User's Manual

32

The revision list can be viewed directly by clicking the title page.
The revision list summarizes the locations of revisions and additions.
Details should always be checked by referring to the relevant text.

SH7730 Group

User's Manual: Hardware

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family/SH7780 Series

SH7730 R8A77301

Renesas Electronics
www.renesas.com

Rev.4.00 2012.03

Notice

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

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

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

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

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

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

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.

5. Reading from/Writing Reserved Bit of Each Register

Note: Treat the reserved bit of register used in each module as follows except in cases where the specifications for values which are read from or written to the bit are provided in the description.
The bit is always read as 0. The write value should be 0 or one, which has been read immediately before writing.
Writing the value, which has been read immediately before writing has the advantage of preventing the bit from being affected on its extended function when the function is assigned.

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 for 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 SH7730 Group RISC (Reduced Instruction Set Computer) microcomputers include a Renesas-original RISC CPU as its core, and the peripheral functions required to configure a system.

Target Users: This manual was written for users who will be using this SH7730 Group in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of SH7730 Group to the target users. Refer to the SH-4A Extended Functions Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- Product names
  The following products are covered in this manual.

Product Classifications and Abbreviations

Basic Classification	Product Code
SH7730	R8A77301

- 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 SH-4A Extended Functions Software Manual.

Rules:

Register name:	The following notation is used for cases when the same or a similar function, e.g. 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 (most significant bit) is on the left and the LSB (least significant bit) is on the right.
Number notation:	Binary is B'xxxx, hexadecimal is H'xxxx, decimal is xxxx.
Signal notation:	An overbar is added to a low-active signal: $\overline{\text{xxxx}}$

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

SH7730 manuals:

Document Title	Document No.
SH7730 Hardware Manual	This manual
SH-4A Extended Functions Software Manual	REJ09B0224

Users manuals for development tools:

Document Title	Document No.
Super™ RISC engine C/C++ Compiler, Assembler, Optimizing Linkage Editor Compiler Package V.9.00 User's Manual	REJ10B0152
SuperH™ RISC engine High-performance Embedded Workshop 3 User's Manual	REJ10B0025
SuperH™ RISC engine High-performance Embedded Workshop 3 Tutorial	REJ10B0023

Application note:

Document Title	Document No.
SuperH™ RISC engine C/C++ Compiler Package Application Note	REJ05B0463

Abbreviations

ACIA	Asynchronous Communication Interface Adapter
AUD	Advanced User Debugger
BSC	Bus State Controller
CPG	Clock Pulse Generator
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller

ETU	Elementary Time Unit
FIFO	First-In First-Out
H-UDI	User Debug Interface
INTC	Interrupt Controller
JTAG	Joint Test Action Group
LSB	Least Significant Bit
MMU	Memory Management Unit
MSB	Most Significant Bit
PFC	Pin Function Controller
RISC	Reduced Instruction Set Computer
SCIF	Serial Communication Interface with FIFO
TLB	Translation Lookaside Buffer
TMU	Timer Unit
UART	Universal Asynchronous Receiver/Transmitter
UBC	User Break Controller
WDT	Watch Dog Timer

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

Contents

Figures

Appendix

Tables

Appendix

Section 1 Overview

1.1 Features of This LSI

This LSI is a single-chip RISC microprocessor that integrates a 32-bit RISC-type SuperH architecture CPU as its core with a floating point unit (FPU), 64-Kbyte large-capacity cache memory, 16-Kbyte on-chip memory, an interrupt controller, and other peripherals.

High-speed data transfers can be performed by an on-chip direct memory access controller (DMAC), and an external memory access support function enables direct connection to different kinds of memory.

Moreover, this LSI incorporates a 16-bit timer pulse unit (TPU), realtime clock (RTC), compare-match timer (CMT), stereo audio recording/playback function interface (SIOF), I^2C bus interface (IIC), serial communication interface with FIFO (SCIF), IrDA interface (IrDA), SIM card interface (SIM), A/D converter (ADC), and D/A converter (DAC).

A powerful built-in power-management function keeps power consumption low, even during high-speed operation. This LSI is ideal for use in electronic devices such as those for applications that require both high-speed operation and low power consumption simultaneously.

The features of this LSI are listed in table 1.1.

Table 1.1 Features of This LSI

Item	Features
CPU	• Renesas original architecture • Upward compatible with SH-1, SH-2, SH-3, and SH4 at instruction set level • 32-bit internal data bus • General-register files — Sixteen 32-bit general registers (eight 32-bit shadow registers) — Seven 32-bit control registers — Four 32-bit system registers • RISC-type instruction set (upward compatible with SH-1, SH-2, SH-3, and SH4 — Instruction length: 16-bit fixed length for improved code efficiency — Load/store architecture — Delayed branch instructions — Instructions executed with conditions — Instruction set based on the C language • Super scalar design which executes two instructions simultaneously • Instruction execution time: Two instructions per cycle (max.) • Virtual address space: 4 Gbytes • Space identifier ASID: 8 bits, 256 virtual address spaces • Built-in multiplier • Eight-stage pipeline

Item	Features
Floating-point unit (FPU)	• Single precision (32 bits) and double precision (64 bits) • IEEE754-compliant data types and exceptions • Rounding mode: Rounding to nearest and rounding to 0 • Handling of non-normalized values: Truncate to 0 or interrupt generation because of compliance to IEEE754 • Floating-point registers: 32 bits × 16 registers × 2 banks (single precision × 16 registers or double precision × 8 registers) × 2 banks • 32-bit CPU-FPU floating-point communication register (FPUL) • FMAC (multiply and accumulate) instruction • FDIV (division)/FSQRT (square root) instructions • FLDI0/FLDI1 (load constants 0 and 1) instructions • Instruction execution time — Latency (FADD/FSUB): 3 cycles (single precision), 5 cycles (double precision) — Latency (FMAC/FMUL): 5 cycles (single precision), 7 cycles (double precision) — Pitch (FADD/FSUB): 1 cycle (single/double precision) — Pitch (FMAC/FMUL): 1 cycle (single precision), 3 cycles (double precision) Note: FMAC instruction only support single precision • 3D graphics instructions (single precision only) — 4-dimensional vector transformation and matrix operation (FTRV): 4 cycles (pitch), 8 cycles (latency) — 4-dimensional inner product (FIPR): 1 cycle (pitch), 5 cycles (latency) • 10-stage pipeline
Memory management unit (MMU)	• 4-Gbyte address space, 256 address space identifiers (8-bit ASID) • Single virtual memory mode and multiple virtual memory mode • Supports multiple page sizes: 1 Kbyte, 4 Kbytes, 64 Kbytes, or 1 Mbyte • 4-entry full associative TLB for instructions • 64-entry full associative TLB for instructions and operands • Supports replacement by software and random counter-based replacement algorithm • Address mapping allows direct access to TLB contents
Cache memory	• Instruction cache (IC) — 32-Kbyte, 4-way set associative — 32-byte block length • Operand cache (OC) — 32-Kbyte, 4-way set associative — 32-byte block length — Selectable write mode (copy-back or write-through)

Item	Features
On-chip memory (IL memory)	• Three independent read/write ports — Instruction fetch access from the CPU — 8-/16-/32-bit operand access from the CPU — 8-/16-/32-/64-bit or 16-/32-byte access from the DMAC • Total of 16 Kbytes
Interrupt controller (INTC)	• 21 external interrupt pins (NMI, IRQ7 to IRQ0, PINTA7 to PINTA0, PINTB3 to PINTB0) — NMI: Fall/rise detection selectable — IRQ: Fall/rise/high-level/low-level detection selectable • 15-level coded external interrupts: IRL3 to IRL0 (shared with IRQ3 to IRQ0) • On-chip peripheral interrupts: Priority can be specified for each module
Bus state controller (BSC)	• Supports physical address space for areas of up to 32 Mbytes, up to 64 Mbytes, and up to 128 Mbytes each. • The following functions can be set independently for each area — Data bus width: 8, 16, or 32 bits (16 or 32 bits for area 0) — Number of access wait cycles: For some areas, different wait cycles can be specified for read and write accesses — Idle wait cycle setting: For continuous access to the same area and to a different area — Supports SRAM, bust ROM, SDRAM, SRAM with byte-select function, and PCMCIA by specifying memory to be connected to each area. — Outputs a chip select signals, $\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$, $\overline{CS5A}$/$\overline{CS5B}$, or $\overline{CS6A}$/$\overline{CS6B}$ to the target area • SDRAM — Up to two 512-Mbit memory devices or up to one 1-Gbit memory device can be connected — Data bus width: 16 bits or 32 bits — Supports auto-refresh or self-refresh functions — Auto-precharge mode or bank active mode can be selected

Item	Features
Direct memory access controller (DMAC)	• Six channels, two channels of which (channels 0, 1) support external requests • Address space: 4 Gbytes on architecture • Data transfer length: Byte, word (2 bytes), longword (4 bytes), 8 bytes, 16 bytes, and 32 bytes • Maximum number of transfer times: 16,777,216 times • Address mode: Dual address mode • Transfer request: Selectable from external request, on-chip peripheral module request, and auto request • Bus mode: Selectable from cycle stealing mode (normal mode and intermittent mode) and burst mode • Priority: Selectable between fixed channel priority mode and round-robin mode • Interrupt request: Generates an interrupt request to the CPU at the end of data transfer • Repeat function: Automatically re-sets the transfer source, destination, and number of transfers at the end of DMA transfer • Reload function: Automatically re-sets the transfer source and destination at the end of the specified number of DMA transfers • External request detection: Selectable from low-level/high-level detection or rise/fall detection of the DREQ input • Transfer request acknowledge signals: Active level is selectable for DACK and TEND

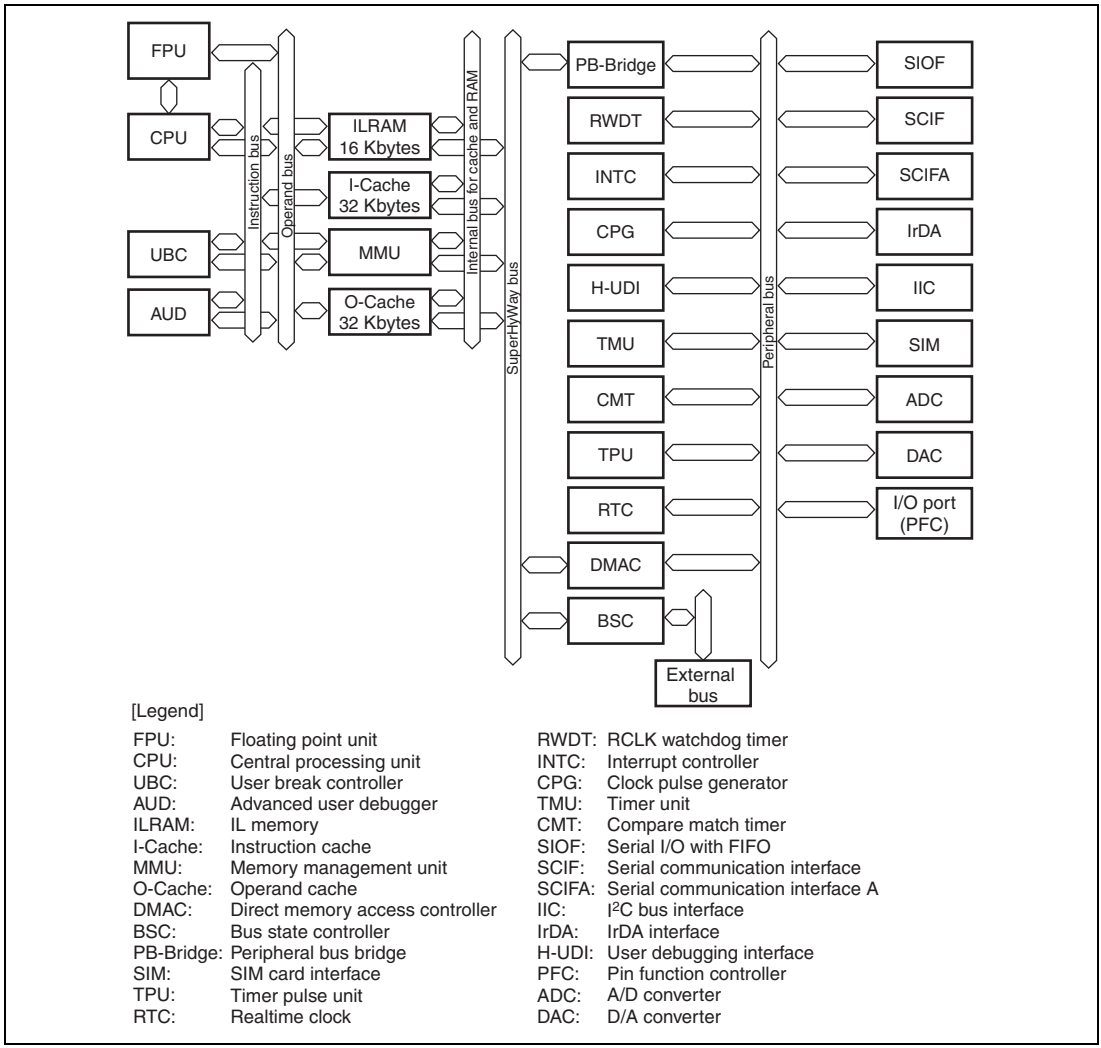
Item	Features
Clock pulse generator (CPG)	• Clock mode: Input clock selectable from external input (EXTAL) and crystal resonator • Output clock: Bus clock (Bϕ) • Generates four types of system clocks — CPU clock (Iϕ): Maximum 266.7 MHz (266 MHz version) Maximum 200 MHz (200 MHz version) — SH (SuperHyway) clock (Sϕ): Maximum 133.4 MHz — Bus clock (Bϕ): Maximum 66.7 MHz — Peripheral clock (Pϕ): Maximum 33.4 MHz • Supports power-down mode — Sleep mode — Software standby mode — Module standby mode
RCLK watchdog timer (RWDT)	• One channel of watchdog timer
16-bit Timer Pulse Unit (TPU)	• Six channels of 16-bit timers • Supports PWM function • Four types of counter input clocks
Realtime clock (RTC)	• Clock/calendar function (BCD representation) • 30-second adjustment function • Alarm/period/carry interrupts • Automatic leap year correction function Note: It is not possible to make the RTC operate alone by shutting off the power to all other modules except for the RTC.
Timer unit (TMU)	• Three channels of 32-bit timers • Incorporates a prescaler • Auto-reload type 32-bit down counter
Compare match timer (CMT)	• Five channels of 32-bit timers (16-bit/32-bit selectable) • Incorporates a prescaler • Compare match function provided on all channels • Generation of interrupt requests and DMA transfer requests
I^2C bus interface (IIC)	• Two channels • Supports multi-master/slave transmission and reception

Item	Features
Serial I/O with FIFO (SIOF)	• One channel • Internal 64-byte (32 bits × 16 stages) transmit/receive FIFOs • Supports 8-/16-bit data and 16-bit stereo audio input/output • Sampling rate clock input from an external pin • Incorporates a prescaler • Module stop function • Generation of interrupt requests and DMA transfer requests
Serial communication interface with FIFO (SCIF)	• Four channels (SCIF0 to SCIF3) • Internal 16-byte (8 bits × 16 stages) transmit/receive FIFOs • Asynchronous mode and clock synchronous mode • Modem control functions (RTS, CTS) on channels 2 and 3 • High-speed UART for Bluetooth • Incorporates a prescaler • Generation of interrupt requests and DMA transfer requests
Serial communication interface with FIFO A (SCIFA)	• Two channels (SCIF4, SCIF5) • Internal 64-byte (8 bits × 64 stages) transmit/receive FIFOs • Asynchronous mode and clock synchronous mode • Modem control functions (RTS, CTS) • High-speed UART for Bluetooth • Incorporates a prescaler • Generation of interrupt requests and DMA transfer requests
IrDA interface (IrDA)	• Two channels • Conforms to version 1.2a
SIM card interface (SIM)	• One channel. Conforms to the ISO 7816-3 data protocol. (T = 0, T = 1) • Asynchronous half-duplex character transmission protocol • Data length: 8 bits • Parity bit generation and check • Selectable output clock cycles per etu (elementary time unit) • Selectable direct convention/inverse convention • Incorporates a prescaler • Clock output level can be fixed (high or low) in idle state • Generation of interrupt requests and DMA transfer requests

Item	Features
A/D converter (ADC)	• 10 bits ±4 LSB, four channels • Conversion time: 16 μs • Input range: 0 to AVcc (maximum 3.6 V)
D/A converter (DAC)	• 10 bits ±4 LSB, two channels • Conversion time: 10 μs • Input range: 0 to AVcc (maximum 3.6 V)
I/O port	• 17 ports (107 pins in total) • Signal direction is switchable for each bit of I/O ports
User break controller (UBC)	• Supports debugging with user break interrupts • Two break channels • All of address, data value, access type, and data size can be set as break conditions • Supports sequential break function
User debugging interface (H-UDI)	• Supports E10A emulator • Realtime branch trace
Package	• 208-pin LQFP (PLQP0208KB-A)
Power-supply voltage	• I/O: 3.3 V ± 0.3 V • Internal: 1.2 ± 0.1 V
Process	• 90-nm CMOS

1.2 Block Diagram

Figure 1.1 shows a block diagram of this LSI.



[Legend]

FPU:	Floating point unit	RWDT:	RCLK watchdog timer
CPU:	Central processing unit	INTC:	Interrupt controller
UBC:	User break controller	CPG:	Clock pulse generator
AUD:	Advanced user debugger	TMU:	Timer unit
ILRAM:	IL memory	CMT:	Compare match timer
I-Cache:	Instruction cache	SIOF:	Serial I/O with FIFO
MMU:	Memory management unit	SCIF:	Serial communication interface
O-Cache:	Operand cache	SCIFA:	Serial communication interface A
DMAC:	Direct memory access controller	IIC:	I^2C bus interface
BSC:	Bus state controller	IrDA:	IrDA interface
PB-Bridge:	Peripheral bus bridge	H-UDI:	User debugging interface
SIM:	SIM card interface	PFC:	Pin function controller
TPU:	Timer pulse unit	ADC:	A/D converter
RTC:	Realtime clock	DAC:	D/A converter

Figure 1.1 Block Diagram of This LSI

1.3 Pin Assignments

Figure 1.2 and table 1.2 show the pin assignments.

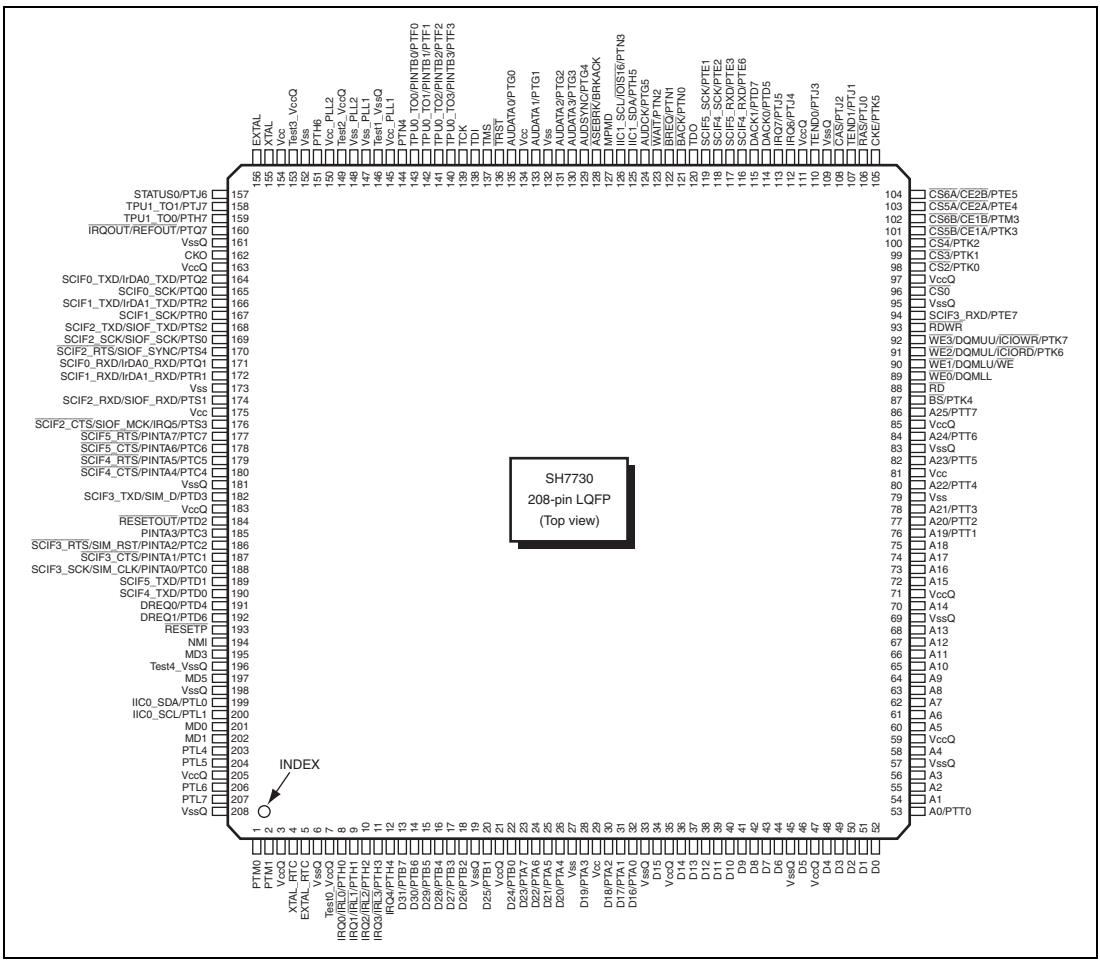


Figure 1.2 Pin Assignments (208-Pin LQFP)

Table 1.2 Pin Assignment

Pin No.	Pin Name	I/O	Description	Power Supply Source
1	AN2/PTM0	I/I	A/D converter input/input port M	AVcc
2	AN3/PTM1	I/I	A/D converter input/input port M	AVcc
3	VccQ	—	I/O power supply (3.3 V)	—
4	XTAL_RTC	O	Crystal resonator connection pin for RTC	VccQ
5	EXTAL_RTC	I	Crystal resonator connection pin for RTC	VccQ
6	VssQ	—	I/O ground (0 V)	—
7	Test0_VccQ	I	Test mode 0	VccQ
8	IRQ0/$\overline{\text{IRL0}}$/PTH0	I/I/I	External interrupt request/external interrupt request/input port H	VccQ
9	IRQ1/$\overline{\text{IRL1}}$/PTH1	I/I/I	External interrupt request/external interrupt request/input port H	VccQ
10	IRQ2/$\overline{\text{IRL2}}$/PTH2	I/I/I	External interrupt request/external interrupt request/input port H	VccQ
11	IRQ3/$\overline{\text{IRL3}}$/PTH3	I/I/I	External interrupt request/external interrupt request/input port H	VccQ
12	IRQ4/PTH4	I/I	External interrupt request/input port H	VccQ
13	D31/PTB7	IO/IO	Data bus/IO port B	VccQ
14	D30/PTB6	IO/IO	Data bus/IO port B	VccQ
15	D29/PTB5	IO/IO	Data bus/IO port B	VccQ
16	D28/PTB4	IO/IO	Data bus/IO port B	VccQ
17	D27/PTB3	IO/IO	Data bus/IO port B	VccQ
18	D26/PTB2	IO/IO	Data bus/IO port B	VccQ
19	VssQ	—	I/O ground (0 V)	—
20	D25/PTB1	IO/IO	Data bus/IO port B	VccQ
21	VccQ	—	I/O power supply (3.3 V)	—
22	D24/PTB0	IO/IO	Data bus/IO port B	VccQ
23	D23/PTA7	IO/IO	Data bus/IO port A	VccQ
24	D22/PTA6	IO/IO	Data bus/IO port A	VccQ
25	D21/PTA5	IO/IO	Data bus/IO port A	VccQ
26	D20/PTA4	IO/IO	Data bus/IO port A	VccQ

Pin No.	Pin Name	I/O	Description	Power Supply Source
27	Vss	—	Internal ground (0 V)	—
28	D19/PTA3	IO/IO	Data bus/IO port A	VccQ
29	Vcc	—	Internal power supply (1.2 V)	—
30	D18/PTA2	IO/IO	Data bus/IO port A	VccQ
31	D17/PTA1	IO/IO	Data bus/IO port A	VccQ
32	D16/PTA0	IO/IO	Data bus/IO port A	VccQ
33	VssQ	—	I/O ground (0 V)	—
34	D15	IO	Data bus	VccQ
35	VccQ	—	I/O power supply (3.3 V)	—
36	D14	IO	Data bus	VccQ
37	D13	IO	Data bus	VccQ
38	D12	IO	Data bus	VccQ
39	D11	IO	Data bus	VccQ
40	D10	IO	Data bus	VccQ
41	D9	IO	Data bus	VccQ
42	D8	IO	Data bus	VccQ
43	D7	IO	Data bus	VccQ
44	D6	IO	Data bus	VccQ
45	VssQ	—	I/O ground (0 V)	—
46	D5	IO	Data bus	VccQ
47	VccQ	—	I/O power supply (3.3 V)	—
48	D4	IO	Data bus	VccQ
49	D3	IO	Data bus	VccQ
50	D2	IO	Data bus	VccQ
51	D1	IO	Data bus	VccQ
52	D0	IO	Data bus	VccQ
53	A0/PTT0	O/IO	Address bus/IO port T	VccQ
54	A1	O	Address bus	VccQ
55	A2	O	Address bus	VccQ
56	A3	O	Address bus	VccQ

Pin No.	Pin Name	I/O	Description	Power Supply Source
57	VssQ	—	I/O ground (0 V)	—
58	A4	O	Address bus	VccQ
59	VccQ	—	I/O power supply (3.3 V)	—
60	A5	O	Address bus	VccQ
61	A6	O	Address bus	VccQ
62	A7	O	Address bus	VccQ
63	A8	O	Address bus	VccQ
64	A9	O	Address bus	VccQ
65	A10	O	Address bus	VccQ
66	A11	O	Address bus	VccQ
67	A12	O	Address bus	VccQ
68	A13	O	Address bus	VccQ
69	VssQ	—	I/O ground (0 V)	—
70	A14	O	Address bus	VccQ
71	VccQ	—	I/O power supply (3.3 V)	—
72	A15	O	Address bus	VccQ
73	A16	O	Address bus	VccQ
74	A17	O	Address bus	VccQ
75	A18	O	Address bus	VccQ
76	A19/PTT1	O/IO	Address bus/IO port T	VccQ
77	A20/PTT2	O/IO	Address bus/IO port T	VccQ
78	A21/PTT3	O/IO	Address bus/IO port T	VccQ
79	Vss	—	Internal ground (0 V)	—
80	A22/PTT4	O/IO	Address bus/IO port T	VccQ
81	Vcc	—	Internal power supply (1.2 V)	—
82	A23/PTT5	O/IO	Address bus/IO port T	VccQ
83	VssQ	—	I/O ground (0 V)	—
84	A24/PTT6	O/IO	Address bus/IO port T	VccQ
85	VccQ	—	I/O power supply (3.3 V)	—

Pin No.	Pin Name	I/O	Description	Power Supply Source
86	A25/PTT7	O/IO	Address bus/IO port T	VccQ
87	$\overline{BS}$/PTK4	O/IO	Bus cycle start signal/IO port K	VccQ
88	$\overline{RD}$	O	Read strobe	VccQ
89	$\overline{WE0}$/DQMLL	O/O	D7 to D0 select signal/DQM (SDRAM)	VccQ
90	$\overline{WE1}$/DQMLU/$\overline{WE}$	O/O/O	D15 to D8 select signal/DQM (SDRAM)/ PCMCIA write enable	VccQ
91	$\overline{WE2}$/DQMUL/$\overline{ICIORD}$/PTK6	O/O/O/IO	D23 to D16 select signal/DQM (SDRAM)/ PCMCIA IO read/IO port K	VccQ
92	$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$/PTK7	O/O/O/IO	D31 to D24 select signal/DQM (SDRAM)/ PCMCIA IO read/IO port K	VccQ
93	$\overline{RDWR}$	O	Read/write	VccQ
94	SCIF3_RXD/PTE7	I/IO	SCIF3 receive data/IO port E	VccQ
95	VssQ	—	I/O ground (0 V)	—
96	$\overline{CS0}$	O	Chip select 0	VccQ
97	VccQ	—	I/O power supply (3.3 V)	—
98	$\overline{CS2}$/PTK0	O/IO	Chip select 2/IO port K	VccQ
99	$\overline{CS3}$/PTK1	O/IO	Chip select 3/IO port K	VccQ
100	$\overline{CS4}$/PTK2	O/IO	Chip select 4/IO port K	VccQ
101	$\overline{CS5B}$/$\overline{CE1A}$/PTK3	O/O/IO	Chip select 5B/CE1 (area 5 PCMCIA)/ IO port K	VccQ
102	$\overline{CS6B}$/$\overline{CE1B}$/PTM3	O/O/IO	Chip select 6B/CE1 (area 6 PCMCIA)/ IO port M	VccQ
103	$\overline{CS5A}$/$\overline{CE2A}$/PTE4	O/O/IO	Chip select 5A/CE2 (area 5 PCMCIA)/ IO port E	VccQ
104	$\overline{CS6A}$/$\overline{CE2B}$/PTE5	O/O/IO	Chip select 6A/CE2 (area 6 PCMCIA)/ IO port E	VccQ
105	CKE/PTK5	O/IO	CK enable (SDRAM)/IO port K	VccQ
106	$\overline{RAS}$/PTJ0	O/IO	RAS (SDRAM)/IO port J	VccQ
107	TEND1/PTJ1	O/IO	DMA transfer end/IO port J	VccQ
108	$\overline{CAS}$/PTJ2	O/IO	CAS (SDRAM)/IO port J	VccQ
109	VssQ	—	I/O ground (0 V)	—
110	TEND0/PTJ3	O/IO	DMA transfer end/IO port J	VccQ

Pin No.	Pin Name	I/O	Description	Power Supply Source
111	VccQ	—	I/O power supply (3.3 V)	—
112	IRQ6/PTJ4	I/I	External interrupt request/input port J	VccQ
113	IRQ7/PTJ5	I/I	External interrupt request/input port J	VccQ
114	DACK0/PTD5	O/IO	DMA acknowledge 0/IO port D	VccQ
115	DACK1/PTD7	O/IO	DMA acknowledge 1/IO port D	VccQ
116	SCIF4_RXD/PTE6	I/I	SCIF4 receive data/input port E	VccQ
117	SCIF5_RXD/PTE3	I/I	SCIF5 receive data/input port E	VccQ
118	SCIF4_SCK/PTE2	IO/IO	SCIF4 serial clock/IO port E	VccQ
119	SCIF5_SCK/PTE1	IO/IO	SCIF5 serial clock/IO port E	VccQ
120	TDO	O	Test data output	VccQ
121	$\overline{\text{BACK}}$/PTN0	O/IO	Bus acknowledge/IO port N	VccQ
122	$\overline{\text{BREQ}}$/PTN1	I/I	Bus request/input port N	VccQ
123	$\overline{\text{WAIT}}$/PTN2	I/I	Hardware wait request/input port N	VccQ
124	AUDCK/PTG5	O/IO	Dedicated pin for emulator/IO port G	VccQ
125	IIC1_SDA/$\overline{\text{ADTRG}}$/PTH5	IO/I/I	IIC1 data/analog trigger/input port H	VccQ
126	IIC1_SCL/$\overline{\text{IOIS16}}$/PTN3	IO/I/I	IIC1 data/IOIS16 (PCMCIA)/input port N	VccQ
127	MPMD	I	ASE mode	VccQ
128	$\overline{\text{ASEBRK}}$/BRKACK	I/O	Dedicated pin for emulator	VccQ
129	AUDSYNC/PTG4	O/IO	Dedicated pin for emulator /IO port G	VccQ
130	AUDATA3/PTG3	O/IO	Dedicated pin for emulator /IO port G	VccQ
131	AUDATA2/PTG2	O/IO	Dedicated pin for emulator /IO port G	VccQ
132	Vss	—	Internal ground (0 V)	—
133	AUDATA1/PTG1	O/IO	Dedicated pin for emulator /IO port G	VccQ
134	Vcc	—	Internal power supply (1.2 V)	—
135	AUDATA0/PTG0	O/IO	Dedicated pin for emulator /IO port G	VccQ
136	$\overline{\text{TRST}}$	I	Test reset	VccQ
137	TMS	I	Test mode switch	VccQ
138	TDI	I	Test data input	VccQ
139	TCK	I	Test clock	VccQ

Pin No.	Pin Name	I/O	Description	Power Supply Source
140	TPU0_TO3/PINTB3/PTF3	O/I/IO	TPU0 output compare/port interrupt/ IO port F	VccQ
141	TPU0_TO2/PINTB2/PTF2	O/I/IO	TPU0 output compare/port interrupt/ IO port F	VccQ
142	TPU0_TO1/PINTB1/PTF1	O/I/IO	TPU0 output compare/port interrupt/ IO port F	VccQ
143	TPU0_TO0/PINTB0/PTF0	O/I/IO	TPU0 output compare/port interrupt/ IO port F	VccQ
144	PTN4	IO	IO port N	VccQ
145	Vcc_PLL1	—	PLL1 power supply (1.2 V)	—
146	Test1_VssQ	I	Test mode 1	VccQ
147	Vss_PLL1	—	PLL1 ground (0 V)	—
148	Vss_PLL2	—	PLL2 ground (0 V)	—
149	Test2_VccQ	I	Test mode 2	VccQ
150	Vcc_PLL2	—	PLL2 power supply (1.2 V)	—
151	PTH6	I	Input port H	VccQ
152	Vss	—	Internal ground (0 V)	—
153	Test3_VccQ	I	Test mode 3	VccQ
154	Vcc	—	Internal power supply (1.2 V)	—
155	XTAL	O	Crystal resonator connection pin	VccQ
156	EXTAL	I	External clock/ Crystal resonator connection pin	VccQ
157	STATUS0/PTJ6	O/IO	Processor status/IO port J	VccQ
158	TPU1_TO1/PTJ7	O/IO	TPU1 output compare/IO port J	VccQ
159	TPU1_TO0/PTH7	O/IO	TPU1 output compare/IO port H	VccQ
160	$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$/PTQ7	O/O/IO	Interrupt request notification/ refresh output/IO port Q	VccQ
161	VssQ	—	I/O ground (0 V)	—
162	CKO	O	System clock output	VccQ

Pin No.	Pin Name	I/O	Description	Power Supply Source
163	VccQ	—	I/O power supply (3.3 V)	—
164	SCIF0_TXD/IrDA0_TXD/PTQ2	O/O/IO	SCIF0 transmit data/IrDA transmit data/ IO port Q	VccQ
165	SCIF0_SCK/PTQ0	IO/IO	SCIF0 serial clock/IO port Q	VccQ
166	SCIF1_TXD/IrDA1_TXD/PTR2	O/O/IO	SCIF1 transmit data/IrDA transmit data/ IO port R	VccQ
167	SCIF1_SCK/PTR0	IO/IO	SCIF1 serial clock/IO port R	VccQ
168	SCIF2_TXD/SIOF_TXD/PTS2	O/O/IO	SCIF2 transmit data/SIOF transmit data/ IO port S	VccQ
169	SCIF2_SCK/SIOF_SCK/PTS0	IO/IO/IO	SCIF2 serial clock/SIOF serial clock/ IO port S	VccQ
170	$\overline{\text{SCIF2\_RTS}}$/SIOF_SYNC/ PTS4	O/IO/IO	SCIF2 transmit request/SIOF frame sync signal/IO port S	VccQ
171	SCIF0_RXD/IrDA0_RXD/PTQ1	I/I/I	SCIF0 receive data/IrDA receive data/ input port Q	VccQ
172	SCIF1_RXD/IrDA1_RXD/PTR1	I/I/I	SCIF1 receive data/IrDA receive data/ input port R	VccQ
173	Vss	—	Internal ground (0 V)	—
174	SCIF2_RXD/SIOF_RXD/PTS1	I/I/I	SCIF2 receive data/SIOF receive data/ input port S	VccQ
175	Vcc	—	Internal power supply (1.2 V)	—
176	$\overline{\text{SCIF2\_CTS}}$/SIOF_MCK/ IRQ5/PTS3	I/I/I/I	SCIF2 transmit clear/SIOF master clock input/external interrupt request/input port S	VccQ
177	$\overline{\text{SCIF5\_RTS}}$/PINTA7/PTC7	O/I/IO	SCIF5 transmit request/port interrupt/ IO port C	VccQ
178	$\overline{\text{SCIF5\_CTS}}$/PINTA6/PTC6	I/I/IO	SCIF5 transmit clear/port interrupt/ IO port C	VccQ
179	$\overline{\text{SCIF4\_RTS}}$/PINTA5/PTC5	O/I/IO	SCIF4 transmit request/port interrupt/ IO port C	VccQ
180	$\overline{\text{SCIF4\_CTS}}$/PINTA4/PTC4	I/I/IO	SCIF4 transmit clear/port interrupt/ IO port C	VccQ
181	VssQ	—	I/O ground (0 V)	—
182	SCIF3_TXD/SIM_D/PTD3	O/IO/IO	SCIF3 transmit data/smart card transmit and receive data/IO port D	VccQ

Pin No.	Pin Name	I/O	Description	Power Supply Source
183	VccQ	—	I/O power supply (3.3 V)	—
184	$\overline{\text{RESETOUT}}$/PTD2	O/IO	Power-on reset output/IO port D	VccQ
185	PINTA3/PTC3	I/IO	Port interrupt/IO port C	VccQ
186	$\overline{\text{SCIF3\_RTS}}$/SIM_RST/ PINTA2/PTC2	O/O/I/IO	SCIF3 transmit request/smart card reset/ port interrupt/IO port C	VccQ
187	$\overline{\text{SCIF3\_CTS}}$/PINTA1/PTC1	I/I/IO	SCIF3 transmit clear/port interrupt/IO port C	VccQ
188	SCIF3_SCK/SIM_CLK/ PINTA0/PTC0	IO/O/I/IO	SCIF3 serial clock/smart card clock/ port interrupt/IO port C	VccQ
189	SCIF5_TXD/PTD1	O/IO	SCIF5 transmit data/IO port D	VccQ
190	SCIF4_TXD/PTD0	O/IO	SCIF4 transmit data/IO port D	VccQ
191	DREQ0/PTD4	I/I	DMA request 0/input port D	VccQ
192	DREQ1/PTD6	I/I	DMA request 1/input port D	VccQ
193	$\overline{\text{RESETP}}$	I	Power-on reset request	VccQ
194	NMI	I	Non-maskable interrupt request	VccQ
195	MD3	I	Area 0 bus width setting	VccQ
196	Test4_VssQ	I	Test mode 4	VccQ
197	MD5	I	Data alignment setting	VccQ
198	VssQ	—	I/O ground (0 V)	—
199	IIC0_SDA/PTL0	IO/I	IIC0 data/input port L	VccQ
200	IIC0_SCL/PTL1	IO/I	IIC0 clock/input port L	VccQ
201	MD0	I	Clock mode setting	VccQ
202	MD1	I	Clock mode setting	VccQ
203	DA1/PTL4	O/I	DA converter output/input port L	AVcc
204	DA0/PTL5	O/I	DA converter output/input port L	AVcc
205	AVcc	—	Analog power supply (3.3 V)	—
206	AN0/PTL6	I/I	AD converter input/input port L	AVcc
207	AN1/PTL7	I/I	AD converter input/input port L	AVcc
208	AVss	—	Analog ground (0 V)	—

1.3.1 Pin Function

Table 1.3 Pin Functions

Classification	Symbol	Function	I/O	Description
Power supply	Vcc	Power supply	—	Power supply for the internal modules and ports for the system. Connect all Vcc pins to the system power supply. There will be no operation if any pins are open.
	Vss	Ground	—	Ground pin. Connect all Vss pins to the system power supply (0 V). There will be no operation if any pins are open.
	VccQ	Power supply	—	Power supply for I/O pins. Connect all VccQ pins to the system power supply. There will be no operation if any pins are open.
	VssQ	Ground	—	Ground pin. Connect all VssQ pins to the system power supply (0 V). There will be no operation if any pins are open.
	AVcc	Analog power supply	—	Analog power supply
	AVss	Analog ground	—	Analog ground and reference voltage for A/D conversion
Clock	Vcc_PLL1	PLL1 power supply	—	Power supply for the on-chip PLL1 oscillator.
	Vss_PLL1	PLL1 ground	—	Ground pin for the on-chip PLL1 oscillator.
	Vcc_PLL2	PLL2 power supply	—	Power supply for the on-chip PLL2 oscillator.
	Vss_PLL2	PLL2 ground	—	Ground pin for the on-chip PLL2 oscillator.
	XTAL	Crystal resonator Connection	Output	Connects the crystal resonator
	EXTAL	Crystal resonator Connection/ external clock input	Input	For connection of the crystal resonator; also, used as an external clock input pin.
	CKO	Clock output pin	Output	Used as an external clock output pin.

Classification	Symbol	Function	I/O	Description
Operating mode control	MD0, MD1	Clock mode setting	Input	Sets the clock operating mode.
	MD3	Area 0 bus width setting	Input	Specifies area 0 bus width (16/32 bits)
	MD5	Data alignment setting	Input	Selects data alignment (big endian or little endian)
System control	$\overline{\text{RESETP}}$	Power-on reset request	Input	This LSI enters the power-on reset state when this pin becomes low level.
	$\overline{\text{RESETOUT}}$	Power-on reset output signal	Output	Becomes low level while this LSI is being power-on reset.
	STATUS0	Processing state 0	Output	Becomes high level in software standby mode.
Interrupt controller (INTC)	NMI	Non maskable interrupt input pin	Input	Interrupt request signal that is not maskable
	IRQ7 to IRQ4, IRQ3/$\overline{\text{IRL3}}$ to IRQ0/$\overline{\text{IRL0}}$	External interrupt input pins	Input	Inputs of interrupt request signals
	$\overline{\text{IRQOUT}}$	Interrupt request output pin	Output	Signal indicating that an interrupt request has been generated.
	PINTA7 to PINTA0, PINTB3 to PINTB0	Port-interrupt input pins	Input	Inputs of port interrupt request signals
Bus Control.	A25 to A0	Address bus	Input	Address bus
	D31 to D0	Data bus	Output	Data bus
	$\overline{\text{BS}}$	Bus cycle start	Output	Bus cycle start
	$\overline{\text{CS0}}$, $\overline{\text{CS2}}$ to $\overline{\text{CS4}}$	Chip select	Output	Chip select
	$\overline{\text{CS5A}}$/$\overline{\text{CE2A}}$		Output	Chip select Active only for address maps 1 and 3 Corresponds to PCMCIA card select signals D15 to D8 when the PCMCIA is used.
	$\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$		Output	Chip select Corresponds to PCMCIA card select signals D7 to D0 when the PCMCIA is used.

Classification	Symbol	Function	I/O	Description
Bus control	$\overline{CS6A}/\overline{CE2B}$	Chip select	Output	Chip select Active only for address maps 1 and 3 Corresponds to PCMCIA card select signals D15 to D8 when the PCMCIA is used.
	$\overline{CS6B}/\overline{CE1B}$		Output	Chip select Corresponds to PCMCIA card select signals D7 to D0 when the PCMCIA is used.
	$\overline{RDWR}$	Read/write	Output	Read or write signal Connects to WE pins when SDRAM or byte-selection SRAM is connected.
	$\overline{RD}$	Read strobe	Output	Read pulse signal (read data output enable signal) A strobe signal to indicate the memory read cycle when the PCMCIA is used.
	$\overline{WE3}$/DQMUU /$\overline{ICIOWR}$	Select signal for D31 to D24 /DQM (SDRAM) /PCMCIA I/O write	Output	Indicates that D31 to D24 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected. Corresponds to signals D31 to D24 when SDRAM is connected. Functions as the I/O write strobe signal when the PCMCIA is used.
	$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$	Select signal for D23 to D16 /DQM (SDRAM) /PCMCIA I/O read	Output	Indicates that D23 to D16 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected. Corresponds to signals D23 to D16 when the SDRAM is used. Functions as the I/O read strobe signal when the PCMCIA is used.
	$\overline{WE1}$/DQMLU/ $\overline{WE}$	Select signal for D15 to D8 /DQM (SDRAM) /PCMCIA I/O write enable	Output	Indicates that D15 to D8 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected. Corresponds to signals D15 to D8 when the SDRAM is used. Functions as the memory write enable signal when the PCMCIA is used.

Classification	Symbol	Function	I/O	Description
Bus control	$\overline{WE0}$/DQMLL	Select signal for D7 to D0 /DQM (SDRAM)	Output	Indicates that D7 to D0 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected. Corresponds to select signals D7 to D0 when the SDRAM is used.
	$\overline{RAS}$	RAS (SDRAM)	Output	Connects to RAS pin when SDRAM is connected.
	$\overline{CAS}$	CAS (SDRAM)	Output	Connects to CAS pin when SDRAM is connected.
	CKE	CK enable (SDRAM)	Output	Connects to CKE pin when SDRAM is connected.
	$\overline{IOIS16}$	PCMCIA 16 bits I/O	Input	PCMCIA 16-bit I/O signal Valid only in little endian mode. Pulled low in big endian mode.
	$\overline{WAIT}$	Hardware wait request	Input	External wait input
	$\overline{BREQ}$	Bus Request	Input	Bus request input
	$\overline{BACK}$	Bus acknowledge	Output	Bus acknowledge output
	$\overline{REFOUT}$	Refresh request	Output	Refresh request output when a bus is released
Direct memory access controller (DMAC)	DREQ0, DREQ1	DMA transfer request	Input	DMA transfer request input from external device.
	DACK0, DACK1	DMA transfer request acknowledge	Output	Strobe as a response to the DMA transfer request, which is output to external device
	TEND0, TEND1	DMA transfer end notification	Output	DMA transfer end output to external device
16-bits timer pulse unit (TPU)	TPU0TO3 to TPU0TO0, TPU1TO1, TPU1TO0	Compare match output	Output	Compare match output/PMW output pin
Realtime clock (RTC)	EXTAL_RTC	Crystal resonator connection	Output	Connects the crystal resonator for the RTC.
	XTAL_RTC	Crystal resonator connection	Input	Connects the crystal resonator for the RTC.

Classification	Symbol	Function	I/O	Description
I2C bus interface (IIC)	IIC0_SCL, IIC1_SCL	I2C serial clock	I/O	I2C serial clock input/output pin
	IIC0_SDA, IIC1_SDA	I2C serial data	Output	I2C serial data input/output pin
Serial I/O with FIFO (SIOF)	SIOF_MCK	Master clock	Input	Master clock input pin
	SIOF_SCK	Serial clock	I/O	Serial clock (common to transmission/reception)
	SIOF_SYNC	Frame synchronization	I/O	Frame synchronization signal (common to transmission/reception)
	SIOF_TXD	Transmit data	Output	Transmit data pin
	SIOF_RXD	Receive data	Input	Receive data pin
Serial Communication interface with FIFO (SCIF)	SCIF3_TXD to SCIF0_TXD	Transmit data	Output	Transmit data pin
	SCIF3_RXD to SCIF0_RXD	Receive data	Input	Receive data pin
	SCIF3_SCK to SCIF0_SCK	Serial clock	I/O	Clock I/O pin
	$\overline{\text{SCIF2\_RTS}}$, $\overline{\text{SCIF3\_RTS}}$	Transmit request	Output	Transmit request output pin
	$\overline{\text{SCIF2\_CTS}}$, $\overline{\text{SCIF3\_CTS}}$	Transmit enable	Input	Transmit enable input pin
Serial communication interface A with FIFO (SCIFA)	SCIF4_TXD, SCIF5_TXD	Transmit data	Output	Transmit data pin
	SCIF4_RXD, SCIF5_RXD	Receive data	Input	Receive data pin
	SCIF4_SCK, SCIF5_SCK	Serial clock	I/O	Clock I/O pin

Classification	Symbol	Function	I/O	Description
Serial communication interface A with FIFO (SCIFA)	$\overline{\text{SCIF4\_RTS}}$, $\overline{\text{SCIF5\_RTS}}$	Transmit request	Output	Transmit request output pin
	$\overline{\text{SCIF4\_CTS}}$, $\overline{\text{SCIF5\_CTS}}$	Transmit enable	Input	Transmit enable input pin
IrDA interface (IrDA)	IrDA0_RXD, IrDA1_RXD	IrDA receive data	Input	Infrared receive (light-receive) pulse input
	IrDA0_TXD, IrDA1_TXD	IrDA transmit data	Output	Infrared transmit (light-emit) pulse output
SIM card module (SIM)	SIM_D	Smart card data	I/O	Smart card data input/output
	SIM_CLK	Smart card clock	Output	Smart card clock output
	SIM_RST	Smart card reset	Output	Smart card reset output
A/D converter (ADC)	AVcc	Analog power supply	—	Analog power supply
	AVss	Analog ground	—	Analog ground and reference voltage for A/D conversion
	AN3 to AN0	Analog input	Input	Analog input pins
	$\overline{\text{ADTRG}}$	Analog trigger	Input	External trigger input for starting A/D conversion
D/A converter (DAC)	AVcc	Analog power supply	—	Analog power supply
	AVss	Analog ground	—	Analog ground and reference voltage for D/A conversion
	DA0, DA1	Analog output	Output	Analog output pins
I/O port	PTA7 to PTA0	General I/O Port	I/O	8-bit general I/O port pins.
	PTB7 to PTB0		I/O	8-bit general I/O port pins.
	PTC7 to PTC0		I/O	8-bit general I/O port pins.
	PTD7		I/O	8-bit general I/O and Input port pins.
	PTD6		Input	
	PTD5		I/O	

Classification	Symbol	Function	I/O	Description
I/O port	PTD4	General I/O Port	Input	8-bit general I/O and Input port pins.
	PTD3 to PTD0		I/O	
	PTE7		I/O	7-bit general I/O and Input port pins.
	PTE6		Input	
	PTE5, PTE4		I/O	
	PTE3		Input	
	PTE2, PTE1		I/O	
	PTF3 to PTF0		I/O	4-bit general I/O port pins.
	PTG5 to PTG0		I/O	6-bit general I/O port pins.
	PTH7		I/O	8-bit general I/O and Input port pins.
	PTH6 to PTH0		Input	
	PTJ7, PTJ6		I/O	8-bit general I/O and Input port pins.
	PTJ5, PTJ4		Input	
	PTJ3 to PTJ0		I/O	
	PTK7 to PTK0		I/O	8-bit general I/O port pins.
	PTL7 to PTE4_PTL1, PTL0		Input	4-bit general Input port pins.
	PTM3		I/O	3-bit general I/O and Input port pins.
	PTM1, PTM0		Input	
	PTN4		I/O	5-bit general I/O and Input port pins.
	PTN3 to PTN1		Input	
	PTN0		I/O	
	PTQ7 to PTQ2		I/O	4-bit general I/O and Input port pins.
	PTQ1		Input	
	PTQ0		I/O	
	PTR2		I/O	3-bit general I/O and Input port pins.
	PTR1		Input	
	PTR0		I/O	

Classification	Symbol	Function	I/O	Description
I/O port	PTS4	General I/O Port	I/O	5-bit general I/O and Input port pins.
	PTS3		Input	
	PTS2		I/O	
	PTS1		Input	
	PTS0		I/O	
	PTT7 to PTT0		I/O	8-bit general I/O port pins.
User debugging interface (H-UDI)	TCK	Test clock	Input	Test clock input pin
	TMS	Test mode switch	Input	Test mode select signal input pin
	$\overline{\text{TRST}}$	Test reset	Input	Initialization signal input pin
	TDI	Test data input	Input	Serial Input pin for instructions and data
	TDO	Test data output	Output	Serial output pin for instructions and data
	MPMD	ASE mode	Input	A low level on this pin places the chip in ASE mode, enabling use of the emulation support mode functions. When using an emulator such as the E10A, fix this pin at a low level.
E10A interface	$\overline{\text{ASEBRK}}$	Pins for an emulator	Input	For details, see the emulator's manual.
	BRKACK	Pins for an emulator	Output	For details, see the emulator's manual.
Advanced user debugger (AUD)	AUDSYNC	Pins for an emulator	Output	For details, see the emulator's manual.
	AUDCK	Pins for an emulator	Output	For details, see the emulator's manual.
	AUDATA3 to AUDATA0	Pins for an emulator	Output	For details, see the emulator's manual.

Note: Pins Test0_VccQ, Test1_VccQ, Test2_VccQ, Test3_VccQ, and Test4_VssQ are used for testing the chip prior to shipment from the factory. In ordinary use, they should be pulled up or pulled down to fix their electric potential.

1.4 Product Lineup

Table 1.4 Product Lineup

Abbreviation	Parts No.	Power Supply Voltage I/O	Power Supply Voltage Internal	Operating Frequency	Operating Temperature	Package
R8A77301	R8A77301C266FPV	3.3 ± 0.3 V	1.2 ± 0.1 V	266.7 MHz	Standard temperature range product -20 to 75 °C	208 Pin Plastic LQFP (PLQP0208KB-A)
	R8A77301C200FPV			200 MHz		
	R8A77301D266FPV			266.7 MHz	Extended temperature range product -40to 85 °C	
	R8A77301D200FPV			200 MHz		

Section 2 Programming Model

The programming model of the SH-4A is explained in this section. This SLI has registers and data formats as shown below.

2.1 Data Formats

The data formats supported in this LSI are shown in figure 2.1.

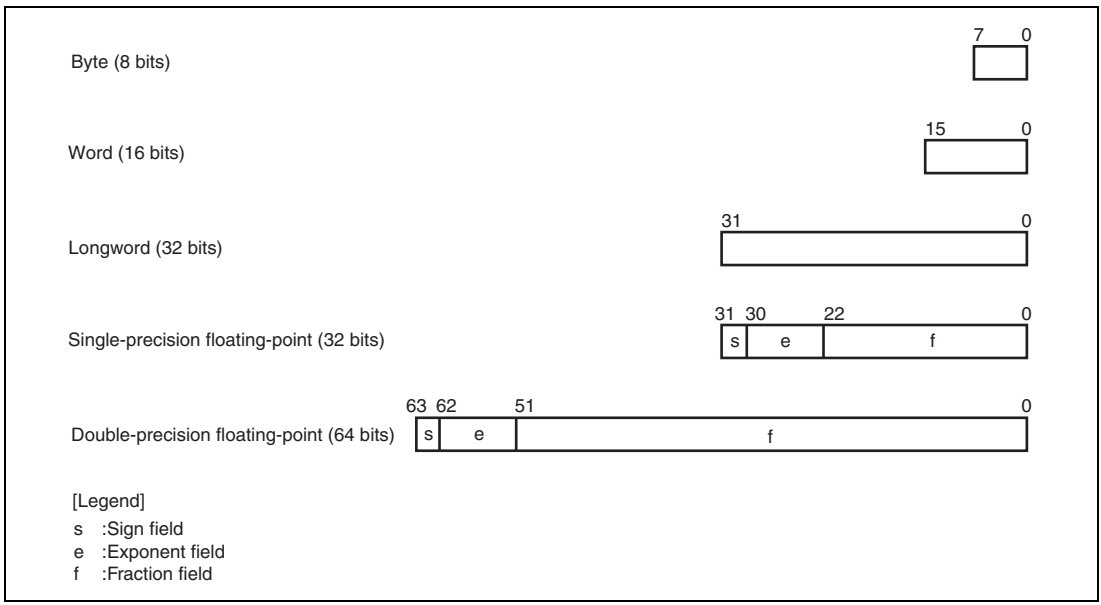


Figure 2.1 Data Formats

2.2 Register Descriptions

2.2.1 Privileged Mode and Banks

(1) Processing Modes

This LSI has two processing modes, user mode and privileged mode. This LSI normally operates in user mode, and switches to privileged mode when an exception occurs or an interrupt is accepted. There are four kinds of registers—general registers, system registers, control registers, and floating-point registers—and the registers that can be accessed differ in the two processing modes.

(2) General Registers

There are 16 general registers, designated R0 to R15. General registers R0 to R7 are banked registers which are switched by a processing mode change.

- Privileged mode

  In privileged mode, the register bank bit (RB) in the status register (SR) defines which banked register set is accessed as general registers, and which set is accessed only through the load control register (LDC) and store control register (STC) instructions.

  When the RB bit is 1 (that is, when bank 1 is selected), the 16 registers comprising bank 1 general registers R0_BANK1 to R7_BANK1 and non-banked general registers R8 to R15 can be accessed as general registers R0 to R15. In this case, the eight registers comprising bank 0 general registers R0_BANK0 to R7_BANK0 are accessed by the LDC/STC instructions.
  When the RB bit is 0 (that is, when bank 0 is selected), the 16 registers comprising bank 0 general registers R0_BANK0 to R7_BANK0 and non-banked general registers R8 to R15 can be accessed as general registers R0 to R15. In this case, the eight registers comprising bank 1 general registers R0_BANK1 to R7_BANK1 are accessed by the LDC/STC instructions.
- User mode

  In user mode, the 16 registers comprising bank 0 general registers R0_BANK0 to R7_BANK0 and non-banked general registers R8 to R15 can be accessed as general registers R0 to R15. The eight registers comprising bank 1 general registers R0_BANK1 to R7_BANK1 cannot be accessed.

(3) Control Registers

Control registers comprise the global base register (GBR) and status register (SR), which can be accessed in both processing modes, and the saved status register (SSR), saved program counter (SPC), vector base register (VBR), saved general register 15 (SGR), and debug base register (DBR), which can only be accessed in privileged mode. Some bits of the status register (such as the RB bit) can only be accessed in privileged mode.

(4) System Registers

System registers comprise the multiply-and-accumulate registers (MACH/MACL), the procedure register (PR), and the program counter (PC). Access to these registers does not depend on the processing mode.

(5) Floating-Point Registers and System Registers Related to FPU

There are thirty-two floating-point registers, FR0–FR15 and XF0–XF15. FR0–FR15 and XF0–XF15 can be assigned to either of two banks (FPR0_BANK0–FPR15_BANK0 or FPR0_BANK1–FPR15_BANK1).

FR0–FR15 can be used as the eight registers DR0/2/4/6/8/10/12/14 (double-precision floating-point registers, or pair registers) or the four registers FV0/4/8/12 (register vectors), while XF0–XF15 can be used as the eight registers XD0/2/4/6/8/10/12/14 (register pairs) or register matrix XMTRX.

System registers related to the FPU comprise the floating-point communication register (FPUL) and the floating-point status/control register (FPSCR). These registers are used for communication between the FPU and the CPU, and the exception handling setting.

Register values after a reset are shown in table 2.1.

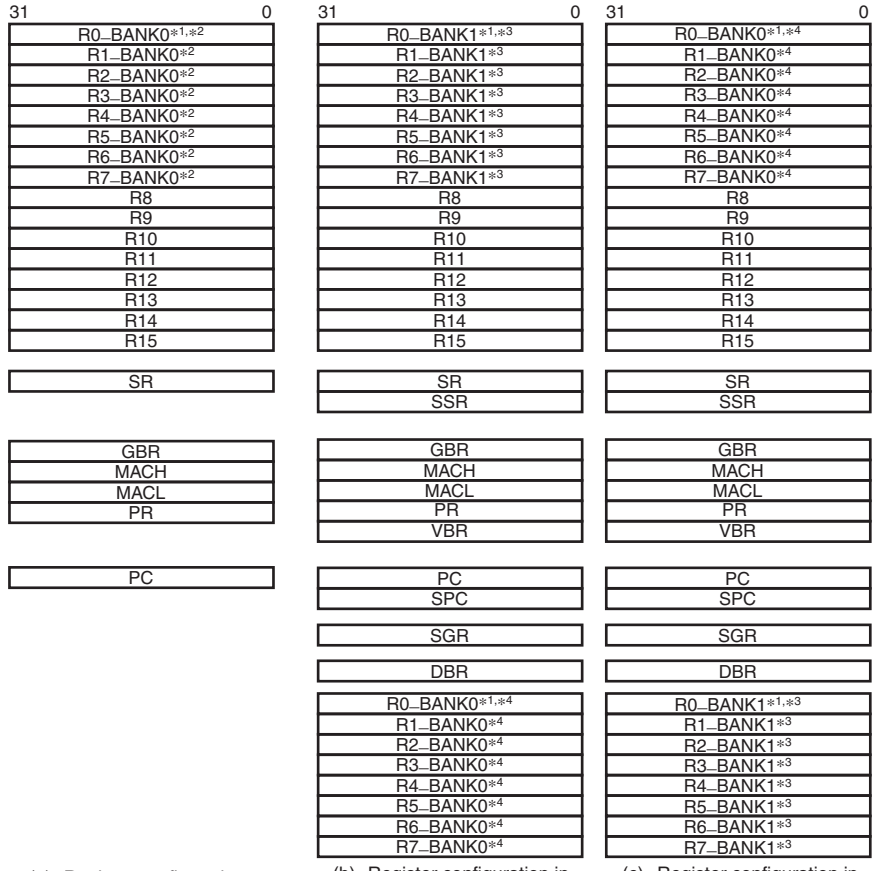
Table 2.1 Initial Register Values

Type	Registers	Initial Value*
General registers	R0_BANK0 to R7_BANK0, R0_BANK1 to R7_BANK1, R8 to R15	Undefined
Control registers	SR	MD bit = 1, RB bit = 1, BL bit = 1, FD bit = 0, IMASK = B'1111, reserved bits = 0, others = undefined
	GBR, SSR, SPC, SGR, DBR	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	H'A0000000
Floating-point registers	FR0 to FR15, XF0 to XF15, FPUL	Undefined
	FPSCR	H'00040001

Note: * Initialized by a power-on reset and manual reset.

The CPU register configuration in each processing mode is shown in figure 2.2.

User mode and privileged mode are switched by the processing mode bit (MD) in the status register.



(a) Register configuration in user mode

(b) Register configuration in privileged mode (RB = 1)

(c) Register configuration in privileged mode (RB = 0)

Notes: 1. R0 is used as the index register in indexed register-indirect addressing mode and indexed GBR indirect addressing mode.
2. Banked registers
3. Banked registers
Accessed as general registers when the RB bit is set to 1 in SR. Accessed only by LDC/STC instructions when the RB bit is cleared to 0.
4. Banked registers
Accessed as general registers when the RB bit is cleared to 0 in SR. Accessed only by LDC/STC instructions when the RB bit is set to 1.

Figure 2.2 CPU Register Configuration in Each Processing Mode

2.2.2 General Registers

Figure 2.3 shows the relationship between the processing modes and general registers. This LSI has twenty-four 32-bit general registers (R0_BANK0 to R7_BANK0, R0_BANK1 to R7_BANK1, and R8 to R15). However, only 16 of these can be accessed as general registers R0 to R15 in one processing mode. This LSI has two processing modes, user mode and privileged mode.

- R0_BANK0 to R7_BANK0
  Allocated to R0 to R7 in user mode (SR.MD = 0)
  Allocated to R0 to R7 when SR.RB = 0 in privileged mode (SR.MD = 1).
- R0_BANK1 to R7_BANK1
  Cannot be accessed in user mode.
  Allocated to R0 to R7 when SR.RB = 1 in privileged mode.

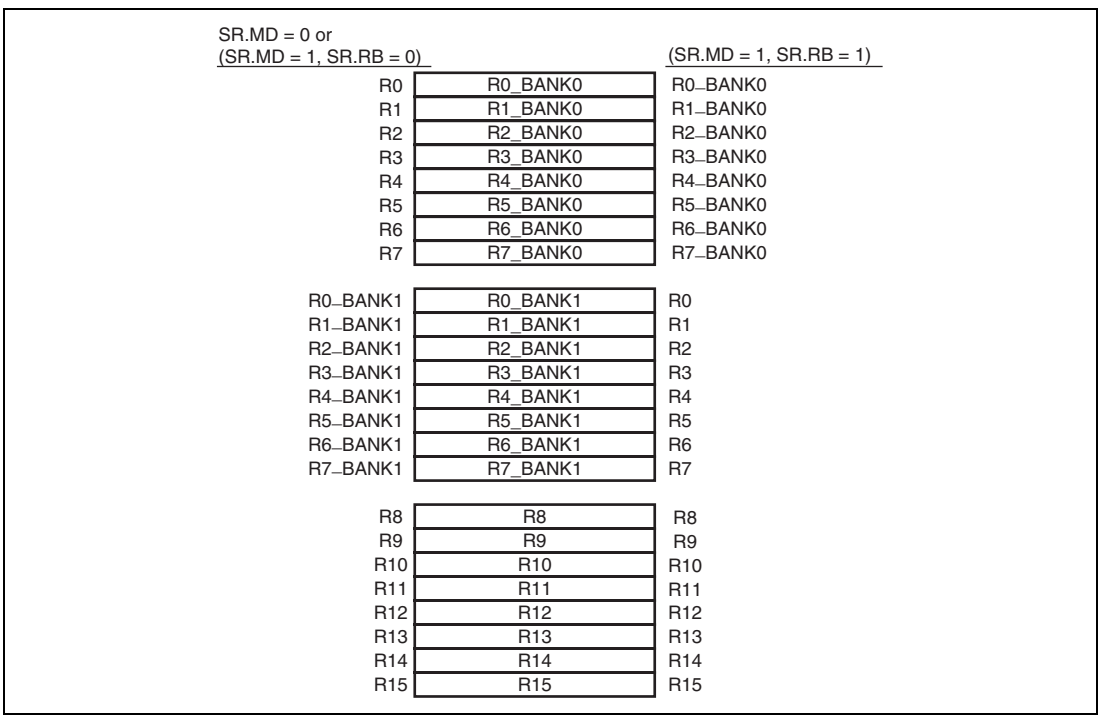


Figure 2.3 General Registers

Note on Programming: As the user's R0 to R7 are assigned to R0_BANK0 to R7_BANK0, and after an exception or interrupt R0 to R7 are assigned to R0_BANK1 to R7_BANK1, it is not necessary for the interrupt handler to save and restore the user's R0 to R7 (R0_BANK0 to R7_BANK0).

2.2.3 Floating-Point Registers

Figure 2.4 shows the floating-point register configuration. There are thirty-two 32-bit floating-point registers, FPR0_BANK0 to FPR15_BANK0, AND FPR0_BANK1 to FPR15_BANK1, comprising two banks. These registers are referenced as FR0 to FR15, DR0/2/4/6/8/10/12/14, FV0/4/8/12, XF0 to XF15, XD0/2/4/6/8/10/12/14, or XMTRX. Reference names of each register are defined depending on the state of the FR bit in FPSCR (see figure 2.4).

1. Floating-point registers, FPRn_BANKj (32 registers)
   FPR0_BANK0 to FPR15_BANK0
   FPR0_BANK1 to FPR15_BANK1

2. Single-precision floating-point registers, FRi (16 registers)
   When FPSCR.FR = 0, FR0 to FR15 are assigned to FPR0_BANK0 to FPR15_BANK0;
   when FPSCR.FR = 1, FR0 to FR15 are assigned to FPR0_BANK1 to FPR15_BANK1.

3. Double-precision floating-point registers or single-precision floating-point registers, DRi (8 registers): A DR register comprises two FR registers.
   DR0 = {FR0, FR1}, DR2 = {FR2, FR3}, DR4 = {FR4, FR5}, DR6 = {FR6, FR7},
   DR8 = {FR8, FR9}, DR10 = {FR10, FR11}, DR12 = {FR12, FR13}, DR14 = {FR14, FR15}

4. Single-precision floating-point vector registers, FVi (4 registers): An FV register comprises four FR registers.
   FV0 = {FR0, FR1, FR2, FR3}, FV4 = {FR4, FR5, FR6, FR7},
   FV8 = {FR8, FR9, FR10, FR11}, FV12 = {FR12, FR13, FR14, FR15}

5. Single-precision floating-point extended registers, XFi (16 registers)
   When FPSCR.FR = 0, XF0 to XF15 are assigned to FPR0_BANK1 to FPR15_BANK1;
   when FPSCR.FR = 1, XF0 to XF15 are assigned to FPR0_BANK0 to FPR15_BANK0.

6. Double-precision floating-point extended registers, XDi (8 registers): An XD register comprises two XF registers.
   XD0 = {XF0, XF1}, XD2 = {XF2, XF3}, XD4 = {XF4, XF5}, XD6 = {XF6, XF7},
   XD8 = {XF8, XF9}, XD10 = {XF10, XF11}, XD12 = {XF12, XF13}, XD14 = {XF14, XF15}

7. Single-precision floating-point extended register matrix, XMTRX: XMTRX comprises all 16 XF registers.

$$\text{XMTRX} = \begin{bmatrix} \text{XF0} & \text{XF4} & \text{XF8} & \text{XF12} \\ \text{XF1} & \text{XF5} & \text{XF9} & \text{XF13} \\ \text{XF2} & \text{XF6} & \text{XF10} & \text{XF14} \\ \text{XF3} & \text{XF7} & \text{XF11} & \text{XF15} \end{bmatrix}$$

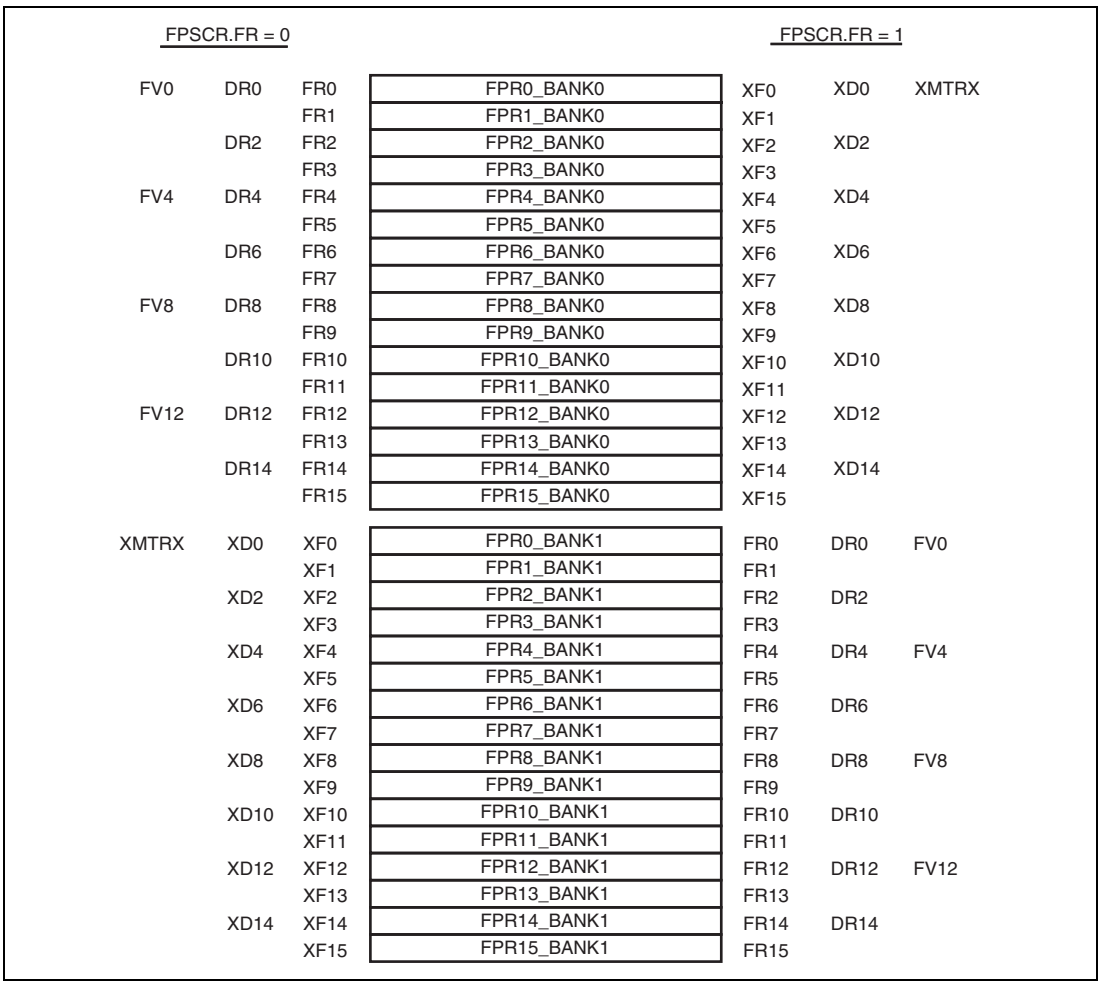


Figure 2.4 Floating-Point Registers

2.2.4 Control Registers

(1) Status Register (SR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	MD	RB	BL	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FD	—	—	—	—	—	M	Q	IMASK				—	—	S	T
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
30	MD	1	R/W	Processing Mode Selects the processing mode. 0: User mode (Some instructions cannot be executed and some resources cannot be accessed.) 1: Privileged mode This bit is set to 1 by an exception or interrupt.
29	RB	1	R/W	Privileged Mode General Register Bank Specification Bit 0: R0_BANK0 to R7_BANK0 are accessed as general registers R0 to R7 and R0_BANK1 to R7_BANK1 can be accessed using LDC/STC instructions 1: R0_BANK1 to R7_BANK1 are accessed as general registers R0 to R7 and R0_BANK0–R7_BANK0 can be accessed using LDC/STC instructions This bit is set to 1 by an exception or interrupt.
28	BL	1	R/W	Exception/Interrupt Block Bit This bit is set to 1 by a reset, a general exception, or an interrupt. While this bit is set to 1, an interrupt request is masked. In this case, this processor enters the reset state when a general exception other than a user break occurs.

Bit	Bit Name	Initial Value	R/W	Description
27 to 16	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
15	FD	0	R/W	FPU Disable Bit When this bit is set to 1 and an FPU instruction is not in a delay slot, a general FPU disable exception occurs. When this bit is set to 1 and an FPU instruction is in a delay slot, a slot FPU disable exception occurs. (FPU instructions: H'F*** instructions and LDS (.L)/STS(.L) instructions using FPUL/FPSCR)
14 to 10	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
9	M	0	R/W	M Bit Used by the DIV0S, DIV0U, and DIV1 instructions.
8	Q	0	R/W	Q Bit Used by the DIV0S, DIV0U, and DIV1 instructions.
7 to 4	IMASK	1111	R/W	Interrupt Mask Level Bits An interrupt whose priority is equal to or less than the value of the IMASK bits is masked. It can be chosen by CPU operation mode register (CPUOPM) whether the level of IMASK is changed to accept an interrupt or not when an interrupt is occurred. For details, see appendix A, CPU Operation Mode Register (CPUOPM).
3, 2	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
1	S	0	R/W	S Bit Used by the MAC instruction.
0	T	0	R/W	T Bit Indicates true/false condition, carry/borrow, or overflow/underflow. For details, see section 3, Instruction Set.

(2) Saved Status Register (SSR) (32 bits, Privileged Mode, Initial Value = Undefined)

The contents of SR are saved to SSR in the event of an exception or interrupt.

(3) Saved Program Counter (SPC) (32 bits, Privileged Mode, Initial Value = Undefined)

The address of an instruction at which an interrupt or exception occurs is saved to SPC.

(4) Global Base Register (GBR) (32 bits, Initial Value = Undefined)

GBR is referenced as the base address of addressing @(disp,GBR) and @(R0,GBR).

(5) Vector Base Register (VBR) (32 bits, Privileged Mode, Initial Value = H'00000000)

VBR is referenced as the branch destination base address in the event of an exception or interrupt. For details, see section 5, Exception Handling.

(6) Saved General Register 15 (SGR) (32 bits, Privileged Mode, Initial Value = Undefined)

The contents of R15 are saved to SGR in the event of an exception or interrupt.

(7) Debug Base Register (DBR) (32 bits, Privileged Mode, Initial Value = Undefined)

When the user break debugging function is enabled (CBCR.UBDE = 1), DBR is referenced as the branch destination address of the user break handler instead of VBR.

2.2.5 System Registers

(1) Multiply-and-Accumulate Registers (MACH and MACL) (32 bits, Initial Value = Undefined)

MACH and MACL are used for the added value in a MAC instruction, and to store the operation result of a MAC or MUL instruction.

(2) Procedure Register (PR) (32 bits, Initial Value = Undefined)

The return address is stored in PR in a subroutine call using a BSR, BSRF, or JSR instruction. PR is referenced by the subroutine return instruction (RTS).

(3) Program Counter (PC) (32 bits, Initial Value = H'A0000000)

PC indicates the address of the instruction currently being executed.

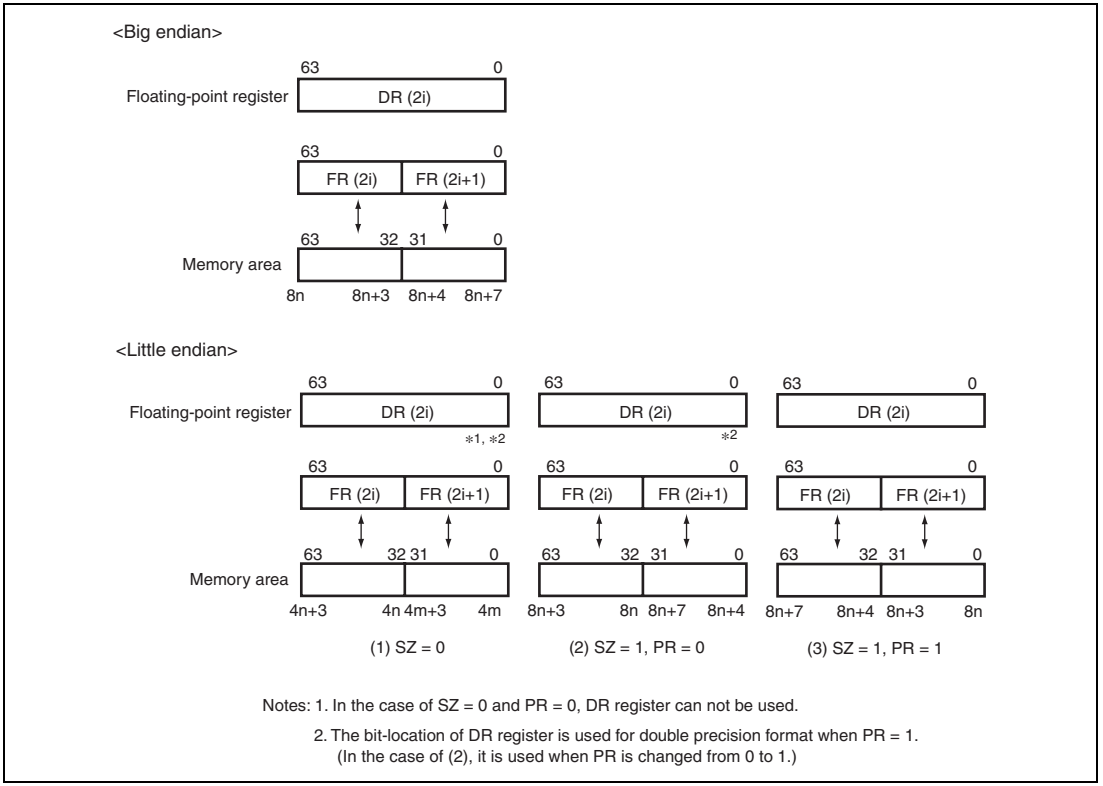
(4) Floating-Point Status/Control Register (FPSCR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	FR	SZ	PR	DN	Cause	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Cause				Enable (EN)					Flag					RM	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
21	FR	0	R/W	Floating-Point Register Bank 0: FPR0_BANK0 to FPR15_BANK0 are assigned to FR0 to FR15 and FPR0_BANK1 to FPR15_BANK1 are assigned to XF0 to XF15 1: FPR0_BANK0 to FPR15_BANK0 are assigned to XF0 to XF15 and FPR0_BANK1 to FPR15_BANK1 are assigned to FR0 to FR15
20	SZ	0	R/W	Transfer Size Mode 0: Data size of FMOV instruction is 32-bits 1: Data size of FMOV instruction is a 32-bit register pair (64 bits) For relationship between the SZ bit, PR bit, and endian, see figure 2.5.
19	PR	0	R/W	Precision Mode 0: Floating-point instructions are executed as single-precision operations 1: Floating-point instructions are executed as double-precision operations (graphics support instructions are undefined) For relationship between the SZ bit, PR bit, and endian, see figure 2.5
18	DN	1	R/W	Denormalization Mode 0: Denormalized number is treated as such 1: Denormalized number is treated as zero

Bit	Bit Name	Initial Value	R/W	Description
17 to 12	Cause	000000	R/W	FPU Exception Cause Field FPU Exception Enable Field FPU Exception Flag Field Each time an FPU operation instruction is executed, the FPU exception cause field is cleared to 0. When an FPU exception occurs, the bits corresponding to FPU exception cause field and flag field are set to 1. The FPU exception flag field remains set to 1 until it is cleared to 0 by software. For bit allocations of each field, see table 2.2.
11 to 7	Enable (EN)	00000	R/W	
6 to 2	Flag	00000	R/W	
1, 0	RM	01	R/W	Rounding Mode These bits select the rounding mode. 00: Round to Nearest 01: Round to Zero 10: Reserved 11: Reserved



Notes: 1. In the case of SZ = 0 and PR = 0, DR register can not be used.

2. The bit-location of DR register is used for double precision format when PR = 1.
(In the case of (2), it is used when PR is changed from 0 to 1.)

Figure 2.5 Relationship between SZ bit and Endian

Table 2.2 Bit Allocation for FPU Exception Handling

Field Name		FPU Error (E)	Invalid Operation (V)	Division by Zero (Z)	Overflow (O)	Underflow (U)	Inexact (I)
Cause	FPU exception cause field	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12
Enable	FPU exception enable field	None	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7
Flag	FPU exception flag field	None	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2

(5) Floating-Point Communication Register (FPUL) (32 bits, Initial Value = Undefined)

Information is transferred between the FPU and CPU via FPUL.

2.3 Memory-Mapped Registers

Some control registers are mapped to the following memory areas. Each of the mapped registers has two addresses.

H'1C00 0000 to H'1FFF FFFF
H'FC00 0000 to H'FFFF FFFF

These two areas are used as follows.

- H'1C00 0000 to H'1FFF FFFF

  This area must be accessed using the address translation function of the MMU.

  Setting the page number of this area to the corresponding field of the TLB enables access to a memory-mapped register.

  The operation of an access to this area without using the address translation function of the MMU is not guaranteed.
- H'FC00 0000 to H'FFFF FFFF

  Access to area H'FC00 0000 to H'FFFF FFFF in user mode will cause an address error. Memory-mapped registers can be referenced in user mode by means of access that involves address translation.

Note: Do not access addresses to which registers are not mapped in either area. The operation of an access to an address with no register mapped is undefined. Also, memory-mapped registers must be accessed using a fixed data size. The operation of an access using an invalid data size is undefined.

2.4 Data Formats in Registers

Register operands are always longwords (32 bits). When a memory operand is only a byte (8 bits) or a word (16 bits), it is sign-extended into a longword when loaded into a register.

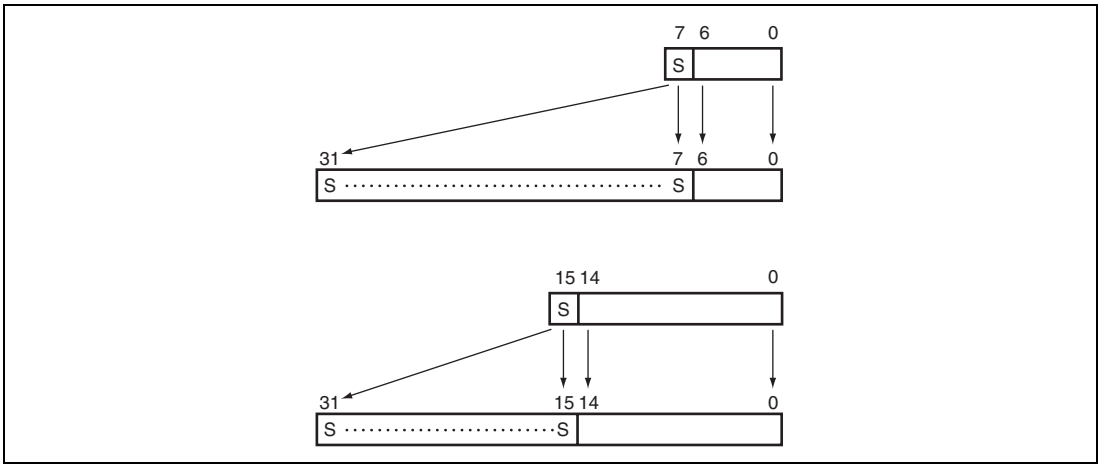


Figure 2.6 Formats of Byte Data and Word Data in Register

2.5 Data Formats in Memory

Memory data formats are classified into bytes, words, and longwords. Memory can be accessed in an 8-bit byte, 16-bit word, or 32-bit longword form. A memory operand less than 32 bits in length is sign-extended before being loaded into a register.

A word operand must be accessed starting from a word boundary (even address of a 2-byte unit: address 2n), and a longword operand starting from a longword boundary (even address of a 4-byte unit: address 4n). An address error will result if this rule is not observed. A byte operand can be accessed from any address.

Big endian or little endian byte order can be selected for the data format. The endian should be set with the external pin after a power-on reset. The endian cannot be changed dynamically. Bit positions are numbered left to right from most-significant to least-significant. Thus, in a 32-bit longword, the leftmost bit, bit 31, is the most significant bit and the rightmost bit, bit 0, is the least significant bit.

The data format in memory is shown in figure 2.7.

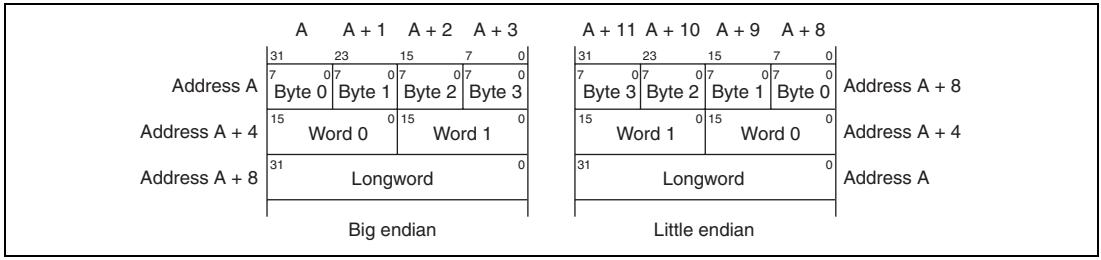


Figure 2.7 Data Formats in Memory

For the 64-bit data format, see figure 2.5.

2.6 Processing States

This LSI has major three processing states: the reset state, instruction execution state, and power-down state.

(1) Reset State

In this state the CPU is reset. The reset state is divided into the power-on reset state and the manual reset.

In the power-on reset state, the internal state of the CPU and the on-chip peripheral module registers are initialized. In the manual reset state, the internal state of the CPU and some registers of on-chip peripheral modules are initialized. For details, see register descriptions for each section of the hardware manual of the product.

(2) Instruction Execution State

In this state, the CPU executes program instructions in sequence. The Instruction execution state has the normal program execution state and the exception handling state.

(3) Power-Down State

In a power-down state, CPU halts operation and power consumption is reduced. The power-down state is entered by executing a SLEEP instruction. There are two modes in the power-down state: sleep mode and standby mode. For details, see the Power-Down section of the hardware manual of the product.

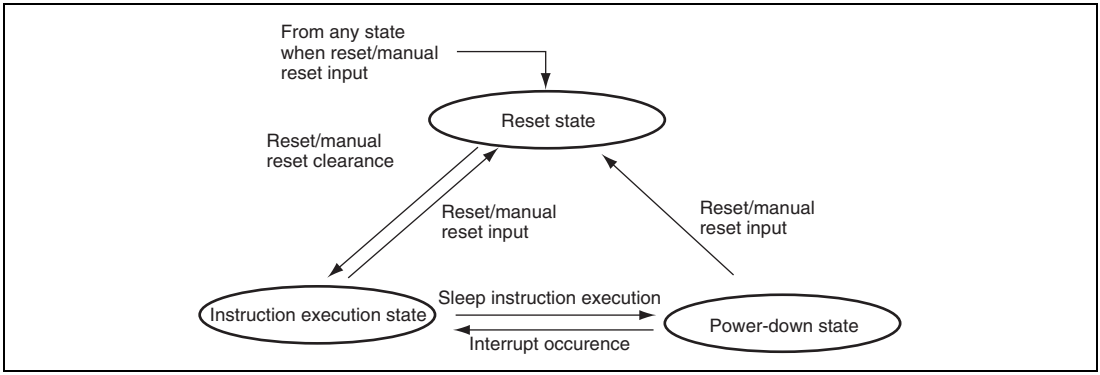


Figure 2.8 Processing State Transitions

2.7 Usage Notes

2.7.1 Notes on Self-Modifying Code

To accelerate the processing speed, the instruction prefetching capability of this LSI has been significantly enhanced from that of the SH-4. Therefore, in the case when a code in memory is rewritten and attempted to be executed immediately, there is increased possibility that the code before being modified, which has already been prefetched, is executed.

To ensure execution of the modified code, one of the following sequence of instructions should be executed between the code rewriting instruction and execution of the modified code.

(1) When the Codes to be Modified are in Non-Cacheable Area

```
SYNCO
ICBI @Rn
```

The target for the ICBI instruction can be any address within the range where no address error exception occurs.

(2) When the Codes to be Modified are in Cacheable Area (Write-Through)

```
SYNCO
ICBI @Rn
```

All instruction cache areas corresponding to the modified codes should be invalidated by the ICBI instruction. The ICBI instruction should be issued to each cache line. One cache line is 32 bytes.

(3) When the Codes to be Modified are in Cacheable Area (Copy-Back)

```
OCBP @Rm or OCBWB @Rm
SYNCO
ICBI @Rn
```

All operand cache areas corresponding to the modified codes should be written back to the main memory by the OCBP or OCBWB instruction. Then all instruction cache areas corresponding to the modified codes should be invalidated by the ICBI instruction. The OCBP, OCBWB, and ICBI instruction should be issued to each cache line. One cache line is 32 bytes.

Note: Self-modifying code is the processing which executes instructions while dynamically rewriting the codes in memory.

Section 3 Instruction Set

This LSI's instruction set is implemented with 16-bit fixed-length instructions. This LSI can use byte (8-bit), word (16-bit), longword (32-bit), and quadword (64-bit) data sizes for memory access. Single-precision floating-point data (32 bits) can be moved to and from memory using longword or quadword size. Double-precision floating-point data (64 bits) can be moved to and from memory using longword size. When this LSI moves byte-size or word-size data from memory to a register, the data is sign-extended.

3.1 Execution Environment

(1) PC

At the start of instruction execution, the PC indicates the address of the instruction itself.

(2) Load-Store Architecture

This LSI has a load-store architecture in which operations are basically executed using registers. Except for bit-manipulation operations such as logical AND that are executed directly in memory, operands in an operation that requires memory access are loaded into registers and the operation is executed between the registers.

(3) Delayed Branches

Except for the two branch instructions BF and BT, this LSI's branch instructions and RTE are delayed branches. In a delayed branch, the instruction following the branch is executed before the branch destination instruction.

(4) Delay Slot

This execution slot following a delayed branch is called a delay slot. For example, the BRA execution sequence is as follows:

Table 3.1 Execution Order of Delayed Branch Instructions

	Instructions			Execution Order
	BRA	TARGET	(Delayed branch instruction)	BRA
	ADD		(Delay slot)	↓
	:			ADD
	:			↓
TARGET	target-inst		(Branch destination instruction)	target-inst

A slot illegal instruction exception may occur when a specific instruction is executed in a delay slot. For details, see section 5, Exception Handling. The instruction following BF/S or BT/S for which the branch is not taken is also a delay slot instruction.

(5) T Bit

The T bit in SR is used to show the result of a compare operation, and is referenced by a conditional branch instruction. An example of the use of a conditional branch instruction is shown below.

```
ADD     #1, R0      ; T bit is not changed by ADD operation
CMP/EQ  R1, R0      ; If R0 = R1, T bit is set to 1
BT      TARGET      ; Branches to TARGET if T bit = 1 (R0 = R1)
```

In an RTE delay slot, the SR bits are referenced as follows. In instruction access, the MD bit is used before modification, and in data access, the MD bit is accessed after modification. The other bits—S, T, M, Q, FD, BL, and RB—after modification are used for delay slot instruction execution. The STC and STC.L SR instructions access all SR bits after modification.

(6) Constant Values

An 8-bit constant value can be specified by the instruction code and an immediate value. 16-bit and 32-bit constant values can be defined as literal constant values in memory, and can be referenced by a PC-relative load instruction.

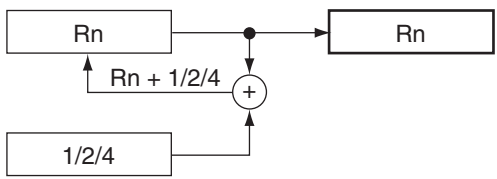
```
MOV.W   @(disp, PC), Rn
MOV.L   @(disp, PC), Rn
```

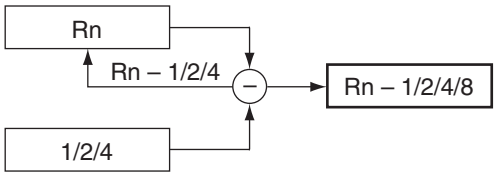
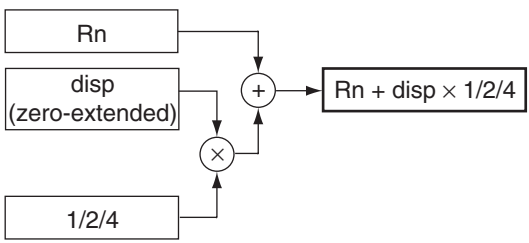
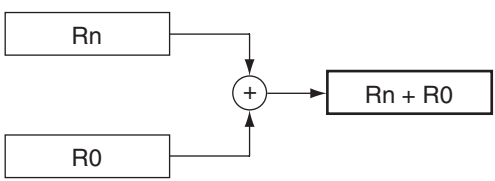
There are no PC-relative load instructions for floating-point operations. However, it is possible to set 0.0 or 1.0 by using the FLDI0 or FLDI1 instruction on a single-precision floating-point register.

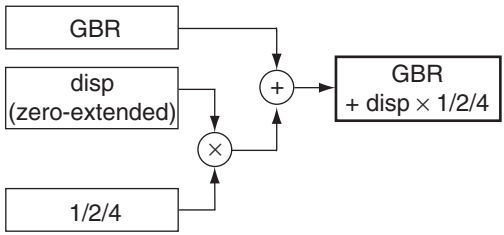
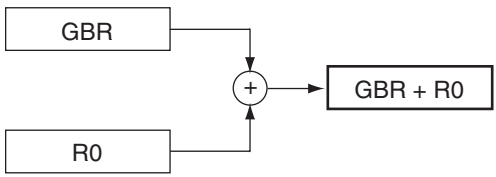
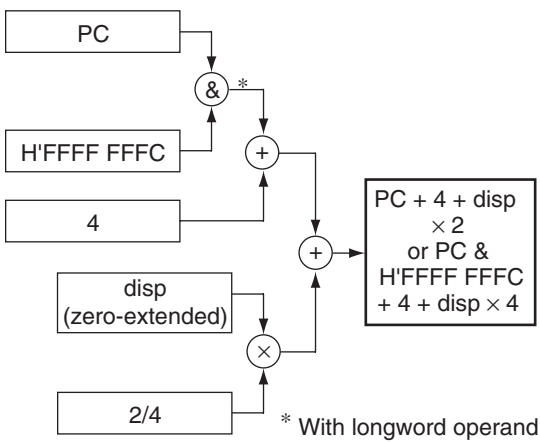
3.2 Addressing Modes

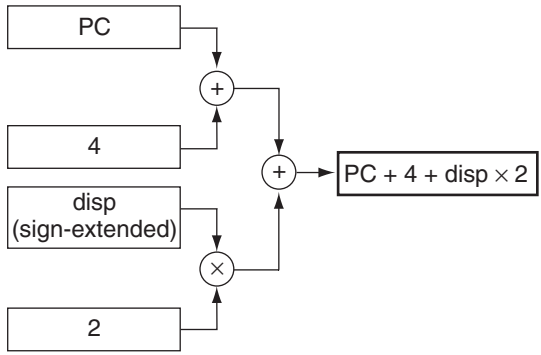
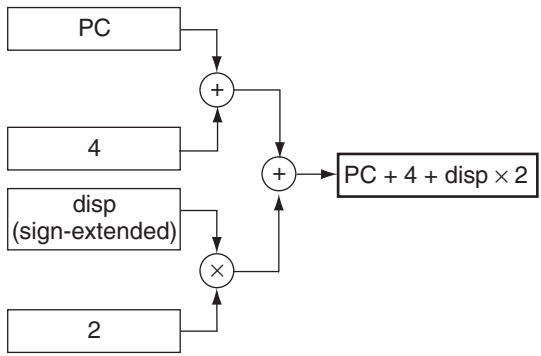
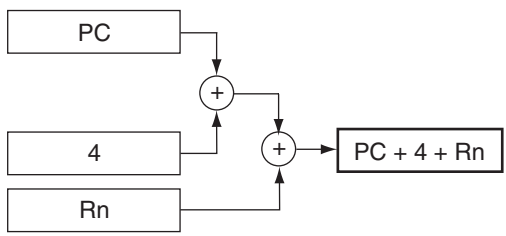
Addressing modes and effective address calculation methods are shown in table 3.2. When a location in virtual memory space is accessed (AT in MMUCR = 1), the effective address is translated into a physical memory address. If multiple virtual memory space systems are selected (SV in MMUCR = 0), the least significant bit of PTEH is also referenced as the access ASID. For details, see section 7, Memory Management Unit (MMU).

Table 3.2 Addressing Modes and Effective Addresses

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register direct	Rn	Effective address is register Rn. (Operand is register Rn contents.)	—
Register indirect	@Rn	Effective address is register Rn contents. 	Rn → EA (EA: effective address)
Register indirect with post-increment	@Rn+	Effective address is register Rn contents. A constant is added to Rn after instruction execution: 1 for a byte operand, 2 for a word operand, 4 for a longword operand, 8 for a quadword operand. 	Rn → EA After instruction execution Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn Quadword: Rn + 8 → Rn

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register indirect with pre-decrement	@–Rn	Effective address is register Rn contents, decremented by a constant beforehand: 1 for a byte operand, 2 for a word operand, 4 for a longword operand, 8 for a quadword operand. 	Byte: Rn – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn Quadword: Rn – 8 → Rn Rn → EA (Instruction executed with Rn after calculation)
Register indirect with displacement	@(disp:4, Rn)	Effective address is register Rn contents with 4-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size. 	Byte: Rn + disp → EA Word: Rn + disp × 2 → EA Longword: Rn + disp × 4 → EA
Indexed register indirect	@(R0, Rn)	Effective address is sum of register Rn and R0 contents. 	Rn + R0 → EA

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
GBR indirect with displacement	@(disp:8, GBR)	Effective address is register GBR contents with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size. 	Byte: GBR + disp → EA Word: GBR + disp × 2 → EA Longword: GBR + disp × 4 → EA
Indexed GBR indirect	@(R0, GBR)	Effective address is sum of register GBR and R0 contents. 	GBR + R0 → EA
PC-relative with displacement	@(disp:8, PC)	Effective address is PC + 4 with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 2 (word), or 4 (longword), according to the operand size. With a longword operand, the lower 2 bits of PC are masked. 	Word: PC + 4 + disp × 2 → EA Longword: PC & H'FFFF FFFC + 4 + disp × 4 → EA

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
PC-relative	disp:8	Effective address is PC + 4 with 8-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + 4 + disp × 2 → Branch-Target
	disp:12	Effective address is PC + 4 with 12-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + 4 + disp × 2 → Branch-Target
	Rn	Effective address is sum of PC + 4 and Rn. 	PC + 4 + Rn → Branch-Target

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Immediate	#imm:8	8-bit immediate data imm of TST, AND, OR, or XOR instruction is zero-extended.	—
	#imm:8	8-bit immediate data imm of MOV, ADD, or CMP/EQ instruction is sign-extended.	—
	#imm:8	8-bit immediate data imm of TRAPA instruction is zero-extended and multiplied by 4.	—

Note: For the addressing modes below that use a displacement (disp), the assembler descriptions in this manual show the value before scaling (×1, ×2, or ×4) is performed according to the operand size. This is done to clarify the operation of the LSI. Refer to the relevant assembler notation rules for the actual assembler descriptions.

@ (disp:4, Rn) ; Register indirect with displacement
@ (disp:8, GBR) ; GBR indirect with displacement
@ (disp:8, PC) ; PC-relative with displacement
disp:8, disp:12 ; PC-relative

3.3 Instruction Set

Table 3.3 shows the notation used in the SH instruction lists shown in tables 3.4 to 3.13.

Table 3.3 Notation Used in Instruction List

Item	Format	Description	
Instruction mnemonic	OP.Sz SRC, DEST	OP:	Operation code
		Sz:	Size
		SRC:	Source operand
		DEST:	Source and/or destination operand
		Rm:	Source register
		Rn:	Destination register
		imm:	Immediate data
		disp:	Displacement
Operation notation		→, ←	Transfer direction
		(xx)	Memory operand
		M/Q/T	SR flag bits
		&	Logical AND of individual bits
		|	Logical OR of individual bits
		∧	Logical exclusive-OR of individual bits
		~	Logical NOT of individual bits
		<<n, >>n	n-bit shift
Instruction code	MSB ↔ LSB	mmmm:	Register number (Rm, FRm)
		nnnn:	Register number (Rn, FRn)
		0000:	R0, FR0
		0001:	R1, FR1
		:	
		1111:	R15, FR15
		mmm:	Register number (DRm, XDm, Rm_BANK)
		nnn:	Register number (DRn, XDn, Rn_BANK)
		000:	DR0, XD0, R0_BANK
		001:	DR2, XD2, R1_BANK
		:	
		111:	DR14, XD14, R7_BANK
		mm:	Register number (FVm)
		nn:	Register number (FVn)
		00:	FV0
		01:	FV4
		10:	FV8
		11:	FV12
		iiii:	Immediate data
		dddd:	Displacement

Item	Format	Description
Privileged mode		"Privileged" means the instruction can only be executed in privileged mode.
T bit	Value of T bit after instruction execution	—: No change

Note: Scaling (×1, ×2, ×4, or ×8) is executed according to the size of the instruction operand.

Table 3.4 Fixed-Point Transfer Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit
MOV	#imm,Rn	imm → sign extension → Rn	1110nnnniiiiiiii	—	—
MOV.W	@(disp*,PC), Rn	(disp × 2 + PC + 4) → sign extension → Rn	1001nnnnddddddd	—	—
MOV.L	@(disp*,PC), Rn	(disp × 4 + PC & H'FFFF FFFC + 4) → Rn	1101nnnnddddddd	—	—
MOV	Rm,Rn	Rm → Rn	0110nnnnmmmm0011	—	—
MOV.B	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0000	—	—
MOV.W	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0001	—	—
MOV.L	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0010	—	—
MOV.B	@Rm,Rn	(Rm) → sign extension → Rn	0110nnnnmmmm0000	—	—
MOV.W	@Rm,Rn	(Rm) → sign extension → Rn	0110nnnnmmmm0001	—	—
MOV.L	@Rm,Rn	(Rm) → Rn	0110nnnnmmmm0010	—	—
MOV.B	Rm,@-Rn	Rn-1 → Rn, Rm → (Rn)	0010nnnnmmmm0100	—	—
MOV.W	Rm,@-Rn	Rn-2 → Rn, Rm → (Rn)	0010nnnnmmmm0101	—	—
MOV.L	Rm,@-Rn	Rn-4 → Rn, Rm → (Rn)	0010nnnnmmmm0110	—	—
MOV.B	@Rm+,Rn	(Rm)→ sign extension → Rn, Rm + 1 → Rm	0110nnnnmmmm0100	—	—
MOV.W	@Rm+,Rn	(Rm) → sign extension → Rn, Rm + 2 → Rm	0110nnnnmmmm0101	—	—
MOV.L	@Rm+,Rn	(Rm) → Rn, Rm + 4 → Rm	0110nnnnmmmm0110	—	—
MOV.B	R0,@(disp*,Rn)	R0 → (disp + Rn)	10000000nnnndddd	—	—
MOV.W	R0,@(disp*,Rn)	R0 → (disp × 2 + Rn)	10000001nnnndddd	—	—
MOV.L	Rm,@(disp*,Rn)	Rm → (disp × 4 + Rn)	0001nnnnmmmmdddd	—	—
MOV.B	@(disp*,Rm),R0	(disp + Rm) → sign extension → R0	10000100mmmmdddd	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit
MOV.W	@(disp*,Rm),R0	(disp × 2 + Rm) → sign extension → R0	10000101mmmmdddd	—	—
MOV.L	@(disp*,Rm),Rn	(disp × 4 + Rm) → Rn	0101nnnnmmmmdddd	—	—
MOV.B	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0100	—	—
MOV.W	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0101	—	—
MOV.L	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0110	—	—
MOV.B	@(R0,Rm),Rn	(R0 + Rm) → sign extension → Rn	0000nnnnmmmm1100	—	—
MOV.W	@(R0,Rm),Rn	(R0 + Rm) → sign extension → Rn	0000nnnnmmmm1101	—	—
MOV.L	@(R0,Rm),Rn	(R0 + Rm) → Rn	0000nnnnmmmm1110	—	—
MOV.B	R0,@(disp*,GBR)	R0 → (disp + GBR)	11000000dddddddd	—	—
MOV.W	R0,@(disp*,GBR)	R0 → (disp × 2 + GBR)	11000001dddddddd	—	—
MOV.L	R0,@(disp*,GBR)	R0 → (disp × 4 + GBR)	11000010dddddddd	—	—
MOV.B	@(disp*,GBR),R0	(disp + GBR) → sign extension → R0	11000100dddddddd	—	—
MOV.W	@(disp*,GBR),R0	(disp × 2 + GBR) → sign extension → R0	11000101dddddddd	—	—
MOV.L	@(disp*,GBR),R0	(disp × 4 + GBR) → R0	11000110dddddddd	—	—
MOVA	@(disp*,PC),R0	disp × 4 + PC & H'FFFF FFFC + 4 → R0	11000111dddddddd	—	—
MOVCO.L	R0,@Rn	LDST → T If (T == 1) R0 → (Rn) 0 → LDST	0000nnnn01110011	—	LDST
MOVLI.L	@Rm,R0	1 → LDST (Rm) → R0 When interrupt/exception occurred 0 → LDST	0000mmmm01100011	—	—
MOVUA.L	@Rm,R0	(Rm) → R0 Load non-boundary alignment data	0100mmmm10101001	—	—
MOVUA.L	@Rm+,R0	(Rm) → R0, Rm + 4 → Rm Load non-boundary alignment data	0100mmmm11101001	—	—
MOVT	Rn	T → Rn	0000nnnn00101001	—	—
SWAP.B	Rm,Rn	Rm → swap lower 2 bytes → Rn	0110nnnnmmmm1000	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit
SWAP.W	Rm,Rn	Rm → swap upper/lower words → Rn	0110nnnnmmmm1001	—	—
XTRCT	Rm,Rn	Rm:Rn middle 32 bits → Rn	0010nnnnmmmm1101	—	—

Note: * The assembler of Renesas uses the value after scaling (×1, ×2, or ×4) as the displacement (disp).

Table 3.5 Arithmetic Operation Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit
ADD	Rm,Rn	Rn + Rm → Rn	0011nnnnmmmm1100	—	—
ADD	#imm,Rn	Rn + imm → Rn	0111nnnniiiiiiii	—	—
ADDC	Rm,Rn	Rn + Rm + T → Rn, carry → T	0011nnnnmmmm1110	—	Carry
ADDV	Rm,Rn	Rn + Rm → Rn, overflow → T	0011nnnnmmmm1111	—	Overflow
CMP/EQ	#imm,R0	When R0 = imm, 1 → T Otherwise, 0 → T	10001000iiiiiiii	—	Comparison result
CMP/EQ	Rm,Rn	When Rn = Rm, 1 → T Otherwise, 0 → T	0011nnnnmmmm0000	—	Comparison result
CMP/HS	Rm,Rn	When Rn ≥ Rm (unsigned), 1 → T Otherwise, 0 → T	0011nnnnmmmm0010	—	Comparison result
CMP/GE	Rm,Rn	When Rn ≥ Rm (signed), 1 → T Otherwise, 0 → T	0011nnnnmmmm0011	—	Comparison result
CMP/HI	Rm,Rn	When Rn > Rm (unsigned), 1 → T Otherwise, 0 → T	0011nnnnmmmm0110	—	Comparison result
CMP/GT	Rm,Rn	When Rn > Rm (signed), 1 → T Otherwise, 0 → T	0011nnnnmmmm0111	—	Comparison result
CMP/PZ	Rn	When Rn ≥ 0, 1 → T Otherwise, 0 → T	0100nnnn00010001	—	Comparison result
CMP/PL	Rn	When Rn > 0, 1 → T Otherwise, 0 → T	0100nnnn00010101	—	Comparison result
CMP/STR	Rm,Rn	When any bytes are equal, 1 → T Otherwise, 0 → T	0010nnnnmmmm1100	—	Comparison result
DIV1	Rm,Rn	1-step division (Rn ÷ Rm)	0011nnnnmmmm0100	—	Calculation result

Instruction		Operation	Instruction Code	Privileged	T Bit
DIV0S	Rm,Rn	MSB of Rn → Q, MSB of Rm → M, M^Q → T	0010nnnnmmmm0111	—	Calculation result
DIV0U		0 → M/Q/T	0000000000011001	—	0
DMULS.L	Rm,Rn	Signed, Rn × Rm → MAC, 32 × 32 → 64 bits	0011nnnnmmmm1101	—	—
DMULU.L	Rm,Rn	Unsigned, Rn × Rm → MAC, 32 × 32 → 64 bits	0011nnnnmmmm0101	—	—
DT	Rn	Rn – 1 → Rn; when Rn = 0, 1 → T When Rn ≠ 0, 0 → T	0100nnnn00010000	—	Comparison result
EXTS.B	Rm,Rn	Rm sign-extended from byte → Rn	0110nnnnmmmm1110	—	—
EXTS.W	Rm,Rn	Rm sign-extended from word → Rn	0110nnnnmmmm1111	—	—
EXTU.B	Rm,Rn	Rm zero-extended from byte → Rn	0110nnnnmmmm1100	—	—
EXTU.W	Rm,Rn	Rm zero-extended from word → Rn	0110nnnnmmmm1101	—	—
MAC.L	@Rm+,@Rn+	Signed, (Rn) × (Rm) + MAC → MAC Rn + 4 → Rn, Rm + 4 → Rm 32 × 32 + 64 → 64 bits	0000nnnnmmmm1111	—	—
MAC.W	@Rm+,@Rn+	Signed, (Rn) × (Rm) + MAC → MAC Rn + 2 → Rn, Rm + 2 → Rm 16 × 16 + 64 → 64 bits	0100nnnnmmmm1111	—	—
MUL.L	Rm,Rn	Rn × Rm → MACL 32 × 32 → 32 bits	0000nnnnmmmm0111	—	—
MULS.W	Rm,Rn	Signed, Rn × Rm → MACL 16 × 16 → 32 bits	0010nnnnmmmm1111	—	—
MULU.W	Rm,Rn	Unsigned, Rn × Rm → MACL 16 × 16 → 32 bits	0010nnnnmmmm1110	—	—
NEG	Rm,Rn	0 – Rm → Rn	0110nnnnmmmm1011	—	—
NEGC	Rm,Rn	0 – Rm – T → Rn, borrow → T	0110nnnnmmmm1010	—	Borrow
SUB	Rm,Rn	Rn – Rm → Rn	0011nnnnmmmm1000	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit
SUBC	Rm,Rn	Rn – Rm – T → Rn, borrow → T	0011nnnnmmmm1010	—	Borrow
SUBV	Rm,Rn	Rn – Rm → Rn, underflow → T	0011nnnnmmmm1011	—	Underflow

Table 3.6 Logic Operation Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit
AND	Rm,Rn	Rn & Rm → Rn	0010nnnnmmmm1001	—	—
AND	#imm,R0	R0 & imm → R0	11001001iiiiiiii	—	—
AND.B	#imm, @(R0,GBR)	(R0 + GBR) & imm → (R0 + GBR)	11001101iiiiiiii	—	—
NOT	Rm,Rn	~Rm → Rn	0110nnnnmmmm0111	—	—
OR	Rm,Rn	Rn | Rm → Rn	0010nnnnmmmm1011	—	—
OR	#imm,R0	R0 | imm → R0	11001011iiiiiiii	—	—
OR.B	#imm, @(R0,GBR)	(R0 + GBR) | imm → (R0 + GBR)	11001111iiiiiiii	—	—
TAS.B	@Rn	When (Rn) = 0, 1 → T Otherwise, 0 → T In both cases, 1 → MSB of (Rn)	0100nnnn00011011	—	Test result
TST	Rm,Rn	Rn & Rm; when result = 0, 1 → T Otherwise, 0 → T	0010nnnnmmmm1000	—	Test result
TST	#imm,R0	R0 & imm; when result = 0, 1 → T Otherwise, 0 → T	11001000iiiiiiii	—	Test result
TST.B	#imm, @(R0,GBR)	(R0 + GBR) & imm; when result = 0, 1 → T Otherwise, 0 → T	11001100iiiiiiii	—	Test result
XOR	Rm,Rn	Rn ∧ Rm → Rn	0010nnnnmmmm1010	—	—
XOR	#imm,R0	R0 ∧ imm → R0	11001010iiiiiiii	—	—
XOR.B	#imm, @(R0,GBR)	(R0 + GBR) ∧ imm → (R0 + GBR)	11001110iiiiiiii	—	—

Table 3.7 Shift Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit
ROTL	Rn	T ← Rn ← MSB	0100nnnn00000100	—	MSB
ROTR	Rn	LSB → Rn → T	0100nnnn00000101	—	LSB
ROTCL	Rn	T ← Rn ← T	0100nnnn00100100	—	MSB
ROTCR	Rn	T → Rn → T	0100nnnn00100101	—	LSB
SHAD	Rm,Rn	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> Rm → [MSB → Rn]	0100nnnnmmmm1100	—	—
SHAL	Rn	T ← Rn ← 0	0100nnnn00100000	—	MSB
SHAR	Rn	MSB → Rn → T	0100nnnn00100001	—	LSB
SHLD	Rm,Rn	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> Rm → [0 → Rn]	0100nnnnmmmm1101	—	—
SHLL	Rn	T ← Rn ← 0	0100nnnn00000000	—	MSB
SHLR	Rn	0 → Rn → T	0100nnnn00000001	—	LSB
SHLL2	Rn	Rn << 2 → Rn	0100nnnn00001000	—	—
SHLR2	Rn	Rn >> 2 → Rn	0100nnnn00001001	—	—
SHLL8	Rn	Rn << 8 → Rn	0100nnnn00011000	—	—
SHLR8	Rn	Rn >> 8 → Rn	0100nnnn00011001	—	—
SHLL16	Rn	Rn << 16 → Rn	0100nnnn00101000	—	—
SHLR16	Rn	Rn >> 16 → Rn	0100nnnn00101001	—	—

Table 3.8 Branch Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit
BF	label	When T = 0, disp × 2 + PC + 4 → PC When T = 1, nop	10001011dddddddd	—	—
BF/S	label	Delayed branch; when T = 0, disp × 2 + PC + 4 → PC When T = 1, nop	10001111dddddddd	—	—
BT	label	When T = 1, disp × 2 + PC + 4 → PC When T = 0, nop	10001001dddddddd	—	—
BT/S	label	Delayed branch; when T = 1, disp × 2 + PC + 4 → PC When T = 0, nop	10001101dddddddd	—	—
BRA	label	Delayed branch, disp × 2 + PC + 4 → PC	1010dddddddddddd	—	—
BRAF	Rn	Delayed branch, Rn + PC + 4 → PC	0000nnnn00100011	—	—
BSR	label	Delayed branch, PC + 4 → PR, disp × 2 + PC + 4 → PC	1011dddddddddddd	—	—
BSRF	Rn	Delayed branch, PC + 4 → PR, Rn + PC + 4 → PC	0000nnnn00000011	—	—
JMP	@Rn	Delayed branch, Rn → PC	0100nnnn00101011	—	—
JSR	@Rn	Delayed branch, PC + 4 → PR, Rn → PC	0100nnnn00001011	—	—
RTS		Delayed branch, PR → PC	0000000000001011	—	—

Table 3.9 System Control Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit
CLRMAC		0 → MACH, MACL	0000000000101000	—	—
CLRS		0 → S	0000000001001000	—	—
CLRT		0 → T	0000000000001000	—	0
ICBI	@Rn	Invalidates instruction cache block	0000nnnn11100011	—	—
LDC	Rm,SR	Rm → SR	0100mmmm00001110	Privileged	LSB
LDC	Rm,GBR	Rm → GBR	0100mmmm00011110	—	—
LDC	Rm,VBR	Rm → VBR	0100mmmm00101110	Privileged	—
LDC	Rm,SGR	Rm → SGR	0100mmmm00111010	Privileged	—
LDC	Rm,SSR	Rm → SSR	0100mmmm00111110	Privileged	—
LDC	Rm,SPC	Rm → SPC	0100mmmm01001110	Privileged	—

Instruction		Operation	Instruction Code	Privileged	T Bit
LDC	Rm,DBR	Rm → DBR	0100mmmm11111010	Privileged	—
LDC	Rm,Rn_BANK	Rm → Rn_BANK (n = 0 to 7)	0100mmmm1nnn1110	Privileged	—
LDC.L	@Rm+,SR	(Rm) → SR, Rm + 4 → Rm	0100mmmm00000111	Privileged	LSB
LDC.L	@Rm+,GBR	(Rm) → GBR, Rm + 4 → Rm	0100mmmm00010111	—	—
LDC.L	@Rm+,VBR	(Rm) → VBR, Rm + 4 → Rm	0100mmmm00100111	Privileged	—
LDC.L	@Rm+,SGR	(Rm) → SGR, Rm + 4 → Rm	0100mmmm00110110	Privileged	—
LDC.L	@Rm+,SSR	(Rm) → SSR, Rm + 4 → Rm	0100mmmm00110111	Privileged	—
LDC.L	@Rm+,SPC	(Rm) → SPC, Rm + 4 → Rm	0100mmmm01000111	Privileged	—
LDC.L	@Rm+,DBR	(Rm) → DBR, Rm + 4 → Rm	0100mmmm11110110	Privileged	—
LDC.L	@Rm+,Rn_BANK	(Rm) → Rn_BANK, Rm + 4 → Rm	0100mmmm1nnn0111	Privileged	—
LDS	Rm,MACH	Rm → MACH	0100mmmm00001010	—	—
LDS	Rm,MACL	Rm → MACL	0100mmmm00011010	—	—
LDS	Rm,PR	Rm → PR	0100mmmm00101010	—	—
LDS.L	@Rm+,MACH	(Rm) → MACH, Rm + 4 → Rm	0100mmmm00000110	—	—
LDS.L	@Rm+,MACL	(Rm) → MACL, Rm + 4 → Rm	0100mmmm00010110	—	—
LDS.L	@Rm+,PR	(Rm) → PR, Rm + 4 → Rm	0100mmmm00100110	—	—
LDTLB		PTEH/PTEL (/PTEA) → TLB	0000000000111000	Privileged	—
MOVCA.L	R0,@Rn	R0 → (Rn) (without fetching cache block)	0000nnnn11000011	—	—
NOP		No operation	0000000000001001	—	—
OCBI	@Rn	Invalidates operand cache block	0000nnnn10010011	—	—
OCBP	@Rn	Writes back and invalidates operand cache block	0000nnnn10100011	—	—
OCBWB	@Rn	Writes back operand cache block	0000nnnn10110011	—	—
PREF	@Rn	(Rn) → operand cache	0000nnnn10000011	—	—
PREFI	@Rn	Reads 32-byte instruction block into instruction cache	0000nnnn11010011	—	—
RTE		Delayed branch, SSR/SPC → SR/PC	0000000000101011	Privileged	—
SETS		1 → S	0000000001011000	—	—
SETT		1 → T	0000000000011000	—	1
SLEEP		Sleep or standby	0000000000011011	Privileged	—
STC	SR,Rn	SR → Rn	0000nnnn00000010	Privileged	—
STC	GBR,Rn	GBR → Rn	0000nnnn00010010	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit
STC	VBR,Rn	VBR → Rn	0000nnnn00100010	Privileged	—
STC	SSR,Rn	SSR → Rn	0000nnnn00110010	Privileged	—
STC	SPC,Rn	SPC → Rn	0000nnnn01000010	Privileged	—
STC	SGR,Rn	SGR → Rn	0000nnnn00111010	Privileged	—
STC	DBR,Rn	DBR → Rn	0000nnnn11111010	Privileged	—
STC	Rm_BANK,Rn	Rm_BANK → Rn (m = 0 to 7)	0000nnnn1mmm0010	Privileged	—
STC.L	SR,@-Rn	Rn – 4 → Rn, SR → (Rn)	0100nnnn00000011	Privileged	—
STC.L	GBR,@-Rn	Rn – 4 → Rn, GBR → (Rn)	0100nnnn00010011	—	—
STC.L	VBR,@-Rn	Rn – 4 → Rn, VBR → (Rn)	0100nnnn00100011	Privileged	—
STC.L	SSR,@-Rn	Rn – 4 → Rn, SSR → (Rn)	0100nnnn00110011	Privileged	—
STC.L	SPC,@-Rn	Rn – 4 → Rn, SPC → (Rn)	0100nnnn01000011	Privileged	—
STC.L	SGR,@-Rn	Rn – 4 → Rn, SGR → (Rn)	0100nnnn00110010	Privileged	—
STC.L	DBR,@-Rn	Rn – 4 → Rn, DBR → (Rn)	0100nnnn11110010	Privileged	—
STC.L	Rm_BANK,@-Rn	Rn – 4 → Rn, Rm_BANK → (Rn) (m = 0 to 7)	0100nnnn1mmm0011	Privileged	—
STS	MACH,Rn	MACH → Rn	0000nnnn00001010	—	—
STS	MACL,Rn	MACL → Rn	0000nnnn00011010	—	—
STS	PR,Rn	PR → Rn	0000nnnn00101010	—	—
STS.L	MACH,@-Rn	Rn – 4 → Rn, MACH → (Rn)	0100nnnn00000010	—	—
STS.L	MACL,@-Rn	Rn – 4 → Rn, MACL → (Rn)	0100nnnn00010010	—	—
STS.L	PR,@-Rn	Rn – 4 → Rn, PR → (Rn)	0100nnnn00100010	—	—
SYNCO		Data accesses invoked by the following instructions are not executed until execution of data accesses which precede this instruction has been completed.	0000000010101011	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit
TRAPA	#imm	PC + 2 → SPC, SR → SSR, R15 → SGR, 1 → SR.MD/BL/RB, #imm << 2 → TRA, H'160 → EXPEVT, VBR + H'0100 → PC	11000011iiiiiiii	—	—

Table 3.10 Floating-Point Single-Precision Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit
FLDI0	FRn	H'0000 0000 → FRn	1111nnnn10001101	—	—
FLDI1	FRn	H'3F80 0000 → FRn	1111nnnn10011101	—	—
FMOV	FRm,FRn	FRm → FRn	1111nnnnmmmm1100	—	—
FMOV.S	@Rm,FRn	(Rm) → FRn	1111nnnnmmmm1000	—	—
FMOV.S	@(R0,Rm),FRn	(R0 + Rm) → FRn	1111nnnnmmmm0110	—	—
FMOV.S	@Rm+,FRn	(Rm) → FRn, Rm + 4 → Rm	1111nnnnmmmm1001	—	—
FMOV.S	FRm,@Rn	FRm → (Rn)	1111nnnnmmmm1010	—	—
FMOV.S	FRm,@-Rn	Rn-4 → Rn, FRm → (Rn)	1111nnnnmmmm1011	—	—
FMOV.S	FRm,@(R0,Rn)	FRm → (R0 + Rn)	1111nnnnmmmm0111	—	—
FMOV	DRm,DRn	DRm → DRn	1111nnn0mmm01100	—	—
FMOV	@Rm,DRn	(Rm) → DRn	1111nnn0mmmm1000	—	—
FMOV	@(R0,Rm),DRn	(R0 + Rm) → DRn	1111nnn0mmmm0110	—	—
FMOV	@Rm+,DRn	(Rm) → DRn, Rm + 8 → Rm	1111nnn0mmmm1001	—	—
FMOV	DRm,@Rn	DRm → (Rn)	1111nnnnmmm01010	—	—
FMOV	DRm,@-Rn	Rn-8 → Rn, DRm → (Rn)	1111nnnnmmm01011	—	—
FMOV	DRm,@(R0,Rn)	DRm → (R0 + Rn)	1111nnnnmmm00111	—	—
FLDS	FRm,FPUL	FRm → FPUL	1111mmmm00011101	—	—
FSTS	FPUL,FRn	FPUL → FRn	1111nnnn00001101	—	—
FABS	FRn	FRn & H'7FFF FFFF → FRn	1111nnnn01011101	—	—
FADD	FRm,FRn	FRn + FRm → FRn	1111nnnnmmmm0000	—	—
FCMP/EQ	FRm,FRn	When FRn = FRm, 1 → T Otherwise, 0 → T	1111nnnnmmmm0100	—	Comparison result
FCMP/GT	FRm,FRn	When FRn > FRm, 1 → T Otherwise, 0 → T	1111nnnnmmmm0101	—	Comparison result
FDIV	FRm,FRn	FRn/FRm → FRn	1111nnnnmmmm0011	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit
FLOAT	FPUL,FRn	(float) FPUL → FRn	1111nnnn00101101	—	—
FMAC	FR0,FRm,FRn	FR0*FRm + FRn → FRn	1111nnnnmmmm1110	—	—
FMUL	FRm,FRn	FRn*FRm → FRn	1111nnnnmmmm0010	—	—
FNEG	FRn	FRn ∧ H'8000 0000 → FRn	1111nnnn01001101	—	—
FSQRT	FRn	√FRn → FRn	1111nnnn01101101	—	—
FSUB	FRm,FRn	FRn – FRm → FRn	1111nnnnmmmm0001	—	—
FTRC	FRm,FPUL	(long) FRm → FPUL	1111mmmm00111101	—	—

Table 3.11 Floating-Point Double-Precision Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit
FABS	DRn	DRn & H'7FFF FFFF FFFF FFFF → DRn	1111nnn001011101	—	—
FADD	DRm,DRn	DRn + DRm → DRn	1111nnn0mmm00000	—	—
FCMP/EQ	DRm,DRn	When DRn = DRm, 1 → T Otherwise, 0 → T	1111nnn0mmm00100	—	Comparison result
FCMP/GT	DRm,DRn	When DRn > DRm, 1 → T Otherwise, 0 → T	1111nnn0mmm00101	—	Comparison result
FDIV	DRm,DRn	DRn /DRm → DRn	1111nnn0mmm00011	—	—
FCNVDS	DRm,FPUL	double_to_ float(DRm) → FPUL	1111mmm010111101	—	—
FCNVSD	FPUL,DRn	float_to_ double (FPUL) → DRn	1111nnn010101101	—	—
FLOAT	FPUL,DRn	(float)FPUL → DRn	1111nnn000101101	—	—
FMUL	DRm,DRn	DRn *DRm → DRn	1111nnn0mmm00010	—	—
FNEG	DRn	DRn ^ H'8000 0000 0000 0000 → DRn	1111nnn001001101	—	—
FSQRT	DRn	√DRn → DRn	1111nnn001101101	—	—
FSUB	DRm,DRn	DRn – DRm → DRn	1111nnn0mmm00001	—	—
FTRC	DRm,FPUL	(long) DRm → FPUL	1111mmm000111101	—	—

Table 3.12 Floating-Point Control Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit
LDS	Rm,FPSCR	Rm → FPSCR	0100mmmm01101010	—	—
LDS	Rm,FPUL	Rm → FPUL	0100mmmm01011010	—	—
LDS.L	@Rm+,FPSCR	(Rm) → FPSCR, Rm+4 → Rm	0100mmmm01100110	—	—
LDS.L	@Rm+,FPUL	(Rm) → FPUL, Rm+4 → Rm	0100mmmm01010110	—	—
STS	FPSCR,Rn	FPSCR → Rn	0000nnnn01101010	—	—
STS	FPUL,Rn	FPUL → Rn	0000nnnn01011010	—	—
STS.L	FPSCR,@-Rn	Rn – 4 → Rn, FPSCR → (Rn)	0100nnnn01100010	—	—
STS.L	FPUL,@-Rn	Rn – 4 → Rn, FPUL → (Rn)	0100nnnn01010010	—	—

Table 3.13 Floating-Point Graphics Acceleration Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit
FMOV	DRm,XDn	DRm → XDn	1111nnn1mmm01100	—	—
FMOV	XDm,DRn	XDm → DRn	1111nnn0mmm11100	—	—
FMOV	XDm,XDn	XDm → XDn	1111nnn1mmm11100	—	—
FMOV	@Rm,XDn	(Rm) → XDn	1111nnn1mmmm1000	—	—
FMOV	@Rm+,XDn	(Rm) → XDn, Rm + 8 → Rm	1111nnn1mmmm1001	—	—
FMOV	@(R0,Rm),XDn	(R0 + Rm) → XDn	1111nnn1mmmm0110	—	—
FMOV	XDm,@Rn	XDm → (Rn)	1111nnnnmmm11010	—	—
FMOV	XDm,@-Rn	Rn – 8 → Rn, XDm → (Rn)	1111nnnnmmm11011	—	—
FMOV	XDm,@(R0,Rn)	XDm → (R0 + Rn)	1111nnnnmmm10111	—	—
FIPR	FVm,FVn	inner_product (FVm, FVn) → FR[n+3]	1111nnmm11101101	—	—
FTRV	XMTRX,FVn	transform_vector (XMTRX, FVn) → FVn	1111nn0111111101	—	—
FRCHG		~FPSCR.FR → FPSCR.FR	1111101111111101	—	—
FSCHG		~FPSCR.SZ → FPSCR.SZ	1111001111111101	—	—
FPCHG		~FPSCR.PR → FPSCR.PR	1111011111111101	—	—
FSRRA	FRn	1/sqrt(FRn) → FRn	1111nnnn01111101	—	—
FSCA	FPUL,DRn	sin(FPUL) → FRn cos(FPUL) → FR[n + 1]	1111nnn011111101	—	—

Note: * sqrt(FRn) is the square root of FRn.

Section 4 Pipelining

This LSI is a 2-ILP (instruction-level-parallelism) superscalar pipelining microprocessor. Instruction execution is pipelined, and two instructions can be executed in parallel.

4.1 Pipelines

Figure 4.1 shows the basic pipelines. Normally, a pipeline consists of eight stages: instruction fetch (I1/I2/I3), decode and register read (ID), execution (E1/E2/E3), and write-back (WB). An instruction is executed as a combination of basic pipelines.

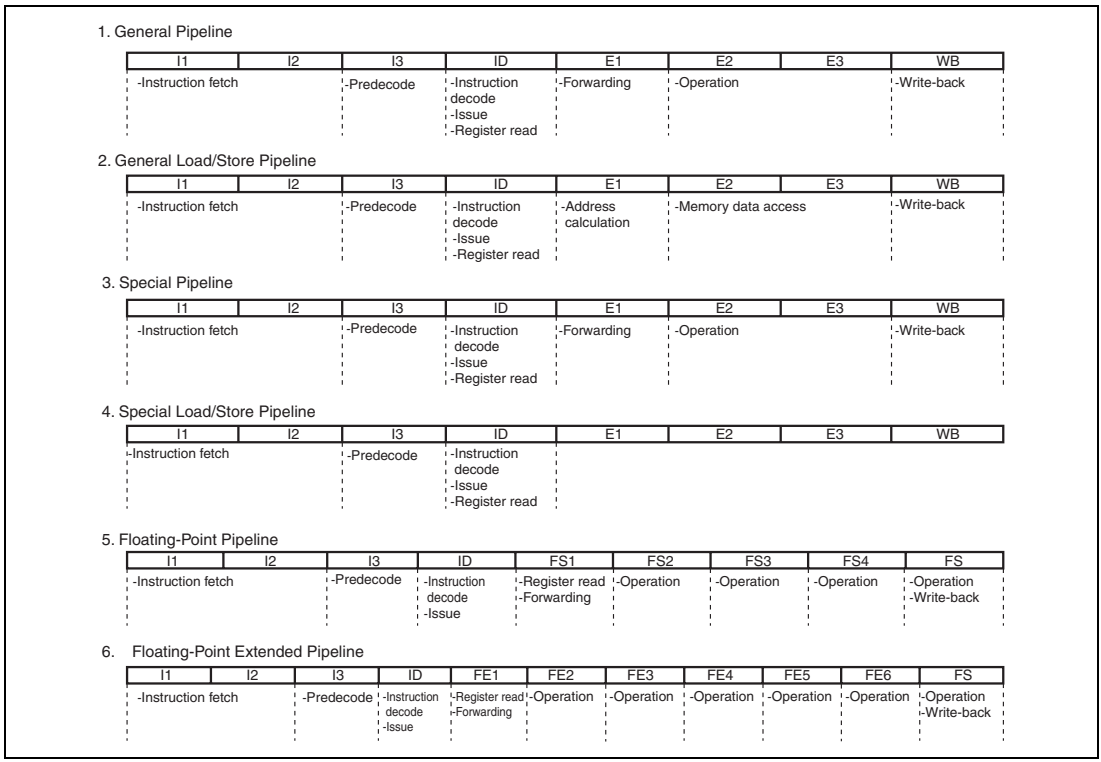


Figure 4.1 Basic Pipelines

Figure 4.2 shows the instruction execution patterns. Representations in figure 4.2 and their descriptions are listed in table 4.1.

Table 4.1 Representations of Instruction Execution Patterns

Representation	Description
E1 E2 E3 WB	CPU EX pipe is occupied
S1 S2 S3 WB	CPU LS pipe is occupied (with memory access)
s1 s2 s3 WB	CPU LS pipe is occupied (without memory access)
E1/S1	Either CPU EX pipe or CPU LS pipe is occupied
E1S1 , E1s1	Both CPU EX pipe and CPU LS pipe are occupied
M2 M3 MS	CPU MULT operation unit is occupied
FE1 FE2 FE3 FE4 FE5 FE6 FS	FPU-EX pipe is occupied
FS1 FS2 FS3 FS4 FS	FPU-LS pipe is occupied
ID	ID stage is locked
└──	Both CPU and FPU pipes are occupied

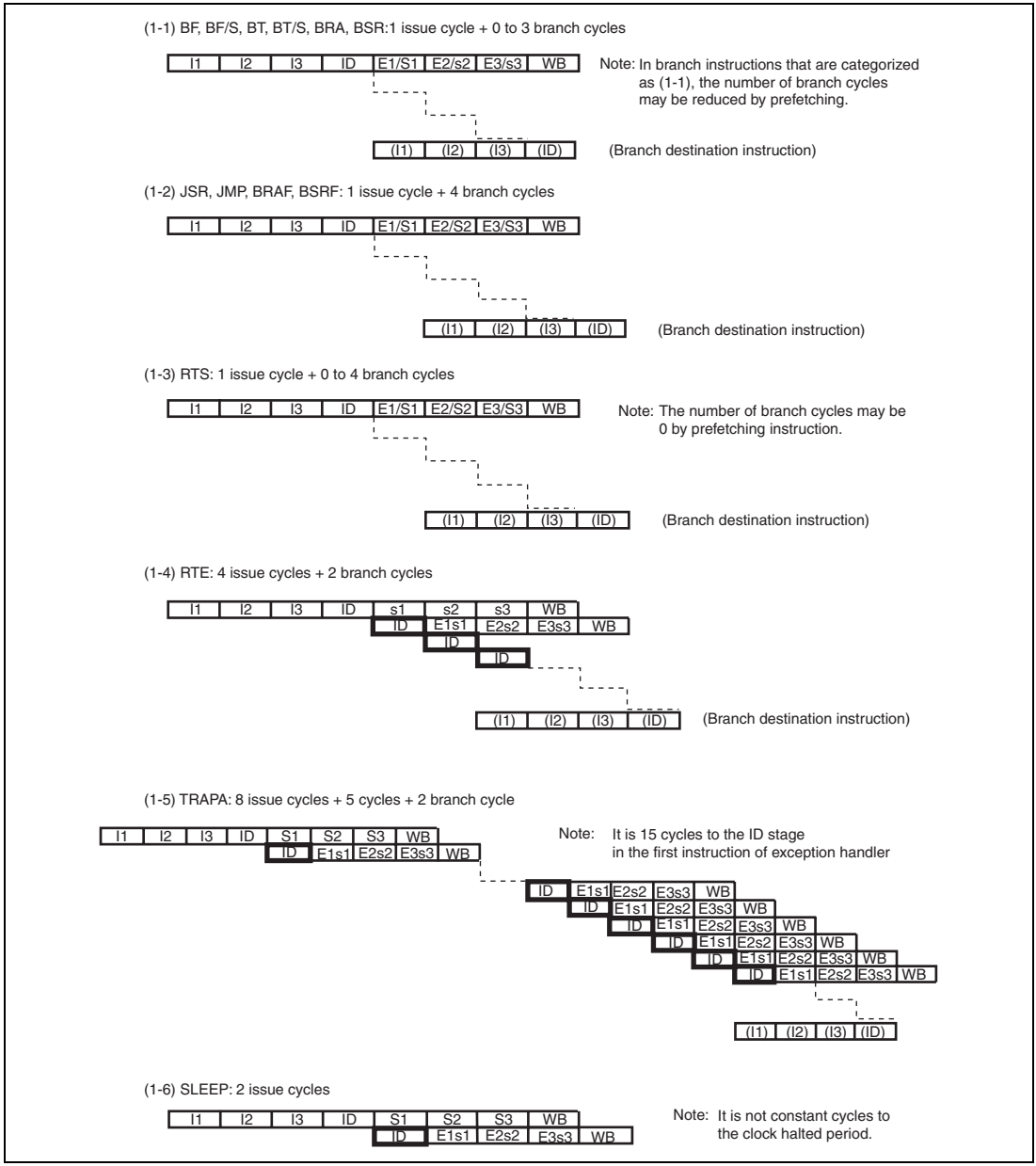


Figure 4.2 Instruction Execution Patterns (1)

(2-1) 1-step operation (EX type): 1 issue cycle

EXT[SU].[BW], MOVT, SWAP, XTRCT, ADD*, CMP*, DIV*, DT, NEG*, SUB*, AND, AND#, NOT, OR, OR#, TST, TST#, XOR, XOR#, ROT*, SHA*, SHL*, CLRS, CLRT, SETS, SETT

Note: Except for AND#, OR#, TST#, and XOR# instructions using GBR relative addressing mode

I1	I2	I3	ID	E1	E2	E3	WB

(2-2) 1-step operation (LS type): 1 issue cycle

MOVA

I1	I2	I3	ID	s1	s2	s3	WB

(2-3) 1-step operation (MT type): 1 issue cycle

MOV#, NOP

I1	I2	I3	ID	E1/S1	E2/s2	E3/s3	WB

(2-4) MOV (MT type): 1 issue cycle

MOV

I1	I2	I3	ID	E1/s1	E2/s2	E3/S3	WB

Figure 4.2 Instruction Execution Patterns (2)

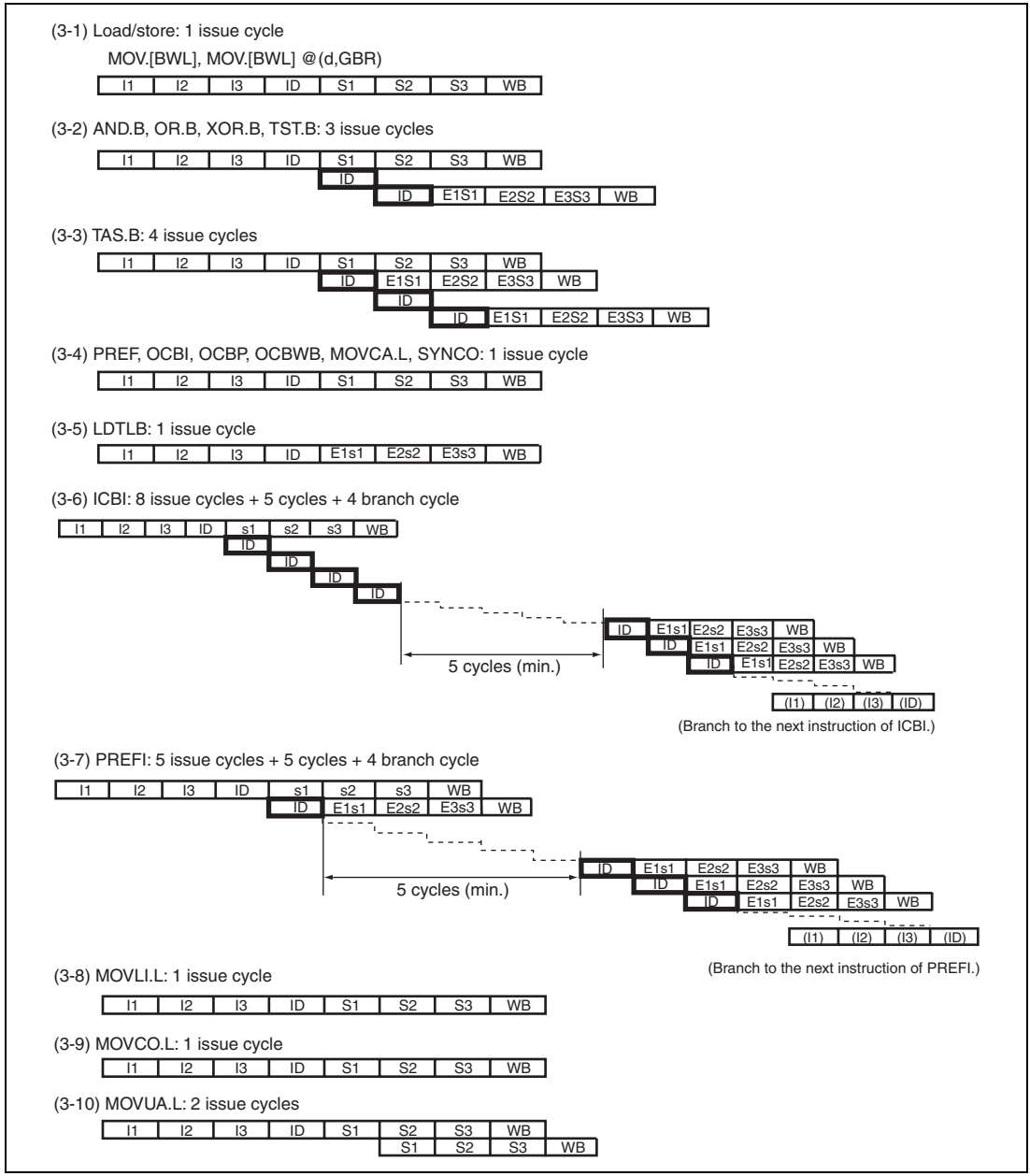


Figure 4.2 Instruction Execution Patterns (3)

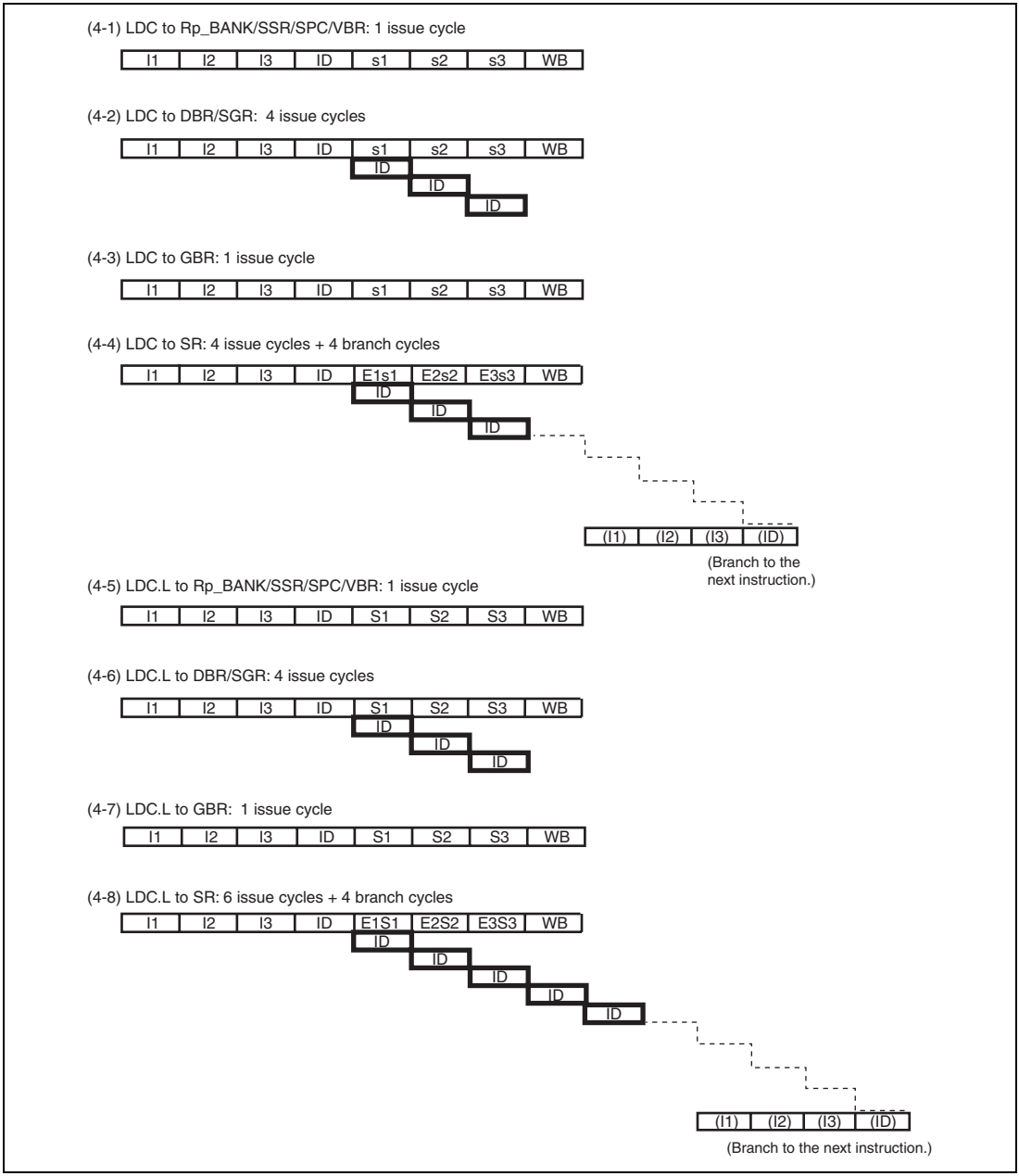


Figure 4.2 Instruction Execution Patterns (4)

(4-9) STC from DBR/GBR/Rp_BANK/SSR/SPC/VBR/SGR: 1 issue cycle

I1	I2	I3	ID	s1	s2	s3	WB

(4-10) STC from SR: 1 issue cycle

I1	I2	I3	ID	E1s1	E2s2	E3s3	WB

(4-11) STC.L from DBR/GBR/Rp_BANK/SSR/SPC/VBR/SGR: 1 issue cycle

I1	I2	I3	ID	S1	S2	S3	WB

(4-12) STC.L from SR: 1 issue cycle

I1	I2	I3	ID	E1S1	E2S2	E3S3	WB

(4-13) LDS to PR: 1 issue cycle

I1	I2	I3	ID	s1	s2	s3	WB

(4-14) LDS.L to PR: 1 issue cycle

I1	I2	I3	ID	S1	S2	S3	WB

(4-15) STS from PR: 1 issue cycle

I1	I2	I3	ID	s1	s2	s3	WB

(4-16) STS.L from PR: 1 issue cycle

I1	I2	I3	ID	S1	S2	S3	WB

(4-17) BSRF, BSR, JSR delay slot instructions (PR set): 0 issue cycle

(I1)	(I2)	(I3)	(ID)	(??1)	(??2)	(??3)	(WB)

Notes: The value of PR is changed in the E3 stage of delay slot instruction.
When the STS and STS.L instructions from PR are used as delay slot instructions, changed PR value is used.

Figure 4.2 Instruction Execution Patterns (5)

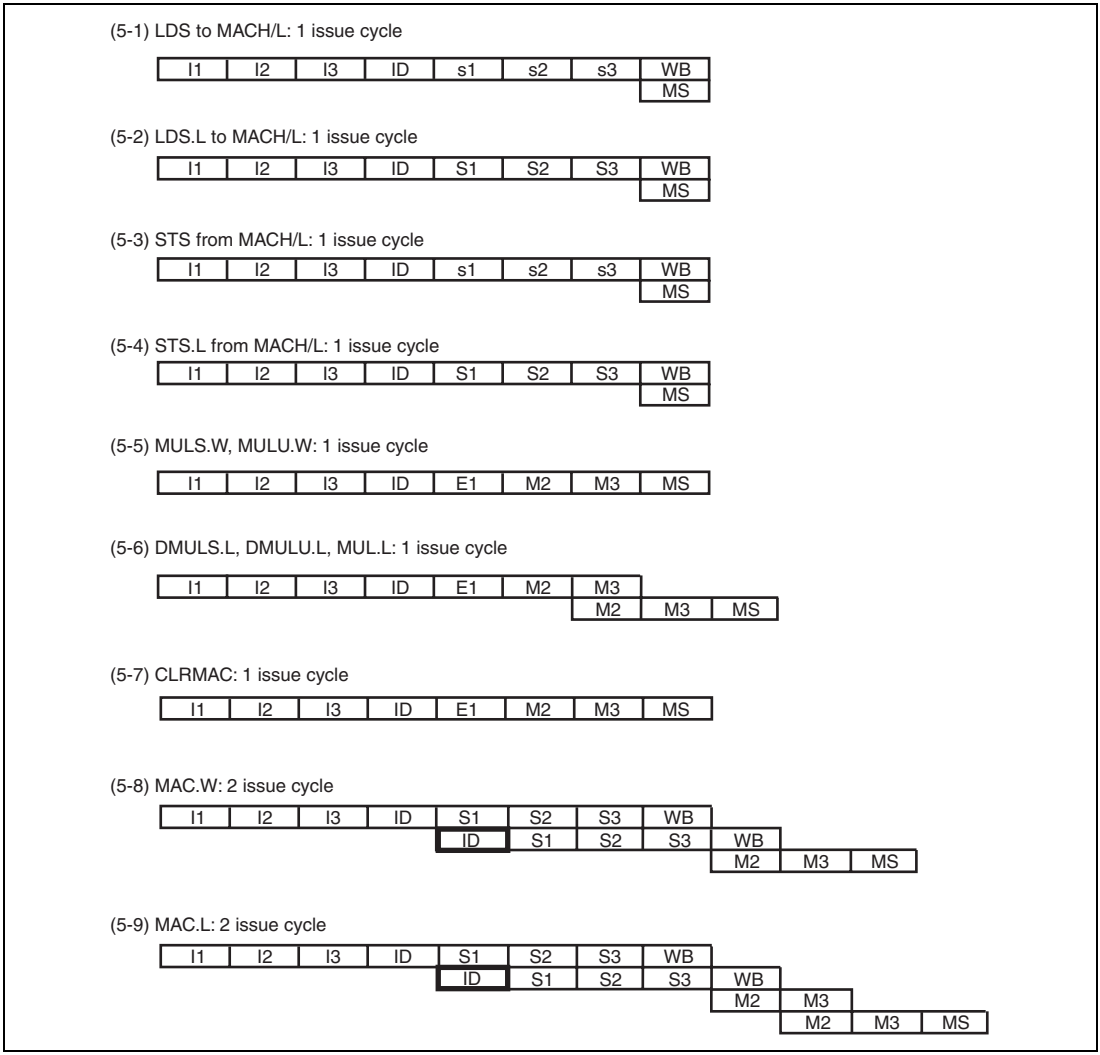


Figure 4.2 Instruction Execution Patterns (6)

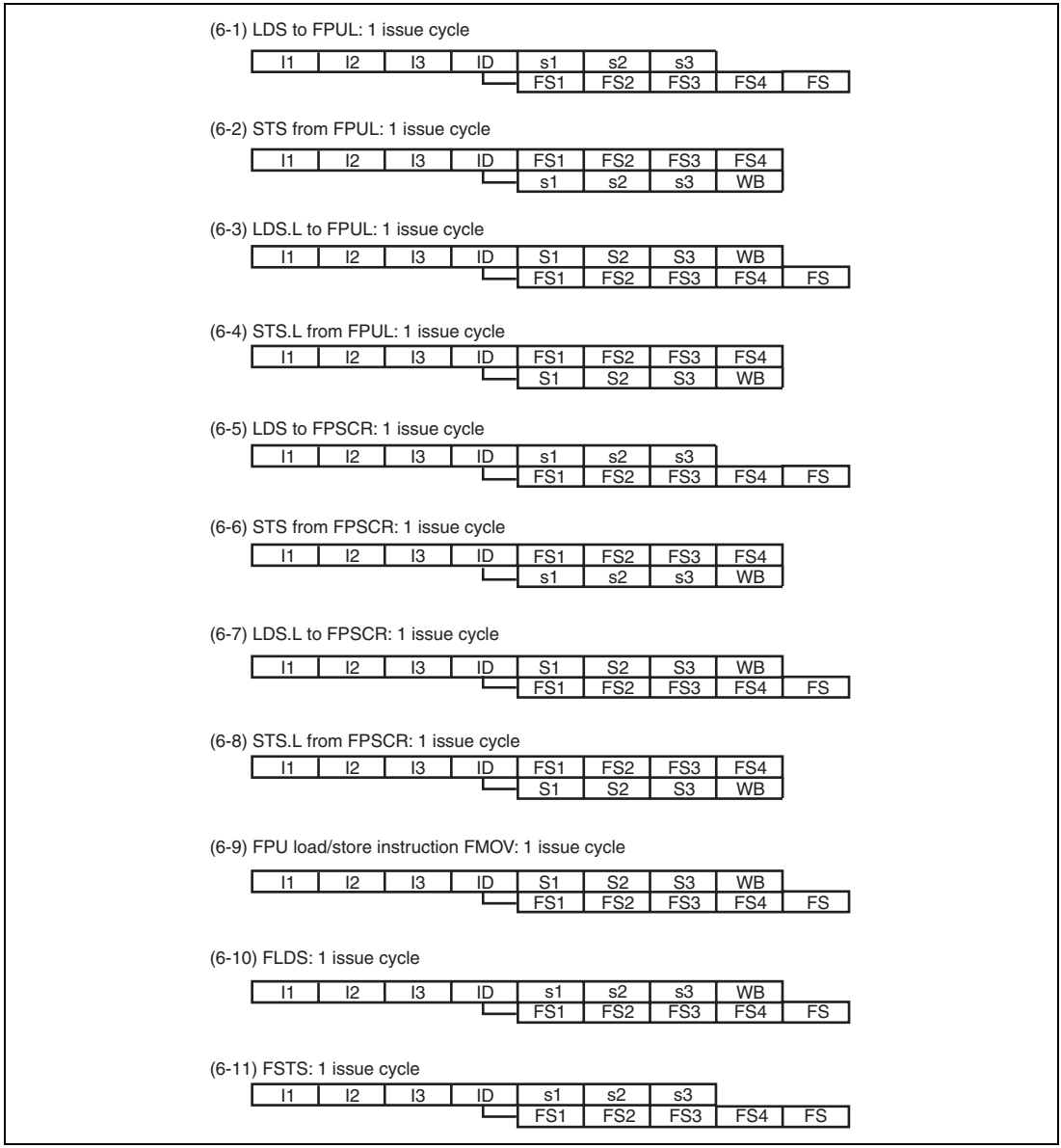


Figure 4.2 Instruction Execution Patterns (7)

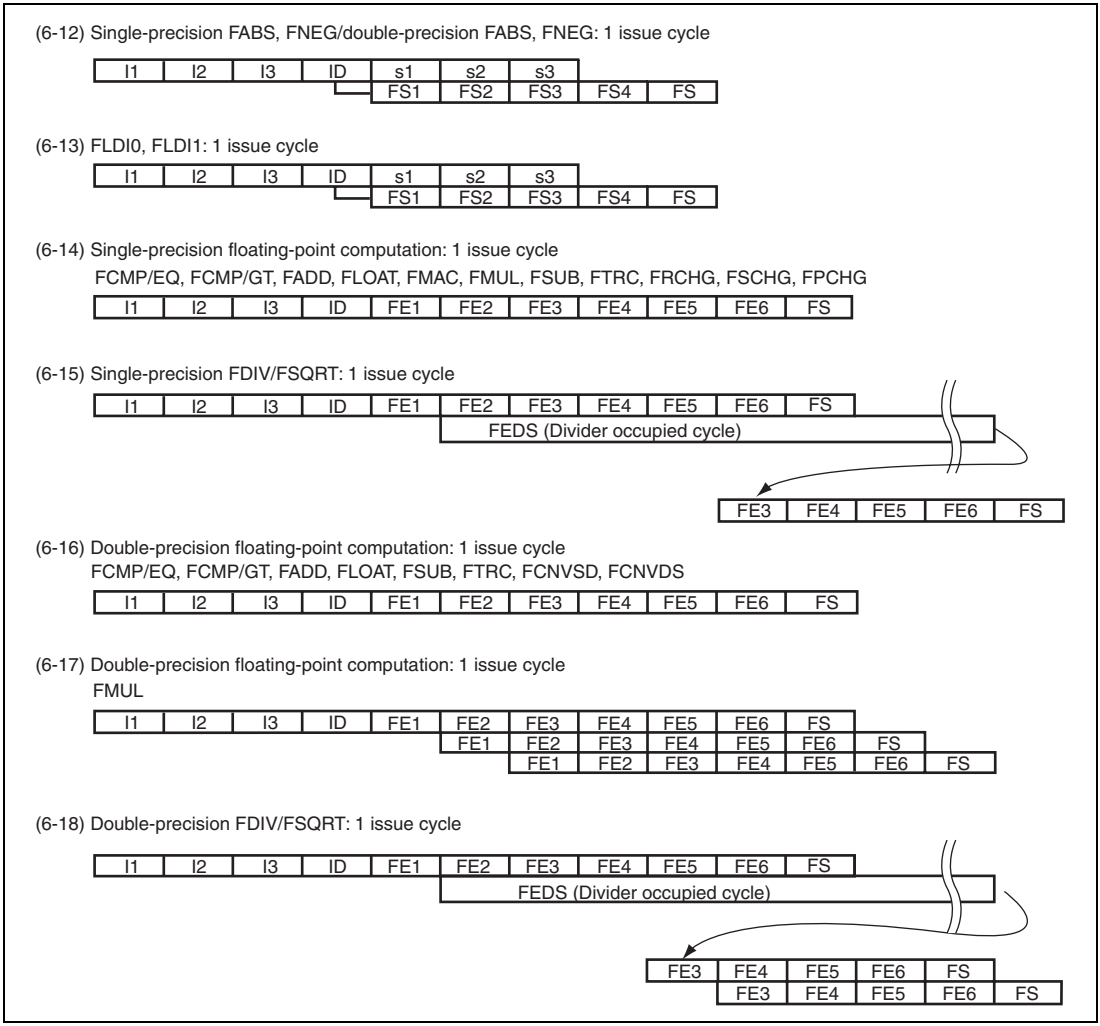


Figure 4.2 Instruction Execution Patterns (8)

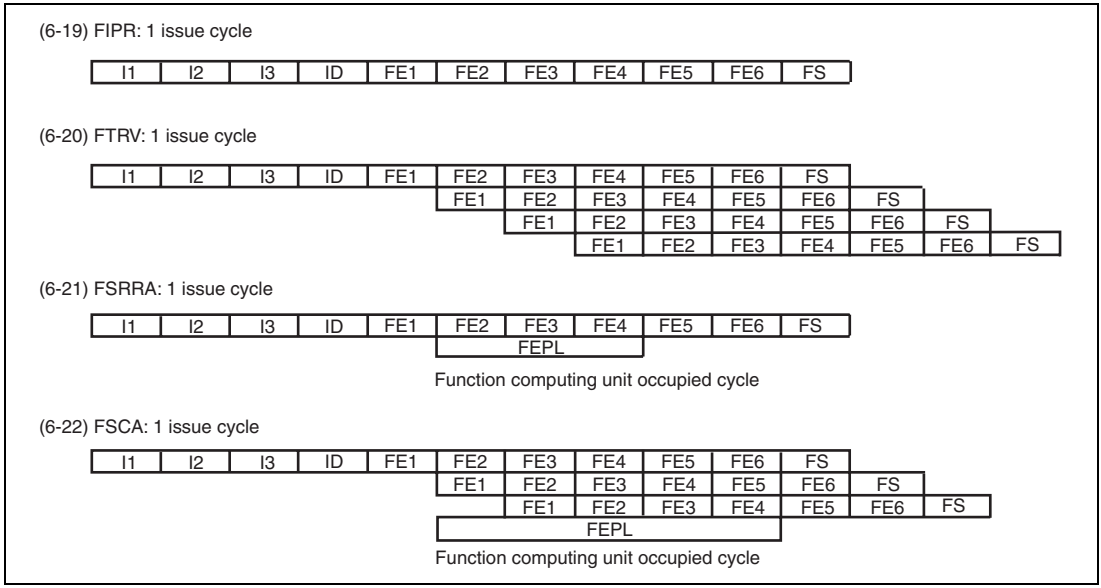


Figure 4.2 Instruction Execution Patterns (9)

4.2 Parallel-Executability

Instructions are categorized into six groups according to the internal function blocks used, as shown in table 4.2. Table 4.3 shows the parallel-executability of pairs of instructions in terms of groups. For example, ADD in the EX group and BRA in the BR group can be executed in parallel.

Table 4.2 Instruction Groups

Instruction Group	Instruction			
EX	ADD	DT	ROTL	SHLR8
	ADDC	EXTS	ROTR	SHLR16
	ADDV	EXTU	SETS	SUB
	AND #imm,R0	MOVT	SETT	SUBC
	AND Rm,Rn	MUL.L	SHAD	SUBV
	CLRMAC	MULS.W	SHAL	SWAP
	CLRS	MULU.W	SHAR	TST #imm,R0
	CLRT	NEG	SHLD	TST Rm,Rn
	CMP	NEGC	SHLL	XOR #imm,R0
	DIV0S	NOT	SHLL2	XOR Rm,Rn
	DIV0U	OR #imm,R0	SHLL8	XTRCT
	DIV1	OR Rm,Rn	SHLL16	
	DMUS.L	ROTCL	SHLR	
	DMULU.L	ROTCR	SHLR2	
MT	MOV #imm,Rn	MOV Rm,Rn	NOP	
BR	BF	BRAF	BT	JSR
	BF/S	BSR	BT/S	RTS
	BRA	BSRF	JMP	
LS	FABS	FMOV.S FR,@adr	MOV.[BWL] @adr,R	STC CR2,Rn
	FNEG	FSTS	MOV.[BWL] R,@adr	STC.L CR2,@-Rn
	FLDI0	LDC Rm,CR1	MOVA	STS SR2,Rn
	FLDI1	LDC.L @Rm+,CR1	MOVCA.L	STS.L SR2,@-Rn
	FLDS	LDS Rm,SR1	MOVUA	STS SR1,Rn
	FMOV @adr,FR	LDS Rm,SR2	OCBI	STS.L SR1,@-Rn
	FMOV FR,@adr	LDS.L @adr,SR2	OCBP	
	FMOV FR,FR	LDS.L @Rm+,SR1	OCBWB	
	FMOV.S @adr,FR	LDS.L @Rm+,SR2	PREF	

Instruction Group	Instruction			
FE	FADD	FDIV	FRCHG	FSCA
	FSUB	FIPR	FSCHG	FSRRA
	FCMP (S/D)	FLOAT	FSQRT	FPCHG
	FCNVDS	FMAC	FTRC	
	FCNVSD	FMUL	FTRV	
CO	AND.B #imm,@(R0,GBR)	LDC.L @Rm+,SR	PREFI	TRAPA
	ICBI	LDTLB	RTE	TST.B #imm,@(R0,GBR)
	LDC Rm,DBR	MAC.L	SLEEP	XOR.B #imm,@(R0,GBR)
	LDC Rm, SGR	MAC.W	STC SR,Rn	
	LDC Rm,SR	MOVCO	STC.L SR,@-Rn	
	LDC.L @Rm+,DBR	MOVLI	SYNCO	
	LDC.L @Rm+,SGR	OR.B #imm,@(R0,GBR)	TAS.B	

[Legend]
R: Rm/Rn
@adr: Address
SR1: MACH/MACL/PR
SR2: FPUL/FPSCR
CR1: GBR/Rp_BANK/SPC/SSR/VBR
CR2: CR1/DBR/SGR
FR: FRm/FRn/DRm/DRn/XDm/XDn

The parallel execution of two instructions can be carried out under following conditions.

1. Both addr (preceding instruction) and addr+2 (following instruction) are specified within the minimum page size (1 Kbyte).
2. The execution of these two instructions is supported in table 4.3, Combination of Preceding and Following Instructions.
3. Data used by an instruction of addr does not conflict with data used by a previous instruction
4. Data used by an instruction of addr+2 does not conflict with data used by a previous instruction
5. Both instructions are valid

Table 4.3 Combination of Preceding and Following Instructions

		Preceding Instruction (addr)					
		EX	**MT**	**BR**	**LS**	**FE**	**CO**
Following Instruction (addr+2)	EX	No	Yes	Yes	Yes	Yes	
	MT	Yes	Yes	Yes	Yes	Yes	
	BR	Yes	Yes	No	Yes	Yes	
	LS	Yes	Yes	Yes	No	Yes	
	FE	Yes	Yes	Yes	Yes	No	
	CO						No

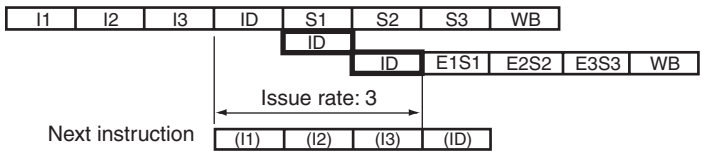
4.3 Issue Rates and Execution Cycles

Instruction execution cycles are summarized in table 4.4. Instruction Group in the table 4.4 corresponds to the category in the table 4.2. Penalty cycles due to a pipeline stall are not considered in the issue rates and execution cycles in this section.

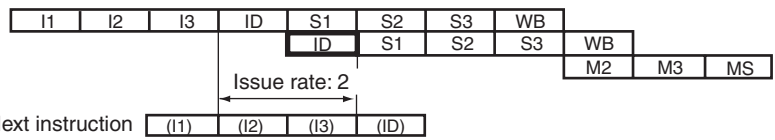
1. Issue Rate

Issue rates indicates the issue period between one instruction and next instruction.

E.g. AND.B instruction



E.g. MAC.W instruction

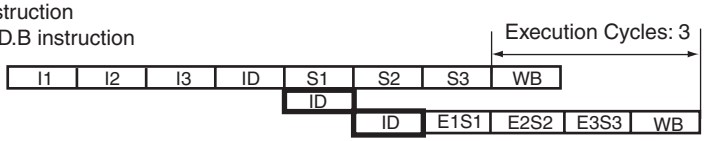


2. Execution Cycles

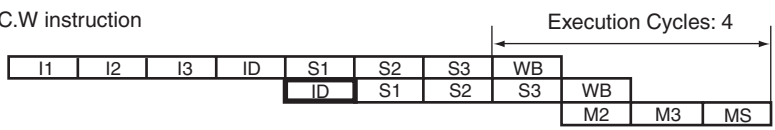
Execution cycles indicates the cycle counts an instruction occupied the pipeline based on the next rules.

CPU instruction

E.g. AND.B instruction

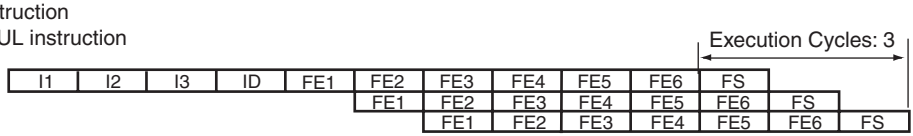


E.g. MAC.W instruction



FPU instruction

E.g. FMUL instruction



E.g. FDIV instruction

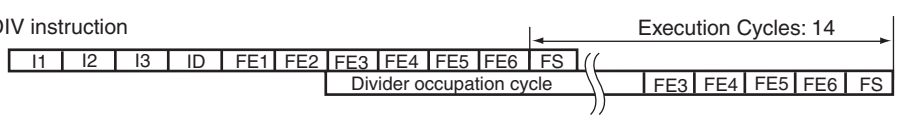


Table 4.4 Issue Rates and Execution Cycles

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Data transfer instructions	1	EXTS.B	Rm,Rn	EX	1	1	2-1
	2	EXTS.W	Rm,Rn	EX	1	1	2-1
	3	EXTU.B	Rm,Rn	EX	1	1	2-1
	4	EXTU.W	Rm,Rn	EX	1	1	2-1
	5	MOV	Rm,Rn	MT	1	1	2-4
	6	MOV	#imm,Rn	MT	1	1	2-3
	7	MOVA	@(disp,PC),R0	LS	1	1	2-2
	8	MOV.W	@(disp,PC),Rn	LS	1	1	3-1
	9	MOV.L	@(disp,PC),Rn	LS	1	1	3-1
	10	MOV.B	@Rm,Rn	LS	1	1	3-1
	11	MOV.W	@Rm,Rn	LS	1	1	3-1
	12	MOV.L	@Rm,Rn	LS	1	1	3-1
	13	MOV.B	@Rm+,Rn	LS	1	1	3-1
	14	MOV.W	@Rm+,Rn	LS	1	1	3-1
	15	MOV.L	@Rm+,Rn	LS	1	1	3-1
	16	MOV.B	@(disp,Rm),R0	LS	1	1	3-1
	17	MOV.W	@(disp,Rm),R0	LS	1	1	3-1
	18	MOV.L	@(disp,Rm),Rn	LS	1	1	3-1
	19	MOV.B	@(R0,Rm),Rn	LS	1	1	3-1
	20	MOV.W	@(R0,Rm),Rn	LS	1	1	3-1
	21	MOV.L	@(R0,Rm),Rn	LS	1	1	3-1
	22	MOV.B	@(disp,GBR),R0	LS	1	1	3-1
	23	MOV.W	@(disp, GBR),R0	LS	1	1	3-1
	24	MOV.L	@(disp, GBR),R0	LS	1	1	3-1
	25	MOV.B	Rm,@Rn	LS	1	1	3-1
	26	MOV.W	Rm,@Rn	LS	1	1	3-1
	27	MOV.L	Rm,@Rn	LS	1	1	3-1
	28	MOV.B	Rm,@-Rn	LS	1	1	3-1
	29	MOV.W	Rm,@-Rn	LS	1	1	3-1

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Data transfer instructions	30	MOV.L	Rm,@-Rn	LS	1	1	3-1
	31	MOV.B	R0,@(disp,Rn)	LS	1	1	3-1
	32	MOV.W	R0,@(disp,Rn)	LS	1	1	3-1
	33	MOV.L	Rm,@(disp,Rn)	LS	1	1	3-1
	34	MOV.B	Rm,@(R0,Rn)	LS	1	1	3-1
	35	MOV.W	Rm,@(R0,Rn)	LS	1	1	3-1
	36	MOV.L	Rm,@(R0,Rn)	LS	1	1	3-1
	37	MOV.B	R0,@(disp,GBR)	LS	1	1	3-1
	38	MOV.W	R0,@(disp,GBR)	LS	1	1	3-1
	39	MOV.L	R0,@(disp,GBR)	LS	1	1	3-1
	40	MOVCA.L	R0,@Rn	LS	1	1	3-4
	41	MOVCO.L	R0,@Rn	CO	1	1	3-9
	42	MOVLI.L	@Rm,R0	CO	1	1	3-8
	43	MOVUA.L	@Rm,R0	LS	2	2	3-10
	44	MOVUA.L	@Rm+,R0	LS	2	2	3-10
	45	MOVT	Rn	EX	1	1	2-1
	46	OCBI	@Rn	LS	1	1	3-4
	47	OCBP	@Rn	LS	1	1	3-4
	48	OCBWB	@Rn	LS	1	1	3-4
	49	PREF	@Rn	LS	1	1	3-4
	50	SWAP.B	Rm,Rn	EX	1	1	2-1
	51	SWAP.W	Rm,Rn	EX	1	1	2-1
	52	XTRCT	Rm,Rn	EX	1	1	2-1
Fixed-point arithmetic instructions	53	ADD	Rm,Rn	EX	1	1	2-1
	54	ADD	#imm,Rn	EX	1	1	2-1
	55	ADDC	Rm,Rn	EX	1	1	2-1
	56	ADDV	Rm,Rn	EX	1	1	2-1
	57	CMP/EQ	#imm,R0	EX	1	1	2-1
	58	CMP/EQ	Rm,Rn	EX	1	1	2-1
	59	CMP/GE	Rm,Rn	EX	1	1	2-1

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Fixed-point arithmetic instructions	60	CMP/GT	Rm,Rn	EX	1	1	2-1
	61	CMP/HI	Rm,Rn	EX	1	1	2-1
	62	CMP/HS	Rm,Rn	EX	1	1	2-1
	63	CMP/PL	Rn	EX	1	1	2-1
	64	CMP/PZ	Rn	EX	1	1	2-1
	65	CMP/STR	Rm,Rn	EX	1	1	2-1
	66	DIV0S	Rm,Rn	EX	1	1	2-1
	67	DIV0U		EX	1	1	2-1
	68	DIV1	Rm,Rn	EX	1	1	2-1
	69	DMULS.L	Rm,Rn	EX	1	2	5-6
	70	DMULU.L	Rm,Rn	EX	1	2	5-6
	71	DT	Rn	EX	1	1	2-1
	72	MAC.L	@Rm+,@Rn+	CO	2	5	5-9
	73	MAC.W	@Rm+,@Rn+	CO	2	4	5-8
	74	MUL.L	Rm,Rn	EX	1	2	5-6
	75	MULS.W	Rm,Rn	EX	1	1	5-5
	76	MULU.W	Rm,Rn	EX	1	1	5-5
	77	NEG	Rm,Rn	EX	1	1	2-1
	78	NEGC	Rm,Rn	EX	1	1	2-1
	79	SUB	Rm,Rn	EX	1	1	2-1
	80	SUBC	Rm,Rn	EX	1	1	2-1
	81	SUBV	Rm,Rn	EX	1	1	2-1
Logical instructions	82	AND	Rm,Rn	EX	1	1	2-1
	83	AND	#imm,R0	EX	1	1	2-1
	84	AND.B	#imm,@(R0,GBR)	CO	3	3	3-2
	85	NOT	Rm,Rn	EX	1	1	2-1
	86	OR	Rm,Rn	EX	1	1	2-1
	87	OR	#imm,R0	EX	1	1	2-1
	88	OR.B	#imm,@(R0,GBR)	CO	3	3	3-2
	89	TAS.B	@Rn	CO	4	4	3-3

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Logical instructions	90	TST	Rm,Rn	EX	1	1	2-1
	91	TST	#imm,R0	EX	1	1	2-1
	92	TST.B	#imm,@(R0,GBR)	CO	3	3	3-2
	93	XOR	Rm,Rn	EX	1	1	2-1
	94	XOR	#imm,R0	EX	1	1	2-1
	95	XOR.B	#imm,@(R0,GBR)	CO	3	3	3-2
Shift instructions	96	ROTL	Rn	EX	1	1	2-1
	97	ROTR	Rn	EX	1	1	2-1
	98	ROTCL	Rn	EX	1	1	2-1
	99	ROTCR	Rn	EX	1	1	2-1
	100	SHAD	Rm,Rn	EX	1	1	2-1
	101	SHAL	Rn	EX	1	1	2-1
	102	SHAR	Rn	EX	1	1	2-1
	103	SHLD	Rm,Rn	EX	1	1	2-1
	104	SHLL	Rn	EX	1	1	2-1
	105	SHLL2	Rn	EX	1	1	2-1
	106	SHLL8	Rn	EX	1	1	2-1
	107	SHLL16	Rn	EX	1	1	2-1
	108	SHLR	Rn	EX	1	1	2-1
	109	SHLR2	Rn	EX	1	1	2-1
	110	SHLR8	Rn	EX	1	1	2-1
	111	SHLR16	Rn	EX	1	1	2-1
Branch instructions	112	BF	disp	BR	1+0 to 2	1	1-1
	113	BF/S	disp	BR	1+0 to 2	1	1-1
	114	BT	disp	BR	1+0 to 2	1	1-1
	115	BT/S	disp	BR	1+0 to 2	1	1-1
	116	BRA	disp	BR	1+0 to 2	1	1-1
	117	BRAF	Rm	BR	1+3	1	1-2
	118	BSR	disp	BR	1+0 to 2	1	1-1
	119	BSRF	Rm	BR	1+3	1	1-2

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Branch instructions	120	JMP	@Rn	BR	1+3	1	1-2
	121	JSR	@Rn	BR	1+3	1	1-2
	122	RTS		BR	1+0 to 3	1	1-3
System control instruction	123	NOP		MT	1	1	2-3
	124	CLRMAC		EX	1	1	5-7
	125	CLRS		EX	1	1	2-1
	126	CLRT		EX	1	1	2-1
	127	ICBI	@Rn	CO	8+5+3	13	3-6
	128	SETS		EX	1	1	2-1
	129	SETT		EX	1	1	2-1
	130	PREFI	@Rn	CO	5+5+3	10	3-7
	131	SYNCO		CO	Undefined	Undefined	3-4
	132	TRAPA	#imm	CO	8+5+1	13	1-5
	133	RTE		CO	4+1	4	1-4
	134	SLEEP		CO	Undefined	Undefined	1-6
	135	LDTLB		CO	1	1	3-5
	136	LDC	Rm,DBR	CO	4	4	4-2
	137	LDC	Rm,SGR	CO	4	4	4-2
	138	LDC	Rm,GBR	LS	1	1	4-3
	139	LDC	Rm,Rp_BANK	LS	1	1	4-1
	140	LDC	Rm,SR	CO	4+3	4	4-4
	141	LDC	Rm,SSR	LS	1	1	4-1
	142	LDC	Rm,SPC	LS	1	1	4-1
	143	LDC	Rm,VBR	LS	1	1	4-1
	144	LDC.L	@Rm+,DBR	CO	4	4	4-6
	145	LDC.L	@Rm+,SGR	CO	4	4	4-6
	146	LDC.L	@Rm+,GBR	LS	1	1	4-7
	147	LDC.L	@Rm+,Rp_BANK	LS	1	1	4-5
	148	LDC.L	@Rm+,SR	CO	6+3	4	4-8
	149	LDC.L	@Rm+,SSR	LS	1	1	4-5

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
System control instructions	150	LDC.L	@Rm+,SPC	LS	1	1	4-5
	151	LDC.L	@Rm+,VBR	LS	1	1	4-5
	152	LDS	Rm,MACH	LS	1	1	5-1
	153	LDS	Rm,MACL	LS	1	1	5-1
	154	LDS	Rm,PR	LS	1	1	4-13
	155	LDS.L	@Rm+,MACH	LS	1	1	5-2
	156	LDS.L	@Rm+,MACL	LS	1	1	5-2
	157	LDS.L	@Rm+,PR	LS	1	1	4-14
	158	STC	DBR,Rn	LS	1	1	4-9
	159	STC	SGR,Rn	LS	1	1	4-9
	160	STC	GBR,Rn	LS	1	1	4-9
	161	STC	Rp_BANK,Rn	LS	1	1	4-9
	162	STC	SR,Rn	CO	1	1	4-10
	163	STC	SSR,Rn	LS	1	1	4-9
	164	STC	SPC,Rn	LS	1	1	4-9
	165	STC	VBR,Rn	LS	1	1	4-9
	166	STC.L	DBR,@-Rn	LS	1	1	4-11
	167	STC.L	SGR,@-Rn	LS	1	1	4-11
	168	STC.L	GBR,@-Rn	LS	1	1	4-11
	169	STC.L	Rp_BANK,@-Rn	LS	1	1	4-11
	170	STC.L	SR,@-Rn	CO	1	1	4-12
	171	STC.L	SSR,@-Rn	LS	1	1	4-11
	172	STC.L	SPC,@-Rn	LS	1	1	4-11
	173	STC.L	VBR,@-Rn	LS	1	1	4-11
	174	STS	MACH,Rn	LS	1	1	5-3
	175	STS	MACL,Rn	LS	1	1	5-3
	176	STS	PR,Rn	LS	1	1	4-15
	177	STS.L	MACH,@-Rn	LS	1	1	5-4
	178	STS.L	MACL,@-Rn	LS	1	1	5-4
	179	STS.L	PR,@-Rn	LS	1	1	4-16

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Single-precision floating-point instructions	180	FLDI0	FRn	LS	1	1	6-13
	181	FLDI1	FRn	LS	1	1	6-13
	182	FMOV	FRm,FRn	LS	1	1	6-9
	183	FMOV.S	@Rm,FRn	LS	1	1	6-9
	184	FMOV.S	@Rm+,FRn	LS	1	1	6-9
	185	FMOV.S	@(R0,Rm),FRn	LS	1	1	6-9
	186	FMOV.S	FRm,@Rn	LS	1	1	6-9
	187	FMOV.S	FRm,@-Rn	LS	1	1	6-9
	188	FMOV.S	FRm,@(R0,Rn)	LS	1	1	6-9
	189	FLDS	FRm,FPUL	LS	1	1	6-10
	190	FSTS	FPUL,FRn	LS	1	1	6-11
	191	FABS	FRn	LS	1	1	6-12
	192	FADD	FRm,FRn	FE	1	1	6-14
	193	FCMP/EQ	FRm,FRn	FE	1	1	6-14
	194	FCMP/GT	FRm,FRn	FE	1	1	6-14
	195	FDIV	FRm,FRn	FE	1	14	6-15
	196	FLOAT	FPUL,FRn	FE	1	1	6-14
	197	FMAC	FR0,FRm,FRn	FE	1	1	6-14
	198	FMUL	FRm,FRn	FE	1	1	6-14
	199	FNEG	FRn	LS	1	1	6-12
	200	FSQRT	FRn	FE	1	14	6-15
	201	FSUB	FRm,FRn	FE	1	1	6-14
	202	FTRC	FRm,FPUL	FE	1	1	6-14
	203	FMOV	DRm,DRn	LS	1	1	6-9
	204	FMOV	@Rm,DRn	LS	1	1	6-9
	205	FMOV	@Rm+,DRn	LS	1	1	6-9
	206	FMOV	@(R0,Rm),DRn	LS	1	1	6-9
	207	FMOV	DRm,@Rn	LS	1	1	6-9
	208	FMOV	DRm,@-Rn	LS	1	1	6-9
	209	FMOV	DRm,@(R0,Rn)	LS	1	1	6-9

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Double-precision floating-point instructions	210	FABS	DRn	LS	1	1	6-12
	211	FADD	DRm,DRn	FE	1	1	6-16
	212	FCMP/EQ	DRm,DRn	FE	1	1	6-16
	213	FCMP/GT	DRm,DRn	FE	1	1	6-16
	214	FCNVDS	DRm,FPUL	FE	1	1	6-16
	215	FCNVSD	FPUL,DRn	FE	1	1	6-16
	216	FDIV	DRm,DRn	FE	1	30	6-18
	217	FLOAT	FPUL,DRn	FE	1	1	6-16
	218	FMUL	DRm,DRn	FE	1	3	6-17
	219	FNEG	DRn	LS	1	1	6-12
	220	FSQRT	DRn	FE	1	30	6-18
	221	FSUB	DRm,DRn	FE	1	1	6-16
	222	FTRC	DRm,FPUL	FE	1	1	6-16
FPU system control instructions	223	LDS	Rm,FPUL	LS	1	1	6-1
	224	LDS	Rm,FPSCR	LS	1	1	6-5
	225	LDS.L	@Rm+,FPUL	LS	1	1	6-3
	226	LDS.L	@Rm+,FPSCR	LS	1	1	6-7
	227	STS	FPUL,Rn	LS	1	1	6-2
	228	STS	FPSCR,Rn	LS	1	1	6-6
	229	STS.L	FPUL,@-Rn	LS	1	1	6-4
	230	STS.L	FPSCR,@-Rn	LS	1	1	6-8
Graphics acceleration instructions	231	FMOV	DRm,XDn	LS	1	1	6-9
	232	FMOV	XDm,DRn	LS	1	1	6-9
	233	FMOV	XDm,XDn	LS	1	1	6-9
	234	FMOV	@Rm,XDn	LS	1	1	6-9
	235	FMOV	@Rm+,XDn	LS	1	1	6-9
	236	FMOV	@(R0,Rm),XDn	LS	1	1	6-9
	237	FMOV	XDm,@Rn	LS	1	1	6-9
	238	FMOV	XDm,@-Rn	LS	1	1	6-9
	239	FMOV	XDm,@(R0,Rn)	LS	1	1	6-9
	240	FIPR	FVm,FVn	FE	1	1	6-19

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Graphics acceleration instructions	241	FRCHG		FE	1	1	6-14
	242	FSCHG		FE	1	1	6-14
	243	FPCHG		FE	1	1	6-14
	244	FSRRA	FRn	FE	1	1	6-21
	245	FSCA	FPUL,DRn	FE	1	3	6-22
	246	FTRV	XMTRX,FVn	FE	1	4	6-20

Section 5 Exception Handling

5.1 Summary of Exception Handling

Exception handling processing is handled by a special routine which is executed by a reset, general exception handling, or interrupt. For example, if the executing instruction ends abnormally, appropriate action must be taken in order to return to the original program sequence, or report the abnormality before terminating the processing. The process of generating an exception handling request in response to abnormal termination, and passing control to a user-written exception handling routine, in order to support such functions, is given the generic name of exception handling.

The exception handling in this LSI is of three kinds: resets, general exceptions, and interrupts.

5.2 Register Descriptions

Table 5.1 lists the configuration of registers related exception handling.

Table 5.1 Register Configuration

Register Name	Abbr.	R/W	P4 Address*	Area 7 Address*	Access Size
TRAPA exception register	TRA	R/W	H'FF00 0020	H'1F00 0020	32
Exception event register	EXPEVT	R/W	H'FF00 0024	H'1F00 0024	32
Interrupt event register	INTEVT	R/W	H'FF00 0028	H'1F00 0028	32
Non-support detection exception register	EXPMASK	R/W	H'FF2F 0004	H'1F2F 0004	32

Note: * P4 is the address when virtual address space P4 area is used. Area 7 is the address when physical address space area 7 is accessed by using the TLB.

Table 5.2 States of Register in Each Operating Mode

Register Name	Abbr.	Power-on Reset	Manual Reset	Sleep	Standby
TRAPA exception register	TRA	Undefined	Undefined	Retained	Retained
Exception event register	EXPEVT	H'0000 0000	H'0000 0020	Retained	Retained
Interrupt event register	INTEVT	Undefined	Undefined	Retained	Retained
Non-support detection exception register	EXPMASK	H'0000 001F	H'0000 001F	Retained	Retained

5.2.1 TRAPA Exception Register (TRA)

The TRAPA exception register (TRA) consists of 8-bit immediate data (imm) for the TRAPA instruction. TRA is set automatically by hardware when a TRAPA instruction is executed. TRA can also be modified by software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TRACODE								—	—
Initial value:	0	0	0	0	0	0	—	—	—	—	—	—	—	—	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
9 to 2	TRACODE	Undefined	R/W	TRAPA Code 8-bit immediate data of TRAPA instruction is set
1, 0	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.

5.2.2 Exception Event Register (EXPEVT)

The exception event register (EXPEVT) consists of a 12-bit exception code. The exception code set in EXPEVT is that for a reset or general exception event. The exception code is set automatically by hardware when an exception occurs. EXPEVT can also be modified by software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	EXPCODE											
Initial value:	0	0	0	0	0	0	0	0	0	0	0/1	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
11 to 0	EXPCODE	H'000 or H'020	R/W	Exception Code The exception code for a reset or general exception is set. For details, see table 5.3.

5.2.3 Interrupt Event Register (INTEVT)

The interrupt event register (INTEVT) consists of a 14-bit exception code. The exception code is set automatically by hardware when an exception occurs. INTEVT can also be modified by software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	INTCODE													
Initial value:	0	0	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
13 to 0	INTCODE	Undefined	R/W	Exception Code The exception code for an interrupt is set. For details, see table 5.3.

5.2.4 Non-Support Detection Exception Register (EXPMASK)

The non-support detection exception register (EXPMASK) is used to enable or disable the generation of exceptions in response to the use of any of functions 1 to 3 listed below. The functions of 1 to 3 are planned not to be supported in the future SuperH-family products. The exception generation functions of EXPMASK can be used in advance of execution; the detection function then checks for the use of these functions in the software. This will ease the transfer of software to the future SuperH-family products that do not support the respective functions.

1. Handling of an instruction other than the NOP instruction in the delay slot of the RTE instruction.
2. Handling of the SLEEP instruction in the delay slot of the branch instruction.
3. Performance of IC/OC memory-mapped associative write operations.

According to the value of EXPMASK, functions 1 and 2 can generate a slot illegal instruction exception, and 3 can generate a data address error exception.

Generation of each exception can be disabled by writing 1 to the corresponding bit in EXPMASK. However, it is recommended that the above functions should not be used when making a program to maintain the compatibility with the future products.

Use the store instruction of the CPU to update EXPMASK. After updating the register and then reading the register once, execute either of the following instructions. Executing either instruction guarantees the operation with the updated register value.

- Execute the RTE instruction.
- Execute the ICBI instruction for any address (including non-cacheable area).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	MM CAW	–	–	BRDS SLP	RTE DS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved For details on reading/writing these bits, see General Precautions on Handling of Product.
4	MMCAW	1	R/W	Memory-Mapped Cache Associative Write 0: Memory-mapped cache associative write is disabled. (A data address error exception will occur.) 1: Memory-mapped cache associative write is enabled. For further details, refer to section 8.6.5, Memory-Mapped Cache Associative Write Operation.
3, 2	—	All 1	R	Reserved For details on reading/writing these bits, see General Precautions on Handling of Product.
1	BRDSSLP	1	R/W	Delay Slot SLEEP Instruction 0: The SLEEP instruction in the delay slot is disabled. (The SLEEP instruction is taken as a slot illegal instruction.) 1: The SLEEP instruction in the delay slot is enabled.
0	RTEDS	1	R/W	RTE Delay Slot 0: An instruction other than the NOP instruction in the delay slot of the RTE instruction is disabled. (An instruction other than the NOP instruction is taken as a slot illegal instruction). 1: An instruction other than the NOP instruction in the delay slot of the RTE instruction is enabled.

Note: The initial values of bits 4, 1, and 0 depend on the product. See the manual of the product for details.

5.3 Exception Handling Functions

5.3.1 Exception Handling Flow

In exception handling, the contents of the program counter (PC), status register (SR), and R15 are saved in the saved program counter (SPC), saved status register (SSR), and saved general register15 (SGR), and the CPU starts execution of the appropriate exception handling routine according to the vector address. An exception handling routine is a program written by the user to handle a specific exception. The exception handling routine is terminated and control returned to the original program by executing a return-from-exception instruction (RTE). This instruction restores the PC and SR contents and returns control to the normal processing routine at the point at which the exception occurred. The SGR contents are not written back to R15 with an RTE instruction.

The basic processing flow is as follows. For the meaning of the SR bits, see section 2, Programming Model.

1. The PC, SR, and R15 contents are saved in SPC, SSR, and SGR, respectively.
2. The block bit (BL) in SR is set to 1.
3. The mode bit (MD) in SR is set to 1.
4. The register bank bit (RB) in SR is set to 1.
5. In a reset, the FPU disable bit (FD) in SR is cleared to 0.
6. The exception code is written to bits 11 to 0 of the exception event register (EXPEVT) or interrupt event register (INTEVT).
7. When the interrupt mode switch bit (INTMU) in CPUOPM has been 1, the interrupt mask level bit (IMASK) in SR is changed to accepted interrupt level.
8. The CPU branches to the determined exception handling vector address, and the exception handling routine begins.

5.3.2 Exception Handling Vector Addresses

The reset vector address is fixed at H'A0000000. Exception and interrupt vector addresses are determined by adding the offset for the specific event to the vector base address, which is set by software in the vector base register (VBR). In the case of the TLB miss exception, for example, the offset is H'00000400, so if H'9C080000 is set in VBR, the exception handling vector address will be H'9C080400. If a further exception occurs at the exception handling vector address, a duplicate exception will result, and recovery will be difficult; therefore, addresses that are not to be converted (in P1 and P2 areas) should be specified for vector addresses.

5.4 Exception Types and Priorities

Table 5.3 shows the types of exceptions, with their relative priorities, vector addresses, and exception/interrupt codes.

Table 5.3 Exceptions

					Exception Transition Direction*3		
Exception Category	**Execution Mode**	**Exception**	**Priority Level*2**	**Priority Order*2**	**Vector Address**	**Offset**	**Exception Code*4**
Reset	Abort type	Power-on reset	1	1	H'A000 0000	—	H'000
		Manual reset	1	2	H'A000 0000	—	H'020
		H-UDI reset	1	1	H'A000 0000	—	H'000
		Instruction TLB multiple-hit exception	1	3	H'A000 0000	—	H'140
		Data TLB multiple-hit exception	1	4	H'A000 0000	—	H'140
General exception	Re-execution type	User break before instruction execution*1	2	0	(VBR/DBR)	H'100/—	H'1E0
		Instruction address error	2	1	(VBR)	H'100	H'0E0
		Instruction TLB miss exception	2	2	(VBR)	H'400	H'040
		Instruction TLB protection violation exception	2	3	(VBR)	H'100	H'0A0
		General illegal instruction exception	2	4	(VBR)	H'100	H'180
		Slot illegal instruction exception	2	4	(VBR)	H'100	H'1A0
		General FPU disable exception	2	4	(VBR)	H'100	H'800
		Slot FPU disable exception	2	4	(VBR)	H'100	H'820
		Data address error (read)	2	5	(VBR)	H'100	H'0E0
		Data address error (write)	2	5	(VBR)	H'100	H'100
		Data TLB miss exception (read)	2	6	(VBR)	H'400	H'040
		Data TLB miss exception (write)	2	6	(VBR)	H'400	H'060
		Data TLB protection violation exception (read)	2	7	(VBR)	H'100	H'0A0
		Data TLB protection violation exception (write)	2	7	(VBR)	H'100	H'0C0
		FPU exception	2	8	(VBR)	H'100	H'120
		Initial page write exception	2	9	(VBR)	H'100	H'080

Exception Category	Execution Mode	Exception	Priority Level*2	Priority Order*2	Exception Transition Direction*3		Exception Code*4
					Vector Address	Offset	
General exception	Completion type	Unconditional trap (TRAPA)	2	4	(VBR)	H'100	H'160
		User break after instruction execution*1	2	10	(VBR/DBR)	H'100/—	H'1E0
Interrupt	Completion type	Nonmaskable interrupt	3	—	(VBR)	H'600	H'1C0
		General interrupt request	4	—	(VBR)	H'600	—

Note: 1. When UBDE in CBCR = 1, PC = DBR. In other cases, PC = VBR + H'100.
2. Priority is first assigned by priority level, then by priority order within each level (the lowest number represents the highest priority).
3. Control passes to H'A000 0000 in a reset, and to [VBR + offset] in other cases.
4. Stored in EXPEVT for a reset or general exception, and in INTEVT for an interrupt.

5.5 Exception Flow

5.5.1 Exception Flow

Figure 5.1 shows an outline flowchart of the basic operations in instruction execution and exception handling. For the sake of clarity, the following description assumes that instructions are executed sequentially, one by one. Figure 5.1 shows the relative priority order of the different kinds of exceptions (reset, general exception, and interrupt). Register settings in the event of an exception are shown only for SSR, SPC, SGR, EXPEVT/INTEVT, SR, and PC. However, other registers may be set automatically by hardware, depending on the exception. For details, see section 5.6, Description of Exceptions. Also, see section 5.6.4, Priority Order with Multiple Exceptions, for exception handling during execution of a delayed branch instruction and a delay slot instruction, or in the case of instructions in which two data accesses are performed.

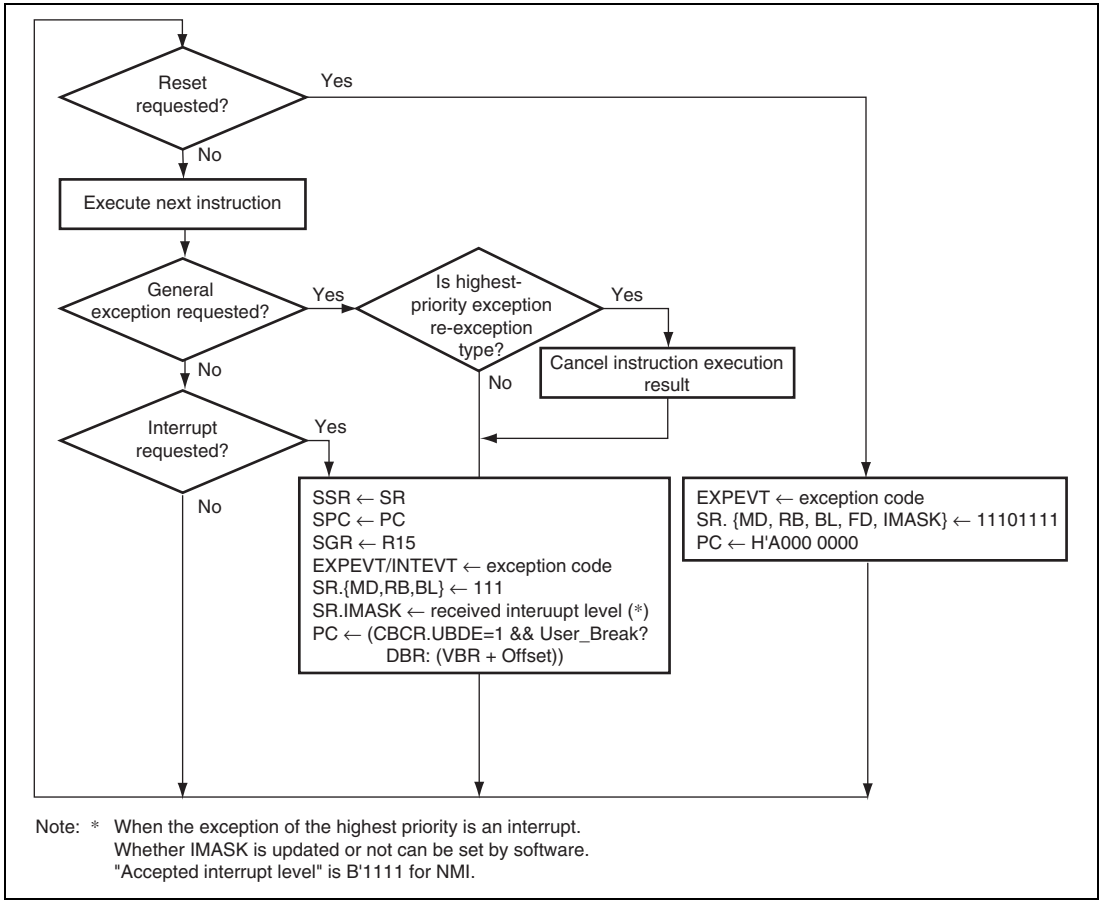


Figure 5.1 Instruction Execution and Exception Handling

5.5.2 Exception Source Acceptance

A priority ranking is provided for all exceptions for use in determining which of two or more simultaneously generated exceptions should be accepted. Five of the general exceptions—general illegal instruction exception, slot illegal instruction exception, general FPU disable exception, slot FPU disable exception, and unconditional trap exception—are detected in the process of instruction decoding, and do not occur simultaneously in the instruction pipeline. These exceptions therefore all have the same priority. General exceptions are detected in the order of instruction execution. However, exception handling is performed in the order of instruction flow (program order). Thus, an exception for an earlier instruction is accepted before that for a later instruction. An example of the order of acceptance for general exceptions is shown in figure 5.2.

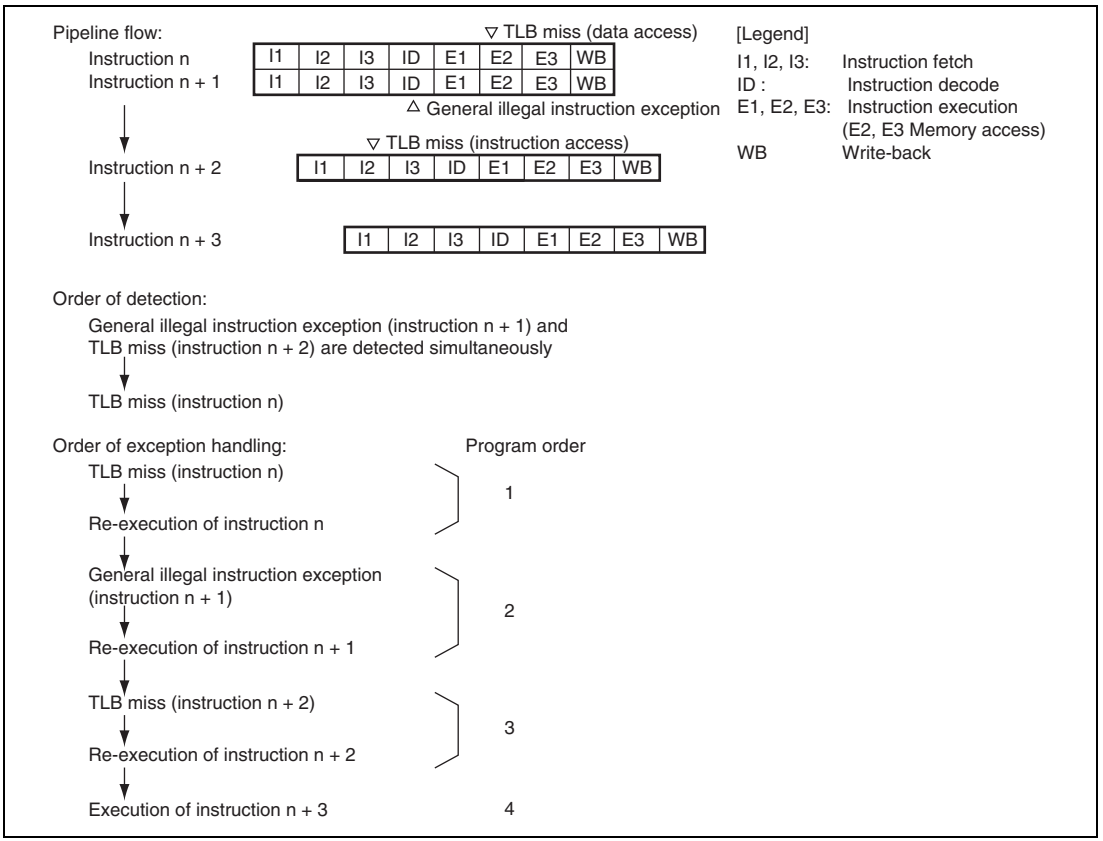


Figure 5.2 Example of General Exception Acceptance Order

5.5.3 Exception Requests and BL Bit

When the BL bit in SR is 0, general exceptions and interrupts are accepted.

When the BL bit in SR is 1 and an general exception other than a user break is generated, the CPU's internal registers and the registers of the other modules are set to their states following a manual reset, and the CPU branches to the same address as in a reset (H'A0000000). For the operation in the event of a user break, see section 30, User Break Controller (UBC). If an ordinary interrupt occurs, the interrupt request is held pending and is accepted after the BL bit has been cleared to 0 by software. If a nonmaskable interrupt (NMI) occurs, it can be held pending or accepted according to the setting made by software. For further details, refer to the hardware manual of the product.

Thus, normally, SPC and SSR are saved and then the BL bit in SR is cleared to 0, to enable multiple exception state acceptance.

5.5.4 Return from Exception Handling

The RTE instruction is used to return from exception handling. When the RTE instruction is executed, the SPC contents are restored to PC and the SSR contents to SR, and the CPU returns from the exception handling routine by branching to the SPC address. If SPC and SSR were saved to external memory, set the BL bit in SR to 1 before restoring the SPC and SSR contents and issuing the RTE instruction.

5.6 Description of Exceptions

The various exception handling operations explained here are exception sources, transition address on the occurrence of exception, and processor operation when a transition is made.

5.6.1 Resets

(1) Power-On Reset

- Condition:
  Power-on reset request
- Operations:
  Exception code H'000 is set in EXPEVT, initialization of the CPU and on-chip peripheral module is carried out, and then a branch is made to the reset vector (H'A0000000). For details, see the section 14, Reset and Power-Down Modes, and the register descriptions in the relevant sections. A power-on reset should be executed when power is supplied.

(2) Manual Reset

- Condition:
  Manual reset request
- Operations:
  Exception code H'020 is set in EXPEVT, initialization of the CPU and on-chip peripheral module is carried out, and then a branch is made to the branch vector (H'A0000000). The registers initialized by a power-on reset and manual reset are different. For details, see the section 14, Reset and Power-Down Modes, and the register descriptions in the relevant sections. In cases where the register descriptions in the relevant sections do not include the descriptions of manual reset, the same states as those of the power-on reset are applied.

(3) H-UDI Reset

- Source: SDIR.TI[7:4] = B'0110 (negation) or B'0111 (assertion)
- Transition address: H'A0000000
- Transition operations:
  Exception code H'000 is set in EXPEVT, initialization of VBR and SR is performed, and a branch is made to PC = H'A0000000.
  CPU and on-chip peripheral module initialization is performed. For details, see section 31, User Debugging Interface (H-UDI), and the register descriptions in the relevant sections.

(4) Instruction TLB Multiple Hit Exception

- Source: Multiple ITLB address matches
- Transition address: H'A0000000
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  Exception code H'140 is set in EXPEVT, initialization of VBR and SR is performed, and a branch is made to PC = H'A0000000.

  CPU and on-chip peripheral module initialization is performed in the same way as in a manual reset. For details, see the register descriptions in the relevant sections.

(5) Data TLB Multiple-Hit Exception

- Source: Multiple UTLB address matches
- Transition address: H'A0000000
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  Exception code H'140 is set in EXPEVT, initialization of VBR and SR is performed, and a branch is made to PC = H'A0000000.

  CPU and on-chip peripheral module initialization is performed in the same way as in a manual reset. For details, see the register descriptions in the relevant sections.

5.6.2 General Exceptions

(1) Data TLB Miss Exception

- Source: Address mismatch in UTLB address comparison
- Transition address: VBR + H'00000400
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'040 (for a read access) or H'060 (for a write access) is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0400.

  To speed up TLB miss processing, the offset is separate from that of other exceptions.

```
Data_TLB_miss_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = read_access ? H'0000 0040 : H'0000 0060;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0400;
}
```

(2) Instruction TLB Miss Exception

- Source: Address mismatch in ITLB address comparison
- Transition address: VBR + H'00000400
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'40 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0400.

  To speed up TLB miss processing, the offset is separate from that of other exceptions.

```
ITLB_miss_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0040;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0400;
}
```

(3) Initial Page Write Exception

- Source: TLB is hit in a store access, but dirty bit D = 0
- Transition address: VBR + H'00000100
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'080 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

```
Initial_write_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0080;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

(4) Data TLB Protection Violation Exception

- Source: The access does not accord with the UTLB protection information (PR bits or EPR bits) shown in table 5.4 and table 5.5.

Table 5.4 UTLB Protection Information (TLB Compatible Mode)

PR	Privileged Mode	User Mode
00	Only read access possible	Access not possible
01	Read/write access possible	Access not possible
10	Only read access possible	Only read access possible
11	Read/write access possible	Read/write access possible

Table 5.5 UTLB Protection Information (TLB Extended Mode)

EPR [5]	Read Permission in Privileged Mode
0	Read access possible
1	Read access not possible

EPR [4]	Write Permission in Privileged Mode
0	Write access possible
1	Write access not possible

EPR [2]	Read Permission in User Mode
0	Read access possible
1	Read access not possible

EPR [1]	Write Permission in User Mode
0	Write access possible
1	Write access not possible

- Transition address: VBR + H'00000100
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

Exception code H'0A0 (for a read access) or H'0C0 (for a write access) is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

```
Data_TLB_protection_violation_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = read_access ? H'0000 00A0 : H'0000 00C0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

(5) Instruction TLB Protection Violation Exception

- Source: The access does not accord with the ITLB protection information (PR bits or EPR bits) shown in table 5.6 and table5.7.

Table 5.6 ITLB Protection Information (TLB Compatible Mode)

PR	Privileged Mode	User Mode
0	Access possible	Access not possible
1	Access possible	Access possible

Table 5.7 ITLB Protection Information (TLB Extended Mode)

EPR [5], EPR [3]	Execution Permission in Privileged Mode
11	Execution of instructions possible
10	Instruction fetch not possible Execution of Rn access by ICBI possible
00	Execution of instructions not possible

EPR [2], EPR [0]	Execution Permission in User Mode
11, 01	Execution of instructions possible
10	Instruction fetch not possible Execution of Rn access by ICBI possible
00	Execution of instructions not possible

- Transition address: VBR + H'00000100
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'0A0 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

```
ITLB_protection_violation_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 00A0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

(6) Data Address Error

- Sources:
  - Word data access from other than a word boundary (2n +1)
  - Longword data access from other than a longword data boundary (4n +1, 4n + 2, or 4n +3) (Except MOVLIA)
  - Quadword data access from other than a quadword data boundary (8n +1, 8n + 2, 8n +3, 8n + 4, 8n + 5, 8n + 6, or 8n + 7)
  - Access to area H'80000000 to H'FFFFFFFF in user mode

    Areas H'E0000000 to H'E3FFFFFF and H'E5000000 to H'E5FFFFFF can be accessed in user mode. For details, see section 7, Memory Management Unit (MMU) and section 9, On-Chip Memory.
  - The MMCAW bit in EXPMASK is 0, and the IC/OC memory mapped associative write is performed. For details of memory mapped associative write, see section 8.6.5, Memory-Mapped Cache Associative Write Operation.
- Transition address: VBR + H'0000100

- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'0E0 (for a read access) or H'100 (for a write access) is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. For details, see section 7, Memory Management Unit (MMU).

```
Data_address_error()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = read_access? H'0000 00E0: H'0000 0100;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

(7) Instruction Address Error

- Sources:
  - — Instruction fetch from other than a word boundary (2n +1)
  - — Instruction fetch from area H'80000000 to H'FFFFFFFF in user mode

    Area H'E5000000 to H'E5FFFFFF can be accessed in user mode. For details, see section 9, On-Chip Memory.
- Transition address: VBR + H'00000100
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in the SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'0E0 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. For details, see section 7, Memory Management Unit (MMU).

```
Instruction_address_error()
{
   TEA = EXCEPTION_ADDRESS;
   PTEH.VPN = PAGE_NUMBER;
   SPC = PC;
   SSR = SR;
   SGR = R15;
   EXPEVT = H'0000 00E0;
   SR.MD = 1;
   SR.RB = 1;
   SR.BL = 1;
   PC = VBR + H'0000 0100;
}
```

(8) Unconditional Trap

- Source: Execution of TRAPA instruction
- Transition address: VBR + H'00000100
- Transition operations:

  As this is a processing-completion-type exception, the PC contents for the instruction following the TRAPA instruction are saved in SPC. The value of SR and R15 when the TRAPA instruction is executed are saved in SSR and SGR. The 8-bit immediate value in the TRAPA instruction is multiplied by 4, and the result is set in TRA [9:0]. Exception code H'160 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

```
TRAPA_exception()
{
    SPC = PC + 2;
    SSR = SR;
    SGR = R15;
    TRA = imm << 2;
    EXPEVT = H'0000 0160;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

(9) General Illegal Instruction Exception

- Sources:
  - — Decoding of an undefined instruction not in a delay slot
    Delayed branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S
    Undefined instruction: H'FFFD
  - — Decoding in user mode of a privileged instruction not in a delay slot
    Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP, but excluding LDC/STC instructions that access GBR
- Transition address: VBR + H'00000100
- Transition operations:
  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.
  Exception code H'180 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. Operation is not guaranteed if an undefined code other than H'FFFD is decoded.

```
General_illegal_instruction_exception()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0180;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

(10) Slot Illegal Instruction Exception

- Sources:
  - — Decoding of an undefined instruction in a delay slot
    Delayed branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S
    Undefined instruction: H'FFFD
  - — Decoding of an instruction that modifies PC in a delay slot
    Instructions that modify PC: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT, BF, BT/S, BF/S, TRAPA, LDC Rm,SR, LDC.L @Rm+,SR, ICBI, PREFI
  - — Decoding in user mode of a privileged instruction in a delay slot
    Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP, but excluding LDC/STC instructions that access GBR
  - — Decoding of a PC-relative MOV instruction or MOVA instruction in a delay slot
  - — The BRDSSLP bit in EXPMASK is 0, and the SLEEP instruction in the delay slot is executed.
  - — The RTEDS bit in EXPMASK is 0, and an instruction other than the NOP instruction in the delay slot is executed.
- Transition address: VBR + H'000 0100
- Transition operations:
  The PC contents for the preceding delayed branch instruction are saved in SPC. The SR and R15 contents when this exception occurred are saved in SSR and SGR.
  Exception code H'1A0 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. Operation is not guaranteed if an undefined code other than H'FFFD is decoded.

```
Slot_illegal_instruction_exception()
{
    SPC = PC - 2;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 01A0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

(11) General FPU Disable Exception

- Source: Decoding of an FPU instruction* not in a delay slot with SR.FD = 1
- Transition address: VBR + H'00000100
- Transition operations:

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'800 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

Note: * FPU instructions are instructions in which the first 4 bits of the instruction code are F (but excluding undefined instruction H'FFFD), and the LDS, STS, LDS.L, and STS.L instructions corresponding to FPUL and FPSCR.

```
General_fpu_disable_exception()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0800;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

(12) Slot FPU Disable Exception

- Source: Decoding of an FPU instruction in a delay slot with SR.FD =1
- Transition address: VBR + H'00000100
- Transition operations:

  The PC contents for the preceding delayed branch instruction are saved in SPC. The SR and R15 contents when this exception occurred are saved in SSR and SGR.

  Exception code H'820 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

```
Slot_fpu_disable_exception()
{
    SPC = PC - 2;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0820;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

(13) Pre-Execution User Break/Post-Execution User Break

- Source: Fulfilling of a break condition set in the user break controller
- Transition address: VBR + H'00000100, or DBR
- Transition operations:

  In the case of a post-execution break, the PC contents for the instruction following the instruction at which the breakpoint is set are set in SPC. In the case of a pre-execution break, the PC contents for the instruction at which the breakpoint is set are set in SPC.

  The SR and R15 contents when the break occurred are saved in SSR and SGR. Exception code H'1E0 is set in EXPEVT.

  The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. It is also possible to branch to PC = DBR.

  For details of PC, etc., when a data break is set, see section 30, User Break Controller (UBC).

```
User_break_exception()
{
    SPC = (pre_execution break? PC : PC + 2);
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 01E0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = (BRCR.UBDE==1 ? DBR : VBR + H'0000 0100);
}
```

(14) FPU Exception

- Source: Exception due to execution of a floating-point operation
- Transition address: VBR + H'00000100
- Transition operations:

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR . The R15 contents at this time are saved in SGR. Exception code H'120 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

```
FPU_exception()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0120;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

5.6.3 Interrupts

(1) NMI (Nonmaskable Interrupt)

- Source: NMI pin edge detection
- Transition address: VBR + H'00000600
- Transition operations:

  The PC and SR contents for the instruction immediately after this exception is accepted are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'1C0 is set in INTEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0600. When the BL bit in SR is 0, this interrupt is not masked by the interrupt mask bits in SR, and is accepted at the highest priority level. When the BL bit in SR is 1, a software setting can specify whether this interrupt is to be masked or accepted. When the INTMU bit in CPUOPM is 1 and the NMI interrupt is accessed, B'1111 is set to IMASK bit in SR. For details, see section 10, Interrupt Controller (INTC).

```
NMI()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    INTEVT = H'0000 01C0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    If (cond) SR.IMASK = B'1111;
    PC = VBR + H'0000 0600;
}
```

(2) General Interrupt Request

- Source: The interrupt mask level bits setting in SR is smaller than the interrupt level of interrupt request, and the BL bit in SR is 0 (accepted at instruction boundary).
- Transition address: VBR + H'00000600
- Transition operations:

  The PC contents immediately after the instruction at which the interrupt is accepted are set in SPC. The SR and R15 contents at the time of acceptance are set in SSR and SGR.

The code corresponding to the each interrupt source is set in INTEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to VBR + H'0600. When the INTMU bit in CPUOPM is 1, IMASK bit in SR is changed to accepted interrupt level. For details, see section 10, Interrupt Controller (INTC).

```
Module_interruption()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    INTEVT = H'0000 0400 ~ H'0000 3FE0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    if (cond) SR.IMASK = level_of accepted_interrupt ();
    PC = VBR + H'0000 0600;
}
```

5.6.4 Priority Order with Multiple Exceptions

With some instructions, such as instructions that make two accesses to memory, and the indivisible pair comprising a delayed branch instruction and delay slot instruction, multiple exceptions occur. Care is required in these cases, as the exception priority order differs from the normal order.

(1) Instructions that Make Two Accesses to Memory

With MAC instructions, memory-to-memory arithmetic/logic instructions, TAS instructions, and MOVUA instructions, two data transfers are performed by a single instruction, and an exception will be detected for each of these data transfers. In these cases, therefore, the following order is used to determine priority.

1. Data address error in first data transfer
2. TLB miss in first data transfer
3. TLB protection violation in first data transfer
4. Initial page write exception in first data transfer
5. Data address error in second data transfer
6. TLB miss in second data transfer

7. TLB protection violation in second data transfer
8. Initial page write exception in second data transfer

(2) Indivisible Delayed Branch Instruction and Delay Slot Instruction

As a delayed branch instruction and its associated delay slot instruction are indivisible, they are treated as a single instruction. Consequently, the priority order for exceptions that occur in these instructions differs from the usual priority order. The priority order shown below is for the case where the delay slot instruction has only one data transfer.

1. A check is performed for the interrupt type and re-execution type exceptions of priority levels 1 and 2 in the delayed branch instruction.
2. A check is performed for the interrupt type and re-execution type exceptions of priority levels 1 and 2 in the delay slot instruction.
3. A check is performed for the completion type exception of priority level 2 in the delayed branch instruction.
4. A check is performed for the completion type exception of priority level 2 in the delay slot instruction.
5. A check is performed for priority level 3 in the delayed branch instruction and priority level 3 in the delay slot instruction. (There is no priority ranking between these two.)
6. A check is performed for priority level 4 in the delayed branch instruction and priority level 4 in the delay slot instruction. (There is no priority ranking between these two.)

If the delay slot instruction has a second data transfer, two checks are performed in step 2, as in the above case (Instructions that make two accesses to memory).

If the accepted exception (the highest-priority exception) is a delay slot instruction re-execution type exception, the branch instruction PR register write operation (PC → PR operation performed in a BSR, BSRF, or JSR instruction) is not disabled. Note that in this case, the contents of PR register are not guaranteed.

5.7 Usage Notes

(1) Return from Exception Handling

A. Check the BL bit in SR with software. If SPC and SSR have been saved to memory, set the BL bit in SR to 1 before restoring them.

B. Issue an RTE instruction. When RTE is executed, the SPC contents are saved in PC, the SSR contents are saved in SR, and branch is made to the SPC address to return from the exception handling routine.

(2) If a General Exception or Interrupt Occurs When BL Bit in SR = 1

A. General exception

When a general exception other than a user break occurs, the PC value for the instruction at which the exception occurred in SPC, and a manual reset is executed. The value in EXPEVT at this time is H'00000020; the SSR contents are undefined.

B. Interrupt

If an ordinary interrupt occurs, the interrupt request is held pending and is accepted after the BL bit in SR has been cleared to 0 by software. If a nonmaskable interrupt (NMI) occurs, it can be held pending or accepted according to the setting made by software.

In sleep or standby mode, however, an interrupt is accepted even if the BL bit in SR is set to 1.

(3) SPC when an Exception Occurs

A. Re-execution type general exception

The PC value for the instruction at which the exception occurred is set in SPC, and the instruction is re-executed after returning from the exception handling routine. If an exception occurs in a delay slot instruction, however, the PC value for the delayed branch instruction is saved in SPC regardless of whether or not the preceding delay slot instruction condition is satisfied.

B. Completion type general exception or interrupt

The PC value for the instruction following that at which the exception occurred is set in SPC. If an exception occurs in a branch instruction with delay slot, however, the PC value for the branch destination is saved in SPC.

(4) RTE Instruction Delay Slot

A. The instruction in the delay slot of the RTE instruction is executed only after the value saved in SSR has been restored to SR. The acceptance of the exception related to the instruction access is determined depending on SR before restoring, while the acceptance of other exceptions is determined depending on the processing mode by SR after restoring or the BL bit. The completion type exception is accepted before branching to the destination of RTE instruction. However, if the re-execution type exception is occurred, the operation cannot be guaranteed.

B. The user break is not accepted by the instruction in the delay slot of the RTE instruction.

(5) Changing the SR Register Value and Accepting Exception

A. When the MD or BL bit in the SR register is changed by the LDC instruction, the acceptance of the exception is determined by the changed SR value, starting from the next instruction.* In the completion type exception, an exception is accepted after the next instruction has been executed. However, an interrupt of completion type exception is accepted before the next instruction is executed.

Note: * When the LDC instruction for SR is executed, following instructions are fetched again and the instruction fetch exception is evaluated again by the changed SR.

Section 6 Floating-Point Unit (FPU)

6.1 Features

The FPU has the following features.

- Conforms to IEEE754 standard
- 32 single-precision floating-point registers (can also be referenced as 16 double-precision registers)
- Two rounding modes: Round to Nearest and Round to Zero
- Two denormalization modes: Flush to Zero and Treat Denormalized Number
- Six exception sources: FPU Error, Invalid Operation, Divide By Zero, Overflow, Underflow, and Inexact
- Comprehensive instructions: Single-precision, double-precision, graphics support, and system control
- In this LSI, the following three instructions are added on to the instruction set of the SH-4
  FSRRA, FSCA, and FPCHG

When the FD bit in SR is set to 1, the FPU cannot be used, and an attempt to execute an FPU instruction will cause an FPU disable exception (general FPU disable exception or slot FPU disable exception).

6.2 Data Formats

6.2.1 Floating-Point Format

A floating-point number consists of the following three fields:

- Sign bit (s)
- Exponent field (e)
- Fraction field (f)

This LSI can handle single-precision and double-precision floating-point numbers, using the formats shown in figures 6.1 and 6.2.

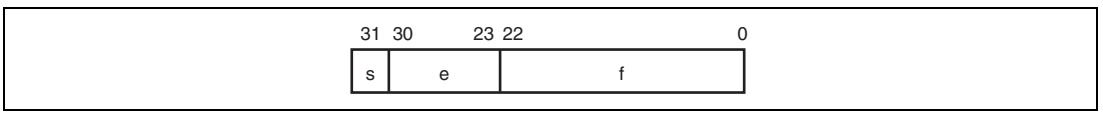


Figure 6.1 Format of Single-Precision Floating-Point Number

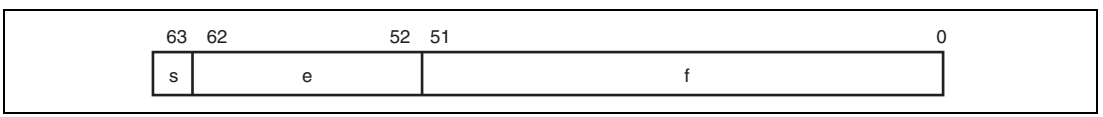


Figure 6.2 Format of Double-Precision Floating-Point Number

The exponent is expressed in biased form, as follows:

$$e = E + bias$$

The range of unbiased exponent E is $E_{min} - 1$ to $E_{max} + 1$. The two values $E_{min} - 1$ and $E_{max} + 1$ are distinguished as follows. $E_{min} - 1$ indicates zero (both positive and negative sign) and a denormalized number, and $E_{max} + 1$ indicates positive or negative infinity or a non-number (NaN). Table 6.1 shows floating-point formats and parameters.

Table 6.1 Floating-Point Number Formats and Parameters

Parameter	Single-Precision	Double-Precision
Total bit width	32 bits	64 bits
Sign bit	1 bit	1 bit
Exponent field	8 bits	11 bits
Fraction field	23 bits	52 bits
Precision	24 bits	53 bits
Bias	+127	+1023
E_{max}	+127	+1023
E_{min}	–126	–1022

Floating-point number value v is determined as follows:

If $E = E_{max} + 1$ and $f \neq 0$, v is a non-number (NaN) irrespective of sign s
If $E = E_{max} + 1$ and $f = 0$, $v = (-1)^s$ (infinity) [positive or negative infinity]
If $E_{min} \leq E \leq E_{max}$, $v = (-1)^s 2^E$ (1.f) [normalized number]
If $E = E_{min} - 1$ and $f \neq 0$, $v = (-1)^s 2^{Emin}$ (0.f) [denormalized number]
If $E = E_{min} - 1$ and $f = 0$, $v = (-1)^s 0$ [positive or negative zero]

Table 6.2 shows the ranges of the various numbers in hexadecimal notation. For the signaling non-number and quiet non-number, see section 6.2.2, Non-Numbers (NaN). For the denormalized number, see section 6.2.3, Denormalized Numbers.

