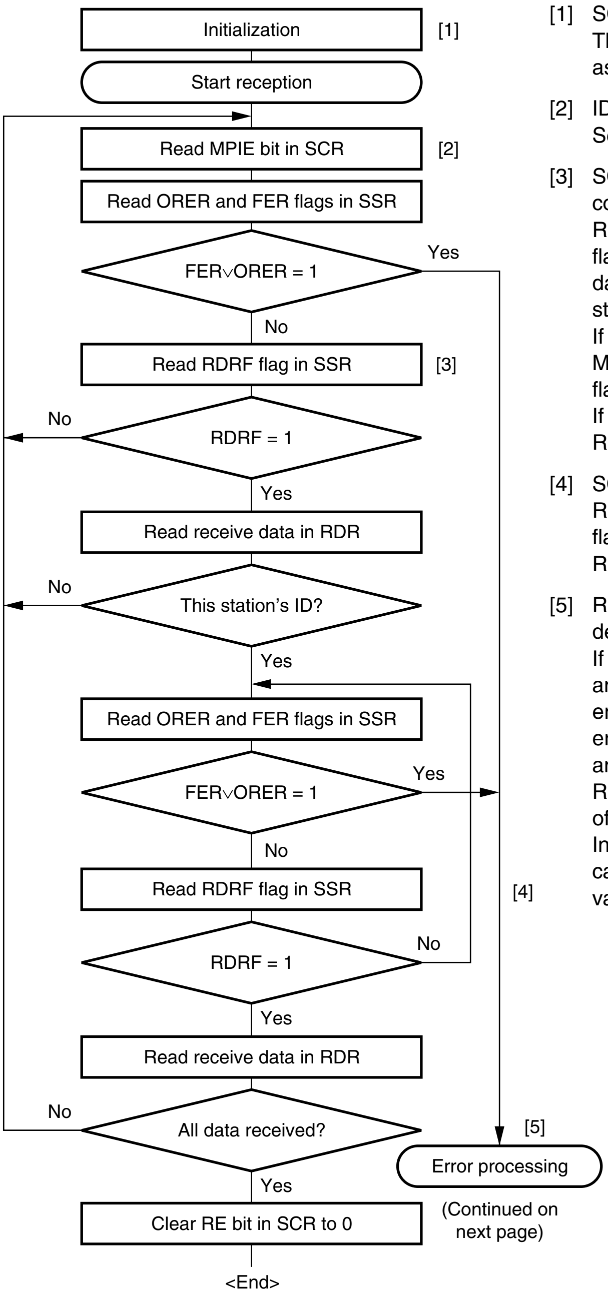
10.5.2 Multiprocessor Serial Data Reception

Figure 10.13 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 10.12 shows an example of SCI operation for multiprocessor format reception.



Start bit
Data (ID1)
MPB
Stop bit
Start bit
Data (Data1)
MPB
Stop bit
1
0
D0
D1
D7
1
1
0
D0
D1
D7
0
1
Idle state (mark state)
MPIE
RDRF
RDR value
ID1
MPIE = 0
RXI interrupt request (multiprocessor interrupt) generated
RDR data read and RDRF flag cleared to 0 in RXI interrupt service routine
If not this station's ID, MPIE bit is set to 1 again
RXI interrupt request is not generated, and RDR retains its state
(a) Data does not match station's ID
Start bit
Data (ID2)
MPB
Stop bit
Start bit
Data (Data2)
MPB
Stop bit
1
0
D0
D1
D7
1
1
0
D0
D1
D7
0
1
Idle state (mark state)
MPIE
RDRF
RDR value
ID1
ID2
Data2
MPIE = 0
RXI interrupt request (multiprocessor interrupt) generated
RDR data read and RDRF flag cleared to 0 in RXI interrupt service routine
Matches this station's ID, so reception continues, and data is received in RXI interrupt service routine
MPIE bit set to 1 again
(b) Data matches station's ID

Figure 10.12 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 10.13 Sample Multiprocessor Serial Reception Flowchart (1)

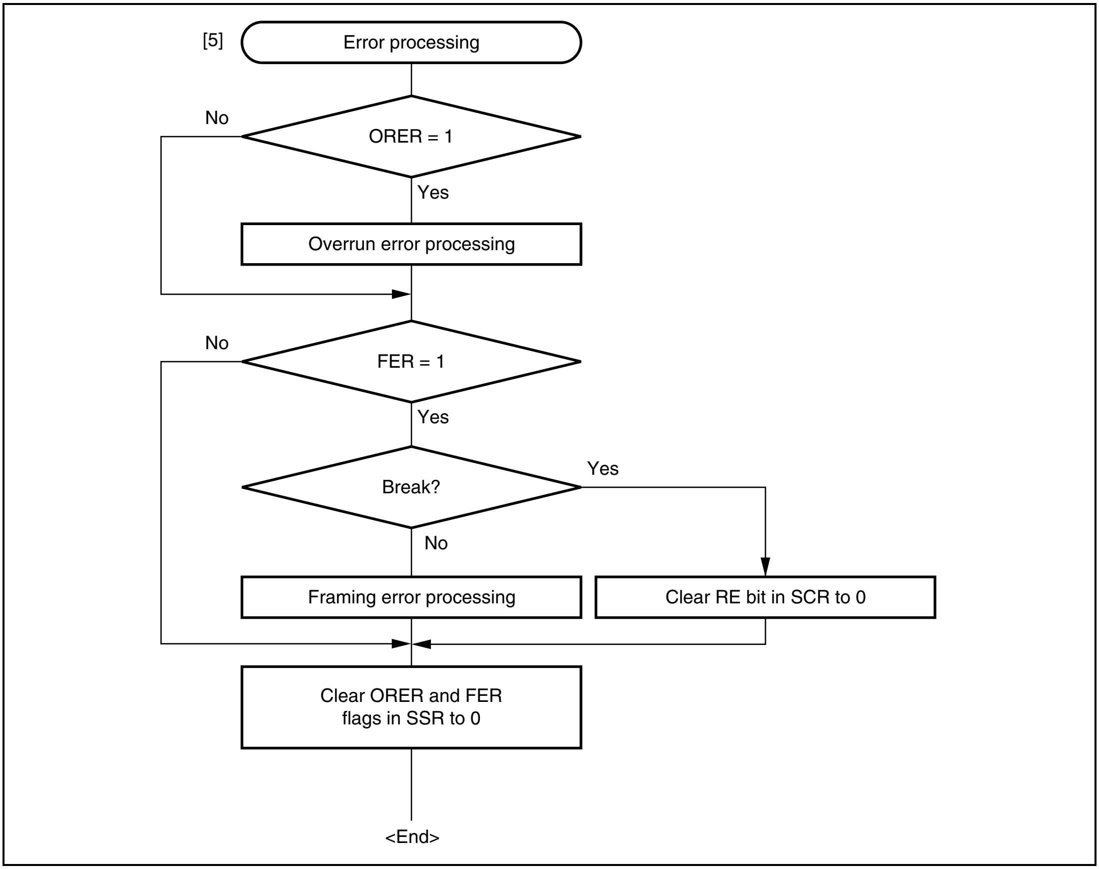
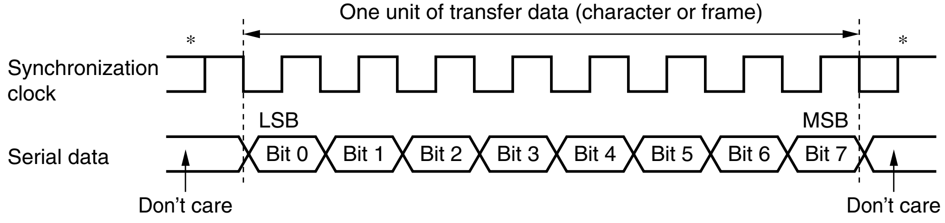


Figure 10.13 Sample Multiprocessor Serial Reception Flowchart (2)

10.6 Operation in Clocked Synchronous Mode

Figure 10.14 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.



Note: * High except in continuous transfer

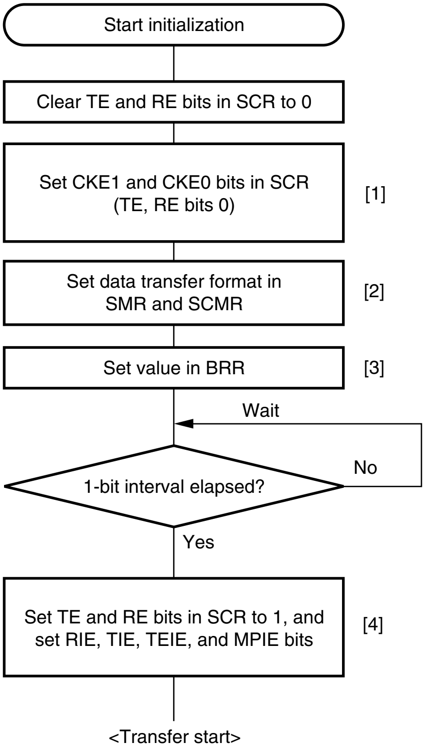
Figure 10.14 Data Format in Synchronous Communication (For LSB-First)

10.6.1 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.

10.6.2 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 10.15. 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.


Start initialization
Clear TE and RE bits in SCR to 0
Set CKE1 and CKE0 bits in SCR (TE, RE bits 0)
[1]
Set data transfer format in SMR and SCMR
[2]
Set value in BRR
[3]
Wait
No
1-bit interval elapsed?
Yes
Set TE and RE bits in SCR to 1, and set RIE, TIE, TEIE, and MPIE bits
[4]
<Transfer start>

[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 10.15 Sample SCI Initialization Flowchart

10.6.3 Serial Data Transmission (Clocked Synchronous Mode)

Figure 10.16 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 10.17 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.

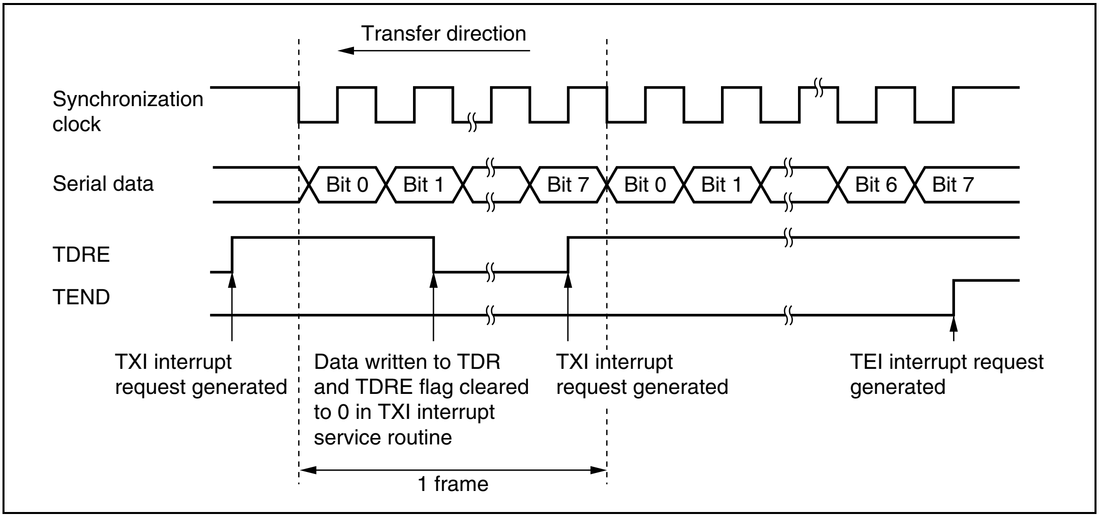
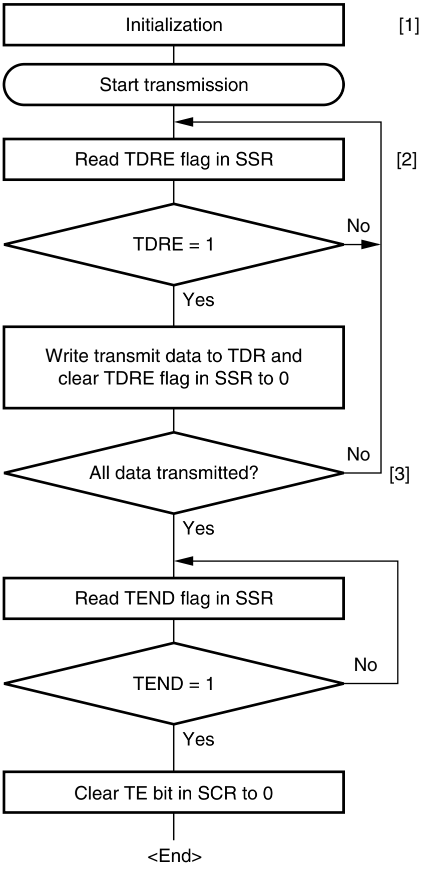

Transfer direction
Synchronization clock
Serial data
Bit 0
Bit 1
Bit 7
Bit 0
Bit 1
Bit 6
Bit 7
TDRE
TEND
TXI interrupt request generated
Data written to TDR and TDRE flag cleared to 0 in TXI interrupt service routine
TXI interrupt request generated
TEI interrupt request generated
1 frame

Figure 10.16 Sample SCI Transmission Operation in Clocked Synchronous Mode



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.

Figure 10.17 Sample Serial Transmission Flowchart

10.6.4 Serial Data Reception (Clocked Synchronous Mode)

Figure 10.18 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.

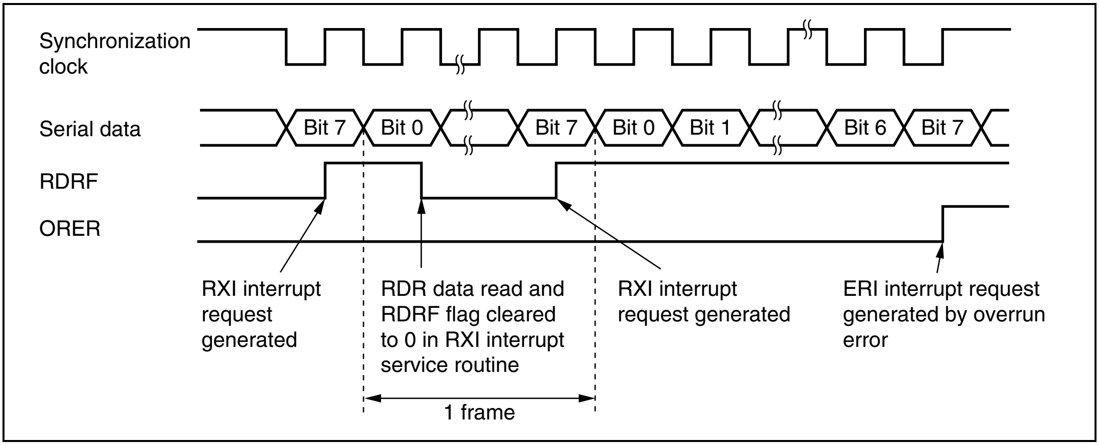
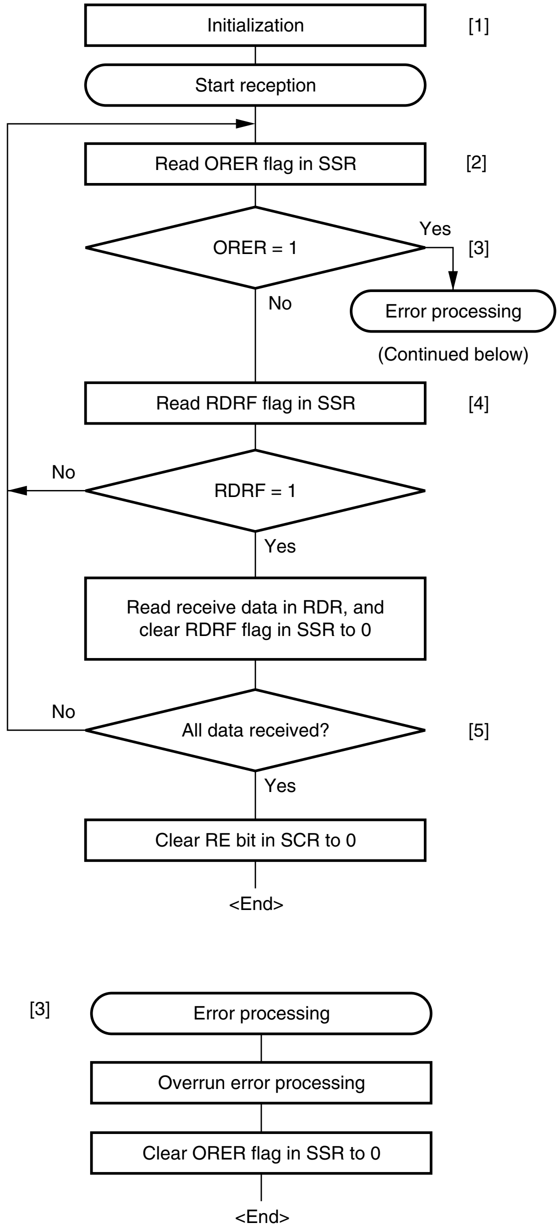

Synchronization clock
Serial data
Bit 7
Bit 0
Bit 7
Bit 0
Bit 1
Bit 6
Bit 7
RDRF
ORER
RXI interrupt request generated
RDR data read and RDRF flag cleared to 0 in RXI interrupt service routine
RXI interrupt request generated
ERI interrupt request generated by overrun error
1 frame

Figure 10.18 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 10.19 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.

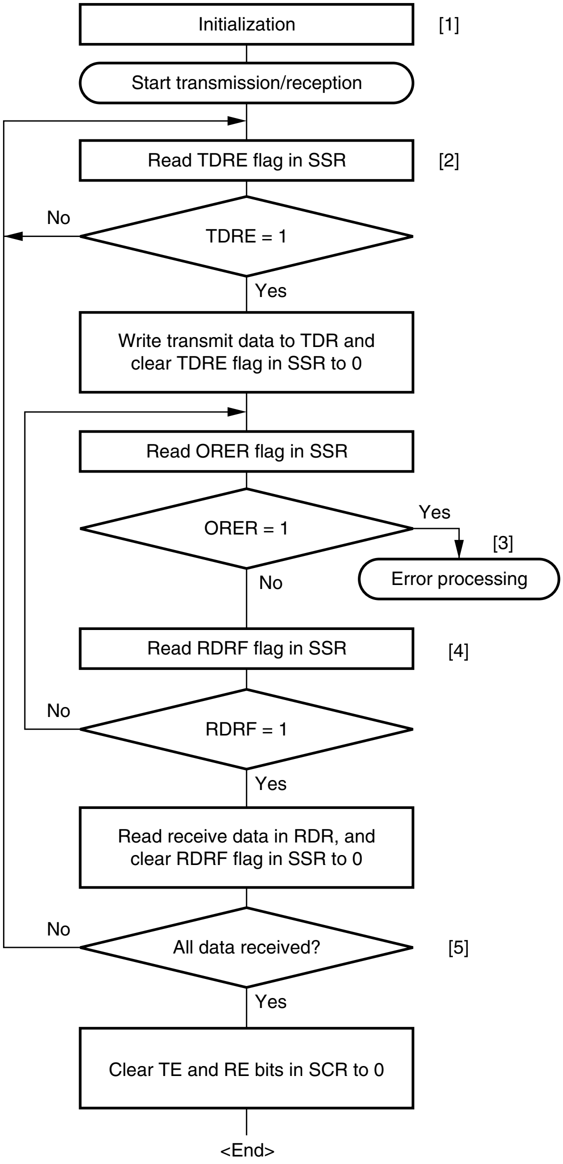
[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0.
Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[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.

Figure 10.19 Sample Serial Reception Flowchart

10.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode)

Figure 10.20 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. 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.


Initialization
[1]
Start transmission/reception
Read TDRE flag in SSR
[2]
No
TDRE = 1
Yes
Write transmit data to TDR and clear TDRE flag in SSR to 0
Read ORER flag in SSR
Yes
ORER = 1
[3]
No
Error processing
Read RDRF flag in SSR
[4]
No
RDRF = 1
Yes
Read receive data in RDR, and clear RDRF flag in SSR to 0
No
All data received?
[5]
Yes
Clear TE and RE bits in SCR to 0
<End>

[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.

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.

Figure 10.20 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations

10.7 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.

10.7.1 Pin Connection Example

Figure 10.21 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.

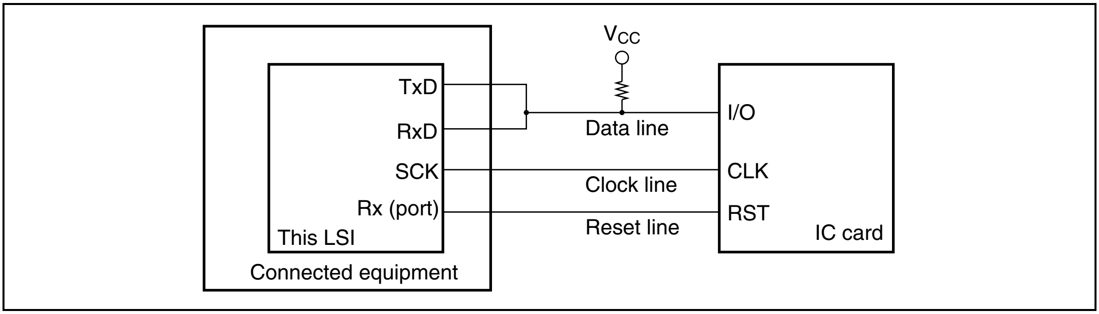


Figure 10.21 Schematic Diagram of Smart Card Interface Pin Connections

10.7.2 Data Format (Except for Block Transfer Mode)

Figure 10.22 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.

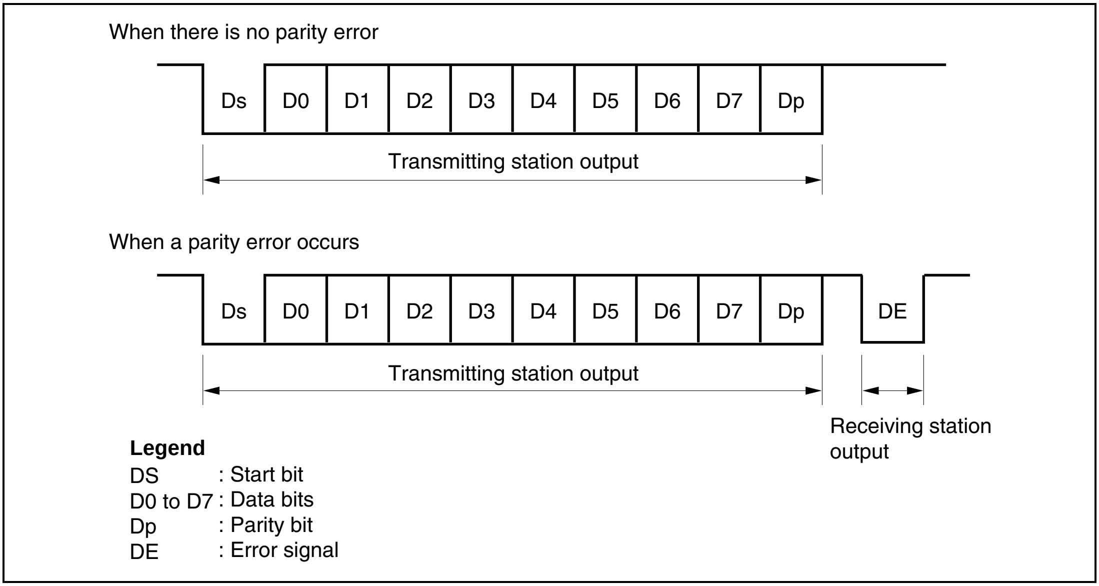


Figure 10.22 Normal Smart Card Interface Data Format

Data transfer with other types of IC cards (direct convention and inverse convention) are performed as described in the following.

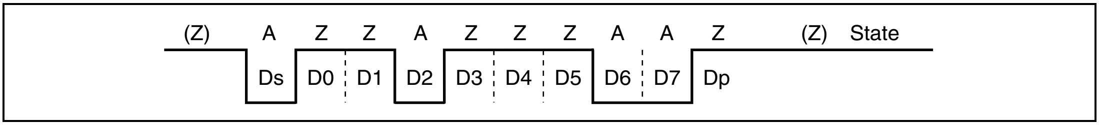


Figure 10.23 Direct Convention (SDIR = SINV = O/$\overline{\text{E}}$ = 0)

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/$\overline{\text{E}}$ bit in SMR to 0 to select even parity mode.

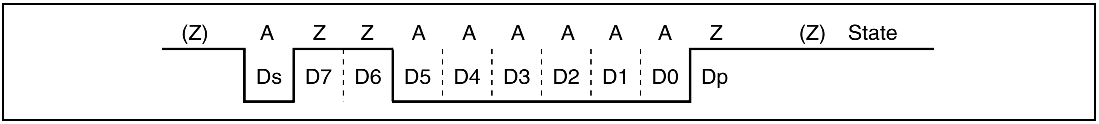


Figure 10.24 Inverse Convention (SDIR = SINV = O/$\overline{\text{E}}$ = 1)

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/$\overline{\text{E}}$ bit in SMR to 1 to invert the parity bit for both transmission and reception.

10.7.3 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.

10.7.4 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 10.25, 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 (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.

$$\begin{aligned} M &= (0.5 - 1/2 \times 372) \times 100\% \\ &= 49.866\% \end{aligned}$$

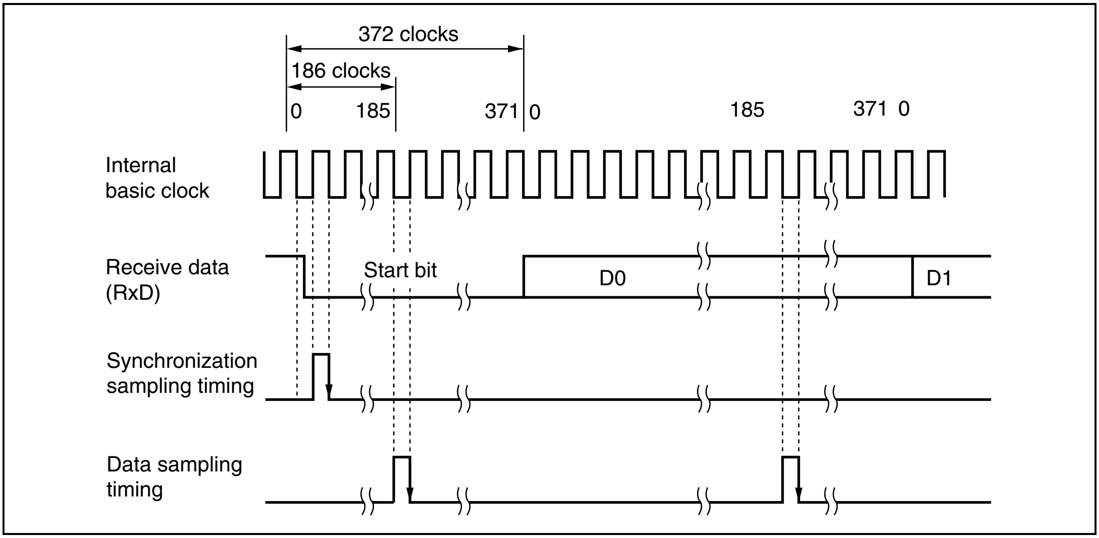


Figure 10.25 Receive Data Sampling Timing in Smart Card Interface Mode (Using Clock of 372 Times the Transfer Rate)

10.7.5 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/$\overline{\text{E}}$, BCP0, BCP1, CKS0, 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.

10.7.6 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 10.26 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 10.28 shows a flowchart for transmission. In the event of an error in transmission, the SCI retransmits the same data automatically. During this period, the TEND flag remains cleared to 0. Therefore, the SCI 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.

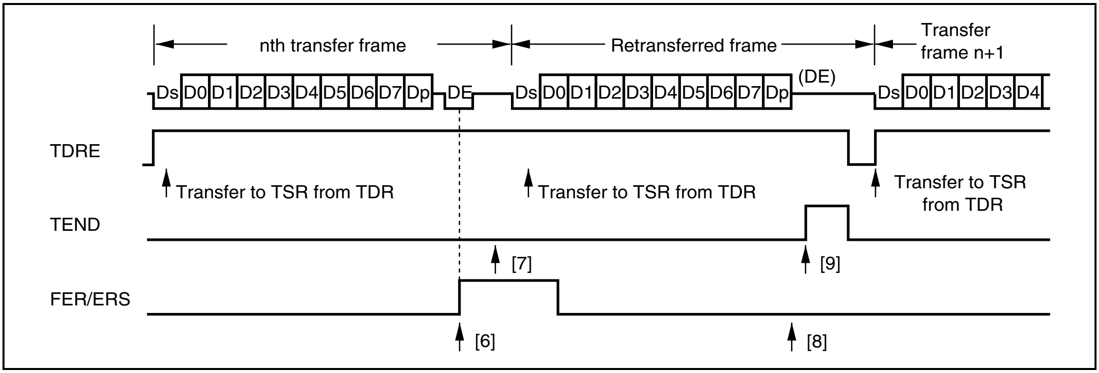


Figure 10.26 Retransfer Operation in SCI Transmit Mode

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 10.27.

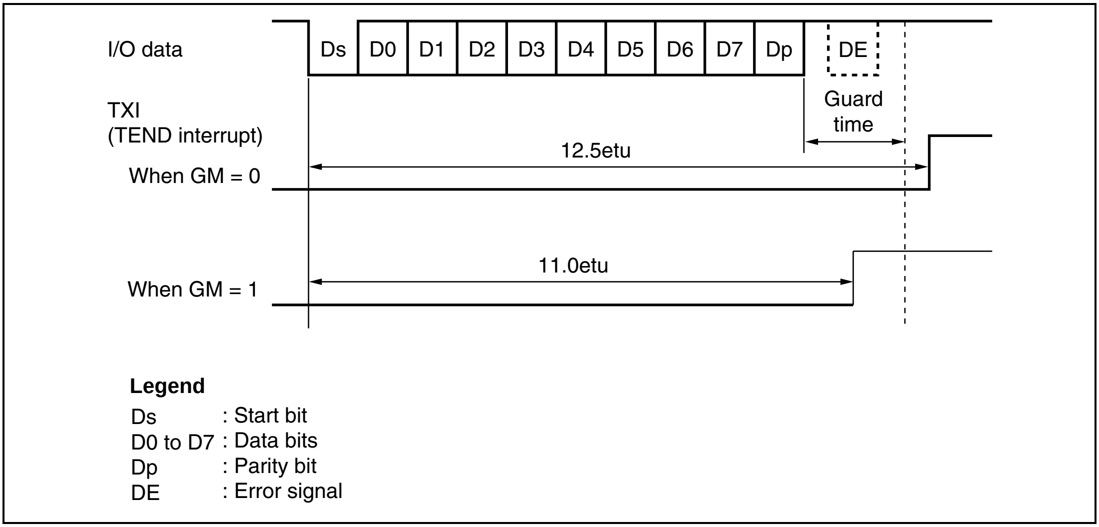


Figure 10.27 TEND Flag Generation Timing in Transmission Operation

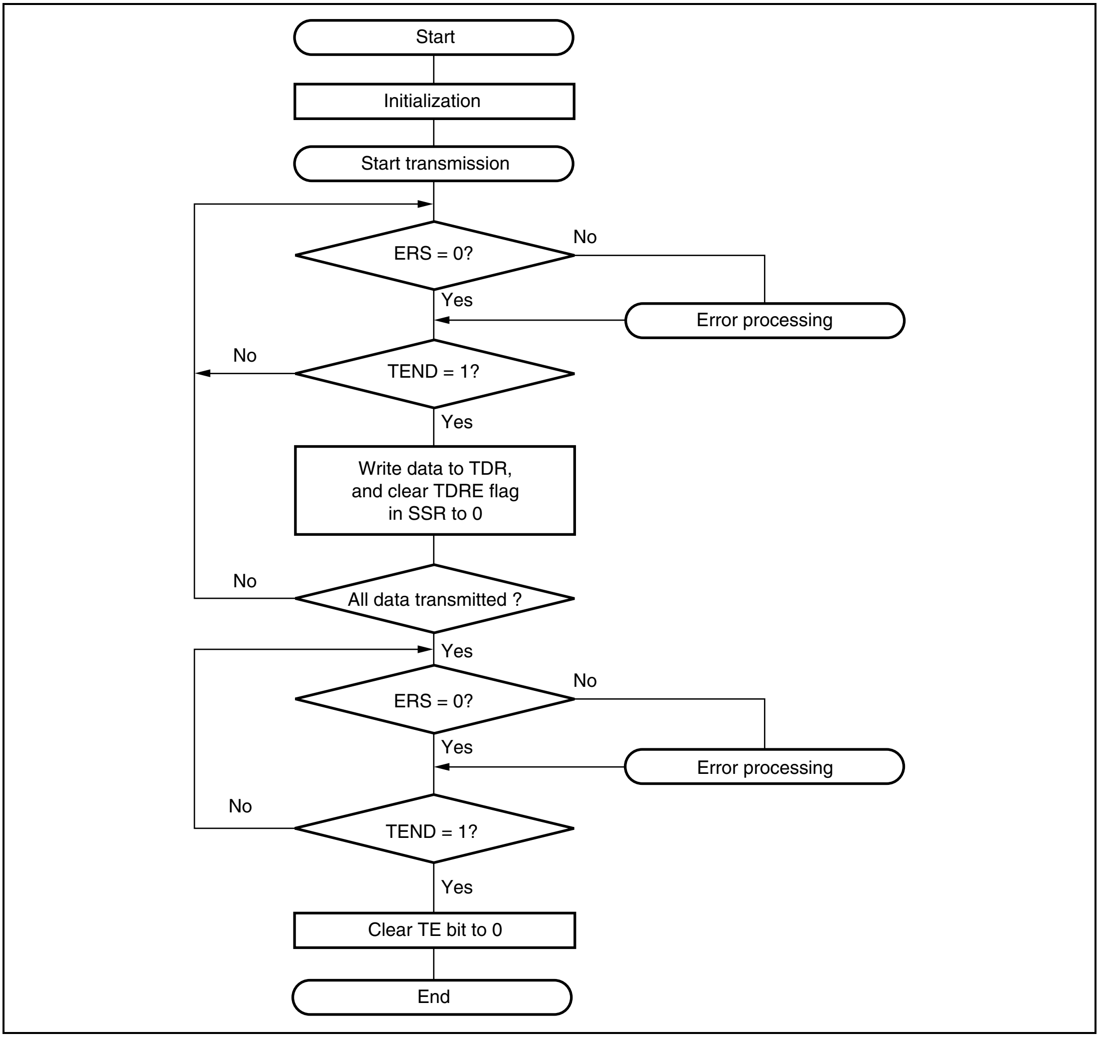


Figure 10.28 Example of Transmission Processing Flow

10.7.7 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 10.29 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 10.30 shows a flowchart for reception. 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. 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, refer to section 10.4, Operation in Asynchronous Mode.

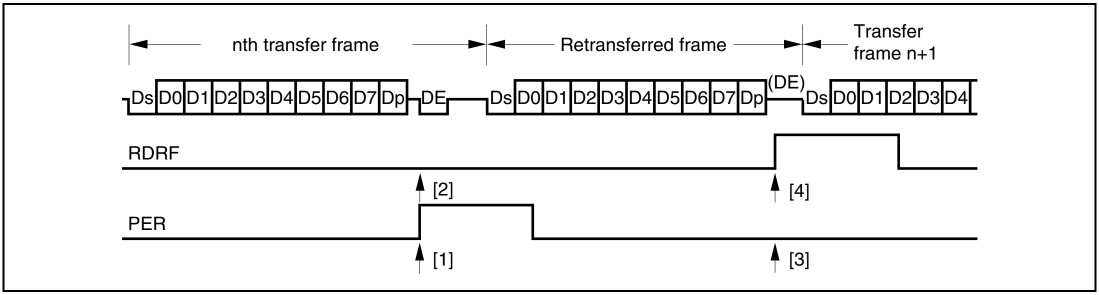


Figure 10.29 Retransfer Operation in SCI Receive Mode

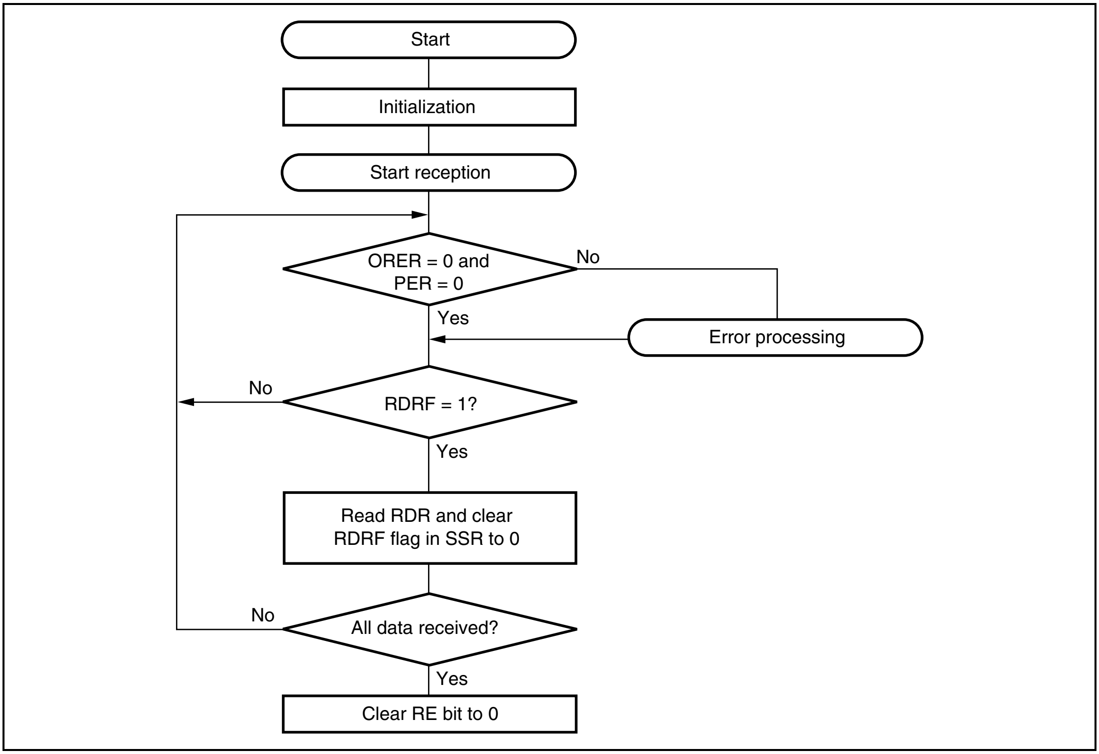


Figure 10.30 Example of Reception Processing Flow

10.7.8 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 10.31 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.

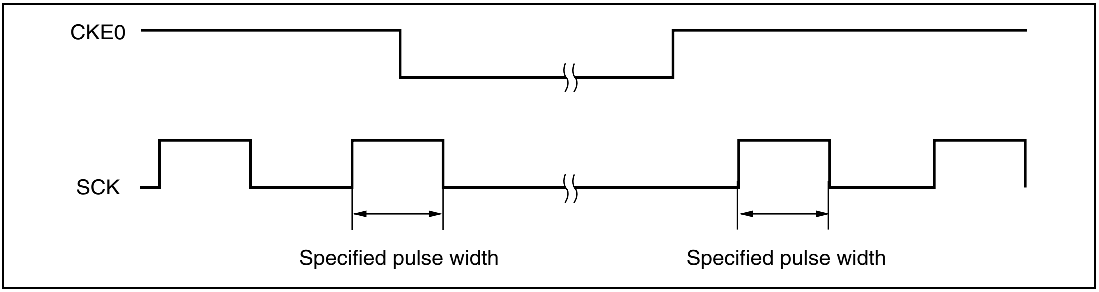


Figure 10.31 Timing for Fixing Clock Output Level

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.

Powering On: To secure clock duty 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 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.

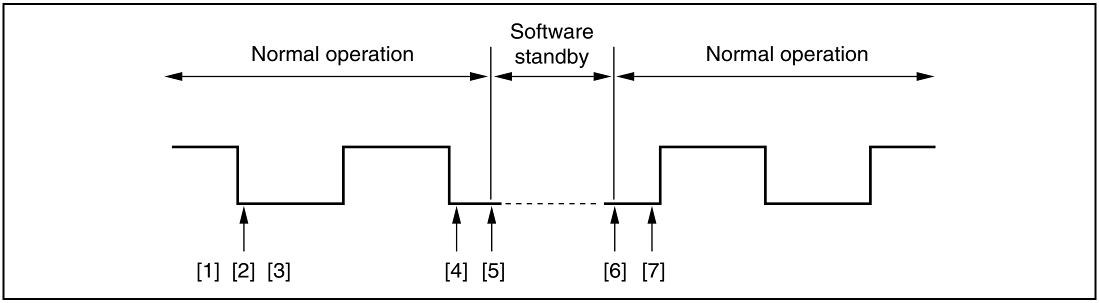


Figure 10.32 Clock Halt and Restart Procedure

10.8 Interrupt Sources

10.8.1 Interrupts in Normal Serial Communication Interface Mode

Table 10.12 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.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the ORER, PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated.

A TEI interrupt is 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 10.12 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag
0	ERI0	Receive Error	ORER, FER, PER
	RXI0	Receive Data Full	RDRF
	TXI0	Transmit Data Empty	TDRE
	TEI0	Transmission End	TEND
1	ERI1	Receive Error	ORER, FER, PER
	RXI1	Receive Data Full	RDRF
	TXI1	Transmit Data Empty	TDRE
	TEI1	Transmission End	TEND
2	ERI2	Receive Error	ORER, FER, PER
	RXI2	Receive Data Full	RDRF
	TXI2	Transmit Data Empty	TDRE
	TEI2	Transmission End	TEND

10.8.2 Interrupts in Smart Card Interface Mode

Table 10.13 shows the interrupt sources in Smart Card interface mode. The transmit end interrupt (TEI) request cannot be used in this mode.

Table 10.13 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag
0	ERI0	Receive Error, detection	ORER, PER, ERS
	RXI0	Receive Data Full	RDRF
	TXI0	Transmit Data Empty	TEND
1	ERI1	Receive Error, detection	ORER, PER, ERS
	RXI1	Receive Data Full	RDRF
	TXI1	Transmit Data Empty	TEND
2	ERI2	Receive Error, detection	ORER, PER, ERS
	RXI2	Receive Data Full	RDRF
	TXI2	Transmit Data Empty	TEND

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. In the event of an error, the SCI retransmits the same data automatically. During this period, the TEND flag remains cleared to 0. Therefore, the SCI 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.

In receive operations, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If an error occurs, an error flag is set but the RDRF flag is not. Consequently, an ERI interrupt request is sent to the CPU. Therefore, the error flag should be cleared.

10.9 Usage Notes

10.9.1 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, refer to section 16, Power-Down Modes.

10.9.2 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.

10.9.3 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 PCR and PDR 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 PCR to 1 and PDR to 0, and then clear TE to 0. When TE is cleared to 0, the transmitter is initialized regardless of the current transmission state, the TxD pin becomes an I/O port, and 0 is output from the TxD pin.

10.9.4 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.

Section 11 Hitachi Controller Area Network (HCAN)

The HCAN is a module for controlling a controller area network (CAN) for realtime communication in vehicular and industrial equipment systems, etc. For details on CAN specification, refer to Bosch CAN Specification Version 2.0 1991, Robert Bosch GmbH.

The block diagram of the HCAN is shown in figure 11.1.

11.1 Features

- CAN version: Bosch 2.0B active compatible
  - Communication systems: NRZ (Non-Return to Zero) system (with bit-stuffing function)
  - Broadcast communication system
  - Transmission path: Bidirectional 2-wire serial communication
  - Communication speed: Max. 1 Mbps
  - Data length: 0 to 8 bytes
- Number of channels: 1
- Data buffers: 16 (one receive-only buffer and 15 buffers settable for transmission/reception)
- Data transmission: Two methods
  - Mailbox (buffer) number order (low-to-high)
  - Message priority (identifier) reverse-order (high-to-low)
- Data reception: Two methods
  - Message identifier match (transmit/receive-setting buffers)
  - Reception with message identifier masked (receive-only)
- CPU interrupts: 12
  - Error interrupt
  - Reset processing interrupt
  - Message reception interrupt
  - Message transmission interrupt
- HCAN operating modes
- Support for various modes
  - Hardware reset
  - Software reset
  - Normal status (error-active, error-passive)
  - Bus off status
  - HCAN configuration mode
  - HCAN sleep mode
  - HCAN halt mode

- Module stop mode can be set

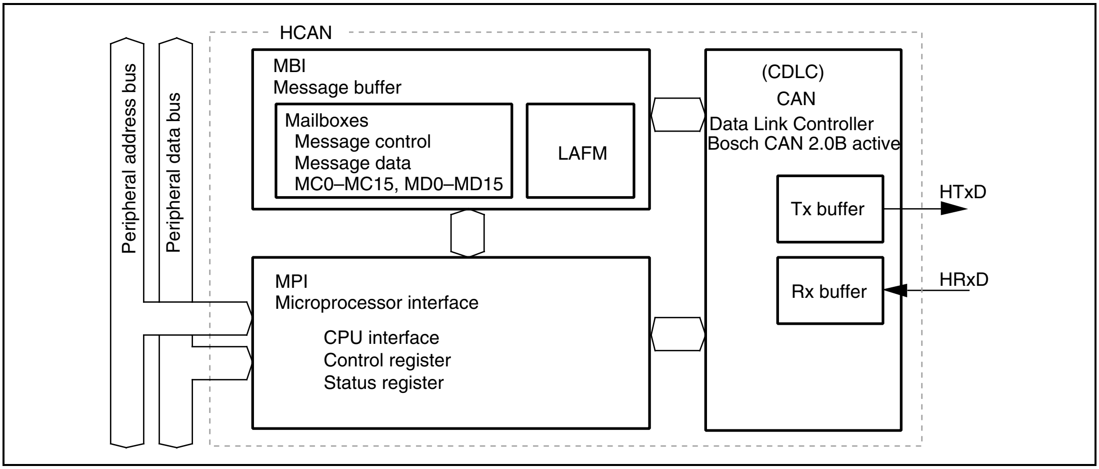


Figure 11.1 HCAN Block Diagram

- Message Buffer Interface (MBI)

  The MBI, consisting of mailboxes and a local acceptance filter mask (LAFM), stores CAN transmit/receive messages (identifiers, data, etc.) Transmit messages are written by the CPU. For receive messages, the data received by the CDLC is stored automatically.
- Microprocessor Interface (MPI)

  The MPI, consisting of a bus interface, control register, status register, etc., controls HCAN internal data, status, and so forth.
- CAN Data Link Controller (CDLC)

  The CDLC transmits and receives of messages conforming to the Bosch CAN Ver. 2.0B active standard (data frames, remote frames, error frames, overload frames, inter-frame spacing), as well as CRC checking, bus arbitration, and other functions.

11.2 Input/Output Pins

Table 11.1 shows the HCAN's pins.

When using HCAN pins, settings must be made in the HCAN configuration mode (during initialization: MCR0 = 1 and GSR3 = 1).

Table 11.1 Pin Configuration

Name	Abbreviation	Input/Output	Function
HCAN transmit data pin	HTxD	Output	CAN bus transmission pin
HCAN receive data pin	HRxD	Input	CAN bus reception pin

A bus driver is necessary for the interface between the pins and the CAN bus. A Philips PCA82C250 compatible model is recommended.

11.3 Register Descriptions

The HCAN has the following registers.

- Master control register (MCR)
- General status register (GSR)
- Bit configuration register (BCR)
- Mailbox configuration register (MBCR)
- Transmit wait register (TXPR)
- Transmit wait cancel register (TXCR)
- Transmit acknowledge register (TXACK)
- Abort acknowledge register (ABACK)
- Receive complete register (RXPR)
- Remote request register (RFPR)
- Interrupt register (IRR)
- Mailbox interrupt mask register (MBIMR)
- Interrupt mask register (IMR)
- Receive error counter (REC)
- Transmit error counter (TEC)
- Unread message status register (UMSR)
- Local acceptance filter mask H (LAFMH)
- Local acceptance filter mask L (LAFML)
- Message control (8-bit × 8 registers × 16 sets) (MC0 to MC15)
- Message data (8-bit × 8 registers × 16 sets) (MD0 to MD15)

- HCAN monitor register (HCANMON)

11.3.1 Master Control Register (MCR)

MCR controls the HCAN.

Bit	Bit Name	Initial Value	R/W	Description
7	MCR7	0	R/W	HCAN Sleep Mode Release When this bit is set to 1, the HCAN automatically exits HCAN sleep mode on detection of CAN bus operation.
6	—	0	R	Reserved This bit is always read as 0. Only 0 should be written to this bit.
5	MCR5	0	R/W	HCAN Sleep Mode When this bit is set to 1, the HCAN transits to HCAN sleep mode. When this bit is cleared to 0, HCAN sleep mode is released.
4	—	0	R	Reserved
3	—	0	R	These bits are always read as 0. Only 0 should be written to these bits.
2	MCR2	0	R/W	Message Transmission Method 0: Transmission order determined by message identifier priority 1: Transmission order determined by mailbox (buffer) number priority (TXPR1 > TXPR15)
1	MCR1	0	R/W	Halt Request When this bit is set to 1, the HCAN transits to HCAN HALT mode. When this bit is cleared to 0, HCAN HALT mode is released.

Bit	Bit Name	Initial Value	R/W	Description
0	MCR0	1	R/W	Reset Request When this bit is set to 1, the HCAN transits to reset mode. For details, refer to section 11.4.1, Hardware and Software Resets. [Setting conditions] • Power-on reset • Hardware standby • Software standby • 1-write (software reset) [Clearing condition] • When 0 is written to this bit while the GSR3 bit in GSR is 1

11.3.2 General Status Register (GSR)

GSR indicates the status of the HCAN.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. Only 0 should be written to these bits.
3	GSR3	1	R	Reset Status Bit Indicates whether the HCAN module is in the normal operating state or the reset state. This bit cannot be modified. [Setting condition] • When entering configuration mode after the HCAN internal reset has finished • Sleep mode [Clearing condition] • When entering normal operation mode after the MCR0 bit in MCR is cleared to 0 (Note that there is a delay between clearing of the MCR0 bit and the GSR3 bit.)

Bit	Bit Name	Initial Value	R/W	Description
2	GSR2	1	R	Message Transmission Status Flag Flag that indicates whether the module is currently in the message transmission period. This bit cannot be modified. [Setting condition] • Start of message transmission (SOF) [Clearing condition] • Interval of three bits after EOF (End of Frame)
1	GSR1	0	R	Transmit/Receive Warning Flag This bit cannot be modified. [Clearing condition] • When TEC < 96 and REC < 96 or TEC ≥ 256 [Setting condition] • When TEC ≥ 96 or REC ≥ 96
0	GSR0	0	R	Bus Off Flag This bit cannot be modified. [Setting condition] • When TEC ≥ 256 (bus off state) [Clearing condition] • Recovery from bus off state

11.3.3 Bit Configuration Register (BCR)

BCR sets HCAN bit timing parameters and the baud rate prescaler. For details on parameters, refer to section 11.4.2, Initialization after Hardware Reset.

Bit	Bit Name	Initial Value	R/W	Description
15	BCR7	0	R/W	Re-Synchronization Jump Width (SJW)
14	BCR6	0	R/W	Set the maximum bit synchronization width.
				00: 1 time quantum
				01: 2 time quanta
				10: 3 time quanta
				11: 4 time quanta
13	BCR5	0	R/W	Baud Rate Prescaler (BRP)
12	BCR4	0	R/W	Set the length of time quantum.
11	BCR3	0	R/W	000000: 2 × system clock
10	BCR2	0	R/W	000001: 4 × system clock
9	BCR1	0	R/W	000010: 6 × system clock
8	BCR0	0	R/W	:
				111111: 128 × system clock
7	BCR15	0	R/W	Bit Sample Point (BSP)
				Sets the point at which data is sampled.
				0: Bit sampling at one point (end of time segment 1 (TSEG1))
				1: Bit sampling at three points (end of TSEG1 and preceding and following one time quantum)
6	BCR14	0	R/W	Time Segment 2 (TSEG2)
5	BCR13	0	R/W	Set the TSEG2 width within a range of 2 to 8 time quanta.
4	BCR12	0	R/W	000: Setting prohibited
				001: 2 time quanta
				010: 3 time quanta
				011: 4 time quanta
				100: 5 time quanta
				101: 6 time quanta
				110: 7 time quanta
				111: 8 time quanta

Bit	Bit Name	Initial Value	R/W	Description
3	BCR11	0	R/W	Time Segment 1 (TSEG1)
2	BCR10	0	R/W	Set the TSEG1 (PRSEG + PHSEG1) width to between 4 and 16 time quanta.
1	BCR9	0	R/W	0000: Setting prohibited 0001: Setting prohibited 0010: Setting prohibited 0011: 4 time quanta 0100: 5 time quanta 0101: 6 time quanta 0110: 7 time quanta 0111: 8 time quanta 1000: 9 time quanta 1001: 10 time quanta 1010: 11 time quanta 1011: 12 time quanta 1100: 13 time quanta 1101: 14 time quanta 1110: 15 time quanta 1111: 16 time quanta
0	BCR8	0	R/W	

11.3.4 Mailbox Configuration Register (MBCR)

MBCR sets the transfer direction for each mailbox.

Bit	Bit Name	Initial Value	R/W	Description
15	MBCR7	0	R/W	These bits set the transfer direction for the corresponding mailboxes from 1 to 15. MBCRn determines the transfer direction for mailbox n (n =1 to 15). 0: Corresponding mailbox is set for transmission 1: Corresponding mailbox is set for reception Bit 8 is reserved. This bit is always read as 1 and the write value should always be 1.
14	MBCR6	0	R/W	
13	MBCR5	0	R/W	
12	MBCR4	0	R/W	
11	MBCR3	0	R/W	
10	MBCR2	0	R/W	
9	MBCR1	0	R/W	
8	—	1	R	
7	MBCR15	0	R/W	
6	MBCR14	0	R/W	
5	MBCR13	0	R/W	
4	MBCR12	0	R/W	
3	MBCR11	0	R/W	
2	MBCR10	0	R/W	
1	MBCR9	0	R/W	
0	MBCR8	0	R/W	

11.3.5 Transmit Wait Register (TXPR)

TXPR sets a transmit wait after a transmit message is stored in a mailbox (buffer) (CAN bus arbitration wait).

Bit	Bit Name	Initial Value	R/W	Description
15	TXPR7	0	R/W	These bits set a transmit wait (CAN bus arbitration wait) for the corresponding mailboxes 1 to 15. When TXPRn (n = 1 to 15) is set to 1, the message in mailbox n becomes the transmit wait state. [Clearing condition] • Completion of message transmission • Completion of transmission cancellation Bit 8 is reserved. This bit is always read as 1 and the write value should always be 1.
14	TXPR6	0	R/W	
13	TXPR5	0	R/W	
12	TXPR4	0	R/W	
11	TXPR3	0	R/W	
10	TXPR2	0	R/W	
9	TXPR1	0	R/W	
8	—	0	R	
7	TXPR15	0	R/W	
6	TXPR14	0	R/W	
5	TXPR13	0	R/W	
4	TXPR12	0	R/W	
3	TXPR11	0	R/W	
2	TXPR10	0	R/W	
1	TXPR9	0	R/W	
0	TXPR8	0	R/W	

11.3.6 Transmit Wait Cancel Register (TXCR)

TXCR controls canceling transmission of transmit wait messages in mailboxes (buffers).

Bit	Bit Name	Initial Value	R/W	Description
15	TXCR7	0	R/W	These bits cancel the transmit wait message in the corresponding mailboxes 1 to 15. When TXCRn (n = 1 to 15) is set to 1, the transmit wait message in mailbox n is canceled. [Clearing condition] • Completion of TXPR clearing when transmit message is canceled normally Bit 8 is reserved. This bit is always read as 0 and the write value should always be 0.
14	TXCR6	0	R/W	
13	TXCR5	0	R/W	
12	TXCR4	0	R/W	
11	TXCR3	0	R/W	
10	TXCR2	0	R/W	
9	TXCR1	0	R/W	
8	—	0	R	
7	TXCR15	0	R/W	
6	TXCR14	0	R/W	
5	TXCR13	0	R/W	
4	TXCR12	0	R/W	
3	TXCR11	0	R/W	
2	TXCR10	0	R/W	
1	TXCR9	0	R/W	
0	TXCR8	0	R/W	

11.3.7 Transmit Acknowledge Register (TXACK)

TXACK is a status register that indicates the normal transmission of mailbox (buffer) transmit messages.

Bit	Bit Name	Initial Value	R/W	Description
15	TXACK7	0	R/W	These bits are status flags that indicate error-free transmission of the transmit message in the corresponding mailboxes 1 to 15. When the message in mailbox n (n = 1 to 15) has been transmitted error-free, TXACKn is set to 1. [Setting condition] • Completion of message transmission for corresponding mailbox [Clearing condition] • Writing 1 Bit 8 is reserved. This bit is always read as 0 and the write value should always be 0.
14	TXACK6	0	R/W	
13	TXACK5	0	R/W	
12	TXACK4	0	R/W	
11	TXACK3	0	R/W	
10	TXACK2	0	R/W	
9	TXACK1	0	R/W	
8	—	0	R	
7	TXACK15	0	R/W	
6	TXACK14	0	R/W	
5	TXACK13	0	R/W	
4	TXACK12	0	R/W	
3	TXACK11	0	R/W	
2	TXACK10	0	R/W	
1	TXACK9	0	R/W	
0	TXACK8	0	R/W	

11.3.8 Abort Acknowledge Register (ABACK)

ABACK is a status register that indicates the normal cancellation (aborting) of mailbox (buffer) transmit messages.

Bit	Bit Name	Initial Value	R/W	Description
15	ABACK7	0	R/(W)*	These bits are status flags that indicate error-free cancellation (abortion) of the transmit message in the corresponding mailboxes 1 to 15. When the message in mailbox n (n = 1 to 15) has been canceled error-free, ABACKn is set to 1. [Setting condition] • Completion of transmit message cancellation for corresponding mailbox [Clearing condition] • Writing 1 Bit 8 is reserved. This bit is always read as 0. The write value should always be 0.
14	ABACK6	0	R/(W)*	
13	ABACK5	0	R/(W)*	
12	ABACK4	0	R/(W)*	
11	ABACK3	0	R/(W)*	
10	ABACK2	0	R/(W)*	
9	ABACK1	0	R/(W)*	
8	—	0	R	
7	ABACK15	0	R/(W)*	
6	ABACK14	0	R/(W)*	
5	ABACK13	0	R/(W)*	
4	ABACK12	0	R/(W)*	
3	ABACK11	0	R/(W)*	
2	ABACK10	0	R/(W)*	
1	ABACK9	0	R/(W)*	
0	ABACK8	0	R/(W)*	

Note: * Only 1 can be written for clearing the flag.

11.3.9 Receive Complete Register (RXPR)

RXPR is a status register that indicates the normal reception of messages (data frame or remote frame) in mailboxes (buffers). For reception of a remote frame, when a bit in this register is set to 1, the corresponding remote request register (RFPR) bit is also set to 1 simultaneously.

Bit	Bit Name	Initial Value	R/W	Description
15	RXPR7	0	R/(W)*	When the message in mailbox n (n = 0 to 15) has been received error-free, RXPRn is set to 1. [Setting condition] • Completion of message (data frame or remote frame) reception in corresponding mailbox [Clearing condition] • Writing 1
14	RXPR6	0	R/(W)*	
13	RXPR5	0	R/(W)*	
12	RXPR4	0	R/(W)*	
11	RXPR3	0	R/(W)*	
10	RXPR2	0	R/(W)*	
9	RXPR1	0	R/(W)*	
8	RXPR0	0	R/(W)*	
7	RXPR15	0	R/(W)*	
6	RXPR14	0	R/(W)*	
5	RXPR13	0	R/(W)*	
4	RXPR12	0	R/(W)*	
3	RXPR11	0	R/(W)*	
2	RXPR10	0	R/(W)*	
1	RXPR9	0	R/(W)*	
0	RXPR8	0	R/(W)*	

Note: * Only 1 can be written for clearing the flag.

11.3.10 Remote Request Register (RFPR)

RFPR is a status register that indicates normal reception of remote frames in mailboxes (buffers). When a bit in this register is set to 1, the corresponding receive complete register (RXPR) bit is also set to 1 simultaneously.

Bit	Bit Name	Initial Value	R/W	Description
15	RFPR7	0	R/(W)*	When mailbox n (n = 0 to 15) has received the remote frame error-free, RFPRn (n = 0 to 15) is set to 1. [Setting condition] • Completion of remote frame reception in corresponding mailbox [Clearing condition] • Writing 1
14	RFPR6	0	R/(W)*	
13	RFPR5	0	R/(W)*	
12	RFPR4	0	R/(W)*	
11	RFPR3	0	R/(W)*	
10	RFPR2	0	R/(W)*	
9	RFPR1	0	R/(W)*	
8	RFPR0	0	R/(W)*	
7	RFPR15	0	R/(W)*	
6	RFPR14	0	R/(W)*	
5	RFPR13	0	R/(W)*	
4	RFPR12	0	R/(W)*	
3	RFPR11	0	R/(W)*	
2	RFPR10	0	R/(W)*	
1	RFPR9	0	R/(W)*	
0	RFPR8	0	R/(W)*	

Note: * Only 1 can be written for clearing the flag.

11.3.11 Interrupt Register (IRR)

IRR is an interrupt flag register.

Bit	Bit Name	Initial Value	R/W	Description
15	IRR7	0	R/(W)*	Overload Frame [Setting condition] • When an overload frame is transmitted in error active/passive state [Clearing condition] • Writing 1
14	IRR6	0	R/(W)*	Bus Off Interrupt Flag Status flag indicating the bus off state caused by the transmit error counter. [Setting condition] • When TEC ≥ 256 [Clearing condition] • Writing 1
13	IRR5	0	R/(W)*	Error Passive Interrupt Flag Status flag indicating the error passive state caused by the transmit/receive error counter. [Setting condition] When TEC ≥ 128 or REC ≥ 128 [Clearing condition] • Writing 1
12	IRR4	0	R/(W)*	Receive Overload Warning Interrupt Flag Status flag indicating the error warning state caused by the receive error counter. [Setting condition] When REC ≥ 96 [Clearing condition] • Writing 1

Bit	Bit Name	Initial Value	R/W	Description
11	IRR3	0	R/(W)*	Transmit Overload Warning Interrupt Flag Status flag indicating the error warning state caused by the transmit error counter. [Setting condition] • When TEC ≥ 96 [Clearing condition] • Writing 1
10	IRR2	0	R	Remote Frame Request Interrupt Flag Status flag indicating that a remote frame has been received in a mailbox (buffer). [Setting condition] • When remote frame reception is completed, when corresponding MBIMR = 0 [Clearing condition] • Clearing of all bits in RFPR (remote request register)
9	IRR1	0	R	Receive Message Interrupt Flag Status flag indicating that a mailbox (buffer) receive message has been received normally. [Setting condition] • When data frame or remote frame reception is completed, when corresponding MBIMR = 0 [Clearing condition] • Clearing of all bits in RXPR (receive complete register)

Bit	Bit Name	Initial Value	R/W	Description
8	IRR0	1	R/(W)*	Reset Interrupt Flag Status flag indicating that the HCAN module has been reset. This bit cannot be masked by the interrupt mask register (IMR). If this bit is not cleared to 0 after entering power-on reset or returning from software standby mode, interrupt processing will start immediately when the interrupt controller enables interrupts. [Setting condition] • When the reset operation has finished after entering power-on reset or software standby mode [Clearing condition] • Writing 1
7 to 5	—	All 0	—	Reserved These bits are always read as 0. Only 0 should be written to these bits.
4	IRR12	0	R/(W)*	Bus Operation Interrupt Flag Status flag indicating detection of a dominant bit due to bus operation when the HCAN module is in HCAN sleep mode. [Setting condition] • Bus operation (dominant bit) detection in HCAN sleep mode [Clearing condition] • Writing 1
3	—	0	—	Reserved
2	—	0	—	These bits are always read as 0. Only 0 should be written to these bits.
1	IRR9	0	R	Unread Interrupt Flag Status flag indicating that a receive message has been overwritten before being read. [Setting condition] • When UMSR (unread message status register) is set [Clearing condition] • Clearing of all bits in UMSR (unread message status register)

Bit	Bit Name	Initial Value	R/W	Description
0	IRR8	0	R/(W)*	Mailbox Empty Interrupt Flag Status flag indicating that the next transmit message can be stored in the mailbox. [Setting condition] • When TXPR (transmit wait register) is cleared by completion of transmission or completion of transmission abort [Clearing condition] • Writing 1

Note: * Only 1 can be written for clearing the flag.

11.3.12 Mailbox Interrupt Mask Register (MBIMR)

MBIMR controls the enabling or disabling of individual mailbox (buffer) interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
15	MBIMR7	1	R/W	Mailbox Interrupt Mask (MBIMRx) When MBIMRn (n = 1 to 15) is cleared to 0, the interrupt request in mailbox n is enabled. When set to 1, the interrupt request is masked. The interrupt source in a transmit mailbox is TXPR clearing caused by transmission end or transmission cancellation. The interrupt source in a receive mailbox is RXPR setting on reception end.
14	MBIMR6	1	R/W	
13	MBIMR5	1	R/W	
12	MBIMR4	1	R/W	
11	MBIMR3	1	R/W	
10	MBIMR2	1	R/W	
9	MBIMR1	1	R/W	
8	MBIMR0	1	R/W	
7	MBIMR15	1	R/W	
6	MBIMR14	1	R/W	
5	MBIMR13	1	R/W	
4	MBIMR12	1	R/W	
3	MBIMR11	1	R/W	
2	MBIMR10	1	R/W	
1	MBIMR9	1	R/W	
0	MBIMR8	1	R/W	

11.3.13 Interrupt Mask Register (IMR)

IMR enables or disables interrupt requests by the IRR interrupt flags. The reset interrupt flag cannot be masked.

Bit	Bit Name	Initial Value	R/W	Description
15	IMR7	1	R/W	Overload Frame/Bus Off Recovery Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt request by IRR7) is enabled. When set to 1, OVR0 is masked.
14	IMR6	1	R/W	Bus Off Interrupt Mask When this bit is cleared to 0, ERS0 (interrupt request by IRR6) is enabled. When set to 1, ERS0 is masked.
13	IMR5	1	R/W	Error Passive Interrupt Mask When this bit is cleared to 0, ERS0 (interrupt request by IRR5) is enabled. When set to 1, ERS0 is masked.
12	IMR4	1	R/W	Receive Overload Warning Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt request by IRR4) is enabled. When set to 1, OVR0 is masked.
11	IMR3	1	R/W	Transmit Overload Warning Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt request by IRR3) is enabled. When set to 1, OVR0 is masked.
10	IMR2	1	R/W	Remote Frame Request Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt request by IRR2) is enabled. When set to 1, OVR0is masked.
9	IMR1	1	R/W	Receive Message Interrupt Mask When this bit is cleared to 0, RM1 (interrupt request by IRR1) is enabled. When set to 1, RMI is masked.
8	—	0	R	Reserved This bit is always read as 0. Only 0 should be written to this bit.
7 to 5	—	All 1	R	Reserved These bits are always read as 1. Only 1 should be written to these bits.

Bit	Bit Name	Initial Value	R/W	Description
4	IMR12	1	R/W	Bus Operation Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt request by IRR12) is enabled. When set to 1, OVR0 is masked.
3	—	1	R	Reserved
2	—	1	R	These bits are always read as 1. Only 1 should be written to these bits.
1	IMR9	1	R/W	Unread Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt request by IRR9) is enabled. When set to 1, OVR0 is masked.
0	IMR8	1	R/W	Mailbox Empty Interrupt Mask When this bit is cleared to 0, SLE0 (interrupt request by IRR8) is enabled. When set to 1, SLE0 is masked.

11.3.14 Receive Error Counter (REC)

REC is an 8-bit read-only register that functions as a counter indicating the number of receive message errors on the CAN bus. The count value is stipulated in the CAN protocol.

11.3.15 Transmit Error Counter (TEC)

TEC is an 8-bit read-only register that functions as a counter indicating the number of transmit message errors on the CAN bus. The count value is stipulated in the CAN protocol.

11.3.16 Unread Message Status Register (UMSR)

UMSR is a status register that indicates, for individual mailboxes (buffers), that a received message has been overwritten by a new receive message before being read. When overwritten by a new message, data in the unread receive message is lost.

Bit	Bit Name	Initial Value	R/W	Description
15	UMSR7	0	R/(W)*	[Setting condition] When a new message is received before RXPR is cleared [Clearing condition] Writing 1
14	UMSR6	0	R/(W)*	
13	UMSR5	0	R/(W)*	
12	UMSR4	0	R/(W)*	
11	UMSR3	0	R/(W)*	
10	UMSR2	0	R/(W)*	
9	UMSR1	0	R/(W)*	
8	UMSR0	0	R/(W)*	
7	UMSR15	0	R/(W)*	
6	UMSR14	0	R/(W)*	
5	UMSR13	0	R/(W)*	
4	UMSR12	0	R/(W)*	
3	UMSR11	0	R/(W)*	
2	UMSR10	0	R/(W)*	
1	UMSR9	0	R/(W)*	
0	UMSR8	0	R/(W)*	

Note: * Only 1 can be written for clearing the flag.

11.3.17 Local Acceptance Filter Masks (LAFML, LAFMH)

LAFML and LAFMH set the identifier bits of the message to be stored in mailbox 0 as Don't Care. For details, refer to section 11.4.4, Message Reception. The relationship between the identifier bits and mask bits are shown in the following.

LAFML

Bit	Bit Name	Initial Value	R/W	Description
15	LAFML7	0	R/W	When this bit is set to 1, ID-7 of the receive message identifier is not compared.
14	LAFML6	0	R/W	When this bit is set to 1, ID-6 of the receive message identifier is not compared.
13	LAFML5	0	R/W	When this bit is set to 1, ID-5 of the receive message identifier is not compared.
12	LAFML4	0	R/W	When this bit is set to 1, ID-4 of the receive message identifier is not compared.
11	LAFML3	0	R/W	When this bit is set to 1, ID-3 of the receive message identifier is not compared.
10	LAFML2	0	R/W	When this bit is set to 1, ID-2 of the receive message identifier is not compared.
9	LAFML1	0	R/W	When this bit is set to 1, ID-1 of the receive message identifier is not compared.
8	LAFML0	0	R/W	When this bit is set to 1, ID-0 of the receive message identifier is not compared.
7	LAFML15	0	R/W	When this bit is set to 1, ID-15 of the receive message identifier is not compared.
6	LAFML14	0	R/W	When this bit is set to 1, ID-14 of the receive message identifier is not compared.
5	LAFML13	0	R/W	When this bit is set to 1, ID-13 of the receive message identifier is not compared.
4	LAFML12	0	R/W	When this bit is set to 1, ID-12 of the receive message identifier is not compared.
3	LAFML11	0	R/W	When this bit is set to 1, ID-11 of the receive message identifier is not compared.
2	LAFML10	0	R/W	When this bit is set to 1, ID-10 of the receive message identifier is not compared.
1	LAFML9	0	R/W	When this bit is set to 1, ID-9 of the receive message identifier is not compared.
0	LAFML8	0	R/W	When this bit is set to 1, ID-8 of the receive message identifier is not compared.

LAFMH

Bit	Bit Name	Initial Value	R/W	Description
15	LAFMH7	0	R/W	When this bit is set to 1, ID-20 of the receive message identifier is not compared.
14	LAFMH6	0	R/W	When this bit is set to 1, ID-19 of the receive message identifier is not compared.
13	LAFMH5	0	R/W	When this bit is set to 1, ID-18 of the receive message identifier is not compared.
12 to 10	—	All 0	R	Reserved These bits are always read as 0. Only 0 should be written to these bits.
9	LAFMH1	0	R/W	When this bit is set to 1, ID-17 of the receive message identifier is not compared.
8	LAFMH0	0	R/W	When this bit is set to 1, ID-16 of the receive message identifier is not compared.
7	LAFMH15	0	R/W	When this bit is set to 1, ID-28 of the receive message identifier is not compared.
6	LAFMH14	0	R/W	When this bit is set to 1, ID-27 of the receive message identifier is not compared.
5	LAFMH13	0	R/W	When this bit is set to 1, ID-26 of the receive message identifier is not compared.
4	LAFMH12	0	R/W	When this bit is set to 1, ID-25 of the receive message identifier is not compared.
3	LAFMH11	0	R/W	When this bit is set to 1, ID-24 of the receive message identifier is not compared.
2	LAFMH10	0	R/W	When this bit is set to 1, ID-23 of the receive message identifier is not compared.
1	LAFMH9	0	R/W	When this bit is set to 1, ID-22 of the receive message identifier is not compared.
0	LAFMH8	0	R/W	When this bit is set to 1, ID-21 of the receive message identifier is not compared.

11.3.18 Message Control (MC0 to MC15)

The message control register sets consist of eight 8-bit registers for one mailbox. The HCAN has 16 sets of these registers. Because message control registers are in RAM, their initial values after power-on are undefined. Be sure to initialize them by writing 0 or 1. Figure 11.2 shows the register names for each mailbox.

Mail box 0	MC0[1]	MC0[2]	MC0[3]	MC0[4]	MC0[5]	MC0[6]	MC0[7]	MC0[8]
Mail box 1	MC1[1]	MC1[2]	MC1[3]	MC1[4]	MC1[5]	MC1[6]	MC1[7]	MC1[8]
Mail box 2	MC2[1]	MC2[2]	MC2[3]	MC2[4]	MC2[5]	MC2[6]	MC2[7]	MC2[8]
Mail box 3	MC3[1]	MC3[2]	MC3[3]	MC3[4]	MC3[5]	MC3[6]	MC3[7]	MC3[8]
Mail box 15	MC15[1]	MC15[2]	MC15[3]	MC15[4]	MC15[5]	MC15[6]	MC15[7]	MC15[8]

Figure 11.2 Message Control Register Configuration

The setting of message control registers are shown in the following. Figures 11.3 and 11.4 show the correspondence between the identifiers and register bit names.

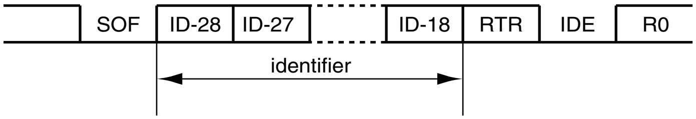


Figure 11.3 Standard Format

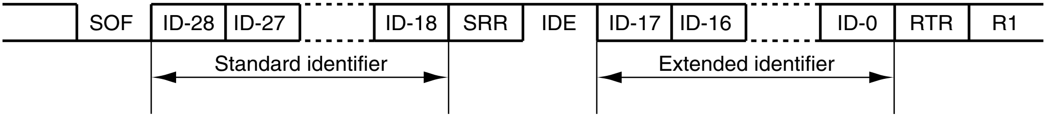


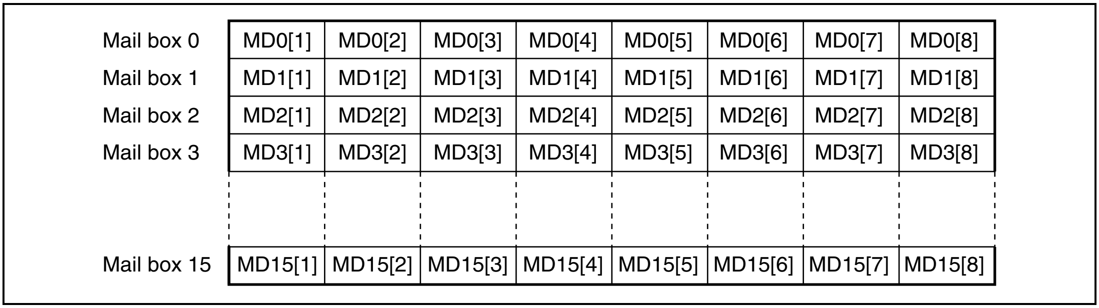
Figure 11.4 Extended Format

Register Name	Bit	Bit Name	R/W	Description
MCx[1]	7 to 4	—	R/W	The initial value of these bits is undefined; they must be initialized (by writing 0 or 1).
	3 to 0	DLC3 to DLC0	R/W	Data Length Code Set the data length of a data frame or the data length requested in a remote frame within the range of 0 to 8 bits. 0000: 0 byte 0001: 1 byte 0010: 2 bytes 0011: 3 bytes 0100: 4 bytes 0101: 5 bytes 0110: 6 bytes 0111: 7 bytes 1000: 8 bytes : : 1111: 8 bytes
MCx[2]	7 to 0	—	R/W	The initial value of these bits is undefined; they must be initialized (by writing 0 or 1).
MCx[3]	7 to 0	—	R/W	
MCx[4]	7 to 0	—	R/W	
MCx[5]	7 to 5	ID-20 to ID-18	R/W	Sets ID-20 to ID-18 in the identifier.
	4	RTR	R/W	Remote Transmission Request Used to distinguish between data frames and remote frames. 0: Data frame 1: Remote frame
	3	IDE	R/W	Identifier Extension Used to distinguish between the standard format and extended format of data frames and remote frames. 0: Standard format 1: Extended format
	2	—	R/W	The initial value of this bit is undefined. It must be initialized by writing 0 or 1.
	1 to 0	ID-17 to ID-16	R/W	Sets ID-17 and ID-16 in the identifier.
MCx[6]	7 to 0	ID-28 to ID-21	R/W	Sets ID-28 to ID-21 in the identifier.
MCx[7]	7 to 0	ID-7 to ID-0	R/W	Sets ID-7 to ID-0 in the identifier.
MCx[8]	7 to 0	ID-15 to ID-8	R/W	Sets ID-15 to ID-8 in the identifier.