Table 6.2 Floating-Point Ranges

Type	Single-Precision	Double-Precision
Signaling non-number	H'7FFF FFFF to H'7FC0 0000	H'7FFF FFFF FFFF FFFF to H'7FF8 0000 0000 0000
Quiet non-number	H'7FBF FFFF to H'7F80 0001	H'7FF7 FFFF FFFF FFFF to H'7FF0 0000 0000 0001
Positive infinity	H'7F80 0000	H'7FF0 0000 0000 0000
Positive normalized number	H'7F7F FFFF to H'0080 0000	H'7FEF FFFF FFFF FFFF to H'0010 0000 0000 0000
Positive denormalized number	H'007F FFFF to H'0000 0001	H'000F FFFF FFFF FFFF to H'0000 0000 0000 0001
Positive zero	H'0000 0000	H'0000 0000 0000 0000
Negative zero	H'8000 0000	H'8000 0000 0000 0000
Negative denormalized number	H'8000 0001 to H'807F FFFF	H'8000 0000 0000 0001 to H'800F FFFF FFFF FFFF
Negative normalized number	H'8080 0000 to H'FF7F FFFF	H'8010 0000 0000 0000 to H'FFEF FFFF FFFF FFFF
Negative infinity	H'FF80 0000	H'FFF0 0000 0000 0000
Quiet non-number	H'FF80 0001 to H'FFBF FFFF	H'FFF0 0000 0000 0001 to H'FFF7 FFFF FFFF FFFF
Signaling non-number	H'FFC0 0000 to H'FFFF FFFF	H'FFF8 0000 0000 0000 to H'FFFF FFFF FFFF FFFF

6.2.2 Non-Numbers (NaN)

Figure 6.3 shows the bit pattern of a non-number (NaN). A value is NaN in the following case:

- Sign bit: Don't care
- Exponent field: All bits are 1
- Fraction field: At least one bit is 1

The NaN is a signaling NaN (sNaN) if the MSB of the fraction field is 1, and a quiet NaN (qNaN) if the MSB is 0.

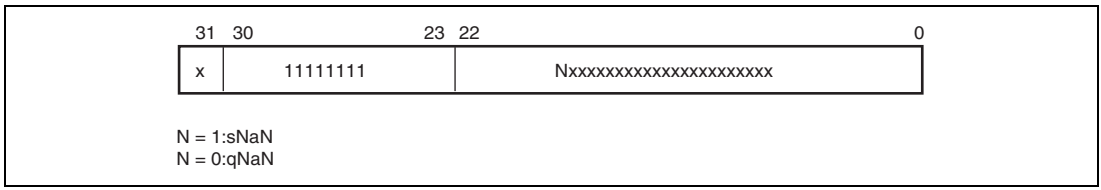


Figure 6.3 Single-Precision NaN Bit Pattern

An sNaN is assumed to be the input data in an operation, except the transfer instructions between registers, FABS, and FNEG, that generates a floating-point value.

- When the EN.V bit in FPSCR is 0, the operation result (output) is a qNaN.
- When the EN.V bit in FPSCR is 1, an invalid operation exception will be generated. In this case, the contents of the operation destination register are unchanged.

Following three instructions are used as transfer instructions between registers.

- FMOV FRm,FRn
- FLDS FRm,FPUL
- FSTS FPUL,FRn

If a qNaN is input in an operation that generates a floating-point value, and an sNaN has not been input in that operation, the output will always be a qNaN irrespective of the setting of the EN.V bit in FPSCR. An exception will not be generated in this case.

The qNAN values as operation results are as follows:

- Single-precision qNaN: H'7FBF FFFF
- Double-precision qNaN: H'7FF7 FFFF FFFF FFFF

See section 11, Instruction Descriptions of the SH-4A Extended Functions Software Manual for details of floating-point operations when a non-number (NaN) is input.

6.2.3 Denormalized Numbers

For a denormalized number floating-point value, the exponent field is expressed as 0, and the fraction field as a non-zero value.

When the DN bit in FPSCR of the FPU is 1, a denormalized number (source operand or operation result) is always positive or negative zero in a floating-point operation that generates a value (an operation other than transfer instructions between registers, FNEG, or FABS).

When the DN bit in FPSCR is 0, a denormalized number (source operand or operation result) is processed as it is. See section 11, Instruction Descriptions of the SH-4A Extended Functions Software Manual for details of floating-point operations when a denormalized number is input.

6.3 Register Descriptions

6.3.1 Floating-Point Registers

Figure 6.4 shows the floating-point register configuration. There are thirty-two 32-bit floating-point registers comprised with two banks: FPR0_BANK0 to FPR15_BANK0, and FPR0_BANK1 to FPR15_BANK1. These thirty-two registers are referenced as FR0 to FR15, DR0/2/4/6/8/10/12/14, FV0/4/8/12, XF0 to XF15, XD0/2/4/6/8/10/12/14, and XMTRX. Corresponding registers to FPR0_BANK0 to FPR15_BANK0, and FPR0_BANK1 to FPR15_BANK1 are determined according to the FR bit of FPSCR.

1. Floating-point registers, FPRi_BANKj (32 registers)
   FPR0_BANK0 to FPR15_BANK0
   FPR0_BANK1 to FPR15_BANK1

2. Single-precision floating-point registers, FRi (16 registers)
   When FPSCR.FR = 0, FR0 to FR15 are allocated to FPR0_BANK0 to FPR15_BANK0;
   when FPSCR.FR = 1, FR0 to FR15 are allocated to FPR0_BANK1 to FPR15_BANK1.

3. Double-precision floating-point registers, DRi (8 registers): A DR register comprises two FR registers.
   DR0 = {FR0, FR1}, DR2 = {FR2, FR3}, DR4 = {FR4, FR5}, DR6 = {FR6, FR7},
   DR8 = {FR8, FR9}, DR10 = {FR10, FR11}, DR12 = {FR12, FR13}, DR14 = {FR14, FR15}

4. Single-precision floating-point vector registers, FVi (4 registers): An FV register comprises four FR registers.
   FV0 = {FR0, FR1, FR2, FR3}, FV4 = {FR4, FR5, FR6, FR7},
   FV8 = {FR8, FR9, FR10, FR11}, FV12 = {FR12, FR13, FR14, FR15}

5. Single-precision floating-point extended registers, XFi (16 registers)
   When FPSCR.FR = 0, XF0 to XF15 are allocated to FPR0_BANK1 to FPR15_BANK1;
   when FPSCR.FR = 1, XF0 to XF15 are allocated to FPR0_BANK0 to FPR15_BANK0.

6. Double-precision floating-point extended registers, XDi (8 registers): An XD register comprises two XF registers.
   XD0 = {XF0, XF1}, XD2 = {XF2, XF3}, XD4 = {XF4, XF5}, XD6 = {XF6, XF7},
   XD8 = {XF8, XF9}, XD10 = {XF10, XF11}, XD12 = {XF12, XF13}, XD14 = {XF14, XF15}

7. Single-precision floating-point extended register matrix, XMTRX: XMTRX comprises all 16 XF registers.

$$XMTRX = \begin{bmatrix} XF0 & XF4 & XF8 & XF12 \\ XF1 & XF5 & XF9 & XF13 \\ XF2 & XF6 & XF10 & XF14 \\ XF3 & XF7 & XF11 & XF15 \end{bmatrix}$$

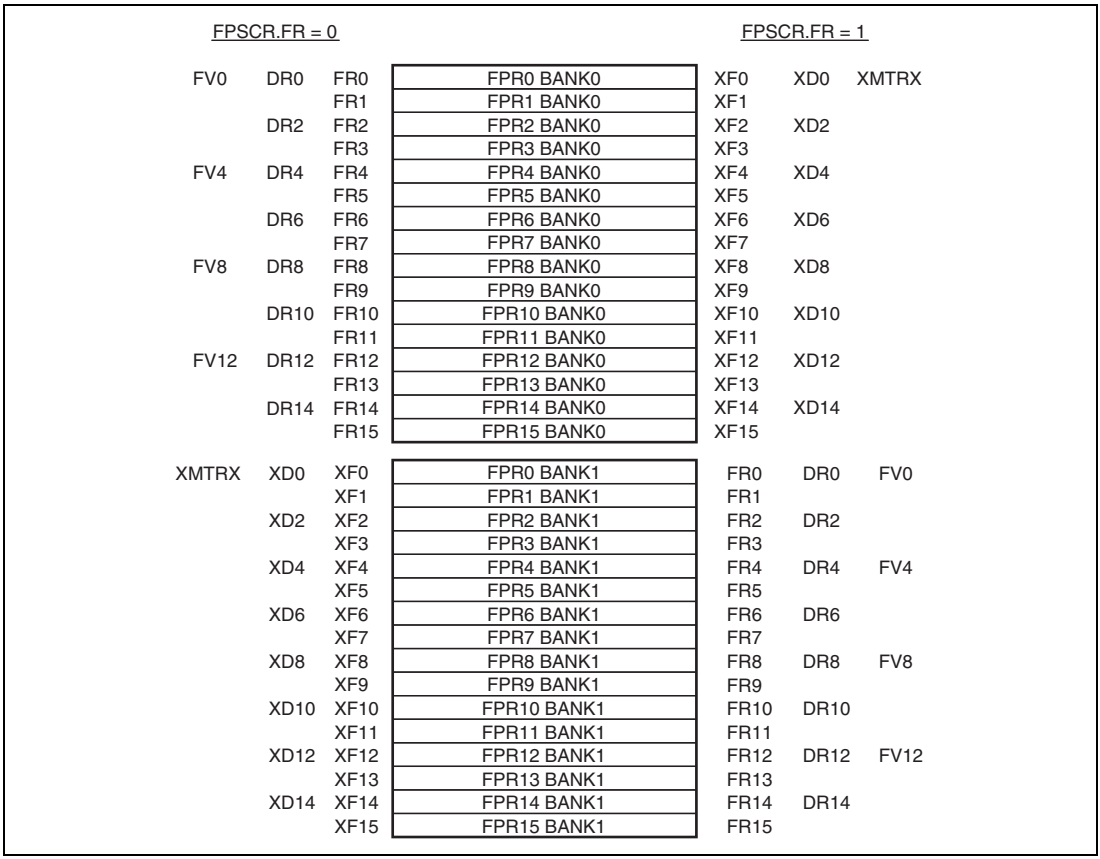


Figure 6.4 Floating-Point Registers

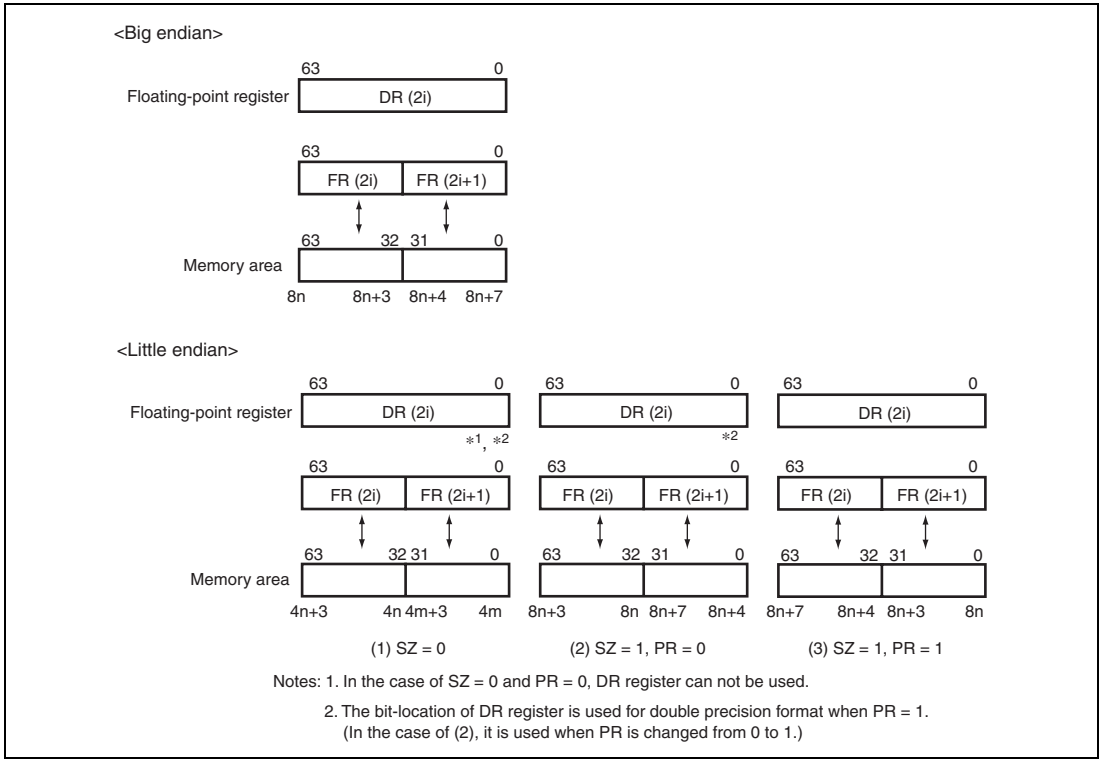
6.3.2 Floating-Point Status/Control Register (FPSCR)

bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	FR	SZ	PR	DN	Cause	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Cause				Enable (EN)					Flag					RM	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	FR	0	R/W	Floating-Point Register Bank 0: FPR0_BANK0 to FPR15_BANK0 are assigned to FR0 to FR15 and FPR0_BANK1 to FPR15_BANK1 are assigned to XF0 to XF15 1: FPR0_BANK0 to FPR15_BANK0 are assigned to XF0 to XF15 and FPR0_BANK1 to FPR15_BANK1 are assigned to FR0 to FR15
20	SZ	0	R/W	Transfer Size Mode 0: Data size of FMOV instruction is 32-bits 1: Data size of FMOV instruction is a 32-bit register pair (64 bits) For relations between endian and the SZ and PR bits, see figure 6.5.
19	PR	0	R/W	Precision Mode 0: Floating-point instructions are executed as single-precision operations 1: Floating-point instructions are executed as double-precision operations (graphics support instructions are undefined) For relations between endian and the SZ and PR bits, see figure 6.5.
18	DN	1	R/W	Denormalization Mode 0: Denormalized number is treated as such 1: Denormalized number is treated as zero

Bit	Bit Name	Initial Value	R/W	Description
17 to 12	Cause	All 0	R/W	FPU Exception Cause Field
11 to 7	Enable	All 0	R/W	FPU Exception Enable Field
6 to 2	Flag	All 0	R/W	FPU Exception Flag Field Each time an FPU operation instruction is executed, the FPU exception cause field is cleared to 0. When an FPU exception occurs, the bits corresponding to FPU exception cause field and flag field are set to 1. The FPU exception flag field remains set to 1 until it is cleared to 0 by software. For bit allocations of each field, see table 6.3.
1	RM1	0	R/W	Rounding Mode
0	RM0	1	R/W	These bits select the rounding mode. 00: Round to Nearest 01: Round to Zero 10: Reserved 11: Reserved



Notes: 1. In the case of SZ = 0 and PR = 0, DR register can not be used.

2. The bit-location of DR register is used for double precision format when PR = 1.
(In the case of (2), it is used when PR is changed from 0 to 1.)

Figure 6.5 Relation between SZ Bit and Endian

Table 6.3 Bit Allocation for FPU Exception Handling

Field Name		FPU Error (E)	Invalid Operation (V)	Division by Zero (Z)	Overflow (O)	Underflow (U)	Inexact (I)
Cause	FPU exception cause field	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12
Enable	FPU exception enable field	None	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7
Flag	FPU exception flag field	None	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2

6.3.3 Floating-Point Communication Register (FPUL)

Information is transferred between the FPU and CPU via FPUL. FPUL is a 32-bit system register that is accessed from the CPU side by means of LDS and STS instructions. For example, to convert the integer stored in general register R1 to a single-precision floating-point number, the processing flow is as follows:

R1 → (LDS instruction) → FPUL → (single-precision FLOAT instruction) → FR1

6.4 Rounding

In a floating-point instruction, rounding is performed when generating the final operation result from the intermediate result. Therefore, the result of combination instructions such as FMAC, FTRV, and FIPR will differ from the result when using a basic instruction such as FADD, FSUB, or FMUL. Rounding is performed once in FMAC, but twice in FADD, FSUB, and FMUL.

Which of the two rounding methods is to be used is determined by the RM bits in FPSCR.

FPSCR.RM[1:0] = 00: Round to Nearest
FPSCR.RM[1:0] = 01: Round to Zero

(1) Round to Nearest

The operation result is rounded to the nearest expressible value. If there are two nearest expressible values, the one with an LSB of 0 is selected.

If the unrounded value is $2^{Emax} (2 - 2^{-P})$ or more, the result will be infinity with the same sign as the unrounded value. The values of Emax and P, respectively, are 127 and 24 for single-precision, and 1023 and 53 for double-precision.

(2) Round to Zero

The digits below the round bit of the unrounded value are discarded.

If the unrounded value is larger than the maximum expressible absolute value, the value will become the maximum expressible absolute value with the same sign as unrounded value.

6.5 Floating-Point Exceptions

6.5.1 General FPU Disable Exceptions and Slot FPU Disable Exceptions

FPU-related exceptions are occurred when an FPU instruction is executed with SR.FD set to 1. When the FPU instruction is in other than delayed slot, the general FPU disable exception is occurred. When the FPU instruction is in the delay slot, the slot FPU disable exception is occurred.

6.5.2 FPU Exception Sources

The exception sources are as follows:

- FPU error (E): When FPSCR.DN = 0 and a denormalized number is input
- Invalid operation (V): In case of an invalid operation, such as NaN input
- Division by zero (Z): Division with a zero divisor
- Overflow (O): When the operation result overflows
- Underflow (U): When the operation result underflows
- Inexact exception (I): When overflow, underflow, or rounding occurs

The FPU exception cause field in FPSCR contains bits corresponding to all of above sources E, V, Z, O, U, and I, and the FPU exception flag and enable fields in FPSCR contain bits corresponding to sources V, Z, O, U, and I, but not E. Thus, FPU errors cannot be disabled.

When an FPU exception occurs, the corresponding bit in the FPU exception cause field is set to 1, and 1 is added to the corresponding bit in the FPU exception flag field. When an FPU exception does not occur, the corresponding bit in the FPU exception cause field is cleared to 0, but the corresponding bit in the FPU exception flag field remains unchanged.

6.5.3 FPU Exception Handling

FPU exception handling is initiated in the following cases:

- FPU error (E): FPSCR.DN = 0 and a denormalized number is input
- Invalid operation (V): FPSCR.Enable.V = 1 and (instruction = FTRV or invalid operation)
- Division by zero (Z): FPSCR.Enable.Z = 1 and division with a zero divisor or the input of FSRRA is zero
- Overflow (O): FPSCR.Enable.O = 1 and possibility of operation result overflow
- Underflow (U): FPSCR.Enable.U = 1 and possibility of operation result underflow
- Inexact exception (I): FPSCR.Enable.I = 1 and instruction with possibility of inexact operation result

Please refer section 11, Instruction Descriptions of the SH-4A Extended Functions Software Manual about the FPU exception case in detail.

All exception events that originate in the FPU are assigned as the same exception event. The meaning of an exception is determined by software by reading from FPSCR and interpreting the information it contains. Also, the destination register is not changed by any FPU exception handling operation.

If the FPU exception sources except for above are generated, the bit corresponding to source V, Z, O, U, or I is set to 1, and a default value is generated as the operation result.

- Invalid operation (V): qNaN is generated as the result.
- Division by zero (Z): Infinity with the same sign as the unrounded value is generated.
- Overflow (O):
  When rounding mode = RZ, the maximum normalized number, with the same sign as the unrounded value, is generated.
  When rounding mode = RN, infinity with the same sign as the unrounded value is generated.
- Underflow (U):
  When FPSCR.DN = 0, a denormalized number with the same sign as the unrounded value, or zero with the same sign as the unrounded value, is generated.
  When FPSCR.DN = 1, zero with the same sign as the unrounded value, is generated.
- Inexact exception (I): An inexact result is generated.

6.6 Graphics Support Functions

This LSI supports two kinds of graphics functions: new instructions for geometric operations, and pair single-precision transfer instructions that enable high-speed data transfer.

6.6.1 Geometric Operation Instructions

Geometric operation instructions perform approximate-value computations. To enable high-speed computation with a minimum of hardware, this LSI ignores comparatively small values in the partial computation results of four multiplications. Consequently, the error shown below is produced in the result of the computation:

$$\text{Maximum error} = \text{MAX (individual multiplication result} \times 2^{-\text{MIN (number of multiplier significant digits}-1,\ \text{number of multiplicand significant digits}-1)}) + \text{MAX (result value} \times 2^{-23},\ 2^{-149})$$

The number of significant digits is 24 for a normalized number and 23 for a denormalized number (number of leading zeros in the fractional part).

In a future version of the SH Series, the above error is guaranteed, but the same result between different processor cores is not guaranteed.

(1) FIPR FVm, FVn (m, n: 0, 4, 8, 12)

This instruction is basically used for the following purposes:

- Inner product ($m \neq n$):
  This operation is generally used for surface/rear surface determination for polygon surfaces.
- Sum of square of elements ($m = n$):
  This operation is generally used to find the length of a vector.

Since an inexact exception is not detected by an FIPR instruction, the inexact exception (I) bit in both the FPU exception cause field and flag field are always set to 1 when an FIPR instruction is executed. Therefore, if the I bit is set in the FPU exception enable field, FPU exception handling will be executed.

(2) FTRV XMTRX, FVn (n: 0, 4, 8, 12)

This instruction is basically used for the following purposes:

- Matrix (4 × 4) · vector (4):
  This operation is generally used for viewpoint changes, angle changes, or movements called vector transformations (4-dimensional). Since affine transformation processing for angle + parallel movement basically requires a 4 × 4 matrix, this LSI supports 4-dimensional operations.
- Matrix (4 × 4) × matrix (4 × 4):
  This operation requires the execution of four FTRV instructions.

Since an inexact exception is not detected by an FIRV instruction, the inexact exception (I) bit in both the FPU exception cause field and flag field are always set to 1 when an FTRV instruction is executed. Therefore, if the I bit is set in the FPU exception enable field, FPU exception handling will be executed. It is not possible to check all data types in the registers beforehand when executing an FTRV instruction. If the V bit is set in the FPU exception enable field, FPU exception handling will be executed.

(3) FRCHG

This instruction modifies banked registers. For example, when the FTRV instruction is executed, matrix elements must be set in an array in the background bank. However, to create the actual elements of a translation matrix, it is easier to use registers in the foreground bank. When the LDS instruction is used on FPSCR, this instruction takes four to five cycles in order to maintain the FPU state. With the FRCHG instruction, the FR bit in FPSCR can be changed in one cycle.

6.6.2 Pair Single-Precision Data Transfer

In addition to the powerful new geometric operation instructions, this LSI also supports high-speed data transfer instructions.

When the SZ bit is 1, this LSI can perform data transfer by means of pair single-precision data transfer instructions.

- FMOV DRm/XDm, DRn/XDRn (m, n: 0, 2, 4, 6, 8, 10, 12, 14)
- FMOV DRm/XDm, @Rn (m: 0, 2, 4, 6, 8, 10, 12, 14; n: 0 to 15)

These instructions enable two single-precision (2 × 32-bit) data items to be transferred; that is, the transfer performance of these instructions is doubled.

- FSCHG

This instruction changes the value of the SZ bit in FPSCR, enabling fast switching between use and non-use of pair single-precision data transfer.

6.7 Notes on Use of FPU Operation Instructions

Care should be taken regarding the following when using FPU operation instructions.

Description

During execution of a double-precision FDIV instruction or double-precision FSQRT instruction (during cycles occupied by the divider), when an exception or interrupt other than an FPU exception prevents execution of a subsequent FPU instruction, and when an FPU operation instruction is executed within a fixed cycle duration following the start of the exception handler routine, the value of CAUSE in FPSCR may be incorrect.

Conditions

The value of CAUSE in FPSCR may be incorrect when all of conditions 1) to 6) below are met:

1) The FPU is set to double-precision mode (FPSCR.PR=1).

2) A double-precision FDIV instruction or double-precision FSQRT instruction is being executed.

   This instruction is referred to below as “instruction 1.”

3) During execution of a double-precision FDIV instruction or double-precision FSQRT instruction (during cycles occupied by the divider), an exception or interrupt other than an FPU exception prevents execution of a subsequent FPU instruction.

   Note that the exceptions referred to here include both completion type exceptions and re-execution type exceptions. In the case in question the SPC points to the FPU operation instruction concerned, or the FPU operation instruction concerned is the next instruction after a branch instruction with delay slot (the next instruction after the delay slot instruction) and the SPC points to the branch instruction two addresses before the FPU operation instruction concerned.

   This unexecuted instruction is referred to below as “instruction 2.”

4) Another FPU operation instruction is executed after the start of the exception handler routine.

   This instruction is referred to as “instruction 3.” “Instruction 3” may be within the exception handler routine or it may be one of a sequence of instructions after returning from the exception handler routine by using RTE.

5) When the number of instructions executed up to instruction 3, counting from the start of the exception handler routine, is within one of the following ranges:

(5-1) Within 26 instructions (including the RTE delay slot) when instruction 3 comes before the return from the exception handler routine by using RTE.

(5-2) Within 13 instructions when instruction 3 comes after the return from the exception handler routine by using RTE and the number of instructions executed by the exception handler routine is odd.

(5-3) Within 14 instructions when instruction 3 comes after the return from the exception handler routine by using RTE and the number of instructions executed by the exception handler routine is even.

6) When the number of instructions executed up to instruction 3, counting from instruction 1, is within one of the following ranges:

(6-1) Within 29 instructions (including the RTE delay slot) when instruction 3 comes before the return from the exception handler routine by using RTE.

(6-2) Within 16 instructions when instruction 3 comes after the return from the exception handler routine by using RTE and the number of instructions executed by the exception handler routine is odd.

(6-3) Within 17 instructions when instruction 3 comes after the return from the exception handler routine by using RTE and the number of instructions executed by the exception handler routine is even.

Note that instruction 2 is not executed, so it is not included in the count of executed instructions.

Note: The FPU operation instructions referred to above (instruction 2 and instruction 3) can be any of the following: FADD, FSUB, FMUL, FMAC, FCMP, FIPR, FTRV, FSCA, FSRRA, FCNVDS, FCNVSD, FTRC, and FLOAT. Note that when the FPCHG instruction does not occur in the interval from instruction 1 to instruction 3 and instructions 2 and 3 are double-precision instructions, instruction 2 and instruction 3 may be double-precision FADD, double-precision FSUB, double-precision FMUL, double-precision FCMP, FCNVDS, FCNVSD, double-precision FTRC, or double-precision FLOAT.

Workarounds

When using the value of the CAUSE bit field, item 1 or 2 below may be used as a workaround.

In the following discussion, the term “FPU operation instruction” refers to any of the following: FADD, FSUB, FMUL, FMAC, FCMP, FIPR, FTRV, FSCA, FSRRA, FCNVDS, FCNVSD, FTRC, and FLOAT. “Exception handler routine” refers to the exception handler routine of an exception (exception or interrupt) other than an FPU exception.

1. Ensure that both 1.1 and 1.2 are satisfied.

   1.1 When an FPU operation instruction is executed within an exception handler routine (from the exception occurrence to the RTE delay slot instruction), ensure that it is the 27th or later instruction from the start of the exception handler routine.

   1.2 When an FPU operation instruction is executed in a sequence of instructions in the original program returned to from an exception handler routine, ensure that 13 or more instructions (counting from the start of the exception handler routine) are executed within the exception handler routine before returning to the original program.

2. When a FPU operation instruction occurs within the range from the start of the exception handler routine specified in conditions 5 and 6 above, execute the LDS, LDS.L, STS or STS.L instruction on FPSCR before executing the FPU operation instruction.

Section 7 Memory Management Unit (MMU)

This LSI supports an 8-bit address space identifier, a 32-bit virtual address space, and a 29-bit or 32-bit physical address space. Address translation from virtual addresses to physical addresses is enabled by the memory management unit (MMU) in this LSI. The MMU performs high-speed address translation by caching user-created address translation table information in an address translation buffer (translation lookaside buffer: TLB).

This LSI has four instruction TLB (ITLB) entries and 64 unified TLB (UTLB) entries. UTLB copies are stored in the ITLB by hardware. A paging system is used for address translation. It is possible to set the virtual address space access right and implement memory protection independently for privileged mode and user mode.

The MMU of this LSI runs in several operating modes. In view of physical address mapping ranges, 29-bit address mode and 32-bit address extended mode are provided. In view of flag functions of the MMU, TLB compatible mode (four paging sizes with four protection bits) and TLB extended mode (eight paging sizes with six protection bits) are provided.

Selection between TLB compatible mode and TLB extended mode is made by setting the relevant control register (bit ME in the MMUCR register) by software.

The flag functions of the MMU are explained in parallel for both TLB compatible mode and TLB extended mode.

7.1 Overview of MMU

The MMU was conceived as a means of making efficient use of physical memory. As shown in (0) in figure 7.1, when a process is smaller in size than the physical memory, the entire process can be mapped onto physical memory, but if the process increases in size to the point where it does not fit into physical memory, it becomes necessary to divide the process into smaller parts, and map the parts requiring execution onto physical memory as occasion arises ((1) in figure 7.1). Having this mapping onto physical memory executed consciously by the process itself imposes a heavy burden on the process. The virtual memory system was devised as a means of handling all physical memory mapping to reduce this burden ((2) in figure 7.1). With a virtual memory system, the size of the available virtual memory is much larger than the actual physical memory, and processes are mapped onto this virtual memory. Thus processes only have to consider their operation in virtual memory, and mapping from virtual memory to physical memory is handled by the MMU. The MMU is normally managed by the OS, and physical memory switching is carried out so as to enable the virtual memory required by a process to be mapped smoothly onto physical memory. Physical memory switching is performed via secondary storage, etc.

The virtual memory system that came into being in this way works to best effect in a time sharing system (TSS) that allows a number of processes to run simultaneously ((3) in figure 7.1). Running a number of processes in a TSS did not increase efficiency since each process had to take account of physical memory mapping. Efficiency is improved and the load on each process reduced by the use of a virtual memory system ((4) in figure 7.1). In this virtual memory system, virtual memory is allocated to each process. The task of the MMU is to map a number of virtual memory areas onto physical memory in an efficient manner. It is also provided with memory protection functions to prevent a process from inadvertently accessing another process's physical memory.

When address translation from virtual memory to physical memory is performed using the MMU, it may happen that the translation information has not been recorded in the MMU, or the virtual memory of a different process is accessed by mistake. In such cases, the MMU will generate an exception, change the physical memory mapping, and record the new address translation information.

Although the functions of the MMU could be implemented by software alone, having address translation performed by software each time a process accessed physical memory would be very inefficient. For this reason, a buffer for address translation (the translation lookaside buffer: TLB) is provided by hardware, and frequently used address translation information is placed here. The TLB can be described as a cache for address translation information. However, unlike a cache, if address translation fails—that is, if an exception occurs—switching of the address translation information is normally performed by software. Thus memory management can be performed in a flexible manner by software.

There are two methods by which the MMU can perform mapping from virtual memory to physical memory: the paging method, using fixed-length address translation, and the segment method, using variable-length address translation. With the paging method, the unit of translation is a fixed-size address space called a page.

In the following descriptions, the address space in virtual memory in this LSI is referred to as virtual address space, and the address space in physical memory as physical address space.

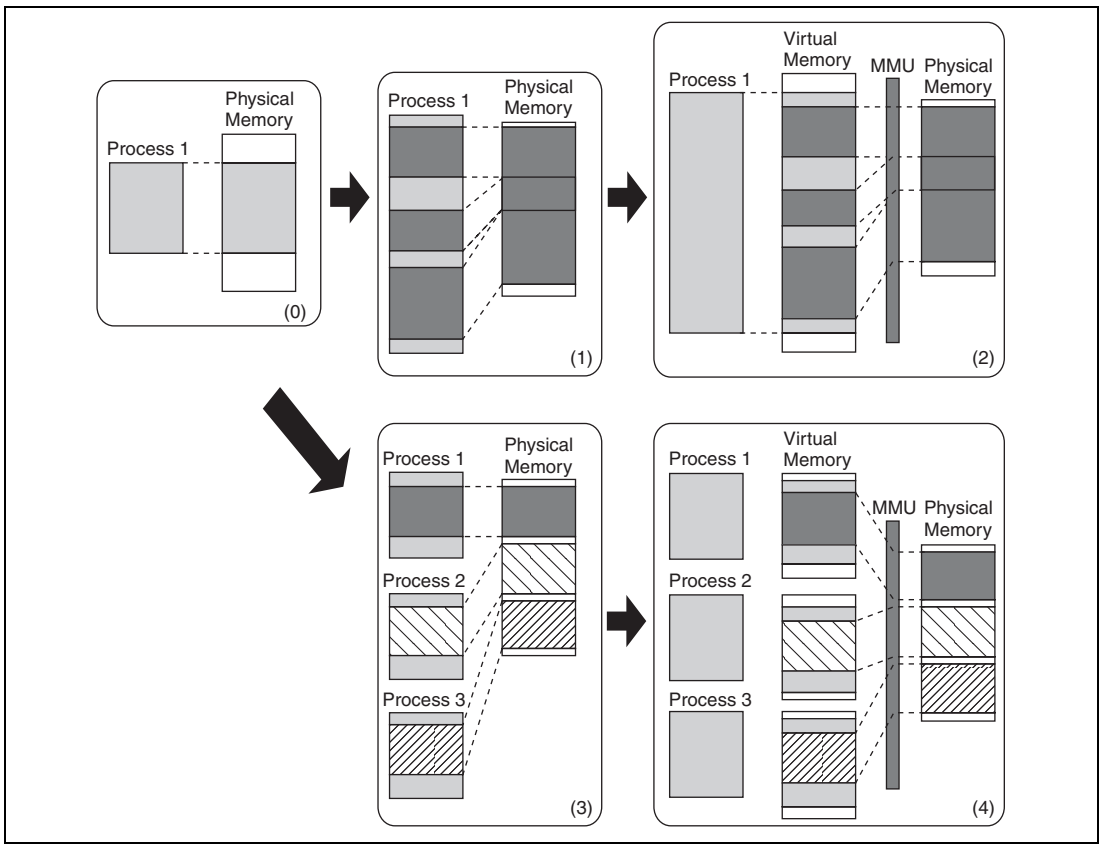


Figure 7.1 Role of MMU

7.1.1 Address Spaces

(1) Virtual Address Space

This LSI supports a 32-bit virtual address space, and can access a 4-Gbyte address space. The virtual address space is divided into a number of areas, as shown in figures 7.2 and 7.3. In privileged mode, the 4-Gbyte space from the P0 area to the P4 area can be accessed. In user mode, a 2-Gbyte space in the U0 area can be accessed. When the SQMD bit in the MMU control register (MMUCR) is 0, a 64-Mbyte space in the store queue area can be accessed. When the RMD bit in the on-chip memory control register (RAMCR) is 1, a 16-Mbyte space in on-chip memory area can be accessed. Accessing areas other than the U0 area, store queue area, and on-chip memory area in user mode will cause an address error.

When the AT bit in MMUCR is set to 1 and the MMU is enabled, the P0, P3, and U0 areas can be mapped onto any physical address space in 1-, 4-, 64-Kbyte, or 1-Mbyte page units in TLB compatible mode and in 1-, 4-, 8-, 64-, 256-Kbyte, 1-, 4-, or 64-Mbyte page units in TLB extended mode. By using an 8-bit address space identifier, the P0, P3, and U0 areas can be increased to a maximum of 256. Mapping from the virtual address space to the 29-bit physical address space is carried out using the TLB.

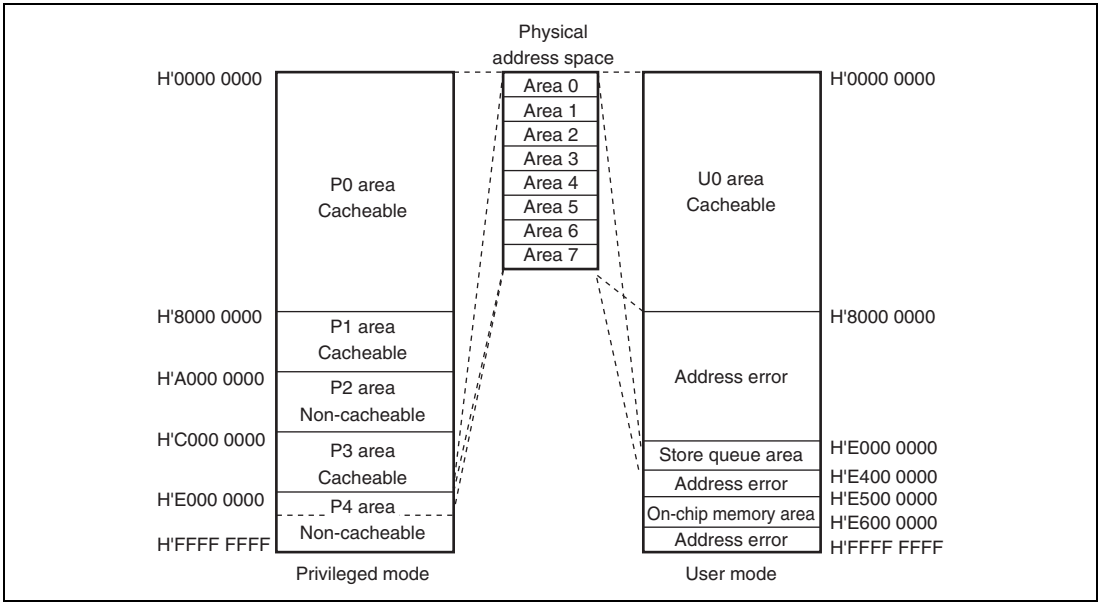


Figure 7.2 Virtual Address Space (AT in MMUCR= 0)

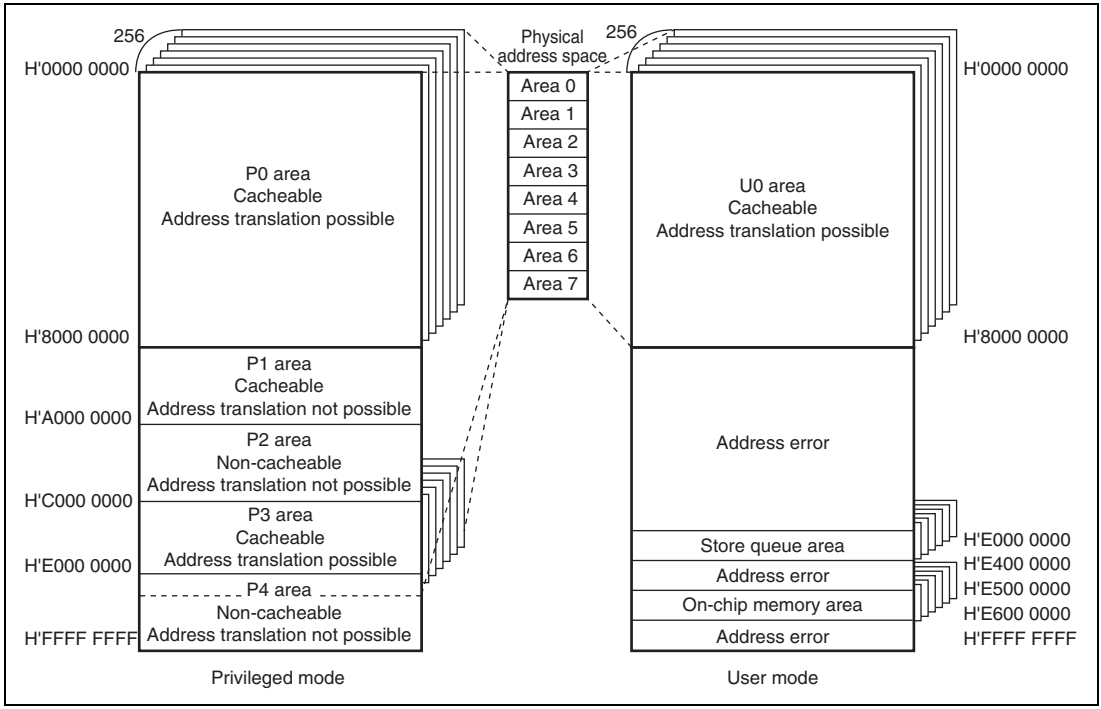


Figure 7.3 Virtual Address Space (AT in MMUCR= 1)

(a) P0, P3, and U0 Areas

The P0, P3, and U0 areas allow address translation using the TLB and access using the cache.

When the MMU is disabled, replacing the upper 3 bits of an address with 0s gives the corresponding physical address. Whether or not the cache is used is determined by the CCR setting. When the cache is used, switching between the copy-back method and the write-through method for write accesses is specified by the WT bit in CCR.

When the MMU is enabled, these areas can be mapped onto any physical address space in 1-, 4-, 64-Kbyte, or 1-Mbyte page units in TLB compatible mode and in 1-, 4-, 8-, 64, 256-Kbyte, 1-, 4-, or 64-Mbyte page units in TLB extended mode using the TLB. When CCR is in the cache enabled state and the C bit for the corresponding page of the TLB entry is 1, accesses can be performed using the cache. When the cache is used, switching between the copy-back method and the write-through method for write accesses is specified by the WT bit of the TLB entry.

When the P0, P3, and U0 areas are mapped onto the control register area which is allocated in the area 7 in physical address space by means of the TLB, the C bit for the corresponding page must be cleared to 0.

(b) P1 Area

The P1 area does not allow address translation using the TLB but can be accessed using the cache.

Regardless of whether the MMU is enabled or disabled, clearing the upper 3 bits of an address to 0 gives the corresponding physical address. Whether or not the cache is used is determined by the CCR setting. When the cache is used, switching between the copy-back method and the write-through method for write accesses is specified by the CB bit in CCR.

(c) P2 Area

The P2 area does not allow address translation using the TLB and access using the cache.

Regardless of whether the MMU is enabled or disabled, clearing the upper 3 bits of an address to 0 gives the corresponding physical address.

(d) P4 Area

The P4 area is mapped onto the internal resource of this LSI. This area except the store queue and on-chip memory areas does not allow address translation using the TLB. This area cannot be accessed using the cache. The P4 area is shown in detail in figure 7.4.

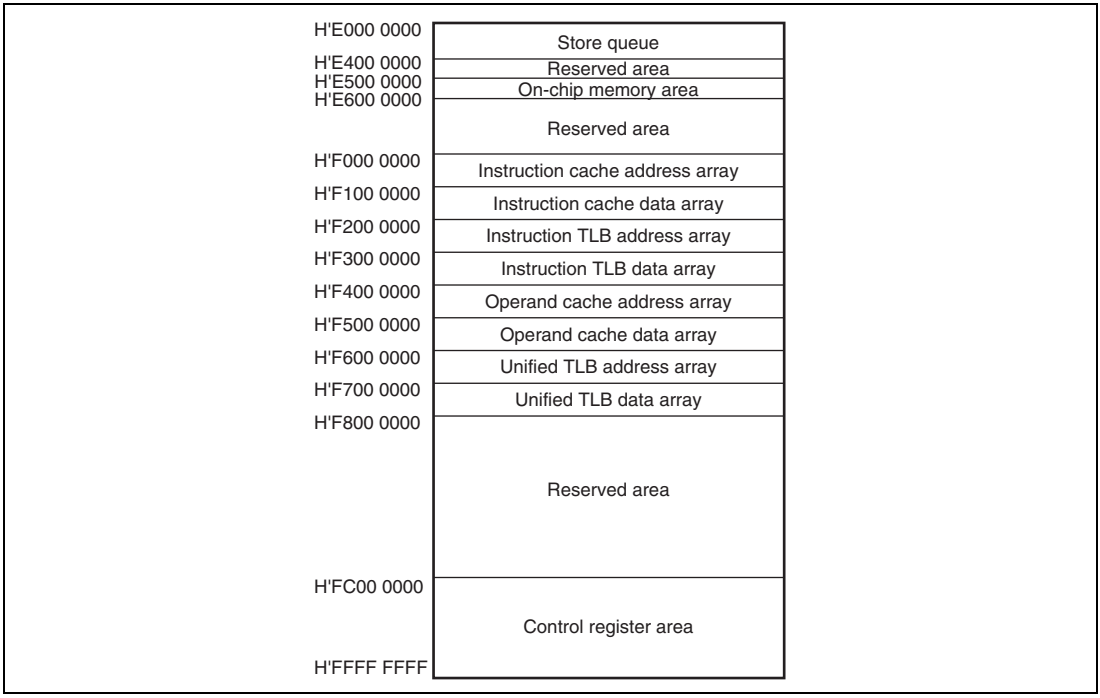


Figure 7.4 P4 Area

The area from H'E000 0000 to H'E3FF FFFF comprises addresses for accessing the store queues (SQs). In user mode, the access right is specified by the SQMD bit in MMUCR. For details, see section 8.7, Store Queues.

The area from H'E500 0000 to H'E5FF FFFF comprises addresses for accessing the on-chip memory. In user mode, the access right is specified by the RMD bit in RAMCR. For details, see section 9, On-Chip Memory.

The area from H'F000 0000 to H'F0FF FFFF is used for direct access to the instruction cache address array. For details, see section 8.6.1, IC Address Array.

The area from H'F100 0000 to H'F1FF FFFF is used for direct access to the instruction cache data array. For details, see section 8.6.2, IC Data Array.

The area from H'F200 0000 to H'F2FF FFFF is used for direct access to the instruction TLB address array. For details, see section 7.7.1, ITLB Address Array.

The area from H'F300 0000 to H'F37F FFFF is used for direct access to instruction TLB data array. For details, see section 7.7.2, ITLB Data Array (TLB Compatible Mode) and section 7.7.3, ITLB Data Array (TLB Extended Mode).

The area from H'F400 0000 to H'F4FF FFFF is used for direct access to the operand cache address array. For details, see section 8.6.3, OC Address Array.

The area from H'F500 0000 to H'F5FF FFFF is used for direct access to the operand cache data array. For details, see section 8.6.4, OC Data Array.

The area from H'F600 0000 to H'F60F FFFF is used for direct access to the unified TLB address array. For details, see section 7.7.4, UTLB Address Array.

The area from H'F700 0000 to H'F70F FFFF is used for direct access to unified TLB data array. For details, see section 7.7.5, UTLB Data Array (TLB Compatible Mode) and section 7.7.6, UTLB Data Array (TLB Extended Mode).

The area from H'FC00 0000 to H'FFFF FFFF is the on-chip peripheral module control register area. For details, see register descriptions in each section.

(2) Physical Address Space

This LSI supports a 29-bit physical address space. The physical address space is divided into eight areas as shown in figure 7.5. Area 7 is a reserved area. For details, see section 11, Bus State Controller (BSC).

Only when area 7 in the physical address space is accessed using the TLB, addresses H'1C00 0000 to H'1FFF FFFF of area 7 are not designated as a reserved area, but are equivalent to the control register area in the P4 area, in the virtual address space.

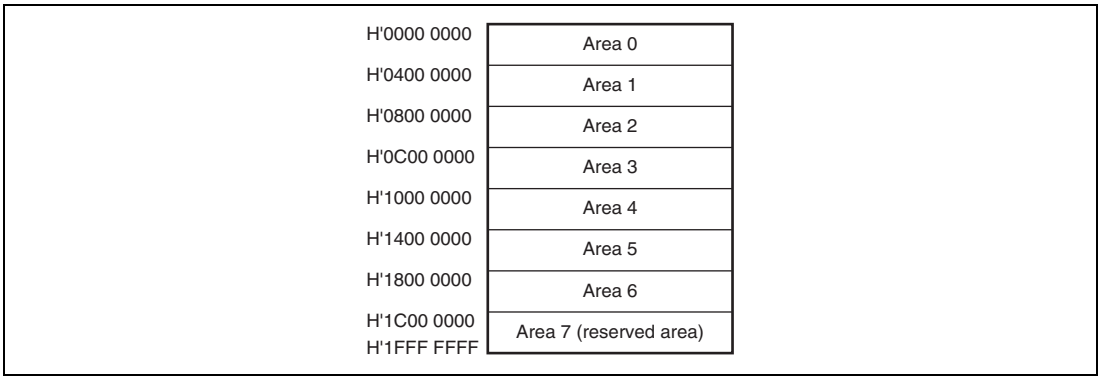


Figure 7.5 Physical Address Space

(3) Address Translation

When the MMU is used, the virtual address space is divided into units called pages, and translation to physical addresses is carried out in these page units. The address translation table in external memory contains the physical addresses corresponding to virtual addresses and additional information such as memory protection codes. Fast address translation is achieved by caching the contents of the address translation table located in external memory into the TLB. In this LSI basically, the ITLB is used for instruction accesses and the UTLB for data accesses. In the event of an access to an area other than the P4 area, the accessed virtual address is translated to a physical address. If the virtual address belongs to the P1 or P2 area, the physical address is uniquely determined without accessing the TLB. If the virtual address belongs to the P0, U0, or P3 area, the TLB is searched using the virtual address, and if the virtual address is recorded in the TLB, a TLB hit is made and the corresponding physical address is read from the TLB. If the accessed virtual address is not recorded in the TLB, a TLB miss exception is generated and processing switches to the TLB miss exception handling routine. In the TLB miss exception handling routine, the address translation table in external memory is searched, and the corresponding physical address and page management information are recorded in the TLB. After the return from the exception handling routine, the instruction which caused the TLB miss exception is re-executed.

(4) Single Virtual Memory Mode and Multiple Virtual Memory Mode

There are two virtual memory systems, single virtual memory and multiple virtual memory, either of which can be selected with the SV bit in MMUCR. In the single virtual memory system, a number of processes run simultaneously, using virtual address space on an exclusive basis, and the physical address corresponding to a particular virtual address is uniquely determined. In the multiple virtual memory system, a number of processes run while sharing the virtual address space, and particular virtual addresses may be translated into different physical addresses depending on the process. The only difference between the single virtual memory and multiple virtual memory systems in terms of operation is in the TLB address comparison method (see section 7.3.3, Address Translation Method).

(5) Address Space Identifier (ASID)

In multiple virtual memory mode, an 8-bit address space identifier (ASID) is used to distinguish between multiple processes running simultaneously while sharing the virtual address space. Software can set the 8-bit ASID of the currently executing process in PTEH in the MMU. The TLB does not have to be purged when processes are switched by means of ASID.

In single virtual memory mode, ASID is used to provide memory protection for multiple processes running simultaneously while using the virtual address space on an exclusive basis.

Note: Two or more entries with the same virtual page number (VPN) but different ASID must not be set in the TLB simultaneously in single virtual memory mode.

7.2 Register Descriptions

The following registers are related to MMU processing.

Table 7.1 Register Configuration

Register Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Size
Page table entry high register	PTEH	R/W	H'FF00 0000	H'1F00 0000	32
Page table entry low register	PTEL	R/W	H'FF00 0004	H'1F00 0004	32
Translation table base register	TTB	R/W	H'FF00 0008	H'1F00 0008	32
TLB exception address register	TEA	R/W	H'FF00 000C	H'1F00 000C	32
MMU control register	MMUCR	R/W	H'FF00 0010	H'1F00 0010	32
Page table entry assistance register	PTEA	R/W	H'FF00 0034	H'1F00 0034	32
Physical address space control register	PASCR	R/W	H'FF00 0070	H'1F00 0070	32
Instruction re-fetch inhibit control register	IRMCR	R/W	H'FF00 0078	H'1F00 0078	32

Note: * These P4 addresses are for the P4 area in the virtual address space. These area 7 addresses are accessed from area 7 in the physical address space by means of the TLB.

Table 7.2 Register States in Each Processing State

Register Name	Abbreviation	Power-on Reset	Manual Reset	Sleep	Standby
Page table entry high register	PTEH	Undefined	Undefined	Retained	Retained
Page table entry low register	PTEL	Undefined	Undefined	Retained	Retained
Translation table base register	TTB	Undefined	Undefined	Retained	Retained
TLB exception address register	TEA	Undefined	Retained	Retained	Retained
MMU control register	MMUCR	H'0000 0000	H'0000 0000	Retained	Retained
Page table entry assistance register	PTEA	H'0000 xxx0	H'0000 xxx0	Retained	Retained
Physical address space control register	PASCR	H'0000 0000	H'0000 xxx0	Retained	Retained

Register Name	Abbreviation	Power-on Reset	Manual Reset	Sleep	Standby
Instruction re-fetch inhibit control register	IRMCR	H'0000 0000	H'0000 0000	Retained	Retained

7.2.1 Page Table Entry High Register (PTEH)

PTEH consists of the virtual page number (VPN) and address space identifier (ASID). When an MMU exception or address error exception occurs, the VPN of the virtual address at which the exception occurred is set in the VPN bit by hardware. VPN varies according to the page size, but the VPN set by hardware when an exception occurs consists of the upper 22 bits of the virtual address which caused the exception. VPN setting can also be carried out by software. The number of the currently executing process is set in the ASID bit by software. ASID is not updated by hardware. VPN and ASID are recorded in the UTLB by means of the LDTLB instruction.

After the ASID field in PTEH has been updated, execute one of the following three methods before an access (including an instruction fetch) to the P0, P3, or U0 area that uses the updated ASID value is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the P0, P3, or U0 area.
2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the R2 bit in IRMCR is 0 (initial value) before updating the ASID field, the specific instruction does not need to be executed. However, note that the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction after the ASID field has been updated.

Note that the method 3 may not be guaranteed in the future SuperH Series. Therefore, it is recommended that the method 1 or 2 should be used for being compatible with the future SuperH Series.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VPN															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VPN						—	—	ASID							
Initial value:	—	—	—	—	—	—	0	0	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	VPN	Undefined	R/W	Virtual Page Number
9, 8	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
7 to 0	ASID	Undefined	R/W	Address Space Identifier

7.2.2 Page Table Entry Low Register (PTEL)

PTEL is used to hold the physical page number and page management information to be recorded in the UTLB by means of the LDTLB instruction. The contents of this register are not changed unless a software directive is issued.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	PPN												
Initial value:	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PPN						—	V	SZ1	PR1	PR0	SZ0	C	D	SH	WT
Initial value:	—	—	—	—	—	—	0	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
28 to 10	PPN	Undefined	R/W	Physical Page Number
9	—	0	R	Reserved For details on reading from or writing to this bit, see description in General Precautions on Handling of Product.

Bit	Bit Name	Initial Value	R/W	Description
8	V	Undefined	R/W	Page Management Information
7	SZ1	Undefined	R/W	The meaning of each bit is same as that of corresponding bit in Common TLB (UTLB).
6	PR1	Undefined	R/W	
5	PR0	Undefined	R/W	For details, see section 7.3, TLB Functions (TLB Compatible Mode; MMUCR.ME = 0) and section 7.4, TLB Functions (TLB Extended Mode; MMUCR.ME = 1).
4	SZ0	Undefined	R/W	
3	C	Undefined	R/W	Note: SZ1, PR1, SZ0, and PR0 bits are valid only in TLB compatible mode.
2	D	Undefined	R/W	
1	SH	Undefined	R/W	
0	WT	Undefined	R/W	

7.2.3 Translation Table Base Register (TTB)

TTB is used to store the base address of the currently used page table, and so on. The contents of TTB are not changed unless a software directive is issued. This register can be used freely by software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TTB															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTB															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.2.4 TLB Exception Address Register (TEA)

After an MMU exception or address error exception occurs, the virtual address at which the exception occurred is stored. The contents of this register can be changed by software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEA															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEA															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit: 31–0 — TEA: Virtual address at which MMU exception or address error occurred

7.2.5 MMU Control Register (MMUCR)

The individual bits perform MMU settings as shown below. Therefore, MMUCR rewriting should be performed by a program in the P1 or P2 area.

After MMUCR has been updated, execute one of the following three methods before an access (including an instruction fetch) to the P0, P3, U0, or store queue area is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the P0, P3, or U0 area.
2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the R2 bit in IRMCR is 0 (initial value) before updating MMUCR, the specific instruction does not need to be executed. However, note that the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction after MMUCR has been updated.

Note that the method 3 may not be guaranteed in the future SuperH Series. Therefore, it is recommended that the method 1 or 2 should be used for being compatible with the future SuperH Series.

MMUCR contents can be changed by software. However, the LRUI and URC bits may also be updated by hardware.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LRUI						—	—	URB						—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

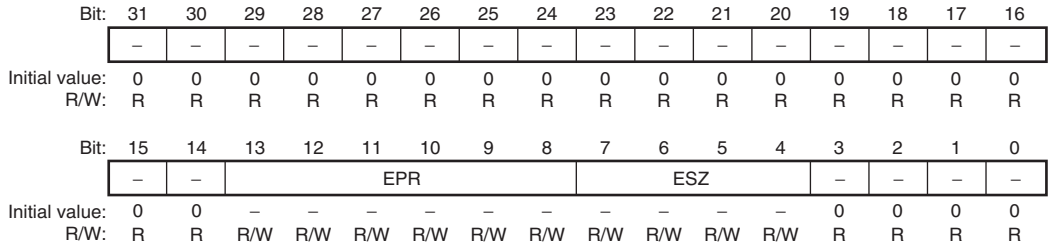
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	URC						SQMD	SV	ME	—	—	—	—	TI	—	AT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	LRUI	000000	R/W	Least Recently Used ITLB These bits indicate the ITLB entry to be replaced. The LRU (least recently used) method is used to decide the ITLB entry to be replaced in the event of an ITLB miss. The entry to be purged from the ITLB can be confirmed using the LRUI bits. LRUI is updated by means of the algorithm shown below. x means that updating is not performed. 000xxx: ITLB entry 0 is used 1xx00x: ITLB entry 1 is used x1x1x0: ITLB entry 2 is used xx1x11: ITLB entry 3 is used xxxxxx: Other than above When the LRUI bit settings are as shown below, the corresponding ITLB entry is updated by an ITLB miss. Ensure that values for which "Setting prohibited" is indicated below are not set at the discretion of software. After a power-on or manual reset, the LRUI bits are initialized to 0, and therefore a prohibited setting is never made by a hardware update. x means "don't care". 111xxx: ITLB entry 0 is updated 0xx11x: ITLB entry 1 is updated x0x0x1: ITLB entry 2 is updated xx0x00: ITLB entry 3 is updated Other than above: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
25, 24	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
23 to 18	URB	000000	R/W	UTLB Replace Boundary These bits indicate the UTLB entry boundary at which replacement is to be performed. Valid only when URB ≠ 0.
17, 16	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
15 to 10	URC	000000	R/W	UTLB Replace Counter These bits serve as a random counter for indicating the UTLB entry for which replacement is to be performed with an LDTLB instruction. This bit is incremented each time the UTLB is accessed. If URB > 0, URC is cleared to 0 when the condition URC = URB is satisfied. Also note that if a value is written to URC by software which results in the condition of URC > URB, incrementing is first performed in excess of URB until URC = H'3F. URC is not incremented by an LDTLB instruction.
9	SQMD	0	R/W	Store Queue Mode Specifies the right of access to the store queues. 0: User/privileged access possible 1: Privileged access possible (address error exception in case of user access)
8	SV	0	R/W	Single Virtual Memory Mode/Multiple Virtual Memory Mode Switching When this bit is changed, ensure that 1 is also written to the TI bit. 0: Multiple virtual memory mode 1: Single virtual memory mode

Bit	Bit Name	Initial Value	R/W	Description
7	ME	0	R/W	TLB Extended Mode Switching 0: TLB compatible mode 1: TLB extended mode For modifying the ME bit value, always set the TI bit to 1 to invalidate the contents of ITLB and UTLB.
6 to 3	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
2	TI	0	R/W	TLB Invalidate Bit Writing 1 to this bit invalidates (clears to 0) all valid UTLB/ITLB bits. This bit is always read as 0.
1	—	0	R	Reserved For details on reading from or writing to this bit, see description in General Precautions on Handling of Product.
0	AT	0	R/W	Address Translation Enable Bit These bits enable or disable the MMU. 0: MMU disabled 1: MMU enabled MMU exceptions are not generated when the AT bit is 0. In the case of software that does not use the MMU, the AT bit should be cleared to 0.

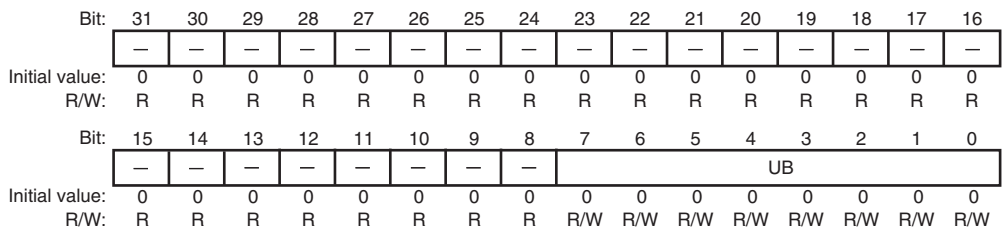
7.2.6 Page Table Entry Assistance Register (PTEA)



Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved For details on reading/writing these bits, see General Precautions on Handling of Product.
13 to 8	EPR	Undefined	R/W	Page Control Information
7 to 4	ESZ	Undefined	R/W	Each bit has the same function as the corresponding bit of the unified TLB (UTLB). For details, see section 7.4, TLB Functions (TLB Extended Mode; MMUCR.ME = 1)
3 to 0	—	All 0	R	Reserved For details on reading/writing these bits, see General Precautions on Handling of Product.

7.2.7 Physical Address Space Control Register (PASCR)

PASCR controls the operation in the physical address space.



Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
7 to 0	UB	H'00	R/W	Buffered Write Control for Each Area (64 Mbytes) When writing is performed without using the cache or in the cache write-through mode, these bits specify whether the next bus access from the CPU waits for the end of writing for each area. 0 : Buffered write (The CPU does not wait for the end of writing bus access and starts the next bus access) 1 : Unbuffered write (The CPU waits for the end of writing bus access and starts the next bus access) UB[7]: Corresponding to the control register area UB[6]: Corresponding to area 6 UB[5]: Corresponding to area 5 UB[4]: Corresponding to area 4 UB[3]: Corresponding to area 3 UB[2]: Corresponding to area 2 UB[1]: Corresponding to area 1 UB[0]: Corresponding to area 0

7.2.8 Instruction Re-Fetch Inhibit Control Register (IRMCR)

When the specific resource is changed, IRMCR controls whether the instruction fetch is performed again for the next instruction. The specific resource means the part of control registers, TLB, and cache.

In the initial state, the instruction fetch is performed again for the next instruction after changing the resource. However, the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction every time the resource is changed. Therefore, it is recommended that each bit in IRMCR is set to 1 and the specific instruction should be executed after all necessary resources have been changed prior to execution of the program which uses changed resources.

For details on the specific sequence, see descriptions in each resource.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	R2	R1	LT	MT	MC
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
4	R2	0	R/W	Re-Fetch Inhibit 2 after Register Change When MMUCR, PASCR, CCR, PTEH, or RAMCR is changed, this bit controls whether re-fetch is performed for the next instruction. 0: Re-fetch is performed 1: Re-fetch is not performed

Bit	Bit Name	Initial Value	R/W	Description
3	R1	0	R/W	Re-Fetch Inhibit 1 after Register Change When a register allocated in addresses H'FF200000 to H'FF2FFFFF is changed, this bit controls whether re-fetch is performed for the next instruction. 0: Re-fetch is performed 1: Re-fetch is not performed
2	LT	0	R/W	Re-Fetch Inhibit after LDTLB Execution This bit controls whether re-fetch is performed for the next instruction after the LDTLB instruction has been executed. 0: Re-fetch is performed 1: Re-fetch is not performed
1	MT	0	R/W	Re-Fetch Inhibit after Writing Memory-Mapped TLB This bit controls whether re-fetch is performed for the next instruction after writing memory-mapped ITLB/UTLB while the AT bit in MMUCR is set to 1. 0: Re-fetch is performed 1: Re-fetch is not performed
0	MC	0	R/W	Re-Fetch Inhibit after Writing Memory-Mapped IC This bit controls whether re-fetch is performed for the next instruction after writing memory-mapped IC while the ICE bit in CCR is set to 1. 0: Re-fetch is performed 1: Re-fetch is not performed

7.3 TLB Functions (TLB Compatible Mode; MMUCR.ME = 0)

7.3.1 Unified TLB (UTLB) Configuration

The UTLB is used for the following two purposes:

1. To translate a virtual address to a physical address in a data access
2. As a table of address translation information to be recorded in the ITLB in the event of an ITLB miss

The UTLB is so called because of its use for the above two purposes. Information in the address translation table located in external memory is cached into the UTLB. The address translation table contains virtual page numbers and address space identifiers, and corresponding physical page numbers and page management information. Figure 7.6 shows the UTLB configuration. The UTLB consists of 64 fully-associative type entries. Figure 7.7 shows the relationship between the page size and address format.

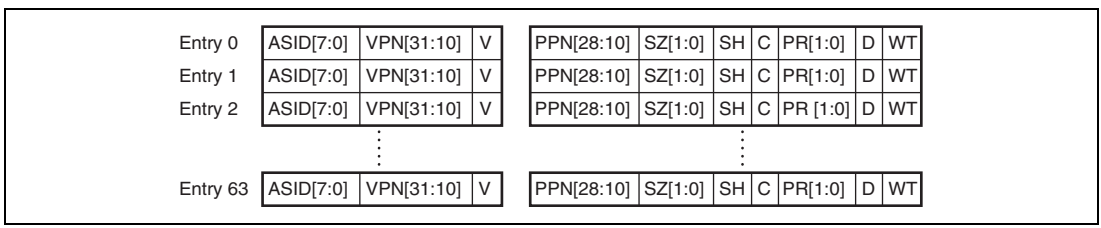


Figure 7.6 UTLB Configuration (TLB Compatible Mode)

[Legend]

- VPN: Virtual page number
  For 1-Kbyte page: Upper 22 bits of virtual address
  For 4-Kbyte page: Upper 20 bits of virtual address
  For 64-Kbyte page: Upper 16 bits of virtual address
  For 1-Mbyte page: Upper 12 bits of virtual address

- ASID: Address space identifier
  Indicates the process that can access a virtual page.
  In single virtual memory mode and user mode, or in multiple virtual memory mode, if the SH bit is 0, this identifier is compared with the ASID in PTEH when address comparison is performed.

- SH: Share status bit
  When 0, pages are not shared by processes.
  When 1, pages are shared by processes.

- SZ[1:0]: Page size bits
  Specify the page size.
  00: 1-Kbyte page
  01: 4-Kbyte page
  10: 64-Kbyte page
  11: 1-Mbyte page

- V: Validity bit
  Indicates whether the entry is valid.
  0: Invalid
  1: Valid
  Cleared to 0 by a power-on reset.
  Not affected by a manual reset.

- PPN: Physical page number
  Upper 22 bits of the physical address of the physical page number.
  With a 1-Kbyte page, PPN[28:10] are valid.
  With a 4-Kbyte page, PPN[28:12] are valid.
  With a 64-Kbyte page, PPN[28:16] are valid.
  With a 1-Mbyte page, PPN[28:20] are valid.
  The synonym problem must be taken into account when setting the PPN (see section 7.5.5, Avoiding Synonym Problems).

- PR[1:0]: Protection key data
  2-bit data expressing the page access right as a code.
  00: Can be read from only in privileged mode
  01: Can be read from and written to in privileged mode
  10: Can be read from only in privileged or user mode
  11: Can be read from and written to in privileged mode or user mode

- C: Cacheability bit
  Indicates whether a page is cacheable.
  0: Not cacheable

1: Cacheable

When the control register area is mapped, this bit must be cleared to 0.

- D: Dirty bit

  Indicates whether a write has been performed to a page.

  0: Write has not been performed

  1: Write has been performed

- WT: Write-through bit

  Specifies the cache write mode.

  0: Copy-back mode

  1: Write-through mode

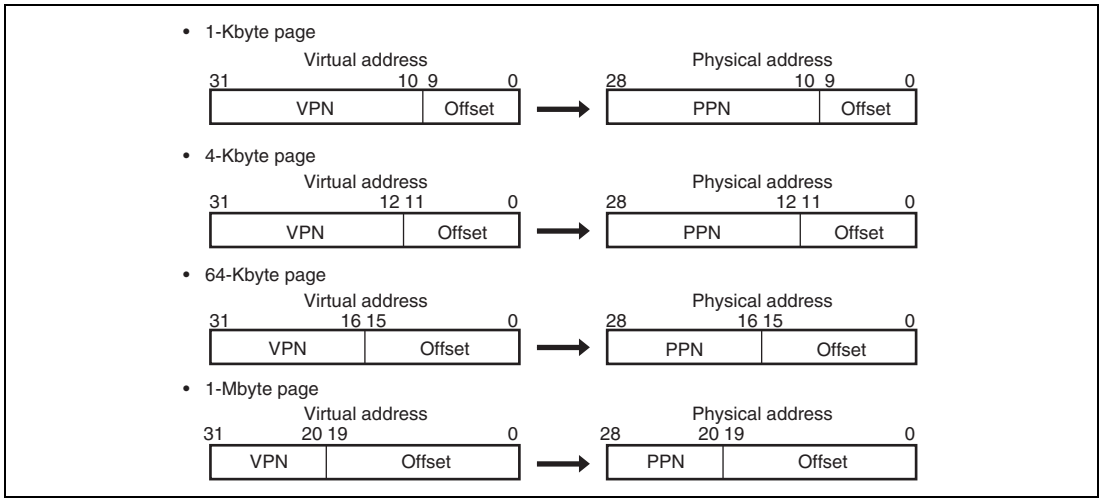


Figure 7.7 Relationship between Page Size and Address Format (TLB Compatible Mode)

7.3.2 Instruction TLB (ITLB) Configuration

The ITLB is used to translate a virtual address to a physical address in an instruction access. Information in the address translation table located in the UTLB is cached into the ITLB. Figure 7.8 shows the ITLB configuration. The ITLB consists of four fully-associative type entries.

Entry 0	ASID[7:0]	VPN[31:10]	V	PPN[28:10]	SZ[1:0]	SH	C	PR
Entry 1	ASID[7:0]	VPN[31:10]	V	PPN[28:10]	SZ[1:0]	SH	C	PR
Entry 2	ASID[7:0]	VPN[31:10]	V	PPN[28:10]	SZ[1:0]	SH	C	PR
Entry 3	ASID[7:0]	VPN[31:10]	V	PPN[28:10]	SZ[1:0]	SH	C	PR

Notes: 1. The D and WT bits are not supported.
2. There is only one PR bit, corresponding to the upper bit of the PR bits in the UTLB.

Figure 7.8 ITLB Configuration (TLB Compatible Mode)

7.3.3 Address Translation Method

Figure 7.9 shows a flowchart of a memory access using the UTLB.

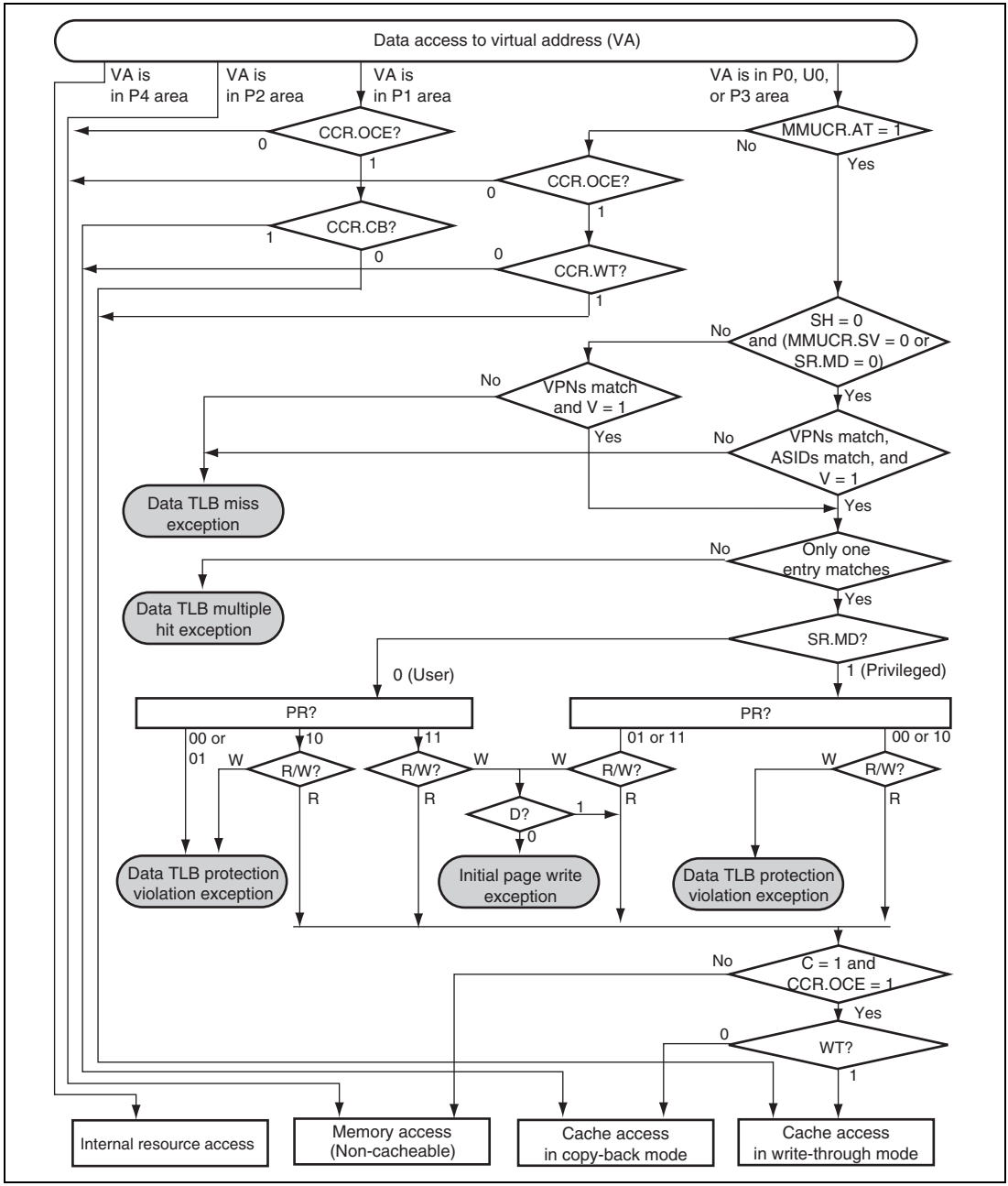


Figure 7.9 Flowchart of Memory Access Using UTLB (TLB Compatible Mode)

Figure 7.10 shows a flowchart of a memory access using the ITLB.

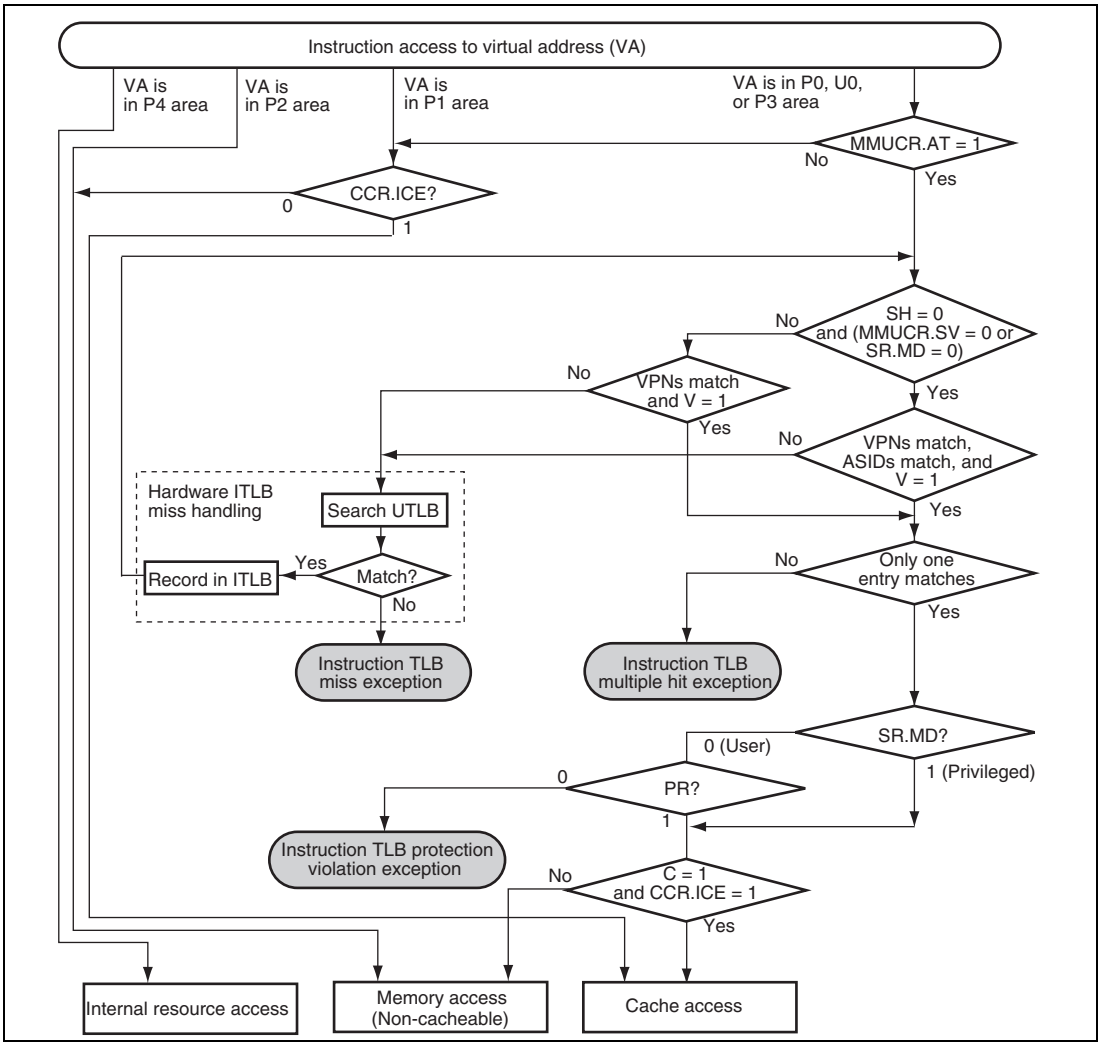


Figure 7.10 Flowchart of Memory Access Using ITLB (TLB Compatible Mode)

7.4 TLB Functions (TLB Extended Mode; MMUCR.ME = 1)

7.4.1 Unified TLB (UTLB) Configuration

Figure 7.11 shows the configuration of the UTLB in TLB extended mode. Figure 7.12 shows the relationship between the page size and address format.

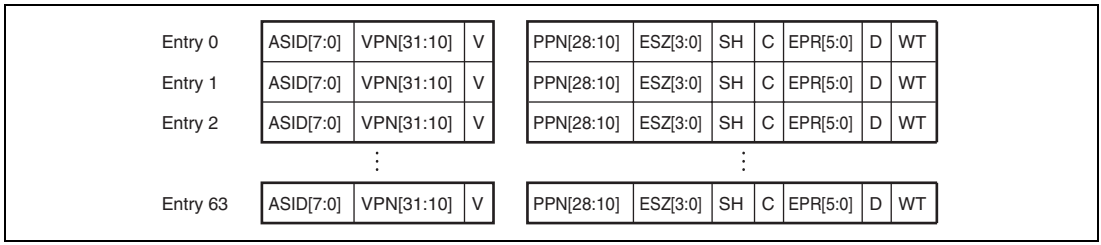


Figure 7.11 UTLB Configuration (TLB Extended Mode)

[Legend]

- VPN: Virtual page number
  For 1-Kbyte page: Upper 22 bits of virtual address
  For 4-Kbyte page: Upper 20 bits of virtual address
  For 8-Kbyte page: Upper 19 bits of virtual address
  For 64-Kbyte page: Upper 16 bits of virtual address
  For 256-Kbyte page: Upper 14 bits of virtual address
  For 1-Mbyte page: Upper 12 bits of virtual address
  For 4-Mbyte page: Upper 10 bits of virtual address
  For 64-Mbyte page: Upper 6 bits of virtual address
- ASID: Address space identifier
  Indicates the process that can access a virtual page.
  In single virtual memory mode and user mode, or in multiple virtual memory mode, if the SH bit is 0, this identifier is compared with the ASID in PTEH when address comparison is performed.
- SH: Share status bit
  When 0, pages are not shared by processes.
  When 1, pages are shared by processes.
- ESZ: Page size bits
  Specify the page size.

0000: 1-Kbyte page
0001: 4-Kbyte page
0010: 8-Kbyte page
0100: 64-Kbyte page
0101: 256-Kbyte page
0111: 1-Mbyte page
1000: 4-Mbyte page
1100: 64-Mbyte page

Note: When a value other than those listed above is recorded, operation is not guaranteed.

- V: Validity bit
  Indicates whether the entry is valid.
  0: Invalid
  1: Valid
  Cleared to 0 by a power-on reset.
  Not affected by a manual reset.
- PPN: Physical page number
  Upper 19 bits of the physical address.
  With a 1-Kbyte page, PPN[28:10] are valid.
  With a 4-Kbyte page, PPN[28:12] are valid.
  With an 8-Kbyte page, PPN[28:13] are valid.
  With a 64-Kbyte page, PPN[28:16] are valid.
  With a 256-Kbyte page, PPN[28:18] are valid.
  With a 1-Mbyte page, PPN[28:20] are valid.
  With a 4-Mbyte page, PPN[28:22] are valid.
  With a 64-Mbyte page, PPN[28:26] are valid.
  The synonym problem must be taken into account when setting the PPN (see section 7.5.5, Avoiding Synonym Problems).
- EPR: Protection key data
  6-bit data expressing the page access right as a code.
  Reading, writing, and execution (instruction fetch) in privileged mode and reading, writing, and execution (instruction fetch) in user mode can be set independently. Each bit is disabled by 0 and enabled by 1.
  EPR[5]: Reading in privileged mode
  EPR[4]: Writing in privileged mode
  EPR[3]: Execution in privileged mode (instruction fetch)

EPR[2]: Reading in user mode

EPR[1]: Writing in user mode

EPR[0]: Execution in user mode (instruction fetch)

- C: Cacheability bit

  Indicates whether a page is cacheable.

  0: Not cacheable

  1: Cacheable

  When the control register area is mapped, this bit must be cleared to 0.

- D: Dirty bit

  Indicates whether a write has been performed to a page.

  0: Write has not been performed.

  1: Write has been performed.

- WT: Write-through bit

  Specifies the cache write mode.

  0: Copy-back mode

  1: Write-through mode

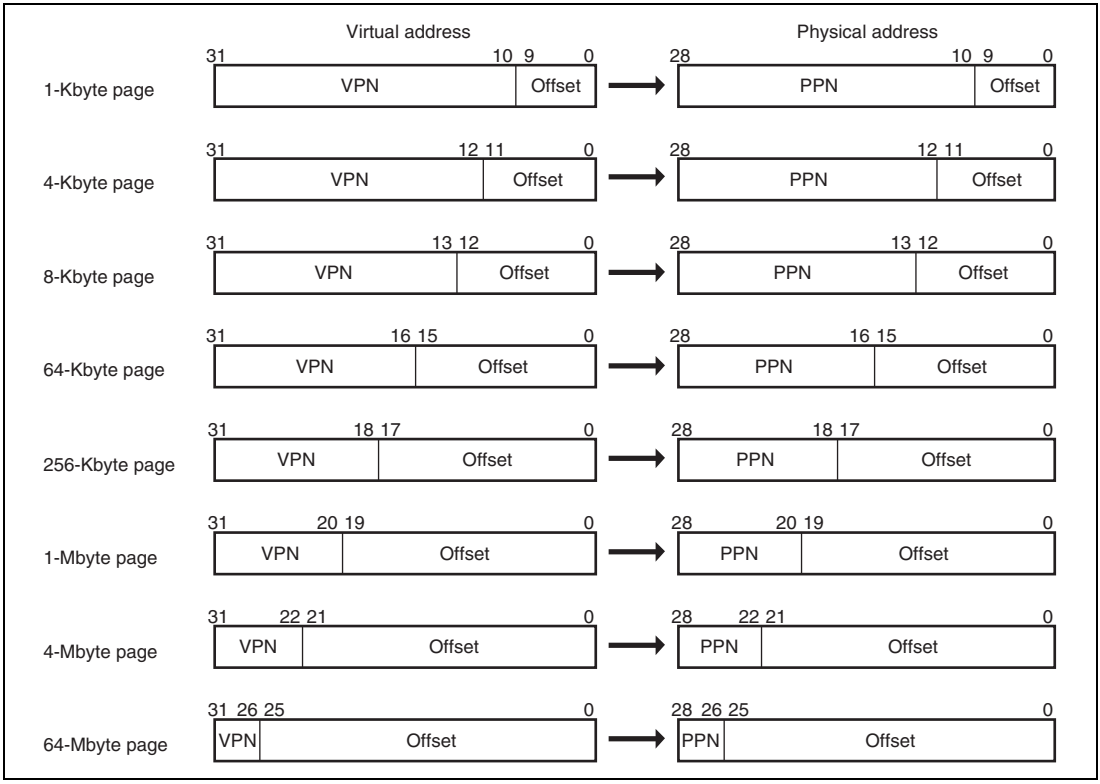


Figure 7.12 Relationship between Page Size and Address Format (TLB Extended Mode)

7.4.2 Instruction TLB (ITLB) Configuration

Figure 7.13 shows the configuration of the ITLB in TLB extended mode.

Entry 0	ASID[7:0]	VPN[31:10]	V	PPN[28:10]	ESZ[3:0]	SH	C	EPR[5]	EPR[3]	EPR[2]	EPR[0]
Entry 1	ASID[7:0]	VPN[31:10]	V	PPN[28:10]	ESZ[3:0]	SH	C	EPR[5]	EPR[3]	EPR[2]	EPR[0]
Entry 2	ASID[7:0]	VPN[31:10]	V	PPN[28:10]	ESZ[3:0]	SH	C	EPR[5]	EPR[3]	EPR[2]	EPR[0]
Entry 3	ASID[7:0]	VPN[31:10]	V	PPN[28:10]	ESZ[3:0]	SH	C	EPR[5]	EPR[3]	EPR[2]	EPR[0]

Note: Bits EPR[4], EPR[1], D, and WT are not supported.

Figure 7.13 ITLB Configuration (TLB Extended Mode)

7.4.3 Address Translation Method

Figure 7.14 is a flowchart of memory access using the UTLB in TLB extended mode.

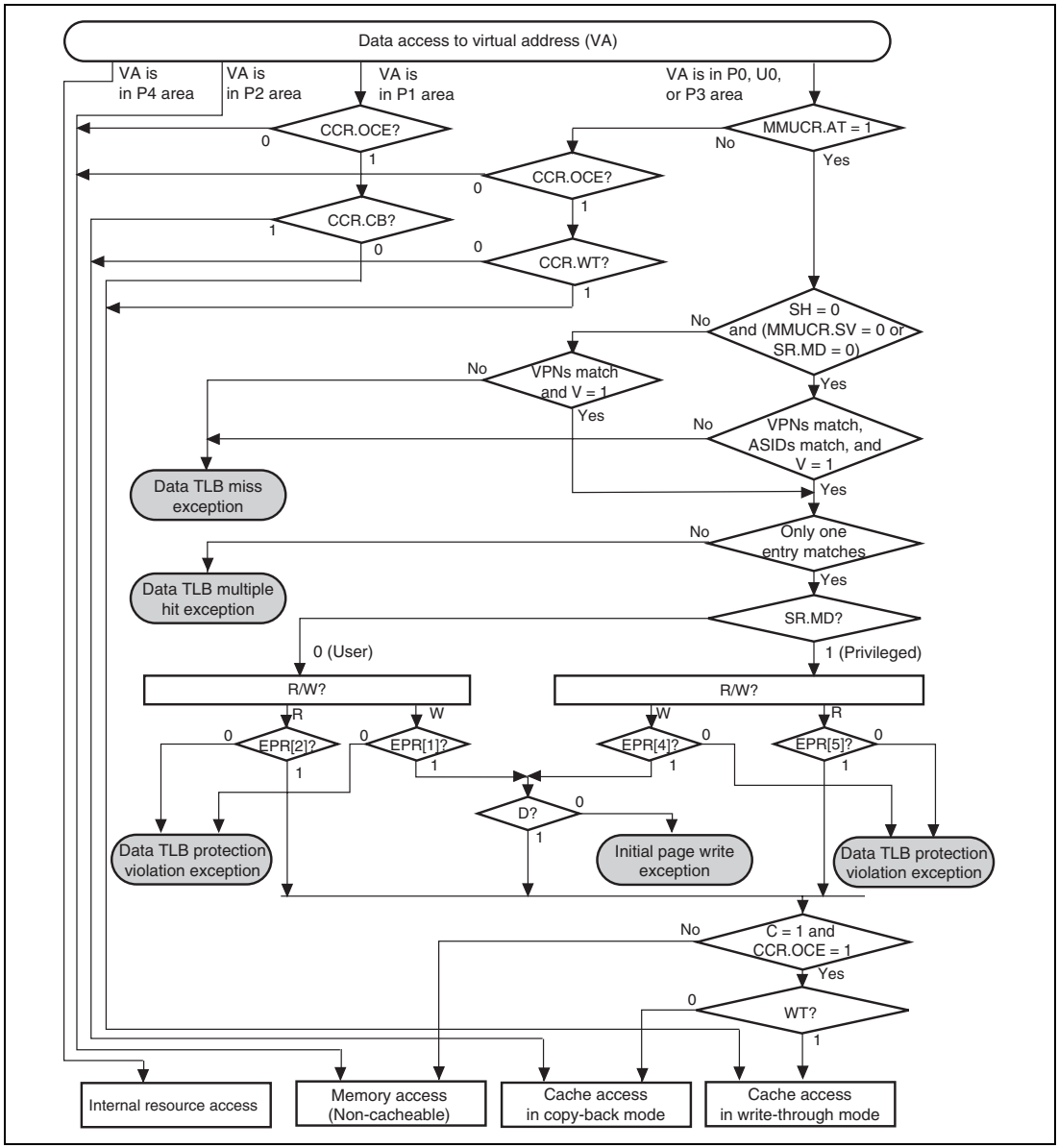


Figure 7.14 Flowchart of Memory Access Using UTLB (TLB Extended Mode)

Figure 7.15 is a flowchart of memory access using the ITLB in TLB extended mode.

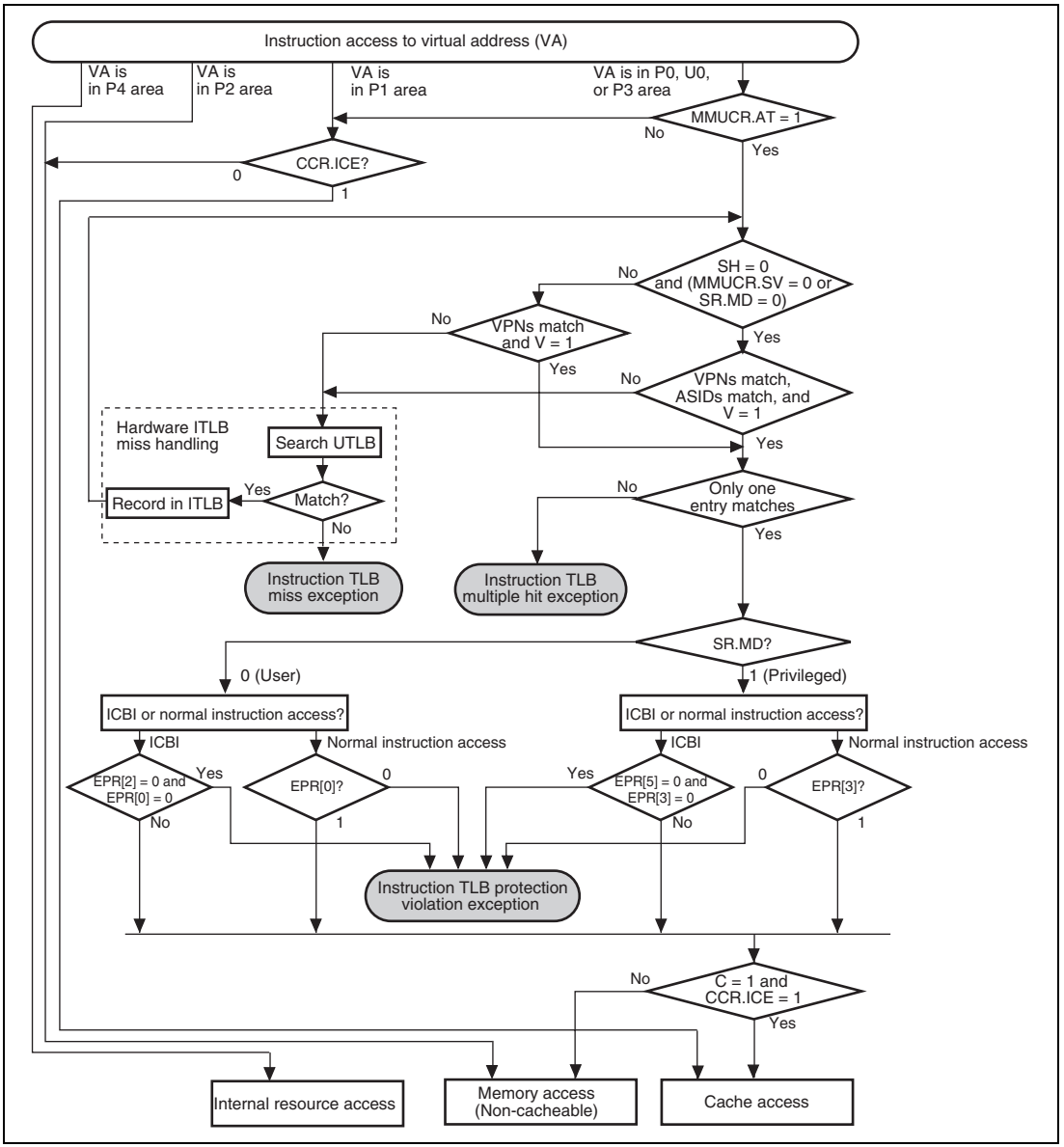


Figure 7.15 Flowchart of Memory Access Using ITLB (TLB Extended Mode)

7.5 MMU Functions

7.5.1 MMU Hardware Management

This LSI supports the following MMU functions.

1. The MMU decodes the virtual address to be accessed by software, and performs address translation by controlling the UTLB/ITLB in accordance with the MMUCR settings.
2. The MMU determines the cache access status on the basis of the page management information read during address translation (C and WT bits).
3. If address translation cannot be performed normally in a data access or instruction access, the MMU notifies software by means of an MMU exception.
4. If address translation information is not recorded in the ITLB in an instruction access, the MMU searches the UTLB. If the necessary address translation information is recorded in the UTLB, the MMU copies this information into the ITLB in accordance with the LRUI bit setting in MMUCR.

7.5.2 MMU Software Management

Software processing for the MMU consists of the following:

1. Setting of MMU-related registers. Some registers are also partially updated by hardware automatically.
2. Recording, deletion, and reading of TLB entries. There are two methods of recording UTLB entries: by using the LDTLB instruction, or by writing directly to the memory-mapped UTLB. ITLB entries can only be recorded by writing directly to the memory-mapped ITLB. Deleting or reading UTLB/ITLB entries is enabled by accessing the memory-mapped UTLB/ITLB.
3. MMU exception handling. When an MMU exception occurs, processing is performed based on information set by hardware.

7.5.3 MMU Instruction (LDTLB)

A TLB load instruction (LDTLB) is provided for recording UTLB entries. When an LDTLB instruction is issued, this LSI copies the contents of PTEH and PTEL (also the contents of PTEA in TLB extended mode) to the UTLB entry indicated by the URC bit in MMUCR. ITLB entries are not updated by the LDTLB instruction, and therefore address translation information purged from the UTLB entry may still remain in the ITLB entry. As the LDTLB instruction changes address translation information, ensure that it is issued by a program in the P1 or P2 area.

After the LDTLB instruction has been executed, execute one of the following three methods before an access (include an instruction fetch) the area where TLB is used to translate the address is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the area where TLB is used to translate the address.
2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the LT bit in IRMCR is 0 (initial value) before executing the LDTLB instruction, the specific instruction does not need to be executed. However, note that the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction after MMUCR has been updated.

Note that the method 3 may not be guaranteed in the future SuperH Series. Therefore, it is recommended that the method 1 or 2 should be used for being compatible with the future SuperH Series.

The operation of the LDTLB instruction is shown in figure 7.16 and 7.17.

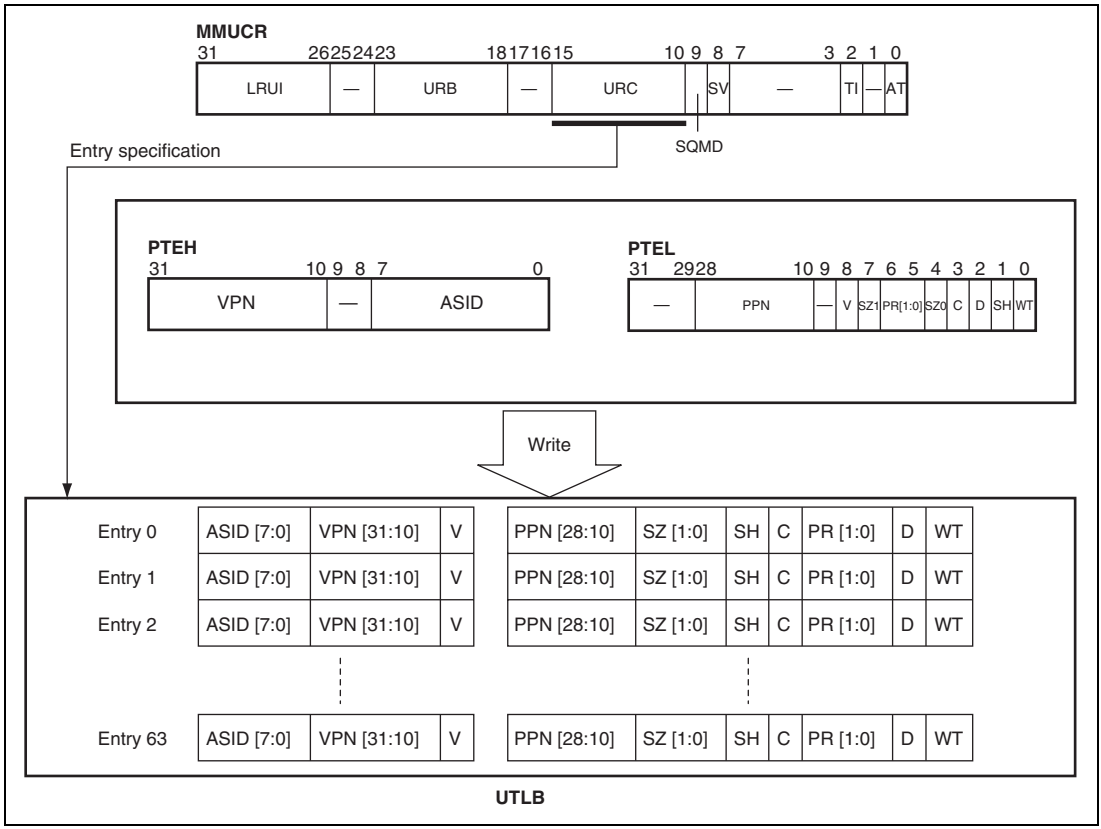


Figure 7.16 Operation of LDTLB Instruction (TLB Compatible Mode)

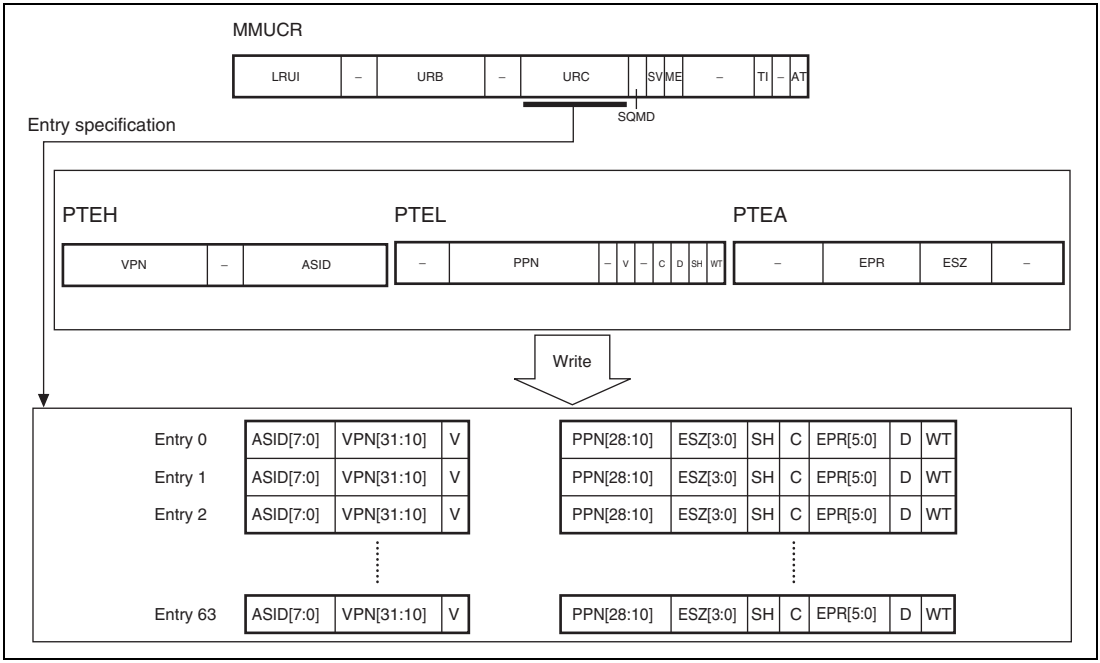


Figure 7.17 Operation of LDTLB Instruction (TLB Extended Mode)

7.5.4 Hardware ITLB Miss Handling

In an instruction access, this LSI searches the ITLB. If it cannot find the necessary address translation information (ITLB miss occurred), the UTLB is searched by hardware, and if the necessary address translation information is present, it is recorded in the ITLB. This procedure is known as hardware ITLB miss handling. If the necessary address translation information is not found in the UTLB search, an instruction TLB miss exception is generated and processing passes to software.

7.5.5 Avoiding Synonym Problems

The following explanation is for the case with 32-Kbyte operand cache.

When information on 1- or 4-Kbyte pages is written as TLB entries, a synonym problem may arise. The problem is that, when a number of virtual addresses are mapped onto a single physical address, the same physical address data is written to a number of cache entries, and it becomes impossible to guarantee data integrity. This problem does not occur with the instruction TLB and instruction cache because only data is read in these cases. In this LSI, entry specification is performed using bits 12 to 5 of the virtual address in order to achieve fast operand cache operation. However, bits 12 to 10 of the virtual address in the case of a 1-Kbyte page, and bit 12 of the virtual address in the case of a 4-Kbyte page, are subject to address translation. As a result, bits 12 to 10 of the physical address after translation may differ from bits 12 to 10 of the virtual address.

Consequently, the following restrictions apply to the writing of address translation information as UTLB entries.

- When address translation information whereby a number of 1-Kbyte page UTLB entries are translated into the same physical address is written to the UTLB, ensure that the VPN[12:10] values are the same.
- When address translation information whereby a number of 4-Kbyte page UTLB entries are translated into the same physical address is written to the UTLB, ensure that the VPN[12] value is the same.
- Do not use 1-Kbyte page UTLB entry physical addresses with UTLB entries of a different page size.
- Do not use 4-Kbyte page UTLB entry physical addresses with UTLB entries of a different page size.

The above restrictions apply only when performing accesses using the cache.

For cache sizes other than 32 Kbytes, the page sizes that can lead to synonym problems and the bits in VPN the value of which should be matched at the time of writing entries to the UTBL are different from those shown in the above explanation. The page sizes that can lead to synonym problems are shown in table 7.3 for cache sizes of 8 Kbytes to 64 Kbytes.

Table 7.3 Cache Size and Countermeasure for Avoiding Synonym Problems

Cache Size	Page Size that can Lead to Synonym Problems	Bits in VPN that should be Matched when Writing to UTLB
8 Kbytes	1 Kbyte	VPN[1:0]
16 Kbytes	1 Kbyte	VPN[11:10]
32 Kbytes	1 Kbyte	VPN[12:10]
	4 Kbytes	VPN[12]
64 Kbytes	1 Kbyte	VPN[13:10]
	4 Kbytes	VPN[13:12]

Note: When multiple items of address translation information use the same physical memory to provide for future expansion of the SuperH RISC engine family, ensure that the VPN[20:10] values are the same. Also, do not use the same physical address for address translation information of different page sizes.

7.6 MMU Exceptions

There are seven MMU exceptions: instruction TLB multiple hit exception, instruction TLB miss exception, instruction TLB protection violation exception, data TLB multiple hit exception, data TLB miss exception, data TLB protection violation exception, and initial page write exception. Refer to figures 7.9, 7.10, 7.14, 7.15, and section 5, Exception Handling for the conditions under which each of these exceptions occurs.

7.6.1 Instruction TLB Multiple Hit Exception

An instruction TLB multiple hit exception occurs when more than one ITLB entry matches the virtual address to which an instruction access has been made. If multiple hits occur when the UTLB is searched by hardware in hardware ITLB miss handling, an instruction TLB multiple hit exception will result.

When an instruction TLB multiple hit exception occurs, a reset is executed and cache coherency is not guaranteed.

(1) Hardware Processing

In the event of an instruction TLB multiple hit exception, hardware carries out the following processing:

1. Sets the virtual address at which the exception occurred in TEA.
2. Sets exception code H'140 in EXPEVT.
3. Branches to the reset handling routine (H'A000 0000).

(2) Software Processing (Reset Routine)

The ITLB entries which caused the multiple hit exception are checked in the reset handling routine. This exception is intended for use in program debugging, and should not normally be generated.

7.6.2 Instruction TLB Miss Exception

An instruction TLB miss exception occurs when address translation information for the virtual address to which an instruction access is made is not found in the UTLB entries by the hardware ITLB miss handling routine. The instruction TLB miss exception processing carried out by hardware and software is shown below. This is the same as the processing for a data TLB miss exception.

(1) Hardware Processing

In the event of an instruction TLB miss exception, hardware carries out the following processing:

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.
3. Sets exception code H'040 in EXPEVT.
4. Sets the PC value indicating the address of the instruction at which the exception occurred in SPC. If the exception occurred at a delay slot, sets the PC value indicating the address of the delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are saved in SGR.
6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.
9. Branches to the address obtained by adding offset H'0000 0400 to the contents of VBR, and starts the instruction TLB miss exception handling routine.

(2) Software Processing (Instruction TLB Miss Exception Handling Routine)

Software is responsible for searching the external memory page table and assigning the necessary page table entry. Software should carry out the following processing in order to find and assign the necessary page table entry.

1. In TLB compatible mode, write to PTEL the values of the PPN, PR, SZ, C, D, SH, V, and WT bits in the page table entry stored in the address translation table for external memory.
   In TLB extended mode, write to PTEL and PTEA the values of the PPN, EPR, ESZ, C, D, SH, V, and WT bits in the page table entry stored in the address translation table for external memory.
2. When the entry to be replaced in entry replacement is specified by software, write the value to the URC bits in MMUCR. If URC is greater than URB at this time, the value should be changed to an appropriate value after issuing an LDTLB instruction.

3. In TLB compatible mode, execute the LDTLB instruction and write the contents of PTEH and PTEL to the TLB.
   In TLB extended mode, execute the LDTLB instruction and write the contents of PTEH, PTEL, PTEA to the UTLB.
4. Finally, execute the exception handling return instruction (RTE) to terminate the exception handling routine and return control to the normal flow. The RTE instruction should be issued at least one instruction after the LDTLB instruction.

For the execution of the LDTLB instruction, see section 7.8.1, Note on Using LDTLB Instruction.

7.6.3 Instruction TLB Protection Violation Exception

An instruction TLB protection violation exception occurs when, even though an ITLB entry contains address translation information matching the virtual address to which an instruction access is made, the actual access type is not permitted by the access right specified by the PR or EPR bit. The instruction TLB protection violation exception processing carried out by hardware and software is shown below.

(1) Hardware Processing

In the event of an instruction TLB protection violation exception, hardware carries out the following processing:

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.
3. Sets exception code H'0A0 in EXPEVT.
4. Sets the PC value indicating the address of the instruction at which the exception occurred in SPC. If the exception occurred at a delay slot, sets the PC value indicating the address of the delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are saved in SGR.
6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.
9. Branches to the address obtained by adding offset H'0000 0100 to the contents of VBR, and starts the instruction TLB protection violation exception handling routine.

(2) Software Processing (Instruction TLB Protection Violation Exception Handling Routine)

Resolve the instruction TLB protection violation, execute the exception handling return instruction (RTE), terminate the exception handling routine, and return control to the normal flow. The RTE instruction should be issued at least one instruction after the LDTLB instruction.

7.6.4 Data TLB Multiple Hit Exception

A data TLB multiple hit exception occurs when more than one UTLB entry matches the virtual address to which a data access has been made.

When a data TLB multiple hit exception occurs, a reset is executed, and cache coherency is not guaranteed. The contents of PPN in the UTLB prior to the exception may also be corrupted.

(1) Hardware Processing

In the event of a data TLB multiple hit exception, hardware carries out the following processing:

1. Sets the virtual address at which the exception occurred in TEA.
2. Sets exception code H'140 in EXPEVT.
3. Branches to the reset handling routine (H'A000 0000).

(2) Software Processing (Reset Routine)

The UTLB entries which caused the multiple hit exception are checked in the reset handling routine. This exception is intended for use in program debugging, and should not normally be generated.

7.6.5 Data TLB Miss Exception

A data TLB miss exception occurs when address translation information for the virtual address to which a data access is made is not found in the UTLB entries. The data TLB miss exception processing carried out by hardware and software is shown below.

(1) Hardware Processing

In the event of a data TLB miss exception, hardware carries out the following processing:

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.

3. Sets exception code H'040 in the case of a read, or H'060 in the case of a write in EXPEVT (OCBP, OCBWB: read; OCBI, MOVCA.L: write).
4. Sets the PC value indicating the address of the instruction at which the exception occurred in SPC. If the exception occurred at a delay slot, sets the PC value indicating the address of the delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are saved in SGR.
6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.
9. Branches to the address obtained by adding offset H'0000 0400 to the contents of VBR, and starts the data TLB miss exception handling routine.

(2) Software Processing (Data TLB Miss Exception Handling Routine)

Software is responsible for searching the external memory page table and assigning the necessary page table entry. Software should carry out the following processing in order to find and assign the necessary page table entry.

1. In TLB compatible mode, write to PTEL the values of the PPN, PR, SZ, C, D, SH, V, and WT bits in the page table entry stored in the address translation table for external memory.
   In TLB extended mode, write to PTEL and PTEA the values of the PPN, EPR, ESZ, C, D, SH, V, and WT bits in the page table entry stored in the address translation table for external memory.
2. When the entry to be replaced in entry replacement is specified by software, write the value to the URC bits in MMUCR. If URC is greater than URB at this time, the value should be changed to an appropriate value after issuing an LDTLB instruction.
3. In TLB compatible mode, execute the LDTLB instruction and write the contents of PTEH and PTEL to the TLB.
   In TLB extended mode, execute the LDTLB instruction and write the contents of PTEH, PTEL, PTEA to the UTLB.
4. Finally, execute the exception handling return instruction (RTE), terminate the exception handling routine, and return control to the normal flow. The RTE instruction should be issued at least one instruction after the LDTLB instruction.

For the execution of the LDTLB instruction, see section 7.8.1, Note on Using LDTLB Instruction.

7.6.6 Data TLB Protection Violation Exception

A data TLB protection violation exception occurs when, even though a UTLB entry contains address translation information matching the virtual address to which a data access is made, the actual access type is not permitted by the access right specified by the PR or EPR bit. The data TLB protection violation exception processing carried out by hardware and software is shown below.

(1) Hardware Processing

In the event of a data TLB protection violation exception, hardware carries out the following processing:

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.
3. Sets exception code H'0A0 in the case of a read, or H'0C0 in the case of a write in EXPEVT (OCBP, OCBWB: read; OCBI, MOVCA.L: write).
4. Sets the PC value indicating the address of the instruction at which the exception occurred in SPC. If the exception occurred at a delay slot, sets the PC value indicating the address of the delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are saved in SGR.
6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.
9. Branches to the address obtained by adding offset H'0000 0100 to the contents of VBR, and starts the data TLB protection violation exception handling routine.

(2) Software Processing (Data TLB Protection Violation Exception Handling Routine)

Resolve the data TLB protection violation, execute the exception handling return instruction (RTE), terminate the exception handling routine, and return control to the normal flow. The RTE instruction should be issued at least one instruction after the LDTLB instruction.

7.6.7 Initial Page Write Exception

An initial page write exception occurs when the D bit is 0 even though a UTLB entry contains address translation information matching the virtual address to which a data access (write) is made, and the access is permitted. The initial page write exception processing carried out by hardware and software is shown below.

(1) Hardware Processing

In the event of an initial page write exception, hardware carries out the following processing:

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.
3. Sets exception code H'080 in EXPEVT.
4. Sets the PC value indicating the address of the instruction at which the exception occurred in SPC. If the exception occurred at a delay slot, sets the PC value indicating the address of the delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are saved in SGR.
6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.
9. Branches to the address obtained by adding offset H'0000 0100 to the contents of VBR, and starts the initial page write exception handling routine.

(2) Software Processing (Initial Page Write Exception Handling Routine)

Software is responsible for the following processing:

1. Retrieve the necessary page table entry from external memory.
2. Write 1 to the D bit in the external memory page table entry.
3. In TLB compatible mode, write to PTEL the values of the PPN, PR, SZ, C, D, SH, V, and WT bits in the page table entry stored in the address translation table for external memory.
   In TLB extended mode, write to PTEL and PTEA the values of the PPN, EPR, ESZ, C, D, SH, V, and WT bits in the page table entry stored in the address translation table for external memory.
4. When the entry to be replaced in entry replacement is specified by software, write that value to the URC bits in MMUCR. If URC is greater than URB at this time, the value should be changed to an appropriate value after issuing an LDTLB instruction.

5. In TLB compatible mode, execute the LDTLB instruction and write the contents of PTEH and PTEL to the TLB.
   In TLB extended mode, execute the LDTLB instruction and write the contents of PTEH, PTEL, PTEA to the UTLB.
6. Finally, execute the exception handling return instruction (RTE), terminate the exception handling routine, and return control to the normal flow. The RTE instruction should be issued at least one instruction after the LDTLB instruction.

7.7 Memory-Mapped TLB Configuration

To enable the ITLB and UTLB to be managed by software, their contents are allowed to be read from and written to by a program in the P1/P2 area with a MOV instruction in privileged mode. Operation is not guaranteed if access is made from a program in another area.

After the memory-mapped TLB has been accessed, execute one of the following three methods before an access (including an instruction fetch) to an area other than the P1/P2 area is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be an area other than the P1/P2 area.
2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the MT bit in IRMCR is 0 (initial value) before accessing the memory-mapped TLB, the specific instruction does not need to be executed. However, note that the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction after MMUCR has been updated.

Note that the method 3 may not be guaranteed in the future SuperH Series. Therefore, it is recommended that the method 1 or 2 should be used for being compatible with the future SuperH Series.

The ITLB and UTLB are allocated to the P4 area in the virtual address space.

In TLB compatible mode, VPN, V, and ASID in the ITLB can be accessed as an address array, PPN, V, SZ, PR, C, and SH as a data array. VPN, D, V, and ASID in the UTLB can be accessed as an address array, PPN, V, SZ, PR, C, D, WT, and SH as a data array. V and D can be accessed from both the address array side and the data array side.

In TLB extended mode, VPN, V, and ASID in the ITLB can be accessed as an address array, PPN, V, ESZ, EPR, C, and SH as a data array. VPN, D, V, and ASID in the UTLB can be accessed as an address array, PPN, V, ESZ, EPR, C, D, WT, and SH as a data array. V and D can be accessed from both the address array side and the data array side.

In both TLB compatible mode and TLB extended mode, only longword access is possible. Instruction fetches cannot be performed in these areas. For reserved bits, a write value of 0 should be specified; their read value is undefined.

7.7.1 ITLB Address Array

The ITLB address array is allocated to addresses H'F200 0000 to H'F2FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and VPN, V, and ASID to be written to the address array are specified in the data field.

In the address field, bits [31:24] have the value H'F2 indicating the ITLB address array and the entry is specified by bits [9:8]. As only longword access is used, 0 should be specified for address field bits [1:0].

In the data field, bits [31:10] indicate VPN, bit [8] indicates V, and bits [7:0] indicate ASID.

The following two kinds of operation can be used on the ITLB address array:

1. ITLB address array read

   VPN, V, and ASID are read into the data field from the ITLB entry corresponding to the entry set in the address field.
2. ITLB address array write

   VPN, V, and ASID specified in the data field are written to the ITLB entry corresponding to the entry set in the address field.

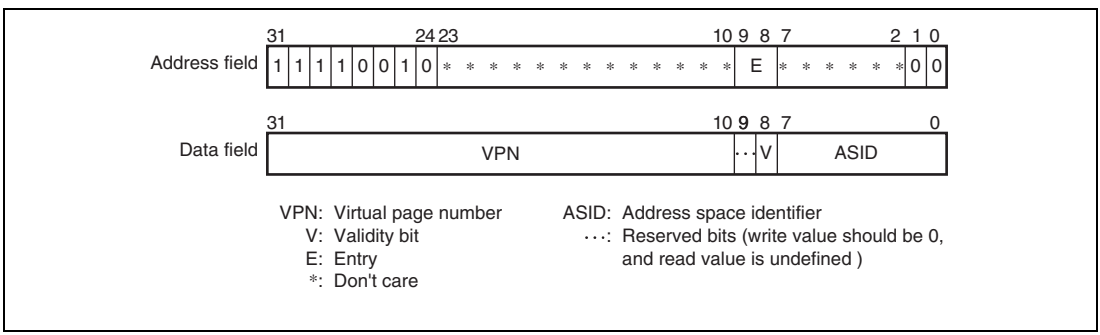


Figure 7.18 Memory-Mapped ITLB Address Array

7.7.2 ITLB Data Array (TLB Compatible Mode)

The ITLB data array is allocated to addresses H'F300 0000 to H'F37F FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and PPN, V, SZ, PR, C, and SH to be written to the data array are specified in the data field.

In the address field, bits [31:23] have the value H'F30 indicating ITLB data array and the entry is specified by bits [9:8].

In the data field, bits [28:10] indicate PPN, bit [8] indicates V, bits [7] and [4] indicate SZ, bit [6] indicates PR, bit [3] indicates C, and bit [1] indicates SH.

The following two kinds of operation can be used on ITLB data array:

1. ITLB data array read

   PPN, V, SZ, PR, C, and SH are read into the data field from the ITLB entry corresponding to the entry set in the address field.
2. ITLB data array write

   PPN, V, SZ, PR, C, and SH specified in the data field are written to the ITLB entry corresponding to the entry set in the address field.

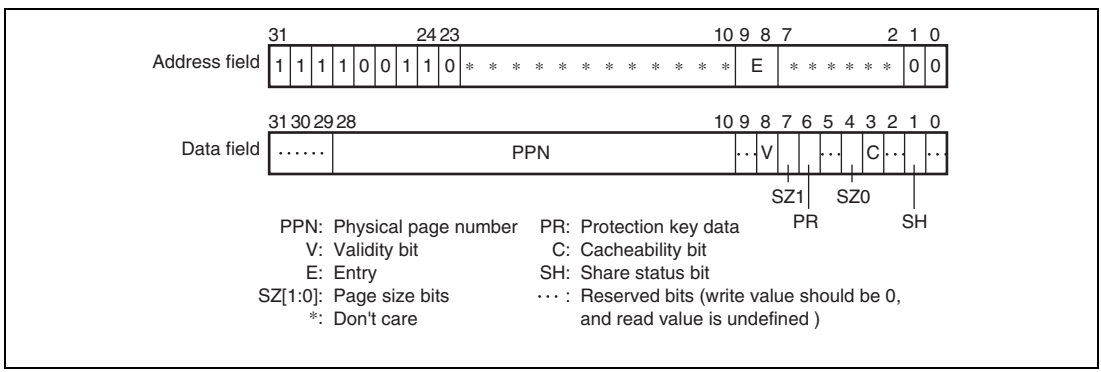


Figure 7.19 Memory-Mapped ITLB Data Array (TLB Compatible Mode)

7.7.3 ITLB Data Array (TLB Extended Mode)

In TLB extended mode the names of the data arrays have been changed from ITLB data array to ITLB data array 1, ITLB data array 2 is added, and the EPR and ESZ bits are accessible. In TLB extended mode, the PR and SZ bits of ITLB data array 1 are reserved and 0 should be specified as the write value for these bits. In addition, when a write to ITLB data array 1 is performed, a write to ITLB data array 2 of the same entry should always be performed.

In TLB compatible mode (MMUCR.ME = 0), ITLB data array 2 cannot be accessed. Operation if they are accessed is not guaranteed.

(1) ITLB Data Array 1

In TLB extended mode, bits 7, 6, and 4 in the data field, which correspond to the PR and SZ bits in compatible mode, are reserved. Specify 0 as the write value for these bits.

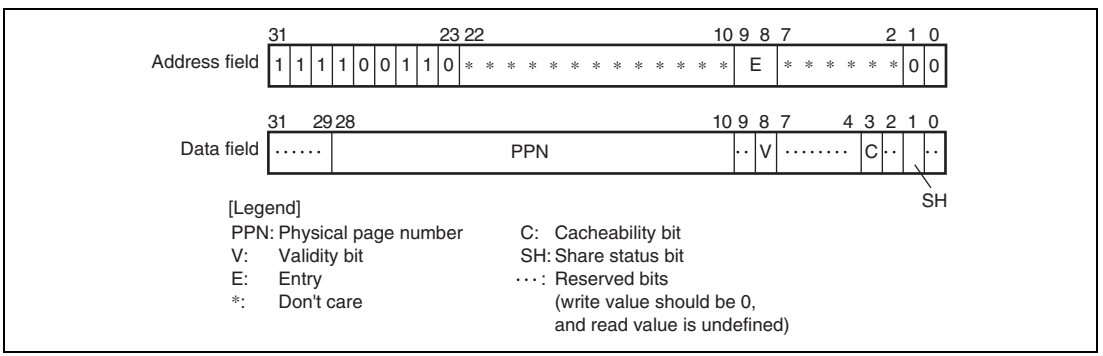


Figure 7.20 Memory-Mapped ITLB Data Array 1 (TLB Extended Mode)

(2) ITLB Data Array 2

The ITLB data array is allocated to addresses H'F380 0000 to H'F3FF FFFF in the P4 area. Access to data array 2 requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and EPR and ESZ to be written to data array 2 are specified in the data field.

In the address field, bits [31:23] have the value H'F38 indicating ITLB data array 2 and the entry is specified by bits [9:8].

In the data field, bits [13], [11], [10], and [8] indicate EPR[5], [3], [2], and [0], and bits [7:4] indicate ESZ, respectively.

The following two kinds of operation can be applied to ITLB data array 2:

1. ITLB data array 2 read

   EPR and ESZ are read into the data field from the ITLB entry corresponding to the entry set in the address field.
2. ITLB data array 2 write

   EPR and ESZ specified in the data field are written to the ITLB entry corresponding to the entry set in the address field.

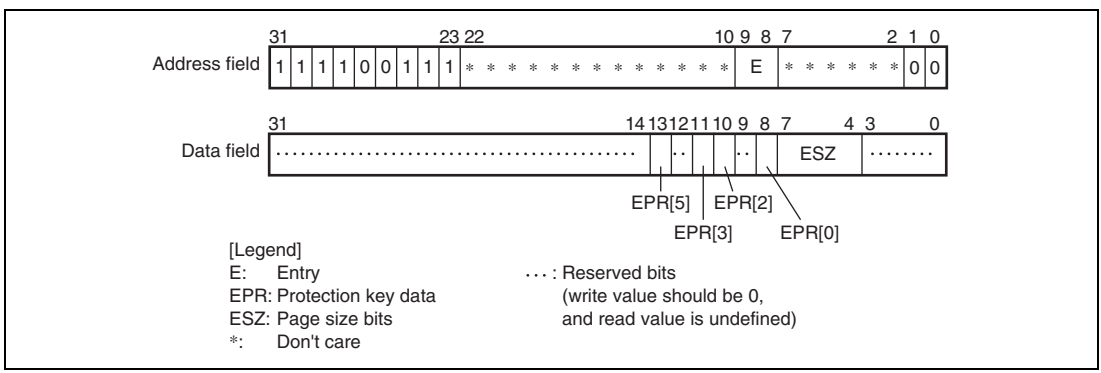


Figure 7.21 Memory-Mapped ITLB Data Array 2 (TLB Extended Mode)

7.7.4 UTLB Address Array

The UTLB address array is allocated to addresses H'F600 0000 to H'F60F FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and VPN, D, V, and ASID to be written to the address array are specified in the data field.

In the address field, bits [31:20] have the value H'F60 indicating the UTLB address array and the entry is specified by bits [13:8]. Bit [7] that is the association bit (A bit) in the address field specifies whether address comparison is performed in a write to the UTLB address array.

In the data field, bits [31:10] indicate VPN, bit [9] indicates D, bit [8] indicates V, and bits [7:0] indicate ASID.

The following three kinds of operation can be used on the UTLB address array:

1. UTLB address array read

   VPN, D, V, and ASID are read into the data field from the UTLB entry corresponding to the entry set in the address field. In a read, associative operation is not performed regardless of whether the association bit specified in the address field is 1 or 0.
2. UTLB address array write (non-associative)

   VPN, D, V, and ASID specified in the data field are written to the UTLB entry corresponding to the entry set in the address field. The A bit in the address field should be cleared to 0.
3. UTLB address array write (associative)

   When a write is performed with the A bit in the address field set to 1, comparison of all the UTLB entries is carried out using the VPN specified in the data field and ASID in PTEH. The usual address comparison rules are followed, but if a UTLB miss occurs, the result is no operation, and an exception is not generated. If the comparison identifies a UTLB entry corresponding to the VPN specified in the data field, D and V specified in the data field are written to that entry. This associative operation is simultaneously carried out on the ITLB, and if a matching entry is found in the ITLB, V is written to that entry. Even if the UTLB comparison results in no operation, a write to the ITLB is performed as long as a matching entry is found in the ITLB. If there is a match in both the UTLB and ITLB, the UTLB information is also written to the ITLB.

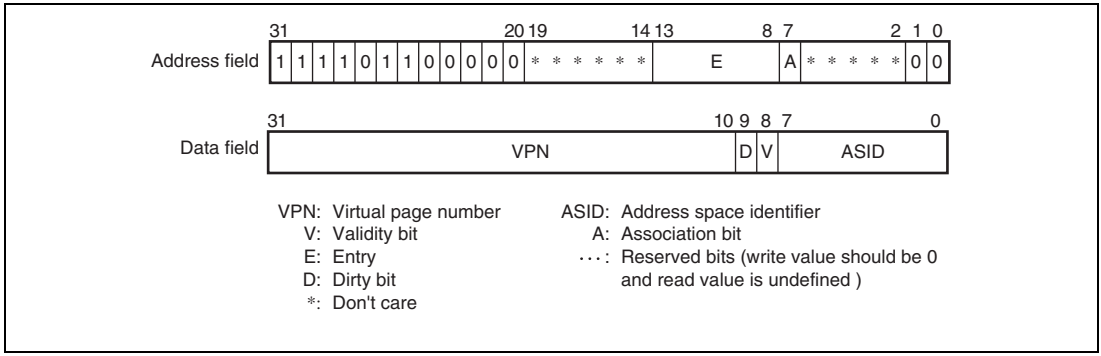


Figure 7.22 Memory-Mapped UTLB Address Array

7.7.5 UTLB Data Array (TLB Compatible Mode)

The UTLB data array is allocated to addresses H'F700 0000 to H'F70F FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and PPN, V, SZ, PR, C, D, SH, and WT to be written to data array are specified in the data field.

In the address field, bits [31:20] have the value H'F70 indicating UTLB data array and the entry is specified by bits [13:8].

In the data field, bits [28:10] indicate PPN, bit [8] indicates V, bits [7] and [4] indicate SZ, bits [6:5] indicate PR, bit [3] indicates C, bit [2] indicates D, bit [1] indicates SH, and bit [0] indicates WT.

The following two kinds of operation can be used on UTLB data array:

1. UTLB data array read

   PPN, V, SZ, PR, C, D, SH, and WT are read into the data field from the UTLB entry corresponding to the entry set in the address field.
2. UTLB data array write

   PPN, V, SZ, PR, C, D, SH, and WT specified in the data field are written to the UTLB entry corresponding to the entry set in the address field.

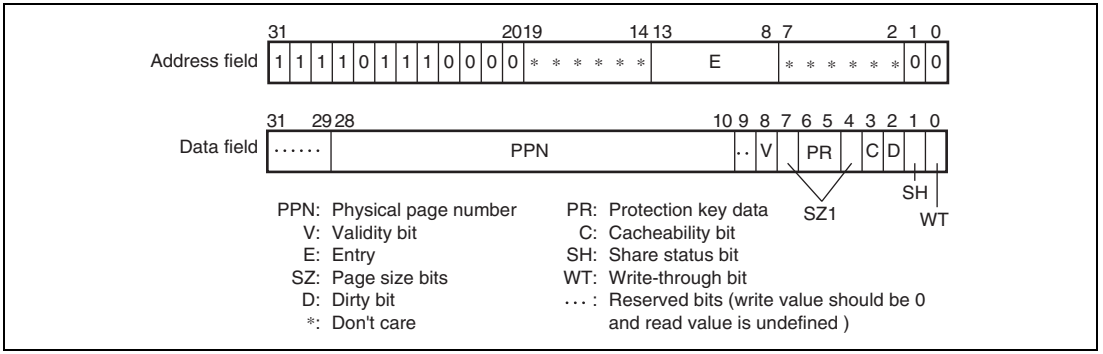


Figure 7.23 Memory-Mapped UTLB Data Array (TLB Compatible Mode)

7.7.6 UTLB Data Array (TLB Extended Mode)

In TLB extended mode, the names of the data arrays have been changed from UTLB data array to UTLB data array 1, UTLB data array 2 is added, and the EPR and ESZ bits are accessible. In TLB extended mode, the PR and SZ bits of UTLB data array 1 are reserved and 0 should be specified as the write value for these bits. In addition, when a write to UTLB data array 1 is performed, a write to UTLB data array 2 of the same entry should always be performed after that.

In TLB compatible mode (MMUCR.ME = 0), UTLB data array 2 cannot be accessed. Operation if they are accessed is not guaranteed.

(1) UTLB Data Array 1

In TLB extended mode, bits 7 to 4 in the data field, which correspond to the PR and SZ bits in compatible mode, are reserved. Specify 0 as the write value for these bits.

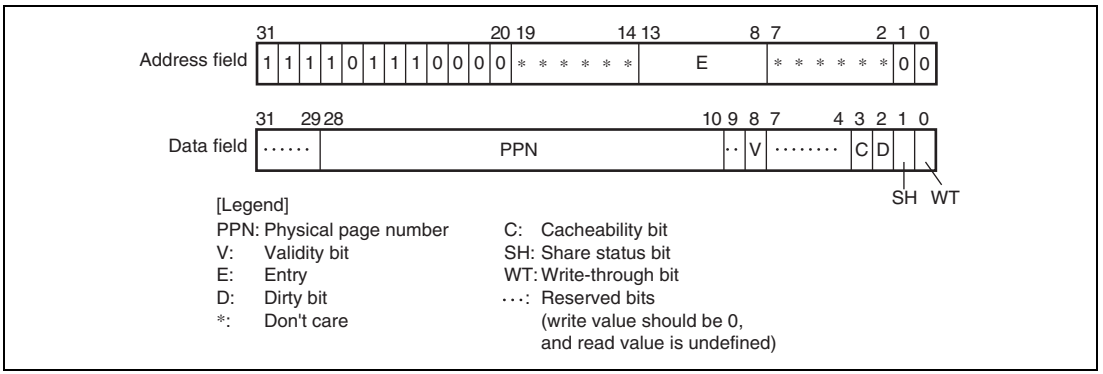


Figure 7.24 Memory-Mapped UTLB Data Array 1 (TLB Extended Mode)

(2) UTLB Data Array 2

The UTLB data array is allocated to addresses H'F780 0000 to H'F78F FFFF in the P4 area. Access to data array 2 requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and EPR and ESZ to be written to data array 2 are specified in the data field.

In the address field, bits [31:20] have the value H'F78 indicating UTLB data array 2 and the entry is specified by bits [13:8].

In the data field, bits [13:8] indicate EPR, and bits [7:4] indicate ESZ, respectively.

The following two kinds of operation can be applied to UTLB data array 2:

1. UTLB data array 2 read

   EPR and ESZ are read into the data field from the UTLB entry corresponding to the entry set in the address field.
2. UTLB data array 2 write

   EPR and ESZ specified in the data field are written to the UTLB entry corresponding to the entry set in the address field.

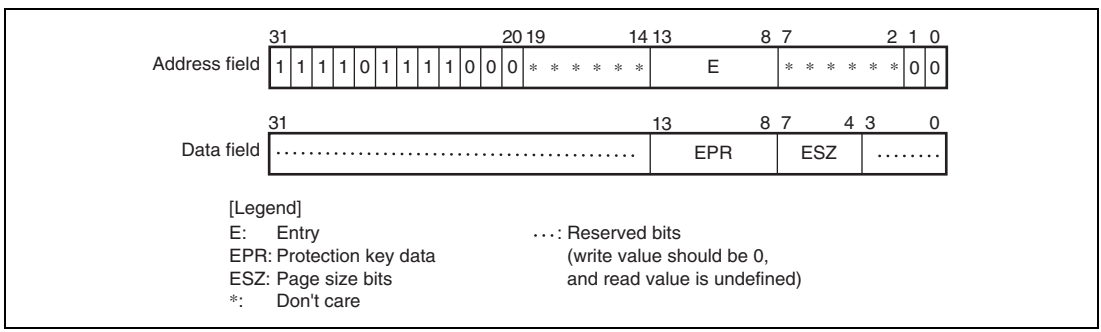


Figure 7.25 Memory-Mapped UTLB Data Array 2 (TLB Extended Mode)

7.8 Usage Notes

7.8.1 Note on Using LDTLB Instruction

When using an LDTLB instruction instead of software to a value to the MMUCR. URC, execute 1 or 2 below.

1. Place the TLB miss exception handling routine*[1] only in the P1, P2 area ,or the on-chip memory so that all the instruction accesses*[3] in the TLB miss exception handling routine should occur solely in the P1, P2 area, or the on-chip memory. Clear the RP bit in the RAMCR register to 0 (initial value), when the TLB miss exception handling routine is placed in the on-chip memory.

   Do not make an attempt to execute the FDIV or FSQRT instruction in the TLB miss exception handling routine.
2. If a TLB miss exception occurs, add 1 to MMUCR.URC before executing an LDTLB instruction.

Notes: 1. An exception handling routine is an entire set of instructions that are executed from the address (VBR + offset) upon occurrence of an exception to the RTE for returning to the original program or to the RTE delay slot.

2. Instruction accesses include the PREFI and ICBI instructions.

7.8.2 Notes on the Unbuffered Write Setting

The setting of the UB bit in the memory management unit (MMU) is used for buffered write control, but in some cases the CPU may not wait for the end of writing bus access before starting the next bus access even when UB is set to 1 (unbuffered write).

When the MMU's UB bit (in the physical address space control register (PASCR)) is cleared to 0 (buffered write), the CPU does not wait for the end of writing bus access before starting the next bus access, and when UB is set to 1 (unbuffered write), the CPU waits for the end of writing bus access before starting the next bus access.

Nevertheless, under the following frequency setting conditions the CPU may not wait for the end of writing bus access before starting the next bus access, even when UB is set to 1.

- When the ratio of the CPU clock frequency to the SuperHyway bus clock frequency is N:1 ($N \geq 4$)

Under these circumstances, there may be cases in which software does not operate as intended if it uses the unbuffered write setting to maintain the sequential relationship between an unbuffered write and the read or write that follows it.

As a workaround, ensure that the conditions described in (1) or (2) below are satisfied.

(1) Set the CPU clock frequency and SuperHyway bus clock frequency to a ratio of N:1 ($N \leq 3$).

(2) Limiting use of the unbuffered write setting

When the ratio of the CPU clock frequency to the SuperHyway bus clock frequency is N:1 ($N \geq 4$), do not set the UB[n] (n = 7 to 0) bits in PASCR to unbuffered write (UB = 1). Instead, use the buffered write setting (UB = 0).

Note that when the buffered write setting is used, there may be cases in which software does not operate as intended if it assumes that the sequential relationship between a write and the read or write that follows it will always be maintained. To guarantee the sequential relationship, either insert a SYNCO instruction or execute an instruction that reads the immediately preceding write destination address after the initial write instruction in the sequence to be maintained.

Section 8 Caches

This LSI has an on-chip 32-Kbyte instruction cache (IC) for instructions and an on-chip 32-Kbyte operand cache (OC) for data.

Note: For the size of instruction cache and operand cache, see the hardware manual of the product. This manual describes the 32-Kbyte case for each cache memory.
For different cache sizes, bit positions different from those shown in figures 8.1, 8.2, and 8.5 to 8.8 apply. The bit positions in ways and entries for various cache sizes are given in the table below. The bit positions in ways apply to figures 8.5 to 8.8, and those in entries apply to figures 8.1, 8.2, 8.5, 8.7, and 8.8.

Cache size	Way	Entry
8 Kbytes	bit[12:11]	bit[10:5]
16 Kbytes	bit[13:12]	bit[11:5]
32 Kbytes	bit[14:13]	bit[12:5]
64 Kbytes	bit[15:14]	bit[13:5]

8.1 Features

The features of the cache are given in table 8.1.

This LSI supports two 32-byte store queues (SQs) to perform high-speed writes to external memory. The features of the store queues are given in table 8.2.

Table 8.1 Cache Features

Item	Instruction Cache	Operand Cache
Capacity	32-Kbyte cache	32-Kbyte cache
Type	4-way set-associative, virtual address index/physical address tag	4-way set-associative, virtual address index/physical address tag
Line size	32 bytes	32 bytes
Entries	256 entries/way	256 entries/way
Write method	—	Copy-back/write-through selectable
Replacement method	LRU (least-recently-used) algorithm	LRU (least-recently-used) algorithm

Table 8.2 Store Queue Features

Item	Store Queues
Capacity	32 bytes × 2
Addresses	H'E000 0000 to H'E3FF FFFF
Write	Store instruction (1-cycle write)
Write-back	Prefetch instruction (PREF instruction)
Access right	When MMU is disabled: Determined by SQMD bit in MMUCR When MMU is enabled: Determined by PR for each page

The operand cache of this LSI is 4-way set associative, each way comprising 256 cache lines. Figure 8.1 shows the configuration of the operand cache.

The instruction cache is 4-way set-associative, each way comprising 256 cache lines. Figure 8.2 shows the configuration of the instruction cache.

This LSI has an IC way prediction scheme to reduce power consumption. In addition, memory-mapped associative writing, which is detectable as an exception, can be enabled by using the non-support detection exception register (EXPMASK). For details, see section 5, Exception Handling.

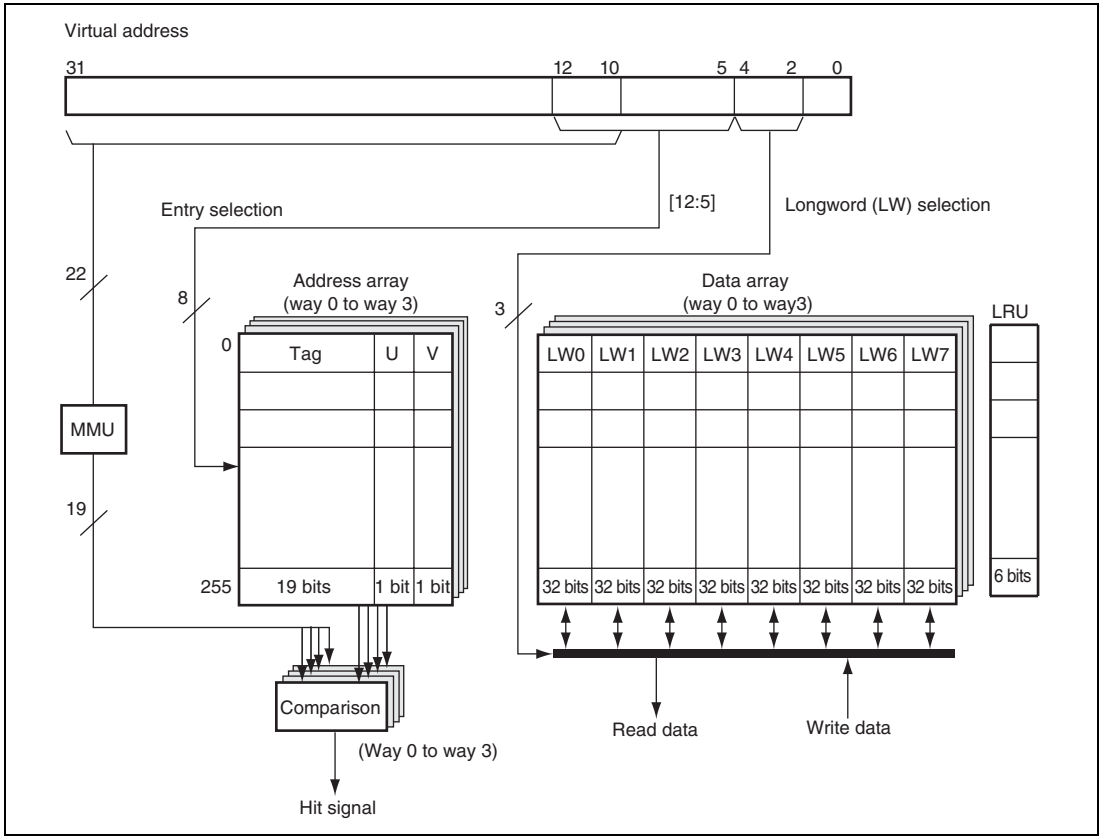


Figure 8.1 Configuration of Operand Cache

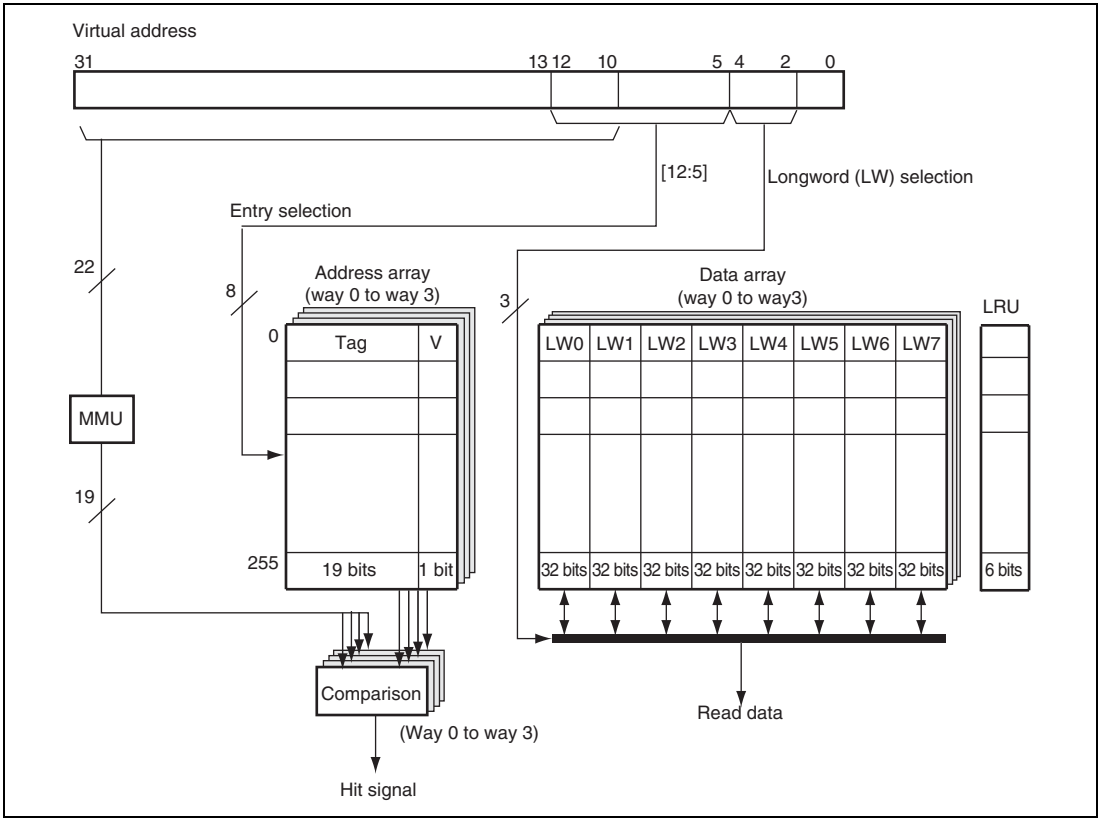


Figure 8.2 Configuration of Instruction Cache

- Tag

  Stores the upper 19 bits of the 29-bit physical address of the data line to be cached. The tag is not initialized by a power-on or manual reset.

- V bit (validity bit)

  Indicates that valid data is stored in the cache line. When this bit is 1, the cache line data is valid. The V bit is initialized to 0 by a power-on reset, but retains its value in a manual reset.

- U bit (dirty bit)

  The U bit is set to 1 if data is written to the cache line while the cache is being used in copy-back mode. That is, the U bit indicates a mismatch between the data in the cache line and the data in external memory. The U bit is never set to 1 while the cache is being used in write-through mode, unless it is modified by accessing the memory-mapped cache (see section 8.6, Memory-Mapped Cache Configuration). The U bit is initialized to 0 by a power-on reset, but retains its value in a manual reset.

- Data array

  The data field holds 32 bytes (256 bits) of data per cache line. The data array is not initialized by a power-on or manual reset.

- LRU

  In a 4-way set-associative method, up to 4 items of data can be registered in the cache at each entry address. When an entry is registered, the LRU bit indicates which of the 4 ways it is to be registered in. The LRU mechanism uses 6 bits of each entry, and its usage is controlled by hardware. The LRU (least-recently-used) algorithm is used for way selection, and selects the less recently accessed way. The LRU bits are initialized to 0 by a power-on reset but not by a manual reset. The LRU bits cannot be read from or written to by software.

8.2 Register Descriptions

The following registers are related to cache.

Table 8.3 Register Configuration

Register Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Size
Cache control register	CCR	R/W	H'FF00 001C	H'1F00 001C	32
Queue address control register 0	QACR0	R/W	H'FF00 0038	H'1F00 0038	32
Queue address control register 1	QACR1	R/W	H'FF00 003C	H'1F00 003C	32
On-chip memory control register	RAMCR	R/W	H'FF00 0074	H'1F00 0074	32

Note: * These P4 addresses are for the P4 area in the virtual address space. These area 7 addresses are accessed from area 7 in the physical address space by means of the TLB.

Table 8.4 Register States in Each Processing State

Register Name	Abbreviation	Power-on Reset	Manual Reset	Sleep	Standby
Cache control register	CCR	H'0000 0000	H'0000 0000	Retained	Retained
Queue address control register 0	QACR0	Undefined	Undefined	Retained	Retained
Queue address control register 1	QACR1	Undefined	Undefined	Retained	Retained
On-chip memory control register	RAMCR	H'0000 0000	H'0000 0000	Retained	Retained

8.2.1 Cache Control Register (CCR)

CCR controls the cache operating mode, the cache write mode, and invalidation of all cache entries.

CCR modifications must only be made by a program in the non-cacheable P2 area or IL memory. After CCR has been updated, execute one of the following three methods before an access (including an instruction fetch) to the cacheable area is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the cacheable area.
2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the R2 bit in IRMCR is 0 (initial value) before updating CCR, the specific instruction does not need to be executed. However, note that the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction after CCR has been updated.

Note that the method 3 may not be guaranteed in the future SuperH Series. Therefore, it is recommended that the method 1 or 2 should be used for being compatible with the future SuperH Series.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	ICI	—	—	ICE	—	—	—	—	OCI	CB	WT	OCE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
11	ICI	0	R/W	IC Invalidation Bit When 1 is written to this bit, the V bits of all IC entries are cleared to 0. This bit is always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
10, 9	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
8	ICE	0	R/W	IC Enable Bit Selects whether the IC is used. Note however when address translation is performed, the IC cannot be used unless the C bit in the page management information is also 1. 0: IC not used 1: IC used
7 to 4	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
3	OCI	0	R/W	OC Invalidation Bit When 1 is written to this bit, the V and U bits of all OC entries are cleared to 0. This bit is always read as 0.
2	CB	0	R/W	Copy-Back Bit Indicates the P1 area cache write mode. 0: Write-through mode 1: Copy-back mode
1	WT	0	R/W	Write-Through Mode Indicates the P0, U0, and P3 area cache write mode. When address translation is performed, the value of the WT bit in the page management information has priority. 0: Copy-back mode 1: Write-through mode
0	OCE	0	R/W	OC Enable Bit Selects whether the OC is used. Note however when address translation is performed, the OC cannot be used unless the C bit in the page management information is also 1. 0: OC not used 1: OC used

8.2.2 Queue Address Control Register 0 (QACR0)

QACR0 specifies the area onto which store queue 0 (SQ0) is mapped when the MMU is disabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	AREA0			—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	—	—	—	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
4 to 2	AREA0	Undefined	R/W	When the MMU is disabled, these bits generate physical address bits [28:26] for SQ0.
1, 0	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.

8.2.3 Queue Address Control Register 1 (QACR1)

QACR1 specifies the area onto which store queue 1 (SQ1) is mapped when the MMU is disabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	AREA1			—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	—	—	—	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
4 to 2	AREA1	Undefined	R/W	When the MMU is disabled, these bits generate physical address bits [28:26] for SQ1.
1, 0	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.

8.2.4 On-Chip Memory Control Register (RAMCR)

RAMCR controls the number of ways in the IC and OC and prediction of the IC way.

RAMCR modifications must only be made by a program in the non-cacheable P2 area. After RAMCR has been updated, execute one of the following three methods before an access (including an instruction fetch) to the cacheable area or the on-chip memory area is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the cacheable area, or the on-chip memory.
2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the R2 bit in IRMCR is 0 (initial value) before updating RAMCR, the specific instruction does not need to be executed. However, note that the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction after RAMCR has been updated.

Note that the method 3 may not be guaranteed in the future SuperH Series. Therefore, it is recommended that the method 1 or 2 should be used for being compatible with the future SuperH Series.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RMD	RP	IC2W	OC2W	ICWPW	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
9	RMD	0	R/W	On-Chip Memory Access Mode Bit For details, see section 9.4, On-Chip Memory Protective Functions.

Bit	Bit Name	Initial Value	R/W	Description
8	RP	0	R/W	On-Chip Memory Protection Enable Bit For details, see section 9.4, On-Chip Memory Protective Functions.
7	IC2W	0	R/W	IC Two-Way Mode bit 0: IC is a four-way operation 1: IC is a two-way operation For details, see section 8.4.3, IC Two-Way Mode.
6	OC2W	0	R/W	OC Two-Way Mode bit 0: OC is a four-way operation 1: OC is a two-way operation For details, see section 8.3.6, OC Two-Way Mode.
5	ICWPD	0	R/W	IC Way Prediction Stop Selects whether the IC way prediction is used. 0: Instruction cache performs way prediction. 1: Instruction cache does not perform way prediction.
4 to 0	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.

8.3 Operand Cache Operation

8.3.1 Read Operation

When the Operand Cache (OC) is enabled (OCE = 1 in CCR) and data is read from a cacheable area, the cache operates as follows:

1. The tag, V bit, U bit, and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tags read from the each way is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU:

- If there is a way whose tag matches and its V bit is 1, see No. 3.
- If there is no way whose tag matches and its V bit is 1 and the U bit of the way which is selected to replace using the LRU bits is 0, see No. 4.
- If there is no way whose tag matches and its V bit is 1 and the U bit of the way which is selected to replace using the LRU bits is 1, see No. 5.

3. Cache hit

   The data indexed by virtual address bits [4:0] is read from the data field of the cache line on the hitted way in accordance with the access size. Then the LRU bits are updated to indicate the hitted way is the latest one.

4. Cache miss (no write-back)

   Data is read into the cache line on the way, which is selected to replace, from the physical address space corresponding to the virtual address. When reading data, the read data is returned to the CPU at the point in time when that data arrives in the cache. While the remaining data on the cache line is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, 1 is written to the V bit and 0 is written to the U bit on the way. Then the LRU bit is updated to indicate the way is latest one.

5. Cache miss (with write-back)

   The tag and data field of the cache line on the way which is selected to replace are saved in the write-back buffer. Then data is read into the cache line on the way which is selected to replace from the physical address space corresponding to the virtual address. When reading data, the read data is returned to the CPU at the point in time when that data arrives in the cache. While the remaining one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, 1 is written to the V bit, and 0 to the U bit. And the LRU bits are updated to indicate the way is latest one. The data in the write-back buffer is then written back to external memory.

8.3.2 Prefetch Operation

When the Operand Cache (OC) is enabled (OCE = 1 in CCR) and data is prefetched from a cacheable area, the cache operates as follows:

1. The tag, V bit, U bit, and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU:

- If there is a way whose tag matches and its V bit is 1, see No. 3.
- If there is no way whose tag matches and the V bit is 1, and the U bit of the way which is selected to replace using the LRU bits is 0, see No. 4.
- If there is no way whose tag matches and the V bit is 1, and the U bit of the way which is selected to replace using the LRU bits is 1, see No. 5.

3. Cache hit

   Then the LRU bits are updated to indicate the hitted way is the latest one.

4. Cache miss (no write-back)

   Data is read into the cache line on the way, which is selected to replace, from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. In the prefetch operation the CPU doesn't wait the data arrives. While the one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, 1 is written to the V bit and 0 is written to the U bit on the way. And the LRU bit is updated to indicate the way is latest one.

5. Cache miss (with write-back)

   The tag and data field of the cache line on the way which is selected to replace are saved in the write-back buffer. Then data is read into the cache line on the way which is selected to replace from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. In the prefetch operation the CPU doesn't wait the data arrives. While the one cache line of data is being read, the CPU can execute the next processing. And the LRU bits are updated to indicate the way is latest one. The data in the write-back buffer is then written back to external memory.

8.3.3 Write Operation

When the Operand Cache (OC) is enabled (OCE = 1 in CCR) and data is written to a cacheable area, the cache operates as follows:

1. The tag, V bit, U bit, and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU:

- If there is a way whose tag matches and its V bit is 1, see No. 3 for copy-back and No. 4 for write-through.
- I If there is no way whose tag matches and its V bit is 1 and the U bit of the way which is selected to replace using the LRU bits is 0, see No. 5 for copy-back and No. 7 for write-through.
- If there is no way whose tag matches and its V bit is 1 and the U bit of the way which is selected to replace using the LRU bits is 1, see No. 6 for copy-back and No. 7 for write-through.

3. Cache hit (copy-back)

   A data write in accordance with the access size is performed for the data field on the hit way which is indexed by virtual address bits [4:0]. Then 1 is written to the U bit. The LRU bits are updated to indicate the way is the latest one.

4. Cache hit (write-through)

   A data write in accordance with the access size is performed for the data field on the hit way which is indexed by virtual address bits [4:0]. A write is also performed to external memory corresponding to the virtual address. Then the LRU bits are updated to indicate the way is the latest one. In this case, the U bit isn't updated.

5. Cache miss (copy-back, no write-back)

   A data write in accordance with the access size is performed for the data field on the hit way which is indexed by virtual address bits [4:0]. Then, the data, excluding the cache-missed data which is written already, is read into the cache line on the way which is selected to replace from the physical address space corresponding to the virtual address.

   Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. While the remaining data on the cache line is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, 1 is written to the V bit and the U bit on the way. Then the LRU bit is updated to indicate the way is latest one.

6. Cache miss (copy-back, with write-back)

   The tag and data field of the cache line on the way which is selected to replace are saved in the write-back buffer. Then a data write in accordance with the access size is performed for the data field on the hit way which is indexed by virtual address bits [4:0]. Then, the data, excluding the cache-missed data which is written already, is read into the cache line on the way which is selected to replace from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. While the remaining data on the cache line is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, 1 is written to the V bit and the U bit on the way. Then the LRU bit is updated to indicate the way is latest one. Then the data in the write-back buffer is then written back to external memory.

7. Cache miss (write-through)

   A write of the specified access size is performed to the external memory corresponding to the virtual address. In this case, a write to cache is not performed.

8.3.4 Write-Back Buffer

In order to give priority to data reads to the cache and improve performance, this LSI has a write-back buffer which holds the relevant cache entry when it becomes necessary to purge a dirty cache entry into external memory as the result of a cache miss. The write-back buffer contains one cache line of data and the physical address of the purge destination.

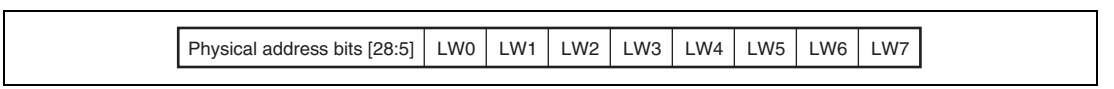


Figure 8.3 Configuration of Write-Back Buffer

8.3.5 Write-Through Buffer

This LSI has a 64-bit buffer for holding write data when writing data in write-through mode or writing to a non-cacheable area. This allows the CPU to proceed to the next operation as soon as the write to the write-through buffer is completed, without waiting for completion of the write to external memory.

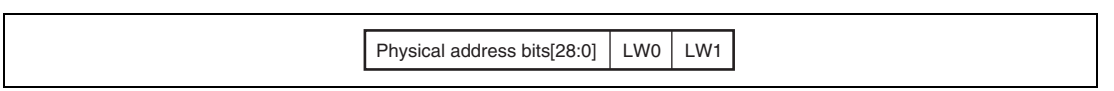


Figure 8.4 Configuration of Write-Through Buffer

8.3.6 OC Two-Way Mode

When the OC2W bit in RAMCR is set to 1, OC two-way mode which only uses way 0 and way 1 in the OC is entered. Thus, power consumption can be reduced. In this mode, only way 0 and way 1 are used even if a memory-mapped OC access is made.

The OC2W bit should be modified by a program in the P2 area. At that time, if the valid line has already been recorded in the OC, data should be written back by software, if necessary, 1 should be written to the OCI bit in CCR, and all entries in the OC should be invalid before modifying the OC2W bit.

8.4 Instruction Cache Operation

8.4.1 Read Operation

When the IC is enabled (ICE = 1 in CCR) and instruction fetches are performed from a cacheable area, the instruction cache operates as follows:

1. The tag, V bit, U bit and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU:

- If there is a way whose tag matches and the V bit is 1, see No. 3.
- If there is no way whose tag matches and the V bit is 1, see No. 4.

3. Cache hit

   The data indexed by virtual address bits [4:2] is read as an instruction from the data field on the hit way. The LRU bits are updated to indicate the way is the latest one.

4. Cache miss

   Data is read into the cache line on the way which selected using LRU bits to replace from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data, and when the corresponding data arrives in the cache, the read data is returned to the CPU as an instruction. While the remaining one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, and 1 is written to the V bit, the LRU bits are updated to indicate the way is the latest one.

8.4.2 Prefetch Operation

When the IC is enabled (ICE = 1 in CCR) and instruction prefetches are performed from a cacheable area, the instruction cache operates as follows:

1. The tag, V bit, Ubit and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting from virtual address translation by the MMU:

- If there is a way whose tag matches and the V bit is 1, see No. 3.
- If there is no way whose tag matches and the V bit is 1, see No. 4.

3. Cache hit

   The LRU bits is updated to indicate the way is the latest one.

4. Cache miss

   Data is read into the cache line on a way which selected using the LRU bits to replace from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. In the prefetch operation, the CPU doesn't wait the data arrived. While the one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, and 1 is written to the V bit, the LRU bits is updated to indicate the way is the latest one.

8.4.3 IC Two-Way Mode

When the IC2W bit in RAMCR is set to 1, IC two-way mode which only uses way 0 and way 1 in the IC is entered. Thus, power consumption can be reduced. In this mode, only way 0 and way 1 are used even if a memory-mapped IC access is made.

The IC2W bit should be modified by a program in the P2 area. At that time, if the valid line has already been recorded in the IC, 1 should be written to the ICI bit in CCR and all entries in the IC should be invalid before modifying the IC2W bit.

8.4.4 Instruction Cache Way Prediction Operation

This LSI incorporates an instruction cache (IC) way prediction scheme to reduce power consumption. This is achieved by activating only the data array that corresponds to a predicted way. When way prediction misses occur, data must be re-read from the right way, which may lead to lower performance in instruction fetching. Setting the ICWPD bit to 1 disables the IC way prediction scheme. Since way prediction misses do not occur in this mode, there is no loss of performance in instruction fetching but the IC consumes more power. The ICWPD bit should be modified by a program in the non-cacheable P2 area. If a valid line has already been recorded in the IC at this time, invalidate all entries in the IC by writing 1 to the ICI bit in CCR before modifying the ICWPD bit.

8.5 Cache Operation Instruction

8.5.1 Coherency between Cache and External Memory

(1) Cache Operation Instruction

Coherency between cache and external memory should be assured by software. In this LSI, the following six instructions are supported for cache operations. Details of these instructions are given in section 11, Instruction Descriptions of the SH-4A Extended Functions Software Manual.

- Operand cache invalidate instruction: OCBI @Rn
  Operand cache invalidation (no write-back)
- Operand cache purge instruction: OCBP @Rn
  Operand cache invalidation (with write-back)
- Operand cache write-back instruction: OCBWB @Rn
  Operand cache write-back
- Operand cache allocate instruction: MOVCA.L R0,@Rn
  Operand cache allocation
- Instruction cache invalidate instruction: ICBI @Rn
  Instruction cache invalidation
- Operand access synchronization instruction: SYNCO
  Wait for data transfer completion

(2) Coherency Control

The operand cache can receive "PURGE" and "FLUSH" transaction from SuperHyway bus to control the cache coherency. Since the address used by the PURGE and FLUSH transaction is a physical address, do not use the 1 Kbyte page size to avoid cache synonym problem in MMU enable mode.

- PURGE transaction
  When the operand cache is enabled, the PURGE transaction checks the operand cache and invalidates the hit entry. If the invalidated entry is dirty, the data is written back to the external memory. If the transaction is not hit to the cache, it is no-operation.

- FLUSH transaction

  When the operand cache is enabled, the FLUSH transaction checks the operand cache and if the hit line is dirty, then the data is written back to the external memory. If the transaction is not hit to the cache or the hit entry is not dirty, it is no-operation.

(3) Changes in Instruction Specifications Regarding Coherency Control

Of the operand cache operating instructions, the coherency control-related specifications of OCBI, OCBP, and OCBWB have been changed from those of the SH-4A with H'20-valued VER bits in the processor version register (PVR).

- Changes in the invalidate instruction OCBI@Rn

  When Rn is designating an address in a non-cacheable area, this instruction is executed as NOP in the SH-4A with H'20-valued VER bits in the processor version register (PVR). In the SH-4A with extended functions, this instruction invalidates the operand cache line designated by way = Rn[14:13] and entry = Rn[12:5] provided that Rn[31:24] = H'F4 (OC address array area). In this process, writing back of the line does not take place even if the line to be invalidated is dirty. This operation is only executable in privileged mode, and an address error exception occurs in user mode. TLB-related exceptions do not occur.

  Do not execute this instruction to invalidate the memory-mapped array areas and control register areas for which Rn[31:24] is not H'F4, and their reserved areas (H'F0 to H'F3, H'F5 to H'FF).

- Changes in the purge instruction OCBP@Rn

  When Rn is designating an address in a non-cacheable area, this instruction is executed as NOP in the SH-4A with H'20-valued VER bits in the processor version register (PVR). In the SH-4A with extended functions, this instruction invalidates the operand cache line designated by way = Rn[14:13] and entry = Rn[12:5] provided that Rn[31:24] = H'F4 (OC address array area). In this process, writing back of the line takes place when the line to be invalidated is dirty. This operation is only executable in privileged mode, and an address error exception occurs in user mode. TLB-related exceptions do not occur.

  Do not execute this instruction to invalidate the memory-mapped array areas and control register areas for which Rn[31:24] is not H'F4, and their reserved areas (H'F0 to H'F3, H'F5 to H'FF).

- Changes in the write-back instruction OCBWB@Rn

  When Rn is designating an address in a non-cacheable area, this instruction is executed as NOP in the SH-4A with H'20-valued VER bits in the processor version register (PVR). In the SH-4A with extended functions, provided that Rn[31:24] = H'F4 (OC address array area), this instruction writes back the operand cache line designated by way = Rn[14:13] and entry = Rn[12:5] if it is dirty and clears the dirty bit to 0. This operation is only executable in privileged mode, and an address error exception occurs in user mode. TLB-related exceptions do not occur.

  Do not execute this instruction to invalidate the memory-mapped array areas and control register areas for which Rn[31:24] is not H'F4, and their reserved areas (H'F0 to H'F3, H'F5 to H'FF).

8.5.2 Prefetch Operation

This LSI supports a prefetch instruction to reduce the cache fill penalty incurred as the result of a cache miss. If it is known that a cache miss will result from a read or write operation, it is possible to fill the cache with data beforehand by means of the prefetch instruction to prevent a cache miss due to the read or write operation, and so improve software performance. If a prefetch instruction is executed for data already held in the cache, or if the prefetch address results in a UTLB miss or a protection violation, the result is no operation, and an exception is not generated. Details of the prefetch instruction are given in section 11, Instruction Descriptions of the SH-4A Extended Functions Software Manual.

- Prefetch instruction (OC) : PREF @Rn
- Prefetch instruction (IC) : PREFI @Rn

8.6 Memory-Mapped Cache Configuration

The IC and OC can be managed by software. The contents of IC data array can be read from or written to by a program in the P2 area by means of a MOV instruction in privileged mode. The contents of IC address array can also be read from or written to in privileged mode by a program in the P2 area or the IL memory area by means of a MOV instruction. Operation is not guaranteed if access is made from a program in another area. In this case, execute one of the following three methods for executing a branch to the P0, U0, P1, or P3 area.

1. Execute a branch using the RTE instruction.
2. Execute a branch to the P0, U0, P1, or P3 area after executing the ICBI instruction for any address (including non-cacheable area).
3. If the MC bit in IRMCR is 0 (initial value) before making an access to the memory-mapped IC, the specific instruction does not need to be executed. However, note that the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction after making an access to the memory-mapped IC.

Note that the method 3 may not be guaranteed in the future SuperH Series. Therefore, it is recommended that the method 1 or 2 should be used for being compatible with the future SuperH Series.

In privileged mode, the OC contents can be read from or written to by a program in the P1 or P2 area by means of a MOV instruction. Operation is not guaranteed if access is made from a program in another area. The IC and OC are allocated to the P4 area in the virtual address space. Only data accesses can be used on both the IC address array and data array and the OC address array and data array, and accesses are always longword-size. Instruction fetches cannot be performed in these areas. For reserved bits, a write value of 0 should be specified and the read value is undefined.

8.6.1 IC Address Array

The IC address array is allocated to addresses H'F000 0000 to H'F0FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the write tag and V bit are specified in the data field.

In the address field, bits [31:24] have the value H'F0 indicating the IC address array, and the way is specified by bits [14:13] and the entry by bits [12:5]. The association bit (A bit) [3] in the address field specifies whether or not association is performed when writing to the IC address array. As only longword access is used, 0 should be specified for address field bits [1:0].

In the data field, the tag is indicated by bits [31:10], and the V bit by bit [0]. As the IC address array tag is 19 bits in length, data field bits [31:29] are not used in the case of a write in which association is not performed. Data field bits [31:29] are used for the virtual address specification only in the case of a write in which association is performed.

The following three kinds of operation can be used on the IC address array:

1. IC address array read

   The tag and V bit are read into the data field from the IC entry corresponding to the way and entry set in the address field. In a read, associative operation is not performed regardless of whether the association bit specified in the address field is 1 or 0.
2. IC address array write (non-associative)

   The tag and V bit specified in the data field are written to the IC entry corresponding to the way and entry set in the address field. The A bit in the address field should be cleared to 0.
3. IC address array write (associative)

   When a write is performed with the A bit in the address field set to 1, the tag in each way stored in the entry specified in the address field is compared with the tag specified in the data field. The way numbers of bits [14:13] in the address field are not used. If the MMU is enabled at this time, comparison is performed after the virtual address specified by data field bits [31:10] has been translated to a physical address using the ITLB. If the addresses match and the V bit in the way is 1, the V bit specified in the data field is written into the IC entry. In other cases, no operation is performed. This operation is used to invalidate a specific IC entry. If an ITLB miss occurs during address translation, or the comparison shows a mismatch, an exception is not generated, no operation is performed, and the write is not executed.

Note: IC address array associative writing function may not be supported in the future SuperH Series. Therefore, it is recommended that the ICBI instruction should be used to operate the IC definitely by handling ITLB miss and reporting ITLB miss exception.

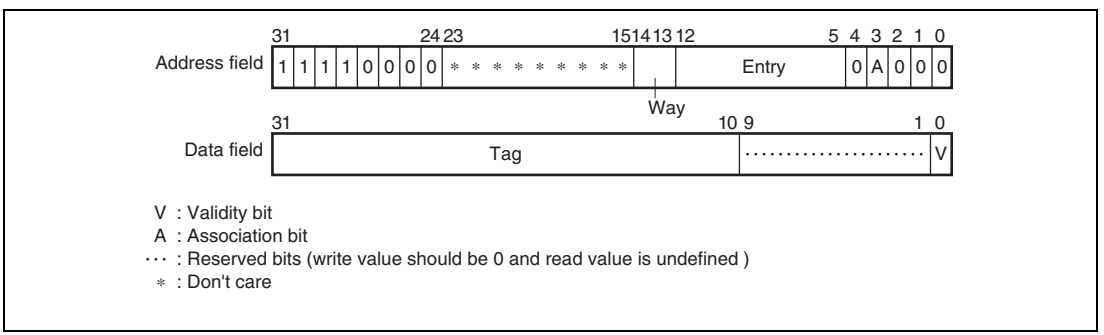


Figure 8.5 Memory-Mapped IC Address Array

8.6.2 IC Data Array

The IC data array is allocated to addresses H'F100 0000 to H'F1FF FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the longword data to be written is specified in the data field.

In the address field, bits [31:24] have the value H'F1 indicating the IC data array, and the way is specified by bits [14:13] and the entry by bits [12:5]. Address field bits [4:2] are used for the longword data specification in the entry. As only longword access is used, 0 should be specified for address field bits [1:0].

The data field is used for the longword data specification.

The following two kinds of operation can be used on the IC data array:

1. IC data array read

   Longword data is read into the data field from the data specified by the longword specification bits in the address field in the IC entry corresponding to the way and entry set in the address field.
2. IC data array write

   The longword data specified in the data field is written for the data specified by the longword specification bits in the address field in the IC entry corresponding to the way and entry set in the address field.

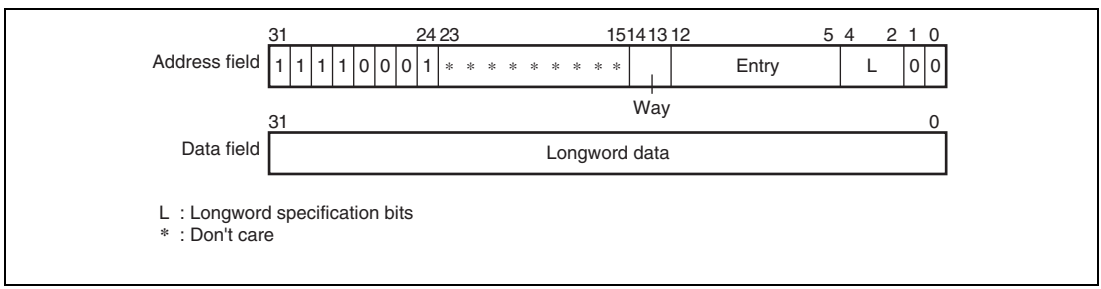


Figure 8.6 Memory-Mapped IC Data Array

8.6.3 OC Address Array

The OC address array is allocated to addresses H'F400 0000 to H'F4FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the write tag, U bit, and V bit are specified in the data field.

In the address field, bits [31:24] have the value H'F4 indicating the OC address array, and the way is specified by bits [14:13] and the entry by bits [12:5]. The association bit (A bit) [3] in the address field specifies whether or not association is performed when writing to the OC address array. As only longword access is used, 0 should be specified for address field bits [1:0].

In the data field, the tag is indicated by bits [31:10], the U bit by bit [1], and the V bit by bit [0]. As the OC address array tag is 19 bits in length, data field bits [31:29] are not used in the case of a write in which association is not performed. Data field bits [31:29] are used for the virtual address specification only in the case of a write in which association is performed.

The following three kinds of operation can be used on the OC address array:

1. OC address array read

   The tag, U bit, and V bit are read into the data field from the OC entry corresponding to the way and entry set in the address field. In a read, associative operation is not performed regardless of whether the association bit specified in the address field is 1 or 0.
2. OC address array write (non-associative)

   The tag, U bit, and V bit specified in the data field are written to the OC entry corresponding to the way and entry set in the address field. The A bit in the address field should be cleared to 0.

   When a write is performed to a cache line for which the U bit and V bit are both 1, after write-back of that cache line, the tag, U bit, and V bit specified in the data field are written.
3. OC address array write (associative)

   When a write is performed with the A bit in the address field set to 1, the tag in each way stored in the entry specified in the address field is compared with the tag specified in the data field. The way numbers of bits [14:13] in the address field are not used. If the MMU is enabled at this time, comparison is performed after the virtual address specified by data field bits [31:10] has been translated to a physical address using the UTLB. If the addresses match and the V bit in the way is 1, the U bit and V bit specified in the data field are written into the OC entry. In other cases, no operation is performed. This operation is used to invalidate a specific OC entry. If the OC entry U bit is 1, and 0 is written to the V bit or to the U bit, write-back is performed. If a UTLB miss occurs during address translation, or the comparison shows a mismatch, an exception is not generated, no operation is performed, and the write is not executed.

Note: OC address array associative writing function may not be supported in the future SuperH Series. Therefore, it is recommended that the OCBI, OCBP, or OCBWB instruction should be used to operate the OC definitely by reporting data TLB miss exception.

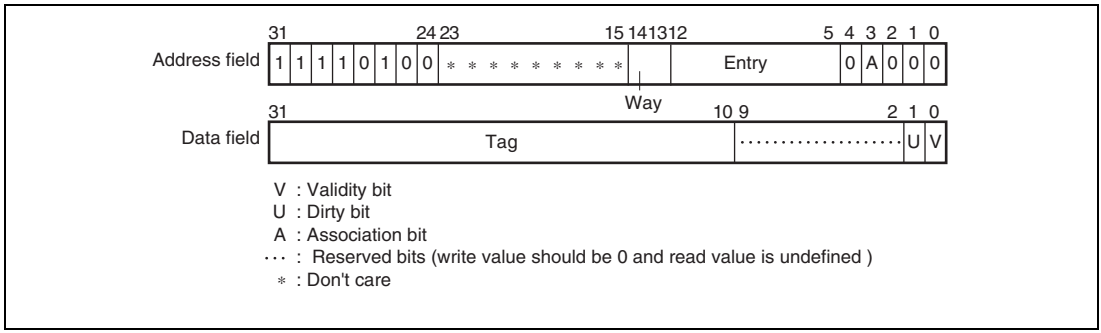


Figure 8.7 Memory-Mapped OC Address Array

8.6.4 OC Data Array

The OC data array is allocated to addresses H'F500 0000 to H'F5FF FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the longword data to be written is specified in the data field.

In the address field, bits [31:24] have the value H'F5 indicating the OC data array, and the way is specified by bits [14:13] and the entry by bits [12:5]. Address field bits [4:2] are used for the longword data specification in the entry. As only longword access is used, 0 should be specified for address field bits [1:0].

The data field is used for the longword data specification.

The following two kinds of operation can be used on the OC data array:

1. OC data array read

   Longword data is read into the data field from the data specified by the longword specification bits in the address field in the OC entry corresponding to the way and entry set in the address field.
2. OC data array write

   The longword data specified in the data field is written for the data specified by the longword specification bits in the address field in the OC entry corresponding to the way and entry set in the address field. This write does not set the U bit to 1 on the address array side.

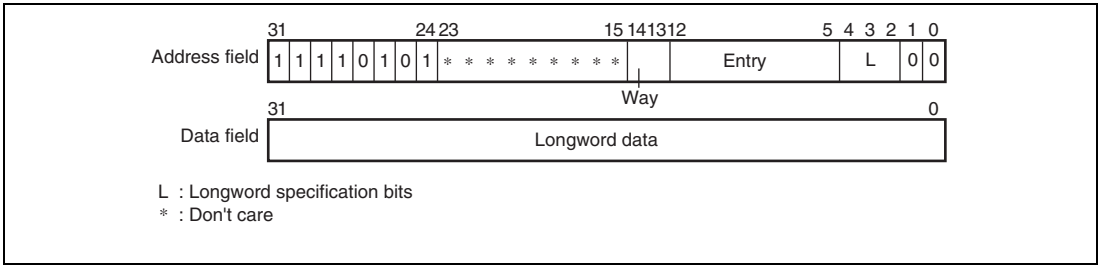


Figure 8.8 Memory-Mapped OC Data Array

8.6.5 Memory-Mapped Cache Associative Write Operation

Associative writing to the IC and OC address arrays may not be supported in future SuperH-family products. The use of instructions ICBI, OCBI, OCBP, and OCBWB is recommended. These instructions handle ITLB misses, and notify instruction TLB miss exceptions and data TLB miss exceptions, thus providing a sure way of controlling the IC and OC. As a transitional measure, this LSI generates address errors when this function is used. If compatibility with previous products is a crucial consideration, on the other hand, the MMCAW bit in EXPMASK (H'FF2F 0004) can be set to 1 to enable this function. However, instructions ICBI, OCBI, OCBP, and OCBWB should be used to guarantee compatibility with future SuperH-family products.

8.7 Store Queues

This LSI supports two 32-byte store queues (SQs) to perform high-speed writes to external memory.

8.7.1 SQ Configuration

There are two 32-byte store queues, SQ0 and SQ1, as shown in figure 8.9. These two store queues can be set independently.

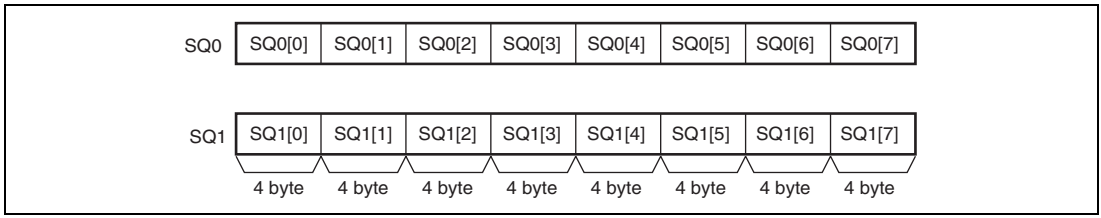


Figure 8.9 Store Queue Configuration

8.7.2 Writing to SQ

A write to the SQs can be performed using a store instruction for addresses H'E000 0000 to H'E3FF FFFC in the P4 area. A longword or quadword access size can be used. The meanings of the address bits are as follows:

[31:26]	: 111000	Store queue specification
[25:6]	: Don't care	Used for external memory transfer/access right
[5]	: 0/1	0: SQ0 specification
		1: SQ1 specification
[4:2]	: LW specification	Specifies longword position in SQ0/SQ1
[1:0]	: 00	Fixed at 0

8.7.3 Transfer to External Memory

Transfer from the SQs to external memory can be performed with a prefetch instruction (PREF). Issuing a PREF instruction for addresses H'E000 0000 to H'E3FF FFFC in the P4 area starts a transfer from the SQs to external memory. The transfer length is fixed at 32 bytes, and the start address is always at a 32-byte boundary. While the contents of one SQ are being transferred to external memory, the other SQ can be written to without a penalty cycle. However, writing to the SQ involved in the transfer to external memory is kept waiting until the transfer is completed.

The physical address bits [28:0] of the SQ transfer destination are specified as shown below, according to whether the MMU is enabled or disabled.

- When MMU is enabled (AT = 1 in MMUCR)

  The SQ area (H'E000 0000 to H'E3FF FFFF) is set in VPN of the UTLB, and the transfer destination physical address in PPN. The ASID, V, SZ, SH, PR, and D bits have the same meaning as for normal address translation, but the C and WT bits have no meaning with regard to this page. When a prefetch instruction is issued for the SQ area, address translation is performed and physical address bits [28:10] are generated in accordance with the SZ bit specification. For physical address bits [9:5], the address prior to address translation is generated in the same way as when the MMU is disabled. Physical address bits [4:0] are fixed at 0. Transfer from the SQs to external memory is performed to this address.
- When MMU is disabled (AT = 0 in MMUCR)

  The SQ area (H'E000 0000 to H'E3FF FFFF) is specified as the address at which a PREF instruction is issued. The meanings of address bits [31:0] are as follows:

[31:26]	: 111000	Store queue specification
[25:6]	: Address	Transfer destination physical address bits [25:6]
[5]	: 0/1	0: SQ0 specification 1: SQ1 specification and transfer destination physical address bit [5]
[4:2]	: Don't care	No meaning in a prefetch
[1:0]	: 00	Fixed at 0

  Physical address bits [28:26], which cannot be generated from the above address, are generated from QACR0 and QACR1.

  QACR0[4:2] : Physical address bits [28:26] corresponding to SQ0

  QACR1[4:2] : Physical address bits [28:26] corresponding to SQ1

Physical address bits [4:0] are always fixed at 0 since burst transfer starts at a 32-byte boundary.

8.7.4 Determination of SQ Access Exception

Determination of an exception in a write to an SQ or transfer to external memory (PREF instruction) is performed as follows according to whether the MMU is enabled or disabled. If an exception occurs during a write to an SQ, the SQ contents before the write are retained. If an exception occurs in a data transfer from an SQ to external memory, the transfer to external memory will be aborted.

- When MMU is enabled (AT = 1 in MMUCR)

  Operation is in accordance with the address translation information recorded in the UTLB, and the SQMD bit in MMUCR. Write type exception judgment is performed for writes to the SQs, and read type exception judgment for transfer from the SQs to external memory (using a PREF instruction). As a result, a TLB miss exception or protection violation exception is generated as required. However, if SQ access is enabled in privileged mode only by the SQMD bit in MMUCR, an address error will occur even if address translation is successful in user mode.

- When MMU is disabled (AT = 0 in MMUCR)

  Operation is in accordance with the SQMD bit in MMUCR.

  0: Privileged/user mode access possible

  1: Privileged mode access possible

  If the SQ area is accessed in user mode when the SQMD bit in MMUCR is set to 1, an address error will occur.

8.7.5 Reading from SQ

In privileged mode in this LSI, reading the contents of the SQs may be performed by means of a load instruction for addresses H'FF00 1000 to H'FF00 103C in the P4 area. Only longword access is possible.

[31:6]	: H'FF00 1000	Store queue specification
[5]	: 0/1	0: SQ0 specification
		1: SQ1 specification
[4:2]	: LW specification	Specifies longword position in SQ0/SQ1
[1:0]	: 00	Fixed at 0

Section 9 On-Chip Memory

This LSI includes the IL memory which is suitable for instruction storage.

9.1 Features

(1) IL Memory

- Capacity
  The IL memory in this LSI is 16 Kbytes.
- Page
  The IL memory is divided into four pages (pages 0, 1, 2, and 3).
- Memory map
  The IL memory is allocated to the addresses shown in table 9.1 in both the virtual address space and the physical address space.

Table 9.1 IL Memory Addresses

	Memory Size
Page	**16 Kbytes**
Page 0	H'E520 0000 to H'E520 0FFF
Page 1	H'E520 1000 to H'E520 1FFF
Page 2	H'E520 2000 to H'E520 2FFF
Page 3	H'E520 3000 to H'E520 3FFF

- Ports
  The page has three independent read/write ports and is connected to the SuperHyway bus, the cache/RAM internal bus, and the instruction bus. The instruction bus is used when the IL memory is accessed through instruction fetch. The cache/RAM internal bus is used when the IL memory is accessed through operand access. The SuperHyway bus is used for IL memory access from the SuperHyway bus master module.
- Priority
  In the event of simultaneous accesses to the same page from different buses, the access requests are processed according to priority. The priority order is: SuperHyway bus > cache/RAM internal bus > instruction bus.

9.2 Register Descriptions

The following register is related to the on-chip memory.

Table 9.2 Register Configuration

Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Access Size
On-chip memory control register	RAMCR	R/W	H'FF00 0074	H'1F00 0074	32

Note: * The P4 address is the address used when using P4 area in the virtual address space. The area 7 address is the address used when accessing from area 7 in the physical address space using the TLB.

Table 9.3 Register States in Each Processing Mode

Name	Abbreviation	Power-On Reset	Sleep	Standby
On-chip memory control register	RAMCR	H'0000 0000	Retained	Retained

9.2.1 On-Chip Memory Control Register (RAMCR)

RAMCR controls the protective functions in the on-chip memory.

When updating RAMCR, please follow limitation described at section 8.2.4, On-Chip Memory Control Register (RAMCR).

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RMD	RP	IC2W	OC2W	ICWPD	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31to10	—	All 0	R	Reserved For read/write in these bits, refer to General Precautions on Handling of Product.

Bit	Bit Name	Initial Value	R/W	Description
9	RMD	0	R/W	On-Chip Memory Access Mode Specifies the right of access to the on-chip memory from the virtual address space. 0: An access in privileged mode is allowed. (An address error exception occurs in user mode.) 1: An access in user/ privileged mode is allowed.
8	RP	0	R/W	On-Chip Memory Protection Enable Selects whether or not to use the protective functions using ITLB and UTLB for accessing the on-chip memory from the virtual address space. 0: Protective functions are not used. 1: Protective functions are used. For further details, refer to section 9.4, On-Chip Memory Protective Functions.
7	IC2W	0	R/W	IC Two-Way Mode For further details, refer to section 8.4.3, IC Two-Way Mode.
6	OC2W	0	R/W	OC Two-Way Mode For further details, refer to section 8.3.6, OC Two-Way Mode.
5	ICWPD	0	R/W	IC Way Prediction Disable For further details, refer to section 8.4.4, Instruction Cache Way Prediction Operation.
4 to 0	—	All 0	R	Reserved For read/write in these bits, refer to General Precautions on Handling of Product.

9.3 Operation

9.3.1 Instruction Fetch Access from the CPU

Instruction fetch access from the CPU is performed directly via the instruction bus for a given virtual address. In the case of successive accesses to the same page of IL memory and as long as no page conflict occurs, the access takes one cycle.

9.3.2 Operand Access from the CPU and Access from the FPU

Note: Operand access is applied for PC relative access (@(disp,pc)).

Operand access from the CPU and access from the FPU are performed via the cache/RAM internal bus. Access via the cache/RAM internal bus takes more than one cycle.

9.3.3 Access from the SuperHyway Bus Master Module

On-chip memory is always accessed by the SuperHyway bus master module, such as DMAC, via the SuperHyway bus which is a physical address bus. The same addresses as for the virtual addresses must be used.

9.4 On-Chip Memory Protective Functions

This LSI implements the following protective functions to the on-chip memory by using the on-chip memory access mode bit (RMD) and the on-chip memory protection enable bit (RP) in the on-chip memory control register (RAMCR).

- Protective functions for access from the CPU and FPU

  When RAMCR.RMD = 0, and the on-chip memory is accessed in user mode, it is determined to be an address error exception.

  When MMUCR.AT = 1 and RAMCR.RP = 1, MMU exception and address error exception are checked in the on-chip memory area which is a part of area P4 as with the area P0/P3/U0.

The above descriptions are summarized in table 9.4.

Table 9.4 Protective Function Exceptions to Access On-Chip Memory

MMUCR.AT	RAMCR.RP	SR.MD	RAMCR. RMD	Always Occurring Exceptions	Possibly Occurring Exceptions
0	x	0	0	Address error exception	—
			1	—	—
		1	x	—	—
1	0	0	0	Address error exception	—
			1	—	—
		1	x	—	—
	1	0	0	Address error exception	—
			1	—	MMU exception
		1	x	—	MMU exception

[Legend] x: Don't care

9.5 Usage Notes

9.5.1 Page Conflict

In the event of simultaneous access to the same page from different buses, page conflict occurs. Although each access is completed correctly, this kind of conflict tends to lower On-chip memory accessibility. Therefore it is advisable to provide all possible preventative software measures. For example, conflicts will not occur if each bus accesses different pages.

9.5.2 Access Across Different Pages

Access from the instruction bus is performed in one cycle when the access is made successively to the same page but takes multiple cycles (a maximum of two wait cycles may be required) when the access is made across pages or the previous access was made to memory other than IL memory. For this reason, from the viewpoint of performance optimization, it is recommended to design the software such that the target page does not change so often in access from the instruction bus. For example, allocating a separate program for each page will deliver better efficiency.

9.5.3 On-Chip Memory Coherency

In order to allocate instructions in the IL memory, write an instruction to the IL memory, execute the following sequence, then branch to the rewritten instruction.

- SYNCO
- ICBI @Rn

In this case, the target for the ICBI instruction can be any address (IL memory address may be possible) within the range where no address error exception occurs, and cache hit/miss is possible.

9.5.4 Sleep Mode

The SuperHyway bus master module, such as DMAC, cannot access OL memory and IL memory in sleep mode.

Section 10 Interrupt Controller (INTC)

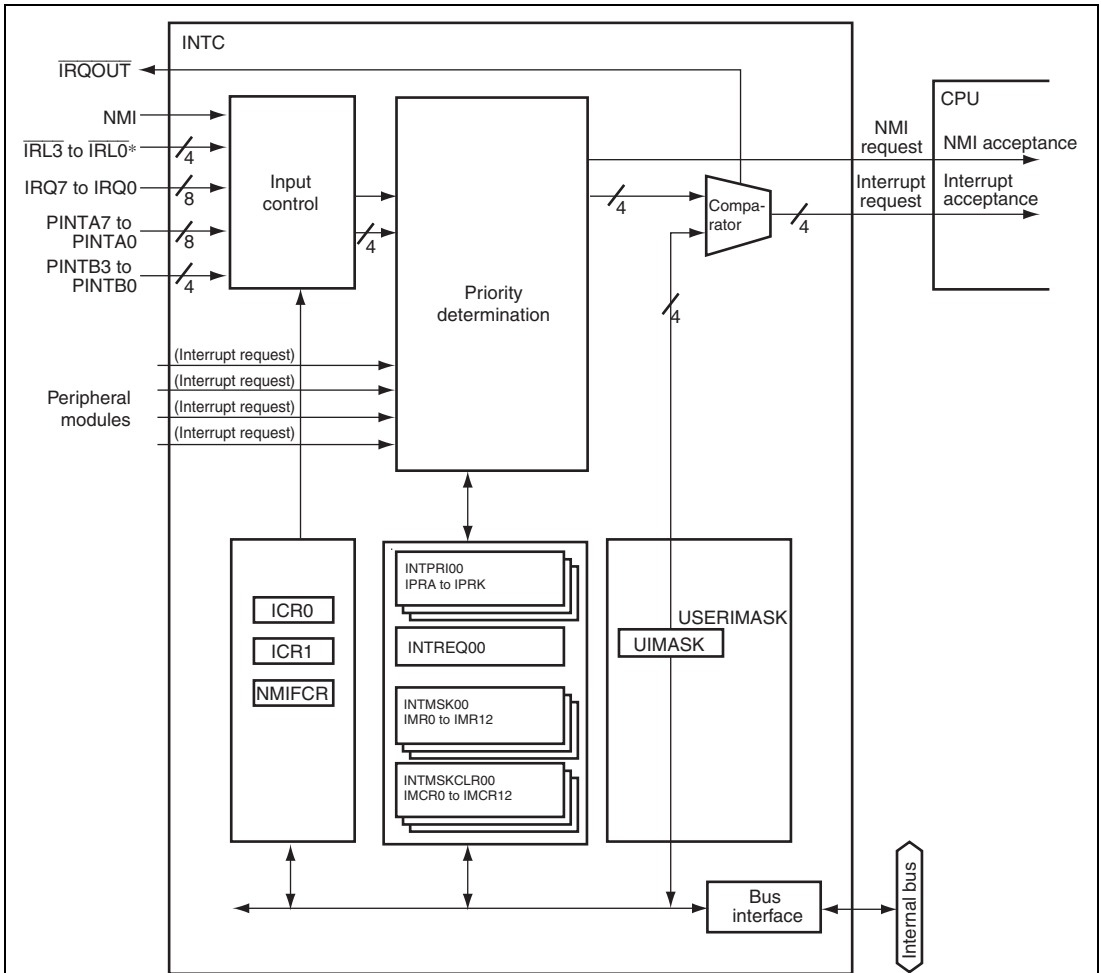
The interrupt controller (INTC) determines the priority of interrupt sources and controls interrupt requests to the CPU. Some INTC registers set the priority of each interrupt and interrupt requests are processed according to the user-set priority.

10.1 Features

The INTC has the following features.

- Fifteen levels of interrupt priority can be set.
  By setting the interrupt priority registers, the priorities of on-chip peripheral module interrupts can be selected from 15 levels for individual request sources.
- NMI noise canceler function
  An NMI input-level bit indicates the NMI pin state. By reading this bit in the interrupt exception handling routine, the pin state can be checked, enabling it to be used as a noise canceler.
- NMI request masking when the block bit (BL) in the status register (SR) is set to 1
  Whether to mask NMI requests when the BL bit in SR is set to 1 can be selected.
- User-mode interrupt disabling function
  Specifying an interrupt mask level in the user interrupt mask level register (USERIMASK) disables interrupts which are not higher in priority than the specified mask level in user mode.

Figure 10.1 shows a block diagram of the INTC.



[Legend]

ICR0, ICR1:	Interrupt control registers 0, 1
NMIFCR:	NMI flag control register
INTPRI00, IPRA to IPRK:	Interrupt priority registers
INTREQ00:	Interrupt request register
INTMSK00, IMR0 to IMR12:	Interrupt mask registers
INTMSKCLR00, IMCR0 to IMCR12:	Interrupt mask clear registers
USERIMASK:	User interrupt mask level register

Note: * IRQ3 to IRQ0 and $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ pin functions are multiplexed on the same pins, so the functions cannot be used at the same time.

Figure 10.1 Block Diagram of INTC

10.2 Input/Output Pins

Table 10.1 shows the INTC pin configuration.

Table 10.1 Pin Configuration

Pin Name	Function	I/O	Description
NMI	Non maskable interrupt input pin	Input	Interrupt request signal that is not maskable
IRQ7 to IRQ0, $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$*1	External interrupt input pins	Input	Inputs of interrupt request signals
$\overline{\text{IRQOUT}}$*2	Interrupt request output pin	Output	Signal indicating that an interrupt request has been generated.
PINTA7 to PINTA0, PINTB3 to PINTB0	Port-interrupt input pins	Input	Inputs of port interrupt request signals

Notes: 1. IRQ3 to IRQ0 and $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ pin functions are multiplexed on the same pins, so the functions cannot be used at the same time.
2. $\overline{\text{IRQOUT}}$ is multiplexed with $\overline{\text{REFOUT}}$ (refresh request signal on bus release).

10.3 Register Descriptions

Table 10.2 shows the INTC register configuration. Table 10.3 shows the register states in each operating mode.

Table 10.2 Register Configuration

Register	Abbreviation	R/W	Address	Access Size
Interrupt control register 0	ICR0	R/W	H'A414 0000	16
Interrupt control register 1	ICR1	R/W	H'A414 001C	16
Interrupt priority register 00	INTPRI00	R/W	H'A414 0010	32
Interrupt request register 00	INTREQ00	R/W	H'A414 0024	8
Interrupt mask register 00	INTMSK00	R/W	H'A414 0044	8
Interrupt mask clear register 00	INTMSKCLR00	W	H'A414 0064	8
NMI flag control register	NMIFCR	R/W	H'A414 00C0	16
User interrupt mask level register	USERIMSK	R/W	H'A470 0000	32
Interrupt priority register A	IPRA	R/W	H'A408 0000	16
Interrupt priority register B	IPRB	R/W	H'A408 0004	16
Interrupt priority register C	IPRC	R/W	H'A408 0008	16

Register	Abbreviation	R/W	Address	Access Size
Interrupt priority register D	IPRD	R/W	H'A408 000C	16
Interrupt priority register E	IPRE	R/W	H'A408 0010	16
Interrupt priority register F	IPRF	R/W	H'A408 0014	16
Interrupt priority register G	IPRG	R/W	H'A408 0018	16
Interrupt priority register H	IPRH	R/W	H'A408 001C	16
Interrupt priority register I	IPRI	R/W	H'A408 0020	16
Interrupt priority register J	IPRJ	R/W	H'A408 0024	16
Interrupt priority register K	IPRK	R/W	H'A408 0028	16
Interrupt mask register 0	IMR0	R/W	H'A408 0080	8
Interrupt mask register 1	IMR1	R/W	H'A408 0084	8
Interrupt mask register 2	IMR2	R/W	H'A408 0088	8
Interrupt mask register 3	IMR3	R/W	H'A408 008C	8
Interrupt mask register 4	IMR4	R/W	H'A408 0090	8
Interrupt mask register 5	IMR5	R/W	H'A408 0094	8
Interrupt mask register 6	IMR6	R/W	H'A408 0098	8
Interrupt mask register 7	IMR7	R/W	H'A408 009C	8
Interrupt mask register 8	IMR8	R/W	H'A408 00A0	8
Interrupt mask register 9	IMR9	R/W	H'A408 00A4	8
Interrupt mask register 10	IMR10	R/W	H'A408 00A8	8
Interrupt mask register 11	IMR11	R/W	H'A408 00AC	8
Interrupt mask register 12	IMR12	R/W	H'A408 00B0	8
Interrupt mask clear register 0	IMCR0	W	H'A408 00C0	8
Interrupt mask clear register 1	IMCR1	W	H'A408 00C4	8
Interrupt mask clear register 2	IMCR2	W	H'A408 00C8	8
Interrupt mask clear register 3	IMCR3	W	H'A408 00CC	8
Interrupt mask clear register 4	IMCR4	W	H'A408 00D0	8
Interrupt mask clear register 5	IMCR5	W	H'A408 00D4	8
Interrupt mask clear register 6	IMCR6	W	H'A408 00D8	8
Interrupt mask clear register 7	IMCR7	W	H'A408 00DC	8
Interrupt mask clear register 8	IMCR8	W	H'A408 00E0	8
Interrupt mask clear register 9	IMCR9	W	H'A408 00E4	8
Interrupt mask clear register 10	IMCR10	W	H'A408 00E8	8
Interrupt mask clear register 11	IMCR11	W	H'A408 00EC	8
Interrupt mask clear register 12	IMCR12	W	H'A408 00F0	8

Table 10.3 Register States in Each Operating Mode

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
ICR0	Initialized	Retained	Retained	Retained
ICR1	Initialized	Retained	Retained	Retained
INTPRI00	Initialized	Retained	Retained	Retained
INTREQ00	Initialized	Retained	Retained	Retained
INTMSK00	Initialized	Retained	Retained	Retained
INTMSKCLR00	Initialized	Retained	Retained	Retained
NMIFCR	Initialized	Retained	Retained	Retained
USERIMASK	Initialized	Retained	Retained	Retained
IPRA	Initialized	Retained	Retained	Retained
IPRB	Initialized	Retained	Retained	Retained
IPRC	Initialized	Retained	Retained	Retained
IPRD	Initialized	Retained	Retained	Retained
IPRE	Initialized	Retained	Retained	Retained
IPRF	Initialized	Retained	Retained	Retained
IPRG	Initialized	Retained	Retained	Retained
IPRH	Initialized	Retained	Retained	Retained
IPRI	Initialized	Retained	Retained	Retained
IPRJ	Initialized	Retained	Retained	Retained
IPRK	Initialized	Retained	Retained	Retained
IMR0	Initialized	Retained	Retained	Retained
IMR1	Initialized	Retained	Retained	Retained
IMR2	Initialized	Retained	Retained	Retained
IMR3	Initialized	Retained	Retained	Retained
IMR4	Initialized	Retained	Retained	Retained
IMR5	Initialized	Retained	Retained	Retained
IMR6	Initialized	Retained	Retained	Retained
IMR7	Initialized	Retained	Retained	Retained
IMR8	Initialized	Retained	Retained	Retained

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
IMR9	Initialized	Retained	Retained	Retained
IMR10	Initialized	Retained	Retained	Retained
IMR11	Initialized	Retained	Retained	Retained
IMR12	Initialized	Retained	Retained	Retained
IMCR0	Initialized	Retained	Retained	Retained
IMCR1	Initialized	Retained	Retained	Retained
IMCR2	Initialized	Retained	Retained	Retained
IMCR3	Initialized	Retained	Retained	Retained
IMCR4	Initialized	Retained	Retained	Retained
IMCR5	Initialized	Retained	Retained	Retained
IMCR6	Initialized	Retained	Retained	Retained
IMCR7	Initialized	Retained	Retained	Retained
IMCR8	Initialized	Retained	Retained	Retained
IMCR9	Initialized	Retained	Retained	Retained
IMCR10	Initialized	Retained	Retained	Retained
IMCR11	Initialized	Retained	Retained	Retained
IMCR12	Initialized	Retained	Retained	Retained

10.3.1 Interrupt Control Register 0 (ICR0)

ICR0 sets the input signal detection mode for the external interrupt input pin NMI, IRQ, IRL, and PINT, and indicates the input signal level at the NMI pin.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMIL	MAI	—	—	—	—	NMIB	NMIE	IRLM	IRLM2	LSH	—	—	—	—	—
Initial value:	0/1*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	0/1*	R	NMI Input Level Indicates the level of the signal input at the NMI pin. This bit can be read to determine the NMI pin level. This bit cannot be modified. 0: NMI input level is low 1: NMI input level is high
14	MAI	0	R/W	NMI Interrupt Mask Selects whether to mask all interrupts while the NMI input level is low regardless of the BL bit setting in SR. 0: Enables interrupts while the NMI input level is low 1: Disables interrupts while the NMI input level is low
13 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	NMIB	0	R/W	NMI Block Mode Selects whether to detect the NMI interrupt immediately or keep it pending until the BL bit in SR is cleared to 0 if the NMI interrupt is input while the BL bit is set to 1. 0: Keeps the NMI interrupt pending while the BL bit in SR is set to 1 1: Detects the NMI interrupt even while the BL bit in SR is set to 1
8	NMIE	0	R/W	NMI Edge Select Selects whether the falling or rising edge of the interrupt request signal at the NMI pin is detected. 0: Interrupt request is detected on falling edge of NMI input 1: Interrupt request is detected on rising edge of NMI input

Bit	Bit Name	Initial Value	R/W	Description
7	IRLM	0	R/W	IRL Pin Mode Selects whether IRQ3/$\overline{\text{IRL3}}$ to IRQ0/$\overline{\text{IRL0}}$ are used as four independent interrupts (IRQ3 to IRQ0) or as 15-level encoded interrupt requests (levels of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ are encoded as H'F to H'1). 0: Used as pins for 15-level encoded interrupts, i.e.$\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$. 1: Used as pins for 4 independent interrupt requests, i.e. IRQ3 to IRQ0.
6	IRLM2	0	R/W	IRL Pin Mode2 This bit sets the IRQ7 to IRQ4 interrupts. 0: Initial value 1: The IRQ7 to IRQ4 interrupts are in use. When the IRQ7 to IRQ4 interrupts are in use, set this bit to 1. When this bit is set to 0, IRQ7 to IRQ4 pin functions should not be selected by setting the Pin Function Controller (PFC). If not, an interrupt that is not intended may occur. These 4 pins, IRQ7 to IRQ4, cannot be used as pins for 15-level encoded interrupts.
5	LSH	0	R/W	Holding function in level detection In level-detection of the IRQ, IRL or PINT interrupts, selects whether or not the interrupt requests are held by the detection circuit. 0: Held 1: Not held When the IRQ interrupt are in use with level sensing and when IRL or PINT interrupts are in use, ordinarily set the LSH bit to 1. Setting the LSH bit to 0 means that even if the external interrupt signal is negated, generation of the interrupt will still indicated within the LSI.
4 to 0	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.

Note: * This bit is set to 1 when the NMI input is at the high level and cleared to 0 when it is at the low level.

10.3.2 Interrupt Control Register 1 (ICR1)

ICR1 specifies the detection mode for the external interrupt input pins IRQ7 to IRQ0 individually: rising edge, falling edge, low level, or high level.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ0S		IRQ1S		IRQ2S		IRQ3S		IRQ4S		IRQ5S		IRQ6S		IRQ7S	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	IRQ0S	00	R/W	IRQn Sense Select These bits select whether interrupt request signals corresponding to pins IRQ7 to IRQ0 are detected by a rising edge, falling edge, low level, or high level.
13, 12	IRQ1S	00	R/W	
11, 10	IRQ2S	00	R/W	
9, 8	IRQ3S	00	R/W	
7, 6	IRQ4S	00	R/W	
5, 4	IRQ5S	00	R/W	
3, 2	IRQ6S	00	R/W	
1, 0	IRQ7S	00	R/W	

IRQnS	Detection Mode
00	Interrupt request is detected on falling edge of IRQn input
01	Interrupt request is detected on rising edge of IRQn input
10	Interrupt request is detected on low level of IRQn input
11	Interrupt request is detected on high level of IRQn input

[Legend] n = 0 to 7

10.3.3 Interrupt Priority Register 00 (INTPRI00)

INTPRI00 is a 32-bit register that specifies priority levels from 15 to 0 for the external interrupt input pins IRQ7 to IRQ0.

Each 4-bit group is set with a value from H'F (1111) to H'0 (0000) to specify the interrupt priority level for the corresponding interrupt. Setting H'F means priority level 15 (the highest level); H'0 means priority level 0 (masking is requested).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IRQ0				IRQ1				IRQ2				IRQ3			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

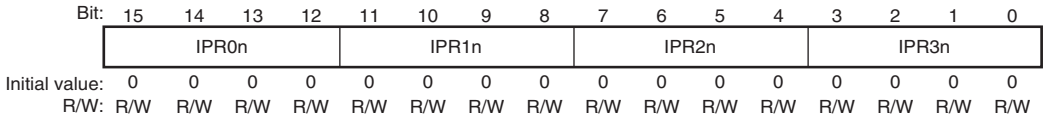
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ4				IRQ5				IRQ6				IRQ7			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	IRQ0	H'0	R/W	These bits set the priority level for each interrupt source in 4-bit units.
27 to 24	IRQ1	H'0	R/W	
23 to 20	IRQ2	H'0	R/W	
19 to 16	IRQ3	H'0	R/W	
15 to 12	IRQ4	H'0	R/W	
11 to 8	IRQ5	H'0	R/W	
7 to 4	IRQ6	H'0	R/W	
3 to 0	IRQ7	H'0	R/W	

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

IPRA to IPRK are 16-bit registers that specify priority levels from 15 to 0 for on-chip peripheral module interrupts.

On-chip peripheral module interrupts are assigned to four 4-bit groups in each register. These 4-bit groups are set with values from H'F (1111) to H'0 (0000) to specify the interrupt priority level for the corresponding interrupt. Setting H'F means priority level 15 (the highest level); H'0 means priority level 0 (interrupt request is masked).



Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IPR0n				IPR1n				IPR2n				IPR3n			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	IPR0n	H'0	R/W	These bits set the priority level for each interrupt source in 4-bit units. For details, see table 10.4.
11 to 8	IPR1n	H'0	R/W	
7 to 4	IPR2n	H'0	R/W	
3 to 0	IPR3n	H'0	R/W	

Table 10.4 Interrupt Sources and IPRA to IPRK

Register	IPR0n	IPR1n	IPR2n	IPR3n
IPRA	TMU0	TMU1	TMU2	RTC
IPRB	—	—	SIM	—
IPRC	—	—	—	—
IPRD	PINTA0 to PINTA7	PINTB0 to PINTB3	IrDA0	IrDA1
IPRE	DMAC (1)	DMAC (2)	—	ADC
IPRF	—	—	—	CMT
IPRG	SCIF0	SCIF1	SCIF2	SCIF3
IPRH	SCIF4	SCIF5	—	IIC0
IPRI	SIOF	—	—	IIC1
IPRJ	—	—	—	—
IPRK	TPU0	TPU1	—	—

Note: —: Reserved. An undefined value will be read. The write value should always be 0.

10.3.5 Interrupt Request Register 00 (INTREQ00)

INTREQ00 is an 8-bit register that indicates interrupt requests from external input pins IRQ7 to IRQ0. This register value is not affected by interrupt mask with the INTPRI00 or INTMSK00 settings.

When edge-detection mode is set for an IRQ pin (ICR1.IRQnS = B'00 or B'01), an interrupt request is cleared by writing 0 to the corresponding IRQn bit after reading IRQn = 1.

Bit:	7	6	5	4	3	2	1	0
	IRQ0	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6	IRQ7
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ0	0	R/W	Edge detection (IRQnS in ICR1 set to B'00 or B'01) Flag indicating detection of IRQn interrupt • When reading 0: No interrupt request detected 1: Interrupt request detected • When writing 0: The bit is cleared to 0 only if it was previously read as 1. 1: Writing 1 is ignored. Write 1 to the bits other than the bit to be cleared. Note: Write 1 to all bits you do not wish to clear to 0. Level detection (IRQnS in ICR1 set to B'10 or B'11) [LSH in ICR0 set to 1] Indicates whether or not a valid interrupt request is being input to the IRQn pin. • When writing 0: No interrupt request being input 1: Interrupt request being input • Writing to these bits is ignored. [LSH in ICR0 cleared to 0] Flag indicating detection of IRQn interrupt • When reading 0: No interrupt request detected 1: Interrupt request detected • Writing to these bits is ignored.
6	IRQ1	0	R/W	
5	IRQ2	0	R/W	
4	IRQ3	0	R/W	
3	IRQ4	0	R/W	
2	IRQ5	0	R/W	
1	IRQ6	0	R/W	
0	IRQ7	0	R/W	

[Legend]
n = 0 to 7

The methods of clearing the bits in this register are as follows.

1. Edge detection

   The interrupt source can be cleared by writing 0 to the corresponding bit after reading it as 1. In this case, write 1 to the bits you do not wish to clear to 0.

2. Level detection (LSH in ICR0 set to 1)

   The corresponding bit is cleared to 0 automatically when the IRQ pin state changes and the interrupt request is negated. It is not necessary to clear the bit in software.

3. Level detection (LSH in ICR0 cleared to 0)

   After the IRQ pin state changes and the interrupt request is negated, write 1 to the corresponding bit in the INTMSK00 register.

10.3.6 Interrupt Mask Register 00 (INTMSK00)

INTMSK00 is an 8-bit register that masks interrupt requests from external interrupt input pins IRQ7 to IRQ0.

To clear an interrupt mask, write 1 to the corresponding bit in INTMSKCLR00. Writing 0 to the corresponding bit in INTMSK00 does not affect the bit value.

When IRL interrupts in IRL3 to IRL0 pins are in use by setting ICR0.IRLM to 0, the interrupt requests from IRQ3 to IRQ0 pins should be masked by setting IRQ3 to IRQ0 bits of this register to 1.

Bit:	7	6	5	4	3	2	1	0
	IRQ0	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6	IRQ7
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ0	0	R/W	IRQn Interrupt Mask
6	IRQ1	0	R/W	0: The corresponding interrupt is not masked
5	IRQ2	0	R/W	1: The corresponding interrupt is masked
4	IRQ3	0	R/W	
3	IRQ4	0	R/W	
2	IRQ5	0	R/W	
1	IRQ6	0	R/W	
0	IRQ7	0	R/W	

[Legend]
n = 7 to 0

10.3.7 Interrupt Mask Clear Register 00 (INTMSKCLR00)

INTMSKCLR00 is an 8-bit write-only register that clears the mask settings for interrupts from external interrupt input pins IRQ7 to IRQ0.

Bit:	7	6	5	4	3	2	1	0
	IRQ0	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6	IRQ7
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ0	0	W	IRQn Interrupt Mask Clear
6	IRQ1	0	W	0: Writing 0 is ignored
5	IRQ2	0	W	1: Clears the corresponding interrupt mask
4	IRQ3	0	W	
3	IRQ4	0	W	
2	IRQ5	0	W	
1	IRQ6	0	W	
0	IRQ7	0	W	

[Legend]
n = 7 to 0

10.3.8 Interrupt Mask Registers 0 to 12 (IMR0 to IMR12)

IMR0 to IMR12 are 8-bit registers that mask on-chip peripheral module interrupts. To mask an interrupt, write 1 to the corresponding bit in IMR0 to IMR12.

To clear an interrupt mask, write 1 to the corresponding bit in IMCR0 to IMCR12. Writing 0 to the corresponding bit in IMR0 to IMR12 does not affect the bit value.

Table 10.5 shows the relationship between IMR0 to IMR12 and each interrupt source.

Bit:	7	6	5	4	3	2	1	0
	IMRn0	IMRn1	IMRn2	IMRn3	IMRn4	IMRn5	IMRn6	IMRn7
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	IMRn0	0	R/W	Interrupt Mask
6	IMRn1	0	R/W	Masks the interrupt request corresponding to each bit. See table 10.5 for the relationship between IMR and each interrupt source.
5	IMRn2	0	R/W	
4	IMRn3	0	R/W	When writing:
3	IMRn4	0	R/W	0: Writing 0 is ignored
2	IMRn5	0	R/W	1: Masks the corresponding interrupt request
1	IMRn6	0	R/W	When reading:
0	IMRn7	0	R/W	0: The corresponding interrupt request is not masked
				1: The corresponding interrupt request is masked

[Legend]
n = 0 to 12

10.3.9 Interrupt Mask Clear Registers 0 to 12 (IMCR0 to IMCR12)

IMCR0 to IMCR12 are 8-bit write-only registers that clear the mask settings for the on-chip peripheral module interrupts. Table 10.5 shows the correspondence between individual bits in IMCR0 to IMCR12 and interrupt sources.

Bit:	7	6	5	4	3	2	1	0
	IMCRn0	IMCRn1	IMCRn2	IMCRn3	IMCRn4	IMCRn5	IMCRn6	IMCRn7
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7	IMCRn0	0	W	Interrupt Mask Clear
6	IMCRn1	0	W	Clears the mask setting for the interrupt corresponding to each bit. See table 10.5 for the relationship between IMCR and each interrupt source.
5	IMCRn2	0	W	
4	IMCRn3	0	W	When writing:
3	IMCRn4	0	W	0: Writing 0 is ignored
2	IMCRn5	0	W	1: Clears the corresponding interrupt mask
1	IMCRn6	0	W	When reading:
0	IMCRn7	0	W	An undefined value will be read

[Legend]
n = 0 to 12

Table 10.5 Correspondence between On-Chip Peripheral Module Interrupt Sources and IMR0 to IMR12, IMCR0 to IMCR12

	Bit [Bit Name]							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Register Name	**[IMR/CR n0]**	**[IMR/CR n1]**	**[IMR/CR n2]**	**[IMR/CR n3]**	**[IMR/CR n4]**	**[IMR/CR n5]**	**[IMR/CR n6]**	**[IMR/CR n7]**
IMR0/IMCR0	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
IMR1/IMCR1	—	DADERR	DEI5	DEI4	DEI3	DEI2	DEI1	DEI0
			(DMAC (2))				(DMAC (1))	
IMR2/IMCR2	—	—	—	ADI	—	—	—	—
	—	—	—	(ADC)	—	—	—	—
IMR3/IMCR3	TEI	TXI	RXI	ERI	—	—	IrDAI1	IrDAI0
			(SIM)		—	—		(IrDA)
IMR4/IMCR4	—	TUNI2	TUNI1	TUNI0	—	—	—	—
	—	(TMU2)	(TMU1)	(TMU0)	—	—	—	—
IMR5/IMCR5	—	—	SCIF5	SCIF4	SCIF3	SCIF2	SCIF1	SCIF0
	—	—		(SCIFA)			(SCIF)	
IMR6/IMCR6	—	—	—	—	—	—	—	SIOFI
	—	—	—	—	—	—	—	(SIOF)
IMR7/IMCR7	IICI0	—	—	—	—	—	—	—
	(IICI0)	—	—	—	—	—	—	—
IMR8/IMCR8	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
IMR9/IMCR9	—	—	—	CMTI	—	—	—	—
	—	—	—	(CMT)	—	—	—	—
IMR10/IMCR10	—	—	—	—	—	CUI	PRI	ATI
	—	—	—	—			(RTC)	
IMR11/IMCR11	IICI1	—	—	—	—	—	PINTBI3 to PINTBI0	PINTAI7 to PINTAI0
	(IICI1)	—	—	—	—	—		(PINT)
IMR11/IMCR12	—	—	—	—	—	—	TPUI1	TPUI0
	—	—	—	—	—	—		(TPU)

[Legend]

n = 0 to 12

Note: —: Reserved. An undefined value will be read. The write value should always be 0.

The corresponding module names are indicated by the entries in parentheses in the rows under the bit names.

10.3.10 User Interrupt Mask Level Register (USERIMASK)

USERIMASK is a 32-bit register that specifies the level of interrupts to be accepted. As this register is allocated to a different 64-Kbyte page than where the other INTC registers are allocated, it can be accessed in user mode by translating its address to the corresponding address in area 7 through the MMU.

When the level of an interrupt is not higher than the interrupt level specified in the UIMASK bit, the interrupt is masked. Specifying H'F masks all interrupts except for NMI. The interrupt with a higher level than that specified in the UIMASK bit is accepted only when the corresponding interrupt mask bit in the interrupt mask register is 0 (interrupt enabled) and the IMASK bit setting in SR is lower than the level of the interrupt. The UIMASK value does not change even after an interrupt is accepted.

This register is initialized to H'0000 0000 (all interrupts enabled) by a power-on reset or manual reset. To prevent unintentional modification, this register can only be written to with bits 31 to 24 set to H'A5.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	UIMASK				—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R/W	Reserved These bits are always read as 0. When writing to the UIMASK bit, be sure to write H'A5 to these bits.
23 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 4	UIMASK	0000	R/W	User Interrupt Mask Level When the level of an interrupt is not higher than this value, the interrupt is masked.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

10.3.11 NMI Flag Control Register (NMIFCR)

NMIFCR is a 16-bit register that has an NMI flag (NMIFL bit) that can be read and cleared by software. The NMIFL bit is automatically set to 1 by hardware when the INTC detects an NMI, and can be cleared by writing 0 through software.

The NMIFL bit does not affect CPU processing with regard to NMI acceptance. The NMI request detected by the INTC is cleared when the CPU accepts it, but the NMIFL bit is not cleared automatically. Even if 0 is written to the NMIFL bit before the CPU accepts the NMI request, the NMI request is not canceled.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMIL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NMIFL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	0	R	NMI Input Level Indicates the level of the signal input at the NMI pin. This bit can be read to determine the NMI pin level. This bit cannot be modified. This bit operates in the same way as the NMIL bit in ICR0. 0: NMI input level is low 1: NMI input level is high
14 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	NMIFL	0	R/W	NMI Interrupt Request Detection Indicates whether an NMI interrupt request signal has been detected. This bit is automatically set to 1 when the INTC detects an NMI interrupt request. Write 0 to clear the bit. Writing 1 is ignored. 0: NMI interrupt request has not been detected 1: NMI interrupt request has been detected

10.4 Interrupt Sources

There are five types of interrupt sources: NMI, IRQ, IRL, PINT, and on-chip peripheral modules. Each interrupt has a priority level (16 to 0), with 1 the lowest and 16 the highest. Priority level 0 masks an interrupt, so the interrupt request is ignored.

10.4.1 NMI Interrupt

The NMI interrupt has the highest priority level of 16. When the BL bit in SR of the CPU is 0, NMI interrupts are always accepted. In sleep or software standby mode, NMI interrupts are accepted regardless of the BL setting. In addition, NMI interrupts are accepted by setting the NMIB bit in ICR0 regardless of the BL setting.

The NMI signal is edge-detected. The NMIE bit in ICR0 is used to select either rising or falling edge detection. After the NMIE bit in ICR0 is modified, NMI interrupts are not detected for a maximum of six bus clock cycles.

When the INTMU bit in CPUOPM is set to 1, the SR interrupt mask level (IMASK in SR) is set to 15 automatically when an NMI interrupt is accepted. The reception of an NMI interrupt has no effect on IMASK in SR when the INTMU bit in CPUOPM has been cleared to 0.

10.4.2 IRQ Interrupts

IRQ interrupts are input on pins IRQ7 to IRQ0. Edge-sensing or level-sensing can be selected by setting the IRQnS bits (n = 0 to 7) in ICR1. When level-sensing is selected, operation differs according to the setting of the LSH bit in ICR0. The initial value of LSH in ICR0 is 0, but it is recommended that it be set to 1 before using the INTC.

1. LSH in ICR0 set to 1

   Interrupt requests are not held internally by the INTC. Maintain the state of the IRQ pin until the interrupt is accepted by the CPU and interrupt handling starts.
2. LSH in ICR0 cleared to 0

   When the INTC detects an interrupt request from the state of the IRQ pin, it holds the interrupt request in the INTREQ00 register. The value is held in INTREQ00 even if the interrupt request is negated at the IRQ pin before the interrupt is accepted by the CPU. After the request is negated at the IRQ pin, the value of INTREQ00 is cleared either when the CPU accepts an interrupt (which need not be an IRQ interrupt) or when the corresponding bit in the INTMSK00 register is set to 1. Clear the INTREQ00 flag before enabling interrupts by clearing the BL bit or executing the RTE instruction.

When the INTMU bit in CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically modified to the level of the accepted interrupt. When the INTMU bit is cleared to 0, the IMASK value in SR is not affected by the accepted interrupt.

10.4.3 IRL Interrupts

IRL interrupts are input via the $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ pins as level sense. When the values of the $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ pins are 0 (B'0000), the highest level interrupt request (interrupt priority level 15) is indicated. When the values of the pins are 15 (B'1111), no interrupt is requested (interrupt priority level 0). Figure10.2 shows an example of connection for an IRL interrupt.

The IRL interrupt scheme includes a noise canceller function and the interrupt is detected when the signal levels sampled at each peripheral module clock cycle are the same for consecutive 2 cycles. This prevents sampling of erroneous levels at transitions on the $\overline{\text{IRL}}$ pins. In standby mode, a noise canceler is driven by the clock for the RTC because the supply of peripheral module clock is stopped. Therefore, when the RTC is not used, recovering from standby mode by the IRL interrupt cannot be executed.

The priority level provided by the IRL interrupt signals should be held until the interrupt handling starts after the interrupt request has been accepted. However, changing to a higher priority level will cause no problem.

The interrupt mask bits (I3 to I0) in the status register (SR) are not affected by the IRL interrupt handling.

When the LSH bit in ICR0 is 0, the interrupt request will be retained inside the LSI even when the interrupt request from outside has been negated. The LSH bit should normally be set to 1.

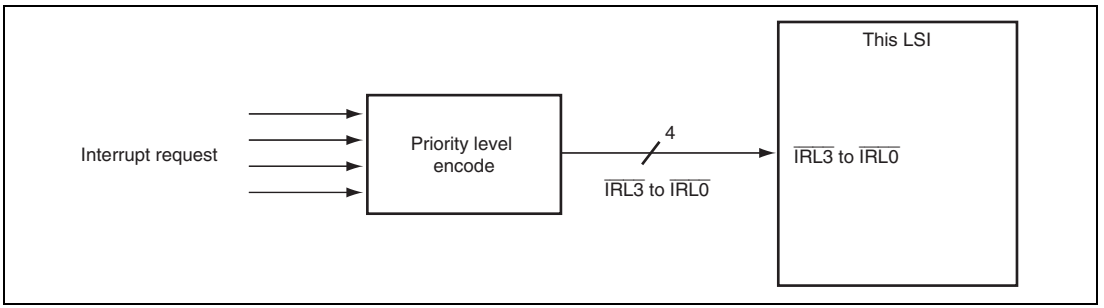


Figure 10.2 Example of IRL Interrupt Connection

10.4.4 PINT Interrupt

PINT interrupts are input via pins PINTA7 to PINTA0, PINTB3 to PINTB0 as level sense. The priority level of PINTA0 to PINTA7 (PINTA) and PINTB3 to PINTB0 (PINTB) can be set by the interrupt priority level register D (IPRD) in a range from 0 to 15. The PINT interrupt signal level should be held until the interrupt handling starts after the interrupt request has been accepted.

The interrupt mask bits I3 to I0 in the status register (SR) are not affected by the PIN interrupt processing routine.

When the LSH bit in ICR0 is 0, there is a possibility that the interrupt request will be retained inside the LSI even when the interrupt request from outside is negated. The LSH bit should normally be set to 1.

10.4.5 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are generated by the peripheral modules.

Not every interrupt source is assigned a different interrupt vector. Sources are reflected in the interrupt event register (INTEVT). It is easy to identify sources by using the value of INTEVT as a branch offset in the exception handling routine.

A priority level (from 15 to 0) can be set for each module by writing to IPRA to IPRK.

When the INTMU bit in the CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically modified to the level of the accepted interrupt. When the INTMU bit in CPUOPM is cleared to 0, the IMASK value in SR is not affected by the accepted interrupt.