Note: x: Mailbox number

11.3.19 Message Data (MD0 to MD15)

The message data register sets consist of eight 8-bit registers for one mailbox. The HCAN has 16 sets of these registers. Because message data registers are in RAM, their initial values after power-on are undefined. Be sure to initialize them by writing 0 or 1. Figure 11.5 shows the register names for each mailbox.



Mail box 0	MD0[1]	MD0[2]	MD0[3]	MD0[4]	MD0[5]	MD0[6]	MD0[7]	MD0[8]
Mail box 1	MD1[1]	MD1[2]	MD1[3]	MD1[4]	MD1[5]	MD1[6]	MD1[7]	MD1[8]
Mail box 2	MD2[1]	MD2[2]	MD2[3]	MD2[4]	MD2[5]	MD2[6]	MD2[7]	MD2[8]
Mail box 3	MD3[1]	MD3[2]	MD3[3]	MD3[4]	MD3[5]	MD3[6]	MD3[7]	MD3[8]
Mail box 15	MD15[1]	MD15[2]	MD15[3]	MD15[4]	MD15[5]	MD15[6]	MD15[7]	MD15[8]

Figure 11.5 Message Data Configuration

11.3.20 HCAN Monitor Register (HCANMON)

HCANMON enables/disables an interrupt by the HCAN reception, controls transmit stop by the HTxD pin, and reflects the states of the HCAN pins.

Bit	Bit Name	Initial Value	R/W	Description
7	RxDIE	0	R/W	HRxD Interrupt Enable Selects whether the $\overline{\text{IRQ2}}$ interrupt is input from the PF0 pin or HRxD pin. 0: $\overline{\text{IRQ2}}$ interrupt generated by input of the PF0 pin 1: $\overline{\text{IRQ2}}$ interrupt generated by input of the HRxD pin
6	TxSTP	0	R/W	HTxD Transmit Stop bit Controls the transmit stop by the HTxD pin. 0: The HTxD pin enables transmission. 1: The HTxD pin is fixed to output 1 and transmission is stopped.
5 to 2	—	Undefined	—	Reserved The read value is undefined. These bits cannot be modified.
1	TxD	Undefined	R	Transmission pin The state of the HTxD pin is read. This bit cannot be modified.
0	RxD	Undefined	R	Reception pin The state of the HRxD pin is read. This bit cannot be modified.

11.4 Operation

11.4.1 Hardware and Software Resets

The HCAN can be reset by a hardware reset or software reset.

- Hardware Reset

  At power-on reset, or in hardware or software standby mode, the HCAN is initialized by automatically setting the MCR reset request bit (MCR0) in MCR and the reset state bit (GSR3) in GSR. At the same time, all internal registers, except for message control and message data registers, are initialized by a hardware reset.
- Software Reset

  The HCAN can be reset by setting the MCR reset request bit (MCR0) in MCR via software. In a software reset, the error counters (TEC and REC) are initialized, however other registers are not. If the MCR0 bit is set while the CAN controller is performing a communication operation (transmission or reception), the initialization state is not entered until message transfer has been completed. The reset status bit (GSR3) in GSR is set on completion of initialization.

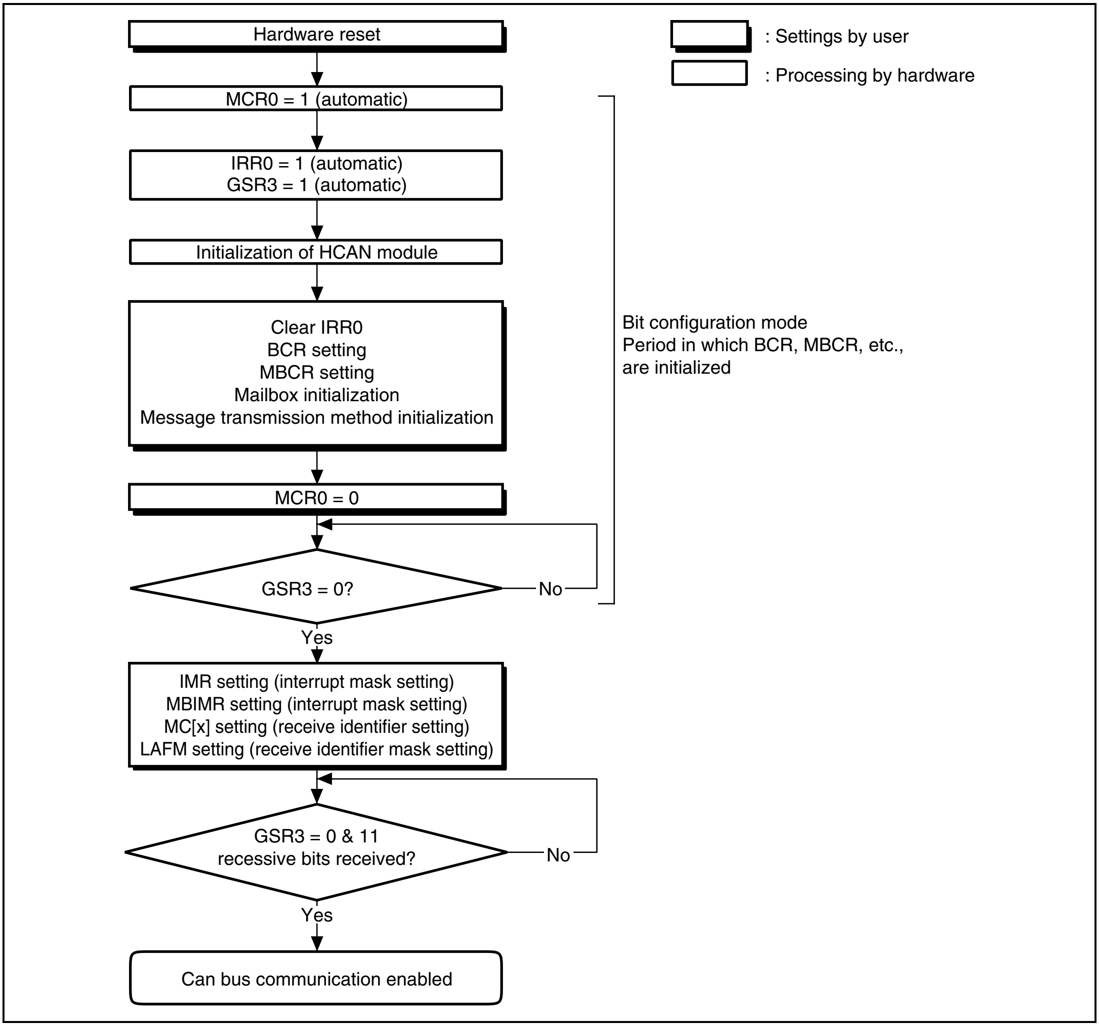
11.4.2 Initialization after Hardware Reset

After a hardware reset, the following initialization processing should be carried out:

1. Clearing of IRR0 bit in the interrupt register (IRR)
2. Bit rate setting
3. Mailbox transmit/receive settings
4. Mailbox (RAM) initialization
5. Message transmission method setting

These initial settings must be made while the HCAN is in bit configuration mode. Configuration mode is a state in which the GSR3 bit in GSR is set to 1 by a reset. Configuration mode is exited by clearing the MCR0 bit in MCR to 0; when the MCR0 bit is cleared to 0, the HCAN automatically clears the GSR3 bit in GSR. There is a delay between clearing the MCR0 bit and clearing the GSR3 bit because the HCAN needs time to be internally reset, there is a delay between clearing of the MCR0 bit and GSR3 bit. After the HCAN exits configuration mode, the power-up sequence begins, and communication with the CAN bus is possible as soon as 11 consecutive recessive bits have been detected.

IRR0 Clearing: The reset interrupt flag (IRR0) is always set after a power-on reset or recovery from software standby mode. As an HCAN interrupt is initiated immediately when interrupts are enabled, IRR0 should be cleared.



Hardware reset
: Settings by user
: Processing by hardware
MCR0 = 1 (automatic)
IRR0 = 1 (automatic)
GSR3 = 1 (automatic)
Initialization of HCAN module
Clear IRR0
BCR setting
MBCR setting
Mailbox initialization
Message transmission method initialization
Bit configuration mode
Period in which BCR, MBCR, etc.,
are initialized
MCR0 = 0
GSR3 = 0?
No
Yes
IMR setting (interrupt mask setting)
MBIMR setting (interrupt mask setting)
MC[x] setting (receive identifier setting)
LAFM setting (receive identifier mask setting)
GSR3 = 0 & 11
recessive bits received?
No
Yes
Can bus communication enabled

Figure 11.6 Hardware Reset Flowchart

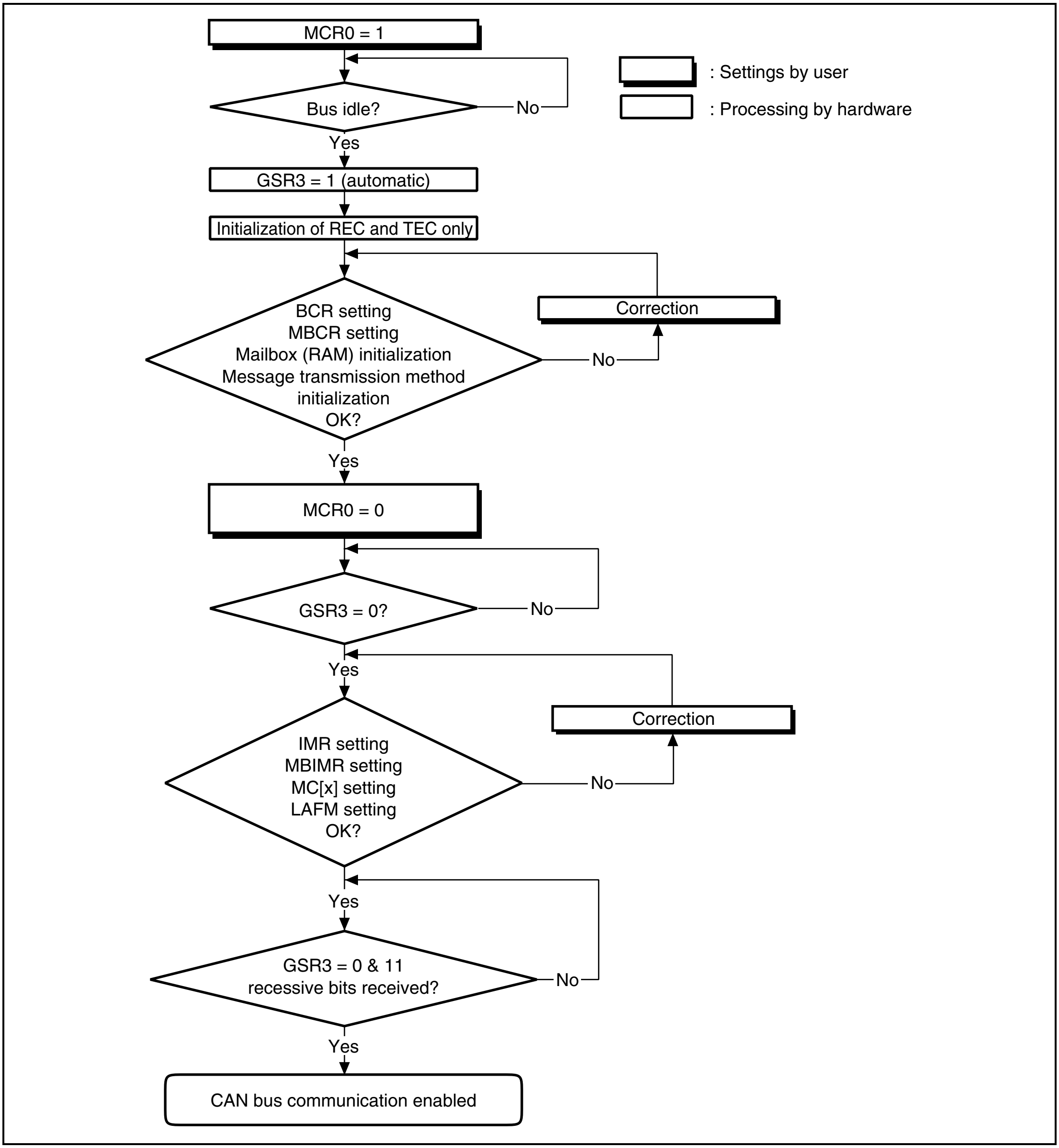


Figure 11.7 Software Reset Flowchart

Bit Rate and Bit Timing Settings: The bit rate and bit timing settings are made in the bit configuration register (BCR). Settings should be made such that all CAN controllers connected to the CAN bus have the same baud rate and bit width. The 1-bit time consists of the total of the settable time quantum (tq).

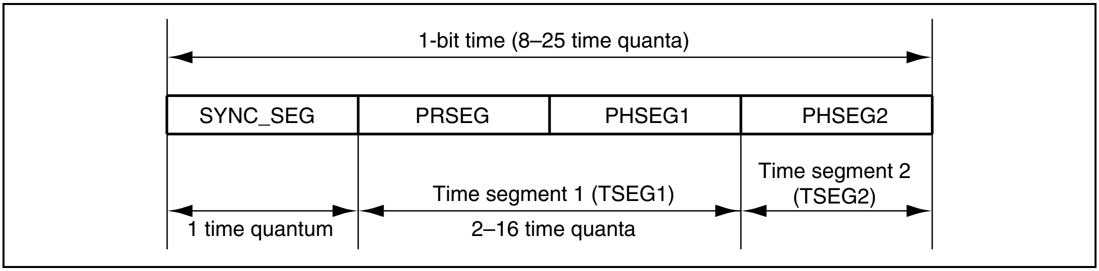


Figure 11.8 Detailed Description of One Bit

SYNC_SEG is a segment for establishing the synchronization of nodes on the CAN bus. Normal bit edge transitions occur in this segment. PRSEG is a segment for compensating for the physical delay between networks. PHSEG1 is a buffer segment for correcting phase drift (positive). This segment is extended when synchronization (resynchronization) is established. PHSEG2 is a buffer segment for correcting phase drift (negative). This segment is shortened when synchronization (resynchronization) is established. Limits on the settable value (TSEG1, TSEG2, BRP, sample point, and SJW) are shown in table 11.2.

Table 11.2 Limits for Settable Value

Name	Abbreviation	Min. Value	Max. Value
Time segment 1	TSEG1	B'0011*[2]	B'1111
Time segment 2	TSEG2	B'001*[3]	B'111
Baud rate prescaler	BRP	B'000000	B'111111
Bit sample point	BSP	B'0	B'1
Re-synchronization jump width	SJW*[1]	B'00	B'11

Notes: 1. SJW is stipulated in the CAN specifications:
$3 \geq \text{SJW} \geq 0$

2. The minimum value of TSEG2 is stipulated in the CAN specifications:
$\text{TSEG2} \geq \text{SJW}$

3. The minimum value of TSEG1 is stipulated in the CAN specifications:
$\text{TSEG1} > \text{TSEG2}$

Time Quanta (TQ) is an integer multiple of the number of system clocks, and is determined by the baud rate prescaler (BRP) as follows. f_{CLK} is the system clock frequency.

$TQ = 2 \times (\text{BPR setting} + 1)/f_{CLK}$

The following formula is used to calculate the 1-bit time and bit rate.

$\text{1-bit time} = TQ \times (3 + TSEG1 + TSEG2)$
$\text{Bit rate} = 1/\text{Bit time}$
$= f_{CLK}/\{2 \times (\text{BPR setting} + 1) \times (3 + TSEG1 + TSEG2)\}$

Note: $f_{CLK} = \phi$ (system clock)
A BCR value is used for BRP, TSEG1, and TSEG2.

Example: With a system clock of 24 MHz, a BRP setting of B'000000, a TSEG1 setting of B'0101, and a TSEG2 setting of B′100:

$\text{Bit rate} = 24/\{2 \times (0 + 1) \times (3 + 5 + 4)\} = 1$ Mbps

Table 11.3 Setting Range for TSEG1 and TSEG2 in BCR

		TSEG2 (BCR[14:12])						
		001	010	011	100	101	110	111
TSEG1 (BCR[11:8])	0011	No	Yes	No	No	No	No	No
	0100	Yes*	Yes	Yes	No	No	No	No
	0101	Yes*	Yes	Yes	Yes	No	No	No
	0110	Yes*	Yes	Yes	Yes	Yes	No	No
	0111	Yes*	Yes	Yes	Yes	Yes	Yes	No
	1000	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1001	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1010	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1011	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1100	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1101	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1110	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1111	Yes*	Yes	Yes	Yes	Yes	Yes	Yes

Note: The time quantum value for TSEG1 and TSEG2 is the TSEG value + 1.
* Only a value other than BRP[13:8] = B’000000 can be set.

Mailbox Transmit/Receive Settings: The HCAN has 16 mailboxes. Mailbox 0 is receive-only, while mailboxes 1 to 15 can be set for transmission or reception. The Initial status of mailboxes 1 to 15 is for transmission. Mailbox transmit/receive settings are not initialized by a software reset.

Mailbox Transmit/Receive Settings: The HCAN has 16 mailboxes. Mailbox 0 is receive-only, while mailboxes 1 to 15 can be set for transmission or reception. The Initial status of mailboxes 1 to 15 is for transmission. Mailbox transmit/receive settings are not initialized by a software reset.

Clearing a bit to 0 in the mailbox configuration register (MBCR) designates the corresponding mailbox for transmission use, whereas a setting of 1 in MBCR designates the corresponding mailbox for reception use. When setting mailboxes for reception, in order to improve message reception efficiency, high-priority messages should be set in low-to-high mailbox order.

Mailbox (Message Control/Data) Initial Settings: Message control/data are held in RAM, and so their initial values are undefined after power is supplied. Initial values must therefore be set in all the mailboxes (by writing 0s or 1s).

Setting the Message Transmission Method: The following two kinds of message transmission methods are available.

- Transmission order determined by message identifier priority
- Transmission order determined by mailbox number priority

Either of the message transmission methods can be selected with the message transmission method bit (MCR2) in the master control register (MCR): When messages are set to be transmitted according to the message identifier priority, if several messages are designated as waiting for transmission (TXPR = 1), the message with the highest priority in the message identifier is stored in the transmit buffer. CAN bus arbitration is then carried out for the message stored in the transmit buffer, and the message is transmitted when the transmission right is acquired. When the TXPR bit is set, the highest-priority message is found and stored in the transmit buffer.

When messages are set to be transmitted according to the mailbox number proiority, if several messages are designated as waiting for transmission (TXPR = 1), messages are stored in the transmit buffer in low-to-high mailbox order. CAN bus arbitration is then carried out for the message stored in the transmit buffer, and the message is transmitted when the transmission right is acquired.

11.4.3 Message Transmission

Messages are transmitted using mailboxes 1 to 15. The transmission procedure after initial settings is described below, and a transmission flowchart is shown in figure 11.9.

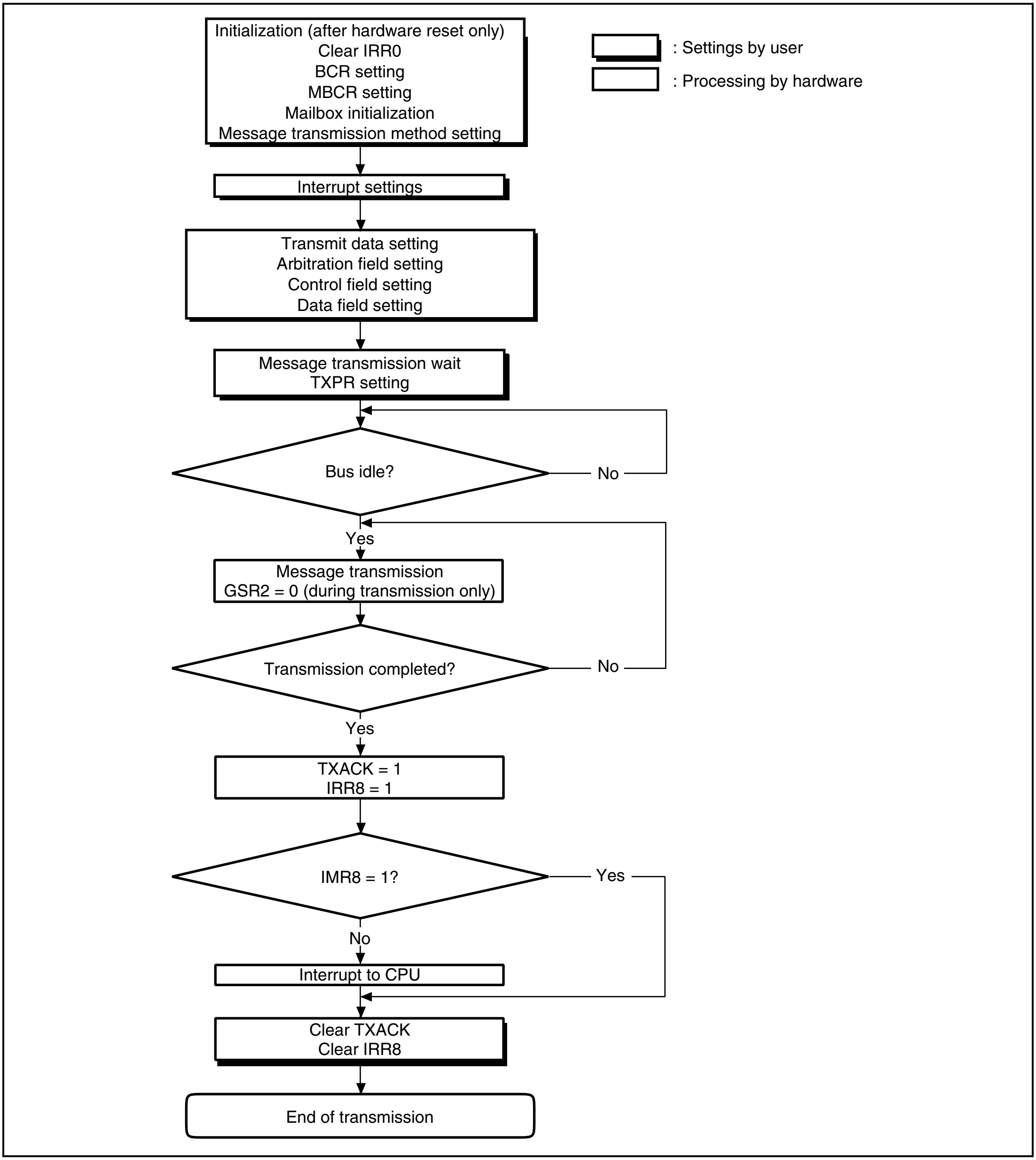


Figure 11.9 Transmission Flowchart

CPU interrupt source settings: The CPU interrupt source is set by the interrupt mask register (IMR) and mailbox interrupt mask register (MBIMR). Transmission acknowledge and transmission abort acknowledge interrupts can be generated for individual mailboxes in the mailbox interrupt mask register (MBIMR).

Arbitration field setting: The arbitration field is set by message control registers MCx[5]–MCx[8] in a transmit mailbox. For a standard format, an 11-bit identifier (ID-28 to ID-18) and the RTR bit are set, and the IDE bit is cleared to 0. For an extended format, a 29-bit identifier (ID-28 to ID-0) and the RTR bit are set, and the IDE bit is set to 1.

Control field setting: In the control field, the byte length of the data to be transmitted is set within the range of zero to eight bytes. The register to be set is the message control register MCx[1] in a transmit mailbox.

Data field setting: In the data field, the data to be transmitted is set within the range zero to eight. The registers to be set are the message data registers MDx[1]–MDx[8]. The byte length of the data to be transmitted is determined by the data length code in the control field. Even if data exceeding the value set in the control field is set in the data field, up to the byte length set in the control field will actually be transmitted.

Message transmission: If the corresponding mailbox transmit wait bit (TXPR1–TXPR15) in the transmit wait register (TXPR) is set to 1 after message control and message data registers have been set, the message enters transmit wait state. If the message is transmitted error-free, the corresponding acknowledge bit (TXACK1–TXACK15) in the transmit acknowledge register (TXACK) is set to 1, and the corresponding transmit wait bit (TXPR1–TXPR15) in the transmit wait register (TXPR) is automatically cleared to 0. Also, if the corresponding bit (MBIMR1-MBIMR15) in the mailbox interrupt mask register (MBIMR) and the mailbox empty interrupt bit (IRR8) in the interrupt mask register (IMR) are both simultaneously set to enable interrupts, interrupts may be sent to the CPU.

If transmission of a transmit message is aborted in the following cases, the message is retransmitted automatically:

- CAN bus arbitration failure (failure to acquire the bus)
- Error during transmission (bit error, stuff error, CRC error, frame error, or ACK error)

Message transmission cancellation: Transmission cancellation can be specified for a message stored in a mailbox as a transmit wait message. A transmit wait message is canceled by setting the bit for the corresponding mailbox (TXCR1–TXCR15) to 1 in the transmit cancel register (TXCR). Clearing the transmit wait register (TXPR) does not cancel transmission. When cancellation is executed, the transmit wait register (TXPR) is automatically reset, and the corresponding bit is set to 1 in the abort acknowledge register (ABACK). An interrupt to the CPU can be requested, and if the mailbox empty interrupt (IRR8) is enabled for the bits (MBIMR1-MBIMR15) corresponding to the mailbox interrupt mask register (MBIMR) and interrupt mask register (IMR), interrupts may be sent to the CPU.

However, a transmit wait message cannot be canceled at the following times:

- During internal arbitration or CAN bus arbitration
- During data frame or remote frame transmission

Figure 11.10 shows a flowchart for transmit message cancellation.

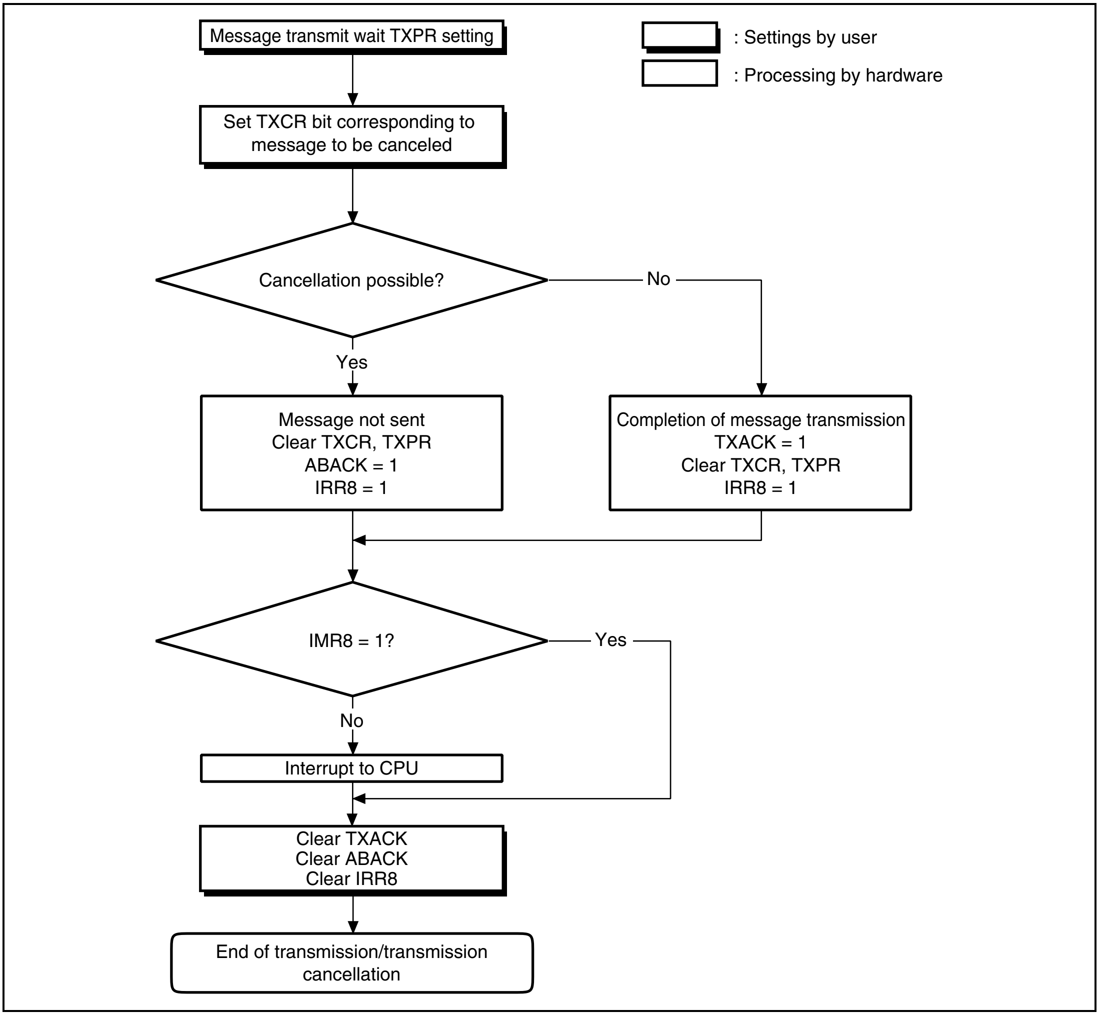


Figure 11.10 Transmit Message Cancellation Flowchart

11.4.4 Message Reception

The reception procedure after initial settings is described below. A reception flowchart is shown in figure 11.11.

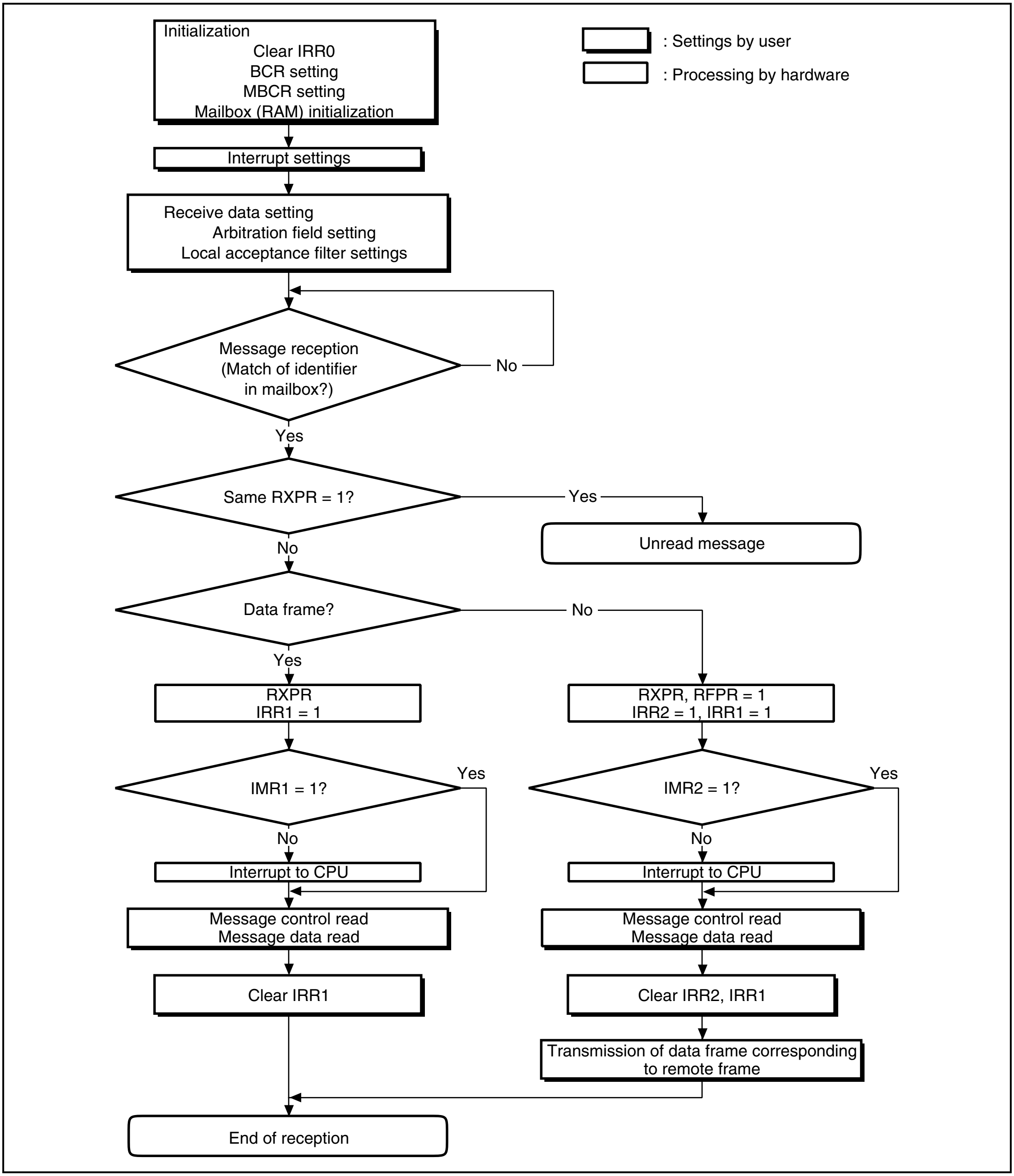


Figure 11.11 Reception Flowchart

CPU interrupt source settings: CPU interrupt source settings are made in the interrupt mask register (IMR) and mailbox interrupt register (MBIMR). The message to be received is also specified. Data frame and remote frame receive wait interrupt requests can be generated for individual mailboxes in the MBIMR.

Arbitration field setting: To receive a message, the message identifier must be set in advance in the message control registers (MCx[1]–MCx[8]) for the receiving mailbox. When a message is received, all the bits in the receive message identifier are compared with those in each message control register identifier, and if a 100% match is found, the message is stored in the matching mailbox. Mailbox 0 has a local acceptance filter mask (LAFM) that allows Don't Care settings to be made. The LAFM setting can be made only for mailbox 0. By making the Don't Care setting for all the bits in the receive message identifier, messages of multiple identifiers can be received.

Examples:

- When the identifier of mailbox 1 is 010_1010_1010 (standard format), only one kind of message identifier can be received by mailbox 1:

  Identifier 1: 010_1010_1010
- When the identifier of mailbox 0 is 010_1010_1010 (standard format) and the LAFM setting is 000_0000_0011 (0: Care, 1: Don't Care), a total of four kinds of message identifiers can be received by mailbox 0:

  Identifier 1: 010_1010_1000

  Identifier 2: 010_1010_1001

  Identifier 3: 010_1010_1010

  Identifier 4: 010_1010_1011

Message reception: When a message is received, a CRC check is performed automatically. If the result of the CRC check is normal, ACK is transmitted in the ACK field irrespective of whether the message can be received or not.

- Data frame reception

  If the received message is confirmed to be error-free by the CRC check, the identifier in the mailbox (and also LAFM in the case of mailbox 0 only) and the identifier of the receive message, are compared. If a complete match is found, the message is stored in the mailbox. The message identifier comparison is carried out on each mailbox in turn, starting with mailbox 0 and ending with mailbox 15. If a complete match is found, the comparison ends at that point, the message is stored in the matching mailbox, and the corresponding receive complete bit (RXPR0–RXPR15) is set in the receive complete register (RXPR). However, when a mailbox 0 LAFM comparison is carried out, even if the identifier matches, the mailbox comparison sequence does not end at that point, but continues with mailbox 1 and then the remaining mailboxes. It is therefore possible for a message matching mailbox 0 to be received by another mailbox. Note that the same message cannot be stored in more than one of mailboxes 1 to 15. On receiving a message, a CPU interrupt request may be generated depending on the mailbox interrupt mask register (MBIMR) and interrupt mask register (IMR) settings.

- Remote frame reception

  Two kinds of messages—data frames and remote frames—can be stored in mailboxes. A remote frame differs from a data frame in that the remote transmission request bit (RTR) in the message control register and the data field is 0 bytes long. The data length to be returned in a data frame must be stored in the data length code (DLC) in the control field.

  When a remote frame (RTR = recessive) is received, the corresponding bit is set in the remote request wait register (RFPR). If the corresponding bit (MBIMR0–MBIMR15) in the mailbox interrupt mask register (MBIMR) and the remote frame request interrupt mask (IRR2) in the interrupt mask register (IMR) are set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

Unread message overwrite: If the receive message identifier matches the mailbox identifier, the receive message is stored in the mailbox regardless of whether the mailbox contains an unread message or not. If a message overwrite occurs, the corresponding bit (UMSR0–UMSR15) is set in the unread message register (UMSR). In overwriting an unread message, when a new message is received before the corresponding bit in the receive complete register (RXPR) has been cleared, the unread message register (UMSR) is set. If the unread interrupt flag (IRR9) in the interrupt mask register (IMR) is set to the interrupt enable value at this time, an interrupt can be sent to the CPU. Figure 11.12 shows a flowchart for unread message overwriting.

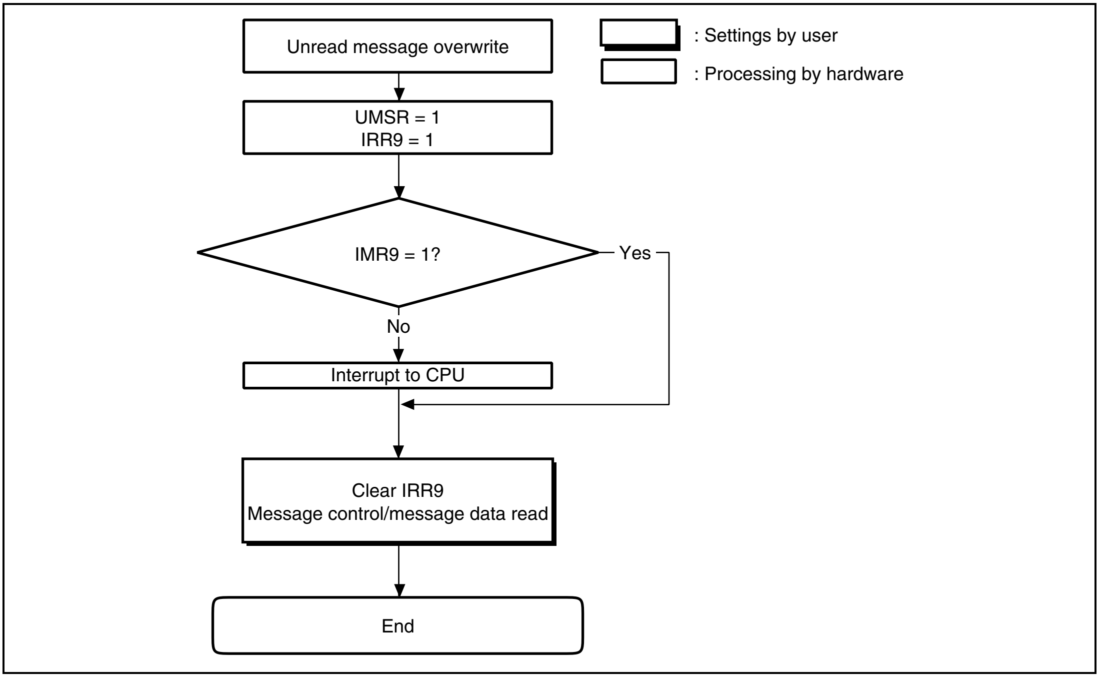


Figure 11.12 Unread Message Overwrite Flowchart

11.4.5 HCAN Sleep Mode

The HCAN is provided with an HCAN sleep mode that places the HCAN module in the sleep state in order to reduce current consumption. Figure 11.13 shows a flowchart of the HCAN sleep mode.

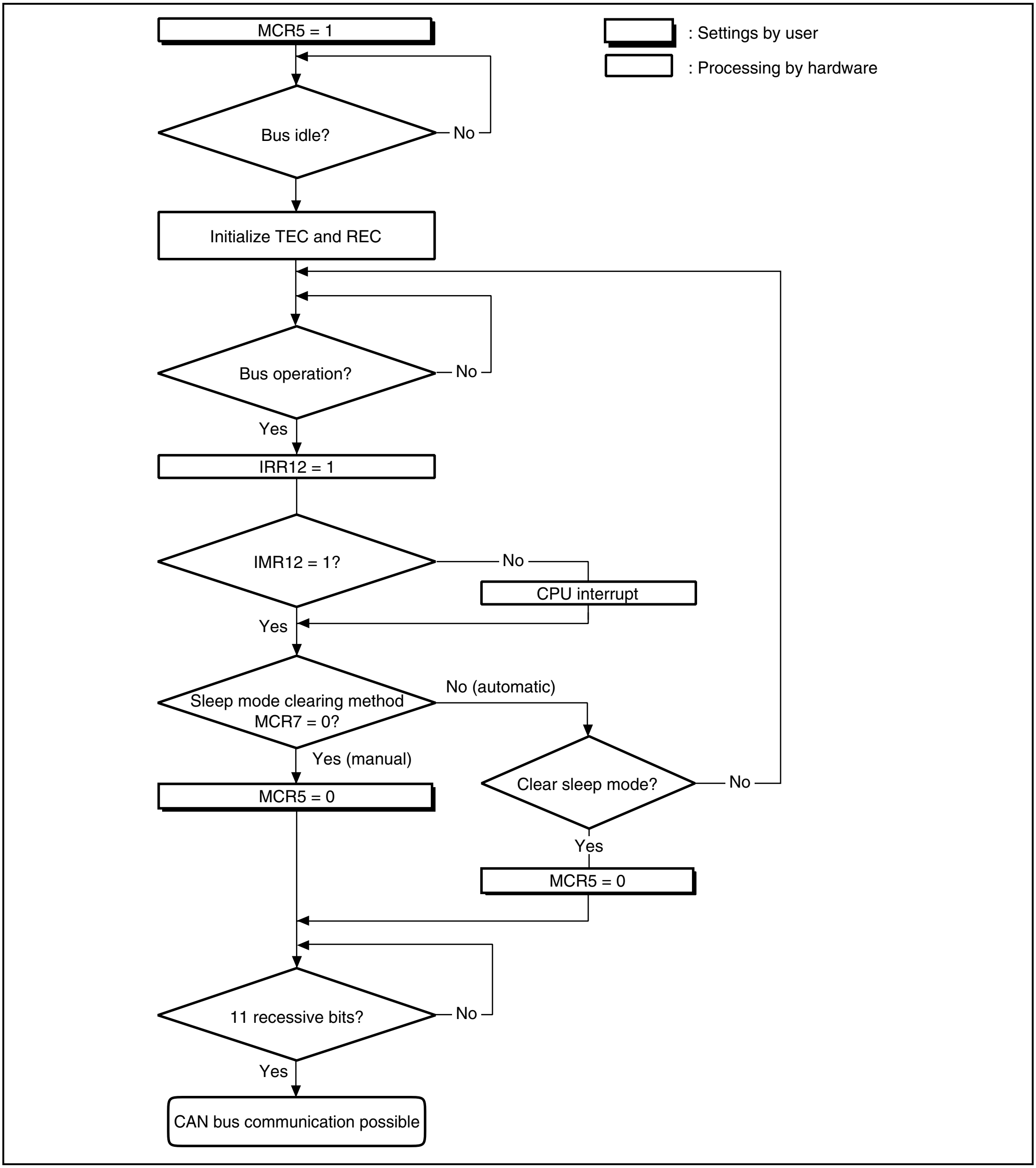


Figure 11.13 HCAN Sleep Mode Flowchart

HCAN sleep mode is entered by setting the HCAN sleep mode bit (MCR5) to 1 in the master control register (MCR). If the CAN bus is operating, the transition to HCAN sleep mode is delayed until the bus becomes idle.

Either of the following methods of clearing HCAN sleep mode can be selected:

- Clearing by software
- Clearing by CAN bus operation

Eleven recessive bits must be received after HCAN sleep mode is cleared before CAN bus communication is re-enabled.

Clearing by software: HCAN sleep mode is cleared by writing a 0 to MCR5 from the CPU.

Clearing by CAN bus operation: The cancellation method is selected by the MCR7 bit setting in MCR. Clearing by CAN bus operation occurs automatically when the CAN bus performs an operation and this change is detected. In this case, the first message is not stored in a mailbox; messages will be received normally from the second message onward. When a change is detected on the CAN bus in HCAN sleep mode, the bus operation interrupt flag (IRR12) is set in the interrupt register (IRR). If the bus interrupt mask (IMR12) in the interrupt mask register (IMR) is set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

11.4.6 HCAN Halt Mode

The HCAN halt mode is provided to enable mailbox settings to be changed without performing an HCAN hardware or software reset. Figure 11.14 shows a flowchart of the HCAN halt mode.

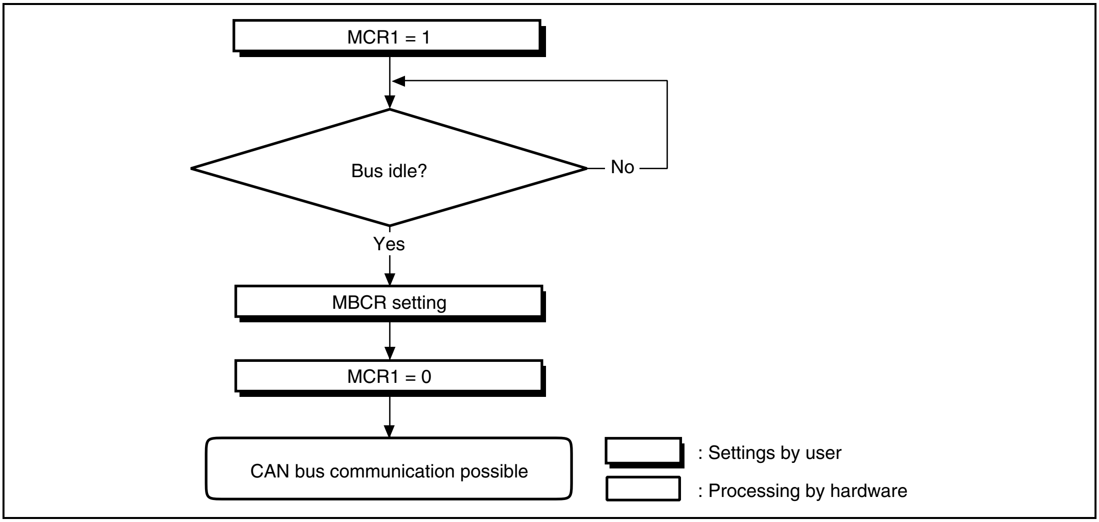


Figure 11.14 HCAN Halt Mode Flowchart

HCAN halt mode is entered by setting the halt request bit (MCR1) to 1 in the master control register (MCR). If the CAN bus is operating, the transition to HCAN halt mode is delayed until the bus becomes idle.

HCAN halt mode is cleared by clearing MCR1 to 0.

11.5 Interrupt Sources

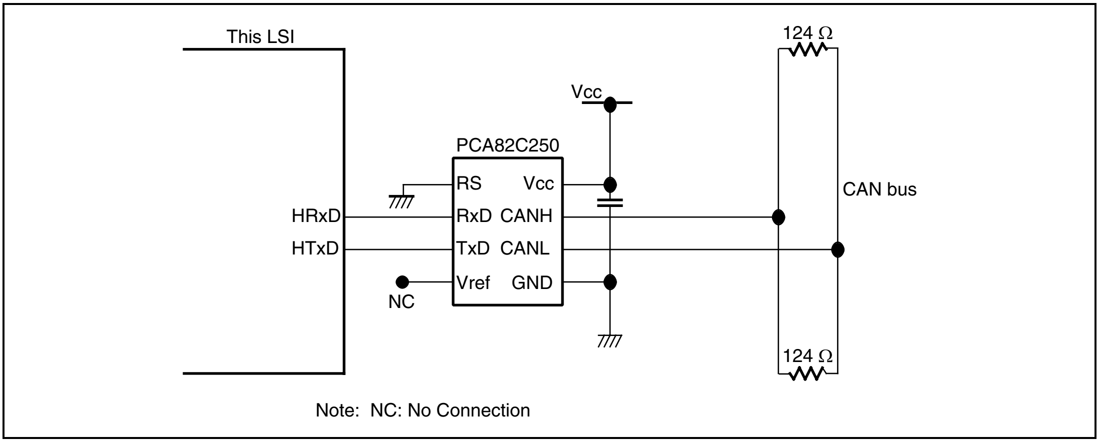
Table 11.4 lists the HCAN interrupt sources. With the exception of the reset processing vector (IRR0), these sources can be masked. Masking is implemented using the mailbox interrupt mask register (MBIMR), interrupt mask register (IMR), and IRQ enable register (IER). For details on the interrupt vector of each interrupt source, refer to section 5, Interrupt Controller.

Table 11.4 HCAN Interrupt Sources

Name	Description	Interrupt Flag
ERS0/OVR0	Error passive interrupt (TEC ≥ 128 or REC ≥ 128)	IRR5
	Bus off interrupt (TEC ≥ 256)	IRR6
	Reset process interrupt by power-on reset	IRR0
	Remote frame reception	IRR2
	Error warning interrupt (TEC ≥ 96)	IRR3
	Error warning interrupt (REC ≥ 96)	IRR4
	Overload frame transmission	IRR7
	Unread message overwrite	IRR9
	Detection of CAN bus operation in HCAN sleep mode	IRR12
RM0	Mailbox 0 message reception	IRR1
RM1	Mailbox 1-15 message reception	IRR1
SLE0	Message transmission/cancellation	IRR8
IRQ2	Generation of IRQ2 interrupt from HRxD input pin by setting RxDIE bit in HCANMON to 1	IRQ2F

11.6 CAN Bus Interface

A bus transceiver IC is necessary to connect this chip to a CAN bus. A Philips PCA82C250 transceiver IC is recommended. Any other product must be compatible with the PCA82C250. Figure 11.15 shows a sample connection diagram.



Note: NC: No Connection

Figure 11.15 High-Speed Interface Using PCA82C250

11.7 Usage Notes

11.7.1 Module Stop Mode Setting

HCAN operation can be disabled or enabled using the module stop control register. The initial setting is for HCAN operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 16, Power-Down Modes.

11.7.2 Reset

The HCAN is reset by a power-on reset, in hardware standby mode, and in software standby mode. All the registers are initialized in a reset, however mailboxes (message control (MCx[x])/message data (MDx[x])) are not. After power-on, mailboxes (message control (MCx[x])/message data (MDx[x])) are initialized, and their values are undefined. Therefore, mailbox initialization must always be carried out after a power-on reset, a transition to hardware standby mode, or software standby mode. The reset interrupt flag (IRR0) is always set after a power-on reset or recovery from software standby mode. As this bit cannot be masked in the interrupt mask register (IMR), if HCAN interrupt enabling is set in the interrupt controller without clearing the flag, an HCAN interrupt will be initiated immediately. IRR0 should therefore be cleared during initialization.

11.7.3 HCAN Sleep Mode

The bus operation interrupt flag (IRR12) in the interrupt register (IRR) is set by CAN bus operation in HCAN sleep mode. Therefore, this flag is not used by the HCAN to indicate sleep mode release. Note that the reset status bit (GSR3) in the general status register (GSR) is set in sleep mode.

11.7.4 Interrupts

When the mailbox interrupt mask register (MBIMR) is set, the interrupt register (IRR8, 2, 1) is not set by reception completion, transmission completion, or transmission cancellation for the set mailboxes.

11.7.5 Error Counters

In the case of error active and error passive, REC and TEC normally count up and down. In the bus-off state, 11-bit recessive sequences are counted (REC + 1) using REC. If REC reaches 96 during the count, IRR4 and GSR1 are set.

11.7.6 Register Access

Byte or word access can be used on all HCAN registers. Longword access cannot be used.

11.7.7 HCAN Medium-Speed Mode

In medium-speed mode, neither read nor write is possible for the HCAN registers.

11.7.8 Register Hold in Standby Modes

All HCAN registers are initialized in hardware standby mode and software standby mode.

11.7.9 Use on Bit Manipulation Instructions

Since the HCAN status flag is cleared by writing 1, do not use the bit manipulation instructions to clear the flag. To clear the flag, use the MOV instructions and write 1 only to the bit to be cleared.

11.7.10 HCAN TXCR Operation

1. When the transmit wait cancel register (TXCR) is used to cancel a transmit wait message in a transmit wait mailbox, the corresponding bit to TXCR and the transmit wait register (TXPR) may not be cleared even if transmission is canceled. This occurs when the following conditions are all satisfied.

- The HRxD pin is stacked to 1 because of a CAN bus error, etc.
- There is at least one mailbox waiting for transmission or being transmitted.
- The message transmission in a mailbox being transmitted is canceled by TXCR.

  If this occurs, transmission is canceled. However, since TXPR and TXCR states are indicated wrongly that a message is being cancelled, transmission cannot be restarted even if the stack state of the HRxD pin is canceled and the CAN bus recovers the normal state. If there are at least two transmission messages, a message which is not being transmitted is canceled and a message being transmitted retains its state.

To avoid this, one of the following countermeasures must be executed.

- Transmission must not be canceled by TXCR. When transmission is normally completed after the CAN bus has recovered, TXPR is cleared and the HCAN recovers the normal state.
- To cancel transmission, the corresponding bit to TXCR must be written to 1 continuously until the bit becomes 0. TXPR and TXCR are cleared and the HCAN recovers the normal state.

2. When the bus-off state is entered while TXPR is set and the transmit wait state is entered, the internal state machine does not operate even if TXCR is set during the bus-off state. Therefore transmission cannot be canceled. The message can be canceled when one message is transmitted or a transmission error occurs after the bus-off state is recovered. To clear a message after the bus-off state is recovered, the following countermeasure must be executed.

- A transmit wait message must be cleared by resetting the HCAN during the bus-off period.

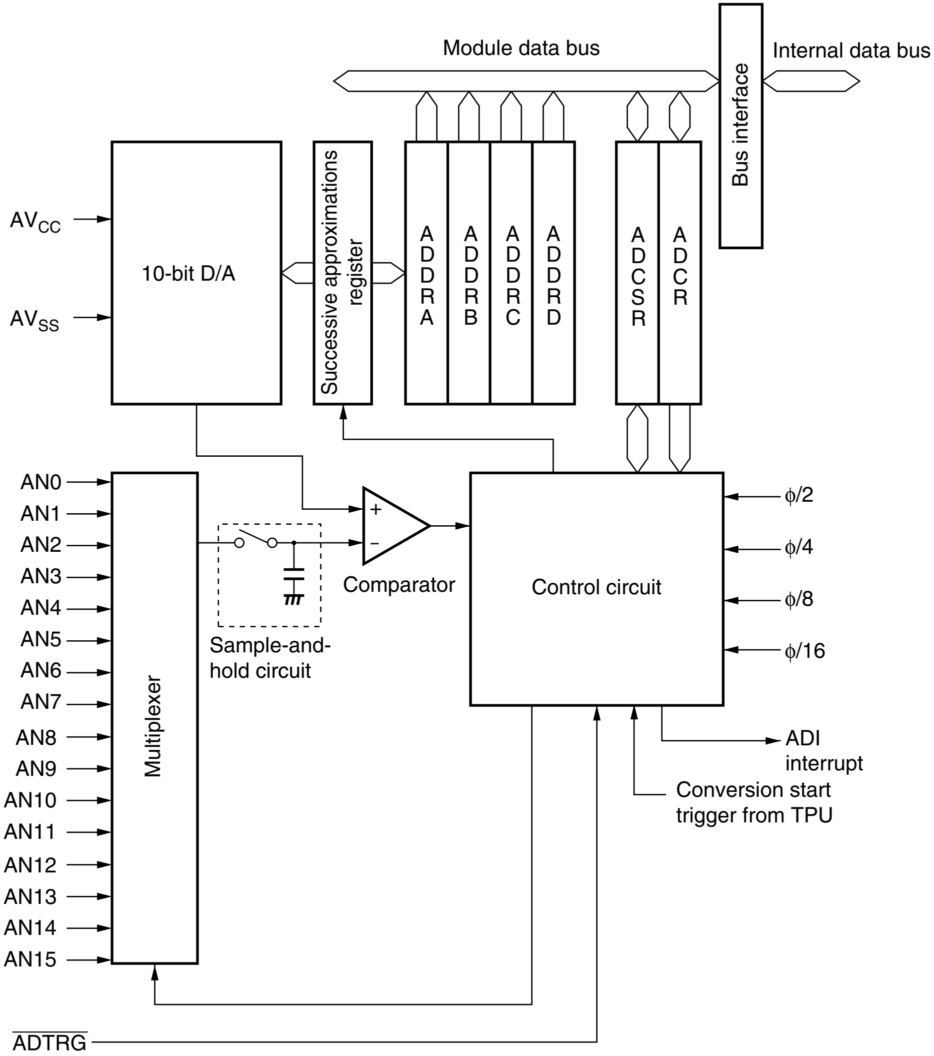
  To reset the HCAN, the module stop bit (MSTPC3 in MSTPCRC) must be set or cleared. In this case, the HCAN is entirely reset. Therefore the initial settings must be made again.

Section 12 A/D Converter

This LSI includes a successive approximation type 10-bit A/D converter that allows up to 16 analog input channels to be selected. The block diagram of the A/D converter is shown in figure 12.1.

12.1 Features

- 10-bit resolution
- 16 input channels
- Conversion time: 11.08 µs per channel (at 24 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



Legend

ADCR : A/D control register

ADCSR : A/D control/status register

ADDRA : A/D data register A

ADDRB : A/D data register B

ADDRC : A/D data register C

ADDRD : A/D data register D

Figure 12.1 Block Diagram of A/D Converter

12.2 Input/Output Pins

Table 12.1 summarizes the input pins used by the A/D converter. The 16 analog input pins are divided into four channel sets and four 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, analog input pins 8 to 11 (AN8 to AN11) comprising group 2, and analog input pins 12 to 15 (AN12 to AN15) comprising group 3. The AVcc and AVss pins are the power supply pins for the analog block in the A/D converter.

Table 12.1 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
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	
Analog input pin 12	AN12	Input	Group 3 analog input pins
Analog input pin 13	AN13	Input	
Analog input pin 14	AN14	Input	
Analog input pin 15	AN15	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input pin for starting A/D conversion

12.3 Register Descriptions

The A/D converter has the following registers. The MSTPA1 bit in the module stop control register A (MSTPCRA) specifies the modes of this module as module stop mode. For details on MSTPCRA, refer to section 16.1.3, Module Stop Control Registers A to C (MSTPCRA to MSTPCRC).

- 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)

12.3.1 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 12.2.

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 12.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel				A/D Data Register to be Stored Results of A/D Conversion
CH3 = 0		CH3 = 1		
Group 0 (CH2 = 0)	Group 1 (CH2 = 1)	Group 2 (CH2 = 0)	Group 3 (CH2 = 1)	
AN0	AN4	AN8	AN12	ADDRA
AN1	AN5	AN9	AN13	ADDRB
AN2	AN6	AN10	AN14	ADDRC
AN3	AN7	AN11	AN15	ADDRD

12.3.2 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 condition] • When 0 is written after reading ADF = 1
6	ADIE	0	R/W	A/D Interrupt Enable A/D conversion end interrupt (ADI) request enabled when 1 is set
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

Bit	Bit Name	Initial Value	R/W	Description	
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, AN1
				0010: AN2	0010: AN0 to AN2
				0011: AN3	0011: AN0 to AN3
				0100: AN4	0100: AN4
				0101: AN5	0101: AN4, AN5
				0110: AN6	0110: AN4 to AN6
				0111: AN7	0111: AN4 to AN7
				1000: AN8	1000: AN8
				1001: AN9	1001: AN8, AN9
				1010: AN10	1010: AN8 to AN10
				1011: AN11	1011: AN8 to AN11
				1100: AN12	1100: AN12
				1101: AN13	1101: AN12, AN13
				1110: AN14	1110: AN12 to AN14
				1111: AN15	1111: AN12 to AN15

12.3.3 A/D Control Register (ADCR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 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	—	1	—	Reserved
4	—	1	—	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 18.7 in section 18, Electrical Characteristics. 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	—	1	—	Reserved
0	—	1	—	These bits are always read as 1.

12.4 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.

12.4.1 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.

12.4.2 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.

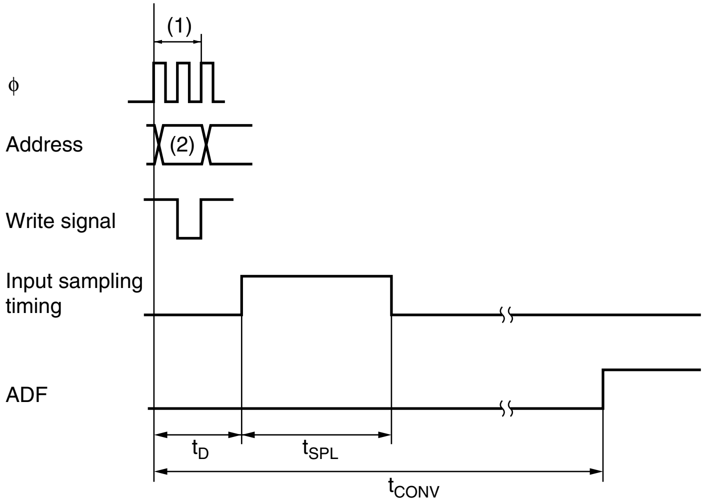
1. When the ADST bit is set to 1 by software, TPU or external trigger input, A/D conversion starts on the first channel in the group (AN0 when CH3 and CH2 = 00, AN4 when CH3 and CH2 = 01, AN8 when CH3 and CH2 = 10, or AN12 when CH3 and CH2 = 11).
2. When A/D conversion for each channel is completed, 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.

12.4.3 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 12.2 shows the A/D conversion timing. Table 12.3 shows the A/D conversion time.

As indicated in figure 12.2, 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 12.3.

In scan mode, the values given in table 12.3 apply to the first conversion time. The values given in table 12.4 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 18.7 in section 18, Electrical Characteristics.



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 12.2 A/D Conversion Timing

Table 12.3 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 12.4 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)

12.4.4 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 12.3 shows the timing.

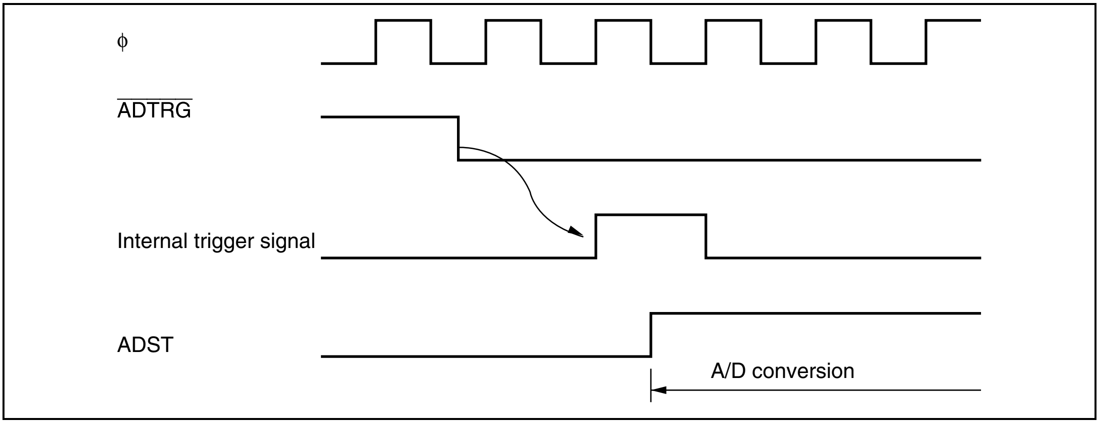


Figure 12.3 External Trigger Input Timing

12.5 Interrupt Sources

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.

Table 12.5 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Source Flag
ADI	A/D conversion completed	ADF

12.6 A/D Conversion Accuracy Definitions

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital output codes
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 12.4).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value B'0000000000 (H'000) to B'0000000001 (H'001) (see figure 12.5).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from B'1111111110 (H'3FE) to B'1111111111 (H'3FF) (see figure 12.5).
- 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 12.5).
- Absolute accuracy

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

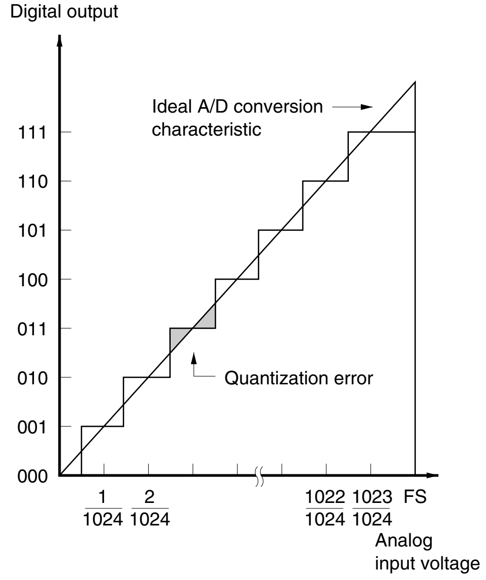


Figure 12.4 A/D Conversion Accuracy Definitions

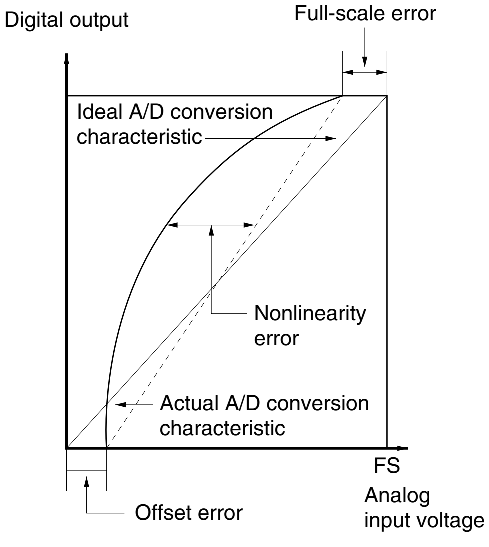


Figure 12.5 A/D Conversion Accuracy Definitions

12.7 Usage Notes

12.7.1 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 16, Power-Down Modes.

12.7.2 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion accuracy is guaranteed for an input signal for which the signal source impedance is 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 accuracy. 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 12.6). When converting a high-speed analog signal, a low-impedance buffer should be inserted.

12.7.3 Influences on Absolute Accuracy

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

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board (i.e, acting as antennas).

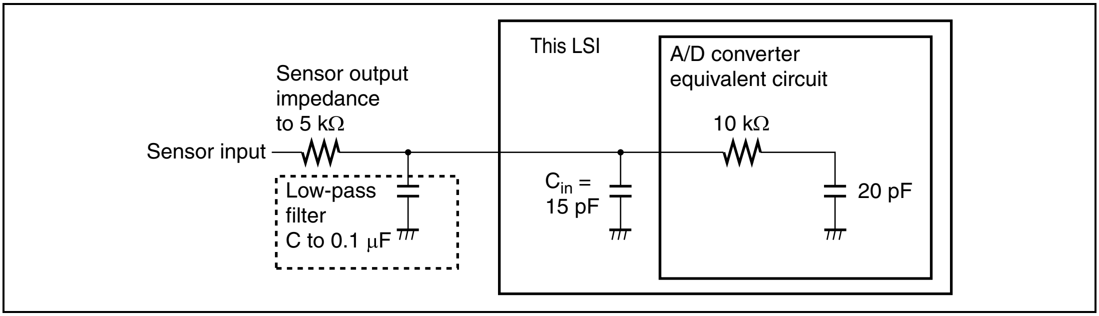

This LSI
A/D converter equivalent circuit
Sensor output impedance to 5 kΩ
Sensor input
10 kΩ
Low-pass filter C to 0.1 µF
C_{in} = 15 pF
20 pF

Figure 12.6 Example of Analog Input Circuit

12.7.4 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 ≤ ANn ≤ 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.

12.7.5 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 AN15), 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.

12.7.6 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 AN15), between AVcc and AVss, as shown in figure 12.7. Also, the bypass capacitors connected to AVcc and the filter capacitor connected to AN0 to AN15 must be connected to AVss.

If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN15) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance (R_{in}), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding circuit constants.

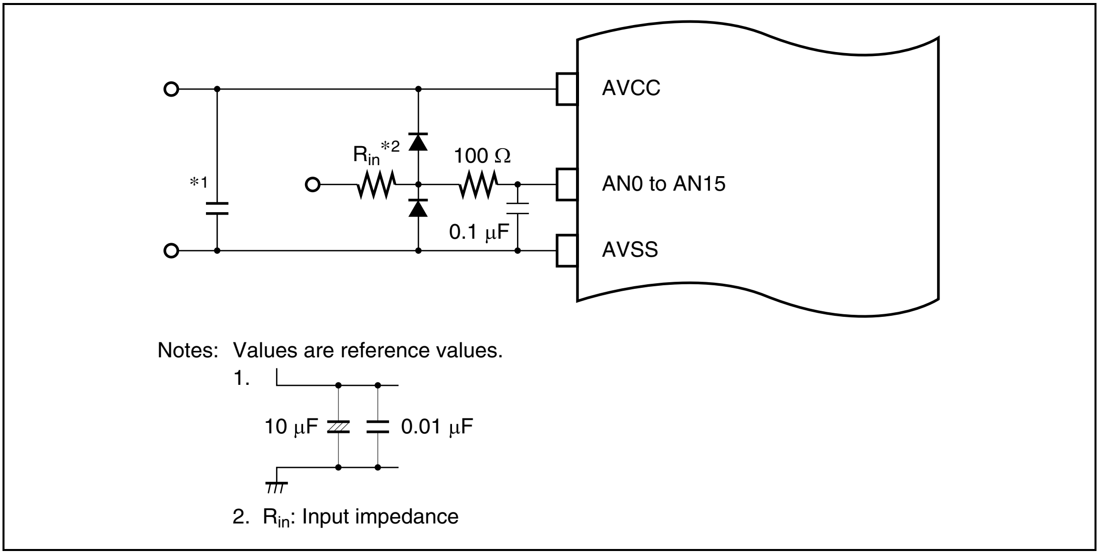


Figure 12.7 Example of Analog Input Protection Circuit

Table 12.6 Analog Pin Specifications

Item	Min.	Max.	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	5	kΩ

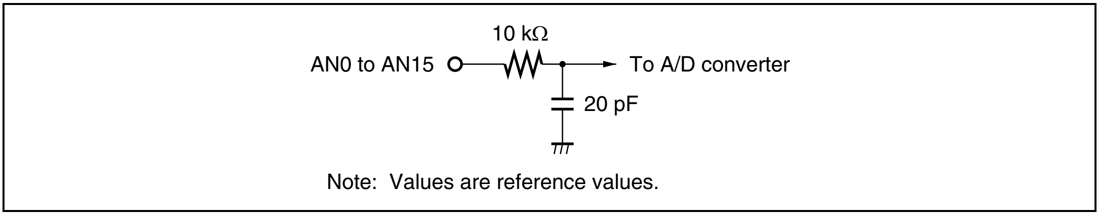


Figure 12.8 Analog Input Pin Equivalent Circuit

Section 13 RAM

This LSI has 4 kbytes of on-chip high-speed static RAM. The RAM is connected to the CPU by a 16-bit data bus, enabling one-state access by the CPU to both byte data and word data.

The on-chip RAM can be enabled or disabled by means of the RAME bit in the system control register (SYSCR). For details on SYSCR, refer to section 3.2.2, System Control Register (SYSCR).

Section 14 ROM

The features of the flash memory are summarized below.

The block diagram of the flash memory is shown in figure 14.1.

14.1 Features

- Size: 64 kbytes
- Programming/erase methods
  - — The flash memory is programmed 128 bytes at a time. Erase is performed in single-block units. The flash memory is configured as follows: 28 kbytes × 1 block, 16 kbytes × 1 block, 8 kbytes × 2 blocks, and 1 kbyte × 4 blocks. To erase the entire flash memory, each block must be erased in turn.
- Reprogramming capability
  - — The flash memory can be reprogrammed up to 100 times.
- Three programming modes
  - — Boot mode
  - — User mode
  - — Programmer 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 of the entire flash memory. In normal user program mode, individual blocks can be erased or programmed.
- Programmer mode
  - — Flash memory can be programmed/erased in programmer mode using a PROM programmer, as well as in on-board programming mode.
- Automatic bit rate adjustment
  - — For data transfer in boot mode, this LSI's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Programming/erasing protection
  - — Sets software protection against flash memory programming/erasing.

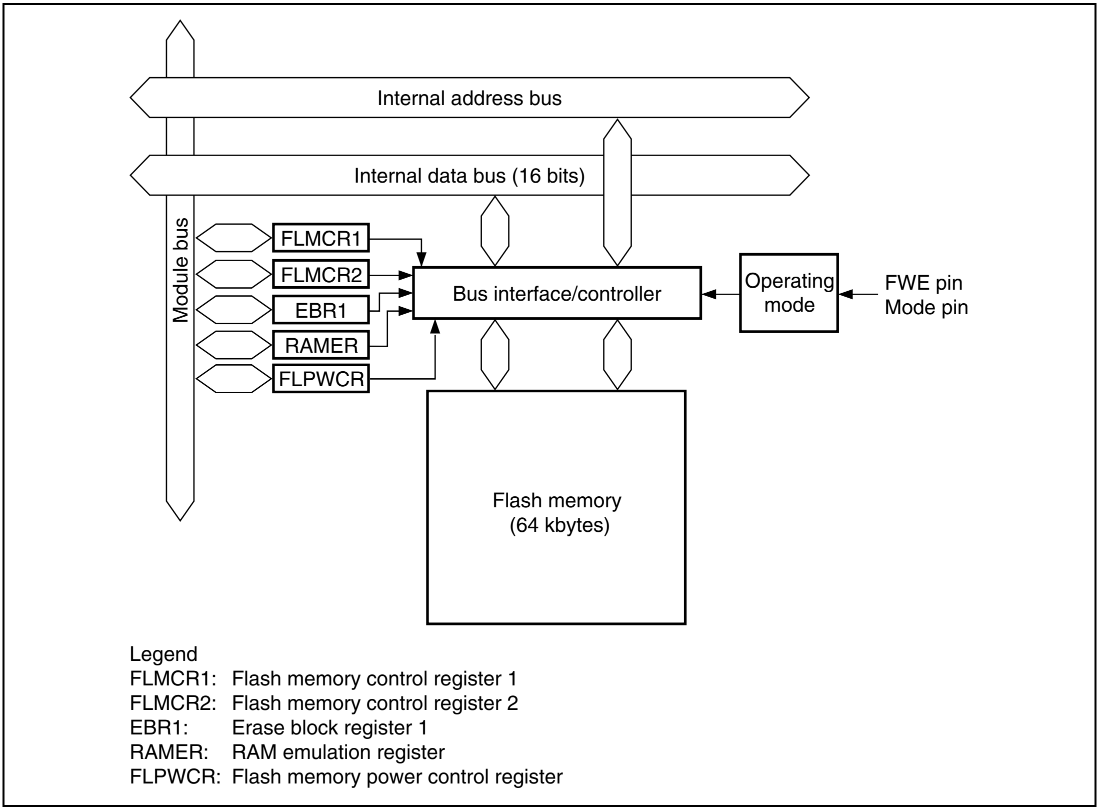

Internal address bus
Internal data bus (16 bits)
Module bus
FLMCR1
FLMCR2
EBR1
RAMER
FLPWCR
Bus interface/controller
Operating mode
FWE pin
Mode pin
Flash memory (64 kbytes)
Legend
FLMCR1: Flash memory control register 1
FLMCR2: Flash memory control register 2
EBR1: Erase block register 1
RAMER: RAM emulation register
FLPWCR: Flash memory power control register

Figure 14.1 Block Diagram of Flash Memory

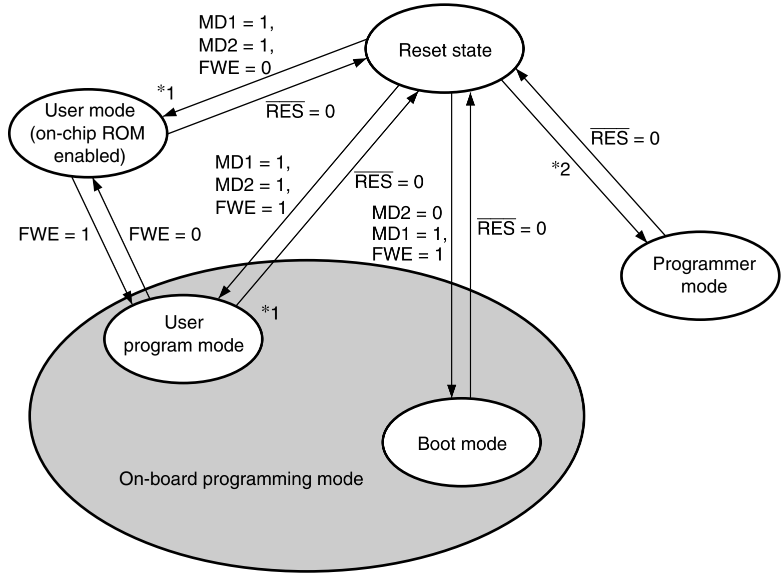
14.2 Mode Transitions

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 14.2. 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 write and erase the flash memory.