The interrupt source flags and interrupt enable flags in each peripheral module must be updated only while the BL bit in SR is set to 1 or corresponding interrupt request is masked by the IMASK bit in SR, IMRs, or USERIMASK. To prevent accepting unintentional interrupts that should have been updated, read the on-chip peripheral register with the corresponding flag, wait for the priority determination time for peripheral modules shown in table 10.8 (e.g. a period required to read a register in INTC once which are driven by the peripheral module clock), and then clear the BL bit to 0 or clear the corresponding interrupt mask by changing the mask setting. Thus, the necessary interval for internal processing is ensured. To update multiple flags, after updating the last flag, read only the register that includes the last flag.

If a flag is updated while the BL bit is 0, execution may branch to the interrupt handling routine with INTEVT = 0; interrupt handling may start depending on the timing relationship between flag updating and interrupt request detection in the LSI. In this case, operation can be continued without causing any problems by executing the RTE instruction.

10.4.6 Interrupt Exception Handling and Priority

Tables 10.6 and 10.7 show the interrupt sources, the codes for the interrupt event register (INTEVT), and the interrupt priority.

Each interrupt source is assigned to a unique INTEVT code. The start address of the exception handling routine is common for all interrupt sources. This is why, for instance, the value of INTEVT is used as an offset at the start of the exception handling routine to branch execution in order to identify the interrupt source.

On-chip peripheral module interrupt priorities can be set freely between 15 and 0 for each module by using IPRA to IPRK. A reset assigns priority level 0 to the on-chip peripheral module interrupts.

If the same priority level is assigned to two or more interrupt sources and interrupts from those sources occur simultaneously, their priority is determined according to the default priority indicated at the right in tables 10.6 and 10.7.

Interrupt priority registers and interrupt mask registers must be updated only while the BL bit in SR is set to 1. To prevent accepting unintentional interrupts, read any interrupt priority register and then clear the BL bit to 0, which ensures the necessary interval for internal processing.

Table 10.6 External Interrupt Sources and Priority

Interrupt Source		INTEVT Code	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Range	Default Priority
NMI		H'1C0	16	—	—	High
IRL	IRL[3:0] = 0	H'200	15	—	—	
	IRL[3:0] = 1	H'220	14	—	—	
	IRL[3:0] = 2	H'240	13	—	—	
	IRL[3:0] = 3	H'260	12	—	—	
	IRL[3:0] = 4	H'280	11	—	—	
	IRL[3:0] = 5	H'2A0	10	—	—	
	IRL[3:0] = 6	H'2C0	9	—	—	
	IRL[3:0] = 7	H'2E0	8	—	—	
	IRL[3:0] = 8	H'300	7	—	—	
	IRL[3:0] = 9	H'320	6	—	—	
	IRL[3:0] = A	H'340	5	—	—	
	IRL[3:0] = B	H'360	4	—	—	
	IRL[3:0] = C	H'380	3	—	—	
	IRL[3:0] = D	H'3A0	2	—	—	
	IRL[3:0] = E	H'3C0	1	—	—	
IRQ	IRQ0	H'600	15 to 0 (0)	INTPRI00 (31 to 28)	—	
	IRQ1	H'620	15 to 0 (0)	INTPRI00 (27 to 24)	—	
	IRQ2	H'640	15 to 0 (0)	INTPRI00 (23 to 20)	—	
	IRQ3	H'660	15 to 0 (0)	INTPRI00 (19 to 16)	—	
	IRQ4	H'680	15 to 0 (0)	INTPRI00 (15 to 12)	—	
	IRQ5	H'6A0	15 to 0 (0)	INTPRI00 (11 to 8)	—	
	IRQ6	H'6C0	15 to 0 (0)	INTPRI00 (7 to 4)	—	
	IRQ7	H'6E0	15 to 0 (0)	INTPRI00 (3 to 0)	—	Low

Table 10.7 On-Chip Peripheral Module Interrupt Sources and Priority

Interrupt Source		INTEVT Code	Interrupt Priority (Initial Value)	Corresponding IPR (Bit Numbers)	Priority within IPR Setting Range	Default Priority
HUDI		H'5E0	15	—	—	High
SIM	ERI	H'700	15 to 0 (0)	IPRB (7 to 4)	High	
	RXI	H'720	15 to 0 (0)			
	TXI	H'740	15 to 0 (0)			
	TEI	H'760	15 to 0 (0)		Low	
IIC1	IICI1	H'7E0	15 to 0 (0)	IPRI (3 to 0)	—	
DMAC (1)	DEI0	H'800	15 to 0 (0)	IPRE (15 to 12)	High	
	DEI1	H'820	15 to 0 (0)			
	DEI2	H'840	15 to 0 (0)			
	DEI3	H'860	15 to 0 (0)		Low	
IrDA	IrDAI0	H'940	15 to 0 (0)	IPRD (7 to 4)	—	
	IrDAI1	H'960	15 to 0 (0)	IPRD (3 to 0)	—	
ADC	ADI	H'980	15 to 0 (0)	IPRE (3 to 0)	—	
TPU	TPUI0	H'9A0	15 to 0 (0)	IPRK (15 to 12)	—	
	TPUI1	H'9C0	15 to 0 (0)	IPRK (11 to 8)	—	
DMAC (2)	DEI4	H'B80	15 to 0 (0)	IPRE (11 to 8)	High	
	DEI5	H'BA0	15 to 0 (0)			
	DADERR	H'BC0	15 to 0 (0)		Low	
SCIF	SCIFI0	H'C00	15 to 0 (0)	IPRG (15 to 12)	High	
	SCIFI1	H'C20	15 to 0 (0)	IPRG (11 to 8)		
	SCIFI2	H'C40	15 to 0 (0)	IPRG (7 to 4)		
	SCIFI3	H'C60	15 to 0 (0)	IPRG (3 to 0)	Low	
SCIFA	SCIFI4	H'C80	15 to 0 (0)	IPRH (15 to 12)	—	
	SCIFI5	H'CA0	15 to 0 (0)	IPRH (11 to 8)	—	
IIC0	IICI0	H'E60	15 to 0 (0)	IPRH (3 to 0)	—	
CMT	CMTI	H'F00	15 to 0 (0)	IPRF (3 to 0)	—	
SIOF	SIOFI	H'F20	15 to 0 (0)	IPRI (15 to 12)	—	Low

Interrupt Source		INTEVT Code	Interrupt Priority (Initial Value)	Corresponding IPR (Bit Numbers)	Priority within IPR Setting Range	Default Priority
TMU0	TUNI0	H'400	15 to 0 (0)	IPRA (15 to 12)	—	High
TMU1	TUNI1	H'420	15 to 0 (0)	IPRA (11 to 8)	—	
TMU2	TUNI2	H'440	15 to 0 (0)	IPRA (7 to 4)	—	
RTC	ATI	H'480	15 to 0 (0)	IPRA (3 to 0)	High	
	PRI	H'4A0	15 to 0 (0)			
	CUI	H'4C0	15 to 0 (0)		Low	
PINT	PINTAI7 to 0	H'4E0	15 to 0 (0)	IPRD (15 to 12)	—	
	PINTBI3 to 0	H'500	15 to 0 (0)	IPRD (11 to 8)	—	Low

10.5 Operation

10.5.1 Interrupt Sequence

The sequence of interrupt operations is described below. Figures 10.3 and 10.4 are flowcharts of the operations.

1. The interrupt request sources send interrupt request signals to the INTC.
2. The INTC selects the highest-priority interrupt from the sent interrupt requests according to the interrupt priority registers. Lower-priority interrupts are held pending. If two of these interrupts have the same priority level or if multiple interrupts occur within a single module, the interrupt with the highest priority is selected according to tables 10.6 and 10.7.
3. The priority level of the interrupt selected by the INTC is compared with the interrupt mask level (IMASK) set in SR of the CPU. If the priority level is higher than the mask level, the INTC accepts the interrupt and sends an interrupt request signal to the CPU.
4. The CPU accepts an interrupt at a break in instructions.
5. The interrupt source code is set in the interrupt event register (INTEVT).
6. SR and program counter (PC) are saved to SSR and SPC, respectively. R15 is saved to SGR at this time.
7. The BL, MD, and RB bits in SR are set to 1.
8. Execution jumps to the start address of the interrupt exception handling routine (the sum of the value set in the vector base register (VBR) and H'0000 0600).

In the exception handling routine, execution may branch with the INTEVT value used as its offset in order to identify the interrupt source. This enables execution to branch to the handling routine for the individual interrupt source.

Notes: 1. When the INTMU bit in the CPU operating mode register (CPUOPM) is set to 1, the interrupt mask level (IMASK) in SR is automatically set to the level of the accepted interrupt. When the INTMU bit is cleared to 0, the IMASK value in SR is not affected by the accepted interrupt.

2. The interrupt source flag should be cleared in the interrupt handler. To ensure that an interrupt source that should have been cleared is not inadvertently accepted again, read the interrupt source flag, wait for the priority determination time for peripheral modules shown in table 10.8 (e.g. a period required to read a register in INTC once which is driven by the peripheral module clock), and then clear the BL bit or execute an RTE instruction.

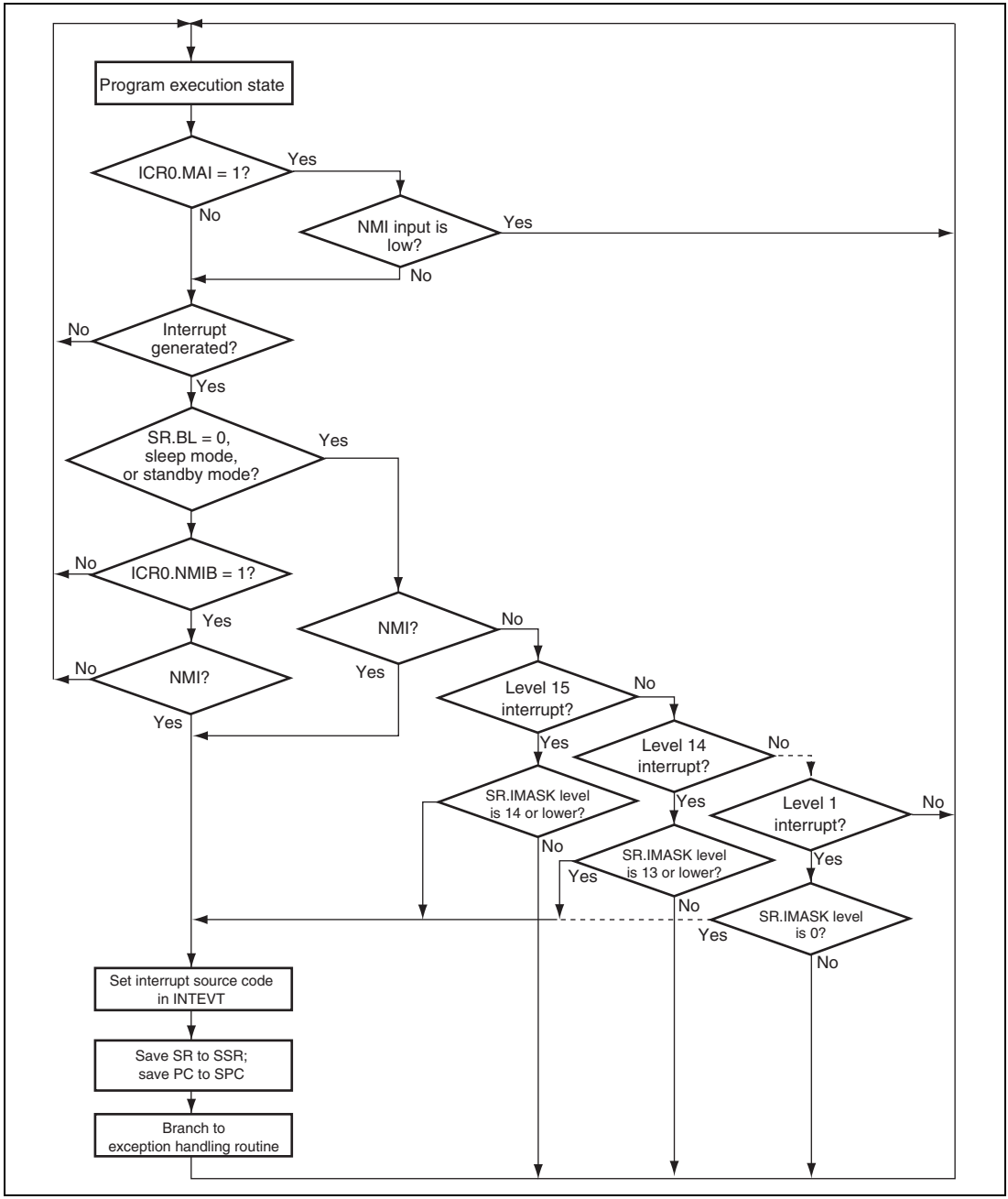


Figure 10.3 Interrupt Operation Flowchart (when CPUOPM.INTMU = 0)

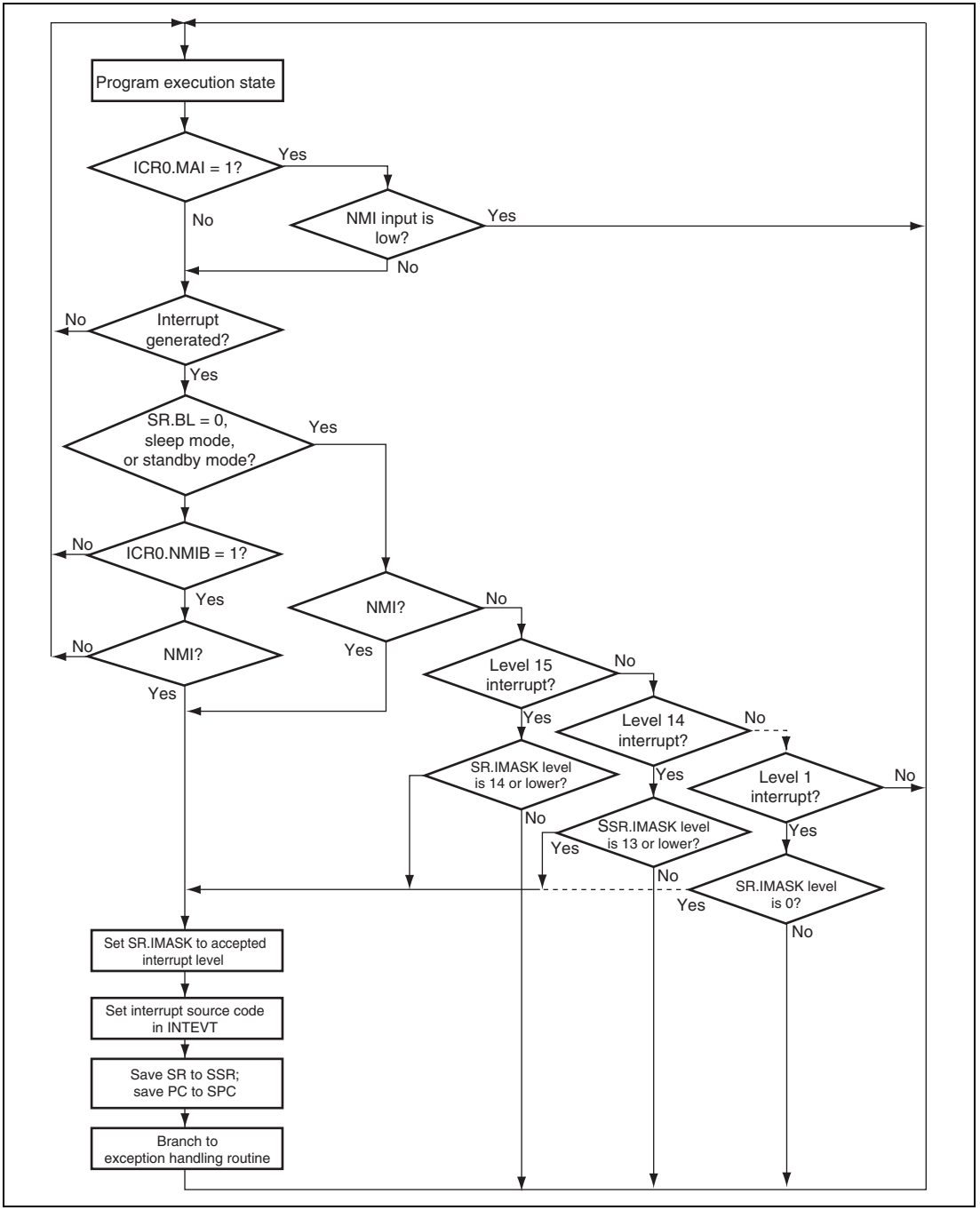


Figure 10.4 Interrupt Operation Flowchart (when CPUOPM.INTMU = 1)

10.5.2 Multiple Interrupts

When handling multiple interrupts, an interrupt handling routine should include the following procedures:

1. To identify the interrupt source, branch to a specific interrupt handling routine for the interrupt source by using the INTEVT code as an offset.
2. Clear the interrupt source in each specific interrupt handling routine.
3. Save SSR and SPC to the stack.
4. Clear the BL bit in SR. When the INTMU bit in CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically modified to the level of the accepted interrupt. When the INTMU bit in CPUOPM is cleared to 0, set the IMASK bit in SR by software to the accepted interrupt level.
5. Handle the interrupt as required.
6. Set the BL bit in SR to 1.
7. Restore SSR and SPC from memory.
8. Execute the RTE instruction.

When these procedures are followed in order, an interrupt of higher priority than the one being handled can be accepted if multiple interrupts occur after step 4. This reduces the interrupt response time for urgent processing.

10.5.3 Interrupt Masking by MAI Bit

Setting the MAI bit in ICR0 to 1 masks interrupts while the NMI signal is low regardless of the BL and IMASK bit settings in SR.

- Normal operation or sleep mode

  All interrupts are masked while the NMI signal is low. Note that only NMI interrupts due to NMI signal input occur.
- Standby mode

  All interrupts including NMI are masked while the NMI signal is low. While the MAI bit is set to 1, the NMI interrupt cannot be used to clear standby mode.

10.5.4 Interrupt Disabling Function in User Mode

Setting the interrupt mask level in USERIMASK disables interrupts having an equal or lower priority level than the specified mask level. This function can disable less-urgent interrupts in a task (such as device driver) operating in user mode to accelerate urgent processing.

USERIMASK is allocated to a different 64-Kbyte page than where the other INTC registers are allocated. When accessing this register in user mode, translate the address through the MMU. In the system that uses a multitasking OS, processes that can access USERIMASK must be controlled by using memory protection functions of the MMU. When terminating the task or switching to another task, be sure to clear USERIMASK to 0 before quitting the task. If the UIMASK bits are left set to a non-zero value, interrupts which are not higher in priority than the UIMASK level are held disabled, and correct operation may not be performed (for example, the OS cannot switch tasks).

A sample sequence of user-mode interrupt disabling operation is described below.

1. Classify interrupts into A and B shown below, and assign higher interrupt levels to A than B.
   A. Interrupts that should be accepted in the device driver
      (interrupts used by the OS, such as timer interrupts)
   B. Interrupts that should be disabled in the device driver
2. Make the MMU settings so that the address space including USERIMASK can only be accessed by the device driver in which interrupts should be disabled.
3. Branch to the device driver.
4. Specify the UIMASK bits so that interrupts B are masked in the device driver operating in user mode.
5. Perform urgent processing in the device driver.
6. Clear the UIMASK bits to 0 to return from the device driver processing.

10.6 Interrupt Response Time

Table 10.8 shows the interrupt response time, which is the interval from when an interrupt request occurs until the interrupt exception handling is started and the start instruction of the exception handling routine is fetched.

Table 10.8 Interrupt Response Time

		Number of States			
Item		**NMI**	**IRQ**	**Peripheral Module**	**Remarks**
Priority determination time		5 Bcyc + 2 Pcyc	4 Bcyc + 2 Pcyc	5 Pcyc	
Wait time until the CPU finishes the current sequence			S - 1 (≥ 0) × Icyc		
Interval from when interrupt exception handling begins (saving SR and PC) until an SuperHyway bus request is issued to fetch the start instruction of the exception handling routine			11 Icyc + 1 Scyc		
Response time	Total	(S + 10) Icyc + 1 Scyc + 5 Bcyc + 2 Pcyc	(S + 10) Icyc + 1 Scyc + 4 Bcyc + 2 Pcyc	(S + 10) Icyc + 1 Scyc + 5 Pcyc	
	Minimum	18 Icyc + S × Icyc	17 Icyc + S × Icyc	16 Icyc + S × Icyc	When Icyc:Scyc:Bcyc: Pcyc = 1:1:1:1

[Legend]

Icyc: Period for one CPU clock cycle

Scyc: Period for one SH clock cycle

Bcyc: Period for one bus clock cycle

Pcyc: Period for one peripheral clock cycle

S: Number of instruction execution states

10.7 Usage Notes

10.7.1 Notes on Level Sensing Interrupt

When the IRQ interrupt are in use with level sensing and when IRL or PINT interrupts are in use, ordinarily set the LSH bit in ICR0 to 1.

Setting the LSH bit to 0 means that even if the external interrupt signal is negated, generation of the interrupt will still indicated within the LSI.

Section 11 Bus State Controller (BSC)

The bus state controller (BSC) outputs control signals for various types of memory that is connected to the external address space and external devices. The BSC functions enable this LSI to connect with SRAM, SDRAM, and other memory storage devices, and external devices.

11.1 Features

The BSC has the following features:

1. External address space
- A maximum 32 or 64 Mbytes for each of areas 0, 2, 3, 4, 5A, 5B, 6A, and 6B, which makes a total of up to 384 Mbytes of external address space (divided into eight areas). (Address map 1)
- A maximum 64 Mbytes for each of areas 0, 2, 3, 4, 5 and 6, which makes a total of up to 384 Mbytes of external address space (divided into six areas). (Address map 2)
- Areas 2 and 3 are merged to form a maximum of 128 Mbytes of area. (Address map 3)
- Each area can be specified as normal space or space of any memory type among byte-selection SRAM, burst ROM (asynchronous), SDRAM, and PCMCIA.
- Data bus width (8, 16, or 32 bits) is selectable for each area.
  For area 0, data bus width is either 16 or 32 bits.
- Controls insertion of wait cycles for each area.
- Controls insertion of wait cycles for each read access and write access.
- Idle cycles in continuous access can be set independently for five cases: read-write (in same space/different space), read-read (in same space/different space), and the first cycle is a write access.

2. Normal space interface
- Supports the interface that can directly connect to SRAM.

3. Burst ROM (clock asynchronous) interface
- High-speed access to the ROM that has the page mode function.

4. SDRAM interface
- Can set the SDRAM in up to two areas.
- Multiplex output for row address/column address.
- Efficient access by single read/single write.
- High-speed access by bank-active mode.
- Supports an auto-refresh and self-refresh.
- Supports low-power function.

5. Byte-selection SRAM interface
- Can connect directly to a byte-selection SRAM.

6. PCMCIA interface
- Supports IC memory cards and I/O card interfaces defined in the JEIDA specifications Ver. 4.2 (PCMCIA2.1).
- Controls the insertion of the wait state using software.
- Supports the bus sizing function of the I/O bus width (only in little endian mode).

7. Bus arbitration
- Outputs a bus acknowledge after receiving a bus request from an external device.

8. Refresh function
- Supports the auto-refresh and self-refresh functions.
- Specifies the refresh interval using the refresh counter and clock selection.
- Can execute concentrated refresh by specifying the refresh counts (1, 2, 4, 6, or 8).

9. Interval timer using refresh counter
- Generates an interrupt request by a compare match.

Note: The PCMCIA interface provided by the BSC only supports the signals and bus protocols shown in table 11.1.

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

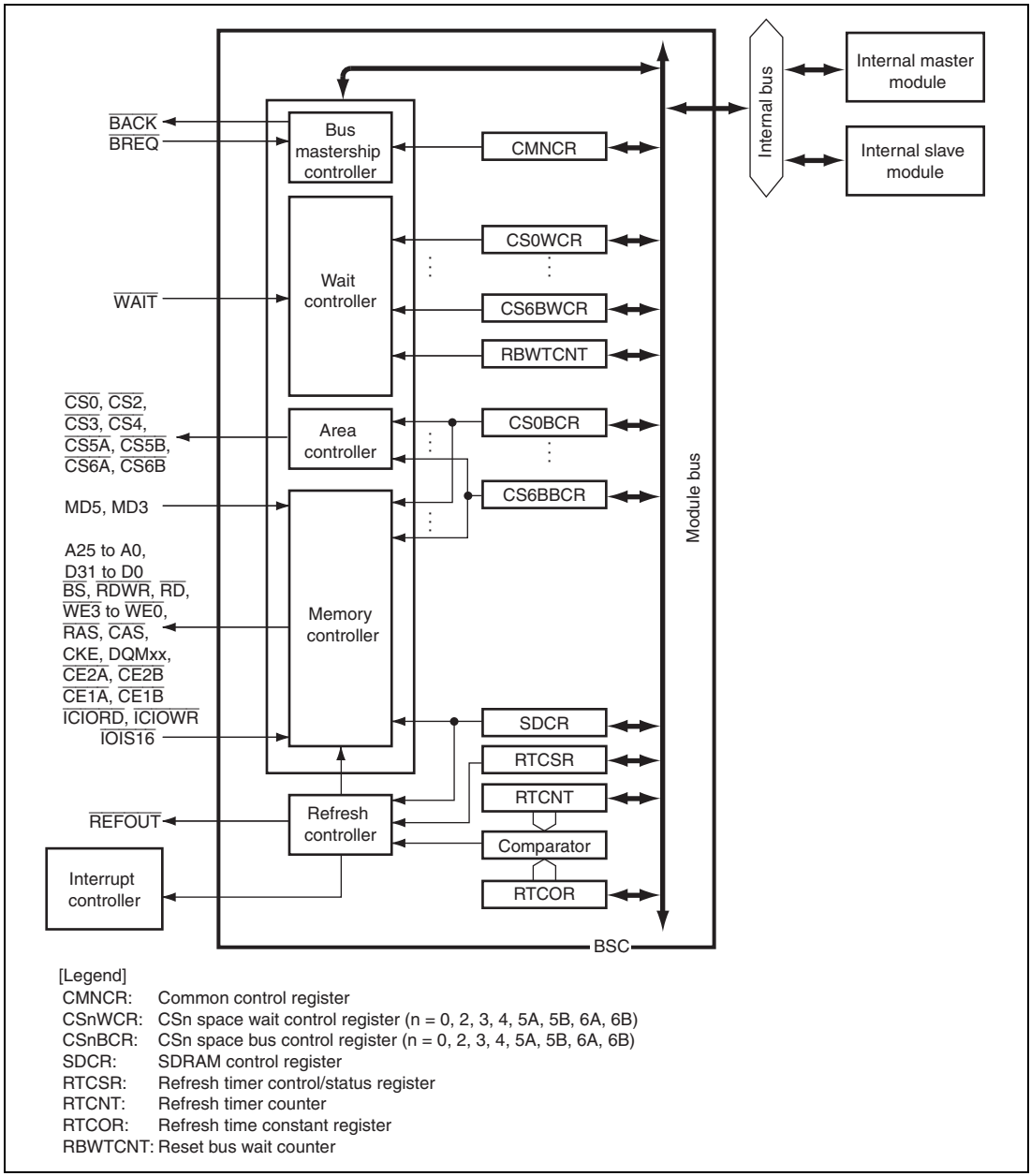


Figure 11.1 Block Diagram of BSC

11.2 Input/Output Pins

The configuration of pins in this module is shown in table 11.1.

Table 11.1 Pin Configuration

Name	I/O	Function
A25 to A0	O	Address bus
D31 to D0	I/O	Data bus
$\overline{BS}$	O	Bus cycle start Asserted when a normal space, burst ROM (asynchronous), or PCMCIA is accessed. Asserted by the same timing as $\overline{CAS}$ in SDRAM access.
$\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$	O	Chip select
$\overline{CS5A}$/$\overline{CE2A}$	O	Chip select Active only for address maps 1 and 3 Corresponds to PCMCIA card select signals D15 to D8 when the PCMCIA is used.
$\overline{CS5B}$/$\overline{CE1A}$	O	Chip select Outputs the area 5 CS signal for address map 2. Corresponds to PCMCIA card select signals D7 to D0 when the PCMCIA is used.
$\overline{CS6A}$/$\overline{CE2B}$	O	Chip select Active only for address maps 1 and 3 Corresponds to PCMCIA card select signals D15 to D8 when the PCMCIA is used.
$\overline{CS6B}$/$\overline{CE1B}$	O	Chip select Outputs the area 6 CS signal for address map 2. Corresponds to PCMCIA card select signals D7 to D0 when the PCMCIA is used.
$\overline{RDWR}$	O	Read/write Connects to $\overline{WE}$ pins when SDRAM or byte-selection SRAM is connected.
$\overline{RD}$	O	Read pulse signal (read data output enable signal) A strobe signal to indicate the memory read cycle when the PCMCIA is used.

Name	I/O	Function
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$	O	Indicates that D31 to D24 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected. Corresponds to signals D31 to D24 when SDRAM is connected. Functions as the I/O write strobe signal when the PCMCIA is used.
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$	O	Indicates that D23 to D16 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected. Corresponds to signals D23 to D16 when the SDRAM is used. Functions as the I/O read strobe signal when the PCMCIA is used.
$\overline{WE1}$/DQMLU/$\overline{WE}$	O	Indicates that D15 to D8 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected. Corresponds to signals D15 to D8 when the SDRAM is used. Functions as the memory write enable signal when the PCMCIA is used.
$\overline{WE0}$/DQMLL	O	Indicates that D7 to D0 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected. Corresponds to select signals D7 to D0 when the SDRAM is used.
$\overline{RAS}$	O	Connects to $\overline{RAS}$ pin when SDRAM is connected.
$\overline{CAS}$	O	Connects to $\overline{CAS}$ pin when SDRAM is connected.
CKE	O	Connects to CKE pin when SDRAM is connected.
$\overline{IOIS16}$	I	PCMCIA 16-bit I/O signal Valid only in little endian mode. Pulled low in bit endian mode.
$\overline{WAIT}$	I	External wait input
$\overline{BREQ}$	I	Bus request input
$\overline{BACK}$	O	Bus acknowledge output
MD5, MD3	I	MD5: Selects data alignment (big endian or little endian) MD3: Specifies area 0 bus width (16/32 bits)
$\overline{REFOUT}$	O	Refresh request output when a bus is released

11.3 Area Overview

11.3.1 Area Division

In the architecture of this LSI, both virtual spaces and physical spaces have 32-bit address spaces. The upper three bits divide into the P0 to P4 areas, and specify the cache access method. For details see section 8, Caches. The remaining 29 bits are used for division of the space into ten areas (address map 1), nine areas (address map 3), or eight areas (address map 2) according to the MAP bit in CMNCR setting. The BSC performs control for this 29-bit space.

As listed in tables 11.2 and 11.4, this LSI can be connected directly to eight, seven, or six physical space areas of memory, and it outputs chip select signals ($\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$, $\overline{CS5A}$, $\overline{CS5B}$, $\overline{CS6A}$, and $\overline{CS6B}$) for each of them. $\overline{CS0}$ is asserted during area 0 access; $\overline{CS5A}$ is asserted during area 5A access when address map 1 is selected; and $\overline{CS5B}$ is asserted when address map 2 is selected.

11.3.2 Shadow Area

Areas 0, 2 to 4, 5A, 5B, 6A, and 6B are decoded by physical addresses A28 to A25, which correspond to areas 000 to 111. Address bits 31 to 29 are ignored. This means that the range of area 0 addresses, for example, is H'00000000 to H'03FFFFFF, and its corresponding shadow space is the address space in P1 to P3 areas obtained by adding to it H'20000000 × n (n = 1 to 6).

The address range for area 7 is H'1C000000 to H'1FFFFFFF. The address space H'1C000000 + H'20000000 × n to H'1FFFFFFF + H'20000000 × n (n = 0 to 6) corresponding to the area 7 shadow space is reserved, so do not use it.

Area P4 (H'E0000000 to H'EFFFFFFF) is an I/O area and is assigned for internal register addresses. Therefore, area P4 does not become shadow space.

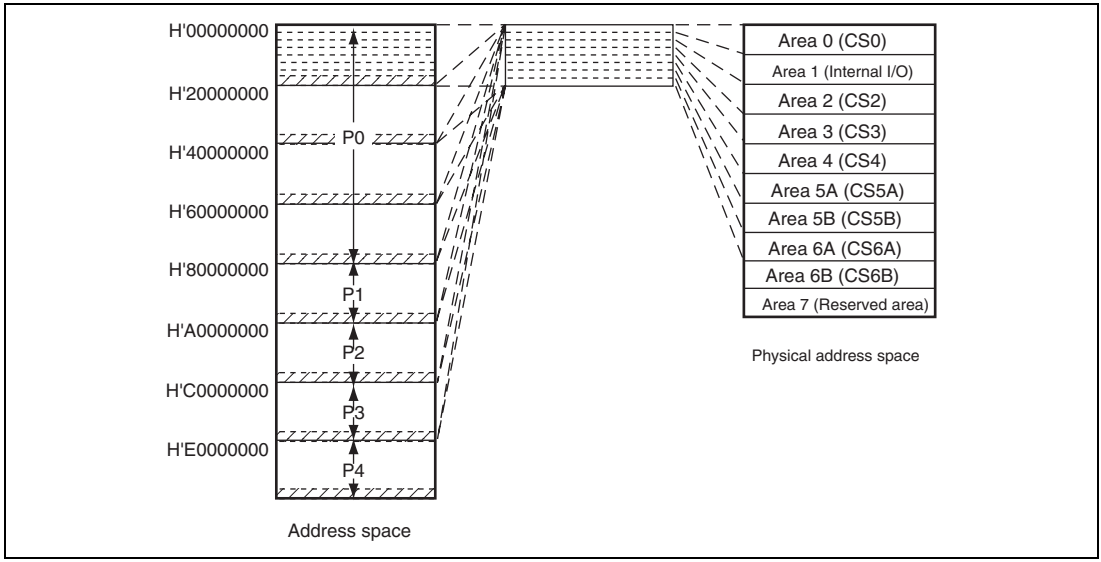


Figure 11.2 Address Space

11.3.3 Address Map

The external address space has a capacity of 384 Mbytes and is used by dividing eight partial spaces (address map 1), seven partial spaces (address map 3), or six partial spaces (address map 2). The kind of memory to be connected and the data bus width are specified in each partial space. The address map for the external address space is shown in tables 11.2 to 11.4.

Table 11.2 Address Space Map 1 (CMNCR.MAP[1:0] = B'00)

Physical Address	Area	Memory to be Connected	Capacity
H'00000000 to H'03FFFFFF	Area 0	Normal memory Burst ROM (Asynchronous)	64 Mbytes
H'04000000 to H'07FFFFFF	Area 1	Internal I/O register area*[2]	64 Mbytes
H'08000000 to H'0BFFFFFF	Area 2	Normal memory Byte-selection SRAM SDRAM*[3]	64 Mbytes
H'0C000000 to H'0FFFFFFF	Area 3	Normal memory Byte-selection SRAM SDRAM*[3]	64 Mbytes
H'10000000 to H'13FFFFFF	Area 4	Normal memory Byte-selection SRAM Burst ROM (Asynchronous)	64 Mbytes
H'14000000 to H'15FFFFFF	Area 5A	Normal memory	32 Mbytes
H'16000000 to H'17FFFFFF	Area 5B	Normal memory Byte-selection SRAM	32 Mbytes
H'18000000 to H'19FFFFFF	Area 6A	Normal memory	32 Mbytes
H'1A000000 to H'1BFFFFFF	Area 6B	Normal memory Byte-selection SRAM	32 Mbytes
H'1C000000 to H'1FFFFFFF	Area 7	Reserved area*[1]	64 Mbytes

Notes: 1. Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed.
2. Set the top three bits of the address to 101 to allocate in the P2 space.
3. If SDRAM is only connected to one of the two possible areas, connect the SDRAM in area 3. In such cases, normal memory or byte-selection SRAM may be connected in area 2.

Table 11.3 Address Space Map 2 (CMNCR.MAP[1:0] = B'01)

Physical Address	Area	Memory to be Connected	Capacity
H'00000000 to H'03FFFFFF	Area 0	Normal memory Burst ROM (Asynchronous)	64 Mbytes
H'04000000 to H'07FFFFFF	Area 1	Internal I/O register area*3	64 Mbytes
H'08000000 to H'0BFFFFFF	Area 2	Normal memory Byte-selection SRAM SDRAM*4	64 Mbytes
H'0C000000 to H'0FFFFFFF	Area 3	Normal memory Byte-selection SRAM SDRAM*4	64 Mbytes
H'10000000 to H'13FFFFFF	Area 4	Normal memory Byte-selection SRAM Burst ROM (Asynchronous)	64 Mbytes
H'14000000 to H'17FFFFFF	Area 5*2	Normal memory Byte-selection SRAM PCMCIA	64 Mbytes
H'18000000 to H'1BFFFFFF	Area 6*2	Normal memory Byte-selection SRAM PCMCIA	64 Mbytes
H'1C000000 to H'1FFFFFFF	Area 7	Reserved area*1	64 Mbytes

Notes: 1. Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed.

2. For area 5, registers CS5BBCR and CS5BWCR are valid and CS5B is valid as the chip select signal.

   For area 6, registers CS6BBCR and CS6BWCR are valid and CS6B is valid as the chip select signal.

3. Set the top three bits of the address to 101 to allocate in the P2 space.

4. If SDRAM is only connected to one of the two possible areas, connect the SDRAM in area 3. In such cases, normal memory or byte-selection SRAM may be connected in area 2.

Table 11.4 Address Space Map 3 (CMNCR.MAP[1:0] = B'10)

Physical Address	Area	Memory to be Connected	Capacity
H'00000000 to H'03FFFFFF	Area 0	Normal memory Burst ROM (Asynchronous)	64 Mbytes
H'04000000 to H'07FFFFFF	Area 1	Internal I/O register area*[2]	64 Mbytes
H'08000000 to H'0FFFFFFF	Area 2/3*[3]	Normal memory Byte-selection SRAM SDRAM	128 Mbytes
H'10000000 to H'13FFFFFF	Area 4	Normal memory Byte-selection SRAM Burst ROM (Asynchronous)	64 Mbytes
H'14000000 to H'15FFFFFF	Area 5A	Normal memory	32 Mbytes
H'16000000 to H'17FFFFFF	Area 5B	Normal memory Byte-selection SRAM	32 Mbytes
H'18000000 to H'19FFFFFF	Area 6A	Normal memory	32 Mbytes
H'1A000000 to H'1BFFFFFF	Area 6B	Normal memory Byte-selection SRAM	32 Mbytes
H'1C000000 to H'1FFFFFFF	Area 7	Reserved area*[1]	64 Mbytes

Notes: 1. Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed.
2. Set the top three bits of the address to 101 to allocate in the P2 space.
3. Areas 2 and 3 are merged into one area. In this case, registers CS3BCR and CS3WCR are valid and $\overline{CS3}$ is valid as the chip select signal.

11.3.4 Area 0 Memory Type and Memory Bus Width

The memory bus width in this LSI can be set for each area. In area 0, external pins can be used to select word (16 bits) or longword (32 bits) on power-on reset. The memory bus width of the other area is set by the register. The correspondence between the memory type, external pins (MD3), and bus width is listed in the table below.

Table 11.5 Correspondence between External Pins (MD3), Memory Type of CS0, and Memory Bus Width

MD3	Memory Type	Bus Width
0	Normal memory	16 bits
1		32 bits

11.3.5 Data Alignment

This LSI supports the big endian and little endian methods of data alignment. The data alignment is specified using the external pin (MD5) at power-on reset as shown in table 11.6.

Table 11.6 Correspondence between External Pin (MD5) and Data Alignment

MD5	Data Alignment
0	Big endian
1	Little endian

11.4 Register Descriptions

Table 11.7 shows the BSC register configuration. Table 11.8 shows the register states in each operating mode.

Table 11.7 Register Configuration

Name	Abbreviation	R/W	Address	Access Size
Common control register	CMNCR	R/W	H'FEC1 0000	32
Bus control register for CS0	CS0BCR	R/W	H'FEC1 0004	32
Bus control register for CS2	CS2BCR	R/W	H'FEC1 0008	32
Bus control register for CS3	CS3BCR	R/W	H'FEC1 000C	32
Bus control register for CS4	CS4BCR	R/W	H'FEC1 0010	32
Bus control register for CS5A	CS5ABCR	R/W	H'FEC1 0014	32
Bus control register for CS5B	CS5BBCR	R/W	H'FEC1 0018	32
Bus control register for CS6A	CS6ABCR	R/W	H'FEC1 001C	32
Bus control register for CS6B	CS6BBCR	R/W	H'FEC1 0020	32
Wait control register for CS0	CS0WCR	R/W	H'FEC1 0024	32
Wait control register for CS2	CS2WCR	R/W	H'FEC1 0028	32
Wait control register for CS3	CS3WCR	R/W	H'FEC1 002C	32
Wait control register for CS4	CS4WCR	R/W	H'FEC1 0030	32
Wait control register for CS5A	CS5AWCR	R/W	H'FEC1 0034	32
Wait control register for CS5B	CS5BWCR	R/W	H'FEC1 0038	32
Wait control register for CS6A	CS6AWCR	R/W	H'FEC1 003C	32
Wait control register for CS6B	CS6BWCR	R/W	H'FEC1 0040	32
SDRAM control register	SDCR	R/W	H'FEC1 0044	32
Refresh timer control/status register	RTCSR	R/W	H'FEC1 0048	32
Refresh timer counter	RTCNT	R/W	H'FEC1 004C	32
Refresh time constant register	RTCOR	R/W	H'FEC1 0050	32
SDRAM mode register	SDMR2	W	H'FEC1 4xxx	—
SDRAM mode register	SDMR3	W	H'FEC1 5xxx	—
Reset bus wait counter	RBWTCNT	—	—	—

Table 11.8 Register States in Each Operating Mode

Name	Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
Common control register	CMNCR	Initialized	Retained	—	Retained
Bus control register for CS0	CS0BCR	Initialized	Retained	—	Retained
Bus control register for CS2	CS2BCR	Initialized	Retained	—	Retained
Bus control register for CS3	CS3BCR	Initialized	Retained	—	Retained
Bus control register for CS4	CS4BCR	Initialized	Retained	—	Retained
Bus control register for CS5A	CS5ABCR	Initialized	Retained	—	Retained
Bus control register for CS5B	CS5BBCR	Initialized	Retained	—	Retained
Bus control register for CS6A	CS6ABCR	Initialized	Retained	—	Retained
Bus control register for CS6B	CS6BBCR	Initialized	Retained	—	Retained
Wait control register for CS0	CS0WCR	Initialized	Retained	—	Retained
Wait control register for CS2	CS2WCR	Initialized	Retained	—	Retained
Wait control register for CS3	CS3WCR	Initialized	Retained	—	Retained
Wait control register for CS4	CS4WCR	Initialized	Retained	—	Retained
Wait control register for CS5A	CS5AWCR	Initialized	Retained	—	Retained
Wait control register for CS5B	CS5BWCR	Initialized	Retained	—	Retained
Wait control register for CS6A	CS6AWCR	Initialized	Retained	—	Retained
Wait control register for CS6B	CS6BWCR	Initialized	Retained	—	Retained
SDRAM control register	SDCR	Initialized	Retained	—	Retained
Refresh timer control/status register	RTCSR	Initialized	Retained	—	Retained
Refresh timer counter	RTCNT	Initialized	Retained	—	Retained
Refresh time constant register	RTCOR	Initialized	Retained	—	Retained
SDRAM mode register	SDMR2	—	—	—	—
SDRAM mode register	SDMR3	—	—	—	—
Reset bus wait counter	RBWTCNT	Initialized	Retained	—	Retained

11.4.1 Common Control Register (CMNCR)

CMNCR is a 32-bit register that controls the common items for each area. Do not access external memory other than area 0 until the CMNCR initialization is complete.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	CKO STP	CKO DRV	—	—	—	—	—	—	—	DM STP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BSD	MAP[1:0]		BLOCK	—	—	—	—	—	—	—	END IAN	—	HIZ MEM	HIZ CNT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0/1*	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	CKOSTP	0	R/W	CKO Stop 0: Outputs CKO. 1: Stops CKO and outputs a low level. Note: Just after the CKOSTP bit has been set to 1, an invalid waveform may be output as the CKO signal before it becomes stable at low level.
24	CKODRV	0	R/W	CKO, CKE Drive Control Controls the operation selected by bit 0 (HIZCNT) setting. See bit 0 (HIZCNT) for details.
23 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
16	DMSTP	0	R/W	Power-Down Mode This bit controls a low-power-consumption mode setting for the BSC. Clearing this bit to 0 stops the BSC's internal circuit for control of external bus mastership ($\overline{\text{BREQ}}$/$\overline{\text{BACK}}$), reducing power consumption. If control of external bus mastership ($\overline{\text{BREQ}}$/$\overline{\text{BACK}}$) is to be used, set this bit to 1. 0: The BSC enters power-down mode. 1: The BSC releases power-down mode.
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	BSD	0	R/W	Bus Access Start Timing Specification After Bus Acknowledge Specifies the bus access start timing after the external bus acknowledge signal is received. 0: Starts the external access at the same timing as the address drive start after the bus acknowledge signal is received. 1: Starts the external access one cycle following the address drive start after the bus acknowledge signal is received.
13, 12	MAP[1:0]	00	R/W	Space Specification Selects the address map for the external address space. The address maps to be selected are shown in tables 11.2 to 11.4. 00: Selects address map 1 01: Selects address map 2 10: Selects address map 3 11: Setting prohibited
11	BLOCK	0	R/W	Bus Lock Bit Specifies whether or not the $\overline{\text{BREQ}}$ signal is received. 0: Receives $\overline{\text{BREQ}}$ 1: Does not receive $\overline{\text{BREQ}}$

Bit	Bit Name	Initial Value	R/W	Description
10 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
3	ENDIAN	0/1*	R	Endian Flag Samples the external pin for specifying endian on power-on reset (MD5). All address spaces are defined by this bit. This is a read-only bit. 0: The external pin for specifying endian (MD5) was low level on power-on reset. This LSI is being operated as big endian. 1: The external pin for specifying endian (MD5) was high level on power-on reset. This LSI is being operated as little endian.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	HIZMEM	0	R/W	High-Z Memory Control Specifies the pin state in standby mode for A25 to A0, $\overline{BS}$, $\overline{CSn}$, $\overline{RDWR}$, $\overline{WEn}$/DQMxx, and $\overline{RD}$. When a bus is released, these pins enter the high-impedance state regardless of the setting of this bit. 0: High impedance in standby mode 1: Driven in standby mode

Bit	Bit Name	Initial Value	R/W	Description
0	HIZCNT	0	R/W	High-Z Control Specifies the states of CKO, CKE, $\overline{RAS}$, and $\overline{CAS}$ in standby mode and bus released state. • When bit 24 (CKODRV) is 0 0: CKO, CKE, $\overline{RAS}$, and $\overline{CAS}$ go high impedance in standby mode and bus released state. 1: CKO, CKE, $\overline{RAS}$, and $\overline{CAS}$ are driven in standby mode and bus released state. • When bit 24 (CKODRV) is 1 0: CKO and CKE are driven; $\overline{RAS}$, and $\overline{CAS}$ go high impedance in standby mode and bus released state. 1: CKO, CKE, $\overline{RAS}$, and $\overline{CAS}$ are driven in standby mode and bus released state.

Note: * The external pin (MD5) for specifying endian is sampled on power-on reset. When big endian is specified, this bit is read as 0 and when little endian is specified, this bit is read as 1.

11.4.2 CSn Space Bus Control Register (CSnBCR)

This register specifies the type of memory connected to each space, data-bus width of each space, and the number of wait cycles between access cycles.

Do not access external memory other than area 0 until the CSnBCR initialization is completed.

(n = 0, 2, 3, 4, 5A, 5B, 6A, 6B)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	BAS	—	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADR SFIX	—	—	SW[1:0]		WR[3:0]				WM	—	—	—	—	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	IWW[2:0]	011	R/W	Idle Cycles between Write-Read Cycles and Write-Write Cycles These bits specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycles are the write-read cycle and write-write cycle. 000: Setting prohibited 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
27 to 25	IWRWD [2:0]	011	R/W	Idle Cycles for Another Space Read-Write Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycle is a read-write one in which continuous accesses switch between different spaces. 000: No idle cycle inserted 001: 1 idle cycles inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
24 to 22	IWRWS [2:0]	011	R/W	Idle Cycles for Read-Write in Same Space Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-write cycle of which continuous accesses are for the same space. 000: No idle cycle inserted 001: 1 idle cycles inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
21 to 19	IWRRD [2:0]	011	R/W	Idle Cycles for Read-Read in Another Space Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous accesses switch between different spaces. 000: No idle cycle inserted 001: 1 idle cycles inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
18 to 16	IWRRS [2:0]	011	R/W	Idle Cycles for Read-Read in Same Space Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous accesses are for the same space. 000: No idle cycle inserted 001: 1 idle cycles inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	TYPE[3:0]	0000	R/W	Memory Type Specify the type of memory connected to a space. 0000: Normal space 0001: Burst ROM (clock asynchronous) 0010: Setting prohibited 0011: Byte-selection SRAM 0100: SDRAM 0101: PCMCIA 0110: Setting prohibited 0111: Setting prohibited 1000: Setting prohibited 1001: Setting prohibited 1010: Setting prohibited 1011: Setting prohibited 1100: Setting prohibited 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited Note: Memory type for area 0 immediately after reset is normal space. Either normal space or burst ROM (asynchronous) can be selected by these bits. For details on memory type in each area, see tables 11.2 to 11.4.
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10, 9	BSZ[1:0]	11*	R/W	Data Bus Width Specify the data bus width of spaces. 00: Setting prohibited 01: 8 bits 10: 16 bits 11: 32 bits Notes: 1. The data bus width for area 0 is specified by the external pin. The BSZ[1:0] bit setting in CS0BCR is ignored. 2. If area 5 or area 6 is specified as PCMCIA space, the bus width can be specified as either 8 bits or 16 bits. 3. If area 2 or area 3 is specified as SDRAM space, the bus width can be specified as either 16 bits or 32 bits.
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * CS0BCR samples the external pins (MD3 and MD4) that specify the bus width at power-on reset.

11.4.3 CSn Space Wait Control Register (CSnWCR)

This register specifies various wait cycles for memory accesses. The bit configuration of this register varies as shown below according to the memory type (TYPE3, TYPE2, TYPE1, or TYPE0) specified by the CSn space bus control register (CSnBCR). Specify CSnWCR before accessing the target area. Specify CSnBCR first, then specify CSnWCR.

(n = 0, 2, 3, 4, 5A, 5B, 6A, 6B)

(1) Normal Space and Byte-Selection SRAM

- CS0WCR, CS6AWCR, CS6BWCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	BAS	—	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADR SFIX	—	—	SW[1:0]		WR[3:0]				WM	—	—	—	—	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	Byte Access Selection for Byte-Selection SRAM Specifies the $\overline{\text{WEn}}$ and $\overline{\text{RDWR}}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{\text{WEn}}$ signal at the read/write timing and asserts the $\overline{\text{RDWR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WEn}}$ signal during the read/write access cycle and asserts the $\overline{\text{RDWR}}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
18 to 16	WW[2:0]	000	R/W	Number of Wait Cycles in Write Access Specify the number of wait cycles necessary for write access. 000: Same number of cycles set by WR[3:0] (read access wait) 001: 0 cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15	ADRSFIX	0	R/W	Address Update Disable (valid only for CS6A) 0: Normal address output 1: Address is not updated for the second and subsequent access cycles in burst access
14, 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address/$\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$/$\overline{\text{WEn}}$ Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ or $\overline{\text{WEn}}$ assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	WR[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles necessary for read/write access. However, if WW[2:0] is set to a nonzero value, the number of wait cycles for write access is determined by the WW[2:0] setting. 0000: 0 cycles 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited
6	WM	1	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	HW[1:0]	00	R/W	Number of Delay Cycles from $\overline{RD}$/$\overline{WEn}$ Negation to Address/$\overline{CSn}$ Negation Specify the number of delay cycles from $\overline{RD}$ or $\overline{WEn}$ negation to address and $\overline{CSn}$ negation. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS2WCR, CS3WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	BW[1:0]		PMD	BAS	—	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SW[1:0]		WR[3:0]				WM	—	—	—	—	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23, 22	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inserted to the second and subsequent access cycles in a burst access. Valid for byte-selection SRAM with page mode specified (PMD bit = 1). 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles Note: Bit position is different from that of burst ROM (asynchronous).

Bit	Bit Name	Initial Value	R/W	Description
21	PMD	0	R/W	Page Mode Specification for Byte-Selection SRAM Specifies the page mode for byte-selection SRAM. 0: Non-page mode access 1: Page mode access
20	BAS	0	R/W	Byte Access Selection for Byte-Selection SRAM Specifies the $\overline{\text{WEn}}$ and $\overline{\text{RDWR}}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{\text{WEn}}$ signal at the read/write timing and asserts the $\overline{\text{RDWR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WEn}}$ signal during the read/write access cycle and asserts the $\overline{\text{RDWR}}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Wait Cycles in Write Access Specify the number of wait cycles necessary for write access. 000: Same number of cycles set by WR[3:0] (read access wait) 001: 0 cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address/$\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$/$\overline{\text{WEn}}$ Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ or $\overline{\text{WEn}}$ assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10 to 7	WR[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles necessary for read/write access. However, if WW[2:0] is set to a non-zero value, the number of wait cycles for write access is determined by the WW[2:0] setting. 0000: 0 cycles 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
6	WM	1	R/W	External Wait Mask Specification Specify whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Number of Delay Cycles from $\overline{RD}$/$\overline{WEn}$ Negation to Address/$\overline{CSn}$ Negation Specify the number of delay cycles from $\overline{RD}$ or $\overline{WEn}$ negation to address and $\overline{CSn}$ negation. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS4WCR, CS5AWCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	BAS	—	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SW[1:0]		WR[3:0]				WM	—	—	—	—	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
20	BAS	0	R/W	Byte Access Selection for Byte-Selection SRAM Specifies the $\overline{\text{WEn}}$ and $\overline{\text{RDWR}}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{\text{WEn}}$ signal at the read/write timing and asserts the $\overline{\text{RDWR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WEn}}$ signal during the read/write access cycle and asserts the $\overline{\text{RDWR}}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Wait Cycles in Write Access Specify the number of wait cycles necessary for write access. 000: Same number of cycles set by WR[3:0] (read access wait) 001: 0 cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address/$\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$/$\overline{\text{WEn}}$ Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ or $\overline{\text{WEn}}$ assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	WR[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles necessary for read/write access. However, if WW[2:0] is set to a non-zero value, the number of wait cycles for write access is determined by the WW[2:0] setting. 0000: 0 cycles 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited
6	WM	1	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycles is 0. 0: External wait is valid 1: External wait is ignored

Bit	Bit Name	Initial Value	R/W	Description
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Number of Delay Cycles from $\overline{RD}$/$\overline{WEn}$ Negation to Address/$\overline{CSn}$ Negation Specify the number of delay cycles from $\overline{RD}$ or $\overline{WEn}$ negation to address and $\overline{CSn}$ negation. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS5BWCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	BAS	—	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SW[1:0]		WR[3:0]				WM	—	—	—	—	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	Byte Access Selection for Byte-Selection SRAM Specifies the $\overline{WEn}$ and $\overline{RDWR}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{WEn}$ signal at the read/write timing and asserts the $\overline{RDWR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read/write access cycle and asserts the $\overline{RDWR}$ signal at the write timing.

Bit	Bit Name	Initial Value	R/W	Description
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Wait Cycles in Write Access Specify the number of wait cycles necessary for write access. 000: Same number of cycles set by WR[3:0] (read access wait) 001: 0 cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address/$\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$/$\overline{\text{WEn}}$ Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ or $\overline{\text{WEn}}$ assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	WR[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles necessary for read/write access. However, if WW[2:0] is set to a non-zero value, the number of wait cycles for write access is determined by the WW[2:0] setting. 0000: 0 cycles 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited
6	WM	1	R/W	External Wait Mask Specification Specify whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycles is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	HW[1:0]	00	R/W	Number of Delay Cycles from $\overline{RD}$/$\overline{WEn}$ Negation to Address/$\overline{CSn}$ Negation Specify the number of delay cycles from $\overline{RD}$ or $\overline{WEn}$ negation to address and $\overline{CSn}$ negation. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

(2) Burst ROM (Asynchronous)

- CS0WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BW[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SW[1:0]		W[3:0]				WM	—	—	—	—	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inserted to the second and subsequent access cycles in a burst read access. 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address/$\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$/$\overline{\text{WEn}}$ Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ or $\overline{\text{WEn}}$ assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted in write access cycles and the first read access cycle. 0000: 0 cycles 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
6	WM	1	R/W	External Wait Mask Specification Specify whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycles is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Number of Delay Cycles from $\overline{RD}$/$\overline{WEn}$ Negation to Address/$\overline{CSn}$ Negation Specify the number of delay cycles from $\overline{RD}$ or $\overline{WEn}$ negation to address and $\overline{CSn}$ negation. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS4WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BW[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SW[1:0]		W[3:0]				WM	—	—	—	—	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inserted to the second and subsequent access cycles in a burst access. 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address/$\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$/$\overline{\text{WEn}}$ Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ or $\overline{\text{WEn}}$ assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted in write access cycles and the first read access cycle. 0000: 0 cycles 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited
6	WM	1	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycles is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Number of Delay Cycles from $\overline{RD}$/$\overline{WEn}$ Negation to Address/$\overline{CSn}$ Negation Specify the number of delay cycles from $\overline{RD}$ or $\overline{WEn}$ negation to address and $\overline{CSn}$ negation. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

(3) SDRAM

- CS2WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	A2CL[1:0]		—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8, 7	A2CL[1:0]	10	R/W	CAS Latency for Area 2 Specify the CAS latency for area 2. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

- CS3WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TRP[1:0]		—	TRCD[1:0]		—	A3CL[1:0]		—	—	TRWL[1:0]		—	TRC[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14, 13	TRP[1:0]	00	R/W	Number of Cycles from Auto-Precharge/PRE Command to ACTV Command Specify the number of minimum cycles from the start of auto-precharge or issuing of PRE command to the issuing of ACTV command for the same bank. The setting for areas 2 and 3 is common. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11, 10	TRCD[1:0]	01	R/W	Number of Cycles from ACTV Command to READ(A)/WRIT(A) Command Specify the number of minimum cycles from issuing ACTV command to issuing READ(A)/WRIT(A) command. The setting for areas 2 and 3 is common. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles

Bit	Bit Name	Initial Value	R/W	Description
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8, 7	A3CL[1:0]	10	R/W	CAS Latency for Area 3. Specify the CAS latency for area 3. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles When connecting the SDRAM to area 2 and area 3, set the CAS latency to the bits 8 and 7 in the CS2WCR register and the SDMR2 and SDMR3 registers for SDRAM mode setting. (See table 11.22.)
6	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4, 3	TRWL[1:0]	00	R/W	Number of Cycles from WRITA/WRIT Command to Auto-Precharge/PRE Command Specifies the number of cycles from issuing WRITA/WRIT command to the start of auto-precharge or to issuing PRE command. The setting for areas 2 and 3 is common. 00: 0 cycles 01: 1 cycle 10: 2 cycles 11: 3 cycles

Bit	Bit Name	Initial Value	R/W	Description
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	TRC[1:0]	00	R/W	Number of Cycles from REF Command/Self-Refresh Release to ACTV Command Specify the number of minimum cycles from issuing the REF command or releasing self-refresh to issuing the ACTV command. The setting for areas 2 and 3 is common. 00: 3 cycles 01: 4 cycles 10: 6 cycles 11: 9 cycles

Note: * If both areas 2 and 3 are specified as SDRAM, TRP1[1:0], TRCD[1:0], TRWL[1:0], and TRC[1:0] bit settings are common.
If only one area is connected to the SDRAM, specify area 3. In this case, specify area 2 as normal space or byte-selection SRAM.

(4) PCMCIA

- CS5BWCR, CS6BWCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	SA[1:0]		—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TED[3:0]				PCW[3:0]				WM	—	—	TEH[3:0]			
Initial value:	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	SA[1:0]	00	R/W	Space Attribute Specification Specify memory card interface or I/O card interface when the PCMCIA interface is selected. SA[1] 0: Specifies memory card interface when A25 = 1 1: Specifies I/O card interface when A25 = 1 SA[0] 0: Specifies memory card interface when A25 = 0 1: Specifies I/O card interface when A25 = 0
19 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14 to 11	TED[3:0]	00	R/W	Delay from Address to $\overline{RD}$ or $\overline{WE}$ Assert Specify the delay time from address output to $\overline{RD}$ or $\overline{WE}$ assert in PCMCIA interface. 0000: 0.5 cycle 0001: 1.5 cycles 0010: 2.5 cycles 0011: 3.5 cycles 0100: 4.5 cycles 0101: 5.5 cycles 0110: 6.5 cycles 0111: 7.5 cycles 1000: 8.5 cycles 1001: 9.5 cycles 1010: 10.5 cycles 1011: 11.5 cycles 1100: 12.5 cycles 1101: 13.5 cycles 1110: 14.5 cycles 1111: 15.5 cycles
10 to 7	PCW[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted. 0000: 3 cycles 0001: 6 cycles 0010: 9 cycles 0011: 12 cycles 0100: 15 cycles 0101: 18 cycles 0110: 22 cycles 0111: 26 cycles 1000: 30 cycles 1001: 33 cycles 1010: 36 cycles 1011: 38 cycles 1100: 52 cycles 1101: 60 cycles 1110: 64 cycles 1111: 80 cycles

Bit	Bit Name	Initial Value	R/W	Description
6	WM	1	R/W	External Wait Mask Specification Specify whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	TEH[3:0]	0000	R/W	Delay from $\overline{RD}$ or $\overline{WE}$ Negate to Address Specify the address hold time from $\overline{RD}$ or $\overline{WE}$ negate in the PCMCIA interface. 0000: 0.5 cycle 0001: 1.5 cycles 0010: 2.5 cycles 0011: 3.5 cycles 0100: 4.5 cycles 0101: 5.5 cycles 0110: 6.5 cycles 0111: 7.5 cycles 1000: 8.5 cycles 1001: 9.5 cycles 1010: 10.5 cycles 1011: 11.5 cycles 1100: 12.5 cycles 1101: 13.5 cycles 1110: 14.5 cycles 1111: 15.5 cycles

11.4.4 Reset Bus Wait Counter (RBWTCNT)

RBWTCNT is a 7-bit counter. The counter is cleared to 0 by a power-on reset and it maintains the 0 state during the reset period. This counter starts to increment in synchronization with the CKO pin after a power-on reset is cleared, and stops when the value reaches H'7F. External bus access is suspended while the counter is operating. This counter is provided to assume the minimum time from clearing a reset for flash memory and the like to the first access. This counter cannot be read from or written to.

11.4.5 SDRAM Control Register (SDCR)

SDCR specifies the method to refresh and access SDRAM, and the types of SDRAMs to be connected.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	A2ROW[1:0]		—	A2COL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RFSH	RMODE	PDOWN	BACTV	—	—	—	A3ROW[1:0]		—	A3COL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20, 19	A2ROW [1:0]	00	R/W	Number of Bits of Row Address for Area 2 Specify the number of bits of row address for area 2. 00: 11 bits 01: 12 bits 10: 13 bits 11: Setting prohibited
18	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
17, 16	A2COL [1:0]	00	R/W	Number of Bits of Column Address for Area 2 Specify the number of bits of column address for area 2. 00: 8 bits 01: 9 bits 10: 10 bits 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	RFSH	0	R/W	Refresh Control Specifies whether or not the refresh operation of the SDRAM is performed. 0: No refresh 1: Refresh
10	RMODE	0	R/W	Refresh Control Specifies whether to perform auto-refresh or self-refresh when the RFSH bit is 1. When the RFSH bit is 1 and this bit is 1, self-refresh starts immediately. When the RFSH bit is 1 and this bit is 0, auto-refresh starts according to the contents that are set in RTCSR, RTCNT, and RTCOR. 0: Auto-refresh is performed 1: Self-refresh is performed
9	PDOWN	0	R	Power-Down Mode Specifies whether the SDRAM is entered in power-down mode or not after the access to SDRAM is completed. If this bit is set to 1, the CKE pin is pulled to low to place the SDRAM to power-down mode. 0: Does not place the SDRAM in power-down mode after access completion. 1: Places the SDRAM in power-down mode after access completion.
8	BACTV	0	R/W	Bank Active Mode Specifies to access whether in auto-precharge mode (using READA and WRITA commands) or in bank active mode (using READ and WRIT commands). 0: Auto-precharge mode (using READA and WRITA commands) 1: Bank active mode (using READ and WRIT commands) Note: Bank active mode can be used only in area 3. In this case, the bus width can be selected as 16 or 32 bits. When both areas 2 and 3 are set to SDRAM, specify auto-precharge mode.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4, 3	A3ROW [1:0]	00	R/W	Number of Bits of Row Address for Area 3 Specify the number of bits of the row address for area 3. 00: 11 bits 01: 12 bits 10: 13 bits 11: Setting prohibited
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	A3COL [1:0]	00	R/W	Number of Bits of Column Address for Area 3 Specify the number of bits of the column address for area 3. 00: 8 bits 01: 9 bits 10: 10 bits 11: Setting prohibited

11.4.6 Refresh Timer Control/Status Register (RTCSR)

RTCSR specifies various items about refresh for SDRAM.

When RTCSR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CMF	—	CKS[2:0]			RRC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be H'A55A00.
7	CMF	0	R/W	Compare Match Flag Indicates that a compare match occurs between the refresh timer counter (RTCNT) and refresh time constant register (RTCOR). This bit is set or cleared in the following conditions. 0: Clearing condition: When 0 is written in CMF after reading out RTCSR during CMF = 1. 1: Setting condition: When the condition RTCNT = RTCOR is satisfied.
6	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5 to 3	CKS[2:0]	000	R/W	Clock Select Select the clock input to count-up the refresh timer counter (RTCNT). 000: Stop the counting-up 001: Bϕ/4 010: Bϕ/16 011: Bϕ/64 100: Bϕ/256 101: Bϕ/1024 110: Bϕ/2048 111: Bϕ/4096
2 to 0	RRC[2:0]	000	R/W	Refresh Count Specify the number of continuous refresh cycles, when the refresh request occurs after the coincidence of the values of the refresh timer counter (RTCNT) and the refresh time constant register (RTCOR). These bits can make the period of occurrence of refresh long. 000: Once 001: Twice 010: 4 times 011: 6 times 100: 8 times 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

11.4.7 Refresh Timer Counter (RTCNT)

RTCNT is an 8-bit counter that increments using the clock selected by bits CKS[2:0] in RTCSR. When RTCNT matches RTCOR, RTCNT is cleared to 0. The value in RTCNT returns to 0 after counting up to 255. When the RTCNT is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CNT[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be H'A55A00.
7 to 0	CNT	All 0	R/W	8-Bit Counter

11.4.8 Refresh Time Constant Register (RTCOR)

RTCOR contains the value for comparison with that of the 8-bit counter. RTCOR matches RTCNT, the CMF bit in RTCSR is set to 1 and RTCNT is cleared to 0

When the RFSH bit in SDCR is 1, a memory refresh request is issued by this matching signal. This request is maintained until the refresh operation is performed. If the request is not processed when the next matching occurs, the previous request is ignored.

Clearing the CMF bit does not clear refresh requests. When the RTCOR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	COR[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be H'A55A00.
7 to 0	COR	All 0	R/W	Value for Comparison with That of the 8-Bit Counter

11.4.9 SDRAM Mode Registers 2, 3 (SDMR2 and SRMR3)

For the settings of SDRAM mode registers (SDMR2 and SDMR3), see table 11.23.

11.5 Operation

11.5.1 Endian/Access Size and Data Alignment

This LSI supports big endian, in which the 0 address is the most significant byte (MSByte) in the byte data and little endian, in which the 0 address is the least significant byte (LSByte) in the byte data. Endian is specified on power-on reset by the external pin (MD5). When MD5 pin is low level on power-on reset, the endian will become big endian and when MD5 pin is high level on power-on reset, the endian will become little endian.

Three data bus widths (8 bits, 16 bits, and 32 bits) are available for normal memory and byte-selection SRAM. Two data bus widths (16 bits and 32 bits) are available for SDRAM. Two data bus widths (8 bits and 16 bits) are available for PCMCIA interface. Data alignment is performed in accordance with the data bus width of the device and endian. This also means that when longword data is read from a byte-width device, the read operation must be done four times. In this LSI, data alignment and conversion of data length is performed automatically between the respective interfaces.

Tables 11.9 to 11.14 show the relationship between endian, device data width, and access unit.

Table 11.9 32-Bit External Device/Big Endian Access and Data Alignment

	Data Bus				Strobe Signals			
Operation	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$/ DQMUU	$\overline{WE2}$/ DQMUL	$\overline{WE1}$/ DQMLU	$\overline{WE0}$/ DQMLL
Byte access at 0	Data 7 to 0	—	—	—	Assert	—	—	—
Byte access at 1	—	Data 7 to 0	—	—	—	Assert	—	—
Byte access at 2	—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 3	—	—	—	Data 7 to 0	—	—	—	Assert
Word access at 0	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert	—	—
Word access at 2	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Longword access at 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

Table 11.10 16-Bit External Device/Big Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$/ DQMUU	$\overline{WE2}$/ DQMUL	$\overline{WE1}$/ DQMLU	$\overline{WE0}$/ DQMLL
Byte access at 0		—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 1		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 2		—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 3		—	—	—	Data 7 to 0	—	—	—	Assert
Word access at 0		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Word access at 2		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Longword access at 0	1st time at 0	—	—	Data 31 to 24	Data 23 to 16	—	—	Assert	Assert
	2nd time at 2	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert

Table 11.11 8-Bit External Device/Big Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$/ DQMUU**	**$\overline{WE2}$/ DQMUL**	**$\overline{WE1}$/ DQMLU**	**$\overline{WE0}$/ DQMLL**
Byte access at 0		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 1		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 2		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 3		—	—	—	Data 7 to 0	—	—	—	Assert
Word access at 0	1st time at 0	—	—	—	Data 15 to 8	—	—	—	Assert
	2nd time at 1	—	—	—	Data 7 to 0	—	—	—	Assert
Word access at 2	1st time at 2	—	—	—	Data 15 to 8	—	—	—	Assert
	2nd time at 3	—	—	—	Data 7 to 0	—	—	—	Assert
Longword access at 0	1st time at 0	—	—	—	Data 31 to 24	—	—	—	Assert
	2nd time at 1	—	—	—	Data 23 to 16	—	—	—	Assert
	3rd time at 2	—	—	—	Data 15 to 8	—	—	—	Assert
	4th time at 3	—	—	—	Data 7 to 0	—	—	—	Assert

Table 11.12 32-Bit External Device/Little Endian Access and Data Alignment

	Data Bus				Strobe Signals			
Operation	**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$/ DQMUU**	**$\overline{WE2}$/ DQMUL**	**$\overline{WE1}$/ DQMLU**	**$\overline{WE0}$/ DQMLL**
Byte access at 0	—	—	—	Data 7 to 0	—	—		Assert
Byte access at 1	—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 2	—	Data 7 to 0	—	—	—	Assert	—	—
Byte access at 3	Data 7 to 0	—	—	—	Assert	—	—	—
Word access at 0	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Word access at 2	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert	—	—
Longword access at 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

Table 11.13 16-Bit External Device/Little Endian Access and Data Alignment

Operation		Data Bus				Strobe Signals			
		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$/ DQMUU	$\overline{WE2}$/ DQMUL	$\overline{WE1}$/ DQMLU	$\overline{WE0}$/ DQMLL
Byte access at 0		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 1		—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 2		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 3		—	—	Data 7 to 0	—	—	—	Assert	—
Word access at 0		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Word access at 2		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Longword access at 0	1st time at 0	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
	2nd time at 1	—	—	Data 31 to 24	Data 23 to 16	—	—	Assert	Assert

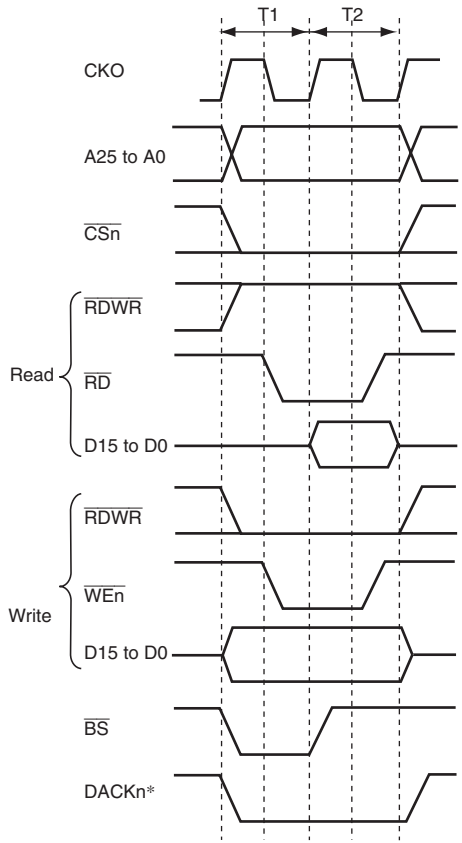
Table 11.14 8-Bit External Device/Little Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$/ DQMUU	$\overline{WE2}$/ DQMUL	$\overline{WE1}$/ DQMLU	$\overline{WE0}$/ DQMLL
Byte access at 0		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 1		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 2		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 3		—	—	—	Data 7 to 0	—	—	—	Assert
Word access at 0	1st time at 0	—	—	—	Data 7 to 0	—	—	—	Assert
	2nd time at 1	—	—	—	Data 15 to 8	—	—	—	Assert
Word access at 2	1st time at 2	—	—	—	Data 7 to 0	—	—	—	Assert
	2nd time at 3	—	—	—	Data 15 to 8	—	—	—	Assert
Longword access at 0	1st time at 0	—	—	—	Data 7 to 0	—	—	—	Assert
	2nd time at 1	—	—	—	Data 15 to 8	—	—	—	Assert
	3rd time at 2	—	—	—	Data 23 to 16	—	—	—	Assert
	4th time at 3	—	—	—	Data 31 to 24	—	—	—	Assert

11.5.2 Normal Space Interface

(1) Basic Timing

For access to a normal space, this LSI uses strobe signal output in consideration of the fact that mainly static RAM will be connected. When using SRAM with a byte-selection pin, see section 11.5.7, Byte-Selection SRAM Interface. Figure 11.3 shows the basic timings of normal space access. A no-wait normal access is completed in two cycles. The $\overline{BS}$ signal is asserted for one cycle to indicate the start of a bus cycle.



Note: * The waveform for DACKn is when active low is specified.

Figure 11.3 Normal Space Basic Access Timing (Access Wait 0)

There is no access size specification when reading. The correct access start address is output in the least significant bit of the address, but since there is no access size specification, 32 bits are always read in case of a 32-bit device, and 16 bits in case of a 16-bit device. When writing, only the $\overline{\text{WEn}}$ signal for the byte to be written is asserted.

It is necessary to output the data that has been read using $\overline{\text{RD}}$ when a buffer is established in the data bus. The $\overline{\text{RDWR}}$ signal is in a read state (high output) when no access has been carried out. Therefore, care must be taken when controlling the external data buffer, to avoid collision.

Figures 11.4 and 11.5 show the basic timings of normal space accesses. If the WM bit of the CSnWCR is cleared to 0, a Tnop cycle is inserted to evaluate the external wait (figure 11.4). If the WM bit of the CSnWCR is set to 1, external waits are ignored and no Tnop cycle is inserted (figure 11.5).

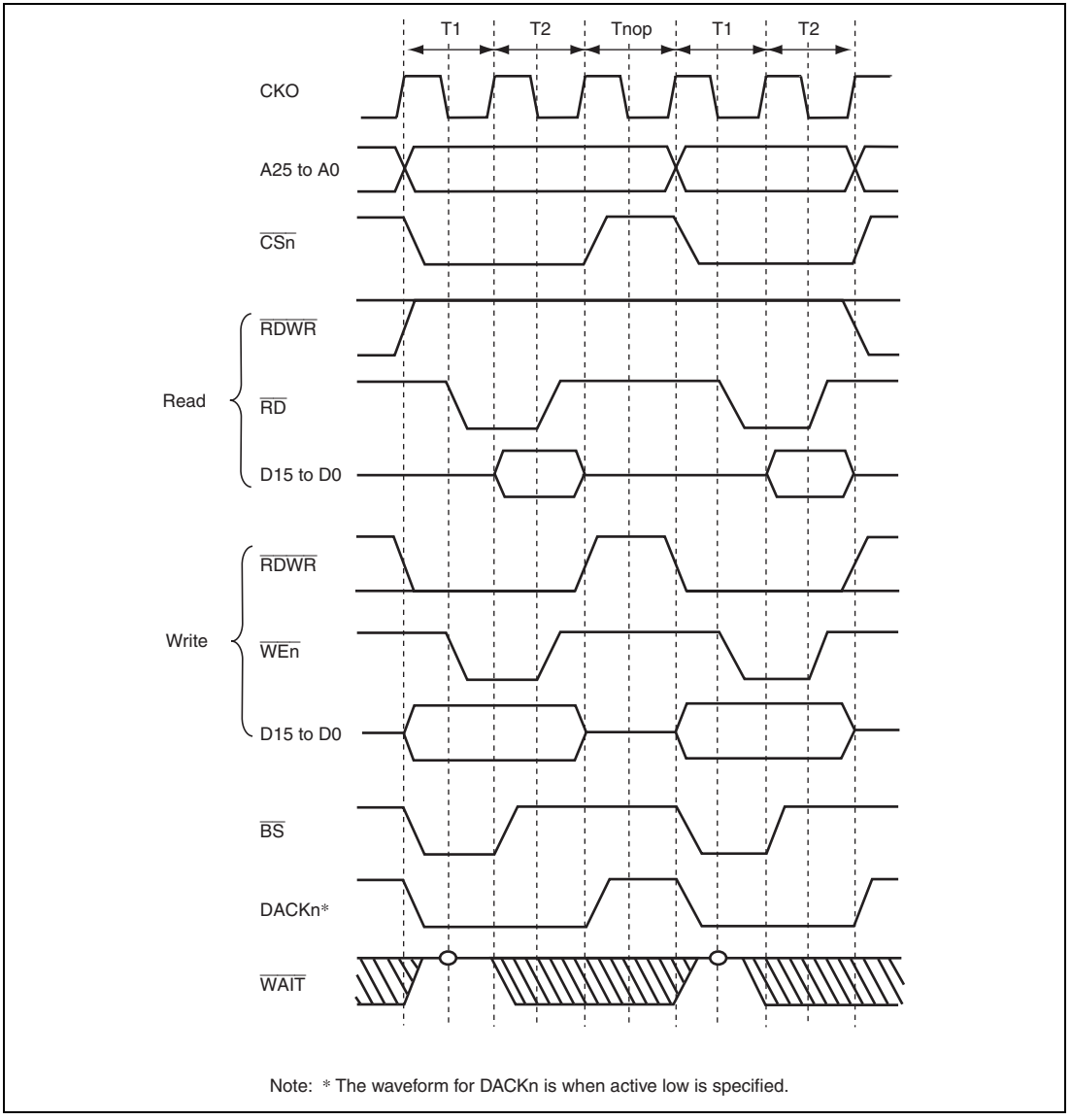


Figure 11.4 Continuous Access for Normal Space 1, Bus Width = 16 bits, Longword Access, CSnWCR.WM Bit = 0 (Access Wait = 0, Cycle Wait = 0)

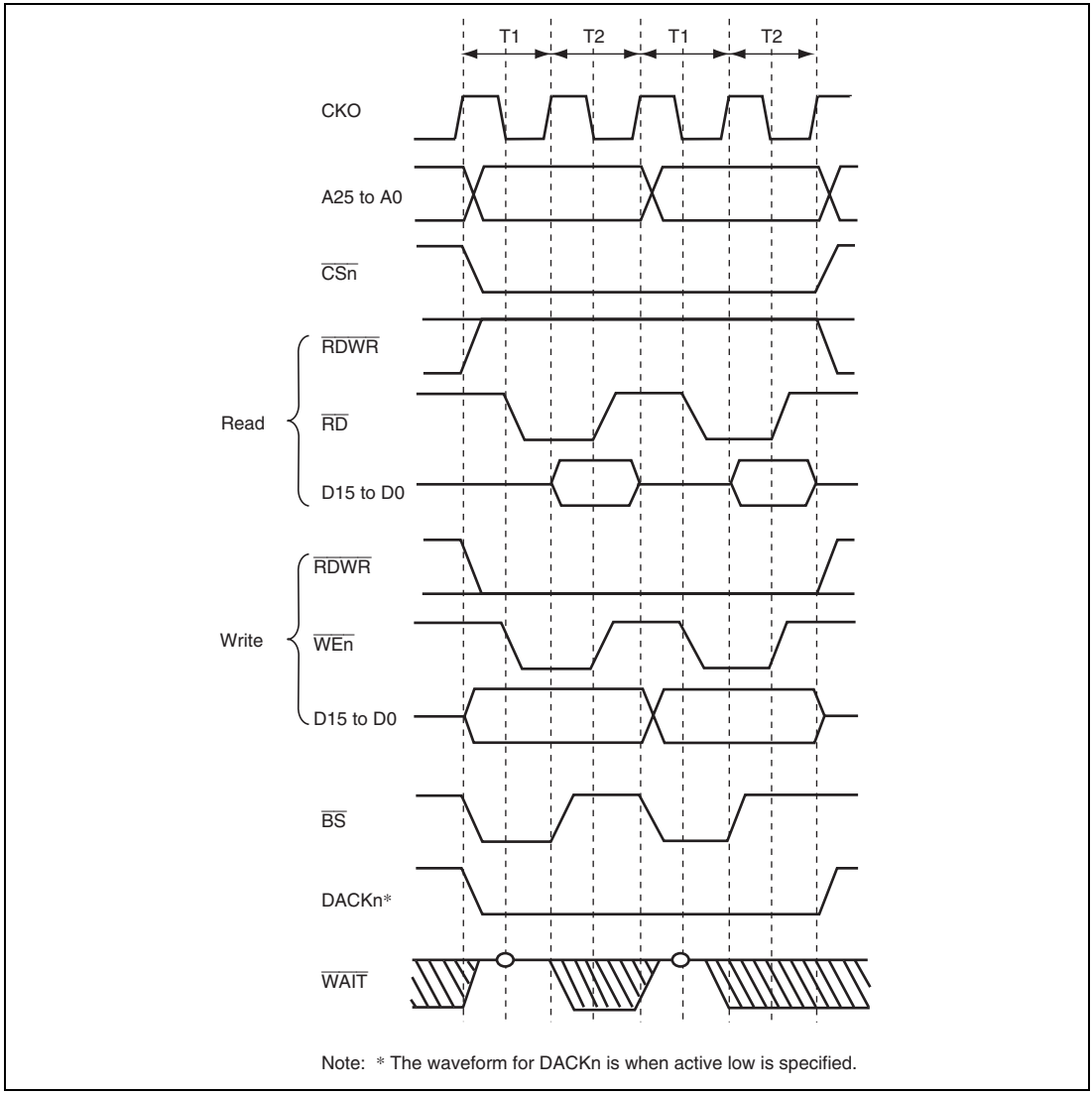


Figure 11.5 Continuous Access for Normal Space 2, Bus Width = 16 bits, Longword Access, CSnWCR.WM Bit = 1 (Access Wait = 0, Cycle Wait = 0)

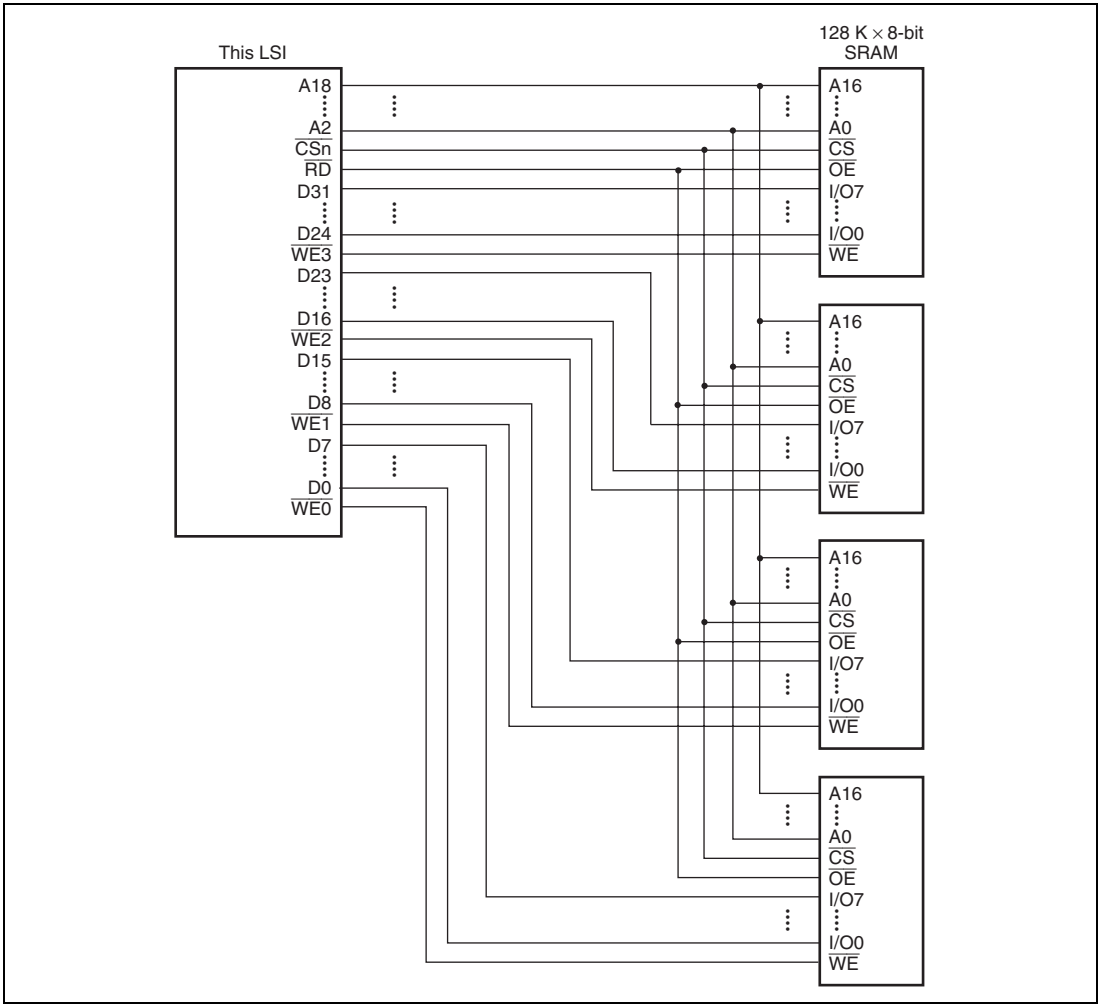


Figure 11.6 Example of 32-Bit Data-Width SRAM Connection

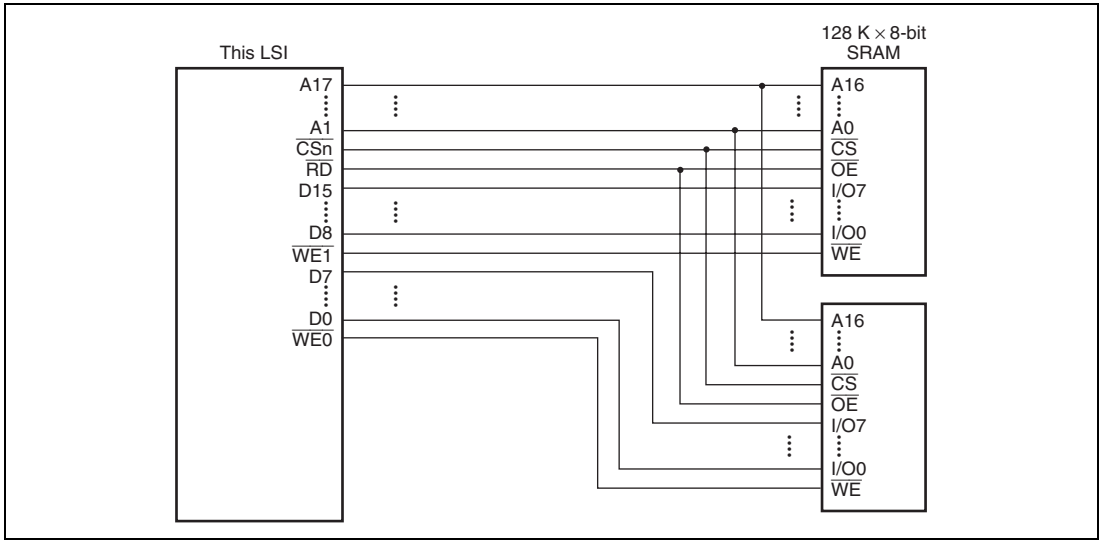


Figure 11.7 Example of 16-Bit Data-Width SRAM Connection

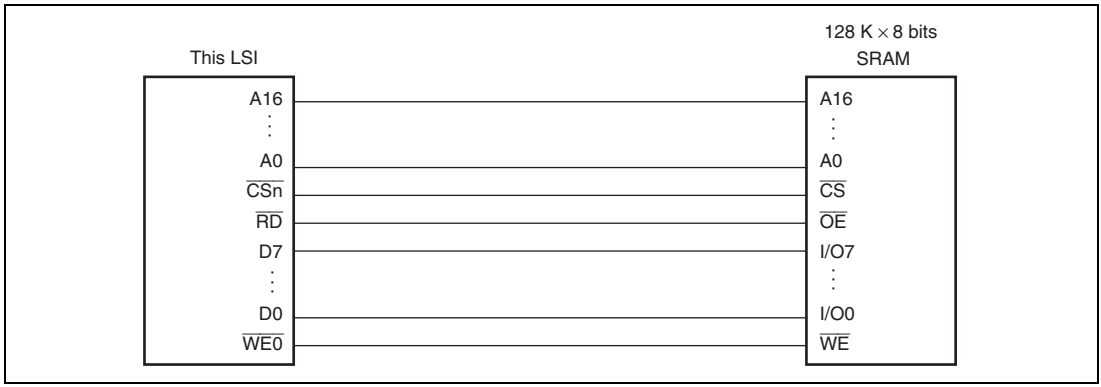


Figure 11.8 Example of 8-Bit Data-Width SRAM Connection

11.5.3 Access Wait Control

Wait-cycle insertion on access to a normal space or byte-selection SRAM is controlled by the settings of bits WR[3:0] and WW[2:0] in CSnWCR. Values for wait-cycle insertion in read access and write access are independent of each other. In the case of access to a normal space or byte-selection SRAM, the specified number of cycles is inserted as Tw, i.e. as the wait cycle, as shown in figure 11.9.

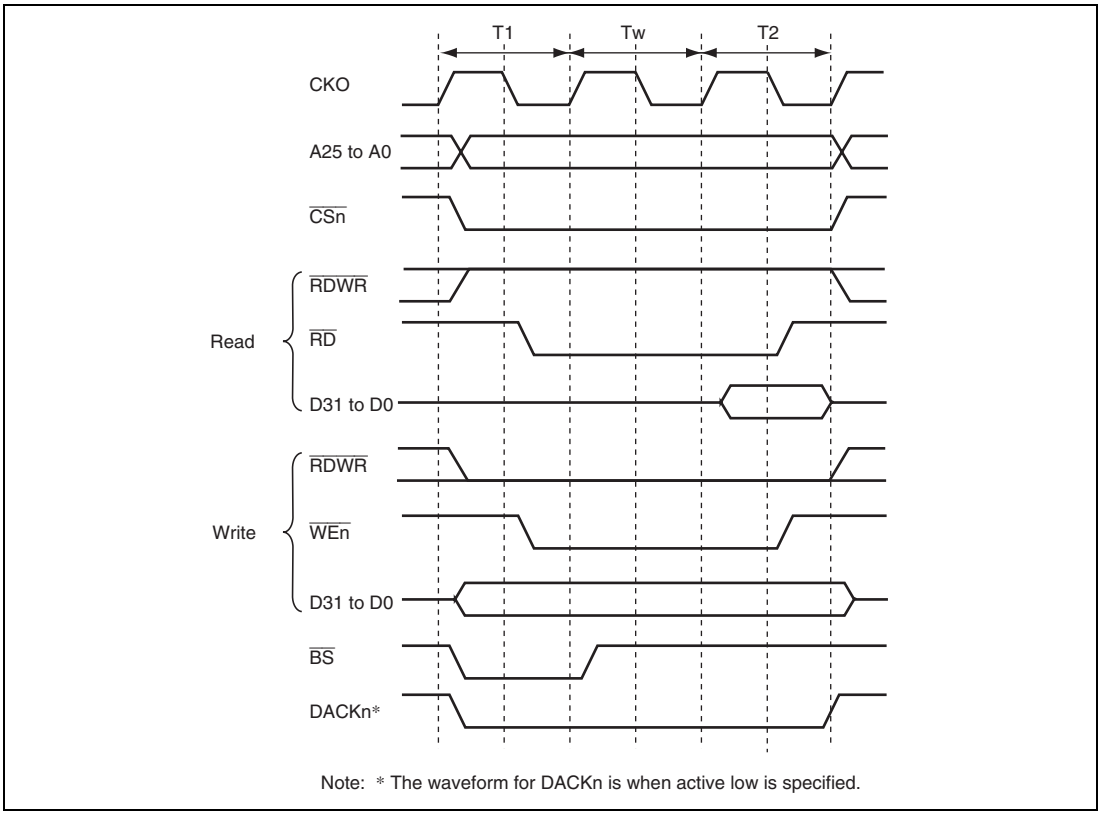


Figure 11.9 Wait Timing for Normal Space Access (Software Wait Only)

When the WM bit in CSnWCR is cleared to 0, the external wait input $\overline{\text{WAIT}}$ signal is also sampled. $\overline{\text{WAIT}}$ pin sampling is shown in figure 11.10. A 2-cycle wait is specified as a software wait. The $\overline{\text{WAIT}}$ signal is sampled on the falling edge of CKO at the transition from the T1 or Tw cycle to the T2 cycle.

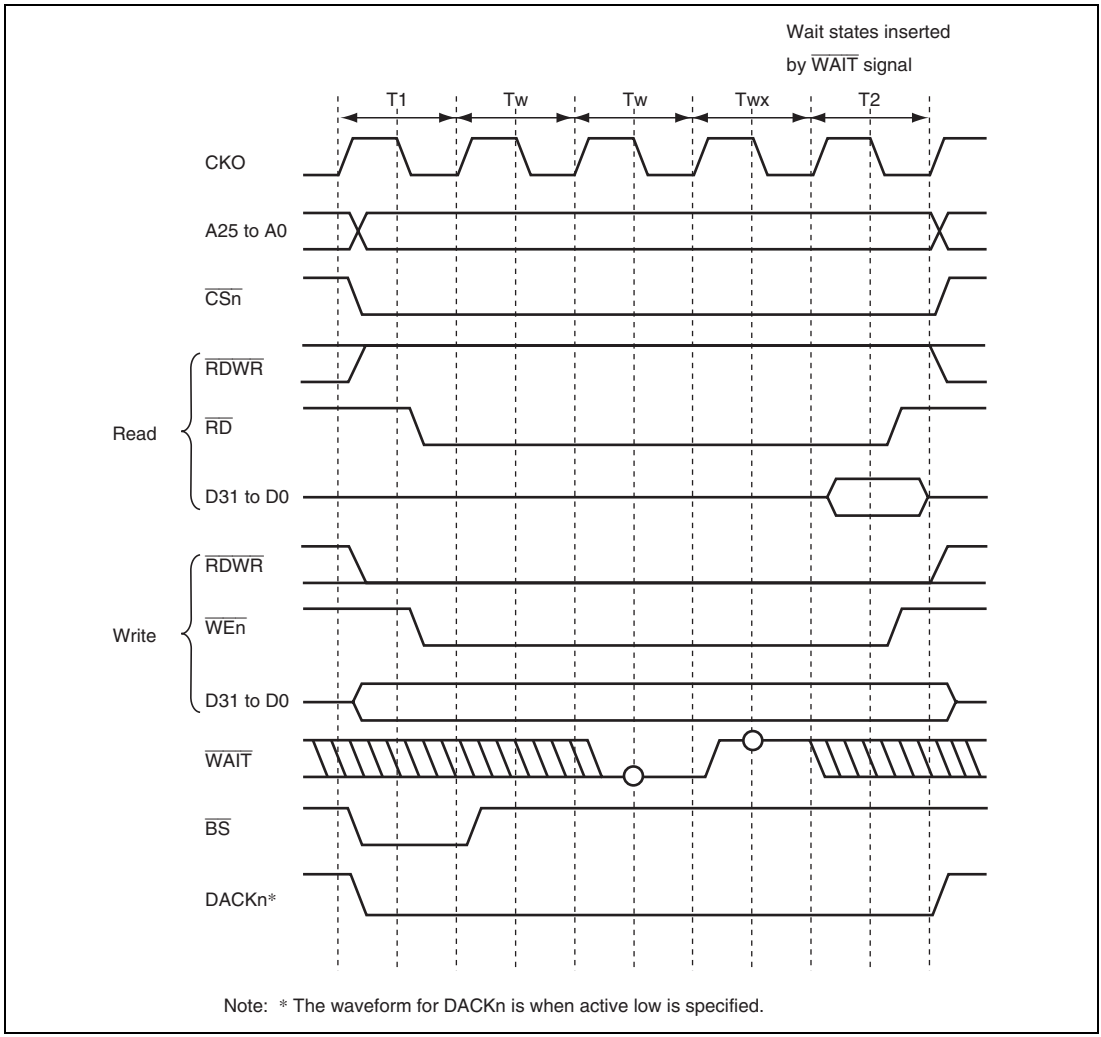


Figure 11.10 Wait State Timing for Normal Space Access (Wait State Insertion using $\overline{\text{WAIT}}$ Signal)

11.5.4 $\overline{\text{CSn}}$ Assert Period Expansion

The number of cycles from $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ assertion can be specified by setting bits SW[1:0] in CSnWCR. The number of cycles from $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ negation to $\overline{\text{CSn}}$ negation can be specified by setting bits HW[1:0]. Therefore, a flexible interface to an external device can be obtained. Figure 11.11 shows an example. A Th cycle and a Tf cycle are added before and after an ordinary cycle, respectively. In these cycles, $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ are not asserted, while other signals are asserted. The data output is prolonged to the Tf cycle, and this prolongation is useful for devices with slow writing operations.

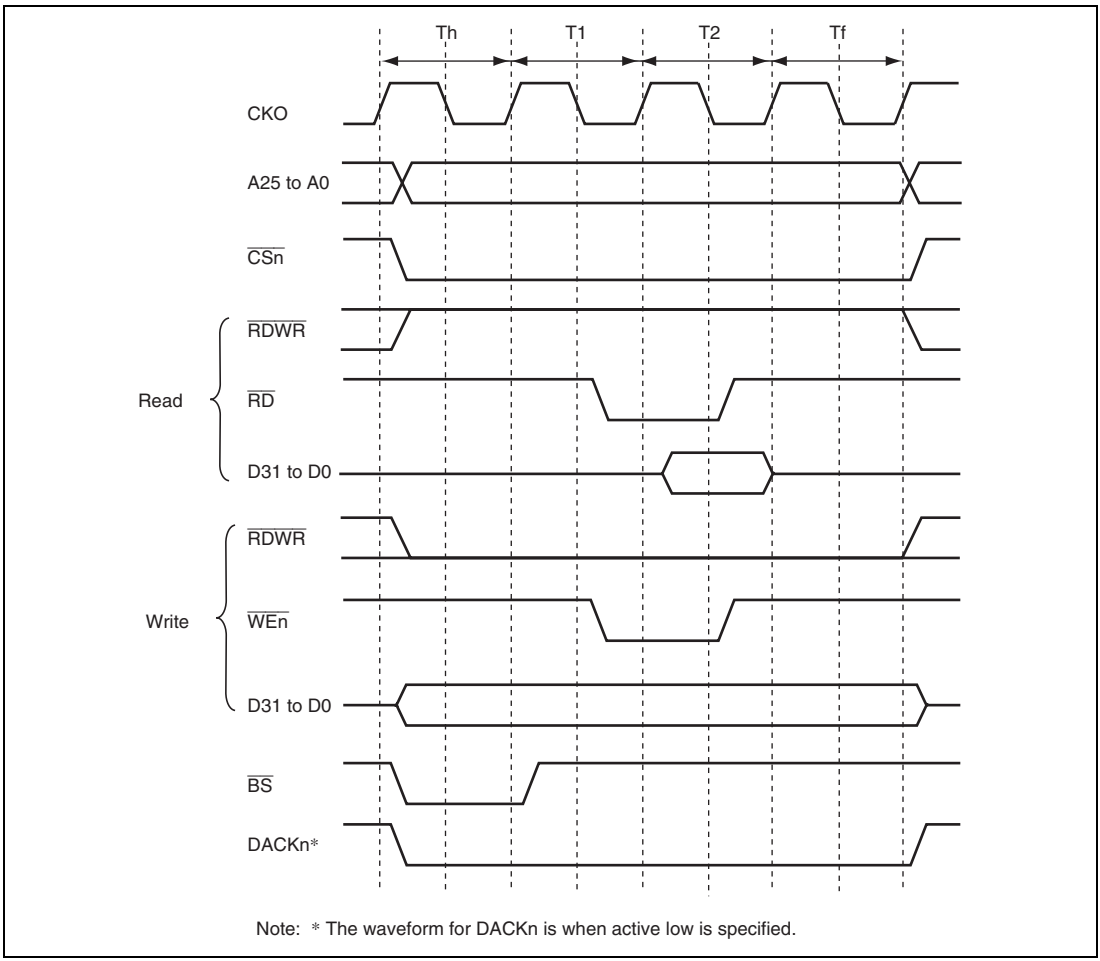


Figure 11.11 $\overline{\text{CSn}}$ Assert Period Expansion

11.5.5 SDRAM Interface

(1) SDRAM Connection

The SDRAM that can be connected to this LSI is a product that has 11/12/13 bits of row address, 8/9/10 bits of column address, 4 or less banks, and uses the A10 pin for setting precharge mode in read and write command cycles.

The control signals for connection of SDRAM are $\overline{RAS}$, $\overline{CAS}$, $\overline{RDWR}$, DQMUU, DQMUL, DQMLU, DQMLL, CKE, $\overline{CS2}$, and $\overline{CS3}$. All the signals other than $\overline{CS2}$ and $\overline{CS3}$ are common to all areas, and signals other than CKE are valid when $\overline{CS2}$ or $\overline{CS3}$ is asserted. SDRAM can be connected to up to 2 spaces. The data bus width of the area that is connected to SDRAM can be set to 32 or 16 bits.

Burst read/single write (burst length 1) and burst read/burst write (burst length 1) are supported as the SDRAM operating mode.

Commands for SDRAM can be specified by $\overline{RAS}$, $\overline{CAS}$, $\overline{RDWR}$, and specific address signals. These commands are shown below.

- NOP
- Auto-refresh (REF)
- Self-refresh (SELF)
- All banks precharge (PALL)
- Specified bank precharge (PRE)
- Bank active (ACTV)
- Read (READ)
- Read with precharge (READA)
- Write (WRIT)
- Write with precharge (WRITA)
- Write mode register (MRS)

The byte to be accessed is specified by DQMUU, DQMUL, DQMLU, and DQMLL. Reading or writing is performed for a byte whose corresponding DQMxx is low. For details on the relationship between DQMxx and the byte to be accessed, refer to section 11.5.1, Endian/Access Size and Data Alignment.

When connecting only one area to SDRAM, specify area 3 as the SDRAM space. With this setting, specify area 2 as a normal memory area or byte-selection SRAM.

Figures 11.12 and 11.13 show examples of the connection of the SDRAM with the LSI.

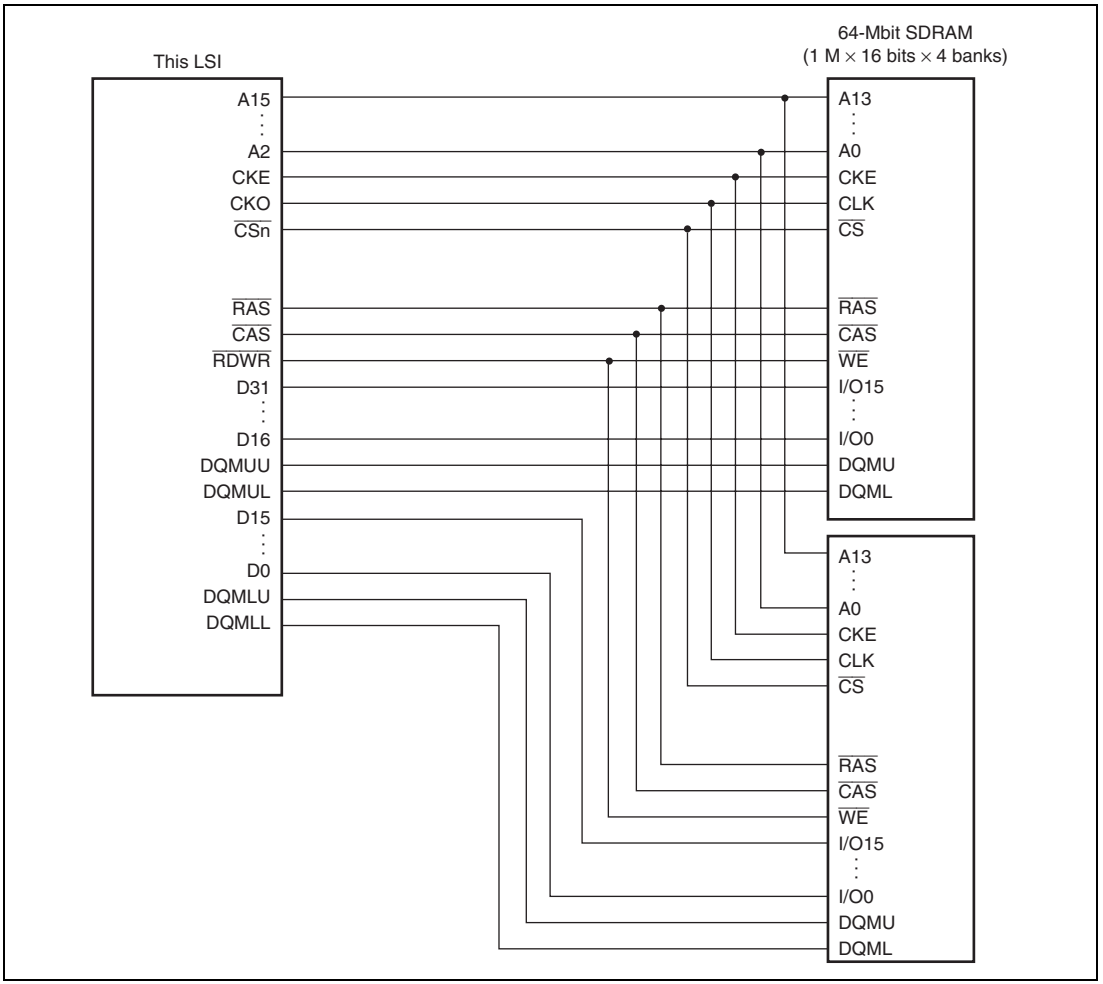


Figure 11.12 Example of 32-Bit Data-Width SDRAM Connection

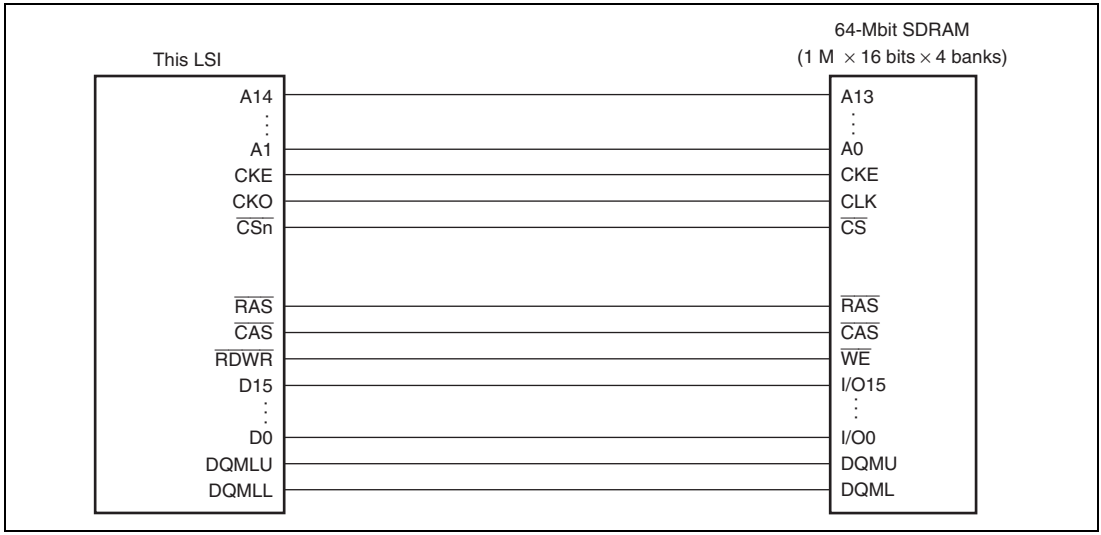


Figure 11.13 Example of 16-Bit Data-Width SDRAM Connection

(2) Address Multiplexing

An address multiplexing is specified so that SDRAM can be connected without external multiplexing circuitry according to the setting of bits BSZ[1:0]in CSnBCR, AxROW[1:0] and AxCOL[1:0] in SDCR. Tables 11.15 to 11.20 show the relationship between the settings of bits BSZ[1:0], AxROW[1:0], and AxCOL[1:0] and the bits output at the address pins. Do not specify those bits in the manner other than this table, otherwise the operation of this LSI is not guaranteed. A25 to A18 are not multiplexed and the original values of address are always output at these pins.

When the data bus width is 16 bits (BSZ[1:0] =B'10), A0 of SDRAM specifies a word address. Therefore, connect this A0 pin of SDRAM to the A1 pin of the LSI; the A1 pin of SDRAM to the A2 pin of the LSI, and so on. When the data bus width is 32 bits (BSZ[1:0] =B'11), the A0 pin of SDRAM specifies a longword address. Therefore, connect this A0 pin of SDRAM to the A2 pin of the LSI; the A1 pin of SDRAM to the A3 pin of the LSI, and so on.

Table 11.15 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)-1

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*[2]	A22*[2]	A12 (BA1)	Specifies bank
A13	A21*[2]	A21*[2]	A11 (BA0)	
A12	A20	L/H*[1]	A10/AP	Specifies address/precharge
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		Unused
A0	A8	A0		
Example of connected memory				
64-Mbit product (512 kwords x 32 bits x 4 banks, column 8 bits product): 1				
16-Mbit product (512 kwords x 16 bits x 2 banks, column 8 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 11.15 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)-2

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**01 (12 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A24	A17		Unused
A16	A23	A16		
A15	A23*2	A23*2	A13 (BA1)	Specifies bank
A14	A22*2	A22*2	A12 (BA0)	
A13	A21	A13	A11	Address
A12	A20	L/H*1	A10/AP	Specifies address/precharge
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		Unused
A0	A8	A0		
Example of connected memory				
128-Mbit product (1 Mword x 32 bits x 4 banks, column 8 bits product): 1				
64-Mbit product (1 Mword x 16 bits x 4 banks, column 8 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 11.16 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (2)-1

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**01 (12 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*[2]	A24*[2]	A13 (BA1)	Specifies bank
A14	A23*[2]	A23*[2]	A12 (BA0)	
A13	A22	A13	A11	Address
A12	A21	L/H*[1]	A10/AP	Specifies address/precharge
A11	A20	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of connected memory				
256-Mbit product (2 Mwords x 32 bits x 4 banks, column 9 bits product): 1				
128-Mbit product (2 Mwords x 16 bits x 4 banks, column 9 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 11.16 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (2)-2

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**01 (12 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*2	A25*2	A13 (BA1)	Specifies bank
A14	A24*2	A24*2	A12 (BA0)	
A13	A23	A13	A11	Address
A12	A22	L/H*1	A10/AP	Specifies address/precharge
A11	A21	A11	A9	Address
A10	A20	A10	A8	
A9	A19	A9	A7	
A8	A18	A8	A6	
A7	A17	A7	A5	
A6	A16	A6	A4	
A5	A15	A5	A3	
A4	A14	A4	A2	
A3	A13	A3	A1	
A2	A12	A2	A0	
A1	A11	A1		Unused
A0	A10	A0		
Example of connected memory				
512-Mbit product (4 Mwords x 32 bits x 4 banks, column 10 bits product): 1				
256-Mbit product (4 Mwords x 16 bits x 4 banks, column 10 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 11.17 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (3)

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**10 (13 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25*[2]	A25*[2]	A14 (BA1)	Specifies bank
A15	A24*[2]	A24*[2]	A13 (BA0)	
A14	A23	A14	A12	Address
A13	A22	A13	A11	
A12	A21	L/H*[1]	A10/AP	Specifies address/precharge
A11	A20	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of connected memory				
512-Mbit product (4 Mwords x 32 bits x 4 banks, column 9 bits product): 1				
256-Mbit product (4 Mwords x 16 bits x 4 banks, column 9 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 11.18 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (4)-1

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21*2	A21*2	A12 (BA1)	Specifies bank
A12	A20*2	A20*2	A11 (BA0)	
A11	A19	L/H*1	A10/AP	Specifies address/precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
16-Mbit product (512 kwords x 16 bits x 2 banks, column 8 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 11.18 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (4)-2

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*[2]	A22*[2]	A13 (BA1)	Specifies bank
A13	A21*[2]	A21*[2]	A12 (BA0)	
A12	A20	A12	A11	Address
A11	A19	L/H*[1]	A10/AP	Specifies address/precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
64-Mbit product (1 Mword x 16 bits x 4 banks, column 8 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 11.19 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (5)-1

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24	A15		
A14	A23*2	A23*2	A13 (BA1)	Specifies bank
A13	A22*2	A22*2	A12 (BA0)	
A12	A21	A12	A11	Address
A11	A20	L/H*1	A10/AP	Specifies address/precharge
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
128-Mbit product (2 Mwords x 16 bits x 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 11.19 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (5)-2

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25	A15		
A14	A24*[2]	A24*[2]	A13 (BA1)	Specifies bank
A13	A23*[2]	A23*[2]	A12 (BA0)	
A12	A22	A12	A11	Address
A11	A21	L/H*[1]	A10/AP	Specifies address/precharge
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords x 16 bits x 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 11.20 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (6)-1

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**10 (13 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*2	A24*2	A14 (BA1)	Specifies bank
A14	A23*2	A23*2	A13 (BA0)	
A13	A22	A13	A12	Address
A12	A21	A12	A11	
A11	A20	L/H*1	A10/AP	Specifies address/precharge
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords x 16 bits x 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 11.20 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (6)-2

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**10 (13 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*2	A25*2	A14 (BA1)	Specifies bank
A14	A24*2	A24*2	A13 (BA0)	
A13	A23	A13	A12	Address
A12	A22	A12	A11	
A11	A21	L/H*1	A10/AP	Specifies address/precharge
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
512-Mbit product (8 Mwords x 16 bits x 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 11.21 Relationship between A3BSZ[1:0], A3ROW[1:0], A3COL[1:0], and Address Multiplex Output (7)

Setting				
A3 BSZ [1:0]	**A3 ROW [1:0]**	**A3 COL [1:0]**		
11 (32 bits)	**10 (13 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26*2	A26*2	A14 (BA1)	Specifies bank
A15	A25*2	A25*2	A13 (BA0)	
A14	A24*2	A14	A12	Address
A13	A23	A13	A11	
A12	A22	L/H*1	A10/AP	Specifies address/precharge
A11	A21	A11	A9	Address
A10	A20	A10	A8	
A9	A19	A9	A7	
A8	A18	A8	A6	
A7	A17	A7	A5	
A6	A16	A6	A4	
A5	A15	A5	A3	
A4	A14	A4	A2	
A3	A13	A3	A1	
A2	A12	A2	A0	
A1	A11	A1		Unused
A0	A10	A0		
Example of connected memory				
512-Mbit product (8 Mwords x 16 bits x 4 banks, column 10 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

(3) Burst Read

A burst read occurs in the following cases with this LSI.

1. Access size in reading is larger than data bus width.
2. 32-byte transfer in cache miss.
3. 8/16/32-byte transfer in DMAC (access to non-cacheable area)

This LSI always accesses the SDRAM with burst length 1. For example, read access of burst length 1 is performed consecutively four times to read 16-byte continuous data from the SDRAM that is connected to a 32-bit data bus.

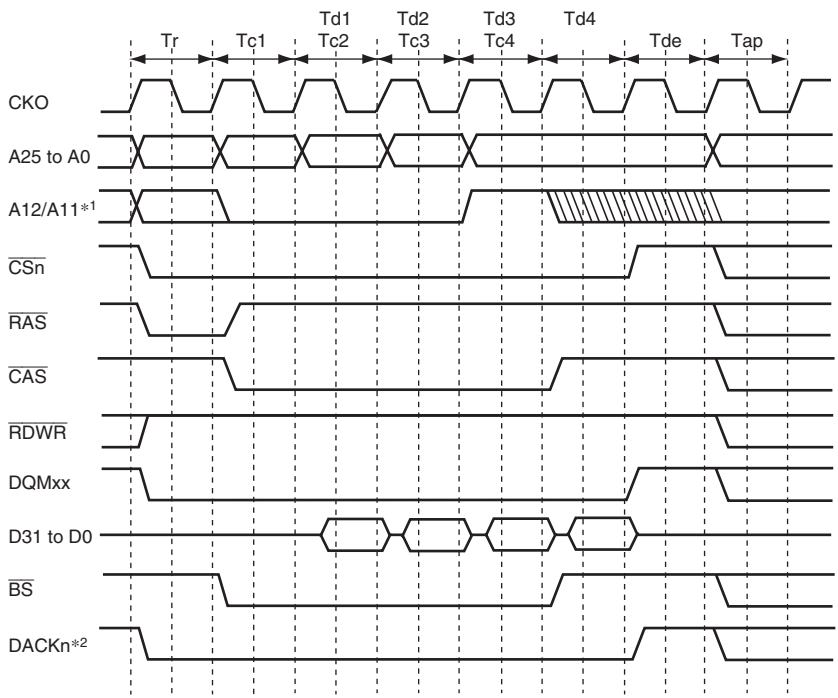
Table 11.22 shows the relationship between the access size and the number of bursts.

Table 11.22 Relationship between Access Size and Number of Bursts

Bus Width	Access Size	Number of Bursts
16 bits	8 bits	1
	16 bits	1
	32 bits	2
	8 bytes	4
	16 bytes	8
	32 bytes	8 x 2
32 bits	8 bits	1
	16 bits	1
	32 bits	1
	8 bytes	2
	16 bytes	4
	32 bytes	4 x 2

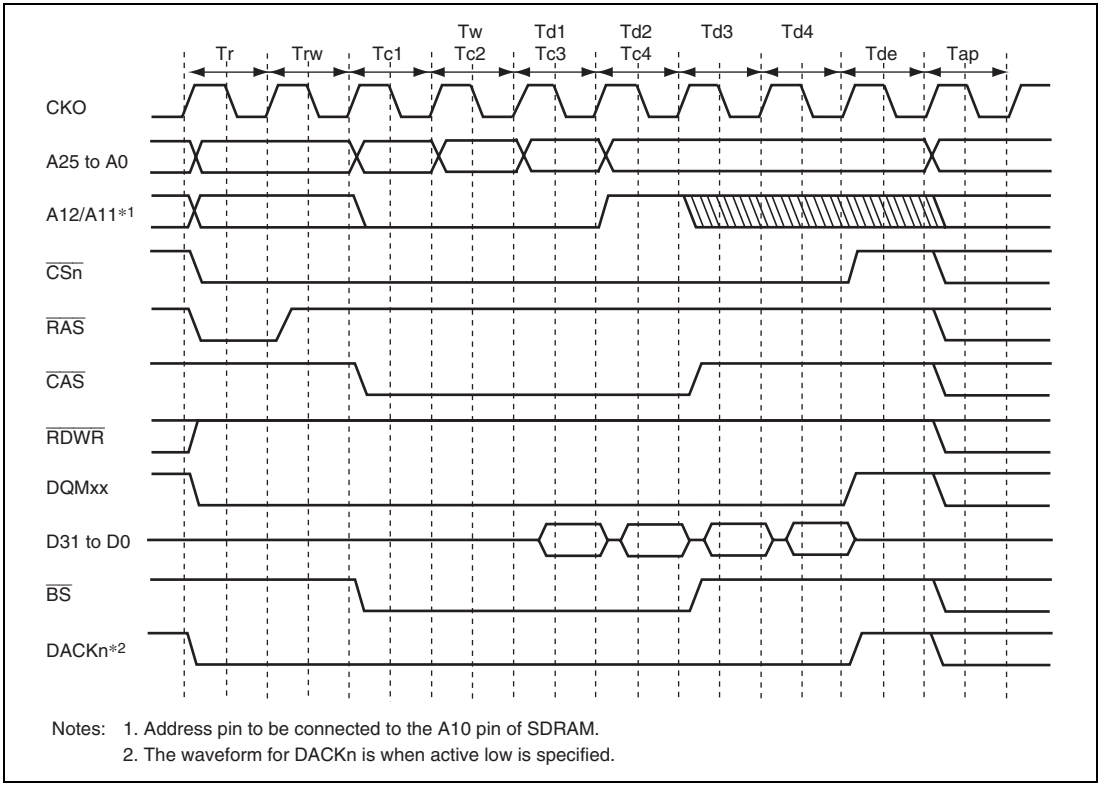
Figures 11.14 and 11.15 show a timing chart in burst read. In burst read, an ACTV command is output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, the READA command is issued in the Tc4 cycle, and the read data is received at the rising edge of the external clock (CKO) in the Td1 to Td4 cycles. The Tap cycle is used to wait for the completion of an auto-precharge induced by the READ command in the SDRAM. In the Tap cycle, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Tap cycles is specified by the TRP1 and TRP0 bits in CS3WCR.

In this LSI, wait cycles can be inserted by specifying each bit in CSnWCR to connect the SDRAM in variable frequencies. Figure 11.15 shows an example in which wait cycles are inserted. The number of cycles from the Tr cycle where the ACTV command is output to the Tc1 cycle where the READ command is output can be specified using the TRCD1 and TRCD0 bits in CS3WCR. If the TRCD1 and TRCD0 bits specify two cycles or more, a Trw cycle where the NOT command is issued is inserted between the Tr cycle and Tc1 cycle. The number of cycles from the Tc1 cycle where the READ command is output to the Td1 cycle where the read data is latched can be specified for the CS2 and CS3 spaces independently, using the A2CL1 and A2CL0 bits in CS2WCR or the A3CL1 and A3CL0 bits in CS3WCR and TRCD0 bit in CS3WCR. The number of cycles from Tc1 to Td1 corresponds to the synchronous DRAM CAS latency. The CAS latency for the synchronous DRAM is normally defined as up to three cycles. However, the CAS latency in this LSI can be specified as 1 to 4 cycles. This CAS latency can be achieved by connecting a latch circuit between this LSI and the synchronous DRAM.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 11.14 Burst Read Basic Timing (Auto-Precharge)



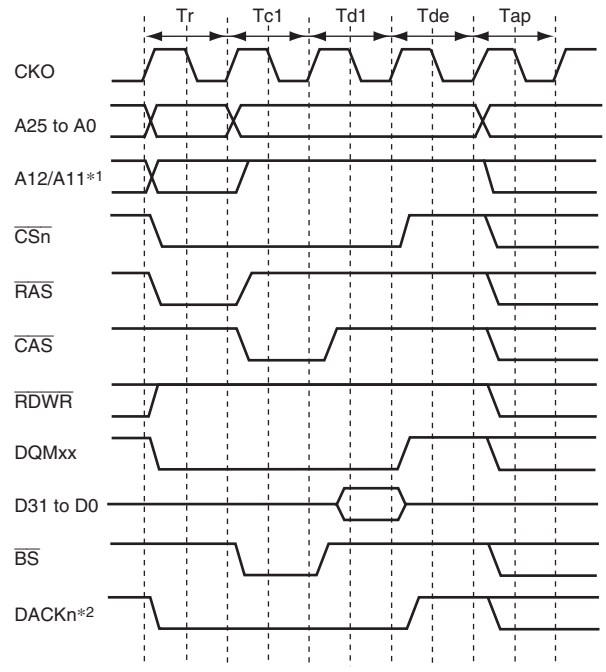
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 11.15 Burst Read Wait Specification Timing (Auto-Precharge)

(4) Single Read

A read access ends in one cycle when data exists in non-cacheable region and the data bus width is larger than or equal to access size. As the burst length is set to 1 in SDRAM burst read/single write mode, only the required data is output. Consequently, no unnecessary bus cycles are generated even when a cache-through area is accessed.

Figure 11.16 shows the single read basic timing.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 11.16 Basic Timing for Single Read (Auto-Precharge)

(5) Burst Write

A burst write occurs in the following cases in this LSI.

1. Access size in writing is larger than data bus width.
2. Copyback of the cache
3. 16-byte transfer in DMAC (access to non-cacheable region)

This LSI always accesses SDRAM with burst length 1. For example, write access of burst length 1 is performed continuously 4 times to write 16-byte continuous data to the SDRAM that is connected to a 32-bit data bus. The relationship between the access size and the number of bursts is shown in table 11.22.

Figure 11.17 shows a timing chart for burst writes. In burst write, an ACTV command is output in the Tr cycle, the WRIT command is issued in the Tc1, Tc2, and Tc3 cycles, and the WRITA command is issued to execute an auto-precharge in the Tc4 cycle. In the write cycle, the write data is output simultaneously with the write command. After the write command with the auto-precharge is output, the Trw1 cycle that waits for the auto-precharge initiation is followed by the Tap cycle that waits for completion of the auto-precharge induced by the WRITA command in the SDRAM. In the Tap cycle, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Trw1 cycles is specified by the TRWL1 and TRWL0 bits in CS3WCR. The number of Tap cycles is specified by the TRP1 and TRP0 bits in CS3WCR.

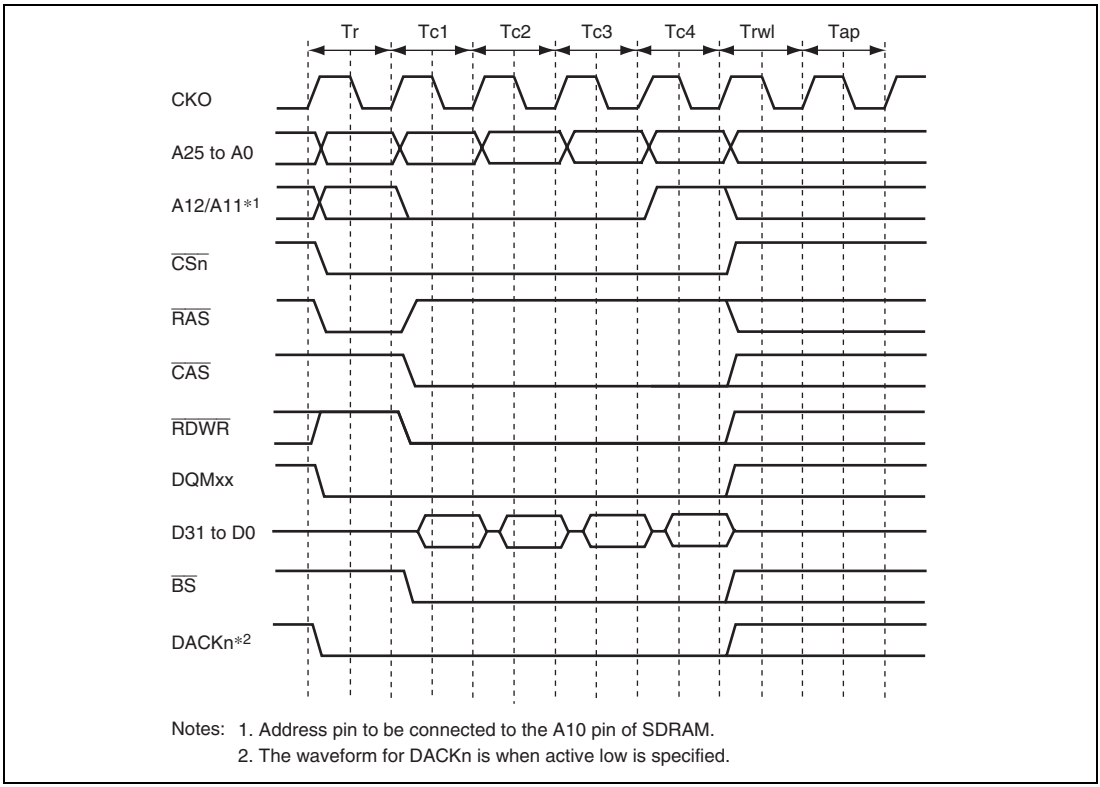
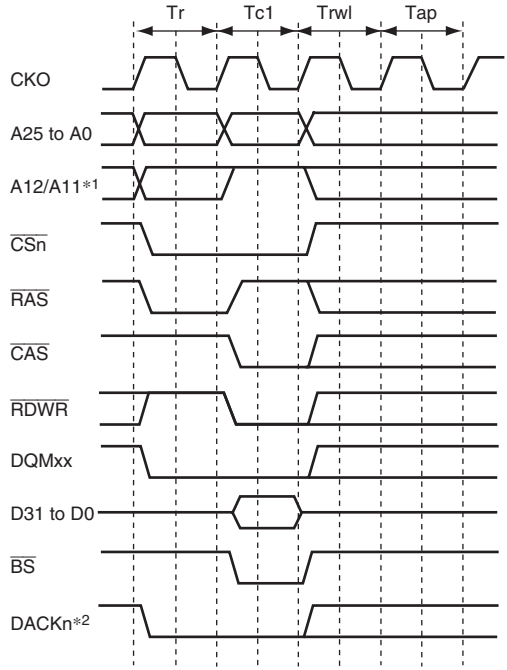


Figure 11.17 Basic Timing for Burst Write (Auto-Precharge)

(6) Single Write

A write access ends in one cycle when data is written in non-cacheable region and the data bus width is larger than or equal to access size.

Figure 11.18 shows the single write basic timing.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 11.18 Basic Timing for Single Write (Auto-Precharge)

(7) Bank Active

The SDRAM bank function is used to support high-speed accesses to the same row address. When the BACTV bit in SDCR is 1, accesses are performed using commands without auto-precharge (READ or WRIT). This function is called bank-active function. This function is valid only for either the upper or lower bits of area 3. When area 3 is set to bank-active mode, area 2 should be set to normal space or byte-selection SRAM. When areas 2 and 3 are both set to SDRAM, auto-precharge mode must be set.

When a bank-active function is used, precharging is not performed when the access ends. When accessing the same row address in the same bank, it is possible to issue the READ or WRIT command immediately, without issuing an ACTV command. As SDRAM is internally divided into several banks, it is possible to activate one row address in each bank. If the next access is to a different row address, a PRE command is first issued to precharge the relevant bank, then when precharging is completed, the access is performed by issuing an ACTV command followed by a READ or WRIT command. If this is followed by an access to a different row address, the access time will be longer because of the precharging performed after the access request is issued. The number of cycles between issuance of the PRE command and the ACTV command is determined by the TRP[1:0] bits in CSnWCR.

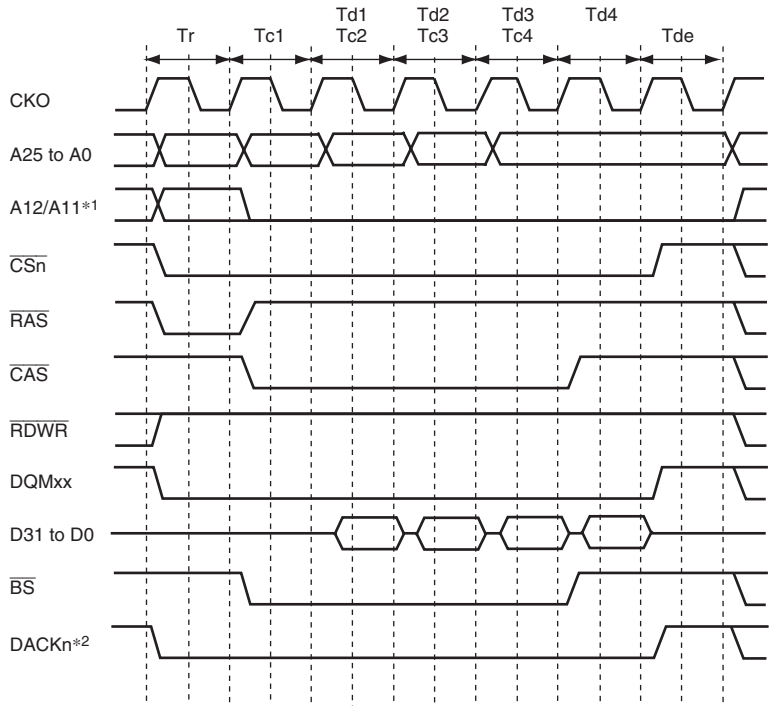
In a write, when an auto-precharge is performed, a command cannot be issued to the same bank for a period of Trwl + Tap cycles after issuance of the WRITA command. When bank active mode is used, READ or WRIT commands can be issued successively if the row address is the same. The number of cycles can thus be reduced by Trwl + Tap cycles for each write.

There is a limit on tRAS, the time for placing each bank in the active state. If there is no guarantee that there will not be a cache hit and another row address will be accessed within the period in which this value is maintained by program execution, it is necessary to set auto-refresh and set the refresh cycle to no more than the maximum value of tRAS.

A burst read cycle without auto-precharge is shown in figure 11.19, a burst read cycle for the same row address in figure 11.20, and a burst read cycle for different row addresses in figure 11.21. Similarly, a single write cycle without auto-precharge is shown in figure 11.22, a single write cycle for the same row address in figure 11.23, and a single write cycle for different row addresses in figure 11.24.

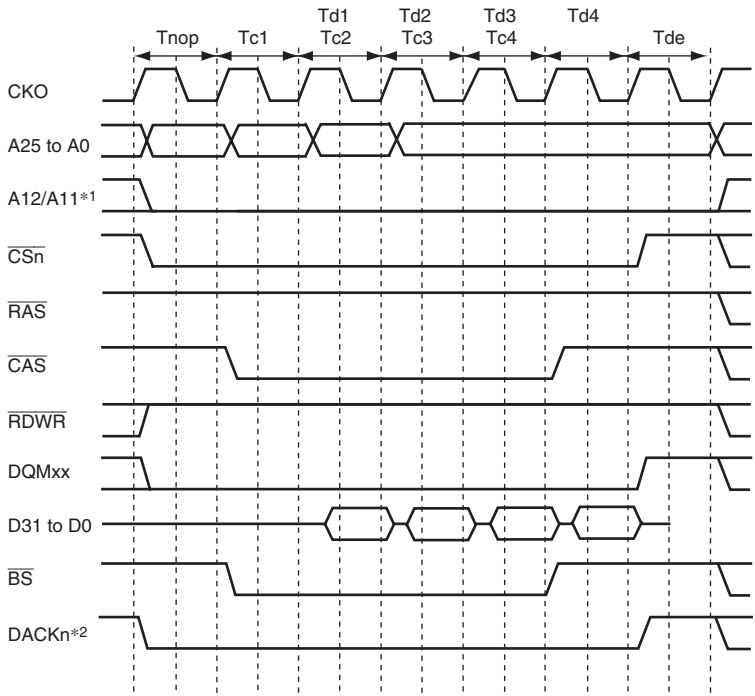
In figure 11.20, a Tnop cycle in which no operation is performed is inserted before the Tc cycle that issues the READ command. The Tnop cycle is inserted to acquire two cycles of CAS latency for the DQMxx signal that specifies the read byte in the data read from the SDRAM. If the CAS latency is specified as two cycles or more, the Tnop cycle is not inserted because the two cycles of latency can be acquired even if the DQMxx signal is asserted after the Tc cycle.

When bank active mode is set, if only accesses to the respective banks in the area 3 space are considered, as long as accesses to the same row address continue, the operation starts with the cycle in figure 11.19 or 9.22, followed by repetition of the cycle in figure 11.20 or 9.23. An access to a different area during this time has no effect. If there is an access to a different row address in the bank active state, after this is detected the bus cycle in figure 11.21 or 9.24 is executed instead of that in figure 11.20 or 9.23. In bank active mode, too, all banks become inactive after a refresh cycle or after the bus is released as the result of bus arbitration.



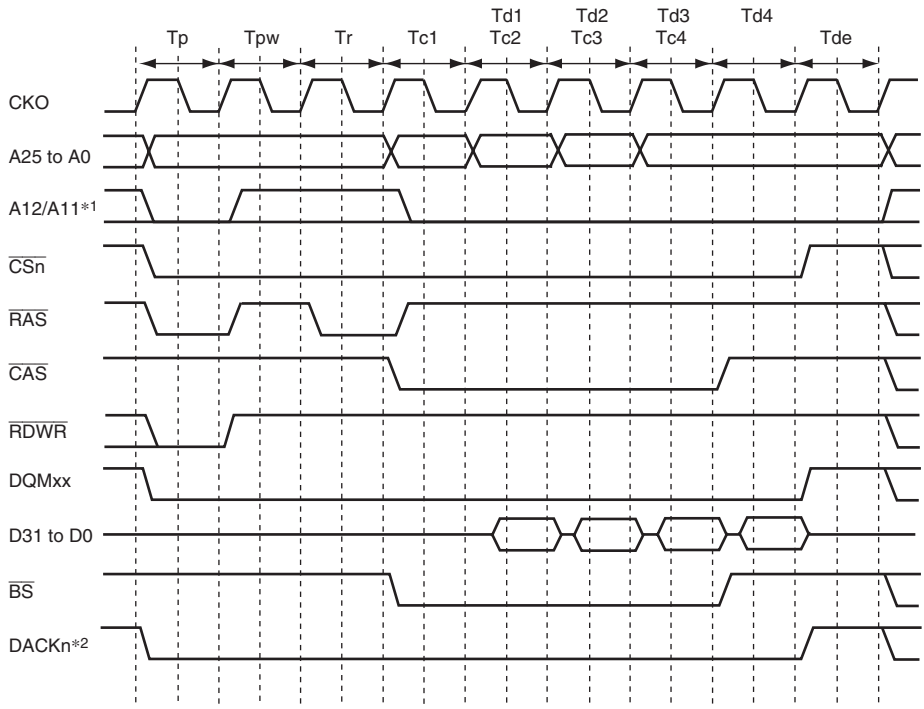
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 11.19 Burst Read Timing (No Auto-Precharge)



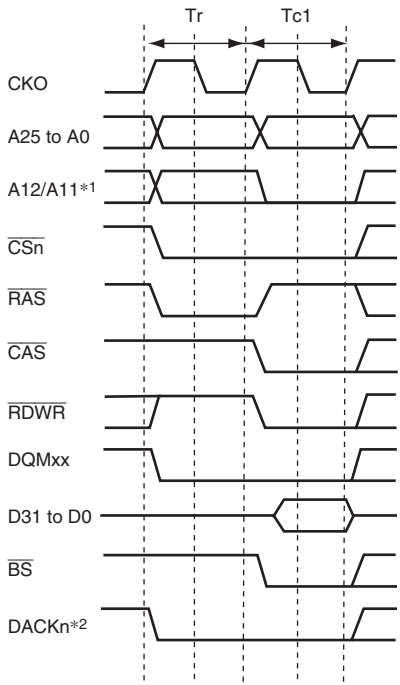
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 11.20 Burst Read Timing (Bank Active, Same Row Address)



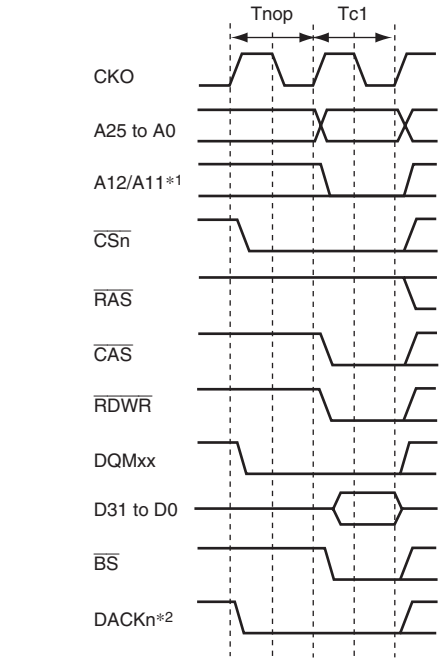
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 11.21 Burst Read Timing (Bank Active, Different Row Addresses)



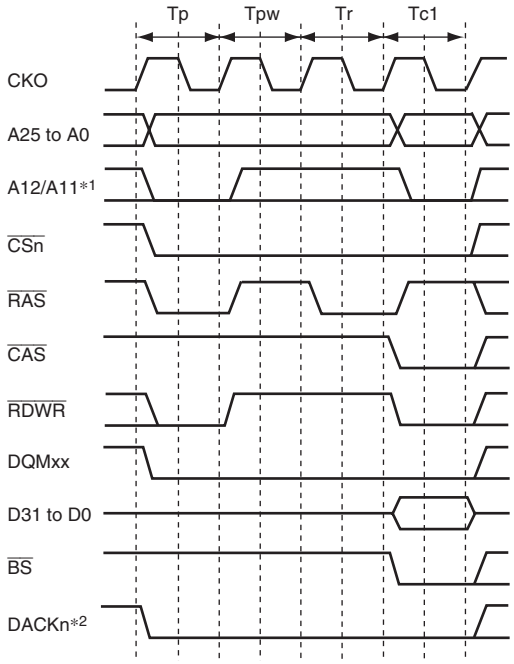
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 11.22 Single Write Timing (No Auto-Precharge)



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 11.23 Single Write Timing (Bank Active, Same Row Address)



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 11.24 Single Write Timing (Bank Active, Different Row Addresses)

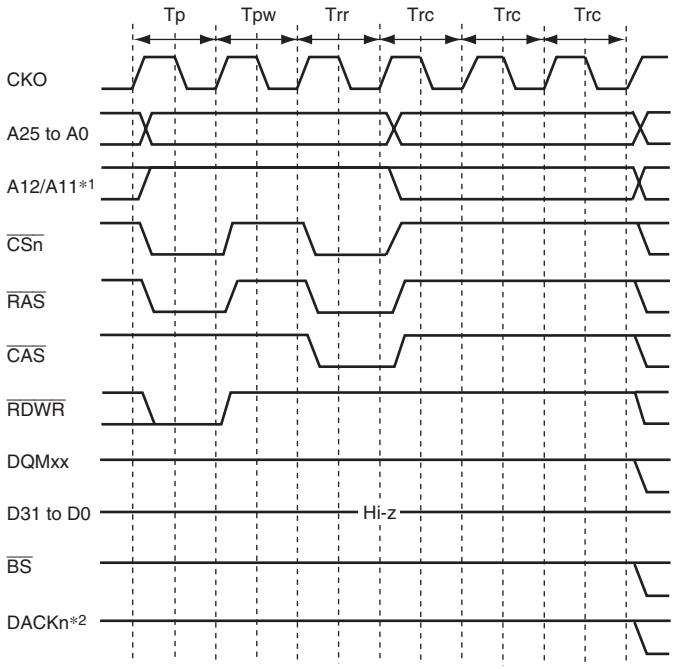
(8) Refreshing

This LSI has a function for controlling SDRAM refreshing. Auto-refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in SDCR. A continuous refreshing can be performed by setting the RRC[2:0] bits in RTCSR. If SDRAM is not accessed for a long period, self-refresh mode, in which the power consumption for data retention is low, can be activated by setting both the RMODE bit and the RFSH bit to 1.

(a) Auto-refreshing

Refreshing is performed at intervals determined by the input clock selected by bits CKS[2:0] in RTCSR, and the value set by in RTCOR. The value of bits CKS[2:0] in RTCOR should be set so as to satisfy the refresh interval stipulation for the SDRAM used. First make the settings for RTCOR, RTCNT, and the RMODE and RFSH bits in SDCR, then make the CKS[2:0] and RRC[2:0] settings. When the clock is selected by bits CKS[2:0], RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared with the RTCOR value, and if the two values are the same, a refresh request is generated and an auto-refresh is performed for the number of times specified by the RRC[2:0]. At the same time, RTCNT is cleared to 0 and the count-up is restarted. Figure 11.25 shows the auto-refresh cycle timing.

After starting, the auto refreshing, PALL command is issued in the Tp cycle to make all the banks to precharged state from active state when some bank is being precharged. Then REF command is issued in the Trr cycle after inserting idle cycles of which number is specified by the TRP[1:0]bits in CSnWCR. A new command is not issued for the duration of the number of cycles specified by the TRC[1:0] bits in CSnWCR after the Trr cycle. The TRC[1:0] bits must be set so as to satisfy the SDRAM refreshing cycle time stipulation (tRC). A NOP cycle is inserted between the Tp cycle and Trr cycle when the setting value of the TRP[1:0] bits in CSnWCR is longer than or equal to 2 cycles.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 11.25 Auto-Refresh Timing

(b) Self-refreshing

Self-refresh mode in which the refresh timing and refresh addresses are generated within the SDRAM. Self-refreshing is activated by setting both the RMODE bit and the RFSH bit in SDCR to 1. After starting the self-refreshing, PALL command is issued in Tp cycle after the completion of the pre-charging bank. A SELF command is then issued after inserting idle cycles of which number is specified by the TRP[1:0] bits in CSnWSR. SDRAM cannot be accessed while in the self-refresh state. Self-refresh mode is cleared by clearing the RMODE bit to 0. After self-refresh mode has been cleared, command issuance is disabled for the number of cycles specified by the TRC[1:0] bits in CSnWCR.

Self-refresh timing is shown in figure 11.26. Settings must be made so that self-refresh clearing and data retention are performed correctly, and auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the state in which auto-refreshing is set, or when exiting standby mode other than through a power-on reset, auto-refreshing is restarted if the RFSH bit is set to 1 and the RMODE bit is cleared to 0 when self-refresh mode is cleared. If the transition from clearing of self-refresh mode to the start of auto-refreshing takes time, this time should be taken into consideration when setting the initial value of RTCNT. Making the RTCNT value 1 less than the RTCOR value will enable refreshing to be started immediately.

After self-refreshing has been set, the self-refresh state continues even if the chip standby state is entered using the LSI standby function, and is maintained even after recovery from standby mode by an interrupt.

The self-refresh state is not cleared by a manual reset.

In case of a power-on reset, the bus state controller's registers are initialized, and therefore the self-refresh state is cleared.

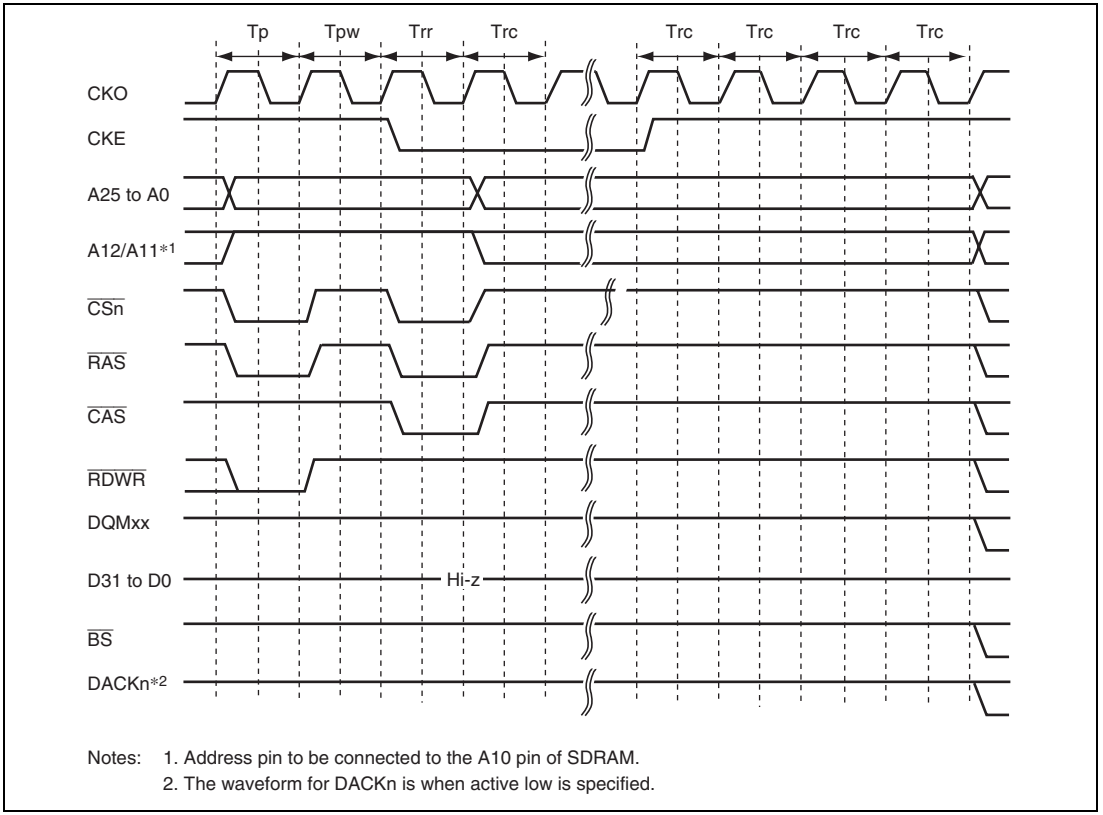


Figure 11.26 Self-Refresh Timing

(9) Relationship between Refresh Requests and Bus Cycles

If a refresh request occurs during bus cycle execution, the refresh cycle must wait for the bus cycle to be completed. If a refresh request occurs while the bus is released by the bus arbitration function, the refresh will not be executed until the bus mastership is acquired. This LSI supports requests by the $\overline{\text{REFOUT}}$ pin for the bus mastership while waiting for the refresh request. The $\overline{\text{REFOUT}}$ pin is asserted low until the bus mastership is acquired.

If a new refresh request occurs while waiting for the previous refresh request, the previous refresh request is deleted. To refresh correctly, a bus cycle longer than the refresh interval or the bus mastership occupation must be prevented from occurring. If a bus mastership is requested during self-refresh, the bus will not be released until the self-refresh is completed.

(10) Power-Down Mode

If the PDOWN bit in SDCR is set to 1, the SDRAM is placed in the power-down mode by bringing the CKE signal to the low level in the non-access cycle. This power-down mode can effectively lower the power consumption in the non-access cycle. However, please note that if an access occurs in power-down mode, a cycle of overhead occurs because a cycle that asserts the CKE in order to cancel power-down mode is inserted.

Figure 11.27 shows the access timing in power-down mode.

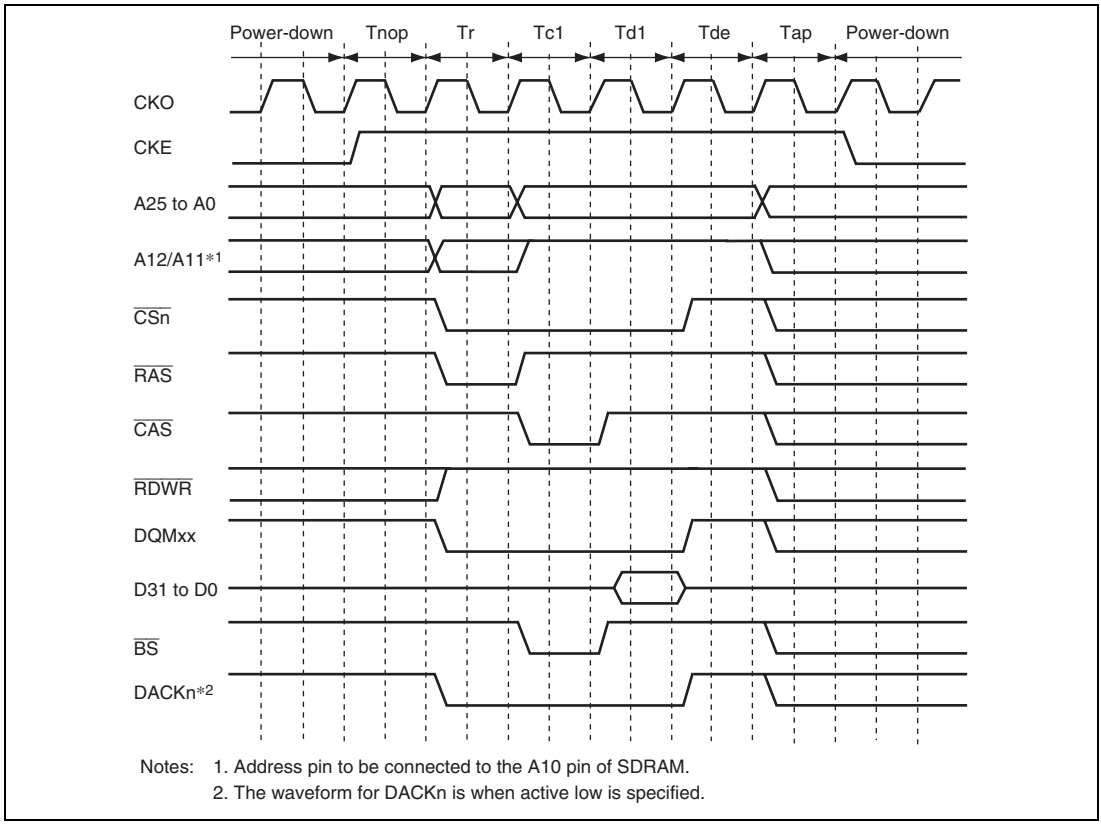


Figure 11.27 Access Timing in Power-Down Mode

(11) Power-On Sequence

In order to use SDRAM, mode setting must first be performed after powering on. To perform SDRAM initialization correctly, the bus state controller registers must first be set, followed by a write to the SDRAM mode register. In SDRAM mode register setting, the address signal value at that time is latched by a combination of the $\overline{\text{CSn}}$, $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and $\overline{\text{RDWR}}$ signals. If the value to be set is X, the bus state controller provides for value X to be written to the SDRAM mode register by performing a write to address H'FEC14000 + X for area 2 SDRAM, and to address H'FEC15000 + X for area 3 SDRAM. In this operation the data is ignored, but the mode write is performed as a byte-size access. To set burst read/single write, CAS latency 2 and 3, wrap type = sequential, and burst length 1 supported by the LSI, arbitrary data is written in a byte-size access to the addresses shown in table 11.23. In this time 0 is output at the external address pins of A12 or later.

Table 11.23 Access Address in SDRAM Mode Register Write

- Setting for Area 2 (SDMR2) RDWR

  Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FEC14440	H'0000440
	3	H'FEC14460	H'0000460
32 bits	2	H'FEC14880	H'0000880
	3	H'FEC148C0	H'00008C0

  Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FEC14040	H'0000040
	3	H'FEC14060	H'0000060
32 bits	2	H'FEC14080	H'0000080
	3	H'FEC140C0	H'00000C0

- Setting for Area 3 (SDMR3)

  Burst read/single write (burst length 1):

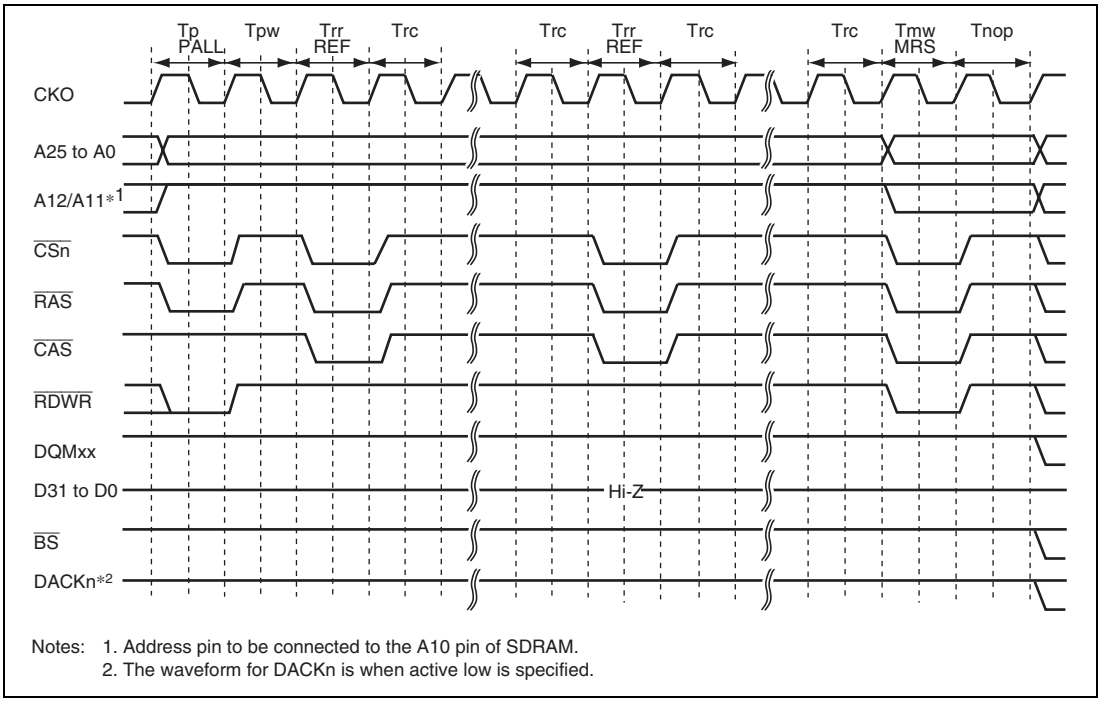
Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FEC15440	H'0000440
	3	H'FEC15460	H'0000460
32 bits	2	H'FEC15880	H'0000880
	3	H'FEC158C0	H'00008C0

  Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FEC15040	H'0000040
	3	H'FEC15060	H'0000060
32 bits	2	H'FEC15080	H'0000080
	3	H'FEC150C0	H'00000C0

Mode register setting timing is shown in figure 11.28. A PALL command (all bank precharge command) is firstly issued. A REF command (auto-refresh command) is then issued 8 times. An MRS command (mode register write command) is finally issued. Idle cycles, of which number is specified by the TRP[1:0] bits in CSnWCR, are inserted between the PALL and the first REF. Idle cycles, of which number is specified by the TRC[1:0]bits in CSnWCR, are inserted between REF and REF, and between the 8th REF and MRS. Idle cycles, of which number is one or more, are inserted between the MRS and a command to be issued next.

It is necessary to keep idle time of certain cycles for SDRAM before issuing PALL command after power-on. Refer the manual of the SDRAM for the idle time to be needed. When the pulse width of the reset signal is longer then the idle time, mode register setting can be started immediately after the reset, but care should be taken when the pulse width of the reset signal is shorter than the idle time.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 11.28 Write Timing for SDRAM Mode Register (Based on JEDEC)

11.5.6 Burst ROM (Clock Asynchronous) Interface

The burst ROM (clock asynchronous) interface is used to access a memory with a high-speed read function using a method of address switching called the burst mode or page mode. In a burst ROM (clock asynchronous) interface, basically the same access as the normal space is performed, but the 2nd and subsequent accesses are performed only by changing the address, without negating the $\overline{RD}$ signal at the end of the 1st cycle. In the 2nd and subsequent accesses, addresses are changed at the falling edge of the CKO.

For the 1st access cycle, the number of wait cycles specified by the W[3:0] bits in CSnWCR is inserted. For the 2nd and subsequent access cycles, the number of wait cycles specified by the BW[1:0] bits in CSnWCR is inserted.

In the access to the burst ROM (clock asynchronous), the $\overline{BS}$ signal is asserted only to the first access cycle. An external wait input is valid only to the first access cycle.

In the single access or write access that do not perform the burst operation in the burst ROM (clock asynchronous) interface, access timing is same as a normal space.

Table 11.24 lists a relationship between bus width, access size, and the number of bursts. Figure 11.29 shows a timing chart.

Note: When using the CS0 space as burst ROM, set CS0BCR and CS0WCR by using a program in a space other than CS0 (on-chip RAM, for example) before accessing the burst ROM.

Table 11.24 Relationship between Bus Width, Access Size, and Number of Bursts

Bus Width	Access Size	Number of Bursts
8 bits	8 bits	1
	16 bits	2
	32 bits	4
	8 bytes	8
	16 bytes	16
	32 bytes	16 x 2
16 bits	8 bits	1
	16 bits	1
	32 bits	2
	8 bytes	4
	16 bytes	8
	32 bytes	8 x 2
32 bits	8 bits	1
	16 bits	1
	32 bits	1
	8 bytes	2
	16 bytes	4
	32 bytes	4 x 2

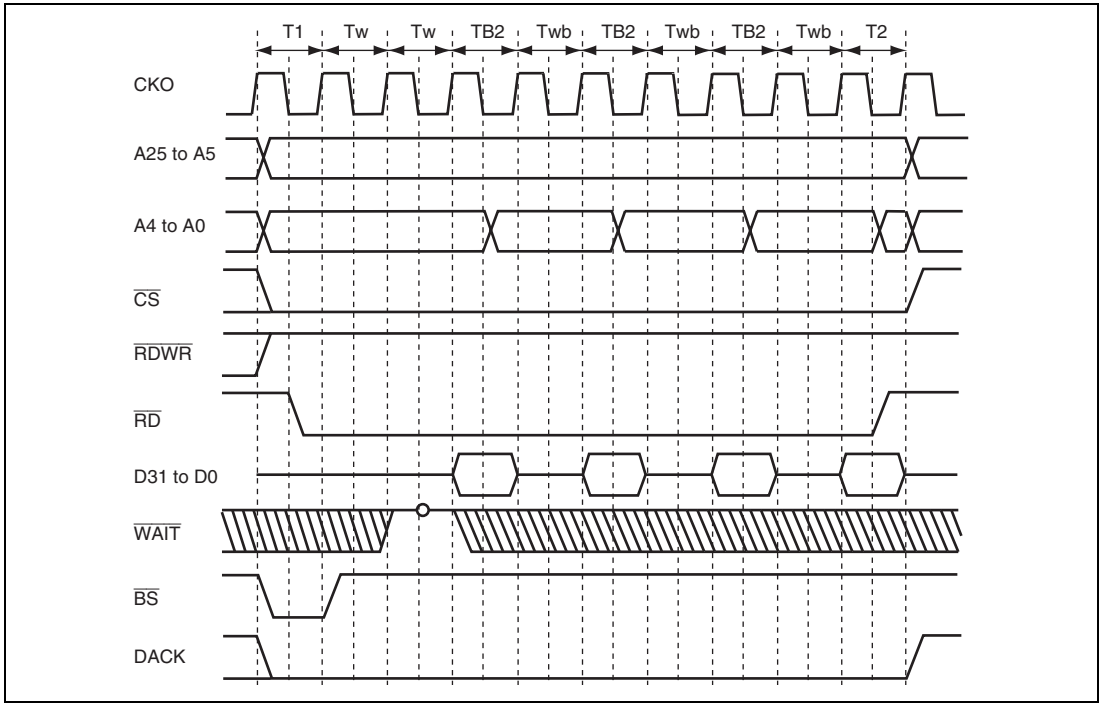


Figure 11.29 Burst ROM (Clock Asynchronous) Access (Bus Width = 32 Bits, 16-byte Transfer (Number of Bursts = 4), Access Wait for First Time = 2, Access Wait for 2nd Time and after = 1)

11.5.7 Byte-Selection SRAM Interface

The byte-selection SRAM interface is for access to an SRAM which has a byte-selection pin ($\overline{\text{WEn}}$). This interface has 16-bit data pins and accesses SRAMs having upper and lower byte selection pins, such as UB and LB.

When the BAS bit in CSnWCR is cleared to 0 (initial value), the write access timing of the byte-selection SRAM interface is the same as that for the normal space interface. While in read access of a byte-selection SRAM interface, the byte-selection signal is output from the $\overline{\text{WEn}}$ pin, which is different from that for the normal space interface. The basic access timing is shown in figure 11.30. In write access, data is written to the memory according to the timing of the byte-selection pin ($\overline{\text{WEn}}$). For details, refer to the data sheet for the corresponding memory.

If the BAS bit in CSnWCR is set to 1, the $\overline{\text{WEn}}$ pin and $\overline{\text{RDWR}}$ pin timings change. Figure 11.31 shows the basic access timing. In write access, data is written to the memory according to the timing of the write enable pin ($\overline{\text{RDWR}}$). The data hold timing from $\overline{\text{RDWR}}$ negation to data write must be acquired by setting the HW[1:0] bits in CSnWCR. Figure 11.32 shows the access timing when a software wait is specified.

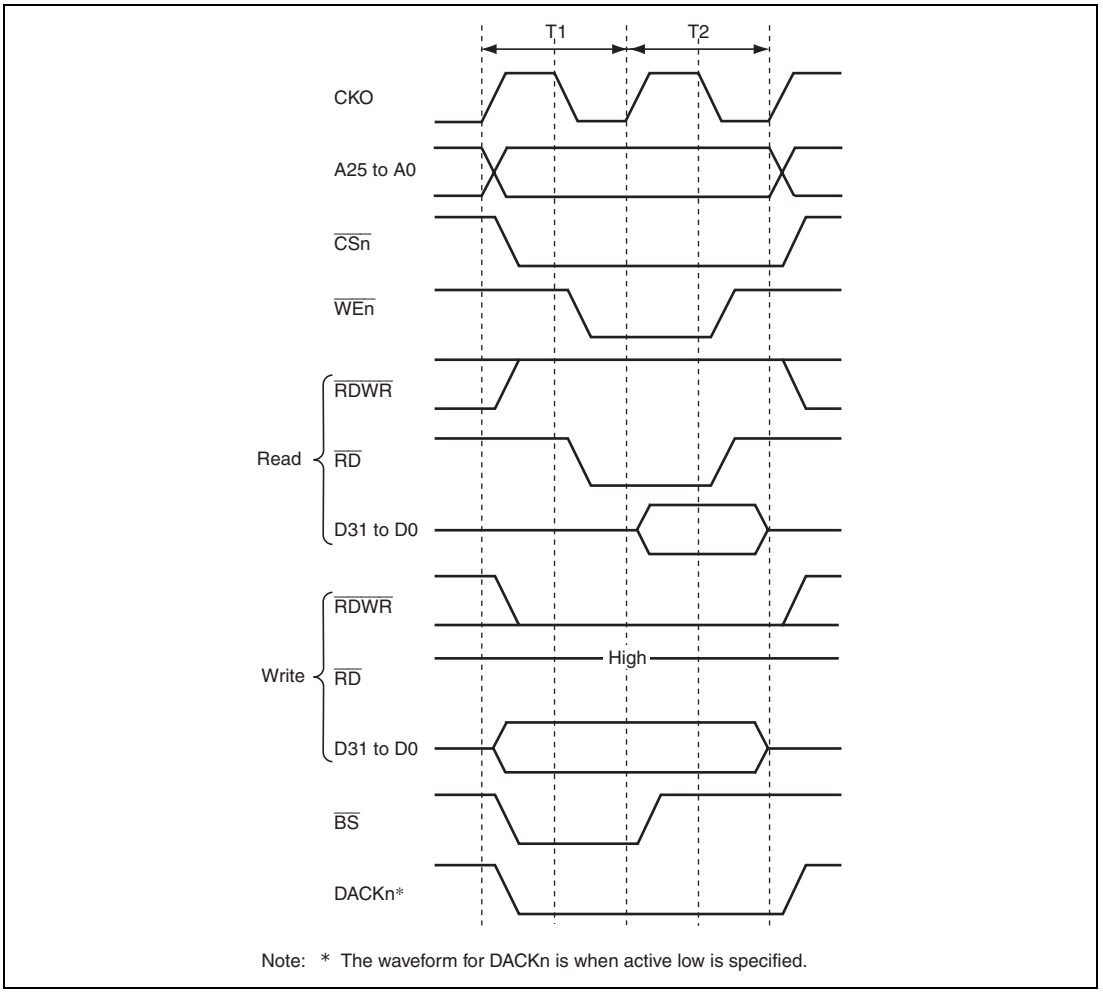
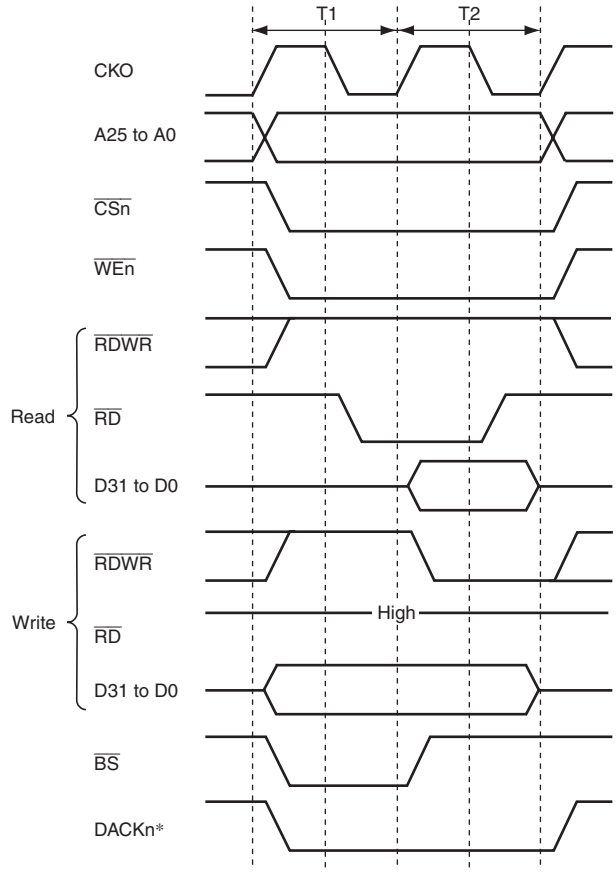
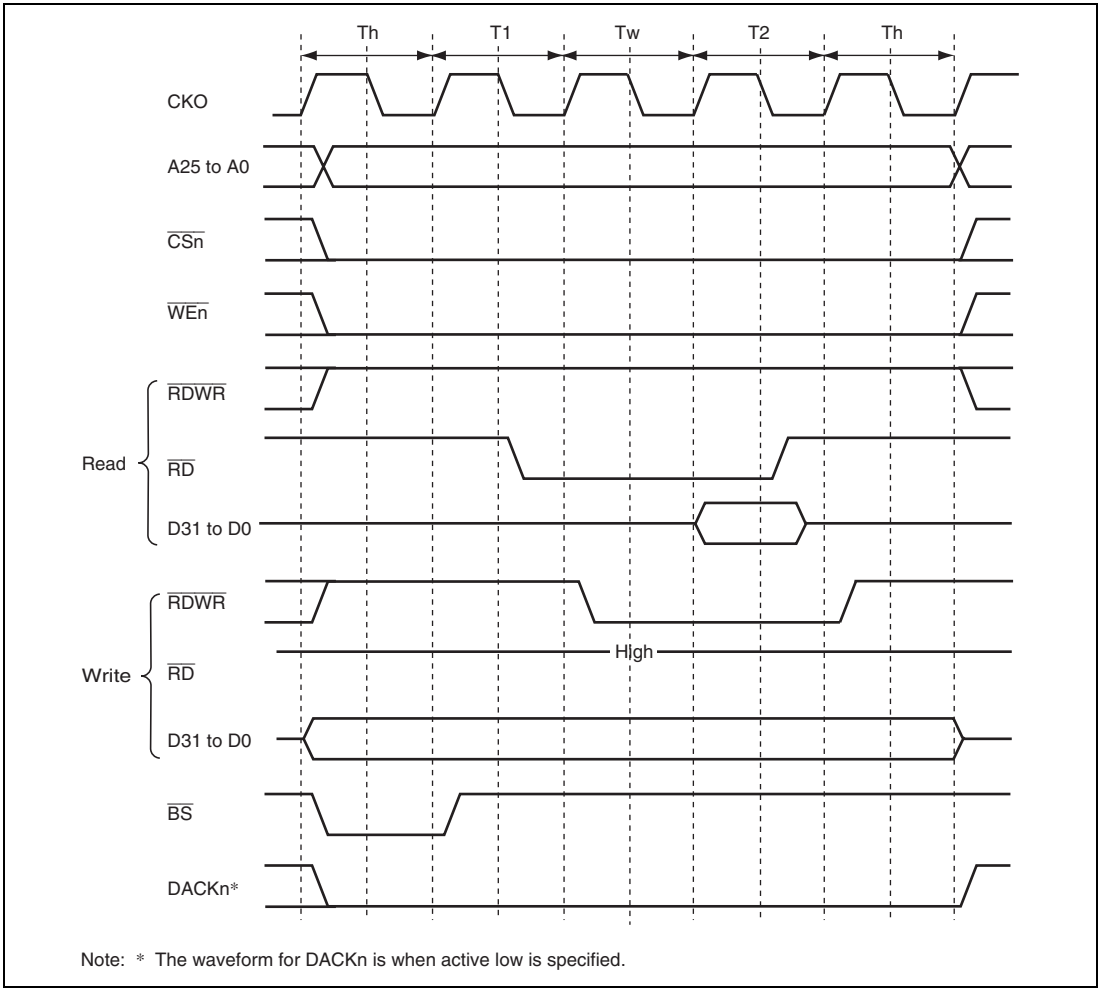


Figure 11.30 Basic Access Timing for Byte-Selection SRAM (BAS = 0)



Note: * The waveform for DACKn is when active low is specified.

Figure 11.31 Basic Access Timing for Byte-Selection SRAM (BAS = 1)



Note: * The waveform for DACKn is when active low is specified.

Figure 11.32 Wait Timing for Byte-Selection SRAM (BAS = 1) (Software Wait Only)

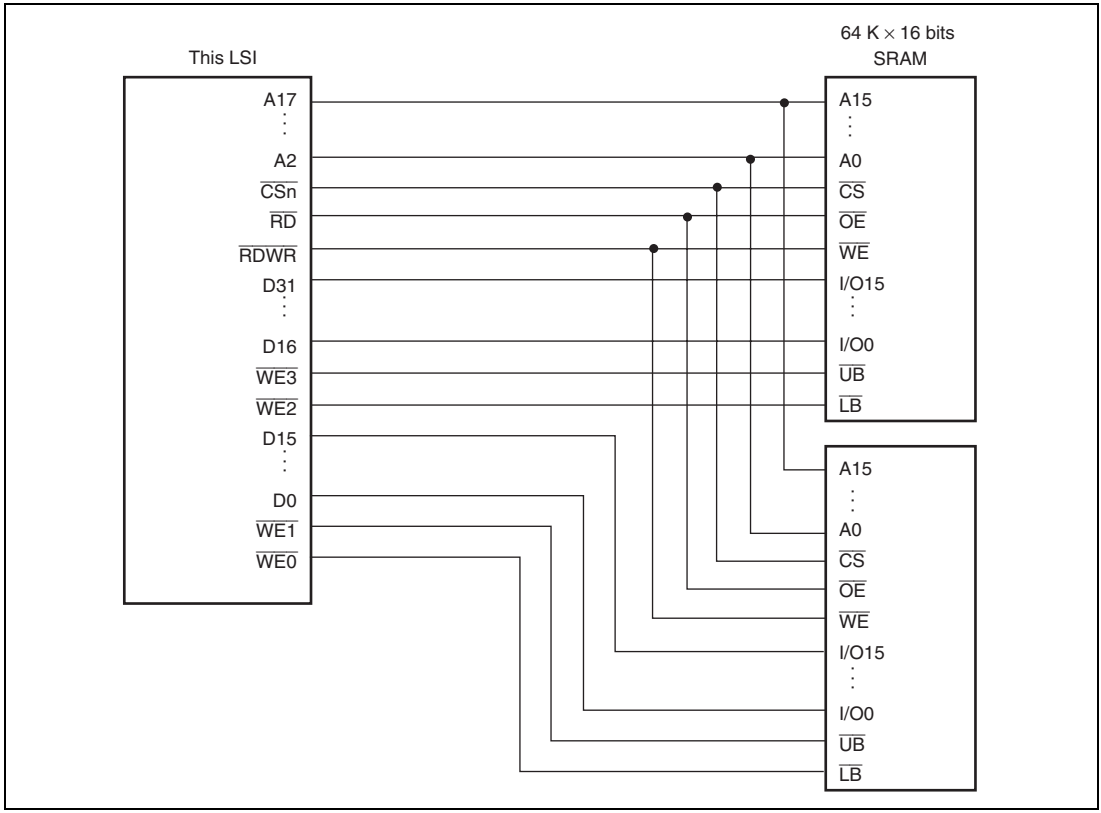


Figure 11.33 Example of Connection with 32-Bit Data-Width Byte-Selection SRAM

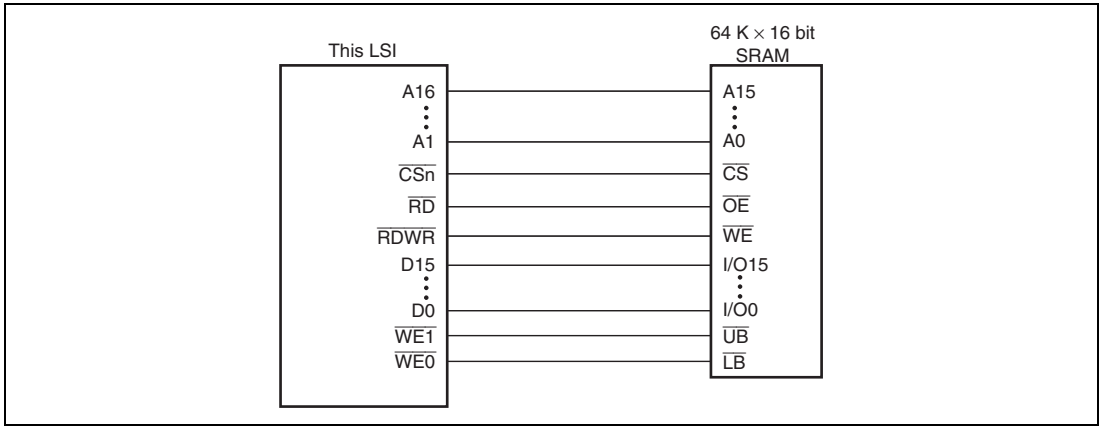


Figure 11.34 Example of Connection with 16-Bit Data-Width Byte-Selection SRAM

11.5.8 PCMCIA Interface

With this LSI, if address map (2) is selected using the MAP bit in CMNCR, the PCMCIA interface can be specified in areas 5 and 6. Areas 5 and 6 in the physical space can be used for the IC memory card and I/O card interface defined in the JEIDA specifications version 4.2 (PCMCIA2.1) by specifying the TYPE[3:0] bits of CSnBCR (n = 5B, 6B) to B'0101. In addition, the SA[1:0] bits of CSnWCR (n = 5B, 6B) assign the upper or lower 32 Mbytes of each area to an IC memory card or I/O card interface. For example, if the SA1 and SA0 bits of the CS5BWCR are set to 1 and cleared to 0, respectively, the upper 32 Mbytes and the lower 32 Mbytes of area 5B are used as an IC memory card interface and I/O card interface, respectively.

When the PCMCIA interface is used, the bus size must be specified as 8 bits or 16 bits using the BSZ[1:0] bits in CS5BBCR or CS6BBCR.

Figure 11.35 shows an example of a connection between this LSI and the PCMCIA card. To enable insertion and removal of the PCMCIA card during system power-on, a three-state buffer must be connected between the LSI and the PCMCIA card.

In the JEIDA and PCMCIA standards, operation in the big endian mode is not clearly defined. Consequently, an original definition is provided for the PCMCIA interface in big endian mode in this LSI.

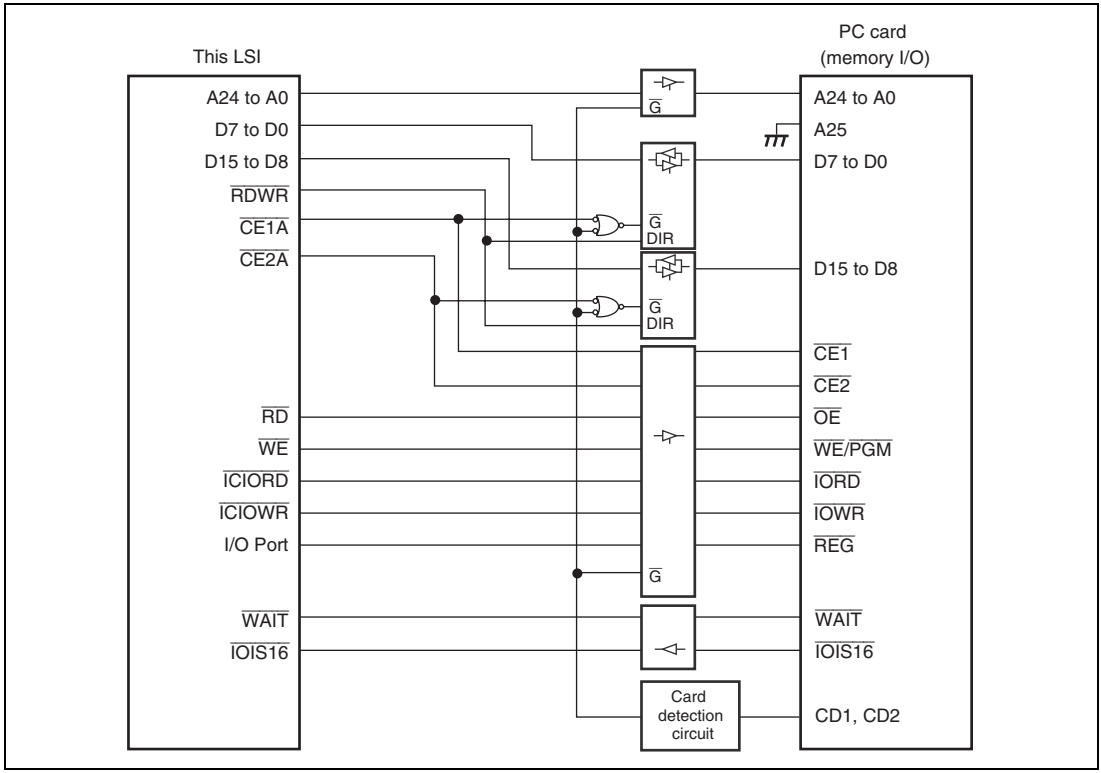


Figure 11.35 Example of PCMCIA Interface Connection

(1) Basic Timing for Memory Card Interface

Figure 11.36 shows the basic timing of the PCMCIA IC memory card interface. If areas 5 and 6 in the physical space are specified as the PCMCIA interface, accessing the common memory areas in areas 5 and 6 automatically accesses the IC memory card interface. If the external bus frequency (CKO) increases, the setup times and hold times for the address pins (A25 to A0) to RD and WE, card enable signals ($\overline{CE1A}$, $\overline{CE2A}$, $\overline{CE1B}$, $\overline{CE2B}$), and write data (D15 to D0) become insufficient. To prevent this error, the LSI can specify the setup times and hold times for areas 5 and 6 in the physical space independently, using CS5BWCR and CS6BWCR. In the PCMCIA interface, as in the normal space interface, a software wait or hardware wait can be inserted using the WAIT pin. Figure 11.37 shows the PCMCIA memory bus wait timing.

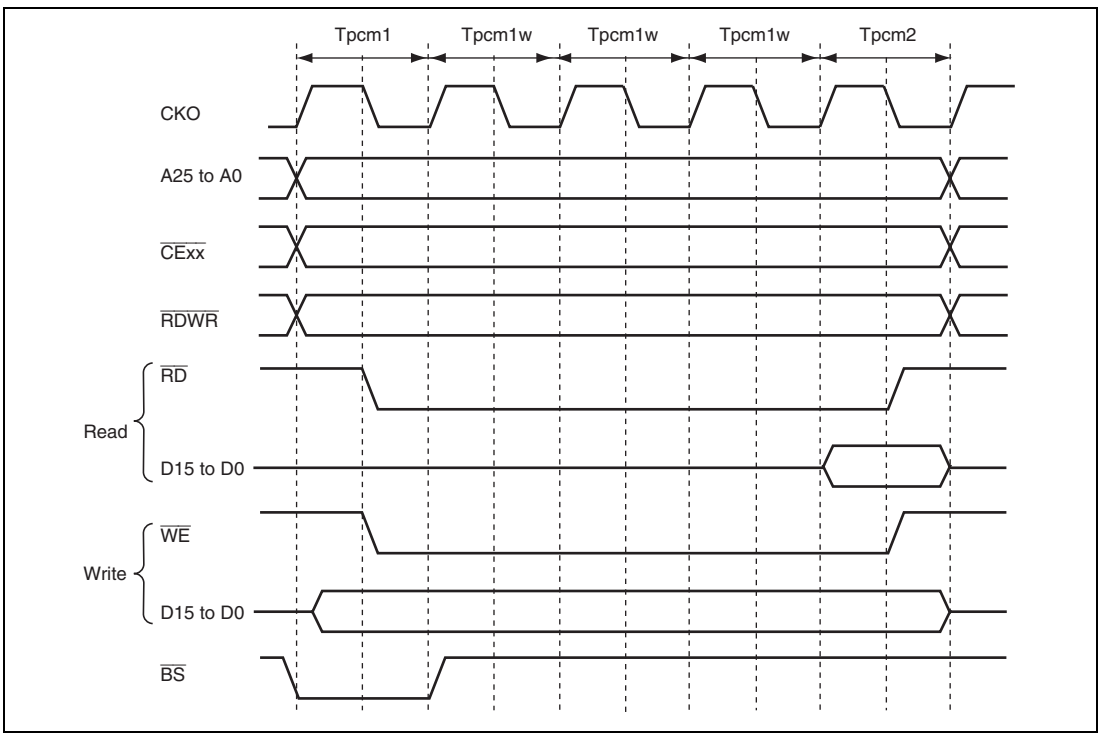


Figure 11.36 Basic Access Timing for PCMCIA Memory Card Interface

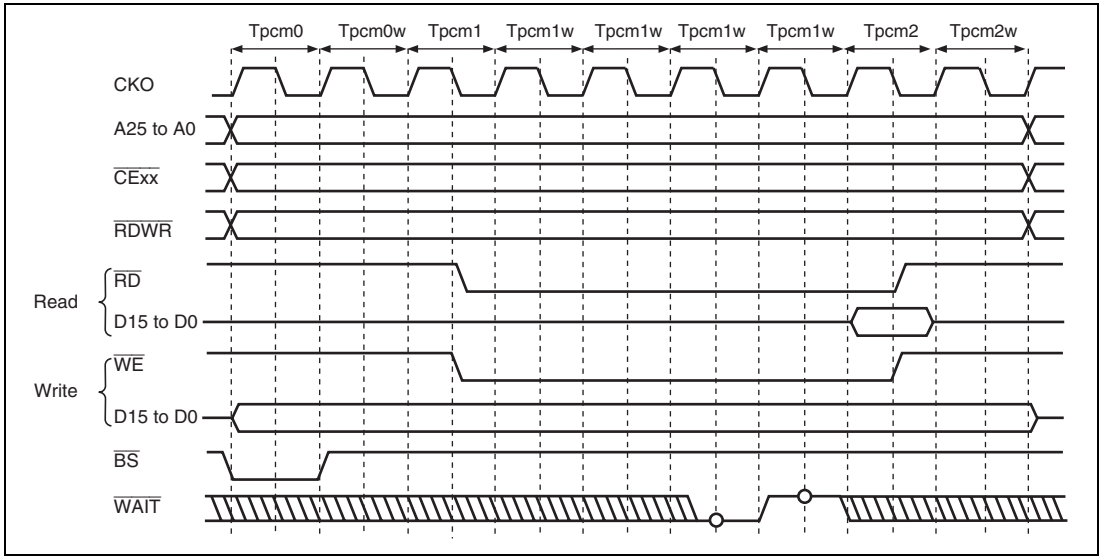


Figure 11.37 Wait Timing for PCMCIA Memory Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait = 1, Hardware Wait = 1)

If all 32 Mbytes of the memory space are used as an IC memory card interface, the $\overline{\text{REG}}$ signal that switches between the common memory and attribute memory can be generated by a port. If the memory space used for the IC memory card interface is 16 Mbytes or less, the A24 pin can be used as the $\overline{\text{REG}}$ signal by using the memory space as a 16-Mbyte common memory space and a 16-Mbyte attribute memory space.

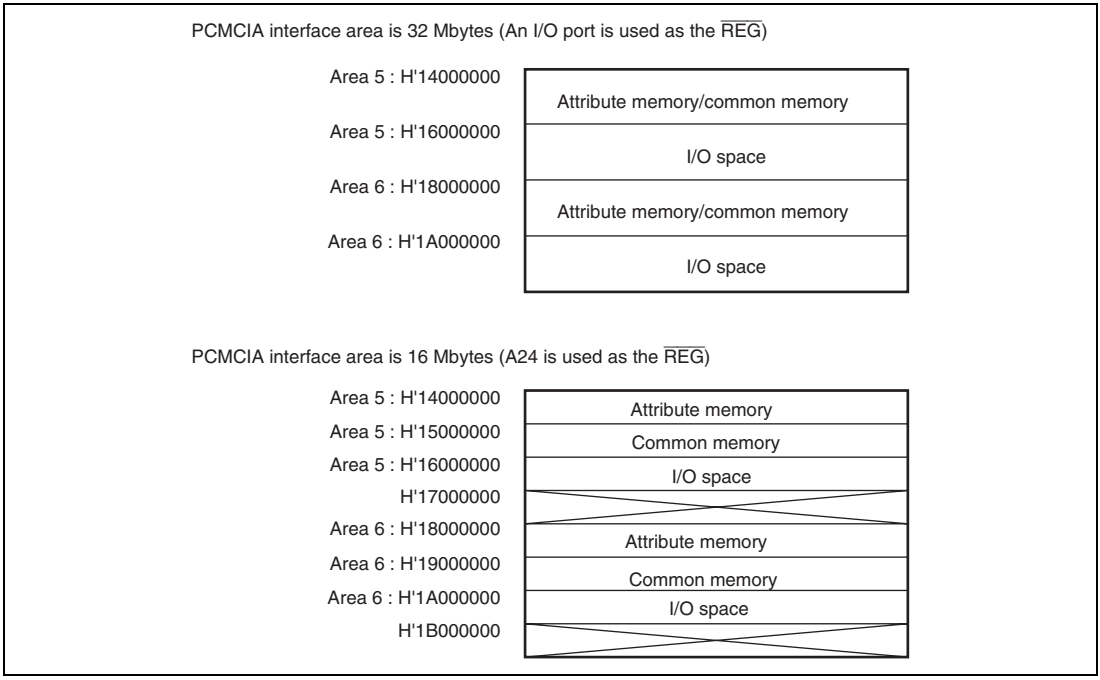


Figure 11.38 Example of PCMCIA Space Assignment (CS5BWCR.SA[1:0] = B'10, CS6BWCR.SA[1:0] = B'10)

(2) Basic Timing for I/O Card Interface

Figures 11.39 and 11.40 show the basic timings for the PCMCIA I/O card interface.

The I/O card and IC memory card interfaces can be switched using an address to be accessed. If area 5 of the physical space is specified as the PCMCIA, the I/O card interface can automatically be accessed by accessing the physical addresses from H'16000000 to H'17FFFFFF. If area 6 of the physical space is specified as the PCMCIA, the I/O card interface can automatically be accessed by accessing the physical addresses from H'1A000000 to H'1BFFFFFF.

Note that areas to be accessed as the PCMCIA I/O card must be non-cached if they are virtual space (space P2 or P3) areas, or a non-cached area specified by the MMU.

If the PCMCIA card is accessed as an I/O card in little endian mode, dynamic bus sizing for the I/O bus can be achieved using the $\overline{\text{IOIS16}}$ signal. If the $\overline{\text{IOIS16}}$ signal is brought high in a word-size I/O bus cycle while the bus width of area 6 is specified as 16 bits, the bus width is recognized as 8 bits and data is accessed twice in 8-bit units in the I/O bus cycle to be executed.

The $\overline{\text{IOIS16}}$ signal is sampled at the falling edge of CKO in the Tpci0, Tpci0w, and Tpci1 cycles when the TED[3:0] bits are specified as 1.5 cycles or more, and is reflected in the CE2 signal 1.5 cycles after the CKO sampling point. The TED[3:0] bits must be specified appropriately to satisfy the setup time from $\overline{\text{ICIORD}}$ and $\overline{\text{ICIOWR}}$ of the PC card to CEn.

Figure 11.41 shows the dynamic bus sizing basic timing.

Note that the $\overline{\text{IOIS16}}$ signal is not supported in big endian mode. In the big endian mode, the $\overline{\text{IOIS16}}$ signal must be fixed low.

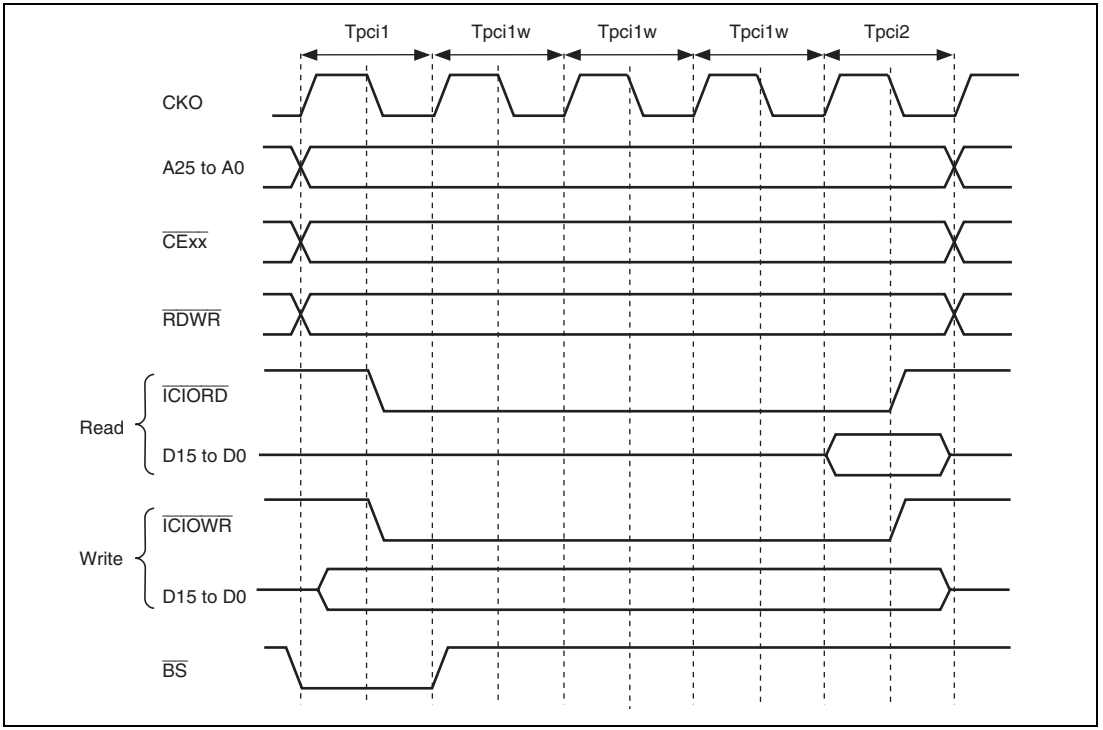


Figure 11.39 Basic Timing for PCMCIA I/O Card Interface

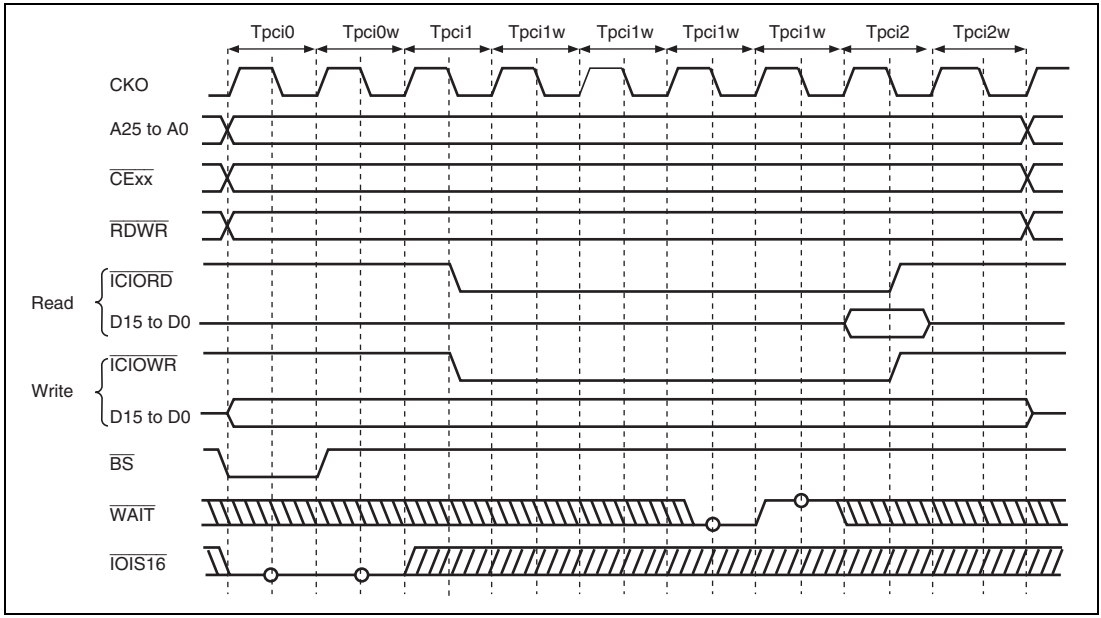


Figure 11.40 Wait Timing for PCMCIA I/O Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait = 1, Hardware Wait = 1)

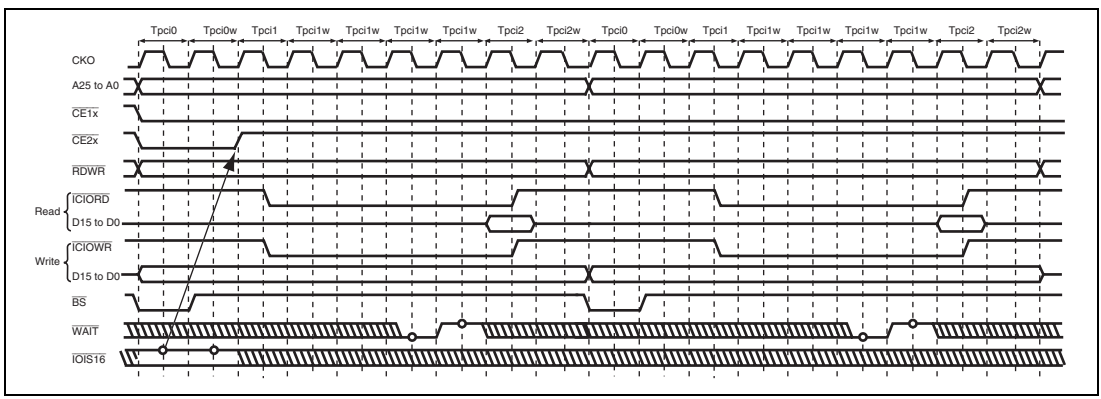


Figure 11.41 Timing for Dynamic Bus Sizing of PCMCIA I/O Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Waits = 3)

11.5.9 Wait between Access Cycles

As the operating frequency of LSIs becomes higher, the off-operation of the data buffer often collides with the next data access when the read operation from devices with slow access speed is completed. As a result of these collisions, the reliability of the device is low and malfunctions may occur. This LSI has a function that avoids data collisions by inserting wait cycles between continuous access cycles.

The number of wait cycles between access cycles can be set by bits IWW[2:0], IWRWD[2:0], IWRWS[2:0], IWRRD[2:0], and IWRRS[2:0] in CSnBCR. The conditions for setting the wait cycles between access cycles (idle cycles) are shown below.

1. Continuous accesses are write-read or write-write
2. Continuous accesses are read-write for different spaces
3. Continuous accesses are read-write for the same space
4. Continuous accesses are read-read for different spaces
5. Continuous accesses are read-read for the same space

11.5.10 Bus Arbitration

To prevent device malfunction while the bus mastership is transferred between master and slave, the LSI negates all of the bus control signals before bus release. When the bus mastership is received, all of the bus control signals are first negated and then driven appropriately. In this case, output buffer contention can be prevented because the master and slave drive the same signals with the same values. In addition, to prevent noise while the bus control signal is in the high impedance state, pull-up resistors must be connected to these control signals.

Bus mastership is transferred at the boundary of bus cycles. Namely, bus mastership is released immediately after receiving a bus request when a bus cycle is not being performed. The release of bus mastership is delayed until the bus cycle is complete when a bus cycle is in progress. Even when from outside the LSI it looks like a bus cycle is not being performed, a bus cycle may be performing internally, started by inserting wait cycles between access cycles. Therefore, it cannot be immediately determined whether or not bus mastership has been released by looking at the $\overline{\text{CS}}$n signal or other bus control signals. The states that do not allow bus mastership release are shown below.

1. 32-byte transfer because of a cache miss
2. During copyback operation for the cache
3. Between the read and write cycles of a TAS instruction
4. Multiple bus cycles generated when the data bus width is smaller than the access size (for example, between bus cycles when longword access is made to a memory with a data bus width of 8 bits)
5. 16-byte or 32-byte transfer by the DMAC
6. Setting the BLOCK bit in CMNCR to 1

Note that refresh requests and bus mastership requests are also accepted during DMA burst transfers.

This LSI has the bus mastership until a bus request is received from another device. Upon acknowledging the assertion (low level) of the external bus request signal $\overline{\text{BREQ}}$, the LSI releases the bus at the completion of the current bus cycle and asserts the $\overline{\text{BACK}}$ signal. After the LSI acknowledges the negation (high level) of the $\overline{\text{BREQ}}$ signal that indicates the slave has released the bus, it negates the $\overline{\text{BACK}}$ signal and resumes the bus usage.

The SDRAM issues an all bank precharge command (PALL) when active banks exist and releases the bus after completion of a PALL command.

The bus sequence is as follows. The address bus and data bus are placed in a high-impedance state synchronized with the rising edge of CKO. The bus mastership enable signal is asserted 0.5 cycles after the above timing, synchronized with the falling edge of CKO. The bus control signals ($\overline{BS}$, $\overline{CSn}$, $\overline{RAS}$, $\overline{CAS}$, DQMxx, $\overline{WEn}$, $\overline{RD}$, and $\overline{RDWR}$) are placed in the high-impedance state at subsequent rising edges of CKO. Bus request signals are sampled at the falling edge of CKO.

The sequence for reclaiming the bus mastership from a slave is described below. 1.5 cycles after the negation of $\overline{BREQ}$ is detected at the falling edge of CKO, the bus control signals are driven high. The $\overline{BACK}$ is negated at the next falling edge of the clock. The fastest timing at which actual bus cycles can be resumed after bus control signal assertion is at the rising edge of the CKO where address and data signals are driven. Figure 11.42 shows the bus arbitration timing.

In an original slave device designed by the user, multiple bus accesses are generated continuously to reduce the overhead caused by bus arbitration. In this case, to execute SDRAM refresh correctly, the slave device must be designed to release the bus mastership within the refresh interval time. To achieve this, the LSI instructs the $\overline{REFOUT}$ pin to request the bus mastership while the SDRAM waits for the refresh. The LSI asserts the $\overline{REFOUT}$ pin until the bus mastership is received. When one cycle has elapsed after de-assertion of the $\overline{BACK}$ signal, the $\overline{REFOUT}$ signal is de-asserted. If the slave releases the bus, this LSI acquires bus mastership to execute an SDRAM refresh.

The bus release by the $\overline{BREQ}$ and $\overline{BACK}$ signal handshaking requires some overhead. If the slave has many tasks, multiple bus cycles should be executed in a bus mastership acquisition. Reducing the cycles required for master to slave bus mastership transitions streamlines the system design.

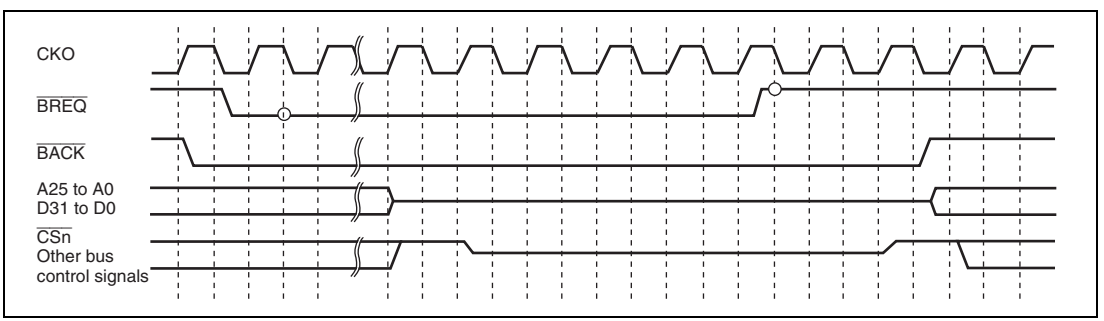


Figure 11.42 Bus Arbitration Timing

11.6 Usage Notes

(1) Reset

The bus state controller (BSC) can be initialized completely only at power-on reset. At power-on reset, all signals are negated and output buffers are turned off regardless of the bus cycle state. All control registers are initialized. In standby, sleep, and manual reset, control registers of the bus state controller are not initialized. At manual reset, the current bus cycle being executed is completed and then the access wait state is entered. If a 16-byte transfer is performed by a cache or if another LSI on-chip bus master module is executed when a manual reset occurs, the current access is cancelled in longword units because the access request is cancelled by the bus master at manual reset. If a manual reset is requested during cache fill operations, the contents of the cache cannot be guaranteed. Since the RTCNT continues counting up during manual reset signal assertion, a refresh request occurs to initiate the refresh cycle. In addition, a bus arbitration request by the $\overline{\text{BREQ}}$ signal can be accepted during manual reset signal assertion.

Some flash memories may specify a minimum time from reset release to the first access. To ensure this minimum time, the bus state controller supports a 7-bit counter (RBWTCNT). The counter is cleared to 0 by a power-on reset and it maintains the 0 state during the reset period. After power-on reset, RBWTCNT is counted up synchronously together with CKO and an external access will not be generated until RBWTCNT is counted up to H'007F. At manual reset, RBWTCNT is not cleared.

(2) Access from the CPU or FPU

In a read access to the cache from the CPU or FPU, the cache is searched. If the cache stores data, the CPU or FPU latches the data and completes the read access. If the cache does not store data, the CPU or FPU performs 32-byte read access to perform cache fill operations via the internal bus. If a cache miss occurs in byte or word operand access or at a branch to an odd word boundary (4n + 2), the CPU or FPU performs 32-byte access to perform a cache fill operation on the external interface.

For a cache-through area, the CPU or FPU performs access according to the actual access addresses. For an instruction fetch to an even word boundary (4n), the CPU or FPU performs longword access. For an instruction fetch to an odd word boundary (4n + 2), the CPU or FPU performs word access.

In a write access to the cache area from the CPU or FPU, the write cycle operation differs according to the cache write methods.

In write-back mode, the cache is first searched. If data is detected at the address corresponding to the cache, the data is then re-written to the cache. In the actual memory, data will not be re-written until data in the corresponding address is re-written. If data is not detected at the address corresponding to the cache, the cache is modified. In this case, data to be modified is first saved to the internal buffer, 32-byte data including the data corresponding to the address is then read, and data in the corresponding access of the cache is finally modified. Following these operations, a write-back cycle for the saved 32-byte data is executed.

In write-through mode, the cache is first searched. If data is detected at the address corresponding to the cache, the data is re-written to the cache simultaneously with the actual write via the internal bus. If data is not detected at the address corresponding to the cache, the cache is not modified but an actual write is performed via the internal bus.

In read cycles, the CPU or FPU is placed in the wait state until read operation has been completed. To continue the process after the data write to the device has been completed, perform a dummy read to the same address to check for completion of the write before the next process to be executed.

The write buffer of the BSC functions in the same way for an access by a bus master other than the CPU such as the DMAC. Accordingly, to perform dual address DMA transfers, the next read cycle is initiated before the previous write cycle is completed. Note, however, that if both the DMA source and destination addresses exist in external memory space, the next write cycle will not be initiated until the previous write cycle is completed.

(3) Access from Internal Bus Masters other than the CPU and FPU

Internal bus masters such as DMAC other than the CPU and FPU cannot access the cache memory. If an internal bus master such as DMAC writes data to an external memory, the contents of the external memory may differ from that of the cache memory. To prevent this problem, if the external memory whose contents is cached is written by an internal bus master such as DMAC other than the CPU and FPU, the corresponding cache memory should be purged by software.

(4) On-Chip Peripheral Module Access

To access an on-chip module register, two or more peripheral module clock (Pϕ) cycles are required from the internal bus (SuperHyway bus). Care must be taken in estimation in the system design process.

(5) External Bus Priority Order

Access via an external bus is performed in the priority order below:

$\overline{\text{BREQ}}$ > Refresh > DMAC > CPU

Note that next transfer is not performed until current transfer (e.g. burst transfer) has completed.

Section 12 Direct Memory Access Controller (DMAC)

This LSI includes the direct memory access controller (DMAC).

The DMAC can be used in place of the CPU to perform high-speed transfers between external devices that have DACK (transfer request acknowledge signal), external memory, on-chip memory, memory-mapped external devices, and on-chip peripheral modules.

12.1 Features

- Six channels (two channels can receive external requests)
- 4-Gbyte physical address space
- Data transfer unit is selectable: Byte, word (2 bytes), longword (4 bytes), 8 bytes, 16 bytes, and 32 bytes
- Maximum transfer count: 16,777,216 transfers
- Address mode: Dual address mode
- Transfer requests:
  External request, on-chip peripheral module request, or auto request can be selected.
  The following modules can issue an on-chip peripheral module request.
  — SCIF0/1/2/3/4/5, IrDA0/1, SIOF, SIM, ADC, DAC, and CMT0/1/2/3/4
- Selectable bus modes:
  Cycle steal mode (normal mode and intermittent mode) or burst mode can be selected.
- Selectable channel priority levels:
  The channel priority levels are selectable between fixed mode and round-robin mode.
- Interrupt request: An interrupt request can be generated to the CPU after half of the transfers ended, all transfers ended, or an address error occurred.
- External request detection: There are following four types of DREQ input detection.
  — Low level detection
  — High level detection
  — Rising edge detection
  — Falling edge detection
- Active level can be specified independently for the transfer request acknowledge signal (DACK) and DMA transfer end signal (TEND).

Figure 12.1 shows the block diagram of the DMAC.

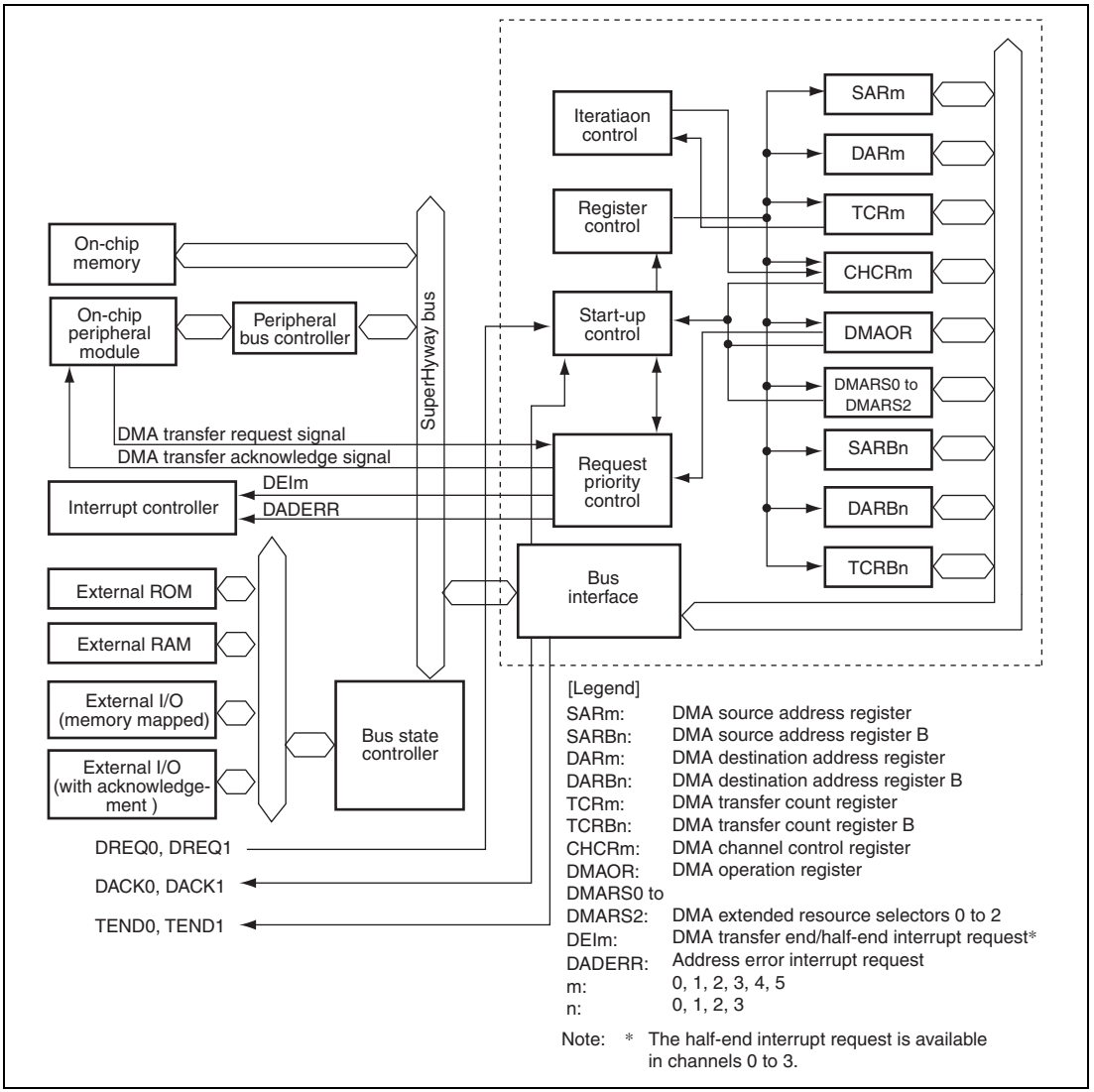


Figure 12.1 Block Diagram of DMAC

12.2 Input/Output Pins

The external pins for the DMAC are described below. Table 12.1 lists the configuration of the pins that are connected to external bus. The DMAC has pins for two channels (channels 0 and 1) for external bus use.

Table 12.1 Pin Configuration

Channel	Function	Pin Name	I/O	Description
0	DMA transfer request	DREQ0*[1]	Input	DMA transfer request input from external device to channel 0
	DMA transfer request acknowledge	DACK0*[2]	Output	Strobe as a response to the DMA transfer request, which is output from channel 0 to external device
	DMA transfer end notification	TEND0*[2]	Output	DMA transfer end output from channel 0 to external device
1	DMA transfer request	DREQ1*[1]	Input	DMA transfer request input from external device to channel 1
	DMA transfer request acknowledge	DACK1*[2]	Output	Strobe as a response to the DMA transfer request, which is output from channel 1 to external device
	DMA transfer end notification	TEND1*[2]	Output	DMA transfer end output from channel 1 to external device

Note: 1. Low-level detection with the initial value.
2. Active-low with the initial value.

12.3 Register Descriptions

Table 12.2 shows the configuration of registers of the DMAC. Table 12.3 shows the state of registers in each processing mode. The SAR for channel 0 is expressed such as SAR_0.

Table 12.2 Register Configuration of DMAC

Channel	Name	Abbreviation	R/W	Address	Access Size
0	DMA source address register_0	SAR_0	R/W	H'FE008020	32
	DMA destination address register_0	DAR_0	R/W	H'FE008024	32
	DMA transfer count register_0	TCR_0	R/W	H'FE008028	32
	DMA channel control register_0	CHCR_0	R/W	H'FE00802C	32
1	DMA source address register_1	SAR_1	R/W	H'FE008030	32
	DMA destination address register_1	DAR_1	R/W	H'FE008034	32
	DMA transfer count register_1	TCR_1	R/W	H'FE008038	32
	DMA channel control register_1	CHCR_1	R/W	H'FE00803C	32
2	DMA source address register_2	SAR_2	R/W	H'FE008040	32
	DMA destination address register_2	DAR_2	R/W	H'FE008044	32
	DMA transfer count register_2	TCR_2	R/W	H'FE008048	32
	DMA channel control register_2	CHCR_2	R/W	H'FE00804C	32
3	DMA source address register_3	SAR_3	R/W	H'FE008050	32
	DMA destination address register_3	DAR_3	R/W	H'FE008054	32
	DMA transfer count register_3	TCR_3	R/W	H'FE008058	32
	DMA channel control register_3	CHCR_3	R/W	H'FE00805C	32
Common	DMA operation register	DMAOR	R/W	H'FE008060	16
4	DMA source address register_4	SAR_4	R/W	H'FE008070	32
	DMA destination address register_4	DAR_4	R/W	H'FE008074	32
	DMA transfer count register_4	TCR_4	R/W	H'FE008078	32
	DMA channel control register_4	CHCR_4	R/W	H'FE00807C	32
5	DMA source address register_5	SAR_5	R/W	H'FE008080	32
	DMA destination address register_5	DAR_5	R/W	H'FE008084	32
	DMA transfer count register_5	TCR_5	R/W	H'FE008088	32
	DMA channel control register_5	CHCR_5	R/W	H'FE00808C	32

Channel	Name	Abbreviation	R/W	Address	Access Size
0	DMA source address register_4	SARB_0	R/W	H'FE008120	32
	DMA destination address register_4	DARB_0	R/W	H'FE008124	32
	DMA transfer count register_4	TCRB_0	R/W	H'FE008128	32
1	DMA source address register B_0	SARB_1	R/W	H'FE008130	32
	DMA destination address register B_0	DARB_1	R/W	H'FE008134	32
	DMA transfer count register B_0	TCRB_1	R/W	H'FE008138	32
2	DMA source address register B_1	SARB_2	R/W	H'FE008140	32
	DMA destination address register B_1	DARB_2	R/W	H'FE008144	32
	DMA transfer count register B_1	TCRB_2	R/W	H'FE008148	32
3	DMA source address register B_2	SARB_3	R/W	H'FE008150	32
	DMA destination address register B_2	DARB_3	R/W	H'FE008154	32
	DMA transfer count register B_2	TCRB_3	R/W	H'FE008158	32
0/1	DMA extended resource selector 0	DMARS0	R/W	H'FE009000	16
2/3	DMA extended resource selector 1	DMARS1	R/W	H'FE009004	16
4/5	DMA extended resource selector 2	DMARS2	R/W	H'FE009008	16

Table 12.3 State of Registers in Each Operating Mode

Channel	Abbreviation	Power-on Reset	Software Standby	Module Standby	Sleep
0	SAR_0	Initialized	Retained	Retained	Retained
	DAR_0	Initialized	Retained	Retained	Retained
	TCR_0	Initialized	Retained	Retained	Retained
	CHCR_0	Initialized	Retained	Retained	Retained
1	SAR_1	Initialized	Retained	Retained	Retained
	DAR_1	Initialized	Retained	Retained	Retained
	TCR_1	Initialized	Retained	Retained	Retained
	CHCR_1	Initialized	Retained	Retained	Retained
2	SAR_2	Initialized	Retained	Retained	Retained
	DAR_2	Initialized	Retained	Retained	Retained
	TCR_2	Initialized	Retained	Retained	Retained
	CHCR_2	Initialized	Retained	Retained	Retained
3	SAR_3	Initialized	Retained	Retained	Retained
	DAR_3	Initialized	Retained	Retained	Retained
	TCR_3	Initialized	Retained	Retained	Retained
	CHCR_3	Initialized	Retained	Retained	Retained
Common	DMAOR	Initialized	Retained	Retained	Retained
4	SAR_4	Initialized	Retained	Retained	Retained
	DAR_4	Initialized	Retained	Retained	Retained
	TCR_4	Initialized	Retained	Retained	Retained
	CHCR_4	Initialized	Retained	Retained	Retained
5	SAR_5	Initialized	Retained	Retained	Retained
	DAR_5	Initialized	Retained	Retained	Retained
	TCR_5	Initialized	Retained	Retained	Retained
	CHCR_5	Initialized	Retained	Retained	Retained
0	SARB_0	Initialized	Retained	Retained	Retained
	DARB_0	Initialized	Retained	Retained	Retained
	TCRB_0	Initialized	Retained	Retained	Retained

Channel	Abbreviation	Power-on Reset	Software Standby	Module Standby	Sleep
1	SARB_1	Initialized	Retained	Retained	Retained
	DARB_1	Initialized	Retained	Retained	Retained
	TCRB_1	Initialized	Retained	Retained	Retained
2	SARB_2	Initialized	Retained	Retained	Retained
	DARB_2	Initialized	Retained	Retained	Retained
	TCRB_2	Initialized	Retained	Retained	Retained
3	SARB_3	Initialized	Retained	Retained	Retained
	DARB_3	Initialized	Retained	Retained	Retained
	TCRB_3	Initialized	Retained	Retained	Retained
0/1	DMARS0	Initialized	Retained	Retained	Retained
2/3	DMARS1	Initialized	Retained	Retained	Retained
4/5	DMARS2	Initialized	Retained	Retained	Retained

12.3.1 DMA Source Address Registers (SAR_0 to SAR_5)

SARs are 32-bit readable/writable registers that specify the source address of a DMA transfer. During a DMA transfer, these registers indicate the next source address.

To transfer data in word or in longword units, specify the address with word or longword address boundary. When transferring data in 8-byte, 16-byte, or 32-byte units, an 8-byte, 16-byte, or 32-byte boundary must be set for the source address value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SAR															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAR															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.3.2 DMA Source Address Registers (SARB_0 to SARB_3)

SARB are 32-bit readable/writable registers that specify the source address of a DMA transfer that is set in SAR again in repeat/reload mode. Data to be written from the CPU to SAR is also written to SARB. To set SARB address that differs from SAR address, write data to SARB after SAR.

To transfer data in word or in longword units, specify the address with word or longword address boundary. When transferring data in 8-byte, 16-byte, or 32-byte units, an 8-byte, 16-byte, or 32-byte boundary must be set for the source address value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SARB															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SARB															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.3.3 DMA Destination Address Registers (DAR_0 to DAR_5)

DARs are 32-bit readable/writable registers that specify the destination address of a DMA transfer. During a DMA transfer, these registers indicate the next destination address.

To transfer data in word or in longword units, specify the address with word or longword address boundary. When transferring data in 8-byte, 16-byte, or 32-byte units, an 8-byte, 16-byte, or 32-byte boundary must be set for the source address value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DAR															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DAR															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.3.4 DMA Destination Address Registers (DARB_0 to DARB_3)

DARB are 32-bit readable/writable registers that specify the destination address of a DMA transfer that is set in DAR again in repeat/reload mode. Data to be written from the CPU to DAR is also written to DARB. To set DARB address that differs from DAR address, write data to DARB after DAR.

To transfer data in word or in longword units, specify the address with word or longword address boundary. When transferring data in 8-byte, 16-byte, or 32-byte units, an 8-byte, 16-byte, or 32-byte boundary must be set for the source address value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DARB															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DARB															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.3.5 DMA Transfer Count Registers (TCR_0 to TCR_5)

TCR are 32-bit readable/writable registers that specify the DMA transfer count. The number of transfers is 1 when the setting is H'00000001, 16,777,215 when H'00FFFFFF is set, and 16,777,216 (the maximum) when H'00000000 is set. During a DMA transfer, these registers indicate the remaining transfer count.

The upper eight bits of TCR are always read as 0, and the write value should always be 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TCR															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCR															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.3.6 DMA Transfer Count Registers (TCRB_0 to TCRB_3)

TCRB are 32-bit readable/writable registers. Data to be written from the CPU to TCR is also written to TCRB. While the half end function is used, TCRB are used as the initial value hold registers to detect a half end. Also, TCRB specify the number of DMA transfers which are set in TCR in repeat mode. TCRB specify the number of DMA transfers and are used as transfer count counters in reload mode.

In reload mode, the lower 16 bits operate as transfer count counters, values of SAR and DAR are updated after the value of the lower 16 bits became 0, and then the value of the upper 16 bits of TCRB are loaded to the lower 16 bits. In upper 16 bits, set the number of transfers which starts reloading. In reload mode, the same number of transfers should be set in both upper and lower 16 bits. Also, set the HIE bit in CHCR to 0 and do not use the half end function.