The differences between boot mode and user program mode are shown in table 14.1.

Figure 14.3 shows the operation flow for boot mode and figure 14.4 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. This LSI transits to programmer mode by using the dedicated PROM programmer.

Figure 14.2 Flash Memory State Transitions

Table 14.1 Differences between Boot Mode and User Program Mode

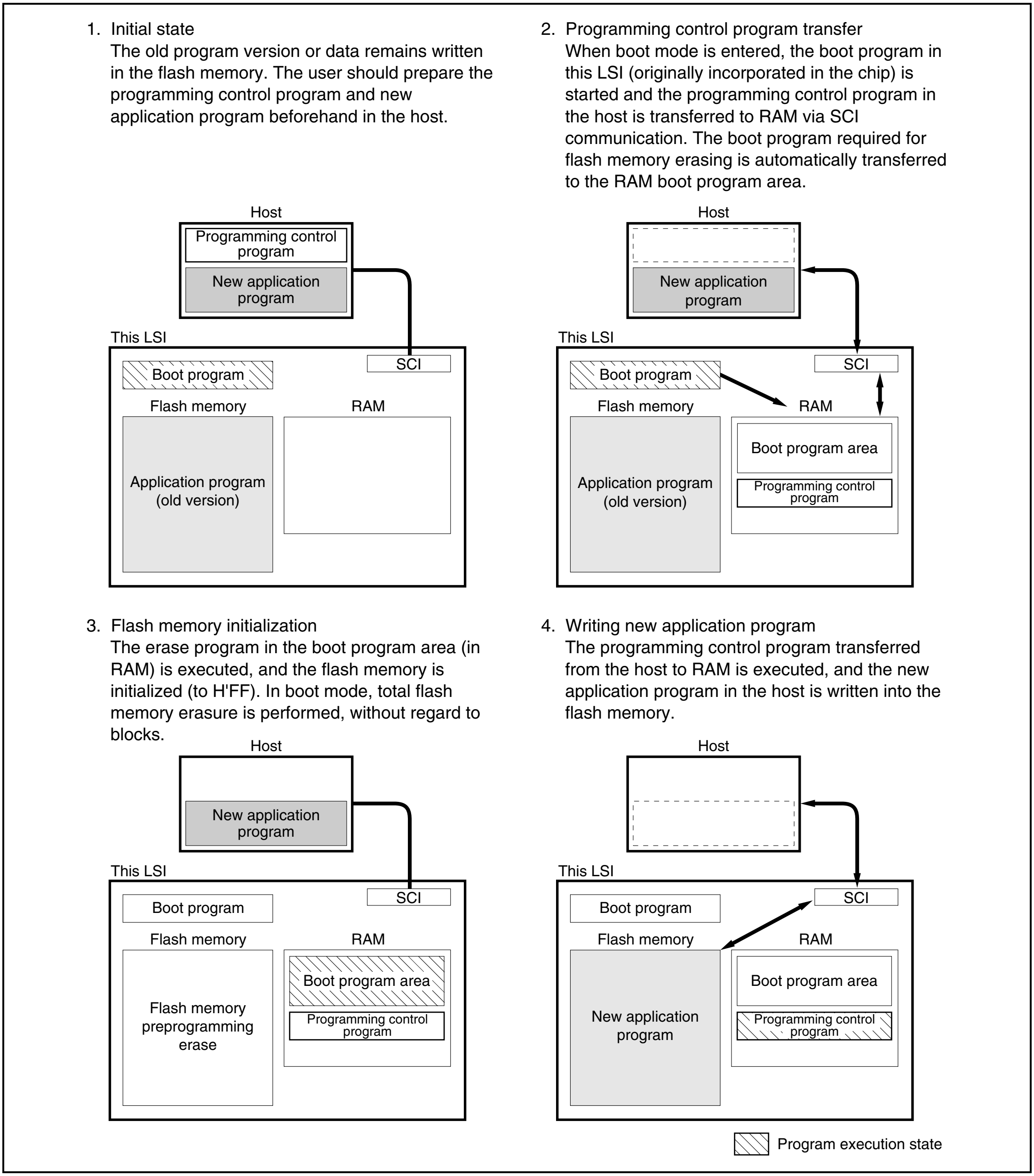
	Boot Mode	User Program Mode
Total erase	Yes	Yes
Block erase	No	Yes
Programming control program*	(2)	(1) (2) (3)

(1) Erase/erase-verify

(2) Program/program-verify

(3) 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.

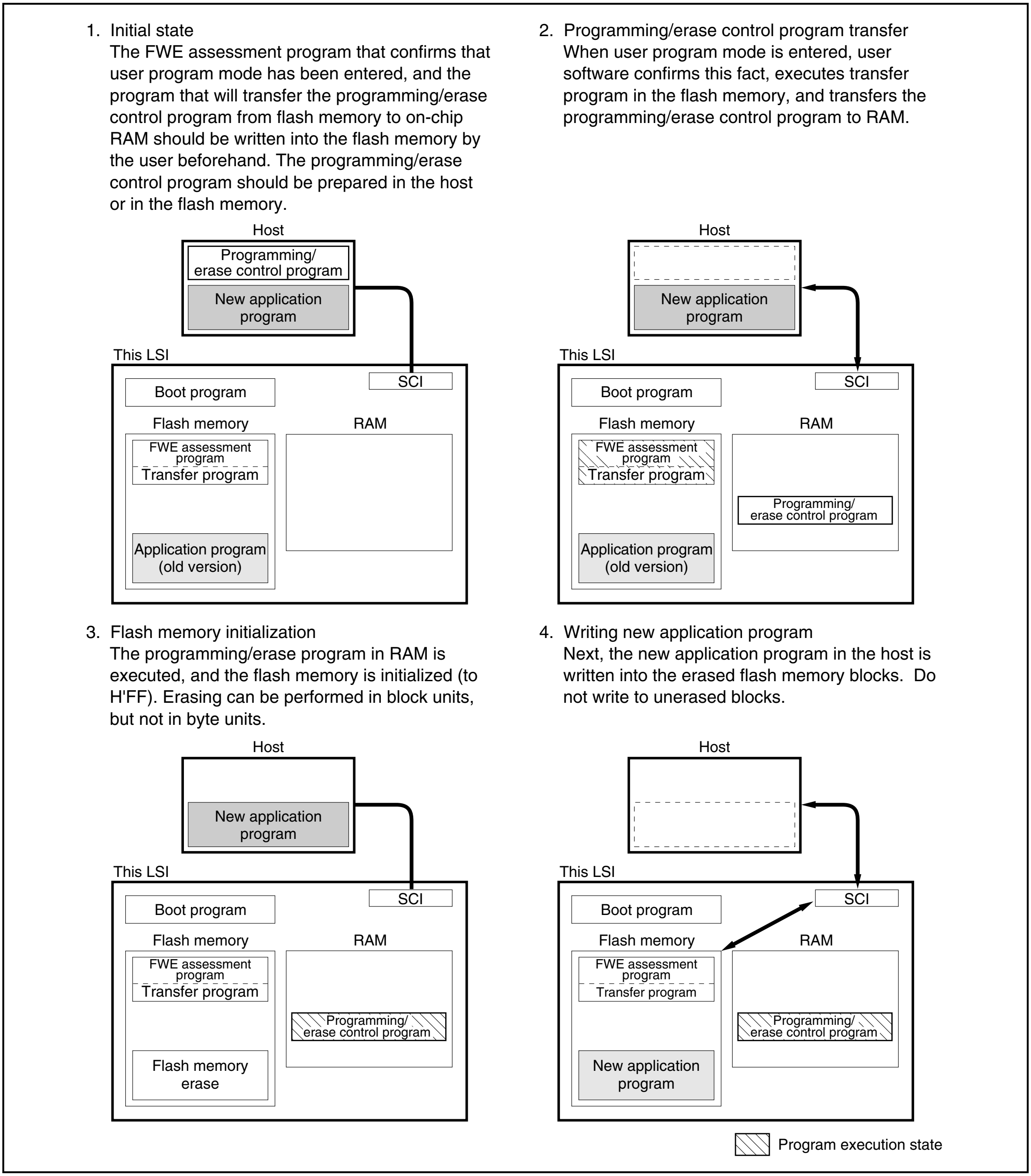
2. Programming control program transfer
   When boot mode is entered, the boot program in this LSI (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically 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.

Host
Programming control program
New application program
This LSI
SCI
Boot program
Flash memory
RAM
Application program (old version)
Boot program area
Programming control program
Flash memory preprogramming erase
New application program
▨ Program execution state

Figure 14.3 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 14.4   User Program Mode

14.3 Block Configuration

Figure 14.5 shows the block configuration of 64-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 28 kbytes (1 block), 16 kbytes (1 block), 8 kbytes (2 blocks), and 1 kbyte (4 blocks). Erasing is performed in these units. Programming is performed in 128-byte units starting from an address with lower eight bits H'00 or H'80.

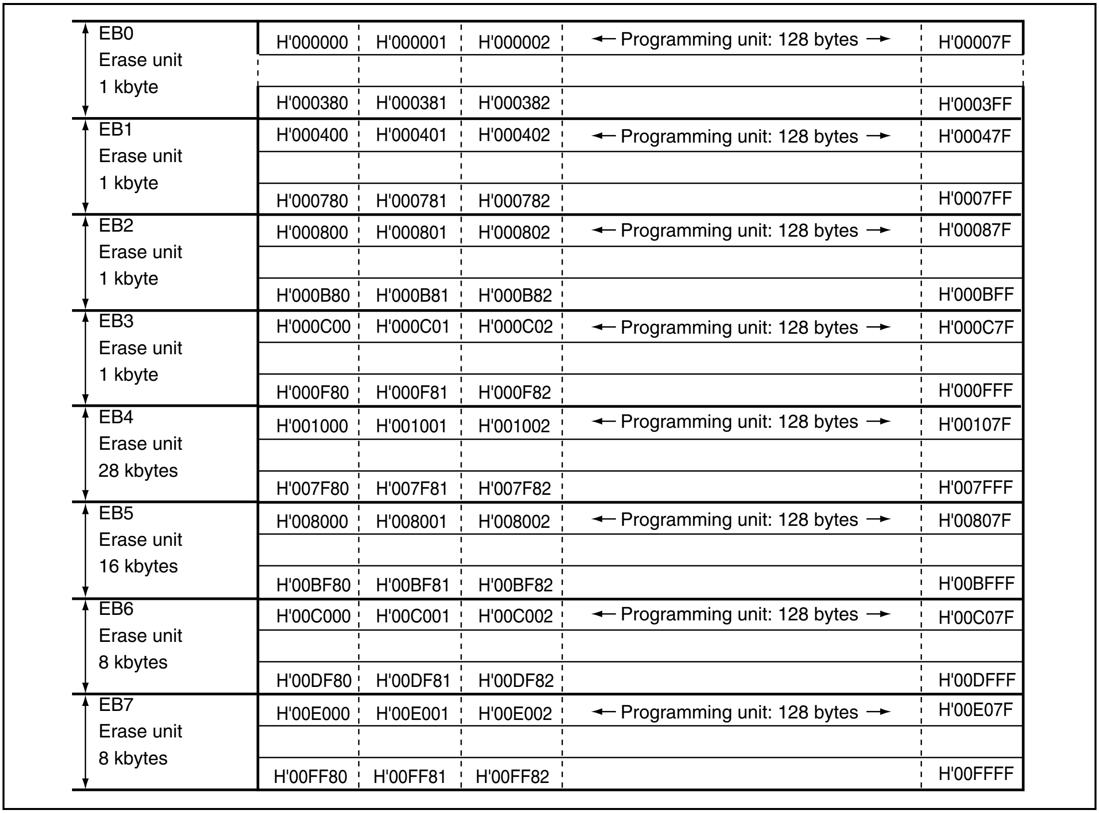

EB0 Erase unit 1 kbyte
H'000000 H'000001 H'000002 ← Programming unit: 128 bytes → H'00007F
H'000380 H'000381 H'000382 H'0003FF
EB1 Erase unit 1 kbyte
H'000400 H'000401 H'000402 ← Programming unit: 128 bytes → H'00047F
H'000780 H'000781 H'000782 H'0007FF
EB2 Erase unit 1 kbyte
H'000800 H'000801 H'000802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

Figure 14.5 Flash Memory Block Configuration

14.4 Input/Output Pins

The flash memory is controlled by means of the pins shown in table 14.2.

Table 14.2 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
TxD2	Output	Serial transmit data output
RxD2	Input	Serial receive data input

14.5 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)
- RAM emulation register (RAMER)
- Flash memory power control register (FLPWCR)

14.5.1 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, refer to section 14.8, Flash Memory Programming/Erasing.

Bit	Bit Name	Initial Value	R/W	Description
7	FWE	—	R	Reflects the input level at the FWE pin. It is cleared to 0 when a low level is input to the FWE pin, and set to 1 when a high level is input.
6	SWE	0	R/W	Software Write Enable Bit When this bit is set to 1, flash memory programming/erasing is enabled. When this bit is cleared to 0, other FLMCR1 register bits and all EBR1 bits cannot be set.
5	ESU1	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.
4	PSU1	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 P1 bit in FLMCR1.
3	EV1	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	PV1	0	R/W	Program-Verify When this bit is set to 1, the flash memory changes to program-verify mode. When it is cleared to 0, program-verify mode is cancelled.
1	E1	0	R/W	Erase When this bit is set to 1, and while the SWE1 and ESU1 bits are 1, the flash memory changes to erase mode. When it is cleared to 0, erase mode is cancelled.
0	P1	0	R/W	Program When this bit is set to 1, and while the SWE1 and PSU1 bits are 1, the flash memory changes to program mode. When it is cleared to 0, program mode is cancelled.

14.5.2 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	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 14.9.3, Error Protection, for details.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

14.5.3 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 to be automatically cleared to 0.

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

14.5.4 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	—	Reserved These bits are always read as 0.
5, 4	—	All 0	R/W	Reserved Only 0 should be written to these bits.
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 are selected to overlap the RAM area of H'FFE000 to H'FFE3FF. The areas correspond with 1-kbyte erase blocks.
0	RAM0	0	R/W	00X: H'000000 to H'0003FF (EB0) 01X: H'000400 to H'0007FF (EB1) 10X: H'000800 to H'000BFF (EB2) 11X: H'000C00 to H'000FFF (EB3) Note: X: Don't care

14.5.5 Flash Memory Power Control Register (FLPWCR)

FLPWCR enables or disables a transition to the flash memory power-down mode when this LSI switches to subactive mode.

Bit	Bit Name	Initial Value	R/W	Description
7	PDWND	0	R/W	When this bit is set to 1, the transition to flash memory power-down mode is disabled.
6 to 0	—	All 0	R	Reserved These bits are always read as 0.

14.6 On-Board Programming Modes

There are two modes for programming/erasing of the flash memory; boot mode, which enables on-board programming/erasing, and programmer mode, in which programming/erasing is performed with a PROM programmer. On-board programming/erasing can also be performed in user program mode. At reset-start in reset mode, this LSI changes to a mode depending on the MD pin settings and FWE pin setting, as shown in table 14.3. The input level of each pin must be defined four states before the reset ends.

When changing to boot mode, the boot program built into this LSI is initiated. The boot program transfers the programming control program from the externally-connected host to on-chip RAM via SCI_2. After erasing the entire flash memory, the programming control program is executed. This can be used for programming initial values in the on-board state or for a forcible return when programming/erasing can no longer be done in user program mode. In user program mode, individual blocks can be erased and programmed by branching to the user program/erase control program prepared by the user.

Table 14.3 Setting On-Board Programming Modes

MD2	MD1	MD0	FWE	LSI State after Reset End
1	1	1	1	User Mode
0	1	1	1	Boot Mode

14.6.1 Boot Mode

Table 14.4 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 14.8, Flash Memory Programming/Erasing.
2. SCI_2 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_2 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 14.5.
5. In boot mode, a part of the on-chip RAM area is used by the boot program. The area H'FFE000 to H'FFE7FF 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_2 (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 a reset. End the reset after driving the reset pin low, waiting at least 20 states, and then setting the mode (MD) pins. Boot mode is also cleared when a WDT overflow occurs.
8. Do not change the MD pin input levels in boot mode.
9. All interrupts are disabled during programming or erasing of the flash memory.

Table 14.4 Boot Mode Operation

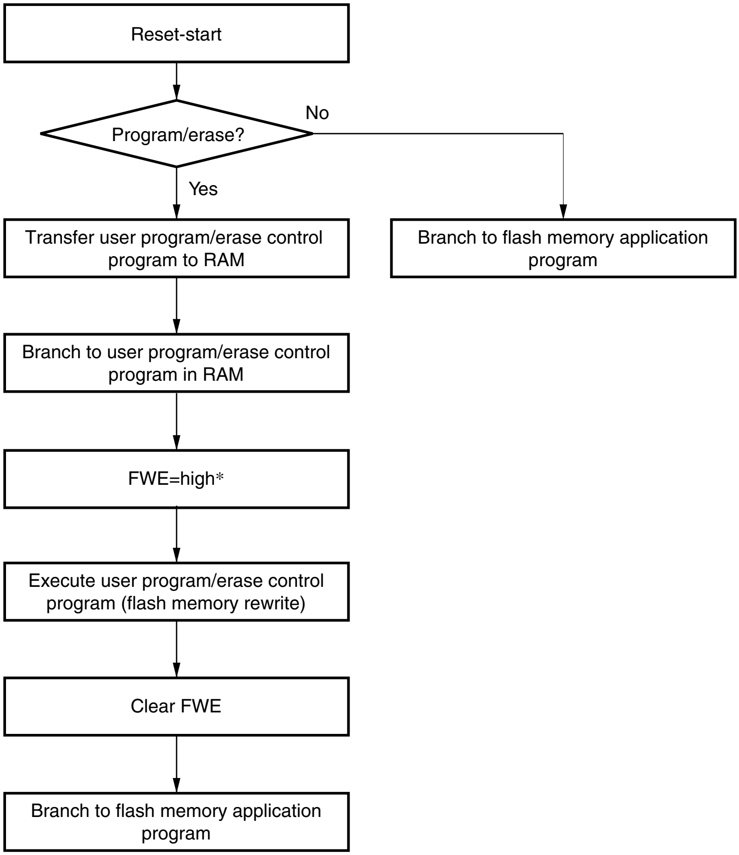
	Host Operation		LSI Operation
Item	Processing Contents	Communications Contents	Processing Contents
Boot mode start			Branches to boot program at reset-start. Boot program initiation
Bit rate adjustment	Continuously transmits data H'00 at specified bit rate. Transmits data H'55 when data H'00 is received error-free. Receives data H'AA.	H'00, H'00 ...... H'00 → ← H'00 H'55 → ← H'AA	· Measures low-level period of receive data H'00. · Calculates bit rate and sets it in BRR of SCI_2. · Transmits data H'00 to host as adjustment end indication. Transmits data H'AA to host when data H'55 is received.
Transfer of programming control program	Transmits number of bytes (N) of programming control program to be transferred as 2-byte data (low-order byte following high-order byte) Transmits 1-byte of programming control program (repeated for N times)	High-order byte and low-order byte → ← Echoback H'XX → ← Echoback	Echobacks the 2-byte data received. Echobacks received data to host and also transfers it to RAM (repeated for N times)
Flash memory erase	Boot program erase error Receives data H'AA.	← H'FF ← H'AA	Checks flash memory data, erases all flash memory blocks in case of written data existing, and transmits data H'AA to host. (If erase could not be done, transmits data H'FF to host and aborts operation.)
			Branches to programming control program transferred to on-chip RAM and starts execution.

Table 14.5 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible

Host Bit Rate	System Clock Frequency Range of LSI
19,200 bps	24 MHz
9,600 bps	8 to 24 MHz
4,800 bps	4 to 24 MHz

14.6.2 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 set branching conditions and provide on-board means of supplying programming data. The flash memory must contain the user program/erase control program or a program that provides the user program/erase control program from external memory. As the flash memory itself cannot be read during programming/erasing, transfer the user program/erase control program to on-chip RAM, as in boot mode. Figure 14.6 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 14.8, Flash Memory Programming/Erasing.


Reset-start
Program/erase?
No
Yes
Transfer user program/erase control program to RAM
Branch to flash memory application program
Branch to user program/erase control program in RAM
FWE=high*
Execute user program/erase control program (flash memory rewrite)
Clear FWE
Branch to flash memory application program

Note: * Do not constantly apply a high level to the FWE pin. Only apply a high level to the FWE pin when programming or erasing the flash memory. To prevent excessive programming or excessive erasing, while a high level is being applied to the FWE pin, activate the watchdog timer in case of handling CPU runaways.

Figure 14.6 Programming/Erasing Flowchart Example in User Program Mode

14.7 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 14.7 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 (EB0).

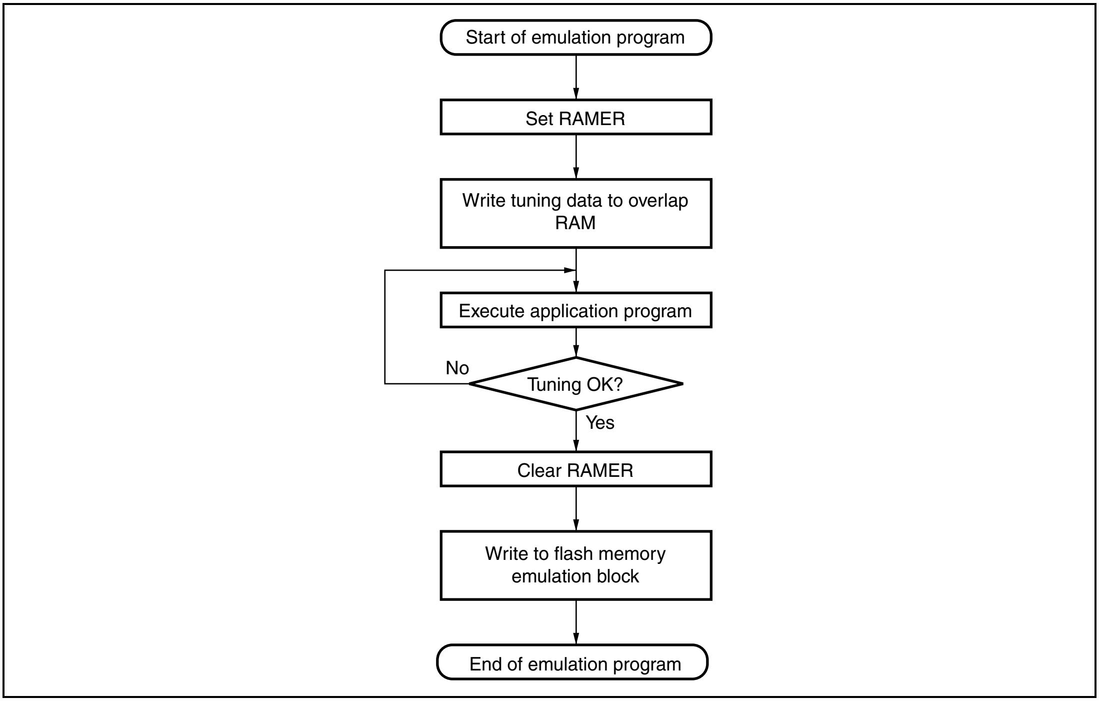

Start of emulation program
Set RAMER
Write tuning data to overlap RAM
Execute application program
No
Tuning OK?
Yes
Clear RAMER
Write to flash memory emulation block
End of emulation program

Figure 14.7 Flowchart for Flash Memory Emulation in RAM

An example in which flash memory block area EB0 is overlapped is shown in figure 14.8.

1. The RAM area to be overlapped is fixed at a 1-kbyte area in the range H'FFE000 to H'FFE3FF.
2. The flash memory area to overlap is selected by RAMER from a 1-kbyte area of the EB0 to EB3 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 P1 or E1 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.

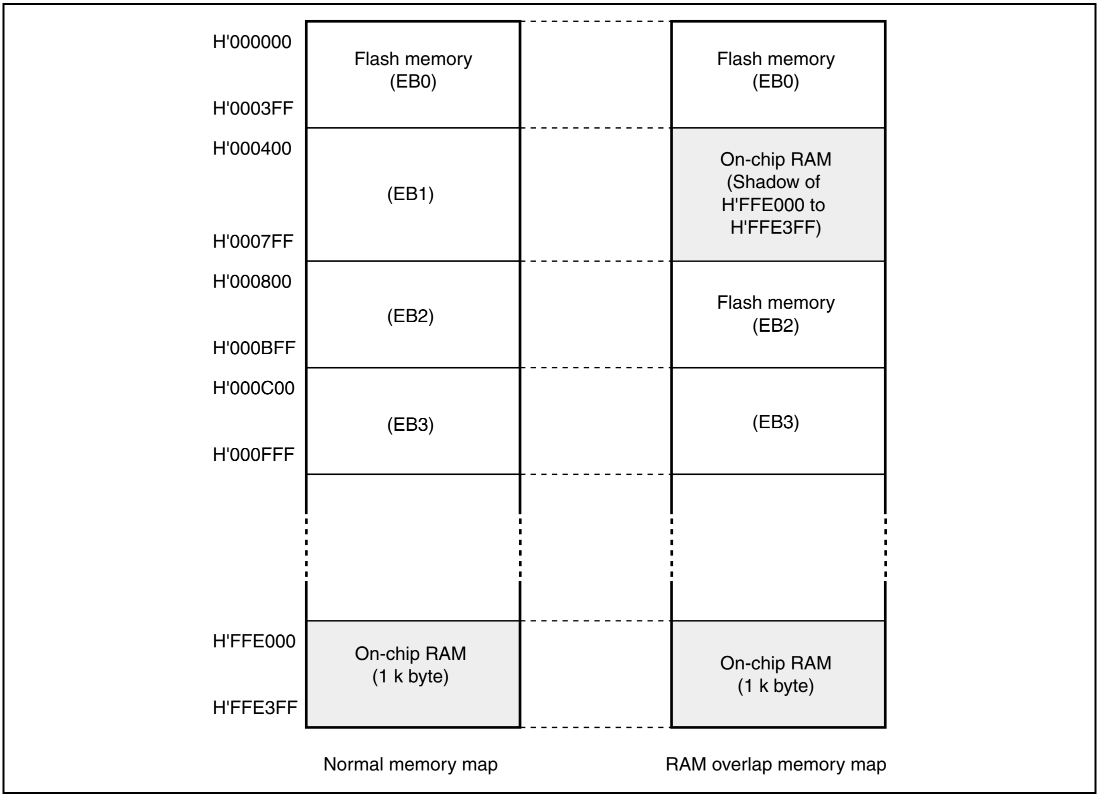


Figure 14.8 Example of RAM Overlap Operation

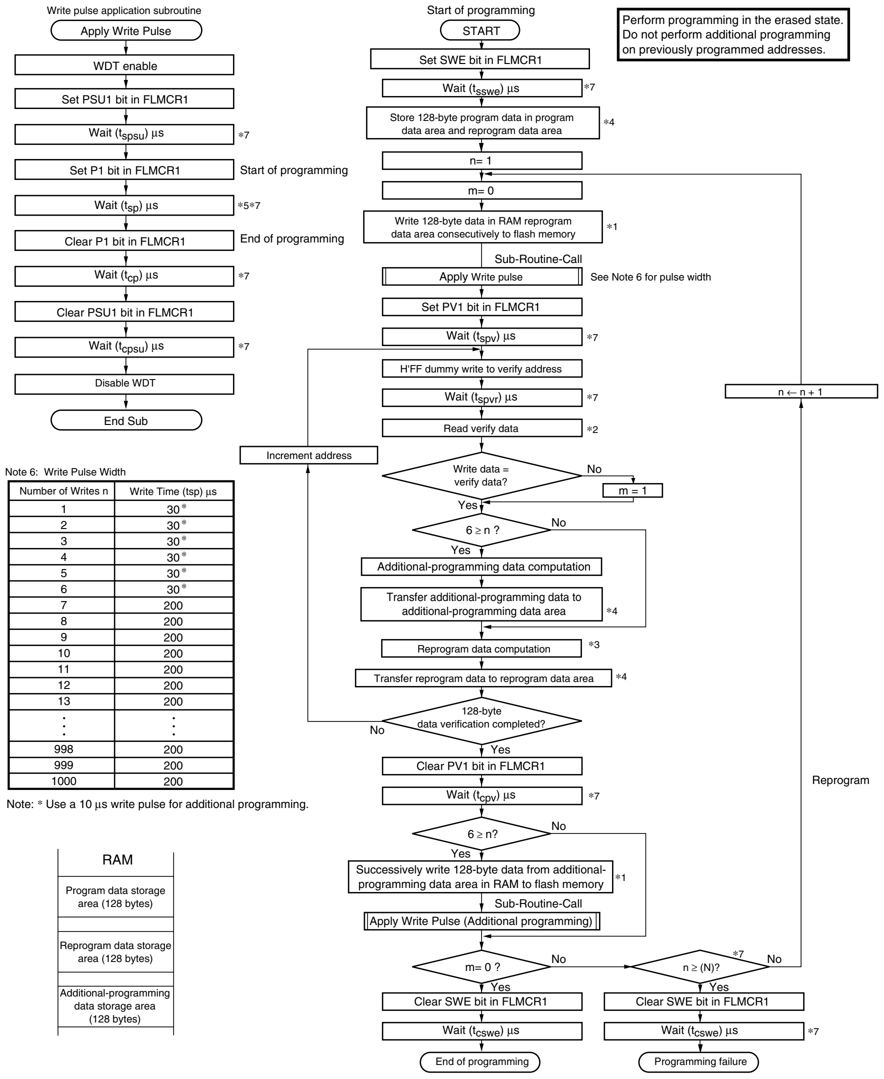
14.8 Flash Memory Programming/Erasing

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 14.8.1, Program/Program-Verify and section 14.8.2, Erase/Erase-Verify, respectively.

14.8.1 Program/Program-Verify

When writing data or programs to the flash memory, the program/program-verify flowchart shown in figure 14.9 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 14.9.
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 P1 bit is set to 1 is the programming time. Figure 14.9 shows the allowable programming times.
6. The watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. An overflow cycle of approximately 6.6 ms is allowed.
7. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower 2 bits are B'00. Verify data can be read in longwords from the address to which a dummy write was performed.
8. The maximum number of repetitions of the program/program-verify sequence of the same bit is 1,000.


Write pulse application subroutine
Apply Write Pulse
WDT enable
Set PSU1 bit in FLMCR1
Wait (t_{spsu}) µs
*7
Set P1 bit in FLMCR1
Start of programming
Wait (t_{sp}) µs
*5*7
Clear P1 bit in FLMCR1
End of programming
Wait (t_{cp}) µs
*7
Clear PSU1 bit in FLMCR1
Wait (t_{cpsu}) µs
*7
Disable WDT
End Sub
Start of programming
START
Perform programming in the erased state. Do not perform additional programming on previously programmed addresses.
Set SWE bit in FLMCR1
Wait (t_{sswe}) µs
*7
Store 128-byte program data in program data area and reprogram data area
*4
n= 1
m= 0
Write 128-byte data in RAM reprogram data area consecutively to flash memory
*1
Sub-Routine-Call
Apply Write pulse
See Note 6 for pulse width
Set PV1 bit in FLMCR1
Wait (t_{spv}) µs
*7
H'FF dummy write to verify address
Wait (t_{spvr}) µs
*7
Read verify data
*2
Increment address
Write data = verify data?
No
m = 1
Yes
6 ≥ n ?
No
Yes
Additional-programming data computation
Transfer additional-programming data to additional-programming data area
*4
Reprogram data computation
*3
Transfer reprogram data to reprogram data area
*4
128-byte data verification completed?
No
Yes
Clear PV1 bit in FLMCR1
Wait (t_{cpv}) µs
*7
6 ≥ n?
No
Yes
Successively write 128-byte data from additional-programming data area in RAM to flash memory
*1
Sub-Routine-Call
Apply Write Pulse (Additional programming)
m= 0 ?
No
Yes
*7
n ≥ (N)?
No
Yes
Clear SWE bit in FLMCR1
Wait (t_{cswe}) µs
End of programming
Clear SWE bit in FLMCR1
Wait (t_{cswe}) µs
*7
Programming failure
n ← n + 1
Reprogram
RAM
Program data storage area (128 bytes)
Reprogram data storage area (128 bytes)
Additional-programming data storage area (128 bytes)

Note 6: Write Pulse Width

Number of Writes n	Write Time (tsp) µs
1	30*
2	30*
3	30*
4	30*
5	30*
6	30*
7	200
8	200
9	200
10	200
11	200
12	200
13	200
⋮	⋮
998	200
999	200
1000	200

Note: * Use a 10 µs write pulse for additional programming.

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 18.5, 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 not to be executed
1	1	1	Additional programming not to be executed

Figure 14.9 Program/Program-Verify Flowchart

14.8.2 Erase/Erase-Verify

When erasing flash memory, the erase/erase-verify flowchart shown in figure 14.10 should be followed.

1. Prewriting (setting erase block data to all 0s) is not necessary.
2. Erasing is performed in block units. Make only a single-bit specification in the erase block register (EBR1). To erase multiple blocks, each block must be erased in turn.
3. The time during which the E1 bit is set to 1 is the flash memory erase time.
4. The watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc. An overflow cycle of approximately 19.8 ms is allowed.
5. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower two bits are B'00. Verify data can be read in longwords from the address to which a dummy write was performed.
6. If the read data is not erased successfully, set erase mode again, and repeat the erase/erase-verify sequence as before. The maximum number of repetitions of the erase/erase-verify sequence is 100.

14.8.3 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.

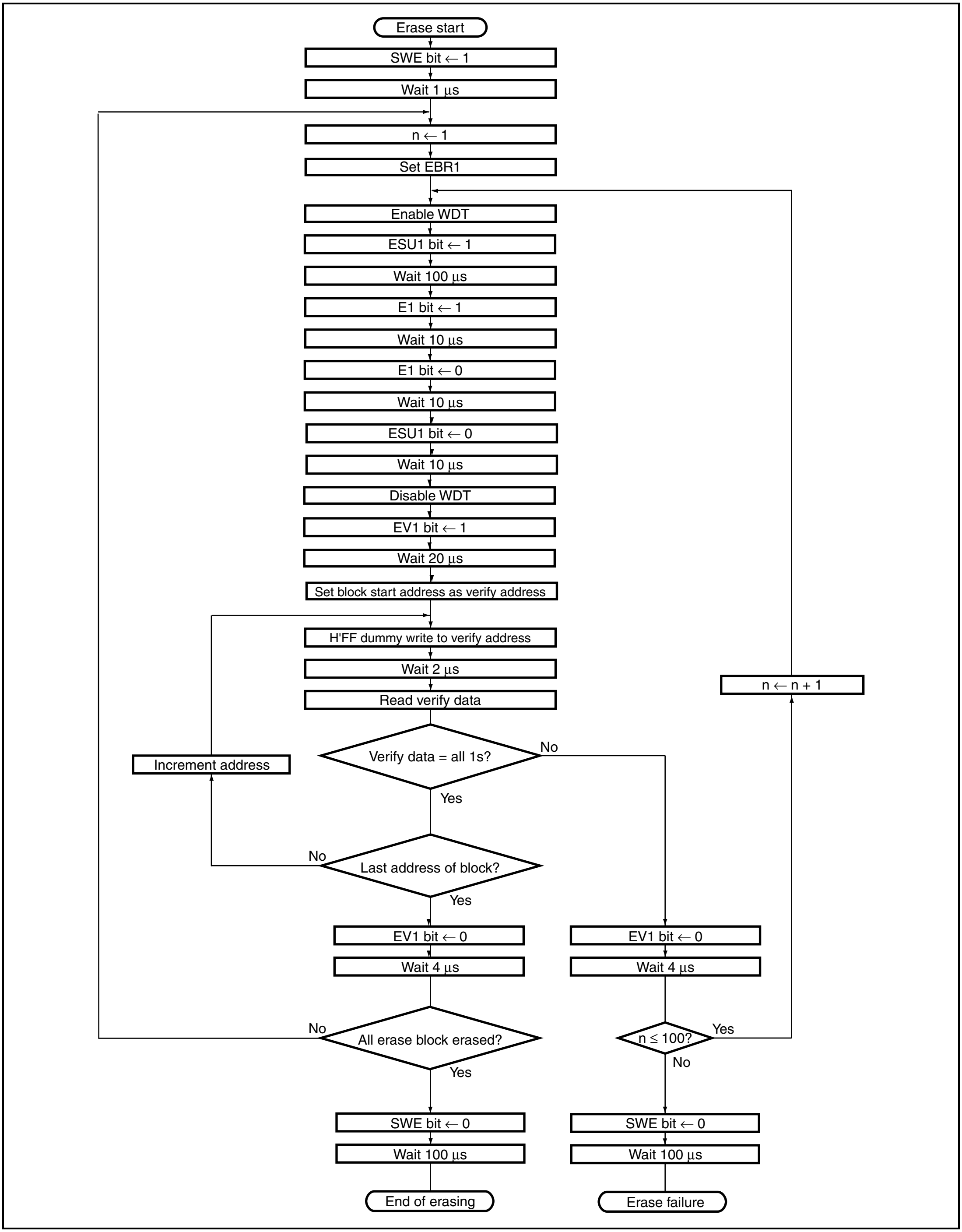


Figure 14.10 Erase/Erase-Verify Flowchart

14.9 Program/Erase Protection

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

14.9.1 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), and erase block register 1 (EBR1) 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.

14.9.2 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 P1 or E1 bit in FLMCR1 does not cause a transition to program mode or erase mode. By setting the erase block register 1 (EBR1), erase protection can be set for individual blocks. When EBR1 is set to H'00, erase protection is set for all blocks.

14.9.3 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 overprogramming or overerasing.

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

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

The FLMCR1, FLMCR2, and EBR1 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 P1 or E1 bit. However, PV1 and EV1 bit setting is enabled, and a transition can be made to verify mode. Error protection can be cleared only by a power-on reset.

14.10 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 Hitachi 64-kbyte flash memory on-chip MCU device type (FZTAT64V5A).

14.11 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.
- Power-down mode

  Part of the power supply circuitry is halted, and the flash memory can be read when the LSI is operating on the subclock.
- Standby mode

  All flash memory circuits are halted.

Table 14.6 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 2 ms, even when the external clock is being used.

Table 14.6 Flash Memory Operating States

LSI Operating State	Flash Memory Operating State
High-speed mode Medium-speed mode Sleep mode	Normal operating mode
Subactive mode Subsleep mode	When PDWND = 0: Power-down mode (read-only) When PDWND = 1: Normal operating mode (read-only)
Watch mode Software standby mode Hardware standby mode	Standby mode

Section 15 Clock Pulse Generator

This LSI has an on-chip clock pulse generator that generates the system clock (ϕ), the bus master clock, internal clock, and subclock. The clock pulse generator consists of an oscillator, PLL circuit, subclock divider, 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 15.1.

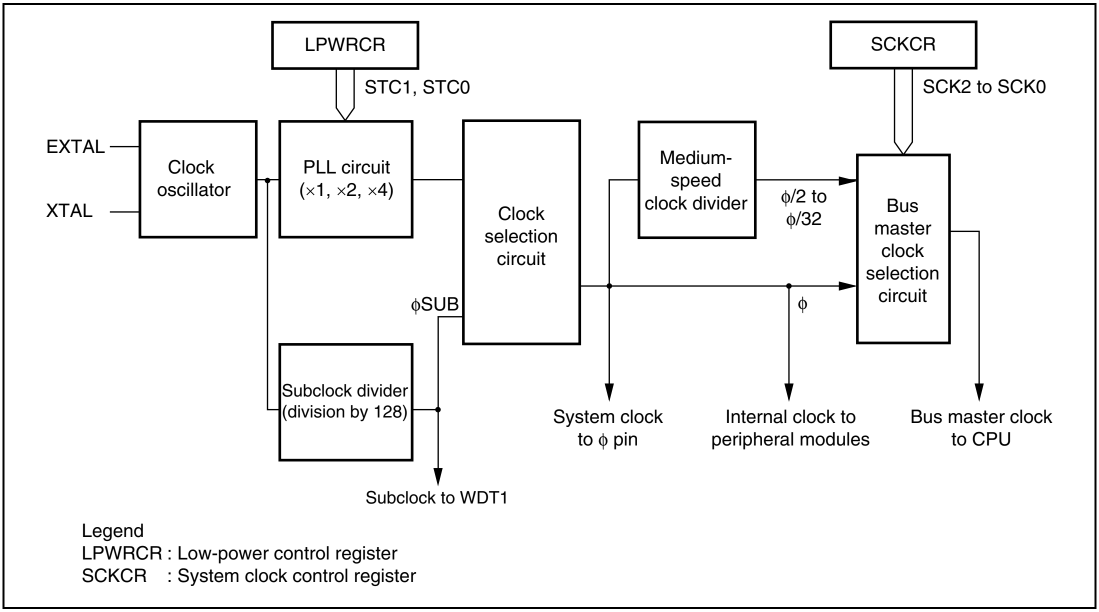

LPWRCR
SCKCR
STC1, STC0
SCK2 to SCK0
EXTAL
XTAL
Clock oscillator
PLL circuit (×1, ×2, ×4)
Clock selection circuit
Medium-speed clock divider
ϕ/2 to ϕ/32
Bus master clock selection circuit
ϕSUB
ϕ
Subclock divider (division by 128)
System clock to ϕ pin
Internal clock to peripheral modules
Bus master clock to CPU
Subclock to WDT1
Legend
LPWRCR : Low-power control register
SCKCR : System clock control register

Figure 15.1 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).

15.1 Register Descriptions

The on-chip clock pulse generator has the following registers.

- System clock control register (SCKCR)
- Low-power control register (LPWRCR)

15.1.1 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, Subactive Mode, Sleep Mode, Subsleep Mode 0: ϕ output 1: Fixed high Software Standby Mode, Watch Mode, Direct Transition 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.
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

Bit	Bit Name	Initial Value	R/W	Description
2	SCK2	0	R/W	System Clock Select 0 to 2
1	SCK1	0	R/W	These bits select the bus master clock.
0	SCK0	0	R/W	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

Legend
X: Don't care

15.1.2 Low-Power Control Register (LPWRCR)

LPWRCR performs power-down mode control, subclock generation control, oscillation circuit feedback resistance control, and frequency multiplication factor setting.

Bit	Bit Name	Initial Value	R/W	Description
7	DTON	0	R/W	See section 16.1.2, Low-Power Control Register (LPWRCR).
6	LSON	0	R/W	
5	—	0	R/W	Reserved Only write 0 to this bit.
4	SUBSTP	0	R/W	Subclock Generation Control 0: Enables subclock generation 1: Disables subclock generation
3	RFCUT	0	R/W	Oscillation Circuit Feedback Resistance Control 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. Change is valid when software standby mode is entered or after software standby mode is recovered. Note: With a crystal resonator, the resonator will not operate if this bit is set to 1.
2	—	0	R/W	Reserved Only write 0 to this bit.

Bit	Bit Name	Initial Value	R/W	Description
1	STC1	0	R/W	Frequency Multiplication Factor
0	STC0	0	R/W	The STC bits specify the frequency multiplication factor of the PLL circuit. 00: ×1 01: ×2 10: ×4 11: Setting prohibited

15.2 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 24 MHz.

15.2.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as shown in the example in figure 15.2. Select the damping resistance R_d according to table 15.1. An AT-cut parallel-resonance crystal should be used.

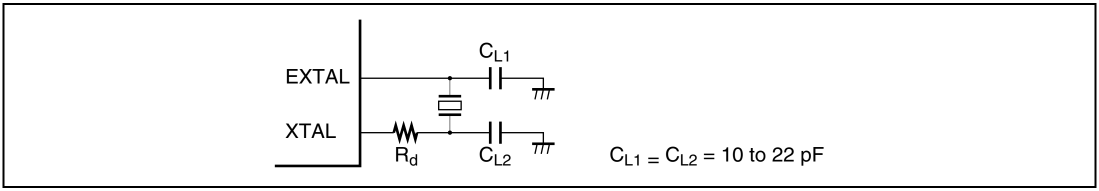


Figure 15.2 Connection of Crystal Resonator (Example)

Table 15.1 Damping Resistance Value

Frequency (MHz)	4	8	12	16	20	24
R_d (Ω)	500	200	0	0	0	0

Figure 15.3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 15.2.

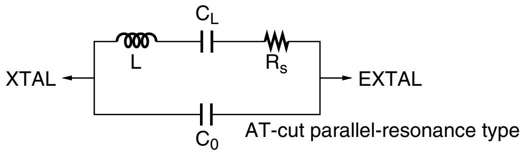


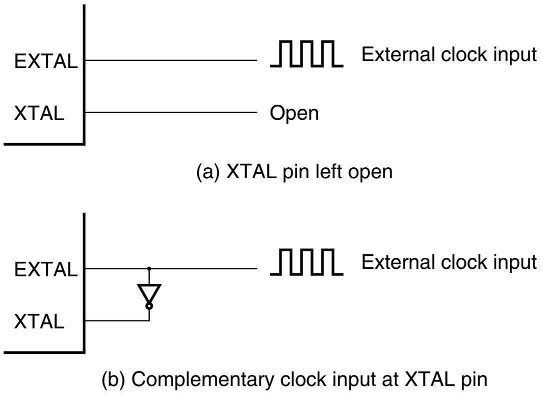
Figure 15.3 Crystal Resonator Equivalent Circuit

Table 15.2 Crystal Resonator Characteristics

Frequency (MHz)	4	8	12	16	20	24
R_s max (Ω)	120	80	60	50	40	40
C_0 max (pF)	7	7	7	7	7	7

15.2.2 External Clock Input

Circuit Configuration: An external clock signal can be input as shown in the examples in figure 15.4. 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.



(a) XTAL pin left open

(b) Complementary clock input at XTAL pin

Figure 15.4 External Clock Input (Examples)

Table 15.3 shows the input conditions for the external clock.

Table 15.3 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 15.5	
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		
Clock low pulse width level	t_{CL}	0.4	0.6	t_{cyc}	φ ≥ 5 MHz	Figure 18.2
		80	—	ns	φ < 5 MHz	
Clock high pulse width level	t_{CH}	0.4	0.6	t_{cyc}	φ ≥ 5 MHz	
		80	—	ns	φ < 5 MHz	

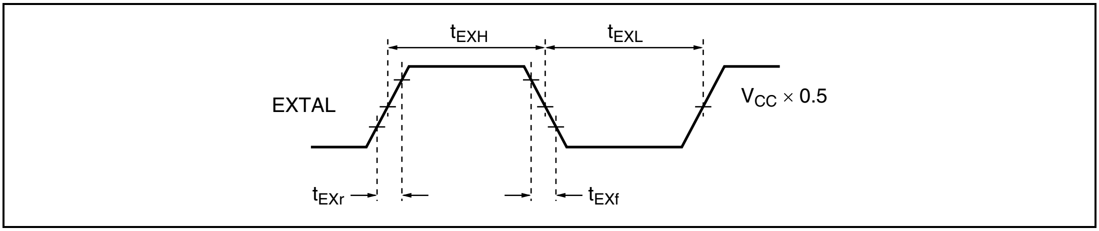


Figure 15.5 External Clock Input Timing

15.3 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, refer to section 16.1.1, 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.

15.4 Subclock Divider

The subclock divider divides the clock generated by the oscillator by 128 to generate a subclock. When using the subclock as a system clock, adjustment by software is needed.

15.5 Medium-Speed Clock Divider

The medium-speed clock divider divides the system clock to generate φ/2, φ/4, φ/8, φ/16, and φ/32.

15.6 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 (φ/2, φ/4, φ/8, φ/16, φ/32).

15.7 Usage Notes

15.7.1 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.

15.7.2 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 15.6. This is to prevent induction from interfering with correct oscillation.

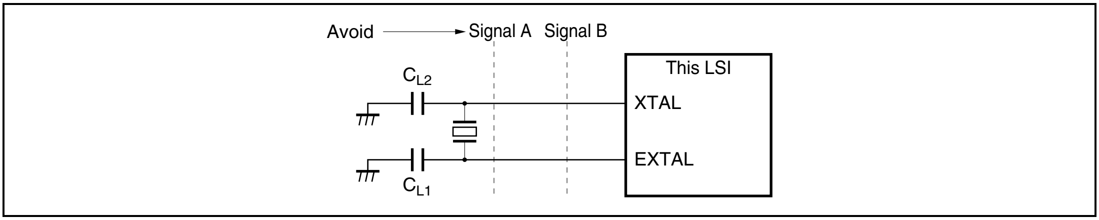


Figure 15.6 Note on Board Design of Oscillator Circuit

Figure 15.7 shows external circuitry recommended to be provided around the PLL circuit. Place oscillation stabilization capacitor C1 and resistor R1 close to the PLLCAP pin, and ensure that no other signal lines cross this line. Separate PLLVcL and 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.

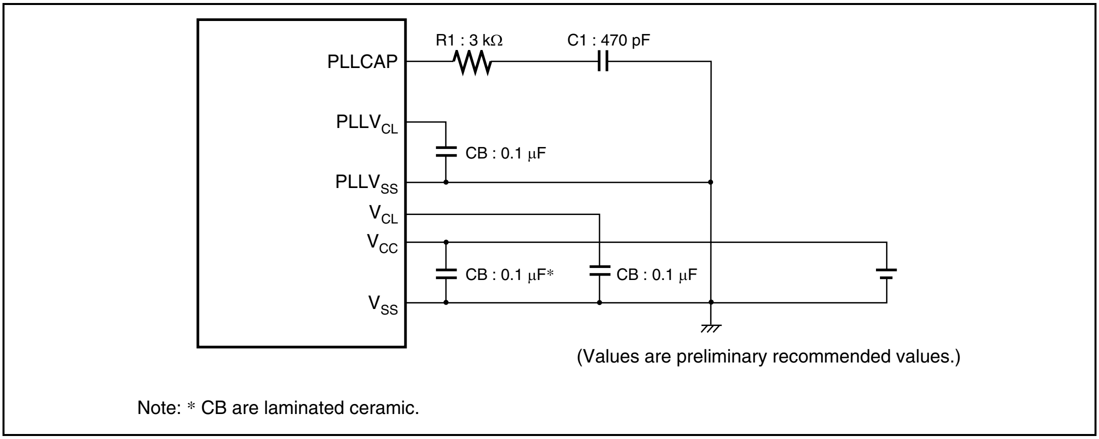


Figure 15.7 External Circuitry Recommended for PLL Circuit

Section 16 Power-Down Modes

In addition to the normal program execution state, this LSI has eight 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.

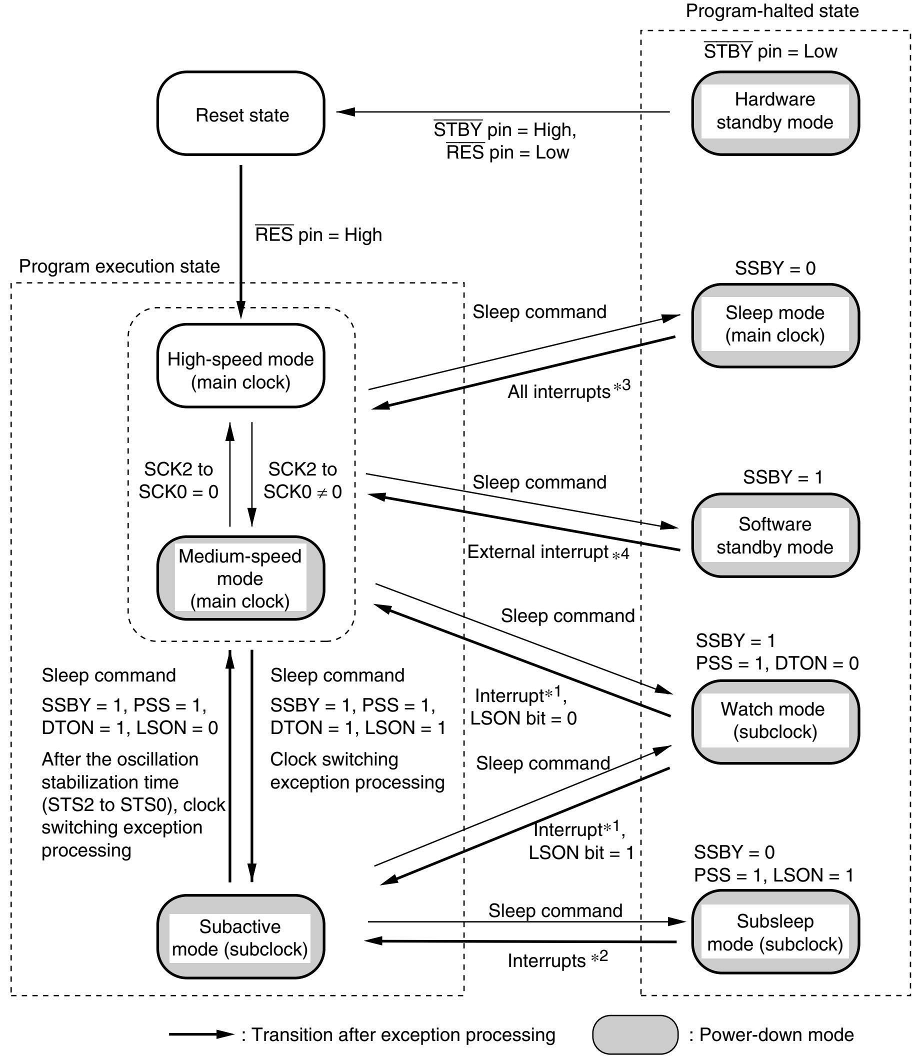
This LSI's operating modes are as follows:

(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

(2) to (9) are power-down modes. Sleep and subsleep modes 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 is in high-speed mode or module stop mode.

Figure 16.1 shows a mode transition. Table 16.1 shows the conditions of transition between modes when executing the SLEEP instruction and the state after transition back from low power mode due to an interrupt. Table 16.2 shows the internal state of the LSI in each mode.



Notes : *1 NMI, IRQ0 to IRQ5, and WDT_1 interrupts

*2 NMI, IRQ0 to IRQ5, WDT_0, and WDT_1 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 RES is driven low.
- From any state, a transition to hardware standby mode occurs when STBY is driven low.
- Always select high-speed mode before making a transition to watch mode or subactive mode.

Figure 16.1 Mode Transition Diagram

Table 16.1 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

Table 16.2 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
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	WDT_0	Functioning	Functioning	Functioning	—	Halted (retained)	Subclock operation	Subclock operation	Halted (retained)	Halted (reset)
	WDT_1	Functioning	Functioning	Functioning	—	Subclock operation	Subclock operation	Subclock operation	Halted (retained)	Halted (reset)
	I/O	Functioning	Functioning	Functioning	Functioning	Retained	Functioning	Retained	Retained	High impedance
	TPU	Functioning	Functioning	Functioning	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	SCI	Functioning	Functioning	Functioning	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)
	HCAN									
	A/D									
	RAM	Functioning	Medium-speed operation	Functioning	Functioning	Retained	Functioning	Retained	Retained	Retained

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).

16.1 Register Descriptions

Registers related to the power down mode are shown below. For details on the system clock control register (SCKCR), refer to section 15.1.1, System Clock Control Register (SCKCR).

- System clock control register (SCKCR)
- Standby control register (SBYCR)
- Low-power control register (LPWRCR)
- Module stop control register A (MSTPCRA)
- Module stop control register B (MSTPCRB)
- Module stop control register C (MSTPCRC)

16.1.1 Standby Control Register (SBYCR)

SBYCR performs software standby mode control.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby This bit determines the operating mode when a transition is made to power-down mode after executing the SLEEP instruction according to the combination of the other control bits. 0: Shifts to sleep mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode. Shifts 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. Shifts to watch mode or high-speed mode when the SLEEP instruction is executed in subactive mode. Note: When entering watch mode or subactive mode, the operating mode must be set to high-speed mode.

Bit	Bit Name	Initial Value	R/W	Description
6	STS2	0	R/W	Standby Timer Select 0 to 2
5	STS1	0	R/W	These bits select the MCU wait time for clock stabilization when software standby mode, watch mode, or subactive mode is cancelled by an external interrupt. With a crystal oscillator (table 16.3), select a wait time of 8ms (oscillation stabilization time) or more, depending on the operating frequency. With an external clock, select a wait time of 2 ms or more.
4	STS0	0	R/W	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
3	—	1	R/W	Reserved Only 1 should be written to this bit.
2 to 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

16.1.2 Low-Power Control Register (LPWRCR)

LPWRCR performs power-down mode control, subclock generation control, oscillation circuit feedback resistance control, and frequency multiplication factor setting.

Bit	Bit Name	Initial Value	R/W	Description
7	DTON	0	R/W	Direct Transition ON Flag 0: When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts to sleep mode, software standby mode, or watch mode*. When the SLEEP instruction is executed in subactive mode, operation shifts to subsleep mode or watch mode. 1: When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts directly to subactive mode*, or shifts to sleep mode or software standby mode. When the SLEEP instruction is executed in subactive mode, operation shifts directly to high-speed mode, or shifts to subsleep mode.
6	LSON	0	R/W	Low-Speed ON Flag 0: When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts to sleep mode, software standby mode, or watch mode*. When the SLEEP instruction is executed in subactive mode, operation shifts to watch mode or shifts directly to high-speed mode. Operation shifts to high-speed mode when watch mode is cancelled. 1: When the SLEEP instruction is executed in high-speed mode, operation shifts to watch mode or subactive mode*. When the SLEEP instruction is executed in sub-active mode, operation shifts to subsleep mode or watch mode. Operation shifts to subactive mode when watch mode is cancelled.
5	—	0	R/W	Reserved This bit can be read from and written to. However, do not write 1 to this bit.
4	SUBSTP	0	R/W	Subclock Generation Control 0: Enables subclock generation 1: Disables subclock generation

Bit	Bit Name	Initial Value	R/W	Description
3	RFCUT	0	R/W	Oscillation Circuit Feedback Resistance Control 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. Change is valid when software standby mode is entered or after software standby mode is recovered. Note: With a crystal resonator, the resonator will not operate if this bit is set to 1.
2	—	0	R/W	Reserved This bit can be read from and written to. However, do not write 1 to this bit.
1	STC1	0	R/W	Frequency Multiplication Factor Setting
0	STC0	0	R/W	These bits specify the frequency multiplication factor of the PLL circuit. 00: x1 01: x2 10: x4 11: Setting prohibited

Note: * Always set high-speed mode when shifting to watch mode or subactive mode.

16.1.3 Module Stop Control Registers A to C (MSTPCRA to MSTPCRC)

MSTPCR performs module stop mode control. 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*	0	R/W	
6	MSTPA6*	0	R/W	
5	MSTPA5	1	R/W	16-bit timer pulse unit (TPU)
4	MSTPA4*	1	R/W	
3	MSTPA3*	1	R/W	
2	MSTPA2*	1	R/W	
1	MSTPA1	1	R/W	A/D converter
0	MSTPA0*	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	
3	MSTPB3*	1	R/W	
2	MSTPB2*	1	R/W	
1	MSTPB1*	1	R/W	
0	MSTPB0*	1	R/W	

MSTPCRC

Bit	Bit Name	Initial Value	R/W	Module
7	MSTPC7*	1	R/W	
6	MSTPC6*	1	R/W	
5	MSTPC5*	1	R/W	
4	MSTPC4*	1	R/W	
3	MSTPC3	1	R/W	Hitachi Controller Area Network (HCAN)
2	MSTPC2*	1	R/W	
1	MSTPC1*	1	R/W	
0	MSTPC0*	1	R/W	

Note: * MSTPA7 and MSTPA6 are readable/writable bits with an initial value of 0 and should always be written with 0.

MSTPA4 to MSTPA2, MSTPA0, MSTPB4 to MSTPB0, MSTPC7 to MSTPC4, and MSTPC2 to MSTPC0 are readable/writable bits with an initial value of 1 and should always be written with 1.

16.2 Medium-Speed Mode

When the SCK0 to SCK2 bits in SCKCR are set to 1, 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 SCK0 to SCK2 bits.

On-chip peripheral modules other than 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 4 states, and internal I/O registers in 8 states.

Medium-speed mode is cleared by clearing all of bits SCK0 to SCK2 to 0. A transition is made to high-speed mode and medium-speed mode is cleared at the end of the current bus cycle.

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, a transition is made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode is restored.

When the SLEEP instruction is executed with the SSBY bit = 1, operation shifts to the software standby mode. When software standby mode is cleared by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is set low and medium-speed mode is cancelled, operation shifts to the reset state. The same applies in the case of a reset caused by overflow of the watchdog timer.

When the STBY pin is driven low, a transition is made to hardware standby mode.

Figure 16.2 shows the timing for transition to and clearance of medium-speed mode.

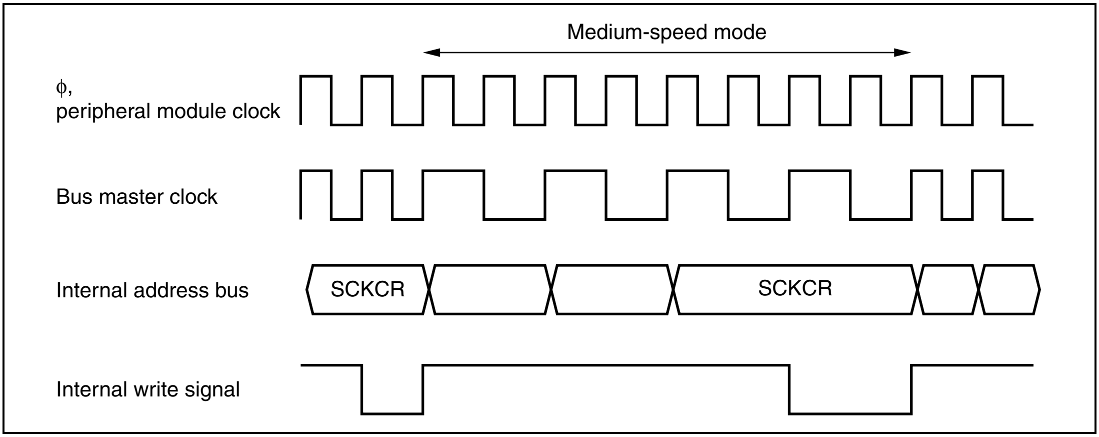


Figure 16.2 Medium-Speed Mode Transition and Clearance Timing

16.3 Sleep Mode

16.3.1 Transition to Sleep Mode

If SLEEP instruction is executed when the SBYCR SSBY bit = 0, the CPU enters the sleep mode. In sleep mode, CPU operation stops, however the contents of the CPU's internal registers are retained. Other peripheral modules do not stop.

16.3.2 Clearing Sleep Mode

Sleep mode is cleared by any interrupt, or signals at the RES, or STBY pins.

- Exiting Sleep Mode by Interrupts:

  When an interrupt occurs, sleep mode is exited and interrupt exception processing starts. Sleep mode is not exited if the interrupt is disabled, or if interrupts other than NMI are masked by the CPU.
- Exiting Sleep Mode by RES pin:

  Setting the RES pin Low selects the reset state. After the stipulated reset input duration, driving the RES pin High restart the CPU performing reset exception processing.
- Exiting Sleep Mode by STBY Pin:

  When the STBY pin level is driven low, a transition is made to hardware standby mode.

16.4 Software Standby Mode

16.4.1 Transition to Software Standby Mode

A transition is made to software standby mode if the SLEEP instruction is executed when the SBYCR SSBY bit is set to 1. In this mode, the CPU, on-chip peripheral modules, and oscillator, all stop. However, the contents of the CPU's internal registers, on-chip RAM data, and the states of on-chip peripheral modules other than the SCI, A/D converter, and the states of I/O ports, are retained. In this mode, the oscillator stops, and therefore power consumption is significantly reduced.

16.4.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$), or by means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

- Clearing with an interrupt

  When an NMI or IRQ0 to IRQ5 interrupt request signal is input, clock oscillation starts, and after the time set in bits STS0 to STS2 in SBYCR has elapsed, stable clocks are supplied to the entire chip, software standby mode is cleared, and interrupt exception handling is started.

  When clearing software standby mode with an IRQ0 to IRQ5 interrupt, set the corresponding enable bit to 1 and ensure that no interrupt with a higher priority than interrupts IRQ0 to IRQ5 is generated. Software standby mode cannot be cleared if the interrupt has been masked on the CPU side.
- Clearing with the $\overline{\text{RES}}$ pin

  When the $\overline{\text{RES}}$ pin is driven low, clock oscillation is started. At the same time as clock oscillation starts, clocks are supplied to the entire chip. Note that the $\overline{\text{RES}}$ pin must be held low until clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin goes high, the CPU begins reset exception handling.
- Clearing with the $\overline{\text{STBY}}$ pin

  When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

16.4.3 Setting Oscillation Stabilization Time after Clearing Software Standby Mode

Bits STS2 to STS0 in SBYCR should be set as described below.

- Using a Crystal Oscillator

  Set bits STS0 to STS2 so that the standby time is at least 8 ms (the oscillation stabilization time).

  Table 16.3 shows the standby times for different operating frequencies and settings of bits STS0 to STS2.

- Using an External Clock

  The PLL circuit requires a time for stabilization. Set bits STS0 to STS2 so that the standby time is at least 2 ms.

Table 16.3 Oscillation Stabilization Time Settings

				24	20	16	12	10	8	6	4	
STS2	**STS1**	**STS0**	**Standby Time**	**MHz**	**MHz**	**MHz**	**MHz**	**MHz**	**MHz**	**MHz**	**MHz**	**Unit**
0	0	0	8192 states	0.34	0.41	0.51	0.68	0.8	1.0	1.3	2.0	ms
		1	16384 states	0.68	0.82	1.0	1.3	1.6	2.0	2.7	4.1	
	1	0	32768 states	1.4	1.6	2.0	2.7	3.3	4.1	5.5	8.2	
		1	65536 states	2.7	3.3	4.1	5.5	6.6	8.2	10.9	16.4	
1	0	0	131072 states	5.5	6.6	8.2	10.9	13.1	16.4	21.8	32.8	
		1	262144 states	10.9	13.1	16.4	21.8	26.2	32.8	43.6	65.6	
	1	0	Reserved	—	—	—	—	—	—	—	—	—
		1	16 states*	0.7	0.8	1.0	1.3	1.6	2.0	1.7	4.0	μs

▨: Recommended time setting

Note: * Cannot be set.

16.4.4 Software Standby Mode Application Example

Figure 16.3 shows an example in which a transition is made to software standby mode at a falling edge on the NMI pin, and software standby mode is cleared at a rising edge on the NMI pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in SYSCR cleared to 0 (falling edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY bit is set to 1, and a SLEEP instruction is executed, causing a transition to software standby mode.

Software standby mode is then cleared at the rising edge on the NMI pin.

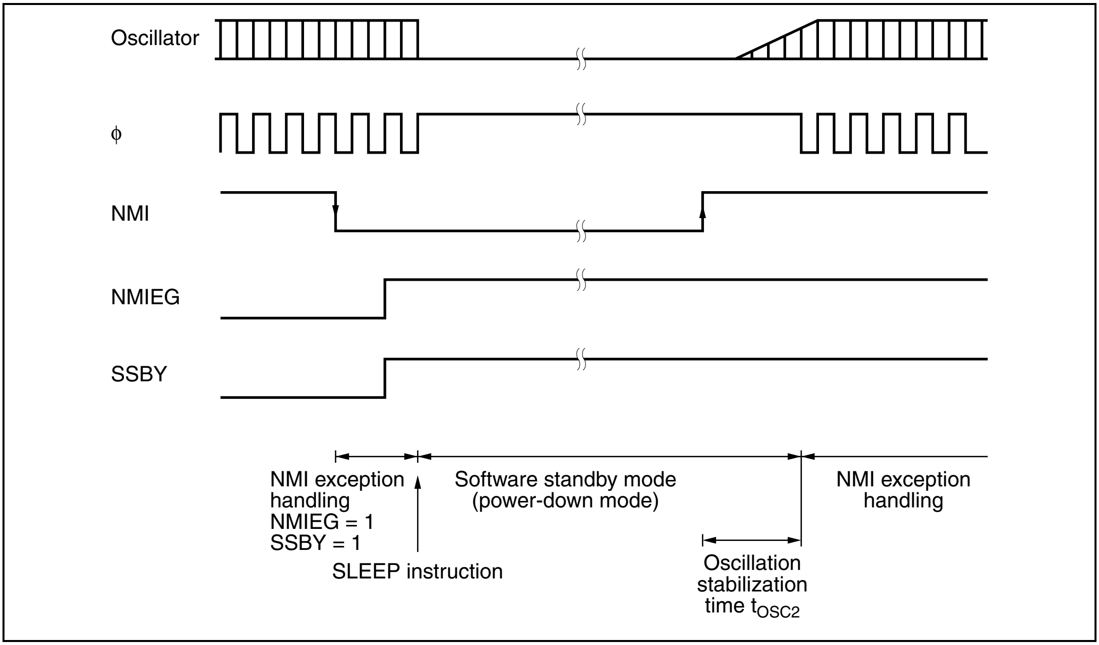

Oscillator
φ
NMI
NMIEG
SSBY
NMI exception handling NMIEG = 1 SSBY = 1
SLEEP instruction
Software standby mode (power-down mode)
Oscillation stabilization time t_{OSC2}
NMI exception handling

Figure 16.3 Software Standby Mode Application Example

16.5 Hardware Standby Mode

16.5.1 Transition to Hardware Standby Mode

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode from any mode.

In hardware standby mode, all functions enter the reset state and stop operation, resulting in a significant reduction in power consumption. As long as the prescribed voltage is supplied, on-chip RAM data is retained. I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 before driving the $\overline{\text{STBY}}$ pin low.

Do not change the state of the mode pins (MD0 to MD2) while this LSI is in hardware standby mode.

16.5.2 Clearing Hardware Standby Mode

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When the $\overline{\text{STBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is set and clock oscillation is started. Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillator stabilizes (at least 8 ms—the 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.

16.5.3 Hardware Standby Mode Timings

Timing of Transition to Hardware Standby Mode

1. To retain RAM contents with the RAME bit set to 1 in SYSCR

   Drive the $\overline{\text{RES}}$ signal low at least 10 states before the $\overline{\text{STBY}}$ signal goes low, as shown in figure 16.4. After $\overline{\text{STBY}}$ has gone low, $\overline{\text{RES}}$ has to wait for at least 0 ns before becoming high.

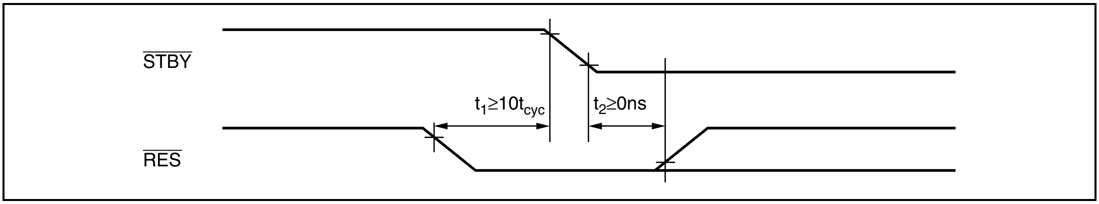


Figure 16.4 Timing of Transition to Hardware Standby Mode

2. To retain RAM contents with the RAME bit cleared to 0 in SYSCR, or when RAM contents do not need to be retained

   $\overline{\text{RES}}$ does not have to be driven low as in the above case.

Timing of Recovery from Hardware Standby Mode

Drive the $\overline{\text{RES}}$ signal low approximately 100 ns or more before $\overline{\text{STBY}}$ goes high to execute a power-on reset.

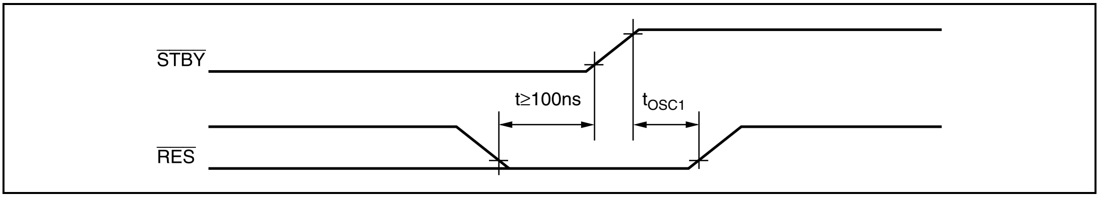


Figure 16.5 Timing of Recovery from Hardware Standby Mode

16.6 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.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the module starts operating at the end of the bus cycle. In module stop mode, the internal states of modules other than the SCI*, HCAN, and A/D converter are retained.

After reset clearance, all modules 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: The internal states of some SCI registers are retained.

16.7 Watch Mode

16.7.1 Transition to Watch Mode

CPU operation makes a transition to watch mode when the SLEEP instruction is executed in high-speed mode or subactive mode with the SSBY bit in SBYCR = 1, the DTON bit in LPWRCR = 0, and the PSS bit in TCSR_1 (WDT_1) = 1.

In watch mode, the CPU is stopped and peripheral modules other than WDT_1 are also stopped. The contents of the CPU's internal registers, on-chip RAM data, and the states of on-chip peripheral modules other than the SCI, HCAN, and A/D converter, and the states of I/O ports, are retained.

16.7.2 Canceling Watch Mode

Watch mode is canceled by any interrupt (WOVI1 interrupt, NMI pin, or $\overline{\mathrm{IRQ0}}$ to $\overline{\mathrm{IRQ5}}$ pin), or signals at the $\overline{\mathrm{RES}}$, or $\overline{\mathrm{STBY}}$ pin.

Canceling Watch Mode by Interrupt: 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 all LSI circuits and interrupt exception processing starts after the time set in the STS2 to STS0 bits of SBYCR has elapsed. In case of an IRQ0 to IRQ5 interrupt, watch mode is not canceled if the corresponding enable bit has been cleared to 0. In case of the interrupt from the on-chip peripheral modules, 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 section 16.4.3, Setting Oscillation Stabilization Time after Clearing Software Standby Mode.

Canceling Watch Mode by $\overline{\text{RES}}$ pin: For canceling watch mode by the $\overline{\text{RES}}$ pin, see section 16.4.2, Clearing Software Standby Mode.