For details on the half end function, see section 12.4.5, Repeat Mode Transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TCRB															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCRB															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.3.7 DMA Channel Control Registers (CHCR_0 to CHCR_5)

CHCR are 32-bit readable/writable registers that control the DMA transfer mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	LCKN	—	—	RPT[2:0]			—	DO	—	TS[3:2]		HE	HIE	AM	AL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R/W	R/W	R/W	R	R/W	R	R/W	R/W	R/(W)*	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DM[1:0]		SM[1:0]		RS[3:0]				DL	DS	TB	TS[1:0]		IE	TE	DE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30	LCKN	0	R/W	Bus Release Enable in Cycle Steal Mode Specifies whether to release the bus to a bus master other than the DMAC between reading and writing in cycle steal mode. With the initial setting, the DMAC retains the bus mastership. Setting this bit to 1 allows acceptance of bus requests from a bus master other than the DMAC, which increases the bus usage rate of the overall system. This bit can be set in cycle steal mode. Do not set it to 1 in burst mode. 0: Bus release between reading and writing is disabled 1: Bus release between reading and writing is enabled

Bit	Bit Name	Initial Value	R/W	Descriptions
29, 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 25	RPT[2:0]	000	R/W	DMA Setting Renewal Specify These bits are enabled in CHCR_0 to CHCR_3. 000: Normal mode (DMAC operation) 001: Repeat mode SAR/DAR/TCR used as repeat area 010: Repeat mode DAR/TCR used as repeat area 011: Repeat mode SAR/TCR used as repeat mode 100: Reserved (setting prohibited) 101: Reload mode SAR/DAR/TCR used as reload area 110: Reload mode DAR/TCR used as reload area 111: Reload mode SAR/TCR used as reload area
22	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
23	DO	0	R/W	DMA Overrun Selects whether detection takes place at overrun 0 or overrun 1 when DREQ level detection is used. This bit is valid only in CHCR_0 and CHCR_1. 0: Detects DREQ by overrun 0 1: Detects DREQ by overrun 1

Bit	Bit Name	Initial Value	R/W	Descriptions
22	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
21, 20	TS[3:2]	00	R/W	DMA Transfer Size Specify With TS[1:0], these bits specify the DMA transfer size. When the transfer source or transfer destination is a register of an on-chip peripheral module with a transfer size set, a proper transfer size for the register should be set. In 2-division transfer mode, 16/32-byte data is halved and transferred in two operations. When 16-byte data is transferred in a peripheral module, 16-byte 2-division transfer should be selected. For the transfer source or destination address specified by SAR or DAR, an address boundary should be set according to the transfer data size. TS[3:0] 0000: Byte units transfer 0001: Word (2-byte) units transfer 0010: Longword (4-byte) units transfer 0011: 16-byte units transfer 0100: 32-byte units transfer 0111: 8-byte units transfer 1011: 16-byte 2-division (8-byte units × 2) transfer 1100: 32-byte 2-division (16-byte units × 2) transfer Other than above: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Descriptions
19	HE	0	R/(W)*	Half End Flag After HIE (bit 18) is set to 1 and the number of transfers become half of TCR (1 bit shift to right) which is set before transfer starts, HE becomes 1. The HE bit is not set when transfers are ended by an NMI interrupt or address error, or by clearing the DE bit and the DME bit in DMAOR before the number of transfers is decreased to half of the TCR value set preceding the transfer. The HE bit is kept set when the transfer ends by an NMI interrupt or address error, or clearing the DE bit (bit 0) or the DME bit in DMAOR after the HE bit is set to 1. To clear the HE bit, write 0 after reading 1 in the HE bit. This bit is valid only in CHCR_0 to CHCR_3. 0: During the DMA transfer or DMA transfer has been interrupted TCR > (TCR set before transfer)/2 [Clearing condition] • Writing 0 after HE = 1 is read. 1: TCR ≤ (TCR set before transfer)/2
18	HIE	0	R/W	Half End Interrupt Enable Specifies whether an interrupt request is generated to the CPU when the read cycle of the transfer that the number of transfers is decreased to half of the TCR value set before the transfer has ended. If the HIE bit is set to 1, an interrupt request is generated to the CPU when the HE bit is set. To confirm that the half of the transfer has ended, execute a dummy read of the destination space after issuing the SYNCO instruction. Clear this bit to 0 while reload mode is set. This bit is valid in CHCR_0 to CHCR_3. 0: Half end interrupt disabled 1: Half end interrupt enabled
17	AM	0	R/W	Acknowledge Mode Selects whether DACK is output in data read cycle or in data write cycle. This bit is valid only in CHCR_0 and CHCR_1. 0: DACK output in read cycle 1: DACK output in write cycle

Bit	Bit Name	Initial Value	R/W	Descriptions
16	AL	0	R/W	Acknowledge Level Specifies whether the DACK and TEND signals are active-high or active-low. This bit is valid only in CHCR0 and CHCR_1. 0: DACK and TEND outputs are active -low 1: DACK and TEND outputs are active -high
15, 14	DM[1:0]	00	R/W	Destination Address Mode Specify whether the DMA destination address is incremented, decremented, or left fixed. 00: Fixed destination address Since the address set in DAR is not modified, the same address is output in the second and subsequent transfers. The address is incremented at the first and second transfers in 16/32-byte division transfer mode. 01: Destination address is incremented +1 in byte units transfer +2 in word units transfer +4 in longword units transfer +8 in 8-byte units transfer +16 in 16-byte units transfer +32 in 32-byte units transfer 10: Destination address is decremented –1 in byte units transfer –2 in word units transfer –4 in longword units transfer Setting prohibited in 8/16/32-byte units transfer 11: Fixed destination address Set to prevent an address from being changed in the objective modules. The address is not changed even in 16/32-byte division transfer mode. Example: When specifying FIFOs in the external devices and peripheral modules.

Bit	Bit Name	Initial Value	R/W	Descriptions
13, 12	SM[1:0]	00	R/W	Source Address Mode Specify whether the DMA source address is incremented, decremented, or left fixed. 00: Fixed source address Since the address set in SAR is not modified, the same address is output in the second and subsequent transfers. The address is incremented at the first and second transfers in 16/32-byte division transfer mode. 01: Source address is incremented +1 in byte units transfer +2 in word units transfer +4 in longword units transfer +8 in 8-byte units transfer +16 in 16-byte units transfer +32 in 32-byte units transfer 10: Source address is decremented –1 in byte units transfer –2 in word units transfer –4 in longword units transfer Setting prohibited in 8/16/32-byte units transfer 11: Fixed source address Set to prevent an address from being changed in the objective modules. The address is not changed even in 16/32-byte division transfer mode. Example: When specifying FIFOs in the external devices and peripheral modules.
11 to 8	RS[3:0]	0000	R/W	Resource Select Specify which transfer requests will be sent to the DMAC. The changing of transfer request source should be done in the state that the DMA enable bit (DE) is set to 0. 0000: External request 0100: Auto request 1000: Selected by DMA extended resource selector (DMARS) Other than above: Setting prohibited Note: External request specification is valid only in CHCR_0 and CHCR_1. External request cannot be selected in CHCR_2 to CHCR_5.

Bit	Bit Name	Initial Value	R/W	Descriptions
7	DL	0	R/W	DREQ Level and DREQ Edge Select
6	DS	0	R/W	Specify the detecting method of the DREQ pin input. These bits are valid only in CHCR_0 and CHCR_1. Even in channels 0 and 1, also, if the transfer request source is specified as an on-chip peripheral module or if an auto-request is specified, these bits are invalid. 00: DREQ detected in low level 01: DREQ detected at falling edge 10: DREQ detected in high level 11: DREQ detected at rising edge
5	TB	0	R/W	Transfer Bus Mode Specifies the bus mode when DMA transfers data. 0: Cycle steal mode 1: Burst mode
4, 3	TS[1:0]	00	R/W	DMA Transfer Size Specify See the description of TS[3:2] (bits 21 and 20).
2	IE	0	R/W	Interrupt Enable Specifies whether an interrupt request is generated to the CPU at the end of the final DMA transfer. Setting this bit to 1 generates an interrupt request (DMINT) to the CPU when the TE bit is set to 1 and a read cycle of the final DMA transfer has ended. To confirm that the final transfer has ended, execute a dummy read of the destination space after issuing the SYNCO instruction. 0: Interrupt request disabled 1: Interrupt request enabled

Bit	Bit Name	Initial Value	R/W	Descriptions
1	TE	0	R/(W)*	Transfer End Flag The TE bit is set to 1 when DMA transfer count register (TCR) is set to 0 (when the DMAC starts executing the final DMA transfer). The TE bit is not set, if DMA transfer ends due to an NMI interrupt or DMA address error before TCR is cleared to 0, or if DMA transfer is ended by clearing the DE bit and DME bit in DMA operation register (DMAOR). To clear the TE bit, the TE bit should be read as 1, and then, 0 is written to. Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled. 0: When DMA transfer is being performed or DMA transfer has been interrupted [Clearing condition]: Write 0 after TE is read as 1 1: TCR = 0 (when the final DMA transfer is being performed or the DMA transfer ends)
0	DE	0	R/W	DMA Enable Enables or disables the DMA transfer. In auto request mode, DMA transfer starts by setting the DE bit and DME bit in DMAOR to 1. In this time, all of the bits TE, NMIF, and AE in DMAOR must be 0. In an external request or peripheral module request, DMA transfer starts if DMA transfer request is generated by the devices or peripheral modules after setting the bits DE and DME to 1. In this case, however, all of the bits TE, NMIF, and AE must be 0, which is the same as in the case of auto request mode. Clearing the DE bit to 0 can terminate the DMA transfer. 0: DMA transfer disabled 1: DMA transfer enabled

Note: * Writing 0 is possible to clear the flag.

12.3.8 DMA Operation Register (DMAOR)

DMAOR is a 16-bit readable/writable register that specifies the priority level of channels in DMA transfer. This register also shows the DMA transfer status and is common to channels 0 to 5.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMS[3:0]				—	—	PR[1:0]		—	—	—	—	—	AE	NMIF	DME
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R	R	R	R/(W)*	R/(W)*	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	CMS[3:0]	0000	R/W	Cycle Steal Mode Select 1, 0 Select either normal mode or intermittent mode in cycle steal mode. It is necessary that all channel's bus modes are set to cycle steal mode to make valid intermittent mode. 0000: Normal mode 0010: Intermittent mode 16 Executes one DMA transfer in each of 16 clocks of an external bus clock. 0011: Intermittent mode 64 Executes one DMA transfer in each of 64 clocks of an external bus clock. 0100: Intermittent mode 256 Executes one DMA transfer in each of 256 clocks of an external bus clock. Other than above: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PR[1:0]	00	R/W	Priority Mode Select the priority level between channels when there are transfer requests for multiple channels simultaneously. 00: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 01: CH0 > CH2 > CH3 > CH1 > CH4 > CH5 10: Setting prohibited 11: Round-robin mode
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	AE	0	R/(W)*	Address Error Flag Indicates that an address error interrupt occurred during DMA transfer. This bit is set under following conditions: • The value set in SAR or DAR does not match to the transfer size boundary. • The transfer source or transfer destination is invalid space. • The transfer source or transfer destination is in module stop mode If this bit is set, DMA transfer is disabled even if the DE bit in CHCR and the DME bit in DMAOR are set to 1. 0: No DMAC address error interrupt [Clearing condition] • Writing AE = 0 after AE = 1 read 1: DMAC address error interrupt occurs

Bit	Bit Name	Initial Value	R/W	Description
1	NMIF	0	R/(W)*	NMI Flag Indicates that an NMI interrupt occurred. If this bit is set, DMA transfer is disabled even if the DE bit in CHCR and the DME bit in DMAOR are set to 1. When the NMI is input, the DMA transfer in progress can be done in at least one transfer unit. When the DMAC is not in operational, the NMIF bit is set to 1 even if the NMI interrupt was input. 0: No NMI interrupt [Clearing condition] • Writing NMIF = 0 after NMIF = 1 read 1: NMI interrupt occurs
0	DME	0	R/W	DMA Master Enable Enables or disables DMA transfers on all channels. If the DME bit and the DE bit in CHCR are set to 1, transfer is enabled. In this time, all of the bits TE in CHCR, NMIF, and AE in DMAOR must be 0. If this bit is cleared during transfer, transfers in all channels are terminated. 0: Disables DMA transfers on all channels 1: Enables DMA transfers on all channels

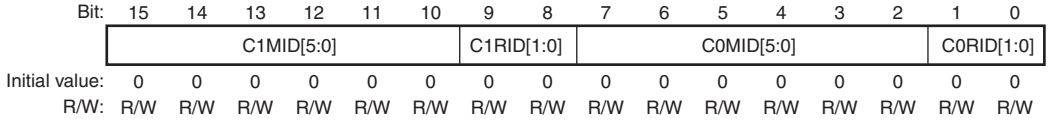
Note: * Writing 0 is possible to clear the flag.

12.3.9 DMA Extended Resource Selectors (DMARS0 to DMARS2)

DMARS are 16-bit readable/writable registers that specify the DMA transfer sources from peripheral modules in each channel. DMARS0 specifies for channels 0 and 1, DMARS1 specifies for channels 2 and 3, and DMARS2 specifies for channels 4 and 5. This register can set the transfer request of SCIF, SCIFA, SIOF, IrDA, SIM, ADC, and CMT.

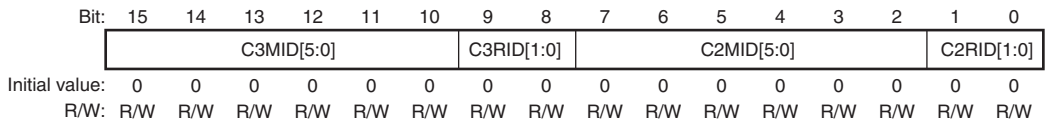
When MID/RID other than the values listed in table 12.4 is set, the operation of this LSI is not guaranteed. The transfer request from DMARS is valid only when the resource select bits RS[3:0] has been set to B'1000 for CHCR_0 to CHCR_5 registers. Otherwise, even if DMARS has been set, transfer request source is not accepted.

- DMARS0



Bit	Bit Name	Initial Value	R/W	Description
15 to 10	C1MID[5:0]	000000	R/W	Transfer request module ID5 to ID0 for DMA channel 1 (MID) See table 12.4.
9, 8	C1RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA channel 1 (RID) See table 12.4.
7 to 2	C0MID[5:0]	000000	R/W	Transfer request module ID5 to ID0 for DMA channel 0 (MID) See table 12.4
1, 0	C0RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA channel 0 (RID) See table 12.4.

- DMARS1



Bit	Bit Name	Initial Value	R/W	Description
15 to 10	C3MID[5:0]	000000	R/W	Transfer request module ID5 to ID0 for DMA channel 3 (MID) See table 12.4.
9, 8	C3RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA channel 3 (RID) See table 12.4.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	C2MID[5:0]	000000	R/W	Transfer request module ID5 to ID0 for DMA channel 2 (MID) See table 12.4.
1, 0	C2RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA channel 2 (RID) See table 12.4.

- DMARS2

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C5MID[5:0]						C5RID[1:0]		C4MID[5:0]						C4RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	C5MID[5:0]	000000	R/W	Transfer request module ID5 to ID0 for DMA channel 5 (MID) See table 12.4.
9, 8	C5RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA channel 5 (RID) See table 12.4.
7 to 2	C4MID[5:0]	000000	R/W	Transfer request module ID5 to ID0 for DMA channel 4 (MID) See table 12.4.
1, 0	C4RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA channel 4 (RID) See table 12.4.

Table 12.4 Transfer Request Sources

Peripheral Module	Setting Value for One Channel (MID + RID)	MID	RID	Function
CMT0	H'03	B'000000	B'11	—
CMT1	H'07	B'000001	B'11	—
CMT2	H'0B	B'000010	B'11	—
CMT3	H'0F	B'000011	B'11	—
CMT4	H'13	B'000100	B'11	—
SCIF0	H'21	B'001000	B'01	Transmission
	H'22		B'10	Reception
SCIF1	H'25	B'001001	B'01	Transmission
	H'26		B'10	Reception
SCIF2	H'29	B'001010	B'01	Transmission
	H'2A		B'10	Reception
SCIF3	H'2D	B'001011	B'01	Transmission
	H'2E		B'10	Reception
SCIF4	H'31	B'001100	B'01	Transmission
	H'32		B'10	Reception
SCIF5	H'35	B'001101	B'01	Transmission
	H'36		B'10	Reception
IrDA0	H'39	B'001110	B'01	Transmission
	H'3A		B'10	Reception
IrDA1	H'3D	B'001111	B'01	Transmission
	H'3E		B'10	Reception
SIOF	H'51	B'010100	B'01	Transmission
	H'52		B'10	Reception
ADC	H'6B	B'011010	B'11	—
SIM	H'A1	B'101000	B'01	Transmission
	H'A2		B'10	Reception

12.4 Operation

When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. In bus mode, burst mode or cycle steal mode can be selected.

12.4.1 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or destination, but they can also be generated by external devices or on-chip peripheral modules that are neither the source nor the destination. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. The request mode is selected in the bits RS[3:0] in CHCR_0 to CHCR_3, and DMARS_0 to DMARS_2.

(1) Auto-Request Mode

When there is no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip peripheral module unable to request a transfer, auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits in CHCR_0 to CHCR_3 and the DME bit in DMAOR are set to 1, the transfer begins so long as the AE and NMIF bits in DMAOR are all 0.

(2) External Request Mode

n this mode, a transfer is initiated by the transfer request signal (DREQ0 or DREQ1) from an external device. This mode is available only in channels 0 and 1. When DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), transfer starts upon a DREQ input.

Choose to detect DREQ by either the edge or level of the signal input with the DL bit and DS bit in CHCR_0 or CHCR_1 as shown in table 12.5. The source of the transfer request does not have to be the data transfer source or destination.

Table 12.5 Selecting External Request Detection by DL and DS Bits

CHCR_0, CHCR_1		
DL	**DS**	**Detection of External Request**
0	0	Low level detection
	1	Falling edge detection
1	0	High level detection
	1	Rising edge detection

When DREQ is accepted, the DREQ pin becomes request accept disabled state. After issuing acknowledge signal DACK for the accepted DREQ, the DREQ pin again becomes request accept enabled state.

When DREQ is used by level detection, there are following two cases by the timing to detect the next DREQ after outputting DACK.

- Overrun 0: Transfer is aborted after the same number of transfer has been performed as requests.
- Overrun 1: Transfer is aborted after transfers have been performed for (the number of requests plus 1) times.

The DO bit in CHCR selects this overrun 0 or overrun 1.

Table 12.6 Selecting External Request Detection with DO Bit

CHCR_0, CHCR_1	
DO	**External Request**
0	Overrun 0
1	Overrun 1

(3) On-Chip Peripheral Module Request Mode

In this mode, a transfer is performed at the transfer request signal of an on-chip peripheral module. Transfer request signals are the transmit data empty transfer request and receive data full transfer request from the SCIF0/1/2/3/4/5, IrDA0/1, SIOF, and SIM and transfer requests from the ADC and CMT0/1/2/3/4, all of which are selected by DMARS0/1/2.

When this mode is selected, if the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), a transfer is performed upon the input of a transfer request signal.

When a transmit data empty transfer request of the SCIF is set as the transfer request, the transfer destination must be the SCIF's transmit data register. Likewise, when receive data full transfer request of the SCIF is set as the transfer request, the transfer source must be the SCIF's receive data register. These conditions also apply to the SIOF, FLCTL, SIUA, and SIUB.

The number of the receive FIFO triggers can be set as a transfer request depending on an on-chip peripheral module. Data needs to be read after the DMA transfer is ended, because data may be remained in the receive FIFO when the receive FIFO trigger condition is not satisfied.

Table 12.7 Selecting On-Chip Peripheral Module Request Modes with Bits RS[3:0]

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Source	Destination	Bus Mode
RS[3:0]	MID	RID					
1000	000000	11	CMT0	Compare-match transfer request	Any	Any	Cycle steal
	000001	11	CMT1	Compare-match transfer request	Any	Any	Cycle steal
	000010	11	CMT2	Compare-match transfer request	Any	Any	Cycle steal
	000011	11	CMT3	Compare-match transfer request	Any	Any	Cycle steal
	000100	11	CMT4	Compare-match transfer request	Any	Any	Cycle steal
	001000	01	SCIF0 transmitter	TXI (transmit FIFO data empty)	Any	SCFTDR0	Cycle steal
		10	SCIF0 receiver	RXI (receive FIFO data full)	SCFRDR0	Any	Cycle steal
	001001	01	SCIF1 transmitter	TXI (transmit FIFO data empty)	Any	SCFTDR1	Cycle steal
		10	SCIF1 receiver	RXI (receive FIFO data full)	SCFRDR1	Any	Cycle steal
	001010	01	SCIF2 transmitter	TXI (transmit FIFO data empty)	Any	SCFTDR2	Cycle steal
		10	SCIF2 receiver	RXI (receive FIFO data full)	SCFRDR2	Any	Cycle steal

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Source	Destination	Bus Mode
RS[3:0]	MID	RID					
1000	001011	01	SCIF3 transmitter	TXI (transmit FIFO data empty)	Any	SCFTDR3	Cycle steal
		10	SCIF3 receiver	RXI (receive FIFO data full)	SCFRDR3	Any	Cycle steal
	001100	01	SCIF4 transmitter	TXI (transmit FIFO data empty)	Any	SCFATDR4	Cycle steal
		10	SCIF4 receiver	RXI (receive FIFO data full)	SCAFRDR4	Any	Cycle steal
	001101	01	SCIF5 transmitter	TXI (transmit FIFO data empty)	Any	SCAFTDR5	Cycle steal
		10	SCIF5 receiver	RXI (receive FIFO data full)	SCAFRDR5	Any	Cycle steal
	001110	01	IrDA0 transmitter	Transmit empty transfer request	Any	IRIF0_UART3	Cycle steal
		10	IrDA0 receiver	Receive full transfer request	IRIF0_UART4	Any	Cycle steal
	001111	01	IrDA1 transmitter	Transmit empty transfer request	Any	IRIF1_UART3	Cycle steal
		10	IrDA1 receiver	Receive full transfer request	IRIF1_UART4	Any	Cycle steal
	010100	01	SIOF transmitter	TXI (transmit FIFO data empty)	Any	SITDR	Cycle steal
		10	SIOF receiver	RXI (receive FIFO data full)	SIRDR	Any	Cycle steal
	011010	11	ADC	ADI (A/D conversion end)	ADDR	Any	Cycle steal
	101000	01	SIM transmitter	TXI (transmit FIFO data empty)	Any	SCTDR	Cycle steal
		10	SIM receiver	RXI (receive FIFO data full)	SCRDR	Any	Cycle steal

12.4.2 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it transfers data according to a predetermined priority. Two modes (fixed mode and round-robin mode) are selected by the bits PR[1:0] in DMAOR.

(1) Fixed Mode

In this mode, the priority levels among the channels remain fixed. There are two kinds of fixed modes as follows:

- CH0 > CH1 > CH2 > CH3 > CH4 > CH5
- CH0 > CH2 > CH3 > CH1 > CH4 > CH5

These are selected by the bits PR[1:0] in DMAOR.

(2) Round-Robin Mode

In round-robin mode each time data of one transfer unit (byte, word, longword, 8-byte, 16-byte, or 32-byte unit) is transferred on one channel, the priority is rotated. The channel on which the transfer was just finished rotates to the bottom of the priority. The round-robin mode operation is shown in figure 12.2. The priority of round-robin mode is CH0 > CH1 > CH2 > CH3 > CH4 > CH5 immediately after reset.

When round-robin mode is specified, do not mix the cycle steal mode and the burst mode in multiple channels' bus modes.

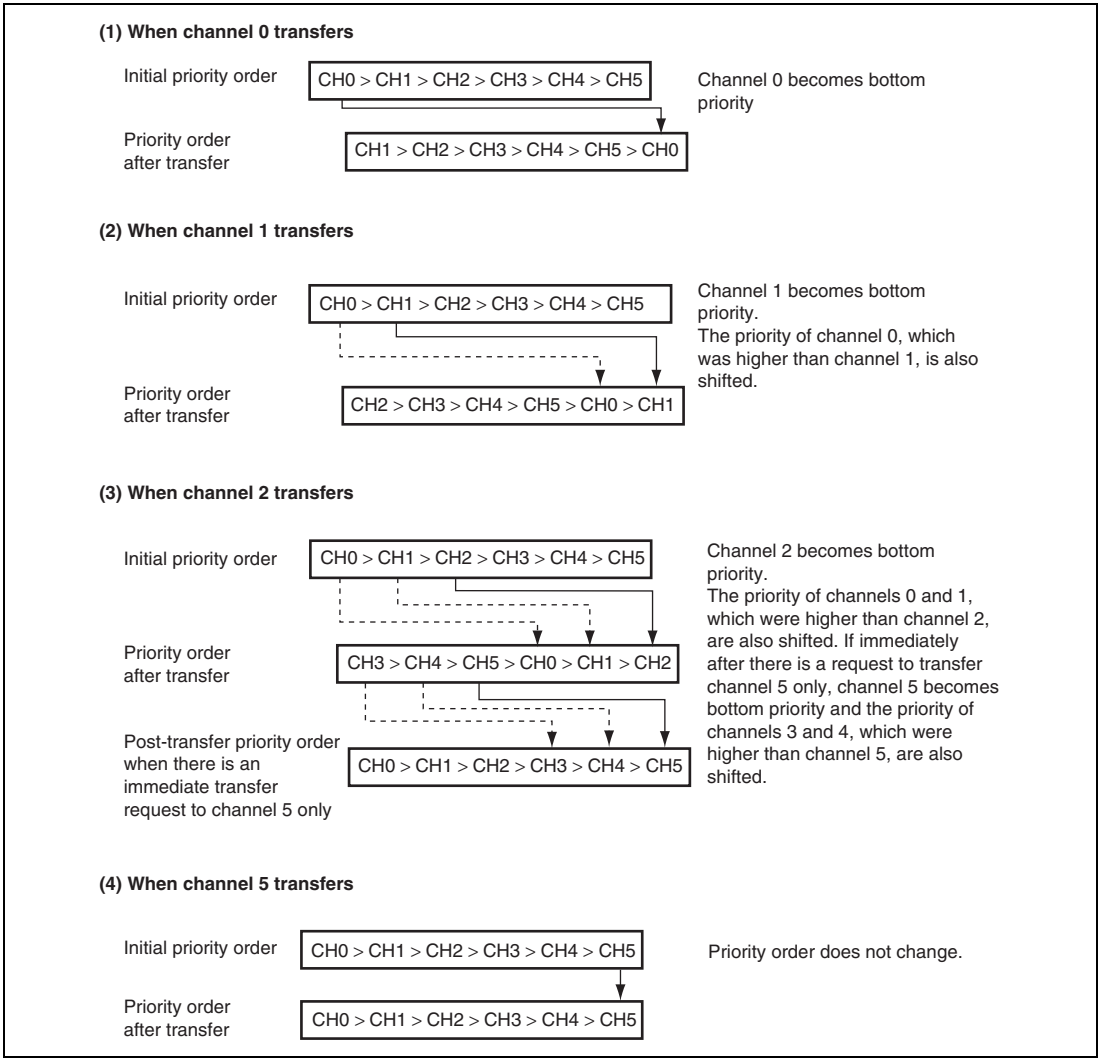


Figure 12.2 Round-Robin Mode

Figure 12.3 shows how the priority changes when channel 0 and channel 3 transfers are requested simultaneously and a channel 1 transfer is requested during the channel 0 transfer. The DMAC operates as follows:

1. Transfer requests are generated simultaneously to channels 0 and 3.
2. Channel 0 has a higher priority, so the channel 0 transfer begins first (channel 3 waits for transfer).
3. A channel 1 transfer request occurs during the channel 0 transfer (channels 1 and 3 are both waiting)
4. When the channel 0 transfer ends, channel 0 becomes lowest priority.
5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 transfer begins (channel 3 waits for transfer).
6. When the channel 1 transfer ends, channel 1 becomes lowest priority.
7. The channel 3 transfer begins.
8. When the channel 3 transfer ends, channels 3 and 2 shift downward in priority so that channel 3 becomes the lowest priority.

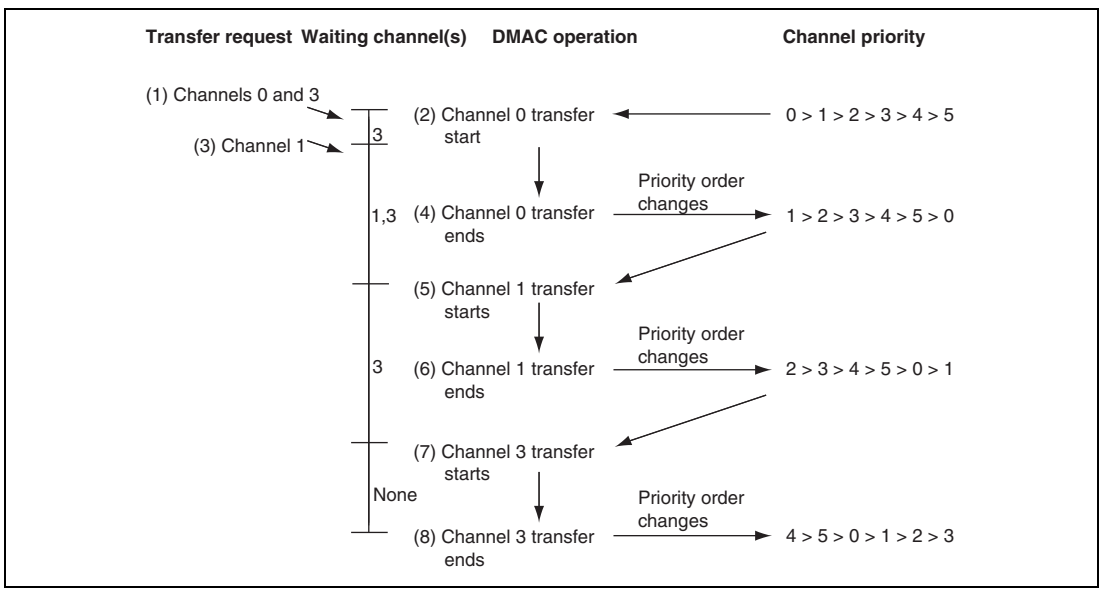


Figure 12.3 Changes in Channel Priority in Round-Robin Mode

12.4.3 DMA Transfer Types

DMA transfer type is dual address mode transfer. They depend on the number of bus cycles of access to source and destination. A data transfer timing depends on the bus mode, which has cycle steal mode and burst mode. The DMAC supports the transfers shown in table 12.8.

Table 12.8 Supported DMA Transfers

	Destination				
Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Peripheral Module**	**IL Memory**
External device with DACK	Not available	Y	Y	Not available	Not available
External memory	Y	Y	Y	Y	Y
Memory-mapped external device	Y	Y	Y	Y	Y
On-chip peripheral module	Not available	Y	Y	Y	Y
IL memory	Not available	Y	Y	Y	Y

[Legend]

Y: Transfer is enabled

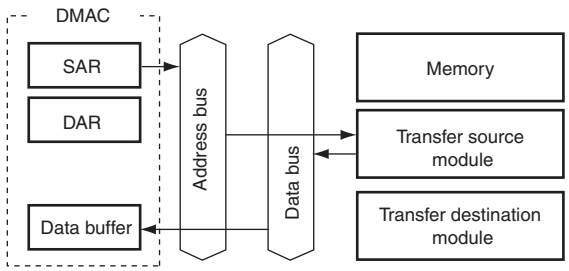
Note: For on-chip peripheral modules, 16-byte transfer is available only by registers which can be accessed in longword units.

(1) Address Modes

(a) Dual Address Mode

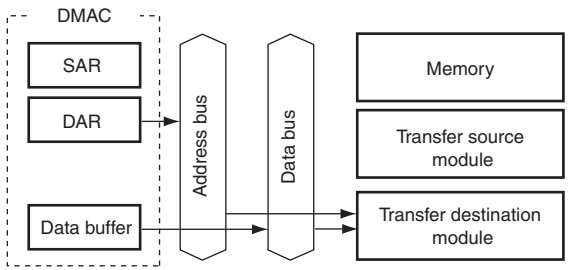
In dual address mode, both the transfer source and destination are accessed by an address. The source and destination can be located externally or internally.

DMA transfer requires two bus cycles because data is read from the transfer source in a data read cycle and written to the transfer destination in a data write cycle. At this time, transfer data is temporarily stored in the DMAC. In the transfer between external memories as shown in figure 12.4, data is read to the DMAC from one external memory in a data read cycle, and then that data is written to the other external memory in a write cycle.



The SAR value is an address, data is read from the transfer source module, and the data is temporarily stored in the DMAC.

First bus cycle



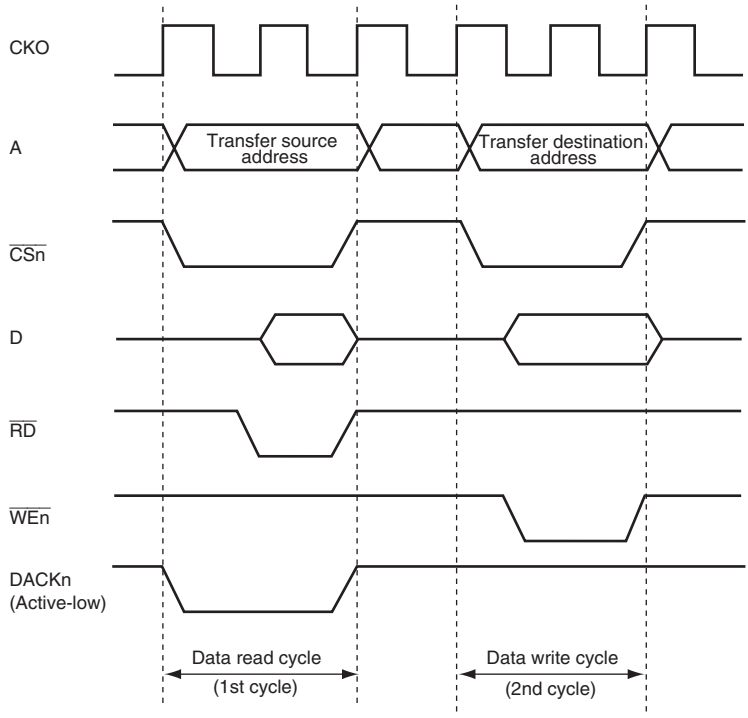
The DAR value is an address and the value stored in the data buffer in the DMAC is written to the transfer destination module.

Second bus cycle

Figure 12.4 Data Flow of Dual Address Mode

Auto request, external request, and on-chip peripheral module request are available for the transfer request. DACK can be output in read cycle or write cycle in dual address mode. CHCR can specify whether the DACK is output in read cycle or write cycle.

Figure 12.5 shows an example of DMA transfer timing in dual address mode.



Note: In transfer between external memories, with DACK output in the read cycle, DACK output timing is the same as that of CSn.

Figure 12.5 Example of DMA Transfer Timing in Dual Mode (Source: Ordinary Memory, Destination: Ordinary Memory)

(2) Bus Modes

There are two bus modes: cycle steal mode and burst mode. Select the mode in the TB bits in CHCR.

(a) Cycle-Steal Mode

In cycle-steal mode, select either normal mode or intermittent mode by the CMS[3:0] bits in DMAOR.

- Normal mode

  In cycle-steal normal mode, the bus mastership is given to another bus master after a one-transfer-unit (byte, word, longword, 8-byte, 16-byte, or 32-byte unit) DMA transfer. When another transfer request occurs, the bus mastership is obtained from the other bus master and a transfer is performed for one transfer unit. When that transfer ends, the bus mastership is passed to the other bus master. This is repeated until the transfer end conditions are satisfied.

  In cycle-steal normal mode, transfer areas are not affected regardless of settings of the transfer request source, transfer source, and transfer destination.

  Figure 12.6 shows an example of DMA transfer timing in cycle-steal normal mode. Transfer conditions shown in the figure are:

  — Dual address mode

  — DREQ low level detection

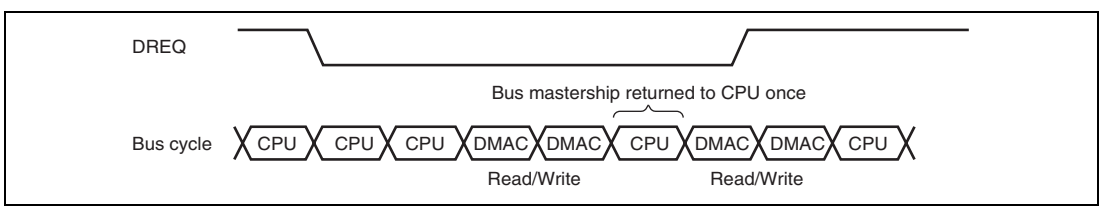


Figure 12.6 DMA Transfer Example in Cycle-Steal Normal Mode (Dual Address, DREQ Low Level Detection)

- Intermittent mode 16, intermittent mode 64, and intermittent mode 256

  In intermittent mode of cycle steal, the DMAC returns the bus mastership to other bus master whenever a unit of transfer (byte, word, longword, 8-byte, 16-byte, or 32-byte unit) is complete. If the next transfer request occurs after that, the DMAC gets the bus mastership from other bus master after waiting for 16, 64, or 256 clocks in Bϕ count. The DMAC then transfers data of one unit and returns the bus mastership to other bus master. These operations are repeated until the transfer end condition is satisfied. It is thus possible to make lower the ratio of bus occupation by DMA transfer than cycle-steal normal mode.

When the DMAC gets again the bus mastership, DMA transfer can be postponed in case of entry updating due to cache miss.

This intermittent mode can be used for all transfer section; transfer request source, transfer source, and transfer destination. The bus modes, however, must be cycle steal mode in all channels.

Figure 12.7 shows an example of DMA transfer timing in cycle steal intermittent mode. Transfer conditions shown in the figure are:

— Dual address mode

— DREQ low level detection

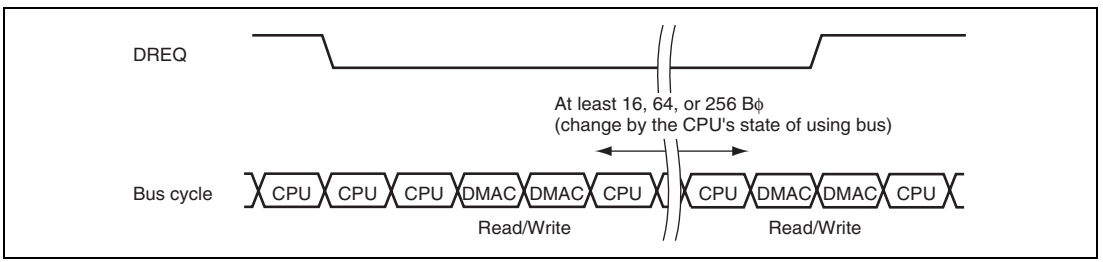


Figure 12.7 Example of DMA Transfer in Cycle Steal Intermittent Mode (Dual Address, DREQ Low Level Detection)

(b) Burst Mode

In burst mode, once the DMAC obtains the bus mastership, the transfer is performed continuously without releasing the bus mastership until the transfer end condition is satisfied. In external request mode with level detection of the DREQ pin, however, when the DREQ pin is not active, the bus mastership passes to the other bus master after the DMAC transfer request that has already been accepted ends, even if the transfer end conditions have not been satisfied.

Burst mode cannot be used when the on-chip peripheral module is the transfer request source.

Figure 12.8 shows DMA transfer timing in burst mode.

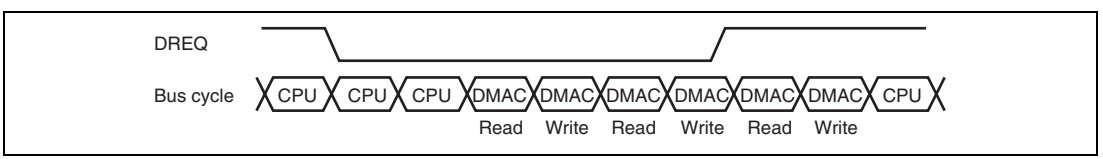


Figure 12.8 DMA Transfer Example in Burst Mode (Dual Address, DREQ Low Level Detection)

(3) Relationship between Request Modes and Bus Modes by DMA Transfer Category

Table 12.9 shows the relationship between request modes and bus modes by DMA transfer category.

Table 12.9 Relationship between Request Modes and Bus Modes by DMA Transfer Category

Address Mode	Transfer Category	Request Mode	Bus Mode	Transfer Size (Bits)	Usable Channels
Dual	External device with DACK and external memory	External	B/C	1/2/4/8/16/32	0, 1
	External device with DACK and memory-mapped external device	External	B/C	1/2/4/8/16/32	0, 1
	External memory and external memory	External, auto	B/C	1/2/4/8/16/32	0 to 5*[3]
	External memory and memory-mapped external device	External, auto	B/C	1/2/4/8/16/32	0 to 5*[3]
	Memory-mapped external device and memory-mapped external device	External, auto	B/C	1/2/4/8/16/32	0 to 5*[3]
	External memory and on-chip peripheral module	All*[1]	C	1/2/4/8/16*[2]	0 to 5*[3]
	Memory-mapped external device and on-chip peripheral module	All*[1]	C	1/2/4/8/16*[2]	0 to 5*[3]
	On-chip peripheral module and on-chip peripheral module	All*[1]	C	1/2/4/8/16*[2]	0 to 5*[3]
	IL memory and memory-mapped external device	External, auto	B/C	1/2/4/8/16/32	0 to 5*[3]
	IL memory and on-chip peripheral module	All*[1]	B/C	1/2/4/8/16*[2]	0 to 5*[3]
	IL memory and external memory	External, auto	B/C	1/2/4/8/16/32	0 to 5*[3]

[Legend]

B: Burst mode

C: Cycle steal mode

Notes: 1. External requests, auto requests, and on-chip peripheral module requests are all available. However the request source register must be designated as the transfer source or the transfer destination.

2. Access size permitted for the on-chip peripheral module register functioning as the transfer source or transfer destination.

3. If the transfer request is an external request, channels 0 and 1 are only available.

(4) Bus Mode and Channel Priority

When the priority is set in fixed mode (CH0 > CH1) and channel 1 is transferring in burst mode, if there is a transfer request to channel 0 with a higher priority, the transfer of channel 0 will begin immediately.

At this time, if channel 0 is also operating in burst mode, the channel 1 transfer will continue after the channel 0 transfer has completely finished.

When channel 0 is in cycle steal mode, channel 0 with a higher priority performs the transfer of one transfer unit and the channel 1 transfer is continuously performed without releasing the bus mastership. The bus mastership will then switch between the two in the order channel 0, channel 1, channel 0, and channel 1. Therefore, the bus state is such that the CPU cycle after the completion of cycle steal mode transfer has been replaced with the channel 1 burst mode transfer. (Hereinafter referred to as burst mode priority execution.)

This example is shown in figure 12.9. When multiple channels are operating in burst modes, the channel with the highest priority is executed first.

When DMA transfer is executed in the multiple channels, the bus mastership will not be given to the bus master until all competing burst transfers are complete.

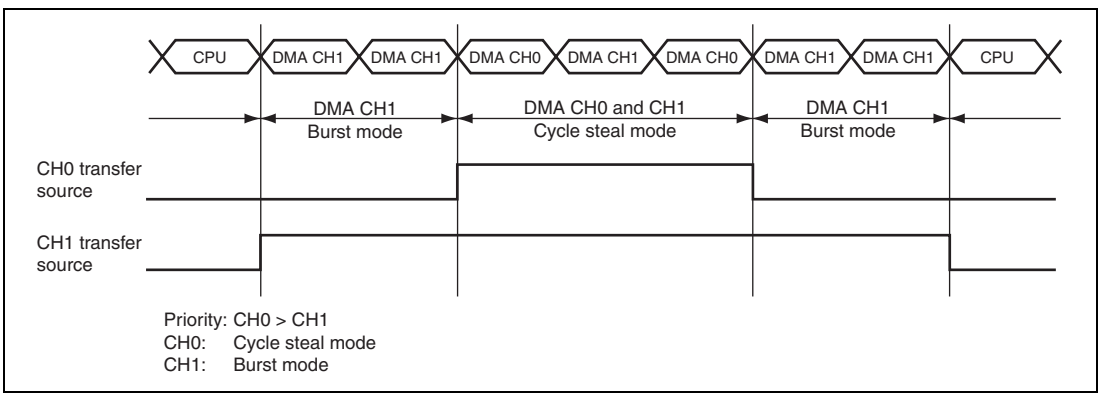


Figure 12.9 Bus State when Multiple Channels are Operating

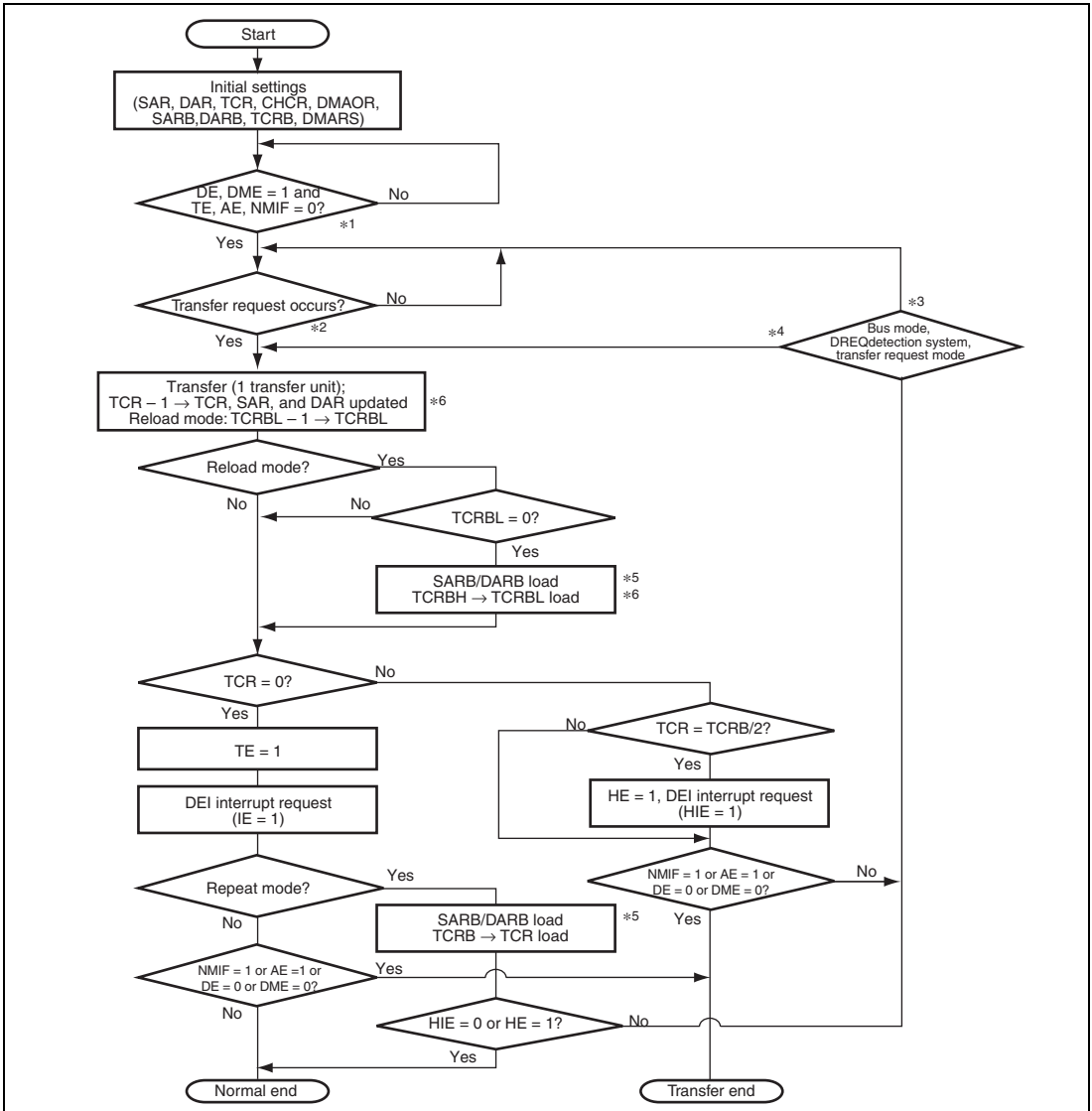
In round-robin mode, the priority changes according to the specification shown in figure 12.2. However, the channel in cycle steal mode cannot be mixed with the channel in burst mode.

12.4.4 DMA Transfer Flow

After the DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count registers (DMATCR), DMA channel control registers (CHCR), DMA operation register (DMAOR), and DMA extended resource selectors (DMARS) are set, the DMAC transfers data according to the following procedure:

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0)
2. When a transfer request occurs while transfer is enabled, the DMAC transfers one transfer unit of data (specified by TS[3:0] setting). In auto request mode, the transfer begins automatically when the DE bit and DME bit are set to 1. The DMATCR value will be decremented for each transfer. The actual transfer flows vary by address mode and bus mode.
3. When the specified number of transfer has been completed (when DMATCR reaches 0), the transfer ends normally. If the IE bit in CHCR is set to 1 at this time, a DEI interrupt is sent to the CPU.
4. When an address error or an NMI interrupt is generated, the transfer is aborted. Transfers are also aborted when the DE bit in CHCR or the DME bit in DMAOR is changed to 0.

Figure 12.10 shows a flowchart of this procedure.



Notes:
1. In repeat mode, a transfer request is acceptted with TE =1 when HIE = 1 and HE = 0.
2. In auto-request mode, transfer starts when bits NMIF, AE, and TE are all 0 or bits TE and HIE are 1 and HE is 0 (in repeat mode), and bits DE and DME are set to 1.
3. DREQ is level detection (external requesrt) in burst mode or cycle-steral mode.
4. DREQ is edge detection (external request) or auto request in burst mode.
5. Loading to SAR and DAR differs according to the operating conditions in each mode.
6. TCRBH represents bits TCRB[13:16] and TCRBL represents bits TCRB [15 to 0].

Figure 12.10 DMA Transfer Flowchart

12.4.5 Repeat Mode Transfer

In a repeat mode transfer, a DMA transfer is repeated without specifying the transfer settings every time before executing a transfer.

Using a repeat mode transfer with the half end function allows a double buffer transfer executed virtually. Following processings can be executed effectively by using a repeat mode transfer. As an example, operation of receiving voice data from the VOICE CODEC and compressing it is explained.

In the following example, processing of compressing 40-word voice data every data reception is explained. In this case, it is assumed that voice data is received by means of SIOF.

1. DMAC settings
- Set address of the SIOF receive data register in SAR
- Set address of an internal memory data store area in DAR
- Set TCR to 80 (H'50)
- Satisfy the following settings of CHCR

  Bits RPT[2:0] = B'010: Repeat mode (use DAR as a repeat area)

  Bit HIE = B'1: TCR/2 interrupt generated

  Bits DM[1:0] = B'01: DAR incremented

  Bits SM[1:0] = B'00: SAR fixed

  Bit IE = B'1: Interrupt enabled

  Bit DE = B'1: DMA transfer enabled
- Set such as bits TB and TS[3:0] according to use conditions
- Set bits CMS[1:0] and PR[1:0] in DMAOR according to use conditions and set the DME bit to B'1

2. Voice data is received and then transferred by SIOF/DMAC
3. TCR is decreased to half of its initial value and an interrupt is generated

   Read CHCR to confirm that the HE bit is set to 1 by an interrupt processing, and compress 40-word voice data from the address set in DAR.
4. TCR is cleared to 0 and an interrupt is generated

   Read CHCR to confirm that the TE bit is set to 1 by an interrupt processing, and compress 40-word voice data from the address set in DAR + 40. After this operation, the value of DARB is copied to DAR in DMAC and initialized, and the value of TCRB is copied to TCR and initialized to 80.

5. Hereafter, steps 2 and 3 are repeated until DME or DE is set to B'0, or an NMI interrupt is generated.

As explained above, a repeat mode transfer enables sequential voice compression by changing buffer for storing data received consequentially and a data buffer for processing signals alternately.

12.4.6 Reload Mode Transfer

In a reload mode transfer, according to the settings of bits RPT[2:0] in CHCR, the value set in SARB/DARB is set to SAR/DAR and the value of bits TCRB[31:16] is set in bits TCRB[15:0] at each transfer set in the bits TCRB[15:0], and the transfer is repeated until TCR becomes 0 without specifying the transfer settings again. A reload mode transfer is effective when repeating data transfer with specific area. Figure 12.11 shows the operation of reload mode transfer.

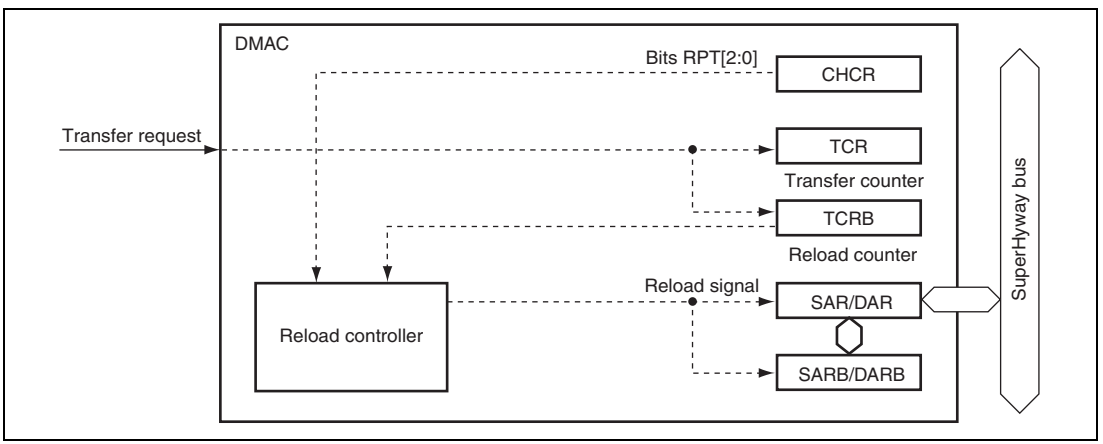


Figure 12.11 Reload Mode Transfer

When a reload mode transfer is executed, TCRB is used as a reload counter. Set TCRB according to section 12.3.6, DMA Transfer Count Registers (TCRB_0 to TCRB_3).

12.4.7 DREQ Pin Sampling Timing

Figures 12.12 to 12.15 show the sample timing of the DREQ input in each bus mode, respectively.

DREQ detection (edge or level detection) sampling takes place at the rising edge of CKO for both assertion and negation.

This means that assertion and negation are possible once every clock cycle, provided the necessary DREQ setup/hold time is guaranteed.

However, there is a non-sensitive period in the case of assertion, and requests are not accepted during this period.

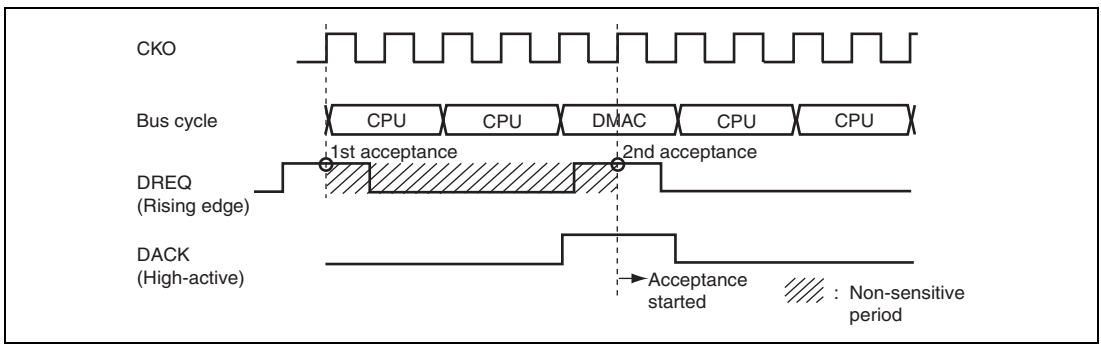


Figure 12.12 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection

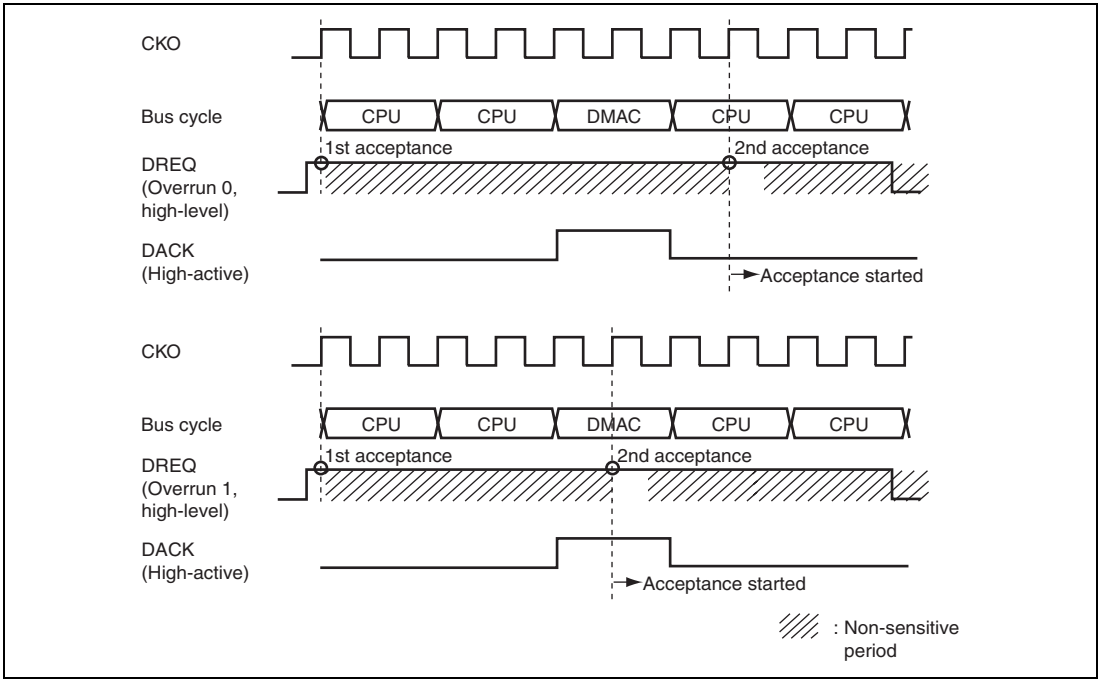


Figure 12.13 Example of DREQ Input Detection in Cycle Steal Mode Level Detection

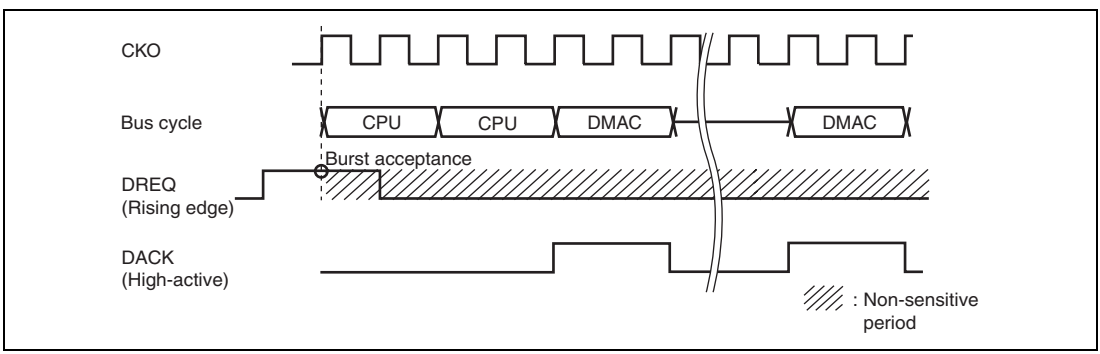


Figure 12.14 Example of DREQ Input Detection in Burst Mode Edge Detection

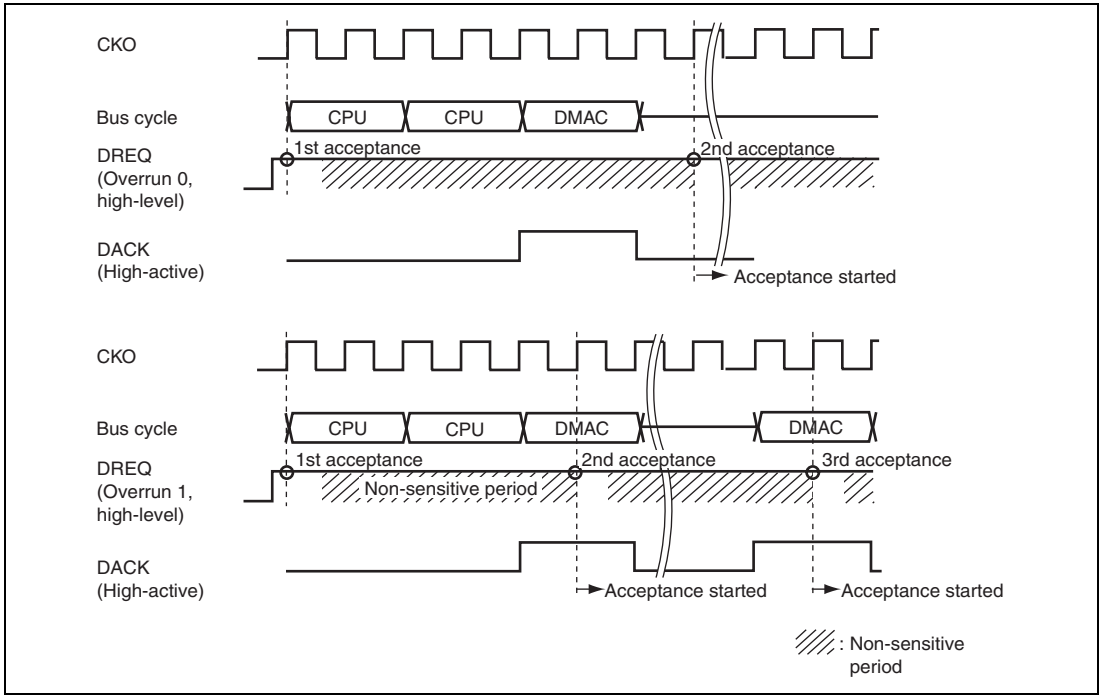


Figure 12.15 Example of DREQ Input Detection in Burst Mode Level Detection

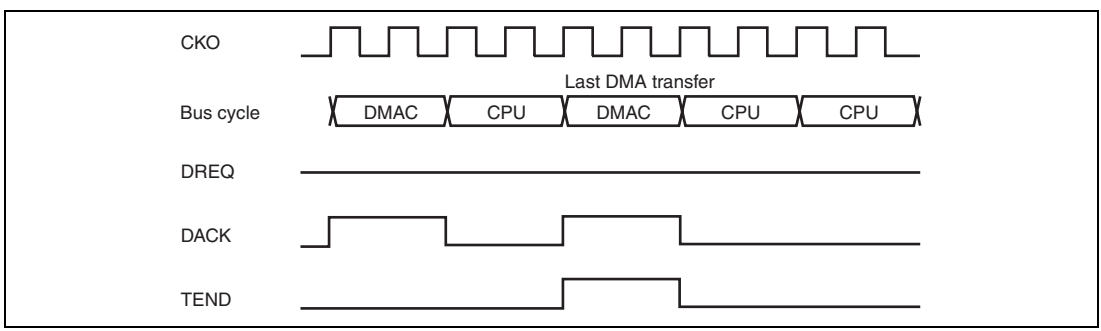
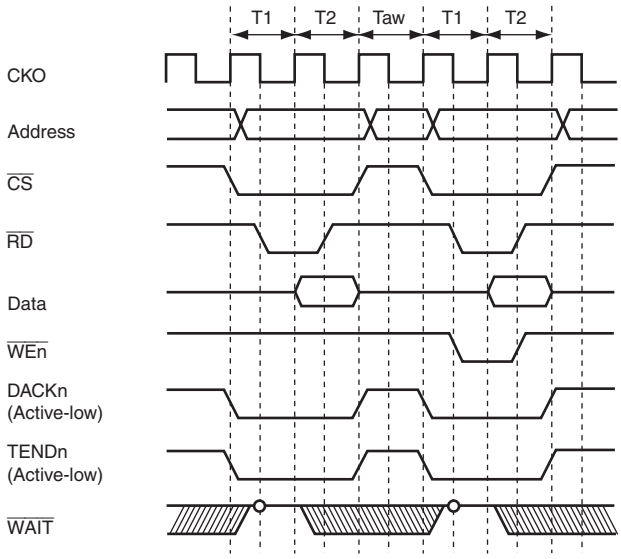


Figure 12.16 DMA Transfer End Signal Timing (Level Detection in Cycle Steal Mode)

When an 8-bit or 16-bit external device is accessed in longword units, or when an 8-bit external device is accessed in word units, the DACK and TEND outputs are divided for data alignment. This example is shown in figure 12.17.



Note: TEND is asserted for the last transfer unit of the DMA transfer. When the transfer unit is divided into several bus cycles and $\overline{CS}$ is negated between bus cycles, TEND is also divided.

Figure 12.17 Example of BSC Ordinary Memory Access (No Wait, Idle Cycle 1, Longword Access to 16-Bit Device)

12.5 Usage Notes

Pay attentions to the following notes when the DMAC is used.

12.5.1 DMA Transfer for Peripheral Modules

When executing a 16-byte DMA transfer for peripheral modules, set the TS[3:0] bits in CHCR to B'1011 and execute in 16-byte 2-division transfer mode. This DMA transfer can only be executed when a 16-byte boundary can be set in SAR or DAR as a transfer source address or transfer destination address. When a transfer source address or transfer destination address is not a 16-byte boundary, data cannot be transferred successfully.

12.5.2 Module Stop

While DMAC is in operation, the DMAC should not be stopped by the module stop register (MSTPCR0). When modules are stopped, transfer contents cannot be guaranteed.

12.5.3 Address Error

When a DMA address error is occurred, set registers of all channels again and then start a transfer.

12.5.4 Notes on Burst Mode Transfer

During a burst mode transfer, following operation should not be executed until the transfer of corresponding channel has completed.

1. Frequency should not be changed.
2. Transition to sleep mode should not be made.
3. Transition to standby mode should not be made.

12.5.5 Notes on Setting of DMA Extended Resource Selectors

Do not set the same transfer requests to DMA extended resource selectors (DMA0_DMARS0 to DMA0_DMARS2, DMA1_DMARS0 to DMA1_DMARS2). When the same transfer requests are set, transfer contents cannot be guaranteed.

12.5.6 Notice on Use of the DMAC Flag Bits

The HE, TE, AE and NMIF flags have the following note.

As for the half end flag and transfer end flag in channel control register (CHCRm.HE and CHCRn.TE, m= 0 to 3, n= 0 to 5) and the address error flag and NMI flag in DMA operation register (DMAOR.AE and DMAOR.NMIF), when reading these flags while they are just setting to 1, the read out value of the corresponding flag is 0, but the internal state of this operation may become same as read out 1. If writing 0 to the corresponding flag after this case, it is equivalent to write 0 after reading the corresponding flag is 1, as a result, the corresponding flag is cleared to 0 unintentionally.

When using corresponding flag, not to clear the flag unintentionally, it is necessary to read and write by the following procedure. When writing register that has the corresponding flags, write 1 to the flag bit except clearing the flag explicitly. Clearing the corresponding flag explicitly, write 0 to the flag bit after reading out 1. Writing 1 to the corresponding bit does not affect the value of the flag.

Note that, when not using corresponding flag, it is no problem to write always 0 (Clearing the corresponding flag explicitly, write 0 to the flag bit after reading out 1).

12.5.7 Restarting a DMA Transfer After Interruption

Follow the procedure below to restart an interrupted DMA transfer in external request mode or on-chip peripheral module request mode.

(1) External request mode

When a DMA transfer is interrupted by an address error, NMI assertion, clearing to 0 of DE in CHCRn (n=0 to 5), or clearing to 0 of DME in DMAOR0, start the DMA transfer after resetting the registers of all the channels.

(2) On-chip peripheral module request mode

(a) Address error or NMI assertion

When a DMA transfer is interrupted by an address error or NMI assertion, after clearing the transfer requests from the peripheral modules of all the channels clear the DME bit to 0 once to disable DMA transfers. Then, after resetting the registers of all the channels, start the DMA transfer.

(b) Transfer interruption by clearing DE bit to 0

Either of the following methods may be used as workarounds.

- After the interruption, clear transfer requests from the peripheral modules of the channel concerned and start the transfer after resetting the registers of the channel concerned.
- After the interruption, clear the DE bit when the transfer requests from the peripheral modules of all channels have been cleared.

(c) Transfer interruption by clearing DME bit to 0

Either of the following methods may be used as workarounds.

- After the interruption, clear transfer requests from the peripheral modules of all channels and start the DMA transfer after resetting the registers of all the channels.
- After the interruption, clear the DME bit to 0 when the transfer requests from the peripheral modules of all channels have been cleared.

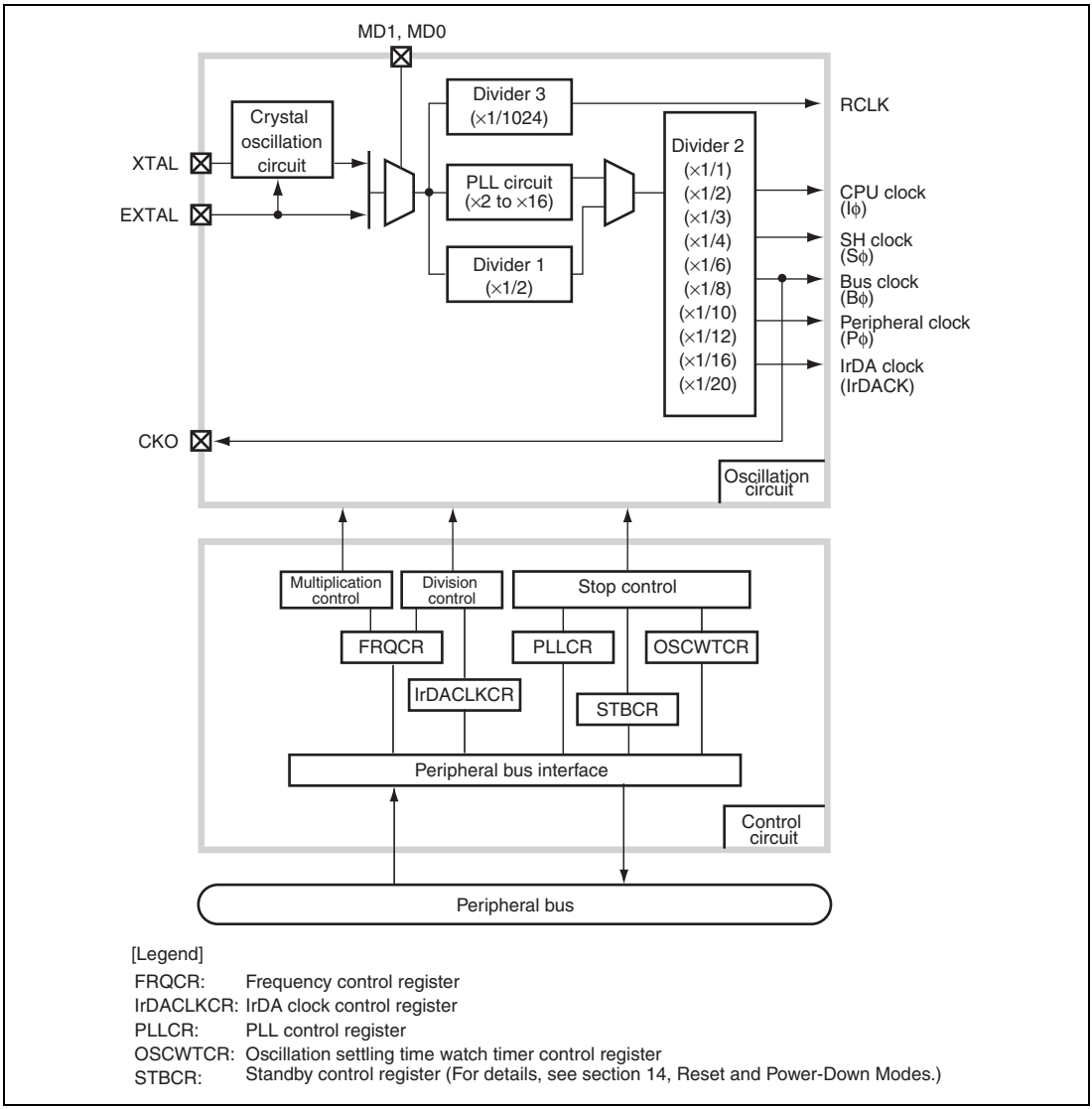
Section 13 Clock Pulse Generator (CPG)

The clock pulse generator (CPG) consists of an oscillator, a PLL circuit, a frequency divider, and related control circuits, and generates various clock signals for use with this LSI circuit.

13.1 Features

- Generation of various clock signals for LSI-internal operations
  CPU clock (Iϕ): Clock for the CPU, FPU, cache, TLB, etc.
  SH clock (Sϕ): Clock for the SuperHyway bus
  Bus clock (Bϕ): Clock for the BSC, DMAC, etc.
  Peripheral clock (Pϕ): Clock for peripheral modules
  RCLK clock (RCLK): Clock for the RWDT
- Generation of various clock signals for external interfaces
  Bus clock (CKO): Clock for the BSC bus interface (same as Bϕ)
  IrDA clock (IrDACK): Clock for the IrDA interface
- Clock modes
  The combination of the division ratios for the CPU clock, SH clock, bus clock, and peripheral clock after a power-on reset can be selected from three clock modes.
- Frequency change function
  The frequency of the CPU clock, SH clock, bus clock, SDRAM clock, and peripheral clock can be changed independently using the PLL circuit and dividers within the CPG. Frequencies are changed by software using the frequency control register (FRQCR) settings.
- Power-down mode control
  The clock can be stopped in sleep mode and software standby mode, and specific modules can be stopped using the module standby function. See section 14, Reset and Power-Down Modes, for details.

A block diagram of the CPG is shown in figure 13.1.



[Legend]

FRQCR: Frequency control register
IrDACLKCR: IrDA clock control register
PLLCR: PLL control register
OSCWTCR: Oscillation settling time watch timer control register
STBCR: Standby control register (For details, see section 14, Reset and Power-Down Modes.)

Figure 13.1 Block Diagram of CPG

The CPG blocks function as follows:

(1) PLL Circuit

The PLL circuit multiples the clock frequency input from the EXTAL pin by the ratio of ×2 to ×16. The multiplication ratio is set by the frequency control register (FRQCR). Turning on and off the PLL circuit is set by the PLL control register (PLLCR).

(2) Divider 1

Divider 1 divides the clock frequency input from the EXTAL pin by two. The clock signal from divider 1 is input to divider 2 when the PLL circuit is off.

(3) Divider 2

Divider 2 receives the clock signal from divider 1 or the PLL circuit as an input and generates the CPU clock, SH clock, bus clock, and peripheral clock. The division ratios are set by the respective frequency control registers.

(4) Divider 3

Divider 3 divides the clock frequency input from the EXTAL pin by 1024.

(5) Control Circuit

The control circuit controls the clock frequency and sets the power-down modes according to the settings of the MD0 and MD1 pins and the frequency control registers.

13.2 Input/Output Pins

Table 13.1 lists the CPG pin configuration.

Table 13.1 Pin Configuration and Functions of CPG

Pin Name	Function	I/O	Description
MD0	Mode control pins	Input	Sets the clock operating mode.
MD1		Input	Sets the clock operating mode.
XTAL	Crystal resonator connection pin	Output	Connects the crystal resonator.
EXTAL	Crystal resonator connection pin / external clock input pin	Input	For connection of the crystal resonator; also, used as an external clock input pin.
CKO	Clock output pin	Output	Used as an external clock output pin.

13.3 Clock Operating Modes

Table 13.2 shows the relationship between the mode control pin (MD1 and MD0) combinations and the initial clock settings after a power-on reset.

Table 13.2 Clock Operating Modes

Clock Mode	Pin Setting		Register Initial Value		Clock Source	PLL (Multiplication Ratio)	Initial Clock Ratio			
	MD1	MD0	FRQCR	PLLCR			Iϕ	Sϕ	Bϕ	Pϕ
0	0	0	H'0755 5558	H'0000 4000	EXTAL*[1]	ON (×8)	2	2	2	1
1	0	1	H'0700 0000	H'0000 0000	EXTAL*[1]	OFF	1/2	1/2	1/2	1/2
2	1	0	H'0755 5558	H'0000 4000	Crystal oscillator*[2]	ON (×8)	2	2	2	1
3	1	1	Setting prohibited							

Notes: 1. The external clock is input on the EXTAL pin.
2. Connect the crystal oscillator to the EXTAL and XTAL pins.

13.4 Register Descriptions

Table 13.3 shows the CPG register configuration. Table 13.4 shows the register states in each operating mode.

Table 13.3 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
Frequency control register	FRQCR	R/W	H'A415 0000	32
PLL control register	PLLCR	R/W	H'A415 0024	32
IrDA clock	IrDACLKCR	R/W	H'A415 0018	32
Oscillation settling time watch timer control register	OSCWTCR	R/W	H'A415 0044	32

Table 13.4 Register States in Each Operating Mode

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
FRQCR	Initialized	Retained	—	Retained
PLLCR	Initialized	Retained	—	Retained
IrDACLKCR	Initialized	Retained	—	Retained
OSCWTCR	Initialized	Retained	—	Retained

13.4.1 Frequency Control Register (FRQCR)

FRQCR is a 32-bit readable/writable register used to specify the frequency multiplication ratio of the PLL circuit, and the frequency division ratio of the CPU clock, SH clock, bus clock, and peripheral clock. FRQCR can be accessed only in longwords.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	HIGH[1:0]		—	STC[4:0]					IFC[3:0]				—	—	—	—
Initial value:	0	0	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1
R/W:	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SFC[3:0]				BFC[3:0]				—	—	—	—	PFC[3:0]			
Initial value:	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1	—*1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	HIGH[1:0]	00	R/W	VCO selection for PLL circuit Set according to the output frequency of the PLL circuit. 00: PLL circuit operates at high speed (PLL circuit multiplication output is 150 MHz or more) 01: Setting prohibited 10: Setting prohibited 11: PLL circuit operates at low speed (PLL circuit multiplication output is 150 MHz or less)
29	—	Undefined*1	R	Reserved The write value should always be 0.
28 to 24	STC[4:0]	Undefined*1	R/W	PLL Circuit Multiplication Ratio Multiplication is by (setting + 1). 00001: ×2 00010: ×3 00011: ×4 00101: ×6 00111: ×8 01111: ×16 Other settings are prohibited

Bit	Bit Name	Initial Value	R/W	Description	
23 to 20	IFC[3:0]	Undefined*1	R/W	CPU Clock (Iϕ) Frequency Division Ratio*2	
				0000: ×1/1	1001: ×1/10
				0010: ×1/2	1010: ×1/12
				0101: ×1/4	1011: ×1/16
				0111: ×1/6	1100: ×1/20
				1000: ×1/8	Other settings are prohibited
19 to 16	—	Undefined*1	R	Reserved	
				The write value should always be 0.	
15 to 12	SFC[3:0]	Undefined*1	R/W	SH Clock (Sϕ) Frequency Division Ratio*2	
				0000: ×1/1	1001: ×1/10
				0010: ×1/2	1010: ×1/12
				0100: ×1/3	1011: ×1/16
				0101: ×1/4	1100: ×1/20
				0111: ×1/6	Other settings are prohibited
				1000: ×1/8	
11 to 8	BFC[3:0]	Undefined*1	R/W	Bus Clock (Bϕ) Frequency Division Ratio*2	
				0000: ×1/1	1001: ×1/10
				0010: ×1/2	1010: ×1/12
				0100: ×1/3	1011: ×1/16
				0101: ×1/4	1100: ×1/20
				0111: ×1/6	Other settings are prohibited
				1000: ×1/8	
7 to 4	—	Undefined*1	R	Reserved	
				The write value should always be 0.	

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	PFC[3:0]	Undefined*[1]	R/W	Peripheral Clock (Pϕ) Frequency Division Ratio*[2] 0000: ×1/1 1001: ×1/10 0010: ×1/2 1010: ×1/12 0100: ×1/3 1011: ×1/16 0101: ×1/4 1100: ×1/20 0111: ×1/6 Other settings are prohibited 1000: ×1/8

Notes: 1. Initial values of bits 29 to 0 depend on the clock mode. See table 13.2 for details.

2. The settings for the clock frequencies of the individual clocks must be in accord with the ratios listed below (N_1 to N_3: 1, 2, 3, …).

   $B\phi : P\phi = N_1 : 1$

   $S\phi : B\phi = N_2 : 1$

   $I\phi : S\phi = N_3 : 1$

   See table 33.6 of section 33, Electrical Characteristics, regarding the frequency ranges of each of the clock signals.

13.4.2 PLL Control Register (PLLCR)

PLLCR is a 32-bit readable/writable register used to turn on or off the PLL circuit. PLLCR can be accessed only in longwords.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PLL1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	—*	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14	PLL1E	Undefined*	R/W	PLL Enable Turns the PLL circuit on or off. 0: PLL circuit is off 1: PLL circuit is on
13 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * Initial value of bit 14 depends on the clock mode. See table 13.2 for details.

13.4.3 IrDA Clock Control Register (IrDACLKCR)

IrDACLKCR is a 32-bit readable/writable register that controls the IrDA clock. IrDACLKCR can be accessed only in longwords.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	EXSRC	—	—	—	DIV[3:0]			
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	EXSRC	1	R/W	Clock Source Select Selects the IrDA clock source. 0: PLL circuit output clock 1: Clock is halted
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	DIV[3:0]	0000	R/W	Division Ratio These bits set the frequency division ratio of the IrDA clock. 0000: × 1/1 0010: × 1/2 0100: × 1/3 0101: × 1/4 0111: × 1/6 1000: × 1/8 1001: × 1/10 1010: × 1/12 1011: × 1/16 1100: × 1/20 Other settings are prohibited

13.4.4 Oscillation Settling Time Watch Timer Control Register (OSCWTCR)

OSCWTCR is a readable/writable register that controls the crystal resonator oscillation settling time watch timer. OSCWTCR can be accessed only in longwords.

The oscillation settling time watch timer control register has control bits for securing the oscillation settling time for the crystal oscillator on exit from software standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNT[7:0]								—	—	TIME	EXOEN	—	CKS[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 8	CNT[7:0]	00000000	R/W	Watch Timer Counter Set the initial value of the oscillation settling time watch timer counter.
7, 6	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	TIME	0	R/W	Clock Settling Time Ensuring Bit Specifies whether to ensure the EXTAL clock settling time on exit from software standby mode when this bit is set to 1. 0: Clock settling time is not ensured 1: Clock settling time is ensured

Bit	Bit Name	Initial Value	R/W	Description
4	EXOEN	0	R/W	EXTAL Stop Clock supply from EXTAL stops and the on-chip crystal oscillator also stops during the standby mode when this bit is set to 1. 0: Does not stop the clock supply from EXTAL nor the on-chip crystal oscillator during the standby mode 1: Stops the clock supply from EXTAL or the on-chip crystal oscillator during the standby mode
3	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	CKS[2:0]	000	R/W	Clock Select Selects the clock for counting by the clock oscillation settling time watch timer among 8 clocks generated by dividing the clock from EXTAL. The overflow cycles in parentheses are the values when the clock from EXTAL is 15 MHz. 000: Clock from EXTAL (17 μs) 001: Clock from EXTAL/4 (68 μs) 010: Clock from EXTAL/16 (273 μs) 011: Clock from EXTAL/32 (546 μs) 100: Clock from EXTAL/64 (1.09 ms) 101: Clock from EXTAL/256 (4.36 ms) 110: Clock from EXTAL/1024 (17.48 ms) 111: Clock from EXTA/4096 (69.91 ms)

13.5 Changing Frequency

The clock controlled by the frequency control register can be changed either by changing the multiplication ratio of the PLL circuit or by changing the division ratio of the divider. All of these are controlled by software through the frequency control register. The methods are described below.

13.5.1 Changing the Multiplication Ratio of PLL Circuit

The multiplication ratio of the PLL circuit can be changed by simply overwriting the STC[4:0] bits in FRQCR. Changing the multiplication ratio suspends CPU operations. After suspension of CPU operations, clock signals output on the CKO pin are unstable until oscillation of the PLL has become stable. Passage of the PLL oscillation settling time is automatically detected within the chip, and CPU operations are restarted once this time has elapsed, indicating that PLL oscillation has become stable.

13.5.2 Changing the Division Ratio

The division ratio can be changed by overwriting each set of bits for setting the division ratio in FRQCR.

The clock changes to the new setting immediately when the contents of FRQCR are changed.

13.5.3 Changing Clock Operating Mode

The values of the mode control pins (MD1 and MD0) that define the clock operating mode are reflected at a power-on reset. Do not change the MD1 and MD0 pin settings during operation.

13.5.4 Turning On/Off of PLL Circuit

The PLL circuit can be turned on or off by rewriting the PLL1E bit in PLLCR.

Similar to when changing the multiplication ratio of the PLL circuit, the oscillation settling time of the PLL circuit is internally detected automatically.

13.6 Procedure for Ensuring the Internal Oscillator Settling Time on Exit from Software Standby Mode

When the clock source is the crystal resonator and this is stopped in software standby mode, OSCWTCR controls the crystal oscillation settling time when an NMI interrupt triggers exit from software standby mode. The procedure is as follows.

1. Set the TIME bit and EXOEN bit of OSCWTCR to 1 before transition to software standby mode.
2. In OSCWTCR, set the clock to be used in the CKS[2:0] bit field and the initial value of the counter in the CNT [7:0] bit field. The values must be such that the counter takes longer to overflow than the clock oscillation settling time.
3. After the STBY bit of STBCR has been set to 1, issuing a SLEEP instruction initiates entry to software standby mode and stops the clock.
4. The watch timer counter (bits CNT[7:0] of OSCWTCR) starts counting on detection of an edge of the NMI signal. When the CNT[7:0] counter overflows, the CPG starts the clock supply and operation of this LSI restarts. At this point, the CNT[7:0] value stops at the set initial value.

13.7 Notes on Board Design

(1) Bypass Capacitor

Insert about 0.1 to 1.0 μF of laminated ceramic capacitors as bypass capacitors for each V_{SS}/V_{CC} pair. Table 13.5 shows the pair of power supply pins. Mount the bypass capacitor near the power supply pins of the LSI. Use components with a frequency characteristic suitable for the operating frequency of the LSI, as well as a suitable capacitance value.

Table 13.5 Pairs of Power Supply Pins

Paired Power Supply Name	Paired Power Supply Pin No.
AVcc – AVss	205 - 208
Vcc – Vss	29 - 27, 81 - 79, 134 - 132, 154 - 152, 175 - 173
Vcc_PLL1 – VSS_PLL1	145 - 147
Vcc_PLL2 – VSS_PLL2	150 - 148
VccQ - VssQ	3 - 6, 21 - 19, 35 - 33, 47 - 45, 59 - 57, 71 - 69, 85 - 83, 97 - 95, 111 - 109, 163 - 161, 183 - 181, 183 - 198

(2) When Using a PLL Oscillator Circuit

Keep the wiring from the PLL V_{CC} and V_{SS} connection pattern to the power supply pins short, and make the pattern width large, to minimize the inductance component.

The analog power supply system of the PLL circuit is sensitive to noise. Therefore system malfunction may occur by the intervention with another power supply. Do not supply the analog power supply with the same resource as the digital power supply of V_{DD} and $V_{CC}Q$.

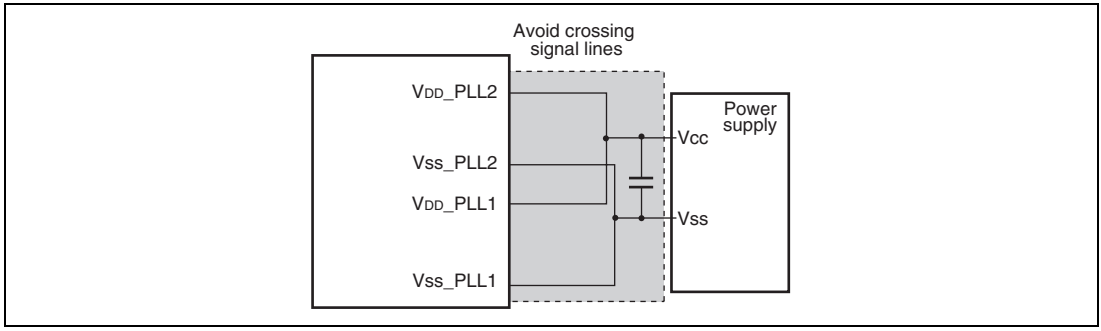


Figure 13.2 Points to Note in Use of the PLL Oscillator Circuit

Section 14 Reset and Power-Down Modes

This LSI monitors the power supply to the LSI, and has the power management function to control the power supply.

This LSI also supports sleep mode, software standby mode, and module standby mode, in which clock supply to the LSI is controlled optimally to save power.

14.1 Features

Supports sleep mode, software standby mode, and module standby state, in which clock supply to the unnecessary operation module or entire LSI is stopped.

14.1.1 Power-Down Modes

This LSI has the following power-down modes and function:

1. Sleep mode: Supply of the clock signal to the CPU is halted.
2. Software standby mode: Supply of clock signals to the CPU and on-chip peripheral modules, etc. is halted.
3. Module standby function: Software specifies halting of the supply of clock signals to specific modules that are not in use.

Table 14.1 shows the transition conditions for entering the modes from the program execution state, as well as the CPU and peripheral module states in each mode and the procedures for canceling each mode.

Table 14.1 States of Power-Down Modes

		State						
Power-Down Mode	**Transition Conditions**	**CPG**	**CPU**	**CPU Register**	**On-Chip Peripheral Modules**	**IL Memory**	**External Memory**	**Canceling Procedure**
Sleep mode	Execute SLEEP instruction with STBY bits cleared to 0 in STBCR	Runs	Halts	Held	Runs	Runs	Auto-refreshing	• Interrupt • Power-on reset
Software standby mode	Execute SLEEP instruction with STBY bit set to 1 in STBCR	Runs/ Halts	Halts	Held	Halts*	Held	Self-refreshing	• NMI, IRQ, PINT, RTC interrupt • Power-on reset
Module standby mode	Set MSTP bit of respective module to 1 in MSTPCR	Runs	Runs/ Halts	Held	Specified module halts	Runs	Auto-refreshing	• Clear MSTP bit to 0

Note: * The RWDT, which is driven by RCLK, stays running.

14.2 Input/Output Pins

Table 14.2 lists the pin configuration related to power-down modes.

Table 14.2 Pin Configuration

Pin Name	Function	I/O	Description
STATUS0	Processing state 0	Output	Becomes high level in software standby mode.
$\overline{\text{RESETP}}$	Reset input pin	Input	This LSI enters the power-on reset state when this pin becomes low level.
$\overline{\text{RESETOUT}}$	Power-on reset output signal	Output	Becomes low level while this LSI is being power-on reset.

14.3 Register Descriptions

Table 14.3 shows the register configuration for power-down modes. Table 14.4 shows the register states in each operating mode.

Table 14.3 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
Standby control register	STBCR	R/W	H'A415 0020	32
Module stop register 0	MSTPCR0	R/W	H'A415 0030	32
Module stop register 1	MSTPCR1	R/W	H'A415 0034	32
Module stop register 2	MSTPCR2	R/W	H'A415 0038	32

Table 14.4 Register States in Each Operating Mode

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
STBCR	Initialized	Retained	—	Retained
MSTPCR0	Initialized	Retained	—	Retained
MSTPCR1	Initialized	Retained	—	Retained
MSTPCR2	Initialized	Retained	—	Retained

14.3.1 Standby Control Register (STBCR)

STBCR is a 32-bit readable/writable register that can select sleep mode and standby mode.

STBCR can be accessed only in longwords.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	STBY	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	STBY	0	R/W	Standby Executing the SLEEP instruction after this bit is set to 1 makes a transition to standby mode.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

14.3.2 Module Stop Register 0 (MSTPCR0)

MSTPCR0 is a 32-bit readable/writable register that can individually start or stop the module assigned to each bit.

MSTPCR0 can be accessed only in longwords.

After cancelling module stop mode for the instruction cache (IC), operand cache (OC), TLB, or IL memory, either of the following preprocessing must be performed before accessing these modules. Note that such module access includes instruction fetch from a relevant module and instruction fetch using a relevant module.

- After reading the changed MSTPn bit once, execute the RTE instruction.
- After reading the changed MSTPn bit once, execute the ICBI instruction for any address. The address can be in a non-cacheable area.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MSTP031	MSTP030	MSTP029	—	MSTP027	—	—	MSTP024	—	MSTP022	MSTP021	—	MSTP019	MSTP018	MSTP017	MSTP016
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R/W	R	R	R/W	R	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSTP015	MSTP014	MSTP013	—	—	—	MSTP009	MSTP008	MSTP007	MSTP006	MSTP005	MSTP004	—	MSTP002	—	—
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	MSTP031	0	R/W	Module Stop Bit 031 Setting this bit to 1 halts supply of the clock signal to the TLB. 0: TLB operates 1: Clock supply to TLB halted
30	MSTP030	0	R/W	Module Stop Bit 030 Setting this bit to 1 halts supply of the clock signal to the instruction cache (IC). 0: IC operates 1: Clock supply to IC halted

Bit	Bit Name	Initial Value	R/W	Description
29	MSTP029	0	R/W	Module Stop Bit 029 Setting this bit to 1 halts supply of the clock signal to the operand cache (OC). 0: OC operates 1: Clock supply to OC halted
28	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
27	MSTP027	0	R/W	Module Stop Bit 027 Setting this bit to 1 halts supply of the clock signal to the IL memory. 0: IL memory operates 1: Clock supply to IL memory halted
26, 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	MSTP024	0	R/W	Module Stop Bit 024 Setting this bit to 1 halts supply of the clock signal to the FPU. 0: FPU operates 1: Clock supply to FPU halted
23	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
22	MSTP022	0	R/W	Module Stop Bit 022 Setting this bit to 1 halts supply of the clock signal to the INTC. 0: INTC operates 1: Clock supply to INTC halted
21	MSTP021	0	R/W	Module Stop Bit 021 Setting this bit to 1 halts supply of the clock signal to the DMAC. 0: DMAC operates 1: Clock supply to DMAC halted

Bit	Bit Name	Initial Value	R/W	Description
20	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
19	MSTP019	0	R/W	Module Stop Bit 019 Setting this bit to 1 halts supply of the clock signal to the H-UDI. 0: H-UDI operates 1: Clock supply to H-UDI halted
18	MSTP018	0	R/W	Module Stop Bit 018 Setting this bit to 1 halts supply of the clock signal to the debugging module (DBG) of the LSI. Clear this bit to 0 when using the H-UDI, UBC, or AUD. 0: DBG operates 1: Clock supply to DBG halted
17	MSTP017	0	R/W	Module Stop Bit 017 Setting this bit to 1 halts supply of the clock signal to the UBC. Clear this bit to 0 when using the H-UDI or AUD. 0: UBC operates 1: Clock supply to UBC halted
16	MSTP016	0	R/W	Module Stop Bit 016 Setting this bit to 1 halts supply of the clock signal to the debugging module (SUBC). 0: SUBC operates 1: Clock supply to SUBC halted
15	MSTP015	0	R/W	Module Stop Bit 015 Setting this bit to 1 halts supply of the clock signal to the TMU. 0: TMU operates 1: Clock supply to TMU halted

Bit	Bit Name	Initial Value	R/W	Description
14	MSTP014	0	R/W	Module Stop Bit 014 Setting this bit to 1 halts supply of the clock signal to the CMT. 0: CMT operates 1: Clock supply to CMT halted
13	MSTP013	0	R/W	Module Stop Bit 013 Setting this bit to 1 halts supply of the clock signal to the RWDT. 0: RWDT operates 1: Clock supply to RWDT halted
12	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	MSTP009	0	R/W	Module Stop Bit 009 Setting this bit to 1 halts supply of the clock signal to the SCIF4 (SCIFA). 0: SCIF4 (SCIFA) operates 1: Clock supply to SCIF4 (SCIFA) halted
8	MSTP008	0	R/W	Module Stop Bit 008 Setting this bit to 1 halts supply of the clock signal to the SCIF5 (SCIFA). 0: SCIF5 (SCIFA) operates 1: Clock supply to SCIF5 (SCIFA) halted
7	MSTP007	0	R/W	Module Stop Bit 007 Setting this bit to 1 halts supply of the clock signal to the SCIF0. 0: SCIF0 operates 1: Clock supply to SCIF0 halted

Bit	Bit Name	Initial Value	R/W	Description
6	MSTP006	0	R/W	Module Stop Bit 006 Setting this bit to 1 halts supply of the clock signal to the SCIF1. 0: SCIF1 operates 1: Clock supply to SCIF1 halted
5	MSTP005	0	R/W	Module Stop Bit 005 Setting this bit to 1 halts supply of the clock signal to the SCIF2. 0: SCIF2 operates 1: Clock supply to SCIF2 halted
4	MSTP004	0	R/W	Module Stop Bit 004 Setting this bit to 1 halts supply of the clock signal to the SCIF3. 0: SCIF3 operates 1: Clock supply to SCIF3 halted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	MSTP002	0	R/W	Module Stop Bit 002 Setting this bit to 1 halts supply of the clock signal to the SIOF. 0: SIOF operates 1: Clock supply to SIOF halted
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: When writing to a certain bit in MSTPCR0, read all values in MSTPCR0 first and rewrite the certain bit, then return the renewed values back to MSTPCR0.

14.3.3 Module Stop Register 1 (MSTPCR1)

MSTPCR1 is a 32-bit readable/writable register that can individually start or stop the module assigned to each bit.

MSTPCR1 can be accessed only in longwords.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MSTP113	—	—	—	MSTP109	MSTP108	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	MSTP113	0	R/W	Module Stop Bit 113 Setting this bit to 1 halts supply of the clock signal to the RTC. 0: RTC operates 1: Clock supply to RTC halted
12 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	MSTP109	0	R/W	Module Stop Bit 109 Setting this bit to 1 halts supply of the clock signal to the IIC0. 0: IIC0 operates 1: Clock supply to IIC0 halted
8	MSTP108	0	R/W	Module Stop Bit 108 Setting this bit to 1 halts supply of the clock signal to the IIC1. 0: IIC1 operates 1: Clock supply to IIC1 halted

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: If changes are to be restricted to specific bits when a new value is written to MSTPCR1, read the current value of MSTPCR1 out, only change the values of the specific bits, and then write the new value back to MSTPCR1.

14.3.4 Module Stop Register 2 (MSTPCR2)

MSTPCR2 is a 32-bit readable/writable register that can individually start or stop the module assigned to each bit.

MSTPCR2 can be accessed only in longwords.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	MSTP227	MSTP226	MSTP225	MSTP224	—	—	MSTP221	MSTP220	—	—	—	MSTP216
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
27	MSTP227	1	R/W	Module Stop Bit 227 Clearing this bit to 0 starts supply of the clock signal to the ADC. 0: ADC operates 1: Clock supply to ADC halted

Bit	Bit Name	Initial Value	R/W	Description
26	MSTP226	1	R/W	Module Stop Bit 226 Clearing this bit to 0 starts supply of the clock signal to the DAC. 0: DAC operates 1: Clock supply to DAC halted
25	MSTP225	1	R/W	Module Stop Bit 225 Clearing this bit to 0 starts supply of the clock signal to the IrDA0. 0: IrDA0 operates 1: Clock supply to IrDA0 halted
24	MSTP224	1	R/W	Module Stop Bit 224 Clearing this bit to 0 starts supply of the clock signal to the IrDA1. 0: IrDA1 operates 1: Clock supply to IrDA1 halted
23, 22	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
21	MSTP221	1	R/W	Module Stop Bit 221 Clearing this bit to 0 starts supply of the clock signal to the TPU0. 0: TPU0 operates 1: Clock supply to TPU0 halted
20	MSTP220	1	R/W	Module Stop Bit 220 Clearing this bit to 0 starts supply of the clock signal to the TPU1. 0: TPU1 operates 1: Clock supply to TPU1 halted
19 to 17	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

Bit	Bit Name	Initial Value	R/W	Description
16	MSTP216	1	R/W	Module Stop Bit 216 Clearing this bit to 0 starts supply of the clock signal to the SIM. 0: SIM operates 1: Clock supply to SIM halted
15 to 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

Note: If changes are to be restricted to specific bits when a new value is written to MSTPCR2, read the current value of MSTPCR2 out, only change the values of the specific bits, and then write the new value back to MSTPCR2.

14.4 Operation

14.4.1 Reset

(1) Power-on reset

Reset this LSI chip by using the power-on reset to recommence execution from the initial state and when power is initially supplied. The $\overline{\text{RESETP}}$ pin is used for the power-on reset. A power-on reset discontinues all processing in execution, all pending processing for events, and is immediately followed by reset processing with the $\overline{\text{RESETOUT}}$ pin driven low. The conditions for generating a power-on reset are as follows.

1. A low level is input on the $\overline{\text{RESETP}}$ pin.
2. When the RWDT starts counting and the counter overflows.

(2) H-UDI reset

When the H-UDI reset assertion command is sent to the H-UDI pins, the system enters the same state as a power-on reset. See section 31, User Debugging Interface (H-UDI) for details on the H-UDI reset.

(3) Manual reset

A manual reset is generated by software. See section 5, Exception Handling for details on the manual reset. $\overline{\text{RESETOUT}}$ pin level does not become low on manual reset.

(4) Exception for multiple hits of instruction TLB

See section 5, Exception Handling for details on the exception for multiple hits of the instruction TLB. The exception for multiple hits of the instruction TLB does not make the level on the $\overline{\text{RESETOUT}}$ pin low.

(5) Exception for multiple hits of data TLB

See section 5, Exception Handling for details on the exception for multiple hits of the data TLB. The exception for multiple hits of the data TLB does not make the level on the $\overline{\text{RESETOUT}}$ pin low.

14.4.2 Sleep Mode

(1) Transition to Sleep Mode

Executing the SLEEP instruction when the STBY bit in STBCR is 0 causes a transition from the program execution state to sleep mode. In sleep mode, supply of the clock signal to the CPU is halted. Although the CPU halts immediately after executing the SLEEP instruction, the contents of the CPU registers remain unchanged. The on-chip peripheral modules continue to operate in sleep mode and the clock continues to be output to the CKO pin.

The procedure for a transition to sleep mode is as follows:

1. Clear the STBY bit in STBCR to 0.
2. Execute the SLEEP instruction.

(2) Exit from Sleep Mode

Exit from sleep mode is driven by an interrupt (NMI, IRQ, or on-chip peripheral module) or a reset.

Interrupts are accepted in sleep mode even when the BL bit in SR is 1. If necessary, place the SPC and SSR on the stack before executing the SLEEP instruction.

(a) Exit Driven by an Interrupt

Exit from sleep mode is triggered by an NMI, IRQ, or on-chip peripheral module interrupt. After the interrupt, interrupt exception handling is executed and a code indicating the interrupt source is set in INTEVT.

(b) Exit Driven by a Reset

Exit from sleep mode is triggered by a power-on reset.

14.4.3 Software Standby Mode

(1) Transition to Software Standby Mode

Executing the SLEEP instruction when the STBY bit in STBCR is 1 causes a transition from the program execution state to software standby mode. In software standby mode, not only the CPU but also the clock and on-chip peripheral modules halt. The clock output from the CKO pin also halts. The RWDT that operates on the RCLK clock, however, continues to operate. The functions of the V_{CC} supplied (I/O) region that needs no clock (such as detection of NMI and IRQ interrupts) also continue to operate.

The contents of the CPU and cache registers remain unchanged. For the register states of the on-chip peripheral modules in software standby mode, refer to the register descriptions in each section.

The procedure for a transition to software standby mode is as follows:

1. Set the STBY bit in STBCR to 1.
2. Execute the SLEEP instruction.
3. Software standby mode is entered and the clocks within the LSI are halted. The output of the STATUS0 pin goes high.

(2) Exit from Software Standby Mode

Exit from software standby mode is driven by an interrupt (NMI, IRQ, CMT, or KEYSC), or a power-on reset.

(a) Exit Driven by an Interrupt

When using an externally input clock as the clock source, or when using the crystal resonator as the clock source and the crystal resonator does not stop oscillating in software standby, the occurrence of an NMI, IRQ (edge-detection), PINT, or RTC interrupt causes software standby mode to be canceled and the STATUS0 pin to go low. Note that in order to cancel software standby mode by means of an IRQ (level-detection) interrupt, it is necessary to supply the clock by connecting EXTAL_RTC and XTAL_RTC to the crystal resonator.

When the internal crystal oscillator is used as the clock supply source and the oscillator is stopped in software standby mode, only the NMI interrupt can trigger exit from software standby mode, in which case the STATUS0 pin goes low. Note that exit from software standby mode cannot be driven by the IRQ, PINT, or RTC interrupts when the internal crystal oscillator has been stopped.

After the interrupt, interrupt exception handling is executed and a code indicating the interrupt source is set in INTEVT. Interrupts are accepted in software standby mode even when the BL bit in SR is 1. If necessary, save the SPC and SSR on the stack before executing the SLEEP instruction.

Immediately after the SLEEP instruction, clock output via the CKO pin is halted until exit from software standby mode

When restarting the internal crystal oscillator, set OSCWTCR of the CPG to ensure the oscillation settling time.

(b) Exit Driven by a Reset

Exit from software standby mode is triggered by a power-on reset or a system reset.

14.4.4 Module Standby Mode

(1) Transition to Module Standby Mode

Setting the MSTP bits in the module stop registers to 1 halts the supply of clocks to the corresponding on-chip peripheral modules. This function can be used to reduce power consumption in normal mode.

Modules in module standby mode keep the state immediately before the transition to the module standby mode. The registers keep the contents before halted, and the external pins keep the functions before halted. At waking up from the module standby state, operation is restarted from the condition immediately before the registers and external pins have halted.

Note: Make sure to set the MSTP bit to 1 while the modules have completed the operation and are in an idle state, with no interrupt sources from the external pins or other modules.

(2) Exit from Module Standby Mode

Exit from module standby mode is triggered by clearing the respective MSTP bit to 0.

14.4.5 Mode Transitions

Figure 14.1 shows the mode transitions.

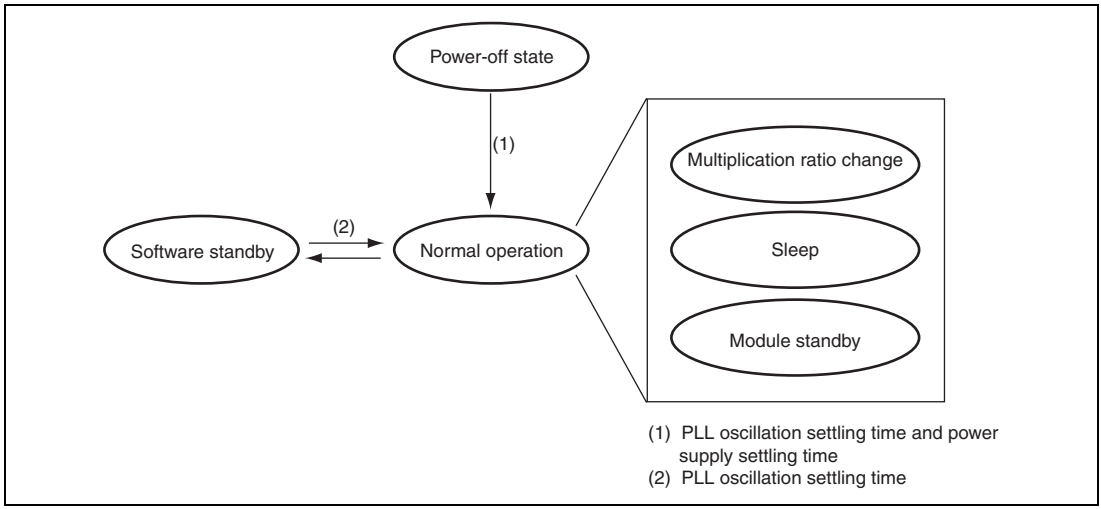


Figure 14.1 Mode Transition Diagram

14.4.6 Output Pins Change Timing

Figure 14.2 shows the state of output pins at a power-on reset.

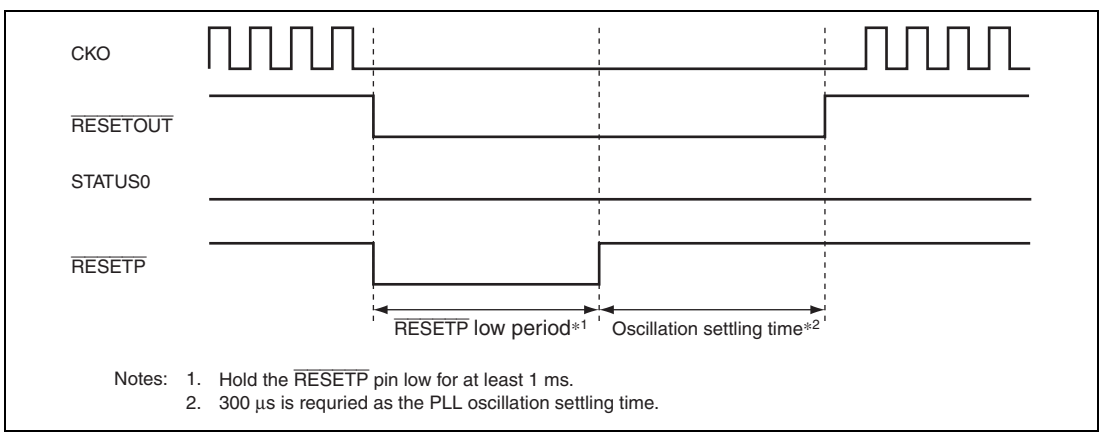


Figure 14.2 State of Output Pins at Power-On Reset

Figure 14.3 shows the state of output pins in software standby mode.

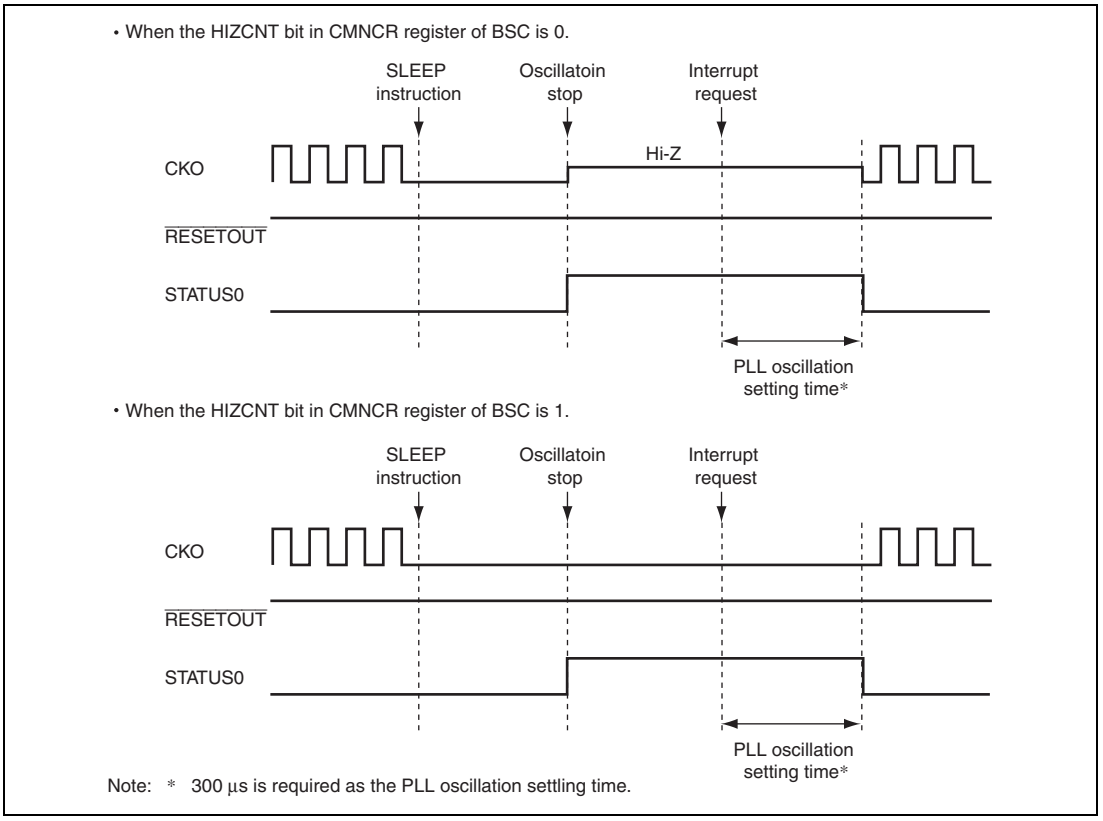


Figure 14.3 State of Output Pins on Exit from Software Standby Mode by Interrupt

Section 15 RCLK Watchdog Timer (RWDT)

This LSI includes the RCLK watchdog timer (RWDT).

This LSI can be reset by the overflow of the counter when the value of the counter has not been updated because of a system runaway.

The RWDT is a single-channel timer that uses a RCLK clock, of which the frequency is 1/1024 of the clock from the EXTAL pin, as an input and can be used as a watchdog timer.

15.1 Features

- Can be used as a watchdog timer. An internal reset is generated when the counter overflows.
- Choice of eight counter input clocks.
  Eight clocks (RCLK/1 to RCLK/4096) that are obtained by dividing the RCLK.

Figures 15.1 shows block diagrams of the RWDT.

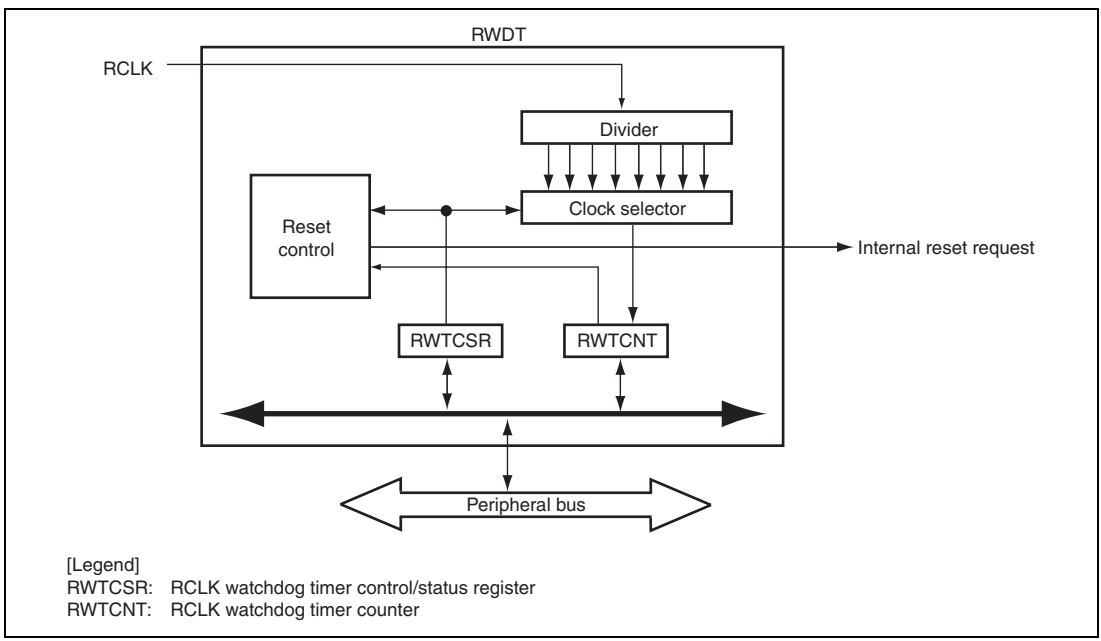


Figure 15.1 Block Diagram of RWDT

15.2 Input/Output Pins for RWDT

The RWDT has no input/output pins.

15.3 Register Descriptions for RWDT

Table 15.1 shows the RWDT register configuration. Table 15.2 shows the register state in each operating mode.

Table 15.1 Register Configuration of RWDT

Name	Abbreviation	R/W	Address	Access Size
RCLK watchdog timer counter	RWTCNT	R/W	H'A4520000	8/16*
RCLK watchdog timer control/status register	RWTCSR	R/W	H'A4520004	8/16*

Note: * Write is performed in 16-bit unit and read in 8-bit unit.

Table 15.2 Register State of RWDT in Each Operating Mode

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
RWTCNT	Initialized	Retained	Retained	Retained
RWTCSR	Initialized	Retained	Retained	Retained

15.3.1 RCLK Watchdog Timer Counter (RWTCNT)

RWTCNT is an 8-bit readable/writable register that increments on the selected clock. When an overflow occurs, it generates a power-on reset. The RWTCNT counter is initialized to H'00 by a power-on reset (including RWDT overflow reset.) Use a word access to write to the RWTCNT counter, with H'5A in the upper byte. Use a byte access to read RWTCNT.

Bit:	7	6	5	4	3	2	1	0
	RWTCNT							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.3.2 RCLK Watchdog Timer Control/Status Register (RWTCSR)

RWTCSR is an 8-bit readable/writable register composed of bits to select the clock used for the count, overflow flags, and enable bit.

RWTCSR is initialized to H'87 by a power-on reset (including RWDT overflow reset). Use a word access to write to RWTCSR, with H'A5 in the upper byte. Use a byte access to read RWTCSR.

BIt:	7	6	5	4	3	2	1	0
	TME	—	WR FLG	WOVF	—	CKS[2:0]		
Initial value:	1*	0	0	0	0	1	1	1
R/W:	R/W	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TME	1*	R/W	Starts and stops timer operation. 0: Timer disabled: Count-up stops and RWTCNT value is retained 1: Timer enabled
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	WRFLG	0	R	Write Status Flag The writing to the RWTCNT is disabled during this bit is 1.The writing to the RWTCNT is masked for the prescribed period to synchronize after the writing to the RWTCNT. Confirm that this bit is 0 to write to continuously the RWTCNT.
4	WOVF	0	R/W	Indicates that the RWTCNT has overflowed. Write 0 to this bit before using the RWDT. 0: No overflow 1: RWTCNT has overflowed
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	CKS[2:0]	111	R/W	RCLK Clock Select These bits select the clock to be used for the RWTCNT count from the eight types obtainable by dividing the RCLK clock. The overflow period that is shown inside the parenthesis in the table is the value when the RCLK clock is 32.768 kHz (EXTAL clock = 33.4 MHz). 000: Rϕ (7.9 ms) 001: Rϕ/4 (31.5 ms) 010: Rϕ/16 (126.0 ms) 011: Rϕ/32 (252.0 ms) 100: Rϕ/64 (503.0 ms) 101: Rϕ/128 (1.0 s) 110: Rϕ/1024 (8.1 s) 111: Rϕ/4096 (32.2 s)

Notes: * If bits CKS[2:0] are modified when the RWDT is operating, the up-count may not be performed correctly. Ensure that the bits CKS[2:0] are modified only when the RWDT is not operating.

15.3.3 Notes on Register Access

The writing procedure to RWTCNT and RWTCSR differs from that of other registers with the purpose of preventing an unintended write. The procedure for writing to these registers is given below.

Writing to RWTCNT and RWTCSR:

- These registers must be written by a word transfer instruction. They cannot be written by a byte or longword transfer instruction.
- When writing to RWTCNT, set the upper byte to H'5A and transfer the lower byte as the write data. When writing to RWTCSR, set the upper byte to H'A5 and transfer the lower byte as the write data.

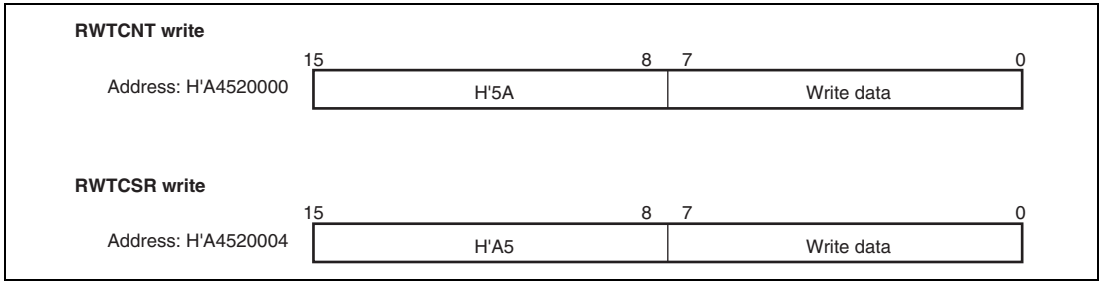


Figure 15.2 Writing to RWTCNT and RWTCSR

15.4 RWDT Usage

15.4.1 Control of System Runaway

Setting the TME bit in RWTCSR to 1 starts counting up on the RCLK. When the counter overflow occurs, an internal reset is again generated. By this function, an internal reset can be automatically generated even when this LSI has caused a system runaway.

1. Clear the WOVF bit in RWTCSR to 0.
2. Set the kind of count clock to the bits CKS[2: 0] in RWTCSR.
3. Start the counting by setting the TME bit in RWTCSR to 1.
4. Write periodically RWTCNT to H'00 so that RWTCNT does not overflow.
5. When RWTCNT overflows, a power-on reset is generated because the RWDT sets the WOVF flag in RWTCSR to 1. At this time, RWTCNT and RWTCSR are initialized.

Section 16 16-Bit Timer Pulse Unit (TPU)

This LSI has two on-chip 16-bit timer pulse units (TPU0 and TPU1). TPU0 consists of 4 channels of 16-bit timers and TPU1 consists of 2 channels of 16-bit timers.

16.1 Features

TPU0 and TPU1 have the following features.

- Maximum of 4 pulse outputs:
  TPU0 and TPU1 have 4 timer general registers (TPUn_TGRA, TPUn_TGRB, TPUn_TGRC, and TPUn_TGRD) for each channel. TPUn_TGRA can be used for output compare setting. TPUn_TGRB, TPUn_TGRC, and TPUn_TGRD in each channel can be used as the timer counter clear registers. TPUn_TGRC and TPUn_TGRD can be used as the buffer registers.
- The following operation can be set for each channel:
  Waveform output on compare match: Selection of 0, 1, or toggle output
  Counter clear operation: Counter clearing on compare match is possible
  PWM mode: PWM output with any desired duty cycle
  Maximum of 4-phase PWM output
- Buffer operation settable for each channel
  Automatic rewriting of output compare register possible
- One interrupt request line for each TPU0 and TPU1.
  Enabling or disabling the compare match/overflow interrupt request can be set independently for each interrupt source.

Table 16.1 describes the TPU functions.

Table 16.1 TPU Functions

		TPU0				TPU1	
Item		**Channel 0**	**Channel 1**	**Channel 2**	**Channel 3**	**Channel 0**	**Channel 1**
Counter clock		Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64
General register		TPU0_TGR0A TPU0_TGR0B	TPU0_TGR1A TPU0_TGR1B	TPU0_TGR2A TPU0_TGR2B	TPU0_TGR3A TPU0_TGR3B	TPU1_TGR0A TPU1_TGR0B	TPU1_TGR1A TPU1_TGR1B
General register/ Buffer register		TPU0_TGR0C TPU0_TGR0D	TPU0_TGR1C TPU0_TGR1D	TPU0_TGR2C TPU0_TGR2D	TPU0_TGR3C TPU0_TGR3D	TPU1_TGR0C TPU1_TGR0D	TPU1_TGR1C TPU1_TGR1D
Output pin		TPU0_TO0	TPU0_TO1	TPU0_TO2	TPU0_TO3	TPU1_TO0	TPU1_TO1
Counter clear function		TPU0_TGR compare match	TPU0_TGR compare match	TPU0_TGR compare match	TPU0_TGR compare match	TPU1_TGR compare match	TPU1_TGR compare match
Compare match output	0 output	O	O	O	O	O	O
	1 output	O	O	O	O	O	O
	Toggle output	O	O	O	O	O	O
PWM mode		O	O	O	O	O	O
Buffer mode		O	O	O	O	O	O
Interrupt source		5 sources • Compare match • Overflow	5 sources • Compare match • Overflow	5 sources • Compare match • Overflow	5 sources • Compare match • Overflow	5 sources • Compare match • Overflow	5 sources • Compare match • Overflow

16.2 Block Diagram

A block diagram of TPU0 and TPU1 is shown in figure 16.1.

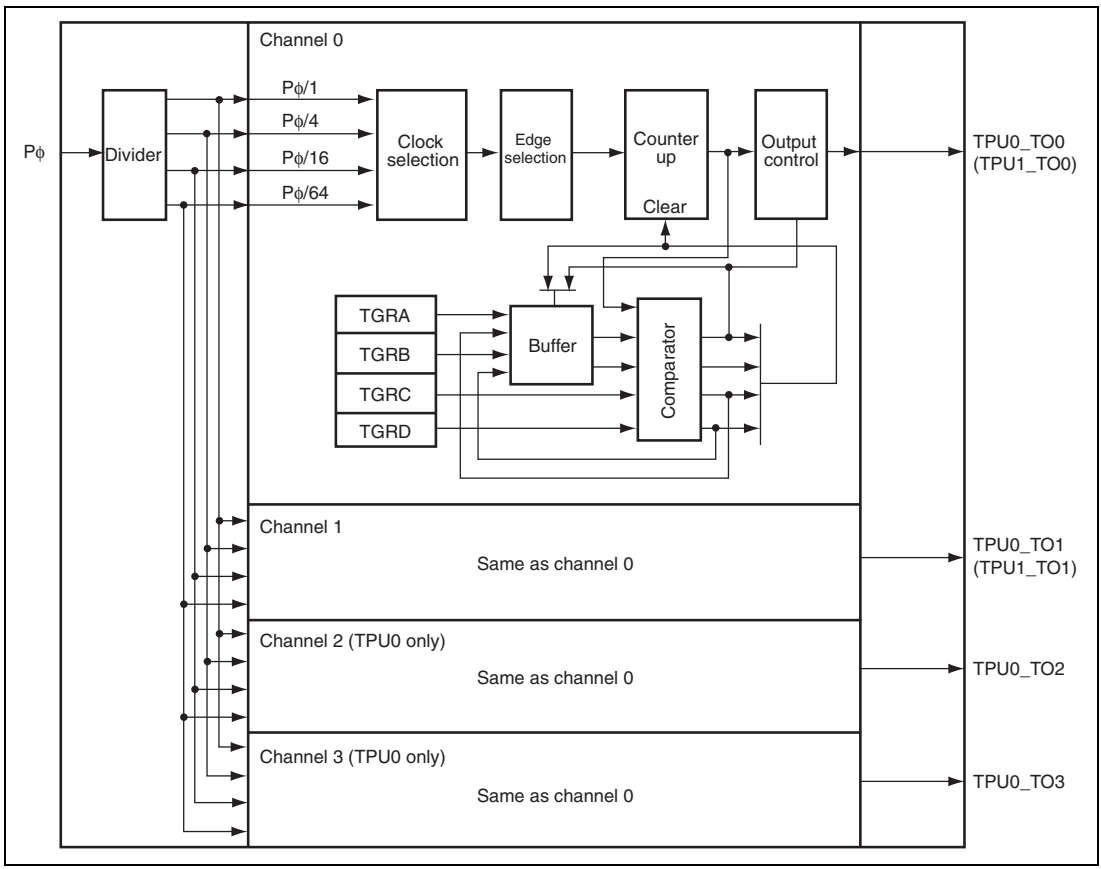


Figure 16.1 TPU0 and TPU1 Block Diagram

16.3 Input/Output Pin

Table 16.2 shows the pin configuration of TPU0 and TPU1.

Table 16.2 Pin Configuration

TPU0, TPU1	Channel	Pin Name	I/O	Functions
TPU0	0	TPU0_TO0	Output	TPU0 output compare match 0
				TPU0_TGR0A output compare output/PMW output pin
	1	TPU0_TO1	Output	TPU0 output compare match 1
				TPU0_TGR1A output compare output/PMW output pin
	2	TPU0_TO2	Output	TPU0 output compare match 2
				TPU0_TGR2A output compare output/PMW output pin
	3	TPU0_TO3	Output	TPU0 output compare match 3
				TPU0_TGR3A output compare output/PMW output pin
TPU1	0	TPU1_TO0	Output	TPU1 output compare match 0
				TPU1_TGR0A output compare output/PMW output pin
	1	TPU1_TO1	Output	TPU1 output compare match 1
				TPU1_TGR1A output compare output/PMW output pin

16.4 Register Descriptions

Table 16.3 shows TPU0 and TPU1 register configuration. Table 16.4 shows the register states in each operating mode.

In this section, registers are noted without distinction of channels, TPU0 and TPU1. The registers are noted as "TPUn_***".

Table 16.3 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
Timer start register	TPU0_TSTR	R/W	H'A463 0000	16
Timer control register 0	TPU0_TCR0	R/W	H'A463 0010	16
Timer mode register 0	TPU0_TMDR0	R/W	H'A463 0014	16
Timer I/O control register 0	TPU0_TIOR0	R/W	H'A463 0018	16
Timer interrupt enable register 0	TPU0_TIER0	R/W	H'A463 001C	16
Timer status register 0	TPU0_TSR0	R/W	H'A463 0020	16
Timer counter 0	TPU0_TCNT0	R/W	H'A463 0024	16
Timer general register 0A	TPU0_TGR0A	R/W	H'A463 0028	16
Timer general register 0B	TPU0_TGR0B	R/W	H'A463 002C	16
Timer general register 0C	TPU0_TGR0C	R/W	H'A463 0030	16
Timer general register 0D	TPU0_TGR0D	R/W	H'A463 0034	16
Timer control register 1	TPU0_TCR1	R/W	H'A463 0050	16
Timer mode register 1	TPU0_TMDR1	R/W	H'A463 0054	16
Timer I/O control register 1	TPU0_TIOR1	R/W	H'A463 0058	16
Timer interrupt enable register 1	TPU0_TIER1	R/W	H'A463 005C	16
Timer status register 1	TPU0_TSR1	R/W	H'A463 0060	16
Timer counter 1	TPU0_TCNT1	R/W	H'A463 0064	16
Timer general register 1A	TPU0_TGR1A	R/W	H'A463 0068	16
Timer general register 1B	TPU0_TGR1B	R/W	H'A463 006C	16
Timer general register 1C	TPU0_TGR1C	R/W	H'A463 0070	16
Timer general register 1D	TPU0_TGR1D	R/W	H'A463 0074	16

Register Name	Abbreviation	R/W	Address	Access Size
Timer control register 2	TPU0_TCR2	R/W	H'A463 0090	16
Timer mode register 2	TPU0_TMDR2	R/W	H'A463 0094	16
Timer I/O control register 2	TPU0_TIOR2	R/W	H'A463 0098	16
Timer interrupt enable register 2	TPU0_TIER2	R/W	H'A463 009C	16
Timer status register 2	TPU0_TSR2	R/W	H'A463 00A0	16
Timer counter 2	TPU0_TCNT2	R/W	H'A463 00A4	16
Timer general register 2A	TPU0_TGR2A	R/W	H'A463 00A8	16
Timer general register 2B	TPU0_TGR2B	R/W	H'A463 00AC	16
Timer general register 2C	TPU0_TGR2C	R/W	H'A463 00B0	16
Timer general register 2D	TPU0_TGR2D	R/W	H'A463 00B4	16
Timer control register 3	TPU0_TCR3	R/W	H'A463 00D0	16
Timer mode register 3	TPU0_TMDR3	R/W	H'A463 00D4	16
Timer I/O control register 3	TPU0_TIOR3	R/W	H'A463 00D8	16
Timer interrupt enable register 3	TPU0_TIER3	R/W	H'A463 00DC	16
Timer status register 3	TPU0_TSR3	R/W	H'A463 00E0	16
Timer counter 3	TPU0_TCNT3	R/W	H'A463 00E4	16
Timer general register 3A	TPU0_TGR3A	R/W	H'A463 00E8	16
Timer general register 3B	TPU0_TGR3B	R/W	H'A463 00EC	16
Timer general register 3C	TPU0_TGR3C	R/W	H'A463 00F0	16
Timer general register 3D	TPU0_TGR3D	R/W	H'A463 00F4	16
Timer start register	TPU1_TSTR	R/W	H'A44F 0000	16
Timer control register 0	TPU1_TCR0	R/W	H'A44F 0010	16
Timer mode register 0	TPU1_TMDR0	R/W	H'A44F 0014	16
Timer I/O control register 0	TPU1_TIOR0	R/W	H'A44F 0018	16
Timer interrupt enable register 0	TPU1_TIER0	R/W	H'A44F 001C	16
Timer status register 0	TPU1_TSR0	R/W	H'A44F 0020	16
Timer counter 0	TPU1_TCNT0	R/W	H'A44F 0024	16
Timer general register 0A	TPU1_TGR0A	R/W	H'A44F 0028	16
Timer general register 0B	TPU1_TGR0B	R/W	H'A44F 002C	16
Timer general register 0C	TPU1_TGR0C	R/W	H'A44F 0030	16
Timer general register 0D	TPU1_TGR0D	R/W	H'A44F 0034	16

Register Name	Abbreviation	R/W	Address	Access Size
Timer control register 1	TPU1_TCR1	R/W	H'A44F 0050	16
Timer mode register 1	TPU1_TMDR1	R/W	H'A44F 0054	16
Timer I/O control register 1	TPU1_TIOR1	R/W	H'A44F 0058	16
Timer interrupt enable register 1	TPU1_TIER1	R/W	H'A44F 005C	16
Timer status register 1	TPU1_TSR1	R/W	H'A44F 0060	16
Timer counter 1	TPU1_TCNT1	R/W	H'A44F 0064	16
Timer general register 1A	TPU1_TGR1A	R/W	H'A44F 0068	16
Timer general register 1B	TPU1_TGR1B	R/W	H'A44F 006C	16
Timer general register 1C	TPU1_TGR1C	R/W	H'A44F 0070	16
Timer general register 1D	TPU1_TGR1D	R/W	H'A44F 0074	16

Table 16.4 Register States in Each Operating Mode

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
TPU0_TSTR	Initialized	Retained	Retained	Retained
TPU0_TCR0 to TPU0_TCR3	Initialized	Retained	Retained	Retained
TPU0_TMDR0 to TPU0_TMDR3	Initialized	Retained	Retained	Retained
TPU0_TIOR0 to TPU0_TIOR3	Initialized	Retained	Retained	Retained
TPU0_TIER0 to TPU0_TIER3	Initialized	Retained	Retained	Retained
TPU0_TSR0 to TPU0_TSR3	Initialized	Retained	Retained	Retained
TPU0_TCNT0 to TPU0_TCNT3	Initialized	Retained	Retained	Retained
TPU0_TGRnA (n = 0 to 3)	Initialized	Retained	Retained	Retained
TPU0_TGRnB (n = 0 to 3)	Initialized	Retained	Retained	Retained
TPU0_TGRnC (n = 0 to 3)	Initialized	Retained	Retained	Retained
TPU0_TGRnD (n = 0 to 3)	Initialized	Retained	Retained	Retained
TPU1_TSTR	Initialized	Retained	Retained	Retained
TPU1_TCR0, TPU1_TCR1	Initialized	Retained	Retained	Retained
TPU1_TMDR0, TPU1_TMDR1	Initialized	Retained	Retained	Retained
TPU1_TIOR0, TPU1_TIOR1	Initialized	Retained	Retained	Retained
TPU1_TIER0, TPU1_TIER1	Initialized	Retained	Retained	Retained
TPU1_TSR0, TPU1_TSR1	Initialized	Retained	Retained	Retained
TPU1_TCNT0, TPU1_TCNT1	Initialized	Retained	Retained	Retained
TPU1_TGRnA (n = 0, 1)	Initialized	Retained	Retained	Retained
TPU1_TGRnB (n = 0, 1)	Initialized	Retained	Retained	Retained
TPU1_TGRnC (n = 0, 1)	Initialized	Retained	Retained	Retained
TPU1_TGRnD (n = 0, 1)	Initialized	Retained	Retained	Retained

16.4.1 Timer Control Register (TPUn_TCR)

TPUn_TCR controls the TPUn_TCNT for each channel. The TPU has one TPUn_TCR register for each channel. TPUn_TCR is initialized to H'0000 at a reset.

TPUn_TCR register settings should be made only while TPUn_TCNT operation is stopped.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
7 to 5	CCLR[2:0]	000	R/W	Counter Clear Select the TPUn_TCNT clearing source. 000: TPUn_TCNT clearing disabled 001: TPUn_TCNT cleared by TPUn_TGRA compare match 010: TPUn_TCNT cleared by TPUn_TGRB compare match 011: Setting prohibited 100: TPUn_TCNT clearing disabled 101: TPUn_TCNT cleared by TPUn_TGRC compare match 110: TPUn_TCNT cleared by TPUn_TGRD compare match 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
4, 3	CKEG[1:0]	00	R/W	Clock Edge Select the input clock edge. When the internal clock is counted using both edges, the input clock period is halved (e.g. Pϕ/4 both edges = Pϕ/2 rising edge). 00: Count at rising edge 01: Count at falling edge* 1X: Count at both edges* [Legend] X: Don't care Note: * If the input clock is Pϕ/1, no operation is performed.
2 to 0	TPSC[2:0]	000	R/W	Timer Prescaler Select the TPUn_TCNT counter clock. The clock source can be selected independently for each channel. Table 18.5 shows the clock sources that can be set for each channel. For more information on counter clock selection, see table 16.6.

Table 16.5 TPU Clock Sources

	Internal Clock			
Channel	**Pϕ/1**	**Pϕ/4**	**Pϕ/16**	**Pϕ/64**
0	O	O	O	O
1	O	O	O	O
2	O	O	O	O
3	O	O	O	O

[Legend]

O: Setting available

Table 16.6 Counter Clock Selection by the TPSC[2:0] Bits

Channel	TPSC[2]	TPSC[1]	TPSC[0]	Description
0 to 3	0	0	0	Internal clock: counts on Pϕ/1 (initial value)
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	*	*	Setting prohibited

[Legend]

*: Don't care.

16.4.2 Timer Mode Register (TPUn_TMDR)

TPUn_TMDR specifies the operation mode for each channel. The TPU has one TPUn_TMDR register for each channel. TPUn_TMDR is initialized to H'0000 at a reset.

TPUn_TMDR register settings should be made only while TPUn_TCNT operation is stopped.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	BFWT	BFB	BFA	—	MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
6	BFWT	0	R/W	Buffer Write Timing Specifies TPUn_TGRA and TPUn_TGRB update timing when TPUn_TGRC and TPUn_TGRD are used as a compare match buffer. When TPUn_TGRC and TPUn_TGRD are not used as a compare match buffer register, this bit does not function. 0: TPUn_TGRA and TPUn_TGRB are rewritten at compare match of each register. 1: TPUn_TGRA and TPUn_TGRB are rewritten at counter clearing.

Bit	Bit Name	Initial Value	R/W	Description
5	BFB	0	R/W	Buffer Operation B Specifies whether TPUn_TGRB is used in normal operation, or TPUn_TGRB and TPUn_TGRD are used in combination for buffer operation. 0: TPUn_TGRB normal operation 1: TPUn_TGRB and TPUn_TGRD used for buffer operation
4	BFA	0	R/W	Buffer Operation A Specifies whether TPUn_TGRA is used in normal operation, or TPUn_TGRA and TPUn_TGRC are used in combination for buffer operation. 0: TPUn_TGRA normal operation 1: TPUn_TGRA and TPUn_TGRC used for buffer operation
3	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
2 to 0	MD[2:0]	000	R/W	Timer Operating Mode Set the timer-operating mode. 000: Normal operating 001: Setting prohibited 010: PWM mode 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

16.4.3 Timer I/O Control Register (TPUn_TIOR)

TPUn_TIOR register controls the TPUn_TO0 to TPUn_TO3 pins, and has one TPUn_TIOR in each channel. The TPUn_TIOR is initialized to H'0000 at a reset.

The TPUn_TIOR register setting should be made only while TPUn_TCNT operation is stopped. Note that the setting of TPUn_TMDR may affect TPUn_TIOR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	IOA[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
2 to 0	IOA[2:0]	000	R/W	I/O Control Bits IOA2 to IOA0 specify the functions of TPUn_TGRA and the TPUn_TO0 to TPUn_TO3 pins. For details, see tables 16.7 and 16.8.

Table 16.7 Settings for Bits IOA[2:0], Initial States of Pin TPU0_TO0 to TPU0_TO3, and Results of Matching with TPU0_TGRA

Channel	IOA[2]	IOA[1]	IOA[0]	Description	
0 to 3	0	0	0	Always output 0 (initial value)	
			1	Initial output of the TPU0_TO0 to TPU0_TO3 pins are 0	Output 0 on compare match with TPU0_TGRA*
		1	0		Output 1 on compare match with TPU0_TGRA
			1		Toggle output on compare match with TPU0_TGRA*
	1	0	0	Always output 1	
			1	Initial output of the TPU0_TO0 to TPU0_TO3 pins are 1	Output 0 on compare match with TPU0_TGRA*
		1	0		Output 1 on compare match with TPU0_TGRA
			1		Toggle output on compare match with TPU0_TGRA*

Note: * Do not use this setting in PWM mode.

Table 16.8 Settings for Bits IOA[2:0], Initial States of Pin TPU1_TO0 and TPU1_TO1, and Results of Matching with TPU1_TGRA

Channel	IOA[2]	IOA[1]	IOA[0]	Description	
0 to 3	0	0	0	Always output 0 (initial value)	
			1	Initial output of the TPU1_TO0 and TPU1_TO1 pins are 0	Output 0 on compare match with TPU1_TGRA*
		1	0		Output 1 on compare match with TPU1_TGRA
			1		Toggle output on compare match with TPU1_TGRA*
	1	0	0	Always output 1	
			1	Initial output of the TPU1_TO0 and TPU1_TO1 pins are 1	Output 0 on compare match with TPU1_TGRA*
		1	0		Output 1 on compare match with TPU1_TGRA
			1		Toggle output on compare match with TPU1_TGRA*

Note: * Do not use this setting in PWM mode.

16.4.4 Timer Interrupt Enable Register (TPUn_TIER)

TPUn_TIER is used to enable and disable interrupt requests for each channel. The TPU has one TPUn_TIER register for each channel. TPUn_TIER is initialized to H'0000 by a reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	TC1EV	TG1ED	TG1EC	TG1EB	TG1EA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
4	TC1EV	0	R/W	Overflow Interrupt Enable When the TCFV flag in TPUn_TSR is set to 1 (a TCNT overflow has occurred), this bit enables or disables interrupt requests corresponding to the state of the TMCFS flag. 0: Interrupt requests by TCFV flag disabled 1: Interrupt requests by TCFV flag enabled
3	TG1ED	0	R/W	TPUn_TGR Interrupt Enable D When the TGFD bit in TPUn_TSR is set to 1 (a compare match between TPUn_TCNT and TPUn_TGRD has occurred), this bit enables or disables interrupt requests corresponding to the state of the TGFD flag. 0: Interrupt requests by TCFD flag disabled 1: Interrupt requests by TCFD flag enabled
2	TG1EC	0	R/W	TPUn_TGR Interrupt Enable C When the TGFC bit in TPUn_TSR is set to 1 (a compare match between TPUn_TCNT and TPUn_TGRC has occurred), this bit enables or disables interrupt requests corresponding to the state of the TGFC flag. 0: Interrupt requests by TCFC flag disabled 1: Interrupt requests by TCFC flag enabled

Bit	Bit Name	Initial Value	R/W	Description
1	TG1EB	0	R/W	TPUn_TGR Interrupt Enable B When the TGFB bit in TPUn_TSR is set to 1 (a compare match between TPUn_TCNT and TPUn_TGRB has occurred), this bit enables or disables interrupt requests corresponding to the state of the TGFB flag. 0: Interrupt requests by TCFB flag disabled 1: Interrupt requests by TCFB flag enabled
0	TG1EA	0	R/W	TPUn_TGR Interrupt Enable A When the TGFA bit in TPUn_TSR is set to 1 (a compare match between TPUn_TCNT and TPUn_TGRA has occurred), this bit enables or disables interrupt requests corresponding to the state of the TGFA flag. 0: Interrupt requests by TCFA flag disabled 1: Interrupt requests by TCFA flag enabled

16.4.5 Timer Status Registers (TPUn_TSR)

TPUn_TSR displays information on the state of each channel. The TPU has one TPUn_TSR register for each channel. TPUn_TSR is initialized to H'0000 by a reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
4	TCFV	0	R/(W)*	Overflow Flag Status flag indicating overflow of TPUn_TCNT [Clearing condition] Writing 0 to the TCFV bit after reading the bit when TCFV = 1 [Setting condition] Overflow of the value in TPUn_TCNT (i.e. the value changing from H'FFFF to H'0000)
3	TGFD	0	R/(W)*	Compare Flag D Status flag indicating a match with TPUn_TGRD [Clearing condition] Writing 0 to the TCFD bit after reading the bit when TCFD = 1 [Setting condition] A match between the values in TPUn_TCNT and TPUn_TGRD
2	TGFC	0	R/(W)*	Compare Flag C Status flag indicating a match with TPUn_TGRC [Clearing condition] Writing 0 to the TCFC bit after reading the bit when TCFC = 1 [Setting condition] A match between the values in TPUn_TCNT and TPUn_TGRC
1	TGFB	0	R/(W)*	Compare Flag B Status flag indicating a match with TPUn_TGRB [Clearing condition] Writing 0 to the TCFB bit after reading the bit when TCFB = 1 [Setting condition] A match between the values in TPUn_TCNT and TPUn_TGRB

Bit	Bit Name	Initial Value	R/W	Description
0	TGFA	0	R/(W)*	Compare Flag A Status flag indicating a match with TPUn_TGRA [Clearing condition] Writing 0 to the TCFA bit after reading the bit when TCFA = 1 [Setting condition] A match between the values in TPUn_TCNT and TPUn_TGRA

Note: * Writing a 0 is the only way to clear this flag.

16.4.6 Timer Counter (TPUn_TCNT)

TPUn_TCNT indicates a 16-bit counter. The TPU has one TPUn_TCNT per channel.

TPUn_TCNT is initialized to H'0000 by a reset.

16.4.7 Timer General Register (TPUn_TGR)

TPUn_TGR indicates a 16-bit general register. Four general registers (TPUn_TGRA, TPUn_TGRB, TPUn_TGRC, and TPUn_TGR) are provided for each channel. TPUn_TGRC and TPUn_TGRD can be designated for operation as buffer registers*. TPUn_TGR is initialized to H'FFFF by a reset.

Note: * The combination of TPUn_TGR and buffer register are TPUn_TGRA—TPUn_TGRC and TPUn_TGRB—TPUn_TGRD.

16.4.8 Timer Start Register (TPUn_TSTR)

TPUn_TSTR starts and stops TCNT operation for channels 0 to 3.

TPUn_TSTR is initialized to H'0000 by a reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CST3*	CST2*	CST1	CST0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
3	CST3*	0	R/W	Counter Start
2	CST2*	0	R/W	These bits select either start or stop of TPUn_TCNT
1	CST1	0	R/W	0: Stops TPUn_TCNTm count operation
0	CST0	0	R/W	1: TPUn_TCNTm count operation [Legend] m = 3 to 0 Note: * TPU1 only consists of channels 1 and 2. Therefore, in the case of TPU1, the write value to the CST3 and CST2 bits should always be 0.

16.5 Operation

16.5.1 Overview

Operation overview for each mode is as follows.

(1) Ordinary Operation

Each channel is provided with TPUn_TCNT and TPUn_TGR registers. TPUn_TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.

(2) Buffer Operation

When a compare match occurs, the buffer register value in the corresponding channel is transferred to TPUn_TGR. Updating timing to rewrite from buffer registers can be selected either when a compare match occurs or when the counter is cleared.

(3) PWM Mode

In PWM mode, PWM waveform is output. The output level can be set by TPUn_TGR. PWM waveform, whose duty is in the range of 0 to 100%, can be output by the settings of TPUn_TGRA and TPUn_TGRB.

16.5.2 Basic Functions

(1) Counter Operation

When the bits CST[3:0] in TPUn_TSTR are set to 1, the TPUn_TCNT for the corresponding channel starts counting. TPUn_TCNT can operate as a free-running counter, periodic counter, and so on.

(a) Example of count operation setting procedure

Figure 16.2 shows an example of the count operation setting procedure.

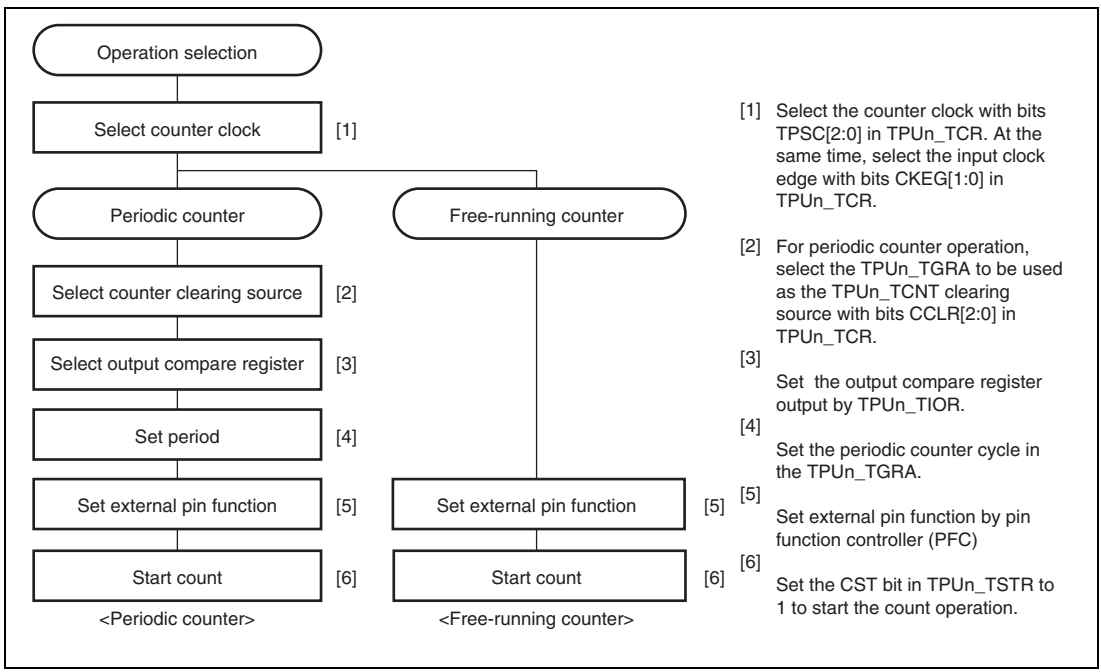


Figure 16.2 Example of Counter Operation Setting Procedure

(b) Free-running count operation and periodic count operation

Immediately after a reset, the TPUn_TCNT counters are all designated as free-running counters. When the relevant bit in TPUn_TSTR is set to 1, the corresponding TPUn_TCNT counter starts up-count operation as a free-running counter. When TPUn_TCNT has overflowed (changes from H'FFFF to H'0000), the TCFV bit in TPUn_TSR is set to 1. TPUn_TCNT starts counting up again from H'0000 after an overflow.

Figure 16.3 illustrates free-running counter operation.

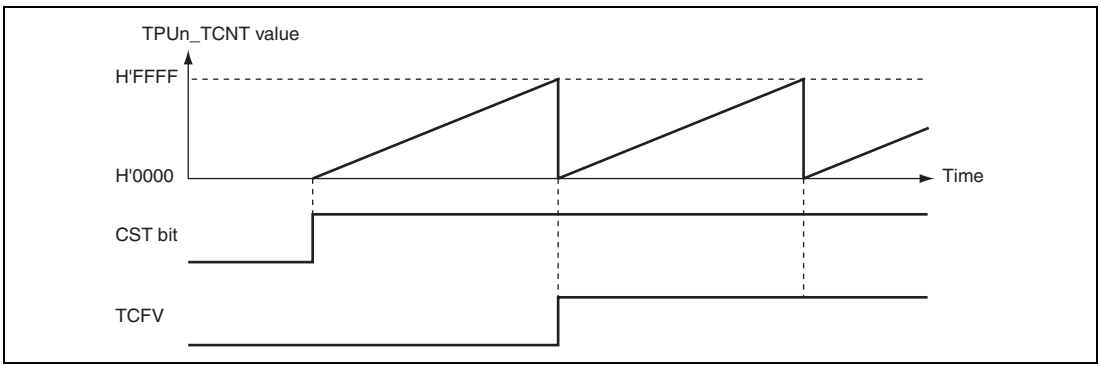


Figure 16.3 Free-Running Counter Operation

When a compare match is selected as the TPUn_TCNT clearing source, the TPUn_TCNT counter for the relevant channel performs periodic count operation. The TPUn_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 CCLR[2:0] in TPUn_TCR. After the settings have been made, TPUn_TCNT starts count-up operation as a periodic counter when the corresponding bit in TPUn_TSTR is set to 1. When the count value matches the value in TPUn_TGR, the TGF bit in TPUn_TSR is set to 1 and TPUn_TCNT is cleared to H'0000.

After a compare match, TPUn_TCNT starts counting up again from H'0000.

Figure 16.4 illustrates periodic counter operation.

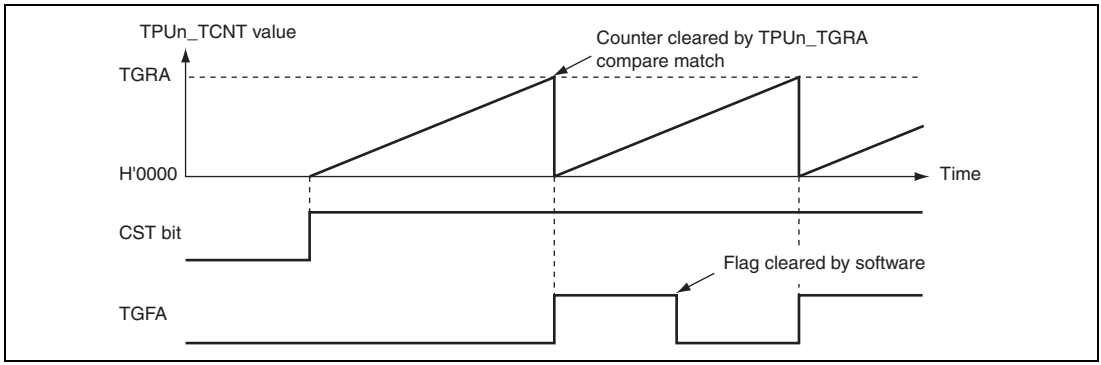


Figure 16.4 Periodic Counter Operation

(2) Waveform Output by Compare Match

The TPU can perform 0-, 1-, or toggle-output from the output pins (TPUn_TO0 to TPUn_TO3 pins) using a TPUn_TGRA compare match.

(a) Example of setting procedure for waveform output by compare match

Figure 16.5 shows an example of the setting procedure for waveform output by a compare match.

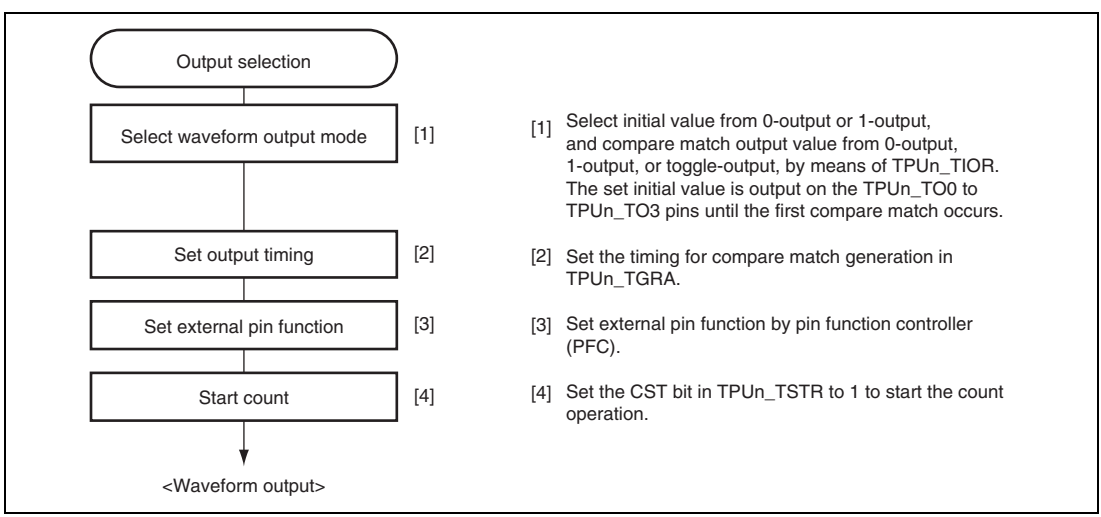


Figure 16.5 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of waveform output operation

Figure 16.6 shows an example of 0-output and 1-output.

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

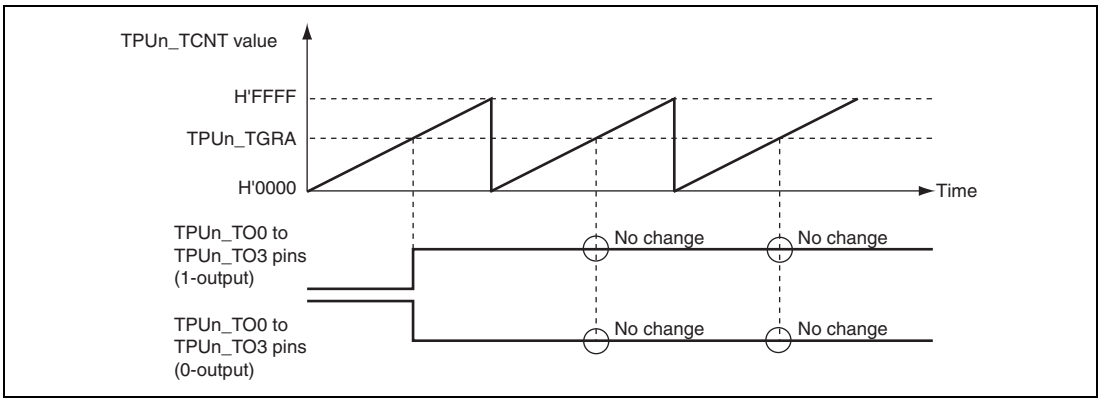


Figure 16.6 Example of 0-Output/1-Output Operation

Figure 16.7 shows an example of toggle output.

In this example, TPUn_TCNT has been designated as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by compare match A.

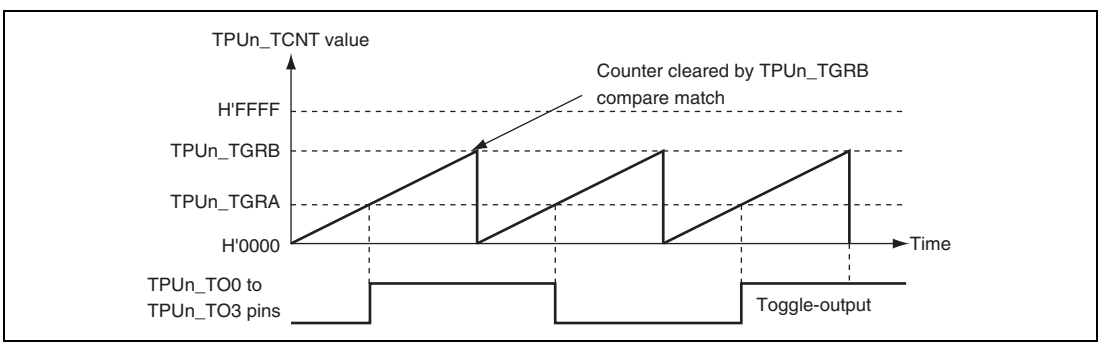


Figure 16.7 Example of Toggle Output Operation

16.5.3 Buffer Operation

TPUn_TGRC and TPUn_TGRD can be used as buffer registers.

Table 16.9 shows the register combinations used in buffer operation.

Table 16.9 Register Combinations in Buffer Operation

Timer General Register	Buffer Register
TPUn_TGRA	TPUn_TGRC
TPUn_TGRB	TPUn_TGRD

When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register. Updating timing to rewrite from buffer registers can be selected either when compare match occurs or when the counter is cleared.

This operation is illustrated in figure 16.8.

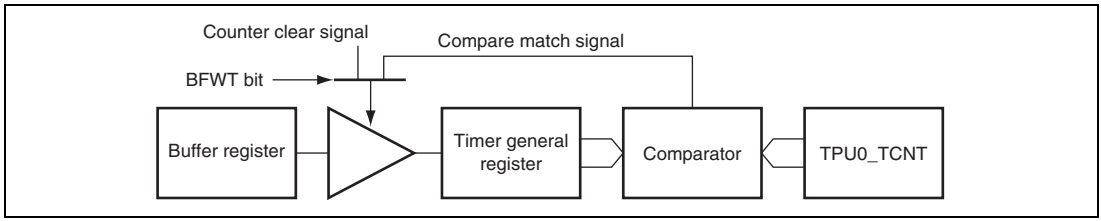


Figure 16.8 Compare Match Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 16.9 shows an example of the buffer operation setting procedure.

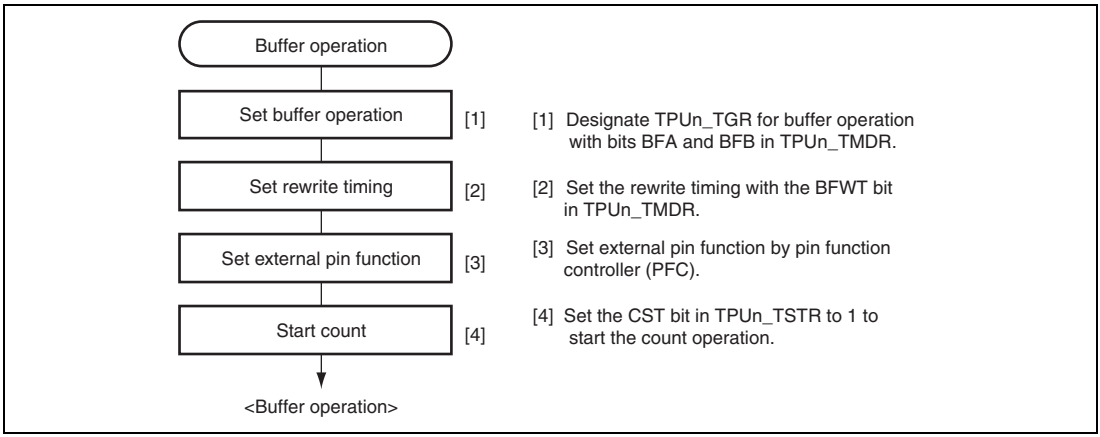


Figure 16.9 Example of Buffer Operation Setting Procedure

(2) Examples of buffer operation

Figure 16.10 shows an operation example in which PWM mode has been designated for channel 0, and buffer operation has been designated for TPUn_TGRA and TPUn_TGRC. The settings used in this example are TPUn_TCNT clearing by compare match B, 1-output (TPUn_TO0 to TPUn_TO3 pins) at compare match A, initial value 0 output by counter clearing, and the rewrite timing from buffer register at counter clearing.

When compare match A occurs, the output changes. When a counter clear is generated by TPUn_TGRB, the output changes and the value in buffer register TPUn_TGRC is simultaneously transferred to the timer general register TPUn_TGRA. This operation is repeated every time compare match A occurs.

For details on PWM modes, see section 16.5.4, PWM Modes.

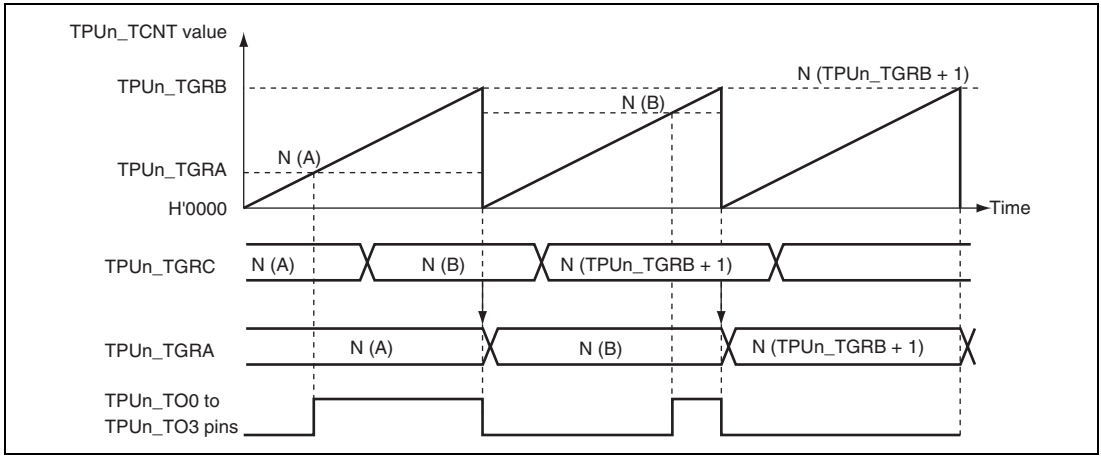


Figure 16.10 Example of Buffer Operation

16.5.4 PWM Modes

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

Designating TPUn_TGRB compare match as the counter clearing source enables the cycle to be set in that register. All channels can be designated for PWM mode independently.

PWM output is generated from the TPUn_TO0 to TPUn_TO3 pin by using TPUn_TGRA and TPUn_TGRB as duty register and periodic register respectively. The initial output specified by TPUn_TIOR is output to TPUn_TO0 to TPUn_TO3 pin by counter clearing due to periodic register compare match. Be sure to set TPUn_TIOR so that the initial output level is different from the compare match output. Selecting the same level or toggle output activates no operation.

Conditions on 0% and 100% duties are shown below.

- 0% duty: When periodic register (TPUn_TGRB) is set to the value equal to duty register TGRB + 1
- 100% duty: When duty register (TPUn_TGRA) is set to 0

In PWM mode, up to four types of PMW outputs are available.

(1) Example of PWM Mode Setting Procedure

Figure 16.11 shows an example of the PWM mode setting procedure.

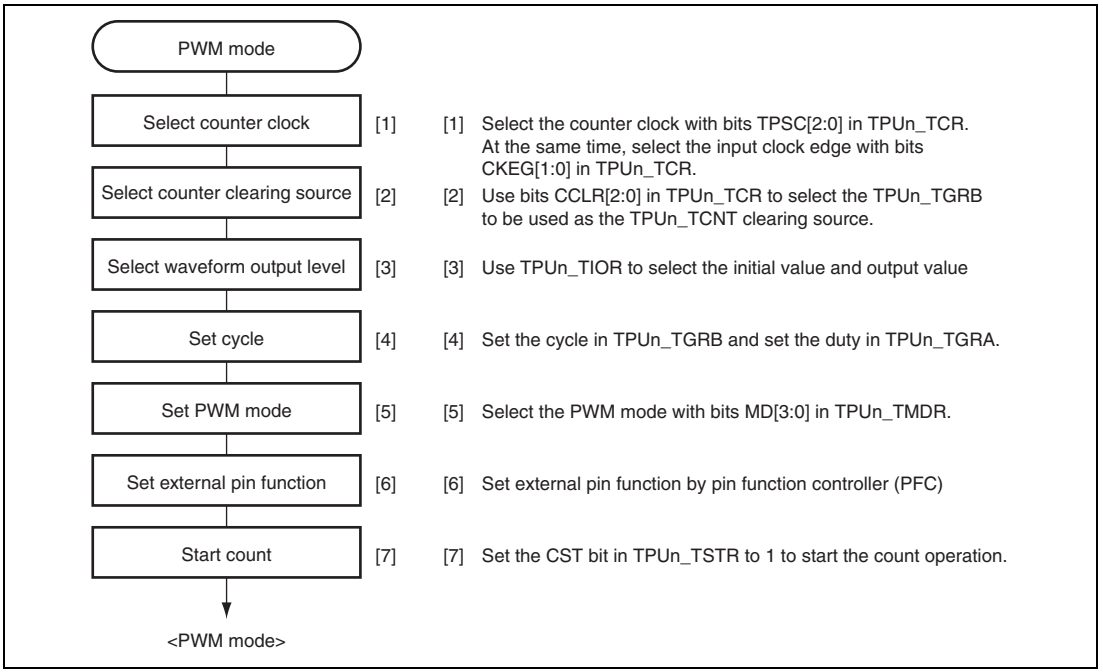


Figure 16.11 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 16.12 shows an example of PWM mode operation.

In this example, TPUn_TGRB compare match is set as the TPUn_TCNT clearing source, 0 is set for the TPUn_TGRA initial output value, and 1 is set as the output value.

In this case, the value set in TPUn_TGRB is used as the cycle, and the value set in TPUn_TGRA as the duty cycle.

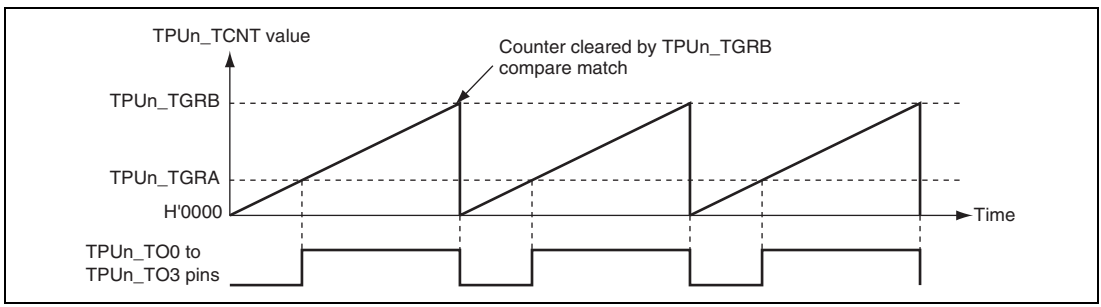


Figure 16.12 Example of PWM Mode Operation (1)

Figure 16.13 shows an example of PWM waveform output with 0% and 100% duties in PWM mode.

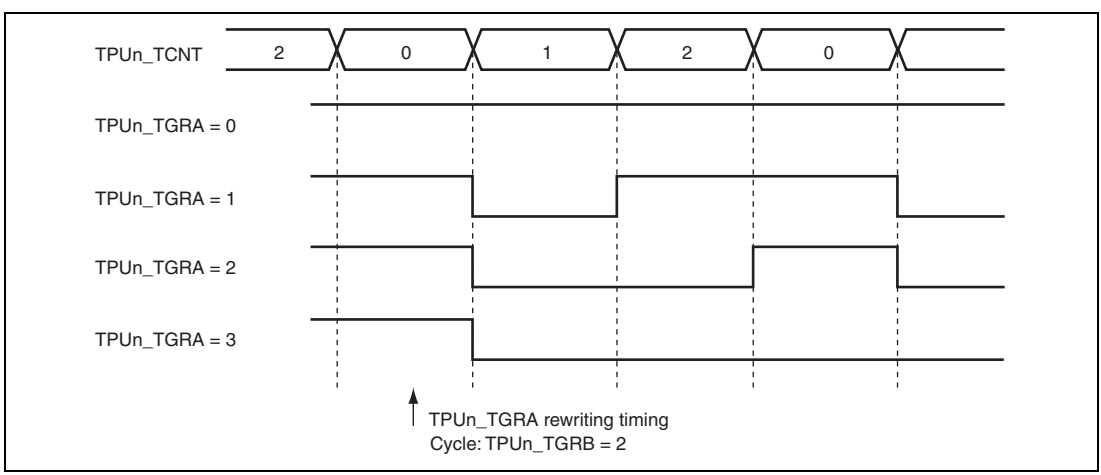


Figure 16.13 Example of PWM Mode Operation (2)

Section 17 Realtime Clock (RTC)

This LSI has a realtime clock (RTC) with its own 32.768-kHz crystal oscillator.

17.1 Features

- Clock and calendar functions (BCD format): Seconds, minutes, hours, date, day of the week, month, and year
- 1-Hz to 64-Hz timer (binary format)
  64-Hz counter indicates the state of the RTC divider circuit between 64 Hz and 1 Hz
- Start/stop function
- 30-second adjust function
- Alarm interrupt: Frame comparison of seconds, minutes, hours, date, day of the week, month, and year can be used as conditions for the alarm interrupt
- Periodic interrupts: the interrupt cycle may be 1/256 second, 1/64 second, 1/16 second, 1/4 second, 1/2 second, 1 second, or 2 seconds
- Carry interrupt: a carry interrupt indicates when a carry occurs during a counter read
- Automatic leap year adjustment

Note: This LSI does not have a separate power supply for the RTC. The RTC has the same power supply as that for input and output (VccQ and VssQ). Operating the RTC alone by shutting down the other power supplies is thus not possible.

Figure 17.1 shows the block diagram of RTC.

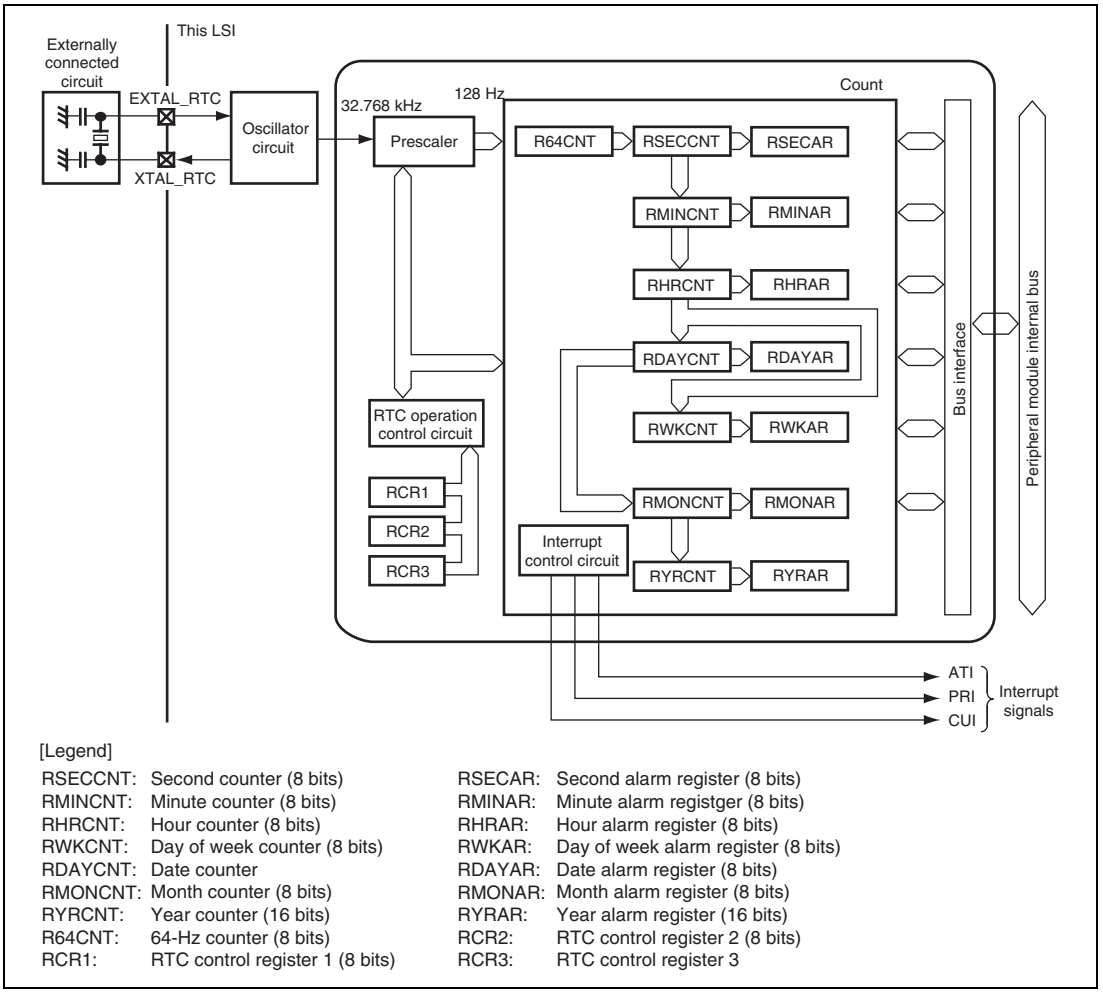


Figure 17.1 RTC Block Diagram

17.2 Input/Output Pin

Table 17.1 shows the RTC pin configuration.

Table 17.1 Pin Configuration

Name	Abbreviation	I/O	Function
Crystal resonator connection for RTC	EXTAL_RTC	Input	Connects the crystal resonator for the RTC.
	XTAL_RTC	Output	Connects the crystal resonator for the RTC.

Note: 1. When RTC is not to be used, pull EXTAL_RTC up to the power supply voltage (VccQ: 3.3V) for input and output. Do not connect anything to XTAL_RTC.
2. EXTAL_RTC is a pin to which a crystal resonator is connected. In putting on external clock to this pin is prohibited.

17.3 Register Descriptions

Table 17.2 shows the register configuration. Table 17.3 shows the register states in each operating mode.

Table 17.2 Register Configuration of RTC

Name	Abbreviation	R/W	Address	Access Size
64-Hz counter	R64CNT	R	H'A465 FEC0	8
Second counter	RSECCNT	R/W	H'A465 FEC2	8
Minute counter	RMINCNT	R/W	H'A465 FEC4	8
Hour counter	RHRCNT	R/W	H'A465 FEC6	8
Day of week counter	RWKCNT	R/W	H'A465 FEC8	8
Date counter	RDAYCNT	R/W	H'A465 FECA	8
Month counter	RMONCNT	R/W	H'A465 FECC	8
Year counter	RYRCNT	R/W	H'A465 FECE	16
Second alarm register	RSECAR	R/W	H'A465 FED0	8
Minute alarm register	RMINAR	R/W	H'A465 FED2	8
Hour alarm register	RHRAR	R/W	H'A465 FED4	8
Day of week alarm register	RWKAR	R/W	H'A465 FED6	8
Date alarm register	RDAYAR	R/W	H'A465 FED8	8
Month alarm register	RMONAR	R/W	H'A465 FEDA	8

Name	Abbreviation	R/W	Address	Access Size
Year alarm register (RYRAR)	RYRAR	R/W	H'A465 FEE0	16
RTC control register 1 (RCR1)	RCR1	R/W	H'A465 FEDC	8
RTC control register 2 (RCR2)	RCR2	R/W	H'A465 FEDE	8
RTC control register 3 (RCR3)	RCR3	R/W	H'A465 FEE4	8

Table 17.3 Register State of RTC in Each Operating Mode

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
R64CNT	Retained	Retained	Retained	Retained
RSECCNT	Retained	Retained	Retained	Retained
RMINCNT	Retained	Retained	Retained	Retained
RHRCNT	Retained	Retained	Retained	Retained
RWKCNT	Retained	Retained	Retained	Retained
RDAYCNT	Retained	Retained	Retained	Retained
RMONCNT	Retained	Retained	Retained	Retained
RYRCNT	Retained	Retained	Retained	Retained
RSECAR	Retained*	Retained	Retained	Retained
RMINAR	Retained*	Retained	Retained	Retained
RHRAR	Retained*	Retained	Retained	Retained
RWKAR	Retained*	Retained	Retained	Retained
RDAYAR	Retained*	Retained	Retained	Retained
RMONAR	Retained*	Retained	Retained	Retained
RYRAR	Initialized	Retained	Retained	Retained
RCR1	Initialized	Retained	Retained	Retained
RCR2	Retained	Retained	Retained	Retained
RCR3	Initialized	Retained	Retained	Retained

Note * There are bits that are initialized by a power-on reset.

17.3.1 64-Hz Counter (R64CNT)

R64CNT indicates the state of the divider circuit between 64 Hz and 1 Hz.

Reading this register, when carry from 128-Hz divider stage is generated, sets the CF bit in the RTC control register 1 (RCR1) to 1 so that the carrying and reading 64 Hz counter are performed at the same time is indicated. In this case, the R64CNT should be read again after writing 0 to the CF bit in RCR1 since the read value is not valid.

After the RESET bit or ADJ bit in the RTC control register 2 (RCR2) is set to 1, the RTC divider circuit is initialized and R64CNT is initialized to H'00.

R64CNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz
Initial value:	0	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

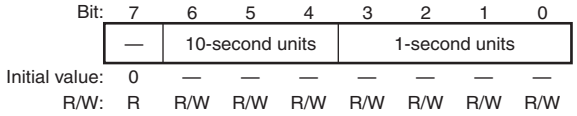
Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. Writing has no effect.
6	1 Hz	Undefined	R	Indicate the state of the divider circuit between 64 Hz and 1 Hz.
5	2 Hz	Undefined	R	
4	4 Hz	Undefined	R	
3	8 Hz	Undefined	R	
2	16 Hz	Undefined	R	
1	32 Hz	Undefined	R	
0	64 Hz	Undefined	R	

17.3.2 Second Counter (RSECCNT)

RSECCNT is used for setting/counting in the BCD-coded second section. The count operation is performed by a carry for each second of the 64-Hz counter.

The range of second can be set is 00 to 59 (decimal). Errant operation will result if any other value is set. Carry out write processing after stopping the count operation with the START bit in RCR2.

RSECCNT is not initialized by a power-on reset or manual reset, or in standby mode.



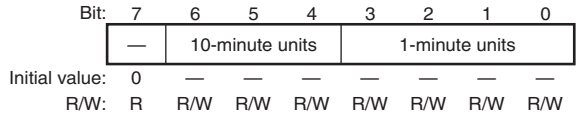
Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	—	Undefined	R/W	Counting Ten's Position of Seconds Counts on 0 to 5 for 60-seconds counting.
3 to 0	—	Undefined	R/W	Counting One's Position of Seconds Counts on 0 to 9 once per second. When a carry is generated, 1 is added to the ten’s position.

17.3.3 Minute Counter (RMINCNT)

RMINCNT is used for setting/counting in the BCD-coded minute section. The count operation is performed by a carry for each minute of the second counter.

The range of minute can be set is 00 to 59 (decimal). Errant operation will result if any other value is set. Carry out write processing after stopping the count operation with the START bit in RCR2.

RMINCNT is not initialized by a power-on reset or manual reset, or in standby mode.



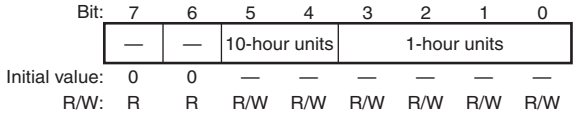
Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0.The write value should always be 0.
6 to 4	—	Undefined	R/W	Counting Ten's Position of Minutes Counts on 0 to 5 for 60-minutes counting.
3 to 0	—	Undefined	R/W	Counting One's Position of Minutes Counts on 0 to 9 once per second. When a carry is generated, 1 is added to the ten's position.

17.3.4 Hour Counter (RHRCNT)

RHRCNT is used for setting/counting in the BCD-coded hour section. The count operation is performed by a carry for each 1 hour of the minute counter.

The range of hour can be set is 00 to 23 (decimal). Errant operation will result if any other value is set. Carry out write processing after stopping the count operation with the START bit in RCR2.

RHRCNT is not initialized by a power-on reset or manual reset, or in standby mode.



Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. Though writing has no effect, the write value should always be 0.
5, 4	—	Undefined	R/W	Counting Ten's Position of Hours Counts on 0 to 2 for ten’s position of hours.
3 to 0	—	Undefined	R/W	Counting One's Position of Hours Counts on 0 to 9 once per hour. When a carry is generated, 1 is added to the ten's position.

17.3.5 Day of Week Counter (RWKCNT)

RWKCNT is used for setting/counting day of week section. The count operation is performed by a carry for each day of the date counter.

The range for day of the week can be set is 0 to 6 (decimal). Errant operation will result if any other value is set. Carry out write processing after stopping the count operation with the START bit in RCR2.

RWKCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	Day-of-week code		
Initial value:	0	0	0	0	0	—	—	—
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. Though writing has n effect, the write value should always be 0.
2 to 0	—	Undefined	R/W	Day-of-Week Counting Day-of-week is indicated with a binary code. 000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: Reserved (setting prohibited)

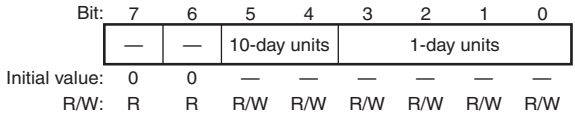
17.3.6 Date Counter (RDAYCNT)

RDAYCNT is used for setting/counting in the BCD-coded date section. The count operation is performed by a carry for each day of the hour counter.

The range of date, which can be set, is 01 to 31 (decimal). Errant operation will result if any other value is set. Carry out write processing after stopping the count operation with the START bit in RCR2.

RDAYCNT is not initialized by a power-on reset or manual reset, or in standby mode.

The range of date changes with each month and in leap years. Please confirm the correct setting. Leap years are recognized by dividing the year counter values by 400, 100, and 4 and obtaining a fractional result of 0. The year counter value of 0000 is included in the leap year.



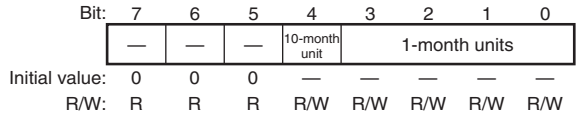
Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	—	Undefined	R/W	Counting Ten's Position of Dates
3 to 0	—	Undefined	R/W	Counting One's Position of Dates Counts on 0 to 9 once per date. When a carry is generated, 1 is added to the ten's position.

17.3.7 Month Counter (RMONCNT)

RMONCNT is used for setting/counting in the BCD-coded month section. The count operation is performed by a carry for each month of the date counter.

The range of month can be set is 01 to 12 (decimal). Errant operation will result if any other value is set. Carry out write processing after stopping the count operation with the START bit in RCR2.

RMONCNT is not initialized by a power-on reset or manual reset, or in standby mode.



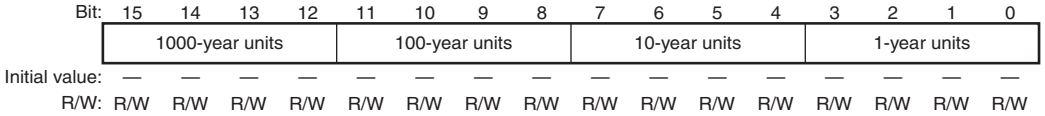
Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. Though writing has no effect, the write value should always be 0.
4	—	Undefined	R/W	Counting Ten's Position of Months
3 to 0	—	Undefined	R/W	Counting One's Position of Months Counts on 0 to 9 once per month. When a carry is generated, 1 is added to the ten's position.

17.3.8 Year Counter (RYRCNT)

RYRCNT is used for setting/counting in the BCD-coded year section. The count operation is performed by a carry for each year of the month counter.

The range for year, which can be set, is 0000 to 9999 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2 or using a carry flag.

RYRCNT is not initialized by a power-on reset or manual reset, or in standby mode.



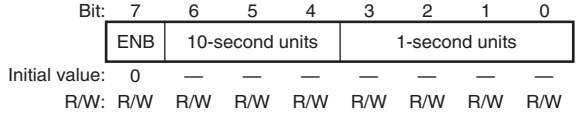
Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	Undefined	R/W	Counting Thousand's Position of Years
11 to 8	—	Undefined	R/W	Counting Hundred's Position of Years
7 to 4	—	Undefined	R/W	Counting Ten's Position of Years
3 to 0	—	Undefined	R/W	Counting One's Position of Years

17.3.9 Second Alarm Register (RSECAR)

RSECAR is an alarm register corresponding to the BCD coded second counter RSECCNT of the RTC. When the ENB bit is set to 1, a comparison with the RSECCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an alarm flag of RCR1 is set to 1.

The range of second alarm, which can be set, is 00 to 59 (decimal) + ENB bits. Errant operation will result if any other value is set.

The ENB bit in RSECAR is initialized to 0 by a power-on reset. The remaining RSECAR fields are not initialized by a power-on reset or manual reset, or in standby mode.



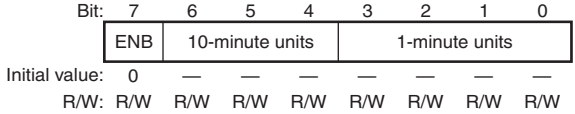
Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RSECCNT value is performed.
6 to 4	—	Undefined	R/W	Ten's position of seconds setting value
3 to 0	—	Undefined	R/W	One's position of seconds setting value

17.3.10 Minute Alarm Register (RMINAR)

RMINAR is an alarm register corresponding to the minute counter RMINCNT. When the ENB bit is set to 1, a comparison with the RMINCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The range of minute alarm, which can be set, is 00 to 59 (decimal). Errant operation will result if any other value is set.

The ENB bit in RMINAR is initialized by a power-on reset. The remaining RMINAR fields are not initialized by a power-on reset or manual reset, or in standby mode.



Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RMINCNT value is performed.
6 to 4	—	Undefined	R/W	Ten's position of minutes setting value
3 to 0	—	Undefined	R/W	One's position of minutes setting value

17.3.11 Hour Alarm Register (RHRAR)

RHRAR is an alarm register corresponding to the BCD coded hour counter RHRCNT of the RTC. When the ENB bit is set to 1, a comparison with the RHRCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The range of hour alarm, which can be set, is 00 to 23 (decimal). Errant operation will result if any other value is set.

The ENB bit in RHRAR is initialized by a power-on reset. The remaining RHRAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	10-hour units		1-hour units			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RHRCNT value is performed.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	—	Undefined	R/W	Ten's position of hours setting value
3 to 0	—	Undefined	R/W	One's position of hours setting value

17.3.12 Day of Week Alarm Register (RWKAR)

RWKAR is an alarm register corresponding to the BCD coded day of week counter RWKCNT. When the ENB bit is set to 1, a comparison with the RWKCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The range of day of the week alarm, which can be set, is 0 to 6 (decimal). Errant operation will result if any other value is set.

The ENB bit in RWKAR is initialized by a power-on reset. The remaining RWKAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	—	—	—	Day-of-week code		
Initial value:	0	0	0	0	0	—	—	—
R/W:	R/W	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RWKCNT value is performed.
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	—	Undefined	R/W	Day of week setting value

Code	0	1	2	3	4	5	6
Day	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

17.3.13 Date Alarm Register (RDAYAR)

RDAYAR is an alarm register corresponding to the BCD coded date counter RDAYCNT. When the ENB bit is set to 1, a comparison with the RDAYCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The range of date alarm, which can be set, is 01 to 31 (decimal). Errant operation will result if any other value is set. The RDAYCNT range that can be set changes with some months and in leap years. Please confirm the correct setting.

The ENB bit in RDAYAR is initialized by a power-on reset. The remaining RDAYAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	10-day units		1-day units			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RDAYCNT value is performed.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	—	Undefined	R/W	Ten's position of dates setting value
3 to 0	—	Undefined	R/W	One's position of dates setting value

17.3.14 Month Alarm Register (RMONAR)

RMONAR is an alarm register corresponding to the month counter RMONCNT. When the ENB bit is set to 1, a comparison with the RMONCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The range of month alarm, which can be set, is 01 to 12 (decimal). Errant operation will result if any other value is set.

The ENB bit in RMONAR is initialized by a power-on reset. The remaining RMONAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	—	10-month unit	1-month units			
Initial value:	0	0	0	—	—	—	—	—
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RMONCNT value is performed.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	—	Undefined	R/W	Ten's position of months setting value
3 to 0	—	Undefined	R/W	One's position of months setting value

17.3.15 Year Alarm Register (RYRAR)

RYRAR is an alarm register corresponding to the year counter RYRCNT. The range of year alarm, which can be set, is 0000 to 9999 (decimal). Errant operation will result if any other value is set.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1000-year units				100-year units				10-year units				1-year units			
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	Undefined	R/W	Thousand's position of years setting value
11 to 8	—	Undefined	R/W	Hundred's position of years setting value
7 to 4	—	Undefined	R/W	Ten's position of years setting value
3 to 0	—	Undefined	R/W	One's position of years setting value

17.3.16 RTC Control Register 1 (RCR1)

RCR1 is a register that affects carry flags and alarm flags. It also selects whether to generate interrupts for each flag.

RCR1 is initialized to H'00 by a power-on reset or a manual reset, and all bits are initialized to 0 except for the CF flag, which is undefined. When using the CF flag, it must be initialized beforehand. This register is not initialized in standby mode.

Bit:	7	6	5	4	3	2	1	0
	CF	—	—	CIE	AIE	—	—	AF
Initial value:	—	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CF	Undefined	R/W	Carry Flag (CUI) Status flag that indicates that a carry has occurred. CF is set to 1 when a count-up to 64-Hz occurs at the second counter carry or 64-Hz counter read. A count register value read at this time cannot be guaranteed; another read is required. 0: No carry of 64-Hz counter by second counter or 64-Hz counter [Clearing condition] When 0 is written to CF 1: Carry of 64-Hz counter by second counter or 64 Hz counter [Setting condition] When the second counter or 64-Hz counter is read during a carry occurrence by the 64-Hz counter, or 1 is written to CF.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	CIE	0	R/W	Carry Interrupt Enable Flag (CUI) When the carry flag (CF) is set to 1, the CIE bit enables interrupts. 0: A carry interrupt is not generated when the CF flag is set to 1 1: A carry interrupt is generated when the CF flag is set to 1
3	AIE	0	R/W	Alarm Interrupt Enable Flag (ATI) When the alarm flag (AF) is set to 1, the AIE bit allows interrupts. 0: An alarm interrupt is not generated when the AF flag is set to 1 1: An alarm interrupt is generated when the AF flag is set to 1

Bit	Bit Name	Initial Value	R/W	Description
2, 1	c	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	AF	0	R/W	Alarm Flag (ATI) The AF flag is set when the alarm time, which is set by an alarm register (ENB bit in RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, or RYRAR is set to 1), and counter match. 0: Alarm register and counter not match [Clearing condition] When 0 is written to AF. 1: Alarm register and counter match* [Setting condition] When alarm register (only a register with ENB bit set to 1) and counter match Note: * Writing 1 holds previous value.

17.3.17 RTC Control Register 2 (RCR2)

RCR2 is a register for periodic interrupt control, 30-second adjustment ADJ, divider circuit RESET, and RTC count control.

RCR2 is initialized to H'09 by a power-on reset. It is initialized except for RTCEN and START by a manual reset. It is not initialized in standby mode, and retains its contents.

Bit:	7	6	5	4	3	2	1	0
	PEF	PES[2:0]			RTCEN	ADJ	RESET	START
Initial value:	0	0	0	0	1	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PEF	0	R/W	Periodic Interrupt Flag (PRI) Indicates interrupt generation with the period designated by the PES[2:0] bits. When set to 1, PEF generates periodic interrupts. 0: Interrupts not generated with the period designated by bits PES[2:0]. [Clearing condition] When 0 is written to PEF 1: Interrupts generated with the period designated by bits PES[2:0]. [Setting condition] When an interrupt is generated with the period designated by bits PES[2:0] or when 1 is written to the PEF flag
6 to 4	PES[2:0]	000	R/W	Interrupt Enable Flags (PRI) These bits specify the periodic interrupt. 000: No periodic interrupts generated 001: Periodic interrupt generated every 1/256 second 010: Periodic interrupt generated every 1/64 second 011: Periodic interrupt generated every 1/16 second 100: Periodic interrupt generated every 1/4 second 101: Periodic interrupt generated every 1/2 second 110: Periodic interrupt generated every 1 second 111: Periodic interrupt generated every 2 seconds
3	—	1	—	Reserved This bit is always read as 1. The write value should always be 1.
2	ADJ	0	R/W	30-Second Adjustment When 1 is written to the ADJ bit, times of 29 seconds or less will be rounded to 00 seconds and 30 seconds or more to 1 minute. The divider circuit (RTC prescaler and R64CNT) will be simultaneously reset. This bit always reads 0. 0: Runs normally. 1: 30-second adjustment.

Bit	Bit Name	Initial Value	R/W	Description
1	RESET	0	R/W	Reset When 1 is written, initializes the divider circuit (RTC prescaler and R64CNT). This bit always reads 0. 0: Runs normally. 1: Divider circuit is reset.
0	START	1	R/W	Start Bit Halts and restarts the counter (clock). 0: Second/minute/hour/day/week/month/year counter halts. 1: Second/minute/hour/day/week/month/year counter runs normally. Note: The 64-Hz counter always runs unless stopped with the RTCEN bit.

17.3.18 RTC Control Register 3 (RCR3)

When the ENB bit is set to 1, RCR3 performs a comparison with the RYRCNT. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an alarm flag of RCR1 is set to 1.

The ENB bit in RYRAR is initialized by a power-on reset. Remaining fields of RCR3 are not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, comparison of the year alarm register (RYRAR) and the year counter (RYRCNT) is performed.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

17.4 Operation

RTC usage is shown below.

17.4.1 Initial Settings of Registers after Power-On

All the registers should be set after the power is turned on.

17.4.2 Setting Time

Figure 17.2 shows how to set the time when the clock is stopped.

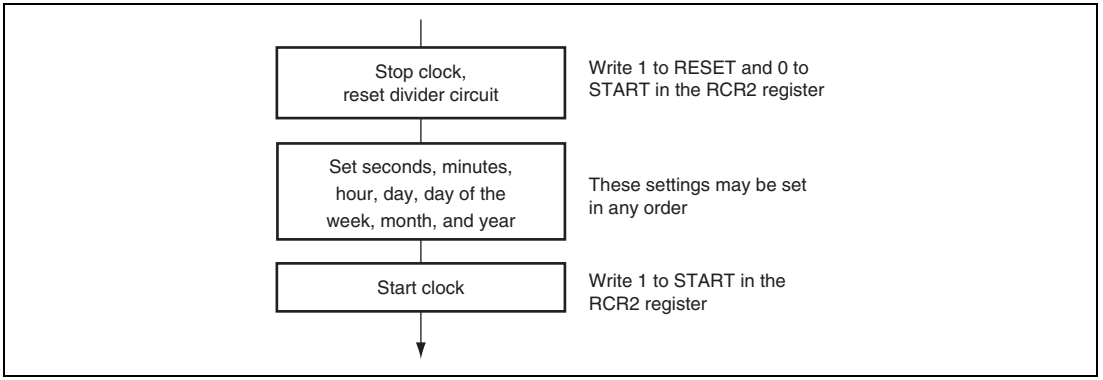


Figure 17.2 Setting Time

17.4.3 Reading Time

Figure 17.3 shows how to read the time.

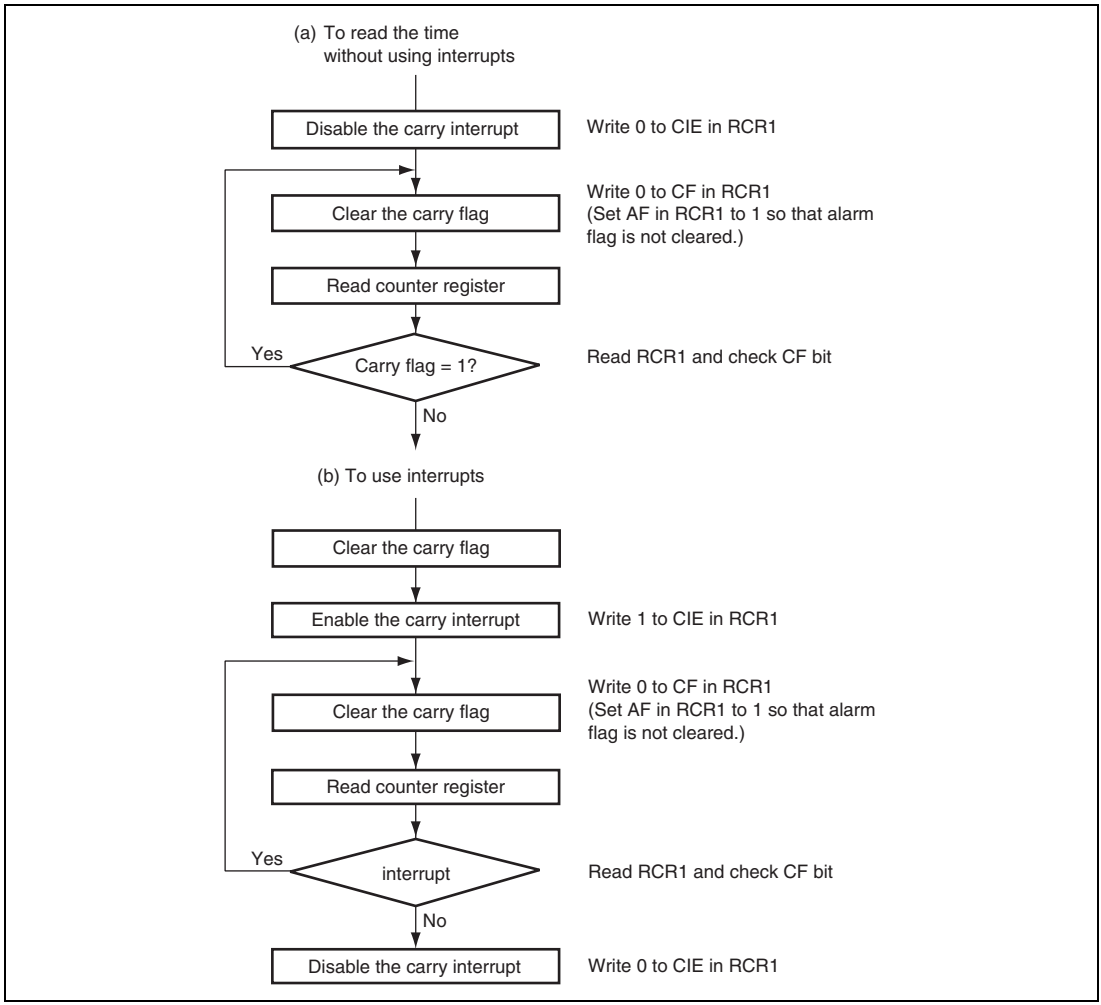


Figure 17.3 Reading Time

If a carry occurs while reading the time, the correct time will not be obtained, so it must be read again. Part (a) in figure 17.3 shows the method of reading the time without using interrupts; part (b) in figure 17.3 shows the method using carry interrupts. To keep programming simple, method (a) should normally be used.

17.4.4 Alarm Function

Figure 17.4 shows how to use the alarm function.

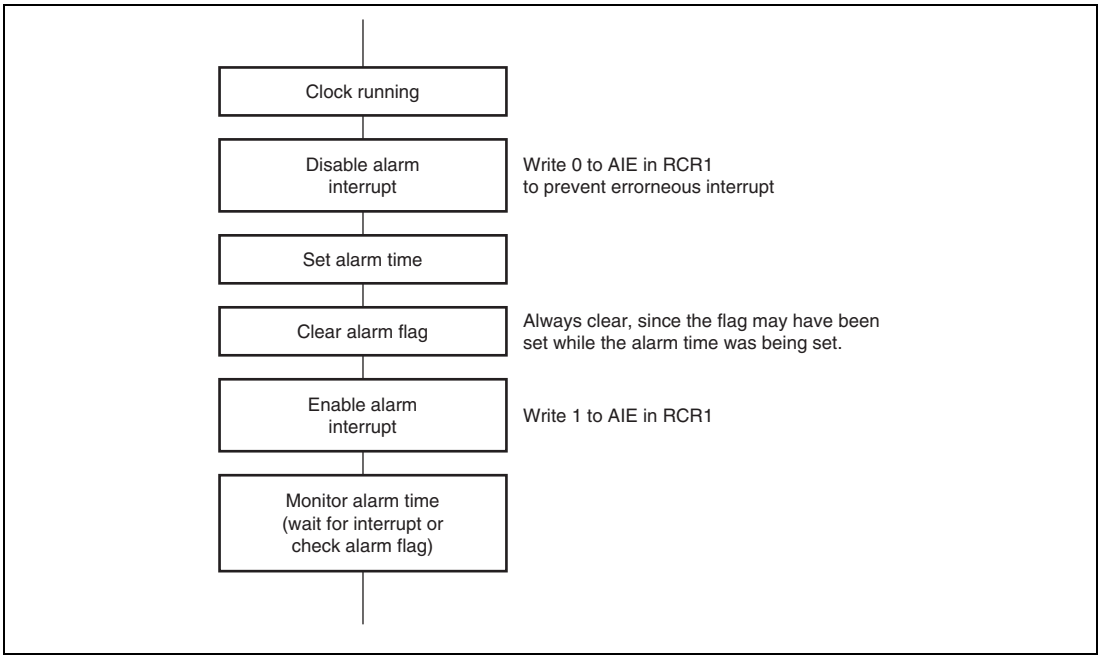


Figure 17.4 Using Alarm Function

Alarms can be generated using seconds, minutes, hours, day of the week, date, month, year, or any combination of these. Set the ENB bit in the register on which the alarm is placed to 1, and then set the alarm time in the lower bits. Clear the ENB bit in the register on which the alarm is not placed to 0.

When the clock and alarm times match, 1 is set in the AF bit in RCR1. Alarm detection can be checked by reading this bit, but normally it is done by interrupt. If 1 is set in the AIE bit in RCR1, an interrupt is generated when an alarm occurs.

The alarm flag is set when the clock and alarm times match. However, the alarm flag can be cleared by writing 0.

17.5 Usage Notes

17.5.1 Register Writing during RTC Count

The following RTC registers cannot be written to during an RTC count (while bit 0 = 1 in RCR2).

RSECCNT, RMINCNT, RHRCNT, RDAYCNT, RWKCNT, RMONCNT, RYRCNT

The RTC count must be stopped before writing to any of the above registers.

17.5.2 Use of Realtime Clock (RTC) Periodic Interrupts

The method of using the periodic interrupt function is shown in figure 17.5.

A periodic interrupt can be generated periodically at the interval set by the flags PES[2:0] in RCR2. When the time set by the PES[2:0] has elapsed, the PEF is set to 1.

The PEF is cleared to 0 upon periodic interrupt generation or when the flags PES[2:0] are set. Periodic interrupt generation can be confirmed by reading this bit, but normally the interrupt function is used.

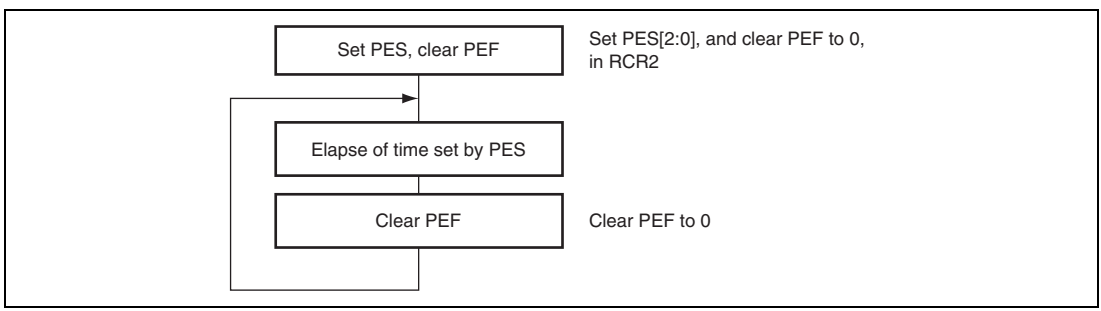


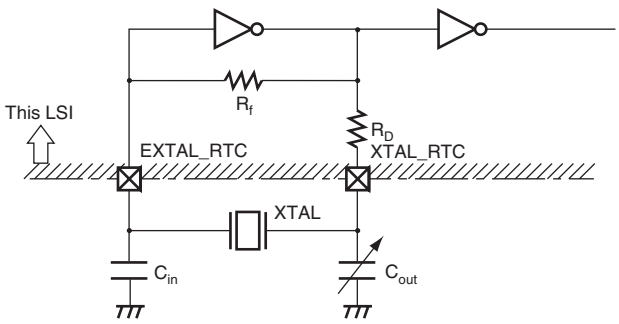
Figure 17.5 Using Periodic Interrupt Function

17.5.3 Transition to Standby Mode after Setting Register

When a transition to standby mode is made after registers in the RTC are set, sometimes counting is not performed correctly. In case the registers are set, be sure to make a transition to standby mode after waiting for two RTC clocks or more.

17.5.4 Crystal Oscillator Circuit

An example of the RTC crystal oscillator circuit is shown in figure 17.6.



Notes:
1. Select either the C_{in} or C_{out} side for frequency adjustment variable capacitor according to requirements such as frequency range, degree of stability, etc.
2. Built-in resistance value R_f (Typ value) = 10 MΩ, R_D (Typ value) = 400 kΩ
3. C_{in} and C_{out} values include floating capacitance due to the wiring. Take care when using a ground plane.
4. The crystal oscillation settling time depends on the mounted circuit constants, stray capacitance, etc., and should be decided after consultation with the crystal resonator manufacturer.
5. Place the crystal resonator and load capacitors C_{in} and C_{out} as close as possible to the chip.
   (Correct oscillation may not be possible if there is externally induced noise in the EXTAL_RTC and XTAL_RTC pins.)
6. Ensure that the crystal resonator connection pin (EXTAL_RTC, XTAL_RTC) wiring is routed as far away as possible from other power lines (except GND) and signal lines.

Figure 17.6 Example of Crystal Oscillator Circuit Connection

17.5.5 Usage of 30-Second Adjustment

When using the 30-second adjustment function, follow the procedure shown in figure 17.7.

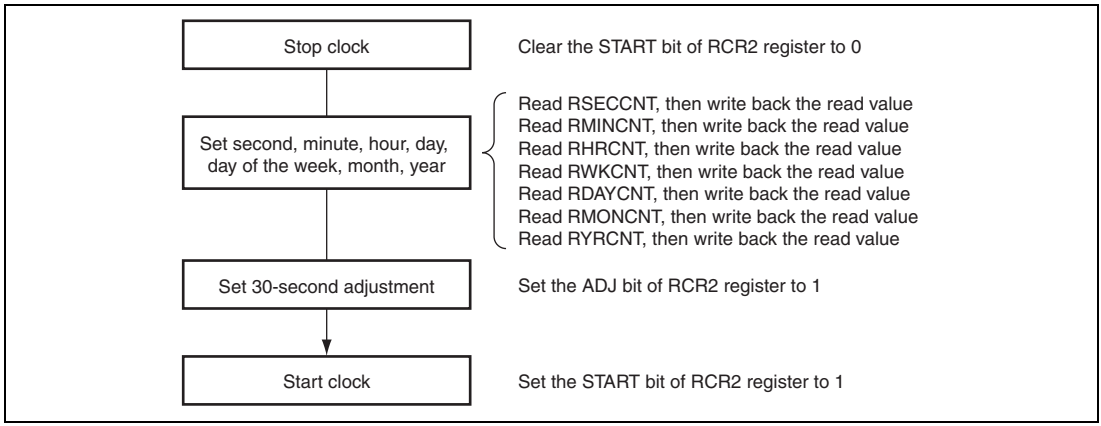


Figure 17.7 Usage of 30-Second Adjustment

When using the 30-second adjustment, the seconds, minutes, hours, date, day of the week, month and year must all be written to the corresponding registers. Firstly, clear the START bit of RCR2 register to 0. After reading out the minute, hour, day, day of the week, month and year, write back the values read and set the ADJ bit of the RCR register to 1. After making the 30-second adjustment, start the clock by setting the START bit of RCR2 register to 1.

Section 18 Timer Unit (TMU)

This LSI includes a three-channel 32-bit timer unit (TMU).

18.1 Features

- Each channel is provided with an auto-reload 32-bit down counter.
- All channels are provided with 32-bit constant registers and 32-bit down counters that can be read or written to at any time.
- All channels generate interrupt requests when the 32-bit down counter underflows (H'00000000 → H'FFFFFFFF)
- Allows selection among five counter input clocks: Pϕ/4, Pϕ/16, Pϕ/64, Pϕ/256, and Pϕ/1024

Figure 18.1 shows a block diagram of the TMU.

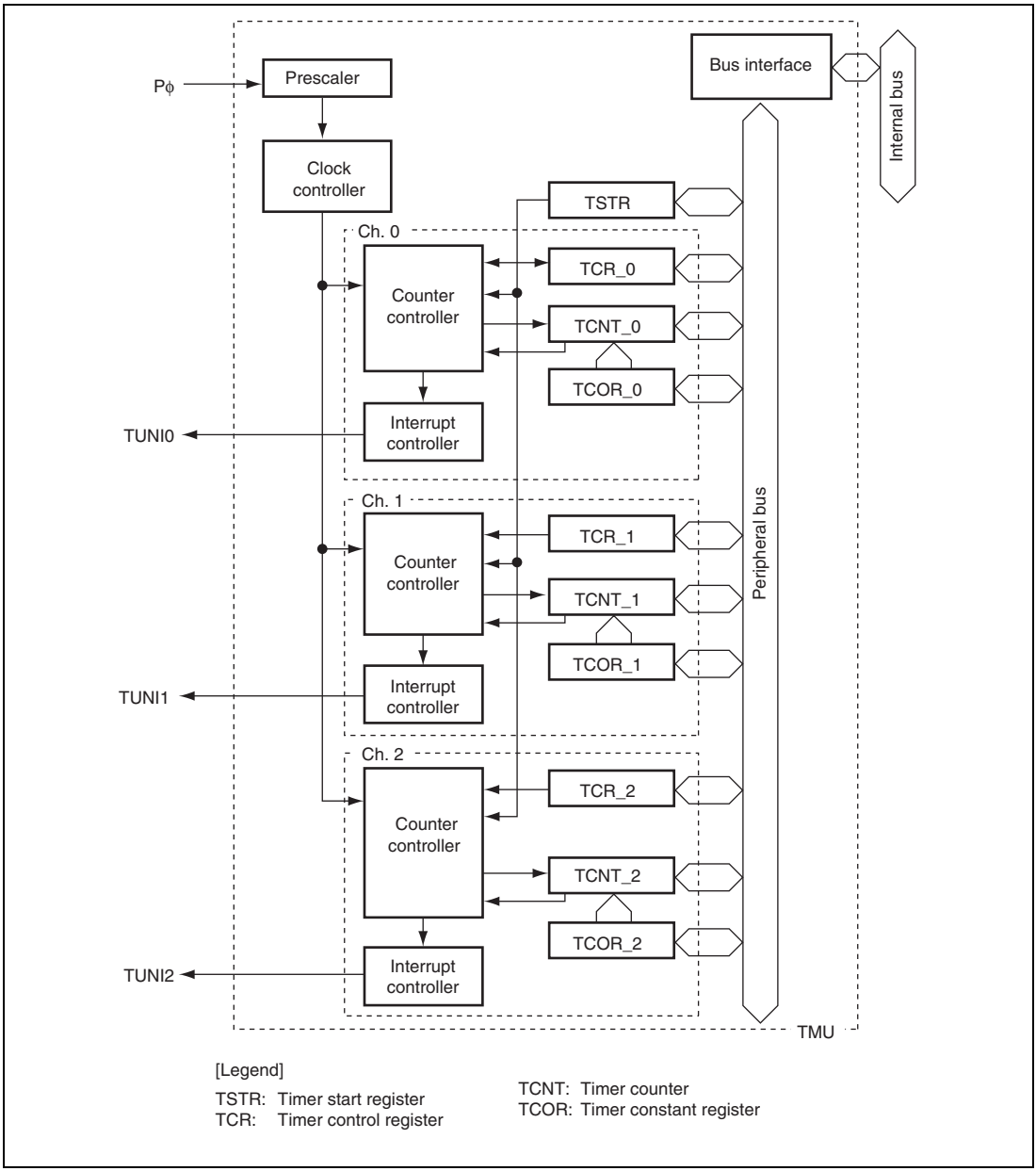


Figure 18.1 Block Diagram of TMU

18.2 Register Descriptions

Table 18.1 shows the TMU register configuration. Table 18.2 shows the register states in each operating mode. Note that the channel numbers are omitted from the register names in descriptions subsequent to these tables.

Table 18.1 Register Configuration

Register	Abbreviation	R/W	Address	Access Size
Timer start register	TSTR	R/W	H'FFD80004	8
Timer constant register_0	TCOR_0	R/W	H'FFD80008	32
Timer counter_0	TCNT_0	R/W	H'FFD8000C	32
Timer control register_0	TCR_0	R/W	H'FFD80010	16
Timer constant register_1	TCOR_1	R/W	H'FFD80014	32
Timer counter_1	TCNT_1	R/W	H'FFD80018	32
Timer control register_1	TCR_1	R/W	H'FFD8001C	16
Timer constant register_2	TCOR_2	R/W	H'FFD80020	32
Timer counter_2	TCNT_2	R/W	H'FFD80024	32
Timer control register_2	TCR_2	R/W	H'FFD80028	16

Table 18.2 Register States in Each Operating Mode

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
TSTR	Initialized	Retained	Retained	Retained
TCOR_0	Initialized	Retained	Retained	Retained
TCNT_0	Initialized	Retained	Retained	Retained
TCR_0	Initialized	Retained	Retained	Retained
TCOR_1	Initialized	Retained	Retained	Retained
TCNT_1	Initialized	Retained	Retained	Retained
TCR_1	Initialized	Retained	Retained	Retained
TCOR_2	Initialized	Retained	Retained	Retained
TCNT_2	Initialized	Retained	Retained	Retained
TCR_2	Initialized	Retained	Retained	Retained

18.2.1 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects whether to operate or halt the timer counters (TCNT).

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	STR2	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	STR2	0	R/W	Counter Start 2 Selects whether to operate or halt timer counter 2 (TCNT_2). 0: TCNT_2 count halted 1: TCNT_2 counts
1	STR1	0	R/W	Counter Start 1 Selects whether to operate or halt timer counter 1 (TCNT_1). 0: TCNT_1 count halted 1: TCNT_1 counts
0	STR0	0	R/W	Counter Start 0 Selects whether to operate or halt timer counter 0 (TCNT_0). 0: TCNT_0 count halted 1: TCNT_0 counts

18.2.2 Timer Control Registers (TCR)

TCR are 16-bit readable/writable registers that control the timer counters (TCNT) and interrupts.

TCR control the issuance of interrupts when the flag (UNF) indicating timer counter (TCNT) underflow has been set to 1, and also carry out counter clock selection.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UNF	—	—	UNIE	—	—	TPSC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*	R	R	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	UNF	0	R/(W)*	Underflow Flag Status flag that indicates occurrence of a TCNT underflow. 0: TCNT has not underflowed [Clearing condition] 0 is written to UNF 1: TCNT has underflowed [Setting condition] TCNT underflows
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	UNIE	0	R/W	Underflow Interrupt Control Controls enabling of interrupt generation when the status flag (UNF) indicating TCNT underflow has been set to 1. 0: Interrupt due to UNF (TUNI) is disabled 1: Interrupt due to UNF (TUNI) is enabled
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	TPSC[2:0]	000	R/W	Timer Prescaler 2, 1, and 0 Select the TCNT count clock. 000: Count on Pϕ/4 001: Count on Pϕ/16 010: Count on Pϕ/64 011: Count on Pϕ/256 100: Count on Pϕ/1024 Others: Setting prohibited

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

18.2.3 Timer Constant Registers (TCOR)

TCOR are 32-bit readable/writable registers that specify the value to be set in TCNT when TCNT underflows.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TCOR															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCOR															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.2.4 Timer Counters (TCNT)

TCNT count down upon input of a clock. The clock input is selected using bits TPSC[1:0] in TCR.

When a TCNT countdown results in an underflow (H'00000000 → H'FFFFFFFF), the underflow flag (UNF) in TCR that is corresponding to the channel is set. The TCOR value is simultaneously set in TCNT itself and the countdown continues from that value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TCNT															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNT															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.3 Operation

Each channel has a 32-bit TCNT and a 32-bit TCOR. TCNT counts down. The auto-reload function enables cyclic counting.

18.3.1 Counter Operation

When bits STR[2:0] in TSTR are set to 1, the corresponding TCNT starts counting. When TCNT underflows, the UNF flag in the corresponding TCR is set. At this moment, if the UNIE bit in TCR is 1, an interrupt request is sent to the CPU. Also, the value in TCOR is copied to TCNT and the count down operation continues.

(1) Count Operation Setting Procedure

An example procedure for setting the count operation is shown in figure 18.2.

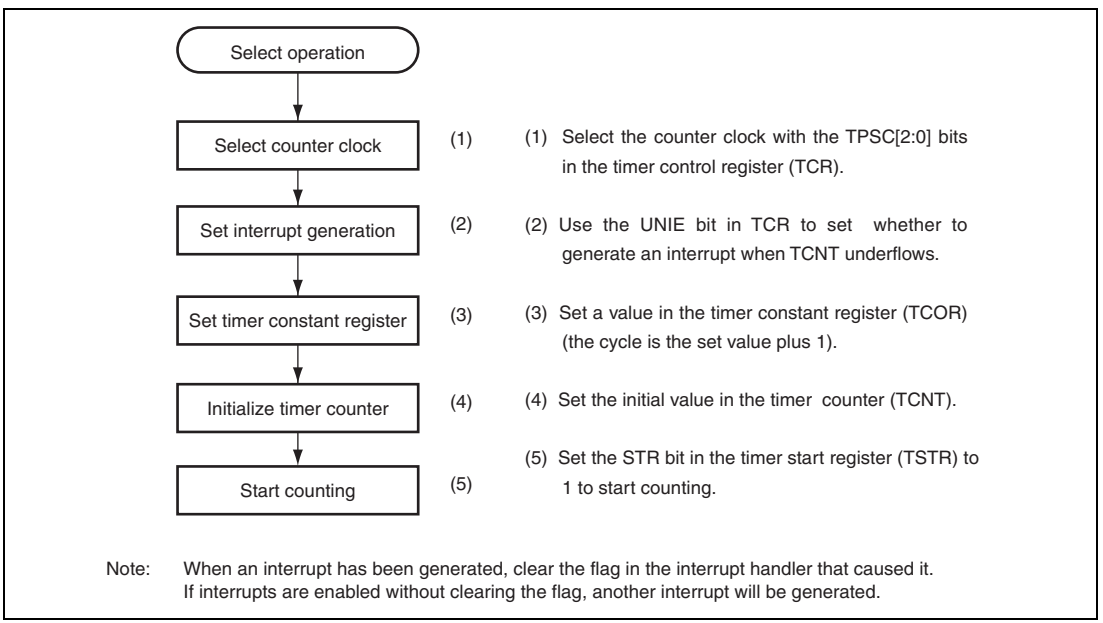


Figure 18.2 Setting Count Operation

(2) Auto-Reload Count Operation

Figure 18.3 shows the TCNT auto-reload operation.

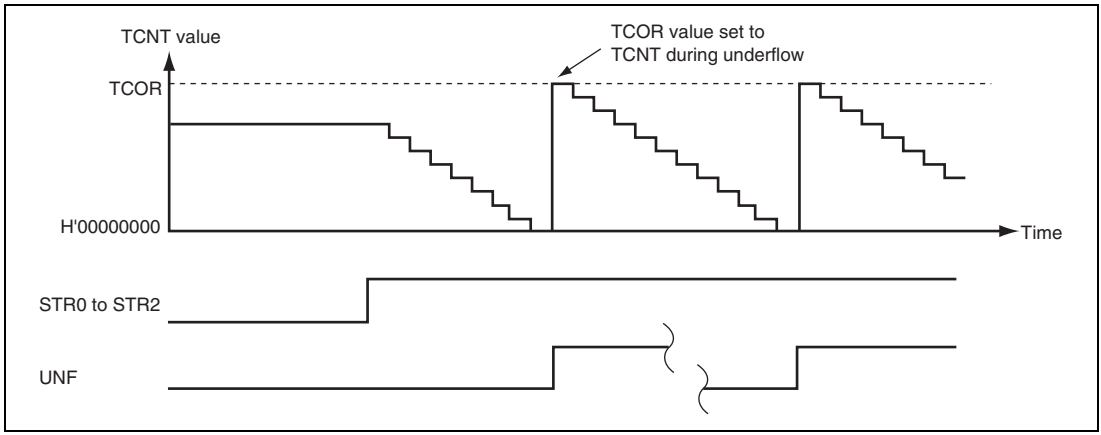


Figure 18.3 Auto-Reload Count Operation

(3) TCNT Count Timing

Setting the bits TPSC[1:0] in TCR allows to select one of the five internal clocks (Pϕ/4, Pϕ/16, Pϕ/64, Pϕ/256, and Pϕ/1024) that are generated by dividing the peripheral module clock. Figure 18.4 shows the timing.

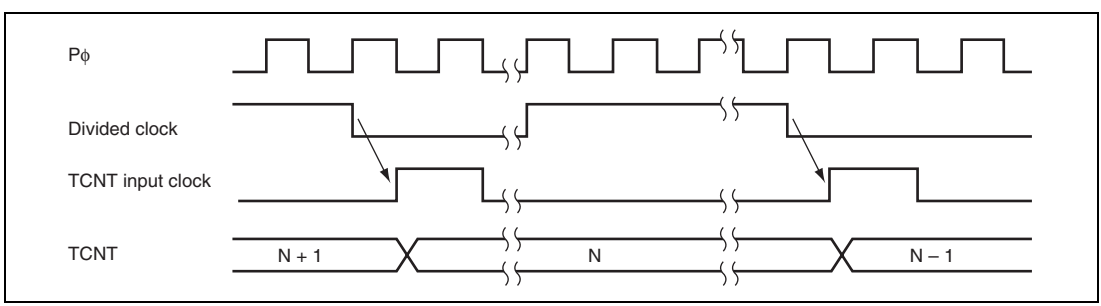


Figure 18.4 Count Timing when Internal Clock is Operating

18.4 Interrupts

There is one source of TMU interrupts: underflow interrupts (TUNI).

18.4.1 Status Flag Set Timing

The UNF bit is set to 1 when TCNT underflows. Figure 18.5 shows the timing.