Canceling Watch Mode by $\overline{\text{STBY}}$ pin: When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

16.8 Subsleep Mode

16.8.1 Transition to 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_1 (WDT_1) = 1, CPU operation shifts to subsleep mode.

In subsleep mode, the CPU is stopped and peripheral modules other than WDT_0 and WDT_1 are also stopped. The contents of the CPU's internal registers, on-chip RAM data, and the states of on-chip peripheral modules other than the SCI, HCAN, and A/D converter, and the states of I/O ports, are retained.

16.8.2 Canceling Subsleep Mode

Subsleep mode is canceled by any interrupt (WOVI0 or WOVI1 interrupt, NMI pin, or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$ pin), or signals at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin.

Canceling Subsleep Mode by Interrupt: When an interrupt occurs, subsleep mode is canceled and interrupt exception processing starts.

In case of an IRQ0 to IRQ5 interrupt, subsleep mode is not canceled if the corresponding enable bit has been cleared to 0. In case of the interrupt from the on-chip peripheral modules, 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.

Canceling Subsleep Mode by $\overline{\text{RES}}$ pin: For canceling subsleep mode by the $\overline{\text{RES}}$ pin, see section 16.4.2, Clearing Software Standby Mode.

Canceling Subsleep Mode by $\overline{\text{STBY}}$ pin: When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

16.9 Subactive Mode

16.9.1 Transition to Subactive Mode

CPU operation makes a transition 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_1 (WDT_1) = 1. When an interrupt occurs in watch mode, and if the LSON bit of LPWRCR is 1, a transition is made to subactive mode. And 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 WDT_0 and WDT_1 are also stopped.

When operating the CPU in subactive mode, the SCK2 to SCK0 bits in SCKCR must be set to 0.

16.9.2 Canceling Subactive Mode

Subactive mode is canceled by the SLEEP instruction or signals at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin.

Canceling Subactive Mode by SLEEP Instruction: 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_1 (WDT_1) = 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_1 (WDT_1) = 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_1 (WDT_1) = 1, a direct transition is made to high-speed mode (SCK0 to SCK2 are all 0).

Canceling Subactive Mode by $\overline{\text{RES}}$ pin: For canceling subactive mode by the $\overline{\text{RES}}$ pin, see section 16.4.2, Clearing Software Standby Mode.

Canceling Subactive Mode by $\overline{\text{STBY}}$ pin: When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

16.10 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 processing starts.

16.10.1 Direct Transitions 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_1 (WDT_1) = 1, to make a direct transition to subactive mode.

16.10.2 Direct Transitions 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_1 (WDT_1) = 1, to make a direct transition to high-speed mode after the time set in the STS2 to STS0 bits of SBYCR has elapsed.

16.11 ϕ Clock Output Disabling Function

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 16.4 shows the state of the ϕ pin in each processing state.

Table 16.4 ϕ Pin State in Each Processing State

Register Settings						
DDR	**PSTOP**	**High-Speed Mode, Medium-Speed Mode**	**Subactive Mode**	**Sleep Mode, Subsleep Mode**	**Software Standby Mode, Watch Mode, Direct Transitions**	**Hardware Standby Mode**
0	X	High impedance	High impedance	High impedance	High impedance	High impedance
1	0	ϕ output	ϕ_{SUB} output	ϕ output	Fixed high	High impedance
1	1	Fixed high	Fixed high	Fixed high	Fixed high	High impedance

Legend
X: Don't care

16.12 Usage Notes

16.12.1 I/O Port Status

In software standby mode, I/O port states are retained. Therefore, there is no reduction in current consumption for the output current when a high-level signal is output.

16.12.2 Current Consumption during Oscillation Stabilization Wait Period

Current consumption increases during the oscillation stabilization wait period.

16.12.3 On-Chip Peripheral Module Interrupt

The on-chip peripheral module (TPU), that halts in subactive mode, cannot cancel that interrupt in subactive mode. Thus, if a transition is made to subactive mode via watch mode when an interrupt has been requested, it will not be possible to clear the CPU interrupt source.

Interrupts should therefore be disabled before executing the SLEEP instruction, then entering watch mode.

16.12.4 Writing to MSTPCR

MSTPCR should only be written to by the CPU.

Section 17 List of Registers

The address list gives information on the on-chip I/O register addresses, how the register bits are configured, and the register states in each operating mode. The information is given as shown below.

1. Register addresses (address order)

- Registers are listed from the lower allocation addresses.
- Registers are classified by functional modules.
- The access size is indicated.

2. Register bits

- Bit configurations of the registers are described in the same order as the register addresses.
- Reserved bits are indicated by — in the bit name column.
- No entry in the bit-name column indicates that the whole register is allocated as a counter or for holding data.

3. Register states in each operating mode

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

17.1 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	Abbrevia-tion	Bit No.	Address*	Module	Data Width	Access State
Master control register	MCR	8	H'F800	HCAN	16	4
General status register	GSR	8	H'F801	HCAN	16	4
Bit configuration register	BCR	16	H'F802	HCAN	16	4
Mailbox configuration register	MBCR	16	H'F804	HCAN	16	4
Transmit wait register	TXPR	16	H'F806	HCAN	16	4
Transmit wait cancel register	TXCR	16	H'F808	HCAN	16	4
Transmit acknowledge register	TXACK	16	H'F80A	HCAN	16	4
Abort acknowledge register	ABACK	16	H'F80C	HCAN	16	4
Receive complete register	RXPR	16	H'F80E	HCAN	16	4
Remote request register	RFPR	16	H'F810	HCAN	16	4
Interrupt register	IRR	16	H'F812	HCAN	16	4
Mailbox interrupt mask register	MBIMR	16	H'F814	HCAN	16	4
Interrupt mask register	IMR	16	H'F816	HCAN	16	4
Receive error counter	REC	8	H'F818	HCAN	16	4
Transmit error counter	TEC	8	H'F819	HCAN	16	4
Unread message status register	UMSR	16	H'F81A	HCAN	16	4
Local acceptance filter mask L	LAFML	16	H'F81C	HCAN	16	4
Local acceptance filter mask H	LAFMH	16	H'F81E	HCAN	16	4
Message control 0[1]	MC0[1]	8	H'F820	HCAN	16	4
Message control 0[2]	MC0[2]	8	H'F821	HCAN	16	4
Message control 0[3]	MC0[3]	8	H'F822	HCAN	16	4
Message control 0[4]	MC0[4]	8	H'F823	HCAN	16	4
Message control 0[5]	MC0[5]	8	H'F824	HCAN	16	4
Message control 0[6]	MC0[6]	8	H'F825	HCAN	16	4
Message control 0[7]	MC0[7]	8	H'F826	HCAN	16	4
Message control 0[8]	MC0[8]	8	H'F827	HCAN	16	4
Message control 1[1]	MC1[1]	8	H'F828	HCAN	16	4

Register Name	Abbrevia-tion	Bit No.	Address*	Module	Data Width	Access State
Message control 1[2]	MC1[2]	8	H'F829	HCAN	16	4
Message control 1[3]	MC1[3]	8	H'F82A	HCAN	16	4
Message control 1[4]	MC1[4]	8	H'F82B	HCAN	16	4
Message control 1[5]	MC1[5]	8	H'F82C	HCAN	16	4
Message control 1[6]	MC1[6]	8	H'F82D	HCAN	16	4
Message control 1[7]	MC1[7]	8	H'F82E	HCAN	16	4
Message control 1[8]	MC1[8]	8	H'F82F	HCAN	16	4
Message control 2[1]	MC2[1]	8	H'F830	HCAN	16	4
Message control 2[2]	MC2[2]	8	H'F831	HCAN	16	4
Message control 2[3]	MC2[3]	8	H'F832	HCAN	16	4
Message control 2[4]	MC2[4]	8	H'F833	HCAN	16	4
Message control 2[5]	MC2[5]	8	H'F834	HCAN	16	4
Message control 2[6]	MC2[6]	8	H'F835	HCAN	16	4
Message control 2[7]	MC2[7]	8	H'F836	HCAN	16	4
Message control 2[8]	MC2[8]	8	H'F837	HCAN	16	4
Message control 3[1]	MC3[1]	8	H'F838	HCAN	16	4
Message control 3[2]	MC3[2]	8	H'F839	HCAN	16	4
Message control 3[3]	MC3[3]	8	H'F83A	HCAN	16	4
Message control 3[4]	MC3[4]	8	H'F83B	HCAN	16	4
Message control 3[5]	MC3[5]	8	H'F83C	HCAN	16	4
Message control 3[6]	MC3[6]	8	H'F83D	HCAN	16	4
Message control 3[7]	MC3[7]	8	H'F83E	HCAN	16	4
Message control 3[8]	MC3[8]	8	H'F83F	HCAN	16	4
Message control 4[1]	MC4[1]	8	H'F840	HCAN	16	4
Message control 4[2]	MC4[2]	8	H'F841	HCAN	16	4
Message control 4[3]	MC4[3]	8	H'F842	HCAN	16	4
Message control 4[4]	MC4[4]	8	H'F843	HCAN	16	4
Message control 4[5]	MC4[5]	8	H'F844	HCAN	16	4
Message control 4[6]	MC4[6]	8	H'F845	HCAN	16	4
Message control 4[7]	MC4[7]	8	H'F846	HCAN	16	4
Message control 4[8]	MC4[8]	8	H'F847	HCAN	16	4
Message control 5[1]	MC5[1]	8	H'F848	HCAN	16	4
Message control 5[2]	MC5[2]	8	H'F849	HCAN	16	4

Register Name	Abbrevia- tion	Bit No.	Address*	Module	Data Width	Access State
Message control 5[3]	MC5[3]	8	H'F84A	HCAN	16	4
Message control 5[4]	MC5[4]	8	H'F84B	HCAN	16	4
Message control 5[5]	MC5[5]	8	H'F84C	HCAN	16	4
Message control 5[6]	MC5[6]	8	H'F84D	HCAN	16	4
Message control 5[7]	MC5[7]	8	H'F84E	HCAN	16	4
Message control 5[8]	MC5[8]	8	H'F84F	HCAN	16	4
Message control 6[1]	MC6[1]	8	H'F850	HCAN	16	4
Message control 6[2]	MC6[2]	8	H'F851	HCAN	16	4
Message control 6[3]	MC6[3]	8	H'F852	HCAN	16	4
Message control 6[4]	MC6[4]	8	H'F853	HCAN	16	4
Message control 6[5]	MC6[5]	8	H'F854	HCAN	16	4
Message control 6[6]	MC6[6]	8	H'F855	HCAN	16	4
Message control 6[7]	MC6[7]	8	H'F856	HCAN	16	4
Message control 6[8]	MC6[8]	8	H'F857	HCAN	16	4
Message control 7[1]	MC7[1]	8	H'F858	HCAN	16	4
Message control 7[2]	MC7[2]	8	H'F859	HCAN	16	4
Message control 7[3]	MC7[3]	8	H'F85A	HCAN	16	4
Message control 7[4]	MC7[4]	8	H'F85B	HCAN	16	4
Message control 7[5]	MC7[5]	8	H'F85C	HCAN	16	4
Message control 7[6]	MC7[6]	8	H'F85D	HCAN	16	4
Message control 7[7]	MC7[7]	8	H'F85E	HCAN	16	4
Message control 7[8]	MC7[8]	8	H'F85F	HCAN	16	4
Message control 8[1]	MC8[1]	8	H'F860	HCAN	16	4
Message control 8[2]	MC8[2]	8	H'F861	HCAN	16	4
Message control 8[3]	MC8[3]	8	H'F862	HCAN	16	4
Message control 8[4]	MC8[4]	8	H'F863	HCAN	16	4
Message control 8[5]	MC8[5]	8	H'F864	HCAN	16	4
Message control 8[6]	MC8[6]	8	H'F865	HCAN	16	4
Message control 8[7]	MC8[7]	8	H'F866	HCAN	16	4
Message control 8[8]	MC8[8]	8	H'F867	HCAN	16	4
Message control 9[1]	MC9[1]	8	H'F868	HCAN	16	4
Message control 9[2]	MC9[2]	8	H'F869	HCAN	16	4
Message control 9[3]	MC9[3]	8	H'F86A	HCAN	16	4

Register Name	Abbrevia- tion	Bit No.	Address*	Module	Data Width	Access State
Message control 9[4]	MC9[4]	8	H'F86B	HCAN	16	4
Message control 9[5]	MC9[5]	8	H'F86C	HCAN	16	4
Message control 9[6]	MC9[6]	8	H'F86D	HCAN	16	4
Message control 9[7]	MC9[7]	8	H'F86E	HCAN	16	4
Message control 9[8]	MC9[8]	8	H'F86F	HCAN	16	4
Message control 10[1]	MC10[1]	8	H'F870	HCAN	16	4
Message control 10[2]	MC10[2]	8	H'F871	HCAN	16	4
Message control 10[3]	MC10[3]	8	H'F872	HCAN	16	4
Message control 10[4]	MC10[4]	8	H'F873	HCAN	16	4
Message control 10[5]	MC10[5]	8	H'F874	HCAN	16	4
Message control 10[6]	MC10[6]	8	H'F875	HCAN	16	4
Message control 10[7]	MC10[7]	8	H'F876	HCAN	16	4
Message control 10[8]	MC10[8]	8	H'F877	HCAN	16	4
Message control 11[1]	MC11[1]	8	H'F878	HCAN	16	4
Message control 11[2]	MC11[2]	8	H'F879	HCAN	16	4
Message control 11[3]	MC11[3]	8	H'F87A	HCAN	16	4
Message control 11[4]	MC11[4]	8	H'F87B	HCAN	16	4
Message control 11[5]	MC11[5]	8	H'F87C	HCAN	16	4
Message control 11[6]	MC11[6]	8	H'F87D	HCAN	16	4
Message control 11[7]	MC11[7]	8	H'F87E	HCAN	16	4
Message control 11[8]	MC11[8]	8	H'F87F	HCAN	16	4
Message control 12[1]	MC12[1]	8	H'F880	HCAN	16	4
Message control 12[2]	MC12[2]	8	H'F881	HCAN	16	4
Message control 12[3]	MC12[3]	8	H'F882	HCAN	16	4
Message control 12[4]	MC12[4]	8	H'F883	HCAN	16	4
Message control 12[5]	MC12[5]	8	H'F884	HCAN	16	4
Message control 12[6]	MC12[6]	8	H'F885	HCAN	16	4
Message control 12[7]	MC12[7]	8	H'F886	HCAN	16	4
Message control 12[8]	MC12[8]	8	H'F887	HCAN	16	4
Message control 13[1]	MC13[1]	8	H'F888	HCAN	16	4
Message control 13[2]	MC13[2]	8	H'F889	HCAN	16	4
Message control 13[3]	MC13[3]	8	H'F88A	HCAN	16	4
Message control 13[4]	MC13[4]	8	H'F88B	HCAN	16	4

Register Name	Abbrevia- tion	Bit No.	Address*	Module	Data Width	Access State
Message control 13[5]	MC13[5]	8	H'F88C	HCAN	16	4
Message control 13[6]	MC13[6]	8	H'F88D	HCAN	16	4
Message control 13[7]	MC13[7]	8	H'F88E	HCAN	16	4
Message control 13[8]	MC13[8]	8	H'F88F	HCAN	16	4
Message control 14[1]	MC14[1]	8	H'F890	HCAN	16	4
Message control 14[2]	MC14[2]	8	H'F891	HCAN	16	4
Message control 14[3]	MC14[3]	8	H'F892	HCAN	16	4
Message control 14[4]	MC14[4]	8	H'F893	HCAN	16	4
Message control 14[5]	MC14[5]	8	H'F894	HCAN	16	4
Message control 14[6]	MC14[6]	8	H'F895	HCAN	16	4
Message control 14[7]	MC14[7]	8	H'F896	HCAN	16	4
Message control 14[8]	MC14[8]	8	H'F897	HCAN	16	4
Message control 15[1]	MC15[1]	8	H'F898	HCAN	16	4
Message control 15[2]	MC15[2]	8	H'F899	HCAN	16	4
Message control 15[3]	MC15[3]	8	H'F89A	HCAN	16	4
Message control 15[4]	MC15[4]	8	H'F89B	HCAN	16	4
Message control 15[5]	MC15[5]	8	H'F89C	HCAN	16	4
Message control 15[6]	MC15[6]	8	H'F89D	HCAN	16	4
Message control 15[7]	MC15[7]	8	H'F89E	HCAN	16	4
Message control 15[8]	MC15[8]	8	H'F89F	HCAN	16	4
Message data 0[1]	MD0[1]	8	H'F8B0	HCAN	16	4
Message data 0[2]	MD0[2]	8	H'F8B1	HCAN	16	4
Message data 0[3]	MD0[3]	8	H'F8B2	HCAN	16	4
Message data 0[4]	MD0[4]	8	H'F8B3	HCAN	16	4
Message data 0[5]	MD0[5]	8	H'F8B4	HCAN	16	4
Message data 0[6]	MD0[6]	8	H'F8B5	HCAN	16	4
Message data 0[7]	MD0[7]	8	H'F8B6	HCAN	16	4
Message data 0[8]	MD0[8]	8	H'F8B7	HCAN	16	4
Message data 1[1]	MD1[1]	8	H'F8B8	HCAN	16	4
Message data 1[2]	MD1[2]	8	H'F8B9	HCAN	16	4
Message data 1[3]	MD1[3]	8	H'F8BA	HCAN	16	4
Message data 1[4]	MD1[4]	8	H'F8BB	HCAN	16	4
Message data 1[5]	MD1[5]	8	H'F8BC	HCAN	16	4

Register Name	Abbrevia- tion	Bit No.	Address*	Module	Data Width	Access State
Message data 1[6]	MD1[6]	8	H'F8BD	HCAN	16	4
Message data 1[7]	MD1[7]	8	H'F8BE	HCAN	16	4
Message data 1[8]	MD1[8]	8	H'F8BF	HCAN	16	4
Message data 2[1]	MD2[1]	8	H'F8C0	HCAN	16	4
Message data 2[2]	MD2[2]	8	H'F8C1	HCAN	16	4
Message data 2[3]	MD2[3]	8	H'F8C2	HCAN	16	4
Message data 2[4]	MD2[4]	8	H'F8C3	HCAN	16	4
Message data 2[5]	MD2[5]	8	H'F8C4	HCAN	16	4
Message data 2[6]	MD2[6]	8	H'F8C5	HCAN	16	4
Message data 2[7]	MD2[7]	8	H'F8C6	HCAN	16	4
Message data 2[8]	MD2[8]	8	H'F8C7	HCAN	16	4
Message data 3[1]	MD3[1]	8	H'F8C8	HCAN	16	4
Message data 3[2]	MD3[2]	8	H'F8C9	HCAN	16	4
Message data 3[3]	MD3[3]	8	H'F8CA	HCAN	16	4
Message data 3[4]	MD3[4]	8	H'F8CB	HCAN	16	4
Message data 3[5]	MD3[5]	8	H'F8CC	HCAN	16	4
Message data 3[6]	MD3[6]	8	H'F8CD	HCAN	16	4
Message data 3[7]	MD3[7]	8	H'F8CE	HCAN	16	4
Message data 3[8]	MD3[8]	8	H'F8CF	HCAN	16	4
Message data 4[1]	MD4[1]	8	H'F8D0	HCAN	16	4
Message data 4[2]	MD4[2]	8	H'F8D1	HCAN	16	4
Message data 4[3]	MD4[3]	8	H'F8D2	HCAN	16	4
Message data 4[4]	MD4[4]	8	H'F8D3	HCAN	16	4
Message data 4[5]	MD4[5]	8	H'F8D4	HCAN	16	4
Message data 4[6]	MD4[6]	8	H'F8D5	HCAN	16	4
Message data 4[7]	MD4[7]	8	H'F8D6	HCAN	16	4
Message data 4[8]	MD4[8]	8	H'F8D7	HCAN	16	4
Message data 5[1]	MD5[1]	8	H'F8D8	HCAN	16	4
Message data 5[2]	MD5[2]	8	H'F8D9	HCAN	16	4
Message data 5[3]	MD5[3]	8	H'F8DA	HCAN	16	4
Message data 5[4]	MD5[4]	8	H'F8DB	HCAN	16	4
Message data 5[5]	MD5[5]	8	H'F8DC	HCAN	16	4
Message data 5[6]	MD5[6]	8	H'F8DD	HCAN	16	4

Register Name	Abbrevia-tion	Bit No.	Address*	Module	Data Width	Access State
Message data 5[7]	MD5[7]	8	H'F8DE	HCAN	16	4
Message data 5[8]	MD5[8]	8	H'F8DF	HCAN	16	4
Message data 6[1]	MD6[1]	8	H'F8E0	HCAN	16	4
Message data 6[2]	MD6[2]	8	H'F8E1	HCAN	16	4
Message data 6[3]	MD6[3]	8	H'F8E2	HCAN	16	4
Message data 6[4]	MD6[4]	8	H'F8E3	HCAN	16	4
Message data 6[5]	MD6[5]	8	H'F8E4	HCAN	16	4
Message data 6[6]	MD6[6]	8	H'F8E5	HCAN	16	4
Message data 6[7]	MD6[7]	8	H'F8E6	HCAN	16	4
Message data 6[8]	MD6[8]	8	H'F8E7	HCAN	16	4
Message data 7[1]	MD7[1]	8	H'F8E8	HCAN	16	4
Message data 7[2]	MD7[2]	8	H'F8E9	HCAN	16	4
Message data 7[3]	MD7[3]	8	H'F8EA	HCAN	16	4
Message data 7[4]	MD7[4]	8	H'F8EB	HCAN	16	4
Message data 7[5]	MD7[5]	8	H'F8EC	HCAN	16	4
Message data 7[6]	MD7[6]	8	H'F8ED	HCAN	16	4
Message data 7[7]	MD7[7]	8	H'F8EE	HCAN	16	4
Message data 7[8]	MD7[8]	8	H'F8EF	HCAN	16	4
Message data 8[1]	MD8[1]	8	H'F8F0	HCAN	16	4
Message data 8[2]	MD8[2]	8	H'F8F1	HCAN	16	4
Message data 8[3]	MD8[3]	8	H'F8F2	HCAN	16	4
Message data 8[4]	MD8[4]	8	H'F8F3	HCAN	16	4
Message data 8[5]	MD8[5]	8	H'F8F4	HCAN	16	4
Message data 8[6]	MD8[6]	8	H'F8F5	HCAN	16	4
Message data 8[7]	MD8[7]	8	H'F8F6	HCAN	16	4
Message data 8[8]	MD8[8]	8	H'F8F7	HCAN	16	4
Message data 9[1]	MD9[1]	8	H'F8F8	HCAN	16	4
Message data 9[2]	MD9[2]	8	H'F8F9	HCAN	16	4
Message data 9[3]	MD9[3]	8	H'F8FA	HCAN	16	4
Message data 9[4]	MD9[4]	8	H'F8FB	HCAN	16	4
Message data 9[5]	MD9[5]	8	H'F8FC	HCAN	16	4
Message data 9[6]	MD9[6]	8	H'F8FD	HCAN	16	4
Message data 9[7]	MD9[7]	8	H'F8FE	HCAN	16	4
Message data 9[8]	MD9[8]	8	H'F8FF	HCAN	16	4

Register Name	Abbrevia- tion	Bit No.	Address*	Module	Data Width	Access State
Message data 10[1]	MD10[1]	8	H'F900	HCAN	16	4
Message data 10[2]	MD10[2]	8	H'F901	HCAN	16	4
Message data 10[3]	MD10[3]	8	H'F902	HCAN	16	4
Message data 10[4]	MD10[4]	8	H'F903	HCAN	16	4
Message data 10[5]	MD10[5]	8	H'F904	HCAN	16	4
Message data 10[6]	MD10[6]	8	H'F905	HCAN	16	4
Message data 10[7]	MD10[7]	8	H'F906	HCAN	16	4
Message data 10[8]	MD10[8]	8	H'F907	HCAN	16	4
Message data 11[1]	MD11[1]	8	H'F908	HCAN	16	4
Message data 11[2]	MD11[2]	8	H'F909	HCAN	16	4
Message data 11[3]	MD11[3]	8	H'F90A	HCAN	16	4
Message data 11[4]	MD11[4]	8	H'F90B	HCAN	16	4
Message data 11[5]	MD11[5]	8	H'F90C	HCAN	16	4
Message data 11[6]	MD11[6]	8	H'F90D	HCAN	16	4
Message data 11[7]	MD11[7]	8	H'F90E	HCAN	16	4
Message data 11[8]	MD11[8]	8	H'F90F	HCAN	16	4
Message data 12[1]	MD12[1]	8	H'F910	HCAN	16	4
Message data 12[2]	MD12[2]	8	H'F911	HCAN	16	4
Message data 12[3]	MD12[3]	8	H'F912	HCAN	16	4
Message data 12[4]	MD12[4]	8	H'F913	HCAN	16	4
Message data 12[5]	MD12[5]	8	H'F914	HCAN	16	4
Message data 12[6]	MD12[6]	8	H'F915	HCAN	16	4
Message data 12[7]	MD12[7]	8	H'F916	HCAN	16	4
Message data 12[8]	MD12[8]	8	H'F917	HCAN	16	4
Message data 13[1]	MD13[1]	8	H'F918	HCAN	16	4
Message data 13[2]	MD13[2]	8	H'F919	HCAN	16	4
Message data 13[3]	MD13[3]	8	H'F91A	HCAN	16	4
Message data 13[4]	MD13[4]	8	H'F91B	HCAN	16	4
Message data 13[5]	MD13[5]	8	H'F91C	HCAN	16	4
Message data 13[6]	MD13[6]	8	H'F91D	HCAN	16	4
Message data 13[7]	MD13[7]	8	H'F91E	HCAN	16	4
Message data 13[8]	MD13[8]	8	H'F91F	HCAN	16	4
Message data 14[1]	MD14[1]	8	H'F920	HCAN	16	4

Register Name	Abbrevia- tion	Bit No.	Address*	Module	Data Width	Access State
Message data 14[2]	MD14[2]	8	H'F921	HCAN	16	4
Message data 14[3]	MD14[3]	8	H'F922	HCAN	16	4
Message data 14[4]	MD14[4]	8	H'F923	HCAN	16	4
Message data 14[5]	MD14[5]	8	H'F924	HCAN	16	4
Message data 14[6]	MD14[6]	8	H'F925	HCAN	16	4
Message data 14[7]	MD14[7]	8	H'F926	HCAN	16	4
Message data 14[8]	MD14[8]	8	H'F927	HCAN	16	4
Message data 15[1]	MD15[1]	8	H'F928	HCAN	16	4
Message data 15[2]	MD15[2]	8	H'F929	HCAN	16	4
Message data 15[3]	MD15[3]	8	H'F92A	HCAN	16	4
Message data 15[4]	MD15[4]	8	H'F92B	HCAN	16	4
Message data 15[5]	MD15[5]	8	H'F92C	HCAN	16	4
Message data 15[6]	MD15[6]	8	H'F92D	HCAN	16	4
Message data 15[7]	MD15[7]	8	H'F92E	HCAN	16	4
Message data 15[8]	MD15[8]	8	H'F92F	HCAN	16	4
HCAN monitor register	HCANMON	8	H'FA00	HCAN	16	4
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
Low-power control register	LPWRCR	8	H'FDEC	SYSTEM	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
Port 1 data direction register	P1DDR	8	H'FE30	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

Register Name	Abbrevia- tion	Bit No.	Address*	Module	Data Width	Access State
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 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
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 _3	TCNT_3	16	H'FE86	TPU_3	16	2
Timer general register A_3	TGRA_3	16	H'FE88	TPU_3	16	2
Timer general register B_3	TGRB_3	16	H'FE8A	TPU_3	16	2
Timer general register C_3	TGRC_3	16	H'FE8C	TPU_3	16	2
Timer general register D_3	TGRD_3	16	H'FE8E	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_4	TCNT_4	16	H'FE96	TPU_4	16	2
Timer general register A_4	TGRA_4	16	H'FE98	TPU_4	16	2
Timer general register B_4	TGRB_4	16	H'FE9A	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_5	TCNT_5	16	H'FEA6	TPU_5	16	2

Register Name	Abbrevia-tion	Bit No.	Address*	Module	Data Width	Access State
Timer general register A_5	TGRA_5	16	H'FEA8	TPU_5	16	2
Timer general register B_5	TGRB_5	16	H'FEAA	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 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 M	IPRM	8	H'FECC	INT	8	2
RAM emulation register	RAMER	8	H'FEDB	ROM	8	2
Port 1 data register	P1DR	8	H'FF00	PORT	8	2
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 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_0	TCNT_0	16	H'FF16	TPU_0	16	2
Timer general register A_0	TGRA_0	16	H'FF18	TPU_0	16	2
Timer general register B_0	TGRB_0	16	H'FF1A	TPU_0	16	2
Timer general register C_0	TGRC_0	16	H'FF1C	TPU_0	16	2

Register Name	Abbrevia-tion	Bit No.	Address*	Module	Data Width	Access State
Timer general register D_0	TGRD_0	16	H'FF1E	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_1	TCNT_1	16	H'FF26	TPU_1	16	2
Timer general register A_1	TGRA_1	16	H'FF28	TPU_1	16	2
Timer general register B_1	TGRB_1	16	H'FF2A	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_2	TCNT_2	16	H'FF36	TPU_2	16	2
Timer general register A_2	TGRA_2	16	H'FF38	TPU_2	16	2
Timer general register B_2	TGRB_2	16	H'FF3A	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
Bit rate register_0	BRR_0	8	H'FF79	SCI_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
Serial mode register_1	SMR_1	8	H'FF80	SCI_1	8	2
Bit rate register_1	BRR_1	8	H'FF81	SCI_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

Register Name	Abbrevia- tion	Bit No.	Address*	Module	Data Width	Access State
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
Flash memory control register 1	FLMCR1	8	H'FFA8	ROM	8	2
Flash memory control register 2	FLMCR2	8	H'FFA9	ROM	8	2
Erase block register 1	EBR1	8	H'FFAA	ROM	8	2
Flash memory power control register	FLPWCR	8	H'FFAC	ROM	8	2
Port 1 register	PORT1	8	H'FFB0	PORT	8	2
Port 4 register	PORT4	8	H'FFB3	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: Lower 16 bits of the address.

17.2 Register Bits

Register bit names of the on-chip peripheral modules are described below.

Each line covers eight bits, and 16-bit registers are shown as 2 lines.

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MCR	MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0	HCAN
GSR	—	—	—	—	GSR3	GSR2	GSR1	GSR0	
BCR	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0	
	BCR15	BCR14	BCR13	BCR12	BCR11	BCR10	BCR9	BCR8	
MBCR	MBCR7	MBCR6	MBCR5	MBCR4	MBCR3	MBCR2	MBCR1	—	
	MBCR15	MBCR14	MBCR13	MBCR12	MBCR11	MBCR10	MBCR9	MBCR8	
TXPR	TXPR7	TXPR6	TXPR5	TXPR4	TXPR3	TXPR2	TXPR1	—	
	TXPR15	TXPR14	TXPR13	TXPR12	TXPR11	TXPR10	TXPR9	TXPR8	
TXCR	TXCR7	TXCR6	TXCR5	TXCR4	TXCR3	TXCR2	TXCR1	—	
	TXCR15	TXCR14	TXCR13	TXCR12	TXCR11	TXCR10	TXCR9	TXCR8	
TXACK	TXACK7	TXACK6	TXACK5	TXACK4	TXACK3	TXACK2	TXACK1	—	
	TXACK15	TXACK14	TXACK13	TXACK12	TXACK11	TXACK10	TXACK9	TXACK8	
ABACK	ABACK7	ABACK6	ABACK5	ABACK4	ABACK3	ABACK2	ABACK1	—	
	ABACK15	ABACK14	ABACK13	ABACK12	ABACK11	ABACK10	ABACK9	ABACK8	
RXPR	RXPR7	RXPR6	RXPR5	RXPR4	RXPR3	RXPR2	RXPR1	RXPR0	
	RXPR15	RXPR14	RXPR13	RXPR12	RXPR11	RXPR10	RXPR9	RXPR8	
RFPR	RFPR7	RFPR6	RFPR5	RFPR4	RFPR3	RFPR2	RFPR1	RFPR0	
	RFPR15	RFPR14	RFPR13	RFPR12	RFPR11	RFPR10	RFPR9	RFPR8	
IRR	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0	
	—	—	—	IRR12	—	—	IRR9	IRR8	
MBIMR	MBIMR7	MBIMR6	MBIMR5	MBIMR4	MBIMR3	MBIMR2	MBIMR1	MBIMR0	
	MBIMR15	MBIMR14	MBIMR13	MBIMR12	MBIMR11	MBIMR10	MBIMR9	MBIMR8	
IMR	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	—	
	—	—	—	IMR12	—	—	IMR9	IMR8	
REC	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TEC	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
UMSR	UMSR7	UMSR6	UMSR5	UMSR4	UMSR3	UMSR2	UMSR1	UMSR0	
	UMSR15	UMSR14	UMSR13	UMSR12	UMSR11	UMSR10	UMSR9	UMSR8	
LAFML	LAFML7	LAFML6	LAFML5	LAFML4	LAFML3	LAFML2	LAFML1	LAFML0	
	LAFML15	LAFML14	LAFML13	LAFML12	LAFML11	LAFML10	LAFML9	LAFML8	
LAFMH	LAFMH7	LAFMH6	LAFMH5	—	—	—	LAFMH1	LAFMH0	
	LAFMH15	LAFMH14	LAFMH13	LAFMH12	LAFMH11	LAFMH10	LAFMH9	LAFMH8	
MC0[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC0[2]	—	—	—	—	—	—	—	—	
MC0[3]	—	—	—	—	—	—	—	—	
MC0[4]	—	—	—	—	—	—	—	—	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MC0[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	HCAN
MC0[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC0[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC0[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC1[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC1[2]	—	—	—	—	—	—	—	—	
MC1[3]	—	—	—	—	—	—	—	—	
MC1[4]	—	—	—	—	—	—	—	—	
MC1[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC1[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC1[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC1[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC2[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC2[2]	—	—	—	—	—	—	—	—	
MC2[3]	—	—	—	—	—	—	—	—	
MC2[4]	—	—	—	—	—	—	—	—	
MC2[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC2[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC2[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC2[8]	ID-15	ID-14	ID-13	ID12	ID-11	ID-10	ID-9	ID-8	
MC3[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC3[2]	—	—	—	—	—	—	—	—	
MC3[3]	—	—	—	—	—	—	—	—	
MC3[4]	—	—	—	—	—	—	—	—	
MC3[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC3[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC3[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC3[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC4[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC4[2]	—	—	—	—	—	—	—	—	
MC4[3]	—	—	—	—	—	—	—	—	
MC4[4]	—	—	—	—	—	—	—	—	
MC4[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC4[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC4[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC4[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC5[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC5[2]	—	—	—	—	—	—	—	—	
MC5[3]	—	—	—	—	—	—	—	—	
MC5[4]	—	—	—	—	—	—	—	—	
MC5[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC5[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MC5[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	HCAN
MC5[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC6[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC6[2]	—	—	—	—	—	—	—	—	
MC6[3]	—	—	—	—	—	—	—	—	
MC6[4]	—	—	—	—	—	—	—	—	
MC6[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC6[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC6[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC6[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC7[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC7[2]	—	—	—	—	—	—	—	—	
MC7[3]	—	—	—	—	—	—	—	—	
MC7[4]	—	—	—	—	—	—	—	—	
MC7[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC7[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC7[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC7[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC8[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC8[2]	—	—	—	—	—	—	—	—	
MC8[3]	—	—	—	—	—	—	—	—	
MC8[4]	—	—	—	—	—	—	—	—	
MC8[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC8[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC8[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC8[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC9[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC9[2]	—	—	—	—	—	—	—	—	
MC9[3]	—	—	—	—	—	—	—	—	
MC9[4]	—	—	—	—	—	—	—	—	
MC9[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC9[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC9[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC9[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC10[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC10[2]	—	—	—	—	—	—	—	—	
MC10[3]	—	—	—	—	—	—	—	—	
MC10[4]	—	—	—	—	—	—	—	—	
MC10[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC10[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC10[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC10[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MC11[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN
MC11[2]	—	—	—	—	—	—	—	—	
MC11[3]	—	—	—	—	—	—	—	—	
MC11[4]	—	—	—	—	—	—	—	—	
MC11[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC11[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC11[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC11[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC12[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC12[2]	—	—	—	—	—	—	—	—	
MC12[3]	—	—	—	—	—	—	—	—	
MC12[4]	—	—	—	—	—	—	—	—	
MC12[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC12[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC12[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC12[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC13[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC13[2]	—	—	—	—	—	—	—	—	
MC13[3]	—	—	—	—	—	—	—	—	
MC13[4]	—	—	—	—	—	—	—	—	
MC13[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC13[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC13[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC13[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC14[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC14[2]	—	—	—	—	—	—	—	—	
MC14[3]	—	—	—	—	—	—	—	—	
MC14[4]	—	—	—	—	—	—	—	—	
MC14[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC14[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC14[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC14[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC15[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC15[2]	—	—	—	—	—	—	—	—	
MC15[3]	—	—	—	—	—	—	—	—	
MC15[4]	—	—	—	—	—	—	—	—	
MC15[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC15[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC15[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC15[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MD0[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD0[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MD0[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	HCAN
MD0[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD0[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD0[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD0[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD0[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD1[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD1[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD1[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD1[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD1[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD1[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD1[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD1[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD2[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD2[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD2[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD2[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD2[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD2[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD2[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD2[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD3[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD3[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD3[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD3[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD3[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD3[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD3[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD3[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD4[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD4[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD4[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD4[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD4[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD4[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD4[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD4[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD5[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD5[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD5[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD5[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MD5[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	HCAN
MD5[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD5[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD5[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD6[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD6[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD6[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD6[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD6[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD6[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD6[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD6[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD7[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD7[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD7[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD7[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD7[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD7[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD7[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD7[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD8[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD8[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD8[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD8[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD8[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD8[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD8[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD8[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD9[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD9[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD9[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD9[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD9[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD9[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD9[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD9[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD10[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD10[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD10[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD10[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD10[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD10[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MD10[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	HCAN
MD10[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD11[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD11[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD11[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD11[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD11[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD11[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD11[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD11[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD12[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD12[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD12[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD12[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD12[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD12[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD12[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD12[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD13[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD13[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD13[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD13[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD13[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD13[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD13[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD13[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD14[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD14[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD14[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD14[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD14[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD14[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD14[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD14[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD15[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD15[2]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD15[3]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD15[4]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD15[5]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD15[6]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD15[7]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MD15[8]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
HCANMON	RxDIE	TxSTP	—	—	—	—	TxD	RxD	HCAN
SBYCR	SSBY	STS2	STS1	STS0	—	—	—	—	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	
LPWRCR	DTON	LSON	—	SUBSTP	RFCUT	—	STC1	STC0	
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	
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	PORT
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	—	—	—	—	
PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR	
PAPCR	—	—	—	—	PA3PCR	PA2PCR	PA1PCR	PA0PCR	
PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR	
PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR	
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	—	—	—	—	
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	
TCNT_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRC_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRD_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
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	
TCNT_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_5
TMDR_5	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_5	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_5	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_5	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	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	
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	
IPRM	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
RAMER	—	—	—	—	RAMS	RAM2	RAM1	RAM0	ROM
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	PORT
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	—	—	—	—	
PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_0
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRC_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRD_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_1
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_2
TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCSR_0	OVF	WT/$\overline{\text{IT}}$	TME	—	—	CKS2	CKS1	CKS0	WDT_0
TCNT_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
RSTCSR	WOVF	RSTE	RSTS	—	—	—	—	—	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SMR_0*3	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_0
(SMR_0*4)	(GM)	(BLK)	(PE)	(O/$\overline{E}$)	(BCP1)	(BCP0)	(CKS1)	(CKS0)	
BRR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_0*3	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
(SSR_0*4)	(TDRE)	(RDRF)	(ORER)	(ERS)	(PER)	(TEND)	(MPB)	(MPBT)	
RDR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_0	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_1*3	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_1
(SMR_1*4)	(GM)	(BLK)	(PE)	(O/$\overline{E}$)	(BCP1)	(BCP0)	(CKS1)	(CKS0)	
BRR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_1*3	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
(SSR_1*4)	(TDRE)	(RDRF)	(ORER)	(ERS)	(PER)	(TEND)	(MPB)	(MPBT)	
RDR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_2*3	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_2
(SMR_2*4)	(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*3	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
(SSR_2*4)	(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	
ADDRBL	AD1	AD0	—	—	—	—	—	—	
ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRCL	AD1	AD0	—	—	—	—	—	—	
ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRDL	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	SCAN	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	
FLMCR1	FWE	SWE	ESU1	PSU1	EV1	PV1	E1	P1	ROM
FLMCR2	FLER	—	—	—	—	—	—	—	
EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
FLPWCR	PDWND	—	—	—	—	—	—	—	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PORT1	P17	P16	P15	P14	P13	P12	P11	P10	PORT
PORT4	P47	P46	P45	P44	P43	P42	P41	P40	
PORT9	P97	P96	P95	P94	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	—	—	—	—	
PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	

Notes: 1. For buffer operation.