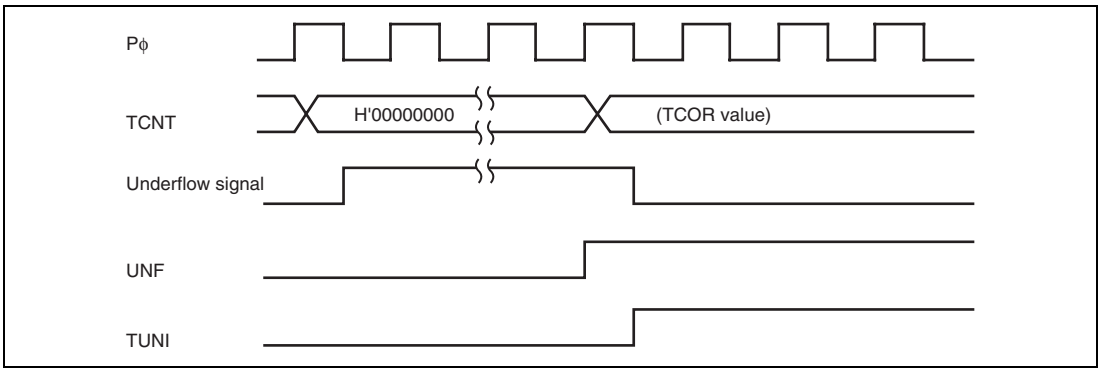


Figure 18.5 UNF Set Timing

18.4.2 Status Flag Clear Timing

The status flag can be cleared by writing 0 from the CPU. Figure 18.6 shows the timing.

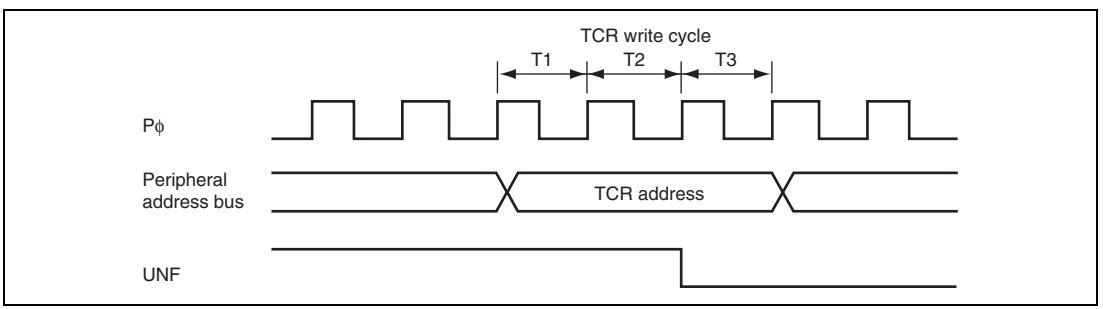


Figure 18.6 Status Flag Clear Timing

18.4.3 Interrupt Sources and Priorities

The TMU generates underflow interrupts for each channel. When the interrupt request flag and interrupt enable bit are both set to 1, the interrupt is requested. A specific code is set in the interrupt event register (INTEVT) for this interrupt and interrupt processing must be executed according to the code.

The priorities between channels are changeable by the interrupt controller. For details, see section 5, Exception Handling, and section 10, Interrupt Controller (INTC). Table 18.3 lists TMU interrupt sources.

Table 18.3 TMU Interrupt Sources

Channel	Interrupt Source	Description	Priority
0	TUNI0	Underflow interrupt 0	High
1	TUNI1	Underflow interrupt 1	↕
2	TUNI2	Underflow interrupt 2	Low

18.5 Usage Notes

18.5.1 Writing to Registers

Synchronization processing is not performed for timer counting during register writes. When writing to registers, be sure to clear the start bits (STR2 to STR0) of the channel in TSTR and halt the timer counting.

18.5.2 Reading Registers

Synchronization processing is performed for timer counting during register reads. When the timer counting and register read are performed simultaneously, the register value stored before the TCNT countdown is read through the synchronization processing.

Section 19 Compare Match Timer (CMT)

This LSI includes 5 channels (channels 0 to 4) of 32-bit compare match timers (CMT).

19.1 Features

- 16 bits/32 bits can be selected.
- Provided with an auto-reload up counter.
- Provided with 32-bit constant registers and 32-bit up counters that can be written or read at any time.
- For each of channels 0 to 4, the counter-input clock is selectable from among 3 signals.
  — Peripheral clock (Pϕ): 1/8, 1/32, and 1/128
- One-shot operation and free-running operation are selectable.
- Allows selection of compare match or overflow for the interrupt source.
- Issuing of DMA transfer requests on compare match or overflow of the counter is selectable on channels 0 to 4.
- Module standby mode can be set.

Figure 19.1 shows a block diagram of the CMT.

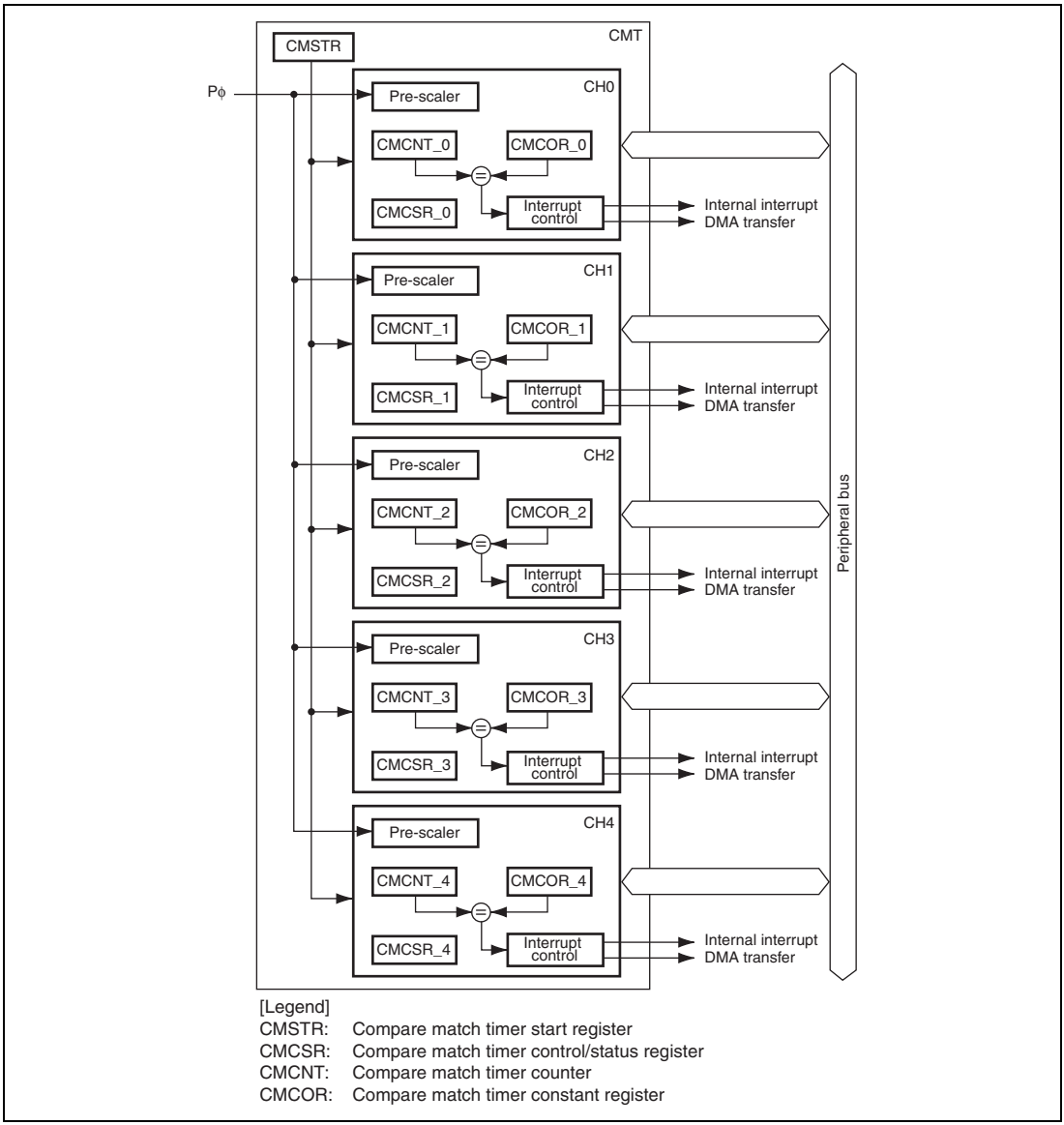


Figure 19.1 Block Diagram of CMT

19.2 Input/output Pins

The CMT has no input/output pins.

19.3 Register Descriptions

Table 19.1 shows the CMT register configuration. Table 19.2 shows the register states in each operating mode. Note that the channel numbers are omitted from the register names in descriptions subsequent to these tables.

Table 19.1 Register Configuration

Channel	Register Name	Abbreviation	R/W	Address	Access Size
Common	Compare match timer start register	CMSTR	R/W	H'A44A 0000	16
0	Compare match timer control/status register_0	CMCSR_0	R/W	H'A44A 0010	16
	Compare match timer counter_0	CMCNT_0	R/W	H'A44A 0014	32
	Compare match timer constant register_0	CMCOR_0	R/W	H'A44A 0018	32
1	Compare match timer control/status register_1	CMCSR_1	R/W	H'A44A 0020	16
	Compare match timer counter_1	CMCNT_1	R/W	H'A44A 0024	32
	Compare match timer constant register_1	CMCOR_1	R/W	H'A44A 0028	32
2	Compare match timer control/status register_2	CMCSR_2	R/W	H'A44A 0030	16
	Compare match timer counter_2	CMCNT_2	R/W	H'A44A 0034	32
	Compare match timer constant register_2	CMCOR_2	R/W	H'A44A 0038	32
3	Compare match timer control/status register_3	CMCSR_3	R/W	H'A44A 0040	16
	Compare match timer counter_3	CMCNT_3	R/W	H'A44A 0044	32
	Compare match timer constant register_3	CMCOR_3	R/W	H'A44A 0048	32
4	Compare match timer control/status register_4	CMCSR_4	R/W	H'A44A 0050	16
	Compare match timer counter_4	CMCNT_4	R/W	H'A44A 0054	32
	Compare match timer constant register_4	CMCOR_4	R/W	H'A44A 0058	32

Table 19.2 Register States in Each Operating Mode

Channel	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
Common	CMSTR	Initialized	Retained	Retained	Retained
0	CMCSR_0	Initialized	Retained	Retained	Retained
	CMCNT_0	Initialized	Retained	Retained	Retained
	CMCOR_0	Initialized	Retained	Retained	Retained
1	CMCSR_1	Initialized	Retained	Retained	Retained
	CMCNT_1	Initialized	Retained	Retained	Retained
	CMCOR_1	Initialized	Retained	Retained	Retained
2	CMCSR_2	Initialized	Retained	Retained	Retained
	CMCNT_2	Initialized	Retained	Retained	Retained
	CMCOR_2	Initialized	Retained	Retained	Retained
3	CMCSR_3	Initialized	Retained	Retained	Retained
	CMCNT_3	Initialized	Retained	Retained	Retained
	CMCOR_3	Initialized	Retained	Retained	Retained
4	CMCSR_4	Initialized	Retained	Retained	Retained
	CMCNT_4	Initialized	Retained	Retained	Retained
	CMCOR_4	Initialized	Retained	Retained	Retained

19.3.1 Compare Match Timer Start Register (CMSTR)

CMSTR is a 16-bit register that selects whether the individual compare match timer counters (CMCNT_4 to CMCNT_0) operate or are halted.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	STR[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	STR[4:0]	All 0	R/W	Count Start Selects whether or not the individual compare match timer counters (CMCNT_4 to CMCNT_0) operate or are halted. 0: Halts counting by CMCNTn. 1: Runs counting by CMCNTn. Note: n = 0 to 4 (channel number).

19.3.2 Compare Match Timer Control/Status Register (CMCSR)

Each CMCSR_n (n = 0 to 4) is a 16-bit register that indicates the occurrence of compare match or overflow events for CMCNT_n, enables interrupts and DMA transfer requests, and sets up the counter input clocks.

Do not change bits other than the CMF and OVF bits during the compare match timer counter (CMCNT) operation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMF	OVF	—	—	—	—	CMS	CMM	—	—	CMR[1:0]		—	CKS[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	CMF	0	R/(W)*	Compare Match Flag This flag indicates whether or not values of the compare match timer counter (CMCNT) and compare match timer constant register (CMCOR) have matched. Software cannot write 1 to the bit. When one-shot is selected for the counter operation, counting resumes by clearing this bit. 0: CMCNT and CMCOR values have not matched [Clearing condition] Writing 0 to CMF after having read CMF = 1 1: CMCNT and CMCOR values have matched
14	OVF	0	R/(W)*	Overflow Flag This flag indicates whether or not the compare match timer counter (CMCNT) has overflowed and been cleared to 0. Software cannot write 1 to this bit. 0: CMCNT has not overflowed [Clearing condition] Writing 0 to OVF after having read OVF = 1 1: CMCNT has overflowed
13 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	CMS	0	R/W	Compare Match Timer Counter Size Selects whether the compare match timer counter (CMCNT) is used as a 16-bit counter or a 32-bit counter. This setting becomes the valid size for the compare match timer constant register (CMCOR). 0: Operates as a 32-bit counter 1: Operates as a 16-bit counter
8	CMM	0	R/W	Compare Match Mode Selects one-shot operation or free-running operation of the counter. 0: One-shot operation 1: Free-running operation

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	CMR[1:0]	00	R/W	Compare Match Request These bits enable or disable generation of DMA transfer requests or internal interrupt requests on a compare match. 00: Disables both DMA transfer requests and internal interrupt requests 01: Enables DMA transfer requests 10: Enables internal interrupt requests 11: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	CKS[2:0]	000	R/W	Clock Select These bits select the clock signal for input to CMCNT. When the start-counting bit (STRn: n=4 to 0) of the corresponding channel is set to 1, incrementation of CMCNT by cycles of the clock signal selected by these bits proceeds. 000: Pϕ/8 001: Pϕ/32 010: Pϕ/128 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

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

19.3.3 Compare Match Timer Counter (CMCNT)

CMCNT is a 32-bit register that is used as an up-counter.

A counter operation is set by the compare match timer control/status register (CMCSR). Therefore, set CMCSR first, before starting a channel operation corresponding to the compare match timer start register (CMSTR). When the 16-bit counter operation is selected by the CMS bit, bits 15 to 0 of this register become valid. When the register should be written to, write the data that is added H'0000 to the upper half in a 32-bit operation. The contents of this register are initialized to H'00000000.

19.3.4 Compare Match Timer Constant Register (CMCOR)

CMCOR is a 32-bit register that sets the compare match period with CMCNT.

When the 16-bit counter operation is selected by the CMS bit in CMCSR, bits 15 to 0 of this register become valid. When the register should be written to, write the data that is added H'0000 to the upper half in a 32-bit operation.

An overflow is detected when CMCNT is cleared to 0 and this register is H'FFFFFFFF. The contents of this register are initialized to H'FFFFFFFF.

19.4 Operation

19.4.1 Counter Operation

The operation of CMT counter n (n= 0 to 4) is commenced by writing 1 to the corresponding STRn bit in CMSTR after making other register settings as required. Complete all of the settings before starting counter operation. Do not change the register settings other than by clearing flag bits.

The counter operates in one of two ways.

- One-Shot Operation

  One-shot operation is selected by clearing the CMM bit in CMCSR to 0. When the value in CMCNT matches the value in CMCOR, the value in CMCNT is cleared to H'00000000 and the CMF bit in CMCSR is set to 1. Counting by CMCNT stops after it has been cleared.

  To detect an overflow interrupt, set the value in CMCOR to H'FFFFFFFF. When the value in CMCNT matches the value in CMCOR, CMCNT is cleared to H'00000000 and the CMF and OVF bits in CMCSR are set to 1.

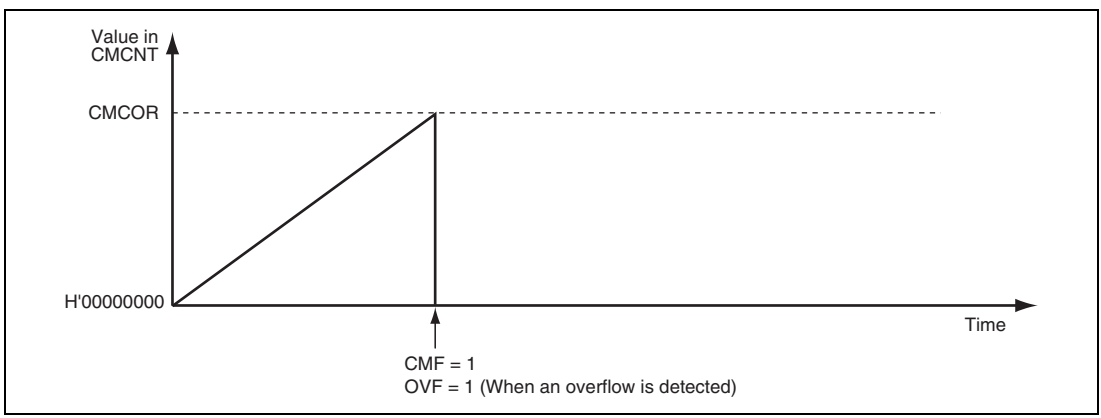


Figure 19.2 Counter Operation (One-Shot Operation)

- Free-Running Operation

  Free-running operation is selected by setting the CMM bit in CMCSR to 1. When the value in CMCNT matches the value in CMCOR, CMCNT is cleared to H'00000000 and the CMF bit in CMCSR is set to 1. CMCNT resumes counting-up after it has been cleared.

To detect an overflow interrupt, set CMCOR to H'FFFFFFFF. When the values in CMCNT and CMCOR match, CMCNT is cleared to H'00000000 and the CMF and OVF bits in CMCSR are set to 1.

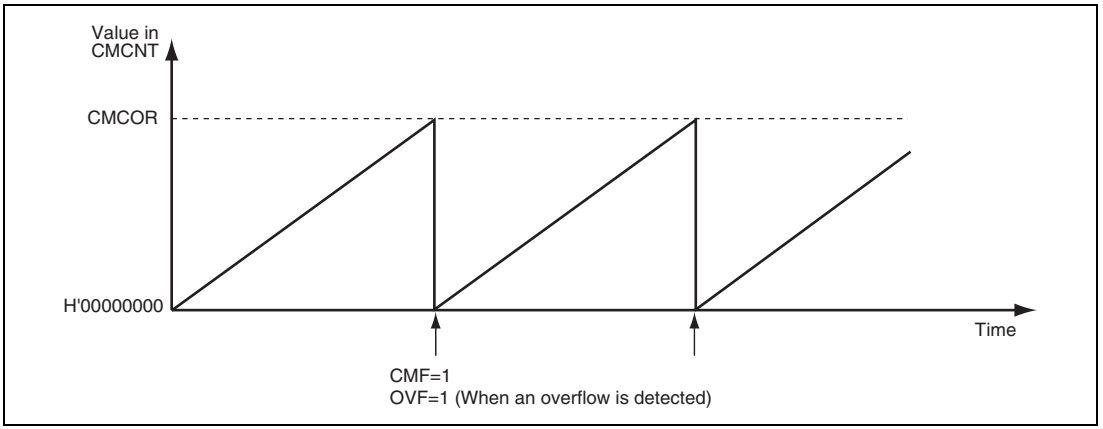


Figure 19.3 Counter Operation (Free-Running Operation)

19.4.2 Counter Size

In this module, the size of the counter is selectable as either 16 or 32 bits. This is selected by the CMS bit in CMCSR.

When the 16-bit size is selected, use a 32-bit value which has H'0000 as its upper half to set CMCOR.

To detect an overflow interrupt, the value must be set to H'0000FFFF.

19.4.3 Timing for Counting by CMCNT

In this module, the clock signal for each counter can be selected from among the following:

- Peripheral clock (Pϕ) scaling for CMCNT_0 to CMCNT_4: 1/8, 1/32, and 1/128

The clock signal for each counter is selected by the CKS[2:0] bits in CMCSR. CMCNT is incremented on rising edges of the selected clock.

19.4.4 DMA Transfer Requests and Internal Interrupt Requests for the CPU

The setting of the CMR[1:0] bits in CMCSR selects generation of DMA transfer requests (or an internal interrupt request for the CPU) on a compare match.

The specification of DMA transfer requests varies with the CMT channel as stated below.

1. Channels 0 and 1 send one DMA request per compare match event.
2. Channels 2 to 4 keep sending each request until the number of transfers set in the DMAC is complete, and automatically stop sending the request when the set number of transfers is complete.

To cancel an internal interrupt request for the CPU, clear the CMF bit to 0. Do this in the handler for CMT interrupts.

19.4.5 Compare Match Flag Set Timing

The CMF bit in CMCSR is set to 1 by the compare match signal generated when CMCOR and CMCNT match. The compare match signal is generated upon the final state of the match (timing at which the CMCNT value is updated to H'0000). Consequently, after CMCOR and CMCNT match, a compare match signal will not be generated until a CMCNT counter clock is input. Figure 19.4 shows the set timing of the CMF bit.

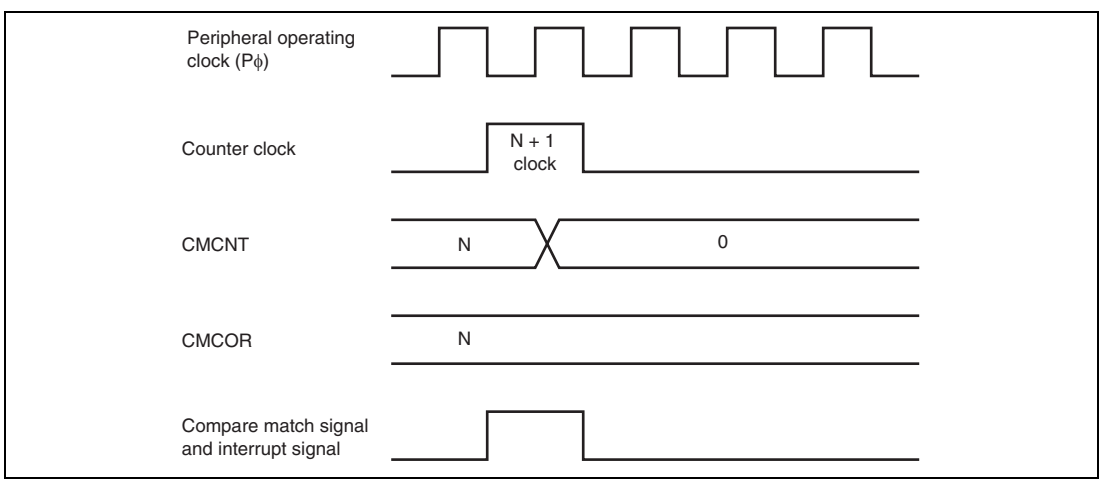


Figure 19.4 CMF Set Timing

Section 20 I²C Bus Interface (IIC)

The I²C bus interface supports and provides a subset of the Philips I²C (Inter-IC) bus interface functions. However, the configuration of the registers that control the I²C bus differs partly from the Philips register configuration.

The I²C bus interface has 2 channels.

20.1 Features

- Supports master mode and slave mode
- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independent from each other, the continuous transmission/reception can be performed.
- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization function

  In master mode, the state of SCL is monitored per bit, and the timing is synchronized automatically. If transmission/reception is not yet possible, set the SCL to low until preparations are completed.
- Six interrupt sources

  Transmit data empty (including slave-address match), transmit end, receive data full (including slave-address match), arbitration lost, NACK detection, and stop condition detection
- Direct bus drive

  Two pins, SCL and SDA pins, function as NMOS open-drain outputs when the bus drive function is selected.

Figure 20.1 shows a block diagram of the I²C bus interface.

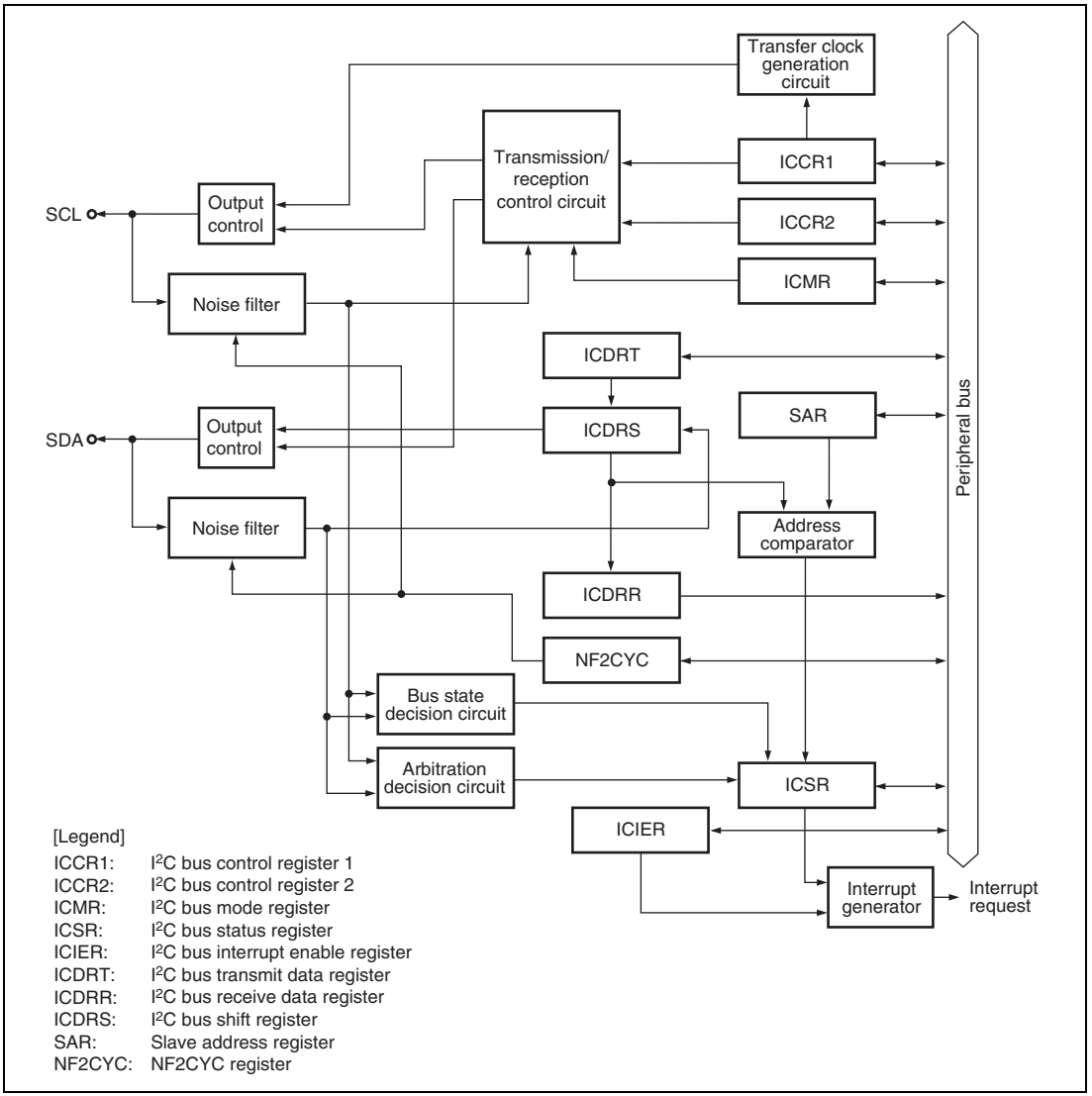


Figure 20.1 Block Diagram of I²C Bus Interface

20.2 Input/Output Pins

Table 20.1 shows the pin configuration of the I²C bus interface.

Table 20.1 Pin Configuration

Channel	Pin Name	I/O	Function
0	IIC0_SCL	I/O	I²C serial clock input/output
	IIC0_SDA	I/O	I²C serial data input/output
1	IIC1_SCL	I/O	I²C serial clock input/output
	IIC1_SDA	I/O	I²C serial data input/output

Note: The interface numbers are abbreviated and the respective sets of pins are collectively denoted by ACL and SDL.

Figure 20.2 shows an example of I/O pin connections to external circuits.

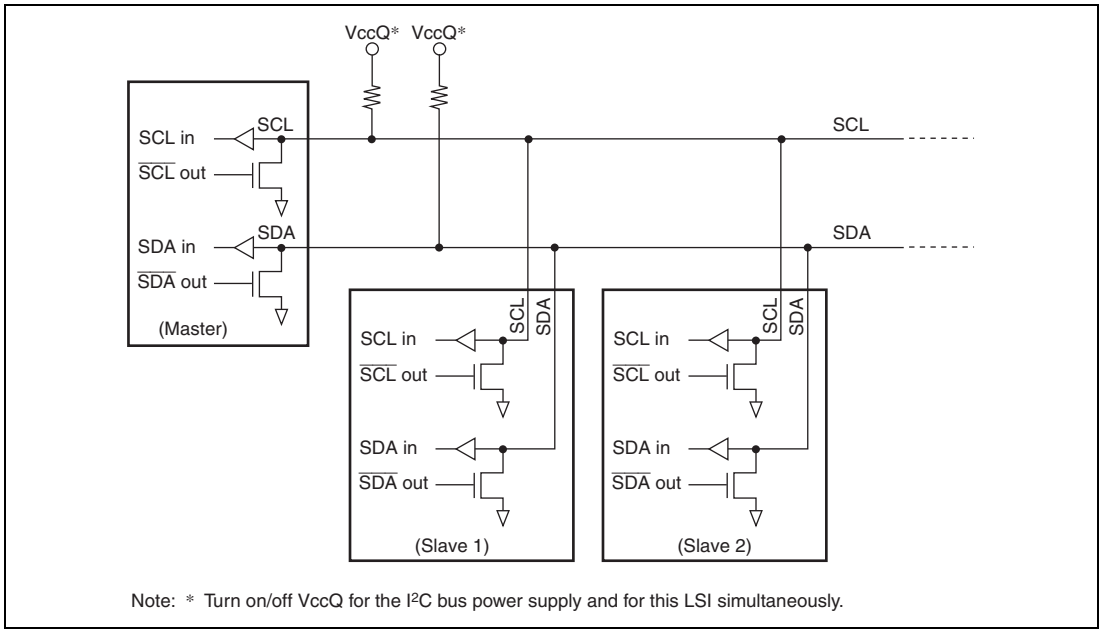


Figure 20.2 External Circuit Connections of I/O Pins

20.3 Register Descriptions

Table 20.2 shows the register configuration. Table 20.3 shows the register states in each operating mode. Note that the channel numbers are omitted from the register names in descriptions subsequent to these tables.

Table 20.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Address	Access Size
0	I²C bus control register 1	ICCR1_0	R/W	H'A447 0000	8
	I²C bus control register 2	ICCR2_0	R/W	H'A447 0001	8
	I²C bus mode register	ICMR_0	R/W	H'A447 0002	8
	I²C bus interrupt enable register	ICIER_0	R/W	H'A447 0003	8
	I²C bus status register	ICSR_0	R/W	H'A447 0004	8
	Slave address register	SAR_0	R/W	H'A447 0005	8
	I²C bus transmit data register	ICDRT_0	R/W	H'A447 0006	8
	I²C bus receive data register	ICDRR_0	R	H'A447 0007	8
	NF2CYC register	NF2CYC_0	R/W	H'A447 0008	8
1	I²C bus control register 1	ICCR1_1	R/W	H'A475 0000	8
	I²C bus control register 2	ICCR2_1	R/W	H'A475 0001	8
	I²C bus mode register	ICMR_1	R/W	H'A475 0002	8
	I²C bus interrupt enable register	ICIER_1	R/W	H'A475 0003	8
	I²C bus status register	ICSR_1	R/W	H'A475 0004	8
	Slave address register	SAR_1	R/W	H'A475 0005	8
	I²C bus transmit data register	ICDRT_1	R/W	H'A475 0006	8
	I²C bus receive data register	ICDRR_1	R	H'A475 0007	8
	NF2CYC register	NF2CYC_1	R/W	H'A475 0008	8

Table 20.3 Register States in Each Operating Mode

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
ICCR1_0	Initialized	Retained	Retained	Retained
ICCR2_0	Initialized	Retained	Retained	Retained
ICMR_0	Initialized	Retained	Retained	Retained
ICIER_0	Initialized	Retained	Retained	Retained
ICSR_0	Initialized	Retained	Retained	Retained
SAR_0	Initialized	Retained	Retained	Retained
ICDRT_0	Initialized	Retained	Retained	Retained
ICDRR_0	Initialized	Retained	Retained	Retained
NF2CYC_0	Initialized	Retained	Retained	Retained
ICCR1_1	Initialized	Retained	Retained	Retained
ICCR2_1	Initialized	Retained	Retained	Retained
ICMR_1	Initialized	Retained	Retained	Retained
ICIER_1	Initialized	Retained	Retained	Retained
ICSR_1	Initialized	Retained	Retained	Retained
SAR_1	Initialized	Retained	Retained	Retained
ICDRT_1	Initialized	Retained	Retained	Retained
ICDRR_1	Initialized	Retained	Retained	Retained
NF2CYC_1	Initialized	Retained	Retained	Retained

20.3.1 I²C Bus Control Register 1 (ICCR1)

ICCR1 is an 8-bit readable/writable register that enables or disables the I²C bus interface, controls transmission or reception, and selects master or slave mode, transmission or reception, and transfer clock frequency in master mode.

Bit:	7	6	5	4	3	2	1	0
	ICE	RCVD	MST	TRS	CKS[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface Enable 0: This module is halted. 1: This module is enabled for transfer operations.
6	RCVD	0	R/W	Reception Disable Enables or disables the next operation when TRS is 0 and ICDRR is read. 0: Enables next reception 1: Disables next reception
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select In master mode with the I²C bus format, when arbitration is lost, MST and TRS are both reset by hardware, causing a transition to slave receive mode. Modification of the TRS bit should be made between transfer frames. When seven bits after the start condition is issued in slave receive mode match the slave address set to SAR and the 8th bit is set to 1, TRS is automatically set to 1. Operating modes are selected as below according to MST and TRS combination. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	CKS[3:0]	0000	R/W	Transfer Clock Select These bits should be set according to the necessary transfer rate (table 20.4) in master mode.

Table 20.4 Transfer Rate

Bit 3	Bit 2	Bit 1	Bit 0		Transfer Rate (kHz)				
CKS3	CKS2	CKS1	CKS0	Clock	Pϕ = 16.7 MHz	Pϕ = 20.0 MHz	Pϕ = 25.0 MHz	Pϕ = 26.7 MHz	Pϕ = 33.3 MHz
0	0	0	0	Pϕ/44	379 kHz	455 kHz	568 kHz	606 kHz	758 kHz
			1	Pϕ/52	321 kHz	385 kHz	481 kHz	513 kHz	641 kHz
		1	0	Pϕ/64	260 kHz	313 kHz	391 kHz	417 kHz	521 kHz
			1	Pϕ/72	231 kHz	278 kHz	347 kHz	370 kHz	463 kHz
	1	0	0	Pϕ/84	198 kHz	238 kHz	298 kHz	317 kHz	397 kHz
			1	Pϕ/92	181 kHz	217 kHz	272 kHz	290 kHz	362 kHz
		1	0	Pϕ/100	167 kHz	200 kHz	250 kHz	267 kHz	333 kHz
			1	Pϕ/108	Setting prohibited				
1	0	0	0	Pϕ/176	94.7 kHz	114 kHz	142 kHz	152 kHz	189 kHz
			1	Pϕ/208	80.1 kHz	96.2 kHz	120 kHz	128 kHz	160 kHz
		1	0	Pϕ/256	65.1 kHz	78.1 kHz	97.7 kHz	104 kHz	130 kHz
			1	Pϕ/288	57.9 kHz	69.4 kHz	86.8 kHz	92.6 kHz	116 kHz
	1	0	0	Pϕ/336	49.6 kHz	59.5 kHz	74.4 kHz	79.4 kHz	99.2 kHz
			1	Pϕ/368	45.3 kHz	54.3 kHz	67.9 kHz	72.5 kHz	90.6 kHz
		1	0	Pϕ/400	41.7 kHz	50.0 kHz	62.5 kHz	66.7 kHz	83.3 kHz
			1	Pϕ/432	Setting prohibited				

Note: The settings should satisfy external specifications.

20.3.2 I²C Bus Control Register 2 (ICCR2)

ICCR2 is an 8-bit readable/writable register that issues start/stop conditions, manipulates the SDA pin, monitors the SCL pin, and controls reset in the control part of the I²C bus.

Bit:	7	6	5	4	3	2	1	0
	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—
Initial value:	0	1	1	1	1	1	0	1
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	BBSY	0	R/W	Bus Busy Enables to confirm whether the I²C bus is occupied or released and to issue start/stop conditions in master mode. With the I²C bus format, this bit is set to 1 when the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. Follow this procedure when also re-transmitting a start condition. Write 0 in BBSY and 0 in SCP to issue a stop condition.
6	SCP	1	R/W	Start/Stop Issue Condition Disable Controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. Even if 1 is written to this bit, the data will not be stored.

Bit	Bit Name	Initial Value	R/W	Description
5	SDAO	1	R/W	SDA Output Value Control This bit is used with SDAOP when modifying output level of SDA. This bit should not be manipulated during transfer. 0: When reading, SDA pin outputs low. When writing, SDA pin is changed to output low. 1: When reading, SDA pin outputs high. When writing, SDA pin is changed to output Hi-Z (outputs high by external pull-up resistance).
4	SDAOP	1	R/W	SDAO Write Protect Controls change of output level of the SDA pin by modifying the SDAO bit. To change the output level, clear SDAO and SDAOP to 0 or set SDAO to 1 and clear SDAOP to 0. This bit is always read as 1.
3	SCLO	1	R	SCL Output Level Monitors SCL output level. When SCLO is 1, SCL pin outputs high. When SCLO is 0, SCL pin outputs low.
2	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
1	IICRST	0	R/W	IIC Control Part Reset Resets the control part except for I²C registers. If the device hangs because of a problem such as a communication failure during I²C bus operation, bits BC2 to BC0 in the ICMR register of the IIC and the internal circuits of the IIC can be reset by setting the IICRST bit to 1.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

20.3.3 I²C Bus Mode Register (ICMR)

ICMR is an 8-bit readable/writable register that selects whether the MSB or LSB is transferred first, performs master mode wait control, and selects the transfer bit count.

Bits BC[2:0] are initialized to H'0 by the IICRST bit in ICCR2.

Bit:	7	6	5	4	3	2	1	0
	MLS	—	—	—	BCWP	BC[2:0]		
Initial value:	0	0	1	1	1	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I²C bus format is used.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	BCWP	1	R/W	BC Write Protect Controls the modification of BC[2:0]. Clear this bit to 0 and then use an MOV instruction to modify the BC[2:0] bits. 0: When writing, values of the BC[2:0] bits are set. 1: When reading, 1 is always read. When writing, settings of the BC[2:0] bits are invalid.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	BC[2:0]	000	R/W	Bit Counter These bits specify the number of bits to be transferred next. When read, the remaining number of transfer bits is indicated. With the I²C bus format, the data is transferred with one addition acknowledge bit. Should be made between transfer frames. If these bits are set to a value other than B'000, the setting should be made while the SCL pin is low. The value returns to B'000 at the end of a data transfer, including the acknowledge bit. These bits are cleared by a power-on reset and in software standby mode and module standby mode. These bits are also cleared by setting the IICRST bit of ICCR2 to 1. 000: 9 bits 001: 2 bits 010: 3 bits 011: 4 bits 100: 5 bits 101: 6 bits 110: 7 bits 111: 8 bits

20.3.4 I²C Bus Interrupt Enable Register (ICIER)

ICIER is an 8-bit readable/writable register that enables or disables interrupt sources and acknowledge bits, sets acknowledge bits to be transferred, and confirms acknowledge bits received.

Bit:	7	6	5	4	3	2	1	0
	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1 or 0, this bit enables or disables the transmit data empty interrupt (TXI). 0: Transmit data empty interrupt request (TXI) is disabled. 1: Transmit data empty interrupt request (TXI) is enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by clearing the TEND bit or the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabled. 1: Transmit end interrupt request (TEI) is enabled.
5	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive data full interrupt request (RXI) when receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) is disabled. 1: Receive data full interrupt request (RXI) is enabled.

Bit	Bit Name	Initial Value	R/W	Description
4	NAKIE	0	R/W	NACK Receive Interrupt Enable Enables or disables the NACK detection interrupt request (NAKI) when the NACKF or AL/OVE bit in ICSR is set. NAKI can be canceled by clearing the NACKF, AL/OVE, or NAKIE bit to 0. 0: NACK receive interrupt request (NAKI) is disabled. 1: NACK receive interrupt request (NAKI) is enabled.
3	STIE	0	R/W	Stop Condition Detection Interrupt Enable Enables or disables the stop condition detection interrupt request (STPI) when the STOP bit in ICSR is set. 0: Stop condition detection interrupt request (STPI) is disabled. 1: Stop condition detection interrupt request (STPI) is enabled.
2	ACKE	0	R/W	Acknowledge Bit Judgment Select 0: The value of the receive acknowledge bit is ignored, and continuous transfer is performed. 1: If the receive acknowledge bit is 1, continuous transfer is halted.
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowledge data that are returned by the receive device. This bit cannot be modified. This bit can be canceled by setting the BBSY bit in ICCR2 to 1. 0: Receive acknowledge = 0 1: Receive acknowledge = 1
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, this bit specifies the bit to be sent at the acknowledge timing. 0: 0 is sent at the acknowledge timing. 1: 1 is sent at the acknowledge timing.

20.3.5 I²C Bus Status Register (ICSR)

ICSR is an 8-bit readable/writable register that confirms interrupt request flags and their status.

Bit:	7	6	5	4	3	2	1	0
	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

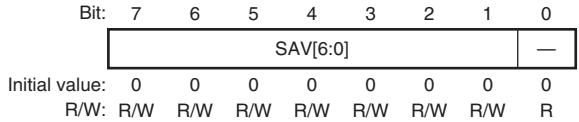
Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	0	R/W	Transmit Data Register Empty [Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1 • When data is written to ICDRT [Setting conditions] • When data is transferred from ICDRT to ICDRS and ICDRT becomes empty • When TRS is set • When the start condition (including retransmission) is issued • When slave mode is changed from receive mode to transmit mode
6	TEND	0	R/W	Transmit End [Clearing conditions] • When 0 is written in TEND after reading TEND = 1 • When data is written to ICDRT [Setting condition] • When the ninth clock of SCL rises with the I²C bus format while the TDRE flag is 1

Bit	Bit Name	Initial Value	R/W	Description
5	RDRF	0	R/W	Receive Data Full [Clearing conditions] • When 0 is written in RDRF after reading RDRF = 1 • When ICDRR is read [Setting condition] • When a receive data is transferred from ICDRS to ICDRR
4	NACKF	0	R/W	No Acknowledge Detection Flag [Clearing condition] • When 0 is written in NACKF after reading NACKF = 1 [Setting condition] • When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1
3	STOP	0	R/W	Stop Condition Detection Flag [Clearing condition] • When 0 is written in STOP after reading STOP = 1 [Setting conditions] • In master mode, when a stop condition is detected after frame transfer • In slave mode, when STOP condition is detected after the first byte slave address, next to detection of start condition, accords with the address set in SAR.

Bit	Bit Name	Initial Value	R/W	Description
2	AL/OVE	0	R/W	Arbitration Lost Flag/Overrun Error Flag Indicates that arbitration was lost in master mode with the I²C bus format. When two or more master devices attempt to seize the bus at nearly the same time, if the I²C bus interface detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been occupied by another master. [Clearing condition] • When 0 is written in AL/OVE after reading AL/OVE = 1 [Setting conditions] • If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode • When the SDA pin outputs high in master mode while a start condition is detected
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA[6:0] in SAR. [Clearing condition] • When 0 is written in AAS after reading AAS = 1 [Setting conditions] • When the slave address is detected in slave receive mode • When the general call address is detected in slave receive mode.
0	ADZ	0	R/W	General Call Address Recognition Flag This bit is valid in slave receive mode with the I²C bus format. [Clearing condition] • When 0 is written in ADZ after reading ADZ = 1 [Setting condition] • When the general call address is detected in slave receive mode

20.3.6 Slave Address Register (SAR)

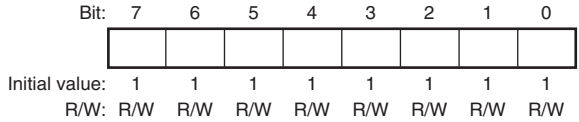
SAR is an 8-bit readable/writable register that selects the communications format and sets the slave address. In slave mode with the I²C bus format, if the upper seven bits of SAR match the upper seven bits of the first frame received after a start condition, this module operates as the slave device.



Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA[6:0]	0000000	R/W	Slave Address These bits set a unique address in these bits, differing form the addresses of other slave devices connected to the I²C bus.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

20.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. If the next transfer data is written to ICDRT while transferring data of ICDRS, continuous transfer is possible.



20.3.8 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit register that stores the receive data. When data of one byte is received, ICDRR transfers the receive data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register, therefore the CPU cannot write to this register.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R

20.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred from ICDRS to ICDRR after data of one byte is received. This register cannot be read directly from the CPU.

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	—	—	—	—	—	—	—	—

20.3.10 NF2CYC Register (NF2CYC)

NF2CYC is an 8-bit readable/writable register that selects the range of the noise filtering for the SCL and SDA pins. For details of the noise filter, see section 20.4.6, Noise Filter.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PRS	NF2CYC
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PRS	0	R/W	Pulse Width Ratio Select Specifies the ratio of the high-level period to the low-level period for the SCL signal. 0: The ratio of high to low is 0.5 to 0.5. 1: The ratio of high to low is about 0.4 to 0.6.
0	NF2CYC	0	R/W	Noise Filtering Range Select 0: The noise less than one cycle of the peripheral clock can be filtered out 1: The noise less than two cycles of the peripheral clock can be filtered out

20.4 Operation

20.4.1 I²C Bus Format

Figure 20.3 shows the I²C bus formats. Figure 20.4 shows the I²C bus timing. The first frame following a start condition always consists of eight bits.

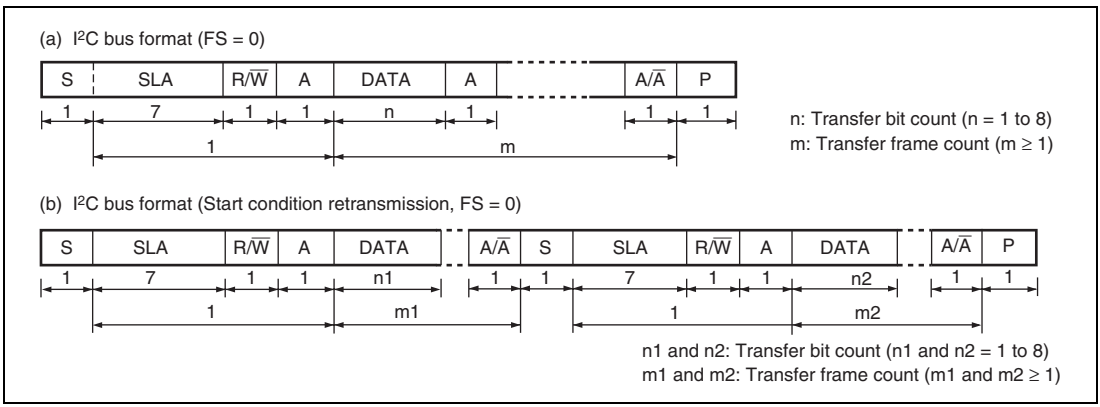


Figure 20.3 I²C Bus Formats

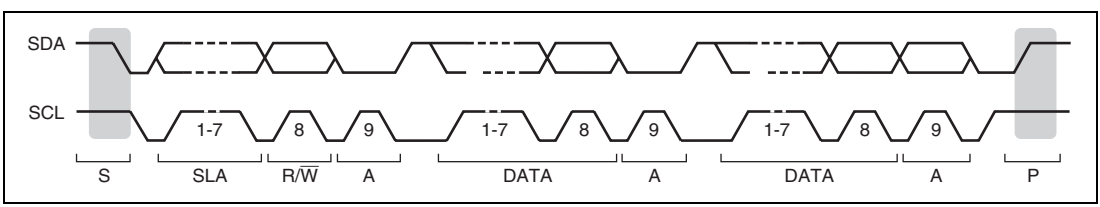


Figure 20.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

R/W: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receive device drives SDA to low.

DATA: Transfer data

P: Stop condition. The master device drives SDA from low to high while SCL is high.

20.4.2 Master Transmit Operation

In master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For master transmit mode operation timing, refer to figures 20.5 and 20.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Also, set bits CKS[3:0] in ICCR1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is released. Set the MST and TRS bits in ICCR1 to select master transmit mode. Then, write 1 to BBSY and 0 to SCP. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first byte data show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cleared to 0, and data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR is set to 1 at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confirm that the slave device has been selected. Then, write second byte data to ICDRT. When ACKBR is 1, the slave device has not been acknowledged, so issue the stop condition. To issue the stop condition, write 0 to BBSY and SCP. SCL is fixed low until the transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the end of last byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) from the receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TEND or NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mode.

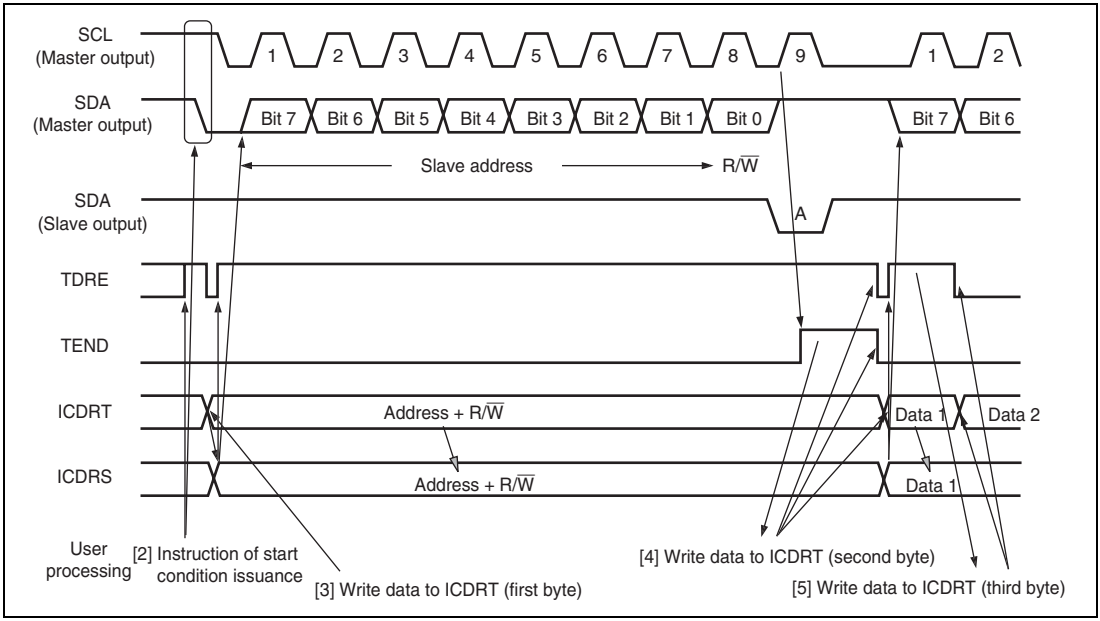


Figure 20.5 Master Transmit Mode Operation Timing (1)

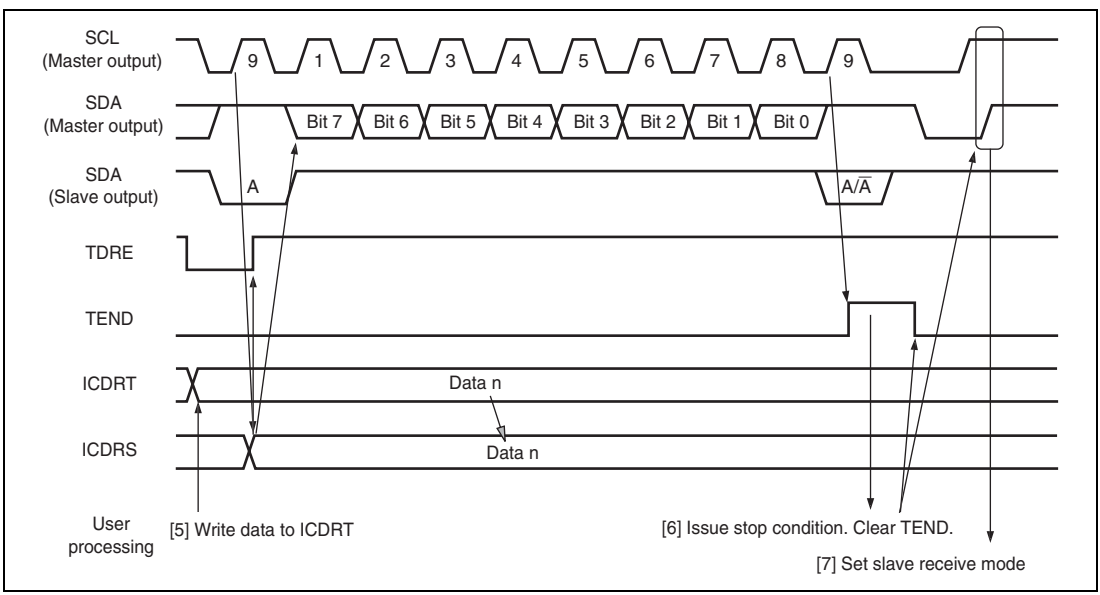


Figure 20.6 Master Transmit Mode Operation Timing (2)

20.4.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data from the slave device, and returns an acknowledge signal. For master receive mode operation timing, refer to figures 20.7 and 20.8. The reception procedure and operations in master receive mode are shown below.

1. Clear the TEND bit in ICSR to 0, then clear the TRS bit in ICCR1 to 0 to switch from master transmit mode to master receive mode. Then, clear the TDRE bit to 0.
2. When ICDRR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. The master device outputs the level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.
3. After the reception of first frame data is completed, the RDRF bit in ICSR is set to 1 at the rise of 9th receive clock pulse. At this time, the receive data is read by reading ICDRR, and RDRF is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. If 8th receive clock pulse falls after reading ICDRR by the other processing while RDRF is 1, SCL is fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1 before reading ICDRR. This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, issue the stage condition.
7. When the STOP bit in ICSR is set to 1, read ICDRR. Then clear the RCVD bit to 0.
8. The operation returns to the slave receive mode.

Note: If only one byte is received, read ICDRR (dummy-read) after the RCVD bit in ICCR1 is set.

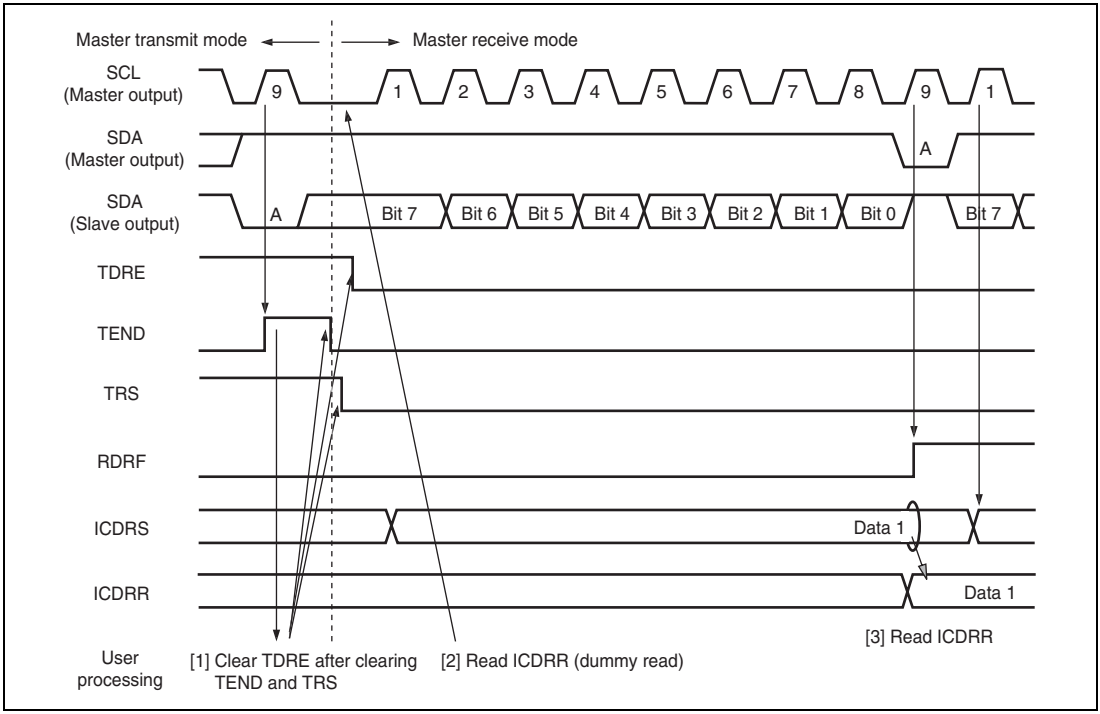


Figure 20.7 Master Receive Mode Operation Timing (1)

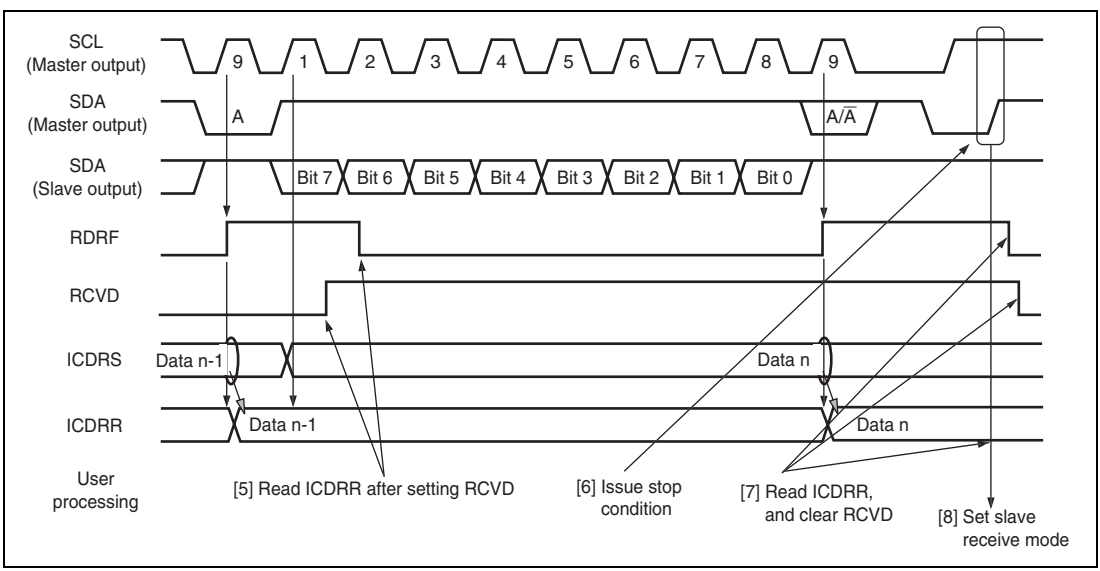


Figure 20.8 Master Receive Mode Operation Timing (2)

20.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. For slave transmit mode operation timing, refer to figures 20.9 and 20.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At this time, if the 8th bit data (R/W) is 1, the TRS bit in ICCR1 and the TDRE bit in ICSR are set to 1, and the mode changes to slave transmit mode automatically. The continuous transmission is performed by writing transmit data to ICDRT every time TDRE is set.
3. If TDRE is set after writing last transmit data to ICDRT, wait until TEND in ICSR is set to 1, with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for the end processing, and read ICDRR (dummy read). SCL is opened.
5. Clear TDRE.

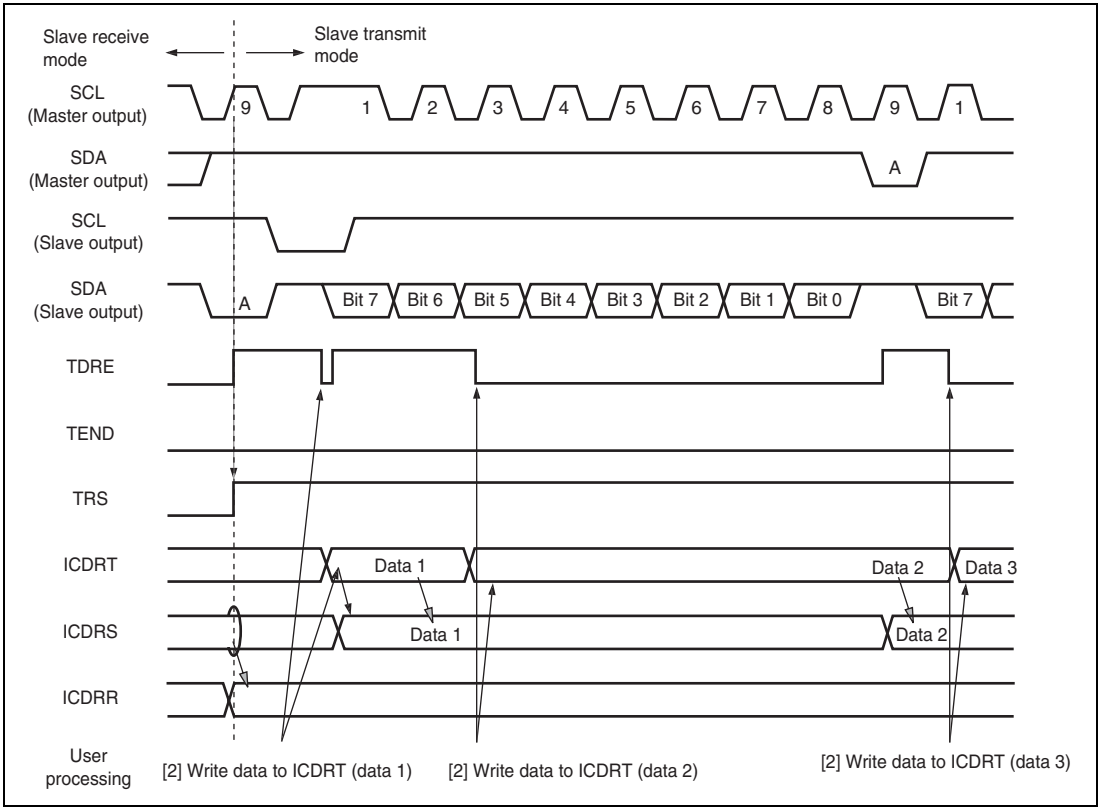


Figure 20.9 Slave Transmit Mode Operation Timing (1)

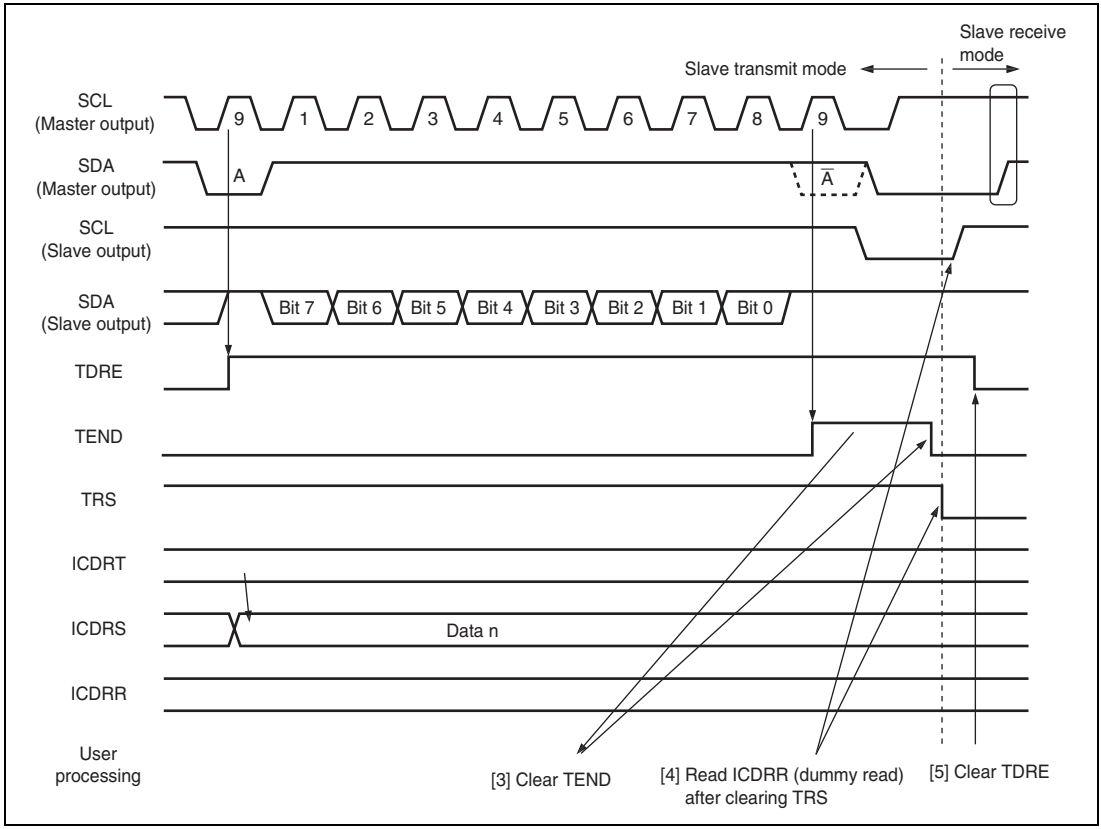


Figure 20.10 Slave Transmit Mode Operation Timing (2)

20.4.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For slave receive mode operation timing, refer to figures 20.11 and 20.12. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (Since the read data show the slave address and R/W, it is not used.)

3. Read ICDRR every time RDRF is set. If 8th receive clock pulse falls while RDRF is 1, SCL is fixed low until ICDRR is read. The change of the acknowledge before reading ICDRR, to be returned to the master device, is reflected to the next transmit frame.
4. The last byte data is read by reading ICDRR.

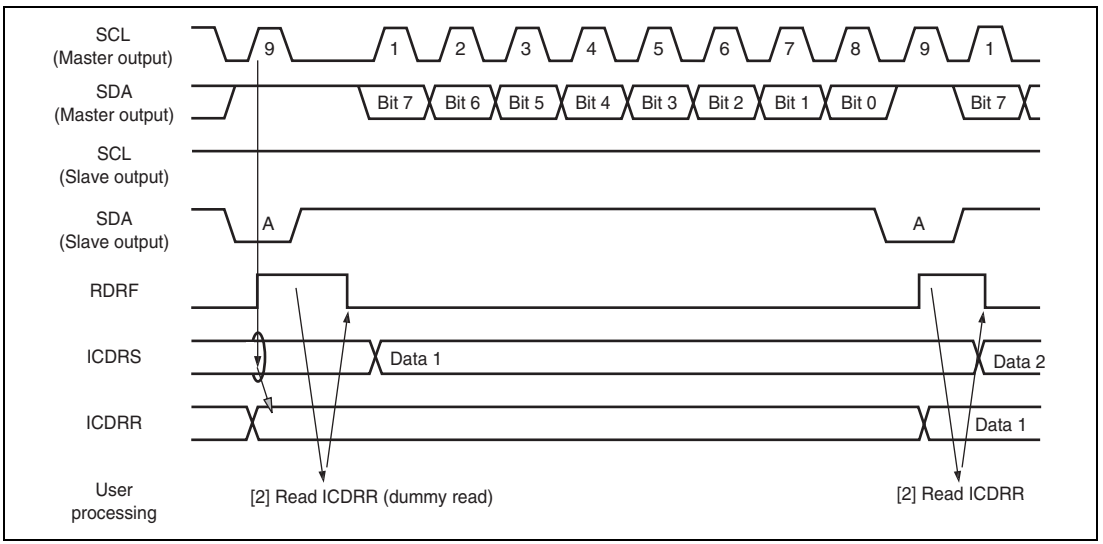


Figure 20.11 Slave Receive Mode Operation Timing (1)

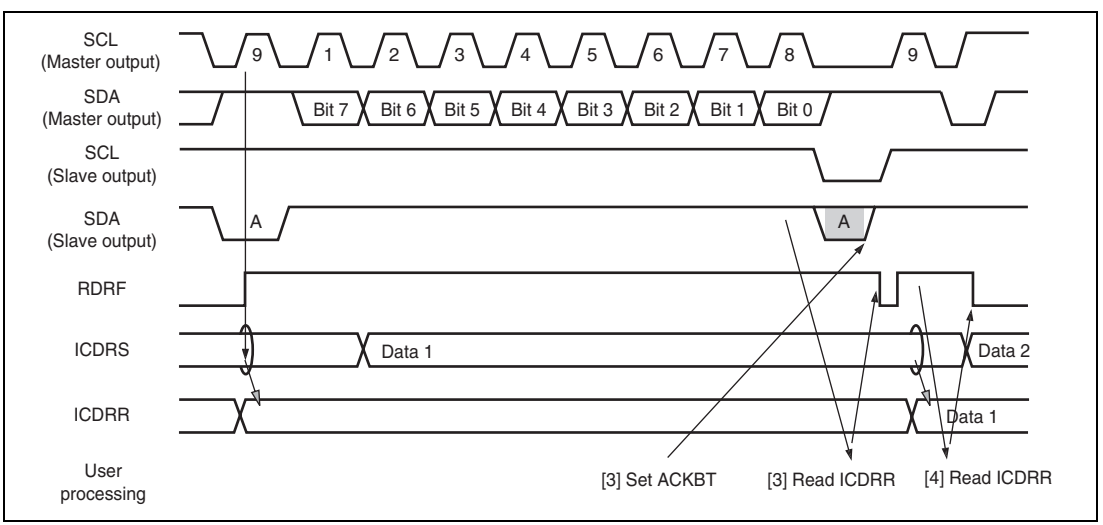


Figure 20.12 Slave Receive Mode Operation Timing (2)

20.4.6 Noise Filter

The logic levels at the SCL and SDA pins are routed through noise filters before being latched internally. Figure 20.13 shows a block diagram of the noise filter circuit.

The noise filter consists of three cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the peripheral clock. When NF2CYC is set to 0, this signal is not passed forward to the next circuit unless the outputs of both latches agree. When NF2CYC is set to 1, this signal is not passed forward to the next circuit unless the outputs of three latches agree. If they do not agree, the previous value is held.

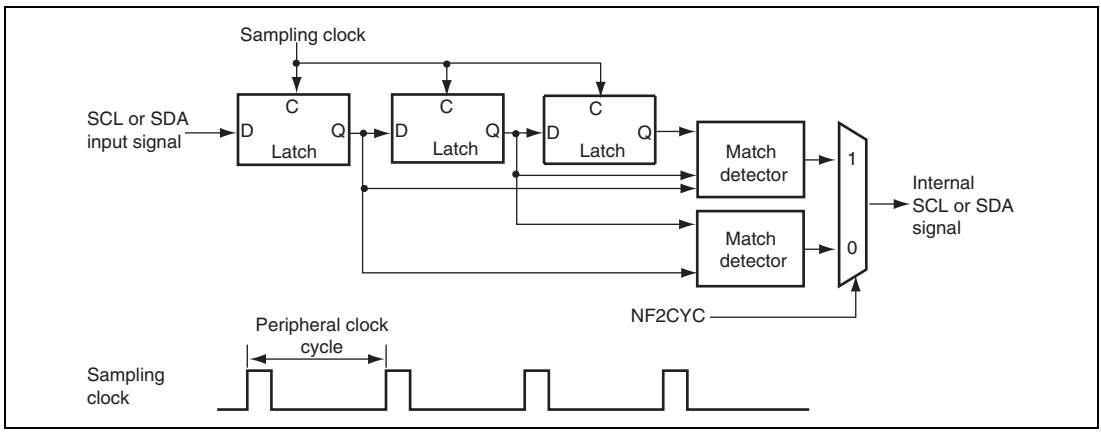


Figure 20.13 Block Diagram of Noise Filter

20.4.7 Example of Use

Flowcharts in respective modes that use the I²C bus interface are shown in figures 20.14 to 20.17.

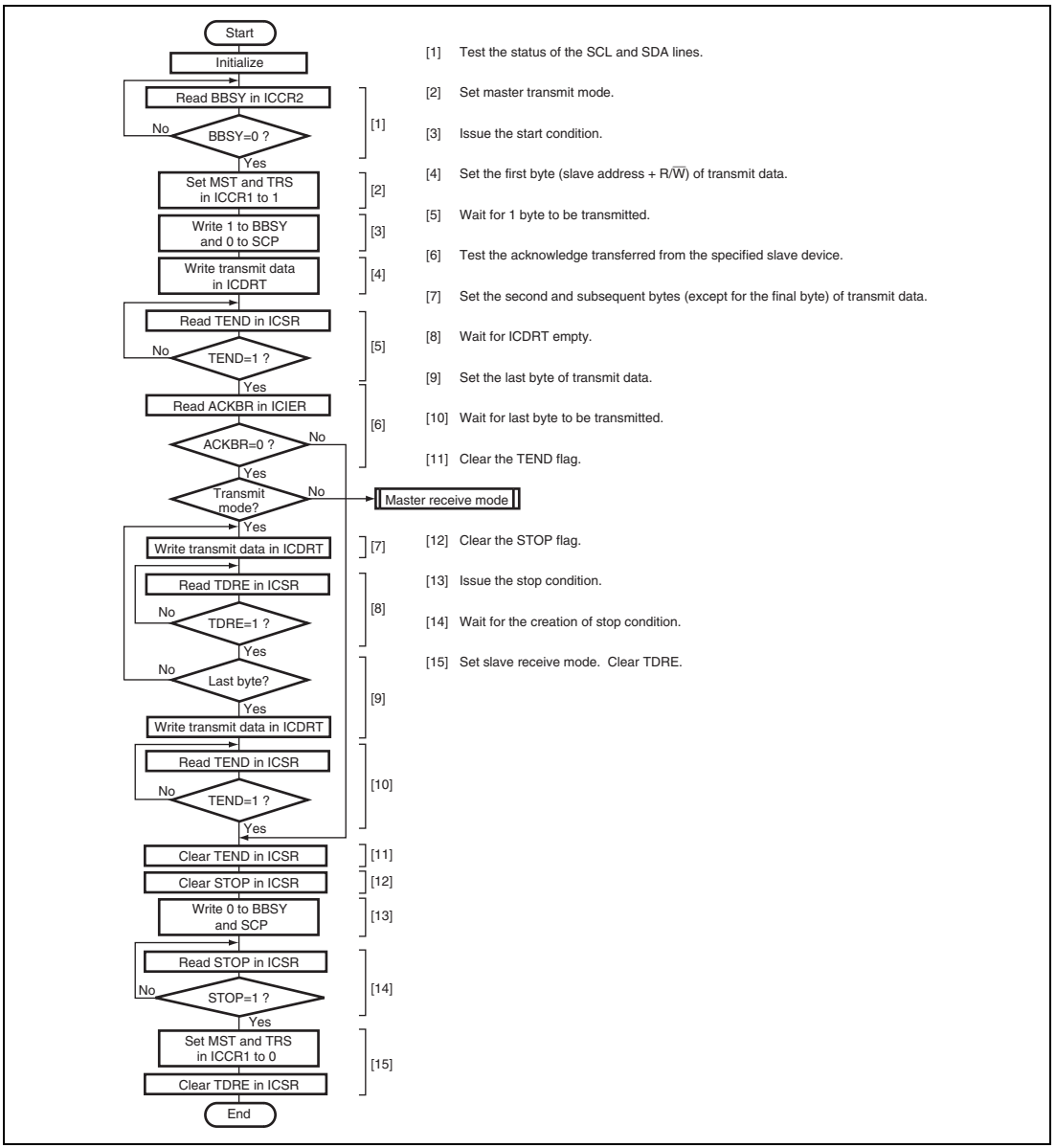
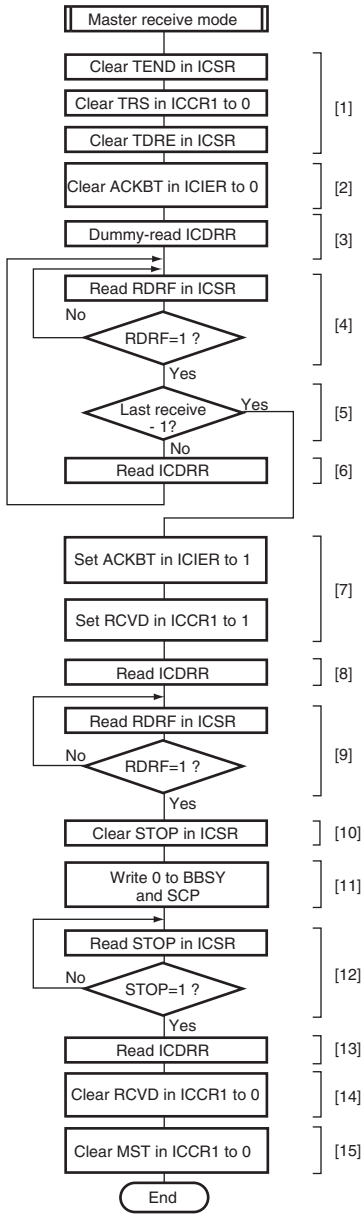


Figure 20.14 Sample Flowchart for Master Transmit Mode



[1] Clear TEND, select master receive mode, and then clear TDRE. *

[2] Set acknowledge to the transmit device. *

[3] Dummy-read ICDDR. *

[4] Wait for 1 byte to be received

[5] Check whether it is the (last receive - 1).

[6] Read the receive data.

[7] Set acknowledge of the final byte. Disable continuous reception (RCVD = 1).

[8] Read the (final byte - 1) of received data.

[9] Wait for the last byte to be receive.

[10] Clear the STOP flag.

[11] Issue the stop condition.

[12] Wait for the creation of stop condition.

[13] Read the last byte of receive data.

[14] Clear RCVD.

[15] Set slave receive mode.

Notes: * Make sure that no interrupt will be generated during steps [1] to [3].

When the size of receive data is only one byte in reception, steps [2] to [6] are skipped after step [1], before jumping to step [7]. The step [8] is dummy-read in ICDRR.

Figure 20.15 Sample Flowchart for Master Receive Mode

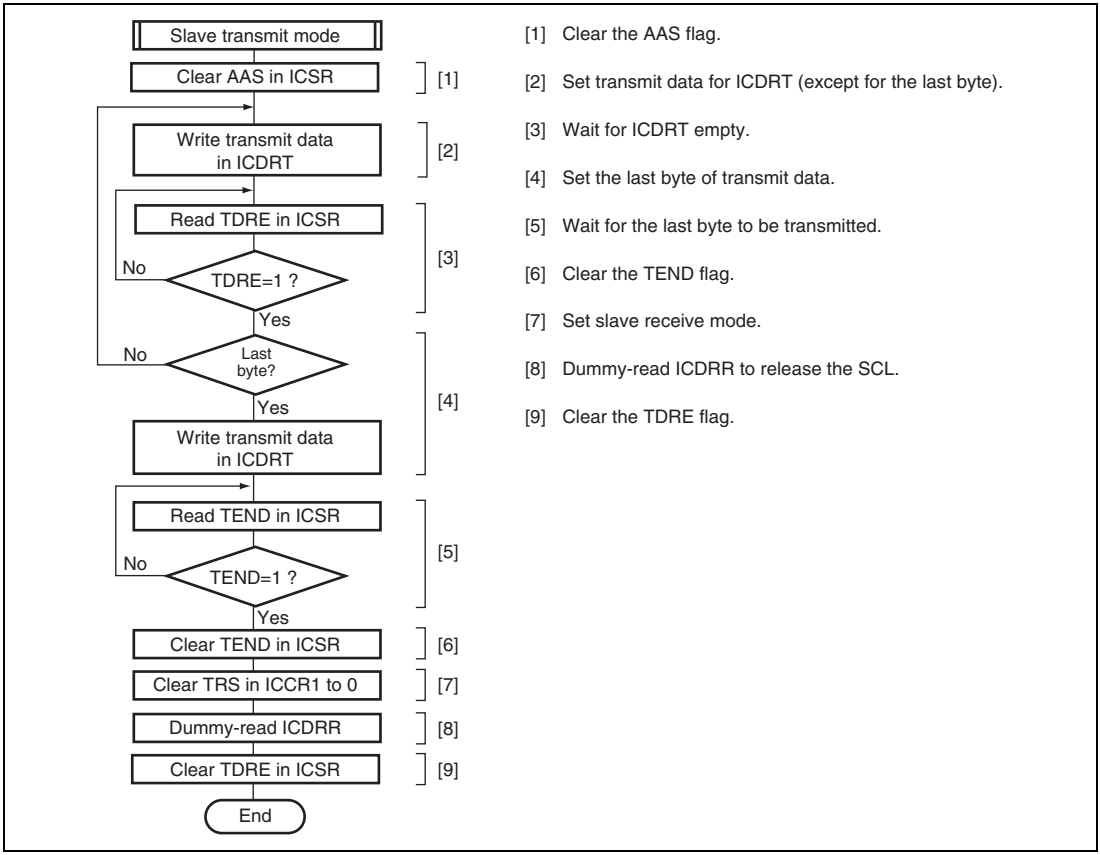


Figure 20.16 Sample Flowchart for Slave Transmit Mode

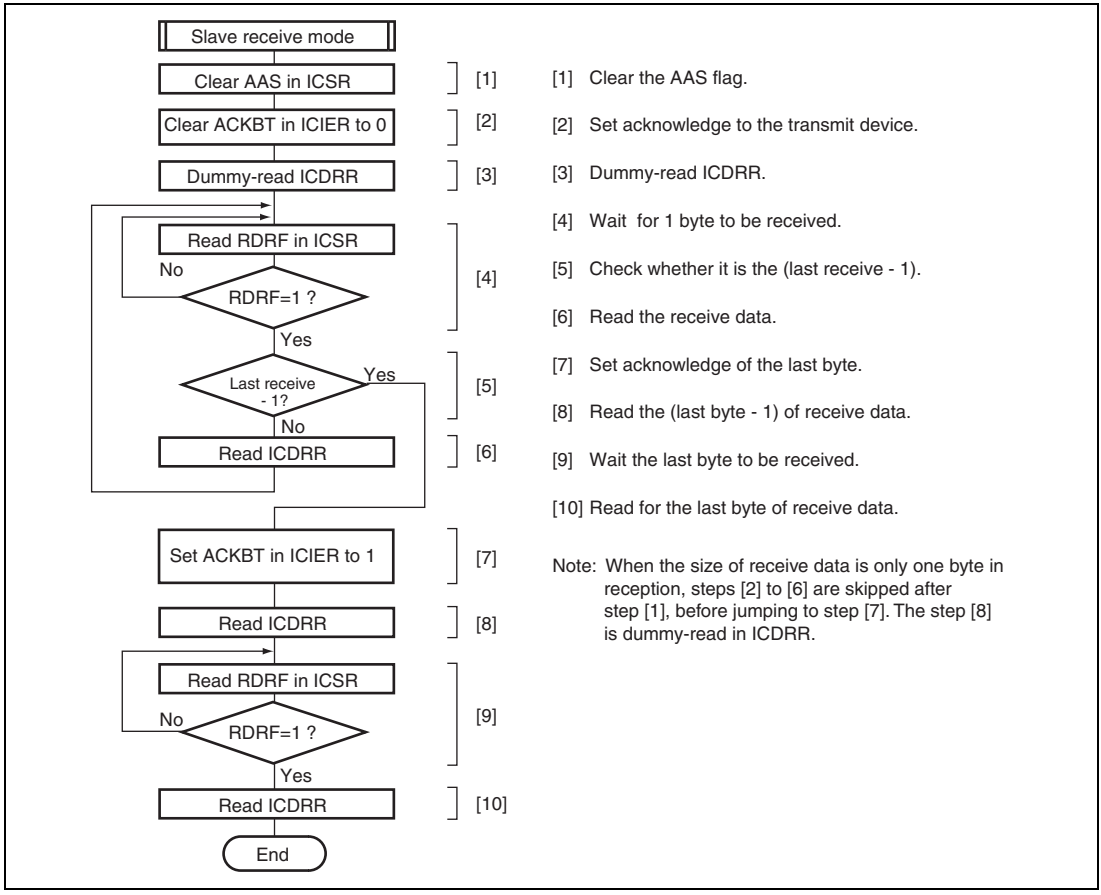


Figure 20.17 Sample Flowchart for Slave Receive Mode

20.5 Interrupt Requests

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK detection, STOP recognition, and arbitration lost/overrun error. Table 20.5 shows the contents of each interrupt request.

Table 20.5 Interrupt Requests

Interrupt Request	Abbreviation	Interrupt Condition
Transmit data Empty	TXI	(TDRE = 1) • (TIE = 1)
Transmit end	TEI	(TEND = 1) • (TEIE = 1)
Receive data full	RXI	(RDRF = 1) • (RIE = 1)
STOP recognition	STPI	(STOP = 1) • (STIE = 1)
NACK detection	NAKI	{(NACKF = 1) + (AL = 1)} • (NAKIE = 1)
Arbitration lost/ overrun error		

When the interrupt condition described in table 20.5 is 1, the CPU executes an interrupt exception handling. Interrupt sources should be cleared in the exception handling. The TDRE and TEND bits are automatically cleared to 0 by writing the transmit data to ICDRT. The RDRF bit is automatically cleared to 0 by reading ICDRR. The TDRE bit is set to 1 again at the same time when the transmit data is written to ICDRT. Therefore, when the TDRE bit is cleared to 0, then an excessive data of one byte may be transmitted.

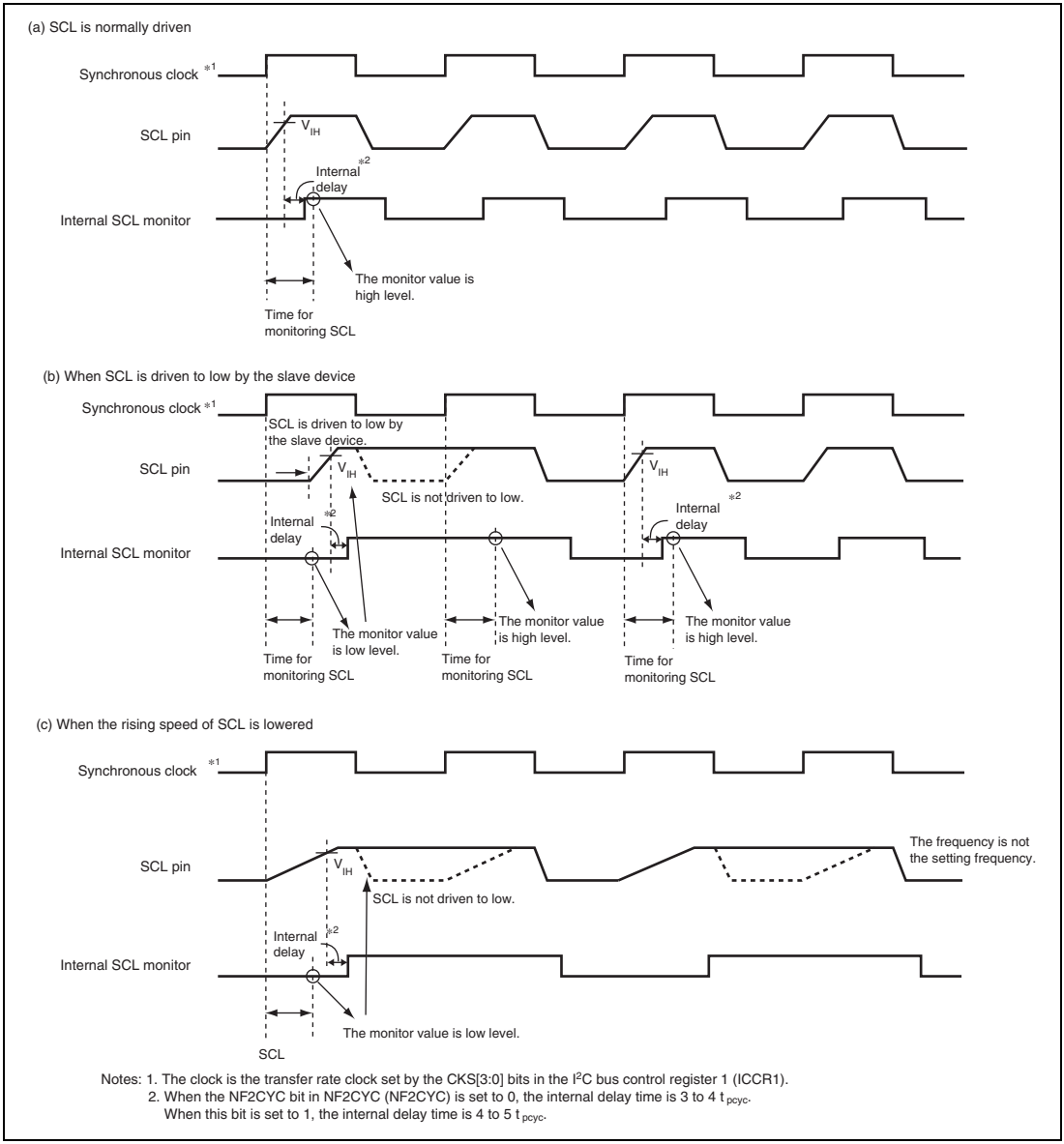
20.6 Bit Synchronous Circuit

In master mode, this module has a possibility that high level period may be short in the two states described below.

- When SCL is driven to low by the slave device
- When the rising speed of SCL is lowered by the load of the SCL line (load capacitance or pull-up resistance)

Therefore, it monitors SCL and communicates by bit with synchronization.

Figure 20.18 shows the timing of the bit synchronous circuit and table 20.6 shows the time when the SCL output changes from low to Hi-Z then SCL is monitored.



Notes: 1. The clock is the transfer rate clock set by the CKS[3:0] bits in the I²C bus control register 1 (ICCR1).
2. When the NF2CYC bit in NF2CYC (NF2CYC) is set to 0, the internal delay time is 3 to 4 t_{pcyc}.
When this bit is set to 1, the internal delay time is 4 to 5 t_{pcyc}.

Figure 20.18 Bit Synchronous Circuit Timing

Table 20.6 Time for Monitoring SCL

CKS3	CKS2	Time for Monitoring SCL*[1]
0	0	9 tpcyc*[2]
	1	21 tpcyc*[2]
1	0	39 tpcyc*[2]
	1	87 tpcyc*[2]

Notes: 1. Monitors the (on-board) SCL level after the time (pcyc) for monitoring SCL has passed since the rising edge of the SCL monitor timing reference clock.
2. pcyc = Pϕ × cyc

20.7 Usage Notes

20.7.1 Restriction on the Setting of Transfer Rate in Multi-Master Operation

When the IIC transfer rate of this LSI is set slower than that of other masters in multi-master operation, the SCL may be output with an unexpected width. To avoid this, set the transfer rate to at least 1/1.8 of the fastest rate among the other masters. For example, when the fastest transfer rate of the other masters is 400 Kbps, the IIC transfer rate of this LSI must be set to at least 223 Kbps (=400/1.8).

20.7.2 Restriction on the Use of Bit-Operation Instructions to Set MST and TRS in Multi-Master Operation

When the MST and the TRS are set for master transfer in multi-master operation, a contradictory event where AL=1 in ICSR and the mode is master transmission (MST=1, TRS=1) may occur, if arbitration lost signal is generated during execution of the bit operation instruction to set TRS. Avoid this in the following way:

1. In multi-master operation, use an MOV instruction to set MST and TRS.
2. When arbitration is lost, ensure that MST = 0 and TRS = 0. If the values are other than MST = 0 and TRS = 0, set the MST and the TRS bits to 0.

20.7.3 Notes on master receive mode

In master receive mode, when SCL is fixed low on the falling edge of the 8th clock while the RDRF bit is set to 1 and ICDRR is read around the falling edge of the 8th clock, the clock is only fixed low in the 8th clock of the next round of data reception.

The SCL is then released from its fixed state without reading ICDRR and the 9th clock is output. As a result, some receive data is lost.

Ways to avoid this phenomenon are listed below.

- Read ICDRR in master receive mode before the rising edge of the 8th clock.
- Set RCVD to 1 in master receive mode and perform communication in units of one byte.

20.7.4 Note on Setting ACKBT in Master Receive Mode

In master receive mode operation, set ACKBT before the falling edge of the 8th SCL cycle of the last date being continuously transferred. Not doing so can lead to an overrun for the slave transmission device.

20.7.5 Issuance of Stop Condition and Repeated Start Condition

A stop condition or repeated start condition should be issued after the fall of the ninth clock pulse is recognized. The fall of the ninth clock pulse can be recognized by checking the SCLO bit in the I²C bus control register 2 (ICCR2). When a stop condition or repeated start condition is issued at a specific timing under the conditions 1 or 2 shown below, the condition may not be output successfully. Issuance under other than these conditions will succeed with no problem.

1. When the SCL signal did not rise within the time specified in section 20.6, Bit Synchronous Circuit, due to the load of the SCL bus (load capacitance or pull-up resistor).
2. When the bit synchronous circuit is activated because the low-level periods of the eighth and ninth clock pulses are extended by the slave device.

Section 21 Serial I/O with FIFO (SIOF)

This LSI includes a single channel of clock-synchronized serial I/O module with FIFO (SIOF).

The SIOF is a module dedicated to the audio CODEC interface. Therefore, the SIOF is not suitable for the applications other than the audio CODEC interface.

21.1 Features

- Serial transfer
  16-stage 32-bit FIFOs (transmission and reception are independent of each other)
  Supports 8-bit data/16-bit data/16-bit stereo audio input/output
  MSB first for data transmission
  Supports a maximum of 48-kHz sampling rate
  Synchronization by either frame synchronization pulse or left/right channel switch
  Supports CODEC control data interface
  Connectable to linear, audio, or A-Law or μ-Law CODEC chip
  Supports both master and slave modes
- Serial clock
  An external pin input or internal clock (Pϕ) can be selected as the clock source.
- Interrupts: One type
- DMA transfer
  Supports DMA transfer by a transfer request for transmission and reception

Figure 21.1 shows a block diagram of the SIOF.

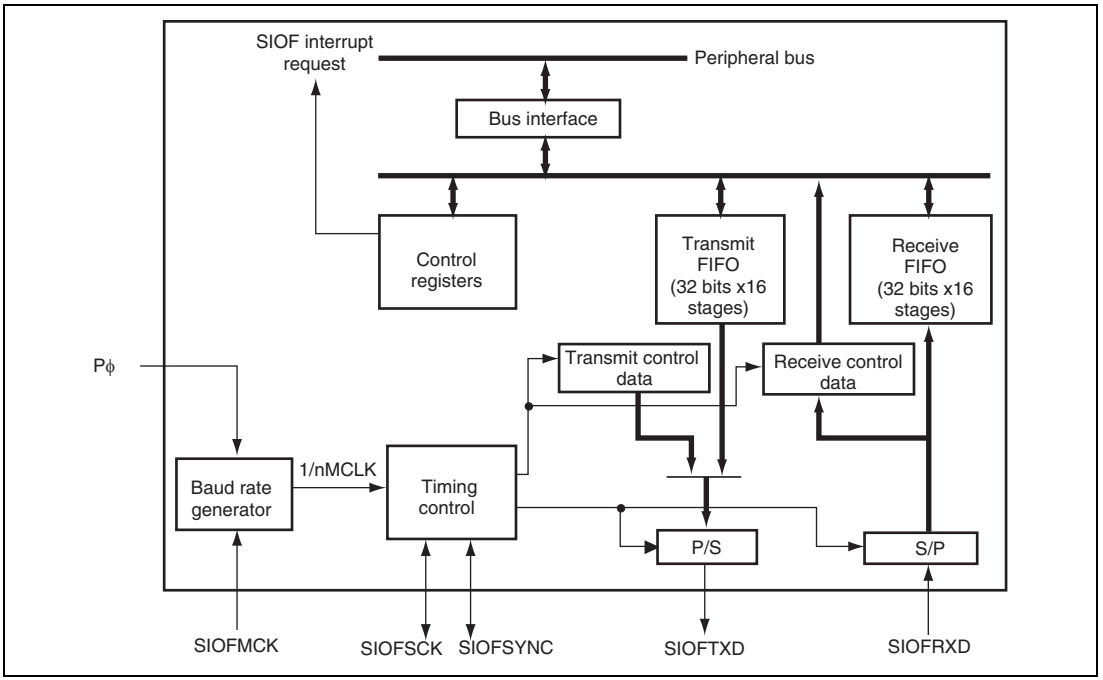


Figure 21.1 Block Diagram of SIOF

21.2 Input/Output Pins

The pin configuration in this module is shown in table 21.1.

Table 21.1 Pin Configuration

Pin Name	Abbreviation*	I/O	Description
SIOF_MCK	SIOFMCK	Input	Master clock input
SIOF_SCK	SIOFSCK	I/O	Serial clock (common to transmission/reception)
SIOF_SYNC	SIOFSYNC	I/O	Frame synchronization signal (common to transmission/reception)
SIOF_TXD	SIOFTXD	Output	Transmit data
SIOF_RXD	SIOFRXD	Input	Receive data

Note: * In the following descriptions, SIOFMCK, SIOFSCK, SIOFSYNC, SIOFTXD, and SIOFRXD are used as generic names.

21.3 Register Descriptions

Table 21.2 shows the SIOF register configuration. Table 21.3 shows the register states in each operating mode.

Table 21.2 Register Configuration

Name	Abbreviation	R/W	Address	Access Size
Mode register	SIMDR	R/W	H'A441 0000	16
Clock select register	SISCR	R/W	H'A441 0002	16
Transmit data assign register	SITDAR	R/W	H'A441 0004	16
Receive data assign register	SIRDAR	R/W	H'A441 0006	16
Control data assign register	SICDAR	R/W	H'A441 0008	16
Control register	SICTR	R/W	H'A441 000C	16
FIFO control register	SIFCTR	R/W	H'A441 0010	16
Status register	SISTR	R/W	H'A441 0014	16
Interrupt enable register	SIIER	R/W	H'A441 0016	16
Transmit data register	SITDR	W	H'A441 0020	32
Receive data register	SIRDR	R	H'A441 0024	32
Transmit control data register	SITCR	R/W	H'A441 0028	32
Receive control data register	SIRCR	R/W	H'A441 002C	32

Table 21.3 Register States in Each Operating Mode

Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
SIMDR	Initialized	Retained	Retained	Retained
SISCR	Initialized	Retained	Retained	Retained
SITDAR	Initialized	Retained	Retained	Retained
SIRDAR	Initialized	Retained	Retained	Retained
SICDAR	Initialized	Retained	Retained	Retained
SICTR	Initialized	Retained	Retained	Retained
SIFCTR	Initialized	Retained	Retained	Retained
SISTR	Initialized	Retained	Retained	Retained
SIIER	Initialized	Retained	Retained	Retained
SITDR	Undefined	Retained	Retained	Retained
SIRDR	Undefined	Retained	Retained	Retained
SITCR	Initialized	Retained	Retained	Retained
SIRCR	Undefined	Retained	Retained	Retained

21.3.1 Mode Register (SIMDR)

SIMDR is a 16-bit readable/writable register that sets the SIOF operating mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRMD[1:0]		SYNCAT	REDG	FL[3:0]				TXDIZ	RCIM	SYNCAC	SYNCDL	—	—	—	—
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	TRMD[1:0]	10	R/W	Transfer Mode 1, 0 Select transfer mode as shown in table 21.4. 00: Slave mode 1 01: Slave mode 2 10: Master mode 1 11: Master mode 2
13	SYNCAT	0	R/W	SIOFSYNC Pin Valid Timing Indicates the position of the SIOFSYNC signal to be output as a synchronization pulse. 0: At the start-bit data of frame 1: At the last-bit data of slot
12	REDG	0	R/W	Receive Data Sampling Edge 0: The SIOFRXD signal is sampled at the falling edge of SIOFSCK 1: The SIOFRXD signal is sampled at the rising edge of SIOFSCK Note: The timing to transmit the SIOFTXD signal is at the opposite edge of the timing that samples the SIOFRXD. This bit is valid only in master mode.

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	FL[3:0]	0000	R/W	Frame Length 3 to 0 00xx: Data length is 8 bits and frame length is 8 bits. 0100: Data length is 8 bits and frame length is 16 bits. 0101: Data length is 8 bits and frame length is 32 bits. 0110: Data length is 8 bits and frame length is 64 bits. 0111: Data length is 8 bits and frame length is 128 bits. 10xx: Data length is 16 bits and frame length is 16 bits. 1100: Data length is 16 bits and frame length is 32 bits. 1101: Data length is 16 bits and frame length is 64 bits. 1110: Data length is 16 bits and frame length is 128 bits. 1111: Data length is 16 bits and frame length is 256 bits. Note: When data length is specified as 8 bits, control data cannot be transmitted or received. x: Don't care
7	TXDIZ	0	R/W	SIOFTXD Pin Output when Transmission is Invalid* 0: High output (1 output) when invalid 1: High-impedance state when invalid Note: Invalid means when disabled, and when a slot that is not assigned as transmit data or control data is being transmitted.
6	RCIM	0	R/W	Receive Control Data Interrupt Mode 0: Sets the RCRDY bit in SISTR when the contents of SIRCR change. 1: Sets the RCRDY bit in SISTR each time when the SIRCR receives the control data.
5	SYNCAC	0	R/W	SIOFSYNC Pin Polarity Valid when the SIOFSYNC signal is output as a synchronous pulse. 0: Active-high 1: Active-low

Bit	Bit Name	Initial Value	R/W	Description
4	SYNCDL	0	R/W	Data Pin Bit Delay for SIOFSYNC Pin Valid when the SIOFSYNC signal is output as synchronous pulse. Only one-bit delay is valid for transmission in slave mode. 0: No bit delay 1: 1-bit delay
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Table 21.4 shows the operation in each transfer mode.

Table 21.4 Operation in Each Transfer Mode

Transfer Mode	Master/Slave	SIOFSYNC	Bit Delay	Control Data Method*
Slave mode 1	Slave	Synchronous pulse	SYNCDL bit	Slot position
Slave mode 2	Slave	Synchronous pulse		Secondary FS
Master mode 1	Master	Synchronous pulse		Slot position
Master mode 2	Master	L/R	No	Not supported

Note: * The control data method is valid only when the FL bit is specified as 1xxx (x: Don't care).

21.3.2 Control Register (SICTR)

SICTR is a 16-bit readable/writable register that sets the SIOF operating state.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKE	FSE	—	—	—	—	TXE	RXE	—	—	—	—	—	—	TXRST	RXRST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	SCKE	0	R/W	Serial Clock Output Enable This bit is valid in master mode. 0: Disables the SIOFSCK output (outputs 0) 1: Enables the SIOFSCK output If this bit is set to 1, the SIOF initializes the baud rate generator and initiates the operation. At the same time, the SIOF outputs the clock generated by the baud rate generator to the SIOFSCK pin.
14	FSE	0	R/W	Frame Synchronization Signal Output Enable This bit is valid in master mode. 0: Disables the SIOFSYNC output (outputs 0) 1: Enables the SIOFSYNC output If this bit is set to 1, the SIOF initializes the frame counter and initiates the operation.
13 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	TXE	0	R/W	Transmit Enable 0: Disables data transmission from the SIOFTXD pin 1: Enables data transmission from the SIOFTXD pin • This bit setting becomes valid at the start of the next frame (at the rising edge of the SIOFSYNC signal). • When the 1 setting for this bit becomes valid, the SIOF issues a transmit transfer request according to the setting of the TFWM bit in SIFCTR. When transmit data is stored in the transmit FIFO, transmission of data from the SIOFTXD pin begins. This bit is initialized upon a transmit reset.
8	RXE	0	R/W	Receive Enable 0: Disables data reception from SIOFRXD 1: Enables data reception from SIOFRXD • This bit setting becomes valid at the start of the next frame (at the rising edge of the SIOFSYNC signal). • When the 1 setting for this bit becomes valid, the SIOF begins the reception of data from the SIOFRXD pin. When receive data is stored in the receive FIFO, the SIOF issues a reception transfer request according to the setting of the RFWM bit in SIFCTR. This bit is initialized upon receive reset.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	TXRST	0	R/W	Transmit Reset 0: Does not reset transmit operation 1: Resets transmit operation • This bit setting becomes valid immediately. • When the 1 setting for this bit becomes valid, the SIOF immediately sets transmit data from the SIOFTXD pin to 1, and initializes the transmit-related status. The following are initialized. — Transmit FIFO write pointer — TCRDY, TFEMP, and TDREQ bits in SISTR • SIOF automatically clears this bit upon the completion of reset. Thus, this bit is always read as 0.
0	RXRST	0	R/W	Receive Reset 0: Does not reset receive operation 1: Resets receive operation • This bit setting becomes valid immediately • When the 1 setting for this bit becomes valid, the SIOF immediately disables reception from the SIOFRXD pin, and initializes the receive-related status. The following are initialized. — Receive FIFO read pointer — RCRDY, RFFUL, and RDREQ bits in SISTR • SIOF automatically clears this bit upon the completion of reset. Thus, this bit is always read as 0.

21.3.3 Transmit Data Register (SITDR)

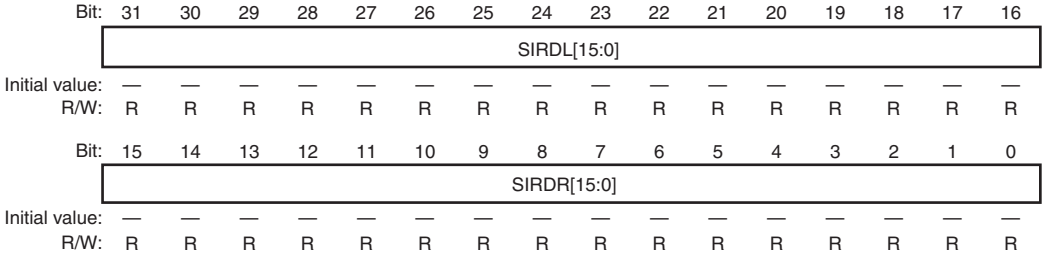
SITDR is a 32-bit write-only register that specifies the target channel (left or right) for data to be output from the SIOFTXD pin.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SITDL[15:0]															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SITDR[15:0]															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SITDL [15:0]	Undefined	W	Left-Channel Transmit Data Specify data to be output from the SIOFTXD pin as left-channel data. The position of the left-channel data in the transmit frame is specified by the TDLA bit in SITDAR. • These bits are valid only when the TDLE bit in SITDAR is set to 1.
15 to 0	SITDR [15:0]	Undefined	W	Right-Channel Transmit Data Specify data to be output from the SIOFTXD pin as right-channel data. The position of the right-channel data in the transmit frame is specified by the TDRA bit in SITDAR. • These bits are valid only when the TDRE bit and TLREP bit in SITDAR are set to 1 and cleared to 0, respectively.

21.3.4 Receive Data Register (SIRDR)

SIRDR is a 32-bit read-only register that reads receive data of the SIOF. SIRDR stores data in the receive FIFO.



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SIRDL [15:0]	Undefined	R	Left-Channel Receive Data Store data received from the SIOFRXD pin as left-channel data. The position of the left-channel data in the receive frame is specified by the RDLA bit in SIRDAR. • These bits are valid only when the RDLE bit in SIRDAR is set to 1.
15 to 0	SIRDR [15:0]	Undefined	R	Right-Channel Receive Data Store data received from the SIOFRXD pin as right-channel data. The position of the right-channel data in the receive frame is specified by the RDRA bit in SIRDAR. • These bits are valid only when the RDRE bit in SIRDAR is set to 1.

21.3.5 Transmit Control Data Register (SITCR)

SITCR is a 32-bit readable/writable register that specifies transmit control data of the SIOF. SITCR can be specified only when the FL bit in SIMDR is specified as 1xxx (x: Don't care).

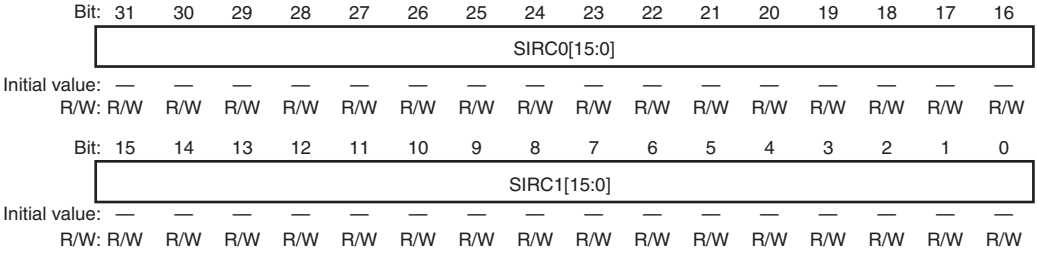
SITCR is initialized by the conditions specified in table 21.3, Register State of SIOF in Each Processing Mode, or by a transmit reset caused by the TXRST bit in SICTR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SITC0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SITC1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SITC0 [15:0]	H'0000	R/W	Control Channel 0 Transmit Data Specify data to be output from the SIOFTXD pin as control channel 0 transmit data. The position of the control channel 0 data in the transmit or receive frame is specified by the CD0A bit in SICDAR. • These bits are valid only when the CD0E bit in SICDAR is set to 1.
15 to 0	SITC1 [15:0]	H'0000	R/W	Control Channel 1 Transmit Data Specify data to be output from the SIOFTXD pin as control channel 1 transmit data. The position of the control channel 1 data in the transmit or receive frame is specified by the CD1A bit in SICDAR. • These bits are valid only when the CD1E bit in SICDAR is set to 1.

21.3.6 Receive Control Data Register (SIRCR)

SIRCR is a 32-bit readable/writable register that stores receive control data of the SIOF. SIRCR can be specified only when the FL bit in SIMDR is specified as 1xxx (x: Don't care).



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SIRC0 [15:0]	Undefined	R/W	Control Channel 0 Receive Data Store data received from the SIOFRXD pin as control channel 0 receive data. The position of the control channel 0 data in the transmit or receive frame is specified by the CD0A bit in SICDAR. • These bits are valid only when the CD0E bit in SICDAR is set to 1.
15 to 0	SIRC1 [15:0]	Undefined	R/W	Control Channel 1 Receive Data Store data received from the SIOFRXD pin as control channel 1 receive data. The position of the control channel 1 data in the transmit or receive frame is specified by the CD1A bit in SICDAR. • These bits are valid only when the CD1E bit in SICDAR is set to 1.

21.3.7 Status Register (SISTR)

SISTR is a 16-bit readable/writable register that shows the SIOF state. Each bit in this register becomes an SIOF interrupt source when the corresponding bit in SIIER is set to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TCRDY	TFEMP	TDREQ	—	RCRDY	RFFUL	RDREQ	—	—	SAERR	FSERR	TFOVF	TFUDF	RFUDF	RFOVF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	TCRDY	0	R	Transmit Control Data Ready 0: Indicates that a write to SITCR is disabled 1: Indicates that a write to SITCR is enabled • If SITCR is written when this bit is cleared to 0, SITCR is over-written and the previous contents of SITCR are not output from the SIOFTXD pin. • This bit is valid when the TXE bit in SITCR is set to 1. • This bit indicates a state of the SIOF. If SITCR is written, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
13	TFEMP	0	R	Transmit FIFO Empty 0: Indicates that transmit FIFO is not empty 1: Indicates that transmit FIFO is empty • This bit is valid when the TXE bit in SICTR is 1. • This bit indicates a state; if SITDR is written, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
12	TDREQ	0	R	Transmit Data Transfer Request 0: Indicates that the size of empty area in the transmit FIFO does not exceed the size specified by the TFWM bit in SIFCTR. 1: Indicates that the size of empty area in the transmit FIFO exceeds the size specified by the TFWM bit in SIFCTR. A transmit data transfer request is issued when the empty area in the transmit FIFO exceeds the size specified by the TFWM bit in SIFCTR. When using transmit data transfer through the DMAC, this bit is always cleared by one DMAC access. After DMAC access, when conditions for setting this bit are satisfied, the SIOF again indicates 1 for this bit. • This bit is valid when the TXE bit in SICTR is 1. • This bit indicates a state; if the size of empty area in the transmit FIFO is less than the size specified by the TFWM bit in SIFCTR, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	RCRDY	0	R	Receive Control Data Ready 0: Indicates that the SIRCR stores no valid data. 1: Indicates that the SIRCR stores valid data. • If SIRCR is written when this bit is set to 1, SIRCR is modified by the latest data. • This bit is valid when the RXE bit in SICTR is set to 1. • This bit indicates a state of the SIOF. If SIRCR is read, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
9	RFFUL	0	R	Receive FIFO Full 0: Receive FIFO not full 1: Receive FIFO full • This bit is valid when the RXE bit in SICTR is 1. • This bit indicates a state; if SIRDR is read, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
8	RDREQ	0	R	Receive Data Transfer Request 0: Indicates that the size of valid area in the receive FIFO does not exceed the size specified by the RFWM bit in SIFCTR. 1: Indicates that the size of valid area in the receive FIFO exceeds the size specified by the RFWM bit in SIFCTR. A receive data transfer request is issued when the valid data area in the receive FIFO exceeds the size specified by the RFWM bit in SIFCTR. When using receive data transfer through the DMAC, this bit is always cleared by one DMAC access. After DMAC access, when conditions for setting this bit are satisfied, the SIOF again indicates 1 for this bit. • This bit is valid when the RXE bit in SICTR is 1. • This bit indicates a state; if the size of valid data area in the receive FIFO is less than the size specified by the RFWM bit in SIFCTR, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	SAERR	0	R/W	Slot Assign Error 0: Indicates that no slot assign error occurs 1: Indicates that a slot assign error occurs A slot assign error occurs when the specifications in SITDAR, SIRDAR, and SICDAR overlap. If a slot assign error occurs, the SIOF does not transmit data to the SIOFTXD pin and does not receive data from the SIOFRXD pin. Note that the SIOF does not clear the TXE bit or RXE bit in SICTR at a slot assign error. • This bit is valid when the TXE bit or RXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
4	FSERR	0	R/W	Frame Synchronization Error 0: Indicates that no frame synchronization error occurs 1: Indicates that a frame synchronization error occurs A frame synchronization error occurs when the next frame synchronization timing appears before the previous data or control data transfers have been completed. If a frame synchronization error occurs, the SIOF performs transmission or reception for slots that can be transferred. • This bit is valid when the TXE or RXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
3	TFOVF	0	R/W	Transmit FIFO Overflow 0: No transmit FIFO overflow 1: Transmit FIFO overflow A transmit FIFO overflow means that there has been an attempt to write to SITDR when the transmit FIFO is full. When a transmit FIFO overflow occurs, the SIOF indicates overflow, and writing is invalid. • This bit is valid when the TXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
2	TFUDF	0	R/W	Transmit FIFO Underflow 0: No transmit FIFO underflow 1: Transmit FIFO underflow A transmit FIFO underflow means that loading for transmission has occurred when the transmit FIFO is empty. When a transmit FIFO underflow occurs, the SIOF repeatedly sends the previous transmit data. • This bit is valid when the TXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
1	RFUDF	0	R/W	Receive FIFO Underflow 0: No receive FIFO underflow 1: Receive FIFO underflow A receive FIFO underflow means that reading of SIRDR has occurred when the receive FIFO is empty. When a receive FIFO underflow occurs, the value of data read from SIRDR is not guaranteed. • This bit is valid when the RXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
0	RFOVF	0	R/W	Receive FIFO Overflow 0: No receive FIFO overflow 1: Receive FIFO overflow A receive FIFO overflow means that writing has occurred when the receive FIFO is full. When a receive FIFO overflow occurs, the SIOF indicates overflow, and receive data is lost. • This bit is valid when the RXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

21.3.8 Interrupt Enable Register (SIIER)

SIIER is a 16-bit readable/writable register that enables the issue of SIOF interrupts. When each bit in this register is set to 1 and the corresponding bit in SISTR is set to 1, the SIOF issues an interrupt.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDMAE	TCRDYE	TFEMPE	TDREQE	RDMAE	RCRDYE	RFFULE	RDREQE	—	—	SAERRE	FSERRE	TFOVFE	TFUDFE	RFUDFE	RFOVFE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	TDMAE	0	R/W	Transmit Data DMA Transfer Request Enable Transmits an interrupt as an interrupt to the CPU/DMA transfer request. The TDREQE bit can be set as transmit interrupts. 0: Used as a CPU interrupt 1: Used as a DMA transfer request to the DMAC
14	TCRDYE	0	R/W	Transmit Control Data Ready Enable 0: Disables interrupts due to transmit control data ready 1: Enables interrupts due to transmit control data ready
13	TFEMPE	0	R/W	Transmit FIFO Empty Enable 0: Disables interrupts due to transmit FIFO empty 1: Enables interrupts due to transmit FIFO empty
12	TDREQE	0	R/W	Transmit Data Transfer Request Enable 0: Disables interrupts due to transmit data transfer requests 1: Enables interrupts due to transmit data transfer requests
11	RDMAE	0	R/W	Receive Data DMA Transfer Request Enable Transmits an interrupt as an interrupt to the CPU/DMA transfer request. The RDREQE bit can be set as receive interrupts. 0: Used as a CPU interrupt 1: Used as a DMA transfer request to the DMAC
10	RCRDYE	0	R/W	Receive Control Data Ready Enable 0: Disables interrupts due to receive control data ready 1: Enables interrupts due to receive control data ready

Bit	Bit Name	Initial Value	R/W	Description
9	RFFULE	0	R/W	Receive FIFO Full Enable 0: Disables interrupts due to receive FIFO full 1: Enables interrupts due to receive FIFO full
8	RDREQE	0	R/W	Receive Data Transfer Request Enable 0: Disables interrupts due to receive data transfer requests 1: Enables interrupts due to receive data transfer requests
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	SAERRE	0	R/W	Slot Assign Error Enable 0: Disables interrupts due to slot assign error 1: Enables interrupts due to slot assign error
4	FSERRE	0	R/W	Frame Synchronization Error Enable 0: Disables interrupts due to frame synchronization error 1: Enables interrupts due to frame synchronization error
3	TFOVFE	0	R/W	Transmit FIFO Overflow Enable 0: Disables interrupts due to transmit FIFO overflow 1: Enables interrupts due to transmit FIFO overflow
2	TFUDFE	0	R/W	Transmit FIFO Underflow Enable 0: Disables interrupts due to transmit FIFO underflow 1: Enables interrupts due to transmit FIFO underflow
1	RFUDFE	0	R/W	Receive FIFO Underflow Enable 0: Disables interrupts due to receive FIFO underflow 1: Enables interrupts due to receive FIFO underflow
0	RFOVFE	0	R/W	Receive FIFO Overflow Enable 0: Disables interrupts due to receive FIFO overflow 1: Enables interrupts due to receive FIFO overflow

21.3.9 FIFO Control Register (SIFCTR)

SIFCTR is a 16-bit readable/writable register that indicates the area available for the transmit/receive FIFO transfer.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TFWM[2:0]			TFUA[4:0]					RFWM[2:0]			RFUA[4:0]				
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	TFWM[2:0]	000	R/W0	Transmit FIFO Watermark 000: Issue a transfer request when 16 stages of the transmit FIFO are empty. 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: Issue a transfer request when 12 or more stages of the transmit FIFO are empty. 101: Issue a transfer request when 8 or more stages of the transmit FIFO are empty. 110: Issue a transfer request when 4 or more stages of the transmit FIFO are empty. 111: Issue a transfer request when 1 or more stages of transmit FIFO are empty. • A transfer request to the transmit FIFO is issued by the TDREQE bit in SISTR. • The transmit FIFO is always used as 16 stages of the FIFO regardless of these bit settings.
12 to 8	TFUA[4:0]	10000	R	Transmit FIFO Usable Area Indicate the number of words that can be transferred by the CPU or DMAC as B'00000 (full) to B'10000 (empty).

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	RFWM[2:0]	000	R/W	Receive FIFO Watermark 000: Issue a transfer request when 1 stage or more of the receive FIFO are valid. 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: Issue a transfer request when 4 or more stages of the receive FIFO are valid. 101: Issue a transfer request when 8 or more stages of the receive FIFO are valid. 110: Issue a transfer request when 12 or more stages of the receive FIFO are valid. 111: Issue a transfer request when 16 stages of the receive FIFO are valid. • A transfer request to the receive FIFO is issued by the RDREQE bit in SISTR. • The receive FIFO is always used as 16 stages of the FIFO regardless of these bit settings.
4 to 0	RFUA[4:0]	00000	R	Receive FIFO Usable Area Indicate the number of words that can be transferred by the CPU or DMAC as B'00000 (empty) to B'10000 (full).

21.3.10 Clock Select Register (SISCR)

SISCR is a 16-bit readable/writable register that sets the serial clock generation conditions for the master clock. SISCR can be specified when the bits TRMD[1:0] in SIMDR are specified as B'10 or B'11.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSSEL	MSIMM	—	BRPS[4:0]					—	—	—	—	—	BRDV[2:0]		
Initial value:	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	MSSEL	1	R/W	Master Clock Source Selection The master clock is the clock input to the baud rate generator. 0: Uses the input signal of the SIOFMCK pin as the master clock 1: Uses Pϕ as the master clock
14	MSIMM	1	R/W	Master Clock Direct Selection 0: Uses the output clock of the baud rate generator as the serial clock 1: Uses the master clock itself as the serial clock
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12 to 8	BRPS[4:0]	00000	R/W	Prescalar Setting Set the master clock division ratio according to the count value of the prescalar of the baud rate generator. The range of settings is from B'00000 (× 1/1) to B'11111 (× 1/32).
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	BRDV[2:0]	000	R/W	Baud rate generator's Division Ratio Setting Set the frequency division ratio for the output stage of the baud rate generator. 000: Prescalar output × 1/2 001: Prescalar output × 1/4 010: Prescalar output × 1/8 011: Prescalar output × 1/16 100: Prescalar output × 1/32 101: Setting prohibited 110: Setting prohibited 111: Prescalar output × 1/1 • Setting 111 is valid only when the bits BRPS[4:0] are set to B'00000 or B'00001. • The final frequency division ratio of the baud rate generator is determined by BRPS × BRDV (maximum 1/1024)

21.3.11 Transmit Data Assign Registers (SITDAR)

SITDAR is a 16-bit readable/writable register that specifies the position of the transmit data in a frame.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDLE	—	—	—	TDLA[3:0]				TDRE	TLREP	—	—	TDRA[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	TDLE	0	R/W	Transmit Left-Channel Data Enable 0: Disables left-channel data transmission 1: Enables left-channel data transmission

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	TDLA[3:0]	0000	R/W	Transmit Left-Channel Data Assigns 3 to 0 Specify the position of left-channel data in a transmit frame as B'0000 (0) to B'1110 (14). 1111: Setting prohibited • Transmit data for the left channel is specified in the SITDL bit in SITDR.
7	TDRE	0	R/W	Transmit Right-Channel Data Enable 0: Disables right-channel data transmission 1: Enables right-channel data transmission
6	TLREP	0	R/W	Transmit Left-Channel Repeat 0: Transmits data specified in the SITDR bit in SITDR as right-channel data 1: Repeatedly transmits data specified in the SITDL bit in SITDR as right-channel data • This bit setting is valid when the TDRE bit is set to 1. • When this bit is set to 1, the SITDR settings are ignored.
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	TDRA[3:0]	0000	R/W	Transmit Right-Channel Data Assigns 3 to 0 Specify the position of right-channel data in a transmit frame as B'0000 (0) to B'1110 (14). 1111: Setting prohibited • Transmit data for the right channel is specified in the SITDR bit in SITDR.

21.3.12 Receive Data Assign Register (SIRDAR)

SIRDAR is a 16-bit readable/writable register that specifies the position of the receive data in a frame.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDLE	—	—	—	RDLA[3:0]				RDRE	—	—	—	RDRA[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	RDLE	0	R/W	Receive Left-Channel Data Enable 0: Disables left-channel data reception 1: Enables left-channel data reception
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	RDLA[3:0]	0000	R/W	Receive Left-Channel Data Assigns 3 to 0 Specify the position of left-channel data in a receive frame as B'0000 (0) to B'1110 (14). 1111: Setting prohibited • Receive data for the left channel is stored in the SIRDL bit in SIRDR.
7	RDRE	0	R/W	Receive Right-Channel Data Enable 0: Disables right-channel data reception 1: Enables right-channel data reception
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	RDRA[3:0]	0000	R/W	Receive Right-Channel Data Assigns 3 to 0 Specify the position of right-channel data in a receive frame as B'0000 (0) to B'1110 (14). 1111: Setting prohibited • Receive data for the right channel is stored in the SIRDR bit in SIRDR.

21.3.13 Control Data Assign Register (SICDAR)

SICDAR is a 16-bit readable/writable register that specifies the position of the control data in a frame. SICDAR can be specified only when the FL bit in SIMDR is specified as 1xxx (x: Don't care).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CD0E	—	—	—	CD0A[3:0]				CD1E	—	—	—	CD1A[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	CD0E	0	R/W	Control Channel 0 Data Enable 0: Disables transmission and reception of control channel 0 data 1: Enables transmission and reception of control channel 0 data
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	CD0A[3:0]	0000	R/W	Control Channel 0 Data Assigns 3 to 0 Specify the position of control channel 0 data in a receive or transmit frame as B'0000 (0) to B'1110 (14). 1111: Setting prohibited • Transmit data for the control channel 0 data is specified in the SITD0 bit in SITCR. • Receive data for the control channel 0 data is stored in the SIRD0 bit in SIRCR.
7	CD1E	0	R/W	Control Channel 1 Data Enable 0: Disables transmission and reception of control channel 1 data 1: Enables transmission and reception of control channel 1 data
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	CD1A[3:0]	0000	R/W	Control Channel 1 Data Assigns 3 to 0 Specify the position of control channel 1 data in a receive or transmit frame as B'0000 (0) to B'1110 (14). 1111: Setting prohibited • Transmit data for the control channel 1 data is specified in the SITD1 bit in SITCR. • Receive data for the control channel 1 data is stored in the SIRD1 bit in SIRCR.

21.4 Operation

21.4.1 Serial Clocks

(1) Master/Slave Modes

The following modes are available as the SIOF clock mode.

- Slave mode: SIOFSCK, SIOFSYNC input
- Master mode: SIOFSCK, SIOFSYNC output

(2) Baud Rate Generator

In SIOF master mode, the baud rate generator (BRG) is used to generate the serial clock. The division ratio is from 1/1 to 1/1024.

Figure 21.2 shows connections for supply of the serial clock.

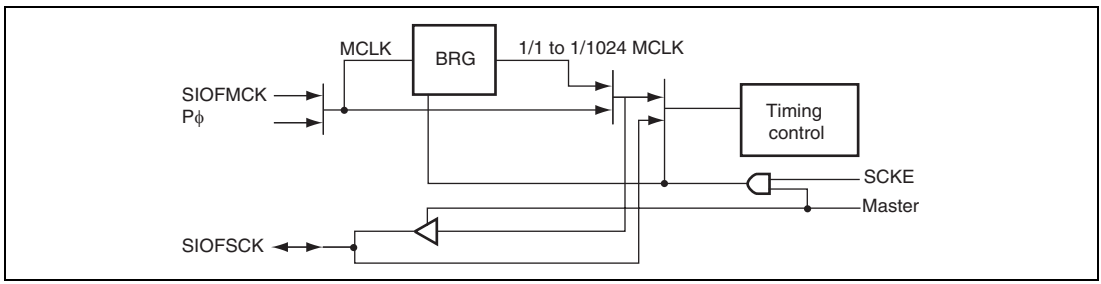


Figure 21.2 Serial Clock Supply

Table 21.5 shows an example of serial clock frequency.

Table 21.5 SIOF Serial Clock Frequency

	Sampling Rate		
Frame Length	**8 kHz**	**44.1 kHz**	**48 kHz**
32 bits	256 kHz	1.4112 MHz	1.536 MHz
64 bits	512 kHz	2.8224 MHz	3.072 MHz
128 bits	1.024 MHz	5.6448 MHz	6.144 MHz
256 bits	2.048 MHz	11.289 MHz	12.289 MHz

21.4.2 Serial Timing

(1) SIOFSYNC

The SIOFSYNC is a frame synchronization signal. Depending on the transfer mode, it has the following two functions.

- Synchronous pulse: 1-bit-width pulse indicating the start of the frame
- L/R: 1/2-frame-width pulse indicating the left-channel stereo data (L) in high level and the right-channel stereo data (R) in low level

Figure 21.3 shows the SIOFSYNC synchronization timing.

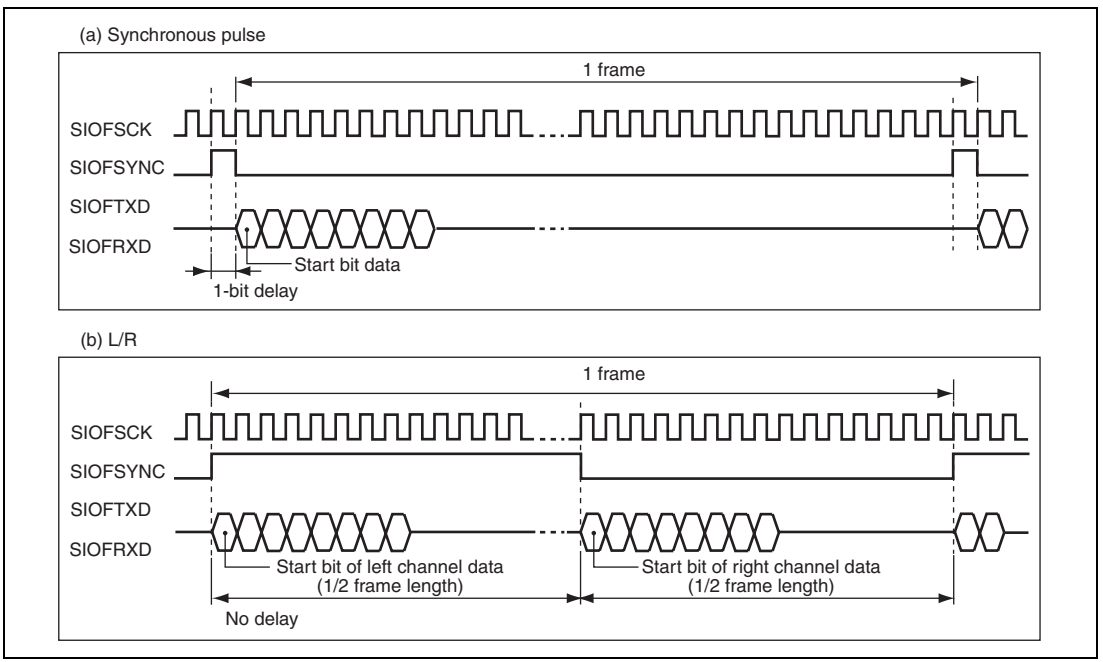


Figure 21.3 Serial Data Synchronization Timing

(2) Transmit/Receive Timing

The SIOFTXD transmit timing and SIOFRXD receive timing relative to the SIOFSCK can be set as the sampling timing in the following two ways. The transmit/receive timing is set using the REDG bit in SIMDR.

- Falling-edge sampling
- Rising-edge sampling

Figure 21.4 shows the transmit/receive timing.

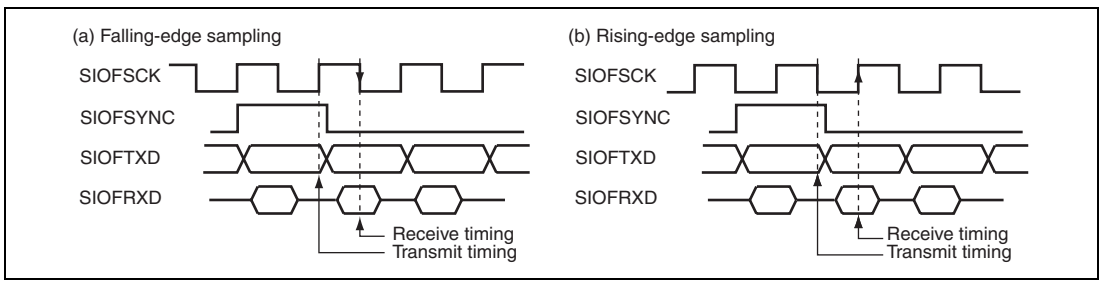


Figure 21.4 SIOF Transmit/Receive Timing

21.4.3 Transfer Data Format

The SIOF performs the following transfer.

- Transmit/receive data: Transfer of 8-bit data/16-bit data/16-bit stereo data
- Control data: Transfer of 16-bit data (uses the specific register as interface)

(1) Transfer Mode

The SIOF supports the following four transfer modes as listed in table 21.6. The transfer mode can be specified by the bits TRMD[1:0] in SIMDR.

Table 21.6 Serial Transfer Modes

Transfer Mode	SIOFSYNC	Bit Delay	Control Data
Slave mode 1	Synchronous pulse	SYNCDL bit	Slot position
Slave mode 2	Synchronous pulse		Secondary FS
Master mode 1	Synchronous pulse		Slot position
Master mode 2	L/R	No	Not supported

(2) Frame Length

The length of the frame to be transferred by the SIOF is specified by the bits FL[3:0] in SIMDR. Table 21.7 shows the relationship between the bits FL[3:0] settings and frame length.

Table 21.7 Frame Length

FL[3:0]	Slot Length	Number of Bits in a Frame	Transfer Data
00XX	8	8	8-bit monaural data
0100	8	16	8-bit monaural data
0101	8	32	8-bit monaural data
0110	8	64	8-bit monaural data
0111	8	128	8-bit monaural data
10xx	16	16	16-bit monaural data
1100	16	32	16-bit monaural stereo data
1101	16	64	16-bit monaural stereo data
1110	16	128	16-bit monaural stereo data
1111	16	256	16-bit monaural stereo data

[Legend]
X: Don't care

(3) Slot Position

The SIOF can specify the position of transmit data, receive data, and control data in a frame (common to transmission and reception) by slot numbers. The slot number of each data is specified by the following registers.

- Transmit data: SITDAR
- Receive data: SIRDAR
- Control data: SICDAR

Only 16-bit data is valid for control data. In addition, control data is always assigned to the same slot number both in transmission and reception.

21.4.4 Register Allocation of Transfer Data

(1) Transmit/Receive Data

Writing and reading of transmit/receive data is performed for the following registers.

- Transmit data writing: SITDR (32-bit access)
- Receive data reading: SIRDR (32-bit access)

Figure 21.5 shows the transmit/receive data and the SITDR and SIRDR bit alignment.

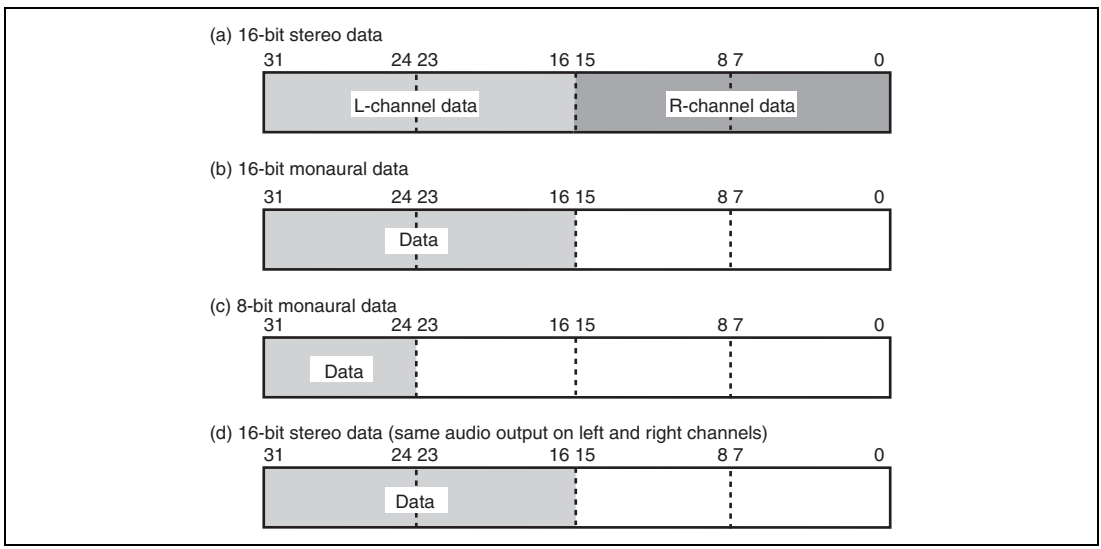


Figure 21.5 Transmit/Receive Data Bit Alignment

Note: In the figure, only the shaded areas are transmitted or received as valid data. Therefore, access must be made in byte units for 8-bit data, and in word units for 16-bit data. Data in unshaded areas is not transmitted or received.

Monaural or stereo can be specified for transmit data by the TDLE bit and TDRE bit in SITDAR. Monaural or stereo can be specified for receive data by the RDLE bit and RDRE bit in SIRDAR. To achieve left and right same audio output while stereo is specified for transmit data, specify the TLREP bit in SITDAR. Table 21.8 and table 21.9 show the audio mode specification for transmit data and that for receive data, respectively.

Table 21.8 Audio Mode Specification for Transmit Data

	Bit		
Mode	**TDLE**	**TDRE**	**TLREP**
Monaural	1	0	X
Stereo	1	1	0
Left and right same audio output	1	1	1

[legend]
X: Don't care

Table 21.9 Audio Mode Specification for Receive Data

	Bit	
Mode	**RDLE**	**RDRE**
Monaural	1	0
Stereo	1	1

Note: Left and right same audio mode is not supported in receive data.
To execute 8-bit monaural transmission or reception, use the left channel.

(2) Control Data

Control data is written to or read from by the following registers.

- Transmit control data write: SITCR (32-bit access)
- Receive control data read: SIRCR (32-bit access)

Figure 21.6 shows the control data and bit alignment in SITCR and SIRCR.

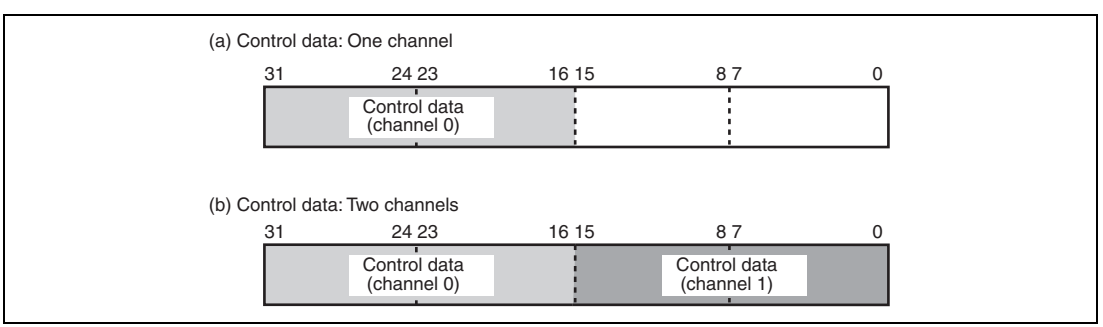


Figure 21.6 Control Data Bit Alignment

The number of channels in control data is specified by the CD0E and CD1E bits in SICDAR. Table 21.10 shows the relationship between the number of channels in control data and bit settings.

Table 21.10 Setting Number of Channels in Control Data

	Bit	
Number of Channels	**CD0E**	**CD1E**
1	1	0
2	1	1

Note: To use only one channel in control data, use channel 0.

21.4.5 Control Data Interface

Control data performs control command output to the CODEC and status input from the CODEC. The SIOF supports the following two control data interface methods.

- Control by slot position
- Control by secondary FS

Control data is valid only when data length is specified as 16 bits.

(1) Control by Slot Position (Master Mode 1, Slave Mode 1)

Control data is transferred for all frames transmitted or received by the SIOF by specifying the slot position of control data. This method can be used in both SIOF master and slave modes. Figure 21.7 shows an example of the control data interface timing by slot position control.

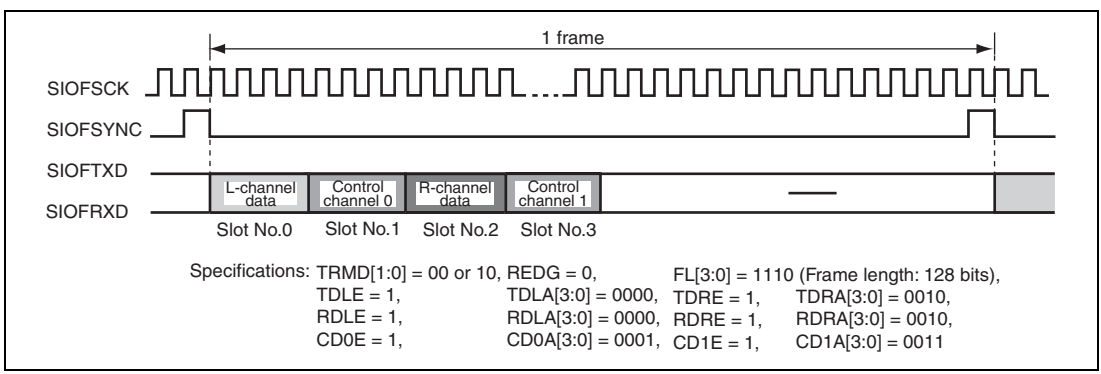


Figure 21.7 Control Data Interface (Slot Position)

(2) Control by Secondary FS (Slave Mode 2)

The CODEC normally outputs the SIOFSYNC signal as synchronization pulse (FS). In this method, the CODEC outputs the secondary FS specific to the control data transfer after 1/2 frame time has been passed (not the normal FS output timing) to transmit or receive control data. This method is valid for SIOF slave mode. The following summarizes the control data interface procedure by the secondary FS.

- Transmit normal transmit data of LSB = 0 (the SIOF forcibly clears 0).
- To execute control data transmission, send transmit data of LSB = 1 (the SIOF forcibly set to 1 by writing SITCDR).
- The CODEC outputs the secondary FS.
- The SIOF transmits or receives (stores in SIRCDR) control data (data specified by SITCDR) synchronously with the secondary FS.

Figure 21.8 shows an example of the control data interface timing by the secondary FS.

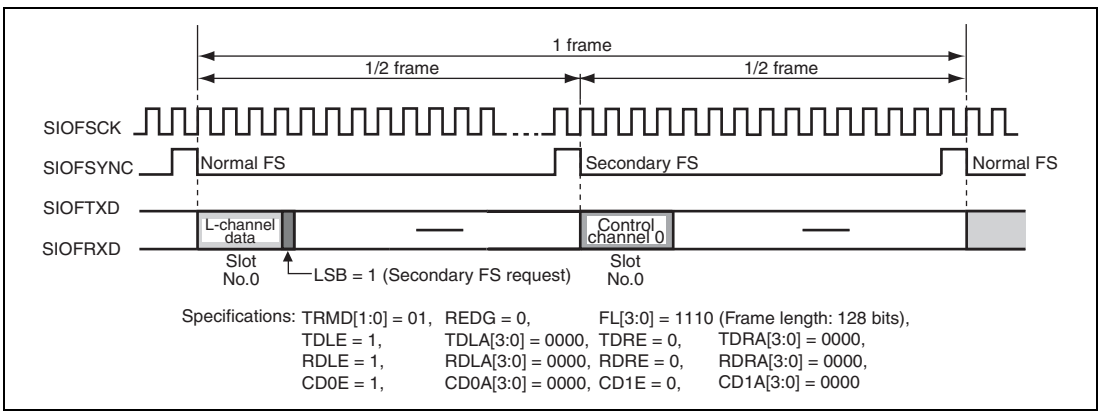


Figure 21.8 Control Data Interface (Secondary FS)

21.4.6 FIFO

(1) Overview

The transmit and receive FIFOs of the SIOF have the following features.

- 16-stage 32-bit FIFOs for transmission and reception
- The FIFO pointer can be updated in one read or write cycle regardless of access size of the CPU and DMAC. (One-stage 32-bit FIFO access cannot be divided into multiple accesses.)

(2) Transfer Request

The transfer request of the FIFO can be issued to the CPU or DMAC as the following interrupt sources.

- FIFO transmit request: TDREQ (transmit interrupt source)
- FIFO receive request: RDREQ (receive interrupt source)

The request conditions for FIFO transmit or receive can be specified individually. The request conditions for the FIFO transmit and receive are specified by the bits TFWM[2:0] and the bits RFWM[2:0] in SIFCTR, respectively. Table 21.11 and table 21.12 summarize the conditions specified by SIFCTR.

Table 21.11 Conditions to Issue Transmit Request

TFWM[2:0]	Number of Requested Stages	Transmit Request	Used Areas
000	1	Empty area is 16 stages	Smallest
100	4	Empty area is 12 stages or more	↕
101	8	Empty area is 8 stages or more	
110	12	Empty area is 4 stages or more	
111	16	Empty area is 1 stage or more	Largest

Table 21.12 Conditions to Issue Receive Request

RFWM[2:0]	Number of Requested Stages	Receive Request	Used Areas
000	1	Valid data is 1 stage or more	Smallest
100	4	Valid data is 4 stages or more	
101	8	Valid data is 8 stages or more	
110	12	Valid data is 12 stages or more	
111	16	Valid data is 16 stages	Largest

The number of stages of the FIFO is always sixteen even if the data area or empty area exceeds the FIFO size (the number of FIFOs). Accordingly, an overflow error or underflow error occurs if data area or empty area exceeds sixteen FIFO stages. The FIFO transmit or receive request is canceled when the above condition is not satisfied even if the FIFO is not empty or full.

(3) Number of FIFOs

The number of FIFO stages used in transmission and reception is indicated by the following register.

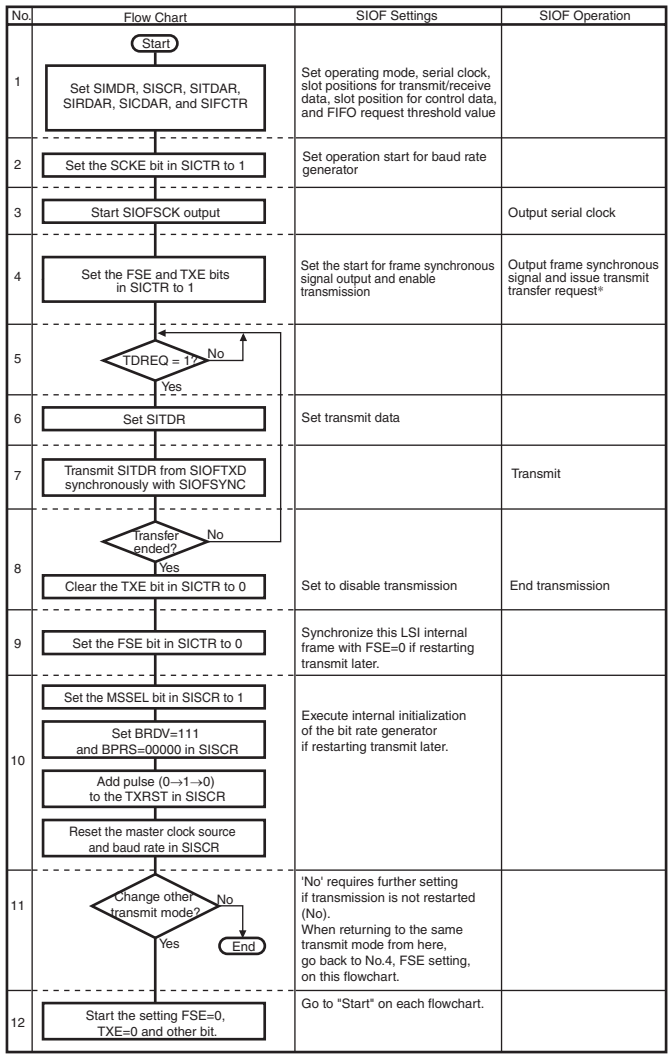
- Transmit FIFO: The number of empty FIFO stages is indicated by the bits TFUA[4:0] in SIFCTR.
- Receive FIFO: The number of valid data stages is indicated by the bits RFUA[4:0] in SIFCTR.

The above indicate possible data numbers that can be transferred by the CPU or DMAC.

21.4.7 Transmit and Receive Procedures

(1) Transmission in Master Mode

Figure 21.9 shows an example of settings and operation for master mode transmission.



Note: * When interrupts due to transmit data underflow are enabled, after setting the no. 6 transmit data, the TXE bit should be set to 1.

Figure 21.9 Example of Transmit Operation in Master Mode

(2) Reception in Master Mode

Figure 21.10 shows an example of settings and operation for master mode reception.

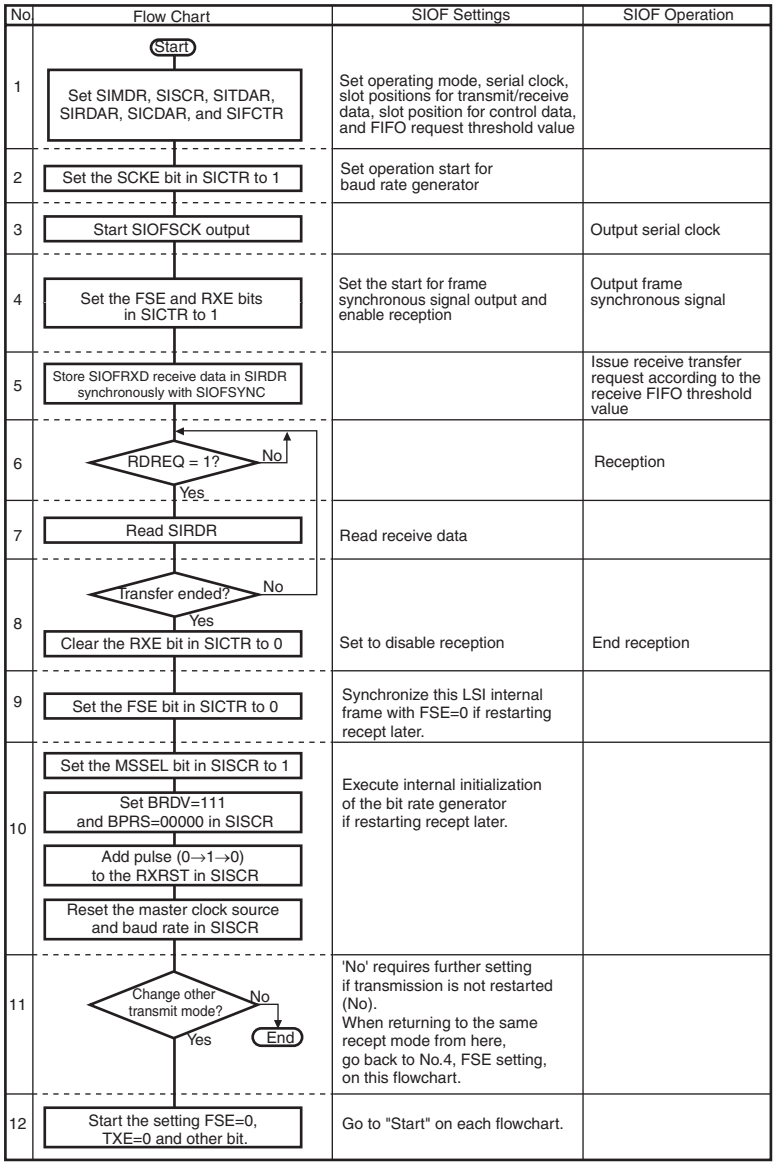


Figure 21.10 Example of Receive Operation in Master Mode

(3) Transmission in Slave Mode

Figure 21.11 shows an example of settings and operation for slave mode transmission.

No.	Flow Chart	SIOF Settings	SIOF Operation
1	Start Set SIMDR, SISCR, SITDAR, SIRDAR, SICDAR, and SIFCTR	Set operating mode, serial clock, slot positions for transmit/receive data, slot position for control data, and FIFO request threshold value	
2	Set the TXE bit in SICTR to 1	Enable transmission	Issue transmit transfer request to enable transmission when frame synchronization signal is input
3	TDREQ = 1? No Yes		
4	Set SITDR	Set transmit data	
5	Transmit SITDR from SIOFTXD synchronously with SIOFSYNC		Transmit
6	Transfer ended? No Yes Clear the TXE bit in SICTR to 0 End	Disable transmission	End transmission

Figure 21.11 Example of Transmit Operation in Slave Mode

(4) Reception in Slave Mode

Figure 21.12 shows an example of settings and operation for slave mode reception.

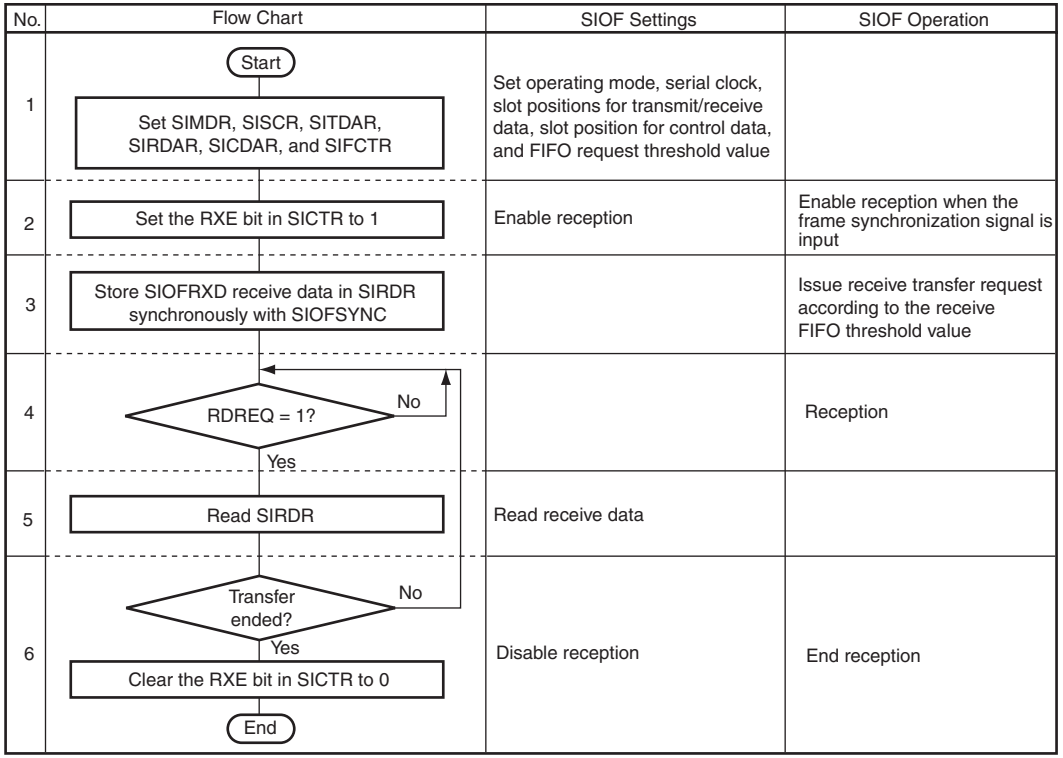


Figure 21.12 Example of Receive Operation in Slave Mode

(5) Transmit/Receive Reset

The SIOF can separately reset the transmit and receive units by setting the following bits to 1.

- Transmit reset: TXRST bit in SICTR
- Receive reset: RXRST bit in SICTR

Table 21.13 shows the details of initialization upon transmit or receive reset.

Table 21.13 Transmit and Receive Reset

Type	Objects Initialized
Transmit reset	Transmit FIFO write pointer
	TCRDY, TFEMP, and TDREQ bits in SISTR
	TXE bit in SICTR
Receive reset	Receive FIFO write pointer
	RCRDY, RFFUL, and RDREQ bits in SISTR
	RXE bit in SICTR

Notes: Refer to the following procedure to operate the transmit reset/receive reset.

1. Set the master clock source in the peripheral clock. (Write 1 (master clock = Pϕ (peripheral clock)) to the MSSEL bit in the SISCR register).
2. Set the prescaler count value of the baud rate generator to 1/1. (Write "00000" (division ratio = 1/1) to BRPS bits 4 to 0 in the SISCR register).
3. Set the division ratio in the bit rate generator's output level to 1/1. (Write "111" (division ratio =1/1) to BRDV bits 2 to 0 in the SISCR register).
4. Reset transmit/receive operation. (To reset, write "1" to the TXRST or RXRST bit in the SICTR register).

21.4.8 Interrupts

The SIOF has one type of interrupt.

(1) Interrupt Sources

Interrupts can be issued by several sources. Each source is shown as an SIOF status in SISTR. Table 21.14 lists the SIOF interrupt sources.

Table 21.14 SIOF Interrupt Sources

No.	Classification	Bit Name	Function Name	Description
1	Transmission	TDREQ	Transmit FIFO transfer request	The transmit FIFO empty area exceeds specified size.
2		TFEMP	Transmit FIFO empty	The transmit FIFO is empty.
3	Reception	RDREQ	Receive FIFO transfer request	The receive FIFO stores data of specified size or more.
4		RFFUL	Receive FIFO full	The receive FIFO is full.
5	Control	TCRDY	Transmit control data ready	The transmit control register is ready to be written.
6		RCRDY	Receive control data ready	The receive control data register stores valid data.
7	Error	TFUDF	Transmit FIFO underflow	Serial data transmit timing has arrived while the transmit FIFO is empty.
8		TFOVF	Transmit FIFO overflow	Write to the transmit FIFO is performed while the transmit FIFO is full.
9		RFOVF	Receive FIFO overflow	Serial data is received while the receive FIFO is full.
10		RFUDF	Receive FIFO underflow	The receive FIFO is read while the receive FIFO is empty.
11		FSERR	FS error	A synchronous signal is input before the specified bit number has been passed (in slave mode).
12		SAERR	Assign error	The same slot is specified in both serial data and control data.

Whether an interrupt is issued or not as the result of an interrupt source is determined by the SIIER settings. If an interrupt source is set to 1 and the corresponding bit in SIIER is set to 1, an SIOF interrupt is issued.

(2) Regarding Transmit and Receive Classification

The transmit sources and receive sources are signals indicating the state; after being set, if the state changes, they are automatically cleared by the SIOF.

When the DMA transfer is used, a DMA transfer request is pulled low (0 level) for one cycle at the end of DMA transfer.

(3) Processing when Errors Occur

On occurrence of each of the errors indicated as a status in SISTR, the SIOF performs the following operations.

- Transmit FIFO underflow (TFUDF)
  The immediately preceding transmit data is again transmitted.
- Transmit FIFO overflow (TFOVF)
  The contents of the transmit FIFO are protected, and the write operation causing the overflow is ignored.
- Receive FIFO overflow (RFOVF)
  Data causing the overflow is discarded and lost.
- Receive FIFO underflow (RFUDF)
  An undefined value is output on the bus.
- FS error (FSERR)
  The internal counter is reset according to the signal in which an error occurs.
- Assign error (SAERR)
  — If the same slot is assigned to both serial data and control data, the slot is assigned to serial data.
  — If the same slot is assigned to two control data items, data cannot be transferred correctly.

21.4.9 Transmit and Receive Timing

Examples of the SIOF serial transmission and reception are shown in figures 21.13 to 21.20.

(1) 8-Bit Monaural Data (1)

Synchronous pulse method, falling edge sampling, slot No.0 used for transmit and receive data, an frame length = 8 bits

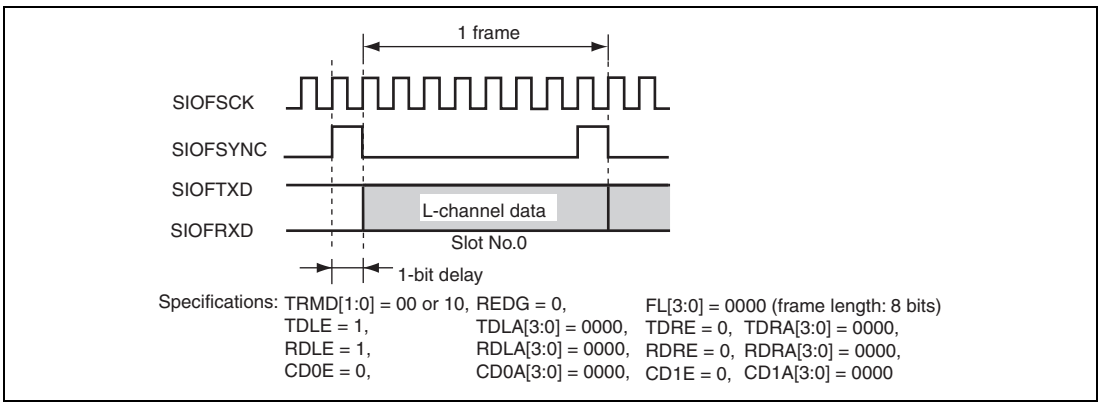


Figure 21.13 Transmit and Receive Timing (8-Bit Monaural Data (1))

(2) 8-Bit Monaural Data (2)

Synchronous pulse method, falling edge sampling, slot No.0 used for transmit and receive data, and frame length = 16 bits

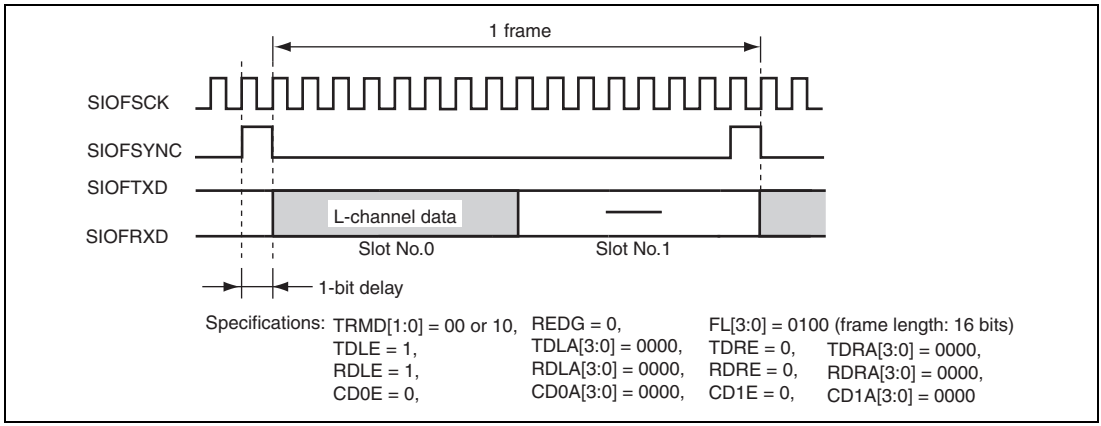


Figure 21.14 Transmit and Receive Timing (8-Bit Monaural Data (2))

(3) 16-Bit Monaural Data

Synchronous pulse method, falling edge sampling, slot No.0 used for transmit and receive data, and frame length = 64 bits

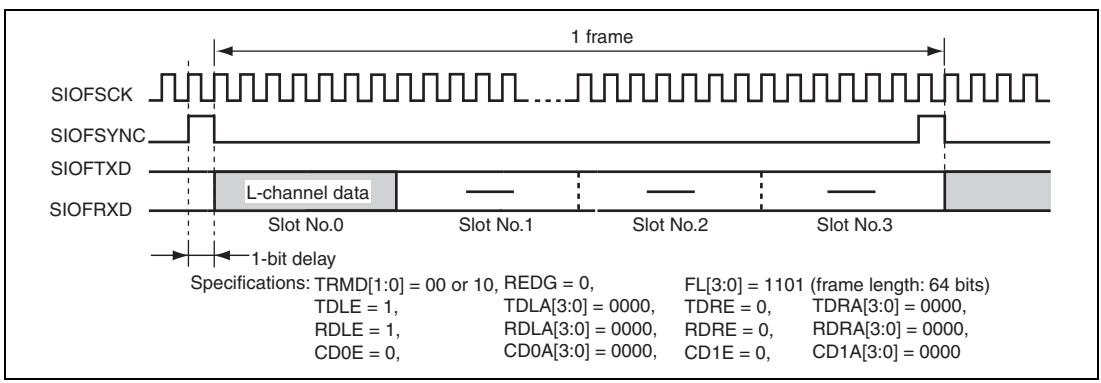


Figure 21.15 Transmit and Receive Timing (16-Bit Monaural Data)

(4) 16-Bit Stereo Data (1)

L/R method, rising edge sampling, slot No.0 used for left channel data, slot No.1 used for right channel data, and frame length = 32 bits

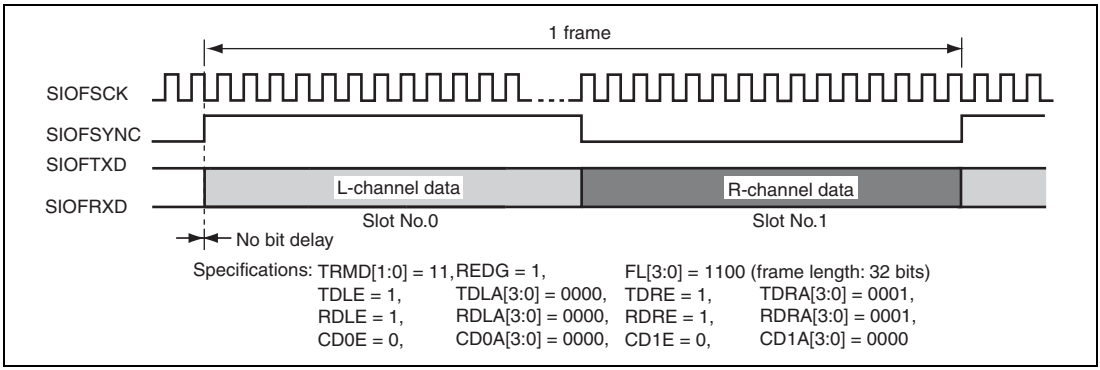


Figure 21.16 Transmit and Receive Timing (16-Bit Stereo Data (1))

(5) 16-Bit Stereo Data (2)

L/R method, rising edge sampling, slot No.0 used for left-channel transmit data, slot No.1 used for left-channel receive data, slot No.2 used for right-channel transmit data, slot No.3 used for right-channel receive data, and frame length = 64 bits

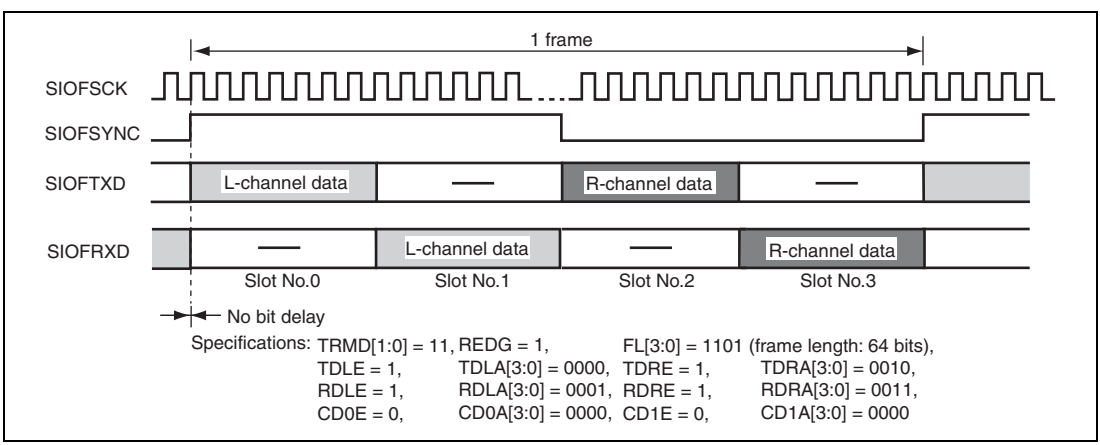


Figure 21.17 Transmit and Receive Timing (16-Bit Stereo Data (2))

(6) 16-Bit Stereo Data (3)

Synchronous pulse method, falling edge sampling, slot No.0 used for left-channel data, slot No.1 used for right-channel data, slot No.2 used for control data for channel 0, slot No.3 used for control data for channel 1, and frame length = 128 bits

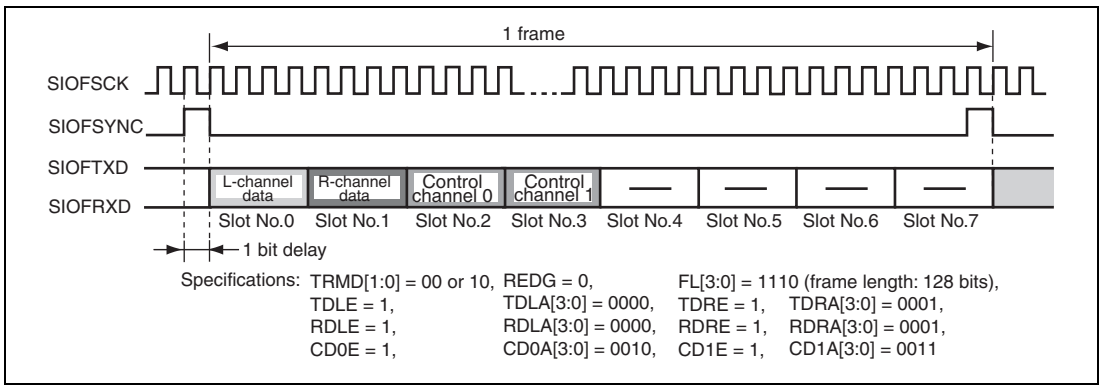


Figure 21.18 Transmit and Receive Timing (16-Bit Stereo Data (3))

(7) 16-Bit Stereo Data (4)

Synchronous pulse method, falling edge sampling, slot No.0 used for left-channel data, slot No.2 used for right-channel data, slot No.1 used for control data for channel 0 , slot No.3 used for control data for channel 1, and frame length = 128 bits

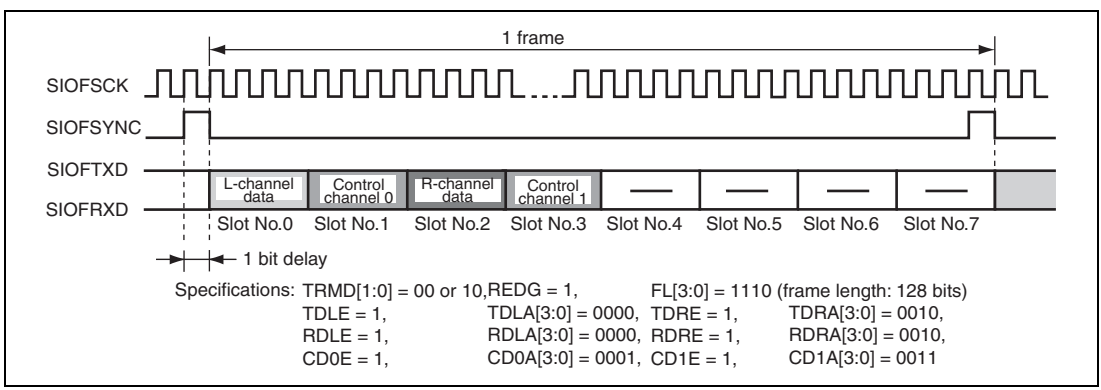


Figure 21.19 Transmit and Receive Timing (16-Bit Stereo Data (4))

(8) Synchronization-Pulse Output Mode at End of Each Slot (SYNCAT Bit = 1)

Synchronous pulse method, falling edge sampling, slot No.0 used for left-channel data, slot No.1 used for right-channel data, slot No.2 used for control data for channel 0, slot No.3 used for control data for channel 1, and frame length = 128 bits

In this mode, valid data must be set to slot No.0.

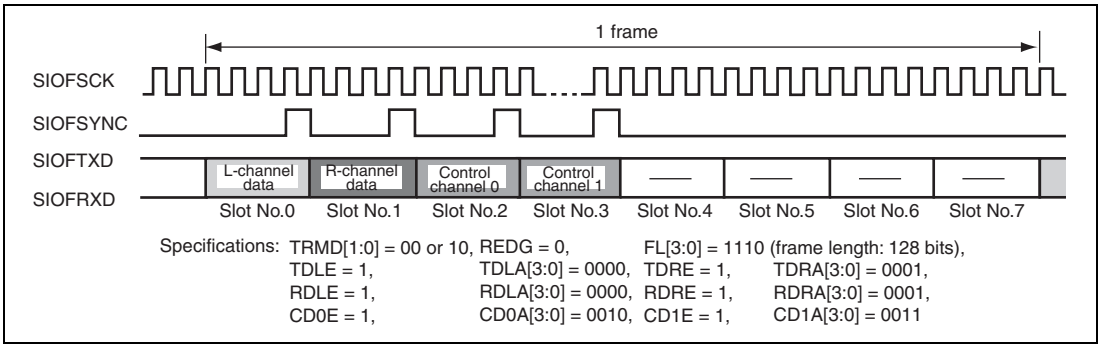


Figure 21.20 Transmit and Receive Timing (16-Bit Stereo Data)

21.5 Usage Note

Pay attention to the following point if you are using the SIOF.

21.5.1 Note on Data Reception in Slave Mode

The following note applies to data reception in slave mode.

In SIOF data reception in slave mode, when a sampling timing with no delay is selected (SIMDR.SYNCDL = 0), since data being received are sampled on rising edges of the SIOF_SCK signal, the hold time for data reception may not be secured.

To avoid this, in both the reception and transmission of data in slave mode, select one bit of delay for the timing of sampling (SIMDR.SYNCDL = 1) with sampling of received data on falling edges of the SIOF_SCK signal.

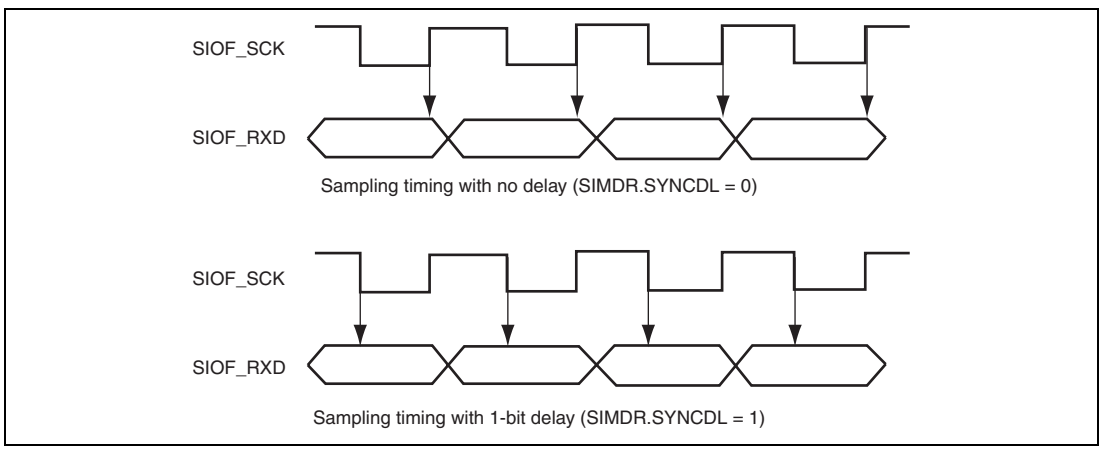


Figure 21.21 Timing of Sampling for Data Reception in Slave Mode

21.5.2 Note on Interrupting Transfers

When a transfer is in progress, do not clear the FSE bit in the SICTR register to 0 to interrupt it.

To interrupt a transfer, clear the TXE or RXE bit in the SICTR register to 0.

Section 22 Serial Communication Interface with FIFO (SCIF)

This LSI has a four-channel serial communication interface with FIFO (SCIF) that supports both asynchronous and clock synchronous serial communication. It also has 16-stage FIFO registers for both transmission and reception independently for each channel that enable this LSI to perform efficient high-speed continuous communication.

22.1 Features

- Asynchronous serial communication:
  - Serial data communication is performed by start-stop in character units. The SCIF can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Receive error detection: Parity, framing, and overrun errors
  - Break detection: Break is detected when a framing error is followed by at least one frame at the space 0 level (low level).
- Clock synchronous serial communication:
  - Serial data communication is synchronized with a clock signal. The SCIF can communicate with other chips having a clock synchronous communication function. There is one serial data communication format.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCIF can transmit and receive simultaneously. Both sections use 16-stage FIFO buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates

- Internal or external transmit/receive clock source: From either baud rate generator (internal) or SCK pin (external)
  Four types of interrupts: Transmit-FIFO-data-empty interrupt, break interrupt, receive-FIFO-data-full interrupt, and receive-error interrupts are requested independently on each channel.
- When the transmit FIFO is empty or the receive FIFO contains any received data, the DMA controller (DMAC) can be activated to perform data transfer by generating a DMA transfer request.
- When the SCIF is not in use, it can be stopped by halting the clock supplied to it, saving power.
- In asynchronous mode, on-chip modem control functions ($\overline{\text{RTS}}$ and $\overline{\text{CTS}}$) (channels 2 and 3).
- The quantity of data in the transmit and receive FIFO data registers and the number of receive errors of the receive data in the receive FIFO data register can be ascertained.
- A time-out error (DR) can be detected when receiving in asynchronous mode.

Figure 22.1 shows a block diagram of the SCIF.

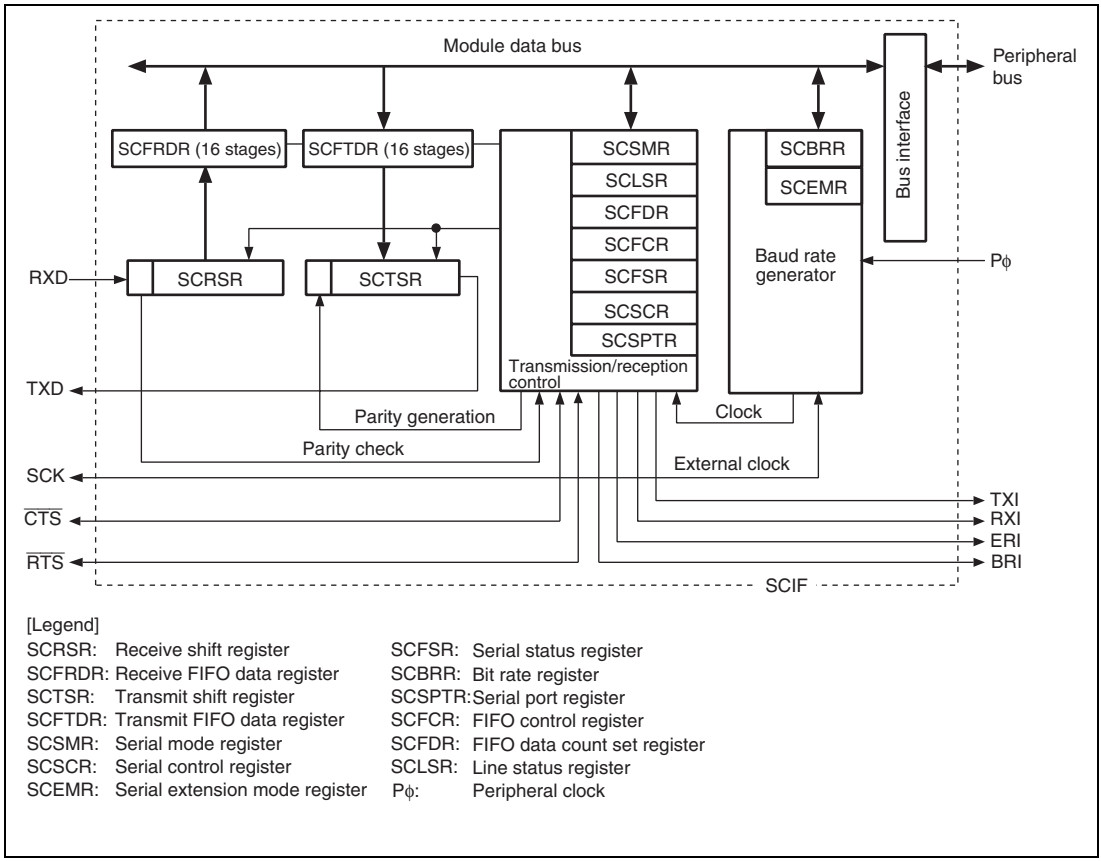


Figure 22.1 Block Diagram of SCIF

22.2 Input/Output Pins

Table 22.1 shows the pin configuration of the SCIF.

Table 22.1 Pin Configuration

Channel	Pin Name	Function	I/O	Description
0	SCIF0_TXD	Transmit data	Output	Transmit data pin
	SCIF0_RXD	Receive data	Input	Receive data pin
	SCIF0_SCK	Serial clock	I/O	Clock I/O pin
1	SCIF1_TXD	Transmit data	Output	Transmit data pin
	SCIF1_RXD	Receive data	Input	Receive data pin
	SCIF1_SCK	Serial clock	I/O	Clock I/O pin
2	SCIF2_TXD	Transmit data	Output	Transmit data pin
	SCIF2_RXD	Receive data	Input	Receive data pin
	SCIF2_SCK	Serial clock	I/O	Clock I/O pin
	$\overline{\text{SCIF2\_RTS}}$	Modem control	Output	RTS output pin
	$\overline{\text{SCIF2\_CTS}}$	Modem control	Input	CTS input pin
3	SCIF3_TXD	Transmit data	Output	Transmit data pin
	SCIF3_RXD	Receive data	Input	Receive data pin
	SCIF3_SCK	Serial clock	I/O	Clock I/O pin
	$\overline{\text{SCIF3\_RTS}}$	Modem control	Output	RTS output pin
	$\overline{\text{SCIF3\_CTS}}$	Modem control	Input	CTS input pin

Note: In the following descriptions, channel numbers in pin names and signal names are omitted and TXD, RXD, SCK, $\overline{\text{RTS}}$, and $\overline{\text{CTS}}$ are used as generic terms.

22.3 Register Descriptions

The SCIF has the following registers. Table 22.3 shows the register states in each operating mode. Note that the channel numbers are omitted from the register names in descriptions subsequent to these tables.

Table 22.2 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
Serial mode register 0	SCSMR0	R/W	H'FFE0 0000	16
Bit rate register 0	SCBRR0	R/W	H'FFE0 0004	8
Serial control register 0	SCSCR0	R/W	H'FFE0 0008	16
Transmit FIFO data register 0	SCFTDR0	W	H'FFE0 000C	8
Serial status register 0	SCFSR0	R/W*[1]	H'FFE0 0010	16
Receive FIFO data register 0	SCFRDR0	R	H'FFE0 0014	8
FIFO control register 0	SCFCR0	R/W	H'FFE0 0018	16
FIFO data count register 0	SCFDR0	R	H'FFE0 001C	16
Line status register 0	SCLSR0	R/W*[2]	H'FFE0 0024	16
Serial mode register 1	SCSMR1	R/W	H'FFE1 0000	16
Bit rate register 1	SCBRR1	R/W	H'FFE1 0004	8
Serial control register 1	SCSCR1	R/W	H'FFE1 0008	16
Transmit FIFO data register 1	SCFTDR1	W	H'FFE1 000C	8
Serial status register 1	SCFSR1	R/W*[1]	H'FFE1 0010	16
Receive FIFO data register 1	SCFRDR1	R	H'FFE1 0014	8
FIFO control register 1	SCFCR1	R/W	H'FFE1 0018	16
FIFO data count register 1	SCFDR1	R	H'FFE1 001C	16
Line status register 1	SCLSR1	R/W*[2]	H'FFE1 0024	16
Serial mode register 2	SCSMR2	R/W	H'FFE2 0000	16
Bit rate register 2	SCBRR2	R/W	H'FFE2 0004	8
Serial control register 2	SCSCR2	R/W	H'FFE2 0008	16
Transmit FIFO data register 2	SCFTDR2	W	H'FFE2 000C	8
Serial status register 2	SCFSR2	R/W*[1]	H'FFE2 0010	16
Receive FIFO data register 2	SCFRDR2	R	H'FFE2 0014	8
FIFO control register 2	SCFCR2	R/W	H'FFE2 0018	16
FIFO data count register 2	SCFDR2	R	H'FFE2 001C	16

Register Name	Abbreviation	R/W	Address	Access Size
Line status register 2	SCLSR2	R/W*2	H'FFE2 0024	16
Serial mode register 3	SCSMR3	R/W	H'FFE3 0000	16
Bit rate register 3	SCBRR3	R/W	H'FFE3 0004	8
Serial control register 3	SCSCR3	R/W	H'FFE3 0008	16
Transmit FIFO data register 3	SCFTDR3	W	H'FFE3 000C	8
Serial status register 3	SCFSR3	R/W*1	H'FFE3 0010	16
Receive FIFO data register 3	SCFRDR3	R	H'FFE3 0014	8
FIFO control register 3	SCFCR3	R/W	H'FFE3 0018	16
FIFO data count register 3	SCFDR3	R	H'FFE3 001C	16
Line status register 3	SCLSR3	R/W*2	H'FFE3 0024	16

Notes: 1. To bits 7 to 4, 1, and 0, only 0 can be written to clear the flag.
2. To bit 0, only 0 can be written to clear the flag.

Table 22.3 Register States in Each Operating Mode

Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
SCSMR0	Initialized	Retained	Retained	Retained
SCBRR0	Initialized	Retained	Retained	Retained
SCSCR0	Initialized	Retained	Retained	Retained
SCFTDR0	Initialized	Retained	Retained	Retained
SCFSR0	Initialized	Retained	Retained	Retained
SCFRDR0	Initialized	Retained	Retained	Retained
SCFCR0	Initialized	Retained	Retained	Retained
SCFDR0	Initialized	Retained	Retained	Retained
SCLSR0	Initialized	Retained	Retained	Retained
SCSMR1	Initialized	Retained	Retained	Retained
SCBRR1	Initialized	Retained	Retained	Retained
SCSCR1	Initialized	Retained	Retained	Retained
SCFTDR1	Initialized	Retained	Retained	Retained
SCFSR1	Initialized	Retained	Retained	Retained
SCFRDR1	Initialized	Retained	Retained	Retained
SCFCR1	Initialized	Retained	Retained	Retained
SCFDR1	Initialized	Retained	Retained	Retained
SCLSR1	Initialized	Retained	Retained	Retained
SCSMR2	Initialized	Retained	Retained	Retained
SCBRR2	Initialized	Retained	Retained	Retained
SCSCR2	Initialized	Retained	Retained	Retained
SCFTDR2	Initialized	Retained	Retained	Retained
SCFSR2	Initialized	Retained	Retained	Retained
SCFRDR2	Initialized	Retained	Retained	Retained
SCFCR2	Initialized	Retained	Retained	Retained
SCFDR2	Initialized	Retained	Retained	Retained
SCLSR2	Initialized	Retained	Retained	Retained

Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
SCSMR3	Initialized	Retained	Retained	Retained
SCBRR3	Initialized	Retained	Retained	Retained
SCSCR3	Initialized	Retained	Retained	Retained
SCFTDR3	Initialized	Retained	Retained	Retained
SCFSR3	Initialized	Retained	Retained	Retained
SCFRDR3	Initialized	Retained	Retained	Retained
SCFCR3	Initialized	Retained	Retained	Retained
SCFDR3	Initialized	Retained	Retained	Retained
SCLSR3	Initialized	Retained	Retained	Retained

22.3.1 Receive Shift Register (SCRSR)

SCRSR receives serial data. Data input at the RXD pin is loaded into SCRSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to SCFRDR.

The CPU cannot read or write to SCRSR directly.

22.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is an 8-bit length16-stage FIFO register that stores serial receive data. The SCIF completes the reception of one byte of serial data by moving the received data from SCRSR into SCFRDR for storage. Thereafter, SCRSR becomes ready for next reception and continuous reception is possible until 16 bytes are stored, which makes SCFRDR full. The CPU can read but not write to SCFRDR. If data is read when there is no receive data in the SCFRDR, the value is undefined.

When SCFRDR is full of receive data, subsequent serial data is lost.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SCFRD[7:0]	Undefined	R	FIFO for serial receive data

22.3.3 Transmit Shift Register (SCTSR)

SCTSR is used to transmit serial data. The SCIF loads transmit data from SCFTDR into SCTSR, then transmits the data serially from the TXD pin, LSB (bit 0) first. After transmitting one data byte, the SCIF automatically loads the next transmit data from SCFTDR into SCTSR and starts transmitting again.

The CPU cannot read from or write to SCTSR directly.

22.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is an 8-bit length16-stage FIFO register that stores data for serial transmission. When data for transmission is written to SCFTDR while the transmit shift register (SCTSR) is empty, the SCIF transfers the data written to SCFTDR to SCTSR and starts serial transmission. Continuous serial transmission can be performed until there is no transmit data left in SCFTDR.

SCFTDR is write-only and cannot be read by the CPU. When SCFTDR is full (16 bytes), no more data can be written. If writing of new data is attempted, the data is ignored.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SCFTD[7:0]	Undefined	W	FIFO for serial transmit data

22.3.5 Serial Mode Register (SCSMR)

SCSMR is a 16-bit register that specifies the serial communication format of the SCIF and selects the clock source for the baud rate generator.

The CPU can always read from and write to SCSMR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CA	CHR	PE	OE	STOP	—	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	CA	0	R/W	Communication Mode Selects whether the SCIF operates in asynchronous or clock synchronous mode. 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	0	R/W	Character Length Selects 7-bit or 8-bit data length in asynchronous mode. When 7-bit data is selected, the MSB (bit 7) of the transmit FIFO data register is not transmitted. In clock synchronous mode, the data length is always 8 bits, regardless of the CHR setting. 0: 8-bit data 1: 7-bit data
5	PE	0	R/W	Parity Enable Selects whether to add a parity bit to transmit data and to check the parity of receive data, in asynchronous mode. In clock synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting. 0: Parity bit not added or checked 1: Parity bit added and checked* Note: * When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (OE) setting. Receive data parity is checked according to the even/odd (OE) mode setting.

Bit	Bit Name	Initial Value	R/W	Description
4	OE	0	R/W	Parity Mode Selects even or odd parity when parity bits are added and checked. The OE setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and checking. The OE setting is ignored in clock synchronous mode, or in asynchronous mode when parity addition and checking is disabled. 0: Even parity*1 1: Odd parity*2 Notes: 1. If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined. 2. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.

Bit	Bit Name	Initial Value	R/W	Description
3	STOP	0	R/W	Stop Bit Length Selects one or two bits as the stop bit length in asynchronous mode. When receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character. The setting of this bit is only valid in asynchronous mode. It is ignored in clock synchronous mode because no stop bits are added. 0: One stop bit*[1] 1: Two stop bits*[2] Notes: 1. When transmitting, a single 1-valued bit is added at the end of each character for transmission. 2. When transmitting, two 1-valued bits are added at the end of each character for transmission.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	CKS[1:0]	00	R/W	Clock Select Select the internal clock source of the on-chip baud rate generator. 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

22.3.6 Serial Control Register (SCSCR)

SCSCR operates the SCIF transmitter/receiver, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write to SCSCR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TIE	RIE	TE	RE	REIE	—	CKE[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables generation of transmit-FIFO-data-empty interrupt (TXI) requests when the serial transmit data is transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), when the quantity of data in SCFTDR becomes less than the specified number of transmission triggers, and when the TDFE flag in the serial status register (SCFSR) is set to1. The TXI interrupt request can be cleared either by writing to SCFTDR a greater quantity of transmit data than the specified transmission trigger number after reading 1 from the TDFE flag and then clearing TDFE to 0, or by clearing TIE to 0. 0: Transmit-FIFO-data-empty interrupt request (TXI) is disabled 1: Transmit-FIFO-data-empty interrupt request (TXI) is enabled

Bit	Bit Name	Initial Value	R/W	Description
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive FIFO data full (RXI) interrupts requested when the RDF flag or DR flag in serial status register (SCFSR) is set to1, receive-error (ERI) interrupts requested when the ER flag in SCFSR is set to1, and break (BRI) interrupts requested when the BRK flag in SCFSR or the ORER flag in line status register (SCLSR) is set to1. RXI interrupt requests can be cleared by reading the DR or RDF flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0. ERI or BRI interrupt requests can be cleared by reading the ER, BR or ORER flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE and REIE to 0. 0: Receive FIFO data full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are disabled 1: Receive FIFO data full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are enabled
5	TE	0	R/W	Transmit Enable Enables or disables serial transmission by the SCIF. With TE set to 1, serial transmission starts when transmit data is written to SCFTDR. 0: Transmission disabled 1: Transmission enabled* Note: * Select the transmit format in SCSMR and SCFCR and reset the transmit FIFO before setting TE to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	RE	0	R/W	Receive Enable Enables or disables serial reception by the SCIF. With RE set to 1, serial reception starts when a start bit is detected in asynchronous mode, or synchronous clock is detected in clock synchronous mode. 0: Reception disabled*[1] 1: Reception enabled*[2] Notes: 1. Clearing RE to 0 does not affect the receive flags (DR, ER, BRK, RDF, FER, PER, and ORER). These flags retain their previous values. 2. Select the receive format in SCSMR and SCFCR and reset the receive FIFO before setting RE to 1.
3	REIE	0	R/W	Receive Error Interrupt Enable Enables or disables the receive-error (ERI) interrupts and break (BRI) interrupts. The setting of REIE bit is valid only when RIE bit is set to 0. ERI or BRI interrupt requests can be cleared by reading the ER, BR or ORER flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE and REIE to 0. Even if RIE is set to 0, when REIE is set to 1, ERI or BRI interrupt requests are enabled. Set so If SCIF wants to inform INTC of ERI or BRI interrupt requests during DMA transfer. 0: Receive-error interrupt (ERI) and break interrupt (BRI) requests are disabled 1: Receive-error interrupt (ERI) and break interrupt (BRI) requests are enabled*

Bit	Bit Name	Initial Value	R/W	Description
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	CKE[1:0]	00	R/W	Clock Enable Select the SCIF clock source and enable or disable clock output from the SCK pin. Depending on CKE[1:0], the SCK pin can be used for serial clock output or serial clock input. CKE[1:0] must be set before selecting the operating mode of the SCIF by the SCSMR register. • Asynchronous mode 00: Internal clock; SCK pin is used as an input pin (input signal is ignored) 01: Setting prohibited 10: External clock; SCK pin used for clock input*1 11: Setting prohibited • Clock synchronous mode 00: Setting prohibited 01: Internal clock; SCK pin used for serial clock output*2 10: External clock; SCK pin used for serial clock input 11: Setting prohibited Notes: 1. The input clock frequency is 16 times the bit rate. 2. The output clock frequency is the same as the bit rate.

22.3.7 Serial Status Register (SCFSR)

SCFSR is a 16-bit register. The upper 8 bits indicate the number of receive errors in the receive FIFO data register (SCFRDR), and the lower 8 bits indicate the states of SCIF operation.

The CPU can always read the upper 8 bits of SCFSR and always read and write from/to the lower 8 bits. However, it cannot write 1 to the flags ER, TEND, TDFE, BRK, RDF, and DR. These flags can be cleared to 0 only after 1 has been read from them. The PER flag and FER flag are read-only and cannot be written to.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PERC[3:0]				FERC[3:0]				ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Initial value:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W*	R/W*	R/W*	R/W*	R	R	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PERC[3:0]	0000	R	Number of Parity Errors Indicate the number of data bytes including a parity error in the receive data stored in the receive FIFO data register (SCFRDR). After the ER bit in SCFSR is set, the value in PERC[3:0] indicates the number of parity errors in SCFRDR. When parity errors have been found in all bytes in the 16 bytes of receive data in SCFRDR, PERC[3:0] shows 0000.
11 to 8	FERC[3:0]	0000	R	Number of Framing Errors Indicate the number of data bytes including a framing error in the receive data stored in SCFRDR. After the ER bit in SCFSR is set, the value in FERC[3:0] indicates the number of framing errors in SCFRDR. When framing errors have been found in all bytes in the 16 bytes of receive data in SCFRDR, FERC[3:0] shows 0000.

Bit	Bit Name	Initial Value	R/W	Description
7	ER	0	R/W*	Receive Error Indicates the occurrence of a framing error or a parity error during reception. Clearing the RE bit to 0 in SCSCR does not affect the ER bit, which retains its value. Even if a receive error has occurred, the receive data is transferred to SCFRDR and the receive operation is continued. Whether or not the data read from SCFRDR includes any error is shown in the FER and PER bits in SCFSR. 0: A framing error or parity error has not occurred during reception. [Clearing conditions] • Power-on reset or manual reset • A 0 is written to ER after 1 is read from it 1: A framing error or parity error has occurred during reception. [Setting conditions] • At the end of reception, the stop bit of the last byte of receive data is checked and it is found to be 0. In two stop-bit mode, only the first stop bit is checked and the second one is not checked. • The total number of 1s in the receive data plus parity bit does not match the even/odd parity specified by the OE bit in SCSMR

Bit	Bit Name	Initial Value	R/W	Description
6	TEND	1	R/W*	Transmit End Indicates that when the last bit of a serial character was transmitted, SCFTDR did not contain valid data, so transmission has ended. 0: Transmission is in progress [Clearing condition] • When 0 is written to TEND after 1 is read from it after transmit data is written to SCFTDR • Data is written to SCFTDR by the DMAC 1: Transmission has ended [Setting conditions] • Power-on reset or manual reset • TE in the serial control register (SCSCR) is clear • SCFTDR does not contain any transmit data when the last bit of a one-byte serial character is transmitted

Bit	Bit Name	Initial Value	R/W	Description
5	TDFE	1	R/W*	Transmit FIFO Data Empty Indicates that writing of transmit data to SCFTDR has been enabled after data is transferred from SCFTDR to SCTSR and the quantity of data in SCFTDR has become equal to or less than the transmission trigger number specified by the TTRG bits in SCFCR. 0: The quantity of transmit data written to SCFTDR is greater than the specified transmission trigger number [Clearing conditions] • After 1 is read from TDFE, data of the quantity exceeding the specified transmission trigger number is written to SCFTDR and then 0 is written to TDFE • Data of the quantity exceeding the specified transmission trigger number is written to SCFTDR by the DMAC 1: The quantity of transmit data in SCFTDR is equal to or less than the specified transmission trigger number [Setting conditions] • Power-on reset or manual reset • The quantity of transmit data in SCFTDR becomes equal to or less than the specified transmission trigger number as a result of transmission Note: Since SCFTDR is a 16-byte FIFO register, the maximum quantity of data that can be written when TDFE is 1 is "16 minus the specified transmission trigger number". If an attempt is made to write more data, the excess data is ignored. The quantity of data in SCFTDR is indicated by the upper 8 bits of SCFDR.

Bit	Bit Name	Initial Value	R/W	Description
4	BRK	0	R/W*	Break Detection Indicates that a break signal has been detected in receive data. 0: No break signal has been received [Clearing conditions] • Power-on reset or manual reset • When 0 is written to BRK after 1 is read from it 1: Break signal has been received After a break is detected, transfer of the receive data (H'00) to SCFRDR stops. When the break ends and the receive signal becomes mark (=1), the transfer of receive data resumes. [Setting condition] • Data causing a framing error is received, and space (=0, ie, low level) lasts for one or more frame length in the subsequent reception
3	FER	0	R	Framing Error Indication In asynchronous mode, indicates a framing error in the received data that is to be read from SCFRDR next. 0: No framing error has occurred in the receive data to be read from SCFRDR next. [Clearing conditions] • Power-on reset or manual reset • No framing error is found in the data to be read from SCFRDR next 1: A framing error has occurred in the next receive data to be read from SCFRDR next. [Setting condition] • A framing error is found in the data to be read from SCFRDR next

Bit	Bit Name	Initial Value	R/W	Description
2	PER	0	R	Parity Error Indication In asynchronous mode, indicates a parity error in the received data that is to be read from SCFRDR next. 0: No parity error has been found in the receive data to be read from SCFRDR next. [Clearing conditions] • Power-on reset or manual reset • No parity error is found in the data to be read from SCFRDR next 1: A parity error has occurred in the next receive data to be read from SCFRDR next. [Setting condition] • A parity error is found in the data to be read from SCFRDR next

Bit	Bit Name	Initial Value	R/W	Description
1	RDF	0	R/W*	Receive FIFO Data Full Indicates that receive data in SCRSR has been transferred to SCFRDR and the quantity of data in SCFRDR has become equal to or more than the receive trigger number specified by the RTRG bits in SCFCR. 0: The quantity of receive data in SCFRDR is less than the specified receive trigger number [Clearing conditions] • Power-on reset or manual reset • After 1 is read from RDF, SCFRDR is read until the quantity of receive data in SCFRDR becomes less than the specified receive trigger number and then 0 is written to RDF • SCFRDR is read by the DMAC until the quantity of receive data in SCFRDR becomes less than the specified receive trigger number 1: The quantity of receive data in SCFRDR is equal to or more than the specified receive trigger number [Setting condition] • Receive data of the quantity equal to or more than the specified receive trigger number is stored in SCFRDR Note: SCFRDR is a 16-byte FIFO register. At least, data of the specified receive trigger number can be read when RDF is 1. If an attempt is made to read after all the data in SCFRDR has been read, undefined data will be read. The quantity of receive data in SCFRDR is indicated by the lower 8 bits of SCFDR.

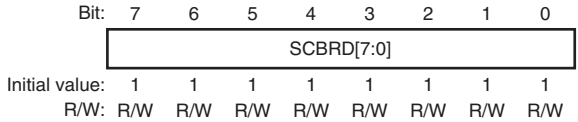
Bit	Bit Name	Initial Value	R/W	Description
0	DR	0	R/W*	Receive Data Ready Indicates that the quantity of data in SCFRDR is less than the specified receive trigger number, and that the next data has not yet been received after the elapse of 15 ETU from the last stop bit in asynchronous mode. In clock synchronous mode, this bit is not set to 1. 0: Reception is in progress, or no receive data remains in SCFRDR after reception ended normally. [Clearing conditions] • Power-on reset or manual reset • After 1 is read from DR, all receive data are read and then 0 is written to DR. • All receive data are read by the DMAC 1: Next data has not been received [Setting condition] • SCFRDR contains less data than the specified receive trigger number, and the next data has not yet been received after the elapse of 15 ETU from the last stop bit. Note: 15 ETU is equivalent to 1.5 frames with the 8-bit, 1-stop-bit format. (ETU: elementary time unit)

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

22.3.8 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that is used to set the bit rate of serial transmission/reception in relation to the operating clock of the baud rate generator selected by the CKS1[1:] bits in SCSMR.

The CPU can always read and write to SCBRR.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SCBRD[7:0]	H'FF	R/W	Bit rate setting

The SCBRR setting is calculated as follows:

- Asynchronous mode:

$$N = \{P\phi / (64 \times 2^{2n-1} \times B)\} \times 10^6 - 1$$

- Clock synchronous mode:

$$N = \{P\phi / (8 \times 2^{2n-1} \times B)\} \times 10^6 - 1$$

B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
(The setting must satisfy the electrical characteristics.)
Pϕ: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 22.4.)

Table 22.4 SCSMR Settings

		SCSMR Setting
n	**Clock Source**	**CKS[1:0]**
0	Pϕ	00
1	Pϕ/4	01
2	Pϕ/16	10
3	Pϕ/64	11

Note: The bit rate error in asynchronous mode is given by the following formula:

$$\text{Error (\%)} = \{\{P\phi / ((N + 1) \times 64 \times 2^{2n-1} \times B)\} \times 10^{6} - 1\} \times 100$$

22.3.9 FIFO Control Register (SCFCR)

SCFCR resets the quantity of data in the transmit and receive FIFO data registers, sets the trigger data quantity, and contains an enable bit for loop-back testing. SCFCR can always be read and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RSTRG[2:0]			RTRG[1:0]		TTRG[1:0]		MCE	TFRST	RFRST	LOOP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	RSTRG[2:0]	000	R/W	$\overline{\text{RTS}}$ Output Active Trigger When the quantity of receive data in SCFRDR becomes more than the number shown below, $\overline{\text{RTS}}$ signal is set to high. 000: 15 001: 1 010: 4 011: 6 100: 8 101: 10 110: 12 111: 14
7, 6	RTRG[1:0]	00	R/W	Receive FIFO Data Trigger Set the quantity of receive data at which the receive data full (RDF) flag in SCFSR is set. The RDF flag is set to 1 when the quantity of receive data stored in SCFRDR has become equal to or more than the set trigger number shown below as the reception proceeds. • Asynchronous mode 00: 1 01: 4 10: 8 11: 14 • Clock synchronous mode 00: 1 01: 2 10: 8 11: 14

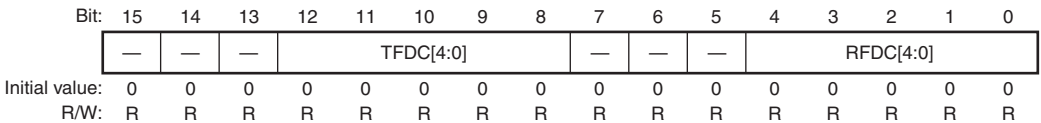
Bit	Bit Name	Initial Value	R/W	Description
5, 4	TTRG[1:0]	00	R/W	Transmit FIFO Data Trigger Set the quantity of remaining transmit data at which the transmit FIFO data register empty (TDFE) flag in SCFSR is set. The TDFE flag is set to 1 when the quantity of transmit data in SCFTDR has become equal to or less than the set trigger number shown below as the transmission proceeds. 00: 8 (8)* 01: 4 (12)* 10: 2 (14)* 11: 0 (16)* Note: * Values in parentheses mean the number of empty bytes in SCFTDR when the TDFE flag is set to 1.
3	MCE	0	R/W	Modem Control Enable Enables modem control signals $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$. In clock synchronous mode, MCE bit should always be 0. 0: Modem signal disabled* 1: Modem signal enabled Note: * $\overline{\text{CTS}}$ is fixed at active 0 regardless of the input value, and $\overline{\text{RTS}}$ is also fixed at 0.
2	TFRST	0	R/W	Transmit FIFO Data Register Reset Invalidates the transmit data in SCFTDR to reset SCFTDR in an empty state. 0: Resetting disabled* 1: Resetting enabled Note: * Resetting is performed by a power-on reset or manual reset, or when a standby state is entered.
1	RFRST	0	R/W	Receive FIFO Data Register Reset Invalidates the receive data in SCFRDR to reset SCFRDR in an empty state. 0: Resetting disabled* 1: Resetting enabled Note: * Resetting is performed by a power-on reset or manual reset, or when a standby state is entered.

Bit	Bit Name	Initial Value	R/W	Description
0	LOOP	0	R/W	Loop-Back Test Internally connects the transmit output pin (TXD) and receive input pin (RXD) and internally connects the $\overline{\text{RTS}}$ pin and $\overline{\text{CTS}}$ pin and enables loop-back testing. 0: Loop back test disabled 1: Loop back test enabled

22.3.10 FIFO Data Count Set Register (SCFDR)

SCFDR is a 16-bit register that indicates the quantity of data stored in SCFTDR and SCFRDR.

It indicates the quantity of transmit data in SCFTDR with the upper 8 bits, and the quantity of receive data in SCFRDR with the lower 8 bits. SCFDR can always be read by the CPU.



Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 8	TFDC[4:0]	00000	R	Number of Data Bytes in Transmit FIFO Indicate the quantity of non-transmitted data stored in SCFTDR. H'00 means no transmit data, and H'10 means that SCFTDR is full of transmit data (16 bytes).
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	RFDC[4:0]	00000	R	Number of Data Bytes in Receive FIFO Indicate the quantity of receive data stored in SCFRDR. H'00 means no receive data, and H'10 means that SCFRDR full of receive data (16 bytes).

22.3.11 Line Status Register (SCLSR)

The CPU can always read or write to SCLSR, but cannot write 1 to the ORER flag. This flag can be cleared to 0 only if it has first been read (after being set to 1).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ORER
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ORER	0	R/W*	Overrun Error Indicates the occurrence of an overrun error. 0: Receiving is in progress or has ended normally*[1] [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ORER after 1 is read from it 1: An overrun error has occurred*[2] [Setting condition] • Next serial reception is finished while SCFRDR is full of 16 bytes of receive data. Notes: 1. Clearing the RE bit to 0 in SCSCR does not affect the ORER bit, which retains the value before RE is cleared. 2. SCFRDR retains the data before an overrun error has occurred, and the next received data is discarded. When the ORER bit is set to 1, the SCIF cannot continue the next serial reception.

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

22.4 Operation

22.4.1 Overview

For serial communication, the SCIF has an asynchronous mode in which characters are synchronized individually, and a clock synchronous mode in which communication is synchronized with clock pulses.

The SCIF has a 16-stage FIFO buffer for both transmission and receptions, reducing the overhead of the CPU, and enabling continuous high-speed communication. Furthermore, the SCIF has $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ signals to be used as modem control signals.

The transmission/reception format is selected by SCSMR as shown in table 22.5. The SCK pin function is determined by the combination of the CA bit in SCSMR and the CKE[1:0] bits in SCSCR.

(1) Asynchronous Mode

- Data length is selectable: 7 or 8 bits
- Parity bit is selectable. So is the stop bit length (1 or 2 bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, receive FIFO data full, overrun errors, receive data ready, and breaks.
- The number of stored data bytes is indicated for both the transmit and receive FIFO registers.
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the clock of on-chip baud rate generator.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

(2) Clock Synchronous Mode

- Data length is 8 bits only.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the clock of the on-chip baud rate generator, and outputs this clock to external devices as the synchronous clock.
  - When an external clock is selected, the SCIF is driven by the synchronization clock that is input from the SCK pin.

Table 22.5 SCSMR Settings and SCIF Communication Formats

SCSMR Settings					SCIF Communication Format		
Bit 7 CA	Bit 6 CHR	Bit 5 PE	Bit 3 STOP	Mode	Data Length	Parity Bit	Stop Bit Length
0	0	0	0	Asynchronous	8 bits	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
	1	0	0		7 bits	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
1	x	x	x	Clock synchronous	8 bits	Not set	None

[Legend]

x: Don't care

22.4.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCIF are independent, so full duplex communication is possible. The transmitter and receiver are 16-byte FIFO buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 22.2 shows the general format of asynchronous serial communication.

In asynchronous serial communication, the communication line is normally held in the mark (high) state. The SCIF monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCIF synchronizes at the falling edge of the start bit. The SCIF samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

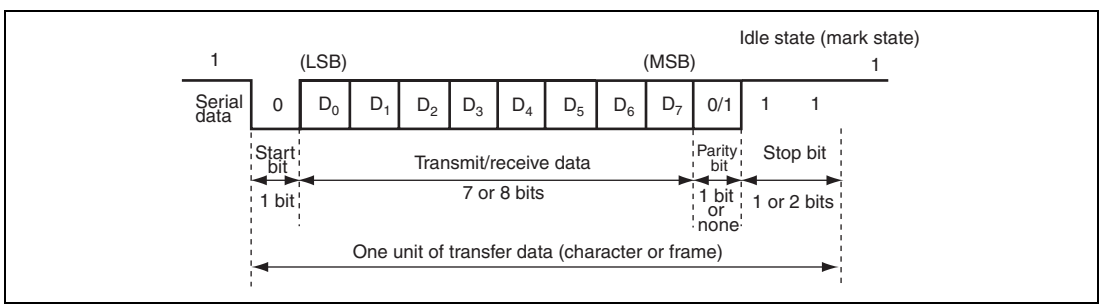


Figure 22.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

(1) Transmit/Receive Formats

Table 22.6 lists the eight communication formats that can be selected in asynchronous mode. The format is selected by settings in SCSMR.

Table 22.6 Serial Communication Formats (Asynchronous Mode)

SCSMR Bits			Serial Transmit/Receive Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	START	8-bit data								STOP		
0	0	1	START	8-bit data								STOP	STOP	
0	1	0	START	8-bit data								P	STOP	
0	1	1	START	8-bit data								P	STOP	STOP
1	0	0	START	7-bit data							STOP			
1	0	1	START	7-bit data							STOP	STOP		
1	1	0	START	7-bit data							P	STOP		
1	1	1	START	7-bit data							P	STOP	STOP	

[Legend]
START: Start bit
STOP: Stop bit
P: Parity bit

(2) Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCIF transmit/receive clock.

(3) SCIF Initialization (Asynchronous Mode)

Before transmitting or receiving, clear the TE and RE bits to 0 in SCSCR, then initialize the SCIF as follows. When changing the operation mode or the communication format, always clear the TE and RE bits to 0 before following the procedure given below.

1 Clearing TE to 0 initializes SCTSR. Clearing TE and RE to 0, however, does not initialize SCFSR, SCFTDR, or SCFRDR, which retain their contents.
2. Clear TE to 0 after all transmit data has been transmitted and the TEND flag in the SCFSR is set. The TE bit can be cleared to 0 during transmission, but the transmit data goes to the Mark state after the bit is cleared to 0. Set the TFRST bit in SCFCR to 1 and reset SCFTDR before TE is set again to start transmission.

Figure 22.3 shows a sample flowchart for initializing the SCIF.

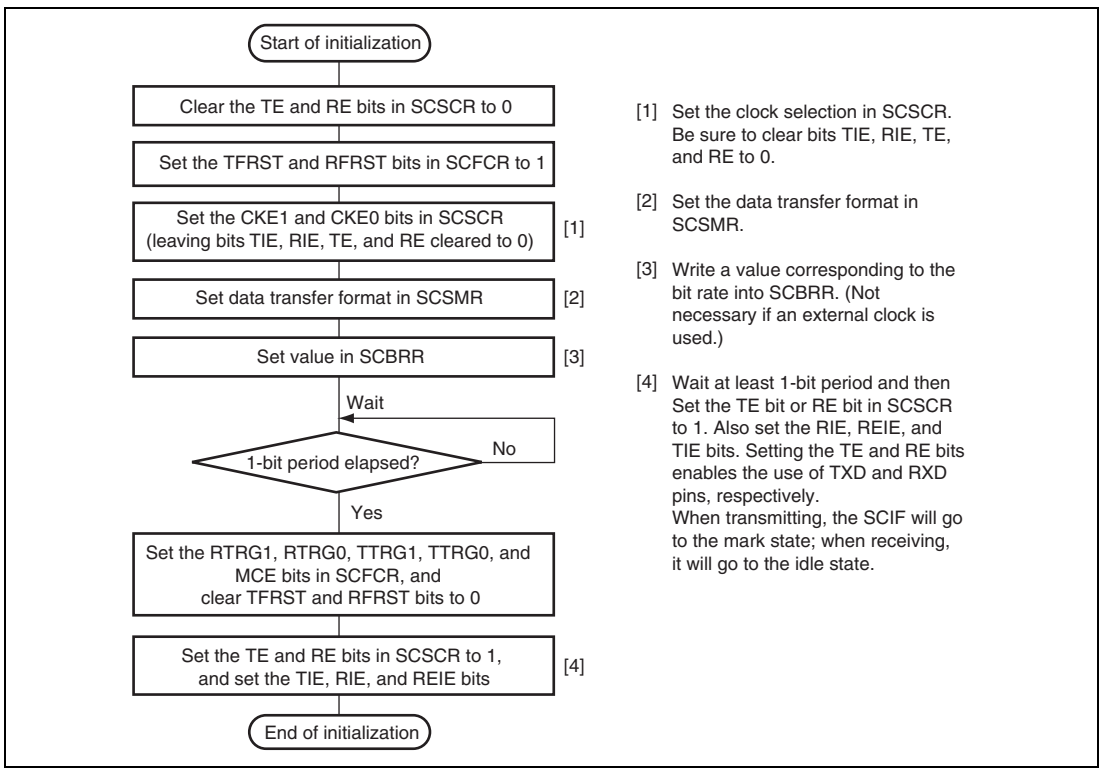


Figure 22.3 Sample Flowchart for SCIF Initialization

(4) Transmitting Serial Data (Asynchronous Mode)

Figure 22.4 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

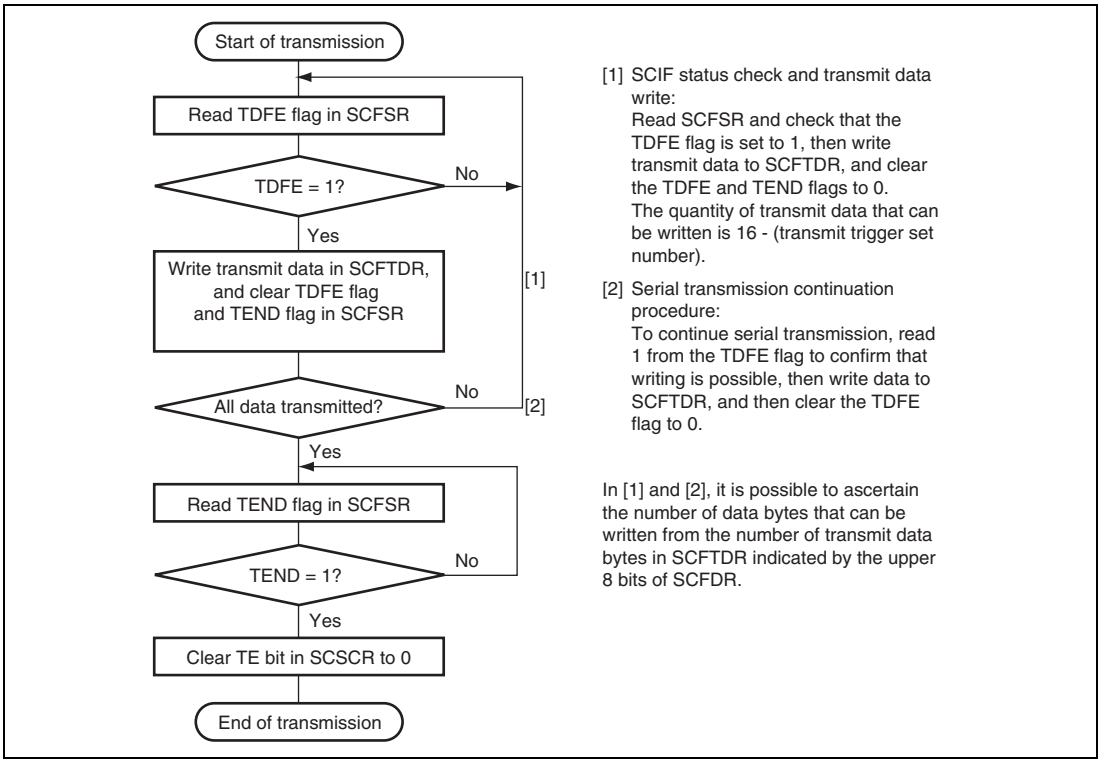


Figure 22.4 Sample Flowchart for Transmitting Serial Data

In serial transmission, the SCIF operates as described below.

1. When data is written into SCFTDR, the SCIF transfers the data from SCFTDR to SCTSR and starts transmission. Confirm that the TDFE flag in SCFSR is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is at least (16 – transmit trigger number setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls below the transmit trigger number set in SCFCR, the TDFE flag is set. If the TIE bit in SCSR is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   The serial transmit data is sent from the TXD pin in the following order.

   A. Start bit: One-bit 0 is output.
   B. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
   C. Parity bit: One parity bit (even or odd parity) is output. (A format in which a parity bit is not output can also be selected.)
   D. Stop bit(s): One or two 1 bits (stop bits) are output.
   E. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.

   When there is no transmit data after the stop bit is sent, the TEND flag in SCFSR is set to 1 and the SCIF enters a mark state, in which 1 is output from the TXD pin.

Figure 22.5 shows an example of the operation for transmission.

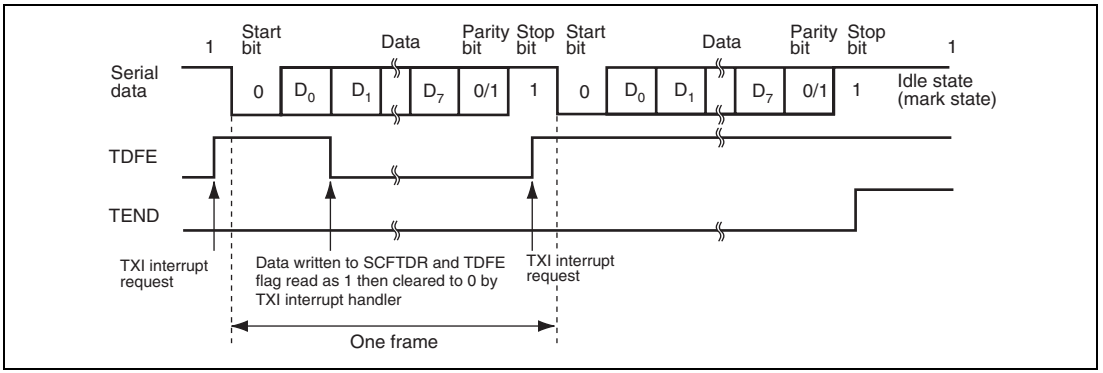


Figure 22.5 Example of Transmit Operation (8-Bit Data, Parity, 1 Stop Bit)

4. When modem control is enabled, transmission can be stopped and restarted in accordance with the $\overline{\text{CTS}}$ input value. When $\overline{\text{CTS}}$ is set to 1, if transmission is in progress, the line goes to the mark state after transmission of one frame. When $\overline{\text{CTS}}$ is set to 0, the next transmit data is output starting from the start bit.

Figure 22.6 shows an example of the operation when modem control is used.

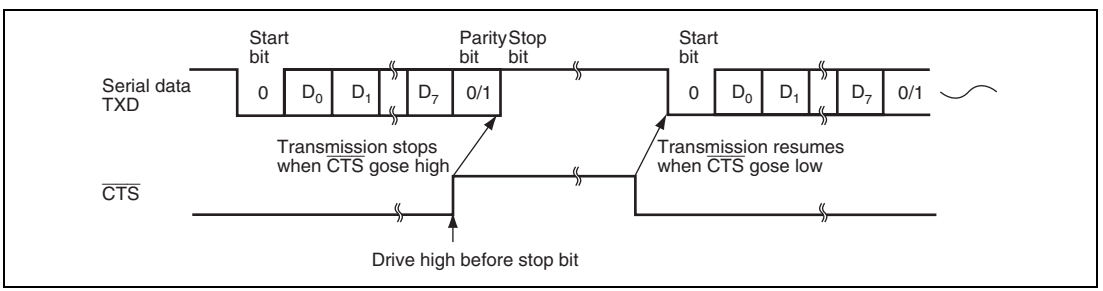


Figure 22.6 Example of Operation Using Modem Control ($\overline{\text{CTS}}$)

(5) Receiving Serial Data (Asynchronous Mode)

Figures 22.7 and 22.8 show sample flowcharts for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

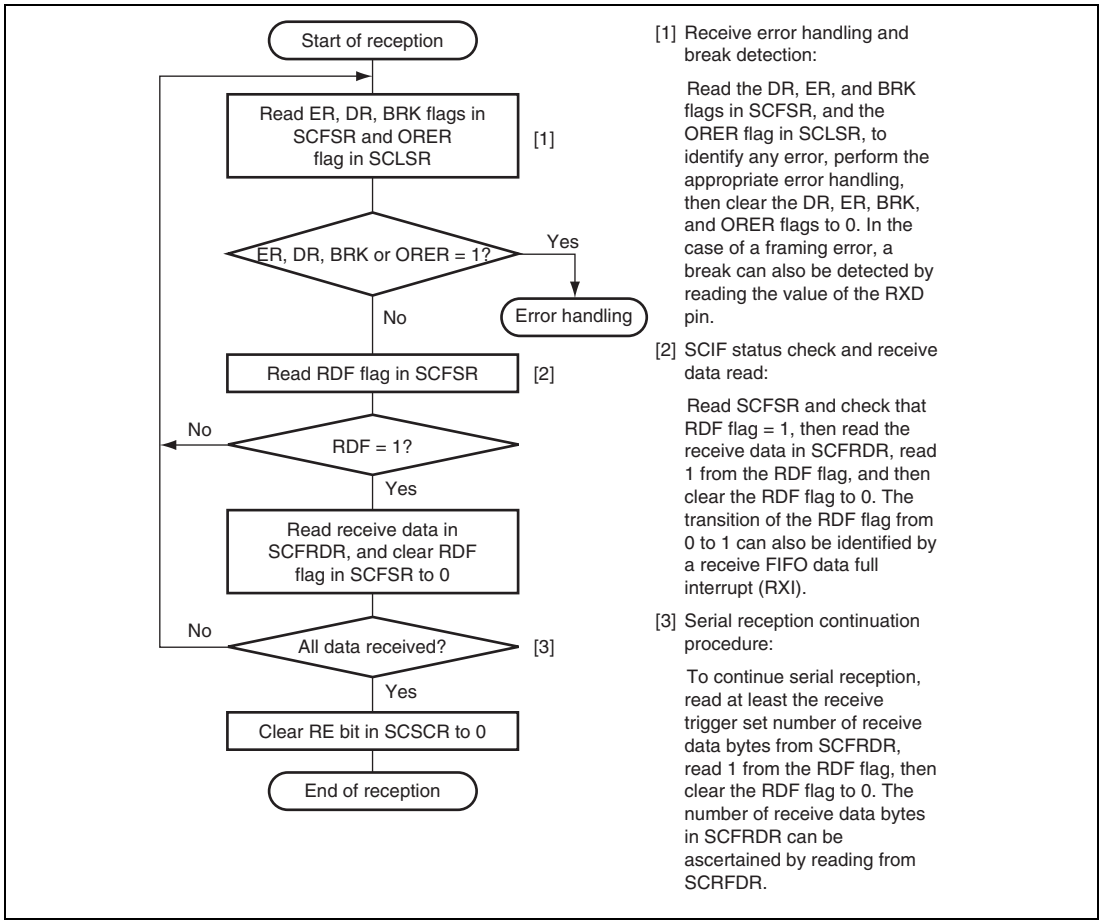
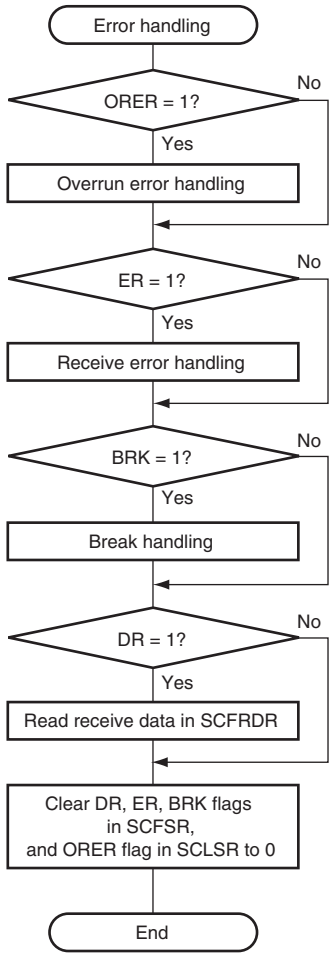


Figure 22.7 Sample Flowchart for Receiving Serial Data



- Whether a framing error or parity error has occurred in the receive data that is to be read from SCFRDR can be ascertained from the FER and PER bits in SCFSR.
- When a break signal is received, receive data is not transferred to SCFRDR while the BRK flag is set. However, note that the last data in SCFRDR is H'00, and the break data in which a framing error occurred is stored.

Figure 22.8 Sample Flowchart for Receiving Serial Data (cont)

In serial reception, the SCIF operates as described below.

1. The SCIF monitors the transmission line, and if a 0 start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in SCRSR in LSB-to-MSB order.
3. The parity bit and stop bit are received.

   After receiving these bits, the SCIF carries out the following checks.

   A. Stop bit check: The SCIF checks whether the stop bit is 1. If there are two stop bits, only the first is checked.

   B. The SCIF checks whether receive data can be transferred from the receive shift register (SCRSR) to SCFRDR.

   C. Overrun check: The SCIF checks that the ORER flag is 0, indicating that the overrun error has not occurred.

   D. Break check: The SCIF checks that the BRK flag is 0, indicating that the break state is not set.*

   If all the above checks are passed, the receive data is stored in SCFRDR.

   Note: * Even when a parity error or a framing error occurs, reception will not be suspended.

4. If the RIE bit in SCSCR is set to 1 when the RDF or DR flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated. If the RIE bit or the REIE bit in SCSCR is set to 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated. If the RIE bit or the REIE bit in SCSCR is set to 1 when the BRK or ORER flag changes to 1, a break reception interrupt (BRI) request is generated.

Figure 22.9 shows an example of the operation for reception.

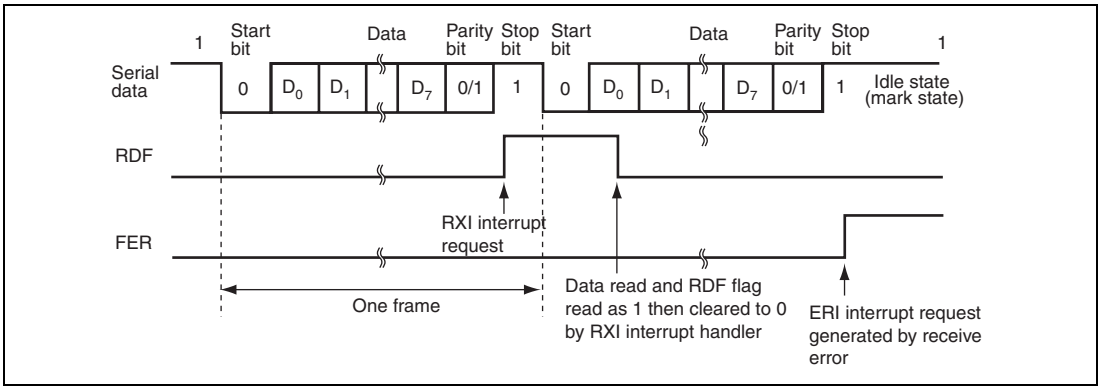


Figure 22.9 Example of SCIF Receive Operation (8-Bit Data, Parity, 1 Stop Bit)

5. When modem control is enabled, the $\overline{\text{RTS}}$ signal is output when SCFRDR is empty. When $\overline{\text{RTS}}$ is 0, reception is possible. When $\overline{\text{RTS}}$ is 1, this indicates that SCFRDR exceeds the number set for the $\overline{\text{RTS}}$ output active trigger, which is set by bits 10 to 8 in SCFCR. $\overline{\text{RTS}}$ also becomes 1 while the RE bit in SCSCR is 0.

Figure 22.10 shows an example of the operation when modem control is used.

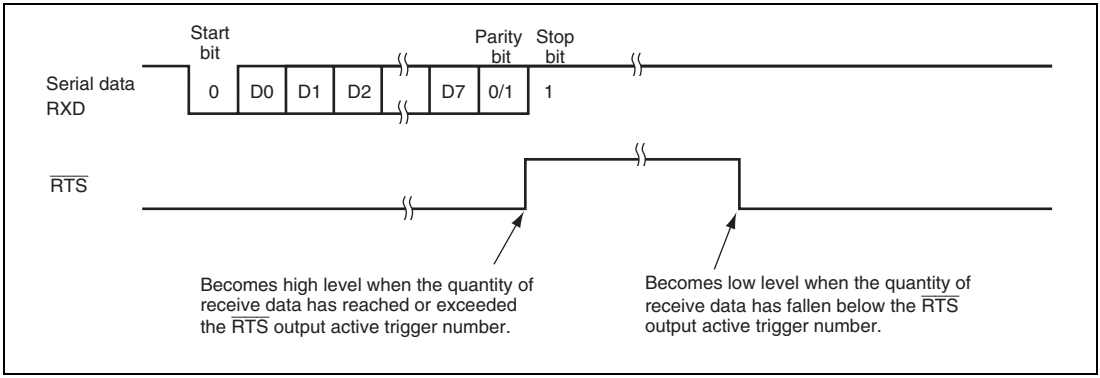


Figure 22.10 Example of Operation Using Modem Control ($\overline{\text{RTS}}$)

22.4.3 Operation in Clock Synchronous Mode

In clock synchronous mode, the SCIF transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCIF transmitter and receiver are independent, so full-duplex communication is possible while sharing the same clock. The transmitter and receiver are also 16-byte FIFO buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 22.11 shows the general format in clock synchronous serial communication.

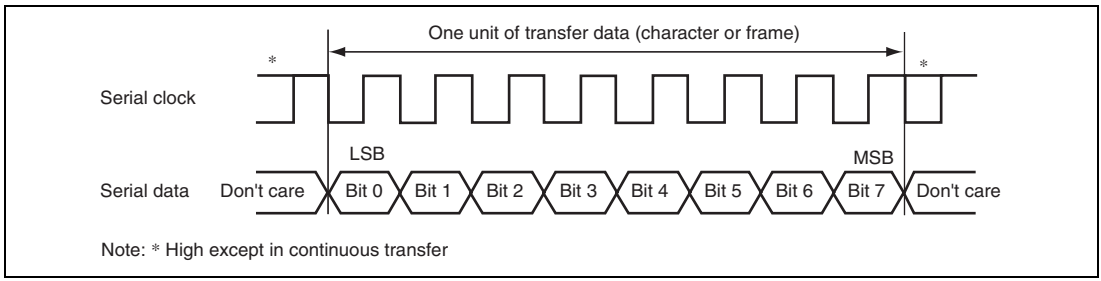


Figure 22.11 Data Format in Clock Synchronous Communication

In clock synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clock.

In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the last data, the communication line remains in the state of the last data.

In clock synchronous mode, the SCIF receives data by synchronizing with the rising edge of the serial clock.

(1) Transmit/Receive Formats

The data length is fixed at eight bits. No parity bit can be added.

(2) Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCIF transmit/receive clock.

(3) SCIF Initialization (Clock Synchronous Mode)

Before transmitting, receiving, or changing the mode or communication format, the software must clear the TE and RE bits to 0 in SCSCR, then initialize the SCIF. Clearing TE to 0 initializes SCTSR. Clearing RE to 0, however, does not initialize the RDF, PER, FER, and ORER flags and SCRDR, which retain their contents.

Figure 22.12 shows a sample flowchart for initializing the SCIF.

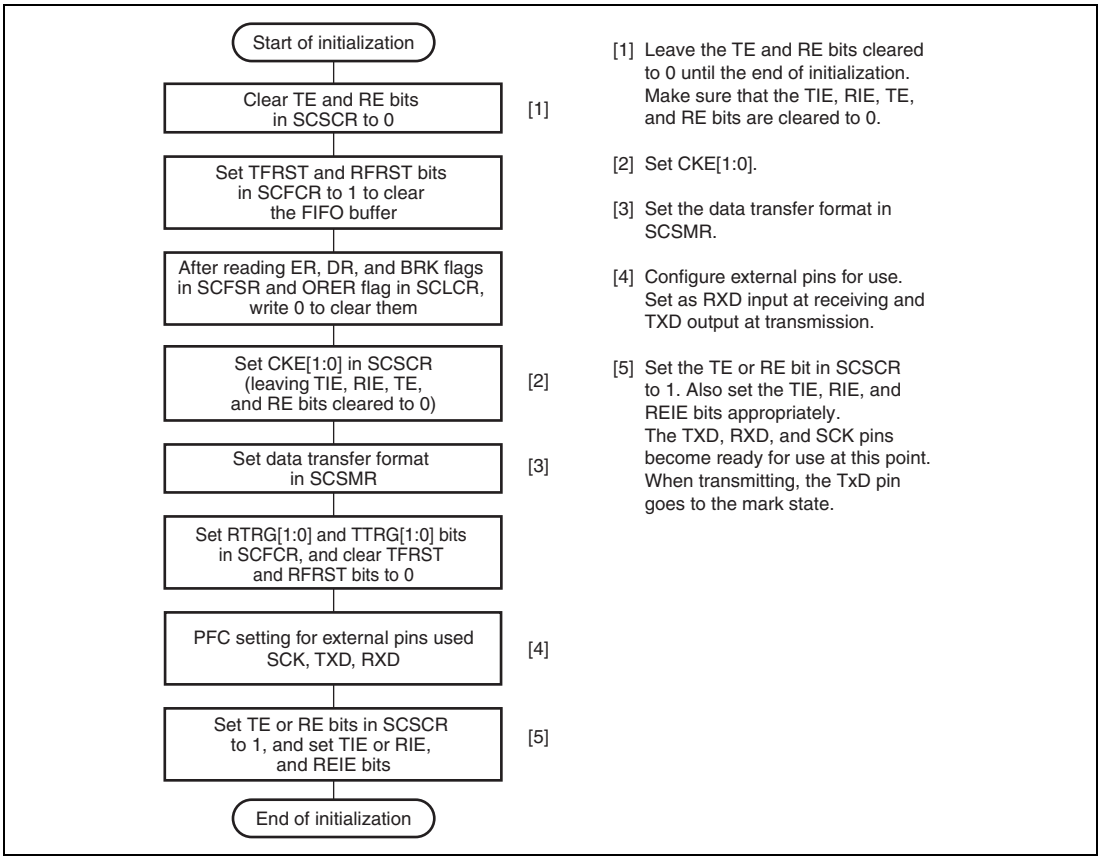


Figure 22.12 Sample Flowchart for SCIF Initialization

(4) Transmitting Serial Data (Clock Synchronous Mode)

Figure 22.13 shows a sample flowchart for transmitting serial data.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

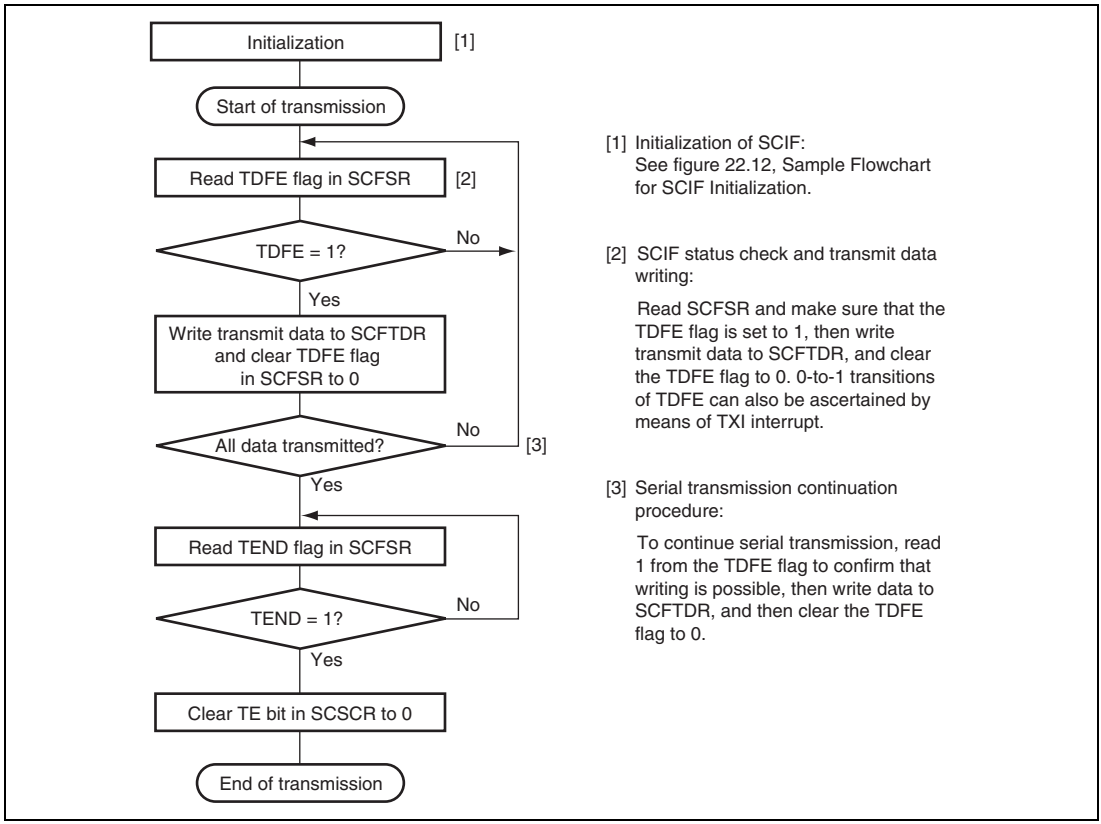


Figure 22.13 Sample Flowchart for Transmitting Serial Data

In serial transmission, the SCIF operates as described below.

1. When data is written into SCFTDR, the SCIF transfers the data in SCFTDR to SCTSR and starts transmission. Confirm that the TDFE flag in SCFSR is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is at least (16 – transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls below the transmit trigger number set in SCFCR, the TDFE flag is set. If the TIE bit in SCSR is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   If clock output mode is selected, the SCIF outputs eight synchronous clock pulses. If an external clock source is selected, the SCIF outputs data in synchronization with the input clock. Data is output from the TXD pin in order from the LSB (bit 0) to the MSB (bit 7).
3. The SCIF checks the SCFTDR transmit data at the timing for sending the MSB (bit 7). If data is present, the data is transferred from SCFTDR to SCTSR, and then serial transmission of the next frame is started. If there is no transmit data, the TEND flag in SCFSR is set to 1 after the MSB (bit 7) is sent and the TXD pin holds its state.

Figure 22.14 shows an example of SCIF transmit operation.

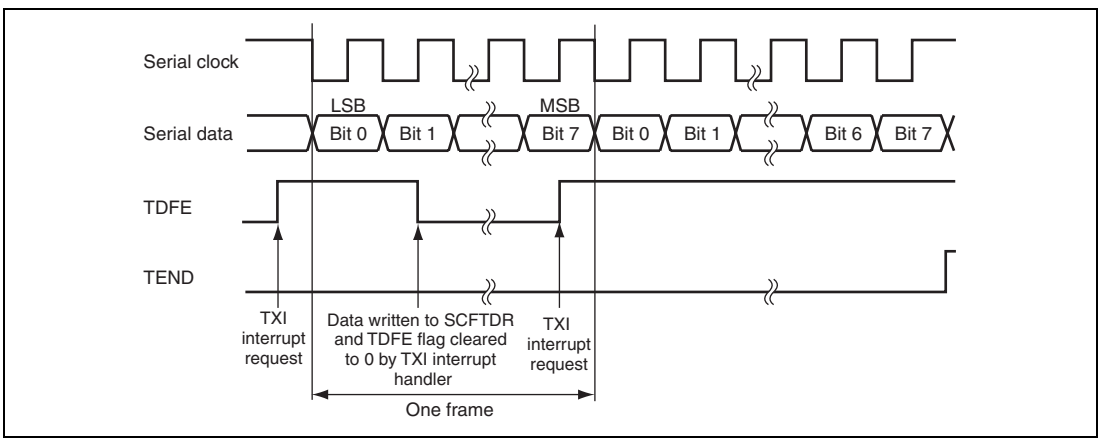


Figure 22.14 Example of SCIF Transmit Operation

(5) Receiving Serial Data (Clock Synchronous Mode)

Figures 22.15 and 22.16 show sample flowcharts for receiving serial data. When switching from asynchronous mode to clock synchronous mode without SCIF initialization, make sure that the ORER bit in SCLSR and the PER and FER bits in SCFCR are cleared to 0.

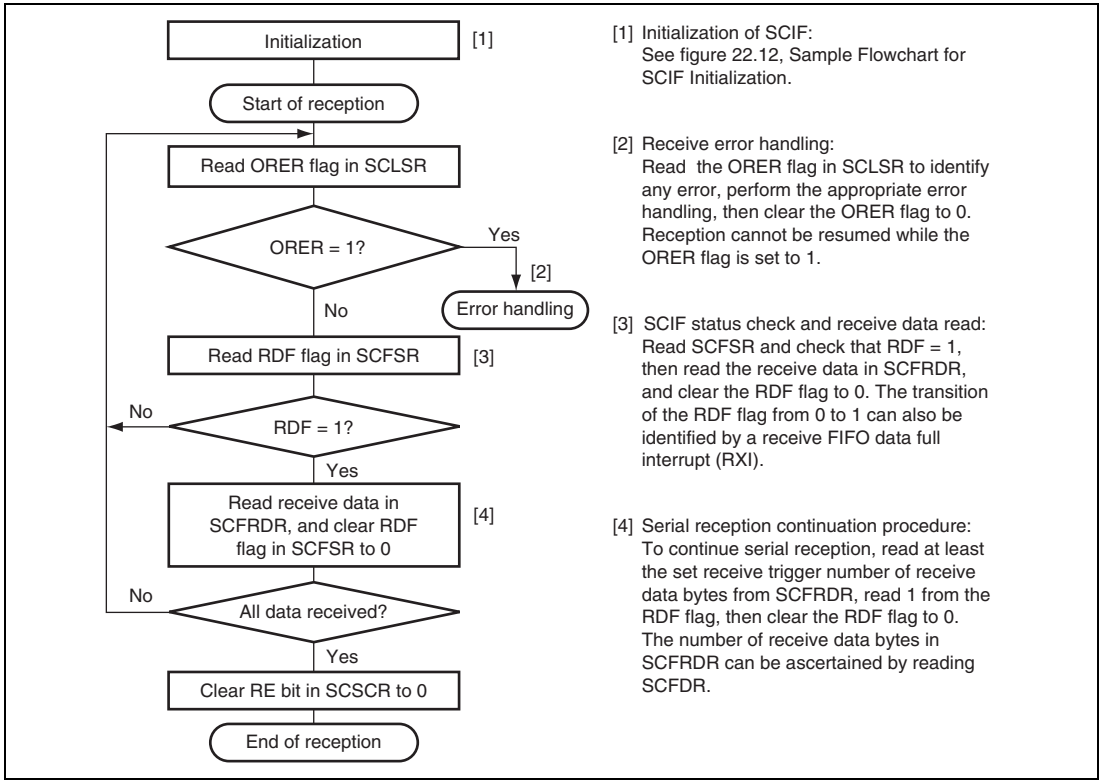


Figure 22.15 Sample Flowchart for Receiving Serial Data (1)

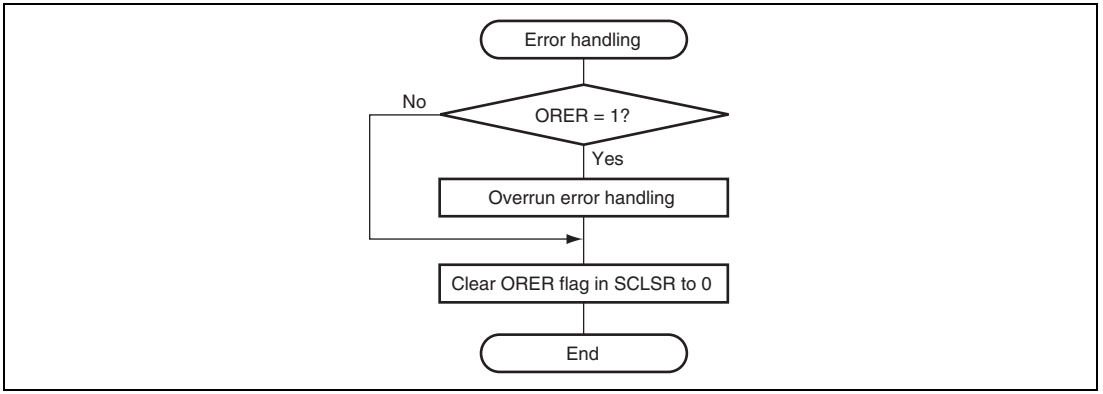


Figure 22.16 Sample Flowchart for Receiving Serial Data (2)

In serial reception, the SCIF operates as described below.

1. The SCIF starts the reception in synchronization with the serial clock output.
2. Receive data is shifted into SCRSR in order from the LSB to the MSB. After receiving the data, the SCIF checks the receive data can be loaded from SCRSR into SCFRDR or not. If this check is passed, the SCIF stores the received data in SCFRDR. If the check is not passed (overrun error is detected), further reception is prevented.
3. After setting RDF to 1, if the receive FIFO data full interrupt enable bit (RIE) is set to 1 in SCSCR, the SCIF requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the receive-data-full interrupt enable bit (RIE) or the receive error interrupt enable bit (REIE) in SCSCR is also set to 1, the SCIF requests a break interrupt (BRI).

Figure 22.17 shows an example of SCIF receive operation.

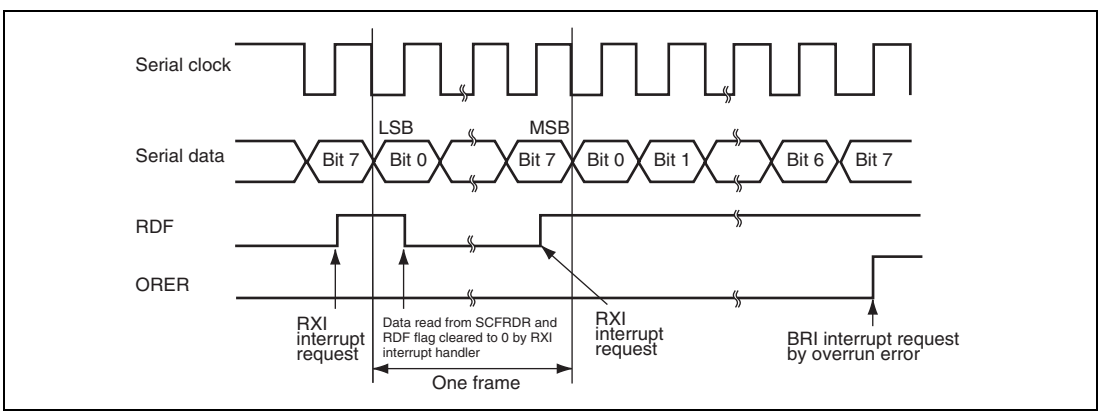


Figure 22.17 Example of SCIF Receive Operation

- Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode)

  Figure 22.18 shows a sample flowchart for transmitting and receiving serial data simultaneously.

  Use the following procedure for the simultaneous transmission/reception of serial data, after enabling the SCIF for transmission/reception.

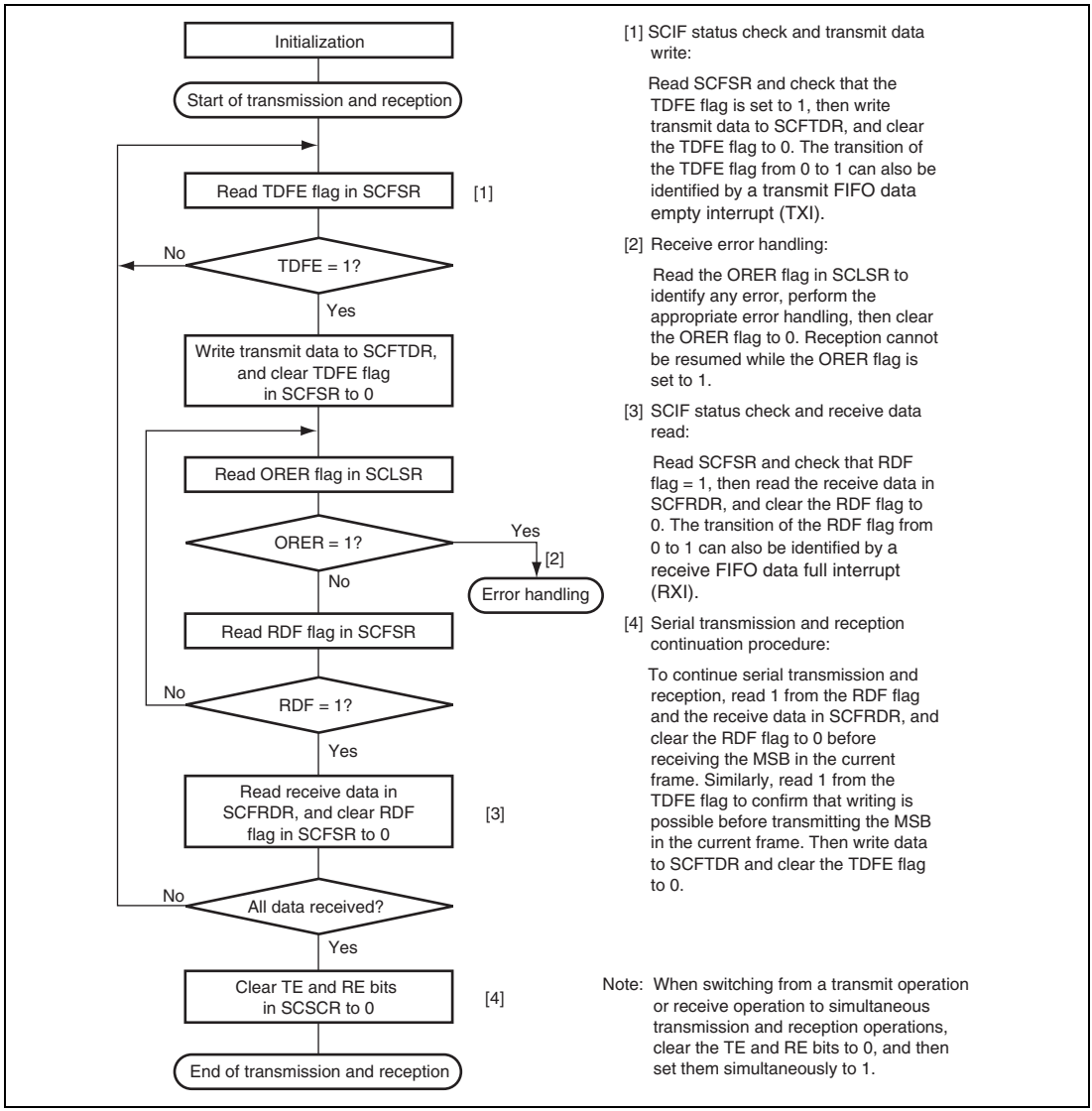


Figure 22.18 Sample Flowchart for Transmitting/Receiving Serial Data

22.5 SCIF Interrupt Sources and DMAC

The SCIF has four types of interrupt sources for each channel: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive-FIFO-data-full (RXI), and break (BRI). However, only a single INTEVT code is assigned per channel, so interrupt sources must be identified by software. The interrupt sources are enabled or disabled separately for each channel by means of the TIE, RIE, and REIE bits in SCSCR.

When the TXI request is enabled by the TIE bit and the TDFE flag in SCFSR is set to 1, a TXI interrupt request and transmit-FIFO-data-empty DMA transfer request are generated. When the TXI request is disabled by the TIE bit and the TDFE flag in SCFSR is set to 1, only a transmit-FIFO-data-empty DMA transfer request is generated. This transmit-FIFO-data-empty DMA transfer request can activate the DMAC to perform data transfer.

When the RXI request is enabled by the RIE bit and the RDF flag or the DR flag in SCFSR is set to 1, an RXI interrupt request and receive-FIFO-data-full DMA transfer request are generated. When the RXI request is disabled by the RIE bit and the RDF flag or the DR flag in SCFSR is set to 1, only a receive-FIFO-data-full DMA transfer request is generated. This receive-FIFO-data-full DMA transfer request can activate the DMAC to perform data transfer. Note that the RXI interrupt request or receive-FIFO-data-full DMA transfer request resulting from the DR flag is only generated in asynchronous mode.

When the BRK flag in SCFSR or the ORER flag in SCLSR is set to 1, a BRI interrupt request is generated.

To perform transmission/reception using the DMAC, configure and enable the DMAC first and configure the SCIF next. The SCIF should be configured such that the RXI and TXI interrupt requests are not sent to the interrupt controller. If not configured as such, the interrupt requests sent to the interrupt controller will be cleared by the DMAC regardless of the interrupt handling program.

Clearing the RIE bit to 0 and setting the REIE bit to 1 in SCSCR generates only an ERI and BRI interrupt request without generating an RXI interrupt request.

22.6 Usage Notes

Note the following when using the SCIF.

22.6.1 SCFTDR Writing and TDFE Flag

The TDFE flag in SCFSR is set when the number of transmit data bytes written in SCFTDR has fallen below the transmit trigger number set by bits TTRG[1:0] in SCFCR. After the TDFE flag is set, transmit data up to the number of empty bytes in SCFTDR can be written, allowing efficient continuous transmission.

However, if the number of data bytes written in SCFTDR is equal to or less than the transmit trigger number, the TDFE flag will be set to 1 again after being read as 1 and cleared to 0. TDFE flag clearing should therefore be carried out when SCFTDR contains more than the transmit trigger number of transmit data bytes.

The number of transmit data bytes in SCFTDR can be ascertained through SCFDR.

22.6.2 SCFRDR Reading and RDF Flag

The RDF flag in SCFSR is set when the number of receive data bytes in SCFRDR has become equal to or greater than the receive trigger number set by bits RTRG[1:0] in SCFCR. After RDF flag is set, receive data equivalent to the trigger number can be read from SCFRDR, allowing efficient continuous reception.

However, if the number of data bytes in SCFRDR exceeds the trigger number, the RDF flag will be set to 1 again if it is cleared to 0. The RDF flag should therefore be cleared to 0 after being read as 1 after reading the number of the received data in SCFRDR which is less than the trigger number.

The number of receive data bytes in SCFRDR can be ascertained through SCFDR.

22.6.3 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)

The SCIF operates on a base clock with a frequency 16 times the bit rate. In reception, the SCIF synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. This is shown in figure 22.19.

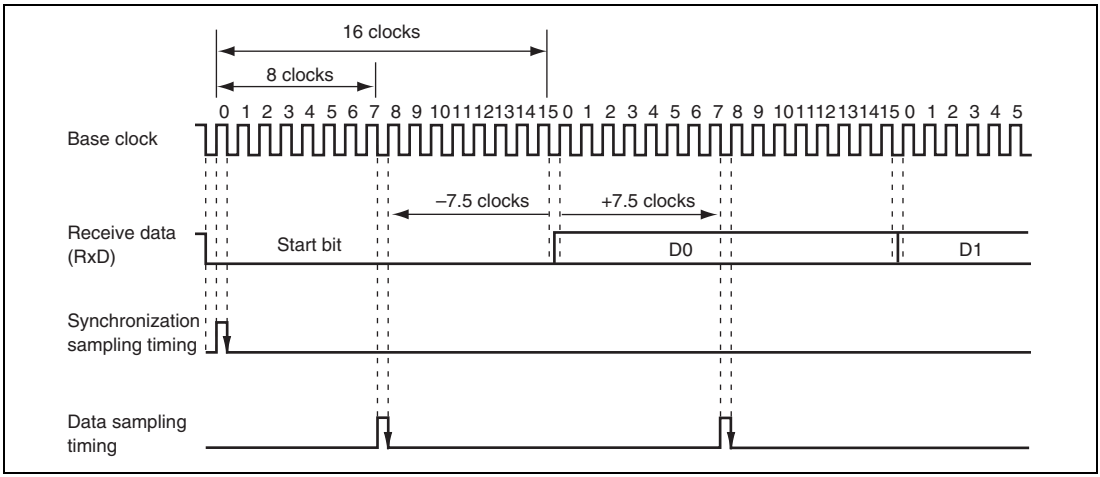


Figure 22.19 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed by equation 1.

Equation 1:

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

Where: M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16 or 8)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation 1, if F = 0, D = 0.5 and N = 16, the receive margin is 46.875%, as given by equation 2.

Equation 2:

When D = 0.5 and F = 0:

$M = (0.5 - 1/(2 \times 16)) \times 100\%$

$= 46.875\%$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

22.6.4 Using the DMAC

When performing transmission/reception using the DMAC, the SCIF should be configured such that the RXI and TXI interrupt requests are not sent to the interrupt controller. If not configured as such, however, the interrupt requests sent to the interrupt controller will be cleared by the DMAC regardless of the interrupt handling program.

22.6.5 Interrupts

Although the SCIF has four types of interrupt sources for each channel, only a single INTEVT code is assigned per channel. Therefore, interrupt sources must be identified by software.

Section 23 Serial Communication Interface with FIFO A (SCIFA)

This LSI has two channels (channel 4 and channel 5) of serial communication interface (SCIFA) that includes FIFO buffers. The SCIFA can perform asynchronous and synchronous serial communications. It has 64-stage FIFO registers for both transmission and reception, which allow efficient high-speed continuous communication.

23.1 Features

- Asynchronous or synchronous mode can be selected for serial communication mode.
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (internal) or SCK pin (external)
- Six types of interrupts (asynchronous mode):

  Transmit-data-stop, transmit-FIFO-data-empty, receive-FIFO-data-full, receive-error (framing error/parity error), break-receive, and receive-data-ready interrupts. A common interrupt vector is assigned to each interrupt source.
- Two types of interrupts (synchronous mode)
- The direct memory access controller (DMAC) can be activated to transfer data in the event of transmit-FIFO-data-empty and receive-FIFO-data-full.
- On-chip modem control functions (CTS and RTS)
- Transmit data stop function is available
- While the SCIFA is not used, it can be stopped by stopping the clock for it to reduce power consumption.
- The number of data bytes in the transmit and receive FIFO registers and the number of receive errors of the receive data in the receive FIFO register can be known.
- Full-duplex communication capability

  The transmitter and receiver are independent units, enabling transmission and reception to be performed simultaneously.

  The transmitter and receiver both have a 64-stage FIFO buffer structure, enabling fast and continuous serial data transmission and reception.

- Asynchronous mode:

  Serial data communications are performed by start-stop in character units. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are eight selectable serial data communication formats.

  — Data length: Seven or eight bits
  — Stop bit length: One or two bits
  — Parity: Even, odd, or none
  — LSB first
  — Receive error detection: Parity, framing, and overrun errors
  — Break detection: Break is detected when the receive data next the generated framing error is the space 0 level and has the framing error.

- Synchronous mode:

  Serial data communication is synchronized with a clock. Serial data communication can be carried out with other chips that have a synchronous communication function.

  — Data length: 8 bits
  — LSB-first transfer

Figure 23.1 shows the block diagram of SCIFA.

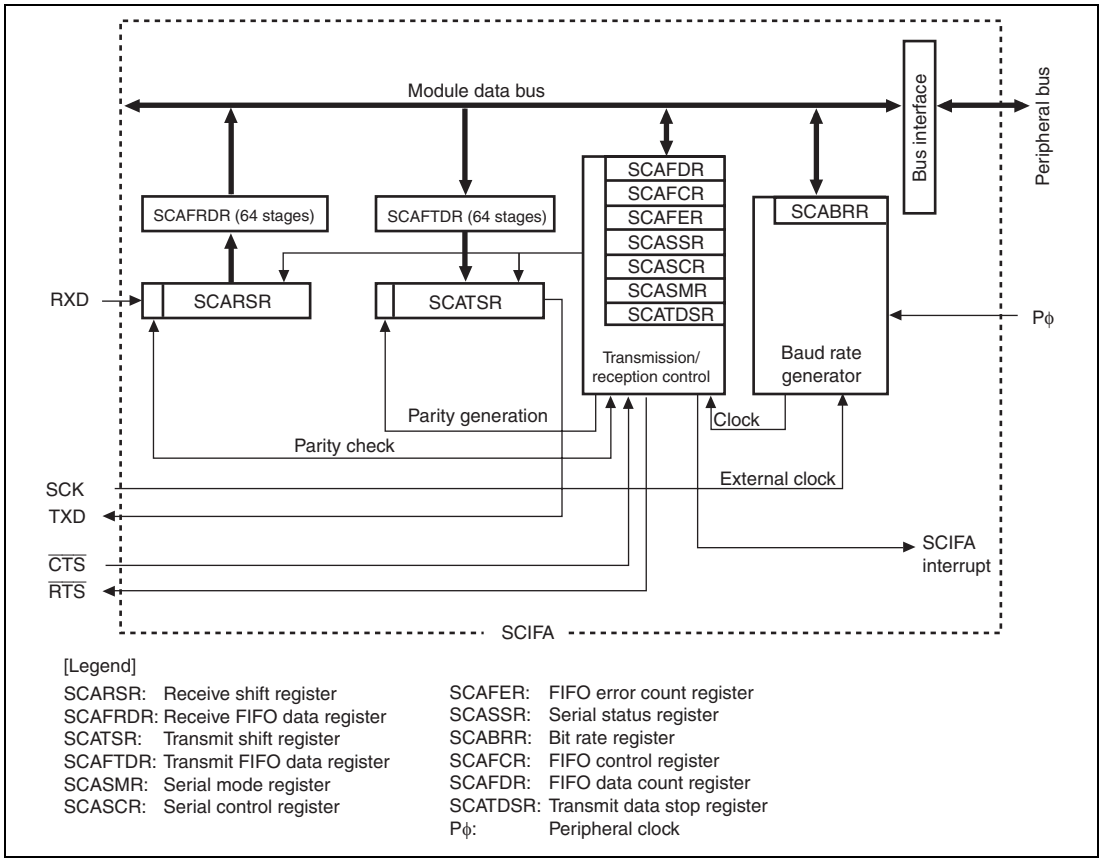


Figure 23.1 Block Diagram of SCIFA

23.2 Input/Output Pins

Table 23.1 shows the pin configuration of SCIFA.

Table 23.1 Pin configuration

Channel	Pin Name	I/O	Function
4	SCIF4_SCK	Input/output	Clock input/output
	SCIF4_RXD	Input	Receive data input
	SCIF4_TXD	Output	Transmit data output
	$\overline{\text{SCIF4\_CTS}}$	Input	Clear to send
	$\overline{\text{SCIF4\_RTS}}$	Output	Request to send
5	SCIF5_SCK	Input/output	Clock input/output
	SCIF5_RXD	Input	Receive data input
	SCIF5_TXD	Output	Transmit data output
	$\overline{\text{SCIF5\_CTS}}$	Input	Clear to send
	$\overline{\text{SCIF5\_RTS}}$	Output	Request to send

Note: In the following description, channel numbers in pin names are omitted and SCK, RXD, TXD, $\overline{\text{CTS}}$, and $\overline{\text{RTS}}$ are used as the generic abbreviations.

23.3 Register Descriptions

The register configuration of the SCIFA is shown in table 23.2, and the register states in each processing mode are shown in table 23.3. Note that the channel numbers are omitted from the register names in descriptions subsequent to these tables.

Table 23.2 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
Serial mode register A4	SCASMR4	R/W	H'FFE4 0000	16
Bit rate register A4	SCABRR4	R/W	H'FFE4 0004	8
Serial control register A4	SCASCR4	R/W	H'FFE4 0008	16
Transmit data stop register A4	SCATDSR4	R/W	H'FFE4 000C	8
FIFO error count register A4	SCAFER4	R	H'FFE4 0010	16
Serial status register A4	SCASSR4	R/W*	H'FFE4 0014	16
FIFO control register A4	SCAFCR4	R/W	H'FFE4 0018	16
FIFO data count register A4	SCAFDR4	R	H'FFE4 001C	16
Transmit FIFO data register A4	SCAFTDR4	W	H'FFE4 0020	8
Receive FIFO data register A4	SCAFRDR4	R	H'FFE4 0024	8
Serial mode register A5	SCASMR5	R/W	H'FFE5 0000	16
Bit rate register A5	SCABRR5	R/W	H'FFE5 0004	8
Serial control register A5	SCASCR5	R/W	H'FFE5 0008	16
Transmit data stop register A5	SCATDSR5	R/W	H'FFE5 000C	8
FIFO error count register A5	SCAFER5	R	H'FFE5 0010	16
Serial status register A5	SCASSR5	R/W*	H'FFE5 0014	16
FIFO control register A5	SCAFCR5	R/W	H'FFE5 0018	16
FIFO data count register A5	SCAFDR5	R	H'FFE5 001C	16
Transmit FIFO data register A5	SCAFTDR5	W	H'FFE5 0020	8
Receive FIFO data register A5	SCAFRDR5	R	H'FFE5 0024	8

Note: * To bits 9 to 7, 5, 4, 1, and 0, only 0 can be written to clear the flag.

Table 23.3 Register States in Each Operating Mode

Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
SCASMR4	Initialized	Retained	Retained	Retained
SCABRR4	Initialized	Retained	Retained	Retained
SCASCR4	Initialized	Retained	Retained	Retained
SCATDSR4	Initialized	Retained	Retained	Retained
SCAFER4	Initialized	Retained	Retained	Retained
SCASSR4	Initialized	Retained	Retained	Retained
SCAFCR4	Initialized	Retained	Retained	Retained
SCAFDR4	Initialized	Retained	Retained	Retained
SCAFTDR4	Initialized	Retained	Retained	Retained
SCAFRDR4	Initialized	Retained	Retained	Retained
SCASMR5	Initialized	Retained	Retained	Retained
SCABRR5	Initialized	Retained	Retained	Retained
SCASCR5	Initialized	Retained	Retained	Retained
SCATDSR5	Initialized	Retained	Retained	Retained
SCAFER5	Initialized	Retained	Retained	Retained
SCASSR5	Initialized	Retained	Retained	Retained
SCAFCR5	Initialized	Retained	Retained	Retained
SCAFDR5	Initialized	Retained	Retained	Retained
SCAFTDR5	Initialized	Retained	Retained	Retained
SCAFRDR5	Initialized	Retained	Retained	Retained

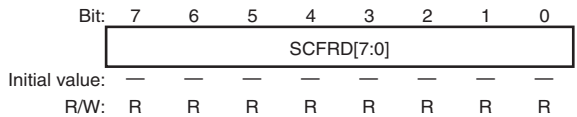
23.3.1 Receive Shift Register (SCARSR)

SCARSR receives serial data. Data input at the RXD pin is loaded into the SCARSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to SCAFRDR, which is a receive FIFO data register. The CPU cannot read from or write to the SCARSR directly.

23.3.2 Receive FIFO Data Register (SCAFRDR)

The 64-byte receive FIFO data register (SCAFRDR) stores serial receive data. The SCIFA completes the reception of one byte of serial data by moving the received data from SCARSR into SCAFRDR for storage. Continuous receive can be performed until 64 bytes are stored, which makes SCAFRDR full.

The CPU can read but cannot write to SCAFRDR. When data is read without received data in SCAFRDR, the value is undefined. When the received data in this register becomes full, the subsequent serial data is lost.



Bit	Bit Name	Initial value	R/W	Description
7 to 0	SCFRD[7:0]	Undefined	R	FIFO Data Registers for Serial Receive Data

23.3.3 Transmit Shift Register (SCATSR)

SCATSR transmits serial data. The SCIFA loads transmit data from SCAFTDR into SCATSR, then transmits the data serially from the TXD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from SCAFTDR into SCATSR and starts transmitting again. The CPU cannot read or write SCATSR directly.

23.3.4 Transmit FIFO Data Register (SCAFTDR)

SCAFTDR is a 64-byte 8-bit-length FIFO register that stores data for serial transmission. When the SCIFA detects that SCATSR is empty, it moves transmit data written in SCAFTDR into SCATSR and starts serial transmission. Continuous serial transmission is performed until SCAFTDR becomes empty. As SCAFTDR is write-only, it cannot be read by the CPU.

When the transmit data in SCAFTDR is full (64 bytes), next data cannot be written. If attempted to write, the data is ignored.

Bit:	7	6	5	4	3	2	1	0
	SCFTD[7:0]							
Initial value:	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial value	R/W	Description
7 to 0	SCFTD[7:0]	Undefined	W	FIFO Data Registers for Serial Transmit Data

23.3.5 Serial Mode Register (SCASMR)

SCASMR is a 16-bit register that specifies the SCIFA serial communication format and selects the clock source for the baud rate generator and the sampling rate.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SRC[2:0]			CA	CHR	PE	OE	STOP	—	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	SRC[2:0]	All 0	R/W	Sampling Control Select sampling rate. 000: Sampling rate 1/16 001: Sampling rate 1/5 010: Sampling rate 1/7 011: Sampling rate 1/11 100: Sampling rate 1/13 101: Sampling rate 1/17 110: Sampling rate 1/19 111: Sampling rate 1/27
7	CA	0	R/W	Communication Mode Selects whether the SCIFA operates in asynchronous or synchronous mode. 0: Asynchronous mode 1: Synchronous mode
6	CHR	0	R/W	Character Length Selects seven-bit or eight-bit data. This bit is only valid in asynchronous mode. In synchronous mode, the data length is always eight bits, regardless of the CHR setting. 0: Eight-bit data 1: Seven-bit data* Note: * When seven-bit data is selected, the MSB (bit 7) in SCAFTDR is not transmitted.

Bit	Bit Name	Initial Value	R/W	Description
5	PE	0	R/W	Parity Enable Selects whether to add a parity bit to transmit data and to check the parity of receive data. This setting is only valid in asynchronous mode. In synchronous mode, parity bit addition and checking is not performed, regardless of the PE setting. 0: Parity bit not added or checked 1: Parity bit added and checked Note: * When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (OE) setting. Receive data parity is checked according to the even/odd (OE) mode setting.
4	OE	0	R/W	Parity Mode Selects even or odd parity when parity bits are added and checked. The OE setting is used only when the PE is set to 1 to enable parity addition and check. The OE setting is ignored when parity addition and check is disabled. 0: Even parity*[1] 1: Odd parity*[2] Notes: 1. If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined. 2. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.

Bit	Bit Name	Initial Value	R/W	Description
3	STOP	0	R/W	Stop Bit Length Selects one or two bits as the stop bit length. In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character. This setting is only valid in asynchronous mode. In synchronous mode, this setting is invalid since stop bits are not added. 0: One stop bit*1 1: Two stop bits*2 Notes: 1. In transmitting, a single bit of 1 is added at the end of each transmitted character. 2. In transmitting, two bits of 1 are added at the end of each transmitted character.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	CKS[1:0]	00	R/W	Clock Select These bits select the internal clock source of the on-chip baud rate generator. 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ is the peripheral clock.

Note: In synchronous mode, bits other than CKS[1:0] are fixed to 0.

23.3.6 Serial Control Register (SCASCR)

SCASCR is a 16-bit readable/writable register that operates the SCI transmitter/receiver, enables/disables interrupt requests, and selects the transmit/receive clock source.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDRQE	RDRQE	—	—	TSIE	ERIE	BRIE	DRIE	TIE	RIE	TE	RE	—	—	CKE[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	TDRQE	0	R/W	Transmit Data Transfer Request Enable Selects whether to issue the transmit-FIFO-data-empty interrupt request or DMA transfer request when TIE = 1 and transmit FIFO empty interrupt is generated at the transmission. 0: Interrupt request is issued to CPU 1: Transmit data transfer request is issued to DMAC
14	RDRQE	0	R/W	Receive Data Transfer Request Enable Selects whether to issue the receive-FIFO-data-full interrupt or DMA transfer request when RIE = 1 and receive FIFO data full interrupt is generated at the reception. 0: Interrupt request is issued to CPU 1: Receive data transfer request is issued to DMAC
13,12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	TSIE	0	R/W	Transmit Data Stop Interrupt Enable Enables or disables the generation of the transmit-data-stop interrupt requested when the TSE bit in SCAFCR is enabled and the TSF flag in SCASSR is set to 1. 0: The transmit-data-stop-interrupt disabled* 1: The transmit-data-stop-interrupt enabled Note: * The transmit data stop interrupt request is cleared by reading the TSF flag after it has been set to 1, then clearing the flag to 0, or clearing the TSIE bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
10	ERIE	0	R/W	Receive Error Interrupt Enable Enables or disables the generation of a receive-error (framing error/parity error) interrupt requested when the ER flag in SCASSR is set to 1. 0: The receive-error interrupt disabled* 1: The receive-error interrupt enabled Note: * The receive-error interrupt request is cleared by reading the ER flag after it has been set to 1, then clearing the flag to 0, or clearing the ERIE bit to 0.
9	BRIE	0	R/W	Break Interrupt Enable Enables or disables the generation of break-receive interrupt requested when the BRK flag in SCASSR is set to 1. 0: The break-receive interrupt disabled* 1: The break receive interrupt enabled Note: * The break-receive interrupt request is cleared by reading the BRK flag after it has been set to 1, then clearing the flag to 0, or clearing the BRIE bit to 0.
8	DRIE	0	R/W	Receive Data Ready Interrupt Enable Disables or enables the generation of receive-data-ready interrupt when the DR flag in SCASSR is set to 1. 0: The receive-data-ready interrupt disabled 1: The receive-data-ready interrupt enabled Note: * The receive-data-ready interrupt request is cleared by reading the DR flag after it has been set to 1, then clearing the flag to 0, or clearing the DRIE bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables the transmit-FIFO-data-empty interrupt requested when the TDFE flag of SCASSR is set to 1. 0: Transmit-FIFO-data-empty interrupt request disabled* 1: Transmit-FIFO-data-empty interrupt request enabled Note: * The transmit-FIFO-data empty interrupt request can be cleared by writing the greater number of transmit data bytes than the specified transmission trigger number to SCAFTDR and by clearing TDFE to 0 after reading 1 from TDFE, or can be cleared by clearing TIE to 0.
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive-FIFO-data-full interrupt requested when the RDF flag of SCASSR is set to1. 0: Receive-FIFO-data-full interrupt request disabled* 1: Receive-FIFO-data-full interrupt request enabled Note: * The receive-FIFO-data -full interrupt request can be cleared by reading the RDF flag after it has been set to 1, then clearing the flag to 0, or by clearing the RIE bit to 0.
5	TE	0	R/W	Transmit Enable Enables or disables the SCIFA serial transmitter. 0: Transmitter disabled 1: Transmitter enabled* Note: * SCASMR and SCAFCR should be set to select the transmit format and reset the transmit FIFO before setting the TE bit to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	RE	0	R/W	Receive Enable Enables or disables the SCIFA serial receiver. 0: Receiver disabled*1 1: Receiver enabled*2 Notes: 1. Clearing RE to 0 does not affect the receive flags (DR, ER, BRK, RDF, FER, PER, and ORER). These flags retain their previous values. 2. SCASMR and SCAFCR should be set to select the receive format and reset the receive FIFO before setting the RE bit to 1.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	CKE[1:0]	00	R/W	Clock Enable These bits select the SCIFA clock source and should be set before selecting the SCIFA operating mode by SCASMR. • Asynchronous mode 00: Internal clock; SCK pin used as input pin (input signal is ignored)*1 01: Setting prohibited 10: External clock, SCK pin used for clock input*3 11: Setting prohibited • Synchronous mode 00: Setting prohibited 01: Internal clock, SCK pin used for synchronous clock output*2 10: External clock, SCK pin used for clock input 11: Setting prohibited Notes: 1. When the data sampling is done using on-chip baud rate generator, CKE[1:0] should be set to 00. 2. The output clock frequency is the same as the bit rate. 3. Input a clock with a frequency 8 times the bit rate. The sampling rate is fixed at 1/16. When the external clock is not input, CKE[1:0] should be set to 00.

23.3.7 FIFO Error Count Register (SCAFER)

SCAFER is a 16-bit read-only register that indicates the number of receive data errors (framing error/parity error).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PER[5:0]						—	—	FER[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15,14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13 to 8	PER[5:0]	All 0	R	Parity Error Indicate the number of data bytes which contain parity errors in receive data stored in SCAFRDR in asynchronous mode. PER[5:0] indicate the number of data bytes with parity errors after the ER bit in SCASSR is set. If all 64-byte receive data in SCAFRDR have parity errors, bits PER[5:0] are all 0s.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	FER[5:0]	All 0	R	Framing Error Indicate the number of data bytes which contain framing errors in receive data stored in SCAFRDR in asynchronous mode. FER[5:0] indicate the number of data bytes with framing errors after the ER bit in SCASSR is set. If all 64-byte receive data in SCAFRDR have framing errors, bits FER[5:0] are all 0s.

23.3.8 Serial Status Register (SCASSR)

SCASSR is a 16-bit readable/writable register that indicates SCIFA states. The ORER, TSF, ER, TDFE, BRK, RDF, or DR flag cannot be set to 1. These flags can be cleared to 0 only if they have first been read (after being set to 1). The flags TEND, FER, and PER are read-only bits and cannot be modified.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ORER	TSF	ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Initial value:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R	R/(W)*	R/(W)*	R	R	R/(W)*	R/(W)*

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	ORER	0	R/(W)*	Overrun Error Flag Indicates that the overrun error occurred during reception. This bit is valid only in asynchronous mode. 0: Indicates during reception, or reception has been completed without any error*[1] [Clearing conditions] Power-on reset, manual reset Writing 0 after reading ORER = 1 1: Indicates that the overrun error is generated during reception*[2] [Setting condition] When receive FIFO is full and the next serial data reception is completed Notes: 1. When the RE bit in SCASCR is cleared to 0, the ORER flag is not affected and retains its previous state. 2. SCAFRDR holds the data received before the overrun error, and newly received data is lost. When ORER is set to 1, subsequent serial data reception cannot be carried out.

Bit	Bit Name	Initial Value	R/W	Description
8	TSF	0	R/(W)*	Transmit Data Stop Flag Indicates that the number of transmit data bytes matches the value set in SCATDSR. 0: Transmit data number does not match the value set in SCATDSR [Clearing conditions] • Power-on reset, manual reset • Writing 0 after reading TSF = 1 1: Transmit data number matches the value set in SCATDSR
7	ER	0	R/(W)*	Receive Error Indicates that a framing error or parity error occurred during reception in asynchronous mode.*[1] 0: Receive is normally completed without any framing or parity error [Clearing conditions] Power-on reset, manual reset ER is read as 1, then written to with 0. 1: A framing error or a parity error has occurred during receiving [Setting conditions] • The stop bit is 0 after checking whether or not the last stop bit of the received data is 1 at the end of one-data receive.*[2] • The total number of 1's in the received data and in the parity bit does not match the even/odd parity specification specified by the OE bit in SCASMR. Notes: 1. Indicates clearing the RE bit to 0 in SCASCR does not affect the ER bit, which retains its previous value. Even if a receive error occurs, the received data is transferred to SCAFRDR and the receive operation is continued. Whether or not the data read from SCRDR includes a receive error can be detected by the FER and PER bits in SCASSR. 2. n the stop mode, only the first stop bit is checked; the second stop bit is not checked.

Bit	Bit Name	Initial Value	R/W	Description
6	TEND	1	R	Transmit End Indicates that when the last bit of a serial character was transmitted, SCAFTDR did not contain valid data, so transmission has ended. 0: Transmission is in progress [Clearing condition] • Data is written to SCAFTDR. 1: End of transmission [Setting condition] • SCAFTDR contains no transmit data when the last bit of a one-byte serial character is transmitted.
5	TDFE	1	R/(W)*	Transmit FIFO Data Empty Indicates that data is transferred from SCAFTDR to SCATSR, the number of data bytes in SCAFTDR becomes less than the transmission trigger number specified by the TTRG[1:0] bits in SCAFCR, and writing the transmit data to SCAFTDR is enabled. 0: The number of transmit data bytes written to SCAFTDR is greater than the specified transmission trigger number [Clearing condition] • Data exceeding the specified transmission trigger number is written to SCAFTDR, software reads TDFE after it has been set to 1, then writes 0 to TDFE. 1: The number of transmit data bytes in SCAFTDR becomes less than the specified transmission trigger number [Setting conditions] • Power-on reset • The number of transmit data bytes in SCAFTDR becomes less than the specified transmission trigger number as a result of transmission* Note: * Since SCAFTDR is a 64-byte FIFO register, the maximum number of data bytes which can be written when TDFE is 1 is "64 minus the specified transmission trigger number". If attempted to write excess data, the data is ignored. The number of data bytes in SCAFTDR is indicated in SCAFDR.

Bit	Bit Name	Initial Value	R/W	Description
4	BRK	0	R/(W)*	Break Detection Indicates that a break signal is detected in received data in asynchronous mode. 0: No break signal is being received [Clearing conditions] • Power-on reset, manual reset • BRK is read as 1, then written to with 0 1: A break signal is received * [Setting conditions] Data including a framing error is received • A framing error with space 0 occurs in the subsequent received data Note: * When a break is detected, transfer of the received data (H'00) to SCAFRDR stops after detection. When the break ends and the receive signal becomes mark 1, the transfer of the received data resumes.
3	FER	0	R	Framing Error Indicates a framing error in the data read from SCAFRDR in asynchronous mode. 0: No framing error occurred in the data read from SCAFRDR [Clearing conditions] • Power-on reset, manual reset • No framing error is present in the data read from SCAFRDR 1: A framing error occurred in the data read from SCAFRDR [Setting condition] • A framing error is present in the data read from SCAFRDR

Bit	Bit Name	Initial Value	R/W	Description
2	PER	0	R	Parity Error Indicates a parity error in the data read from SCAFRDR in asynchronous mode. 0: No parity error occurred in the data read from SCAFRDR [Clearing conditions] • Power-on reset • No parity error is present in the data read from SCAFRDR 1: A parity error occurred in the data read from SCAFRDR [Setting condition] • A parity error is present in the data read from SCAFRDR

Bit	Bit Name	Initial Value	R/W	Description
1	RDF	0	R/(W)*	Receive FIFO Data Full Indicates that received data is transferred to SCAFRDR, the number of data bytes in SCAFRDR becomes more than the reception trigger number specified by the RTRG[1:0] bits in SCAFCR. 0: The number of transmit data bytes written to SCAFRDR is less than the specified reception trigger number [Clearing conditions] • Power-on reset • SCAFRDR is read until the number of receive data bytes in SCAFRDR becomes less than the specified reception trigger number, and RDF is read as 1, then written to with 0. 1: The number of receive data bytes in SCAFRDR is more than the specified reception trigger number [Setting condition] • Receive data of the number of bytes greater than the specified reception trigger number is being stored to SCAFRDR.* Note: * Since SCAFTDR is a 64-byte FIFO register, the maximum number of data bytes which can be read when RDF is 1 is the specified reception trigger number. If attempted to read after all data in SCAFRDR have been read, the data is undefined. The number of receive data bytes in SCAFRDR is indicated by the lower bits of SCAFTDR.

Bit	Bit Name	Initial Value	R/W	Description
0	DR	0	R/(W)*	Receive Data Ready Indicates that SCAFRDR stores data which is less than the specified reception trigger number, and that next data is not yet received after 15 etu has elapsed from the last stop bit in asynchronous mode. 0: Receive is in progress, or no received data remains in SCAFRDR after the receive ended normally. [Clearing conditions] (Initial value) • Power-on reset • All receive data in SCAFRDR is read, and DR is read as 1, then written to with 0. 1: Next receive data is not received [Setting condition] • SCAFRDR stores the data which is less than the specified reception trigger number, and that next data is not yet received after 15 etu has elapsed from the last stop bit.* Note: * This is equivalent to 1.5 frames with the 8-bit 1-stop-bit format. (etu: Element Time Unit)

Note: * The only value that can be written is 0 to clear the flag.

23.3.9 Bit Rate Register (SCABRR)

SCABRR is an eight-bit readable/writable register that, together with the baud rate generator clock source selected by the CKS[1:0] bits in SCASMR, determines the serial transmit/receive bit rate.

Bit:	7	6	5	4	3	2	1	0
	SCBRD[7:0]							
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SCBRD[7:0]	H'FF	R/W	Bit Rate Setting

The SCABRR setting is calculated as follows:

Asynchronous Mode:

1. When sampling rate is 1/16

$$N = \frac{P\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$

2. When sampling rate is 1/5

$$N = \frac{P\phi}{10 \times 2^{2n-1} \times B} \times 10^6 - 1$$

3. When sampling rate is 1/7

$$N = \frac{P\phi}{14 \times 2^{2n-1} \times B} \times 10^6 - 1$$

4. When sampling rate is 1/11

$$N = \frac{P\phi}{22 \times 2^{2n-1} \times B} \times 10^6 - 1$$

5. When sampling rate is 1/13

$$N = \frac{P\phi}{26 \times 2^{2n-1} \times B} \times 10^6 - 1$$

6. When sampling rate is 1/17

$$N = \frac{P\phi}{34 \times 2^{2n-1} \times B} \times 10^6 - 1$$

7. When sampling rate is 1/19

$$N = \frac{P\phi}{38 \times 2^{2n-1} \times B} \times 10^6 - 1$$

8. When sampling rate is 1/27

$$N = \frac{P\phi}{54 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Synchronous Mode:

$$N = \frac{P\phi}{4 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)
N: SCABRR setting for baud rate generator
Asynchronous mode: $0 \leq N \leq 255$
Synchronous mode: $1 \leq N \leq 255$
Pϕ: Peripheral module operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See the table below for the relation between n and the clock.)

Table 23.4 SCASMR Setting

		SCASMR Setting
n	**Clock Source**	**CKS[1:0]**
0	Pϕ	00
1	Pϕ/4	01
2	Pϕ/16	10
3	Pϕ/64	11

Find the bit rate error in asynchronous mode by the following formula:

1. When sampling rate is 1/16

$$\text{Error (\%)} = \left(\frac{P\phi \times 10^6}{(1+N) \times B \times 32 \times 2^{2n-1}} - 1 \right) \times 100$$

2. When sampling rate is 1/5

$$\text{Error (\%)} = \left(\frac{P\phi \times 10^6}{(1+N) \times B \times 10 \times 2^{2n-1}} - 1 \right) \times 100$$

3. When sampling rate is 1/7

$$\text{Error (\%)} = \left(\frac{P\phi \times 10^6}{(1+N) \times B \times 14 \times 2^{2n-1}} - 1 \right) \times 100$$

4. When sampling rate is 1/11

$$\text{Error (\%)} = \left(\frac{P\phi \times 10^6}{(1+N) \times B \times 22 \times 2^{2n-1}} - 1 \right) \times 100$$

5. When sampling rate is 1/13

$$\text{Error (\%)} = \left(\frac{P\phi \times 10^6}{(1+N) \times B \times 26 \times 2^{2n-1}} - 1 \right) \times 100$$

6. When sampling rate is 1/17

$$\text{Error (\%)} = \left(\frac{P\phi \times 10^6}{(1+N) \times B \times 34 \times 2^{2n-1}} - 1 \right) \times 100$$

7. When sampling rate is 1/19

$$\text{Error (\%)} = \left(\frac{P\phi \times 10^6}{(1+N) \times B \times 38 \times 2^{2n-1}} - 1 \right) \times 100$$

8. When sampling rate is 1/27

$$\text{Error (\%)} = \left(\frac{P\phi \times 10^6}{(1+N) \times B \times 54 \times 2^{2n-1}} - 1 \right) \times 100$$

23.3.10 FIFO Control Register (SCAFCR)

SCAFCR is a 16-bit readable/writable register that resets the number of data bytes in the transmit and receive FIFO registers, sets the number of trigger data, and contains an enable bit for the loop back test.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSE	TCRST	—	—	—	RSTRG[2:0]			RTRG[1:0]		TTRG[1:0]		MCE	TFRST	RFRST	LOOP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	TSE	0	R/W	Transmit Data Stop Enable Enables or disables transmit data stop function. This function is enabled only in asynchronous mode. Since this function is not supported in synchronous mode, clear this bit to 0 in synchronous mode. 0: Transmit data stop function disabled 1: Transmit data stop function enabled
14	TCRST	0	R/W	Transmit Count Reset Clears the transmit count to 0. This bit is available while the transmit data stop function is enabled. 0: Transmit count reset disabled* 1: Transmit count reset enabled (cleared to 0) Note: * The transmit count is reset (cleared to 0) by a power-on reset or manual reset.

Bit	Bit Name	Initial Value	R/W	Description
13 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	RSTRG [2:0]	000	R/W	Trigger of the $\overline{\text{RTS}}$ Output Active The $\overline{\text{RTS}}$ signal goes to high, when the number of receive data bytes stored in SCAFRDR has become equal to or more than the trigger number setting listed below. 000: 63 001: 1 010: 8 011: 16 100: 32 101: 48 110: 54 111: 60
7, 6	RTRG[1:0]	00	R/W	Receive FIFO Data Trigger Number Set the number of receive data bytes at which the receive data full (RDF) flag in SCASSR is set. The RDF flag is set when the number of receive data bytes stored in SCAFRDR has become equal to or more than the trigger number setting listed below. 00: 1 01: 16 10: 32 11: 48

Bit	Bit Name	Initial Value	R/W	Description
5, 4	TTRG[1:0]	00	R/W	Transmit FIFO Data Trigger Number Set the number of remaining transmit data bytes at which the transmit FIFO data register empty (TDFE) flag in SCASSR is set. The TDFE flag is set when the number of transmit data bytes in SCAFTDR has become equal to or less than the trigger number setting listed below. 00: 32 (32) 01: 16 (49) 10: 2 (62) 11: 0 (64) Note: * Values in brackets mean the number of empty bytes in SCAFTDR when the TDFE is set.
3	MCE	0	R/W	Modem Control Enable Enables the modem control signals $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$. 0: Disables the modem signal* 1: Enables the modem signal Note: * The $\overline{\text{CTS}}$ is fixed to active 0 regardless of the input value, and the $\overline{\text{RTS}}$ is also fixed to 0.
2	TFRST	0	R/W	Transmit FIFO Data Register Reset Cancels the transmit data in the transmit FIFO data register and resets the data to the empty state. 0: Disables reset operation* 1: Enables reset operation Note: * The reset is executed in a power-on reset or a manual reset.
1	RFRST	0	R/W	Receive FIFO Data Register Reset Cancels the receive data in the receive FIFO data register and resets the data to the empty state. 0: Disables reset operation* 1: Enables reset operation Note: * The reset is executed in a power-on reset or a manual reset.

Bit	Bit Name	Initial Value	R/W	Description
0	LOOP	0	R/W	Loop Back Test Internally connects the transmit output pin (TXD) and receive input pin (RXD) and enables the loop back test. 0: Disables the loop back test 1: Enables the loop back test

23.3.11 FIFO Data Count Register (SCAFDR)

SCAFDR is a 16-bit register which indicates the number of data bytes stored in SCAFRDR. The SCAFDR is always read from the CPU.

The bits 14 to 8 of this register indicate the number of transmit data bytes stored in SCAFTDR that have not yet been transmitted.

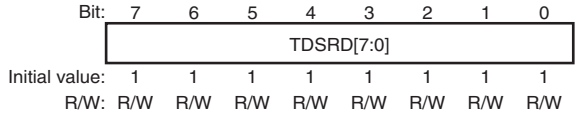
The bits 6 to 0 of this register indicate the number of receive data bytes stored in SCAFRDR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	T[6:0]							—	R[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 8	T[6:0]	H'00	R	These bits indicate the number of non-transmitted data stored in SCAFTDR. The H'00 means no transmit data, and the H'40 means that SCAFTDR is full of transmit data.
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 0	R[6:0]	H'00	R	These bits indicate the number of receive data bytes stored in SCAFRDR. The H'00 means no receive data, and the H'40 means that SCAFRDR is full of receive data.

23.3.12 Transmit Data Stop Register (SCATDSR)

SCATDSR is an 8-bit readable/writable register that sets the number of data bytes to be transmitted. This register is available when the TSE bit in SCAFCR is enabled. The transmit operation stops after all data set by this register have been transmitted. Settable values are H'00 (1 byte) to H'FF (256 bytes). The initial value of this register is H'FF.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	TDSRD[7:0]	H'FF	R/W	Transmit Data Stop Setting

23.4 Operation

23.4.1 Overview

For serial communication, the SCIFA has asynchronous mode in which characters are synchronized individually and synchronous mode in which synchronization is achieved with clock pulses. The SCIFA has the 64-byte FIFO buffer for both transmission and reception, reduces an overhead of the CPU, and enables continuous high-speed communication.

23.4.2 Asynchronous Mode

Operation in asynchronous mode is described below.

The transmission and reception format is selected in SCASMR, as listed in table 23.5.

- Data length is selectable from seven or eight bits.
- Parity and multiprocessor bits are selectable. So is the stop bit length (one or two bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, overrun errors, receive FIFO data full, receive data ready, and breaks.
- The number of stored data for both the transmit and receive FIFO registers is displayed.
- Clock source: Internal clock/external clock
  - Internal clock: SCIFA operates using the on-chip baud rate generator
  - External clock: The sampling rate is fixed at 1/16, so a clock with a frequency 8 times the bit rate is required. (The internal baud rate generator should not be used.)

Table 23.5 SCASMR Setting and SCIFA Transmit/Receive Format

SCASMR Setting				SCIFA Transmit/Receive Format			
Bit 6	Bit 5	Bit 3					
CHR	PE	STOP	Mode	Data Length	Multi-processor Bit	Parity Bit	Stop Bit Length
0	0	0	Asynchronous mode	8-bit data	None	None	1 bit
		1					2 bits
	1	0				Yes	1 bit
		1					2 bits
1	0	0		7-bit data		None	1 bit
		1					2 bits
	1	0				Yes	1 bit
		1					2 bits

23.4.3 Serial Operation

(1) Transmit/Receive Formats

Table 23.6 lists eight communication formats that can be selected. The format is selected by settings in SCASMR.

Table 23.6 Serial Transmit/Receive Formats

SCASMR Setting			Serial Transmit/Receive Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	START	8-Bit data								STOP		
0	0	1	START	8-Bit data								STOP	STOP	
0	1	0	START	8-Bit data								P	STOP	
0	1	1	START	8-Bit data								P	STOP	STOP
1	0	0	START	7-Bit data							STOP			
1	0	1	START	7-Bit data							STOP	STOP		
1	1	0	START	7-Bit data							P	STOP		
1	1	1	START	7-Bit data							P	STOP	STOP	

(2) 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 SCIFA's serial clock, according to the setting of the CKE bit in SCASCR.

The sampling rate is fixed at 1/16 when an external clock is input on the SCK pin, so input a clock with a frequency 8 times the bit rate.

(3) Transmitting and Receiving Data

(a) SCIFA Initialization

Before transmitting or receiving, clear the TE and RE bits to 0 in SCASCR, then initialize the SCIFA as follows.

When changing the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 initializes SCATSR. Clearing TE and RE to 0, however, does not initialize SCASSR, SCAFTDR, or SCAFRDR, which retain their previous contents.

Clear TE to 0 after all transmit data are transmitted and the TEND bit in the SCASSR is set. The transmitting data enters the high impedance state after clearing to 0 although the bit can be cleared to 0 in transmitting. Set the TFRST bit in the SCAFCR to 1 and reset SCAFTDR before TE is set again to start transmission.

Figure 23.2 is a sample flowchart for initializing the SCIFA.

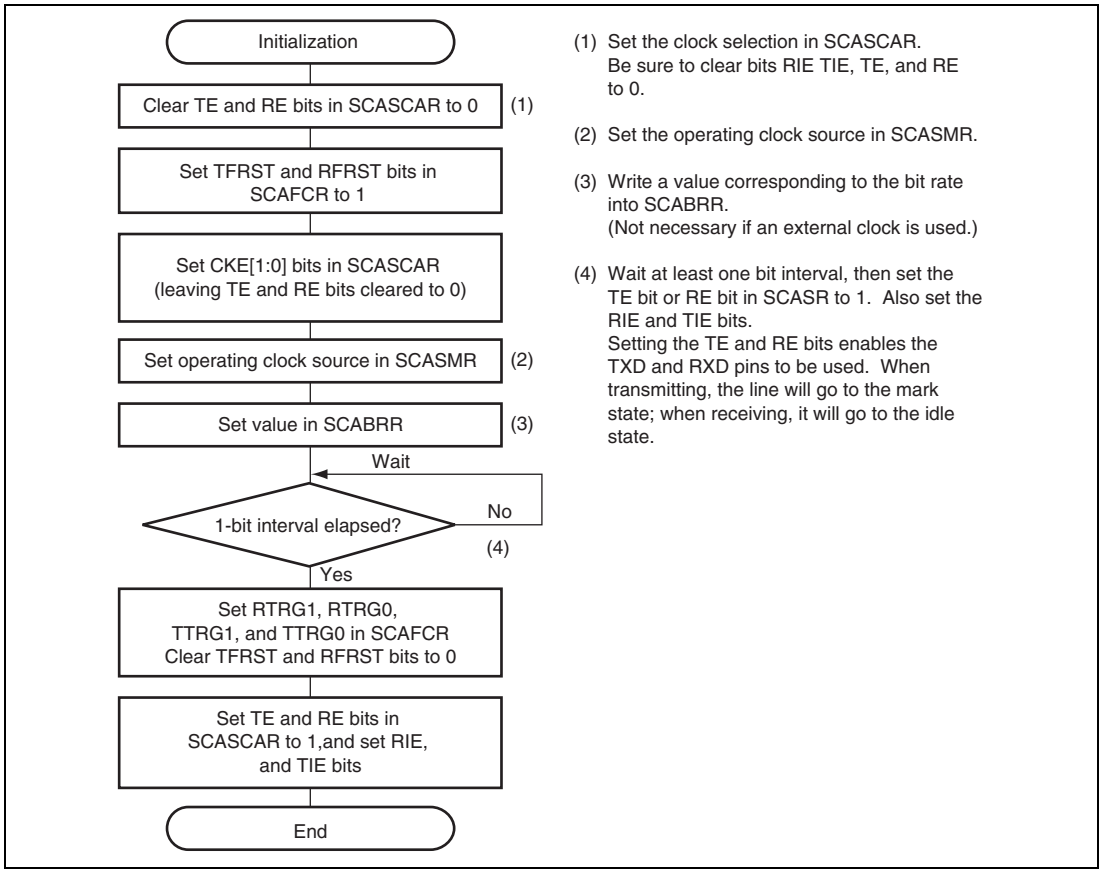
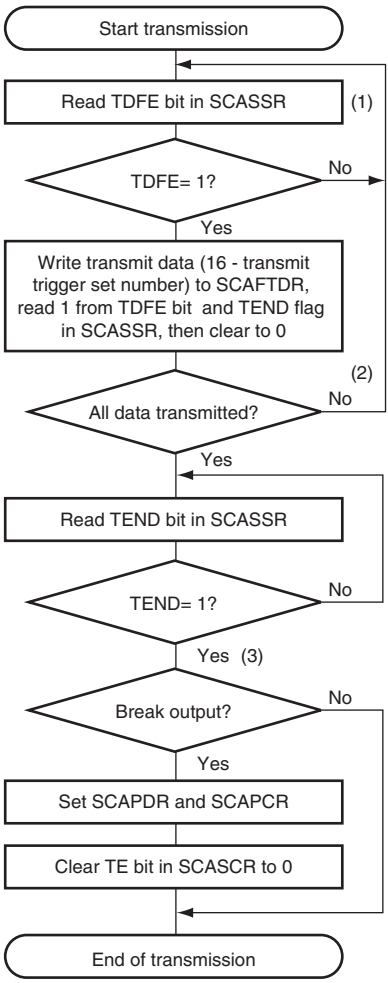


Figure 23.2 Sample SCIFA Initialization Flowchart

(b) Serial Data Transmission

Figure 23.3 shows a sample serial transmission flowchart. After SCIFA transmission is enabled, use the following procedure to perform serial data transmission.



(1) SCIFA status check and transmit data write:
Read SCASSR and check that the TDFE flag is set to 1, then write transmit data to SCAFTDR, read 1 from the TDFE and TEND flags, then clear these flags to 0.
The number of transmit data bytes that can be written is 64 - (transmit trigger number setting).

(2) Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, then write data to SCAFTDR, and then clear the TDFE flag to 0.

(3) Break output at the end of serial transmission:
To output a break in serial transmission, set the port data register (PxDR) and the control register (PxCR) of the port that is multiplexed with TXD, then clear the TE bit to 0 in the serial control register (SACSCR).
In steps 1 and 2, it is possible to ascertain the number of data bytes that can be written to SCAFTDR from the number of transmit data bytes indicated by the upper 8 bits of the FIFO data count register (SCAFDR).

Figure 23.3 Sample Serial Transmission Flowchart

In serial transmission, the SCIFA operates as described below.

1. When data is written into SCAFTDR, the SCIFA transfers the data from SCAFTDR to SCATSR and starts transmitting. Confirm that the TDFE flag in SCASSR is set to 1 before writing transmit data to SCAFTDR. The number of data bytes that can be written is (64 – transmission trigger number setting).
2. When data is transferred from SCAFTDR to SCATSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCAFTDR. When the number of transmit data bytes in SCAFTDR falls below the transmission trigger number set in SCAFCR, the TDFE flag is set. If the TIE bit in SCASCR is set to 1 at this time, a transmit-FIFO-data-empty interrupt request is generated.
   When the number of transmit data bytes matches the data set in SCATDSR while the transmit data stop function is used, the transmit operation is stopped and the TSF flag in SCASSR is set. When the TSIE bit in SCASCR is set to 1, transmit data stop interrupt request is generated. A common interrupt vector is assigned to the transmit-FIFO-data-empty interrupt and the transmit-data-stop interrupt.
   The serial transmit data is sent from the TXD pin in the following order.
   A. Start bit: One-bit 0 is output.
   B. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
   C. Parity bit: One parity bit (even or odd parity) is output. (A format in which a parity bit is not output can also be selected.)
   D. Stop bit(s): One- or two-bit 1s (stop bits) are output.
   E. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCIFA checks SCAFTDR transmit data at the timing for sending the stop bit. If data is present, the data is transferred from SCAFTDR to SCATSR, the stop bit is sent, and then serial transmission of the next frame is started.
   If there is no transmit data, the TEND flag in SCASSR is set to 1, the stop bit is sent, and then the line goes to the mark state in which 1 is output continuously.
   Figure 23.4 shows an example of the operation for transmission in asynchronous mode.

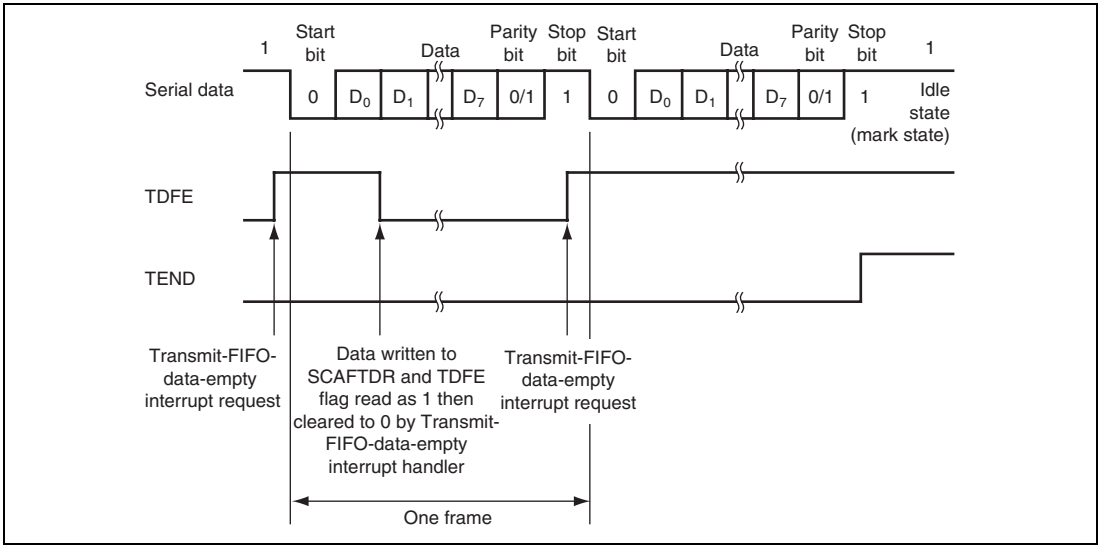


Figure 23.4 Example of Transmit Operation (Example with 8-Bit Data, Parity, One Stop Bit)

— Transmit data stop function

When the value of the SCATDSR register and the number of transmit data bytes match, transmit operation stops. Setting the TSIE bit (interrupt enable bit) allows the generation of an interrupt.

Figure 23.5 shows an example of the operation for transmit data stop function.

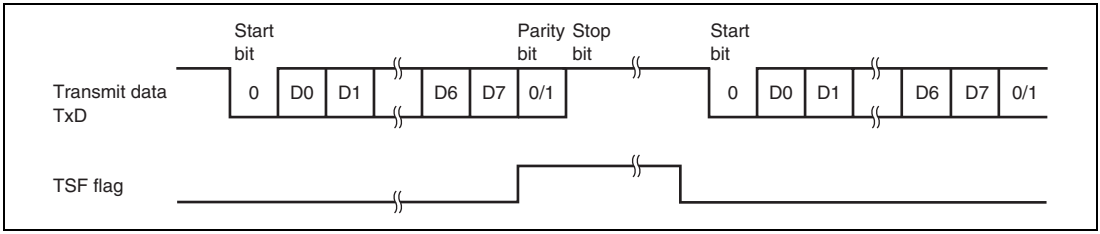


Figure 23.5 Example of Transmit Data Stop Function

Figure 23.6 shows the transmit data stop function flowchart.

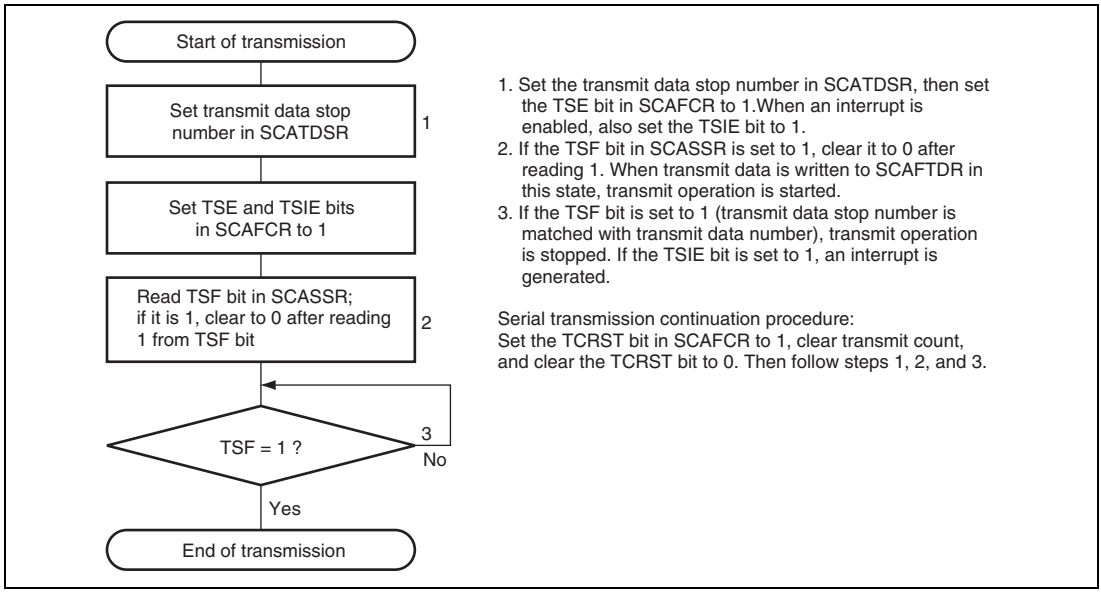


Figure 23.6 Transmit Data Stop Function Flowchart

(c) Serial Data Reception

Figures 23.7 and 18.8 show sample serial reception flowcharts. After SCIFA reception is enabled, use the following procedure to perform serial data reception.

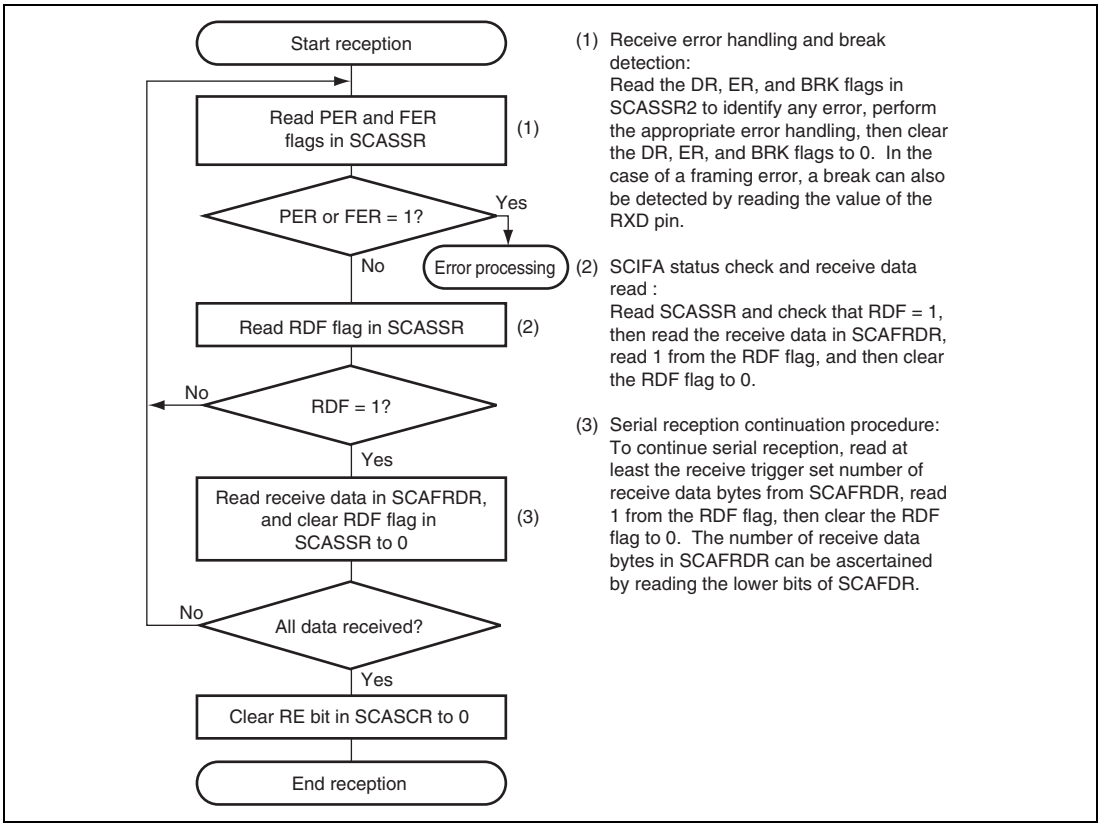
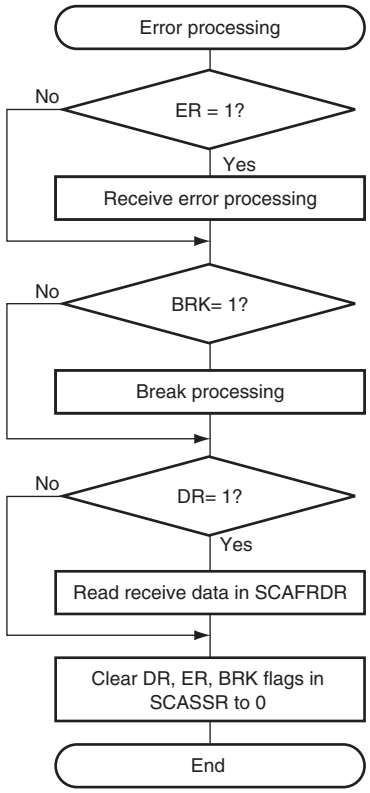


Figure 23.7 Sample Serial Reception Flowchart (1)



1. Whether a framing error or parity error has occurred in the receive data read from SCAFRDR can be ascertained from the FER and PER bits in SCASSR.
2. When a break signal is received, receive data is not transferred to SCAFRDR while the BRK flag is set. However, note that the last data in SCAFRDR is H'00 and the break data in which a framing error occurred is stored.

Figure 23.8 Sample Serial Reception Flowchart (2)

In serial reception, the SCIFA operates as described below.

1. The SCIFA monitors the transmission line, and if a 0 start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in SCARSR in LSB-to-MSB order.
3. The parity bit and stop bit are received.

   After receiving these bits, the SCIFA carries out the following checks.

   A. Stop bit check: The SCIFA checks whether the stop bit is 1. If there are two stop bits, only the first is checked.

   B. The SCIFA checks whether receive data can be transferred from SCARSR to SCAFRDR.

   C. Break check: The SCIFA checks that the BRK flag is 0, indicating that the break state is not set.

   If all the above checks are passed, the receive data is stored in SCAFRDR.

Note: Even when the receive error (framing error/parity error) is generated, receive operation is continued.

4. If the RIE bit in SCASCR is set to 1 when the RDF flag changes to 1, a receive-FIFO-data-full interrupt request is generated.
   If the ERIE bit in SCASCR is set to 1 when the ER flag changes to 1, a receive-error interrupt request is generated.
   If the BRIE bit in SCASCR is set to 1 when the BRK flag changes to 1, a break reception interrupt request is generated.
   If the DRIE bit in SCASCR is set to 1 when the DR flag changes to 1, a receive data ready interrupt request is generated.
   Note that a common vector is assigned to each interrupt source.

Figure 23.9 shows an example of the operation for reception.

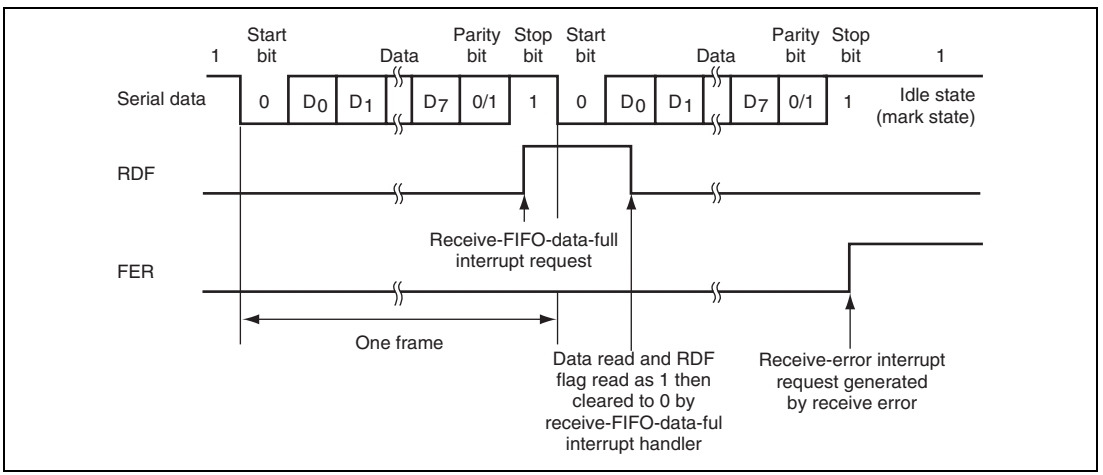


Figure 23.9 Example of SCIFA Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit)

When modem control is enabled, transmission can be stopped and restarted in accordance with the $\overline{\text{CTS}}$ input value. When $\overline{\text{CTS}}$ is set to 1, if transmission is in progress, the line goes to the mark state after transmission of one frame. When $\overline{\text{CTS}}$ is set to 0, the next transmit data is output starting from the start bit.

Figure 23.10 shows an example of the operation when modem control is used.

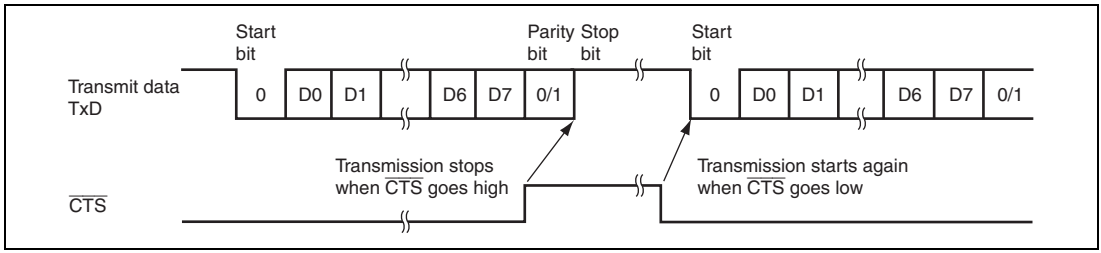


Figure 23.10 Example of $\overline{\text{CTS}}$ Control Operation

When modem control is enabled, the $\overline{\text{RTS}}$ signal goes high after the number of receive FIFO (SCAFRDR) has exceeded the number of $\overline{\text{RTS}}$ output triggers.

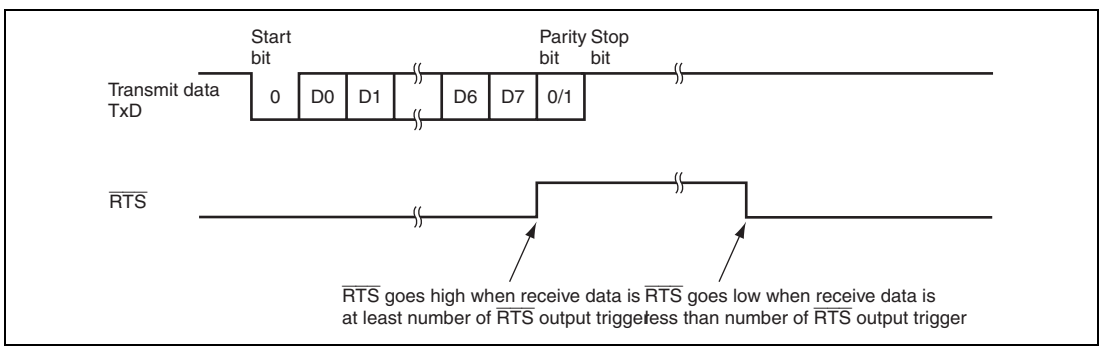


Figure 23.11 Example of $\overline{\text{RTS}}$ Control Operation

23.4.4 Synchronous Mode

Operation in synchronous mode is described below.

The SCIFA has 64-stage FIFO buffers for both transmission and reception, reducing the CPU overhead and enabling fast, continuous communication to be performed.

The operating clock source is selected using SCASMR. The SCIFA clock source is determined by the CKE[1:0] bits in SCASCR.

- Transmit/receive format: Fixed 8-bit data
- Indication of the number of data bytes stored in the transmit and receive FIFO registers
- Internal clock or external clock used as the SCIFA clock source

  When the internal clock is selected:
  The SCIFA operates on the baud rate generator clock and outputs a serial clock from SCK pin.

  When the external clock is selected:
  The SCIFA operates on the external clock input through the SCK pin.

23.4.5 Serial Operation in Synchronous Mode

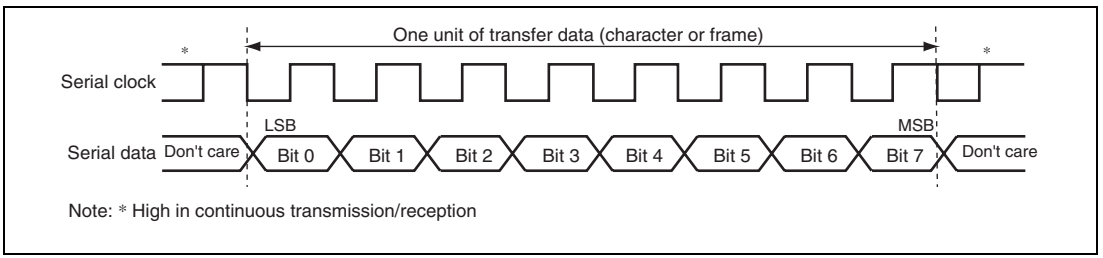


Figure 23.12 Data Format in Synchronous Communication

In synchronous serial communication, data on the communication line is output from a falling edge of the serial clock to the next falling edge. Data is guaranteed valid at the rising edge of the serial clock.

In serial communication, each character is output starting with the LSB and ending with the MSB. After the MSB is output, the communication line remains in the state of the MSB.

In synchronous mode, the SCIFA receives data in synchronization with the rising edge of the serial clock.

(1) Data Transfer Format

A fixed 8-bit data format is used. No parity or multiprocessor bits are added.

(2) Clock

An internal clock generated by the on-chip baud rate generator or an external clock input through the SCK pin can be selected as the serial clock for the SCIFA, according to the setting of the CKE[1:0] bits in SCASCR.

Eight serial clock pulses are output in the transfer of one character, and when no transmission/reception is performed, the clock is fixed high. However, when the operation mode is reception only, the synchronous clock output continues while the RE bit is set to 1. To fix the clock high every time one character is transferred, write to SCAFTDR the same number of dummy data bytes as the data bytes to be received and set the TE and RE bits to 1 at the same time to transmit the dummy data. When the specified number of data bytes are transmitted, the clock is fixed high.

(3) Data Transfer Operations

(a) SCIFA Initialization

Before transmitting and receiving data, it is necessary to clear the TE and RE bits in SCASCR to 0, then initialize the SCIFA as described below.

When the clock source, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, SCATSR is initialized. Note that clearing the TE and RE bits to 0 does not change the contents of SCASSR, SCAFTDR, or SCAFRDR. The TE bit should be cleared to 0 after all transmit data has been sent and the TEND bit in SCASSR has been set to 1. The TE bit should not be cleared to 0 during transmission; if attempted, the TXD pin will go to the high-impedance state. Before setting TE to 1 again to start transmission, the TFRST bit in SCAFCR should first be set to 1 to reset SCAFTDR.

Figure 23.13 shows sample SCIFA initialization flowcharts.

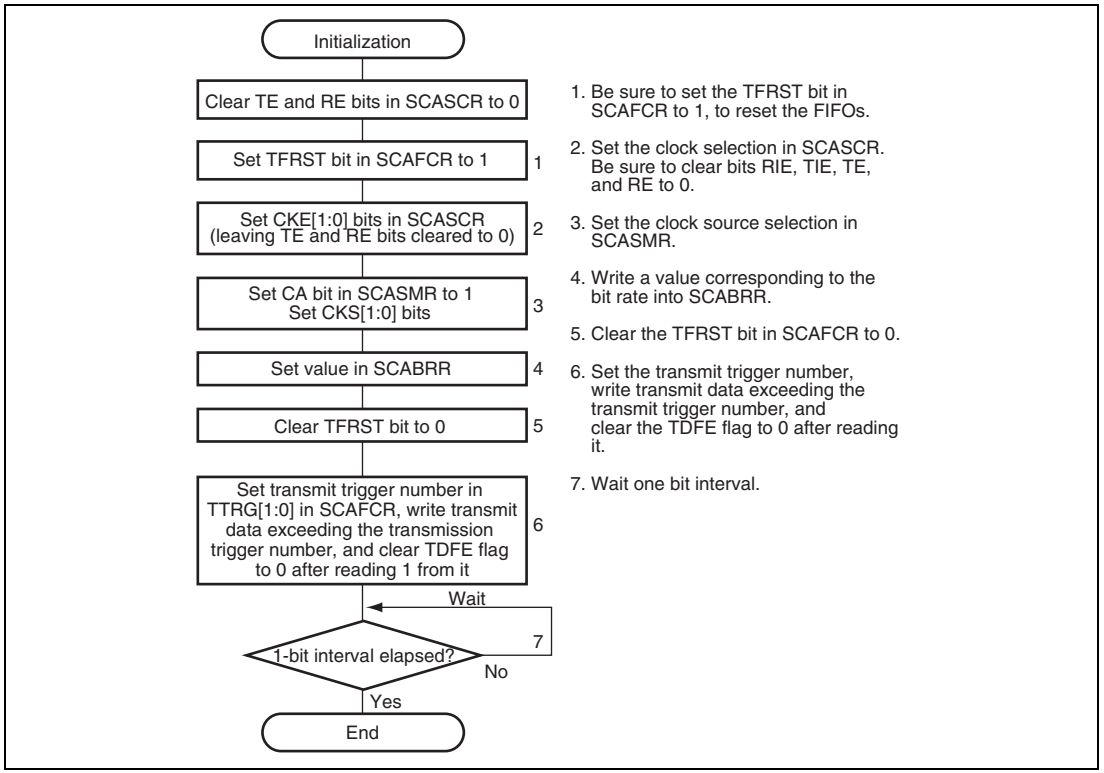


Figure 23.13 Sample SCIFA Initialization Flowchart (1) (Transmission)

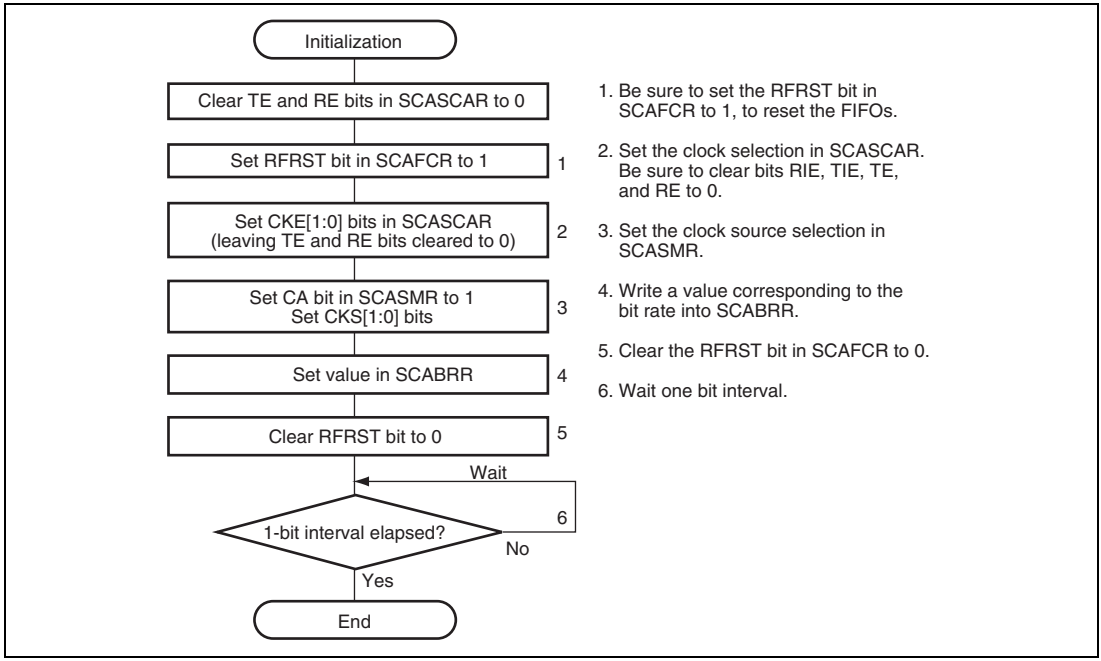


Figure 23.14 Sample SCIFA Initialization Flowchart (2) (Reception)

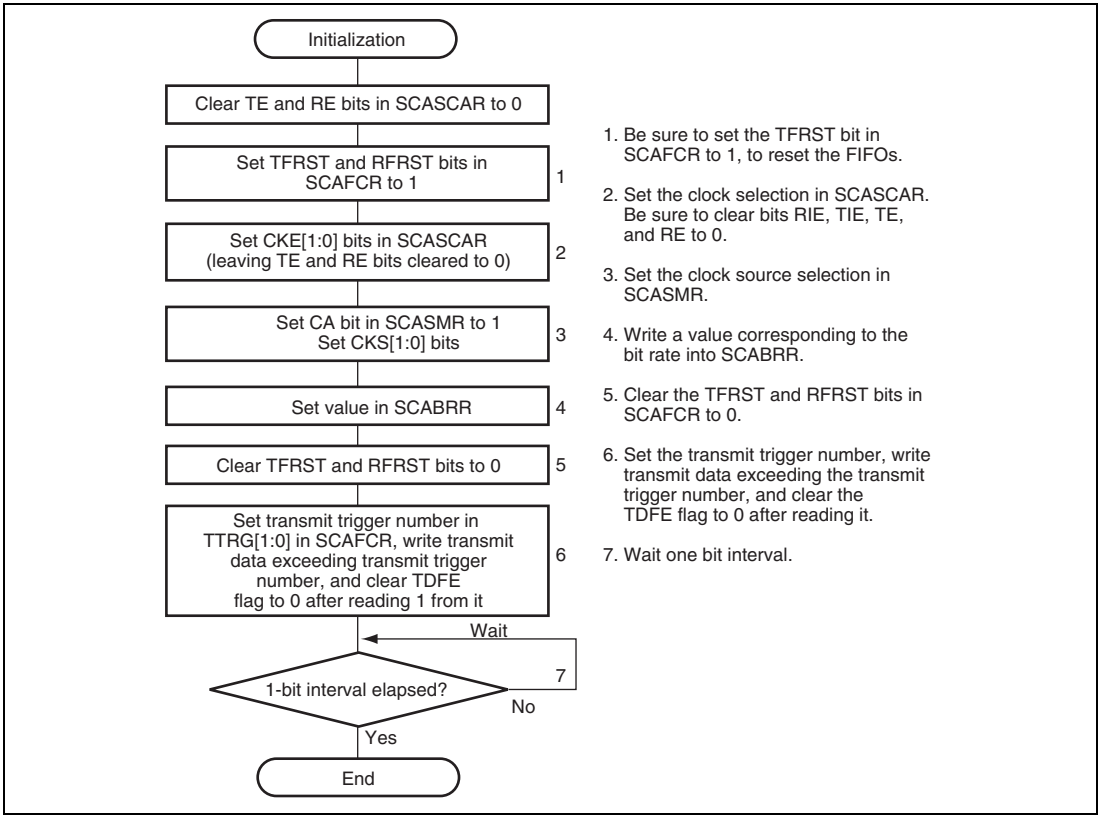


Figure 23.15 Sample SCIFA Initialization Flowchart (3)
(Simultaneous Transmission and Reception)

(b) Serial Data Transmission

Figures 23.16 and 23.17 show sample flowcharts for serial transmission.

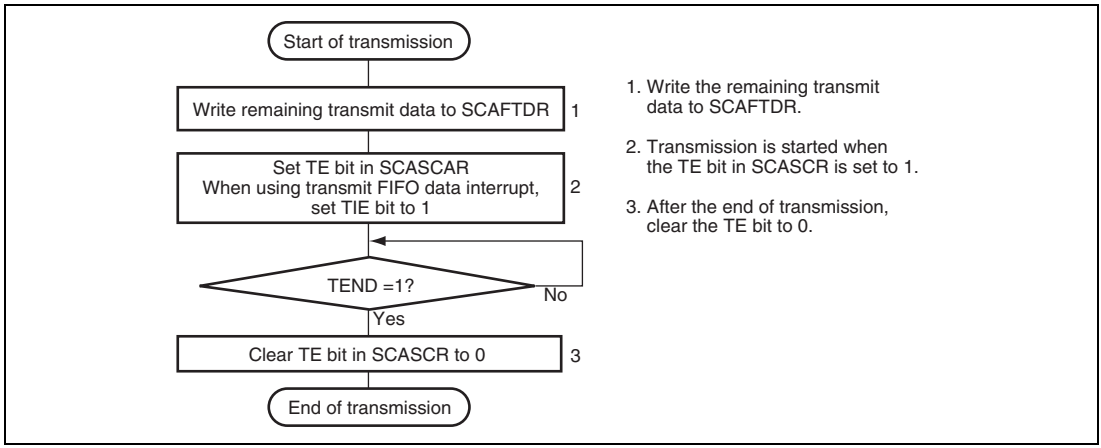


Figure 23.16 Sample Serial Transmission Flowchart (1)
(First Transmission after Initialization)

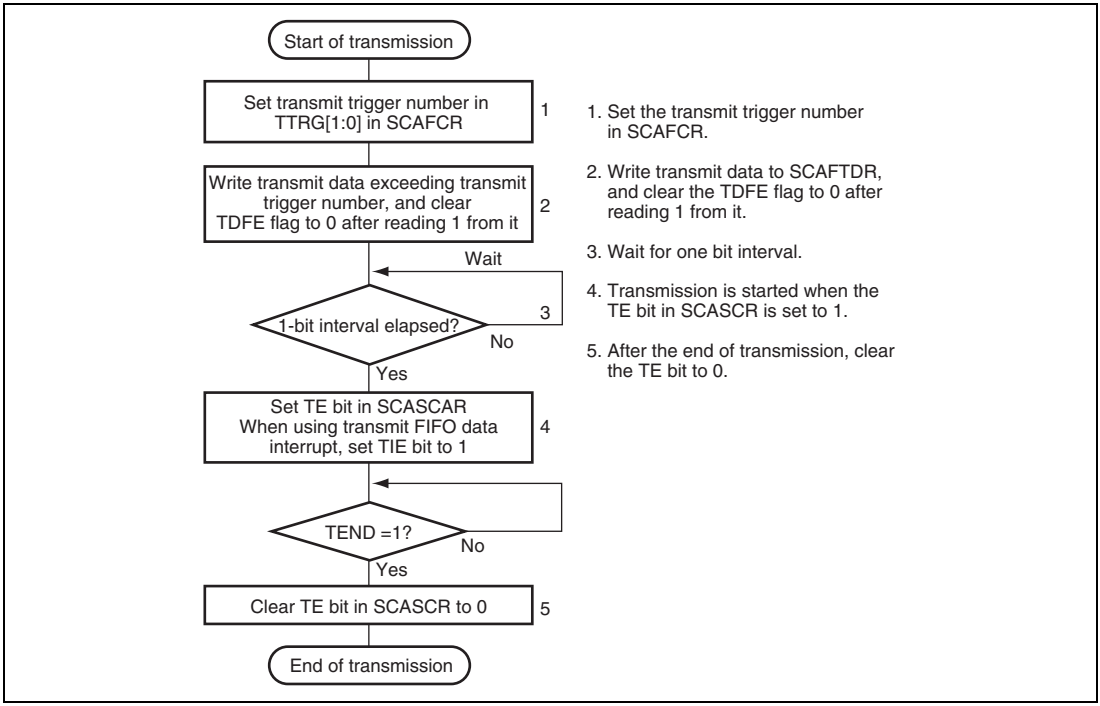


Figure 23.17 Sample Serial Transmission Flowchart (2)
(Second and Subsequent Transmission)

(c) Serial Data Reception

Figures 23.18 and 23.19 show sample flowcharts for serial reception.

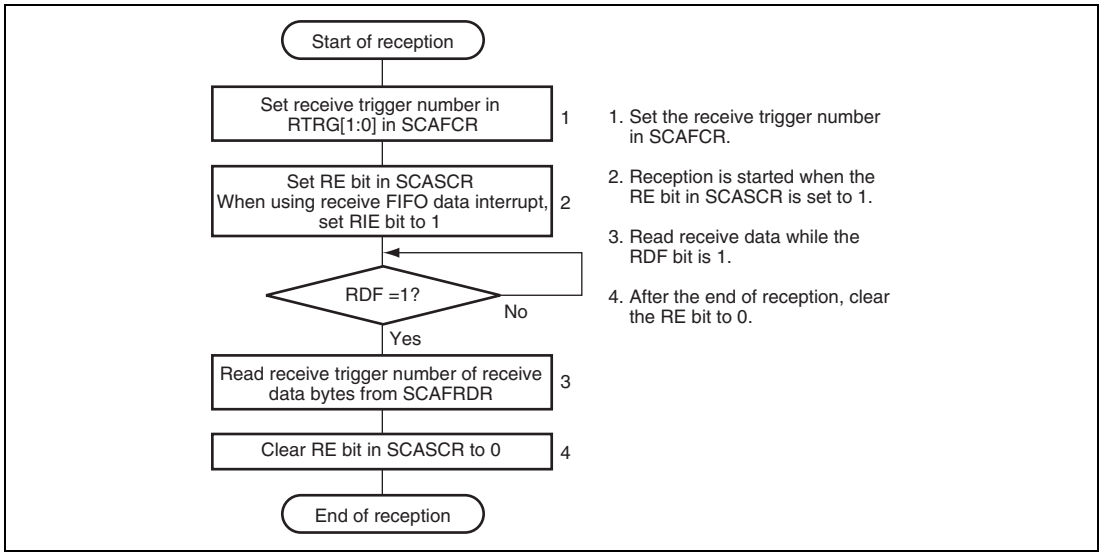
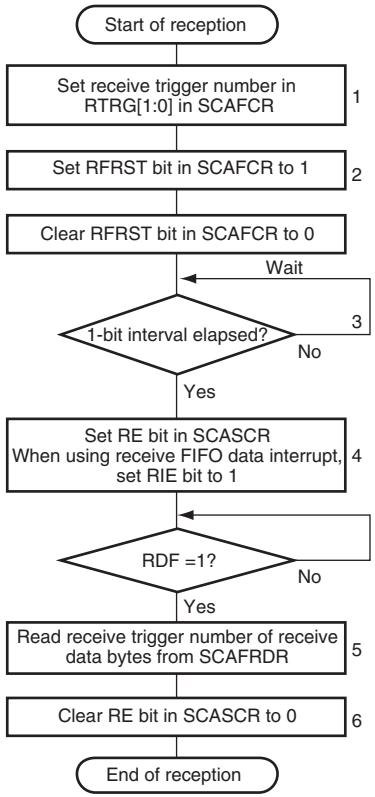


Figure 23.18 Sample Serial Reception Flowchart (1)
(First Reception after Initialization)



1. Set the receive trigger number in SCAFCR.
2. Reset the receive FIFO.
3. Wait for one bit interval.
4. Reception is started when the RE bit in SCASCR is set to 1.
5. Read receive data while the RDF bit is 1.
6. After the end of reception, clear the RE bit to 0.

Figure 23.19 Sample Serial Reception Flowchart (2)
(Second and Subsequent Reception)

(d) Simultaneous Serial Data Transmission and Reception

Figures 23.20, and 23.21 show sample flowcharts for simultaneous serial transmission and reception.

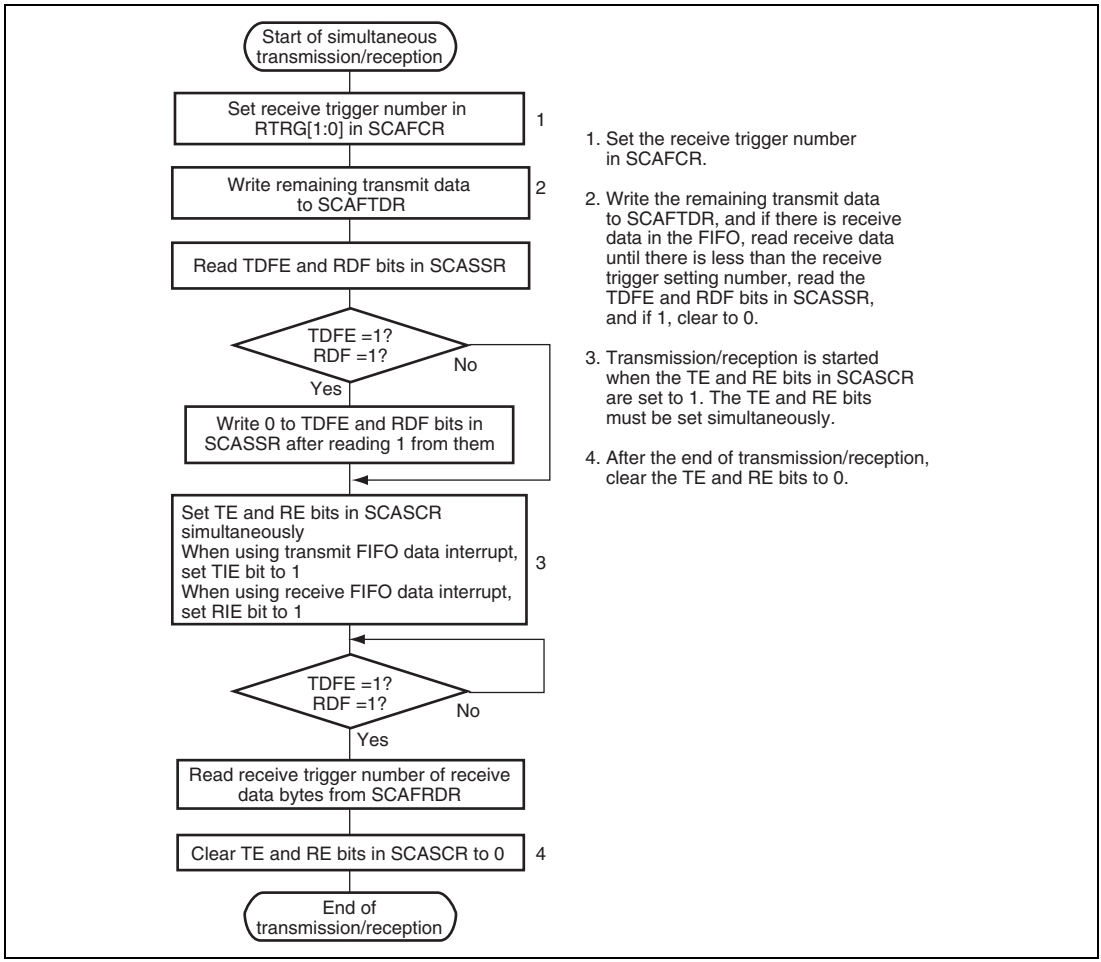


Figure 23.20 Sample Simultaneous Serial Transmission and Reception Flowchart (1) (First Transfer after Initialization)

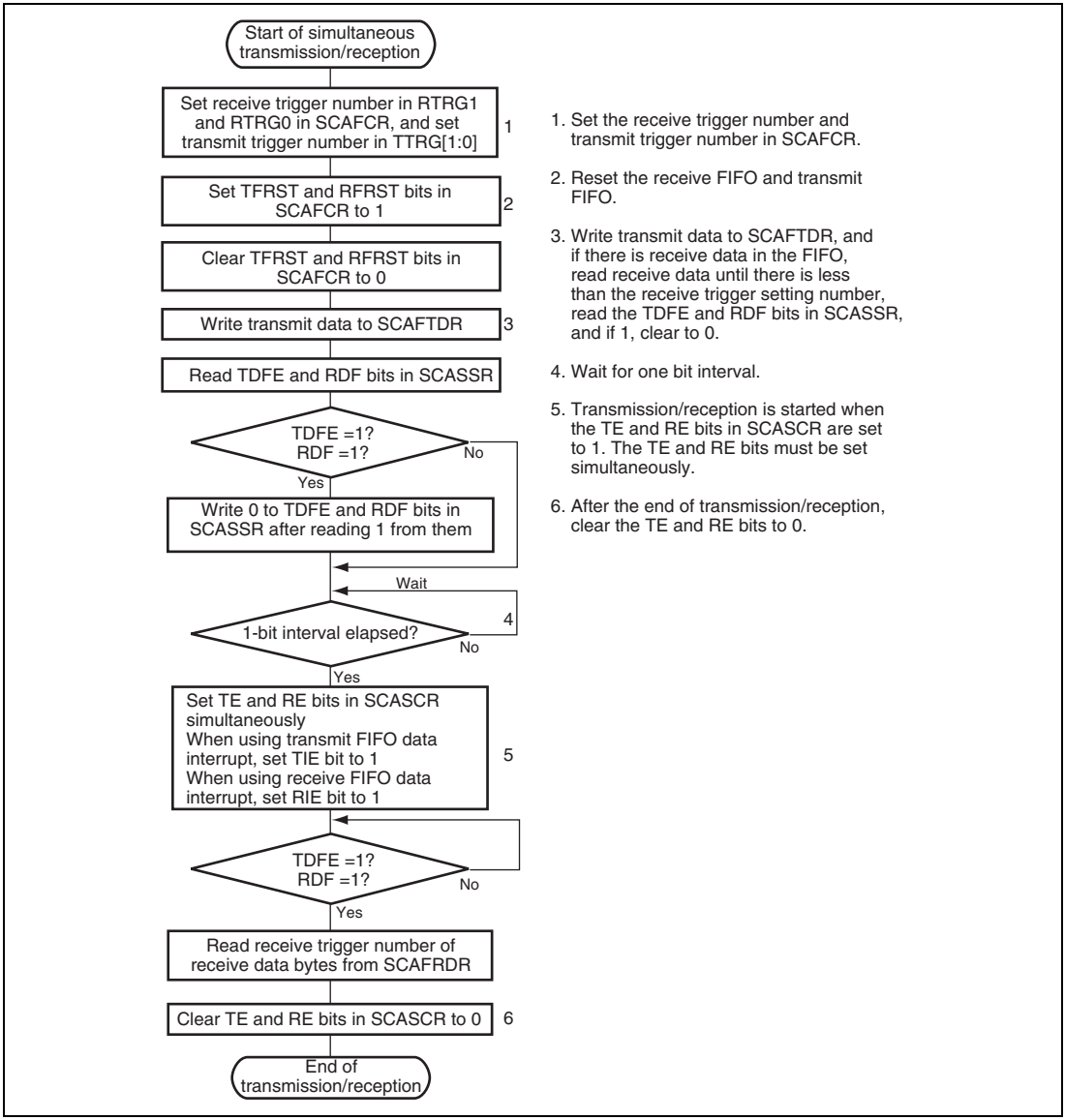


Figure 23.21 Sample Simultaneous Serial Transmission and Reception Flowchart (2) (Second and Subsequent Transfer)

23.5 Interrupt Sources and DMAC

In asynchronous mode, the SCIFA supports six interrupts: transmit-FIFO-data-empty, transmit data stop, receive-error, receive-FIFO-data-full, break receive, and receive data ready. A common interrupt vector is assigned to each interrupt source.

In synchronous mode, the SCIFA supports two interrupts: transmit-FIFO-data-empty and receive-FIFO-data-full.

Table 23.7 shows the interrupt sources. The interrupt sources are enabled or disabled by means of the TIE, RIE, ERIE, BRIE, DRIE, and TSIE bits in SCASCR.

When the TDFE flag in SCASSR is set to 1, the transmit-FIFO-data-empty interrupt request is generated. When the TSF flag in SCASSR is set to 1, the transmit-data-stop interrupt request is generated. Activating the DMAC and transferring data can be performed by the transmit-FIFO-data-empty interrupt request. The DMAC transfer request is automatically cleared when the number of data bytes written to SCAFTDR by the DMAC is increased more than that of setting transmit triggers.

When the RDF flag in SCASSR is set to 1, a receive-FIFO-data-full interrupt request is generated. Activating the DMAC and transferring data can be performed by the receive-FIFO-data-full interrupt request. The DMAC transfer request is automatically cleared when receive data is read from SCAFRDR by the DMAC until the number of receive data bytes in SCAFRDR is decreased less than that of receive triggers.

When executing the data transmission/reception using the DMAC, configure the DMAC first and enable it, then configure the SCIFA. The completion of the DMA transfer is the completion of transmission/reception.

An interrupt request is generated when the ER flag in SCASSR is set to1; the BRK flag in SCASSR is set to 1; the DR flag in SCASSR is set to 1; or the TSF flag in SCASSR is set to 1. A common interrupt vector is assigned to each interrupt source. The activation of DMAC and generation of an interrupt are not executed at the same time by the same source. To activate the DMAC, set the interrupt enable bit (TIE or RIE) corresponding to the generated interrupt source and the appropriate transfer enable bit (TDRQE or RDRQE) to 1.

Table 23.7 SCIFA Interrupt Sources

Interrupt Source	DMAC Activation
Interrupt initiated by receive error (ER), break (BRK), data ready (DR), or transmit data stop (TSF)	Not possible
Interrupt initiated by receive FIFO data full flag (RDF) or transmit FIFO data empty (TDFE)	Possible

23.6 Usage Notes

Note the following points when using the SCIFA.

(1) SCAFTDR Writing and the TDFE Flag

The TDFE flag in the serial status register (SCASSR) is set when the number of transmit data bytes written in SCAFTDR has fallen below the transmission trigger number set by bits TTRG[1:0] in SCAFCR. After TDFE is set, transmit data up to the number of empty bytes in SCAFTDR can be written, allowing efficient continuous transmission.
However, if the number of data bytes written in SCAFTDR is less than or equal to the transmission trigger number, the TDFE flag will be set to 1 again after being cleared to 0. The TDFE flag should therefore be cleared to 0 after a number of data bytes exceeding the transmission trigger number has been written to SCAFTDR.
The number of transmit data bytes in SCAFTDR can be found in the bits T[6:0] of SCAFDR.

(2) SCAFRDR Reading and the RDF Flag

The RDF flag in SCASSR is set when the number of receive data bytes in SCAFRDR has become equal to or greater than the reception trigger number set by bits RTRG[1:0] in SCAFCR. After RDF is set, receive data equivalent to the trigger number can be read from SCAFRDR, allowing efficient continuous reception.
However, if the number of data bytes in SCAFRDR exceeds the trigger number, the RDF flag will be set to 1 again after being cleared to 0. The RDF flag should therefore be cleared to 0 when 1 has been written to RDF after all receive data has been read.
The number of receive data bytes in SCAFRDR can be found in the bits R[6:0] of SCAFDR.

(3) Break Detection and Processing

Break signals can be detected by reading the RXD pin directly when a framing error (FER) is detected. In the break state the input from the RXD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. Note that, although transfer of receive data to SCAFRDR is halted in the break state, the SCIFA receiver continues to operate.

(4) Receive Data Sampling Timing and Receive Margin

An example with a sampling rate 1/16 is given. The SCIFA operates on a base clock with a frequency of 8 times the transfer rate. In reception, the SCIFA synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. The timing is shown in figure 23.22.

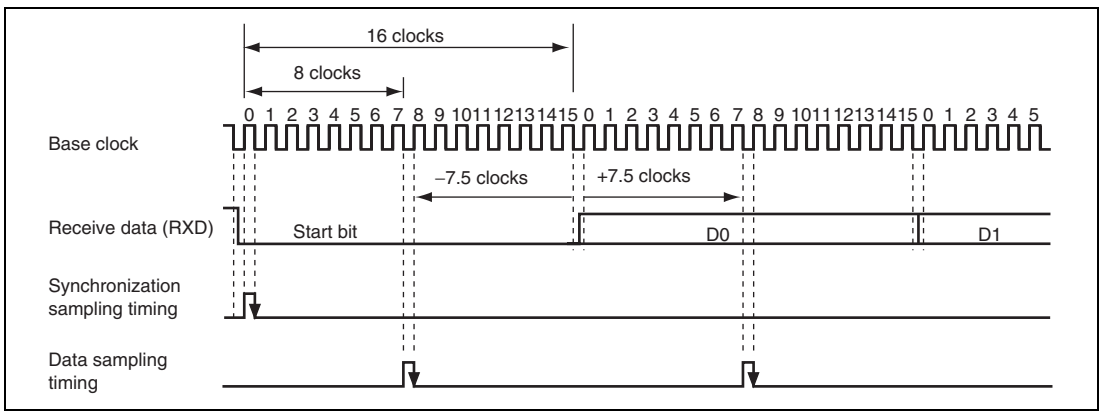


Figure 23.22 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation (1).

Equation 1:

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)\, F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\% \quad \cdots\cdots\cdots\cdots (1)$$

M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation (1), if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation (2).

When D = 0.5 and F = 0:

$$M = (0.5 - 1/(2 \times 16)) \times 100\%$$
$$= 46.875\% \quad \cdots\cdots\cdots\cdots (2)$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

(5) Limitation on Simultaneous Transmission and Reception in Clock-Synchronous Mode

When transmission and reception are taking place simultaneously, receive operation continues until the FIFO is full even if transmit operation has completed and TXD and SCK output have stopped.

To stop transmission and reception at the same time, write a quantity of transmit data to the transmit FIFO equal to the valid receive data count before enabling transmission and reception, and after transmission and reception finish, read the valid receive data and discard any remaining receive data.

Section 24 IrDA Interface (IrDA)

The IrDA interface (IrDA) performs infrared data communication conforming to IrDA standard 1.2a through an external infrared transceiver unit connected to this LSI.

The IrDA includes a UART block to control data transmission and reception as well as an infrared transmit and receive (light-emit and light-receive) pulse modulator/demodulator block and a CRC engine block in front of the UART. The UART block controls serial data transmission and reception in the asynchronous mode. The infrared transmit and receive pulse modulator/demodulator block controls communication pulses and checks pulses received through infrared baseband modulation/demodulation conforming to IrDA standard 1.0. The CRC engine block reads 8-bit input data and outputs a 16-bit CRC calculation result.

24.1 Features

The IrDA has the following UART features.

- Asynchronous serial communication
  - Data length: Eight bits
  - Stop bit: One bit
  - Parity bit: None
- Reception error detection: Overrun error and framing error
- Baud rate error correction: 16 decimal fractions can be selected.
- Baud rate count: Up to 65536 can be specified.

The IrDA has the following infrared transmit and receive pulse modulator/demodulator features.

- Infrared transmit (light-emit) pulse width: 1-bit width × 3/16 or 1.63 μs can be selected.
- Pulse width check: An out-of-standard pulse (insufficient or excess width) can be detected.
- 1.8432-MHz clock generator
  - Up to 16 can be specified for the integer part of the baud rate count.
  - The fraction part can be selected from 16 values.

The IrDA has the following CRC calculation features.

- Generator polynomial: $X^{16} + X^{12} + X^5 + 1$
- Data input
  - Input in bytes
  - CRC is calculated in 8-bit units starting from the lower bits.
- CRC output: 16-bit CRC is output.
- Maximum data length: 4096 bytes

Figure 24.1 shows a block diagram of the IrDA.

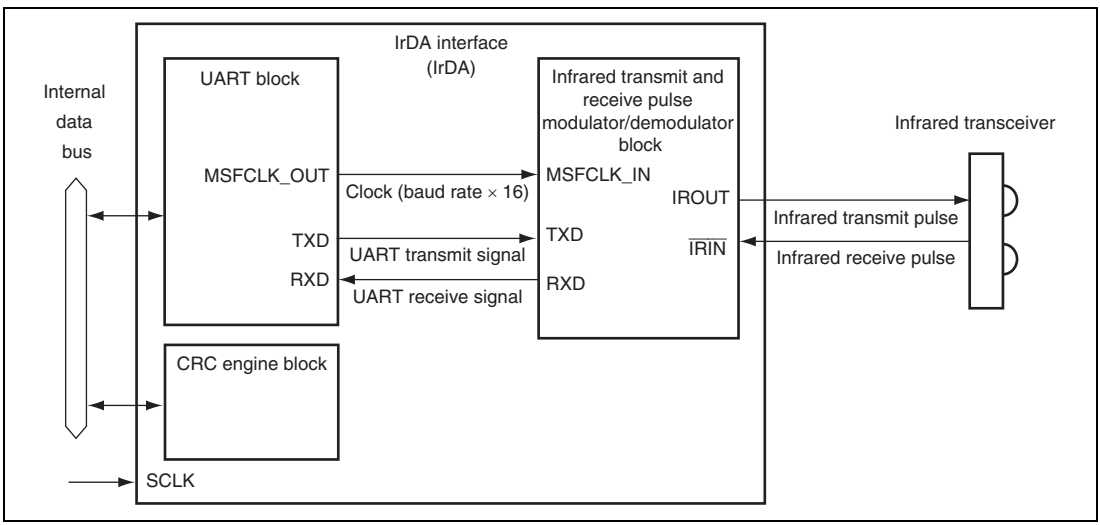


Figure 24.1 Block Diagram of IrDA

24.2 Input/Output Pins

Table 24.1 shows the IrDA pin configuration.

Table 24.1 Pin Configuration

Channel	Pin Name	Abbreviation	I/O	Function
0	IrDA0_RXD	$\overline{\text{IRIN}}$	Input	Infrared receive (light-receive) pulse input
	IrDA0_TXD	IROUT	Output	Infrared transmit (light-emit) pulse output
1	IrDA1_RXD	$\overline{\text{IRIN}}$	Input	Infrared receive (light-receive) pulse input
	IrDA1_TXD	IROUT	Output	Infrared transmit (light-emit) pulse output

Note: Since the pin functions are the same for each channel, abbreviations are used for the pin names in the following description.

24.3 Register Descriptions

Table 24.2 shows the IrDA register configuration. Table 24.3 shows the register states in each operating mode. Note that the channel numbers are omitted from the register names in descriptions subsequent to these tables.

Table 24.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Address	Access Size
0	DMA receive interrupt source clear register	IRIF0_RINTCLR	W	H'A45D 0016	16
	DMA transmit interrupt source clear register	IRIF0_TINTCLR	W	H'A45D 0018	16
	IrDA-SIR10 control register	IRIF0_SIR0	R/W	H'A45D 0020	8/16
	IrDA-SIR10 baud rate error correction register	IRIF0_SIR1	R/W	H'A45D 0022	8/16
	IrDA-SIR10 baud rate count set register	IRIF0_SIR2	R/W	H'A45D 0024	8/16
	IrDA-SIR10 status register	IRIF0_SIR3	R	H'A45D 0026	8/16
	Hardware frame processing set register	IRIF0_SIR_FRM	R/W	H'A45D 0028	8/16
	EOF value register	IRIF0_SIR_EOF	R/W	H'A45D 002A	8/16
	Flag clear register	IRIF0_SIR_FLG	W	H'A45D 002C	8/16
	UART status register 2	IRIF0_SIR_STS2	R/W	H'A45D 002E	8/16
	UART control register	IRIF0_UART0	R/W	H'A45D 0030	8/16
	UART status register	IRIF0_UART1	R	H'A45D 0032	8/16
	UART transmit data register	IRIF0_UART3	W	H'A45D 0036	8/16
	UART receive data register	IRIF0_UART4	R	H'A45D 0038	8/16
	UART interrupt mask register	IRIF0_UART5	R/W	H'A45D 003A	8/16
	UART baud rate error correction register	IRIF0_UART6	R/W	H'A45D 003C	8/16
	UART baud rate count set register	IRIF0_UART7	R/W	H'A45D 003E	8/16
	CRC engine control register	IRIF0_CRC0	R/W	H'A45D 0040	8/16
	CRC engine input data register	IRIF0_CRC1	W	H'A45D 0042	8/16
	CRC engine calculation register	IRIF0_CRC2	W	H'A45D 0044	8/16
	CRC engine output data register 1	IRIF0_CRC3	R	H'A45D 0046	8/16
	CRC engine output data register 2	IRIF0_CRC4	R	H'A45D 0048	8/16

Channel	Register Name	Abbreviation	R/W	Address	Access Size
1	DMA receive interrupt source clear register	IRIF1_RINTCLR	W	H'A45E 0016	16
	DMA transmit interrupt source clear register	IRIF1_TINTCLR	W	H'A45E 0018	16
	IrDA-SIR10 control register	IRIF1_SIR0	R/W	H'A45E 0020	8/16
	IrDA-SIR10 baud rate error correction register	IRIF1_SIR1	R/W	H'A45E 0022	8/16
	IrDA-SIR10 baud rate count set register	IRIF1_SIR2	R/W	H'A45E 0024	8/16
	IrDA-SIR10 status register	IRIF1_SIR3	R	H'A45E 0026	8/16
	Hardware frame processing set register	IRIF1_SIR_FRM	R/W	H'A45E 0028	8/16
	EOF value register	IRIF1_SIR_EOF	R/W	H'A45E 002A	8/16
	Flag clear register	IRIF1_SIR_FLG	W	H'A45E 002C	8/16
	UART status register 2	IRIF1_SIR_STS2	R/W	H'A45E 002E	8/16
	UART control register	IRIF1_UART0	R/W	H'A45E 0030	8/16
	UART status register	IRIF1_UART1	R	H'A45E 0032	8/16
	UART transmit data register	IRIF1_UART3	W	H'A45E 0036	8/16
	UART receive data register	IRIF1_UART4	R	H'A45E 0038	8/16
	UART interrupt mask register	IRIF1_UART5	R/W	H'A45E 003A	8/16
	UART baud rate error correction register	IRIF1_UART6	R/W	H'A45E 003C	8/16
	UART baud rate count set register	IRIF1_UART7	R/W	H'A45E 003E	8/16
	CRC engine control register	IRIF1_CRC0	R/W	H'A45E 0040	8/16
	CRC engine input data register	IRIF1_CRC1	W	H'A45E 0042	8/16
	CRC engine calculation register	IRIF1_CRC2	W	H'A45E 0044	8/16
	CRC engine output data register 1	IRIF1_CRC3	R	H'A45E 0046	8/16
	CRC engine output data register 2	IRIF1_CRC4	R	H'A45E 0048	8/16

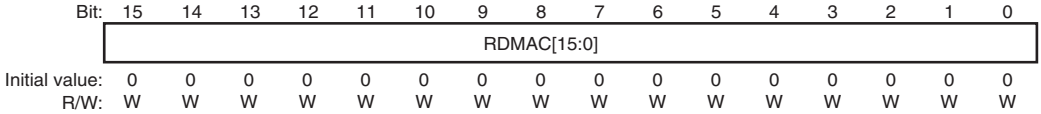
Table 24.3 Register States in Each Operating Mode

Channel	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
0	IRIF0_RINTCLR	Initialized	Retained	Retained	Retained
	IRIF0_TINTCLR	Initialized	Retained	Retained	Retained
	IRIF0_SIR0	Initialized	Retained	Retained	Retained
	IRIF0_SIR1	Initialized	Retained	Retained	Retained
	IRIF0_SIR2	Initialized	Retained	Retained	Retained
	IRIF0_SIR3	Initialized	Retained	Retained	Retained
	IRIF0_SIR_FRM	Initialized	Retained	Retained	Retained
	IRIF0_SIR_EOF	Initialized	Retained	Retained	Retained
	IRIF0_SIR_FLG	Initialized	Retained	Retained	Retained
	IRIF0_SIR_STS2	Initialized	Retained	Retained	Retained
	IRIF0_UART0	Initialized	Retained	Retained	Retained
	IRIF0_UART1	Initialized	Retained	Retained	Retained
	IRIF0_UART3	Initialized	Retained	Retained	Retained
	IRIF0_UART4	Initialized	Retained	Retained	Retained
	IRIF0_UART5	Initialized	Retained	Retained	Retained
	IRIF0_UART6	Initialized	Retained	Retained	Retained
	IRIF0_UART7	Initialized	Retained	Retained	Retained
	IRIF0_CRC0	Initialized	Retained	Retained	Retained
	IRIF0_CRC1	Initialized	Retained	Retained	Retained
	IRIF0_CRC2	Initialized	Retained	Retained	Retained
	IRIF0_CRC3	Initialized	Retained	Retained	Retained
	IRIF0_CRC4	Initialized	Retained	Retained	Retained

Channel	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
1	IRIF1_RINTCLR	Initialized	Retained	Retained	Retained
	IRIF1_TINTCLR	Initialized	Retained	Retained	Retained
	IRIF1_SIR0	Initialized	Retained	Retained	Retained
	IRIF1_SIR1	Initialized	Retained	Retained	Retained
	IRIF1_SIR2	Initialized	Retained	Retained	Retained
	IRIF1_SIR3	Initialized	Retained	Retained	Retained
	IRIF1_SIR_FRM	Initialized	Retained	Retained	Retained
	IRIF1_SIR_EOF	Initialized	Retained	Retained	Retained
	IRIF1_SIR_FLG	Initialized	Retained	Retained	Retained
	IRIF1_SIR_STS2	Initialized	Retained	Retained	Retained
	IRIF1_UART0	Initialized	Retained	Retained	Retained
	IRIF1_UART1	Initialized	Retained	Retained	Retained
	IRIF1_UART3	Initialized	Retained	Retained	Retained
	IRIF1_UART4	Initialized	Retained	Retained	Retained
	IRIF1_UART5	Initialized	Retained	Retained	Retained
	IRIF1_UART6	Initialized	Retained	Retained	Retained
	IRIF1_UART7	Initialized	Retained	Retained	Retained
	IRIF1_CRC0	Initialized	Retained	Retained	Retained
	IRIF1_CRC1	Initialized	Retained	Retained	Retained
	IRIF1_CRC2	Initialized	Retained	Retained	Retained
	IRIF1_CRC3	Initialized	Retained	Retained	Retained
	IRIF1_CRC4	Initialized	Retained	Retained	Retained

24.3.1 DMA Receive Interrupt Source Clear Register (IRIF_RINTCLR)

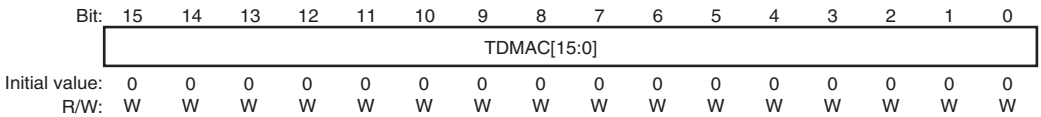
IRIF_RINTCLR is a register that clears a request for DMA transfer of received data.



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	RDMAC [15:0]	H'0000	W	Clear of DMA Transfer Request for Received Data To clear a request, write any word data to this register.

24.3.2 DMA Transmit Interrupt Source Clear Register (IRIF_TINTCLR)

IRIF_TINTCLR is a register that clears a request for DMA transfer of transmit data.



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TDMAC [15:0]	H'0000	W	Clear of DMA Transfer Request for Transmit Data To clear a request, write any word data to this register.

24.3.3 IrDA-SIR10 Control Register (IRIF_SIR0)

IRIF_SIR0 is a register that controls modulation/demodulation of infrared transmit and receive pulses.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IR TPW	IR ERRC
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	IRTPW	0	R/W	Infrared Transmit (Light-Emit) Pulse Width Select Selects the pulse width for infrared transmission. 0: Outputs three cycles of the clock input through MSFCLK_IN. 1: Outputs three cycles of the 1.8432-MHz clock specified by IRIF_SIR1 and IRIF_SIR2.
0	IRERRC	0	R/W	Clear of Error Flag for Infrared Receive (Light-Receive) Pulse Width Clears the flag for an error in the pulse width of infrared reception. 0: Does not clear the error flag. 1: Clears the error flag. Note: This bit is automatically cleared to 0 immediately after set to 1: there is no need to write 0 to this bit.

24.3.4 IrDA-SIR10 Baud Rate Error Correction Register (IRIF_SIR1)

IRIF_SIR1 is a register that specifies error correction of the baud rate (the fraction part of the baud rate count) used in the infrared transmit and receive pulse modulator/demodulator block. This value is used in combination with the IRIF_SIR2 value to generate a 1.8432-MHz clock.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	IRBCA[3:0]				—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 4	IRBCA[3:0]	0000	R/W	Infrared Modulator/Demodulator Baud Rate Error Correction Set These bits specify error correction of the baud rate (the fraction part of the baud rate count) used in the infrared transmit and receive pulse modulator/demodulator block. The value shown to the right of each setting below is used as the fraction part for the baud rate count specified in IRIF_SIR2. Select an appropriate fraction value according to the user system operating specifications. 0000: 0.0000 1000: 0.5000 0001: 0.0625 1001: 0.5625 0010: 0.1250 1010: 0.6250 0011: 0.1875 1011: 0.6875 0100: 0.2500 1100: 0.7500 0101: 0.3125 1101: 0.8125 0110: 0.3750 1110: 0.8750 0111: 0.4375 1111: 0.9375 Note: These bits must not be modified during transmission or reception. If this is attempted, correct operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.3.5 IrDA-SIR10 Baud Rate Count Set Register (IRIF_SIR2)

IRIF_SIR2 is a register that specifies the integer part of the baud rate count used in the infrared transmit and receive pulse modulator/demodulator block. This value is used in combination with the error correction value specified in IRIF_SIR1 to generate a 1.8432-MHz clock.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	IRBC[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	IRBC[3:0]	0000	R/W	Infrared Modulator/Demodulator Baud Rate Count Set These bits specify the integer part of the clock generation dividing count used in the infrared transmit and receive pulse modulator/demodulator block. Note: These bits must not be modified during transmission or reception. If this is attempted, correct operation cannot be guaranteed.

24.3.6 IrDA-SIR10 Status Register (IRIF_SIR3)

IRIF_SIR3 is a register that indicates an error in the width of the received infrared pulse during infrared pulse demodulation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRERR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	IRERR	0	R	Error Flag for Width of Received Infrared Pulse Indicates an error in the width of the received pulse during infrared pulse demodulation. 0: No error has occurred. 1: An error has occurred.

24.3.7 Hardware Frame Processing Set Register (IRIF_SIR_FRM)

IRIF_SIR_FRM is a register that specifies the processing of received data frames.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	EOFD	FRER	—	—	—	—	—	—	—	FRP
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	EOFD	1	R	EOF Detection Flag 0: EOF has been detected. 1: EOF has not been detected.
8	FRER	0	R	Frame Error Bit 0: No frame error has been detected. 1: A frame error has been detected.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	FRP	0	R/W	Frame Processing Set 0: Disables EOF detection. 1: Enables EOF detection.

24.3.8 EOF Value Register (IRIF_SIR_EOF)

IRIF_SIR_EOF is a register that specifies the EOF value.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	EOF[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	EOF[7:0]	H'C1	R/W	EOF Set These bits specify the EOF value to be detected.

24.3.9 Flag Clear Register (IRIF_SIR_FLG)

IRIF_SIR_FLG is a register that clears the frame error flag and EOF flag. Writing any data to the upper or lower eight bits of this register clears the corresponding flag.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FRERC[7:0]								EOFC[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	FRERC[7:0]	H'00	W	Frame Error Flag Clear Writing any byte data to these bits (the upper eight bits of the register) clears the frame error flag.
7 to 0	EOFC[7:0]	H'00	W	EOF Error Flag Clear Writing any byte data to these bits (the lower eight bits of the register) clears the EOF error flag.

24.3.10 UART Status Register 2 (IRIF_SIR_STS2)

IRIF_SIR_STS2 is a register that indicates the operating status during data reception.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	IRSME	IROVE	IRFRE	IRPRE	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	IRSME	0	R/W	Receive Sum Error Flag 0: No receive sum error has occurred. 1: A receive sum error has occurred.

Bit	Bit Name	Initial Value	R/W	Description
5	IROVE	0	R/W	Receive Overrun Error Flag 0: No receive overrun error has occurred. 1: A receive overrun error has occurred.
4	IRFRE	0	R/W	Receive Framing Error Flag 0: No receive framing error has occurred. 1: A receive framing error has occurred.
3	IRPRE	0	R/W	Receive Parity Error Flag 0: No receive parity error has occurred. 1: A receive parity error has occurred.
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: Writing to this register clears all error flags.

24.3.11 UART Control Register (IRIF_UART0)

IRIF_UART0 is a register that controls data transmission and reception.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TBEC	RIE	TIE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	TBEC	0	W	Transmit Data Clear Clears the empty flag for the UART transmit buffer. Although the flag is cleared by writing 1 to this bit, the transmit data register is not cleared. This bit is always read as 0 even after 1 is written to it. 0: Does not clear the flag. 1: Clears the flag.
1	RIE	0	R/W	Receive Enable Starts or stops UART reception. If 0 is written to this bit during reception, the operation stops after one unit of data is received. 0: Stops reception. 1: Starts reception.
0	TIE	0	R/W	Transmit Enable Starts or stops UART transmission. If 0 is written to this bit during transmission, the operation stops after one unit of data is transmitted. 0. Stops transmission. 1: Starts transmission.

24.3.12 UART Status Register (IRIF_UART1)

IRIF_UART1 is a register that includes flags for indicating the UART operation status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	UR SME	UR OVE	UR FRE	UR PRE	RBF	TSBE	TBE
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
8, 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	URSME	0	R	Receive Sum Error Flag This bit is set to 1 when any of the UART error flags (receive parity error flag, receive framing error flag, or receive overrun error flag) is set to 1, and is cleared to 0 when none of the UART error flags is set to 1. The error flag is cleared when the receive data register is read by the system. If the next data is received before the receive data register is read, the error flag is updated according to the latest received data status (previous received data error flags are overwritten). 0: No error has occurred. 1: An error has occurred.
5	UROVE	0	R	Receive Overrun Error Flag This bit is set to 1 when the next received data is stored in the UART receive data register before the previous received data is read from the register by the system, and is cleared to 0 when the receive data register is read by the system. (The previous received data is always overwritten with the latest received data.) 0: No error has occurred. 1: An error has occurred.
4	URFRE	0	R	Receive Framing Error Flag This bit is cleared to 0 when a 1-bit stop bit is added behind the UART received, and is set to 1 when a 2-bit stop bit is added. The error flag is cleared when the receive data register is read by the system. If the next data is received before the receive data register is read, the error flag is updated according to the latest received data status (previous received data error flags are overwritten). 0: No error has occurred. 1: An error has occurred.

Bit	Bit Name	Initial Value	R/W	Description
3	URPRE	0	R	Receive Parity Error Flag Appending of parity bits to data being transmitted or received by the UART has been disabled. Therefore, this flag is always 0. 0: No error has occurred. 1: An error has occurred.
2	RBF	0	R	Receive Buffer Full Flag This bit is set to 1 when received data is stored in the UART receive data register (even if any of the receive parity error, receive framing error, and receive overrun error has occurred), and is cleared to 0 when the receive data register is read by the system. 0: No received data in the buffer. 1: Received data is in the buffer.
1	TSBE	1	R	Transmit Shift Buffer Empty Flag This bit is set to 1 when UART transmission is completed (the UART transmit shift buffer becomes empty), and is 0 during UART transmission 0: During transmission 1: Transmission completed
0	TBE	1	R	Transmit Buffer Empty flag This bit is set to 1 when data is sent from the UART transmit data register to the transmit shift buffer (the transmit data register becomes empty) or when 1 is written to the transmit data clear bit in the UART control register, and is cleared to 0 when transmit data is written to the transmit data register. 0. Transmit data is in the register. 1: No transmit data is in the register.

24.3.13 UART Mode Register (IRIF_UART2)

IRIF_UART2 specifies the data format or mode of transmission for the transmission of serial data. Usage should normally be with this register in the initial state. If the value of IRIF_UART2 is changed, operation is not guaranteed.

24.3.14 UART Transmit Data Register (IRIF_UART3)

IRIF_UART3 is a register that stores transmit data.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TD[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	W	Reserved The write value should always be 0.
7 to 0	TD[7:0]	H'00	W	UART Transmit Data The data to be transmitted should be specified here.

Note: Write to this register while the transmit buffer empty (TBE) flag in the UART status register (IRIF_UART1) is set to 1. If this register is written to while the flag is set to 0, undefined data may be transmitted depending on the timing.

24.3.15 UART Receive Data Register (IRIF_UART4)

IRIF_UART4 is a register that stores received data.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RD[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	RD[7:0]	H'00	R	UART Receive Data These bits store received data.

Note: Read this register while the receive buffer full (RBF) flag in the UART status register (IRIF_UART1) is set to 1. If this register is read while the flag is set to 0, undefined data may be read depending on the timing.

24.3.16 UART Interrupt Mask Register (IRIF_UART5)

IRIF_UART5 is a register that enables or disables UART interrupts.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	RSEIM	—	—	—	RBFIM	TSBEIM	TBEIM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6	RSEIM	0	R/W	Receive Sum Error Flag Interrupt Mask Enables or disables an interrupt by the receive sum error flag. 0: Disables an interrupt. 1: Enables an interrupt.
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	RBFIM	0	R/W	Receive Buffer Full Flag Interrupt Mask Enables or disables an interrupt by the receive buffer full flag. 0: Disables an interrupt. 1: Enables an interrupt.
1	TSBEIM	0	R/W	Transmit Shift Buffer Empty Flag Interrupt Mask Enables or disables an interrupt by the transmit shift buffer empty flag. When the flag is set to 1, interrupt processing is started. 0: Disables an interrupt. 1: Enables an interrupt.
0	TBEIM	0	R/W	Transmit Buffer Empty Flag Interrupt Mask Enables or disables an interrupt by the transmit buffer empty flag. When the flag is set to 1, interrupt processing is started. 0: Disables an interrupt. 1: Enables an interrupt.

Note: The TSBEIM and TBEIM flags must not both be set to 1 (enabled) at the same time.

24.3.17 UART Baud Rate Error Correction Register (IRIF_UART6)

IRIF_UART6 is a register that specifies error correction of the baud rate for data communication.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	UABCA[3:0]				—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 4	UABCA[3:0]	0000	R/W	Baud Rate Error Correction Set These bits specify error correction of the baud rate (the fraction part of the baud rate count) used by the UART in combination with the value specified in the UART baud rate count set register. The value shown to the right of each setting below is used as the fraction part for the baud rate count. Select an appropriate fraction value according to the user system operating specifications. 0000: 0.0000 1000: 0.5000 0001: 0.0625 1001: 0.5625 0010: 0.1250 1010: 0.6250 0011: 0.1875 1011: 0.6875 0100: 0.2500 1100: 0.7500 0101: 0.3125 1101: 0.8125 0110: 0.3750 1110: 0.8750 0111: 0.4375 1111: 0.9375
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.3.18 UART Baud Rate Count Set Register (IRIF_UART7)

IRIF_UART7 is a register that specifies the baud rate count for data communication.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UABC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	UABC[15:0]	H'0000	R/W	Baud Rate Count Set These bits specify the integer part of the baud rate count used by the UART in combination with the value specified in the UART baud rate error correction register. For details of baud rate setting, refer to section 24.4.1 (4), Baud Rate Setting for Data Transmission and Reception.

24.3.19 CRC Engine Control Register (IRIF_CRC0)

IRIF_CRC0 is a register that activates the CRC engine and counts the number of input data items.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRC_RST	—	—	—	CRC_CT[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	CRC_RST	0	W	CRC Engine Reset Clears all registers related to CRC calculation. After reset, this bit automatically returns to 0 and there is no need to write 0 to this bit. 0: Normal CRC calculation 1: CRC engine reset
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
11 to 0	CRC_CT [11:0]	H'000	R	CRC Engine Input Data Count The number of data items that have been input to the CRC engine can be read. After the data count reaches 4096, it will wrap around to 0.

24.3.20 CRC Engine Input Data Register (IRIF_CRC1)

IRIF_CRC1 is a register that stores data to be input to the CRC engine.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CRC_IN[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved The write value should always be 0.
7 to 0	CRC_IN[7:0]	H'00	W	CRC Engine Input Data The data to be input to the CRC engine should be specified here. The specified data is shifted into the CRC calculation register, starting with the LSB (CRC_IN0) of the data into the MSB (CRC_REG15) side of the calculation register. For details, see figure 24.7.

24.3.21 CRC Engine Calculation Register (IRIF_CRC2)

IRIF_CRC2 is a register used for CRC calculation. Generally, this register must be accessed only when the initial value for CRC calculation should be specified.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRC_REG[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	CRC_REG [15:0]	H'0000	W	CRC Engine Calculation Data The initial value for CRC calculation should be specified here.

24.3.22 CRC Engine Output Data Register 1 (IRIF_CRC3)

IRIF_CRC3 is a register that stores the CRC engine calculation result.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRC_OUT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	CRC_OUT [15:0]	H'0000	R	CRC Engine Output Data The result of calculation in the CRC engine can be read.

24.3.23 CRC Engine Output Data Register 2 (IRIF_CRC4)

IRIF_CRC4 is a register that stores the CRC engine calculation result. The bit order (LSB–MSB) of the IRIF_CRC3 value is inverted and stored in this register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRC_OUT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	CRC_OUT [15:0]	H'0000	R	CRC Engine Output Data The result of calculation in the CRC engine can be read. Bit 15 holds the LSB value and bit 0 holds the MSB value of the calculation result.

24.4 Operation

24.4.1 UART

The UART carries out serial communication in asynchronous mode.

(1) Data Format

Figure 24.2 shows the format of data that can be handled in the UART.

- ST bit (start bit)
  The ST bit indicates the beginning of data transmission or reception. A low-level signal of 1-bit duration is sent immediately before the data bits.
- Bits 0 to 7 (data bits)
  The data bits hold the transmit data specified in IRIF_UART3 or the received data to be stored in IRIF_UART4. Eight bits are used as the data bits for one character and the bits are transferred with the LSB first.
- SP bit (stop bit)
  The SP bit indicates the end of data transmission or reception. A high-level signal of 1-bit duration is sent immediately after the data bits.

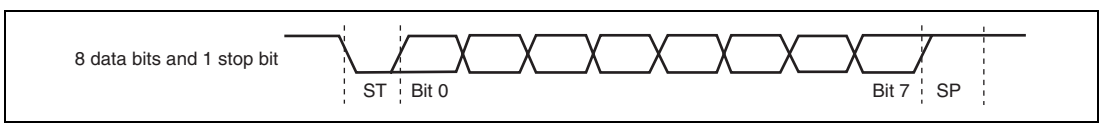


Figure 24.2 Data Transmission and Reception Format

(2) Data Transmission Timing

Figure 24.3 shows the data transmission timing controlled by the UART.

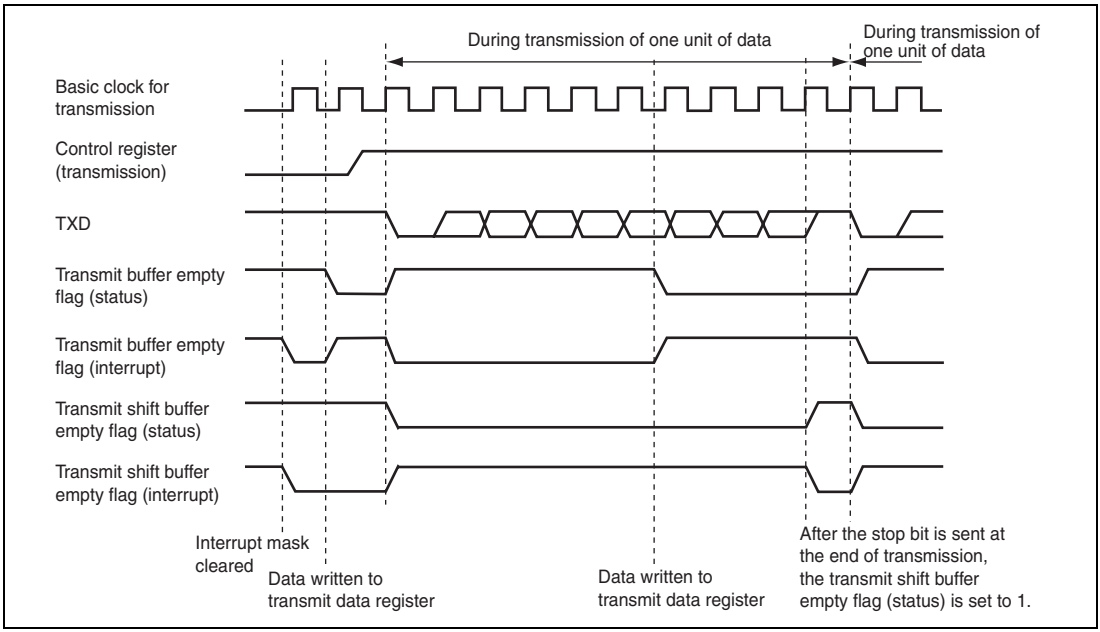
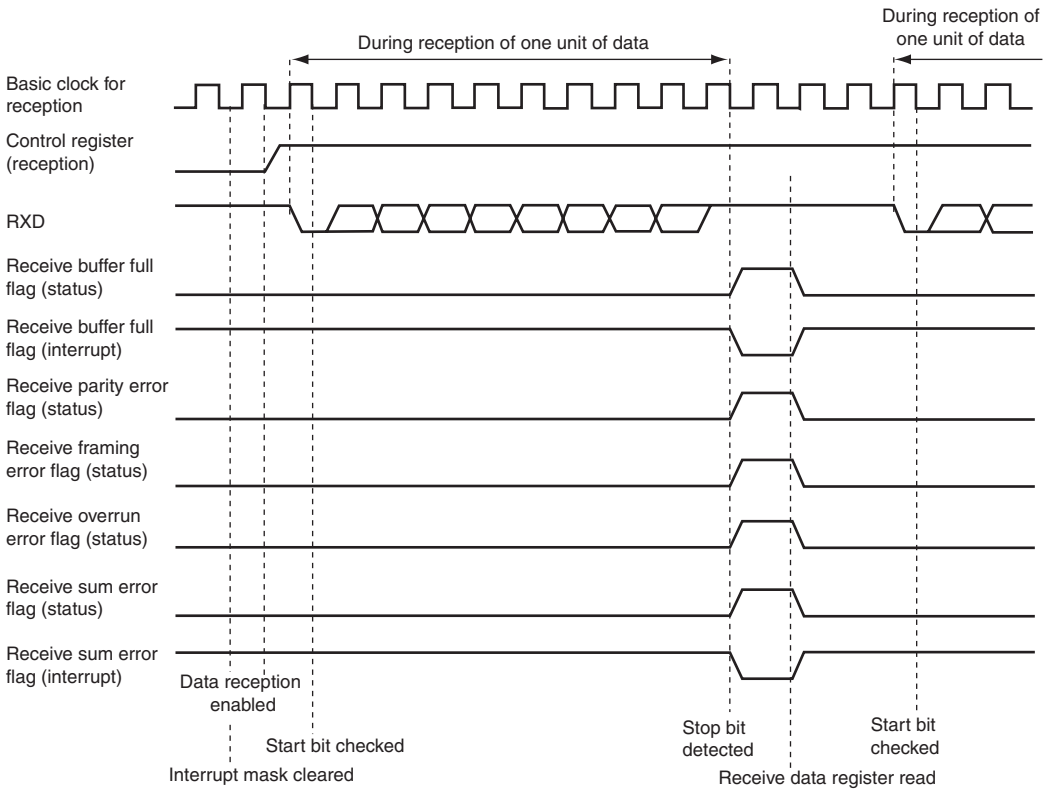


Figure 24.3 Data Transmission Timing

(3) Data Reception Timing

Figure 24.4 shows the data reception timing controlled by the UART.

When the last stop bit of the received data is detected, the received data is stored and the receive flags are set or cleared appropriately.



Note: The data reception control circuit recognizes the start bit when a falling edge is detected on the RXD input pin. After that, the control circuit samples the signal level at the center of the start bit duration, and if a low level is detected on the RXD input pin, the control circuit starts the reception sequence.

Figure 24.4 Data Reception Timing

(4) Baud Rate Setting for Data Transmission and Reception

The baud rate for UART data transmission and reception is calculated by the following equation.

$$\text{Baud rate [bps]} = \frac{\text{System clock (SCLK) [Hz]}}{\text{UABCA} + (\text{UABC} + 1) \times 16}$$

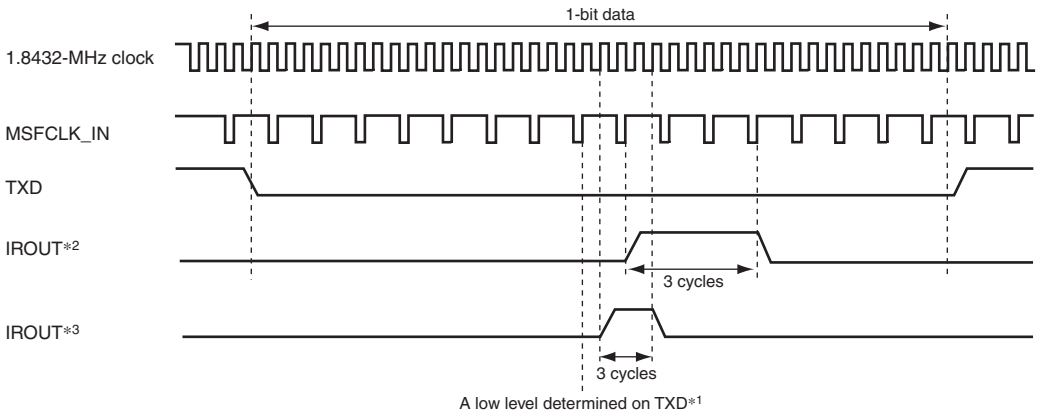
UABC: Baud rate counter value for data transmission and reception
(value specified in the UABC15 to UABC0 bits in IRIF_UART7)
UABCA: Baud rate counter value for data transmission and reception
(value selected with the UABCA3 to UABCA0 bits in IRIF_UART6)

The clock shown in the above equation is the clock input to the controller. Every time the integer part of the baud rate count is reloaded, the fraction part selected by the baud rate error correction register is accumulated. When an overflow occurs during the fraction part accumulation, 1 is added to the integer part and the resultant value is reloaded to the counter. This means that the error in the baud rate count is eliminated by adding 1 to the count when the accumulated error in the fraction part reaches 1.

24.4.2 Transmit and Receive Pulse Modulation and Demodulation

(1) Transmission of Infrared Light-Emit Pulse Data

The data transmitted from the UART is encoded into a waveform conforming to IrDA standard 1.0, and infrared transmit (light-emit) pulse data is sent to the infrared transceiver device. Figure 24.5 shows the encoding timing.

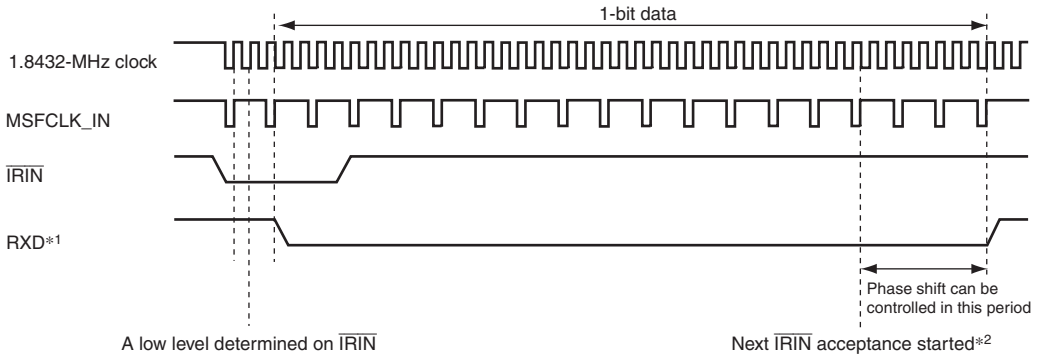


Notes: 1. A low level through TXD is determined when a low level is continuously detected for eight MSFCLK_IN cycles. After that, a pulse having a width equal to three cycles of the specified clock is not output.
2. This waveform shows the output timing at the IROUT pin when the infrared transmit pulse width select bit is set to 0 (three cycles of MSFCLK_IN).
3. This waveform shows the output timing at the IROUT pin when the infrared transmit pulse width select bit is set to 1 (three cycles of the 1.8432-MHz clock).

Figure 24.5 Timing for Encoding Infrared Transmit (Light-Emit) Pulse Data

(2) Reception of Infrared Light-Receive Pulse Data

The infrared receive (light-receive) pulse data is sent from the infrared transceiver and its waveform conforming to IrDA standard 1.0 is decoded and transferred to the UART. Figure 24.6 shows the decoding timing.



Notes: 1. When a low level is detected on the IRIN pin for at least two continuous cycles of the 1.8432-MHz clock (an internally generated clock), a low pulse having a width of 17 MSFCLK_IN cycles is output. This eliminates the phase shift (lag) in the IRIN input pulse for one MSFCLK_IN cycle.

2. The next IRIN input can be detected 14 MSFCLK_IN cycles after a low level is determined on the IRIN pin. Until 14 cycles have passed, a low level on the IRIN pin is not detected as an effective pulse. This eliminates the phase shift (lead) in the IRIN input pulse for one MSFCLK_IN cycle.

Figure 24.6 Timing for Decoding Infrared Receive (Light-Receive) Pulse Data

(3) Internal Clock Generation for Transmit and Receive Pulse Modulation and Demodulation

The 1.8432-MHz clock used in the transmit and receive pulse modulator/demodulator block is generated by the following equation.

$$1.8432\text{-MHz clock} = \frac{\text{System clock (SCLK) [Hz]}}{\text{IRBCA} + (\text{IRBC} + 1)}$$

IRBC: Baud rate counter value for infrared transmit and receive pulse modulation and demodulation (value specified in the IRBC3 to IRBC0 bits in IRIF_SIR2)
IRBCA: Baud rate error correction value for infrared transmit and receive pulse modulation and demodulation (value selected with the IRBCA3 to IRBCA0 bits in IRIF_SIR1)

The 1.8432-MHz clock is used to measure 1.63 μs and is necessary in the following operations.

- Generating a 1.63-μs infrared transmit (light-emit) pulse
- Recognizing an infrared receive (light-receive) pulse
- Detecting an error in the width (shorter than the standard) of an infrared receive (light-receive) pulse

The clock shown in the above equation is the clock input to the controller. Every time the integer part of the baud rate count is reloaded, the fraction part selected by the baud rate error correction register is accumulated. When an overflow occurs during fraction part accumulation, 1 is added to the integer part and the resultant value is reloaded to the counter. This means that the error in the baud rate count is eliminated by adding 1 to the count when the accumulated error in the fraction part reaches 1.

(4) Notes on Infrared Transmit and Receive Pulse Modulation and Demodulation

(a) Errors in the width of infrared receive (light-receive) pulses

The infrared receive pulse error flag (IRERR) is set to 1 when the width of an infrared receive (light-receive) pulse is determined as outside of the standard; that is, in the following cases.

- When a low level of an infrared receive pulse is detected for only one cycle of the 1.8432-MHz clock (shorter than the standard)
- When a low level of an infrared receive pulse is detected for five or more continuous cycles of the MSFCLK_IN clock (longer than the standard)
- When a high level of an infrared receive pulse is detected for only one cycle of the 1.8432-MHz clock (lacking a pulse)

Note that the following case is not detected as an error although this pulse width does not satisfy the standard.

- When a pulse is input for less than one cycle of the 1.8432-MHz clock

Note: When the pulse width is longer than the standard value, the error flag is set to 1 but the pulse is determined as effective and the modulator/demodulator performs the normal operation (outputs a low level through RXD). The above errors in the pulse width can be detected even during data reception after an infrared receive pulse is recognized.

(b) Interface with the infrared transceiver device

The polarity of the input signal to the infrared transceiver device is opposite to that of the output signal as follows.

- Infrared data transmit pin (IROUT): Positive logic output
- Infrared data receive pin ($\overline{\text{IRIN}}$): Negative logic input

(c) Register read and write

Do not write to IRIF_SIR0, IRIF_SIR1, or IRIF_SIR2 during data transmission or reception. If this is attempted, correct data communication cannot be guaranteed.

(d) Infrared transmit (light-emit) pulse width select bit

When a 1.8432-MHz clock is input through MSFCLK_IN (the data transmission and reception function operates at 115.2 kbps), do not set the infrared transmit (light-emit) pulse width select bit to 1. If this is attempted, correct transmit pulses might not be output.

(e) Pulse width

Do not set the baud rate count set register for infrared transmit and receive pulse modulation and demodulation (IRIF_SIR2) to H'0000. If this is attempted, the transmit pulse width may be less than the lower limit (pulse duration minimum: 1.41 μs) prescribed in the standard (Infrared Data Association Serial Infrared Physical Layer Specification Version 1.3).

24.4.3 CRC Engine

(1) CRC Engine Configuration

The CRC engine consists of an input data register, a byte counter, a CRC calculation register, and a CRC output register. Figure 24.7 shows the configuration of the CRC engine.

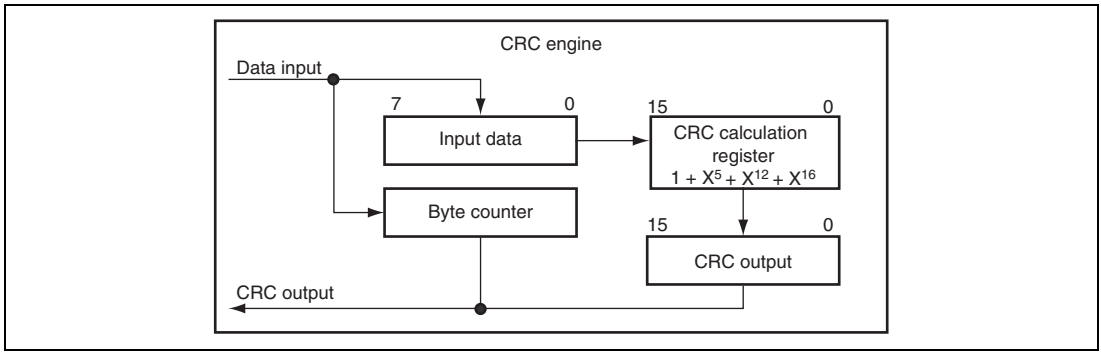


Figure 24.7 CRC Engine Configuration

(2) CRC Engine Operation

Writing 8-bit input data to the CRC engine starts CRC calculation in 8-bit units beginning with the lower bits to output a 16-bit calculation result. Figure 24.8 gives an overview of the CRC calculation.

The CRC generator polynomial is $1 + X^5 + X^{12} + X^{16}$, and the maximum data length is 4096 bytes.

An example of CRC calculation is shown here. After resetting the registers, write H'CC, H'F5, H'F1 and H'A7 as input data in that order. The resultant byte count will be 4 and H'51DF will be output as the CRC calculation result.

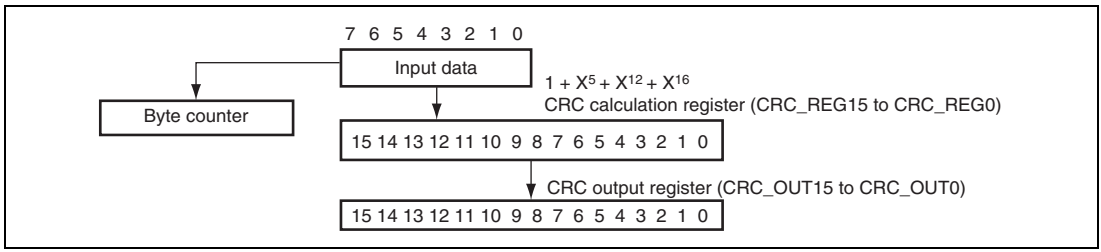


Figure 24.8 CRC Engine Operation

24.4.4 Communication Flow

(1) IrDA Transmission Flow

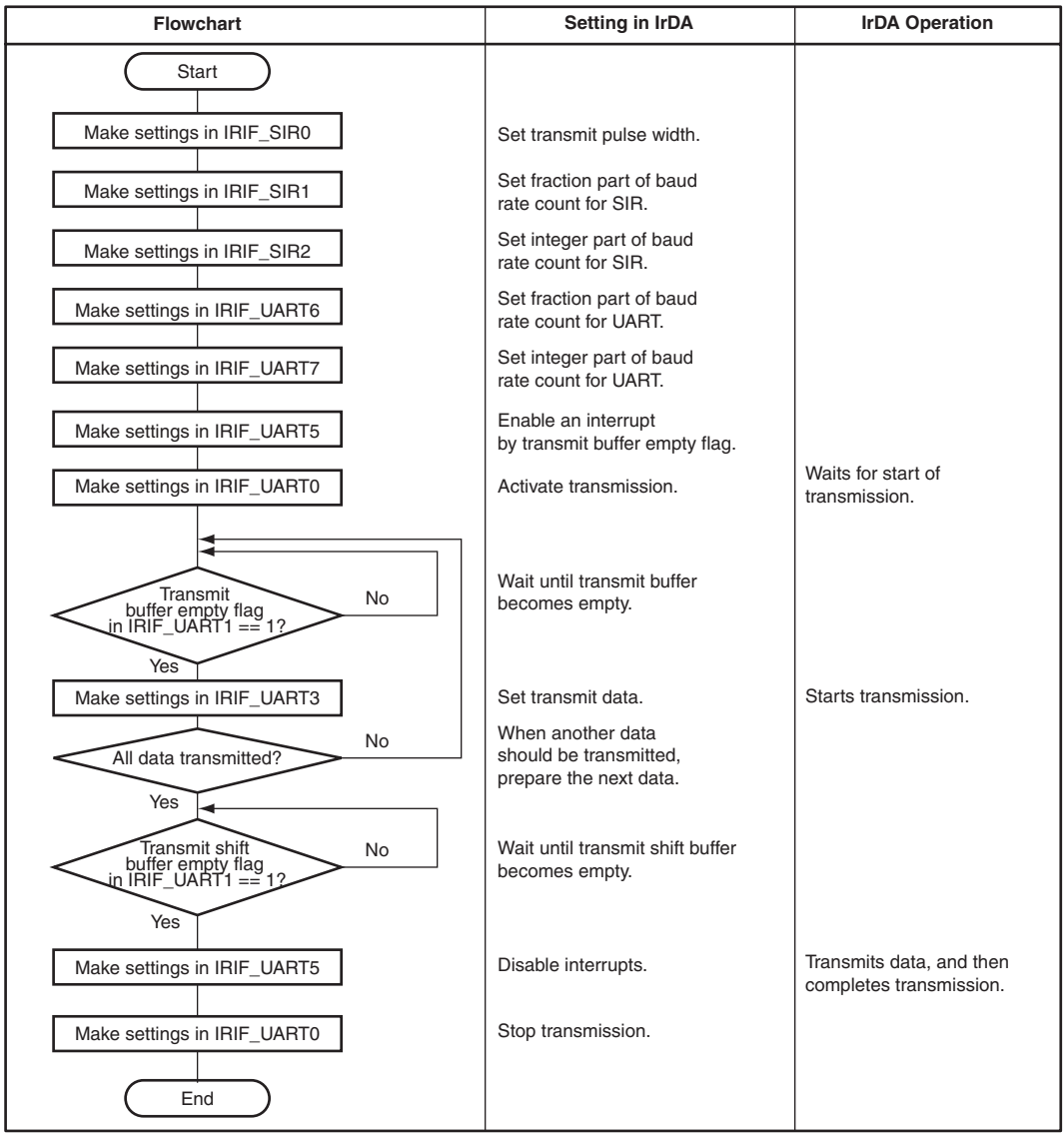
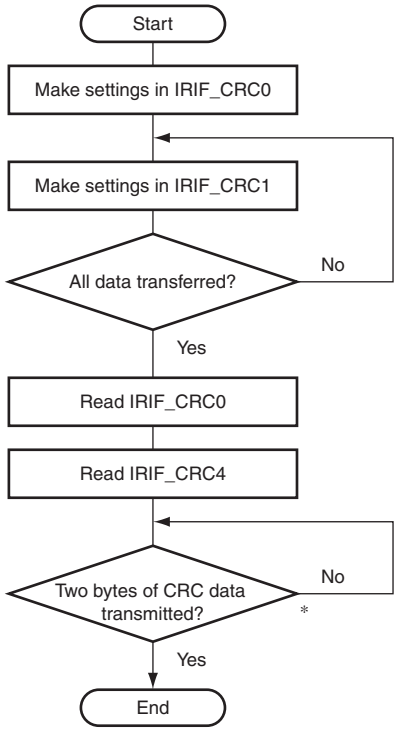


Figure 24.9 IrDA Transmission Flow

(2) IrDA Transmission (CRC Calculation) Flow

Flowchart	Setting in IrDA	IrDA Operation
	Initialize CRC engine. Specify input data for CRC engine. When another CRC data should be input to CRC engine, prepare the next input data. Read value from CRC engine input data byte counter. Read CRC calculation result. When another CRC data should be transmitted, prepare the next CRC transmit data.	

Note: * As shown in the flowchart, after all data has been transferred, the 2-byte calculation result read from IRIF_CRC4 should be transmitted in the order of lower byte to the upper byte.

Figure 24.10 IrDA Transmission (CRC Calculation) Flow

(3) IrDA Reception Flow

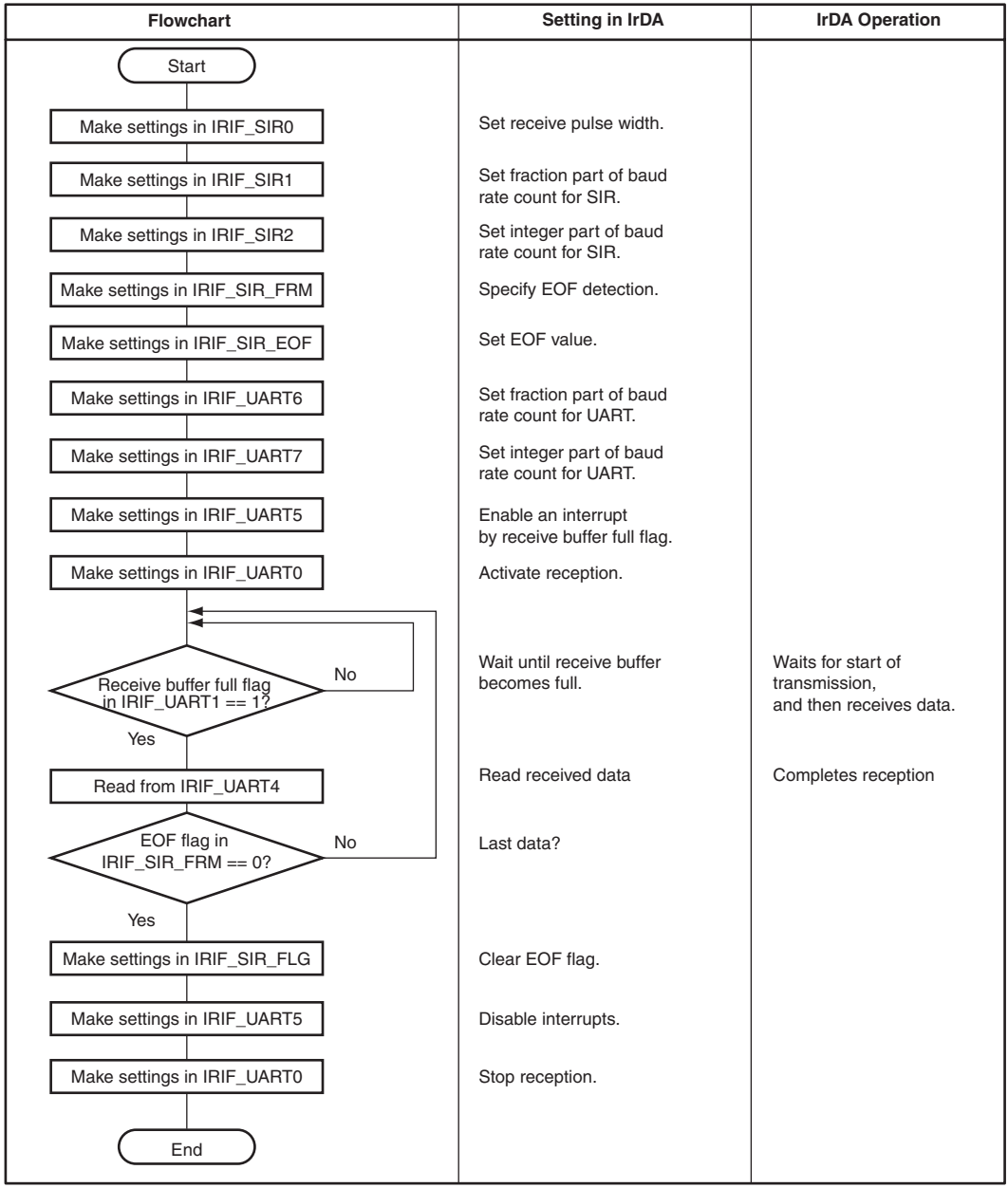


Figure 24.11 IrDA Reception Flow

(4) IrDA Reception (CRC Calculation) Flow

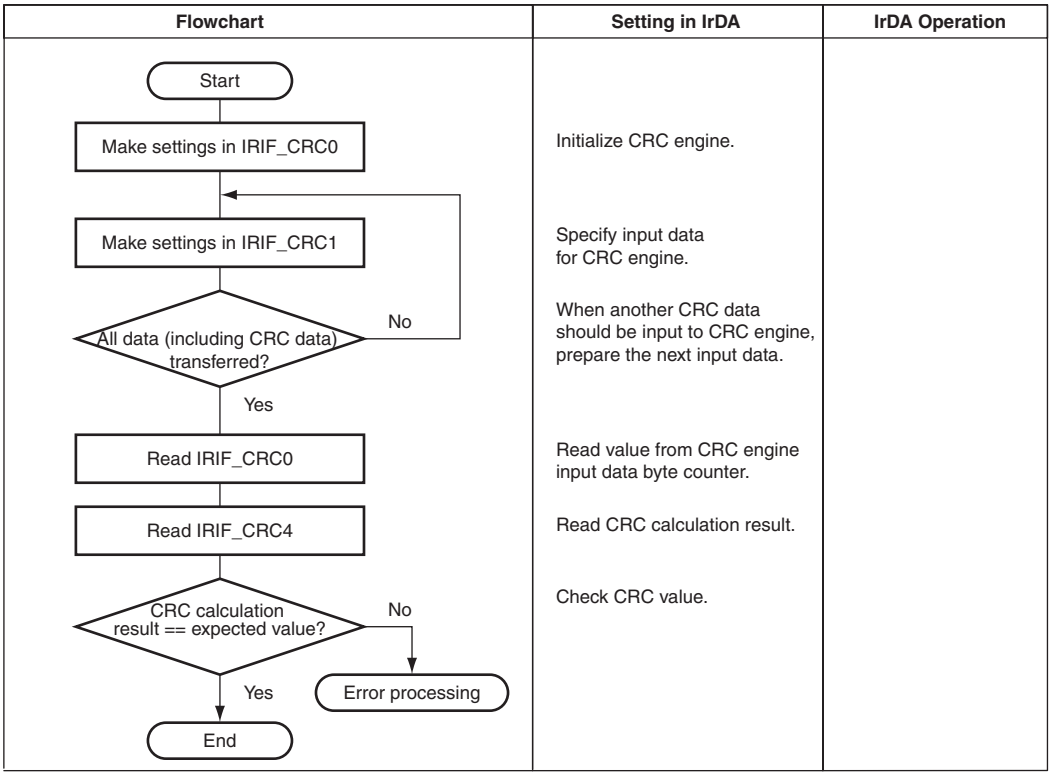


Figure 24.12 IrDA Reception (CRC Calculation) Flow

24.5 Notes on Data Transmission and Reception

(1) Access to Data Receive Buffer

After data reception is completed (the receive buffer becomes full), if read access to the receive buffer register is delayed and then attempted at exactly the same time as when the next data reception is completed, data may be lost without any error interrupt occurring.

Under usual conditions, if the next data reception is completed without reading the previously received data from the receive buffer register, a receive overrun error interrupt should be generated. However, if the previous data is read from the receive buffer register at exactly the same time as when the next data reception is completed, one read operation may be incorrectly recognized as two read operations. In this case, the read value is undefined and a receive overrun error interrupt may not occur.

This problem should be prevented by controlling the operating conditions so that no receive buffer overrun error occurs.

(2) Transmission Jitter

When the IrDA transmit pulse width is set to 1.63 μs and the baud rate is set within a range from 56.6 kbps to 19.2 kbps, the transmission jitter of an IrDA pulse may exceed the upper limit (rate tolerance: ±0.87%) prescribed in the IrDA standard (Infrared Data Association Serial Infrared Physical Layer Specification Version 1.3).

(3) Prohibited Value (H'0001) for IRIF_SIR0 at a Baud Rate of 115 kbps

Do not set IRIF_SIR0 to H'0001 when the baud rate is 115 kbps. If this is attempted, the transmit pulse width may be below the lower limit (pulse duration minimum: 1.41 μs) prescribed in the IrDA standard (Infrared Data Association Serial Infrared Physical Layer Specification Version 1.3).

Section 25 SIM Card Module (SIM)

The smart card interface supports IC cards (smart cards) conforming to the ISO/IEC 7816-3 (Identification Card) specification.

25.1 Features

The smart card interface has the following features.

- General functions
  Asynchronous half-duplex transmission
  Protocol selectable between T = 0 and T = 1 modes
  Data length: 8 bits
  Parity bit generation and check
  Selectable character protection addition time N
  Selectable output clock cycles per etu
  Transmission of error signal (parity error) in receive mode when T = 0
  Detection of error signal and automatic character retransmission in transmit mode when T = 0
  Selectable minimum character interval of 11 etu (N = 255) when T = 1 (etu: Elementary Time Unit)
  Selectable direct convention/inverse convention
  Output clock can be fixed at high or low
- Freely selectable bit rate by on-chip baud rate generator
- Four types of interrupt source
  The four interrupt sources, transmit data empty, receive data full, transmit/receive error, and transmit complete, can be requested separately.
- DMA transfer
  Through DMA transfer requests for transmit data empty and receive data full, the direct memory access controller (DMAC) can be started and used for data transfer.
- The time waiting for the operation when T = 0, and the time waiting for a character when T = 1 can be observed.

Figure 25.1 shows a block diagram of the smart card interface.

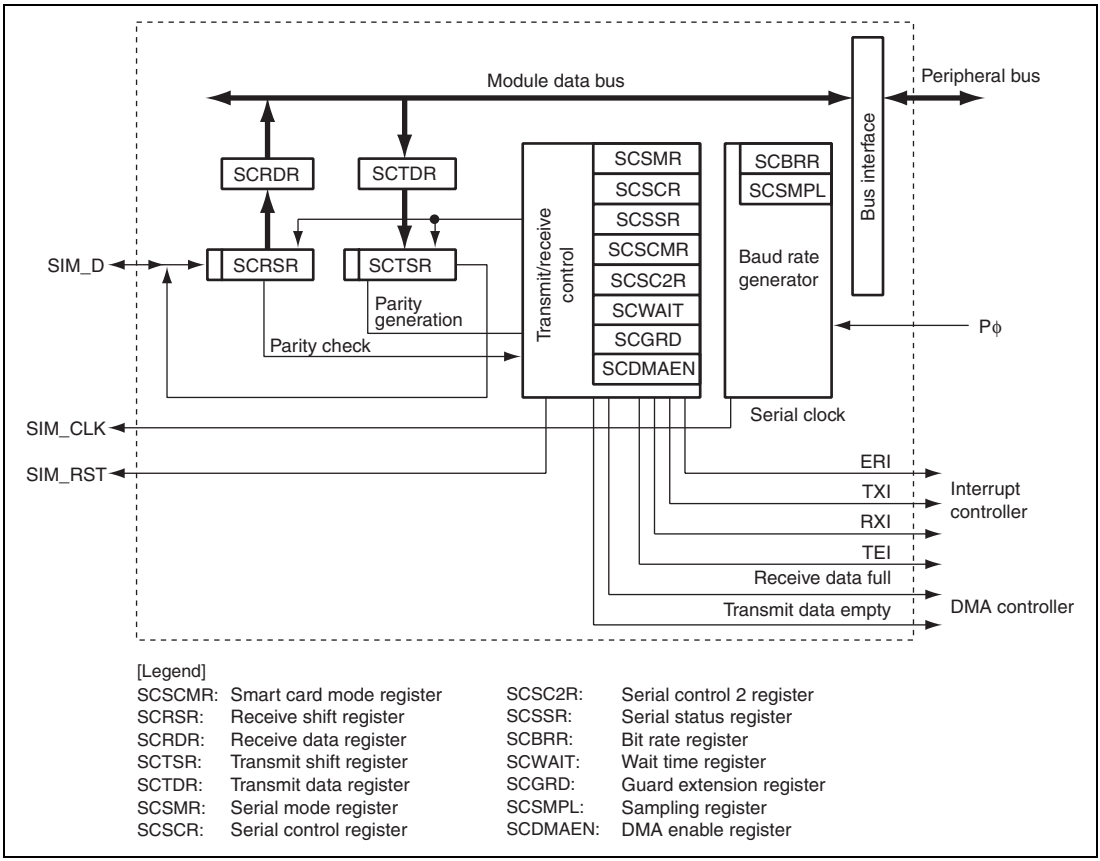


Figure 25.1 Smart Card Interface

25.2 Input/Output Pins

The pin configuration of the smart card interface is shown in table 25.1.

Table 25.1 Pin Configuration

Pin Name	Function	I/O	Description
SIM_D*	Smart card data	I/O	Smart card data input/output
SIM_CLK	Smart card clock	Output	Smart card clock output
SIM_RST	Smart card reset	Output	Smart card reset output

Note: * In explaining transmit and receive operations, the transmit data and receive data sides shall be referred to as TXD and RXD, respectively.

25.3 Register Descriptions

The SIM card module registers are initialized by a reset. Table 25.2 shows the SIM card module register configuration. Table 25.3 shows the register states in each operating mode.

Table 25.2 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
Serial mode register	SCSMR	R/W	H'A449 0000	8
Bit rate register	SCBRR	R/W	H'A449 0002	8
Serial control register	SCSCR	R/W	H'A449 0004	8
Transmit data register	SCTDR	R/W	H'A449 0006	8
Serial status register	SCSSR	R/W	H'A449 0008	8
Receive data register	SCRDR	R	H'A449 000A	8
Smart card mode register	SCSCMR	R/W	H'A449 000C	8
Serial control 2 register	SCSC2R	R/W	H'A449 000E	8
Wait time register	SCWAIT	R/W	H'A449 0010	16
Guard extension register	SCGRD	R/W	H'A449 0012	8
Sampling register	SCSMPL	R/W	H'A449 0014	16
DMA enable register	SCDMAEN	R/W	H'A449 0016	8

Table 25.3 Register States in Each Operating Mode

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
SCSMR	Initialized	Retained	Retained	Retained
SCBRR	Initialized	Retained	Retained	Retained
SCSCR	Initialized	Retained	Retained	Retained
SCTDR	Initialized	Retained	Retained	Retained
SCSSR	Initialized	Retained	Retained	Retained
SCRDR	Initialized	Retained	Retained	Retained
SCSCMR	Initialized	Retained	Retained	Retained
SCSC2R	Initialized	Retained	Retained	Retained
SCWAIT	Initialized	Retained	Retained	Retained
SCGRD	Initialized	Retained	Retained	Retained
SCSMPL	Initialized	Retained	Retained	Retained
SCDMAEN	Initialized	Retained	Retained	Retained

25.3.1 Serial Mode Register (SCSMR)

SCSMR is an 8-bit readable/writable register that selects settings for the communication format of the smart card interface.

Bit:	7	6	5	4	3	2	1	0
	HOEN	LCB	PB	WECC	SDIR	SINV	RST	SMIF
Initial value:	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
4	O/$\overline{E}$	0	R/W	Parity Mode Selects whether even or odd parity is to be used when adding a parity bit and checking parity. 0: Even parity*[1] 1: Odd parity*[2] Notes: 1. When set to even parity, during transmission a parity bit is added such that the sum of 1 bits in the parity bit and transmit characters is even. During reception, a check is performed to ensure that the sum of 1 bits in the parity bit and the receive characters is even. 2. When set to odd parity, during transmission a parity bit is added such that the sum of 1 bits in the parity bit and transmit characters is odd. During reception, a check is performed to ensure that the sum of 1 bits in the parity bit and the receive characters is odd.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

25.3.2 Bit Rate Register (SCBRR)

SCBRR is an 8-bit readable/writable register that sets the serial clock frequency.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	BRR[2:0]		
Initial value:	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	BRR[2:0]	111	R/W	Set the serial clock frequency for transmission/reception.

The SCBRR setting can be determined from the following formula.

$$\text{Serial clock frequency} = \frac{P\phi}{2\,(BRR + 1)}$$

The units of Pϕ (system clock frequency) and serial clock frequency are MHz.

25.3.3 Serial Control Register (SCSCR)

SCSCR is an 8-bit readable/writable register that selects transmit or receive operation, the serial clock output, and whether to enable or disable interrupt requests for the smart card interface.

Bit:	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	WAIT_IE	TEIE	CKE1	CKE0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When serial transmit data is transferred from the transmit data register (SCTDR) to the transmit shift register (SCTSR), and the TDRE flag in the serial status register (SCSSR) is set to 1, transmit data empty interrupt (TXI) requests are enabled/disabled. 0: Disables transmit data empty interrupt (TXI) requests* 1: Enables transmit data empty interrupt (TXI) requests Note: * A TXI can be canceled either by clearing the TDRE flag, or by clearing the TIE bit to 0.
6	RIE	0	R/W	Receive Interrupt Enable When serial receive data is transferred from the receive shift register (SCRSR) to the receive data register (SCRDR), and the RDRF flag in SCSSR is set to 1, receive data full interrupt (RXI) requests, and transmit/receive error interrupt (ERI) requests due to parity errors, overrun errors, and error signal status are enabled/disabled. 0: Disables receive data full interrupt (RXI) requests and transmit/receive error interrupt (ERI) requests*[1]*[2] 1: Enables receive data full interrupt (RXI) requests and transmit/receive error interrupt (ERI) requests*[2] Notes: 1. RXI and ERI interrupt requests can be canceled either by clearing the RDRF, PER, ORER or ERS flag, or by clearing the RIE bit to 0. 2. Wait error interrupt (ERI) requests are enabled or disabled by using the WAIT_IE bit in SCSCR.

Bit	Bit Name	Initial Value	R/W	Description
5	TE	0	R/W	Transmit Enable Enables/disables serial transmit operations. 0: Disables transmission*[1] 1: Enables transmission*[2]*[3] Notes: 1. The TDRE flag in SCSSR is fixed to 1. 2. In this state, if transmit data is written to SCTDR, the transmit operation is initiated. Before setting the TE bit to 1, the serial mode register (SCSMR) and smart card mode register (SCSCMR) must always be set, to determine the transmit format. 3. Even if the TE bit is cleared to 0, the ERS flag is unaffected, and the previous state is retained.
4	RE	0	R/W	Receive Enable Enables/disables serial receive operations. 0: Disables reception*[1] 1: Enables reception*[2] Notes: 1. Clearing the RE bit to 0 has no effect on the RDRF, PER, ORER, or WAIT_ER flag, and the previous state is retained. 2. If the start bit is detected in this state, serial reception is initiated. Before setting the RE bit to 1, SCSMR and SCSCMR must always be set, to determine the receive format.
3	WAIT_IE	0	R/W	Wait Enable Enables/disables wait error interrupt requests. 0: Disables wait error interrupt (ERI) requests 1: Enables wait error interrupt (ERI) requests

Bit	Bit Name	Initial Value	R/W	Description
2	TEIE	0	R/W	Transmit End Interrupt Enable When transmission ends and the TEND flag is set to 1, transmit end interrupt (TEI) requests are enabled/disabled. 0: Disables transmit end interrupt (TEI) requests* 1: Enables transmit end interrupt (TEI) requests* Note: * A TEI can be canceled either by writing transmit data to SCTDR and clearing the TEND bit, or by clearing the TEIE bit to 0 after the TDRE flag in SCSSR is read as 1.
1	CKE1	0	R/W	Clock Enable
0	CKE0	0	R/W	Select the clock source for the smart card interface, and enable/disable clock output from the SIM_CLK pin. 00: Fix the output pin low 01: Clock output as the output pin 10: Fix the output pin high 11: Clock output as the output pin

25.3.4 Transmit Shift Register (SCTSR)

SCTSR is a shift register that transmits serial data.

The smart card interface transfers transmit data from the transmit data register (SCTDR) to SCTSR, and then sends the data in order from the LSB or MSB to the SIM_TXD pin to perform serial data transmission.

When data transmission of one byte has been completed and SCTSR being empty is detected, the transmit data written to SCTDR is automatically transferred to SCTSR, and transmission is initiated. When the TDRE flag in the serial status register (SCSSR) is set to 1, no data is transferred from SCTDR to SCTSR.

25.3.5 Transmit Data Register (SCTDR)

SCTDR is an 8-bit readable/writable register that stores data for serial transmission.

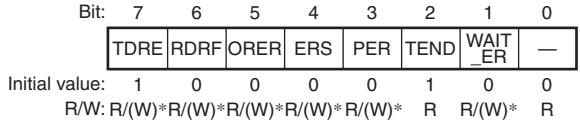
When the smart card interface detects a vacancy in the transmit shift register (SCTSR), transmit data written to SCTDR is transferred to SCTSR, and serial transmission is initiated. During SCTSR serial data transmission, if the next transmit data is written to SCTDR, continuous serial transmission is possible.

Bit:	7	6	5	4	3	2	1	0
	SCTDR[7:0]							
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SCTDR[7:0]	H'FF	R/W	Transmit Data Store data for serial transmission.

25.3.6 Serial Status Register (SCSSR)

SCSSR is an 8-bit readable/writable register that indicates the operating state of the smart card interface.



Bit:	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	ERS	PER	TEND	WAIT_ER	—
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R/(W)*	R

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W*)	Transmit Data Register Empty Indicates that data was transferred from the transmit data register (SCTDR) to the transmit shift register (SCTSR), and that the next serial transmit data can be written to SCTDR. 0: Indicates that valid transmit data is written to SCTDR [Clearing conditions] • When the TE bit in SCSCR is 1, and data is written to SCTDR • When 0 is written to the TDRE bit 1: Indicates that there is no valid transmit data in SCTDR [Setting conditions] • On reset • When the TE bit in SCSCR is 0 • When data is transferred from SCTDR to SCTSR, and data can be written to SCTDR

Bit	Bit Name	Initial Value	R/W	Description
6	RDRF	0	R/(W*)	Receive Data Register Full Indicates that received data is stored in the receive data register (SCRDR). 0: Indicates that no valid received data is stored in SCRDR [Clearing conditions] • On reset • When data is read from SCRDR • When 0 is written to RDRF 1: Indicates that valid received data is stored in SCRDR [Setting condition] • When serial reception is completed normally, and received data is transferred from SCRSR to SCRDR. Note: In T = 0 mode, when a parity error is detected during reception, the SCRDR contents and RDRF flag are unaffected, and the previous state is retained. On the other hand, in T = 1 mode, when a parity error is detected during reception, the received data is transferred to SCRDR, and the RDRF flag is set to 1. In both T = 0 and T = 1 modes, even if the RE bit in the serial control register (SCSCR) is cleared to 0, the SCRDR contents and RDRF flag are unaffected, and the previous state is retained.

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/(W*)	Overrun Error Indicates that an overrun error occurred during reception, resulting in abnormal termination. 0: Indicates that reception is in progress, or that reception was completed normally*1 [Clearing conditions] • On reset • When 0 is written to the ORER bit 1: Indicates that an overrun error occurred during reception*2 [Setting condition] • When the RDRF bit is set to 1 and the next serial reception is completed. Notes: 1. When the RE bit in SCSCR is cleared to 0, the ORER flag is unaffected and the previous state is retained. 2. In SCRDR, the received data before the overrun error occurred is lost, and the data that had been received at the time when the overrun error occurred is retained. Further, with the ORER bit set to 1, subsequent serial reception cannot be continued.
4	ERS	0	R/(W*)	Error Signal Status Indicates the status of error signals returned from the receive side during transmission. In T = 1 mode, this flag is not set. 0: Indicates that an error signal indicating detection of a parity error was not sent from the receive side [Clearing conditions] • On reset • When 0 is written to the ERS bit 1: Indicates that an error signal indicating detection of a parity error was sent from the receive side [Setting condition] • When an error signal is sampled. Note: Even if the TE bit in SCSCR is cleared to 0, the ERS flag is unaffected, and the previous state is retained.

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/(W*)	Parity Error Indicates that a parity error has occurred during reception, resulting in abnormal termination. 0: Indicates that reception is in progress, or that reception was completed normally*[1] [Clearing conditions] • On reset • When 0 is written to the PER bit 1: Indicates that a parity error occurred during reception*[2] [Setting condition] • When the sum of 1 bit in the received data and parity bit does not match the even or odd parity specified by the O/E bit in the serial mode register (SCSMR). Notes: 1. When the RE bit in SCSCR is cleared to 0, the PER flag is unaffected, and the previous state is retained. 2. In T = 0 mode, the data received when a parity error occurs is not transferred to SCRDR, and the RDRF flag is not set. On the other hand, in T = 1 mode, the data received when a parity error occurs is transferred to SCRDR, and the RDRF flag is set. When a parity error occurs, the PER flag should be cleared to 0 before the sampling timing for the next parity bit.

Bit	Bit Name	Initial Value	R/W	Description
2	TEND	1	R	Transmit End Indicates that character transmission in T = 0 mode or block transmission in T = 1 mode is ended. This bit is set to 1 when SCTDR is empty after one byte of serial character and the parity bit have been transmitted. During block transmission in T = 1 mode, writing the next data to SCTDR during transmission of one byte of serial character prevents this bit from being set to 1. The TEND flag is read-only, and cannot be written. 0: Indicates that character transmission in T = 0 mode or block transmission in T = 1 mode is in progress [Clearing condition] • When transmit data is transferred from SCTDR to SCTSR, and character transmission or block transmission is initiated. 1: Indicates that character transmission in T = 0 mode or block transmission in T = 1 mode is ended [Setting conditions] • On reset • When the ERS flag is 0 (normal transmission) after one byte of serial character and a parity bit are transmitted, and also SCTDR is empty Note: The TEND flag is set 1 etu before the end of the character protection time.

Bit	Bit Name	Initial Value	R/W	Description
1	WAIT_ER	0	R/(W*)	Wait Error Indicates the wait timer error status. 0: Indicates that the interval between the start of two successive characters has not exceeded the etu set by SCWAIT. [Clearing conditions] • On reset • When 0 is written to the WAIT_ER flag 1: Indicates that the interval between the start of two successive characters has exceeded the etu set by SCWAIT. [Setting conditions] • In T = 0 mode, when the interval between the start of a character to be received and immediately preceding transmitted or received character exceeds the (value of 60 × SCWAIT: Operation wait time) etu. • In T = 1 mode, when the interval between the start of two successive received characters exceeds the (SCWAIT value: Character protection time) etu. Note: Even if the RE bit in SCSCR is cleared to 0, the WAIT_ER flag is unaffected, and the previous state is retained.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Note: * Writing only 0 is possible to clear the flag.

25.3.7 Receive Shift Register (SCRSR)

SCRSR is a register that receives serial data.

The smart card interface receives serial data input from the SIM_RXD pin in order, from the LSB or MSB, and sets it in SCRSR, converting it to parallel data. When reception of one byte of data is completed, the data is automatically transferred to SCRDR.

25.3.8 Receive Data Register (SCRDR)

SCRDR is an 8-bit read-only register that stores received serial data.

When reception of one byte of serial data is completed, the smart card interface transfers the received serial data from the receive shift register (SCRSR) to SCRDR for storage, and completes the receive operation. Thereafter, SCRSR can receive data. In this way, SCRSR and SCRDR constitute a double buffer, enabling continuous reception of data.

Bit:	7	6	5	4	3	2	1	0
	SCRDR[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SCRDR[7:0]	H'00	R	Receive Data Store received serial data.

25.3.9 Smart Card Mode Register (SCSCMR)

SCSCMR is an 8-bit readable/writable register that selects functions of the smart card interface.

Bit:	7	6	5	4	3	2	1	0
	HOEN	LCB	PB	WECC	SDIR	SINV	RST	SMIF
Initial value:	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	HOEN	0	R/W	High Output Enable Enables or disables temporary output of high level after transmission of one frame of data has finished. 0: Disables the high-output function (initial value) 1: Enables the high-output function

Bit	Bit Name	Initial Value	R/W	Description
6	LCB	0	R/W	Last Character When this bit is set to 1, the character protection time is 2 etu, and the setting of the guard extension register is invalid. 0: Character protection time is determined by the value of the guard extension register 1: Character protection time is 2 etu
5	PB	0	R/W	Protocol Selects the T = 0 or T = 1 protocol. 0: Smart card interface operates according to the T = 0 protocol 1: Smart card interface operates according to the T = 1 protocol
4	WECC	0	R/W	Wait Error Counter Clear Enables or disables clearing of the wait error counter. 0: Wait error counter is not cleared and wait errors are detected (initial value) 1: Wait error counter is cleared and wait errors are not detected
3	SDIR	0	R/W	Smart Card Data Transfer Direction Selects the format for serial/parallel conversion. 0: Transmits the SCTDR contents in LSB-first. Received data is stored in SCRDR as LSB-first. 1: Transmits the SCTDR contents in MSB-first. Received data is stored in SCRDR as MSB-first.
2	SINV	0	R/W	Smart Card Data Inversion Specifies inversion of the data logic level. In combination with the function of bit 3, used for transmission to or reception from the inverse convention card. The SINV bit does not affect the parity bit. 0: Transmits the SCTDR contents without change. Stores received data in SCRDR without change. 1: Inverts the SCTDR contents and transmits it. Inverts received data and stores it in SCRDR.

Bit	Bit Name	Initial Value	R/W	Description
1	RST	0	R/W	Smart Card Reset Controls the output of the SIM_RST pin of the smart card interface. 0: SIM_RST pin of the smart card interface outputs low level 1: SIM_RST pin of the smart card interface outputs high level
0	SMIF	1	R	Smart Card Interface Mode Select This bit is always read as 1. The write value should always be 1.

25.3.10 Serial Control 2 Register (SCSC2R)

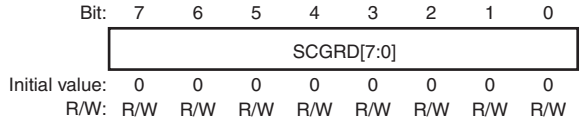
SCSC2R is an 8-bit readable/writable register that enables or disables receive data full interrupt (RXI) requests.

Bit:	7	6	5	4	3	2	1	0
	EIO	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	EIO	0	R/W	Error Interrupt Only When the EIO bit is 1, even if the RIE bit is set to 1, a receive data full interrupt (RXI) request is not sent to the CPU. When the DMAC is used with this setting, the CPU processes only ERI requests. 0: Receive data full interrupt (RXI) requests are determined by the RIE bit setting 1: Receive data full interrupt (RXI) requests are disabled. When the RIE bit is 1, only ERI requests are enabled.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

25.3.11 Guard Extension Register (SCGRD)

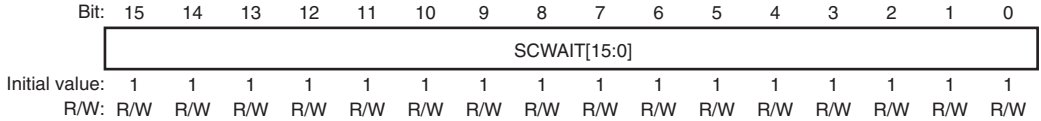
SCGRD is an 8-bit readable/writable register that sets the time added for character protection.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SCGRD[7:0]	H'00	R/W	Guard Extension Set the time added for character protection in smart card mode. The interval between the start of two successive characters is 12 etu (no addition) when the value of this register is H'00, is 13 etu when the value is H'01, and so on, up to 266 etu for H'FE. If the value of this register is H'FF, the interval between the start of two successive characters is 11 etu in T = 1 mode and is 12 etu in T = 0 mode.

25.3.12 Wait Time Register (SCWAIT)

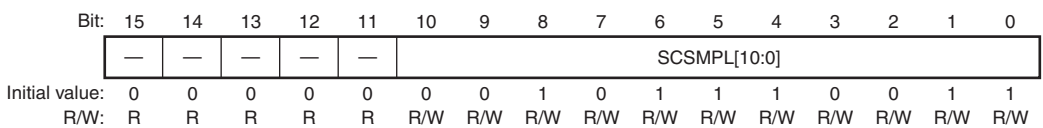
SCWAIT is a 16-bit readable/writable register. If the interval between the start of two successive characters exceeds the set value (in etu units), a wait time error is generated.



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	SCWAIT[15:0]	H'FFFF	R/W	Wait Time Register In T = 0 mode, the operation wait time can be set in this register. If the interval between the start of characters to be received and transmitted or received characters immediately before exceeds the (60 × the value set in this register) etu, the WAIT_ER flag is set to 1. However, if SCWAIT is set to H'0000, the WAIT_ER flag is set after 60 etu. In T = 1 mode, the character wait time can be set in this register. If the interval between the start of two successive received characters exceeds the (the value set in this register) etu, the WAIT_ER flag is set to 1. However, if SCWAIT is set to H'0000, the WAIT_ER flag is set after 1 etu.

25.3.13 Sampling Register (SCSMPL)

SCSMPL is a 16-bit readable/writable register that sets the number of serial clock cycles per etu.



Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SCSMPL[10:0]	H'173	R/W	Setting for Number of Serial Clock Cycles per Etu The number of serial clock cycles per etu is (SCSMPL value + 1). The value written to SCSMPL should always be H'0007 or greater.

25.3.14 DMA Enable Register (SCDMAEN)

SCDMAEN enables or disables DMA transfer.

Bit:	7	6	5	4	3	2	1	0
	RDMAE	TDMAE	—	—	—	—	—	—
Initial value:	1	1	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	RDMAE	1	R/W	Reception DMA Enable Flag Enables or disables DMA transfer at reception. 0: Disables DMA transfer during reception 1: Enables DMA transfer during reception
6	TDMAE	1	R/W	Transmission DMA Enable Flag Enables or disables DMA transfer at transmission. 0: Disables DMA transfer during transmission 1: Enables DMA transfer during transmission
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

25.4 Operation

25.4.1 Overview

The main functions of the smart card interface are as follows.

- One frame consists of the start bit, 8-bit data, and the parity bit.
- During transmission, a character protection time, set using SCGRD and the LCB and PB bits in SCSCMR, is inserted between the end of each parity bit and the beginning of the next frame.
- During reception in T = 0 mode, when a parity error is detected, low level is output for a duration of 1 etu as an error signal, 10.5 etu after the start bit.
- During transmission in T = 0 mode, if an error signal is sampled, after 2 etu or more have elapsed, the same data is automatically transmitted.
- Only asynchronous communication functions are supported; there is no clocked synchronous communication function.

25.4.2 Data Format

Figure 25.2 shows the data format used by the smart card interface. The smart card interface performs a parity check for each frame during reception.

During reception in T = 0 mode, if a parity error is detected, an error signal is returned to the transmit side, requesting data retransmission. When the transmit side samples the error signal, it retransmits the same data.

During reception in T = 1 mode, if a parity error is detected, an error signal is not returned. During transmission, error signals are not sampled and data is not retransmitted.

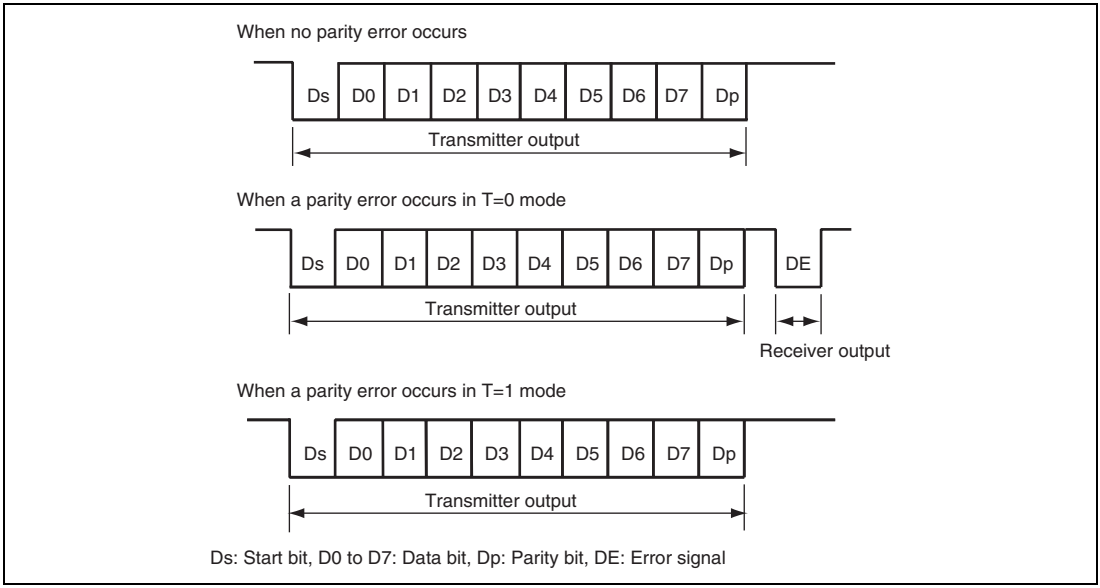


Figure 25.2 Data Format Used by Smart Card Interface

The operation sequence is as follows.

1. When not in use, the data line is in a high-impedance state and fixed at high level by a pull-up resistance.
2. The transmit side initiates transmission of one frame of data. The data frame begins with the start bit (Ds: low level). This is followed by eight data bits (D0 to D7) and the parity bit (Dp).
3. The smart card interface then returns the data line to high impedance. The data line is held at high level by the pull-up resistance.
4. The receive side performs a parity check.

   If there is no parity error and reception is normal, reception of the next frame is awaited, without further action.

   On the other hand, when a parity error has occurred in T = 0 mode, an error signal (DE: low level) is output, requesting data retransmission. After output of an error signal with the specified duration, the receive side again sets the signal line to the high-impedance state. The signal line returns to high level by means of the pull-up resistance. If in T = 1 mode, however, no error signal is output even if a parity error occurs.
5. If the transmit side does not receive an error signal, the next frame is transmitted.

   On the other hand, if in T = 0 mode and an error signal is received, the data for which the error occurred is retransmitted as in step 2 above. In T = 1 mode, however, error signals are not received and retransmission is not performed.

25.4.3 Register Settings

Table 25.4 shows a map of the bits in the registers used by the smart card interface.

Bits for which 0 or 1 is shown must always be set to the value shown. The method for setting the bits other than these is explained below.

Table 25.4 Register Settings for Smart Card Interface

	Bit							
Register	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
SCSMR	0	0	1	O/$\overline{E}$	0	0	0	0
SCBRR	0	0	0	0	0	BRR2	BRR1	BRR0
SCSCR	TIE	RIE	TE	RE	WAIT_IE	TEIE	CKE1	CKE0
SCTDR	SCTDR7	SCTDR6	SCTDR5	SCTDR4	SCTDR3	SCTDR2	SCTDR1	SCTDR0
SCSSR	TDRE	RDRF	ORER	ERS	PER	TEND	WAIT_ER	0
SCRDR	SCRDR7	SCRDR6	SCRDR5	SCRDR4	SCRDR3	SCRDR2	SCRDR1	SCRDR0
SCSCMR	HOEN	LCB	PB	WECC	SDIR	SINV	RST	SMIF
SCSC2R	EIO	0	0	0	0	0	0	0
SCWAIT	SCWAIT15 to SCWAIT0							
SCGRD	SCGRD7 to SCGRD0							
SCSMPL	SCSMPL10 to SCSMPL0, bits 15 to 11 are 0							
SCDMAEN	RDMAE	TDMAE	0	0	0	0	0	0

(1) Serial Mode Register (SCSMR) Setting

When the IC card is set for the direct convention, the O/$\overline{E}$ bit is cleared to 0; for the inverse convention, it is set to 1.

(2) Bit Rate Register (SCBRR) Setting

Sets the bit rate. For the method of computing settings, refer to section 25.4.4, Clocks.

(3) Serial Control Register (SCSCR) Settings

Each interrupt can be enabled and disabled using the TIE, RIE, TEIE, and WAIT_IE bits.

By setting either the TE or RE bit to 1, transmission or reception is selected.

The CKE1 and CKE0 bits are used to select the clock output state. For details, refer to section 25.4.4, Clocks.

(4) Smart Card Mode Register (SCSCMR) Settings

When the IC card is set for the direct convention, both the SDIR and SINV bits are cleared to 0; for the inverse convention, both are set to 1.

Figure 25.3 below shows the register settings and waveform examples at the start character for two types of IC cards (a direct-convention type and an inverse-convention type).

For the direct-convention type, the logical level 1 is assigned to the Z state, and the logical level 0 to the A state, and transmission and reception are performed in LSB-first. The data of the above start character is then H'3B. Even parity is used according to the smart card specification, and so the parity bit is 1.

For the inverse-convention type, the logical level 1 is assigned to the A state, and the logical level 0 to the Z state, and transmission and reception are performed in MSB-first. The data of the start character shown in figure 25.3 is then H'3F. Even parity is used according to the smart card specification, and so the parity bit is 0 corresponding to the Z state.

In addition, only the D7 to D0 bits are inverted by the SINV bit. The O/$\overline{E}$ bit in SCSMR is set to odd parity mode to invert the parity bit. In transmission and reception, the setting condition is similar.

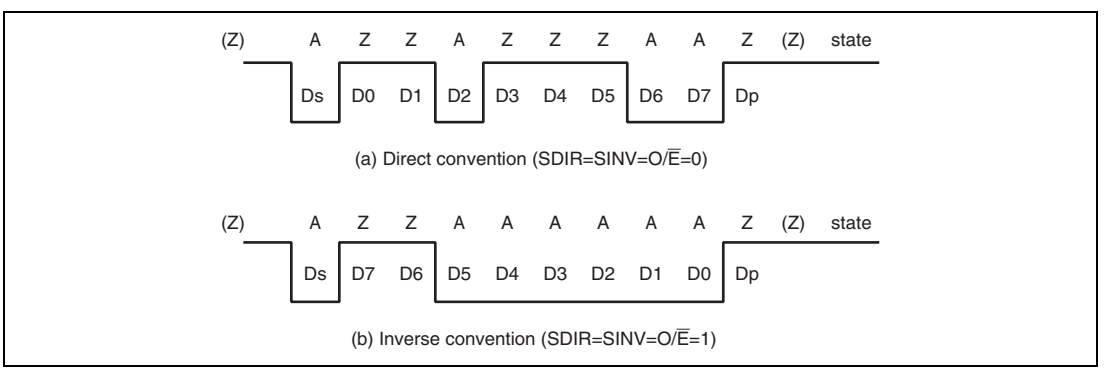


Figure 25.3 Examples of Start Character Waveforms

25.4.4 Clocks

Only the internal clock generated by the on-chip baud rate generator can be used as the transmit/receive clock in the smart card interface. The bit rate is set using the bit rate register (SCBRR) and the sampling register (SCSMPL), using the formula indicated below. Examples of bit rates are listed in table 25.5.

Here, when the CKE0 bit is set to 1 and the clock output is selected, a clock signal is output from the SIM_CLK pin with frequency equal to (SCSMPL + 1) times the bit rate.

$$B = P\phi \times 10^6/\{(S+1) \times 2\ (N+1)\}$$

where

B = Bit rate (bits/s)

Pϕ = Operating frequency of the peripheral module

S = SCSMPL setting ($0 \leq S \leq 2047$)

N = SCBRR setting ($0 \leq N \leq 7$).

Table 25.5 Example of Bit Rates (bits/s) for SCBRR Settings (Pϕ = 19.8 MHz, SCSMPL = 371)

SCBRR Setting	SCK Frequency (MHz)	Bit Rate (bits/s)
7	1.2375	3327
6	1.414	3802
5	1.65	4435
4	1.98	5323
3	2.475	6653
2	3.3	8871
1	4.95	13306
0	9.9	26613

Note: The bit rate is a value that is rounded off below the decimal point.

25.4.5 Data Transmit/Receive Operation

(1) Initialization

Prior to data transmission and reception, the following procedure should be used to initialize the smart card interface. Initialization is also necessary when switching from transmit mode to receive mode, and when switching from receive mode to transmit mode. An example of the initialization process is shown in the flowchart of figure 25.4.

(a) Clear the TE and RE bits in the serial control register (SCSCR) to 0.

(b) Clear the error flags PER, ORER, ERS, and WAIT_ER in the serial status register (SCSSR) to 0.

(c) Set the parity bit (O/$\overline{E}$ bit) in the serial mode register (SCSMR).

(d) Set the LCB, PB, SMIF, SDIR, and SINV bits in the smart card mode register (SCSCMR).

(e) Set the value corresponding to the bit rate to the bit rate register (SCBRR). Also, set the value corresponding to the peripheral module operating frequency per 1 etu to the sampling register (SCSMPL).

(f) Set the value corresponding to the character protection time to the guard extension register (SCGRD). Also, set the value corresponding to the operation work time in T = 0 mode and the value corresponding to the character wait time in T = 1 mode to the wait time register (SCWAIT).

(g) Set the clock source select bits (CKE1 and CKE0 bits) in the serial control register (SCSCR). At this time, the TIE, RIE, TE, RE, TEIE, and WAIT_IE bits should be cleared to 0.

If the CKE0 bit is set to 1, a clock signal is output from the SIM_CLK pin.

(h) After waiting at least 1 etu, set the WECC bit in SCSCMR, and the TIE, RIE, TE, RE, TEIE, and WAIT_IE bits in SCSCR. Except for self-check, the TE bit and RE bit should not be set simultaneously.

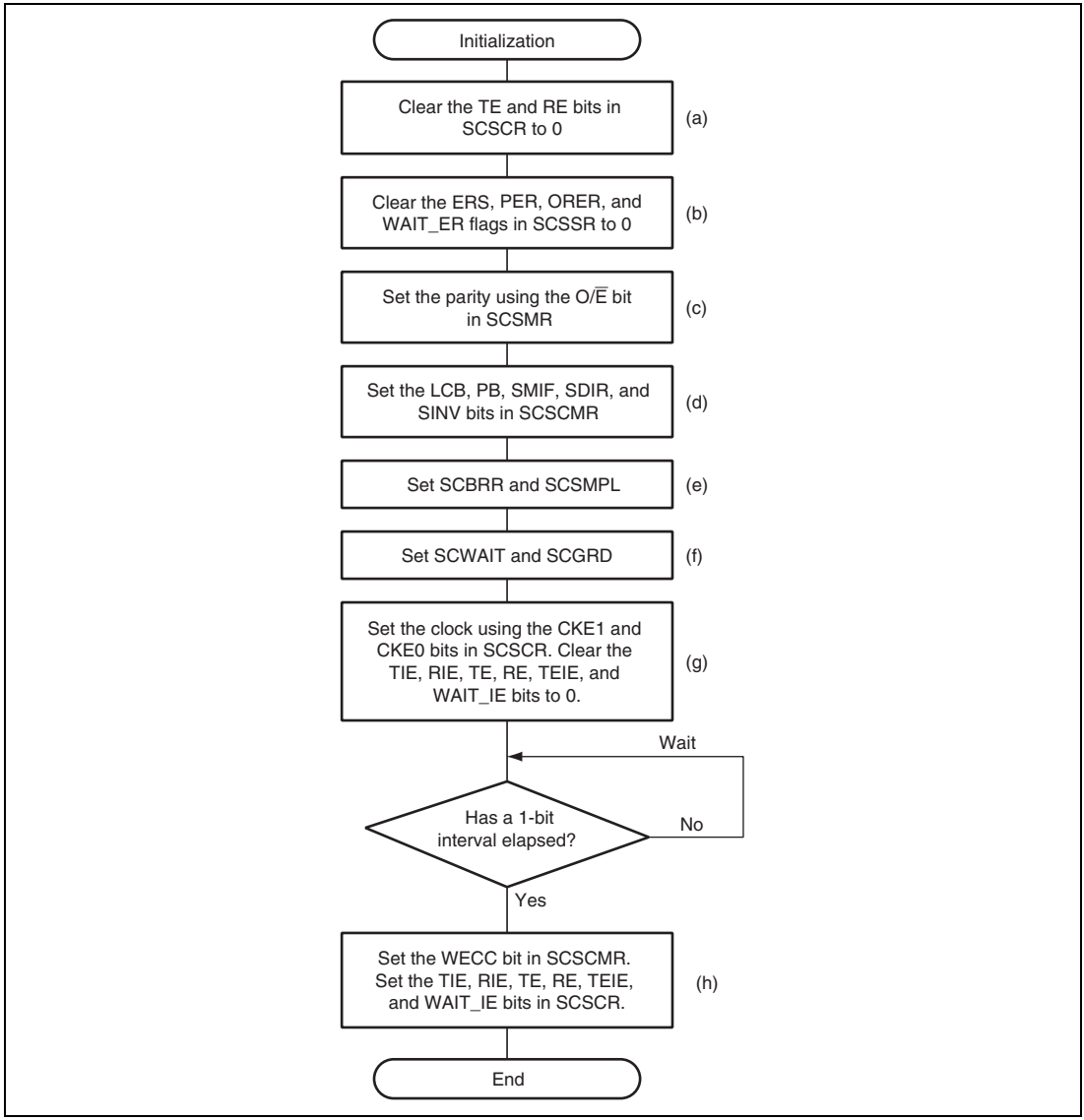


Figure 25.4 Example of Initialization Flow

(2) Serial Data Transmission

Data transmission in smart card mode includes error signal sampling and retransmit processing. An example of transmit processing is shown in figure 25.5.