2. For free operation.

3. Normal serial communication interface mode.

4. Smart Card interface mode.

Some bit functions of SMR differ in normal serial communication interface mode and Smart Card interface mode.

17.3 Register States in Each Operating Mode

Register Name	Reset	High-speed	Medium-speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
MCR	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	HCAN
GSR	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
BCR	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MBCR	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
TXPR	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
TXCR	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
TXACK	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ABACK	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
RXPR	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
RFPR	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IRR	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MBIMR	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IMR	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
REC	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
TEC	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
UMSR	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
LAFML	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
LAFMH	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC3[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC3[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC3[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC3[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC3[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	

Register Name	Reset	High-speed	Medium-speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
MC3[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	HCAN
MC3[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC3[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC4[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC4[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC4[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC4[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC4[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC4[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC4[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC4[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC8[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC8[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC8[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC8[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC8[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC8[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC8[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC8[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC9[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC9[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC9[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC9[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC9[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC9[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC9[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	

Register Name	Reset	High-speed	Medium-speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
MC9[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	HCAN
MC10[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC10[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC10[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC10[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC10[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC10[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC10[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC10[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC12[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC12[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC12[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC12[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC12[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC12[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC12[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC12[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD0[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	

Register Name	Reset	High-speed	Medium-speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
MD0[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	HCAN
MD0[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD0[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD0[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD0[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD0[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD0[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD4[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD4[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD4[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD4[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD4[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD4[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD4[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD4[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD5[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD5[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD5[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD5[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD5[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD5[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD5[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD5[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD6[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD6[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD6[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	

Register Name	Reset	High-speed	Medium-speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
MD6[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	HCAN
MD6[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD6[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD6[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD6[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD9[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD9[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD9[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD9[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD9[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD9[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD9[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD9[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD12[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD12[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD12[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD12[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD12[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	

Register Name	Reset	High-speed	Medium-speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
MD12[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	HCAN
MD12[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD12[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD13[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD13[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD13[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD13[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD13[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD13[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD13[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD13[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[1]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[2]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[3]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[4]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[5]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[6]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[7]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[8]	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
HCANMON	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SBYCR	Initialized	–	–	–	–	–	–	–	–	Initialized	SYSTEM
SYSCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
SCKCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
MDCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
MSTPCRA	Initialized	–	–	–	–	–	–	–	–	Initialized	
MSTPCRB	Initialized	–	–	–	–	–	–	–	–	Initialized	
MSTPCRC	Initialized	–	–	–	–	–	–	–	–	Initialized	
LPWRCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
ISCRH	Initialized	–	–	–	–	–	–	–	–	Initialized	INT
ISCRL	Initialized	–	–	–	–	–	–	–	–	Initialized	
IER	Initialized	–	–	–	–	–	–	–	–	Initialized	
ISR	Initialized	–	–	–	–	–	–	–	–	Initialized	
P1DDR	Initialized	–	–	–	–	–	–	–	–	Initialized	PORT
PADDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PBDDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PCDDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PDDDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PFDDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PAPCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PBPCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PCPCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PDPCR	Initialized	–	–	–	–	–	–	–	–	Initialized	

Register Name	Reset	High-speed	Medium-speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
PAODR	Initialized	–	–	–	–	–	–	–	–	Initialized	PORT
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	
TSR_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCNT_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRA_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRB_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRC_3	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRD_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	
TCNT_4	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRA_4	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRB_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	
TCNT_5	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRA_5	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRB_5	Initialized	–	–	–	–	–	–	–	–	Initialized	
TSTR	Initialized	–	–	–	–	–	–	–	–	Initialized	TPU common
TSYR	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRA	Initialized	–	–	–	–	–	–	–	–	Initialized	INT
IPRB	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRD	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRE	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRF	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRG	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRH	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRJ	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRK	Initialized	–	–	–	–	–	–	–	–	Initialized	
IPRM	Initialized	–	–	–	–	–	–	–	–	Initialized	
RAMER	Initialized	–	–	–	–	–	–	–	–	Initialized	ROM
P1DR	Initialized	–	–	–	–	–	–	–	–	Initialized	PORT
PADR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PBDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PCDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PDDR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PFDR	Initialized	–	–	–	–	–	–	–	–	Initialized	

Register Name	Reset	High- speed	Medium- speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
TCR_0	Initialized	–	–	–	–	–	–	–	–	Initialized	TPU_0
TMDR_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIORH_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIORL_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIER_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TSR_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCNT_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRA_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRB_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRC_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRD_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	TPU_1
TMDR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIOR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIER_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TSR_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCNT_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRA_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRB_1	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCR_2	Initialized	–	–	–	–	–	–	–	–	Initialized	TPU_2
TMDR_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIOR_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TIER_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TSR_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCNT_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRA_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TGRB_2	Initialized	–	–	–	–	–	–	–	–	Initialized	
TCSR_0	Initialized	–	–	–	–	–	–	–	–	Initialized	WDT_0
TCNT_0	Initialized	–	–	–	–	–	–	–	–	Initialized	
RSTCSR	Initialized	–	–	–	–	–	–	–	–	Initialized	
SMR_0	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	SCI_0
BRR_0	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SCR_0	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
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	Initialized	Initialized	Initialized	Initialized	Initialized	
SMR_1	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	SCI_1
BRR_1	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SCR_1	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
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	Initialized	Initialized	Initialized	Initialized	Initialized	
SMR_2	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	SCI_2
BRR_2	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SCR_2	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	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	

Register Name	Reset	High- speed	Medium- speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby	Module
SCMR_2	Initialized	–	–	–	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	SCI_2
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	
FLMCR1	Initialized	–	–	–	–	–	–	–	–	Initialized	ROM
FLMCR2	Initialized	–	–	–	–	–	–	–	–	Initialized	
EBR1	Initialized	–	–	–	–	–	–	–	–	Initialized	
FLPWCR	Initialized	–	–	–	–	–	–	–	–	Initialized	
PORT1	Initialized	–	–	–	–	–	–	–	–	Initialized	PORT
PORT4	Initialized	–	–	–	–	–	–	–	–	Initialized	
PORT9	Initialized	–	–	–	–	–	–	–	–	Initialized	
PORTA	Initialized	–	–	–	–	–	–	–	–	Initialized	
PORTB	Initialized	–	–	–	–	–	–	–	–	Initialized	
PORTC	Initialized	–	–	–	–	–	–	–	–	Initialized	
PORTD	Initialized	–	–	–	–	–	–	–	–	Initialized	
PORTF	Initialized	–	–	–	–	–	–	–	–	Initialized	

Note: – is not initialized.

18.1 Absolute Maximum Ratings

Table 18.1 lists the absolute maximum ratings.

Table 18.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	–0.3 to +7.0	V
Input voltage (XTAL, EXTAL)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (port 4 and 9)	V_{in}	–0.3 to AV_{CC} +0.3	V
Input voltage (except XTAL, EXTAL, port 4 and 9)	V_{in}	–0.3 to V_{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.

18.2 DC Characteristics

Table 18.2 lists the DC characteristics. Table 18.3 lists the permissible output currents.

Table 18.2 DC Characteristics

Conditions: V_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{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

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$	V_T^-	V_{CC} × 0.2	—	—	V	
		V_T^+	—	—	V_{CC} × 0.7	V	
		$V_T^+ - V_T^-$	V_{CC} × 0.05	—	—	V	
Input high voltage	$\overline{\text{RES}}$, $\overline{\text{STBY}}$, NMI, MD2 to MD0, FWE	V_{IH}	V_{CC} × 0.9	—	V_{CC} + 0.3	V	
	EXTAL		V_{CC} × 0.7	—	V_{CC} + 0.3	V	
	Port 1, A to D, F, HRxD		V_{CC} × 0.7	—	V_{CC} + 0.3	V	
	Port 4 and 9		AV_{CC} × 0.7	—	AV_{CC} + 0.3	V	
Input low voltage	$\overline{\text{RES}}$, $\overline{\text{STBY}}$, NMI, MD2 to MD0, FWE	V_{IL}	–0.3	—	V_{CC} × 0.1	V	
	EXTAL		–0.3	—	V_{CC} × 0.2	V	
	Port 1, A to D, F, HRxD		–0.3	—	V_{CC} × 0.2	V	
	Port 4, 9		–0.3	—	AV_{CC} × 0.2	V	
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			V_{CC} – 1.0	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
Input leakage current	$\overline{\text{RES}}$	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
	$\overline{\text{STBY}}$, NMI, MD2 to MD0, FWE, HRxD		—	—	1.0	µA	
	Port 4, 9		—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} – 0.5 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
MOS input pull-up current	Port A to D	$-I_P$	30	—	300	μA	V_{in} = 0 V
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V
	NMI		—	—	30	pF	f = 1 MHz
	All input pins except $\overline{RES}$ and NMI		—	—	15	pF	T_a = 25°C
Current consumption*2	Normal operation	I_{CC}*3	—	TBD V_{CC} = 5.0 V	TBD V_{CC} = 5.5 V	mA	f = 20 MHz
	Sleep mode		—	TBD V_{CC} = 5.0 V	TBD V_{CC} = 5.5 V	mA	f = 20 MHz
	All modules stopped		—	TBD	—	mA	f = 20 MHz, V_{CC} = 5.0 V (reference values)
	Medium-speed mode (ϕ/32)		—	TBD	—	mA	f = 20 MHz, V_{CC} = 5.0 V (reference values)
	Standby mode		—	TBD	TBD	μA	T_a ≤ 50°C
			—	—	TBD	μA	50°C < T_a
Analog power supply current	During A/D conversion	AI_{CC}	—	2.5	4.0	mA	AV_{CC} = 5.0 V
	Idle		—	—	5.0	μA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D converter is not used, do not leave the AV_{CC}, and AV_{SS} pins open. Apply a voltage between 4.0 V and 5.5 V to the AV_{CC} pin by connecting them to V_{CC}, for instance.

2. Current consumption values are for V_{IH} = V_{CC} (EXTAL), AV_{CC} (ports 4 and 9), or V_{CC} (other), and V_{IL} = 0 V, with all output pins unloaded and the on-chip MOS pull-up transistors in the off state.

3. I_{CC} depends on V_{CC} and f as follows:
I_{CC} (max.) = TBD (normal operation)
I_{CC} (max.) = TBD (sleep mode)

Table 18.3 Permissible Output Currents

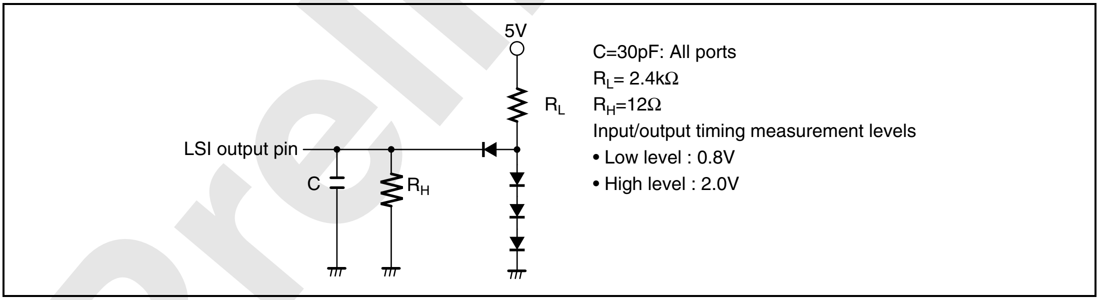
Conditions: V_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)*

Item			Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	All output pins	V_{CC} = 4.5 to 5.5 V	I_{OL}	—	—	10	mA
Permissible output low current (total)	Total of all output pins	V_{CC} = 4.5 to 5.5 V	$\sum I_{OL}$	—	—	100	mA
Permissible output high current (per pin)	All output pins	V_{CC} = 4.5 to 5.5 V	$-I_{OH}$	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	V_{CC} = 4.5 to 5.5 V	$\sum -I_{OH}$	—	—	30	mA

Note: * To protect chip reliability, do not exceed the output current values in table 18.3.

18.3 AC Characteristics

Figure 18.1 shows the test conditions for the AC characteristics.



C=30pF: All ports
R_L= 2.4kΩ
R_H=12Ω
Input/output timing measurement levels
- Low level : 0.8V
- High level : 2.0V

Figure 18.1 Output Load Circuit

18.3.1 Clock Timing

Table 18.4 lists the clock timing

Table 18.4 Clock Timing

Conditions : V_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ϕ = 4 MHz to 24 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
Clock cycle time	t_{cyc}	41.6	250	ns	Figure 18.2
Clock high pulse width	t_{CH}	TBD	—	ns	
Clock low pulse width	t_{CL}	TBD	—	ns	
Clock rise time	t_{Cr}	—	TBD	ns	
Clock fall time	t_{Cf}	—	TBD	ns	
Oscillation stabilization time at reset (crystal)	t_{OSC1}	20	—	ms	Figure 18.3
Oscillation stabilization time in software standby (crystal)	t_{OSC2}	8	—	ms	Figure 18.3
External clock output stabilization delay time	t_{DEXT}	2	—	ms	Figure 18.3

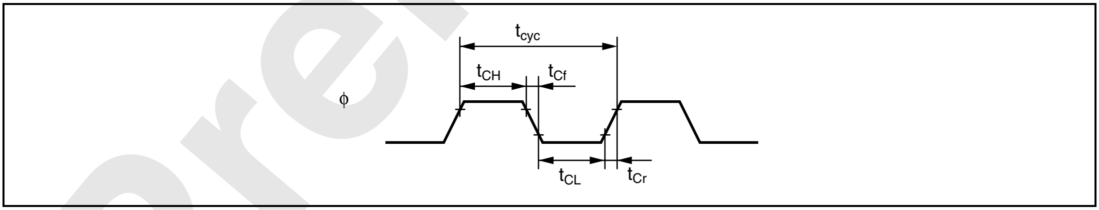


Figure 18.2 System Clock Timing

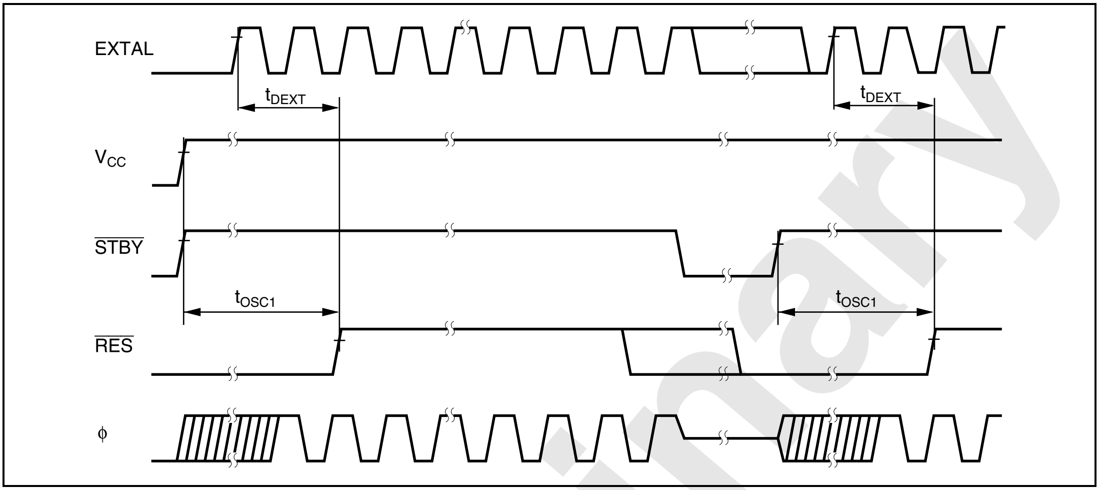


Figure 18.3 Oscillation Stabilization Timing

18.3.2 Control Signal Timing

Table 18.5 lists the control signal timing.

Table 18.5 Control Signal Timing

Conditions: V_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, φ = 4 MHz to 24 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figure 18.4
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 18.5
NMI hold time	t_{NMIH}	10	—	ns	
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	

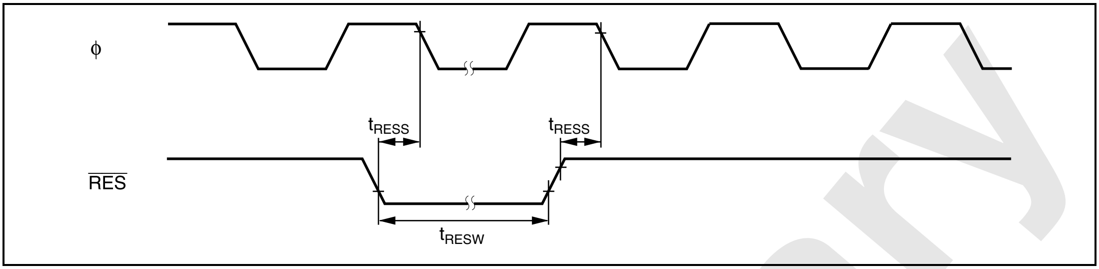


Figure 18.4 Reset Input Timing

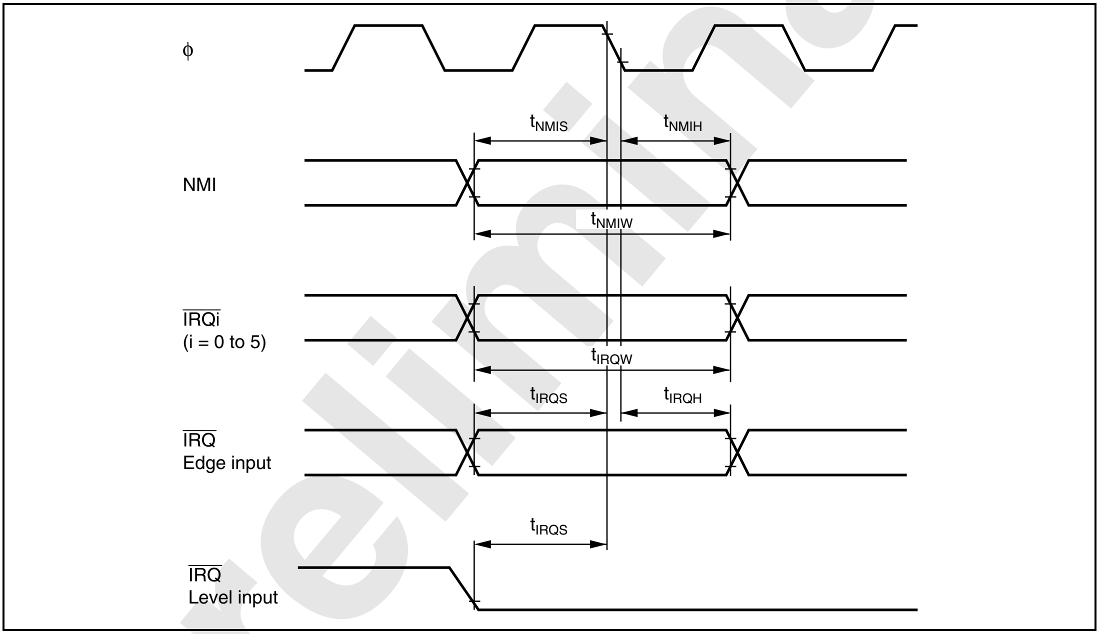


Figure 18.5 Interrupt Input Timing

18.3.3 Timing of On-Chip Peripheral Modules

Table 18.6 lists the timing of on-chip peripheral modules.

Table 18.6 Timing of On-Chip Peripheral Modules

Conditions: V_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ϕ = 4 MHz to 24 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item			Symbol	Min.	Max.	Unit	Test Conditions
I/O port	Output data delay time		t_{PWD}	—	40	ns	Figure 18.6
	Input data setup time		t_{PRS}	25	—		
	Input data hold time		t_{PRH}	25	—		
TPU	Timer output delay time		t_{TOCD}	—	40	ns	Figure 18.7
	Timer input setup time		t_{TICS}	25	—		
	Timer clock input setup time		t_{TCKS}	25	—	ns	Figure 18.8
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—		
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 18.9
		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}	—	40	ns	Figure 18.10
	Receive data setup time (synchronous)		t_{RXS}	40	—		
	Receive data hold time (synchronous)		t_{RXH}	40	—		

Item		Symbol	Min.	Max.	Unit	Test Conditions
A/D converter	Trigger input setup time	t_{TRGS}	30	—	ns	Figure 18.11
HCAN*	Transmit data delay time	t_{HTXD}	—	80	ns	Figure 18.12
	Transmit data setup time	t_{HRXS}	80	—		
	Transmit data hold time	t_{HRXH}	80	—		

Note: * The HCAN input signal is asynchronous. However, its state is judged to have changed at the rising-edge (two clock cycles) of the CK clock signal shown in figure 18.12. The HCAN output signal is also asynchronous. Its state changes based on the rising-edge (two clock cycles) of the CK clock signal shown in figure 18.12.

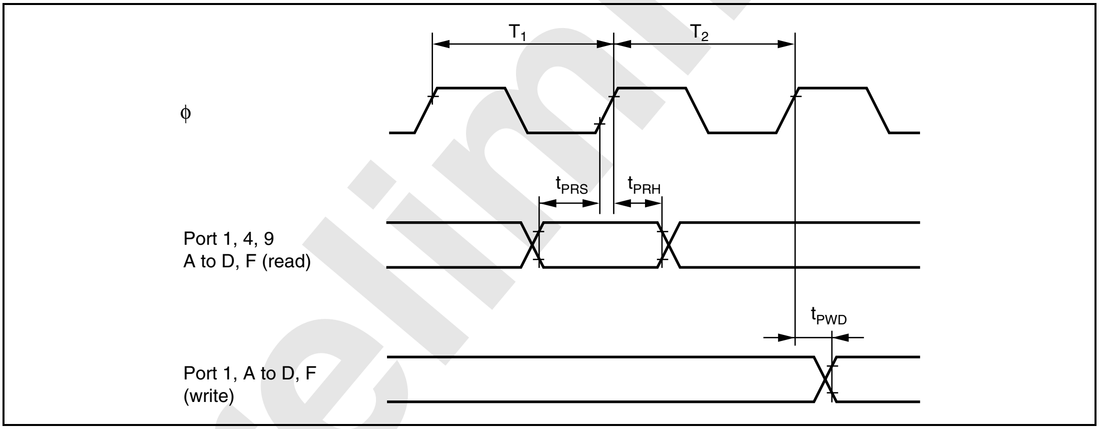
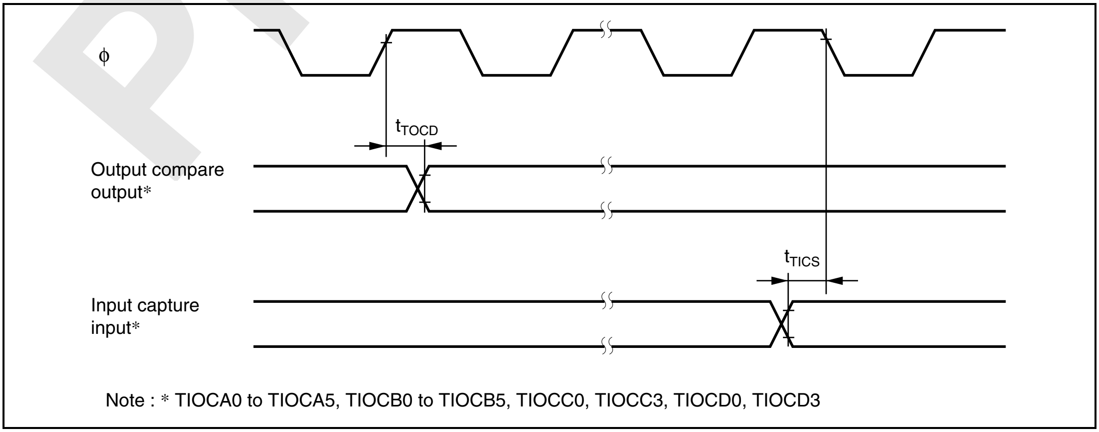


Figure 18.6 I/O Port Input/Output Timing



Note : * TIOCA0 to TIOCA5, TIOCB0 to TIOCB5, TIOCC0, TIOCC3, TIOCD0, TIOCD3

Figure 18.7 TPU Input/Output Timing

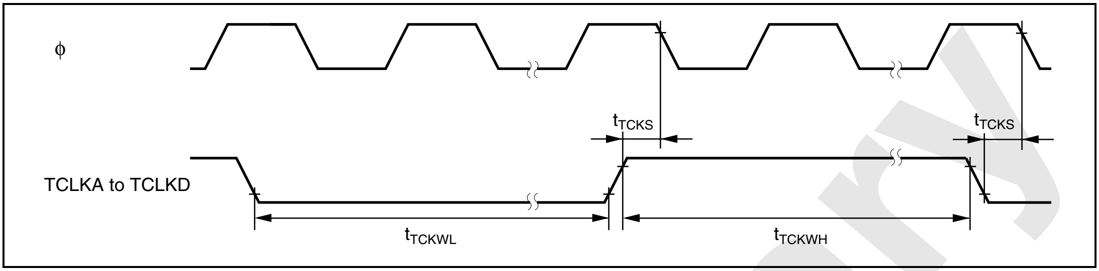


Figure 18.8 TPU Clock Input Timing

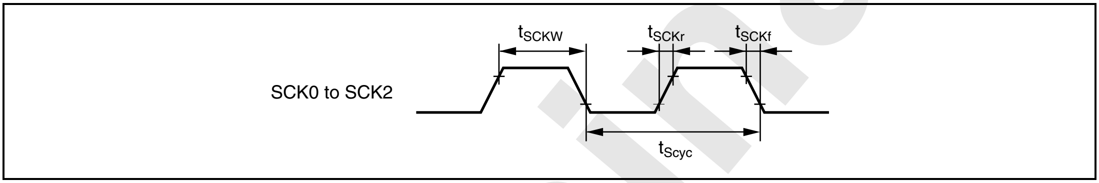


Figure 18.9 SCK Clock Input Timing

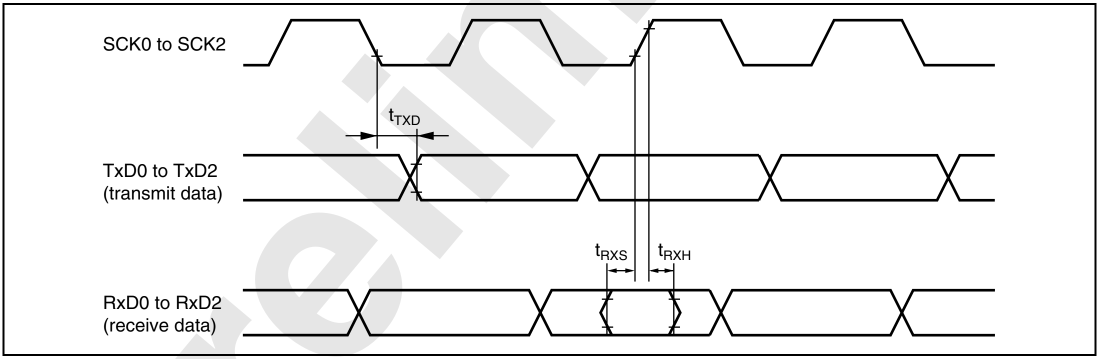


Figure 18.10 SCI Input/Output Timing (Clocked Synchronous Mode)

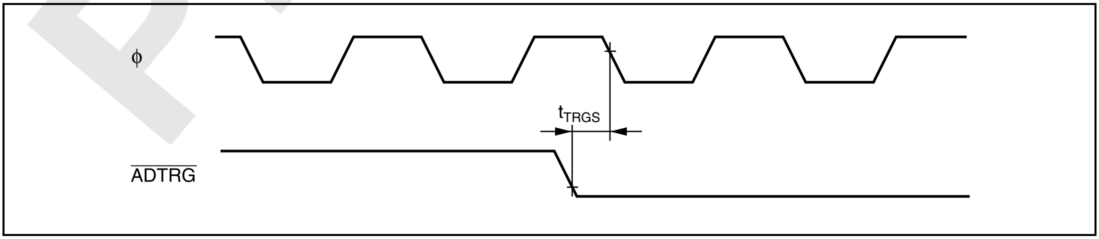


Figure 18.11 A/D Converter External Trigger Input Timing

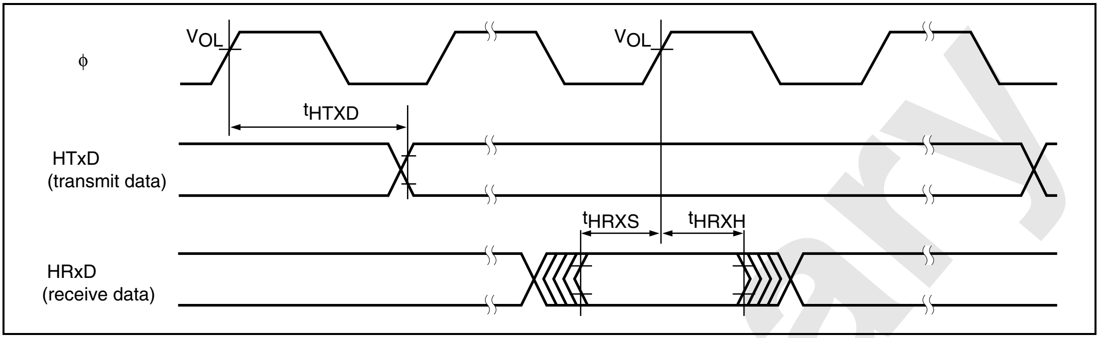


Figure 18.12 HCAN Input/Output Timing

18.4 A/D Conversion Characteristics

Table 18.7 lists the A/D conversion characteristics.

Table 18.7 A/D Conversion Characteristics

Conditions: V_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{SS} = AV_{SS} = 0V, φ = 4 MHz to 24 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Min.	Typ.	Max.	Unit
Resolution	10	10	10	bits
Conversion time	10	—	200	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

18.5 Flash Memory Characteristics

Table 18.8 lists the flash memory characteristics.

Table 18.8 Flash Memory Characteristics

Conditions: V_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V,
T_a = 0 to +75°C (Programming/erasing operating temperature range)

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 PSU1 bit setting*1	t_{spsu}	50	50	—	µs	
	Wait time after P1 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 P1 bit clear*1	t_{cp}	5	5	—	µs	
	Wait time after PSU1 bit clear*1	t_{cpsu}	5	5	—	µs	
	Wait time after PV1 bit setting*1	t_{spv}	4	4	—	µs	
	Wait time after H'FF dummy write*1	t_{spvr}	2	2	—	µs	
	Wait time after PV1 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 ESU1 bit setting*1	t_{sesu}	100	100	—	µs	
	Wait time after E1 bit setting*1, *5	t_{se}	10	10	100	ms	Erase time wait
	Wait time after E1 bit clear*1	t_{ce}	10	10	—	µs	
	Wait time after ESU1 bit clear*1	t_{cesu}	10	10	—	µs	
	Wait time after EV1 bit setting*1	t_{sev}	20	20	—	µs	
	Wait time after H'FF dummy write*1	t_{sevr}	2	2	—	µs	
	Wait time after EV1 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 P1 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 E1 bit in FLMCR1 is set. It does not include the erase verification time.)

4. To specify the maximum programming time value (tp (max.)) in the 128-bytes programming algorithm, set the max. value (1000) for the maximum programming count (N).

The wait time after P1 bit setting should be changed as follows according to the value of the programming counter(n).

Programming counter(n) = 1 to 6: tsp30 = 30 μs
Programming counter(n) = 7 to 1000: tsp200 = 200 μs
[In additional programming]
Programming counter(n) = 1 to 6: tsp10 = 10 μs

5. For the maximum erase time (tE(max.)), the following relationship applies between the wait time after E1 bit setting (tse) and the maximum erase count (N):

tE (max.) = Wait time after E1 bit setting (tse) x maximum erase count (N)

To set the maximum erase time, the values of (tse) and (N) should be set so as to satisfy the above formula.

Examples: When tse = 100 ms, N = 12 times
When tse = 10 ms, N = 120 times

A. I/O Port States in Each Pin State

Port Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Program Execution State Sleep Mode
Port 1	7	T	T	Keep	I/O port
Port 4	7	T	T	T	Input port
Port 9	7	T	T	T	Input port
Port A	7	T	T	Keep	I/O port
Port B	7	T	T	Keep	I/O port
Port C	7	T	T	Keep	I/O port
Port D	7	T	T	Keep	I/O port
PF7	7	T	T	[DDR = 0] T [DDR = 1] H	[DDR = 0] T [DDR = 1] Clock output
PF6 PF5 PF4 PF3 PF2 PF1 PF0	7	T	T	Keep	I/O port
HTxD	7	H	T	H	Output
HRxD	7	Input	T	T	Input

Legend

H: High level

T: High impedance

Keep: Input port becomes high-impedance, output port retains state

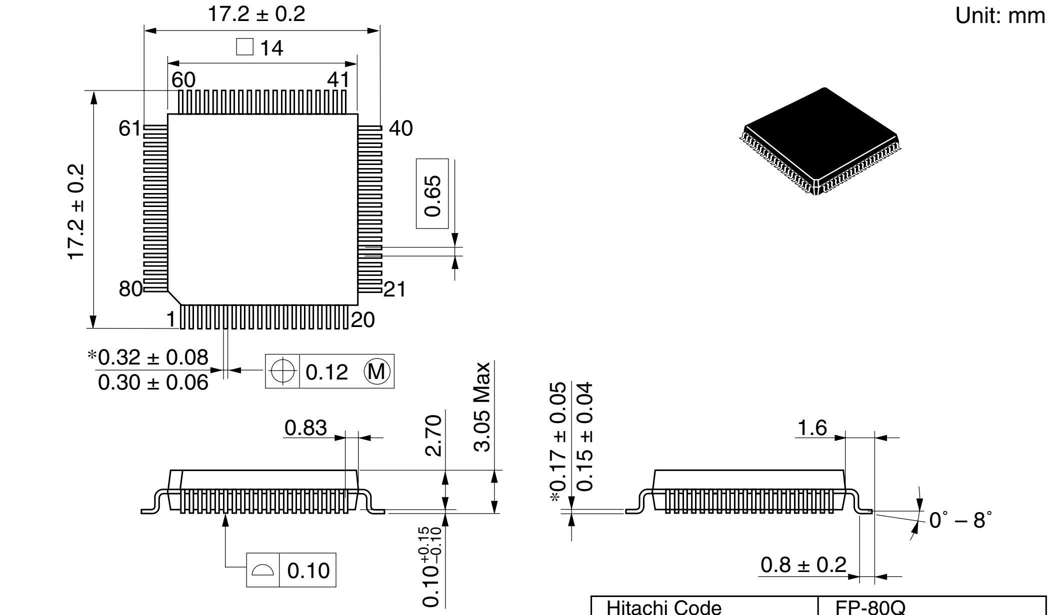
B. Product Code Lineup

Product Classification		Type Name	Model Marking	Package (Code)
H8S/2615	Flash memory version	HD64F2615	HD64F2615	80-pin QFP (FP-80Q)
	Masked ROM version*	HD6432615	HD6432615	

Note: * In planning

C. Package Dimensions

The package dimension that is shown in the Hitachi Semiconductor Package Data Book has priority.



*Dimension including the plating thickness
Base material dimension

Hitachi Code	FP-80Q
JEDEC	—
JEITA	Conforms
Mass (reference value)	1.2 g

Figure C.1 FP-80Q Package Dimensions

Index

H8S/2615 Series Hardware Manual —Preliminary—

Publication Date: 0.5th Edition, March 2003
Published by: Business Operation Division
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
Hitachi Kodaira Semiconductor Co., Ltd.