(a) Follow the Initialization procedure described in section 25.4.5, Data Transmit/Receive Operation to initialize the smart card interface.
(b) Confirm that the ERS bit (error flag) in SCSSR is cleared to 0.
(c) Repeat (b) and (c) until it can be confirmed that the TDRE flag in SCSSR is set to 1.
(d) Write transmit data to SCTDR, and perform transmission. At this time, the TDRE flag is automatically cleared to 0. When transmission of the start bit is started, the TEND flag is automatically cleared to 0, and the TDRE flag is automatically set to 1.
(e) When performing continuous data transmission, return to (b).
(f) Set the WECC bit in SCSCMR as required. When transmission is ended, clear the TE bit to 0.

Interrupt processing can be performed in the above series of processing.

When the TIE bit is set to 1 to enable interrupt requests and if transmission is started and the TDRE flag is set to 1, a transmit data empty interrupt (TXI) request is issued. When the RIE bit is set to 1 to enable interrupt requests and if an error occurs during transmission and the ERS flag is set to 1, a transmit/receive error interrupt (ERI) request is issued.

For details, refer to Interrupt Operations in section 25.4.5, Data Transmit/Receive Operation.

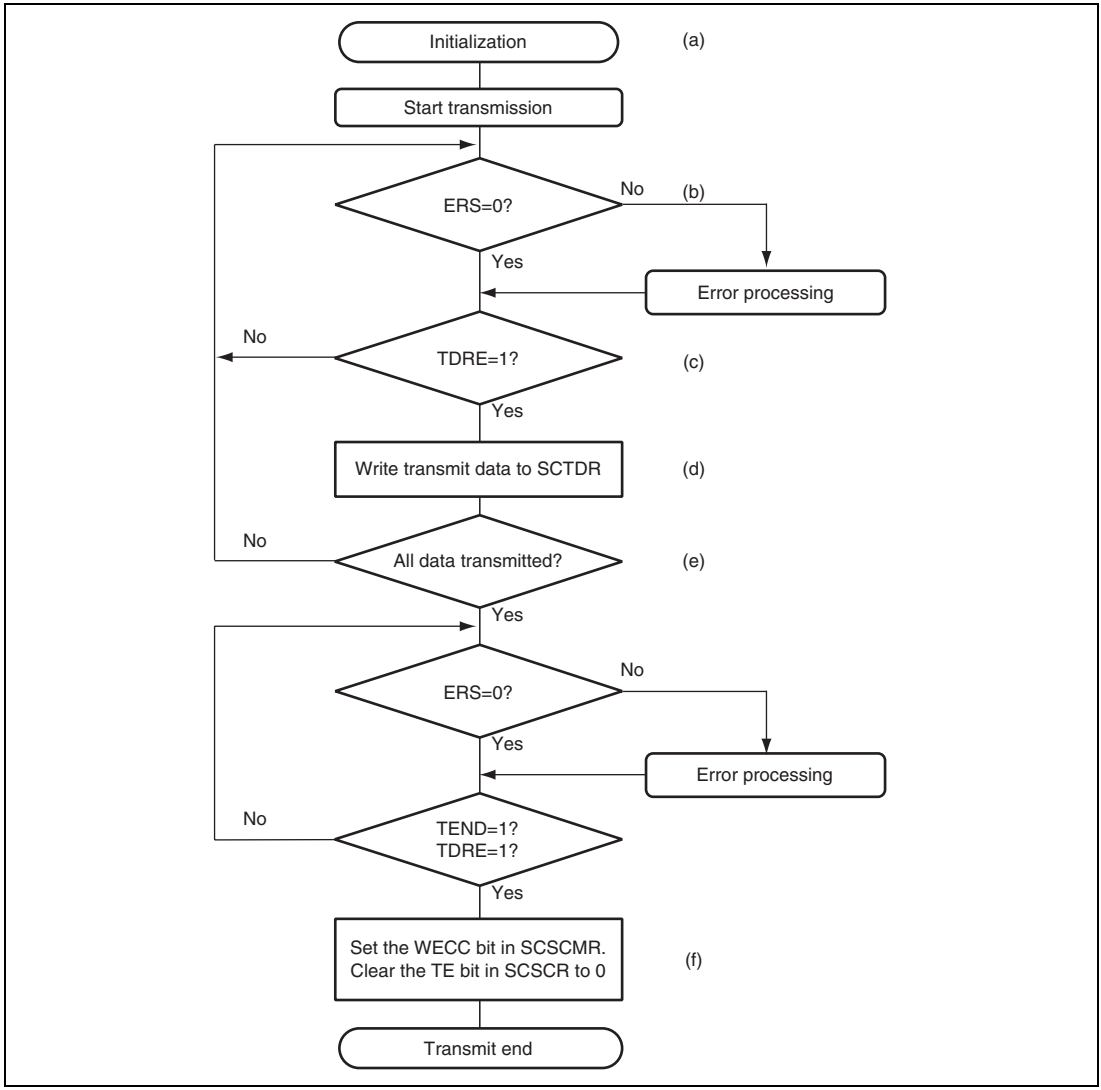


Figure 25.5 Example of Transmit Processing

(3) Serial Data Reception

An example of data receive processing in smart card mode is shown in figure 25.6.

(a) Follow the Initialization procedure described in section 25.4.5, Data Transmit/Receive Operation to initialize the smart card interface.
(b) Confirm that the PER, ORER, and WAIT_ER flags in SCSSR are 0. If one of these flags is set, after performing the prescribed receive error processing, clear the PER, ORER, and WAIT_ER flags to 0.
(c) Repeat (b) and (c) in the figure until it can be confirmed that the RDRF flag is set to 1.
(d) Read received data from SCRDR.
(e) When receiving data continuously, return to (b).
(f) Set the WECC bit in SCSCMR as required. When reception is ended, clear the RE bit to 0.

Interrupt processing can be performed in the above series of processing.

When the RIE bit is set to 1 and the EIO bit is cleared to 0 and if the RDRF flag is set to 1, a receive data full interrupt (RXI) request is issued. If the RIE bit is set to 1, an error occurs during reception, and either the ORER, PER, or WAIT_ER flag is set to 1, a transmit/receive error interrupt (ERI) request is issued.

For details, refer to Interrupt Operations in section 25.4.5, Data Transmit/Receive Operation.

If a parity error occurs during reception and the PER flag is set to 1, in T = 0 mode the received data is not transferred to SCRDR, and so this data cannot be read. In T = 1 mode, received data is transferred to SCRDR, and so this data can be read.

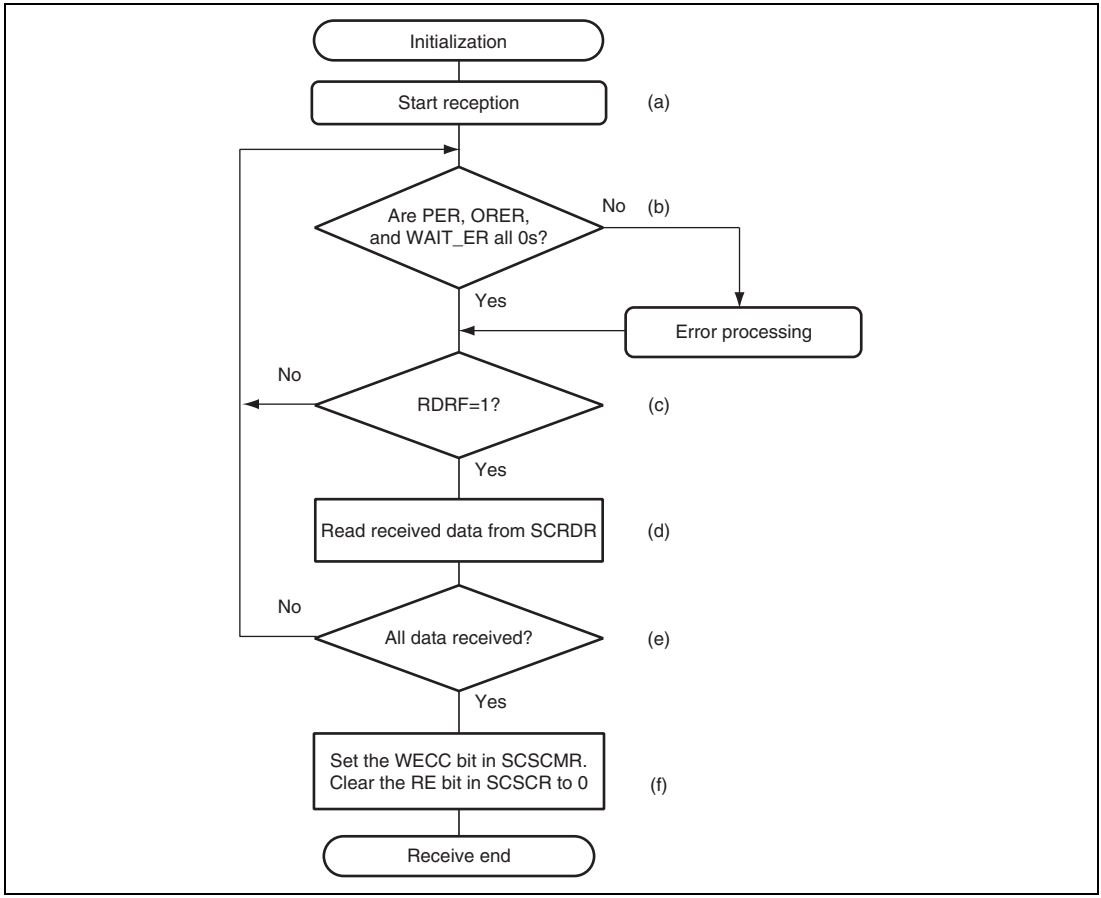


Figure 25.6 Example of Receive Processing

(4) Switching Modes

When switching from receive mode to transmit mode, after confirming that reception has been completed, start initialization, and then clear the RE bit to 0 and set the TE bit to 1. Completion of reception can be confirmed through the RDRF flag.

When switching from transmit mode to receive mode, after confirming that transmission has been completed, start initialization, and then clear the TE bit to 0 and set the RE bit to 1. Completion of transmission can be confirmed through the TDRE and TEND flags.

(5) Interrupt Operations

The smart card interface has four types of interrupt requests: transmit data empty interrupt (TXI) requests, transmit/receive error interrupt (ERI) requests, receive data full interrupt (RXI) requests, and transmit end interrupt (TEI) requests.

- When the TDRE flag in SCSSR is set to 1, a TXI request is issued.
- When the RDRF flag in SCSSR is set to 1, an RXI request is issued.
- When the ERS, ORER, PER, or WAIT_ER flag in SCSSR is set to 1, an ERI request is issued.
- When the TEND flag in SCSSR is set, a TEI request is issued.

Table 25.6 lists the interrupt sources for the smart card interface. Each of the interrupt requests can be enabled or disabled using the TIE, RIE, TEIE, and WAIT_IE bits in SCSCR and the EIO bit in SCSC2R. In addition, each interrupt request can be sent independently to the interrupt controller.

Table 25.6 Interrupt Sources of Smart Card Interface

Operating State		Flags	Mask Bits	Interrupt Sources
Transmit mode	Normal operation	TDRE	TIE	TXI
		TEND	TEIE	TEI
	Error	ERS	RIE	ERI
Receive mode	Normal operation	RDRF	RIE, EIO	RXI
	Error	ORER, PER	RIE	ERI
		WAIT_ER	WAIT_IE	ERI

(6) Data Transfer Using DMAC

The smart card interface enables reception and transmission using the DMAC. When the DMAC is used, set the RDMAE and TDMAE bits in SCDMAEN to 1.

In transmission, when the TDRE flag in SCSSR is set to 1 while the TDMAE bit is 1, a DMA transfer request for transmit data empty is issued. If a DMA transfer request for transmit data empty is set in advance as a DMAC activation source, the DMAC can be activated and made to transfer data when a DMA transfer request for transmit data empty occurs.

When in T = 0 mode and if an error signal is received during transmission, the same data is automatically retransmitted. At the time of this retransmission, no DMA transfer request is issued, and so the number of bytes specified to the DMAC can be transmitted.

When using the DMAC for transmit data processing and performing error processing as a result of an interrupt request sent to the CPU, the TIE bit should be cleared to 0 so that no TXI requests are generated, and the RIE bit should be set to 1 so that an ERI request is issued. The ERS flag set when an error signal is received is not cleared automatically, and so should be cleared by sending an interrupt request to the CPU.

In reception, when the RDRF flag in SCSSR is set to 1 while the RDMAE bit is 1, a DMA transfer request for receive data full is issued. By setting a DMA transfer request for receive data full in advance as a DMAC activation source, the DMAC can be activated and made to transfer data when a DMA transfer request for receive data full occurs.

When in T = 0 mode and if a parity error occurs during reception, a data retransmit request is issued. At this time the RDRF flag is not set, and a DMA transfer request is not issued, so the number of bytes specified to the DMAC can be received.

When using the DMAC for receive data processing and performing error processing as a result of an interrupt request sent to the CPU, the RIE bit should be set to 1, the EIO bit to 1, and the WAIT_ER bit to 1, so that no RXI requests are generated and only ERI requests are generated.

The PER, ORER, and WAIT_ER flags that are set by a receive error are not automatically cleared, and so should be cleared by sending an interrupt request to the CPU.

When using the DMAC for transmission and reception, the DMAC should always be set first and put into the enabled state, before setting the smart card interface.

25.5 Usage Notes

The following matters should be noted when using the smart card interface.

(1) Receive Data Timing and Receive Margin

When SCSMPL holds its initial value, the smart card interface operates at a basic clock frequency 372 times the transfer rate.

During reception, the smart card interface samples the falling edge of the start bit using the serial clock for internal synchronization. Receive data is captured internally at the rising edge of the 186th serial clock pulse. This is shown in figure 25.7.

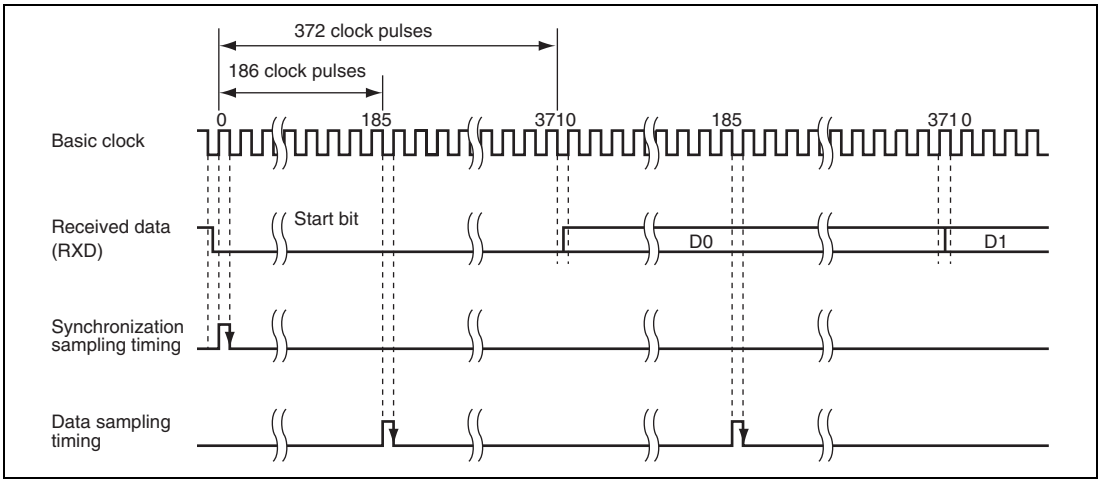


Figure 25.7 Receive Data Sampling Timing in Smart Card Mode

Hence the receive margin can be expressed as follows.

Formula for receive margin in smart card mode:

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

where

M: Receive margin (%)

N: Ratio of the bit rate to the clock (N = 372)

D: Clock duty (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of the deviation of the clock frequency

In the above formula, if F = 0 and D = 0.5, then the receive margin is as follows.

When D = 0.5, F = 0,

$M = (0.5 - 1/2 \times 372) \times 100\%$

$= 49.866\%$.

(2) Retransmit Operation

Retransmit operations when the smart card interface is in receive mode and in transmit mode are described below.

- Retransmission when the smart card interface is in receive mode (T = 0)

  Figure 25.8 shows retransmit operations when the smart card interface is in receive mode.

  (a) If an error is detected as a result of checking the received parity bit, the PER bit in SCSSR is automatically set to 1. At this time, if the RIE bit in SCSCR is set to enable, an ERI request is issued. The PER bit in SCSSR should be cleared to 0 before the sampling timing for the next parity bit.

  (b) The RDRF bit in SCSSR is not set for frames in which a parity error occurs.

  (c) If no error is detected as a result of checking the received parity bit, the PER bit in SCSSR is not set.

  (d) If no error is detected as a result of checking the received parity bit, it is assumed that reception was completed normally, and the RDRF bit in SCSSR is automatically set to 1. If the RIE bit in SCSCR is 1 and the EIO bit is 0, an RXI request is generated.

  (e) If a normal frame is received, the pin retains its high-impedance state at the timing for transmission of error signals.

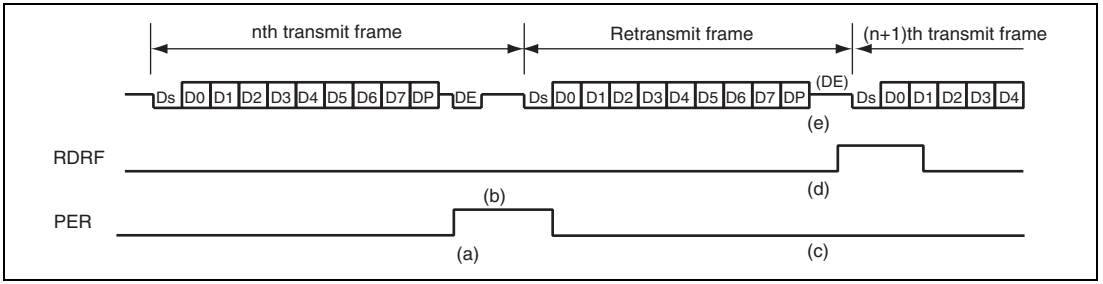


Figure 25.8 Retransmission when Smart Card Interface is in Receive Mode

- Retransmission when the smart card interface is in transmit mode (T = 0)

  Figure 25.9 shows retransmit operations when the smart card interface is in transmit mode.

  (a) After completion of transmission of one frame, if an error signal is returned from the receive side, the ERS bit in SCSSR is set to 1. If the RIE bit in SCSCR is set to enable, an ERI request is issued. The ERS bit in SCSSR should be cleared to 0 before the sampling timing for the next parity bit.

  (b) In T = 0 mode, the TEND bit in SCSSR is not set for a frame when an error signal indicating an error is received.

  (c) If no error signal is returned from the receive side, the ERS bit in SCSSR is not set.

  (d) If no error signal is returned from the receive side, it is assumed that transmission of one frame, including retransmission, is completed. If SCTDR is not empty at this time, the TEND bit in SCSSR is set to 1. At this time, if the TEIE bit in SCSCR is set to enable, a TEI interrupt request is issued.

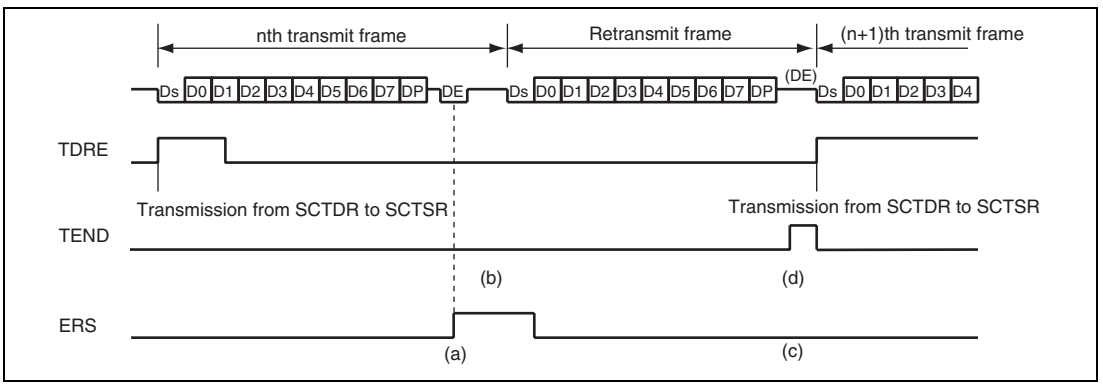


Figure 25.9 Retransmit Standby Mode (Clock Stopped) when Smart Card Interface is in Transmit Mode

(3) Temporary High-Output Function

Setting the HOEN bit in SCSCMR to 1 enables the smart card interface to use the temporary high-output function after transmitting one frame of data. If the temporary high-output function is disabled, when the data line is driven to high impedance by negating the tristate buffer after transmission of one frame, pulling up the data line to high level takes time. When the temporary high-output function is enabled, high-level is forcibly output for one system clock cycle before negating the tristate buffer after transmission of one frame, thus reducing the time to fix the data line to high level. A timechart is shown in figure 25.10.

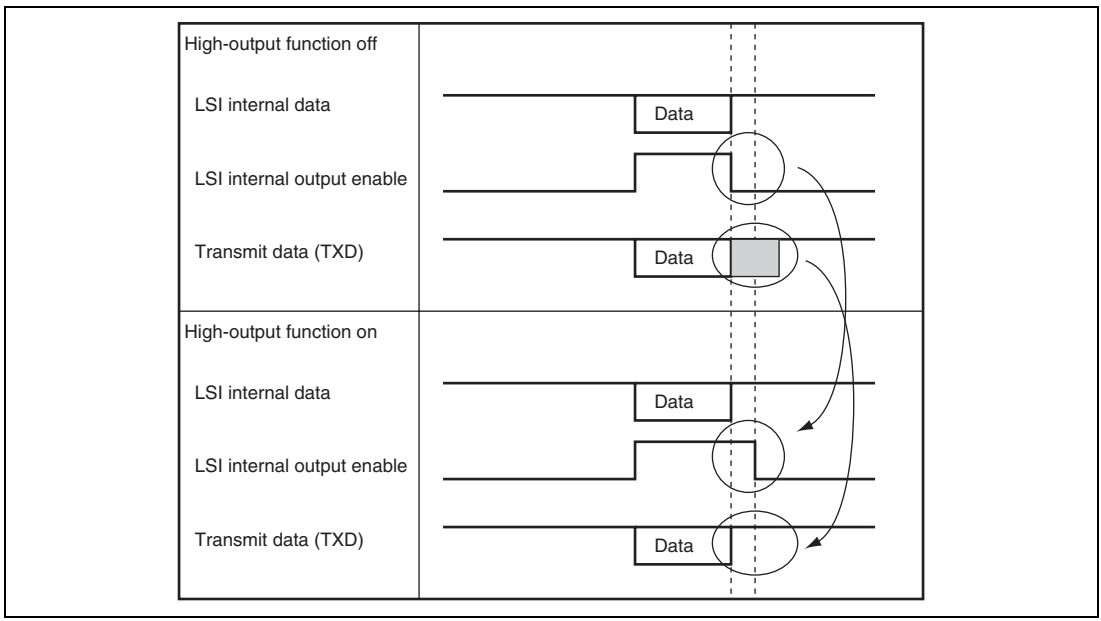


Figure 25.10 High-Output Function Timechart

(4) Standby Mode (Clock Stop)

When switching between smart card interface mode and standby mode, in order to retain the clock duty, the following switching procedure should be used. The switching procedure is shown in figure 25.11.

- When switching from smart card interface mode to standby mode
  - (a) Write 0 to the TE and RE bits in the serial control register (SCSCR), to stop transmit and receive operations. At the same time, set the CKE1 bit to the value for the output-fixed state in standby mode.
  - (b) Write 0 to the CKE0 bit in SCSCR to stop the clock.
  - (c) Wait for one cycle of the serial clock. During this interval, the duty is retained, and the clock output is fixed at the specified level.
  - (d) Make the transition to standby mode.
- To return from standby mode to smart card interface mode
  - (e) Cancel the standby state.
  - (f) Set the CKE1 bit in the serial control register (SCSCR) to the value of the output-fixed state at the beginning of standby (the current SIM_CLK pin state).
  - (g) Write 1 to the CKE0 bit in SCSCR to output a clock signal. Clock signal generation begins at normal duty.

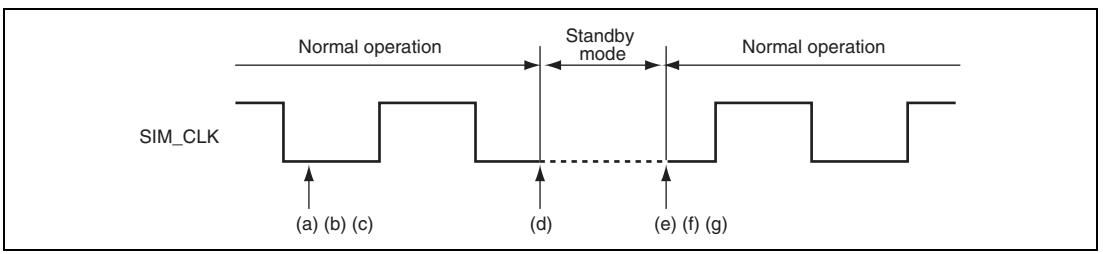


Figure 25.11 Procedure for Stopping Clock and Restarting

(5) Power-On and Clock Output

In order to retain the clock duty from power-on, the following switching procedure should be used.

1. In order to fix the potential, a pull-up resistance/pull-down resistance is used.
2. Use the CKE1 bit in the serial control register (SCSCR) to fix the specified output.
3. Set the CKE0 bit in SCSCR to 1 to start clock output.

(6) Pin Connections

An example of pin connections for the smart card interface is shown in figure 25.12.

In communication with the smart card, transmission and reception are performed using a single data transmit line. The data transmit line should be pulled up by a resistance on the power supply $V_{CC}Q$ side.

When using the clock generated by the smart card interface with the IC card, the SIM_CLK pin output is input to the CLK pin of the IC card. If an internal clock of the IC card is used, this connection is not needed.

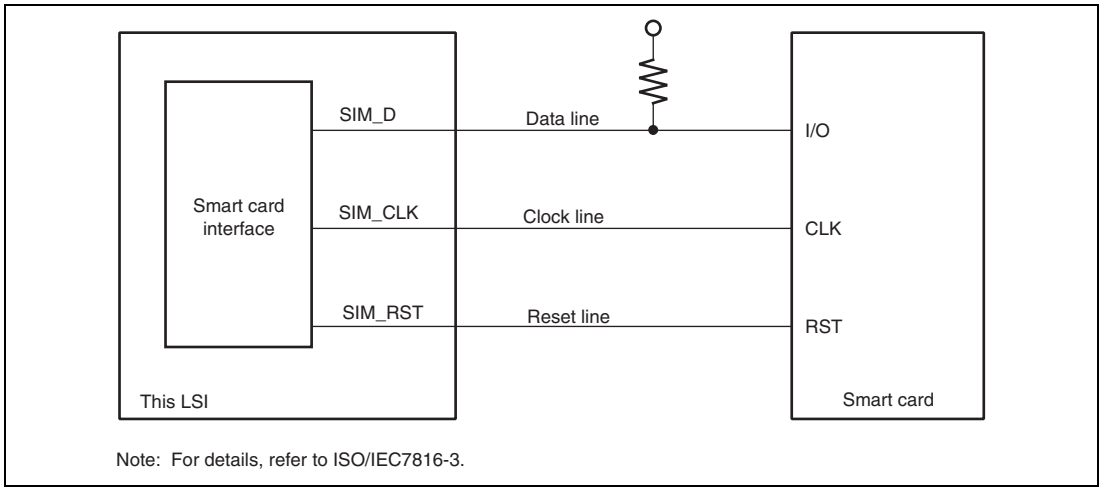


Figure 25.12 Example of Pin Connections in Smart Card Interface

Note: The transmission/reception in loop can perform self-check when the RE and TE bits are set to 1 without connecting to the IC card.

Section 26 A/D Converter

This LSI includes a 10-bit successive-approximation A/D converter allowing selection of up to four analog input channels.

26.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Four input channels
- High-speed conversion
  - Conversion time: maximum 8.5 μs per channel (Pϕ = 33 MHz operation)
- Three conversion modes
  - Single mode: A/D conversion on one channel
  - Multi mode: A/D conversion on one to four channels
  - Scan mode: Continuous A/D conversion on one to four channels
- Four 16-bit data registers
  - A/D conversion results are transferred for storage into 16-bit data registers corresponding to the channels.
- Sample-and-hold function
- A/D interrupt requested at the end of conversion
  - At the end of A/D conversion, an A/D end interrupt (ADI) can be requested.
- A/D conversion can be externally triggered

Figure 26.1 shows a block diagram of the A/D converter.

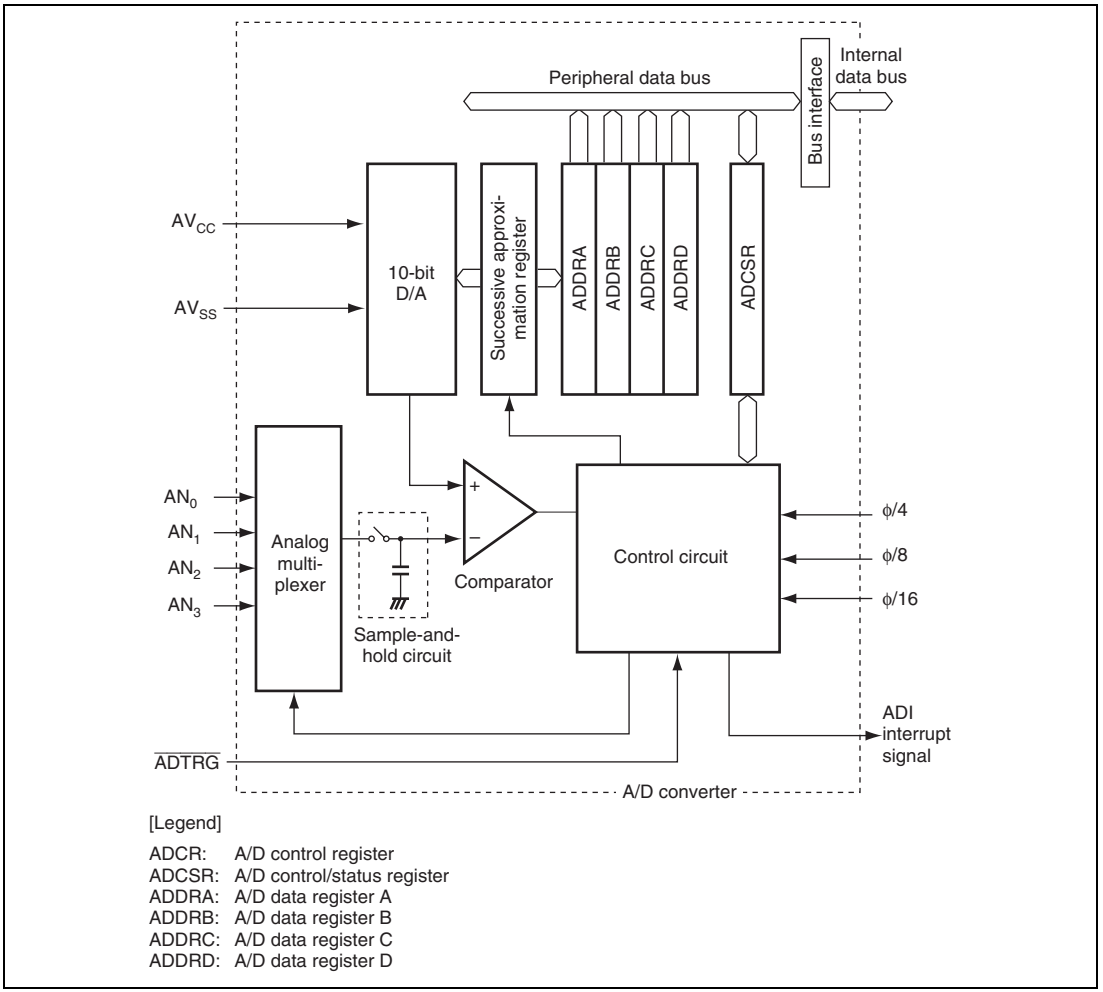


Figure 26.1 Block Diagram of A/D Converter

26.2 Input Pins

Table 26.1 summarizes the A/D converter's input pins. AV_{CC} and AV_{SS} are the power supply inputs for the analog circuits in the A/D converter. AV_{CC} also functions as the A/D converter reference voltage pin.

Table 26.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AVcc	Input	Analog power supply and reference voltage for A/D conversion
Analog ground pin	AVss	Input	Analog ground and reference voltage for A/D conversion
Analog input pin 0	AN0	Input	Analog inputs
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog trigger	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

26.3 Register Descriptions

Table 26.2 shows the ADC module register configuration. Table 26.3 shows the register states in each operating mode.

Table 26.2 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
A/D data register A	ADDRA	R	H'A461 0000	16
A/D data register B	ADDRB	R	H'A461 0002	16
A/D data register C	ADDRC	R	H'A461 0004	16
A/D data register D	ADDRD	R	H'A461 0006	16
A/D control/status register	ADCSR	R/W	H'A461 0008	16

Table 26.3 Register States in Each Operating Mode

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
ADDRA	Initialized	Initialized	Initialized	Retained
ADDRB	Initialized	Initialized	Initialized	Retained
ADDRC	Initialized	Initialized	Initialized	Retained
ADDRD	Initialized	Initialized	Initialized	Retained
ADCSR	Initialized	Initialized	Initialized	Retained

26.3.1 A/D Data Registers A to D (ADDRA to ADDRD)

The four A/D data registers (ADDRA to ADDRD) are 16-bit read-only registers that store the results of A/D conversion.

An A/D conversion produces 10-bit data, which is transferred for storage into the A/D data register corresponding to the selected channel. The upper 8 bits of the result are stored in the upper byte (bits 15 to 6) of the A/D data register. Bits 5 to 0 of an A/D data register are always read as 0. Table 26.4 indicates the pairings of analog input channels and A/D data registers.

The A/D data registers are initialized to H'0000 by a reset and in standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD[9:0]										—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	AD[9:0]	All 0	R	Bit data (10 bits)
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Table 26.4 Analog Input Channels and A/D Data Registers

Analog Input Channel	A/D Data Register
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD

26.3.2 A/D Control/Status Registers (ADCSR)

ADCSR is a 16-bit readable/writable register that selects the mode and controls the A/D converter. ADCSR is initialized to H'0000 by a reset and in standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	DMASL	TRGE[1:0]		—	—	CKS[1:0]		MULTI[1:0]		—	CH[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	ADF	0	R/(W)*	A/D End Flag Indicates the end of A/D conversion. [Clearing conditions] (1) Cleared by reading ADF while ADF = 1, then writing 0 to ADF (2) Cleared when DMAC is activated by ADI interrupt and ADDR is read [Setting conditions] • Single mode: A/D conversion ends • Multi mode: A/D conversion ends cycling through the selected channels • Scan mode: A/D conversion ends cycling through the selected channels Note: Clear this bit by writing 0.
14	ADIE	0	R/W	A/D Interrupt Enable Enables or disables the interrupt (ADI) requested at the end of A/D conversion. Set the ADIE bit while A/D conversion is not being made. 0: A/D end interrupt request (ADI) is disabled 1: A/D end interrupt request (ADI) is enabled

Bit	Bit Name	Initial Value	R/W	Description
13	ADST	0	R/W	A/D Start Starts or stops A/D conversion. The ADST bit remains set to 1 during A/D conversion. 0: A/D conversion is stopped 1: Single mode: A/D conversion starts; ADST is automatically cleared to 0 when conversion ends on all selected channels Multi mode: A/D conversion starts; when conversion is completed cycling through the selected channels, ADST is automatically cleared to 0 Scan mode: A/D conversion starts and continues; A/D conversion is continuously performed until ADST is cleared to 0 by software, by a reset, or by a transition to standby mode
12	DMASL	0	R/W	DMAC Select Selects an interrupt due to the end of A/D conversion or activation of the DMAC. Set the DMASL bit while A/D conversion is not being made. 0: An interrupt by the end of A/D conversion is selected 1: Activation of the DMAC by the end of A/D conversion is selected Always read as 0 when each register of A/D is read through CPU.
11, 10	TRGE[1:0]	00	R/W	Trigger Enable Enables or disables A/D conversion by external trigger input. 00: Disables A/D conversion by external trigger input 01: Reserved (setting prohibited) 10: Reserved (setting prohibited) 11: A/D conversion is started at the falling edge of A/D conversion trigger pin ($\overline{\text{ADTRG}}$)
9, 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CKS[1:0]	01	R/W	Clock Select Selects the A/D conversion time. Clear the ADST bit to 0 before changing the conversion time. 00: Conversion time = 151 states (maximum) 01: Conversion time = 285 states (maximum) 10: Conversion time = 545 states (maximum) 11: Reserved (setting prohibited) When Pϕ ≥ 16.5 MHz, do not set CKS1 and CKS0 to 00. If set, a sufficient conversion time is not assured, causing inaccurate conversion or abnormal operation.
5, 4	MULTI[1:0]	00	R/W	Selects single mode, multi mode, or scan mode. 00: Single mode 01: Reserved (setting prohibited) 10: Multi mode 11: Scan mode
3	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	CH[2:0]	000	R/W	Channel Select These bits and the MULTI bit select the analog input channels. Clear the ADST bit to 0 before changing the channel selection. Single mode / Multi mode or scan mode 000: AN0 / AN0 001: AN1 / AN0, AN1 010: AN2 / AN0 to AN2 011: AN3 / AN0 to AN3 100: Reserved (setting prohibited) 101: Reserved (setting prohibited) 110: Reserved (setting prohibited) 111: Reserved (setting prohibited)

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

26.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has three operating modes: single mode, multi mode, and scan mode.

26.4.1 Single Mode

Single mode should be selected when only one A/D conversion on one channel is required. A/D conversion starts when the ADST bit in the A/D control/status register (ADCSR) is set to 1 by software. The ADST bit remains set to 1 during A/D conversion and is automatically cleared to 0 when conversion ends.

When conversion ends the ADF bit in ADCSR is set to 1. If the ADIE bit in ADCSR is also set to 1 and DMASL is cleared to 0, an ADI interrupt is requested at this time. To clear the ADF flag to 0, first read ADF, then write 0 to ADF.

When the mode or analog input channel must be switched during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the mode or channel is changed.

Typical operations when channel 1 (AN1) is selected in single mode are described next.

Figure 26.2 shows a timing diagram for this example.

1. Supply of the clock to the ADC is started by setting the MSTPCR2.MSTP227 bit to 1 and activates AD conversion operation.
2. Single mode is selected (MULTI = 0), input channel AN1 is selected (CH[2:0] = 001), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
3. When A/D conversion is completed, the result is transferred into ADDRB. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
4. Since ADF = 1, ADIE = 1, and DMSL = 0 an ADI interrupt is requested.
5. The A/D interrupt handling routine starts.
6. The routine reads ADF, then writes 0 to the ADF flag.
7. The routine reads and processes the conversion result (ADDRB = 0).
8. Execution of the A/D interrupt handling routine ends.
9. Supply of the clock to the ADC is halted by setting the MSTPCR2.MSTP227 bit to 0 so that ADC enters module standby state.

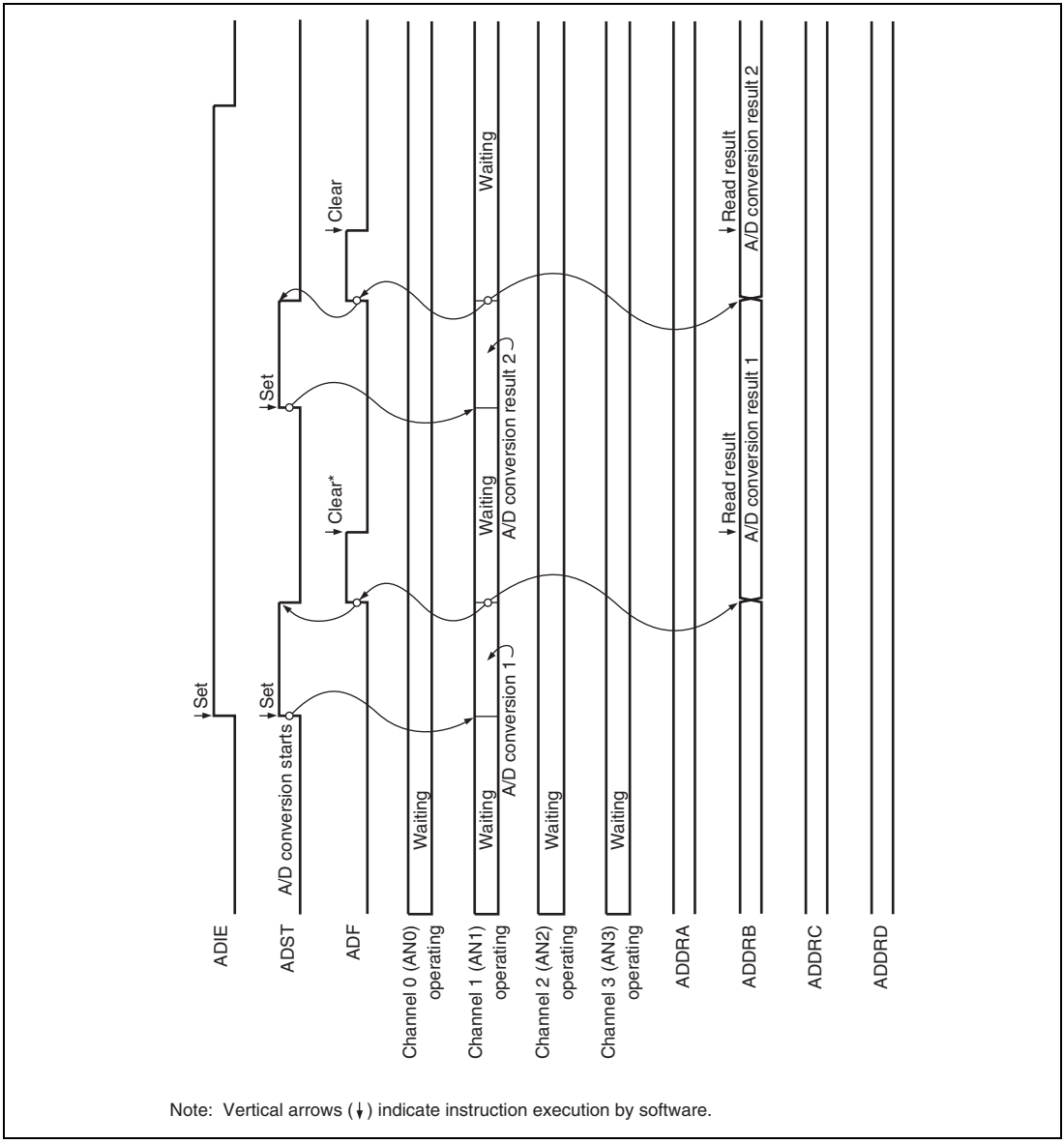


Figure 26.2 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

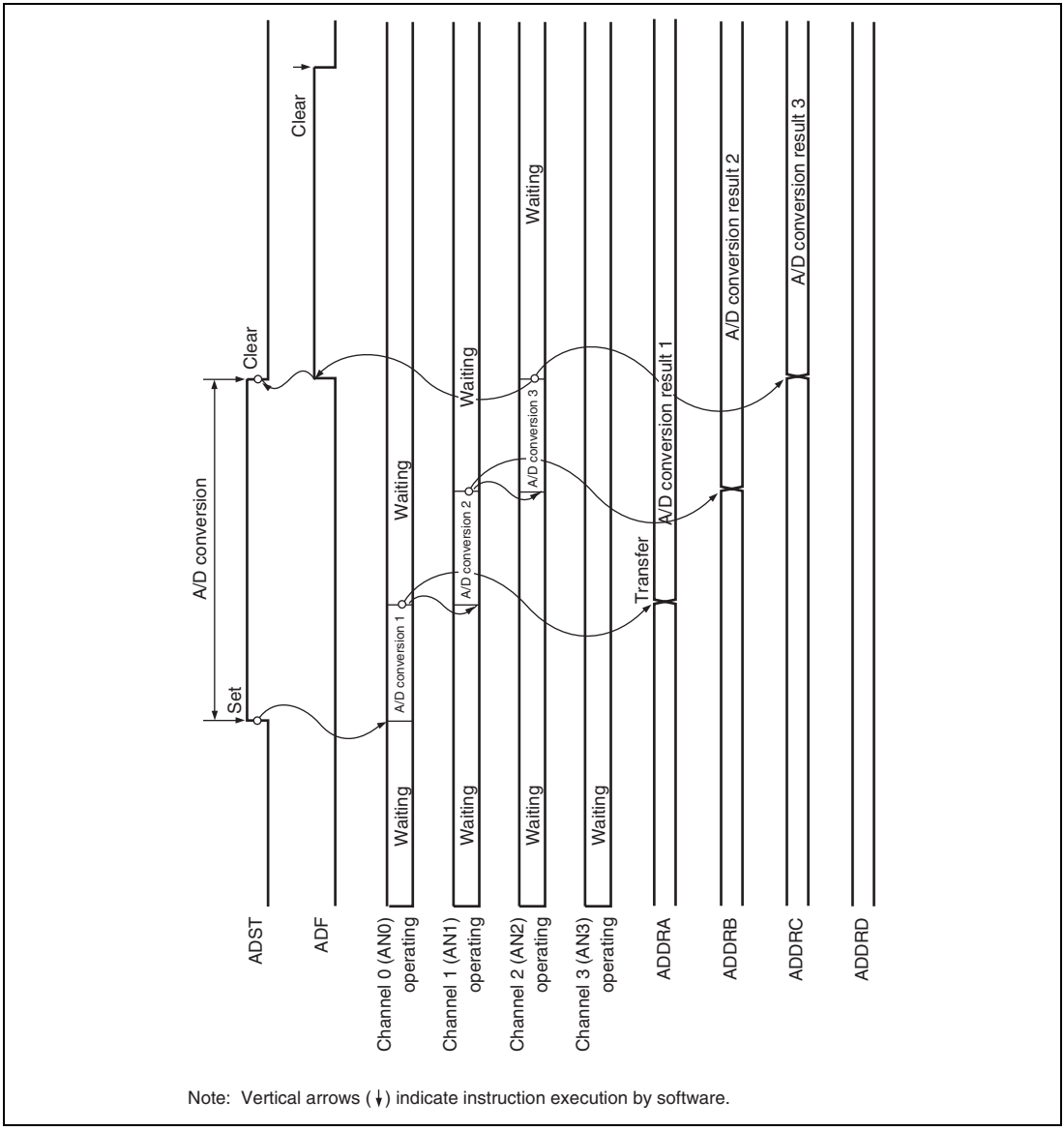
26.4.2 Multi Mode

Multi mode should be selected when performing A/D conversions on one or more channels. When the ADST bit in the A/D conversion control/status register (ADCSR) is set to 1 by software, A/D conversion starts on the first channel (AN0). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN1) starts immediately. When A/D conversions end on the selected channels, the ADST bit is cleared to 0. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the mode or analog input channel selection must be changed during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels (AN0 to AN2) are selected in scan mode are described next. Figure 26.3 shows a timing diagram for this example.

1. Supply of the clock to the ADC is started by setting the MSTPCR2.MSTP227 bit to 1 and activates AD conversion operation.
2. Multi mode is selected (MULTI = 1), analog input channels AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
3. When A/D conversion of the first channel (AN0) is completed, the result is transferred into ADDRA.
4. Next, conversion of the second channel (AN1) starts automatically.
5. Conversion proceeds in the same way through the third channel (AN2).
6. When conversion of all selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and ADST bit is cleared to 0. If the ADIE bit is set to 1, an ADI interrupt is requested at this time.
7. Supply of the clock to the ADC is halted by setting the MSTPCR2.MSTP227 bit to 0 so that ADC enters module standby state.



Note: Vertical arrows (↓) indicate instruction execution by software.

Figure 26.3 Example of A/D Converter Operation (Multi Mode, Channels AN0 to AN2 Selected)

26.4.3 Scan Mode

Scan mode is useful for monitoring analog inputs in a group of one or more channels. When the ADST bit in the A/D control/status register (ADCSR) is set to 1 by software, A/D conversion starts on the first channel (AN0). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN1) starts immediately. A/D conversion continues cyclically on the selected channels until the ADST bit is cleared to 0. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the mode or analog input channel must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels (AN0 to AN2) are selected in scan mode are described next. Figure 26.4 shows a timing diagram for this example.

1. Supply of the clock to the ADC is started by setting the MSTPCR2.MSTP227 bit to 1 and activates AD conversion operation.
2. Scan mode is selected, analog input channels AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
3. When A/D conversion of the first channel (AN0) is completed, the result is transferred into ADDRA.
4. Next, conversion of the second channel (AN1) starts automatically.
5. Conversion proceeds in the same way through the third channel (AN2).
6. When conversion of all the selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set to 1, an ADI interrupt is requested at this time.
7. Steps 3 to 5 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN0).
8. Supply of the clock to the ADC is halted by setting the MSTPCR2.MSTP227 bit to 0 so that ADC enters module standby state.

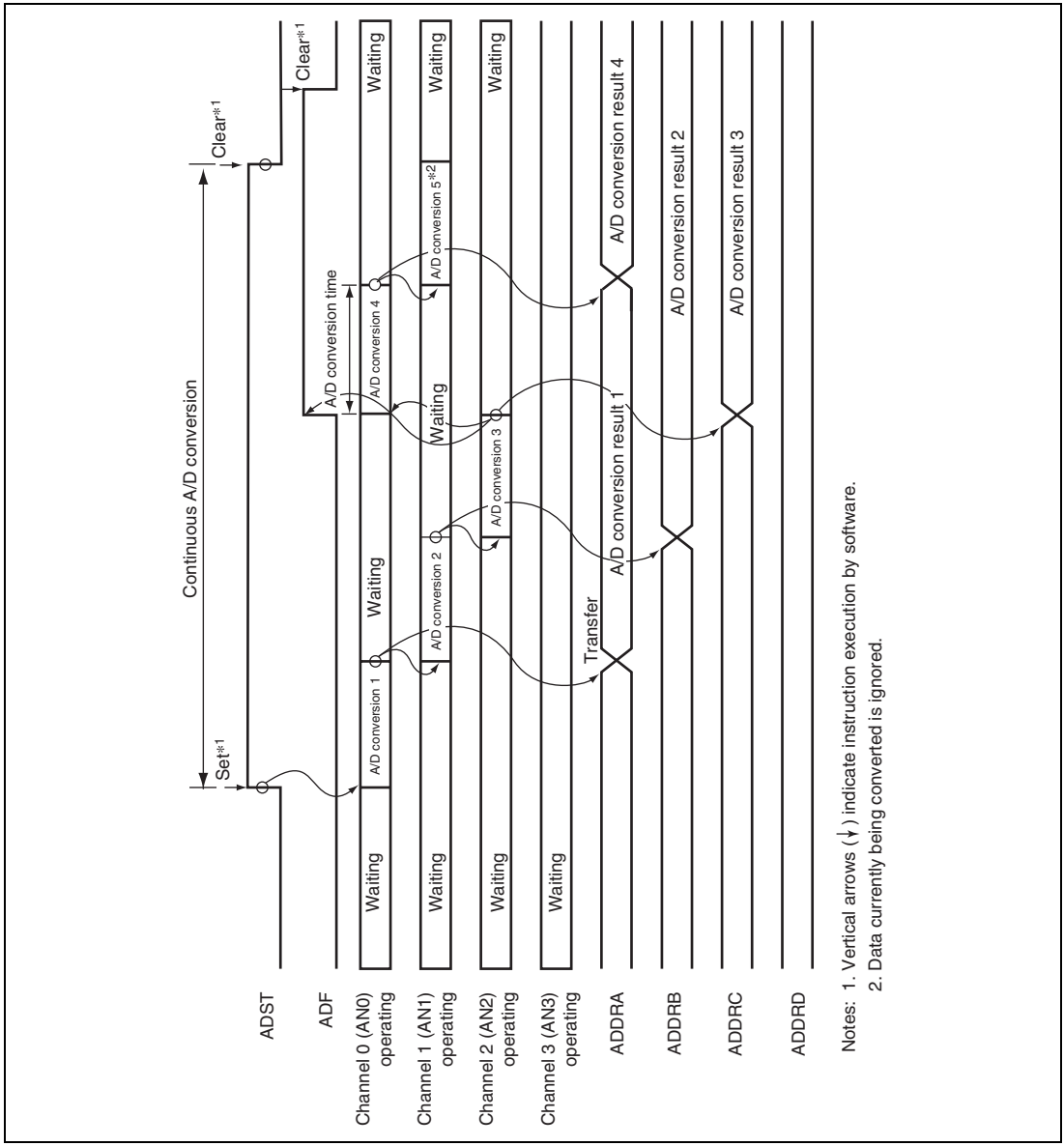


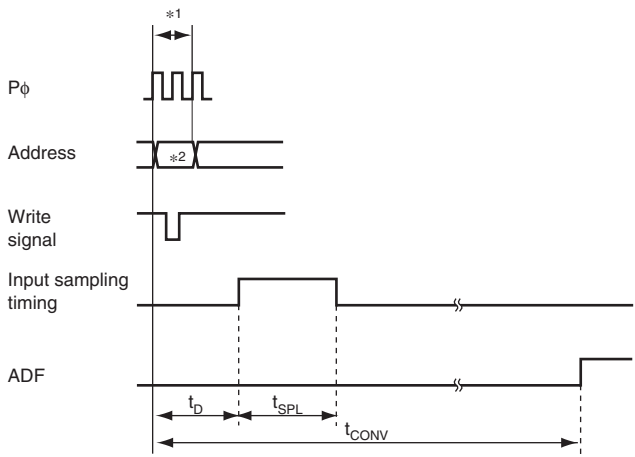
Figure 26.4 Example of A/D Converter Operation (Scan Mode, Channels AN0 to AN2 Selected)

26.4.4 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 at a time t_D after the ADST bit is set to 1, then starts conversion. Figure 26.5 shows the A/D conversion timing. Table 26.5 indicates the A/D conversion time.

As indicated in figure 26.5, the A/D conversion time includes t_D and the input sampling time. 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 26.5.

In multi mode and scan mode, the values given in table 26.5 apply to the first conversion. In the second and subsequent conversions the conversion the conversion time is fixed at 512 states (fixed) when CKS[1:0] = 10, 256 states (fixed) when CKS[1:0] = 01, and 128 states (fixed) when CKS[1:0] = 00.



t_D A/D conversion start delay
t_{SPL} Input sampling time
t_{CONV} A/D conversion time
Notes: 1. ADCSR write cycle
2. ADCSR address

Figure 26.5 A/D Conversion Timing

Table 26.5 A/D Conversion Time (Single Mode)

		CKS1 = 1, CKS0 = 0			CKS1 = 0, CKS0 = 1			CKS1 = 0, CKS0 = 0		
	Symbol	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.
A/D conversion start delay	t_D	18	—	21	10	—	13	6	—	9
Input sampling time	t_{SPL}	—	129	—	—	65	—	—	33	—
A/D conversion time	t_{CONV}	535	—	545	275	—	285	141	—	151

Note: Values in the table are numbers of states (t_{cyc}). Make settings such that the minimum conversion time is satisfied.

26.4.5 External Trigger Input Timing

The A/D conversion can also be started by the external trigger input. The external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin when bits TRGE1 and TRGE0 in A/D control register (ADCR) are set to 1. The falling edge of $\overline{\text{ADTRG}}$ input pin sets the ADST bit in the A/D control/status register (ADCSR) to 1, and then A/D conversion is started.

Other operations are the same as when the ADST bit is set to 1 by software, regardless of the conversion mode.

Figure 26.6 shows the timing.

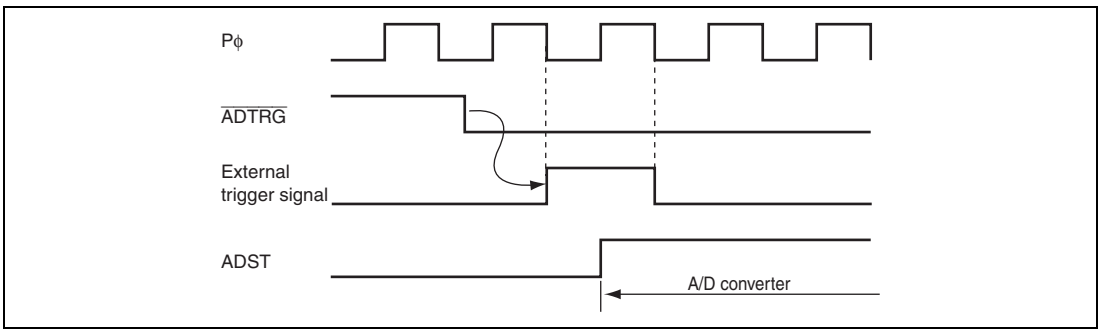


Figure 26.6 External Trigger Input Timing

26.5 Interrupts

The A/D converter generates an interrupt (ADI) at the end of A/D conversion. The ADI interrupt request can be enabled or disabled by the ADIE bit on the DMASL bit in ADCSR.

26.6 Definitions of A/D Conversion Accuracy

The A/D converter compares an analog value input from an analog input channel with its analog reference value and converts it to 10-bit digital data. The absolute accuracy of this A/D conversion is the deviation between the input analog value and the output digital value. It includes the following errors:

- Offset error
- Full-scale error
- Quantization error
- Nonlinearity error

These four error quantities are explained below with reference to figure 26.7. In the figure, the 10 bits of the A/D converter have been simplified to 3 bits.

Offset error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from the minimum (zero voltage) 0000000000 (000 in the figure) to 000000001 (001 in the figure)(figure 26.7, item (1)). Full-scale error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from the 1111111110 (110 in the figure) to the maximum 1111111111 (111 in the figure)(figure 26.7, item (2)). Quantization error is the intrinsic error of the A/D converter and is expressed as 1/2 LSB (figure 26.7, item (3)). Nonlinearity error is the deviation between actual and ideal A/D conversion characteristics between zero voltage and full-scale voltage (figure 26.7, item (4)). Note that it does not include offset, full-scale, or quantization error.

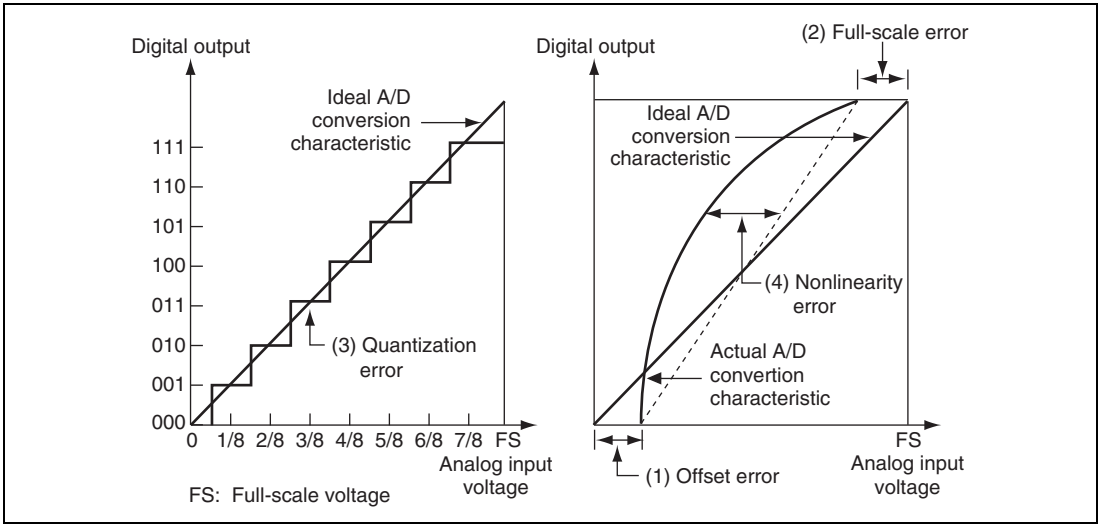


Figure 26.7 Definitions of A/D Conversion Accuracy

26.7 Usage Notes

26.7.1 Allowable Signal-Source Impedance

For the analog input design of this LSI, conversion accuracy is guaranteed for an input signal with signal-source impedance of 5 kΩ or less. The specification is for charging input capacitance of the sample and hold circuit of the A/D converter within sampling time. When the output impedance of the sensor exceeds 5 kΩ, conversion accuracy is not guaranteed due to insufficient charging. If large external capacitance is set at conversion in single mode, signal-source impedance is ignored since input load is only internal input resistance of 3 kΩ. However, an analog signal with large differential coefficient (5 mV/μs or greater) cannot be followed up because of a low-pass filter (figure 26.8). When converting high-speed analog signals or converting in scan mode, insert a low-impedance buffer.

26.7.2 Influence to Absolute Accuracy

By adding capacitance, absolute accuracy may be degraded if noise is on GND because there is coupling with GND. Therefore, connect electrically stable GND such as AVss to prevent absolute accuracy from being degraded.

A filter circuit must not interfere with digital signals, or must not be an antenna on a mounting board.

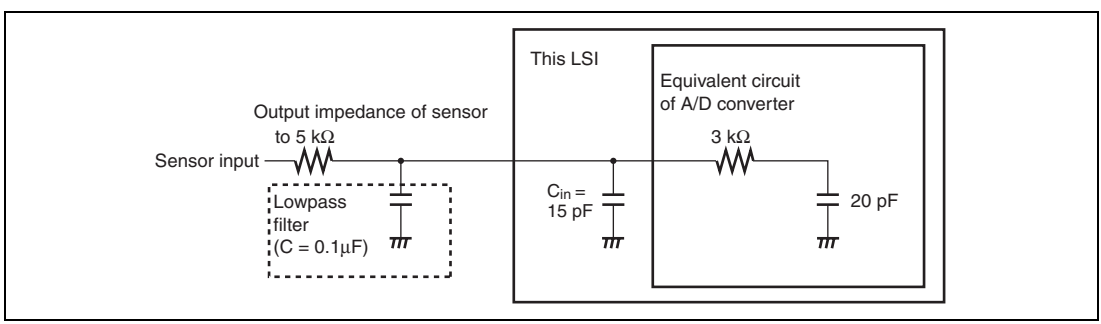


Figure 26.8 Analog Input Circuit Example

26.7.3 Setting Analog Input Voltage

Operating the chip in excess of the following voltage range may result in damage to chip reliability.

- Analog Input Voltage Range: During A/D conversion, the voltages (VANn) input to the analog input pins ANn should be in the range $AV_{SS} \leq VANn \leq AV_{CC}$ (n = 0 to 3).
- Relationships of AV_{CC}, AV_{SS} and $V_{CC}Q$, $V_{SS}Q$: $V_{CC}Q$-0.3 V ≤ AV_{CC} ≤ $V_{CC}Q$ + 0.3 V and AV_{SS} = V_{SS}. Even when the A/D converter is not used, do not open AVcc and AVss. Connect AVcc to VccQ and Avss to VssQ.

26.7.4 Notes on Board Design

In designing a board, separate digital circuits and analog circuits. Do not intersect or locate closely signal lines of a digital circuit and an analog circuit. An analog circuit may malfunction due to induction, thus affecting A/D conversion values. Separate analog input pins (AN0 to AN3) and the analog power voltage (AVcc) from digital circuits with analog ground (AVss). Connect analog ground (AVss) to one point of stable ground (Vss) on the board.

26.7.5 Notes on Countermeasures to Noise

Connect a protective circuit between AVcc and AVss, as shown in figure 26.9, to prevent damage of analog input pins (AN0 to AN3) due to abnormal voltage such as excessive serge. Connect a bypass capacitor that is connected to AVcc and a capacitor for a filter that is connected to AN0 to AN3 to AVss.

When a capacitor for a filter is connected, input currents of AN0 to AN3 are averaged, may causing errors. If A/D conversion is frequently performed in scan mode, voltages of analog input pins cause errors when a current that is charged/discharged for capacitance of a sample & hold circuit in the A/D converter is higher than a current that is input through input impedance (Rin). Therefore, determine a circuit constant carefully.

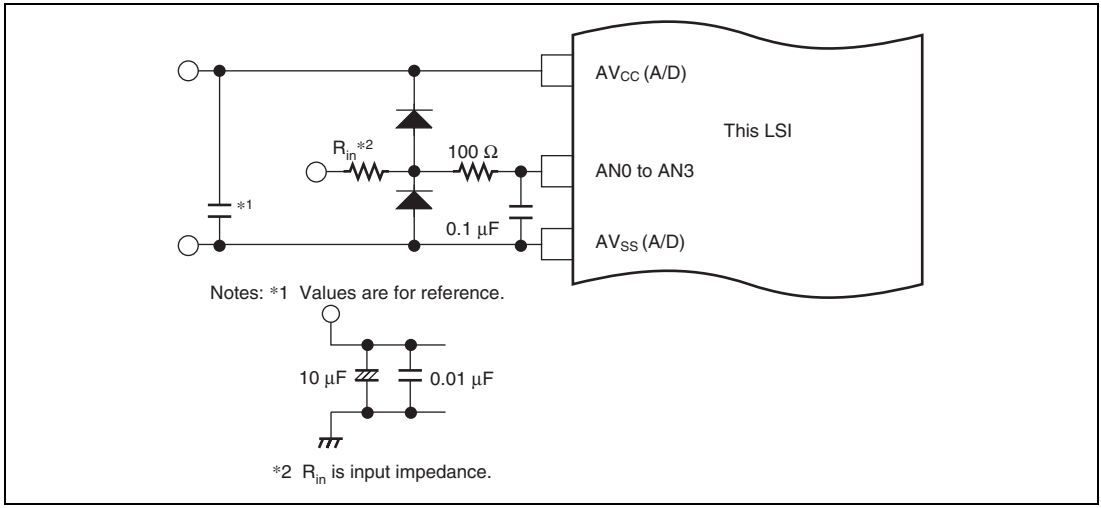


Figure 26.9 Example of Analog Input Protection Circuit

Table 26.6 Analog Input Pin Ratings

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Allowable signal-source impedance	—	5	kΩ

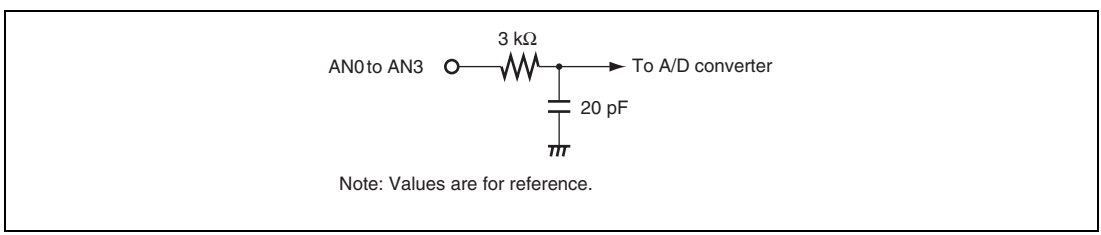


Figure 26.10 Analog Input Pin Equivalent Circuit

26.7.6 Notes on A/D Conversion

Every time A/D conversion is completed, enter the ADC module to the standby state.

When executing A/D conversion again, exit the module standby state and activate the ADC module.

Section 27 D/A Converter (DAC)

This LSI incorporates a two-channel D/A converter (DAC) with the following features.

27.1 Features

- 10-bit resolution
- Two output channels
- Conversion time: Max. 10 μs (when load capacitance is 30 pF)
- Output voltage: 0 V to AVcc (analog power supply)

Figure 27.1 shows the block diagram for the DAC.

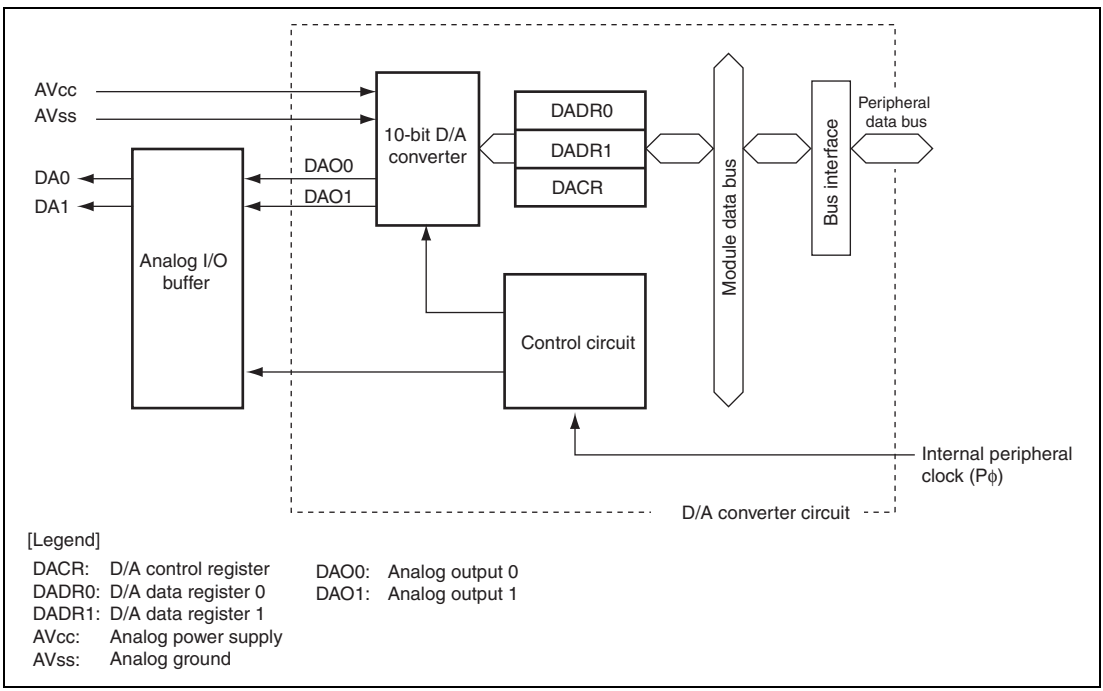


Figure 27.1 Block Diagram of D/A Converter

27.2 Input/Output Pins

Table 27.1 summarizes the input/output pins used by the D/A converter.

Table 27.1 Pin Configuration

Pin Name	I/O	Function
AVcc	—	Analog block power supply and D/A conversion reference voltage
AVss	—	Analog block ground
DA0	Output	Channel 0 analog output
DA1	Output	Channel 1 analog output

27.3 Register Descriptions

Table 27.2 shows the DAC module register configuration. Table 27.3 shows the register states in each operating mode.

Table 27.2 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
D/A data register 0	DADR0	R/W	H'A462 0000	16
D/A data register 1	DADR1	R/W	H'A462 0002	16
A/D control register	DACR	R/W	H'A462 0004	16

Table 27.3 Register States in Each Operating Mode

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
DADR0	Initialized	Retained	Retained	Retained
DADR1	Initialized	Retained	Retained	Retained
DACR	Initialized	Retained	Retained	Retained

27.3.1 D/A Data Registers 0 and 1 (DADR0, DADR1)

DADR0 and DADR1 are 16-bit readable/writable registers that store data for D/A conversion. When the D/A output enable bits (DAOE1, DAOE0) of the DA control register (DACR) are set to 1, the contents of the D/A data register are converted and output to analog output pins (DA0, DA1). The D/A data register is initialized to H'0000 at reset. Note that the D/A data register is not initialized upon entering the software standby, module standby, or hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DAD[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	DAD[9:0]	H'000	R/W	10-bit register that stores data for D/A conversion.

27.3.2 D/A Control Register (DACR)

The DACR register is a 16-bit readable/writable register that controls D/A converter operation. The DACR is initialized to H'0000 at reset. Note that the DACR is not initialized in software standby or module standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DAOE1	DAOE0	—	—	—	—	—	DAE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE1	0	R/W	Controls D/A conversion for channel 1 and analog output. 0: D/A conversion for channel 1 and analog output (DA1) are disabled 1: D/A conversion for channel 1 and analog output (DA1) are enabled
6	DAOE0	0	R/W	Controls D/A conversion for channel 0 and analog output. 0: D/A conversion for channel 0 and analog output (DA0) are disabled 1: D/A conversion for channel 0 and analog output (DA0) are enabled
5 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DAE	0	R/W	Enables or disables D/A conversion. 0: D/A conversion is stopped on both channels 0 and 1, and a low level is output. 1: D/A conversion is performed on both channels 0 and 1

27.4 Operation

The D/A converter incorporates two D/A channels that can operate individually.

The D/A converter executes D/A conversion while analog output is enabled by the D/A control register (DACR). When the D/A data registers (DADR0 and DADR1) are modified, the D/A converter immediately initiates the new data conversion. Setting the DAE bit in DACR to 1 starts D/A conversion, and setting the DAOE1 or DAOE0 bit in DACR to 1 enables the output of the conversion results of the corresponding channel.

An example of D/A conversion on channel 0 is shown below. The operation timing is shown in figure 27.2.

1. Write data for conversion to DADR0.
2. Set the DAE and DAOE0 bits in DACR to 1. D/A conversion starts, and the DA0 output is enabled. The result of conversion is output after the conversion has ended. The output value will be (DADR0 contents/1024) × AVcc.

   The conversion results are output continuously until DADR0 is modified or the DAOE0 bit is cleared to 0.
3. When D/A data register 0 (DMDR0) is modified, the conversion starts again.

   The results are output after the conversion has ended.
4. When the DAOE0 bit is cleared to 0, analog output is disabled (high-impedance state).

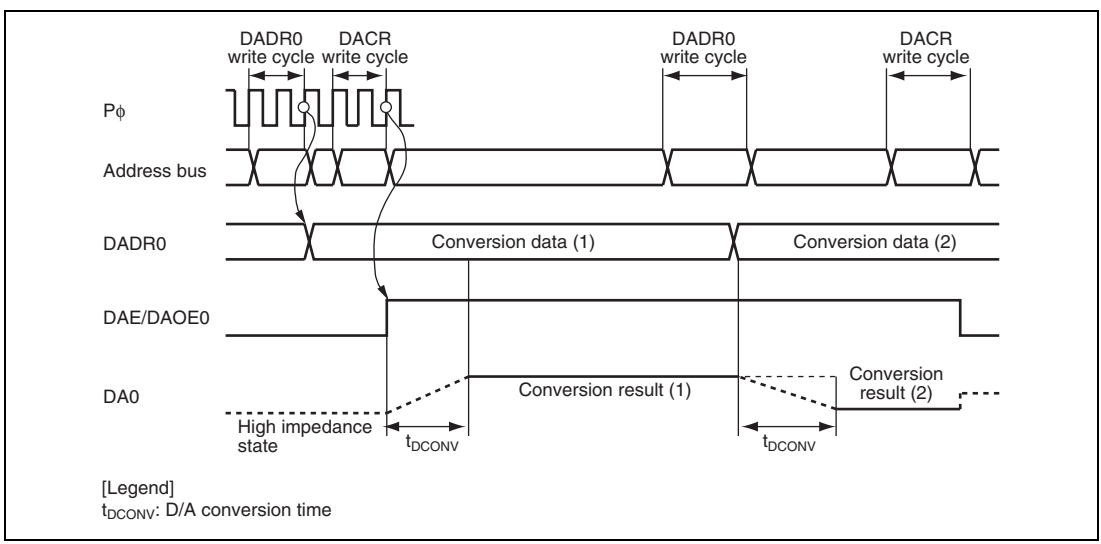


Figure 27.2 D/A Converter Operation Example

Section 28 I/O Port

This LSI has seventeen 8-bit ports* (ports A to T). All port pins are multiplexed with other pin functions. The pin function controller (PFC) handles the selection of pin functions and pull-up/pull-down MOS control. Each port has one data register for the storage of pin data.

Note: * On some ports, not all of the eight bits are effective.

28.1 Register Descriptions

Table 28.1 shows the register configuration of I/O ports. Table 28.2 shows the register states in each operating mode.

Table 28.1 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
Port A data register	PADR	R/W	H'A405 0080	8
Port B data register	PBDR	R/W	H'A405 0082	8
Port C data register	PCDR	R/W	H'A405 0084	8
Port D data register	PDDR	R/W	H'A405 0086	8
Port E data register	PEDR	R/W	H'A405 0088	8
Port F data register	PFDR	R/W	H'A405 008A	8
Port G data register	PGDR	R/W	H'A405 008C	8
Port H data register	PHDR	R/W	H'A405 008E	8
Port J data register	PJDR	R/W	H'A405 0090	8
Port K data register	PKDR	R/W	H'A405 0092	8
Port L data register	PLDR	R/W	H'A405 0094	8
Port M data register	PMDR	R/W	H'A405 0096	8
Port N data register	PNDR	R/W	H'A405 0098	8
Port Q data register	PQDR	R/W	H'A405 009A	8
Port R data register	PRDR	R/W	H'A405 009C	8
Port S data register	PSDR	R/W	H'A405 009E	8
Port T data register	PTDR	R/W	H'A405 00A0	8

Table 28.2 Register States of I/O Ports in Each Operating Mode

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
PADR	Initialized	Retained	—	Retained
PBDR	Initialized	Retained	—	Retained
PCDR	Initialized	Retained	—	Retained
PDDR	Initialized	Retained	—	Retained
PEDR	Initialized	Retained	—	Retained
PFDR	Initialized	Retained	—	Retained
PGDR	Initialized	Retained	—	Retained
PHDR	Initialized	Retained	—	Retained
PJDR	Initialized	Retained	—	Retained
PKDR	Initialized	Retained	—	Retained
PLDR	Initialized	Retained	—	Retained
PMDR	Initialized	Retained	—	Retained
PNDR	Initialized	Retained	—	Retained
PQDR	Initialized	Retained	—	Retained
PRDR	Initialized	Retained	—	Retained
PSDR	Initialized	Retained	—	Retained
PTDR	Initialized	Retained	—	Retained

28.2 Port A

Port A is an input/output port with the pin configuration shown in figure 28.1. Each pin has an input pull-up MOS, which is controlled by the port A control register (PACR) in the PFC.

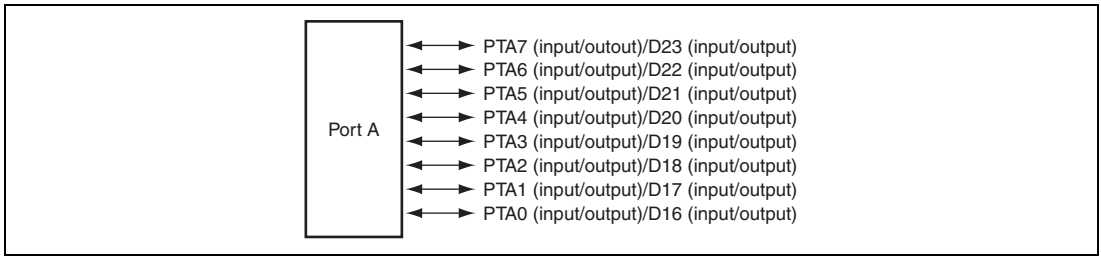


Figure 28.1 Port A

28.2.1 Port A Data Register (PADR)

PADR is a register that stores data for pins PTA7 to PTA0. Bits PA7DT to PA0DT correspond to pins PTA7 to PTA0. For pins that function as general-purpose output pins, a read operation directly reads out the corresponding value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	PA1DT	PA0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DT	0	R/W	Table 28.3 shows the function of PADR.
6	PA6DT	0	R/W	
5	PA5DT	0	R/W	
4	PA4DT	0	R/W	
3	PA3DT	0	R/W	
2	PA2DT	0	R/W	
1	PA1DT	0	R/W	
0	PA0DT	0	R/W	

Table 28.3 Port A Data Register (PADR) Read/Write Operations

PACR State				
PAnMD1	**PAnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PADR value	The value is written to PADR, but does not affect the pin state.
	1	Output	PADR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PADR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PADR, but does not affect the pin state.

Note: n = 7 to 0

28.3 Port B

Port B is an input/output port with the pin configuration shown in figure 28.2. Each pin has an input pull-up MOS, which is controlled by the port B control register (PBCR) in the PFC.

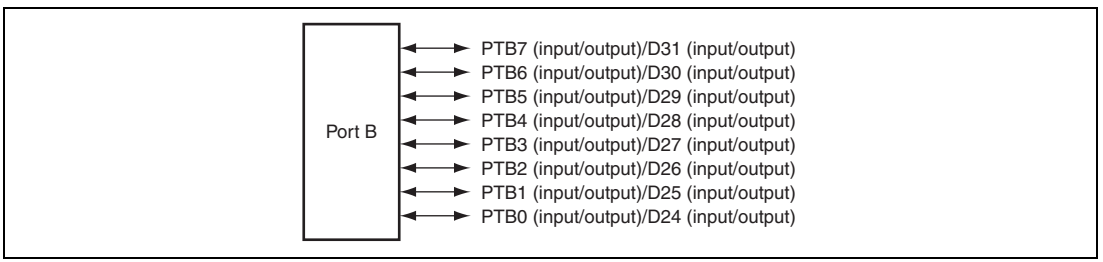


Figure 28.2 Port B

28.3.1 Port B Data Register (PBDR)

PBDR is a register that stores data for pins PTB7 to PTB0. Bits PB7DT to PB0DT correspond to pins PTB7 to PTB0. For pins that function as general-purpose output pins, a read operation directly reads out the corresponding value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DT	0	R/W	Table 28.4 shows the function of PBDR.
6	PB6DT	0	R/W	
5	PB5DT	0	R/W	
4	PB4DT	0	R/W	
3	PB3DT	0	R/W	
2	PB2DT	0	R/W	
1	PB1DT	0	R/W	
0	PB0DT	0	R/W	

Table 28.4 Port B Data Register (PBDR) Read/Write Operations

PBCR State				
PBnMD1	**PBnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PBDR value	The value is written to PBDR, but does not affect the pin state.
	1	Output	PBDR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PBDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PBDR, but does not affect the pin state.

Note: n = 7 to 0

28.4 Port C

Port C is an input/output port with the pin configuration shown in figure 28.3. Each pin has an input pull-up/pull-down MOS, which is controlled by the port C control register (PCCR) in the PFC. Select pull-up or pull-down MOS with PINT control register A (PINTCRA) in the PFC.

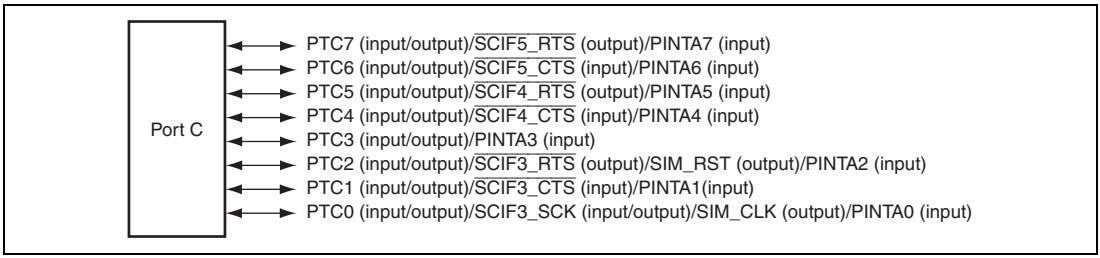


Figure 28.3 Port C

28.4.1 Port C Data Register (PCDR)

PCDR is a register that stores data for pins PTC7 to PTC0. Bits PC7DT to PC0DT correspond to pins PTC7 to PTC0. For pins that function as general-purpose output pins, a read operation directly reads out the corresponding value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DT	0	R/W	Table 28.5 shows the function of PCDR.
6	PC6DT	0	R/W	
5	PC5DT	0	R/W	
4	PC4DT	0	R/W	
3	PC3DT	0	R/W	
2	PC2DT	0	R/W	
1	PC1DT	0	R/W	
0	PC0DT	0	R/W	

Table 28.5 Port C Data Register (PCDR) Read/Write Operations

PCCR State				
PCnMD1	**PCnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PCDR value	The value is written to PCDR, but does not affect the pin state.
	1	Output	PCDR value	The write value is output from the pin.
1	0	Input (Pull-up/ pull-down MOS on)	Pin state	The value is written to PCDR, but does not affect the pin state.
	1	Input (Pull-up/ pull-down MOS off)	Pin state	The value is written to PCDR, but does not affect the pin state.

Note: n = 7 to 0

28.5 Port D

Port D is an input/output port with the pin configuration shown in figure 28.4. Each pin has an input pull-up MOS, which is controlled by the port D control register (PDCR) in the PFC.

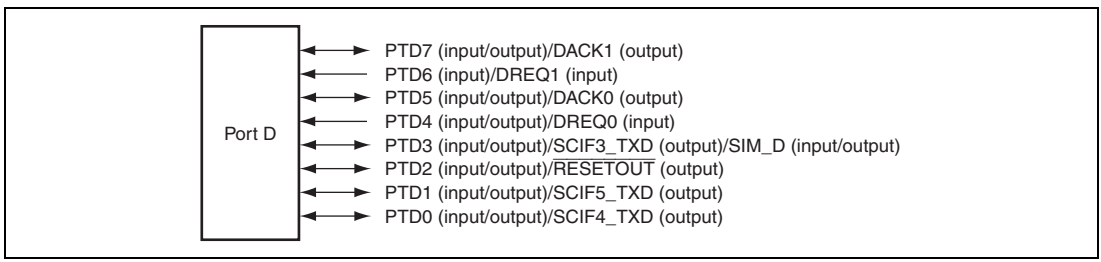


Figure 28.4 Port D

28.5.1 Port D Data Register (PDDR)

PDDR is a register that stores data for pins PTD7 to PTD0. Bits PD7DT to PD0DT correspond to pins PTD7 to PTD0. For pins that function as general-purpose output pins, a read operation directly reads out the corresponding value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DT	0	R/W	Table 28.6 shows the function of PDDR.
6	PD6DT	0	R	
5	PD5DT	0	R/W	
4	PD4DT	0	R	
3	PD3DT	0	R/W	
2	PD2DT	0	R/W	
1	PD1DT	0	R/W	
0	PD0DT	0	R/W	

Table 28.6 Port D Data Register (PDDR) Read/Write Operations

- PD0DT to PG3DT, PD5DT, PD7DT

PDCR State				
PDnMD1	**PDnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PDDR value	The value is written to PDDR, but does not affect the pin state.
	1	Output	PDDR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PDDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PDDR, but does not affect the pin state.

Note: n = 0 to 3, 5, 7

- PD4DT, PD6DT

PDCR State

PDnMD1	PDnMD0	Pin State	Read	Write
0	0	Other function	PDDR value	The value is written to PDDR, but does not affect the pin state.
	1	—	—	—
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PDDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PDDR, but does not affect the pin state.

Note: n = 4, 6

28.6 Port E

Port E is an input/output port with the pin configuration shown in figure 28.5. Each pin has an input pull-up MOS, which is controlled by the port E control register (PECR) in the PFC.

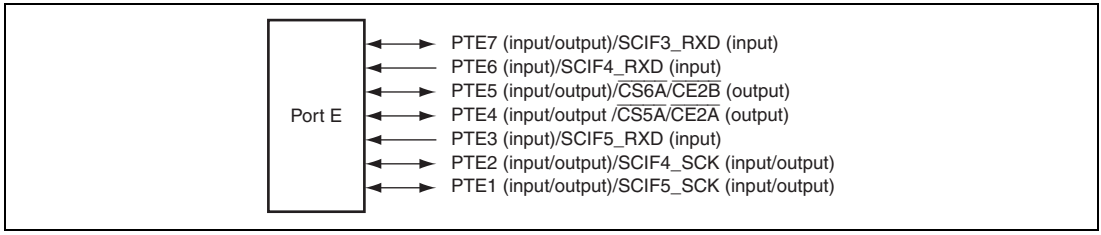


Figure 28.5 Port E

28.6.1 Port E Data Register (PEDR)

PEDR is a register that stores data for pins PTE7 to PTE1. Bits PE7DT to PE1DT correspond to pins PTE7 to PTE1. For pins that function as general-purpose output pins, a read operation directly reads out the corresponding value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	PE7DT	PE6DT	PE5DT	PE4DT	PE3DT	PE2DT	PE1DT	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DT	0	R/W	Table 28.7 shows the function of PEDR.
6	PE6DT	0	R	
5	PE5DT	0	R/W	
4	PE4DT	0	R/W	
3	PE3DT	0	R	
2	PE2DT	0	R/W	
1	PE1DT	0	R/W	
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Table 28.7 Port E Data Register (PEDR) Read/Write Operations

- PE1DT, PE2DT, PE4DT, PE5DT, PE7DT

PECR State				
PEnMD1	**PEnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PEDR value	The value is written to PEDR, but does not affect the pin state.
	1	Output	PEDR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PEDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PEDR, but does not affect the pin state.

Note: n = 1, 2, 4, 5, 7

- PE3DT, PE6DT

PECR State				
PEnMD1	**PEnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PEDR value	The value is written to PEDR, but does not affect the pin state.
	1	—	—	—
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PEDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PEDR, but does not affect the pin state.

Note: n = 3, 6

28.7 Port F

Port F is an input/output port with the pin configuration shown in figure 28.6. Each pin has an input pull-up/pull-down MOS, which is controlled by the port F control register (PFCR) in the PFC. Select pull-up or pull-down MOS with PINT control register B (PINTCRB) in the PFC.

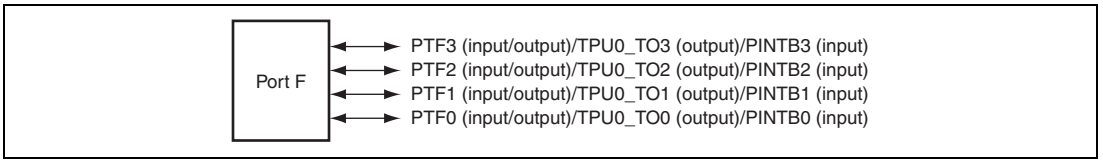


Figure 28.6 Port F

28.7.1 Port F Data Register (PFDR)

PFDR is a register that stores data for pins PTF3 to PTF0. Bits PF3DT to PF0DT correspond to pins PTF3 to PTF0. For pins that function as general-purpose output pins, a read operation directly reads out the value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	PF3DT	PF2DT	PF1DT	PF0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	PF3DT	0	R/W	Table 28.8 shows the function of PFDR.
2	PF2DT	0	R/W	
1	PF1DT	0	R/W	
0	PF0DT	0	R/W	

Table 28.8 Port F Data Register (PFDR) Read/Write Operations

PFCR State				
PFnMD1	**PFnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PFDR value	The value is written to PFDR, but does not affect the pin state.
	1	Output	PFDR value	The write value is output from the pin.
1	0	Input (Pull-up/ pull-down MOS on)	Pin state	The value is written to PFDR, but does not affect the pin state.
	1	Input (Pull-up/ pull-down MOS off)	Pin state	The value is written to PFDR, but does not affect the pin state.

Note: n = 0 to 3

28.8 Port G

Port G is an input/output port with the pin configuration shown in figure 28.7. Each pin has an input pull-up MOS, which is controlled by the port G control register (PGCR) in the PFC.

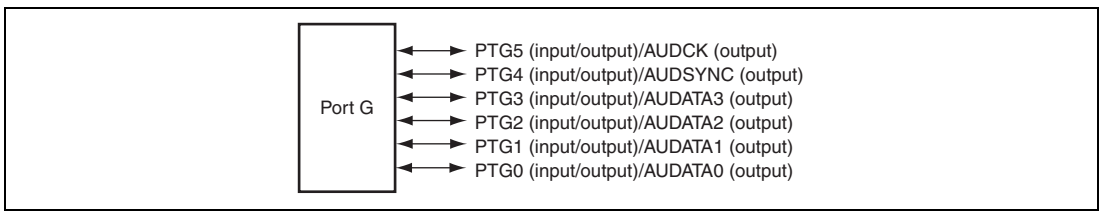


Figure 28.7 Port G

28.8.1 Port G Data Register (PGDR)

PGDR is a register that stores data for pins PTG5 to PTG0. Bits PG5DT to PG0DT correspond to pins PTG5 to PTG0. For pins that function as general-purpose output pins, a read operation directly reads out the value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	—	—	PG5DT	PG4DT	PG3DT	PG2T	PG1DT	PG0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	PG5DT	0	R/W	Table 28.9 shows the function of PGDR.
4	PG4DT	0	R/W	
3	PG3DT	0	R/W	
2	PG2DT	0	R/W	
1	PG1DT	0	R/W	
0	PG0DT	0	R/W	

Table 28.9 Port G Data Register (PGDR) Read/Write Operations

PGCR State				
PGnMD1	**PGnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PGDR value	The value is written to PGDR, but does not affect the pin state.
	1	Output	PGDR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PGDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PGDR, but does not affect the pin state.

Note: n = 0 to 5

28.9 Port H

Port H is an input/output port with the pin configuration shown in figure 28.8. Each pin has an input pull-up MOS, which is controlled by the port H control register (PHCR) in the PFC.

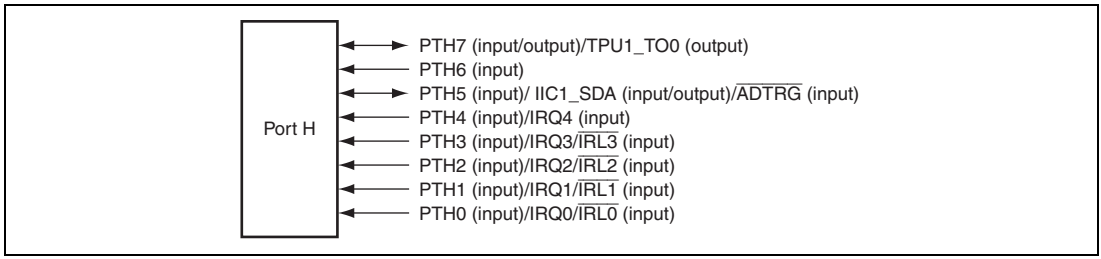


Figure 28.8 Port H

28.9.1 Port H Data Register (PHDR)

PHDR is a register that stores data for pins PTH7 to PTH0. Bits PH7DT to PH0DT correspond to pins PTH7 to PTH0. For pins that function as general-purpose output pins, a read operation reads out the value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	PH7DT	PH6DT	PH5DT	PH4DT	PH3DT	PH2DT	PH1DT	PH0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	PH7DT	0	R/W	Table 28.10 shows the function of PHDR.
6	PH6DT	0	R	
5	PH5DT	0	R	
4	PH4DT	0	R	
3	PH3DT	0	R	
2	PH2DT	0	R	
1	PH1DT	0	R	
0	PH0DT	0	R	

Table 28.10 Port H Data Register (PHDR) Read/Write Operations

- PH7DT

PHCR State				
PHnMD1	**PHnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PHDR value	The value is written to PHDR, but does not affect the pin state.
	1	Output	PHDR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PHDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PHDR, but does not affect the pin state.

Note: n = 7

- PH0DT to PH4DT, PH6DT

PHCR State				
PHnMD1	**PHnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PHDR value	The value is written to PHDR, but does not affect the pin state.
	1	—	—	—
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PHDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PHDR, but does not affect the pin state.

Note: n = 0 to 4, 6

- PH5DT

PHCR State				
PHnMD1	**PHnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PHDR value	The value is written to PHDR, but does not affect the pin state.
	1	—	—	—
1	0	Input	Pin state	The value is written to PHDR, but does not affect the pin state.
	1	—	—	—

Note: n = 5

28.10 Port J

Port J is an input/output port with the pin configuration shown in figure 28.9. Each pin has an input pull-up MOS, which is controlled by the port J control register (PJCR) in the PFC.

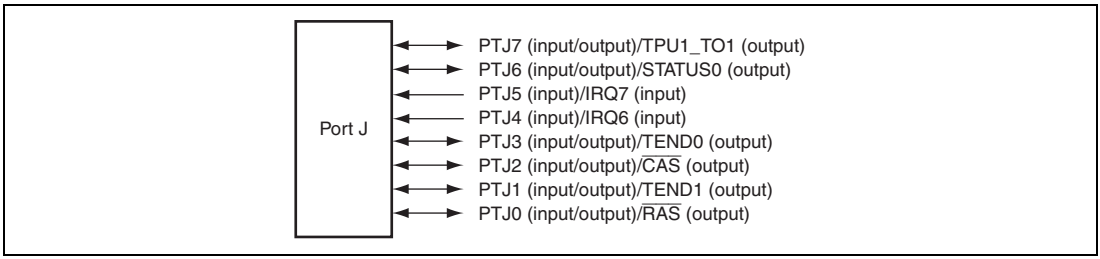


Figure 28.9 Port J

28.10.1 Port J Data Register (PJDR)

PJDR is a register that stores data for pins PTJ7 to PTJ0. Bits PJ7DT to PJ0DT correspond to pins PTJ7 to PTJ0. For pins that function as general-purpose output pins, a read operation directly reads out the value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	PJ7DT	PJ6DT	PJ5DT	PJ4DT	PJ3DT	PJ2DT	PJ1DT	PJ0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PJ7DT	0	R/W	Table 28.11 shows the function of PJDR.
6	PJ6DT	0	R/W	
5	PJ5DT	0	R	
4	PJ4DT	0	R	
3	PJ3DT	0	R/W	
2	PJ2DT	0	R/W	
1	PJ1DT	0	R/W	
0	PJ0DT	0	R/W	

Table 28.11 Port J Data Register (PJDR) Read/Write Operations

- PJ0DT to PJ3DT, PJ6DT, PJ7DT

PJCR State				
PJnMD1	**PJnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PJDR value	The value is written to PJDR, but does not affect the pin state.
	1	Output	PJDR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PJDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PJDR, but does not affect the pin state.

Note: n = 0 to 3, 6, 7

- PJ4DT, PJ5DT

PJCR State				
PJnMD1	**PJnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PJDR value	The value is written to PJDR, but does not affect the pin state.
	1	—	—	—
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PJDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PJDR, but does not affect the pin state.

Note: n = 4, 5

28.11 Port K

Port K is an input/output port with the pin configuration shown in figure 28.10. Each pin has an input pull-up MOS, which is controlled by the port K control register (PKCR) in the PFC.

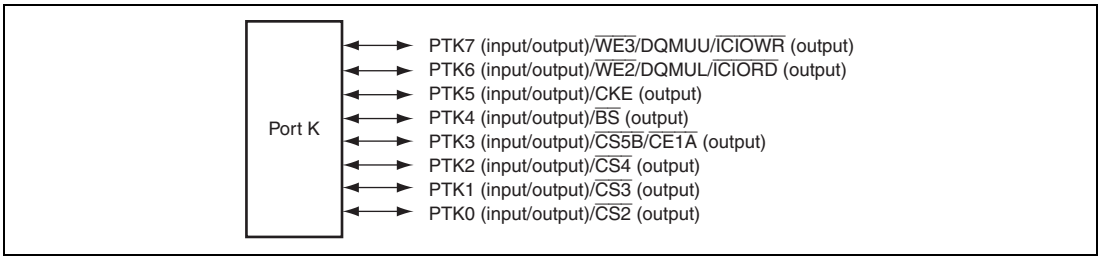


Figure 28.10 Port K

28.11.1 Port K Data Register (PKDR)

PKDR is a register that stores data for pins PTK7 to PTK0. Bits PK7DT to PK0DT correspond to pins PTK7 to PTK0. For pins that function as general-purpose output pins, a read operation directly reads out the value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	PK7DT	PK6DT	PK5DT	PK4DT	PK3DT	PK2DT	PK1DT	PK0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PK7DT	0	R/W	Table 28.12 shows the function of PKDR.
6	PK6DT	0	R/W	
5	PK5DT	0	R/W	
4	PK4DT	0	R/W	
3	PK3DT	0	R/W	
2	PK2DT	0	R/W	
1	PK1DT	0	R/W	
0	PK0DT	0	R/W	

Table 28.12 Port K Data Register (PKDR) Read/Write Operations

PKCR State				
PKnMD1	**PKnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PKDR value	The value is written to PKDR, but does not affect the pin state.
	1	Output	PKDR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PKDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PKDR, but does not affect the pin state.

Note: n = 7 to 0

28.12 Port L

Port L is an input/output port with the pin configuration shown in figure 28.11. Each pin has an input pull-up MOS, which is controlled by the port L control register (PLCR) in the PFC.

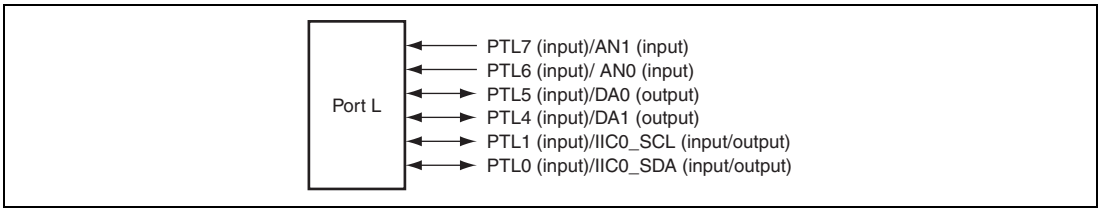


Figure 28.11 Port L

28.12.1 Port L Data Register (PLDR)

PLDR is a register that stores data for pins PTL7 to PTL4, PTL1, and PTL0. Bits PL7DT to PL4DT, PL1DT, and PL0DT correspond to pins PTL7 to PTL4, PTL1, and PTL0. For pins that function as general-purpose output pins, a read operation directly reads out the value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	PL7DT	PL6DT	PL5DT	PL4DT	—	—	PL1DT	PL0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	PL7DT	0	R	Table 28.13 shows the function of PLDR.
6	PL6DT	0	R	
5	PL5DT	0	R	
4	PL4DT	0	R	
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PL1DT	0	R	Table 28.13 shows the function of PLDR.
0	PL0DT	0	R	

Table 28.13 Port L Data Register (PLDR) Read/Write Operations

- PL0DT, PL1DT, PL4DT to PL7DT

PLCR State				
PLnMD1	**PLnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PLDR value	The value is written to PLDR, but does not affect the pin state.
	1	—	—	—
1	0	Input	Pin state	The value is written to PLDR, but does not affect the pin state.
	1	—	—	—

Note: n = 0, 1, 4 to 7

28.13 Port M

Port M is an input/output port with the pin configuration shown in figure 28.12. The PTM3 pin has an input pull-up MOS, which is controlled by the port M control register (PMCR) in the PFC.

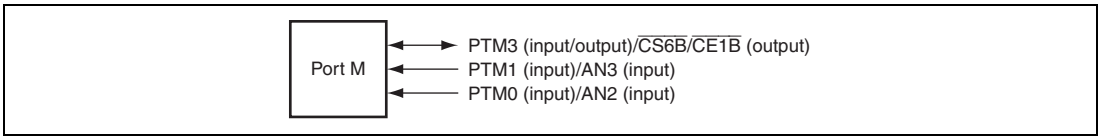


Figure 28.12 Port M

28.13.1 Port M Data Register (PMDR)

PMDR is a register that stores data for pins PTM3, PTM1, and PTM0. Bits PM3DT, PM1DT, and PM0DT correspond to pins PTM7 to PTM0. For pins that function as general-purpose output pins, a read operation directly reads out the value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	PM3DT	—	PM1DT	PM0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	PM3DT	0	R/W	Table 28.14 shows the function of PMDR.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	PM1DT	0	R	Table 28.14 shows the function of PMDR.
0	PM0DT	0	R	

Table 28.14 Port M Data Register (PMDR) Read/Write Operations

- PM3DT

PMCR State				
PMnMD1	**PMnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PMDR value	The value is written to PMDR, but does not affect the pin state.
	1	Output	PMDR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PMDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PMDR, but does not affect the pin state.

Note: n = 3

- PM0DT, PM1DT

PMCR State				
PMnMD1	**PMnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PMDR value	The value is written to PMDR, but does not affect the pin state.
	1	—	—	—
1	0	Input	Pin state	The value is written to PMDR, but does not affect the pin state.
	1	—	—	—

Note: n = 0, 1

28.14 Port N

Port N is an input/output port with the pin configuration shown in figure 28.13. Pins other than the PTN3 pin have an input pull-up MOS, which is controlled by the port N control register (PNCR) in the PFC.

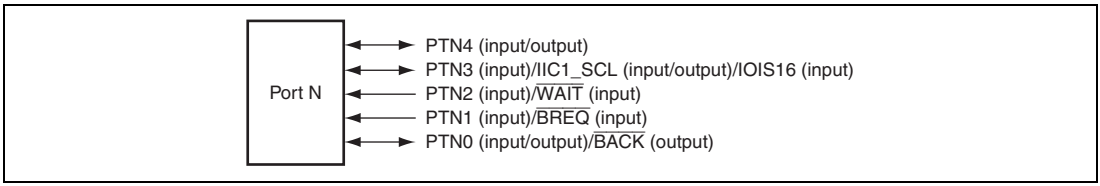


Figure 28.13 Port N

28.14.1 Port N Data Register (PNDR)

PNDR is a register that stores data for pins PTN4 to PTN0. Bits PN4DT to PN0DT correspond to pins PTN4 to PTN0. For pins that function as general-purpose output pins, a read operation directly reads out the value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	PN4DT	PN3DT	PN2DT	PN1DT	PN0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PN4DT	0	R/W	Table 28.15 shows the function of PNDR.
3	PN3DT	0	R	
2	PN2DT	0	R	
1	PN1DT	0	R	
0	PN0DT	0	R/W	

Table 28.15 Port N Data Register (PNDR) Read/Write Operations

- PN0DT

PNCR State				
PNnMD1	**PNnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PNDR value	The value is written to PNDR, but does not affect the pin state.
	1	Output	PNDR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PNDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PNDR, but does not affect the pin state.

Note: n = 0

- PN1DT, PN2DT

PNCR State				
PNnMD1	**PNnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PNDR value	The value is written to PNDR, but does not affect the pin state.
	1	—	—	—
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PNDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PNDR, but does not affect the pin state.

Note: n = 1, 2

- PN3DT

PNCR State				
PNnMD1	**PNnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PNDR value	The value is written to PNDR, but does not affect the pin state.
	1	—	—	—
1	0	Input	Pin state	The value is written to PNDR, but does not affect the pin state.
	1	—	—	—

Note: n = 3

- PN4DT

PNCR State				
PNnMD1	**PNnMD0**	**Pin State**	**Read**	**Write**
0	0	—	—	—
	1	Output	PNDR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PNDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PNDR, but does not affect the pin state.

Note: n = 4

28.15 Port Q

Port Q is an input/output port with the pin configuration shown in figure 28.14. Each pin has an input pull-up MOS, which is controlled by the port Q control register (PQCR) in the PFC.

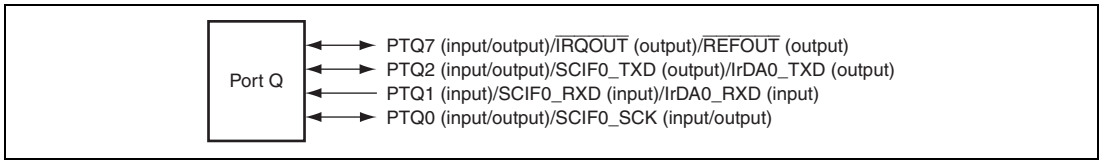


Figure 28.14 Port Q

28.15.1 Port Q Data Register (PQDR)

PQDR is a register that stores data for pins PTQ7, PTQ2 to PTQ0. Bits PQ7DT, PQ2DT to PQ0DT correspond to pins PTQ7, PTQ2 to PTQ0. For pins that function as general-purpose output pins, a read operation directly reads out the value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	PQ7DT	—	—	—	—	PQ2DT	PQ1DT	PQ0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PQ7DT	0	R/W	Table 28.16 shows the function of PQDR.
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	PQ2DT	0	R/W	Table 28.16 shows the function of PQDR.
1	PQ1DT	0	R	
0	PQ0DT	0	R/W	

Table 28.16 Port Q Data Register (PQDR) Read/Write Operations

- PQ0DT, PQ2DT, PQ7DT

PQCR State				
PQnMD1	**PQnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PQDR value	The value is written to PQDR, but does not affect the pin state.
	1	Output	PQDR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PQDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PQDR, but does not affect the pin state.

Note: n = 0, 2, 7

- PQ1DT

PQCR State				
PQnMD1	**PQnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PQDR value	The value is written to PQDR, but does not affect the pin state.
	1	—	—	—
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PQDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PQDR, but does not affect the pin state.

Note: n = 1

28.16 Port R

Port R is an input/output port with the pin configuration shown in figure 28.15. Each pin has an input pull-up MOS, which is controlled by the port R control register (PRCR) in the PFC.

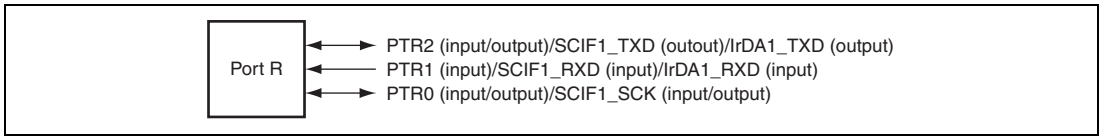


Figure 28.15 Port R

28.16.1 Port R Data Register (PRDR)

PRDR is a register that stores data for pins PTR2 to PTR0. Bits PR2DT to PR0DT correspond to pins PTR2 to PTR0. For pins that function as general-purpose output pins, a read operation directly reads out the value from this register. When the function is a general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	PR2DT	PR1DT	PR0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	PR2DT	0	R/W	Table 28.17 shows the function of PRDR.
1	PR1DT	0	R	
0	PR0DT	0	R/W	

Table 28.17 Port R Data Register (PRDR) Read/Write Operations

- PR0DT, PR2DT

PRCR State				
PRnMD1	**PRnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PRDR value	The value is written to PRDR, but does not affect the pin state.
	1	Output	PRDR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PRDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PRDR, but does not affect the pin state.

Note: n = 0, 2

- PR1DT

PRCR State				
PRnMD1	**PRnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PRDR value	The value is written to PRDR, but does not affect the pin state.
	1	—	—	—
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PRDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PRDR, but does not affect the pin state.

Note: n = 1

28.17 Port S

Port S is an input/output port with the pin configuration shown in figure 28.16. Each pin has an input pull-up MOS, which is controlled by the port S control register (PSCR) in the PFC.

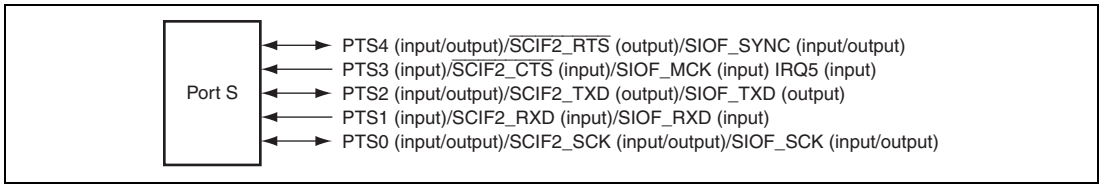


Figure 28.16 Port S

28.17.1 Port S Data Register (PSDR)

PSDR is a register that stores data for pins PTS4 to PTS0. Bits PS4DT to PS0DT correspond to pins PTS4 to PTS0. For pins that function as general-purpose output pins, a read operation directly reads out the value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	PS4DT	PS3DT	PS2DT	PS1DT	PS0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PS4DT	0	R/W	Table 28.18 shows the function of PSDR.
3	PS3DT	0	R	
2	PS2DT	0	R/W	
1	PS1DT	0	R	
0	PS0DT	0	R/W	

Table 28.18 Port S Data Register (PSDR) Read/Write Operations

- PS0DT, PS2DT, PS4DT

PSCR State				
PSnMD1	**PSnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PSDR value	The value is written to PSDR, but does not affect the pin state.
	1	Output	PSDR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PSDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PSDR, but does not affect the pin state.

Note: n = 0, 2, 4

- PS1DT, PS3DT

PSCR State				
PSnMD1	**PSnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PSDR value	The value is written to PSDR, but does not affect the pin state.
	1	—	—	—
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PSDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PSDR, but does not affect the pin state.

Note: n = 1, 3

28.18 Port T

Port T is an input/output port with the pin configuration shown in figure 28.17. Each pin has an input pull-up MOS, which is controlled by the port T control register (PTCR) in the PFC.

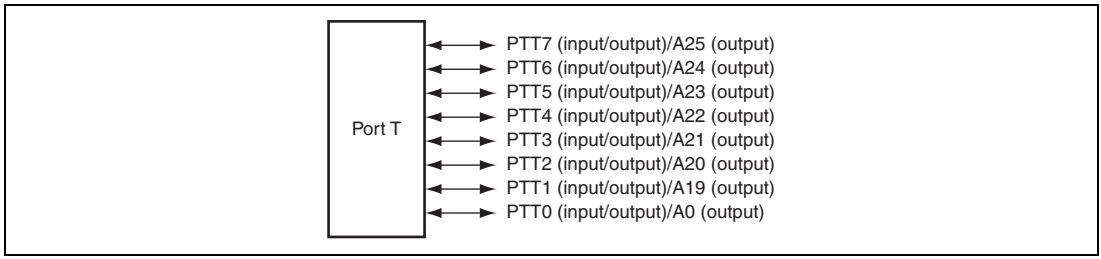


Figure 28.17 Port T

28.18.1 Port T Data Register (PTDR)

PTDR is a register that stores data for pins PTT7 to PTT0. Bits PT7DT to PT0DT correspond to pins PTT7 to PTT0. For pins that function as general-purpose output pins, a read operation directly reads out the value from this register. For pins that function as general-purpose input pins, a read operation reads out the level on the corresponding pin.

Bit:	7	6	5	4	3	2	1	0
	PT7DT	PT6DT	PT5DT	PT4DT	PT3DT	PT2DT	PT1DT	PT0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PT7DT	0	R/W	Table 28.19 shows the function of PTDR.
6	PT6DT	0	R/W	
5	PT5DT	0	R/W	
4	PT4DT	0	R/W	
3	PT3DT	0	R/W	
2	PT2DT	0	R/W	
1	PT1DT	0	R/W	
0	PT0DT	0	R/W	

Table 28.19 Port T Data Register (PTDR) Read/Write Operations

- PT0DT to PT7DT

PTCR State				
PTnMD1	**PTnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PTDR value	PTDR Value is written to PTDR, but does not affect the pin state.
	1	Output	PTDR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PTDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PTDR, but does not affect the pin state.

Note: n = 7 to 0

Section 29 Pin Function Controller (PFC)

29.1 Overview

The pin function controller (PFC) consists of registers to select the functions and I/O directions of multiplexed pins. The pin functions and I/O directions can be individually selected for every pin regardless of the LSI operating mode. Table 29.1 lists the multiplexed pins of this LSI.

The hatched entries in the table are the pin functions that are selected immediately after a reset. Selection between the port function (function 4) and other functions (functions 1 to 3) are made in the port control registers. For how to select a pin function for the pins on which two or more of functions 1 to 3 are multiplexed, see the description of pin select registers (PSELA to PSELC) in this section.

Note: The settings of the I/O buffer Hi-Z control registers A to F take priority over those of the port control registers.

Table 29.1 Multiplexed Pins

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
A	D23 input/output (BSC)	—	—	PTA7 input/output (port)
A	D22 input/output (BSC)	—	—	PTA6 input/output (port)
A	D21 input/output (BSC)	—	—	PTA5 input/output (port)
A	D20 input/output (BSC)	—	—	PTA4 input/output (port)
A	D19 input/output (BSC)	—	—	PTA3 input/output (port)
A	D18 input/output (BSC)	—	—	PTA2 input/output (port)
A	D17 input/output (BSC)	—	—	PTA1 input/output (port)
A	D16 input/output (BSC)	—	—	PTA0 input/output (port)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
B	D31 input/output (BSC)	—	—	PTB7 input/output (port)
B	D30 input/output (BSC)	—	—	PTB6 input/output (port)
B	D29 input/output (BSC)	—	—	PTB5 input/output (port)
B	D28 input/output (BSC)	—	—	PTB4 input/output (port)
B	D27 input/output (BSC)	—	—	PTB3 input/output (port)
B	D26 input/output (BSC)	—	—	PTB2 input/output (port)
B	D25 input/output (BSC)	—	—	PTB1 input/output (port)
B	D24 input/output (BSC)	—	—	PTB0 input/output (port)
C	$\overline{\text{SCIF5\_RTS}}$ output (SCIF5)	—	PINTA7 input (INTC)	PTC7 input/output (port)
C	$\overline{\text{SCIF5\_CTS}}$ input (SCIF5)	—	PINTA6 input (INTC)	PTC6 input/output (port)
C	$\overline{\text{SCIF4\_RTS}}$ output (SCIF4)	—	PINTA5 input (INTC)	PTC5 input/output (port)
C	$\overline{\text{SCIF4\_CTS}}$ input (SCIF4)	—	PINTA4 input (INTC)	PTC4 input/output (port)
C	—	—	PINTA3 input (INTC)	PTC3 input/output (port)
C	$\overline{\text{SCIF3\_RTS}}$ output (SCIF3)	SIM_RST output (SIM)	PINTA2 input (INTC)	PTC2 input/output (port)
C	$\overline{\text{SCIF3\_CTS}}$ input (SCIF3)	—	PINTA1 input (INTC)	PTC1 input/output (port)
C	SCIF3_SCK input/output (SCIF3)	SIM_SCK output (SIM)	PINTA0 input (INTC)	PTC0 input/output (port)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
D	DACK1 output (DMAC)	—	—	PTD7 input/output (port)
D	DREQ1 input (DMAC)	—	—	PTD6 input (port)
D	DACK0 output (DMAC)	—	—	PTD5 input/output (port)
D	DREQ0 input (DMAC)	—	—	PTD4 input (port)
D	SCIF3_TXD output (SCIF3)	SIM_D input/output (SIM)	—	PTD3 input/output (port)
D	$\overline{\text{RESETOUT}}$ output (CPG)	—	—	PTD2 input/output (port)
D	SCIF5_TXD output (SCIF5)	—	—	PTD1 input/output (port)
D	SCIF4_TXD output (SCIF4)	—	—	PTD0 input/output (port)
E	SCIF3_RXD input (SCIF3)	—	—	PTE7 input/output (port)
E	SCIF4_RXD input (SCIF4)	—	—	PTE6 input (port)
E	$\overline{\text{CS6A}}$/$\overline{\text{CE2B}}$ output (BSC)	—	—	PTE5 input/output (port)
E	$\overline{\text{CS5A}}$/$\overline{\text{CE2A}}$ output (BSC)	—	—	PTE4 input/output (port)
E	SCIF5_RXD input (SCIF5)	—	—	PTE3 input (port)
E	SCIF4_SCK input/output (SCIF4)	—	—	PTE2 input/output (port)
E	SCIF5_SCK input/output (SCIF5)	—	—	PTE1 input/output (port)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
F	TPU0_TO3 output (TPU)	—	PINTB3 input (INTC)	PTF3 input/output (port)
F	TPU0_TO2 output (TPU)	—	PINTB2 input (INTC)	PTF2 input/output (port)
F	TPU0_TO1 output (TPU)	—	PINTB1 input (INTC)	PTF1 input/output (port)
F	TPU0_TO0 output (TPU)	—	PINTB0 input (INTC)	PTF0 input/output (port)
G	AUDCK output (AUD)	—	—	PTG5 input/output (port)
G	AUDSYNC output (AUD)	—	—	PTG4 input/output (port)
G	AUDATA3 output (AUD)	—	—	PTG3 input/output (port)
G	AUDATA2 output (AUD)	—	—	PTG2 input/output (port)
G	AUDATA1 output (AUD)	—	—	PTG1 input/output (port)
G	AUDATA0 output (AUD)	—	—	PTG0 input/output (port)
H	TPU1_TO0 output (TPU)	—	—	PTH7 input/output (port)
H	—	—	—	PTH6 input (port)
H	IIC1_SDA input/output (IIC1)	$\overline{\text{ADTRG}}$ input (ADC)	—	PTH5 input (port)
H	IRQ4 input (INTC)	—	—	PTH4 input (port)
H	IRQ3/$\overline{\text{IRL3}}$ input (INTC)	—	—	PTH3 input (port)
H	IRQ2/$\overline{\text{IRL2}}$ input (INTC)	—	—	PTH2 input (port)
H	IRQ1/$\overline{\text{IRL1}}$ input (INTC)	—	—	PTH1 input (port)
H	IRQ0/$\overline{\text{IRL0}}$ input (INTC)	—	—	PTH0 input (port)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
J	TPU1_TO1 output (TPU)	—	—	PT7 input/output (port)
J	STATUS0 output (CPG)	—	—	PTJ6 input/output (port)
J	IRQ7 input (INTC)	—	—	PTJ5 input (port)
J	IRQ6 input (INTC)	—	—	PTJ4 input (port)
J	TEND0 output (DMAC)	—	—	PTJ3 input/output (port)
J	$\overline{\text{CAS}}$ output (BSC)	—	—	PTJ2 input/output (port)
J	TEND1 output (DMAC)	—	—	PTJ1 input/output (port)
J	$\overline{\text{RAS}}$ output (BSC)	—	—	PTJ0 input/output (port)
K	$\overline{\text{WE3}}$/DQMUU/$\overline{\text{ICIOWR}}$ output (BSC)	—	—	PTK7 input/output (port)
K	$\overline{\text{WE2}}$/DQMUL/$\overline{\text{ICIORD}}$ output (BSC)	—	—	PTK6 input/output (port)
K	CKE output (BSC)	—	—	PTK5 input/output (port)
K	$\overline{\text{BS}}$ output (BSC)	—	—	PTK4 input/output (port)
K	$\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$ output (BSC)	—	—	PTK3 input/output (port)
K	$\overline{\text{CS4}}$ output (BSC)	—	—	PTK2 input/output (port)
K	$\overline{\text{CS3}}$ output (BSC)	—	—	PTK1 input/output (port)
K	$\overline{\text{CS2}}$ output (BSC)	—	—	PTK0 input/output (port)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
L	AN1 input (ADC)	—	—	PTL7 input (port)
L	AN0 input (ADC)	—	—	PTL6 input (port)
L	DA0 output (DAC)	—	—	PTL5 input (port)
L	DA1 output (DAC)	—	—	PTL4 input (port)
L	IIC0_SCL input/output (IIC0)	—	—	PTL1 input (port)
L	IIC0_SDA input/output (IIC0)	—	—	PTL0 input (port)
M	$\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$ output (BSC)	—	—	PTM3 input/output (port)
M	AN3 input (ADC)	—	—	PTM1 input (port)
M	AN2 input (ADC)	—	—	PTM0 input (port)
N	—	—	—	PTN4 input/output (port)
N	IIC1_SCL input/output (IIC0)	$\overline{\text{IOIS16}}$ input (BSC)	—	PTN3 input (port)
N	$\overline{\text{WAIT}}$ input (BSC)	—	—	PTN2 input (port)
N	$\overline{\text{BREQ}}$ input (BSC)	—	—	PTN1 input (port)
N	$\overline{\text{BACK}}$ output (BSC)	—	—	PTN0 input/output (port)
Q	$\overline{\text{IRQOUT}}$ output (INTC)	$\overline{\text{REFOUT}}$ output (BSC)	—	PTQ7 input/output (port)
Q	SCIF0_TXD output (SCIF0)	IrDA0_TXD output (IrDA0)	—	PTQ2 input/output (port)
Q	SCIF0_RXD input (SCIF0)	IrDA0_RXD input (IrDA0)	—	PTQ1 input (port)
Q	SCIF0_SCK input/output (SCIF0)	—	—	PTQ0 input/output (port)
R	SCIF1_TXD output (SCIF1)	IrDA1_TXD output (IrDA1)	—	PTR2 input/output (port)
R	SCIF1_RXD input (SCIF1)	IrDA1_RXD input (IrDA1)	—	PTR1 input (port)
R	SCIF1_SCK input/output (SCIF1)	—	—	PTR0 input/output (port)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
S	$\overline{\text{SCIF2\_RTS}}$ output (SCIF2)	SIOF_SYNC output (SIOF)	—	PTS4 input/output (port)
S	$\overline{\text{SCIF2\_CTS}}$ input (SCIF2)	SIOF_MCK input (SIOF)	IRQ5 input (INTC)	PTS3 input (port)
S	SCIF2_TXD output (SCIF2)	SIOF_TXD output (SIOF)	—	PTS2 input/output (port)
S	SCIF2_RXD input (SCIF2)	SIOF_RXD output (SIOF)	—	PTS1 input (port)
S	SCIF2_SCK input/output (SCIF2)	SIOF_SCK input/output (SIOF)	—	PTS0 input/output (port)
T	A25 output (BSC)	—	—	PTT7 input/output (port)
T	A24 output (BSC)	—	—	PTT6 input/output (port)
T	A23 output (BSC)	—	—	PTT5 input/output (port)
T	A22 output (BSC)	—	—	PTT4 input/output (port)
T	A21 output (BSC)	—	—	PTT3 input/output (port)
T	A20 output (BSC)	—	—	PTT2 input/output (port)
T	A19 output (BSC)	—	—	PTT1 input/output (port)
T	A0 output (BSC)	—	—	PTT0 input/output (port)

Note: The hatched entries in the table are the pin functions immediately after a reset.

29.2 Register Descriptions

Table 29.2 shows the PFC register configuration. Table 29.3 shows the register states in each operating mode.

Table 29.2 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
Port A control register	PACR	R/W	H'A4050000	16
Port B control register	PBCR	R/W	H'A4050002	16
Port C control register	PCCR	R/W	H'A4050004	16
Port D control register	PDCR	R/W	H'A4050006	16
Port E control register	PECR	R/W	H'A4050008	16
Port F control register	PFCR	R/W	H'A405000A	16
Port G control register	PGCR	R/W	H'A405000C	16
Port H control register	PHCR	R/W	H'A405000E	16
Port J control register	PJCR	R/W	H'A4050010	16
Port K control register	PKCR	R/W	H'A4050012	16
Port L control register	PLCR	R/W	H'A4050014	16
Port M control register	PMCR	R/W	H'A4050016	16
Port N control register	PNCR	R/W	H'A4050018	16
Port Q control register	PQCR	R/W	H'A405001A	16
Port R control register	PRCR	R/W	H'A405001C	16
Port S control register	PSCR	R/W	H'A405001E	16
Port T control register	PTCR	R/W	H'A4050020	16
Pin select register A	PSELA	R/W	H'A4050100	16
Pin select register B	PSELB	R/W	H'A4050102	16
Pin select register C	PSELC	R/W	H'A4050104	16
I/O buffer Hi-Z control register A	HIZCRA	R/W	H'A4050120	16
I/O buffer Hi-Z control register B	HIZCRB	R/W	H'A4050122	16
I/O buffer Hi-Z control register C	HIZCRC	R/W	H'A4050124	16
I/O buffer Hi-Z control register D	HIZCRD	R/W	H'A4050126	16
I/O buffer Hi-Z control register E	HIZCRE	R/W	H'A4050128	16
I/O buffer Hi-Z control register F	HIZCRF	R/W	H'A405012A	16

Register Name	Abbreviation	R/W	Address	Access Size
Pull-up/pull-down control register	PULCR	R/W	H'A405015E	16
PINT control register A	PINTCRA	R/W	H'A4050040	16
PINT control register B	PINTCRB	R/W	H'A4050042	16

Table 29.3 Register States in Each Operating Mode

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
PACR	Initialized	Retained	—	Retained
PBCR	Initialized	Retained	—	Retained
PCCR	Initialized	Retained	—	Retained
PDCR	Initialized	Retained	—	Retained
PECR	Initialized	Retained	—	Retained
PFCR	Initialized	Retained	—	Retained
PGCR	Initialized	Retained	—	Retained
PHCR	Initialized	Retained	—	Retained
PJCR	Initialized	Retained	—	Retained
PKCR	Initialized	Retained	—	Retained
PLCR	Initialized	Retained	—	Retained
PMCR	Initialized	Retained	—	Retained
PNCR	Initialized	Retained	—	Retained
PQCR	Initialized	Retained	—	Retained
PRCR	Initialized	Retained	—	Retained
PSCR	Initialized	Retained	—	Retained
PTCR	Initialized	Retained	—	Retained
PSELA	Initialized	Retained	—	Retained
PSELB	Initialized	Retained	—	Retained
PSELC	Initialized	Retained	—	Retained
HIZCRA	Initialized	Retained	—	Retained
HIZCRB	Initialized	Retained	—	Retained
HIZCRC	Initialized	Retained	—	Retained
HIZCRD	Initialized	Retained	—	Retained

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
HIZCRE	Initialized	Retained	—	Retained
HIZCRF	Initialized	Retained	—	Retained
PULCR	Initialized	Retained	—	Retained
PINTCRA	Initialized	Retained	—	Retained
PINTCRB	Initialized	Retained	—	Retained

29.2.1 Port A Control Register (PACR)

PACR is a 16-bit readable/writable register that selects pin functions and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA7MD[1:0]		PA6MD[1:0]		PA5MD[1:0]		PA4MD[1:0]		PA3MD[1:0]		PA2MD[1:0]		PA1MD[1:0]		PA0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PA7MD[1:0]	00	R/W	PA7 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
13, 12	PA6MD[1:0]	00	R/W	PA6 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
11, 10	PA5MD[1:0]	00	R/W	PA5 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PA4MD[1:0]	00	R/W	PA4 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7, 6	PA3MD[1:0]	00	R/W	PA3 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5, 4	PA2MD[1:0]	00	R/W	PA2 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3, 2	PA1MD[1:0]	00	R/W	PA1 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1, 0	PA0MD[1:0]	00	R/W	PA0 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

29.2.2 Port B Control Register (PBCR)

PBCR is a 16-bit readable/writable register that selects pin functions and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB7MD[1:0]		PB6MD[1:0]		PB5MD[1:0]		PB4MD[1:0]		PB3MD[1:0]		PB2MD[1:0]		PB1MD[1:0]		PB0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PB7MD[1:0]	00	R/W	PB7 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
13, 12	PB6MD[1:0]	00	R/W	PB6 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
11, 10	PB5MD[1:0]	00	R/W	PB5 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
9, 8	PB4MD[1:0]	00	R/W	PB4 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PB3MD[1:0]	00	R/W	PB3 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5, 4	PB2MD[1:0]	00	R/W	PB2 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3, 2	PB1MD[1:0]	00	R/W	PB1 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1, 0	PB0MD[1:0]	00	R/W	PB0 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

29.2.3 Port C Control Register (PCCR)

PCCR is a 16-bit readable/writable register that selects pin functions and input pull-up/pull-down MOS control. Whether the MOS is pull-up or pull-down is selected in the PINTCRA register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC7MD[1:0]		PC6MD[1:0]		PC5MD[1:0]		PC4MD[1:0]		PC3MD[1:0]		PC2MD[1:0]		PC1MD[1:0]		PC0MD[1:0]	
Initial value:	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PC7MD[1:0]	10	R/W	PC7 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up/pull-down MOS: On) 11: Port input (Pull-up/pull-down MOS: Off)
13, 12	PC6MD[1:0]	10	R/W	PC6 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up/pull-down MOS: On) 11: Port input (Pull-up/pull-down MOS: Off)
11, 10	PC5MD[1:0]	10	R/W	PC5 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up/pull-down MOS: On) 11: Port input (Pull-up/pull-down MOS: Off)
9, 8	PC4MD[1:0]	10	R/W	PC4 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up/pull-down MOS: On) 11: Port input (Pull-up/pull-down MOS: Off)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PC3MD[1:0]	10	R/W	PC3 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up/pull-down MOS: On) 11: Port input (Pull-up/pull-down MOS: Off)
5, 4	PC2MD[1:0]	10	R/W	PC2 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up/pull-down MOS: On) 11: Port input (Pull-up/pull-down MOS: Off)
3, 2	PC1MD[1:0]	10	R/W	PC1 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up/pull-down MOS: On) 11: Port input (Pull-up/pull-down MOS: Off)
1, 0	PC0MD[1:0]	10	R/W	PC0 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up/pull-down MOS: On) 11: Port input (Pull-up/pull-down MOS: Off)

29.2.4 Port D Control Register (PDCR)

PDCR is a 16-bit readable/writable register that selects pin functions and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD7MD[1:0]		PD6MD[1:0]		PD5MD[1:0]		PD4MD[1:0]		PD3MD[1:0]		PD2MD[1:0]		PD1MD[1:0]		PD0MD[1:0]	
Initial value:	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PD7MD[1:0]	10	R/W	PD7 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
13, 12	PD6MD[1:0]	10	R/W	PD6 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
11, 10	PD5MD[1:0]	10	R/W	PD5 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
9, 8	PD4MD[1:0]	10	R/W	PD4 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PD3MD[1:0]	10	R/W	PD3 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5, 4	PD2MD[1:0]	00	R/W	PD2 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3, 2	PD1MD[1:0]	10	R/W	PD1 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1, 0	PD0MD[1:0]	10	R/W	PD0 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

29.2.5 Port E Control Register (PECR)

PECR is a 16-bit readable/writable register that selects pin functions and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE7MD[1:0]		PE6MD[1:0]		PE5MD[1:0]		PE4MD[1:0]		PE3MD[1:0]		PE2MD[1:0]		PE1MD[1:0]		—	—
Initial value:	1	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PE7MD[1:0]	10	R/W	PE7 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
13, 12	PE6MD[1:0]	10	R/W	PE6 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
11, 10	PE5MD[1:0]	00	R/W	PE5 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
9, 8	PE4MD[1:0]	00	R/W	PE4 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PE3MD[1:0]	10	R/W	PE3 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5, 4	PE2MD[1:0]	10	R/W	PE2 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3, 2	PE1MD[1:0]	10	R/W	PE1 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

29.2.6 Port F Control Register (PFCR)

PFCR is a 16-bit readable/writable register that selects pin functions and input pull-up/pull-down MOS control. Whether the MOS is pull-up or pull-down is selected in the PINTCRB register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PF3MD[1:0]		PF2MD[1:0]		PF1MD[1:0]		PF0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7, 6	PF3MD[1:0]	10	R/W	PF3 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up/pull-down MOS: On) 11: Port input (Pull-up/pull-down MOS: Off)
5, 4	PF2MD[1:0]	10	R/W	PF2 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up/pull-down MOS: On) 11: Port input (Pull-up/pull-down MOS: Off)
3, 2	PF1MD[1:0]	10	R/W	PF1 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up/pull-down MOS: On) 11: Port input (Pull-up/pull-down MOS: Off)
1, 0	PF0MD[1:0]	10	R/W	PF0 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up/pull-down MOS: On) 11: Port input (Pull-up/pull-down MOS: Off)

29.2.7 Port G Control Register (PGCR)

PGCR is a 16-bit readable/writable register that selects pin functions and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PG5MD[1:0]		PG4MD[1:0]		PG3MD[1:0]		PG2MD[1:0]		PG1MD[1:0]		PG0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11, 10	PG5MD[1:0]	00	R/W	PG5 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
9, 8	PG4MD[1:0]	00	R/W	PG4 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7, 6	PG3MD[1:0]	00	R/W	PG3 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5, 4	PG2MD[1:0]	00	R/W	PG2 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

Bit	Bit Name	Initial Value	R/W	Description
3, 2	PG1MD[1:0]	00	R/W	PG1 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1, 0	PG0MD[1:0]	00	R/W	PG0 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

29.2.8 Port H Control Register (PHCR)

PHCR is a 16-bit readable/writable register that selects pin functions and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH7MD[1:0]		PH6MD[1:0]		PH5MD[1:0]		PH4MD[1:0]		PH3MD[1:0]		PH2MD[1:0]		PH1MD[1:0]		PH0MD[1:0]	
Initial value:	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PH7MD[1:0]	10	R/W	PH7 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
13, 12	PH6MD[1:0]	10	R/W	PH6 Mode 00: Setting prohibited 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

Bit	Bit Name	Initial Value	R/W	Description
11, 10	PH5MD[1:0]	10	R/W	PH5 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input 11: Setting prohibited
9, 8	PH4MD[1:0]	10	R/W	PH4 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7, 6	PH3MD[1:0]	10	R/W	PH3 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5, 4	PH2MD[1:0]	10	R/W	PH2 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3, 2	PH1MD[1:0]	10	R/W	PH1 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1, 0	PH0MD[1:0]	10	R/W	PH0 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

29.2.9 Port J Control Register (PJCR)

PJCR is a 16-bit readable/writable register that selects pin functions and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PJ7MD[1:0]		PJ6MD[1:0]		PJ5MD[1:0]		PJ4MD[1:0]		PJ3MD[1:0]		PJ2MD[1:0]		PJ1MD[1:0]		PJ0MD[1:0]	
Initial value:	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0
R/W:	R/W	RW	RW	RW	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PJ7MD[1:0]	10	R/W	PJ7 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
13, 12	PJ6MD[1:0]	00	R/W	PJ6 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
11, 10	PJ5MD[1:0]	10	R/W	PJ5 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
9, 8	PJ4MD[1:0]	10	R/W	PJ4 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PJ3MD[1:0]	10	R/W	PJ3 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5, 4	PJ2MD[1:0]	00	R/W	PJ2 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3, 2	PJ1MD[1:0]	10	R/W	PJ1 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1, 0	PJ0MD[1:0]	00	R/W	PJ0 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

29.2.10 Port K Control Register (PKCR)

PKCR is a 16-bit readable/writable register that selects pin functions and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PK7MD[1:0]		PK6MD[1:0]		PK5MD[1:0]		PK4MD[1:0]		PK3MD[1:0]		PK2MD[1:0]		PK1MD[1:0]		PK0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PK7MD[1:0]	00	R/W	PK7 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
13, 12	PK6MD[1:0]	00	R/W	PK6 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
11, 10	PK5MD[1:0]	00	R/W	PK5 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
9, 8	PK4MD[1:0]	00	R/W	PK4 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PK3MD[1:0]	00	R/W	PK3 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5, 4	PK2MD[1:0]	00	R/W	PK2 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3, 2	PK1MD[1:0]	00	R/W	PK1 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1, 0	PK0MD[1:0]	00	R/W	PK0 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

29.2.11 Port L Control Register (PLCR)

PLCR is a 16-bit readable/writable register that selects pin functions.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PL7MD[1:0]		PL6MD[1:0]		PL5MD[1:0]		PL4MD[1:0]		—	—	—	—	PL1MD[1:0]		PL0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PL7MD[1:0]	00	R/W	PL7 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input 11: Setting prohibited
13, 12	PL6MD[1:0]	00	R/W	PL6 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input 11: Setting prohibited
11, 10	PL5MD[1:0]	00	R/W	PL5 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input 11: Setting prohibited
9, 8	PL4MD[1:0]	00	R/W	PL4 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input 11: Setting prohibited
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3, 2	PL1MD[1:0]	10	R/W	PL1 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input 11: Setting prohibited
1, 0	PL0MD[1:0]	10	R/W	PL0 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input 11: Setting prohibited

29.2.12 Port M Control Register (PMCR)

PMCR is a 16-bit readable/writable register that selects pin functions and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PM3MD[1:0]		—	—	PM1MD[1:0]		PM0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7, 6	PM3MD[1:0]	00	R/W	PM3 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

Bit	Bit Name	Initial Value	R/W	Description
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3, 2	PM1MD[1:0]	00	R/W	PM1 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input 11: Setting prohibited
1, 0	PM0MD[1:0]	00	R/W	PM0 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input 11: Setting prohibited

29.2.13 Port N Control Register (PNCR)

PNCR is a 16-bit readable/writable register that selects pin functions and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PN4MD[1:0]		PN3MD[1:0]		PN2MD[1:0]		PN1MD[1:0]		PN0MD[1:0]	
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PN4MD[1:0]	10	R/W	PN4 Mode 00: Setting prohibited 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7, 6	PN3MD[1:0]	10	R/W	PN3 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input 11: Setting prohibited
5, 4	PN2MD[1:0]	00	R/W	PN2 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3, 2	PN1MD[1:0]	00	R/W	PN1 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1, 0	PN0MD[1:0]	00	R/W	PN0 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

29.2.14 Port Q Control Register (PQCR)

PQCR is a 16-bit readable/writable register that selects pin functions and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PQ7MD[1:0]		—	—	—	—	—	—	—	—	PQ2MD[1:0]		PQ1MD[1:0]		PQ0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
R/W:	R/W	R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PQ7MD[1:0]	00	R/W	PQ7 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
13 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PQ2MD[1:0]	10	R/W	PQ2 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3, 2	PQ1MD[1:0]	10	R/W	PQ1 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1, 0	PQ0MD[1:0]	10	R/W	PQ0 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

29.2.15 Port R Control Register (PRCR)

PRCR is a 16-bit readable/writable register that selects pin functions and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	PR2MD[1:0]		PR1MD[1:0]		PR0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PR2MD[1:0]	10	R/W	PR2 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3, 2	PR1MD[1:0]	10	R/W	PR1 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1, 0	PR0MD[1:0]	10	R/W	PR0 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

29.2.16 Port S Control Register (PSCR)

PSCR is a 16-bit readable/writable register that selects pin functions and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PS4MD[1:0]		PS3MD[1:0]		PS2MD[1:0]		PS1MD[1:0]		PS0MD[1:0]	
Initial value:	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PS4MD[1:0]	10	R/W	PS4 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7, 6	PS3MD[1:0]	10	R/W	PS3 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5, 4	PS2MD[1:0]	10	R/W	PS2 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3, 2	PS1MD[1:0]	10	R/W	PS1 Mode 00: Other functions (See table 29.1) 01: Setting prohibited 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PS0MD[1:0]	10	R/W	PS0 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

29.2.17 Port T Control Register (PTCR)

PTCR is a 16-bit readable/writable register that selects pin functions and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PT7MD[1:0]		PT6MD[1:0]		PT5MD[1:0]		PT4MD[1:0]		PT3MD[1:0]		PT2MD[1:0]		PT1MD[1:0]		PT0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PT7MD[1:0]	00	R/W	PT7 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
13, 12	PT6MD[1:0]	00	R/W	PT6 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
11, 10	PT5MD[1:0]	00	R/W	PT5 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PT4MD[1:0]	00	R/W	PT4 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7, 6	PT3MD[1:0]	00	R/W	PT3 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5, 4	PT2MD[1:0]	00	R/W	PT2 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3, 2	PT1MD[1:0]	00	R/W	PT1 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1, 0	PT0MD[1:0]	00	R/W	PT0 Mode 00: Other functions (See table 29.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

29.2.18 Pin Select Register A (PSELA)

PSELA is a 16-bit readable/writable register that selects the pin function for the pins on which two or more "other functions" are multiplexed.

To use one of the two or more "other functions" that are multiplexed on a pin, first set the corresponding bit in PSELA and then set the port control register to select "other functions".

Setting example: To use the SIM_RST function of the PTC2/$\overline{\text{SCIF3\_RTS}}$/SIM_RST/PINTA2 pin

1. Write B'10 to the PSA[3:2] bits in PSELA.

2. Set bits PC2MD[1:0] in the port C control register (PCCR) to B'00 (other functions).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSA[15:14]		—	—	PSA11	PSA10	PSA9	PSA8	PSA7	PSA6	PSA5	PSA4	PSA[3:2]		PSA1	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PSA[15:14]	00	R/W	Selects $\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ 00: Selects $\overline{\text{IRQOUT}}$ 01: Selects $\overline{\text{REFOUT}}$ 10: Selects the logical OR of $\overline{\text{IRQOUT}}$ and $\overline{\text{REFOUT}}$ 11: Selects the logical OR of $\overline{\text{IRQOUT}}$ and $\overline{\text{REFOUT}}$
13, 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	PSA11	0	R/W	Selects TPU0_TO3/PINTB3 0: Selects TPU0_TO3 1: Selects PINTB3
10	PSA10	0	R/W	Selects TPU0_TO2/PINTB2 0: Selects TPU0_TO2 1: Selects PINTB2
9	PSA9	0	R/W	Selects TPU0_TO1/PINTB1 0: Selects TPU0_TO1 1: Selects PINTB1

Bit	Bit Name	Initial Value	R/W	Description
8	PSA8	0	R/W	Selects TPU0_TO0/PINTB0 0: Selects TPU0_TO0 1: Selects PINTB0
7	PSA7	0	R/W	Selects $\overline{\text{SCIF5\_RTS}}$/PINTA7 0: Selects $\overline{\text{SCIF5\_RTS}}$ 1: Selects PINTA7
6	PSA6	0	R/W	Selects $\overline{\text{SCIF5\_CTS}}$/PINTA6 0: Selects $\overline{\text{SCIF5\_CTS}}$ 1: Selects PINTA6
5	PSA5	0	R/W	Selects $\overline{\text{SCIF4\_RTS}}$/PINTA5 0: Selects $\overline{\text{SCIF4\_RTS}}$ 1: Selects PINTA5
4	PSA4	0	R/W	$\overline{\text{SCIF4\_CTS}}$/PINTA4 0: $\overline{\text{SCIF4\_CTS}}$ 1: PINTA4
3, 2	PSA[3:2]	00	R/W	Selects $\overline{\text{SCIF3\_RTS}}$/SIM_RST/PINTA2 00: Selects $\overline{\text{SCIF3\_RTS}}$ 01: Selects PINTA2 10: Selects SIM_RST 11: Setting prohibited
1	PSA1	0	R/W	Selects $\overline{\text{SCIF3\_CTS}}$/PINTA1 0: Selects $\overline{\text{SCIF3\_CTS}}$ 1: Selects PINTA1
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

29.2.19 Pin Select Register B (PSELB)

PSELB is a 16-bit readable/writable register that selects the pin function for the pins on which two or more "other functions" are multiplexed.

To use one of the two or more "other functions" that are multiplexed on a pin, first set the corresponding bit in PSELB and then set the port control register to select "other functions".

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSB[15:14]		PSB13	PSB12	PSB11	PSB10	PSB9	PSB8	PSB7	PSB6	PSB[5:4]		PSB3	PSB2	PSB1	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PSB[15:14]	00	R/W	Selects SCIF3_SCK/SIM_SCK/PINTA0 00: Selects SCIF3_SCK 01: Selects PINTA0 10: Selects SIM_SCK 11: Setting prohibited
13	PSB13	0	R/W	Selects SCIF3_TXD/SIM_D 0: Selects SCIF3_TXD 1: Selects SIM_D
12	PSB12	0	R/W	Selects IIC1_SCL/$\overline{\text{IOIS16}}$ 0: Selects IIC1_SCL 1: Selects $\overline{\text{IOIS16}}$
11	PSB11	0	R/W	Selects IIC1_SDA/$\overline{\text{ADTRG}}$ 0:Selects IIC1_SDA 1: Selects $\overline{\text{ADTRG}}$
10	PSB10	0	R/W	Selects SCIF1_TXD/IrDA1_TXD 0: Selects SCIF1_TXD 1: Selects IrDA1_TXD
9	PSB9	0	R/W	Selects SCIF1_RXD/IrDA1_RXD 0: Selects SCIF1_RXD 1: Selects IrDA1_RXD

Bit	Bit Name	Initial Value	R/W	Description
8	PSB8	0	R/W	Selects SCIF0_TXD/IrDA0_TXD 0: Selects SCIF0_TXD 1: Selects IrDA0_TXD
7	PSB7	0	R/W	Selects SCIF0_RXD/IrDA0_RXD 0: Selects SCIF0_RXD 1: Selects IrDA0_RXD
6	PSB6	0	R/W	Selects $\overline{\text{SCIF2\_RTS}}$/SIOF_SYNC 0: Selects $\overline{\text{SCIF2\_RTS}}$ 1: Selects SIOF_SYNC
5, 4	PSB[5:4]	00	R/W	Selects $\overline{\text{SCIF2\_CTS}}$/SIOF_MCK/IRQ5 00: Selects $\overline{\text{SCIF2\_CTS}}$ 01: Selects IRQ5 10: Selects SIOF_MCK 11: Setting prohibited
3	PSB3	0	R/W	Selects SCIF2_TXD/SIOF_TXD 0: Selects SCIF2_TXD 1: Selects SIOF_TXD
2	PSB2	0	R/W	Selects SCIF2_RXD/SIOF_RXD 0: Selects SCIF2_RXD 1: Selects SIOF_RXD
1	PSB1	0	R/W	Selects SCIF2_SCK/SIOF_SCK 0: Selects SCIF2_SCK 1: Selects SIOF_SCK
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

29.2.20 Pin Select Register C (PSELC)

PESELC is a 16-bit readable/writable register that selects the polarity for the PINTA7 to PINTA0 and PINTB3 to PINTB0 functions.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PSC11	PSC10	PSC9	PSC8	PSC7	PSC6	PSC5	PSC4	PSC3	PSC2	PSC1	PSC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	PSC11	0	R/W	Selects input polarity of PINTB3 function 0: When the pin is a high level, interrupt is accepted. 1: When the pin is a low level, interrupt is accepted.
10	PSC10	0	R/W	Selects input polarity of PINTB2 function 0: When the pin is a high level, interrupt is accepted. 1: When the pin is a low level, interrupt is accepted.
9	PSC9	0	R/W	Selects input polarity of PINTB1 function 0: When the pin is a high level, interrupt is accepted. 1: When the pin is a low level, interrupt is accepted.
8	PSC8	0	R/W	Selects input polarity of PINTB0 function 0: When the pin is a high level, interrupt is accepted. 1: When the pin is a low level, interrupt is accepted.
7	PSC7	0	R/W	Selects input polarity of PINTA7 function 0: When the pin is a high level, interrupt is accepted. 1: When the pin is a low level, interrupt is accepted.
6	PSC6	0	R/W	Selects input polarity of PINTA6 function 0: When the pin is a high level, interrupt is accepted. 1: When the pin is a low level, interrupt is accepted.

Bit	Bit Name	Initial Value	R/W	Description
5	PSC5	0	R/W	Selects input polarity of PINTA5 function 0: When the pin is a high level, interrupt is accepted. 1: When the pin is a low level, interrupt is accepted.
4	PSC4	0	R/W	Selects input polarity of PINTA4 function 0: When the pin is a high level, interrupt is accepted. 1: When the pin is a low level, interrupt is accepted.
3	PSC3	0	R/W	Selects input polarity of PINTA3 function 0: When the pin is a high level, interrupt is accepted. 1: When the pin is a low level, interrupt is accepted.
2	PSC2	0	R/W	Selects input polarity of PINTA2 function 0: When the pin is a high level, interrupt is accepted. 1: When the pin is a low level, interrupt is accepted.
1	PSC1	0	R/W	Selects input polarity of PINTA1 function 0: When the pin is a high level, interrupt is accepted. 1: When the pin is a low level, interrupt is accepted.
0	PSC0	0	R/W	Selects input polarity of PINTA0 function 0: When the pin is a high level, interrupt is accepted. 1: When the pin is a low level, interrupt is accepted.

29.2.21 I/O Buffer Hi-Z Control Register A (HIZCRA)

HIZCRA is a 16-bit readable/writable register that controls the high impedance states of pins on a per-function basis.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HIZA15	HIZA14	HIZA13	HIZA12	HIZA11	HIZA10	HIZA9	HIZA8	HIZA7	HIZA6	HIZA5	HIZA4	HIZA3	HIZA2	HIZA1	HIZA0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	HIZA15	0	R/W	Controls high impedance of the CKE pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
14	HIZA14	0	R/W	Controls high impedance of the $\overline{\text{RDWR}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
13	HIZA13	0	R/W	Controls high impedance of the $\overline{\text{WE3}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
12	HIZA12	0	R/W	Controls high impedance of the $\overline{\text{WE2}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
11	HIZA11	0	R/W	Controls high impedance of the $\overline{\text{WE1}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
10	HIZA10	0	R/W	Controls high impedance of the $\overline{\text{WE0}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
9	HIZA9	0	R/W	Controls high impedance of the $\overline{\text{RD}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
8	HIZA8	0	R/W	Controls high impedance of the $\overline{\text{BS}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance

Bit	Bit Name	Initial Value	R/W	Description
7	HIZA7	0	R/W	Controls high impedance of the $\overline{\text{CS6A}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
6	HIZA6	0	R/W	Controls high impedance of the $\overline{\text{CS5A}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
5	HIZA5	0	R/W	Controls high impedance of the $\overline{\text{CS6B}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
4	HIZA4	0	R/W	Controls high impedance of the $\overline{\text{CS5B}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
3	HIZA3	0	R/W	Controls high impedance of the $\overline{\text{CS4}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
2	HIZA2	0	R/W	Controls high impedance of the $\overline{\text{CS3}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
1	HIZA1	0	R/W	Controls high impedance of the $\overline{\text{CS2}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
0	HIZA0	0	R/W	Controls high impedance of the $\overline{\text{CS0}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance

29.2.22 I/O Buffer Hi-Z Control Register B (HIZCRB)

HIZCRB is a 16-bit readable/writable register that controls high impedance states of pins on a per-function basis.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	HIZB6	HIZB5	HIZB4	HIZB3	HIZB2	HIZB1	HIZB0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	HIZB6	0	R/W	Controls high impedance of the D31 to D16 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
5	HIZB5	0	R/W	Controls high impedance of the A25 to A19 pins 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
4	HIZB4	0	R/W	Controls high impedance of the A0 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
3	HIZB3	0	R/W	Controls high impedance of the $\overline{\text{BREQ}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
2	HIZB2	0	R/W	Controls high impedance of the $\overline{\text{BACK}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance

Bit	Bit Name	Initial Value	R/W	Description
1	HIZB1	0	R/W	Controls high impedance of the $\overline{\text{CAS}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
0	HIZB0	0	R/W	Controls high impedance of the $\overline{\text{RAS}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance

29.2.23 I/O Buffer Hi-Z Control Register C (HIZCRC)

HIZCRC is a 16-bit readable/writable register that controls high impedance states of pins on a per-function basis.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	HIZC11	HIZC10	HIZC9	HIZC8	HIZC7	HIZC6	HIZC5	HIZC4	HIZC3	HIZC2	HIZC1	HIZC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	HIZC11	0	R/W	Controls high impedance of the PTF3 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
10	HIZC10	0	R/W	Controls high impedance of the PTF2 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance

Bit	Bit Name	Initial Value	R/W	Description
9	HIZC9	0	R/W	Controls high impedance of the PTF1 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
8	HIZC8	0	R/W	Controls high impedance of the PTF0 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
7	HIZC7	0	R/W	Controls high impedance of the PTC7 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
6	HIZC6	0	R/W	Controls high impedance of the PTC6 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
5	HIZC5	0	R/W	Controls high impedance of the PTC5 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
4	HIZC4	0	R/W	Controls high impedance of the PTC4 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
3	HIZC3	0	R/W	Controls high impedance of the PTC3 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
2	HIZC2	0	R/W	Controls high impedance of the PTC2 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance

Bit	Bit Name	Initial Value	R/W	Description
1	HIZC1	0	R/W	Controls high impedance of the PTC1 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
0	HIZC0	0	R/W	Controls high impedance of the PTC0 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance

29.2.24 I/O Buffer Hi-Z Control Register D (HIZCRD)

HIZCRD is a 16-bit readable/writable register that controls high impedance states of pins on a per-function basis.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	HIZD8	HIZD7	HIZD6	HIZD5	HIZD4	HIZD3	HIZD2	HIZD1	HIZD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	HIZD8	0	R/W	Controls high impedance of the $\overline{\text{IRQOUT}}$ pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
7	HIZD7	0	R/W	Controls high impedance of the PTJ5 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance

Bit	Bit Name	Initial Value	R/W	Description
6	HIZD6	0	R/W	Controls high impedance of the PTJ4 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
5	HIZD5	0	R/W	Controls high impedance of the PTS3 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
4	HIZD4	0	R/W	Controls high impedance of the PTH4 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
3	HIZD3	0	R/W	Controls high impedance of the PTH3 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
2	HIZD2	0	R/W	Controls high impedance of the PTH2 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
1	HIZD1	0	R/W	Controls high impedance of the PTH1 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
0	HIZD0	0	R/W	Controls high impedance of the PTH0 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance

29.2.25 I/O Buffer Hi-Z Control Register E (HIZCRE)

HIZCRE is a 16-bit readable/writable register that controls high impedance states of pins on a per-function basis.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HIZE15	HIZE14	HIZE13	HIZE12	HIZE11	HIZE10	HIZE9	HIZE8	HIZE7	HIZE6	HIZE5	HIZE4	HIZE3	HIZE2	HIZE1	HIZE0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	HIZE15	0	R/W	Controls high impedance of the PTJ7 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
14	HIZE14	0	R/W	Controls high impedance of the PTH7 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
13	HIZE13	0	R/W	Controls high impedance of PTJ1 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
12	HIZE12	0	R/W	Controls high impedance of the PTJ3 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
11	HIZE11	0	R/W	Controls high impedance of the PTD6 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
10	HIZE10	0	R/W	Controls high impedance of the PTD4 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance

Bit	Bit Name	Initial Value	R/W	Description
9	HIZE9	0	R/W	Controls high impedance of the PTD7 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
8	HIZE8	0	R/W	Controls high impedance of the PTD5 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
7	HIZE7	0	R/W	Controls high impedance of the PTD1/PTE3/PTE1 pins 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
6	HIZE6	0	R/W	Controls high impedance of the PTD0/PTE6/PTE2 pins 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
5	HIZE5	0	R/W	Controls high impedance of PTE7 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
4	HIZE4	0	R/W	Controls high impedance of PTD3 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
3	HIZE3	0	R/W	Controls high impedance of PTS4 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
2	HIZE2	0	R/W	Controls high impedance of PTS2/PTS1/PTS0 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance

Bit	Bit Name	Initial Value	R/W	Description
1	HIZE1	0	R/W	Controls high impedance of PTR2/PTR1/PTR0 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
0	HIZE0	0	R/W	Controls high impedance of PTQ2/PTQ1/PTQ0 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance

29.2.26 I/O Buffer Hi-Z Control Register F (HIZCRF)

HIZCRF is a 16-bit readable/writable register that controls high impedance states of pins on a per-function basis.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HIZF1	HIZF0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	HIZF1	0	R/W	Controls high impedance of the PTN4 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance
0	HIZF0	0	R/W	Controls high impedance of the PTH6 pin 0: I/O buffer operates normally 1: Input of the I/O buffer is disabled and output is high impedance

29.2.27 Pull-up/Pull-down Control Register (PULCR)

PULCR is a 16-bit readable/writable register that controls pull-up/pull-down of pins.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PUL1	PUL0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PUL1	0	R/W	Controls pulling up of the $\overline{\text{ASEBRK}}$ pin 0: Pull-up MOS is on 1: Pull-up MOS is off
0	PUL0	0	R/W	Controls pulling up of the $\overline{\text{TRST}}$ pin 0: Pull-up MOS is on 1: Pull-up MOS is off

29.2.28 PINT Control Register A (PINTCRA)

PINTACR is a 16-bit readable/writable register that enables pull-up/pull-down of PINTA7 to PINTA0 functions and selects between the two. The selection of pull-up/pull-down is effective not only for the PINT function but for all pin functions.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PINTA PEN7	PINTA PUD7	PINTA PEN6	PINTA PUD6	PINTA PEN5	PINTA PUD5	PINTA PEN4	PINTA PUD4	PINTA PEN3	PINTA PUD3	PINTA PEN2	PINTA PUD2	PINTA PEN1	PINTA PUD1	PINTA PEN0	PINTA PUD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PINTAPEN7	0	R/W	Controls PINTA7 function 0: Pull-up/pull-down is MOS on 1: Pull-up/pull-down is MOS off

Bit	Bit Name	Initial Value	R/W	Description
14	PINTAPUD7	0	R/W	Controls PINTA7 function 0: Selects pull-down 1: Selects pull-up
13	PINTAPEN6	0	R/W	Controls PINTA6 function 0: Pull-up/pull-down MOS is on 1: Pull-up/pull-down MOS is off
12	PINTAPUD6	0	R/W	Controls PINTA6 function 0: Selects pull-down 1: Selects pull-up
11	PINTAPEN5	0	R/W	Controls PINTA5 function 0: Pull-up/pull-down MOS is on 1: Pull-up/pull-down MOS is off
10	PINTAPUD5	0	R/W	Controls PINTA5 function 0: Selects pull-down 1: Selects pull-up
9	PINTAPEN4	0	R/W	Controls PINTA4 function 0: Pull-up/pull-down MOS is on 1: Pull-up/pull-down MOS is off
8	PINTAPUD4	0	R/W	Controls PINTA4 function 0: Selects pull-down 1: Selects pull-up
7	PINTAPEN3	0	R/W	Controls PINTA3 function 0: Pull-up/pull-down MOS is on 1: Pull-up/pull-down MOS is off
6	PINTAPUD3	0	R/W	Controls PINTA3 function 0: Selects pull-down 1: Selects pull-up
5	PINTAPEN2	0	R/W	Controls PINTA2 function 0: Pull-up/pull-down MOS is on 1: Pull-up/pull-down MOS is off

Bit	Bit Name	Initial Value	R/W	Description
4	PINTAPUD2	0	R/W	Controls PINTA2 function 0: Selects pull-down 1: Selects pull-up
3	PINTAPEN1	0	R/W	Controls PINTA1 function 0: Pull-up/pull-down MOS is on 1: Pull-up/pull-down MOS is off
2	PINTAPUD1	0	R/W	Controls PINTA1 function 0: Selects pull-down 1: Selects pull-up
1	PINTAPEN0	0	R/W	Controls PINTA0 function 0: Pull-up/pull-down MOS is on 1: Pull-up/pull-down MOS is off
0	PINTAPUD0	0	R/W	Controls PINTA0 function 0: Selects pull-down 1: Selects pull-up

29.2.29 PINT Control Register B (PINTCRB)

PINTBCR is a 16-bit readable/writable register that enables pull-up/pull-down of PINTB3 to PINTB0 functions and selects between the two. The selection of pull-up/pull-down is effective not only for the PINT function but for all pin functions.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PINTB PEN3	PINTB PUD3	PINTB PEN2	PINTB PUD2	PINTB PEN1	PINTB PUD1	PINTB PEN0	PINTB PUD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7	PINTBPEN3	0	R/W	Controls PINTB3 function 0: Pull-up/pull-down MOS on 1: Pull-up/pull-down MOS off
6	PINTBPUD3	0	R/W	Controls PINTB3 function 0: Selects pull-down 1: Selects pull-up
5	PINTBPEN2	0	R/W	Controls PINTB2 function 0: Pull-up/pull-down MOS on 1: Pull-up/pull-down MOS off
4	PINTBPUD2	0	R/W	Controls PINTB2 function 0: Selects pull-down 1: Selects pull-up
3	PINTBPEN1	0	R/W	Controls PINTB1 function 0: Pull-up/pull-down MOS on 1: Pull-up/pull-down MOS off
2	PINTBPUD1	0	R/W	Controls PINTB1 function 0: Selects pull-down 1: Selects pull-up
1	PINTBPEN0	0	R/W	Controls PINTB0 function 0: Pull-up/pull-down MOS on 1: Pull-up/pull-down MOS off
0	PINTBPUD0	0	R/W	Controls PINTB0 function 0: Selects pull-down 1: Selects pull-up

Section 30 User Break Controller (UBC)

The user break controller (UBC) provides versatile functions to facilitate program debugging. These functions help to ease creation of a self-monitor/debugger, which allows easy program debugging using this LSI alone, without using the in-circuit emulator. Various break conditions can be set in the UBC: instruction fetch or read/write access of an operand, operand size, data contents, address value, and program stop timing for instruction fetch.

30.1 Features

1. The following break conditions can be set.

   Break channels: Two (channels 0 and 1)

   User break conditions can be set independently for channels 0 and 1, and can also be set as a single sequential condition for the two channels, that is, a sequential break. (Sequential break involves two cases such that the channel 0 break condition is satisfied in a certain bus cycle and then the channel 1 break condition is satisfied in a different bus cycle, and vice versa.)

- Address

  When 40 bits containing ASID and 32-bit address are compared with the specified value, all the ASID bits can be compared or masked.

  32-bit address can be masked bit by bit, allowing the user to mask the address in desired page sizes such as lower 12 bits (4-Kbyte page) and lower 10 bits (1-Kbyte page).

- Data

  32 bits can be masked only for channel 1.

- Bus cycle

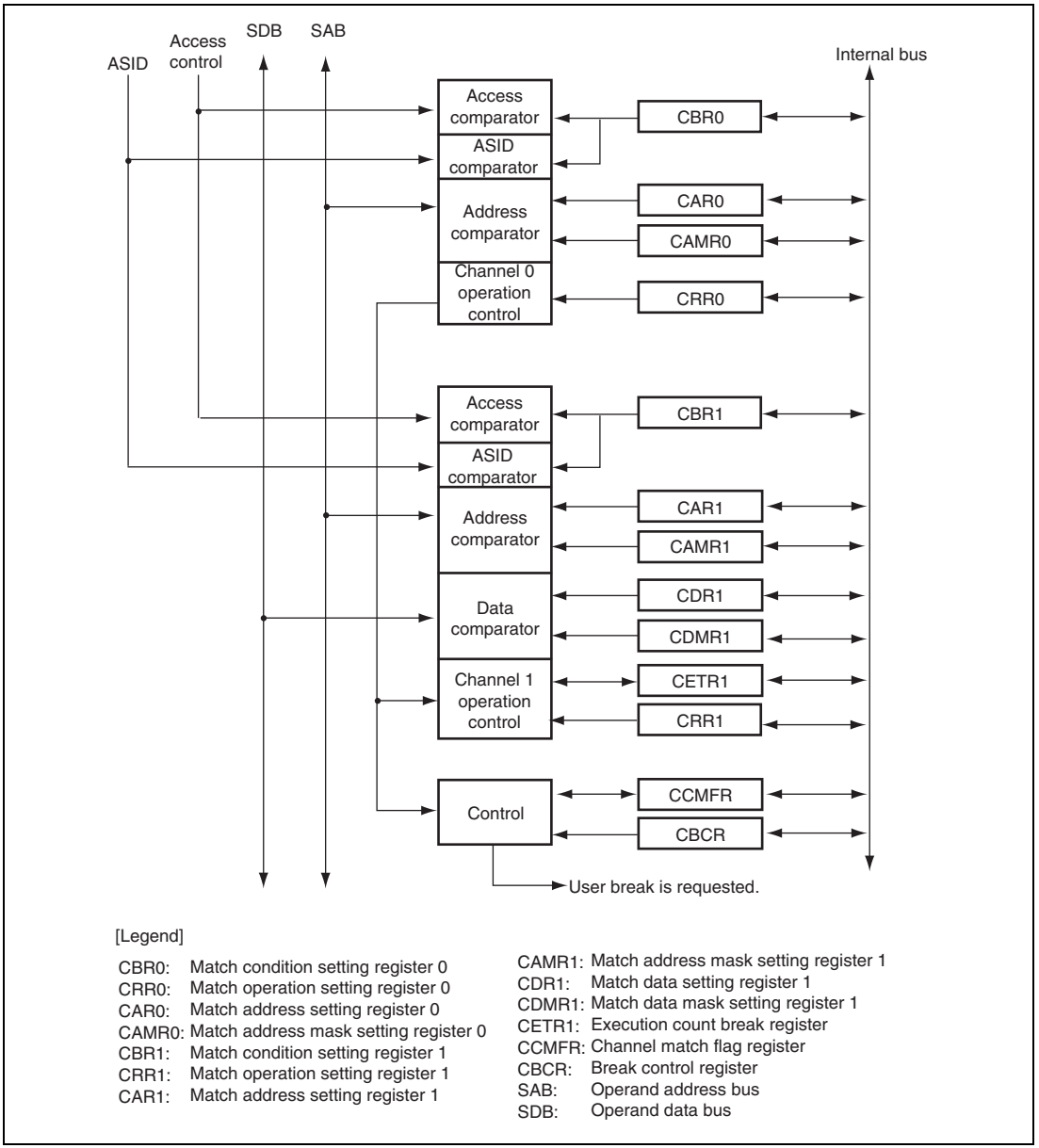
  The program can break either for instruction fetch (PC break) or operand access.

- Read or write access
- Operand sizes

  Byte, word, longword, and quadword are supported.

2. The user-designated exception handling routine for the user break condition can be executed.
3. Pre-instruction-execution or post-instruction-execution can be selected as the PC break timing.
4. A maximum of $2^{12} - 1$ repetition counts can be specified as the break condition (available only for channel 1).

Figure 30.1 shows the UBC block diagram.



[Legend]

CBR0: Match condition setting register 0
CRR0: Match operation setting register 0
CAR0: Match address setting register 0
CAMR0: Match address mask setting register 0
CBR1: Match condition setting register 1
CRR1: Match operation setting register 1
CAR1: Match address setting register 1
CAMR1: Match address mask setting register 1
CDR1: Match data setting register 1
CDMR1: Match data mask setting register 1
CETR1: Execution count break register
CCMFR: Channel match flag register
CBCR: Break control register
SAB: Operand address bus
SDB: Operand data bus

Figure 30.1 Block Diagram of UBC

30.2 Register Descriptions

The UBC has the following registers.

Table 30.1 Register Configuration

Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Access Size
Match condition setting register 0	CBR0	R/W	H'FF200000	H'1F200000	32
Match operation setting register 0	CRR0	R/W	H'FF200004	H'1F200004	32
Match address setting register 0	CAR0	R/W	H'FF200008	H'1F200008	32
Match address mask setting register 0	CAMR0	R/W	H'FF20000C	H'1F20000C	32
Match condition setting register 1	CBR1	R/W	H'FF200020	H'1F200020	32
Match operation setting register 1	CRR1	R/W	H'FF200024	H'1F200024	32
Match address setting register 1	CAR1	R/W	H'FF200028	H'1F200028	32
Match address mask setting register 1	CAMR1	R/W	H'FF20002C	H'1F20002C	32
Match data setting register 1	CDR1	R/W	H'FF200030	H'1F200030	32
Match data mask setting register 1	CDMR1	R/W	H'FF200034	H'1F200034	32
Execution count break register 1	CETR1	R/W	H'FF200038	H'1F200038	32
Channel match flag register	CCMFR	R/W	H'FF200600	H'1F200600	32
Break control register	CBCR	R/W	H'FF200620	H'1F200620	32

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 30.2 Register Status in Each Processing State

Register Name	Abbreviation	Power-on Reset	Manual Reset	Sleep	Standby
Match condition setting register 0	CBR0	H'20000000	Retained	Retained	Retained
Match operation setting register 0	CRR0	H'00002000	Retained	Retained	Retained
Match address setting register 0	CAR0	Undefined	Retained	Retained	Retained
Match address mask setting register 0	CAMR0	Undefined	Retained	Retained	Retained
Match condition setting register 1	CBR1	H'20000000	Retained	Retained	Retained
Match operation setting register 1	CRR1	H'00002000	Retained	Retained	Retained
Match address setting register 1	CAR1	Undefined	Retained	Retained	Retained
Match address mask setting register 1	CAMR1	Undefined	Retained	Retained	Retained
Match data setting register 1	CDR1	Undefined	Retained	Retained	Retained
Match data mask setting register 1	CDMR1	Undefined	Retained	Retained	Retained
Execution count break register 1	CETR1	Undefined	Retained	Retained	Retained
Channel match flag register	CCMFR	H'00000000	Retained	Retained	Retained
Break control register	CBCR	H'00000000	Retained	Retained	Retained

The access size must be the same as the control register size. If the size is different, the register is not written to if attempted, and reading the register returns the undefined value. A desired break may not occur between the time when the instruction for rewriting the control register is executed and the time when the written value is actually reflected on the register. In order to confirm the exact timing when the control register is updated, read the data which has been written most recently. The subsequent instructions are valid for the most recently written register value.

30.2.1 Match Condition Setting Registers 0 and 1 (CBR0 and CBR1)

CBR0 and CBR1 are readable/writable 32-bit registers which specify the break conditions for channels 0 and 1, respectively. The following break conditions can be set in the CBR0 and CBR1: (1) whether or not to include the match flag in the conditions, (2) whether or not to include the ASID, and the ASID value when included, (3) whether or not to include the data value, (4) operand size, (5) whether or not to include the execution count, (6) bus type, (7) instruction fetch cycle or operand access cycle, and (8) read or write access cycle.

- CBR0

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MFE	AIE	MFI						AIV							
Initial value :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	SZ			—	—	—	—	CD		ID		—	RW		CE
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	MFE	0	R/W	Match Flag Enable Specifies whether or not to include the match flag value specified by the MFI bit of this register in the match conditions. When the specified match flag value is 1, the condition is determined to be satisfied. 0: The match flag is not included in the match conditions; thus, not checked. 1: The match flag is included in the match conditions.
30	AIE	0	R/W	ASID Enable Specifies whether or not to include the ASID specified by the AIV bit of this register in the match conditions. 0: The ASID is not included in the match conditions; thus, not checked. 1: The ASID is included in the match conditions.

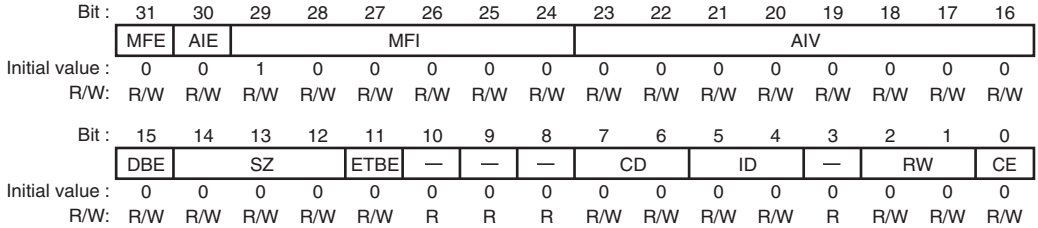
Bit	Bit Name	Initial Value	R/W	Description
29 to 24	MFI	100000	R/W	Match Flag Specify Specifies the match flag to be included in the match conditions. 000000: MF0 bit of the CCMFR register 000001: MF1 bit of the CCMFR register Others: Reserved (setting prohibited) Note: The initial value is the reserved value, but when 1 is written into CBR0[0], MFI must be set to 000000 or 000001. And note that the channel 0 is not hit when MFE bit of this register is 1 and MFI bits are 000000 in the condition of CCRMF.MF0 = 0.
23 to 16	AIV	All 0	R/W	ASID Specify Specifies the ASID value to be included in the match conditions.
15	—	0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
14 to 12	SZ	All 0	R/W	Operand Size Select Specifies the operand size to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 000: The operand size is not included in the match conditions; thus, not checked (any operand size specifies the match condition).*[1] 001: Byte access 010: Word access 011: Longword access 100: Quadword access*[2] Others: Reserved (setting prohibited)
11 to 8	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD	All 0	R/W	Bus Select Specifies the bus to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 00: Operand bus for operand access Others: Reserved (setting prohibited)
5, 4	ID	All 0	R/W	Instruction Fetch/Operand Access Select Specifies the instruction fetch cycle or operand access cycle as the match condition. 00: Instruction fetch cycle or operand access cycle 01: Instruction fetch cycle 10: Operand access cycle 11: Instruction fetch cycle or operand access cycle
3	—	0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
2, 1	RW	All 0	R/W	Bus Command Select Specifies the read/write cycle as the match condition. This bit is valid only when the operand access cycle is specified as a match condition. 00: Read cycle or write cycle 01: Read cycle 10: Write cycle 11: Read cycle or write cycle
0	CE	0	R/W	Channel Enable Validates/invalidates the channel. If this bit is 0, all the other bits of this register are invalid. 0: Invalidates the channel. 1: Validates the channel.

Notes: 1. If the data value is included in the match conditions, be sure to specify the operand size.

2. If the quadword access is specified and the data value is included in the match conditions, the upper and lower 32 bits of 64-bit data are each compared with the contents of both the match data setting register and the match data mask setting register.

- CBR1



Bit	Bit Name	Initial Value	R/W	Description
31	MFE	0	R/W	Match Flag Enable Specifies whether or not to include the match flag value specified by the MFI bit of this register in the match conditions. When the specified match flag value is 1, the condition is determined to be satisfied. 0: The match flag is not included in the match conditions; thus, not checked. 1: The match flag is included in the match conditions.
30	AIE	0	R/W	ASID Enable Specifies whether or not to include the ASID specified by the AIV bit of this register in the match conditions. 0: The ASID is not included in the match conditions; thus, not checked. 1: The ASID is included in the match conditions.
29 to 24	MFI	100000	R/W	Match Flag Specify Specifies the match flag to be included in the match conditions. 000000: The MF0 bit of the CCMFR register 000001: The MF1 bit of the CCMFR register Others: Reserved (setting prohibited) Note: The initial value is the reserved value, but when 1 is written into CBR1[0], MFI must be set to 000000 or 000001. And note that the channel 1 is not hit when MFE bit of this register is 1 and MFI bits are 000001 in the condition of CCRMF.MF1 = 0.

Bit	Bit Name	Initial Value	R/W	Description
23 to 16	AIV	All 0	R/W	ASID Specify Specifies the ASID value to be included in the match conditions.
15	DBE	0	R/W	Data Value Enable*3 Specifies whether or not to include the data value in the match condition. This bit is valid only when the operand access cycle is specified as a match condition. 0: The data value is not included in the match conditions; thus, not checked. 1: The data value is included in the match conditions.
14 to 12	SZ	All 0	R/W	Operand Size Select Specifies the operand size to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 000: The operand size is not included in the match condition; thus, not checked (any operand size specifies the match condition). *1 001: Byte access 010: Word access 011: Longword access 100: Quadword access*2 Others: Reserved (setting prohibited)
11	ETBE	0	R/W	Execution Count Value Enable Specifies whether or not to include the execution count value in the match conditions. If this bit is 1 and the match condition satisfaction count matches the value specified by the CETR1 register, the operation specified by the CRR1 register is performed. 0: The execution count value is not included in the match conditions; thus, not checked. 1: The execution count value is included in the match conditions.
10 to 8	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD	All 0	R/W	Bus Select Specifies the bus to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 00: Operand bus for operand access Others: Reserved (setting prohibited)
5, 4	ID	All 0	R/W	Instruction Fetch/Operand Access Select Specifies the instruction fetch cycle or operand access cycle as the match condition. 00: Instruction fetch cycle or operand access cycle 01: Instruction fetch cycle 10: Operand access cycle 11: Instruction fetch cycle or operand access cycle
3	—	0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
2, 1	RW	All 0	R/W	Bus Command Select Specifies the read/write cycle as the match condition. This bit is valid only when the operand access cycle is specified as a match condition. 00: Read cycle or write cycle 01: Read cycle 10: Write cycle 11: Read cycle or write cycle
0	CE	0	R/W	Channel Enable Validates/invalidates the channel. If this bit is 0, all the other bits in this register are invalid. 0: Invalidates the channel. 1: Validates the channel.

Notes: 1. If the data value is included in the match conditions, be sure to specify the operand size.

2. If the quadword access is specified and the data value is included in the match conditions, the upper and lower 32 bits of 64-bit data are each compared with the contents of both the match data setting register and the match data mask setting register.

3. The OCBI instruction is handled as longword write access without the data value, and the PREF, OCBP, and OCBWB instructions are handled as longword read access without the data value. Therefore, do not include the data value in the match conditions for these instructions.

30.2.2 Match Operation Setting Registers 0 and 1 (CRR0 and CRR1)

CRR0 and CRR1 are readable/writable 32-bit registers which specify the operation to be executed when channels 0 and 1 satisfy the match condition, respectively. The following operations can be set in the CRR0 and CRR1 registers: (1) breaking at a desired timing for the instruction fetch cycle and (2) requesting a break.

- CRR0

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCB	BIE
Initial value :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
13	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
12 to 2	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.

Bit	Bit Name	Initial Value	R/W	Description
1	PCB	0	R/W	PC Break Select Specifies either before or after instruction execution as the break timing for the instruction fetch cycle. This bit is invalid for breaks other than the ones for the instruction fetch cycle. 0: Sets the PC break before instruction execution. 1: Sets the PC break after instruction execution.
0	BIE	0	R/W	Break Enable Specifies whether or not to request a break when the match condition is satisfied for the channel. 0: Does not request a break. 1: Requests a break.

- CRR1

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCB	BIE
Initial value :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

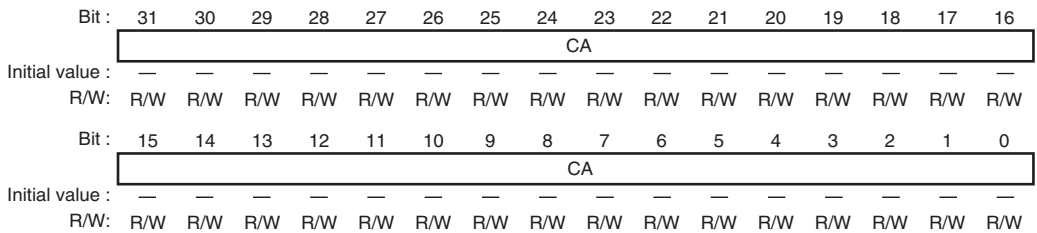
Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
13	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
12 to 2	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.

Bit	Bit Name	Initial Value	R/W	Description
1	PCB	0	R/W	PC Break Select Specifies either before or after instruction execution as the break timing for the instruction fetch cycle. This bit is invalid for breaks other than ones for the instruction fetch cycle. 0: Sets the PC break before instruction execution. 1: Sets the PC break after instruction execution.
0	BIE	0	R/W	Break Enable Specifies whether or not to request a break when the match condition is satisfied for the channel. 0: Does not request a break. 1: Requests a break.

30.2.3 Match Address Setting Registers 0 and 1 (CAR0 and CAR1)

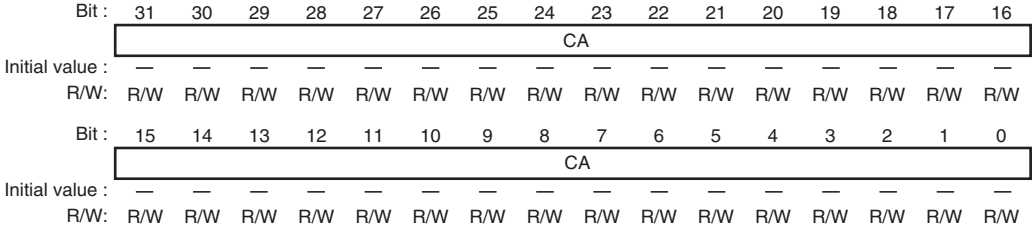
CAR0 and CAR1 are readable/writable 32-bit registers specifying the virtual address to be included in the break conditions for channels 0 and 1, respectively.

- CAR0



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CA	Undefined	R/W	Compare Address Specifies the address to be included in the break conditions. When the operand bus has been specified using the CBR0 register, specify the SAB address in CA[31:0].

- CAR1

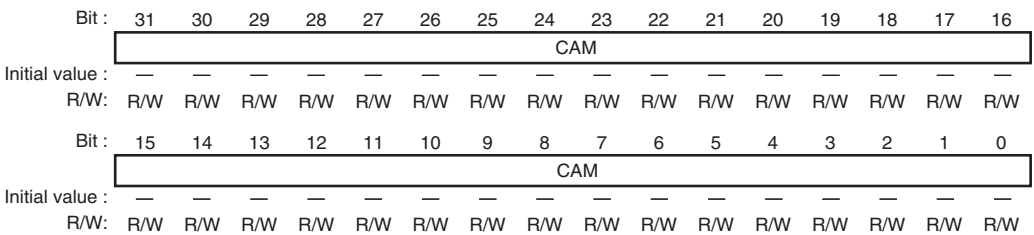


Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CA	Undefined	R/W	Compare Address Specifies the address to be included in the break conditions. When the operand bus has been specified using the CBR1 register, specify the SAB address in CA[31:0].

30.2.4 Match Address Mask Setting Registers 0 and 1 (CAMR0 and CAMR1)

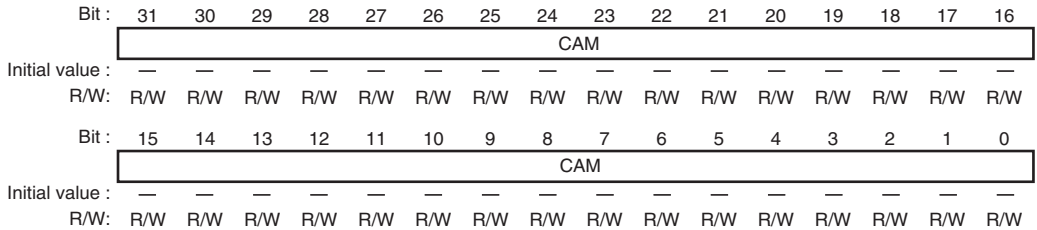
CMAR0 and CMAR1 are readable/writable 32-bit registers which specify the bits to be masked among the address bits specified by using the match address setting register of the corresponding channel. (Set the bits to be masked to 1.)

- CAMR0



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CAM	Undefined	R/W	Compare Address Mask Specifies the bits to be masked among the address bits which are specified using the CAR0 register. (Set the bits to be masked to 1.) 0: Address bits CA[n] are included in the break condition. 1: Address bits CA[n] are masked and not included in the break condition. [n] = any values from 31 to 0

- CAMR1



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CAM	Undefined	R/W	Compare Address Mask Specifies the bits to be masked among the address bits which are specified using the CAR1 register. (Set the bits to be masked to 1.) 0: Address bits CA[n] are included in the break condition. 1: Address bits CA[n] are masked and not included in the break condition. [n] = any values from 31 to 0

30.2.5 Match Data Setting Register 1 (CDR1)

CDR1 is a readable/writable 32-bit register which specifies the data value to be included in the break conditions for channel 1.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CD															
Initial value :	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CD															
Initial value :	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CD	Undefined	R/W	Compare Data Value Specifies the data value to be included in the break conditions. When the operand bus has been specified using the CBR1 register, specify the SDB data value in CD[31:0].

Table 30.3 Settings for Match Data Setting Register

Bus and Size Selected Using CBR1	CD[31:24]	CD[23:16]	CD[15:8]	CD[7:0]
Operand bus (byte)	Don't care	Don't care	Don't care	SDB7 to SDB0
Operand bus (word)	Don't care	Don't care	SDB15 to SDB8	SDB7 to SDB0
Operand bus (longword)	SDB31 to SDB24	SDB23 to SDB16	SDB15 to SDB8	SDB7 to SDB0

Notes: 1. If the data value is included in the match conditions, be sure to specify the operand size.
2. The OCBI instruction is handled as longword write access without the data value, and the PREF, OCBP, and OCBWB instructions are handled as longword read access without the data value. Therefore, do not include the data value in the match conditions for these instructions.
3. If the quadword access is specified and the data value is included in the match conditions, the upper and lower 32 bits of 64-bit data are each compared with the contents of both the match data setting register and match data mask setting register.

30.2.6 Match Data Mask Setting Register 1 (CDMR1)

CDMR1 is a readable/writable 32-bit register which specifies the bits to be masked among the data value bits specified using the match data setting register. (Set the bits to be masked to 1.)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDM															
Initial value :	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDM															
Initial value :	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CDM	Undefined	R/W	Compare Data Value Mask Specifies the bits to be masked among the data value bits specified using the CDR1 register. (Set the bits to be masked to 1.) 0: Data value bits CD[n] are included in the break condition. 1: Data value bits CD[n] are masked and not included in the break condition. [n] = any values from 31 to 0

30.2.7 Execution Count Break Register 1 (CETR1)

CETR1 is a readable/writable 32-bit register which specifies the number of the channel hits before a break occurs. A maximum value of $2^{12} - 1$ can be specified. When the execution count value is included in the match conditions by using the match condition setting register, the value of this register is decremented by one every time the channel is hit. When the channel is hit after the register value reaches H'001, a break occurs.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CET											
Initial value :	0	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
11 to 0	CET	Undefined	R/W	Execution Count Specifies the execution count to be included in the break conditions.

30.2.8 Channel Match Flag Register (CCMFR)

CCMFR is a readable/writable 32-bit register which indicates whether or not the match conditions have been satisfied for each channel. When a channel match condition has been satisfied, the corresponding flag bit is set to 1. To clear the flags, write the data containing value 0 for the bits to be cleared and value 1 for the other bits to this register. (The logical AND between the value which has been written and the current register value is actually written to the register.) Sequential operation using multiple channels is available by using these match flags.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MF1	MF0
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
1	MF1	0	R/W	Channel 1 Condition Match Flag This flag is set to 1 when the channel 1 match condition has been satisfied. To clear the flag, write 0 to this bit. 0: Channel 1 match condition has not been satisfied. 1: Channel 1 match condition has been satisfied.
0	MF0	0	R/W	Channel 0 Condition Match Flag This flag is set to 1 when the channel 0 match condition has been satisfied. To clear the flag, write 0 to this bit. 0: Channel 0 match condition has not been satisfied. 1: Channel 0 match condition has been satisfied.

30.2.9 Break Control Register (CBCR)

CBCR is a readable/writable 32-bit register which specifies whether or not to use the user break debugging support function. For details on the user break debugging support function, refer to section 30.4, User Break Debugging Support Function.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UBDE
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
0	UBDE	0	R/W	User Break Debugging Support Function Enable Specifies whether or not to use the user break debugging support function. 0: Does not use the user break debugging support function. 1: Uses the user break debugging support function.

30.3 Operation Description

30.3.1 Definition of Words Related to Accesses

"Instruction fetch" refers to an access in which an instruction is fetched. For example, fetching the instruction located at the branch destination after executing a branch instruction is an instruction access. "Operand access" refers to any memory access accompanying execution of an instruction. For example, accessing an address (PC + disp × 2 + 4) in the instruction MOV.W@(disp,PC),Rn is an operand access. "Data" is used in contrast to "address".

All types of operand access are classified into read or write access. Special care must be taken in using the following instructions.

- PREF, OCBP, and OCBWB: Instructions for a read access
- MOVCA.L and OCBI: Instructions for a write access
- TAS.B: Instruction for a single read access or a single write access

The operand access accompanying the PREF, OCBP, OCBWB, and OCBI instructions is access without the data value; therefore, do not include the data value in the match conditions for these instructions.

The operand size should be defined for all types of operand access. Available operand sizes are byte, word, longword, and quadword. For operand access accompanying the PREF, OCBP, OCBWB, MOVCA.L, and OCBI instructions, the operand size is defined as longword.

30.3.2 User Break Operation Sequence

The following describes the sequence from when the break condition is set until the user break exception handling is initiated.

1. Specify the operand size, bus, instruction fetch/operand access, and read/write as the match conditions using the match condition setting register (CBR0 or CBR1). Specify the break address using the match address setting register (CAR0 or CAR1), and specify the address mask condition using the match address mask setting register (CAMR0 or CAMR1). To include the ASID in the match conditions, set the AIE bit in the match condition setting register and specify the ASID value by the AIV bit in the same register. To include the data value in the match conditions, set the DBE bit in the match condition setting register; specify the break data using the match data setting register (CDR1); and specify the data mask condition using the match data mask setting register (CDMR1). To include the execution count in the match conditions, set the ETBE bit of the match condition setting register; and specify the execution count using the execution count break register (CETR1). To use the sequential break, set the MFE bit of the match condition setting register; and specify the number of the first channel using the MFI bit.
2. Specify whether or not to request a break when the match condition is satisfied and the break timing when the match condition is satisfied as a result of fetching the instruction using the match operation setting register (CRR0 or CRR1). After having set all the bits in the match condition setting register except the CE bit and the other necessary registers, set the CE bit and read the match condition setting register again. This ensures that the set values in the control registers are valid for the subsequent instructions immediately after reading the register. Setting the CE bit of the match condition setting register in the initial state after reset via the control registers may cause an undesired break.
3. When the match condition has been satisfied, the corresponding condition match flag (MF1 or MF0) in the channel match flag register (CCMFR) is set. A break is also requested to the CPU according to the set values in the match operation setting register (CRR0 or CRR1). The CPU operates differently according to the BL bit value of the SR register: when the BL bit is 0, the CPU accepts the break request and executes the specified exception handling; and when the BL bit is 1, the CPU does not execute the exception handling.
4. The match flags (MF1 and MF0) can be used to confirm whether or not the corresponding match condition has been satisfied. Although the flag is set when the condition is satisfied, it is not cleared automatically; therefore, write 0 to the flag bit by issuing a memory store instruction to the channel match flag register (CCMFR) in order to use the flag again.
5. Breaks may occur virtually at the same time for channels 0 and 1. In this case, only one break request is sent to the CPU; however, the two condition match flags corresponding to these breaks may be set.

6. While the BL bit in the SR register is 1, no break requests are accepted. However, whether or not the condition has been satisfied is determined. When the condition is determined to be satisfied, the corresponding condition match flag is set.
7. If the sequential break conditions are set, the condition match flag is set every time the match conditions are satisfied for each channel. When the conditions have been satisfied for the first channel in the sequence but not for the second channel in the sequence, clear the condition match flag for the first channel in the sequence in order to release the first channel in the sequence from the match state.

30.3.3 Instruction Fetch Cycle Break

1. If the instruction fetch cycle is set in the match condition setting register (CBR0 or CBR1), the instruction fetch cycle is handled as a match condition. To request a break upon satisfying the match condition, set the BIE bit in the match operation setting register (CRR0 or CRR1) of the corresponding channel. Either before or after executing the instruction can be selected as the break timing according to the PCB bit value. If the instruction fetch cycle is specified as a match condition, be sure to clear the LSB to 0 in the match address setting register (CAR0 or CAR1); otherwise, no break occurs.
2. If pre-instruction-execution break is specified for the instruction fetch cycle, the break is requested when the instruction is fetched and determined to be executed. Therefore, this function cannot be used for the instructions which are fetched through overrun (i.e., the instructions fetched during branching or making transition to the interrupt routine but not executed). For priorities of pre-instruction-execution break and the other exceptions, refer to section 5, Exception Handling. If pre-instruction-execution break is specified for the delayed slot of the delayed branch instruction, the break is requested before the delayed branch instruction is executed. However, do not specify pre-instruction-execution break for the delayed slot of the RTE instruction.
3. If post-instruction-execution break is specified for the instruction fetch cycle, the break is requested after the instruction which satisfied the match condition has been executed and before the next instruction is executed. Similar to pre-instruction-execution break, this function cannot be used for the instructions which are fetched through overrun. For priorities of post-instruction-execution break and the other exceptions, refer to section 5, Exception Handling. If post-instruction-execution break is specified for the delayed branch instruction and its delayed slot, the break does not occur until the first instruction at the branch destination.
4. If the instruction fetch cycle is specified as the channel 1 match condition, the DBE bit of match condition setting register CBR1 becomes invalid, the settings of match data setting register CDR1 and match data mask setting register CDMR1 are ignored. Therefore, the data value cannot be specified for the instruction fetch cycle break.

30.3.4 Operand Access Cycle Break

1. Table 30.4 shows the relation between the operand sizes specified using the match condition setting register (CBR0 or CBR1) and the address bits to be compared for the operand access cycle break.

Table 30.4 Relation between Operand Sizes and Address Bits to be Compared

Selected Operand Size	Address Bits to be Compared
Quadword	Address bits A31 to A3
Longword	Address bits A31 to A2
Word	Address bits A31 to A1
Byte	Address bits A31 to A0
Operand size is not included in the match conditions	Address bits A31 to A3 for quadword access Address bits A31 to A2 for longword access Address bits A31 to A1 for word access Address bits A31 to A0 for byte access

The above table means that if address H'00001003 is set in the match address setting register (CAR0 or CAR1), for example, the match condition is satisfied for the following access cycles (assuming that all the other conditions are satisfied):

— Longword access to address H'00001000

— Word access to address H'00001002

— Byte access to address H'00001003

2. When the data value is included in the channel 1 match conditions:

   If the data value is included in the match conditions, be sure to select the quadword, longword, word, or byte as the operand size using the operand size select bit (SZ) of the match condition setting register (CBR1), and also set the match data setting register (CDR1) and the match data mask setting register (CDMR1). With these settings, the match condition is satisfied when both of the address and data conditions are satisfied. The data value and mask control for byte access, word access, and longword access should be set in bits 7 to 0, 15 to 0, and 31 to 0 in the bits CDR1 and CDMR1, respectively. For quadword access, 64-bit data is divided into the upper and lower 32-bit data units, and each unit is independently compared with the specified condition. When either the upper or lower 32-bit data unit satisfies the match condition, the match condition for the 64-bit data is determined to be satisfied.

3. The operand access accompanying the PREF, OCBP, OCBWB, and OCBI instructions are access without the data value; therefore, if the data value is included in the match conditions for these instructions, the match conditions will never be satisfied.

4. If the operand bus is selected, a break occurs after executing the instruction which has satisfied the conditions and immediately before executing the next instruction. However, if the data value is included in the match conditions, a break may occur after executing several instructions after the instruction which has satisfied the conditions; therefore, it is impossible to identify the instruction causing the break. If such a break has occurred for the delayed branch instruction or its delayed slot, the break does not occur until the first instruction at the branch destination.

   However, do not specify the operand break for the delayed slot of the RTE instruction. And if the data value is included in the match conditions, it is not allowed to set the break for the preceding the RTE instruction by one to six instructions.

30.3.5 Sequential Break

1. Sequential break conditions can be specified by setting the MFE and MFI bits in the match condition setting registers (CBR0 and CBR1). (Sequential break involves two cases such that channel 0 break condition is satisfied then channel 1 break condition is satisfied, and vice versa.) To use the sequential break function, clear the MFE bit of the match condition setting register and the BIE bit of the match operation setting register of the first channel in the sequence, and set the MFE bit and specify the number of the second channel in the sequence using the MFI bit in the match condition setting register of the second channel in the sequence. If the sequential break condition is set, the condition match flag is set every time the match condition is satisfied for each channel. When the condition has been satisfied for the first channel in the sequence but not for the second channel in the sequence, clear the condition match flag for the first channel in the sequence in order to release the first channel in the sequence from the match state.
2. For channel 1, the execution count break condition can also be included in the sequential break conditions.
3. If the match conditions for the first and second channels in the sequence are satisfied within a significantly short time, sequential operation may not be guaranteed in some cases, as shown below.

- When the Match Condition is Satisfied at the Instruction Fetch Cycle for Both the First and Second Channels in the Sequence:

Instruction B is 0 instruction after instruction A	Equivalent to setting the same addresses; do not use this setting.
Instruction B is one instruction after instruction A	Sequential operation is not guaranteed.
Instruction B is two or more instructions after instruction A	Sequential operation is guaranteed.

- When the match condition is satisfied at the instruction fetch cycle for the first channel in the sequence whereas the match condition is satisfied at the operand access cycle for the second channel in the sequence:

Instruction B is 0 or one instruction after instruction A	Sequential operation is not guaranteed.
Instruction B is two or more instructions after instruction A	Sequential operation is guaranteed.

- When the match condition is satisfied at the operand access cycle for the first channel in the sequence whereas the match condition is satisfied at the instruction fetch cycle for the second channel in the sequence:

Instruction B is 0 to five instructions after instruction A	Sequential operation is not guaranteed.
Instruction B is six or more instructions after instruction A	Sequential operation is guaranteed.

- When the match condition is satisfied at the operand access cycle for both the first and second channels in the sequence:

Instruction B is 0 to five instructions after instruction A	Sequential operation is not guaranteed.
Instruction B is six or more instructions after instruction A	Sequential operation is guaranteed.

30.3.6 Program Counter Value to be Saved

When a break has occurred, the address of the instruction to be executed when the program restarts is saved in the SPC then the exception handling state is initiated. A unique instruction causing a break can be identified unless the data value is included in the match conditions.

- When the instruction fetch cycle (before instruction execution) is specified as the match condition:

  The address of the instruction which has satisfied the match conditions is saved in the SPC. The instruction which has satisfied the match conditions is not executed, but a break occurs instead. However, if the match conditions are satisfied for the delayed slot instruction, the address of the delayed branch instruction is saved in the SPC.
- When the instruction fetch cycle (after instruction execution) is specified as the match condition:

  The address of the instruction immediately after the instruction which has satisfied the match conditions is saved in the SPC. The instruction which has satisfied the match conditions is executed, then a break occurs before the next instruction. If the match conditions are satisfied for the delayed branch instruction or its delayed slot, these instructions are executed and the address of the branch destination is saved in the SPC.
- When the operand access (address only) is specified as the match condition:

  The address of the instruction immediately after the instruction which has satisfied the break conditions is saved in the SPC. The instruction which has satisfied the match conditions are executed, then a break occurs before the next instruction. However, if the conditions are satisfied for the delayed slot, the address of the branch destination is saved in the SPC.
- When the operand access (address and data) is specified as the match condition:

  If the data value is added to the match conditions, the instruction which has satisfied the match conditions is executed. A user break occurs before executing an instruction that is one through six instructions after the instruction which has satisfied the match conditions. The address of the instruction is saved in the SPC; thus, it is impossible to identify exactly where a break will occur. If the conditions are satisfied for the delayed slot instruction, the address of the branch destination is saved in the SPC. If a branch instruction follows the instruction which has satisfied the match conditions, a break may occur after the delayed instruction and delayed slot are executed. In this case, the address of the branch destination is also saved in the SPC.

30.4 User Break Debugging Support Function

By using the user break debugging support function, the branch destination address can be modified when the CPU accepts the user break request. Specifically, setting the UBDE bit of break control register CBCR to 1 allows branching to the address indicated by DBR instead of branching to the address indicated by the [VBR + offset]. Figure 30.2 shows the flowchart of the user break debugging support function.

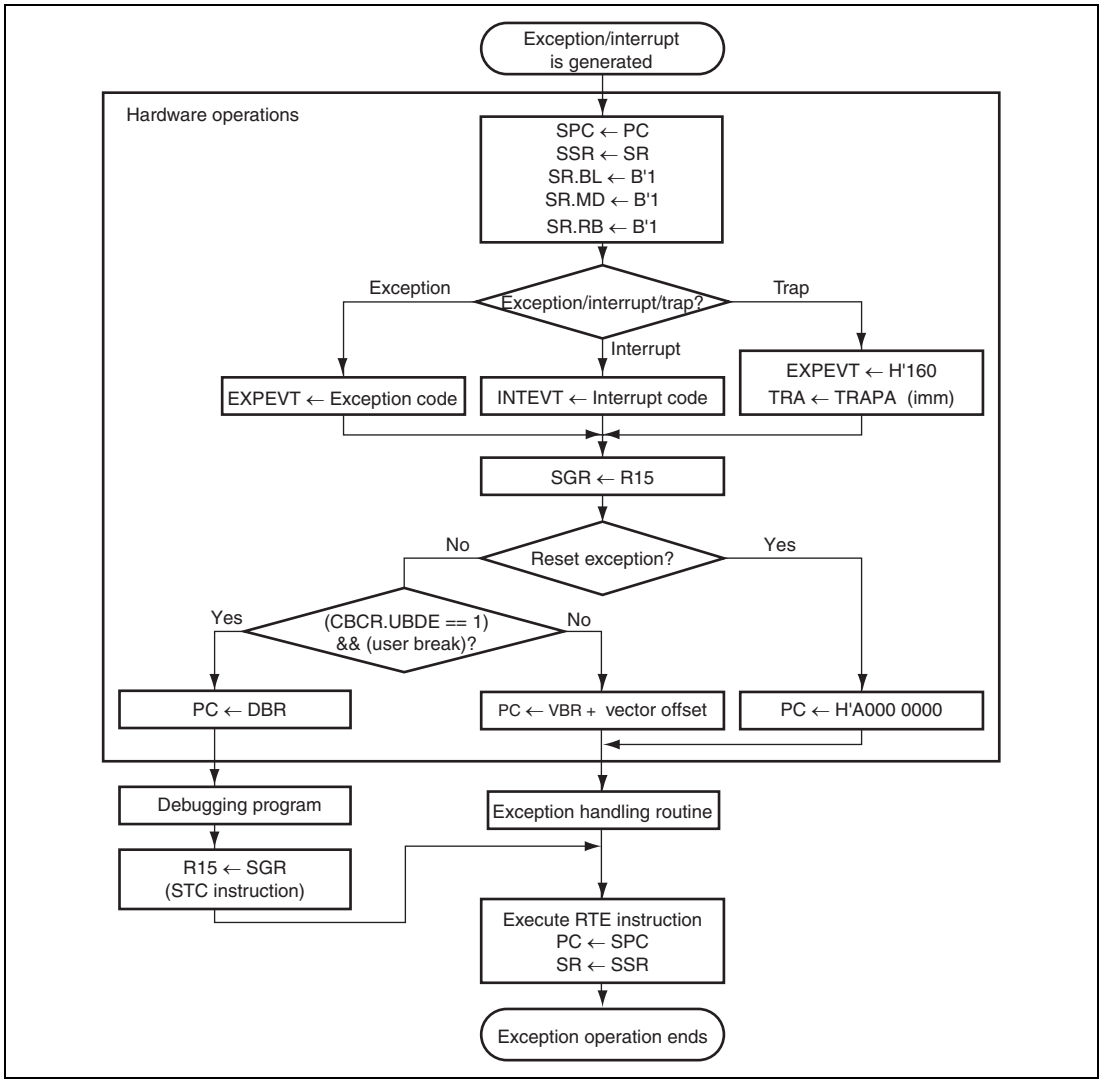


Figure 30.2 Flowchart of User Break Debugging Support Function

30.5 User Break Examples

(1) Match Conditions are Specified for an Instruction Fetch Cycle

- Example 1-1

  Register settings: CBR0 = H'00000013 / CRR0 = H'00002003 / CAR0 = H'00000404 / CAMR0 = H'00000000 / CBR1 = H'00000013 / CRR1 = H'00002001 / CAR1 = H'00008010 / CAMR1 = H'00000006 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

  Address: H'00000404 / Address mask: H'00000000

  Bus cycle: Instruction fetch (after executing the instruction)

  ASID is not included in the conditions.

  — Channel 1:

  Address: H'00008010 / Address mask: H'00000006

  Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

  Bus cycle: Instruction fetch (before executing instruction)

  ASID, data values, and execution count are not included in the conditions.

  With the above settings, the user break occurs after executing the instruction at address H'00000404 or before executing the instruction at address H'00008010 to H'00008016.

- Example 1-2

  Register settings: CBR0 = H'40800013 / CRR0 = H'00002000 / CAR0 = H'00037226 / CAMR0 = H'00000000 / CBR1 = H'C0700013 / CRR1 = H'00002001 / CAR1 = H'0003722E / CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

  Specified conditions: Channel 0 → Channel1 sequential mode

  — Channel 0

  Address: H'00037226 / Address mask: H'00000000 / ASID: H'80

  Bus cycle: Instruction fetch (before executing the instruction)

  — Channel 1

  Address: H'0003722E / Address mask: H'00000000 / ASID: H'70

  Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

  Bus cycle: Instruction fetch (before executing the instruction)

  Data values and execution count are not included in the conditions.

With the above settings, the user break occurs after executing the instruction at address H'00037226 where ASID is H'80 before executing the instruction at address H'0003722E where ASID is H'70.

- Example 1-3

  Register settings: CBR0 = H'00000013 / CRR0 = H'00002001 / CAR0 = H'00027128 / CAMR0 = H'00000000 / CBR1 = H'00000013 / CRR1 = H'00002001 / CAR1 = H'00031415 / CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

    Address: H'00027128 / Address mask: H'00000000

    Bus cycle: Instruction fetch (before executing the instruction)

    ASID is not included in the conditions.

  — Channel 1

    Address: H'00031415 / Address mask: H'00000000

    Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

    Bus cycle: Instruction fetch (before executing the instruction)

    ASID, data values, and execution count are not included in the conditions.

  With the above settings, the user break occurs for channel 0 before executing the instruction at address H'00027128. No user break occurs for channel 1 since the instruction fetch is executed only at even addresses.

- Example 1-4

  Register settings: CBR0 = H'40800013 / CRR0 = H'00002000 / CAR0 = H'00037226 / CAMR0 = H'00000000 / CBR1 = H'C0700013 / CRR1 = H'00002001 / CAR1 = H'0003722E / CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H′00000000 / CBCR = H'00000000

  Specified conditions: Channel 0 → Channel 1 sequential mode

  — Channel 0

    Address: H'00037226 / Address mask: H'00000000 / ASID: H'80

    Bus cycle: Instruction fetch (before executing the instruction)

  — Channel 1

    Address: H'0003722E / Address mask: H'00000000 / ASID: H'70

    Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

    Bus cycle: Instruction fetch (before executing the instruction)

    Data values and execution count are not included in the conditions.

With the above settings, the user break occurs after executing the instruction at address H'00037226 where ASID is H'80 and before executing the instruction at address H'0003722E where ASID is H'70.

- Example 1-5

  Register settings: CBR0 = H'00000013 / CRR0 = H'00002001 / CAR0 = H'00000500 / CAMR0 = H'00000000 / CBR1 = H'00000813 / CRR1 = H'00002001 / CAR1 = H'00001000 / CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000005 / CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

  Address: H'00000500 / Address mask: H'00000000

  Bus cycle: Instruction fetch (before executing the instruction)

  ASID is not included in the conditions.

  — Channel 1

  Address: H'00001000 / Address mask: H'00000000

  Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000005

  Bus cycle: Instruction fetch (before executing the instruction)

  Execution count: 5

  ASID and data values are not included in the conditions.

  With the above settings, the user break occurs for channel 0 before executing the instruction at address H'00000500. The user break occurs for channel 1 after executing the instruction at address H'00001000 four times; before executing the instruction five times.

- Example 1-6

  Register settings: CBR0 = H'40800013 / CRR0 = H'00002003 / CAR0 = H'00008404 / CAMR0 = H'00000FFF / CBR1 = H'40700013 / CRR1 = H'00002001 / CAR1 = H'00008010 / CAMR1 = H'00000006 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

  Address: H'00008404 / Address mask: H'00000FFF / ASID: H'80

  Bus cycle: Instruction fetch (after executing the instruction)

  — Channel 1

  Address: H'00008010 / Address mask: H'00000006 / ASID: H'70

  Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

  Bus cycle: Instruction fetch (before executing the instruction)

  Data values and execution count are not included in the conditions.

With the above settings, the user break occurs after executing the instruction at address H'00008000 to H'00008FFE where ASID is H'80 or before executing the instruction at address H'00008010 to H'00008016 where ASID is H'70.

(2) Match Conditions are Specified for an Operand Access Cycle

- Example 2-1

  Register settings: CBR0 = H'40800023 / CRR0 = H'00002001 / CAR0 = H'00123456 / CAMR0 = H'00000000 / CBR1 = H'4070A025 / CRR1 = H'00002001 / CAR1 = H'000ABCDE / CAMR1 = H'000000FF / CDR1 = H'0000A512 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

  Address: H'00123456 / Address mask: H'00000000 / ASID: H'80

  Bus cycle: Operand bus, operand access, and read (operand size is not included in the conditions.)

  — Channel 1

  Address: H'000ABCDE / Address mask: H'000000FF / ASID: H'70

  Data: H'0000A512 / Data mask: H'00000000 / Execution count: H'00000000

  Bus cycle: Operand bus, operand access, write, and word size

  Execution count is not included in the conditions.

  With these settings, the user break occurs for channel 0 for the following accesses: longword read access to address H'000123454, word read access to address H'000123456, byte read access to address H'000123456 where ASID is H'80. The user break occurs for channel 1 when word H'A512 is written to address H'000ABC00 to H'000ABCFE where ASID is H'70.

30.6 Usage Notes

- A desired break may not occur between the time when the instruction for rewriting the UBC register is executed and the time when the written value is actually reflected on the register.

  After the UBC register is updated, execute one of the following three methods.

  A. Read the updated UBC register, and execute a branch using the RTE instruction.
  (It is not necessary that a branch using the RTE instruction be next to a reading UBC register.)

  B. Execute the ICBI instruction for any address (including non-cacheable area).
  (It is not necessary that the ICBI instruction be next to a reading UBC register.)

  C. Set 0(initial value) to IRMCR.R1 before updating the UBC register and update with following sequence.

  a. Write the UBC register.

  b. Read the UBC register which is updated at 1.

  c. Write the value which is read at 2 to the UBC register.

Note: When two or more UBC registers are updated, executing these methods at each updating the UBC registers is not necessary. At only last updating the UBC register, execute one of these methods.

- The PCB bit of the CRR0 and CRR1 registers is valid only when the instruction fetch is specified as the match condition.
- If the sequential break conditions are set, the sequential break conditions are satisfied when the conditions for the first and second channels in the sequence are satisfied in this order. Therefore, if the conditions are set so that the conditions for channels 0 and 1 should be satisfied simultaneously for the same bus cycle, the sequential break conditions will not be satisfied, causing no break.
- For the SLEEP instruction, do not allow the post-instruction-execution break where the instruction fetch cycle is the match condition. For the instructions preceding the SLEEP instruction by one to five instructions, do not allow the break where the operand access is the match condition.
- If the user break and other exceptions occur for the same instruction, they are determined according to the specified priority. For the priority, refer to section 5, Exception Handling. If the exception having the higher priority occurs, the user break does not occur.
  — The pre-instruction-execution break is accepted prior to any other exception.

— If the post-instruction-execution break and data access break have occurred simultaneously with the re-execution type exception (including the pre-instruction-execution break) having a higher priority, only the re-execution type exception is accepted, and no condition match flags are set. When the exception handling has finished thus clearing the exception source, and when the same instruction has been executed again, the break occurs setting the corresponding flag.

— If the post-instruction-execution break or operand access break has occurred simultaneously with the completion-type exception (TRAPA) having a higher priority, then no user break occurs; however, the condition match flag is set.

- When conditions have been satisfied simultaneously and independently for channels 0 and 1, resulting in identical SPC values for both of the breaks, the user break occurs only once. However, the condition match flags are set for both channels. For example,

  Instruction at address 110 (post-instruction-execution break for instruction fetch for channel 0) → SPC = 112, CCMFR.MF0 = 1

  Instruction at address 112 (pre-instruction-execution break for instruction fetch for channel 1) → SPC = 112, CCMFR.MF1 = 1
- It is not allowed to set the pre-instruction-execution break or the operand break in the delayed slot instruction of the RTE instruction. And if the data value is included in the match conditions of the operand break, do not set the break for the preceding the RTE instruction by one to six instructions.
- If the re-execution type exception and the post-instruction-execution break are in conflict for the instruction requiring two or more execution states, then the re-execution type exception occurs. Here, the CCMFR.MF0 (or CCMFR.MF1) bit may or may not be set to 1 when the break conditions have been satisfied.

Section 31 User Debugging Interface (H-UDI)

The H-UDI is a serial interface which is based on the JTAG (IEEE 1149.4: IEEE Standard Test Access Port and Boundary-Scan Architecture) standard. The H-UDI is also used for emulator connection.

31.1 Features

The H-UDI is a serial interface which is based on the JTAG standard. The H-UDI is also used for emulator connection. When using an emulator, H-UDI functions should not be used. Refer to the appropriate emulator users manual for the method of connecting the emulator.

The H-UDI has six pins: TCK, TMS, TDI, TDO, $\overline{\text{TRST}}$, and $\overline{\text{ASEBRK}}$/BRKACK. The pin functions except $\overline{\text{ASEBRK}}$/BRKACK and serial communications protocol are based on the JTAG standard. This LSI has additional six pins for emulator connection: (AUDSYNC, AUDCK, and AUDATA3 to AUDATA0).

Figure 31.1 shows a block diagram of the H-UDI.

The TAP (Test Access Port) controller and five registers (SDBPR, SDIR, SDDRH, SDDRL, and SDINT). SDBPR supports the JTAG bypass mode, SDIR is used for commands, SDDR is used for data, and SDINT is used for H-UDI interrupts. SDIR is directly accessed from the TDI and TDO pins.

The TAP controller and control registers are initialized by driving the $\overline{\text{TRST}}$ pin low or by applying the TCK signal for five or more clock cycles with the TMS pin set to 1. This initialization sequence is independent of the reset pin for this LSI. Other circuits are initialized by a normal reset.

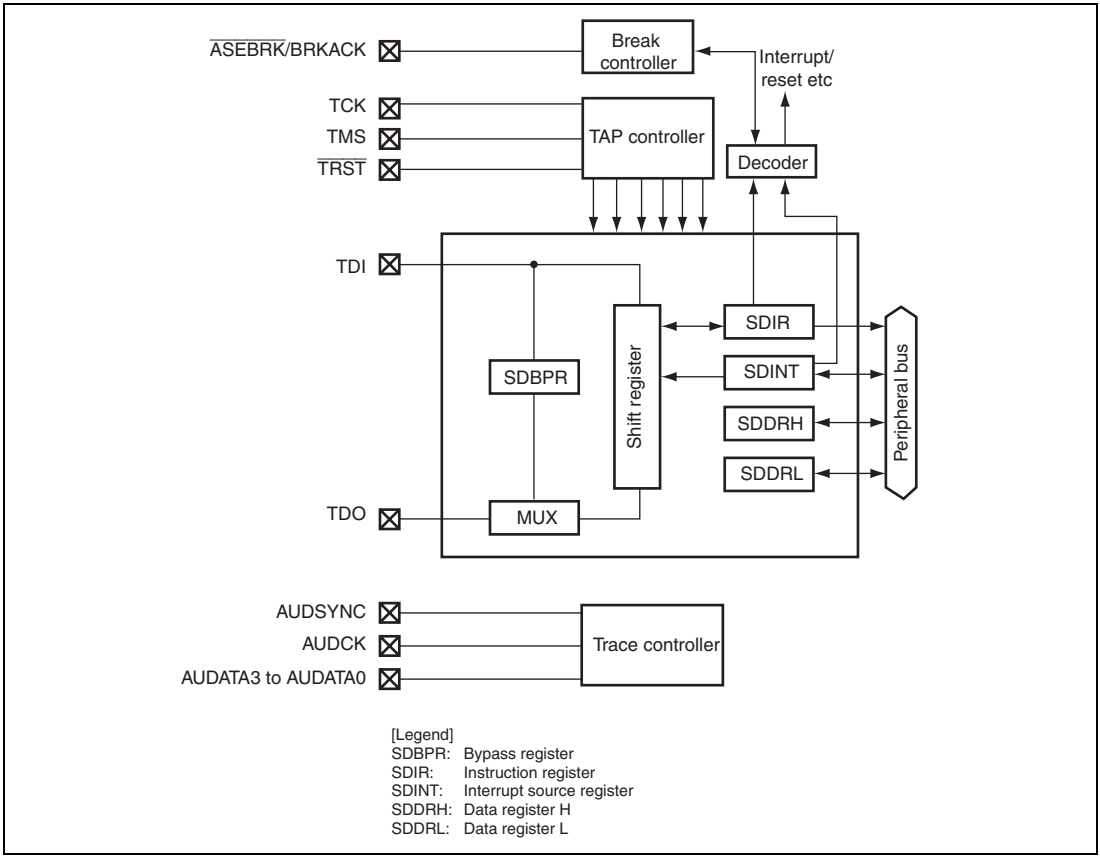


Figure 31.1 H-UDI Block Diagram

31.2 Input/Output Pins

Table 31.1 shows the pin configuration for the H-UDI.

Table 31.1 Pin Configuration

Pin Name	Function	I/O	Description	When Not in Use
TCK	Clock	Input	Functions as the serial clock input pin stipulated in the JTAG standard. Data input to the H-UDI via the TDI pin or data output via the TDO pin is performed in synchronization with this signal.	Open*[1]
TMS	Mode	Input	Mode Select Input Changing this signal in synchronization with the TCK signal determines the significance of data input via the TDI pin. Its protocol is based on the JTAG standard (IEEE standard 1149.1).	Open*[1]
$\overline{\text{TRST}}$*[2]	Reset	Input	H-UDI Reset Input This signal is received asynchronously with a TCK signal. Asserting this signal resets the JTAG interface circuit. When a power is supplied, the $\overline{\text{TRST}}$ pin should be asserted for a given period regardless of whether or not the JTAG function is used, which differs from the JTAG standard.	Fixed to ground*[3]
TDI	Data input	Input	Data Input Data is sent to the H-UDI by changing this signal in synchronization with the TCK signal.	Open*[1]
TDO	Data output	Output	Data Output Data is read from the H-UDI in synchronization with the TCK signal.	Open
$\overline{\text{ASEBRK}}$/ BRKACK	Emulator	I/O	Pins for an emulator	Open*[1]
AUDSYNC, AUDCK, AUDATA3 to AUDATA0	Emulator	Output	Pins for an emulator	Open

Pin Name	Function	I/O	Description	When Not in Use
MPMD	ASE mode (Emulation support mode setting)	Input	A low level on this pin places the chip in ASE mode, enabling use of the emulation support mode functions. When using an emulator such as the E10A, fix this pin at a low level.	Pull up

Notes: 1. This pin is pulled up in this LSI. When using interrupts or resets via the H-UDI or emulator, the use of external pull-up resistors will not cause any problem.
2. When using interrupts or resets via the H-UDI or emulator, the $\overline{\text{TRST}}$ pin should be designed so that it can be controlled independently and can be controlled to retain low level while the $\overline{\text{RESET}}$ pin is asserted at a power-on reset.
3. When connected to a ground pin, the following problem occurs. Since the $\overline{\text{TRST}}$ pin is pulled up within this LSI, a weak current flows when the pin is externally connected to a ground pin. The value of the current is determined by a resistance of the pull-up MOS for the port pin. Although this current does not affect the operation of this LSI, it consumes unnecessary power. Pulling up the $\overline{\text{TRST}}$ pin can be disabled by the pull-down control register (PULCR) of the pin function controller (PFC). For details, see section 29, Pin Function Controller (PFC).

The TCK clock or the CPG of this LSI should be set to ensure that the frequency of the TCK clock is less than the peripheral-clock frequency of this LSI.

31.3 Register Descriptions

The H-UDI has the following registers.

Table 31.2 Register Configuration (1)

		CPU Side				
Register Name	**Abbreviation**	**R/W**	**Area P4 Address*1**	**Area 7 Address*1**	**Size**	**Initial Value*2**
Instruction register	SDIR	R	H'FC11 0000	H'1C11 0000	16	H'0EFF
Data register H	SDDR/SDDRH	R/W	H'FC11 0008	H'1C11 0008	32/16	Undefined
Data register L	SDDRL	R/W	H'FC11 000A	H'1C11 000A	16	Undefined
Interrupt source register	SDINT	R/W	H'FC11 0018	H'1C11 0018	16	H'0000
Bypass register	SDBPR	—	—	—	—	Undefined

Notes: 1. The area P4 address is an address when accessing through area P4 in a virtual address space. The area 7 address is an address when accessing through area 7 in a physical space using the TLB.
2. The low level of the $\overline{\text{TRST}}$ pin or the Test-Logic-Reset state of the TAP controller initializes to these values.

Table 31.3 Register Configuration (2)

		H-UDI Side		
Register Name	**Abbreviation**	**R/W**	**Size**	**Initial Value*1**
Instruction register	SDIR	R/W	32	H'FFFF FFFD (fixed value*2)
Data register H	SDDR/SDDRH	—	—	—
Data register L	SDDRL	—	—	—
Interrupt source register	SDINT	W*3	32	H'0000 0000
Bypass register	SDBPR	R/W	1	Undefined

Notes: 1. The low level of the $\overline{\text{TRST}}$ pin or the Test-Logic-Reset state of the TAP controller initializes to these values.
2. When reading via the H-UDI, the value is always H'FFFF FFFD.
3. Only 1 can be written to the LSB by the H-UDI interrupt command.

Table 31.4 Register States in Each Operating Mode

Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
SDIR	H'0EFF	Retained	Retained	Retained
SDDR/SDDRH	Undefined	Retained	Retained	Retained
SDDRL	Undefined	Retained	Retained	Retained
SDINT	H'0000	Retained	Retained	Retained

31.3.1 Instruction Register (SDIR)

SDIR is a 16-bit read-only register that can be read from the CPU. Commands are set via the serial input (TDI). SDIR is initialized by $\overline{\text{TRST}}$ or in the Test-Logic-Reset state and can be written by the H-UDI irrespective of the CPU mode. Operation is not guaranteed when a reserved command is set to this register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TI[7:0]								—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	TI	B'00001110	R	Test Instruction Bits 7 to 0 01100000: H-UDI reset negate 01110000: H-UDI reset assert 10100000: H-UDI interrupt 00001110: Initial state Other than above: Setting prohibited
7 to 0	—	All 1	R	Reserved These bits are always read as 1.

31.3.2 Data Register H and L (SDDRH and SDDRL)

SDDR is a 32-bit register that comprises two registers: SDDRH and SDDRL. SDDRH and SDDRL are also 16-bit registers that can be read from or written to by the CPU. SDDR as a 32-bit register can be read from or written to by the CPU. The register value is not initialized by a reset for the CPU but is initialized by $\overline{\text{TRST}}$.

- SDDRH

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SDDRH data															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- SDDRL

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDDRL data															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

31.3.3 Interrupt Source Register (SDINT)

SDINT is a 16-bit register that can be read from or written to by the CPU. Specifying an H-UDI interrupt command in SDIR via H-UDI pin (Update-IR) sets the INTREQ bit to 1. While an H-UDI interrupt command is set in SDIR, SDINT which is connected between the TDI and TDO pins can be read as a 32-bit register. In this case, the upper 16 bits will be 0 and the lower 16 bits represent the SDINT value.

Only 0 can be written to the INTREQ bit by the CPU. While this bit is set to 1, an interrupt request will continue to be generated. This bit, therefore, should be cleared by the interrupt handling routine. It is initialized by $\overline{\text{TRST}}$ or in the Test-Logic-Reset state.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INTREQ
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved For reading from or writing to these bits, see General Precautions on Handling of Product.
0	INTREQ	0	R/W	Interrupt Request Indicates whether or not an interrupt by an H-UDI interrupt command has occurred. Clearing this bit to 0 by the CPU cancels an interrupt request. When writing 1 to this bit, the previous value is maintained.

31.3.4 Bypass Register (SDBPR)

SDBPR is an eight-bit register that supports the J-TAG bypass mode. When the BYPASS command is set to the boundary scan TAP controller, the TDI and TDO are connected by way of SDBPR. This register cannot be accessed from the CPU regardless of the LSI mode. Though this register is not initialized by a power-on reset and the $\overline{\text{TRST}}$ pin asserted, initialized to 0 in the Capture-DR state.

31.4 Operation

31.4.1 Boundary Scan TAP Controllers

The H-UDI contains two separate TAP controllers: one for controlling the boundary-scan function and another for controlling the H-UDI reset and interrupt functions. Assertion of TRST, for example at power-on reset, activates the boundary-scan TAP controller and enables the boundary-scan function prescribed in the JTAG standards. Executing a switchover command to the H-UDI allows usage of the H-UDI reset and H-UDI interrupts.

Note: Fix the levels on pins as follows.
High level: EXTAL_RTC, Test0_VccQ, MPMD, and Test2_VccQ pins
Low level: $\overline{\text{BREQ}}$/PIN1 and Test4_VssQ pins
Set the RESETP pin to the low level when power is supplied and to the high level during a boundary scan. Furthermore, input of a clock signal via the EXTAL is required after selection of the clock mode by the settings of the MD1 and MD0 pins.

Table 31.5 shows the commands supported by boundary-scan TAP controller.

Table 31.5 Commands Supported by Boundary-Scan TAP Controller

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description
0	1	0	1	0	1	0	1	IDCODE
1	1	1	1	1	1	1	1	BYPASS
0	0	0	0	0	0	0	0	EXTEST
0	1	0	0	0	0	0	0	SAMPLE/PRELOAD
0	0	0	0	1	0	0	0	H-UDI (switchover command)
Other than above								Setting prohibited

31.4.2 TAP Control

Figure 31.2 shows the internal states of the TAP controller. The state transitions is based on the JTAG standard.

- State transitions occur according to the TMS value at the rising edge of the TCK signal.
- The TDI value is sampled at the rising edge of the TCK signal and shifted at the falling edge of the TCK signal.
- The TDO value is changed at the falling edge of the TCK signal. The TDO signal is in a Hi-Z state other than in the Shift-DR or Shift-IR state.
- A transition to the Test-Logic-Reset by clearing $\overline{\text{TRST}}$ to 0 is performed asynchronously with the TCK signal.

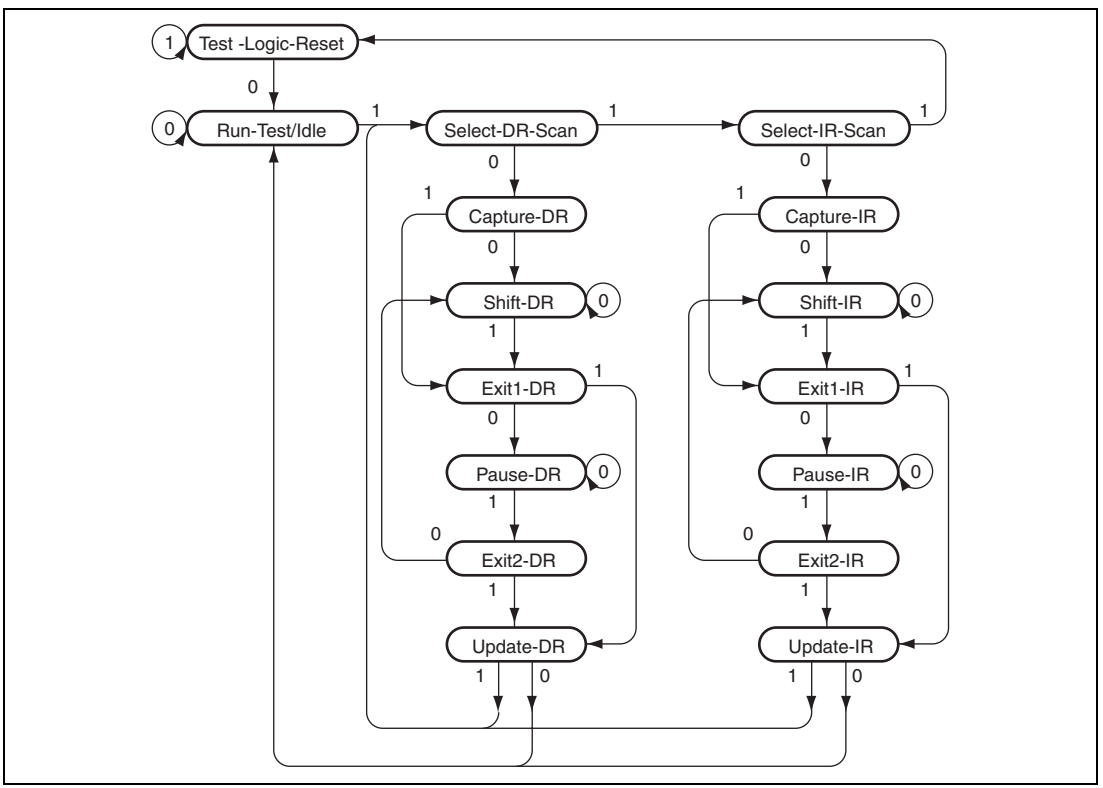


Figure 31.2 TAP Controller State Transitions

31.4.3 H-UDI Reset

A power-on reset is generated by the SDIR command. After the H-UDI reset assert command has been sent from the H-UDI pin, sending the H-UDI reset negate command resets the CPU (see figure 31.3). The required time between the H-UDI reset assert and H-UDI reset negate commands is the same as the time for holding the reset pin low in order to reset this LSI by a power-on reset.

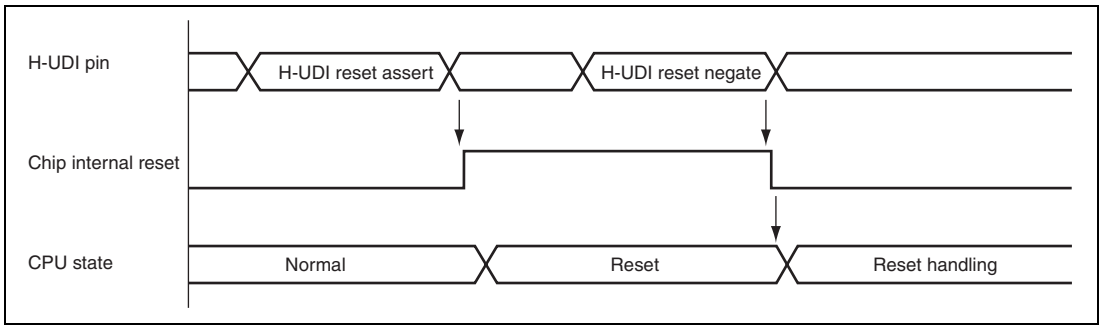


Figure 31.3 H-UDI Reset

31.4.4 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting the appropriate command in SDIR from the H-UDI. An H-UDI interrupt request signal is asserted when the INTREQ bit in SDINT is set to 1 by setting the appropriate command. Since the interrupt request signal is not negated until the INTREQ bit is cleared to 0 by software, it is not possible to lose the interrupt request. While an H-UDI interrupt command is set in SDIR, SDINT is connected between the TDI and TDO pins.

31.5 Usage Notes

- Once an SDIR command is set, it will be changed only by an assertion of the $\overline{\text{TRST}}$ signal, making the TAP controller Test-Logic-Reset state, or writing other commands from the H-UDI.
- The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot be used when using an emulator.

Section 32 List of Registers

32.1 Register Addresses

- Information on the on-chip I/O registers are listed for each functional module in the order of the corresponding section numbers
- Access to reserved addresses, which are not shown in this list, is prohibited. If accessed, operation at the time of access and the subsequent operation are not guaranteed.
- The access size is shown in the number of bits.
- For details on the individual registers, see the register description in the corresponding sections.

Table 32.1 Register Configuration (1)

Module	Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Access Size
Exception Handling	TRAPA exception register	TRA	R/W	H'FF00 0020	H'1F00 0020	32
	Exception event register	EXPEVT	R/W	H'FF00 0024	H'1F00 0024	32
	Interrupt event register	INTEVT	R/W	H'FF00 0028	H'1F00 0028	32
	Non-support detection exception register	EXPMASK	R/W	H'FF2F 0004	H'1F2F 0004	32
MMU	Page table entry high register	PTEH	R/W	H'FF00 0000	H'1F00 0000	32
	Page table entry low register	PTEL	R/W	H'FF00 0004	H'1F00 0004	32
	Translation table base register	TTB	R/W	H'FF00 0008	H'1F00 0008	32
	TLB exception address register	TEA	R/W	H'FF00 000C	H'1F00 000C	32
	MMU control register	MMUCR	R/W	H'FF00 0010	H'1F00 0010	32
	Physical address space control register	PASCR	R/W	H'FF00 0070	H'1F00 0070	32
	Instruction re-fetch inhibit control register	IRMCR	R/W	H'FF00 0078	H'1F00 0078	32
	Page table entry assistance register	PTEA	R/W	H'FF00 0034	H'1F00 0034	32
Caches	Cache control register	CCR	R/W	H'FF00 001C	H'1F00 001C	32
	Queue address control register 0	QACR0	R/W	H'FF00 0038	H'1F00 0038	32
	Queue address control register 1	QACR1	R/W	H'FF00 003C	H'1F00 003C	32
	On-chip memory control register	RAMCR	R/W	H'FF00 0074	H'1F00 0074	32

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 32.1 Register Configuration (2)

Module	Channel	Name	Abbreviation	R/W	Address	Access Size
INTC	—	Interrupt control register 0	ICR0	R/W	H'A414 0000	16
		Interrupt control register 1	ICR1	R/W	H'A414 001C	16
		Interrupt priority register 00	INTPRI00	R/W	H'A414 0010	32
		Interrupt request register 00	INTREQ00	R/W	H'A414 0024	8
		Interrupt mask register 00	INTMSK00	R/W	H'A414 0044	8
		Interrupt mask clear register 00	INTMSKCLR00	W	H'A414 0064	8
		NMI flag control register	NMIFCR	R/W	H'A414 00C0	16
		User interrupt mask level register	USERIMASK	R/W	H'A470 0000	32
		Interrupt priority register A	IPRA	R/W	H'A408 0000	16
		Interrupt priority register B	IPRB	R/W	H'A408 0004	16
		Interrupt priority register C	IPRC	R/W	H'A408 0008	16
		Interrupt priority register D	IPRD	R/W	H'A408 000C	16
		Interrupt priority register E	IPRE	R/W	H'A408 0010	16
		Interrupt priority register F	IPRF	R/W	H'A408 0014	16
		Interrupt priority register G	IPRG	R/W	H'A408 0018	16
		Interrupt priority register H	IPRH	R/W	H'A408 001C	16
		Interrupt priority register I	IPRI	R/W	H'A408 0020	16
		Interrupt priority register J	IPRJ	R/W	H'A408 0024	16
		Interrupt priority register K	IPRK	R/W	H'A408 0028	16
		Interrupt mask register 0	IMR0	R/W	H'A408 0080	8
		Interrupt mask register 1	IMR1	R/W	H'A408 0084	8
		Interrupt mask register 2	IMR2	R/W	H'A408 0088	8
		Interrupt mask register 3	IMR3	R/W	H'A408 008C	8
		Interrupt mask register 4	IMR4	R/W	H'A408 0090	8
		Interrupt mask register 5	IMR5	R/W	H'A408 0094	8
		Interrupt mask register 6	IMR6	R/W	H'A408 0098	8
		Interrupt mask register 7	IMR7	R/W	H'A408 009C	8
		Interrupt mask register 8	IMR8	R/W	H'A408 00A0	8
		Interrupt mask register 9	IMR9	R/W	H'A408 00A4	8
		Interrupt mask register 10	IMR10	R/W	H'A408 00A8	8
		Interrupt mask register 11	IMR11	R/W	H'A408 00AC	8
		Interrupt mask register 12	IMR12	R/W	H'A408 00B0	8
		Interrupt mask clear register 0	IMCR0	W	H'A408 00C0	8
		Interrupt mask clear register 1	IMCR1	W	H'A408 00C4	8
		Interrupt mask clear register 2	IMCR2	W	H'A408 00C8	8

Module	Channel	Name	Abbreviation	R/W	Address	Access Size
INTC	—	Interrupt mask clear register 3	IMCR3	W	H'A408 00CC	8
		Interrupt mask clear register 4	IMCR4	W	H'A408 00D0	8
		Interrupt mask clear register 5	IMCR5	W	H'A408 00D4	8
		Interrupt mask clear register 6	IMCR6	W	H'A408 00D8	8
		Interrupt mask clear register 7	IMCR7	W	H'A408 00DC	8
		Interrupt mask clear register 8	IMCR8	W	H'A408 00E0	8
		Interrupt mask clear register 9	IMCR9	W	H'A408 00E4	8
		Interrupt mask clear register 10	IMCR10	W	H'A408 00E8	8
		Interrupt mask clear register 11	IMCR11	W	H'A408 00EC	8
		Interrupt mask clear register 12	IMCR12	W	H'A408 00F0	8
BSC	—	Common control register	CMNCR	R/W	H'FEC1 0000	32
		CS0 space bus control register	CS0BCR	R/W	H'FEC1 0004	32
		CS2 space bus control register	CS2BCR	R/W	H'FEC1 0008	32
		CS3 space bus control register	CS3BCR	R/W	H'FEC1 000C	32
		CS4 space bus control register	CS4BCR	R/W	H'FEC1 0010	32
		CS5A space bus control register	CS5ABCR	R/W	H'FEC1 0014	32
		CS5B space bus control register	CS5BBCR	R/W	H'FEC1 0018	32
		CS6A space bus control register	CS6ABCR	R/W	H'FEC1 001C	32
		CS6B space bus control register	CS6BBCR	R/W	H'FEC1 0020	32
		CS0 space wait control register	CS0WCR	R/W	H'FEC1 0024	32
		CS2 space wait control register	CS2WCR	R/W	H'FEC1 0028	32
		CS3 space wait control register	CS3WCR	R/W	H'FEC1 002C	32
		CS4 space wait control register	CS4WCR	R/W	H'FEC1 0030	32
		CS5A space wait control register	CS5AWCR	R/W	H'FEC1 0034	32
		CS5B space wait control register	CS5BWCR	R/W	H'FEC1 0038	32
		CS6A space wait control register	CS6AWCR	R/W	H'FEC1 003C	32
		CS6B space wait control register	CS6BWCR	R/W	H'FEC1 0040	32
		SDRAM control register	SDCR	R/W	H'FEC1 0044	32
		Refresh timer control/status register	RTCSR	R/W	H'FEC1 0048	32
		Refresh timer counter	RTCNT	R/W	H'FEC1 004C	32
		Refresh time constant register	RTCOR	R/W	H'FEC1 0050	32
		SDRAM mode register	SDMR2	W	H'FEC1 4xxx	—
		SDRAM mode register	SDMR3	W	H'FEC1 5xxx	—

Module	Channel	Name	Abbreviation	R/W	Address	Access Size
DMAC	0	DMA source address register_0	SAR_0	R/W	H'FE00 8020	32
		DMA destination address register_0	DAR_0	R/W	H'FE00 8024	32
		DMA transfer count register_0	TCR_0	R/W	H'FE00 8028	32
		DMA channel control register_0	CHCR_0	R/W	H'FE00 802C	32
	1	DMA source address register_1	SAR_1	R/W	H'FE00 8030	32
		DMA destination address register_1	DAR_1	R/W	H'FE00 8034	32
		DMA transfer count register_1	TCR_1	R/W	H'FE00 8038	32
		DMA channel control register_1	CHCR_1	R/W	H'FE00 803C	32
	2	DMA source address register_2	SAR_2	R/W	H'FE00 8040	32
		DMA destination address register_2	DAR_2	R/W	H'FE00 8044	32
		DMA transfer count register_2	TCR_2	R/W	H'FE00 8048	32
		DMA channel control register_2	CHCR_2	R/W	H'FE00 804C	32
	3	DMA source address register_3	SAR_3	R/W	H'FE00 8050	32
		DMA destination address register_3	DAR_3	R/W	H'FE00 8054	32
		DMA transfer count register_3	TCR_3	R/W	H'FE00 8058	32
		DMA channel control register_3	CHCR_3	R/W	H'FE00 805C	32
	Common	DMA operation register	DMAOR	R/W	H'FE00 8060	16
	4	DMA source address register_4	SAR_4	R/W	H'FE00 8070	32
		DMA destination address register_4	DAR_4	R/W	H'FE00 8074	32
		DMA transfer count register_4	TCR_4	R/W	H'FE00 8078	32
		DMA channel control register_4	CHCR_4	R/W	H'FE00 807C	32
	5	DMA source address register_5	SAR_5	R/W	H'FE00 8080	32
		DMA destination address register_5	DAR_5	R/W	H'FE00 8084	32
		DMA transfer count register_5	TCR_5	R/W	H'FE00 8088	32
		DMA channel control register_5	CHCR_5	R/W	H'FE00 808C	32
	0	DMA source address register B_0	SARB_0	R/W	H'FE00 8120	32
		DMA destination address register B_0	DARB_0	R/W	H'FE00 8124	32
		DMA transfer count register B_0	TCRB_0	R/W	H'FE00 8128	32
	1	DMA source address register B_1	SARB_1	R/W	H'FE00 8130	32
		DMA destination address register B_1	DARB_1	R/W	H'FE00 8134	32
		DMA transfer count register B_1	TCRB_1	R/W	H'FE00 8138	32
	2	DMA source address register B_2	SARB_2	R/W	H'FE00 8140	32
		DMA destination address register B_2	DARB_2	R/W	H'FE00 8144	32
		DMA transfer count register B_2	TCRB_2	R/W	H'FE00 8148	32

Module	Channel	Name	Abbreviation	R/W	Address	Access Size
DMAC	3	DMA source address register B_3	SARB_3	R/W	H'FE00 8150	32
		DMA destination address register B_3	DARB_3	R/W	H'FE00 8154	32
		DMA transfer count register B_3	TCRB_3	R/W	H'FE00 8158	32
	0, 1	DMA extended resource selector 0	DMARS0	R/W	H'FE00 9000	16
	2, 3	DMA extended resource selector 1	DMARS1	R/W	H'FE00 9004	16
	4, 5	DMA extended resource selector 2	DMARS2	R/W	H'FE00 9008	16
CPG	—	Frequency control register	FRQCR	R/W	H'A415 0000	32
		PLL control register	PLLCR	R/W	H'A415 0024	32
		IrDA clock control register	IrDACLKCR	R/W	H'A415 0018	32
		Oscillation settling time watch timer control register	OSCWTCR	R/W	H'A415 0044	32
Power-Down Modes	—	Standby control register	STBCR	R/W	H'A415 0020	32
		Module stop register 0	MSTPCR0	R/W	H'A415 0030	32
		Module stop register 1	MSTPCR1	R/W	H'A415 0034	32
		Module stop register 2	MSTPCR2	R/W	H'A415 0038	32
RWDT	—	RCLK watchdog timer counter	RWTCNT	R/W	H'A452 0000	8/16
		RCLK watchdog timer control/status register	RWTCSR	R/W	H'A452 0004	8/16
TPU0	Common	Timer start register	TPU0_TSTR	R/W	H'A463 0000	16
	0	Timer control register 0	TPU0_TCR0	R/W	H'A463 0010	16
		Timer mode register 0	TPU0_TMDR0	R/W	H'A463 0014	16
		Timer I/O control register 0	TPU0_TIOR0	R/W	H'A463 0018	16
		Timer interrupt enable register 0	TPU0_TIER0	R/W	H'A463 001C	16
		Timer status register 0	TPU0_TSR0	R/W	H'A463 0020	16
		Timer counter 0	TPU0_TCNT0	R/W	H'A463 0024	16
		Timer general register 0A	TPU0_TGR0A	R/W	H'A463 0028	16
		Timer general register 0B	TPU0_TGR0B	R/W	H'A463 002C	16
		Timer general register 0C	TPU0_TGR0C	R/W	H'A463 0030	16
		Timer general register 0D	TPU0_TGR0D	R/W	H'A463 0034	16
	1	Timer control register 1	TPU0_TCR1	R/W	H'A463 0050	16
		Timer mode register 1	TPU0_TMDR1	R/W	H'A463 0054	16
		Timer I/O control register 1	TPU0_TIOR1	R/W	H'A463 0058	16
		Timer interrupt enable register 1	TPU0_TIER1	R/W	H'A463 005C	16
		Timer status register 1	TPU0_TSR1	R/W	H'A463 0060	16
		Timer counter 1	TPU0_TCNT1	R/W	H'A463 0064	16

Module	Channel	Name	Abbreviation	R/W	Address	Access Size
TPU0	1	Timer general register 1A	TPU0_TGR1A	R/W	H'A463 0068	16
		Timer general register 1B	TPU0_TGR1B	R/W	H'A463 006C	16
		Timer general register 1C	TPU0_TGR1C	R/W	H'A463 0070	16
		Timer general register 1D	TPU0_TGR1D	R/W	H'A463 0074	16
	2	Timer control register 2	TPU0_TCR2	R/W	H'A463 0090	16
		Timer mode register 2	TPU0_TMDR2	R/W	H'A463 0094	16
		Timer I/O control register 2	TPU0_TIOR2	R/W	H'A463 0098	16
		Timer interrupt enable register 2	TPU0_TIER2	R/W	H'A463 009C	16
		Timer status register 2	TPU0_TSR2	R/W	H'A463 00A0	16
		Timer counter 2	TPU0_TCNT2	R/W	H'A463 00A4	16
		Timer general register 2A	TPU0_TGR2A	R/W	H'A463 00A8	16
		Timer general register 2B	TPU0_TGR2B	R/W	H'A463 00AC	16
		Timer general register 2C	TPU0_TGR2C	R/W	H'A463 00B0	16
		Timer general register 2D	TPU0_TGR2D	R/W	H'A463 00B4	16
	3	Timer control register 3	TPU0_TCR3	R/W	H'A463 00D0	16
		Timer mode register 3	TPU0_TMDR3	R/W	H'A463 00D4	16
		Timer I/O control register 3	TPU0_TIOR3	R/W	H'A463 00D8	16
		Timer interrupt enable register 3	TPU0_TIER3	R/W	H'A463 00DC	16
		Timer status register 3	TPU0_TSR3	R/W	H'A463 00E0	16
		Timer counter 3	TPU0_TCNT3	R/W	H'A463 00E4	16
		Timer general register 3A	TPU0_TGR3A	R/W	H'A463 00E8	16
		Timer general register 3B	TPU0_TGR3B	R/W	H'A463 00EC	16
		Timer general register 3C	TPU0_TGR3C	R/W	H'A463 00F0	16
		Timer general register 3D	TPU0_TGR3D	R/W	H'A463 00F4	16
TPU1	Common	Timer start register	TPU1_TSTR	R/W	H'A44F 0000	16
	0	Timer control register 0	TPU1_TCR0	R/W	H'A44F 0010	16
		Timer mode register 0	TPU1_TMDR0	R/W	H'A44F 0014	16
		Timer I/O control register 0	TPU1_TIOR0	R/W	H'A44F 0018	16
		Timer interrupt enable register 0	TPU1_TIER0	R/W	H'A44F 001C	16
		Timer status register 0	TPU1_TSR0	R/W	H'A44F 0020	16
		Timer counter 0	TPU1_TCNT0	R/W	H'A44F 0024	16
		Timer general register 0A	TPU1_TGR0A	R/W	H'A44F 0028	16
		Timer general register 0B	TPU1_TGR0B	R/W	H'A44F 002C	16
		Timer general register 0C	TPU1_TGR0C	R/W	H'A44F 0030	16
		Timer general register 0D	TPU1_TGR0D	R/W	H'A44F 0034	16

Module	Channel	Name	Abbreviation	R/W	Address	Access Size
TPU1	1	Timer control register 1	TPU1_TCR1	R/W	H'A44F 0050	16
		Timer mode register 1	TPU1_TMDR1	R/W	H'A44F 0054	16
		Timer I/O control register 1	TPU1_TIOR1	R/W	H'A44F 0058	16
		Timer interrupt enable register 1	TPU1_TIER1	R/W	H'A44F 005C	16
		Timer status register 1	TPU1_TSR1	R/W	H'A44F 0060	16
		Timer counter 1	TPU1_TCNT1	R/W	H'A44F 0064	16
		Timer general register 1A	TPU1_TGR1A	R/W	H'A44F 0068	16
		Timer general register 1B	TPU1_TGR1B	R/W	H'A44F 006C	16
		Timer general register 1C	TPU1_TGR1C	R/W	H'A44F 0070	16
		Timer general register 1D	TPU1_TGR1D	R/W	H'A44F 0074	16
RTC	—	64 Hz counter	R64CNT	R	H'A465 FEC0	8
		Second counter	RSECCNT	R/W	H'A465 FEC2	8
		Minute counter	RMINCNT	R/W	H'A465 FEC4	8
		Hour counter	RHRCNT	R/W	H'A465 FEC6	8
		Day-of-week counter	RWKCNT	R/W	H'A465 FEC8	8
		Date counter	RDAYCNT	R/W	H'A465 FECA	8
		Month counter	RMONCNT	R/W	H'A465 FECC	8
		Year counter	RYRCNT	R/W	H'A465 FECE	16
		Second alarm register	RSECAR	R/W	H'A465 FED0	8
		Minute alarm register	RMINAR	R/W	H'A465 FED2	8
		Hour alarm register	RHRAR	R/W	H'A465 FED4	8
		Day-of-week alarm register	RWKAR	R/W	H'A465 FED6	8
		Date alarm register	RDAYAR	R/W	H'A465 FED8	8
		Month alarm register	RMONAR	R/W	H'A465 FEDA	8
		RTC control register1	RCR1	R/W	H'A465 FEDC	8
		RTC control register2	RCR2	R/W	H'A465 FEDE	8
		Year alarm register	RYRAR	R/W	H'A465 FEE0	16
		RTC control register3	RCR3	R/W	H'A465 FEE4	8
TMU	—	Timer start register	TSTR	R/W	H'FFD8 0004	8
		Timer constant register_0	TCOR_0	R/W	H'FFD8 0008	32
		Timer counter_0	TCNT_0	R/W	H'FFD8 000C	32
		Timer control register_0	TCR_0	R/W	H'FFD8 0010	16
		Timer constant register_1	TCOR_1	R/W	H'FFD8 0014	32
		Timer counter_1	TCNT_1	R/W	H'FFD8 0018	32
		Timer control register_1	TCR_1	R/W	H'FFD8 001C	16

Module	Channel	Name	Abbreviation	R/W	Address	Access Size
TMU	—	Timer constant register_2	TCOR_2	R/W	H'FFD8 0020	32
		Timer counter_2	TCNT_2	R/W	H'FFD8 0024	32
		Timer control register_2	TCR_2	R/W	H'FFD8 0028	16
CMT	Common	Compare match timer start register	CMSTR	R/W	H'A44A 0000	16
	0	Compare match timer control/status register_0	CMCSR_0	R/W	H'A44A 0010	16
		Compare match timer counter_0	CMCNT_0	R/W	H'A44A 0014	32
		Compare match timer constant register_0	CMCOR_0	R/W	H'A44A 0018	32
	1	Compare match timer control/status register_1	CMCSR_1	R/W	H'A44A 0020	16
		Compare match timer counter_1	CMCNT_1	R/W	H'A44A 0024	32
		Compare match timer constant register_1	CMCOR_1	R/W	H'A44A 0028	32
	2	Compare match timer control/status register_2	CMCSR_2	R/W	H'A44A 0030	16
		Compare match timer counter_2	CMCNT_2	R/W	H'A44A 0034	32
		Compare match timer constant register_2	CMCOR_2	R/W	H'A44A 0038	32
	3	Compare match timer control/status register_3	CMCSR_3	R/W	H'A44A 0040	16
		Compare match timer counter_3	CMCNT_3	R/W	H'A44A 0044	32
		Compare match timer constant register_3	CMCOR_3	R/W	H'A44A 0048	32
	4	Compare match timer control/status register_4	CMCSR_4	R/W	H'A44A 0050	16
		Compare match timer counter_4	CMCNT_4	R/W	H'A44A 0054	32
		Compare match timer constant register_4	CMCOR_4	R/W	H'A44A 0058	32
IIC	0	I^2C bus control register 1	ICCR1_0	R/W	H'A447 0000	8
		I^2C bus control register 2	ICCR2_0	R/W	H'A447 0001	8
		I^2C bus mode register	ICMR_0	R/W	H'A447 0002	8
		I^2C bus interrupt enable register	ICIER_0	R/W	H'A447 0003	8
		I^2C bus status register	ICSR_0	R/W	H'A447 0004	8
		Slave address register	SAR_0	R/W	H'A447 0005	8
		I^2C bus transmit data register	ICDRT_0	R/W	H'A447 0006	8
		I^2C bus receive data register	ICDRR_0	R	H'A447 0007	8
		NF2CYC register	NF2CYC_0	R/W	H'A447 0008	8
	1	I^2C bus control register 1	ICCR1_1	R/W	H'A475 0000	8
		I^2C bus control register 2	ICCR2_1	R/W	H'A475 0001	8
		I^2C bus mode register	ICMR_1	R/W	H'A475 0002	8
		I^2C bus interrupt enable register	ICIER_1	R/W	H'A475 0003	8
		I^2C bus status register	ICSR_1	R/W	H'A475 0004	8
		Slave address register	SAR_1	R/W	H'A475 0005	8

Module	Channel	Name	Abbreviation	R/W	Address	Access Size
IIC	1	I²C bus transmit data register	ICDRT_1	R/W	H'A475 0006	8
		I²C bus receive data register	ICDRR_1	R	H'A475 0007	8
		NF2CYC register	NF2CYC_1	R/W	H'A475 0008	8
SIOF	—	Mode register	SIMDR	R/W	H'A441 0000	16
		Clock select register	SISCR	R/W	H'A441 0002	16
		Transmit data assign register	SITDAR	R/W	H'A441 0004	16
		Receive data assign register	SIRDAR	R/W	H'A441 0006	16
		Control data assign register	SICDAR	R/W	H'A441 0008	16
		Control register	SICTR	R/W	H'A441 000C	16
		FIFO control register	SIFCTR	R/W	H'A441 0010	16
		Status register	SISTR	R/W	H'A441 0014	16
		Interrupt enable register	SIIER	R/W	H'A441 0016	16
		Transmit data register	SITDR	W	H'A441 0020	32
		Receive data register	SIRDR	R	H'A441 0024	32
		Transmit control data register	SITCR	R/W	H'A441 0028	32
		Receive control data register	SIRCR	R/W	H'A441 002C	32
SCIF	0	Serial mode register 0	SCSMR0	R/W	H'FFE0 0000	16
		Bit rate register 0	SCBRR0	R/W	H'FFE0 0004	8
		Serial control register 0	SCSCR0	R/W	H'FFE0 0008	16
		Transmit FIFO data register 0	SCFTDR0	W	H'FFE0 000C	8
		Serial status register 0	SCFSR0	R/W	H'FFE0 0010	16
		Receive FIFO data register 0	SCFRDR0	R	H'FFE0 0014	8
		FIFO control register 0	SCFCR0	R/W	H'FFE0 0018	16
		FIFO data count register 0	SCFDR0	R	H'FFE0 001C	16
		Line status register 0	SCLSR0	R/W	H'FFE0 0024	16
	1	Serial mode register 1	SCSMR1	R/W	H'FFE1 0000	16
		Bit rate register 1	SCBRR1	R/W	H'FFE1 0004	8
		Serial control register 1	SCSCR1	R/W	H'FFE1 0008	16
		Transmit FIFO data register 1	SCFTDR1	W	H'FFE1 000C	8
		Serial status register 1	SCFSR1	R/W	H'FFE1 0010	16
		Receive FIFO data register 1	SCFRDR1	R	H'FFE1 0014	8
		FIFO control register 1	SCFCR1	R/W	H'FFE1 0018	16
		FIFO data count register 1	SCFDR1	R	H'FFE1 001C	16
		Line status register 1	SCLSR1	R/W	H'FFE1 0024	16

Module	Channel	Name	Abbreviation	R/W	Address	Access Size
SCIF	2	Serial mode register 2	SCSMR2	R/W	H'FFE2 0000	16
		Bit rate register 2	SCBRR2	R/W	H'FFE2 0004	8
		Serial control register 2	SCSCR2	R/W	H'FFE2 0008	16
		Transmit FIFO data register 2	SCFTDR2	W	H'FFE2 000C	8
		Serial status register 2	SCFSR2	R/W	H'FFE2 0010	16
		Receive FIFO data register 2	SCFRDR2	R	H'FFE2 0014	8
		FIFO control register 2	SCFCR2	R/W	H'FFE2 0018	16
		FIFO data count register 2	SCFDR2	R	H'FFE2 001C	16
		Line status register 2	SCLSR2	R/W	H'FFE2 0024	16
	3	Serial mode register 3	SCSMR3	R/W	H'FFE3 0000	16
		Bit rate register 3	SCBRR3	R/W	H'FFE3 0004	8
		Serial control register 3	SCSCR3	R/W	H'FFE3 0008	16
		Transmit FIFO data register 3	SCFTDR3	W	H'FFE3 000C	8
		Serial status register 3	SCFSR3	R/W	H'FFE3 0010	16
		Receive FIFO data register 3	SCFRDR3	R	H'FFE3 0014	8
		FIFO control register 3	SCFCR3	R/W	H'FFE3 0018	16
		FIFO data count register 3	SCFDR3	R	H'FFE3 001C	16
		Line status register 3	SCLSR3	R/W	H'FFE3 0024	16
SCIFA	4	Serial mode register A4	SCASMR4	R/W	H'FFE4 0000	16
		Bit rate register A4	SCABRR4	R/W	H'FFE4 0004	8
		Serial control register A4	SCASCR4	R/W	H'FFE4 0008	16
		Transmit data stop register A4	SCATDSR4	R/W	H'FFE4 000C	8
		FIFO error count register A4	SCAFER4	R	H'FFE4 0010	16
		Serial status register A4	SCASSR4	R/W	H'FFE4 0014	16
		FIFO control register A4	SCAFCR4	R/W	H'FFE4 0018	16
		FIFO data count register A4	SCAFDR4	R	H'FFE4 001C	16
		Transmit FIFO data register A4	SCAFTDR4	W	H'FFE4 0020	8
		Receive FIFO data register A4	SCAFRDR4	R	H'FFE4 0024	8
	5	Serial mode register A5	SCASMR5	R/W	H'FFE5 0000	16
		Bit rate register A5	SCABRR5	R/W	H'FFE5 0004	8
		Serial control register A5	SCASCR5	R/W	H'FFE5 0008	16
		Transmit data stop register A5	SCATDSR5	R/W	H'FFE5 000C	8
		FIFO error count register A5	SCAFER5	R	H'FFE5 0010	16
		Serial status register A5	SCASSR5	R/W	H'FFE5 0014	16

Module	Channel	Name	Abbreviation	R/W	Address	Access Size
SCIFA	5	FIFO control register A5	SCAFCR5	R/W	H'FFE5 0018	16
		FIFO data count register A5	SCAFDR5	R	H'FFE5 001C	16
		Transmit FIFO data register A5	SCAFTDR5	W	H'FFE5 0020	8
		Receive FIFO data register A5	SCAFRDR5	R	H'FFE5 0024	8
IrDA	0	IrDA test register	IRIF0_INT2	R/W	H'A45D 0014	8/16
		DMA receive interrupt source clear register	IRIF0_RINTCLR	W	H'A45D 0016	16
		DMA transmit interrupt source clear register	IRIF0_TINTCLR	W	H'A45D 0018	16
		IrDA-SIR10 control register	IRIF0_SIR0	R/W	H'A45D 0020	8/16
		IrDA-SIR10 baud rate error correction register	IRIF0_SIR1	R/W	H'A45D 0022	8/16
		IrDA-SIR10 baud rate count set register	IRIF0_SIR2	R/W	H'A45D 0024	8/16
		IrDA-SIR10 status register	IRIF0_SIR3	R	H'A45D 0026	8/16
		Hardware frame processing set register	IRIF0_SIR_FRM	R/W	H'A45D 0028	8/16
		EOF value register	IRIF0_SIR_EOF	R/W	H'A45D 002A	8/16
		Flag clear register	IRIF0_SIR_FLG	W	H'A45D 002C	8/16
		UART status register 2	IRIF0_SIR_STS2	R/W	H'A45D 002E	8/16
		UART control register	IRIF0_UART0	R/W	H'A45D 0030	8/16
		UART status register	IRIF0_UART1	R	H'A45D 0032	8/16
		UART transmit data register	IRIF0_UART3	W	H'A45D 0036	8/16
		UART receive data register	IRIF0_UART4	R	H'A45D 0038	8/16
		UART interrupt mask register	IRIF0_UART5	R/W	H'A45D 003A	8/16
		UART baud rate error correction register	IRIF0_UART6	R/W	H'A45D 003C	8/16
		UART baud rate count set register	IRIF0_UART7	R/W	H'A45D 003E	8/16
		CRC engine control register	IRIF0_CRC0	R/W	H'A45D 0040	8/16
		CRC engine input data register	IRIF0_CRC1	W	H'A45D 0042	8/16
		CRC engine calculation register	IRIF0_CRC2	W	H'A45D 0044	8/16
		CRC engine output data register 1	IRIF0_CRC3	R	H'A45D 0046	8/16
		CRC engine output data register 2	IRIF0_CRC4	R	H'A45D 0048	8/16
	1	IrDA test register	IRIF1_INT2	R/W	H'A45E 0014	8/16
		DMA receive interrupt source clear register	IRIF1_RINTCLR	W	H'A45E 0016	16
		DMA transmit interrupt source clear register	IRIF1_TINTCLR	W	H'A45E 0018	16
		IrDA-SIR10 control register	IRIF1_SIR0	R/W	H'A45E 0020	8/16
		IrDA-SIR10 baud rate error correction register	IRIF1_SIR1	R/W	H'A45E 0022	8/16
		IrDA-SIR10 baud rate count set register	IRIF1_SIR2	R/W	H'A45E 0024	8/16
		IrDA-SIR10 status register	IRIF1_SIR3	R	H'A45E 0026	8/16

Module	Channel	Name	Abbreviation	R/W	Address	Access Size
IrDA	1	Hardware frame processing set register	IRIF1_SIR_FRM	R/W	H'A45E 0028	8/16
		EOF value register	IRIF1_SIR_EOF	R/W	H'A45E 002A	8/16
		Flag clear register	IRIF1_SIR_FLG	W	H'A45E 002C	8/16
		UART status register 2	IRIF1_SIR_STS2	R/W	H'A45E 002E	8/16
		UART control register	IRIF1_UART0	R/W	H'A45E 0030	8/16
		UART status register	IRIF1_UART1	R	H'A45E 0032	8/16
		UART transmit data register	IRIF1_UART3	W	H'A45E 0036	8/16
		UART receive data register	IRIF1_UART4	R	H'A45E 0038	8/16
		UART interrupt mask register	IRIF1_UART5	R/W	H'A45E 003A	8/16
		UART baud rate error correction register	IRIF1_UART6	R/W	H'A45E 003C	8/16
		UART baud rate count set register	IRIF1_UART7	R/W	H'A45E 003E	8/16
		CRC engine control register	IRIF1_CRC0	R/W	H'A45E 0040	8/16
		CRC engine input data register	IRIF1_CRC1	W	H'A45E 0042	8/16
		CRC engine calculation register	IRIF1_CRC2	W	H'A45E 0044	8/16
		CRC engine output data register 1	IRIF1_CRC3	R	H'A45E 0046	8/16
		CRC engine output data register 2	IRIF1_CRC4	R	H'A45E 0048	8/16
SIM	—	Serial mode register	SCSMR	R/W	H'A449 0000	8
		Bit rate register	SCBRR	R/W	H'A449 0002	8
		Serial control register	SCSCR	R/W	H'A449 0004	8
		Transmit data register	SCTDR	R/W	H'A449 0006	8
		Serial status register	SCSSR	R/W	H'A449 0008	8
		Receive data register	SCRDR	R	H'A449 000A	8
		Smart card mode register	SCSCMR	R/W	H'A449 000C	8
		Serial control 2 register	SCSC2R	R	H'A449 000E	8
		Wait time register	SCWAIT	R/W	H'A449 0010	16
		Guard extension register	SCGRD	R/W	H'A449 0012	8
		Sampling register	SCSMPL	R/W	H'A449 0014	16
		DMA enable register	SCDMAEN	R/W	H'A449 0016	8
ADC	—	A/D data register A	ADDRA	R	H'A461 0000	16
		A/D data register B	ADDRB	R	H'A461 0002	16
		A/D data register C	ADDRC	R	H'A461 0004	16
		A/D data register D	ADDRD	R	H'A461 0006	16
		A/D control/status register	ADCSR	R/W	H'A461 0008	16

Module	Channel	Name	Abbreviation	R/W	Address	Access Size
DAC	—	D/A data register 0	DADR0	R/W	H'A462 0000	16
		D/A data register 1	DADR1	R/W	H'A462 0002	16
		D/A control register	DACR	R/W	H'A462 0004	16
I/O Port	—	Port A data register	PADR	R/W	H'A405 0080	8
		Port B data register	PBDR	R/W	H'A405 0082	8
		Port C data register	PCDR	R/W	H'A405 0084	8
		Port D data register	PDDR	R/W	H'A405 0086	8
		Port E data register	PEDR	R/W	H'A405 0088	8
		Port F data register	PFDR	R/W	H'A405 008A	8
		Port G data register	PGDR	R/W	H'A405 008C	8
		Port H data register	PHDR	R/W	H'A405 008E	8
		Port J data register	PJDR	R/W	H'A405 0090	8
		Port K data register	PKDR	R/W	H'A405 0092	8
		Port L data register	PLDR	R/W	H'A405 0094	8
		Port M data register	PMDR	R/W	H'A405 0096	8
		Port N data register	PNDR	R/W	H'A405 0098	8
		Port Q data register	PQDR	R/W	H'A405 009A	8
		Port R data register	PRDR	R/W	H'A405 009C	8
		Port S data register	PSDR	R/W	H'A405 009E	8
		Port T data register	PTDR	R/W	H'A405 00A0	8
PFC	—	Port A control register	PACR	R/W	H'A405 0000	16
		Port B control register	PBCR	R/W	H'A405 0002	16
		Port C control register	PCCR	R/W	H'A405 0004	16
		Port D control register	PDCR	R/W	H'A405 0006	16
		Port E control register	PECR	R/W	H'A405 0008	16
		Port F control register	PFCR	R/W	H'A405 000A	16
		Port G control register	PGCR	R/W	H'A405 000C	16
		Port H control register	PHCR	R/W	H'A405 000E	16
		Port J control register	PJCR	R/W	H'A405 0010	16
		Port K control register	PKCR	R/W	H'A405 0012	16
		Port L control register	PLCR	R/W	H'A4050014	16
		Port M control register	PMCR	R/W	H'A4050016	16
		Port N control register	PNCR	R/W	H'A4050018	16

Module	Channel	Name	Abbreviation	R/W	Address	Access Size
PFC	—	Port Q control register	PQCR	R/W	H'A405001A	16
		Port R control register	PRCR	R/W	H'A405001C	16
		Port S control register	PSCR	R/W	H'A405001E	16
		Port T control register	PTCR	R/W	H'A4050020	16
		Pin select register A	PSELA	R/W	H'A4050100	16
		Pin select register B	PSELB	R/W	H'A4050102	16
		Pin select register C	PSELC	R/W	H'A4050104	16
		I/O buffer Hi-Z control register A	HIZCRA	R/W	H'A4050120	16
		I/O buffer Hi-Z control register B	HIZCRB	R/W	H'A4050122	16
		I/O buffer Hi-Z control register C	HIZCRC	R/W	H'A4050124	16
		I/O buffer Hi-Z control register D	HIZCRD	R/W	H'A4050126	16
		I/O buffer Hi-Z control register E	HIZCRE	R/W	H'A4050128	16
		I/O buffer Hi-Z control register F	HIZCRF	R/W	H'A405012A	16
		Pull-up/pull-down control register	PULCR	R/W	H'A405015E	16
		PINT control register A	PINTCRA	R/W	H'A4050040	16
		PINT control register B	PINTCRB	R/W	H'A4050042	16

Table 32.1 Register Configuration (3)

Module	Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Access Size
UBC	Match condition setting register 0	CBR0	R/W	H'FF20 0000	H'1F20 0000	32
	Match operation setting register 0	CRR0	R/W	H'FF20 0004	H'1F20 0004	32
	Match address setting register 0	CAR0	R/W	H'FF20 0008	H'1F20 0008	32
	Match address mask setting register 0	CAMR0	R/W	H'FF20 000C	H'1F20 000C	32
	Match condition setting register 1	CBR1	R/W	H'FF20 0020	H'1F20 0020	32
	Match operation setting register 1	CRR1	R/W	H'FF20 0024	H'1F20 0024	32
	Match address setting register 1	CAR1	R/W	H'FF20 0028	H'1F20 0028	32
	Match address mask setting register 1	CAMR1	R/W	H'FF20 002C	H'1F20 002C	32
	Match data setting register 1	CDR1	R/W	H'FF20 0030	H'1F20 0030	32
	Match data mask setting register 1	CDMR1	R/W	H'FF20 0034	H'1F20 0034	32
	Execution count break register 1	CETR1	R/W	H'FF20 0038	H'1F20 0038	32
	Channel match flag register	CCMFR	R/W	H'FF20 0600	H'1F20 0600	32
	Break control register	CBCR	R/W	H'FF20 0620	H'1F20 0620	32
H-UDI	Instruction register	SDIR	R	H'FC11 0000	H'1C11 0000	16
	Data register H	SDDR/SDDRH	R/W	H'FC11 0008	H'1C11 0008	32/16
	Data register L	SDDRL	R/W	H'FC11 000A	H'1C11 000A	16
	Interrupt source register	SDINT	R/W	H'FC11 0018	H'1C11 0018	16
	Bypass register	SDBPR	—	—	—	—

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

32.2 Register States in Each Operating Mode

Table 32.2 Register States in Each Operating Mode (1)

Module	Register Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Exception Handling	TRA	Initialized	Retained	Retained	Retained
	EXPEVT	Initialized	Retained	Retained	Retained
	INTEVT	Initialized	Retained	Retained	Retained
	EXPMASK	Initialized	Initialized	Retained	Retained
MMU	PTEH	Initialized	Initialized	Retained	Retained
	PTEL	Initialized	Initialized	Retained	Retained
	TTB	Initialized	Initialized	Retained	Retained
	TEA	Initialized	Retained	Retained	Retained
	MMUCR	Initialized	Initialized	Retained	Retained
	PASCR	Initialized	Initialized	Retained	Retained
	IRMCR	Initialized	Initialized	Retained	Retained
	PTEA	Initialized	Initialized	Retained	Retained
Cache	CCR	Initialized	Initialized	Retained	Retained
	QACR0	Initialized	Initialized	Retained	Retained
	QACR1	Initialized	Initialized	Retained	Retained
	RAMCR	Initialized	Initialized	Retained	Retained

Table 32.2 Register States in Each Operating Mode (2)

Module	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
INTC	ICR0	Initialized	Retained	Retained	Retained
	ICR1	Initialized	Retained	Retained	Retained
	INTPRI00	Initialized	Retained	Retained	Retained
	INTREQ00	Initialized	Retained	Retained	Retained
	INTMSK00	Initialized	Retained	Retained	Retained
	INTMSKCLR00	Initialized	Retained	Retained	Retained
	NMIFCR	Initialized	Retained	Retained	Retained
	USERIMASK	Initialized	Retained	Retained	Retained
	IPRA	Initialized	Retained	Retained	Retained
	IPRB	Initialized	Retained	Retained	Retained
	IPRC	Initialized	Retained	Retained	Retained
	IPRD	Initialized	Retained	Retained	Retained
	IPRE	Initialized	Retained	Retained	Retained
	IPRF	Initialized	Retained	Retained	Retained
	IPRG	Initialized	Retained	Retained	Retained
	IPRH	Initialized	Retained	Retained	Retained
	IPRI	Initialized	Retained	Retained	Retained
	IPRJ	Initialized	Retained	Retained	Retained
	IPRK	Initialized	Retained	Retained	Retained
	IMR0	Initialized	Retained	Retained	Retained
	IMR1	Initialized	Retained	Retained	Retained
	IMR2	Initialized	Retained	Retained	Retained
	IMR3	Initialized	Retained	Retained	Retained
	IMR4	Initialized	Retained	Retained	Retained
	IMR5	Initialized	Retained	Retained	Retained
	IMR6	Initialized	Retained	Retained	Retained
	IMR7	Initialized	Retained	Retained	Retained
	IMR8	Initialized	Retained	Retained	Retained
	IMR9	Initialized	Retained	Retained	Retained

Module	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
INTC	IMR10	Initialized	Retained	Retained	Retained
	IMR11	Initialized	Retained	Retained	Retained
	IMR12	Initialized	Retained	Retained	Retained
	IMCR0	Initialized	Retained	Retained	Retained
	IMCR1	Initialized	Retained	Retained	Retained
	IMCR2	Initialized	Retained	Retained	Retained
	IMCR3	Initialized	Retained	Retained	Retained
	IMCR4	Initialized	Retained	Retained	Retained
	IMCR5	Initialized	Retained	Retained	Retained
	IMCR6	Initialized	Retained	Retained	Retained
	IMCR7	Initialized	Retained	Retained	Retained
	IMCR8	Initialized	Retained	Retained	Retained
	IMCR9	Initialized	Retained	Retained	Retained
	IMCR10	Initialized	Retained	Retained	Retained
	IMCR11	Initialized	Retained	Retained	Retained
	IMCR12	Initialized	Retained	Retained	Retained
BSC	CMNCR	Initialized	Retained	—	Retained
	CS0BCR	Initialized	Retained	—	Retained
	CS2BCR	Initialized	Retained	—	Retained
	CS3BCR	Initialized	Retained	—	Retained
	CS4BCR	Initialized	Retained	—	Retained
	CS5ABCR	Initialized	Retained	—	Retained
	CS5BBCR	Initialized	Retained	—	Retained
	CS6ABCR	Initialized	Retained	—	Retained
	CS6BBCR	Initialized	Retained	—	Retained
	CS0WCR	Initialized	Retained	—	Retained
	CS2WCR	Initialized	Retained	—	Retained
	CS3WCR	Initialized	Retained	—	Retained
	CS4WCR	Initialized	Retained	—	Retained
	CS5AWCR	Initialized	Retained	—	Retained
	CS5BWCR	Initialized	Retained	—	Retained

Module	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
BSC	CS6AWCR	Initialized	Retained	—	Retained
	CS6BWCR	Initialized	Retained	—	Retained
	SDCR	Initialized	Retained	—	Retained
	RTCSR	Initialized	Retained	—	Retained
	RTCNT	Initialized	Retained	—	Retained
	RTCOR	Initialized	Retained	—	Retained
	SDMR2	—	—	—	—
	SDMR3	—	—	—	—
DMAC	SAR_0	Initialized	Retained	Retained	Retained
	DAR_0	Initialized	Retained	Retained	Retained
	TCR_0	Initialized	Retained	Retained	Retained
	CHCR_0	Initialized	Retained	Retained	Retained
	SAR_1	Initialized	Retained	Retained	Retained
	DAR_1	Initialized	Retained	Retained	Retained
	TCR_1	Initialized	Retained	Retained	Retained
	CHCR_1	Initialized	Retained	Retained	Retained
	SAR_2	Initialized	Retained	Retained	Retained
	DAR_2	Initialized	Retained	Retained	Retained
	TCR_2	Initialized	Retained	Retained	Retained
	CHCR_2	Initialized	Retained	Retained	Retained
	SAR_3	Initialized	Retained	Retained	Retained
	DAR_3	Initialized	Retained	Retained	Retained
	TCR_3	Initialized	Retained	Retained	Retained
	CHCR_3	Initialized	Retained	Retained	Retained
	DMAOR	Initialized	Retained	Retained	Retained
	SAR_4	Initialized	Retained	Retained	Retained
	DAR_4	Initialized	Retained	Retained	Retained
	TCR_4	Initialized	Retained	Retained	Retained
	CHCR_4	Initialized	Retained	Retained	Retained
	SAR_5	Initialized	Retained	Retained	Retained
	DAR_5	Initialized	Retained	Retained	Retained

Module	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
DMAC	TCR_5	Initialized	Retained	Retained	Retained
	CHCR_5	Initialized	Retained	Retained	Retained
	SARB_0	Initialized	Retained	Retained	Retained
	DARB_0	Initialized	Retained	Retained	Retained
	TCRB_0	Initialized	Retained	Retained	Retained
	SARB_1	Initialized	Retained	Retained	Retained
	DARB_1	Initialized	Retained	Retained	Retained
	TCRB_1	Initialized	Retained	Retained	Retained
	SARB_2	Initialized	Retained	Retained	Retained
	DARB_2	Initialized	Retained	Retained	Retained
	TCRB_2	Initialized	Retained	Retained	Retained
	SARB_3	Initialized	Retained	Retained	Retained
	DARB_3	Initialized	Retained	Retained	Retained
	TCRB_3	Initialized	Retained	Retained	Retained
	DMARS0	Initialized	Retained	Retained	Retained
	DMARS1	Initialized	Retained	Retained	Retained
	DMARS2	Initialized	Retained	Retained	Retained
CPG	FRQCR	Initialized	Retained	—	Retained
	PLLCR	Initialized	Retained	—	Retained
	IrDACLKCR	Initialized	Retained	—	Retained
	OSCWTCR	Initialized	Retained	—	Retained
Power Down Mode	STBCR	Initialized	Retained	—	Retained
	MSTPCR0	Initialized	Retained	—	Retained
	MSTPCR1	Initialized	Retained	—	Retained
	MSTPCR2	Initialized	Retained	—	Retained
RWDT	RWTCNT	Initialized	Retained	Retained	Retained
	RWTCSR	Initialized	Retained	Retained	Retained
TPU0	TPU0_TSTR	Initialized	Retained	Retained	Retained
	TPU0_TCR0	Initialized	Retained	Retained	Retained
	TPU0_TMDR0	Initialized	Retained	Retained	Retained
	TPU0_TIOR0	Initialized	Retained	Retained	Retained

Module	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
TPU0	TPU0_TIER0	Initialized	Retained	Retained	Retained
	TPU0_TSR0	Initialized	Retained	Retained	Retained
	TPU0_TCNT0	Initialized	Retained	Retained	Retained
	TPU0_TGR0A	Initialized	Retained	Retained	Retained
	TPU0_TGR0B	Initialized	Retained	Retained	Retained
	TPU0_TGR0C	Initialized	Retained	Retained	Retained
	TPU0_TGR0D	Initialized	Retained	Retained	Retained
	TPU0_TCR1	Initialized	Retained	Retained	Retained
	TPU0_TMDR1	Initialized	Retained	Retained	Retained
	TPU0_TIOR1	Initialized	Retained	Retained	Retained
	TPU0_TIER1	Initialized	Retained	Retained	Retained
	TPU0_TSR1	Initialized	Retained	Retained	Retained
	TPU0_TCNT1	Initialized	Retained	Retained	Retained
	TPU0_TGR1A	Initialized	Retained	Retained	Retained
	TPU0_TGR1B	Initialized	Retained	Retained	Retained
	TPU0_TGR1C	Initialized	Retained	Retained	Retained
	TPU0_TGR1D	Initialized	Retained	Retained	Retained
	TPU0_TCR2	Initialized	Retained	Retained	Retained
	TPU0_TMDR2	Initialized	Retained	Retained	Retained
	TPU0_TIOR2	Initialized	Retained	Retained	Retained
	TPU0_TIER2	Initialized	Retained	Retained	Retained
	TPU0_TSR2	Initialized	Retained	Retained	Retained
	TPU0_TCNT2	Initialized	Retained	Retained	Retained
	TPU0_TGR2A	Initialized	Retained	Retained	Retained
	TPU0_TGR2B	Initialized	Retained	Retained	Retained
	TPU0_TGR2C	Initialized	Retained	Retained	Retained
	TPU0_TGR2D	Initialized	Retained	Retained	Retained
	TPU0_TCR3	Initialized	Retained	Retained	Retained
	TPU0_TMDR3	Initialized	Retained	Retained	Retained
	TPU0_TIOR3	Initialized	Retained	Retained	Retained
	TPU0_TIER3	Initialized	Retained	Retained	Retained

Module	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
TPU0	TPU0_TSR3	Initialized	Retained	Retained	Retained
	TPU0_TCNT3	Initialized	Retained	Retained	Retained
	TPU0_TGR3A	Initialized	Retained	Retained	Retained
	TPU0_TGR3B	Initialized	Retained	Retained	Retained
	TPU0_TGR3C	Initialized	Retained	Retained	Retained
	TPU0_TGR3D	Initialized	Retained	Retained	Retained
TPU1	TPU1_TSTR	Initialized	Retained	Retained	Retained
	TPU1_TCR0	Initialized	Retained	Retained	Retained
	TPU1_TMDR0	Initialized	Retained	Retained	Retained
	TPU1_TIOR0	Initialized	Retained	Retained	Retained
	TPU1_TIER0	Initialized	Retained	Retained	Retained
	TPU1_TSR0	Initialized	Retained	Retained	Retained
	TPU1_TCNT0	Initialized	Retained	Retained	Retained
	TPU1_TGR0A	Initialized	Retained	Retained	Retained
	TPU1_TGR0B	Initialized	Retained	Retained	Retained
	TPU1_TGR0C	Initialized	Retained	Retained	Retained
	TPU1_TGR0D	Initialized	Retained	Retained	Retained
	TPU1_TCR1	Initialized	Retained	Retained	Retained
	TPU1_TMDR1	Initialized	Retained	Retained	Retained
	TPU1_TIOR1	Initialized	Retained	Retained	Retained
	TPU1_TIER1	Initialized	Retained	Retained	Retained
	TPU1_TSR1	Initialized	Retained	Retained	Retained
	TPU1_TCNT1	Initialized	Retained	Retained	Retained
	TPU1_TGR1A	Initialized	Retained	Retained	Retained
	TPU1_TGR1B	Initialized	Retained	Retained	Retained
	TPU1_TGR1C	Initialized	Retained	Retained	Retained
	TPU1_TGR1D	Initialized	Retained	Retained	Retained
RTC	R64CNT	Retained	Retained	Retained	Retained
	RSECCNT	Retained	Retained	Retained	Retained
	RMINCNT	Retained	Retained	Retained	Retained
	RHRCNT	Retained	Retained	Retained	Retained

Module	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
RTC	RWKCNT	Retained	Retained	Retained	Retained
	RDAYCNT	Retained	Retained	Retained	Retained
	RMONCNT	Retained	Retained	Retained	Retained
	RYRCNT	Retained	Retained	Retained	Retained
	RSECAR	Retained*	Retained	Retained	Retained
	RMINAR	Retained*	Retained	Retained	Retained
	RHRAR	Retained*	Retained	Retained	Retained
	RWKAR	Retained*	Retained	Retained	Retained
	RDAYAR	Retained*	Retained	Retained	Retained
	RMONAR	Retained*	Retained	Retained	Retained
	RYRAR	Initialized	Retained	Retained	Retained
	RCR1	Initialized	Retained	Retained	Retained
	RCR2	Retained	Retained	Retained	Retained
	RCR3	Initialized	Retained	Retained	Retained
TMU	TSTR	Initialized	Retained	Retained	Retained
	TCOR_0	Initialized	Retained	Retained	Retained
	TCNT_0	Initialized	Retained	Retained	Retained
	TCR_0	Initialized	Retained	Retained	Retained
	TCOR_1	Initialized	Retained	Retained	Retained
	TCNT_1	Initialized	Retained	Retained	Retained
	TCR_1	Initialized	Retained	Retained	Retained
	TCOR_2	Initialized	Retained	Retained	Retained
	TCNT_2	Initialized	Retained	Retained	Retained
	TCR_2	Initialized	Retained	Retained	Retained
CMT	CMSTR	Initialized	Retained	Retained	Retained
	CMCSR_0	Initialized	Retained	Retained	Retained
	CMCNT_0	Initialized	Retained	Retained	Retained
	CMCOR_0	Initialized	Retained	Retained	Retained
	CMCSR_1	Initialized	Retained	Retained	Retained
	CMCNT_1	Initialized	Retained	Retained	Retained
	CMCOR_1	Initialized	Retained	Retained	Retained

Module	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
CMT	CMCSR_2	Initialized	Retained	Retained	Retained
	CMCNT_2	Initialized	Retained	Retained	Retained
	CMCOR_2	Initialized	Retained	Retained	Retained
	CMCSR_3	Initialized	Retained	Retained	Retained
	CMCNT_3	Initialized	Retained	Retained	Retained
	CMCOR_3	Initialized	Retained	Retained	Retained
	CMCSR_4	Initialized	Retained	Retained	Retained
	CMCNT_4	Initialized	Retained	Retained	Retained
	CMCOR_4	Initialized	Retained	Retained	Retained
IIC	ICCR1_0	Initialized	Retained	Retained	Retained
	ICCR2_0	Initialized	Retained	Retained	Retained
	ICMR_0	Initialized	Retained	Retained	Retained
	ICIER_0	Initialized	Retained	Retained	Retained
	ICSR_0	Initialized	Retained	Retained	Retained
	SAR_0	Initialized	Retained	Retained	Retained
	ICDRT_0	Initialized	Retained	Retained	Retained
	ICDRR_0	Initialized	Retained	Retained	Retained
	NF2CYC_0	Initialized	Retained	Retained	Retained
	ICCR1_1	Initialized	Retained	Retained	Retained
	ICCR2_1	Initialized	Retained	Retained	Retained
	ICMR_1	Initialized	Retained	Retained	Retained
	ICIER_1	Initialized	Retained	Retained	Retained
	ICSR_1	Initialized	Retained	Retained	Retained
	SAR_1	Initialized	Retained	Retained	Retained
	ICDRT_1	Initialized	Retained	Retained	Retained
	ICDRR_1	Initialized	Retained	Retained	Retained
	NF2CYC_1	Initialized	Retained	Retained	Retained
SIOF	SIMDR	Initialized	Retained	Retained	Retained
	SISCR	Initialized	Retained	Retained	Retained
	SITDAR	Initialized	Retained	Retained	Retained
	SIRDAR	Initialized	Retained	Retained	Retained

Module	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
SIOF	SICDAR	Initialized	Retained	Retained	Retained
	SICTR	Initialized	Retained	Retained	Retained
	SIFCTR	Initialized	Retained	Retained	Retained
	SISTR	Initialized	Retained	Retained	Retained
	SIIER	Initialized	Retained	Retained	Retained
	SITDR	Undefined	Retained	Retained	Retained
	SIRDR	Undefined	Retained	Retained	Retained
	SITCR	Initialized	Retained	Retained	Retained
	SIRCR	Undefined	Retained	Retained	Retained
SCIF	SCSMR0	Initialized	Retained	Retained	Retained
	SCBRR0	Initialized	Retained	Retained	Retained
	SCSCR0	Initialized	Retained	Retained	Retained
	SCFTDR0	Initialized	Retained	Retained	Retained
	SCFSR0	Initialized	Retained	Retained	Retained
	SCFRDR0	Initialized	Retained	Retained	Retained
	SCFCR0	Initialized	Retained	Retained	Retained
	SCFDR0	Initialized	Retained	Retained	Retained
	SCLSR0	Initialized	Retained	Retained	Retained
	SCSMR1	Initialized	Retained	Retained	Retained
	SCBRR1	Initialized	Retained	Retained	Retained
	SCSCR1	Initialized	Retained	Retained	Retained
	SCFTDR1	Initialized	Retained	Retained	Retained
	SCFSR1	Initialized	Retained	Retained	Retained
	SCFRDR1	Initialized	Retained	Retained	Retained
	SCFCR1	Initialized	Retained	Retained	Retained
	SCFDR1	Initialized	Retained	Retained	Retained
	SCLSR1	Initialized	Retained	Retained	Retained
	SCSMR2	Initialized	Retained	Retained	Retained
	SCBRR2	Initialized	Retained	Retained	Retained
	SCSCR2	Initialized	Retained	Retained	Retained
	SCFTDR2	Initialized	Retained	Retained	Retained

Module	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
SCIF	SCFSR2	Initialized	Retained	Retained	Retained
	SCFRDR2	Initialized	Retained	Retained	Retained
	SCFCR2	Initialized	Retained	Retained	Retained
	SCFDR2	Initialized	Retained	Retained	Retained
	SCLSR2	Initialized	Retained	Retained	Retained
	SCSMR3	Initialized	Retained	Retained	Retained
	SCBRR3	Initialized	Retained	Retained	Retained
	SCSCR3	Initialized	Retained	Retained	Retained
	SCFTDR3	Initialized	Retained	Retained	Retained
	SCFSR3	Initialized	Retained	Retained	Retained
	SCFRDR3	Initialized	Retained	Retained	Retained
	SCFCR3	Initialized	Retained	Retained	Retained
	SCFDR3	Initialized	Retained	Retained	Retained
	SCLSR3	Initialized	Retained	Retained	Retained
SCIFA	SCASMR4	Initialized	Retained	Retained	Retained
	SCABRR4	Initialized	Retained	Retained	Retained
	SCASCR4	Initialized	Retained	Retained	Retained
	SCATDSR4	Initialized	Retained	Retained	Retained
	SCAFER4	Initialized	Retained	Retained	Retained
	SCASSR4	Initialized	Retained	Retained	Retained
	SCAFCR4	Initialized	Retained	Retained	Retained
	SCAFDR4	Initialized	Retained	Retained	Retained
	SCAFTDR4	Initialized	Retained	Retained	Retained
	SCAFRDR4	Initialized	Retained	Retained	Retained
	SCASMR5	Initialized	Retained	Retained	Retained
	SCABRR5	Initialized	Retained	Retained	Retained
	SCASCR5	Initialized	Retained	Retained	Retained
	SCATDSR5	Initialized	Retained	Retained	Retained
	SCAFER5	Initialized	Retained	Retained	Retained
	SCASSR5	Initialized	Retained	Retained	Retained
	SCAFCR5	Initialized	Retained	Retained	Retained

Module	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
SCIFA	SCAFDR5	Initialized	Retained	Retained	Retained
	SCAFTDR5	Initialized	Retained	Retained	Retained
	SCAFRDR5	Initialized	Retained	Retained	Retained
IrDA	IRIF0_INT2	Initialized	Retained	Retained	Retained
	IRIF0_RINTCLR	Initialized	Retained	Retained	Retained
	IRIF0_TINTCLR	Initialized	Retained	Retained	Retained
	IRIF0_SIR0	Initialized	Retained	Retained	Retained
	IRIF0_SIR1	Initialized	Retained	Retained	Retained
	IRIF0_SIR2	Initialized	Retained	Retained	Retained
	IRIF0_SIR3	Initialized	Retained	Retained	Retained
	IRIF0_SIR_FRM	Initialized	Retained	Retained	Retained
	IRIF0_SIR_EOF	Initialized	Retained	Retained	Retained
	IRIF0_SIR_FLG	Initialized	Retained	Retained	Retained
	IRIF0_SIR_STS2	Initialized	Retained	Retained	Retained
	IRIF0_UART0	Initialized	Retained	Retained	Retained
	IRIF0_UART1	Initialized	Retained	Retained	Retained
	IRIF0_UART2	Initialized	Retained	Retained	Retained
	IRIF0_UART3	Initialized	Retained	Retained	Retained
	IRIF0_UART4	Initialized	Retained	Retained	Retained
	IRIF0_UART5	Initialized	Retained	Retained	Retained
	IRIF0_UART6	Initialized	Retained	Retained	Retained
	IRIF0_UART7	Initialized	Retained	Retained	Retained
	IRIF0_CRC0	Initialized	Retained	Retained	Retained
	IRIF0_CRC1	Initialized	Retained	Retained	Retained
	IRIF0_CRC2	Initialized	Retained	Retained	Retained
	IRIF0_CRC3	Initialized	Retained	Retained	Retained
	IRIF0_CRC4	Initialized	Retained	Retained	Retained
	IRIF1_INT2	Initialized	Retained	Retained	Retained
	IRIF1_RINTCLR	Initialized	Retained	Retained	Retained
	IRIF1_TINTCLR	Initialized	Retained	Retained	Retained
	IRIF1_SIR0	Initialized	Retained	Retained	Retained

Module	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
IrDA	IRIF1_SIR1	Initialized	Retained	Retained	Retained
	IRIF1_SIR2	Initialized	Retained	Retained	Retained
	IRIF1_SIR3	Initialized	Retained	Retained	Retained
	IRIF1_SIR_FRM	Initialized	Retained	Retained	Retained
	IRIF1_SIR_EOF	Initialized	Retained	Retained	Retained
	IRIF1_SIR_FLG	Initialized	Retained	Retained	Retained
	IRIF1_SIR_STS2	Initialized	Retained	Retained	Retained
	IRIF1_UART0	Initialized	Retained	Retained	Retained
	IRIF1_UART1	Initialized	Retained	Retained	Retained
	IRIF1_UART2	Initialized	Retained	Retained	Retained
	IRIF1_UART3	Initialized	Retained	Retained	Retained
	IRIF1_UART4	Initialized	Retained	Retained	Retained
	IRIF1_UART5	Initialized	Retained	Retained	Retained
	IRIF1_UART6	Initialized	Retained	Retained	Retained
	IRIF1_UART7	Initialized	Retained	Retained	Retained
	IRIF1_CRC0	Initialized	Retained	Retained	Retained
	IRIF1_CRC1	Initialized	Retained	Retained	Retained
	IRIF1_CRC2	Initialized	Retained	Retained	Retained
	IRIF1_CRC3	Initialized	Retained	Retained	Retained
	IRIF1_CRC4	Initialized	Retained	Retained	Retained
SIM	SCSMR	Initialized	Retained	Retained	Retained
	SCBRR	Initialized	Retained	Retained	Retained
	SCSCR	Initialized	Retained	Retained	Retained
	SCTDR	Initialized	Retained	Retained	Retained
	SCSSR	Initialized	Retained	Retained	Retained
	SCRDR	Initialized	Retained	Retained	Retained
	SCSCMR	Initialized	Retained	Retained	Retained
	SCSC2R	Initialized	Retained	Retained	Retained
	SCWAIT	Initialized	Retained	Retained	Retained
	SCGRD	Initialized	Retained	Retained	Retained
	SCSMPL	Initialized	Retained	Retained	Retained
	SCDMAEN	Initialized	Retained	Retained	Retained

Module	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
ADC	ADDRA	Initialized	Initialized	Initialized	Retained
	ADDRB	Initialized	Initialized	Initialized	Retained
	ADDRC	Initialized	Initialized	Initialized	Retained
	ADDRD	Initialized	Initialized	Initialized	Retained
	ADCSR	Initialized	Initialized	Initialized	Retained
DAC	DADR0	Initialized	Retained	Retained	Retained
	DADR1	Initialized	Retained	Retained	Retained
	DACR	Initialized	Retained	Retained	Retained
PORT	PADR	Initialized	Retained	—	Retained
	PBDR	Initialized	Retained	—	Retained
	PCDR	Initialized	Retained	—	Retained
	PDDR	Initialized	Retained	—	Retained
	PEDR	Initialized	Retained	—	Retained
	PFDR	Initialized	Retained	—	Retained
	PGDR	Initialized	Retained	—	Retained
	PHDR	Initialized	Retained	—	Retained
	PJDR	Initialized	Retained	—	Retained
	PKDR	Initialized	Retained	—	Retained
	PLDR	Initialized	Retained	—	Retained
	PMDR	Initialized	Retained	—	Retained
	PNDR	Initialized	Retained	—	Retained
	PQDR	Initialized	Retained	—	Retained
	PRDR	Initialized	Retained	—	Retained
	PSDR	Initialized	Retained	—	Retained
	PTDR	Initialized	Retained	—	Retained
PFC	PACR	Initialized	Retained	—	Retained
	PBCR	Initialized	Retained	—	Retained
	PCCR	Initialized	Retained	—	Retained
	PDCR	Initialized	Retained	—	Retained
	PECR	Initialized	Retained	—	Retained
	PFCR	Initialized	Retained	—	Retained

Module	Register Abbreviation	Power-On Reset	Software Standby	Module Standby	Sleep
PFC	PGCR	Initialized	Retained	—	Retained
	PHCR	Initialized	Retained	—	Retained
	PJCR	Initialized	Retained	—	Retained
	PKCR	Initialized	Retained	—	Retained
	PLCR	Initialized	Retained	—	Retained
	PMCR	Initialized	Retained	—	Retained
	PNCR	Initialized	Retained	—	Retained
	PQCR	Initialized	Retained	—	Retained
	PRCR	Initialized	Retained	—	Retained
	PSCR	Initialized	Retained	—	Retained
	PTCR	Initialized	Retained	—	Retained
	PSELA	Initialized	Retained	—	Retained
	PSELB	Initialized	Retained	—	Retained
	PSELC	Initialized	Retained	—	Retained
	HIZCRA	Initialized	Retained	—	Retained
	HIZCRB	Initialized	Retained	—	Retained
	HIZCRC	Initialized	Retained	—	Retained
	HIZCRD	Initialized	Retained	—	Retained
	HIZCRE	Initialized	Retained	—	Retained
	HIZCRF	Initialized	Retained	—	Retained
	PULCR	Initialized	Retained	—	Retained
	PINTCRA	Initialized	Retained	—	Retained
	PINTCRB	Initialized	Retained	—	Retained

Table 32.2 Register States in Each Operating Mode (3)

Module	Register Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
UBC	CBR0	Initialized	Retained	Retained	Retained
	CRR0	Initialized	Retained	Retained	Retained
	CAR0	Initialized	Retained	Retained	Retained
	CAMR0	Initialized	Retained	Retained	Retained
	CBR1	Initialized	Retained	Retained	Retained
	CRR1	Initialized	Retained	Retained	Retained
	CAR1	Initialized	Retained	Retained	Retained
	CAMR1	Initialized	Retained	Retained	Retained
	CDR1	Initialized	Retained	Retained	Retained
	CDMR1	Initialized	Retained	Retained	Retained
	CETR1	Initialized	Retained	Retained	Retained
	CCMFR	Initialized	Retained	Retained	Retained
	CBCR	Initialized	Retained	Retained	Retained
H-UDI	SDIR	Initialized	Retained	Retained	Retained
	SDDR/SDDRH	Initialized	Retained	Retained	Retained
	SDDRL	Initialized	Retained	Retained	Retained
	SDINT	Initialized	Retained	Retained	Retained
	SDBPR	—	—	—	—

Section 33 Electrical Characteristics

33.1 Absolute Maximum Ratings

Table 33.1 shows the absolute maximum ratings.

Table 33.1 Absolute Maximum Ratings

Item		Symbol	Value	Unit
Power supply voltage (I/O)		$V_{CC}Q$	−0.3 to 4.6	V
Power supply voltage (internal)		V_{CC}, V_{CC}–PLL1, V_{CC}–PLL2	−0.3 to 1.8	V
Input voltage		V_{in}	−0.3 to $V_{CC}Q$ + 0.3	V
Analog power supply voltage		AV_{CC}	−0.3 to 4.6	V
Analog input voltage		V_{AN}	−0.3 to AV_{CC} + 0.3	V
Operating temperature*1	Standard temperature range product	T_{opr}	−20 to 75	°C
	Extended temperature range product		−40 to 85	
Storage temperature		T_{stg}	−55 to 125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

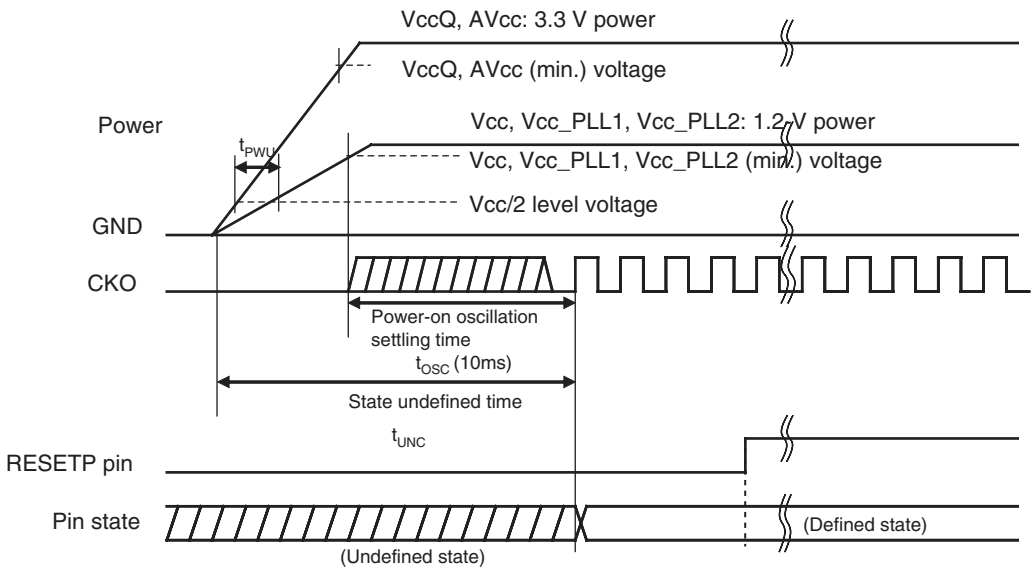
Note: 1. For a list of the operating temperatures of individual product versions, see table 1.4, Product Lineup.

33.2 Power-On and Power-Off Order

(1) Order of turning on 1.2 V power (V_{CC}, V_{CC}_PLL1, and V_{CC}_PLL2) and 3.3 V power ($V_{CC}Q$, AV_{CC})

- First turn on the 3.3 V power, then turn on the 1.2 V power within 1 ms. This interval should be as short as possible.
- Do not allow the voltage of the 1.2 V supply system to be above that of the 3.3 V supply system.
- Until power-on oscillation settling time (10 ms) elapses after all powers have reached respective specified minimum voltages, internal circuits remain unsettled, and so pin states are also undefined. The system design must ensure that these undefined states do not cause erroneous system operation.

  Waveforms at power-on are shown in the following figure.



Note: * Except power/GND, clock related, and analog pins.

Table 33.2 Recommended Timing in Power-On

Item	Symbol	Value	Unit
Time difference between the power-on of ($V_{CC}Q$, AV_{CC}) and (V_{CC}, V_{CC}_PLL1, V_{CC}_PLL2) levels	t_{PWU}	1 (max.)	ms
State undefined time	t_{UNC}	100 (max.)	ms

Note: * The table shown above is recommended values, so they represent guidelines rather than strict requirements.

The state undefined time is the period during which individual power supply voltages rise excessively.

When the power-on oscillation settling time (10 ms) has elapsed after all power supply voltages have reached the respective specified minimum levels, a power-on reset ($\overline{\text{RESETP}}$) can successfully be accepted. Successful acceptance of a power-on reset will settle the pin states. Ensure that the state undefined time is less than or equal to 100 ms.

(2) Power-off order

- In the reverse order of powering-on, first turn off the 1.2 V power, then turn off the 3.3 V power within 10 ms. This interval should be as short as possible. The system design must ensure that the states of pins or undefined period of an internal state do not cause erroneous system operation.
- Do not allow the voltage of the 1.2 V supply system to be above that of the 3.3 V supply system.
- Pin states are undefined while only the 1.5 V power is off. The system design must ensure that these undefined states do not cause erroneous system operation.

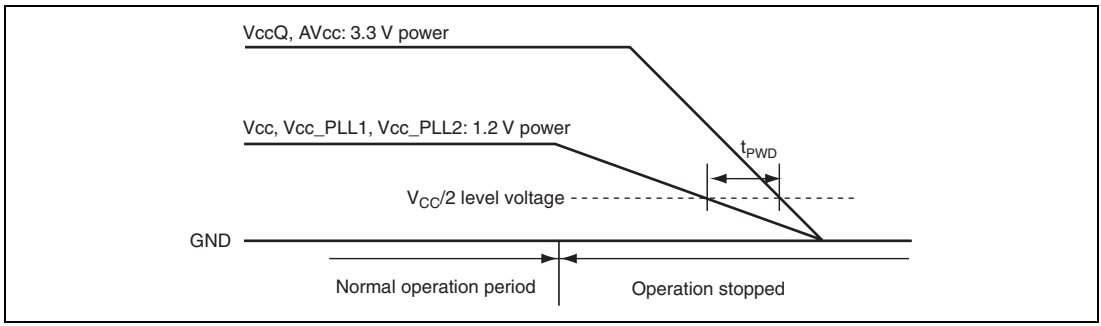


Table 33.3 Recommended Timing in Power-Off

Item	Symbol	Maximum Value	Unit
Time difference between the power-off of ($V_{CC}Q$, AV_{CC}) and (V_{CC}, V_{CC}_PLL1, V_{CC}_PLL2) levels	t_{PWD}	10	ms

Note: * The table shown above is recommended values, so they represent guidelines rather than strict requirements.

33.3 DC Characteristics

Tables 33.4 and 33.5 show the DC characteristics.

Table 33.4 DC Characteristics (1) [Common]

Conditions: T_a = –20°C to +75°C (Standard temperature range product),
T_a = –40°C to +85°C (Extended temperature range product)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage*1*4		$V_{cc}Q$	3.0	3.3	3.6	V	
		V_{cc} V_{cc}_PLL1 V_{cc}_PLL2	1.1	1.2	1.3	V	
Analog (A/D, D/A) power supply voltage*2		AV_{cc}	3.0	3.3	3.6	V	When not in use, the same voltage as $V_{cc}Q$.
Analog (A/D, D/A) power supply current	During A/D conversion	AI_{cc}	—	1.3	2.0	mA	
	During A/D and D/A conversion		—	1.6	2.5	mA	
	Idle		—	1.0	5.0	μA	T_a = 25°C
Supply current*3	Normal operation	I_{cc}	—	135	275	mA	V_{cc} = 1.2V Iϕ = 266.7 MHz
			—	110	220	mA	V_{cc} = 1.2V Iϕ = 200 MHz
		$I_{cc}Q$	—	30	40	mA	$V_{cc}Q$ = 3.3 V Bϕ = 33.4 MHz
	Sleep mode	I_{cc}	—	30	50	mA	When sleep mode is entered after a power-on reset: $V_{cc}Q$ = 3.3 V Bϕ = 33.4 MHz
		$I_{cc}Q$	—	20	25	mA	
	Standby mode	I_{cc}	—	0.4	3	mA	T_a = 25°C $V_{cc}Q$ = 3.3 V V_{cc} = 1.2 V
		$I_{cc}Q$	—	15	30	μA	
Input leakage current	All input pins	$\lvert I_{in} \rvert$	—	—	1.0	μA	V_{in} = 0.5 to $V_{cc}Q$ – 0.5 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Three-state leakage current	Input/output pins, all output pins (off state)	$\lvert I_{STI} \rvert$	—	—	1.0	μA	V_{in} = 0.5 to $V_{CC}Q$ – 0.5 V
Pull-up/ pull-down resistance	Port pins	P_{pull}	20	50	120	kΩ	
Pin capacitance	All pins	C	—	—	10	pF	

Notes: 1. When the PLL is not used, the V_{CC}_PLL1, V_{CC}_PLL2, and V_{SS}_PLL1, V_{SS}_PLL2 should be supplied.
2. AV_{CC} should satisfy the condition $V_{CC}Q - 0.3\ V \le AV_{CC} \le V_{CC}Q + 0.3\ V$. Even when the A/D converter and D/A converter are not used, AVcc and AVss should not be open. Connect AVcc to VccQ, and AVss to VssQ.
3. Supply current values are for V_{IH}min = V_{CC} – 0.5 V and V_{IL}max = 0.5 V with all output pins unloaded.
4. The same voltage should be supplied as Vcc, Vcc_PLL1 and Vcc_PLL2.

Table 33.4 DC Characteristics (2-a) [Except for I²C Related Pins]

Conditions: T_a = –20°C to +75°C (Standard temperature range product),
T_a = –40°C to +85°C (Extended temperature range product)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	$\overline{\text{RESETP}}$, MD0, MD1, MD3, MD5, $\overline{\text{TRST}}$, MPMD, EXTAL, NMI	V_{IH}	$V_{CC}Q \times 0.9$	—	$V_{CC}Q + 0.3$	V	
	PTM0, PTM1, PTL4, PTL5, PTL6, PTL7		2.0	—	$AV_{CC} + 0.3$		
	Other input pins		2.0	—	$V_{CC}Q + 0.3$		
Input low voltage	$\overline{\text{RESETP}}$, MD0, MD1, MD3, MD5, $\overline{\text{TRST}}$, MPMD, EXTAL, NMI	V_{IL}	-0.3	—	$V_{CC}Q \times 0.1$	V	
	PTM0, PTM1, PTL4, PTL5, PTL6, PTL7		-0.3	—	$AV_{CC} \times 0.2$		
	Other input pins		-0.3	—	$V_{CC}Q \times 0.2$		

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	V_{OH}	2.4	—	—	V	$V_{CC}Q$ = 3.0 V I_{OH} = -0.2 mA
		2.0	—	—		$V_{CC}Q$ = 3.0 V I_{OL} = -2.0 mA
Output low voltage	V_{OL}	—	—	0.6	V	$V_{CC}Q$ = 3.6 V I_{OL} = -2.0 mA

Table 33.4 DC Characteristics (2-b) [I²C Related Pins*]

Conditions: T_a = –20°C to +75°C (Standard temperature range product),
T_a = –40°C to +85°C (Extended temperature range product)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage	$V_{CC}Q$	3.0	3.3	3.6	V	
Input high voltage	V_{IH}	$V_{CC}Q \times 0.7$	—	$V_{CC}Q + 0.3$	V	
Input low voltage	V_{IL}	–0.3	—	$V_{CC}Q \times 0.3$	V	
Output low voltage	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
Permissible output low current	I_{OL}	—	—	10	mA	

Note: * The IIC_SCL and IIC_SDA pins (open-drain pins).

Table 33.5 Permissible Output Current Values

Conditions: T_a = –20°C to +75°C (Standard temperature range product),
T_a = –40°C to +85°C (Extended temperature range product)

Item	Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	I_{OL}	—	—	2.0	mA
Permissible output low current (total)	$\sum I_{OL}$	—	—	120	mA
Permissible output high current (per pin)	$-I_{OH}$	—	—	2.0	mA
Permissible output high current (total)	$\sum (-I_{OH})$	—	—	40	mA

Note: * To ensure chip reliability, do not exceed the output current values given in table 33.5.

33.4 AC Characteristics

The input of this LSI is a synchronous input. The setup hold time of each input signal should be kept unless any notice.

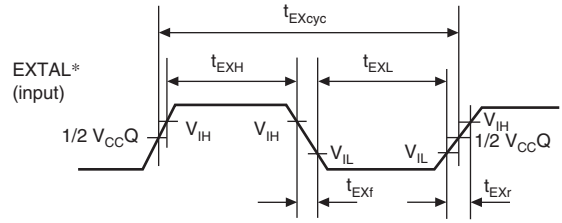
Table 33.6 Maximum Operating Frequencies

Item		Symbol	Min.	Typ.	Max.	Unit	Remarks
Operating frequency	CPU clock (Iϕ)	f	24	—	266.7	MHz	266 MHz version
			24	—	200		200 MHz version
	SH clock (Sϕ)		24	—	133.4		
	Bus clock (Bϕ)		24	—	66.7		
	Peripheral clock (Pϕ)		8	—	33.4		
	PLL circuit output clock		75	—	266.7		266 MHz version
			75	—	200		200 MHz version

33.4.1 Clock Timing

Table 33.7 Clock Timing

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency	f_{EX}	10	66.67	MHz	33.1
EXTAL clock input cycle time	t_{EXcyc}	15	100	ns	
EXTAL clock input low pulse width	t_{EXL}	7	—	ns	
EXTAL clock input high pulse width	t_{EXH}	7	—	ns	
EXTAL clock input rise time	t_{EXr}	—	4	ns	
EXTAL clock input fall time	t_{EXf}	—	4	ns	
CKO clock output frequency	f_{OP}	20	66.7	MHz	33.2
CKO clock output cycle time	t_{cyc}	15	50	ns	
CKO clock output low pulse width	t_{CKOL}	3	—	ns	
CKO clock output high pulse width	t_{CKOH}	3	—	ns	
CKO clock output rise time	t_{CKOr}	—	3	ns	
CKO clock output fall time	t_{CKOf}	—	3	ns	
$\overline{\text{RESETP}}$ power-on oscillation settling time	t_{OSC}	10	—	ms	33.3



Note: * When clock is input from EXTAL pin

Figure 33.1 EXTAL Clock Input Timing

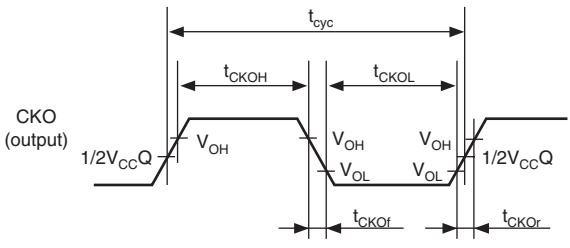
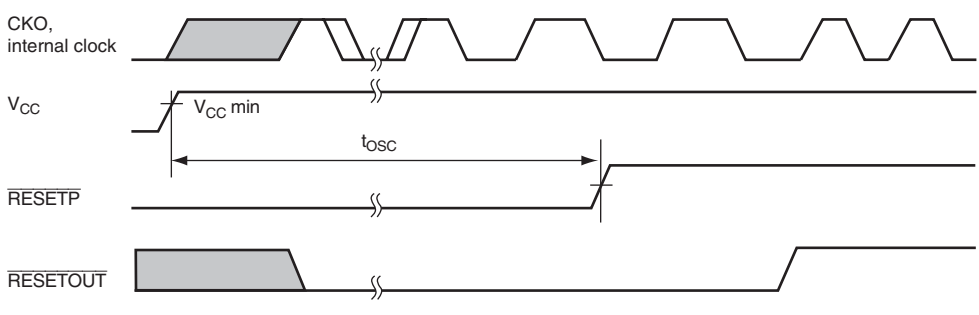


Figure 33.2 CKIO Clock Output Timing



Note: Oscillation settling time when on-chip oscillator is used

Figure 33.3 Power-On Oscillation Settling Time

33.4.2 Control Signal Timing

Table 33.8 Control Signal Timing

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{RESETP}}$ pulse width	t_{RESPW}	1*[2]	—	ms	33.4
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	$1/2t_{cyc}$ + 7	—	ns	33.6
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	$1/2t_{cyc}$ + 2	—	ns	
NMI setup time*[1]	t_{NMIS}	8	—	ns	33.5
NMI hold time	t_{NMIH}	3	—	ns	
IRQ7 to IRQ0 setup time*[1]	t_{IRQS}	8	—	ns	
IRQ7 to IRQ0 hold time	t_{IRQH}	3	—	ns	
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	$1/2t_{cyc}$ + 13	ns	33.6
Bus tri-state delay time 1	t_{BOFF1}	0	30	ns	33.6
Bus tri-state delay time 2	t_{BOFF2}	0	30	ns	33.7
Bus buffer on time 1	t_{BON1}	0	30	ns	
Bus buffer on time 2	t_{BON2}	0	30	ns	

Notes: 1. NMI, and IRQ7 to IRQ0 are asynchronous. Changes are detected at the clock rise when the setup time shown is used. If the setup time cannot be used, detection may be delayed until the next clock rises.
2. In standby mode, $t_{RESPW} = t_{OSC}$ (10 ms).
3. t_{cyc} means the external bus clock (Bϕ) cycle time.

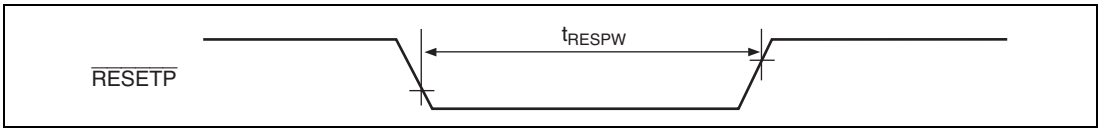


Figure 33.4 Reset Input Timing

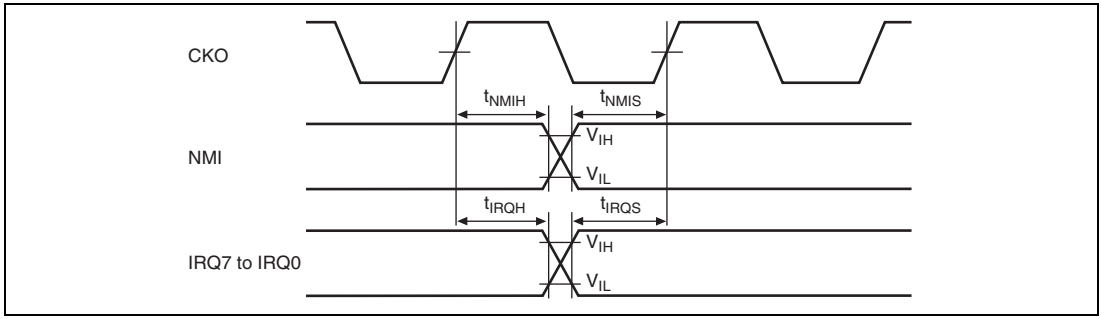
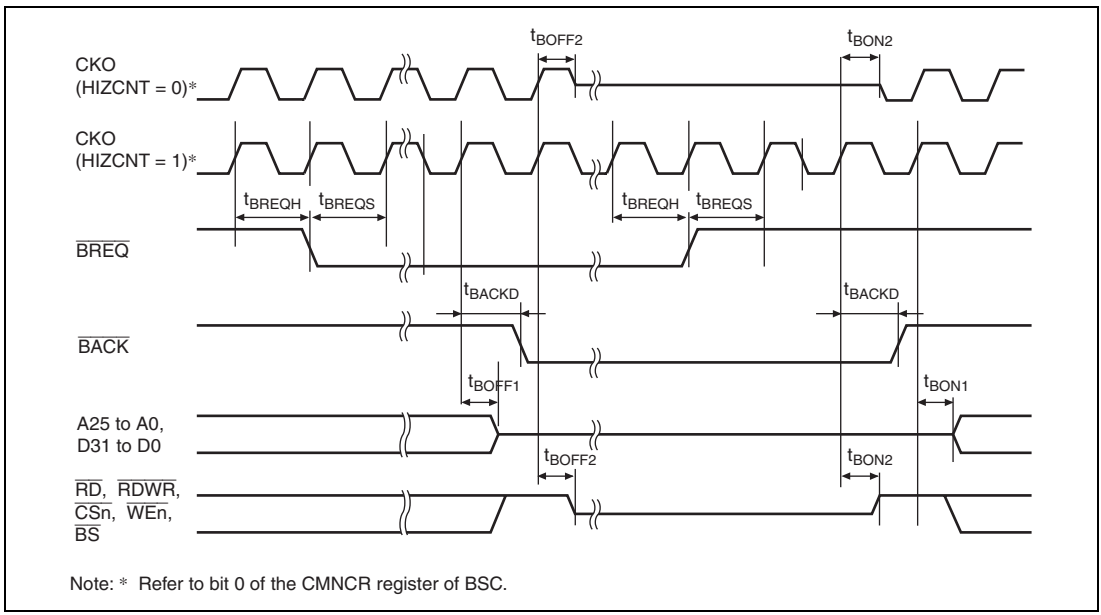


Figure 33.5 Interrupt Signal Input Timing



Note: * Refer to bit 0 of the CMNCR register of BSC.

Figure 33.6 Bus Release Timing

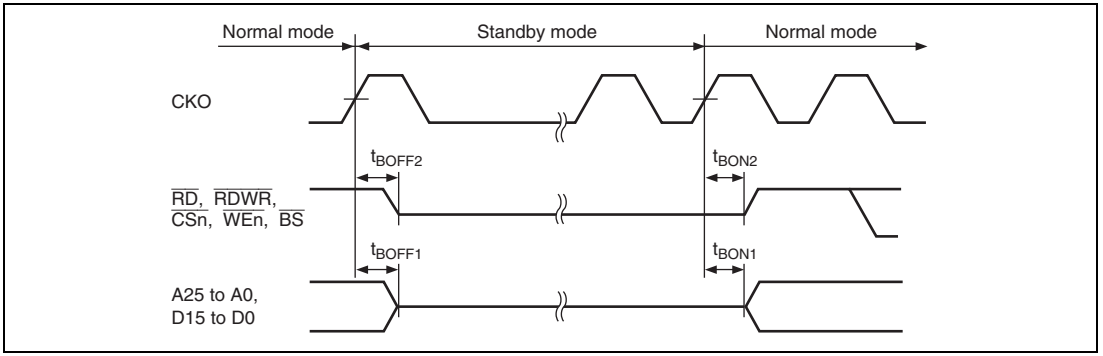


Figure 33.7 Pin Drive Timing at Standby

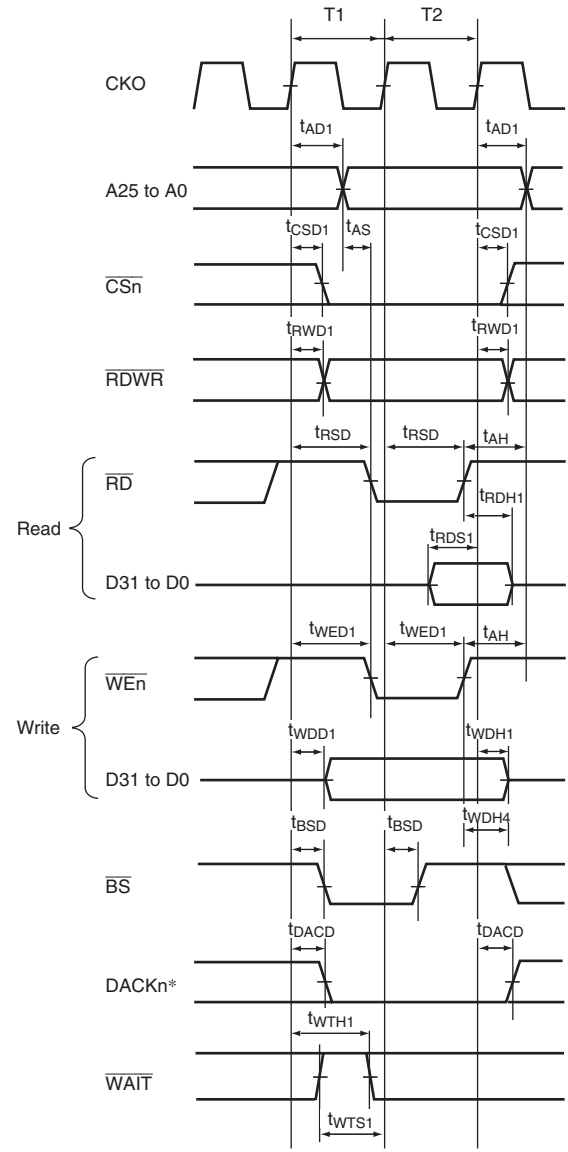
33.4.3 AC Bus Timing

Table 33.9 Bus Timing

Item	Symbol	Min.	Max.	Unit	Figure
Address delay time 1	t_{AD1}	1	10	ns	33.8 to 33.38
Address delay time 2	t_{AD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10	ns	33.15
Address setup time	t_{AS}	0	—	ns	33.8 to 33.15
Address hold time	t_{AH}	0	—	ns	33.8, 33.9
$\overline{BS}$ delay time	t_{BSD}	—	10	ns	33.8 to 33.34
$\overline{CS}$ delay time 1	t_{CSD1}	1	10	ns	33.8 to 33.38
Read/write delay time 1	t_{RWD1}	1	10	ns	33.8 to 33.38
Read strobe delay time	t_{RSD}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10	ns	33.8 to 33.15, 33.35, 33.36
Read data setup time 1	t_{RDS1}	$1/2t_{cyc}$ + 7	—	ns	33.8 to 33.14, 33.33 to 33.38
Read data setup time 2	t_{RDS2}	7	—	ns	33.16 to 33.19, 33.24 to 33.26, 33.33, 33.34
Read data setup time3	t_{RDS3}	$1/2t_{cyc}$ + 7	—	ns	33.15
Read data hold time 1	t_{RDH1}	0	—	ns	33.8 to 33.14, 33.33 to 33.38
Read data hold time 2	t_{RDH2}	2	—	ns	33.16 to 33.19, 33.24 to 33.26, 33.33, 33.34
Read data hold time3	t_{RDH3}	0	—	ns	33.15
Write enable delay time 1	t_{WED1}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10	ns	33.8 to 33.13, 33.37, 33.38
Write enable delay time 2	t_{WED2}	—	10	ns	33.14
Write data delay time 1	t_{WDD1}	—	10	ns	33.8 to 33.14, 33.35 to 33.38
Write data delay time 2	t_{WDD2}	—	10	ns	33.20 to 33.23, 33.27 to 33.29, 33.33, 33.34

Item	Symbol	Min.	Max.	Unit	Figure
Write data hold time 1	t_{WDH1}	1	—	ns	33.8 to 33.14, 33.33 to 33.38
Write data hold time 2	t_{WDH2}	1	—	ns	33.20 to 33.23, 33.27 to 33.29, 33.33, 33.34
Write data hold time4	t_{WDH4}	0	—	ns	33.8
Write data hold time5	t_{WDH5}	1	—	ns	33.35 to 33.38
$\overline{WAIT}$ setup time 1	t_{WTS1}	$1/2t_{cyc} + 7$	—	ns	33.8 to 33.15, 33.36, 33.38
$\overline{WAIT}$ hold time 1	t_{WTH1}	$1/2t_{cyc} + 2$	—	ns	33.8 to 33.15, 33.36, 33.38
$\overline{RAS}$ delay time 1	t_{RASD1}	1	10	ns	33.16 to 33.34
$\overline{CAS}$ delay time 1	t_{CASD1}	1	10	ns	33.16 to 33.34
DQM delay time 1	t_{DQMD1}	1	10	ns	33.16 to 33.34
CKE delay time 1	t_{CKED1}	1	10	ns	33.31 to 33.34
DACK delay time	t_{DACD}	—	13	ns	33.8 to 33.33
$\overline{ICIORD}$ delay time	t_{ICRSD}	—	$1/2t_{cyc} + 10$	ns	33.37, 33.38
$\overline{ICIOWR}$ delay time	t_{ICWSD}	—	$1/2t_{cyc} + 10$	ns	33.37, 33.38
$\overline{IOIS16}$ setup time	t_{IO16S}	$1/2t_{cyc} + 6$	—	ns	33.38
$\overline{IOIS16}$ hold time	t_{IO16H}	$1/2t_{cyc} + 4$	—	ns	33.38
$\overline{REFOUT}$, $\overline{IRQOUT}$ delay time	t_{REFOD}	—	$1/2t_{cyc} + 10$	ns	33.39

33.4.4 Basic Timing



Note: * Waveform when active low is specified for DACKn.

Figure 33.8 Basic Bus Cycle in Normal Space (No Wait)

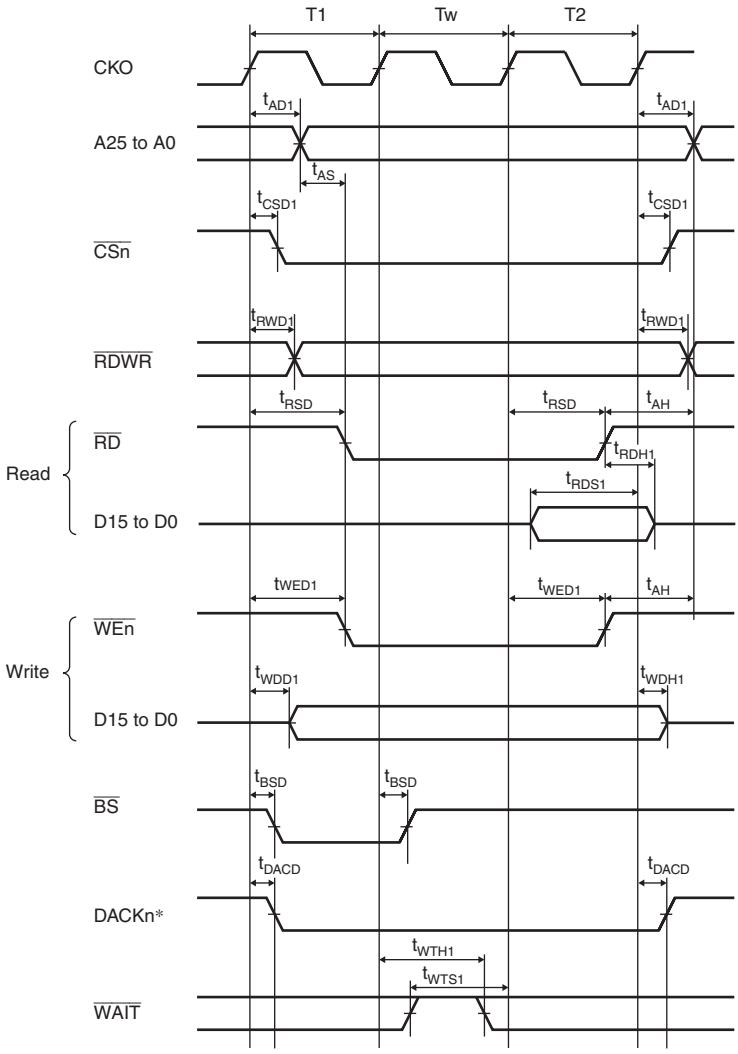
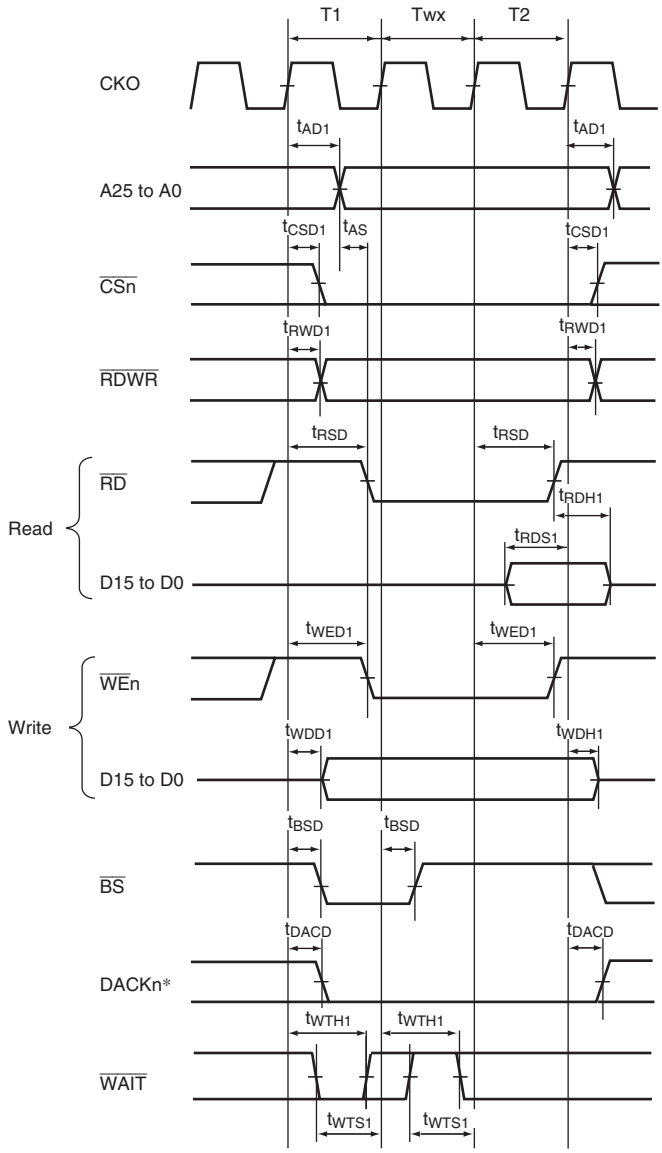


Figure 33.9 Basic Bus Cycle in Normal Space (Software Wait 1)



Note: * Waveform when active low is specified for DACKn.

Figure 33.10 Basic Bus Cycle in Normal Space (Asynchronous External Wait 1 Input)

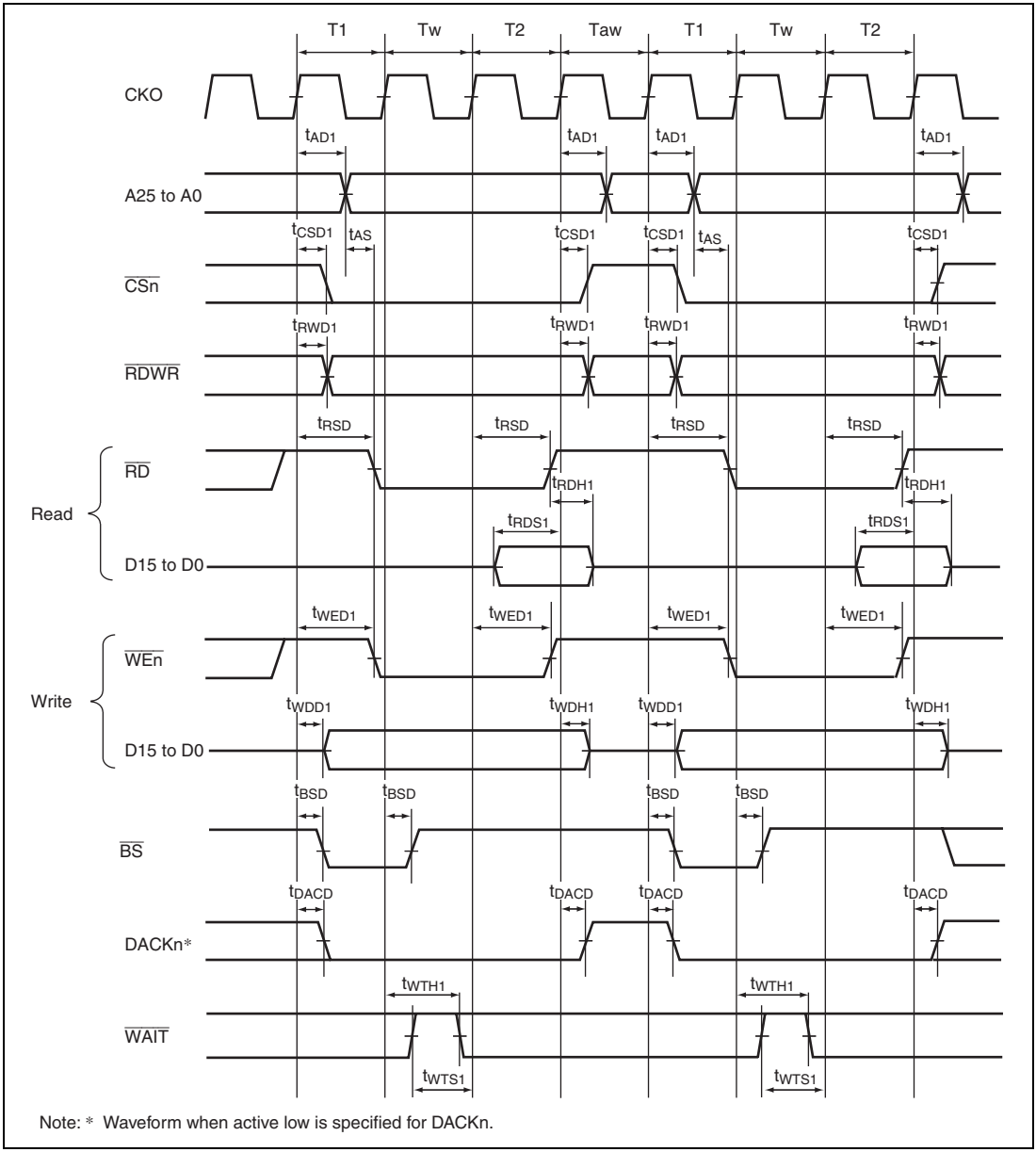


Figure 33.11 Basic Bus Cycle in Normal Space
(Software Wait 1, Asynchronous External Wait Valid (WM Bit = 0), No Idle Cycle)

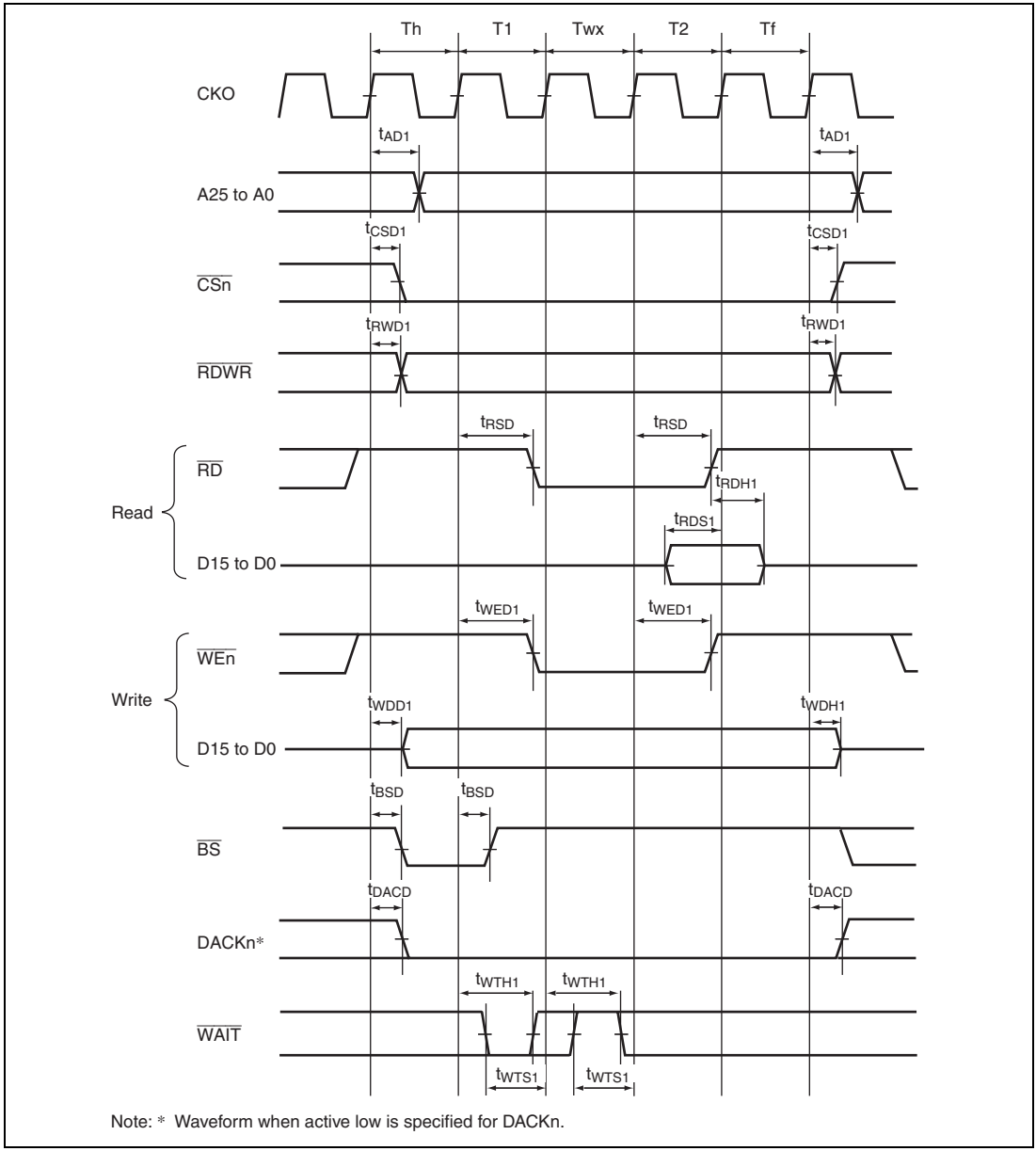
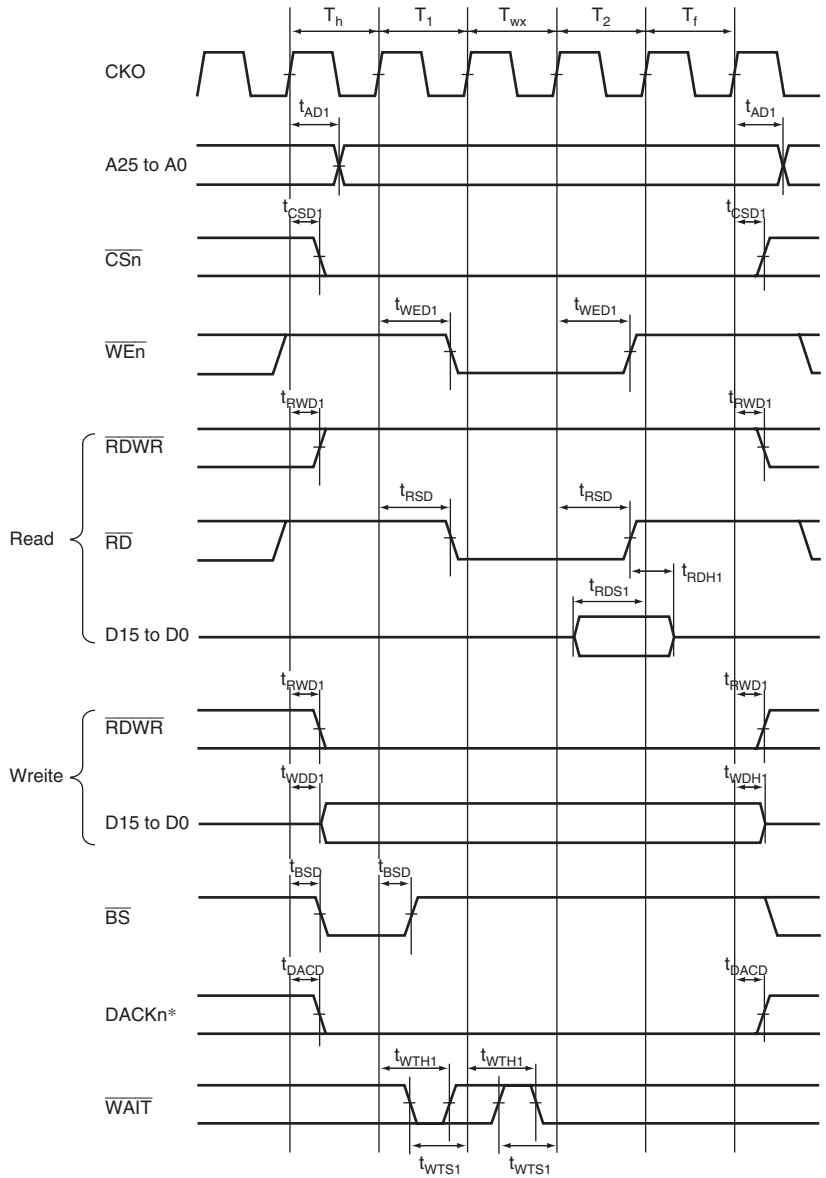
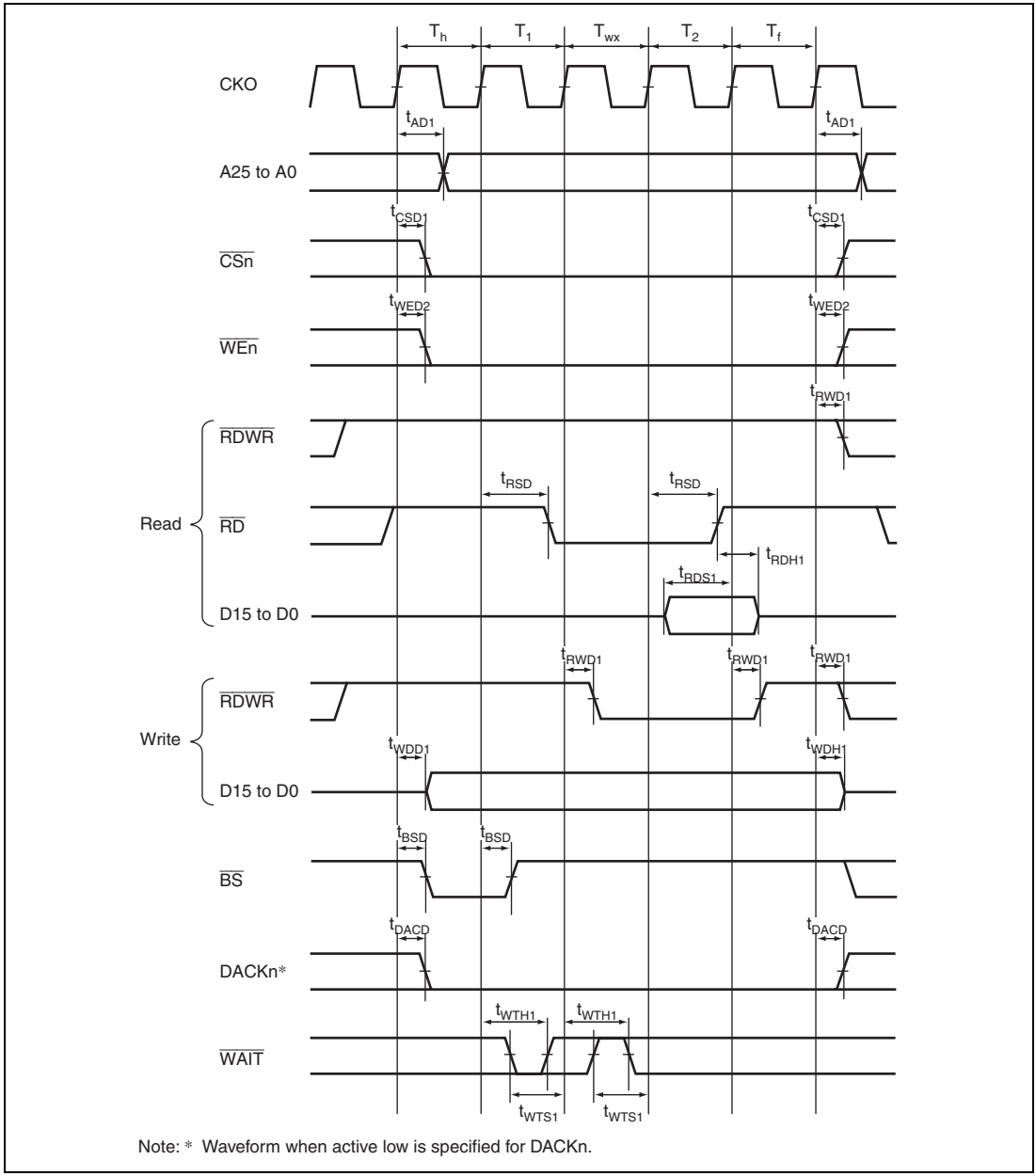


Figure 33.12 CS Extended Bus Cycle in Normal Space (CSnWCR.SW[1:0] =B'01, CSnWCR.HW[1:0] =B'01, External Wait 1 Input)



Note: * Waveform when active low is specified for DACKn.

Figure 33.13 Bus Cycle of SRAM with Byte Selection (CSnWCR.SW[1:0]=B'01, CSnWCR.HW[1:0]=B'01, External Wait 1 Input, BAS = 0 (UB and LB in Write Cycle Controlled))



Note: * Waveform when active low is specified for DACKn.

Figure 33.14 Bus Cycle of SRAM with Byte Selection (CSnWCR.SW[1:0]=B'01, CSnWCR.HW[1:0]=B'01, External Wait 1 Input, BAS = 1 (WE in Write Cycle Controlled))

33.4.5 Burst ROM Timing

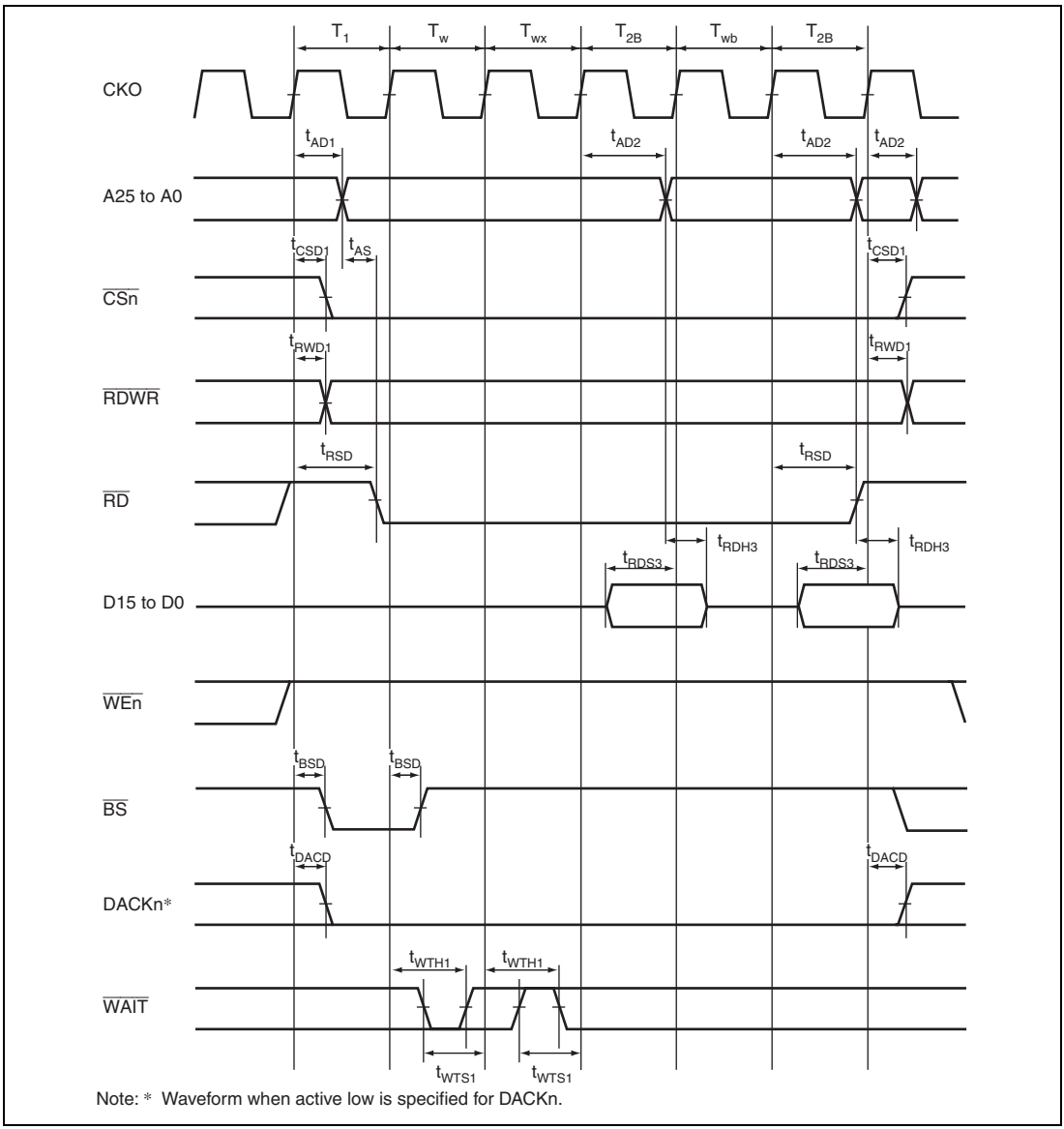
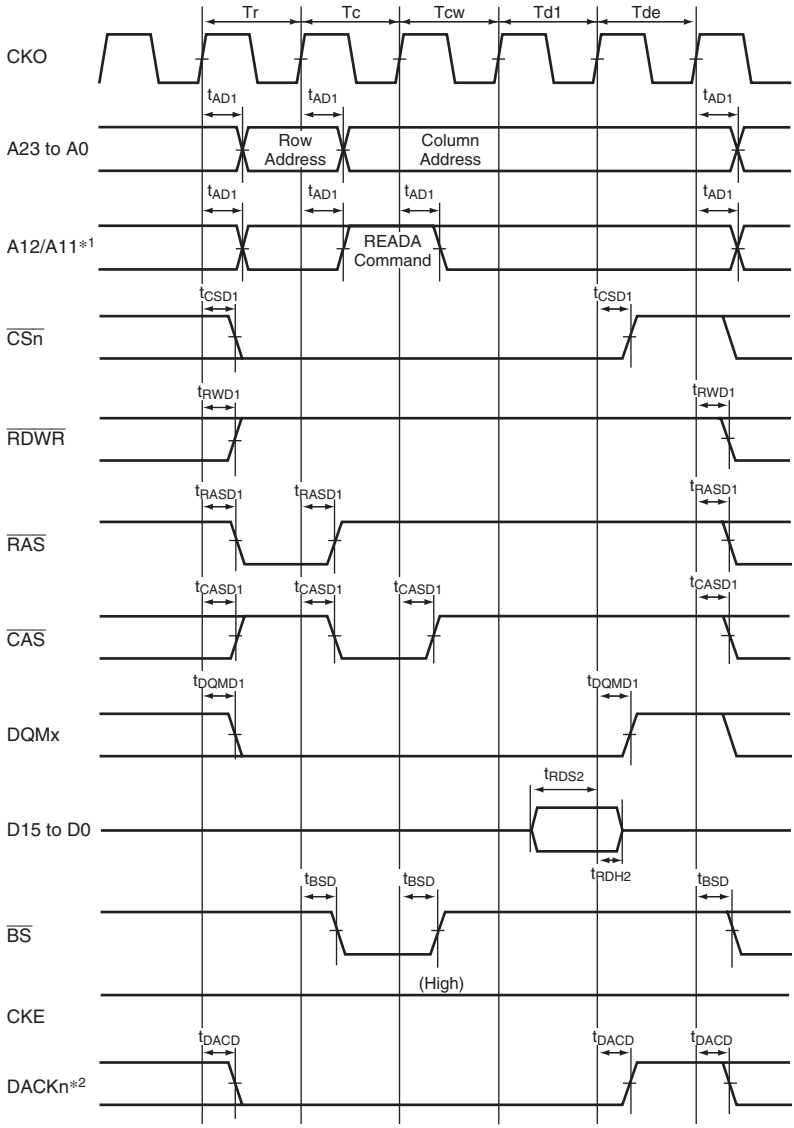


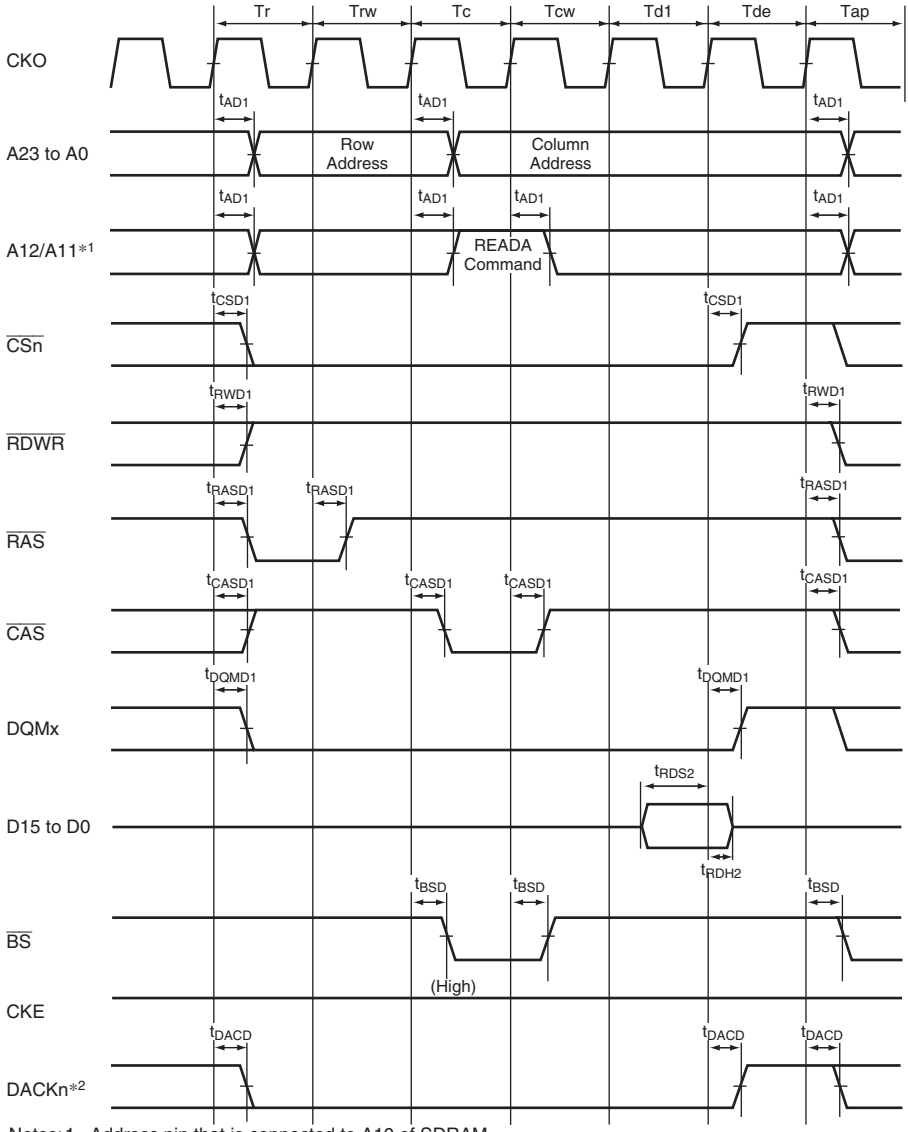
Figure 33.15 Read Bus Cycle of Burst ROM
(Software Wait 1, Asynchronous External Wait 1 Input, Burst Wait 1, Number of Burst = 2)

33.4.6 SDRAM Timing



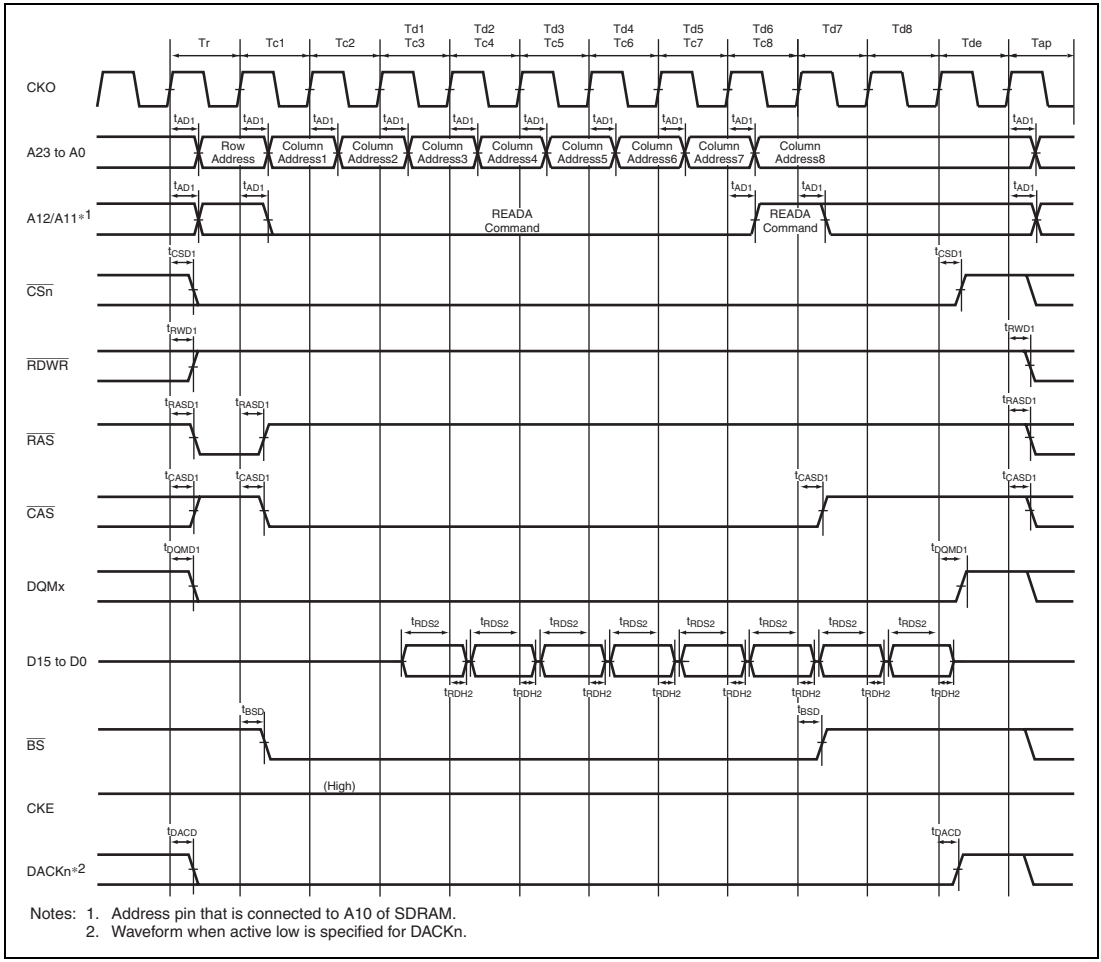
Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.16 Single Read Bus Cycle of SDRAM (Auto Precharge Mode, CAS Latency 2, TRCD = 1 Cycle, TRP = 1 Cycle)



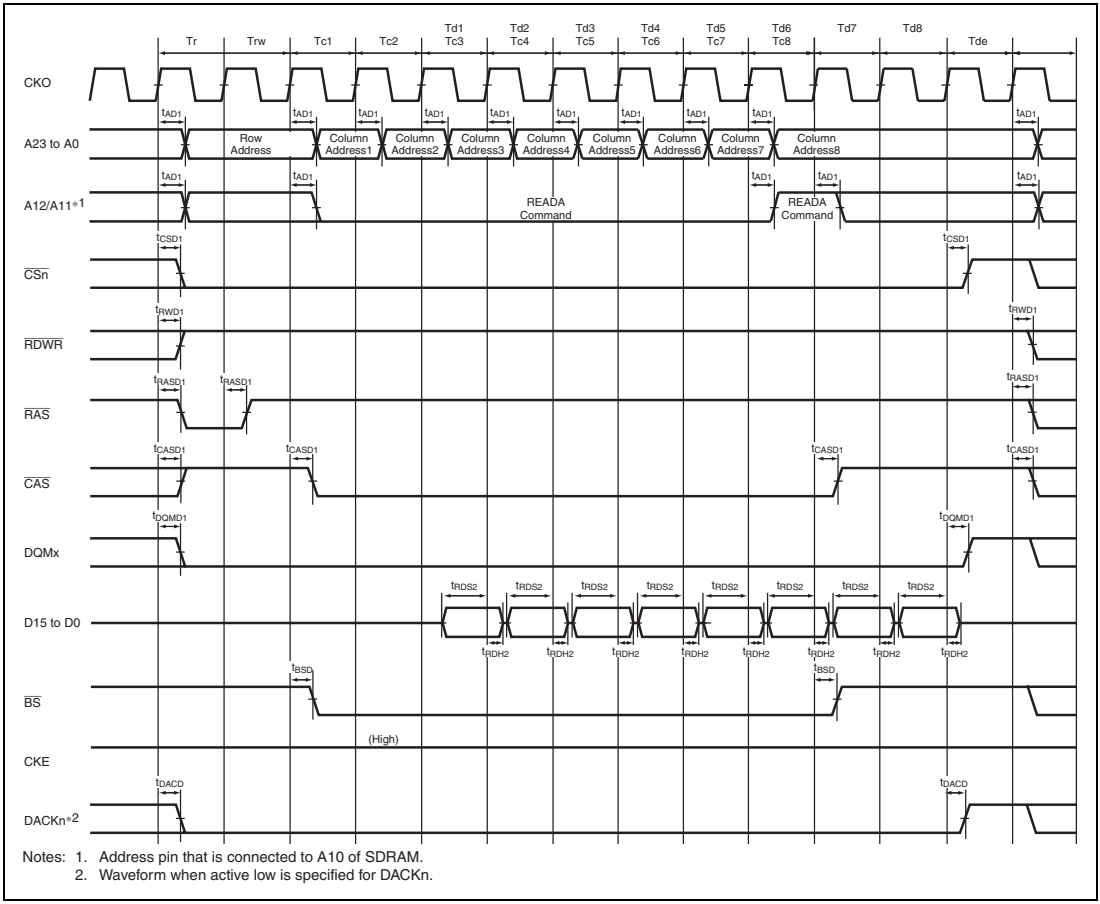
Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.17 Single Read Bus Cycle of SDRAM (Auto Precharge Mode, CAS Latency 2, TRCD = 2 Cycles, TRP = 2 Cycles)



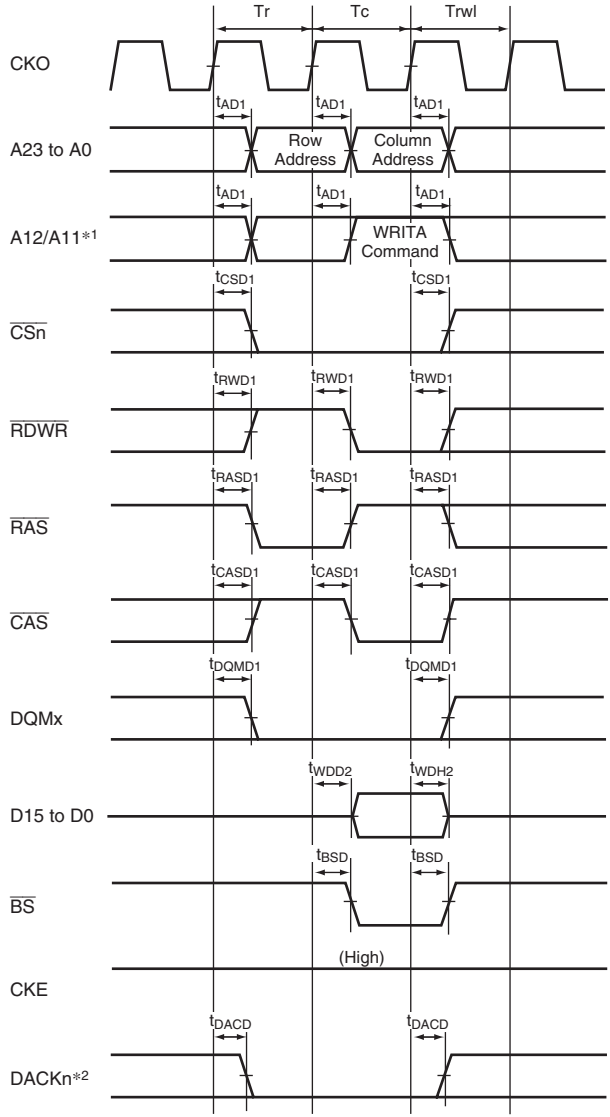
Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.18 Burst Read Bus Cycle of SDRAM (Single Read × 8)
(Auto Precharge Mode, CAS Latency 2, TRCD = 1 Cycle, TRP = 2 Cycles)



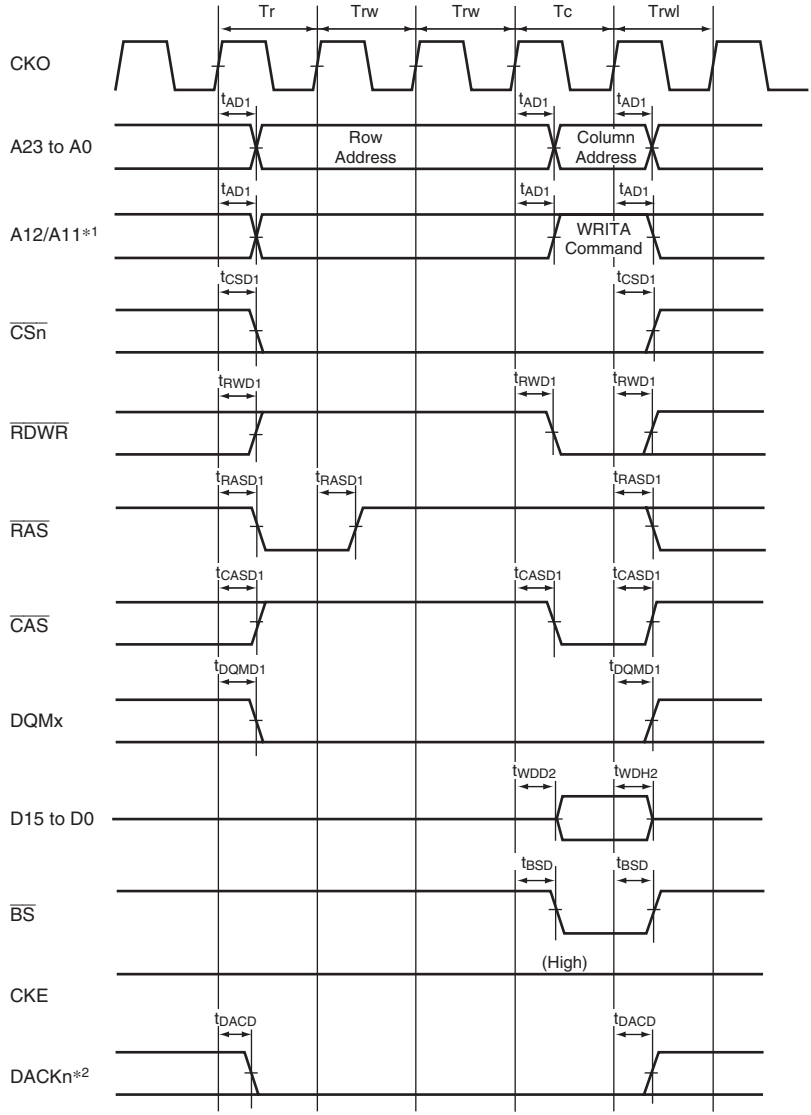
Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.19 Burst Read Bus Cycle of SDRAM (Single Read × 8)
(Auto Precharge Mode, CAS Latency 2, TRCD = 2 Cycles, TRP = 1 Cycle)



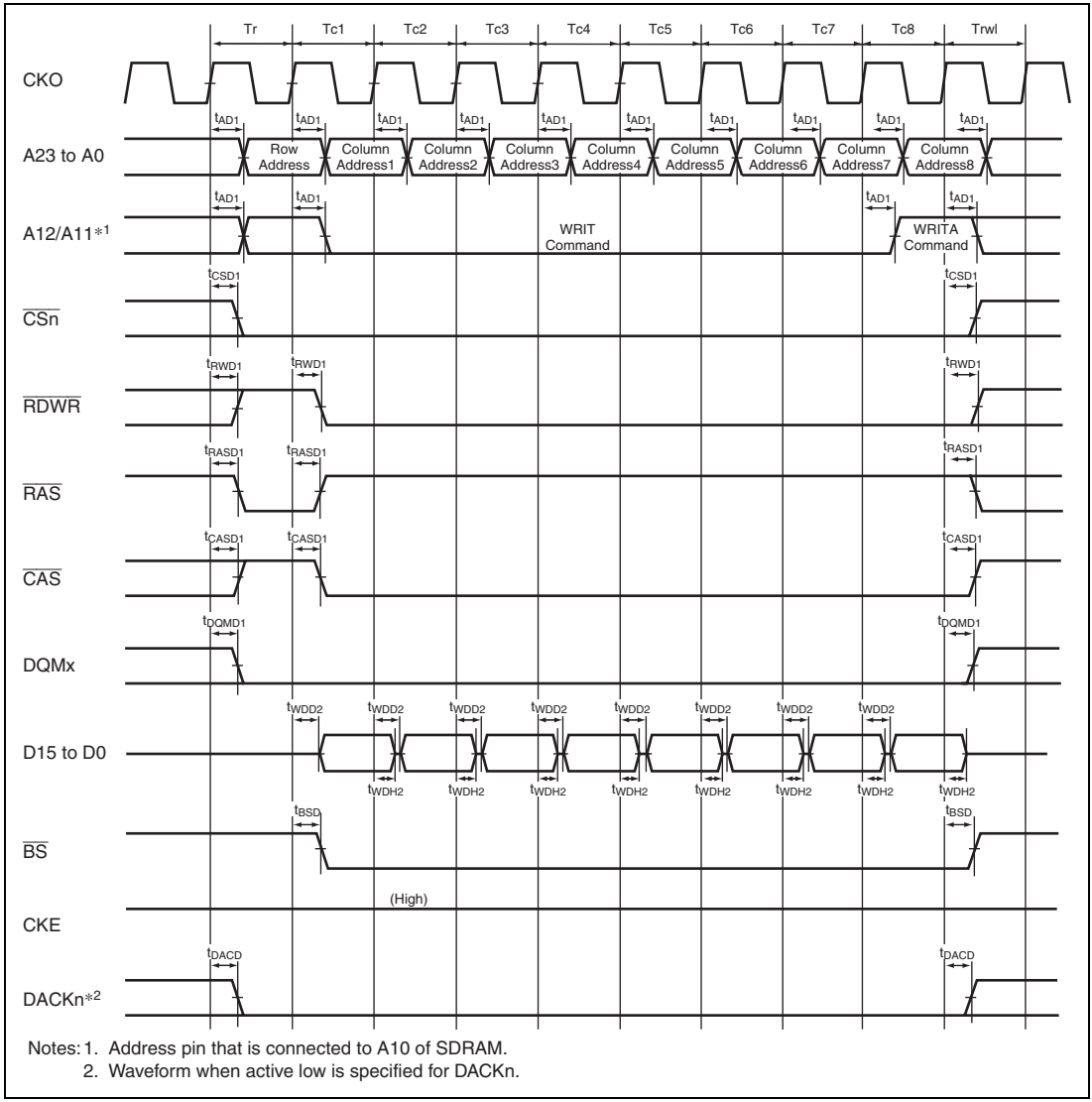
Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.20 Single Write Bus Cycle of SDRAM (Auto Precharge Mode, TRWL = 1 Cycle)



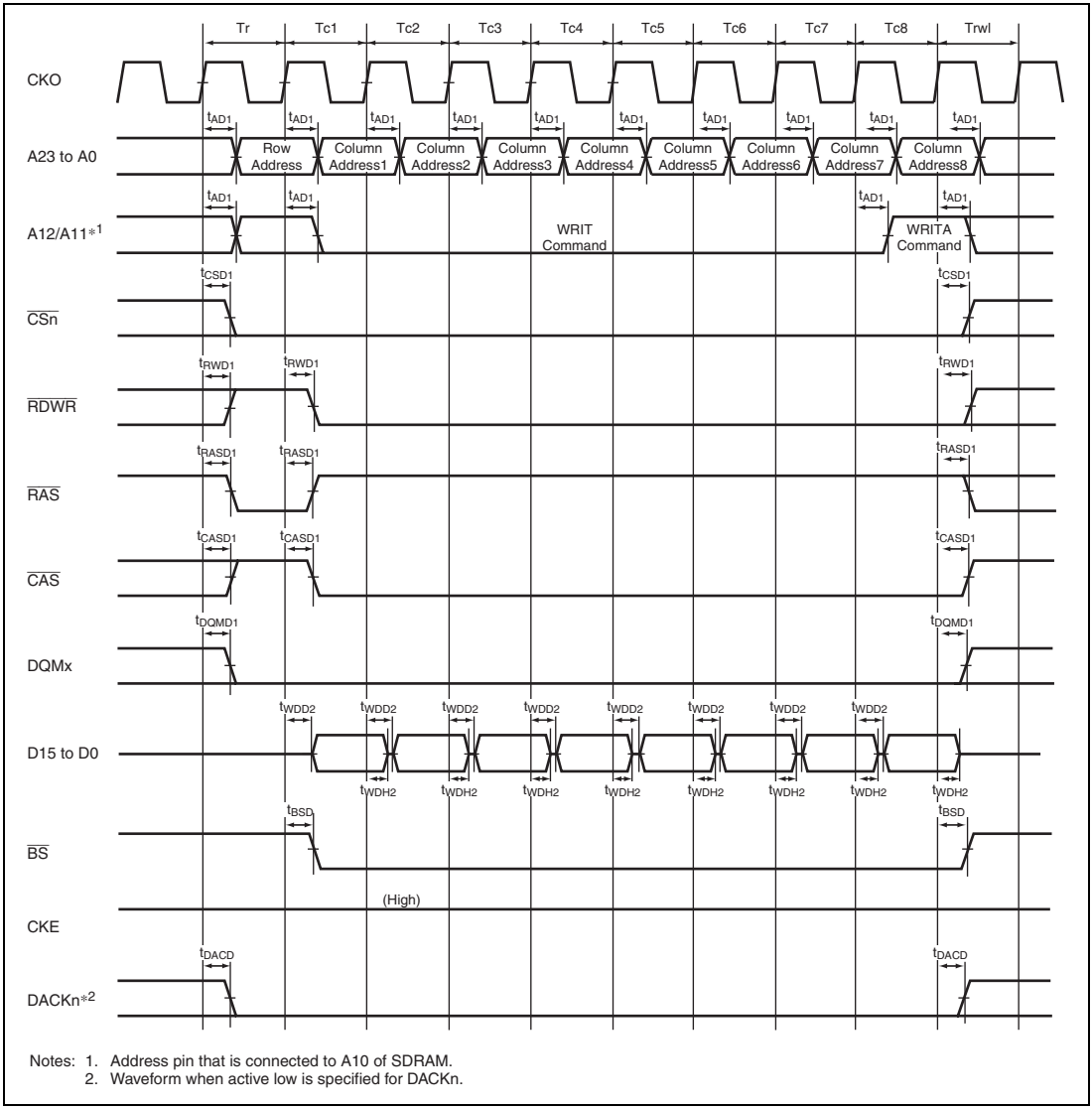
Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.21 Single Write Bus Cycle of SDRAM (Auto Precharge Mode, TRCD = 3 Cycles, TRWL = 1 Cycle)



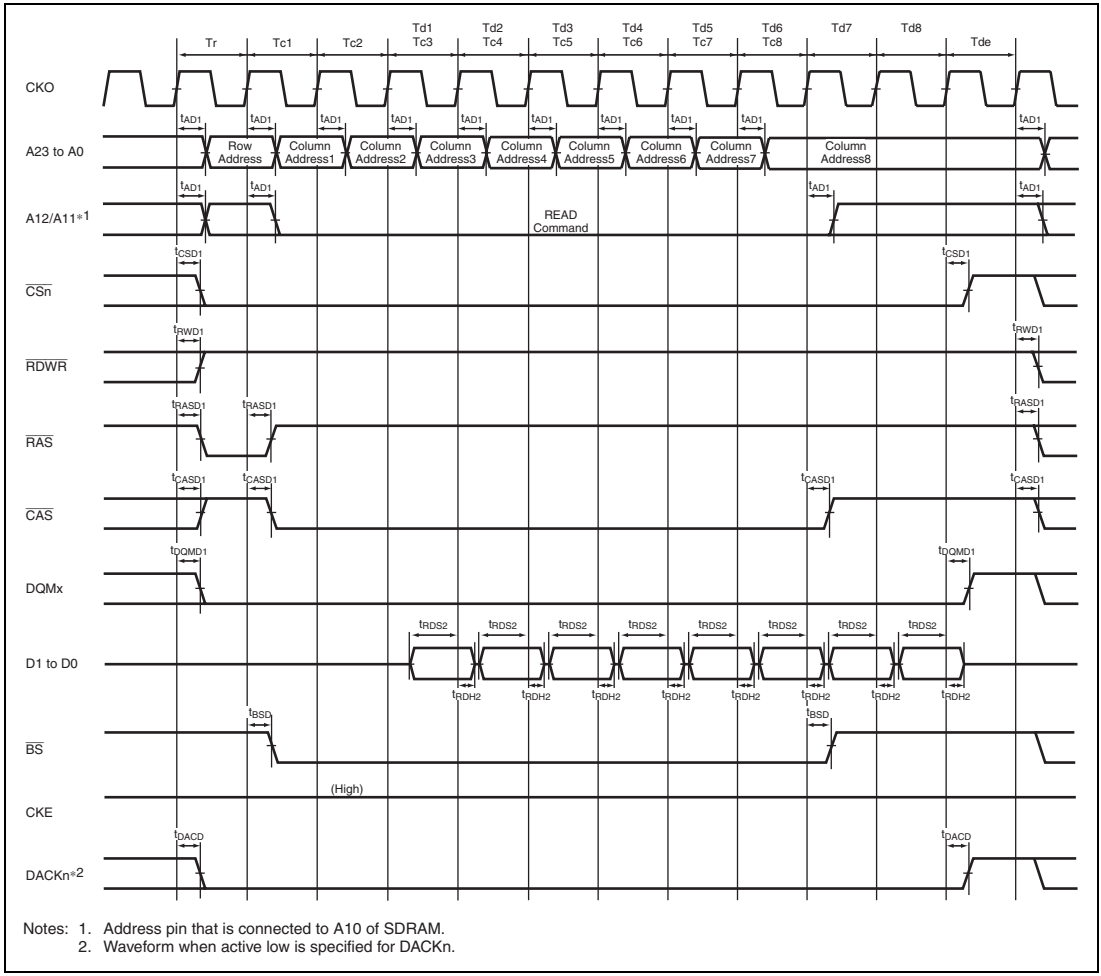
Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.22 Burst Write Bus Cycle of SDRAM (Single Write × 8) (Auto Precharge Mode, TRCD = 1 Cycle, TRWL = 1 Cycle)



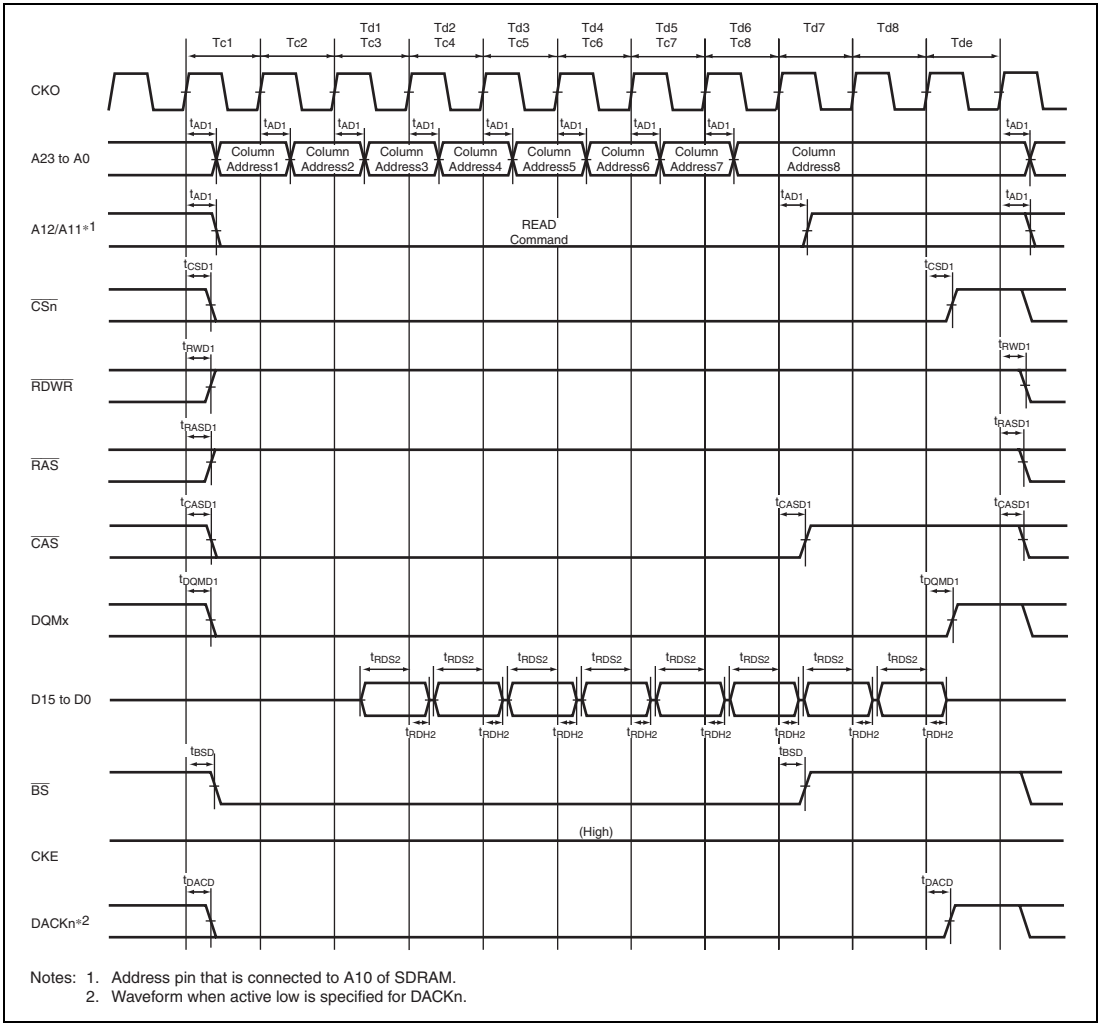
Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.23 Burst Write Bus Cycle of SDRAM (Single Write × 8)
(Auto Precharge Mode, TRCD = 2 Cycles, TRWL = 1 Cycle)



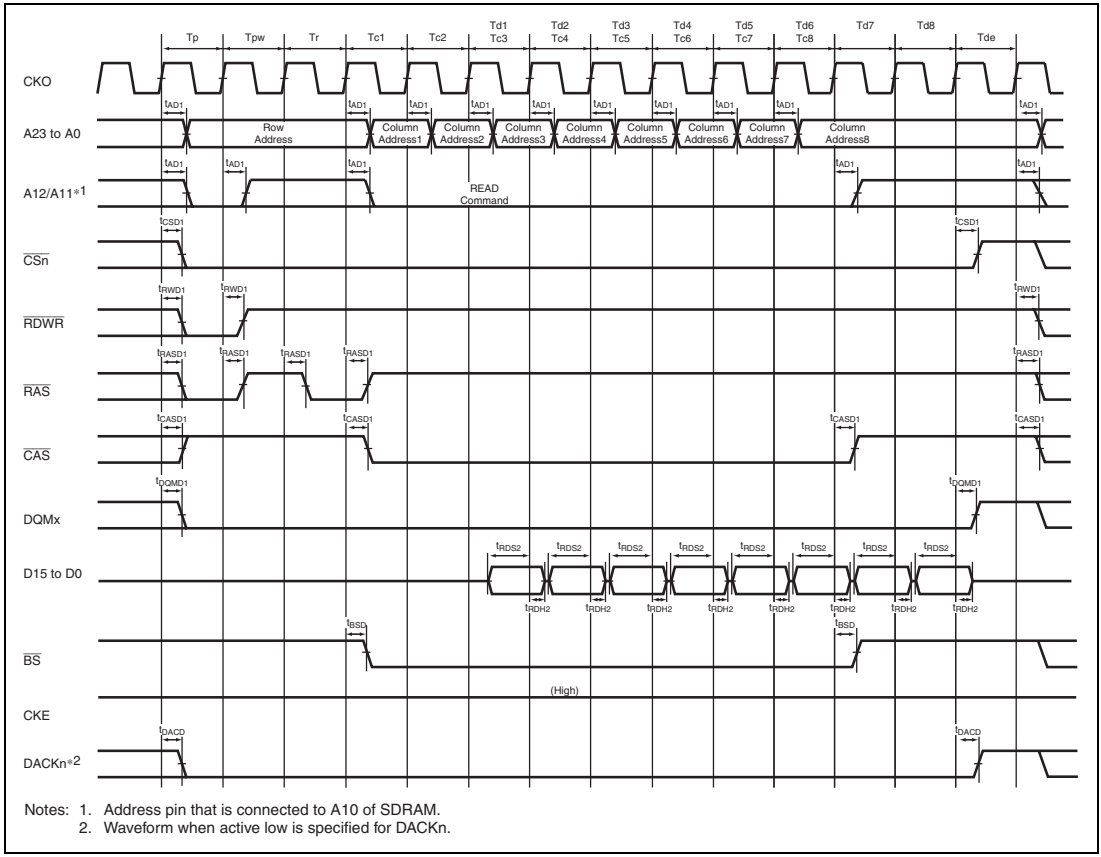
Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.24 Burst Read Bus Cycle of SDRAM (Single Read × 8)
(Bank Active Mode: ACTV + READ Command, CAS Latency 2, TRCD = 1 Cycle)



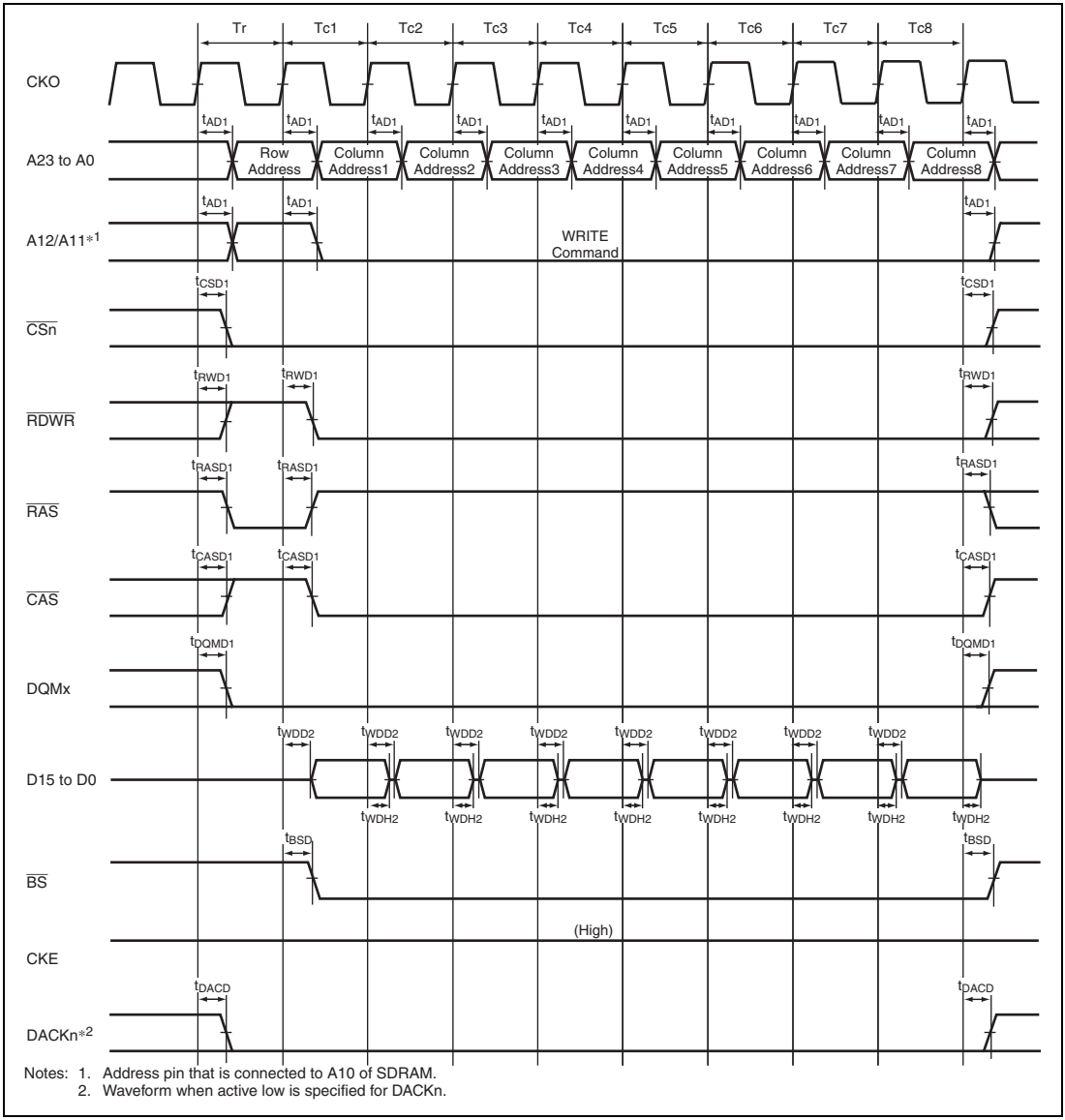
Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.25 Burst Read Bus Cycle of SDRAM (Single Read × 8)
(Bank Active Mode: READ Command,
Same Row Address, CAS Latency 2, TRCD = 1 Cycle)



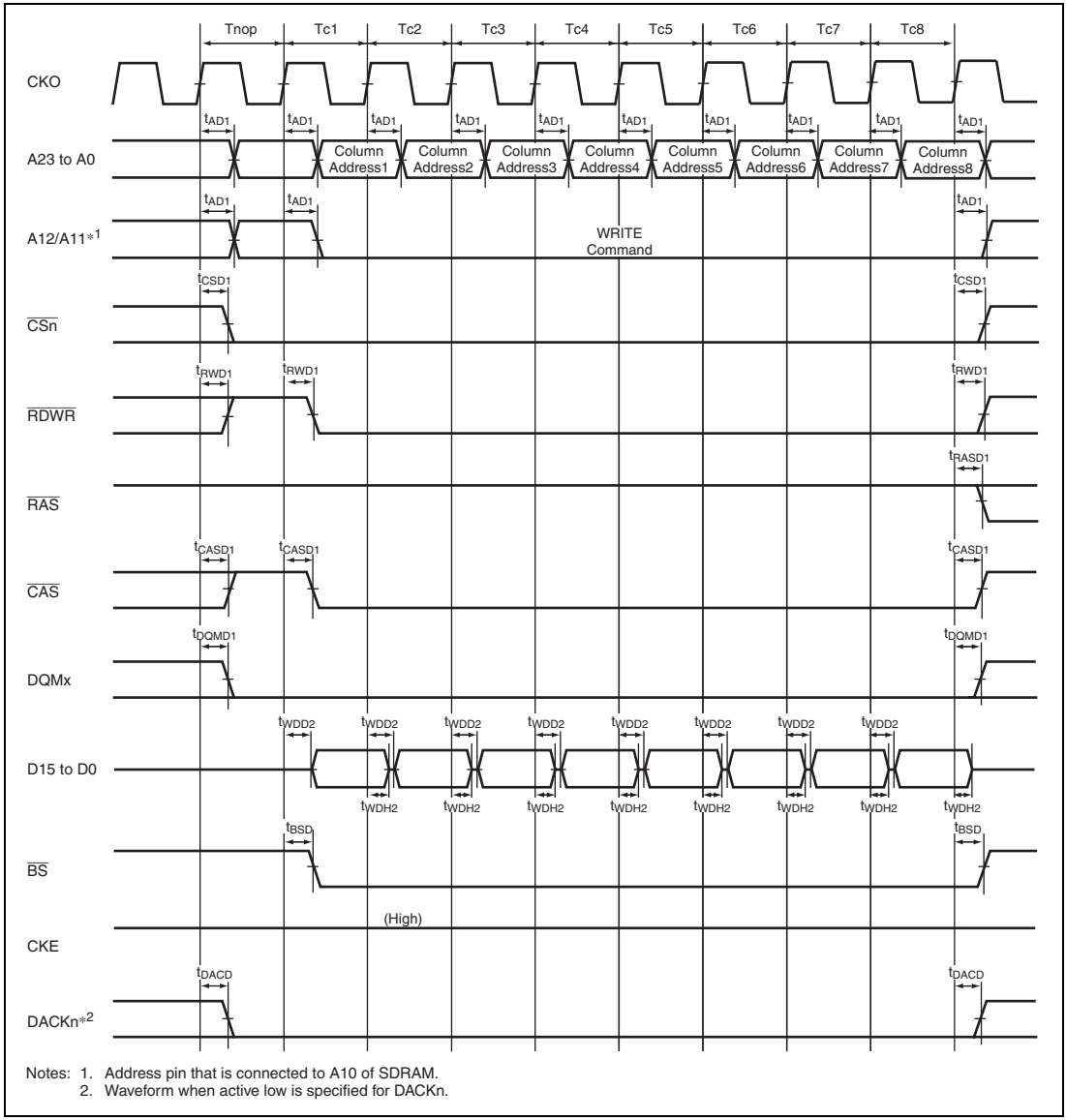
Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.26 Burst Read Bus Cycle of SDRAM (Single Read × 8)
(Bank Active Mode: PRE + ACTV + READ Command,
Different Row Address, CAS Latency 2, TRCD = 1 Cycle)



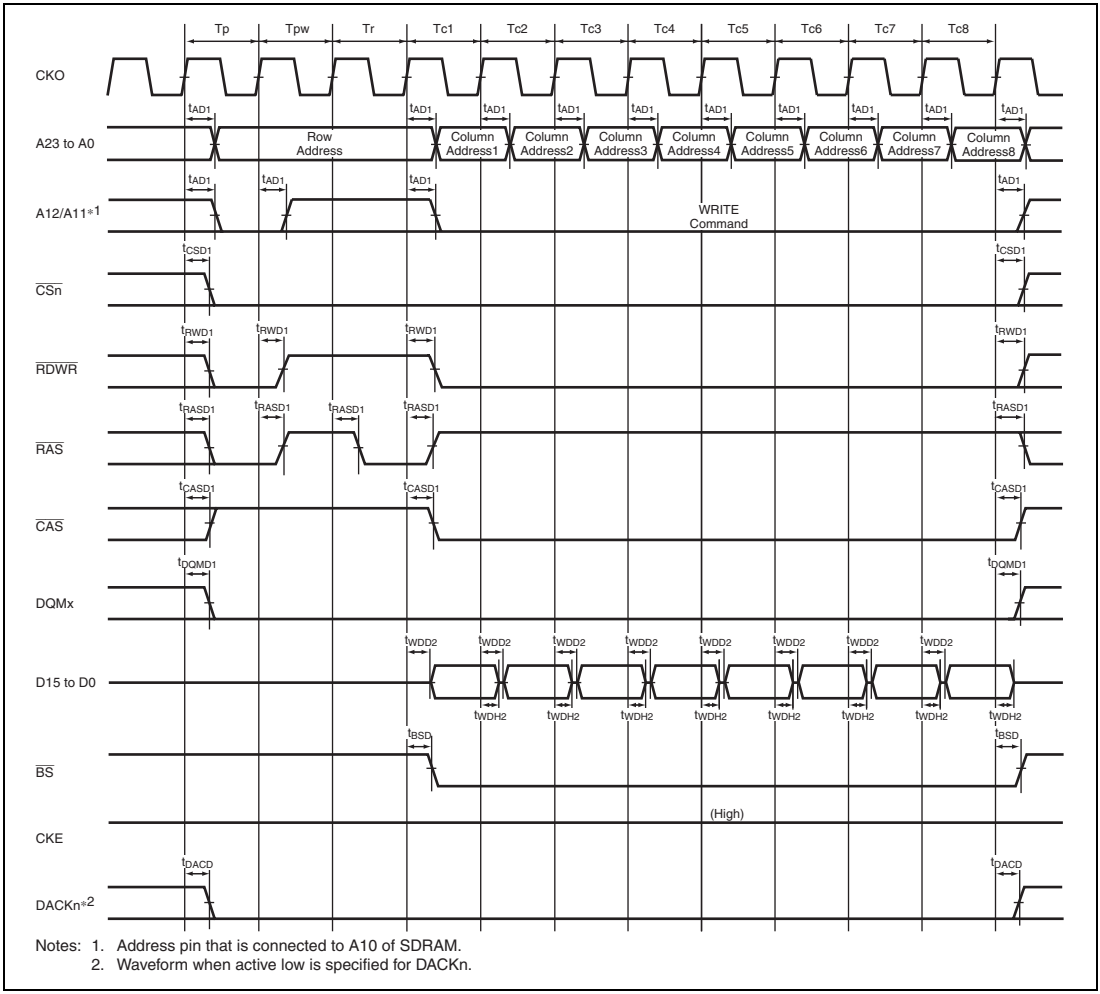
Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.27 Burst Write Bus Cycle of SDRAM (Single Write × 8)
(Bank Active Mode: ACTV + WRIT Command, TRCD = 1 Cycle)



Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.28 Burst Write Bus Cycle of SDRAM (Single Write × 8)
(Bank Active Mode: ACTV + WRIT Command, TRCD = 1 Cycle)



Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.29 Burst Write Bus Cycle of SDRAM (Single Write × 8)
(Bank Active Mode: PRE + ACTV + WRIT Command, TRCD = 1 Cycle)

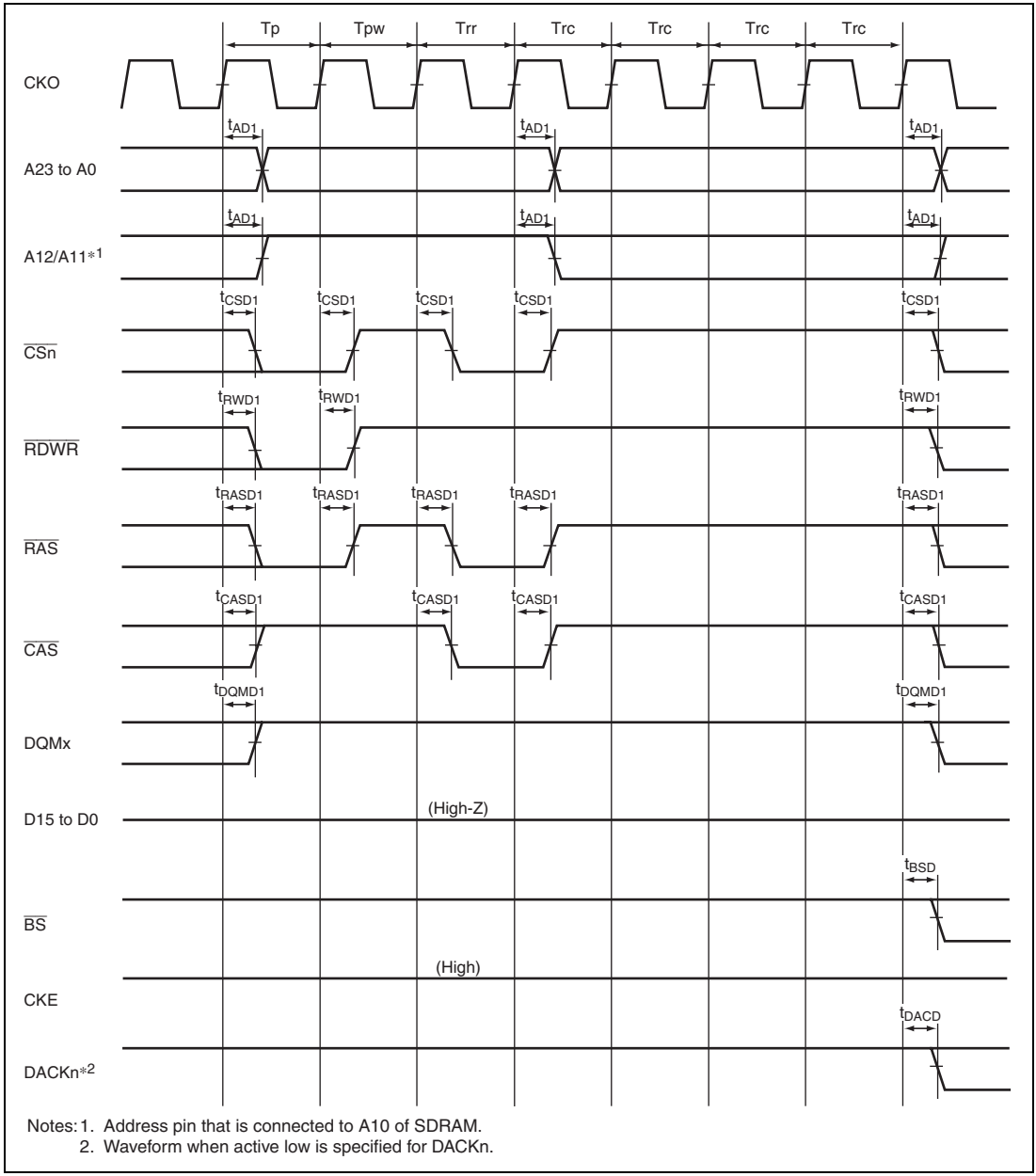
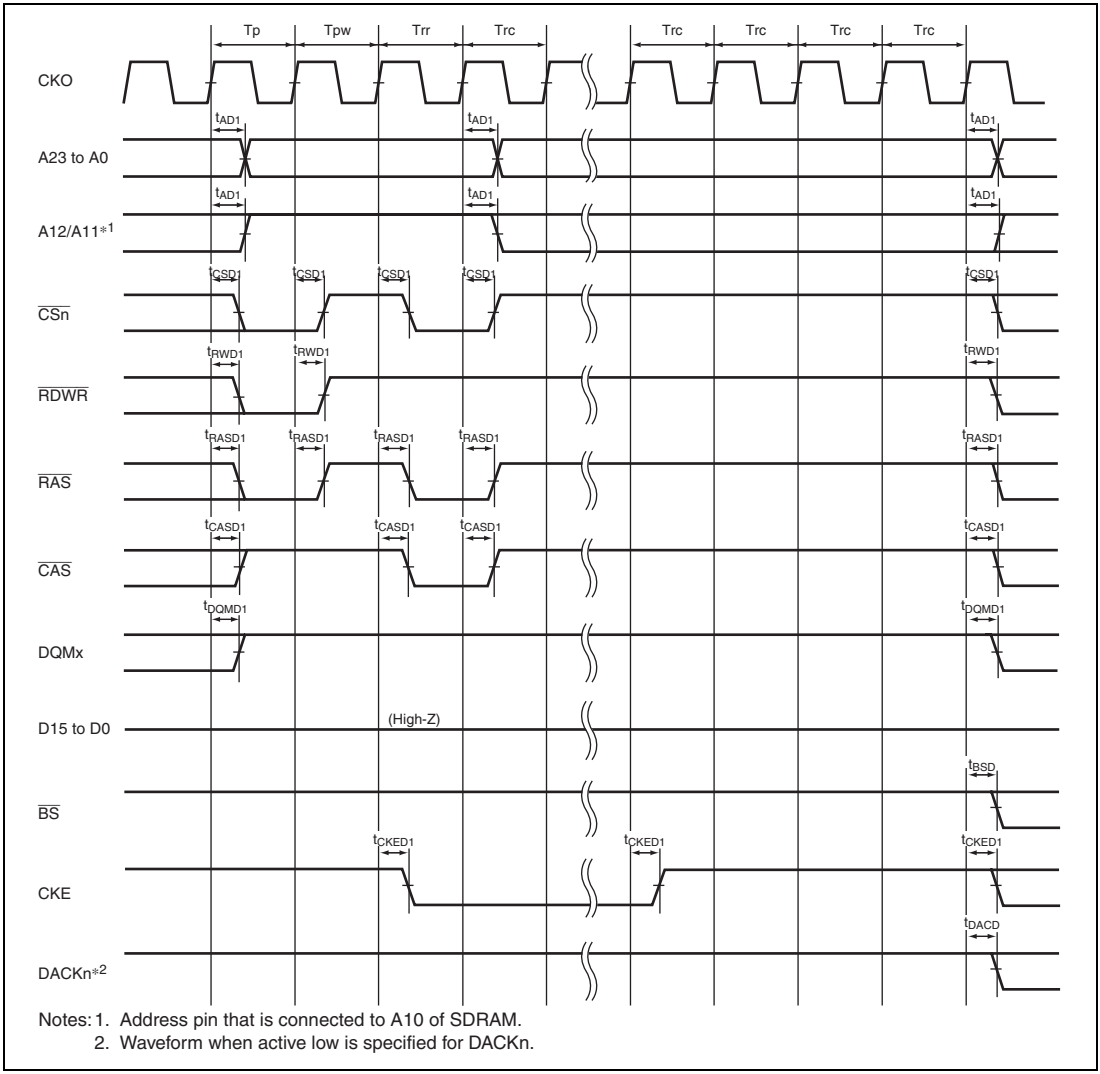
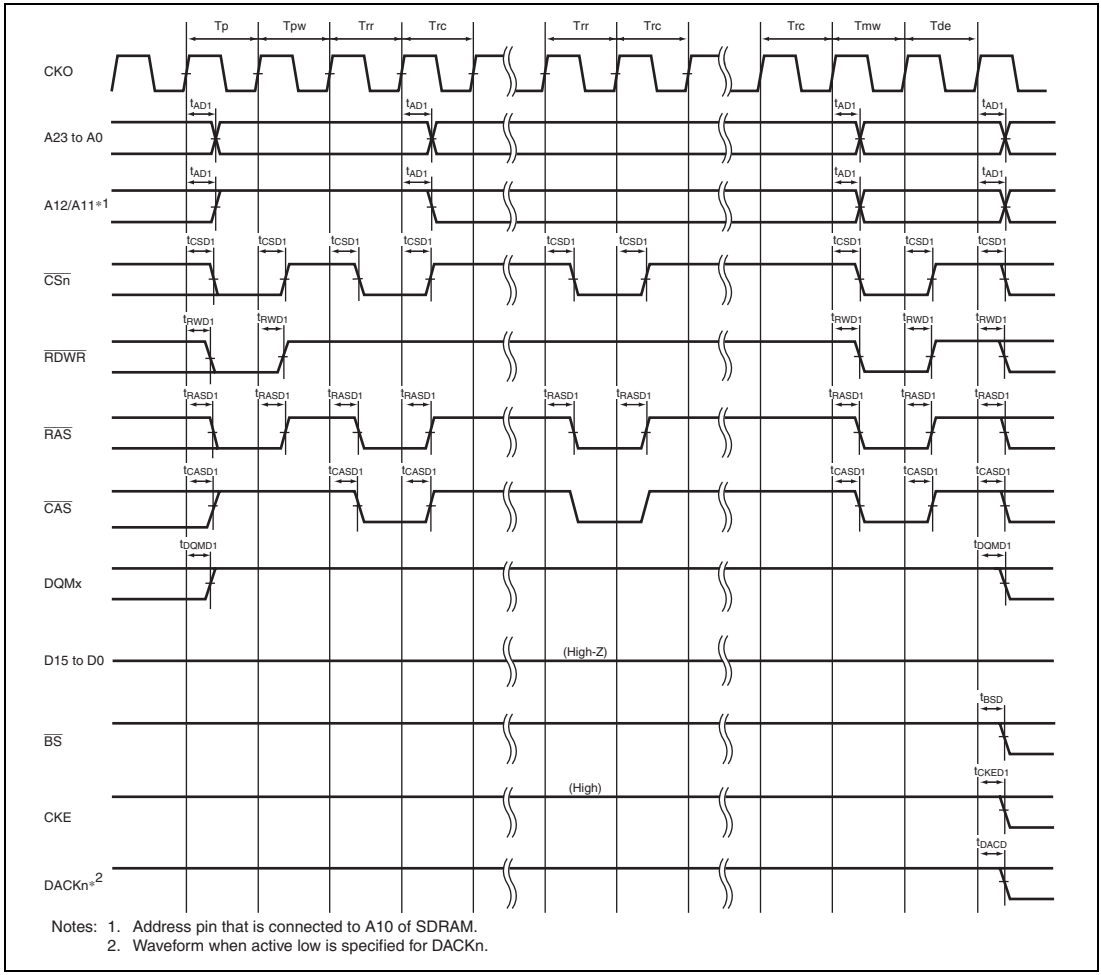


Figure 33.30 Auto Refresh Timing of SDRAM (TRP = 2 Cycles)



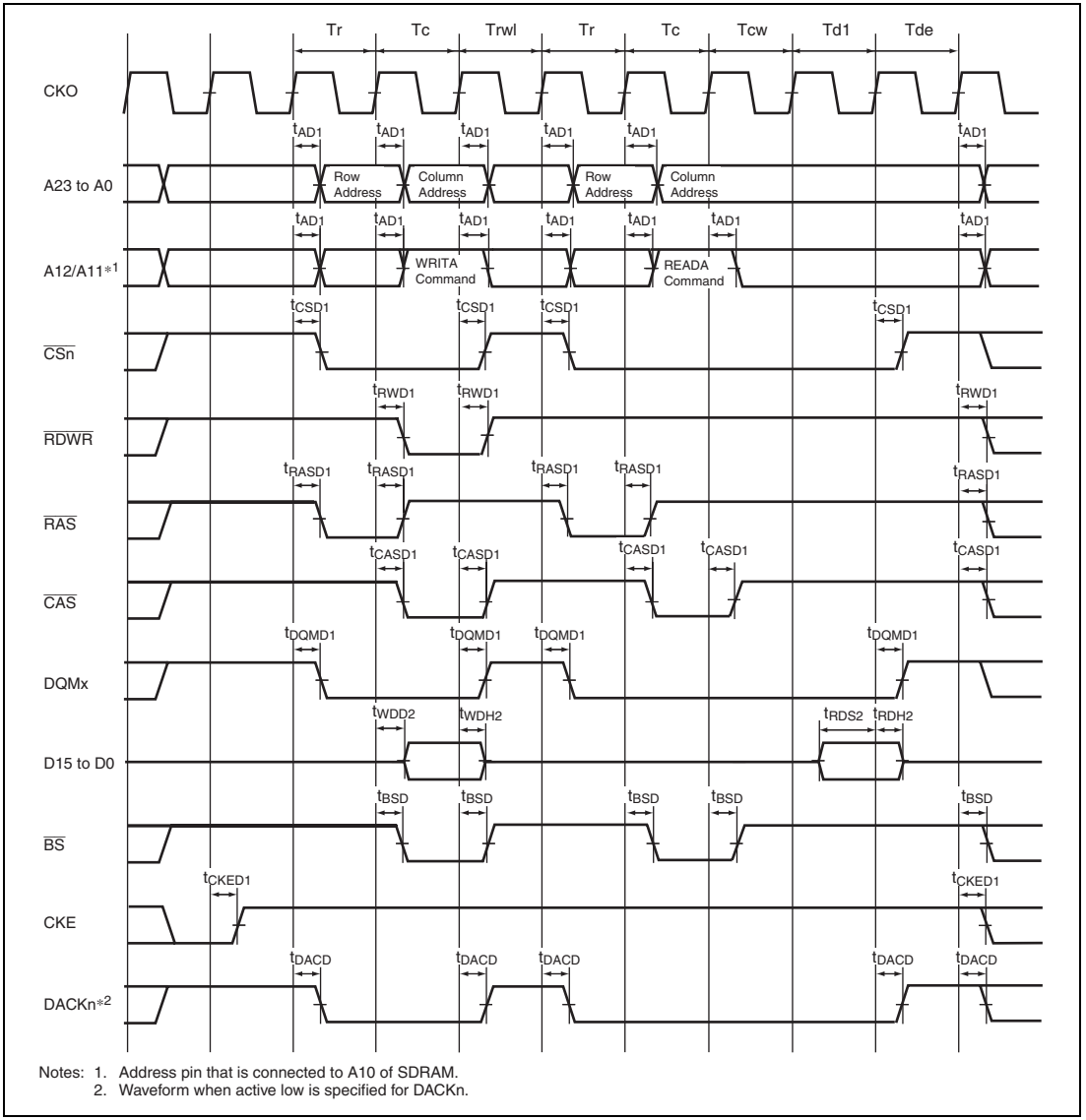
Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.31 Self Refresh Timing of SDRAM (TRP = 2 Cycles)



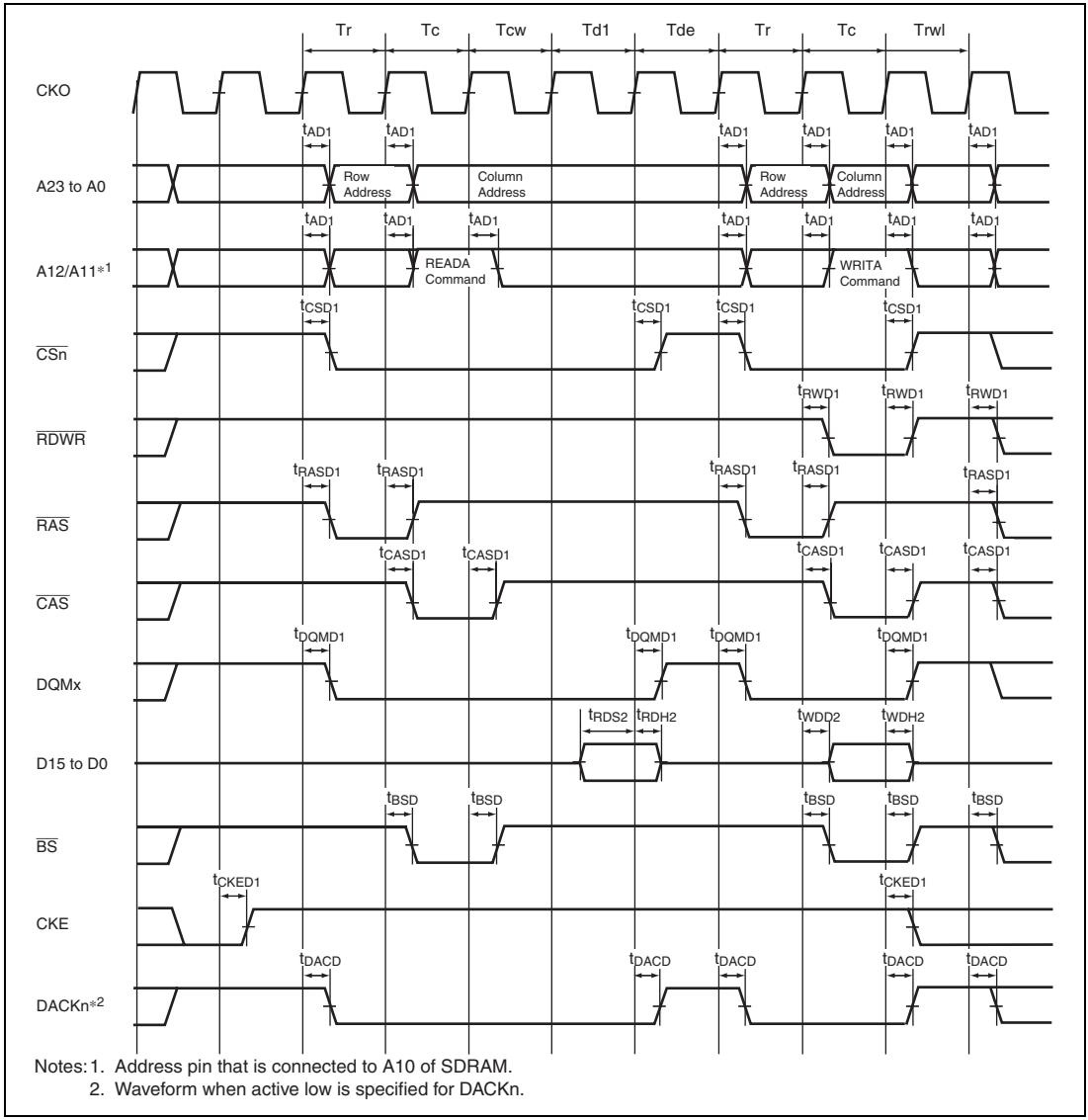
Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.32 Power-On Sequence of SDRAM (Mode Write Timing, TRP = 2 Cycles)



Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.33 Write-to-Read Bus Cycle in Power-Down Mode of SDRAM (Auto Precharge Mode, TRCD = 1 Cycle, TRP = 1 Cycle, TRWL = 1 Cycle)



Notes: 1. Address pin that is connected to A10 of SDRAM.
2. Waveform when active low is specified for DACKn.

Figure 33.34 Read-to-Write Bus Cycle in Power-Down Mode of SDRAM (Auto Precharge Mode, TRCD = 1 Cycle, TRP = 1 Cycle, TRWL = 1 Cycle)

33.4.7 PCMCIA Timing

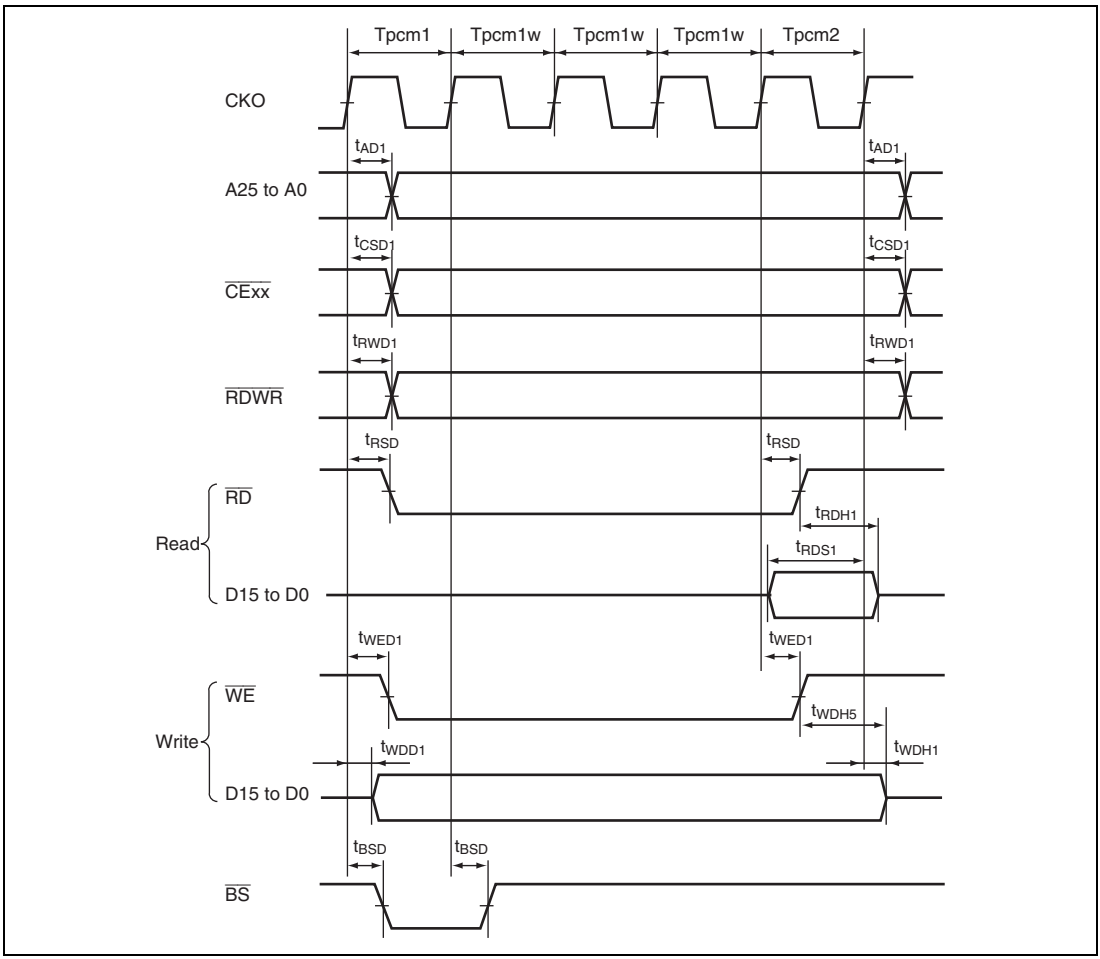


Figure 33.35 PCMCIA Memory Card Interface Bus Timing

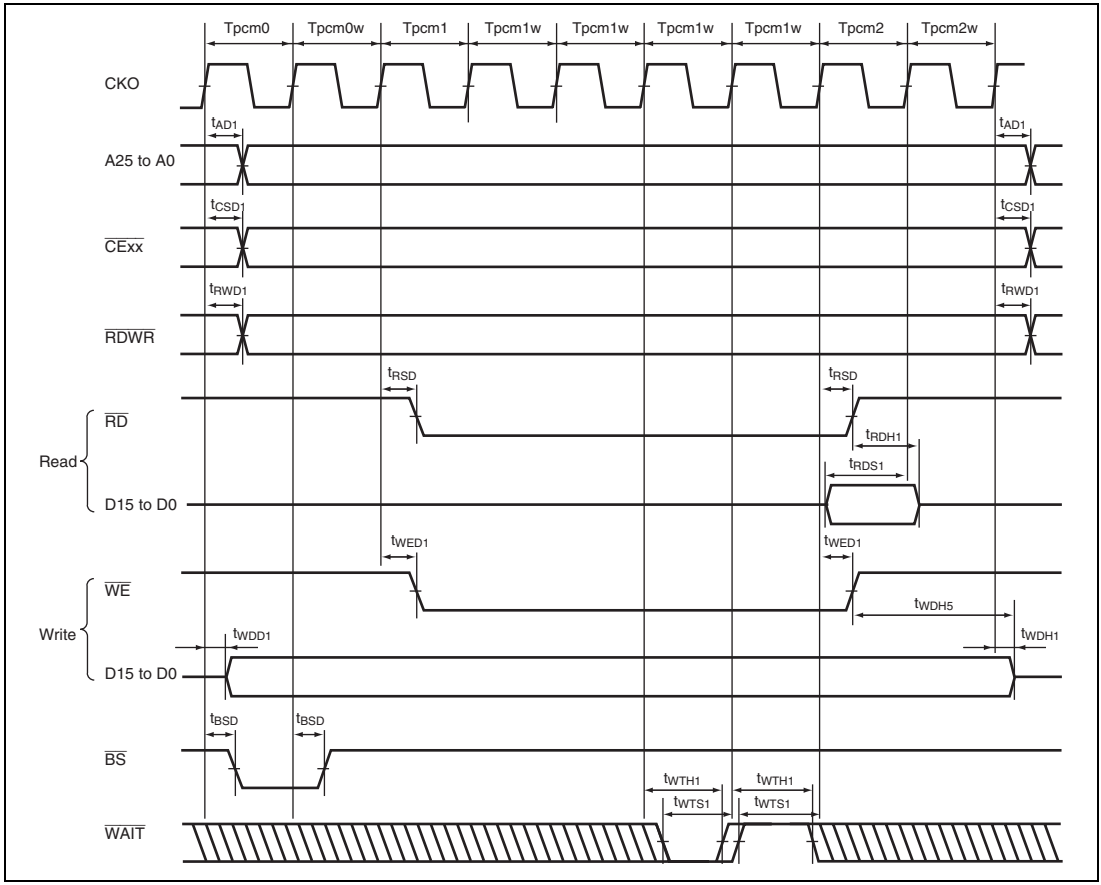


Figure 33.36 PCMCIA Memory Card Interface Bus Timing (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait 1, Hardware Wait 1)

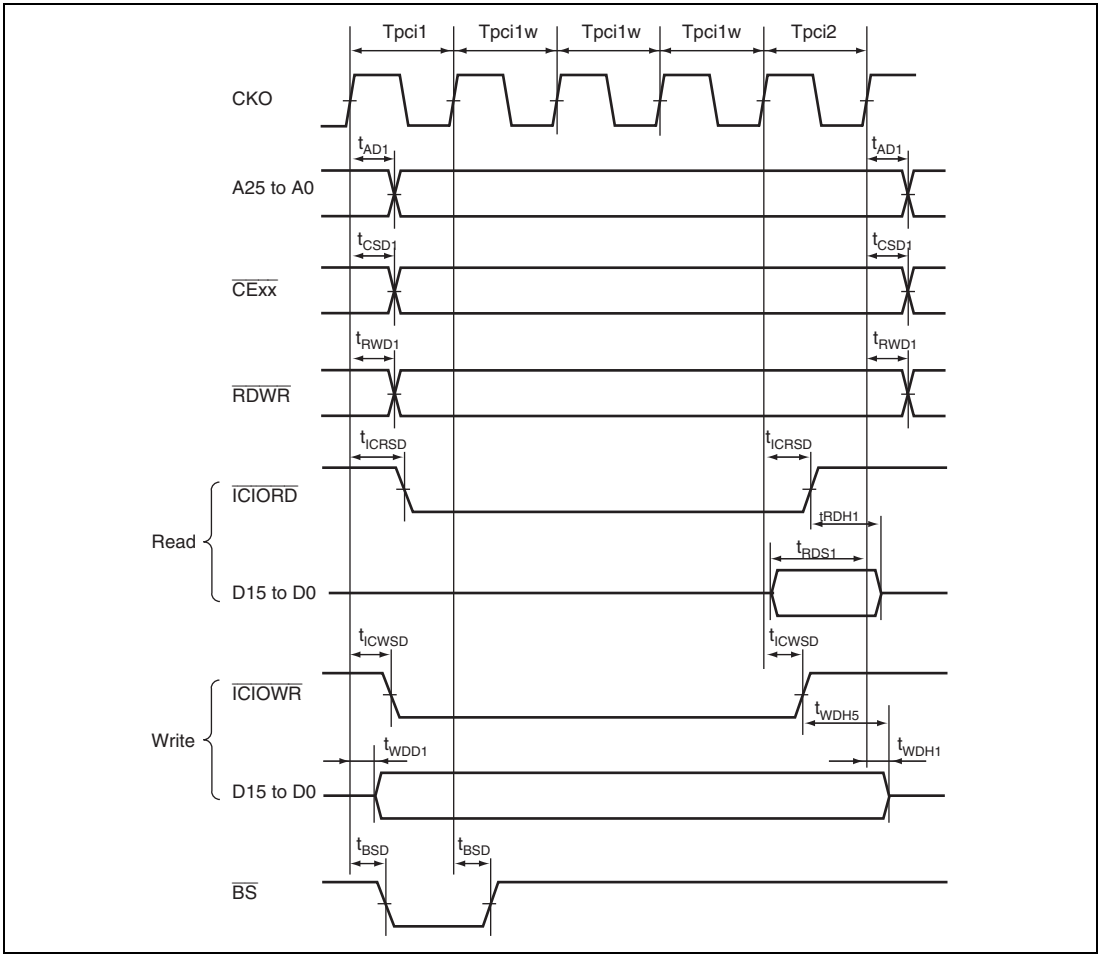


Figure 33.37 PCMCIA I/O Card Interface Bus Timing

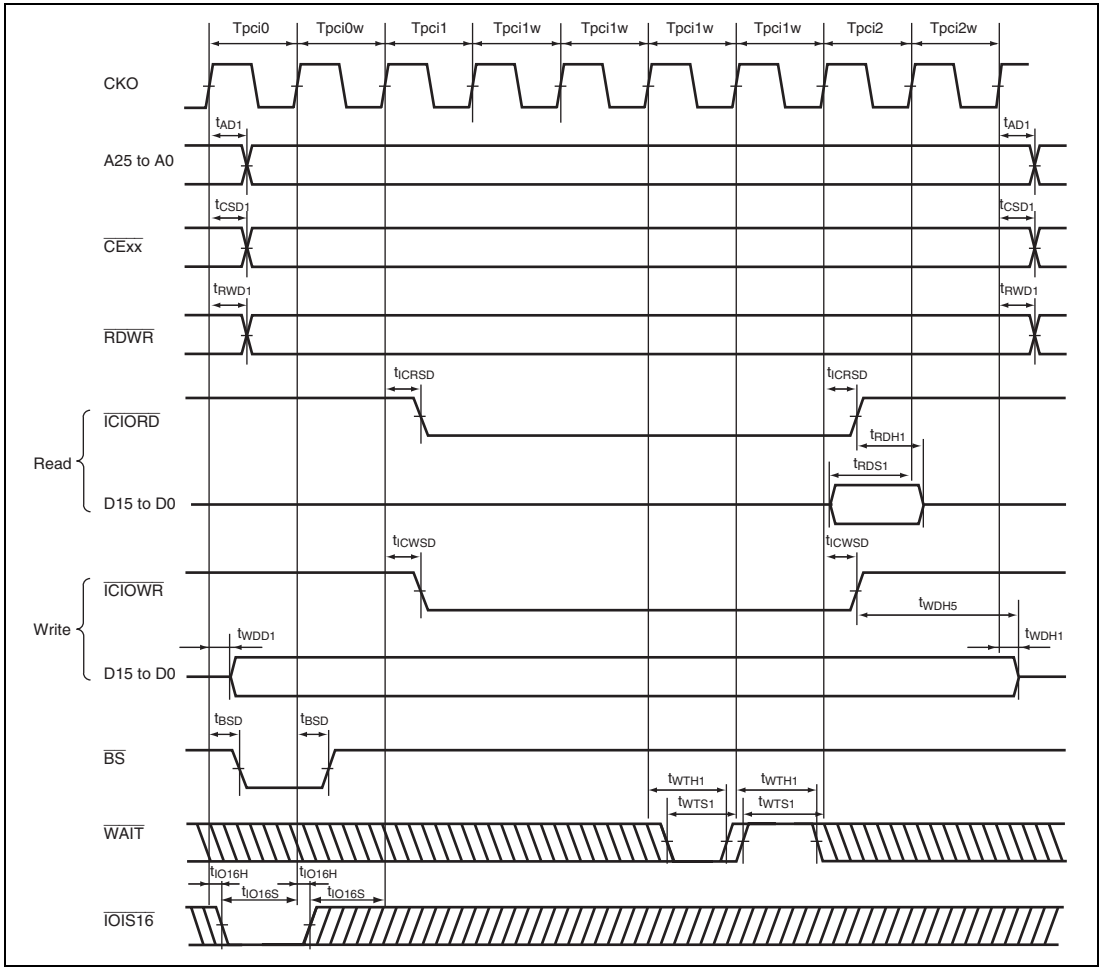


Figure 33.38 PCMCIA I/O Card Interface Bus Timing (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait 1, Hardware Wait 1)

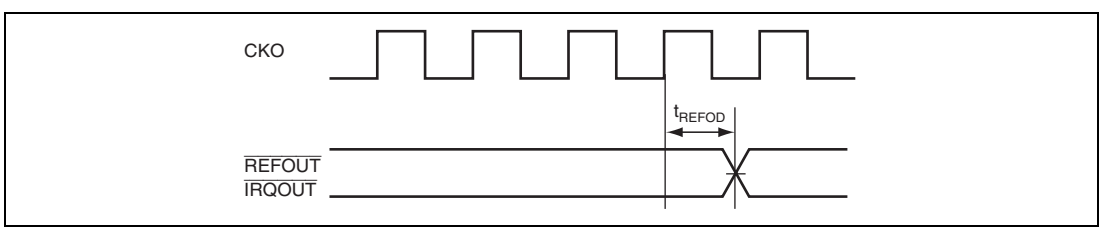


Figure 33.39 $\overline{\text{REFOUT}}$, $\overline{\text{IRQOUT}}$ Delay Time

33.4.8 Peripheral Module Signal Timing

Table 33.10 Peripheral Module Signal Timing

Module	Item	Symbol	Min.	Max.	Unit	Figure
Port	Output data delay time	t_{PORTD}	—	17	ns	33.40
	Input data setup time	t_{PORTS}	15	—		
	Input data hold time	t_{PORTH}	8	—		
DMAC	DREQn setup time	t_{DREQS}	6	—	ns	33.41
	DREQn hold time	t_{DREQH}	4	—		
	DACKn, TENDn delay time	t_{DACD}	—	13		33.42

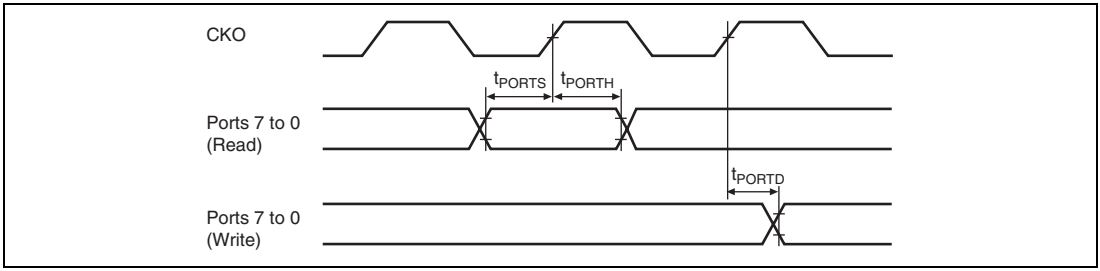


Figure 33.40 I/O Port Timing

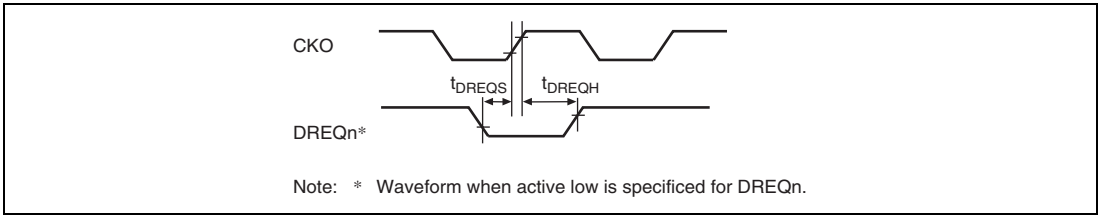


Figure 33.41 DREQ Input Timing (DREQ Low Level is Detected)

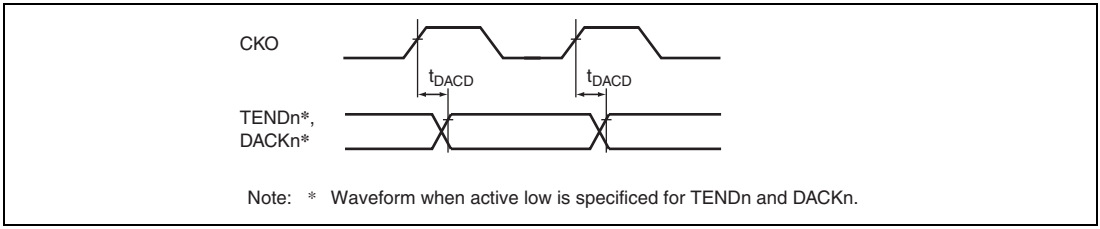


Figure 33.42 TEND, DACK Output Timing

33.4.9 16-Bit Timer Pulse Unit (TPU)

Table 33.11 16-Bit Timer Pulse Unit

Item	Symbol	Min.	Max.	Unit	Figure
Timer output delay time	t_{TOD}	—	15	ns	33.43

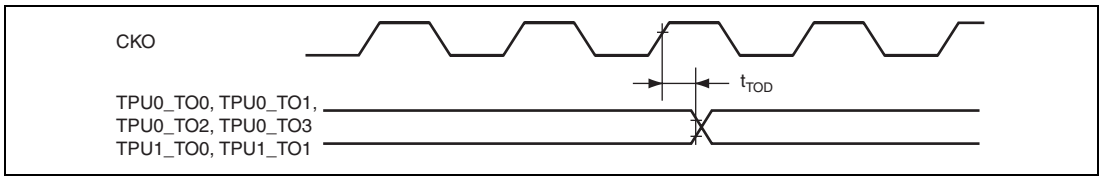


Figure 33.43 TPU Output Timing

33.4.10 RTC Signal Timing

Table 33.12 RTC Signal Timing

Module	Item	Symbol	Min.	Max.	Unit	Figure
RTC	Oscillation settling time	t_{ROSC}	3	—	s	33.44

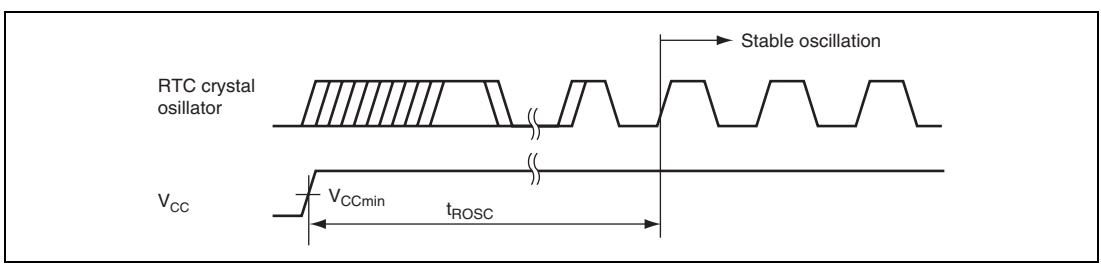


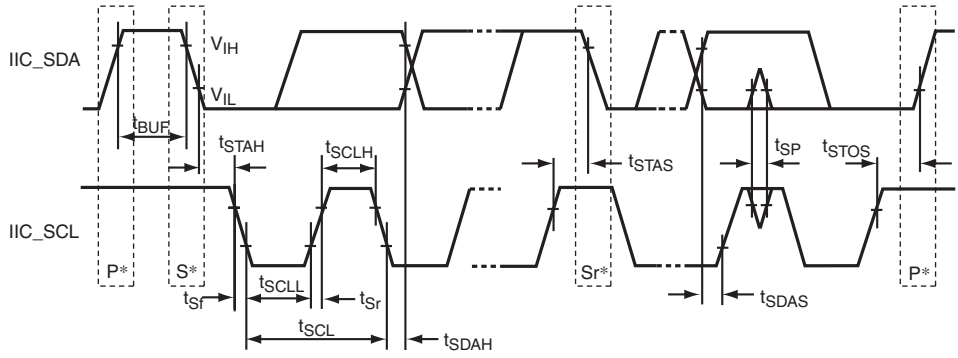
Figure 33.44 Oscillation Settling Time when RTC Crystal Oscillator is Turned On

33.4.11 I²C Bus Interface Timing

Table 33.13 I²C Bus Interface Timing

Item	Symbol	Test Conditions	Value			Unit	Figure
			Min.	Typ.	Max.		
SCL input cycle time	t_{SCL}		$12t_{pcyc}$ + 600	—	—	ns	33.45
SCL input high pulse width	t_{SCLH}		$3t_{pcyc}$ + 300	—	—	ns	
SCL input low pulse width	t_{SCLL}		$5t_{pcyc}$ + 300	—	—	ns	
SCL, SDA input rise time	t_{Sr}		—	—	300	ns	
SCL, SDA input fall time	t_{Sf}		—	—	300	ns	
SCL, SDA input input spike pulse removal time	t_{SP}		—	—	$1t_{cyc}$	ns	
SDA input bus free time	t_{BUF}		$5t_{pcyc}$	—	—	ns	
Start condition input hold time	t_{STAH}		$3t_{pcyc}$	—	—	ns	
Repeated start condition input setup time	t_{STAS}		$3t_{pcyc}$	—	—	ns	
Stop condition input setup time	t_{STOS}		$3t_{pcyc}$	—	—	ns	
Data input setup time	t_{SDAS}		$1t_{pcyc}$ + 20	—	—	ns	
Data input hold time	t_{SDAH}		0	—	—	ns	
SCL, SDA capacitive load	Cb		0	—	400	pF	
SCL, SDA output fall time	t_{Sf}	VccQ = 3.0 V	—	—	250	ns	
		VccQ = 3.6 V	—	—	300	ns	

Note: * t_{pcyc} is the cycle time of the peripheral clock (Pϕ).



* S, P, and Sr indicate as below.
S: Start condition
P: Stop condition
Sr: Repeated start condition 0

Figure 33.45 I²C Bus Interface Input/Output Timing

33.4.12 SIOF Module Signal Timing

Table 33.14 SIOF Module Signal Timing

Item	Symbol	Min.	Max.	Unit	Figure
SIOF_MCK clock input cycle time	t_{MCYC}	t_{pcyc}*	—	ns	33.46
SIOF_MCK input high level width	t_{MWH}	0.4 × t_{MCYC}	—	ns	33.46
SIOF_MCK input low level width	t_{MWL}	0.4 × t_{MCYC}	—	ns	33.46
SIOF_SCK clock cycle time	t_{SICYC}	t_{pcyc}*	—	ns	33.47 to 33.51
SIOF_SCK output high level width	t_{SWHO}	0.4 × t_{SICYC}	—	ns	33.47 to 33.50
SIOF_SCK output low level width	t_{SWLO}	0.4 × t_{SICYC}	—	ns	33.47 to 33.50
SIOF_SYNC output delay time	t_{FSD}	—	20	ns	33.47 to 33.50
SIOF_SCK input high level width	t_{SWHI}	0.4 × t_{SICYC}	—	ns	33.51
SIOF_SCK input low level width	t_{SWLI}	0.4 × t_{SICYC}	—	ns	33.51
SIOF_SYNC input setup time	t_{FSS}	20	—	ns	33.51
SIOF_SYNC input hold time	t_{FSH}	20	—	ns	33.51
SIOF_TXD output delay time	t_{STDD}	—	20	ns	33.47 to 33.51
SIOF_RXD input setup time	t_{SRDS}	20	—	ns	33.47 to 33.51
SIOF_RXD input hold time	t_{SRDH}	20	—	ns	33.47 to 33.51

Note: * t_{Pcyc} is the cycle time of the peripheral clock (Pϕ).

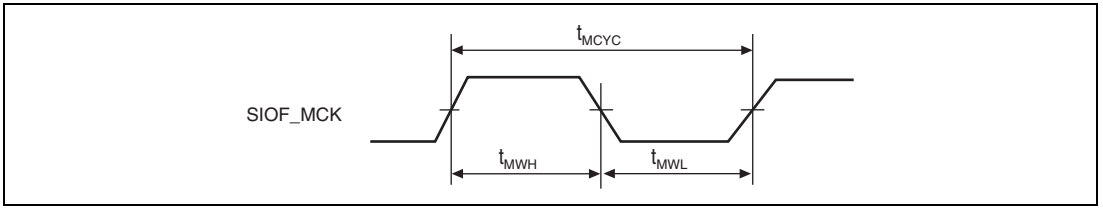


Figure 33.46 SIOF_MCK Input Timing

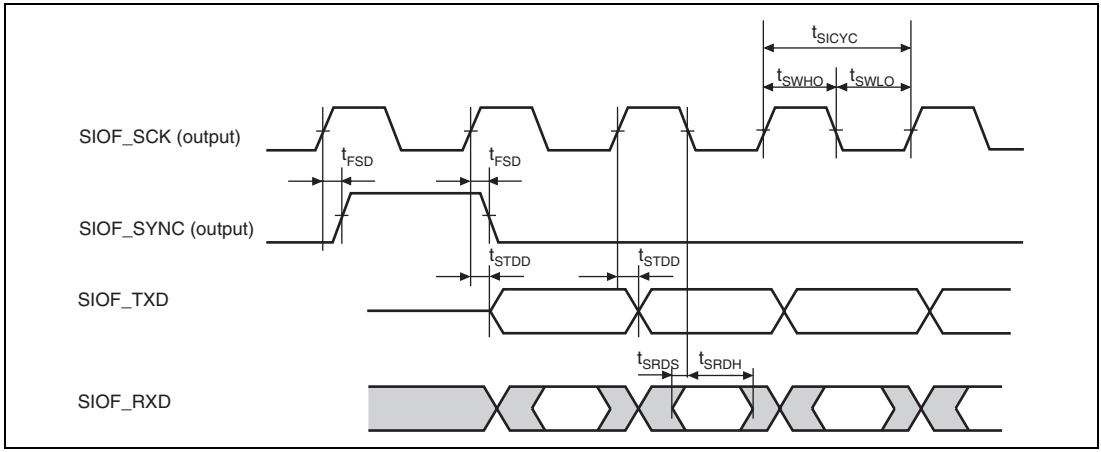


Figure 33.47 SIOF Transmission/Reception Timing (Master Mode 1, Fall Sampling)

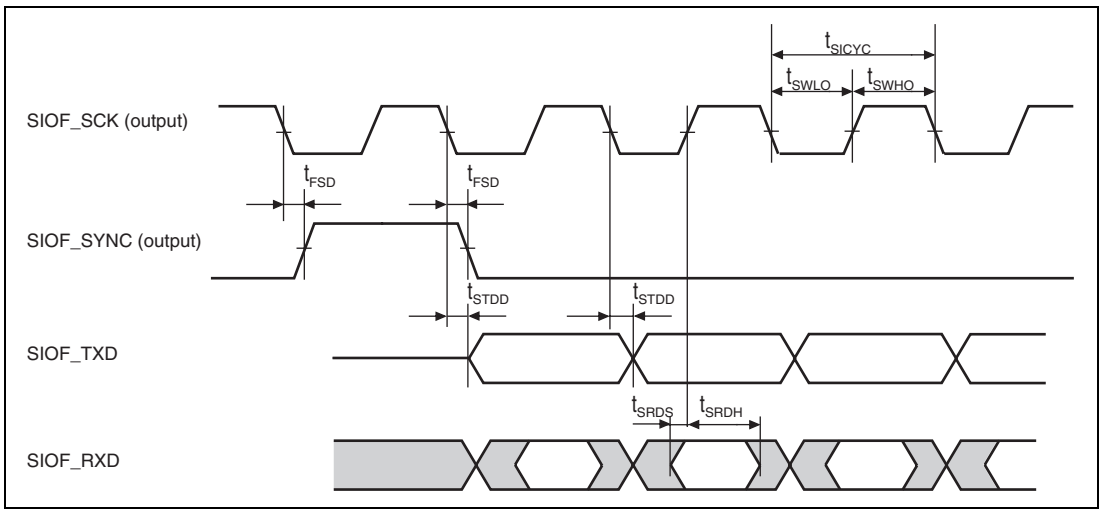


Figure 33.48 SIOF Transmission/Reception Timing (Master Mode 1, Rise Sampling)

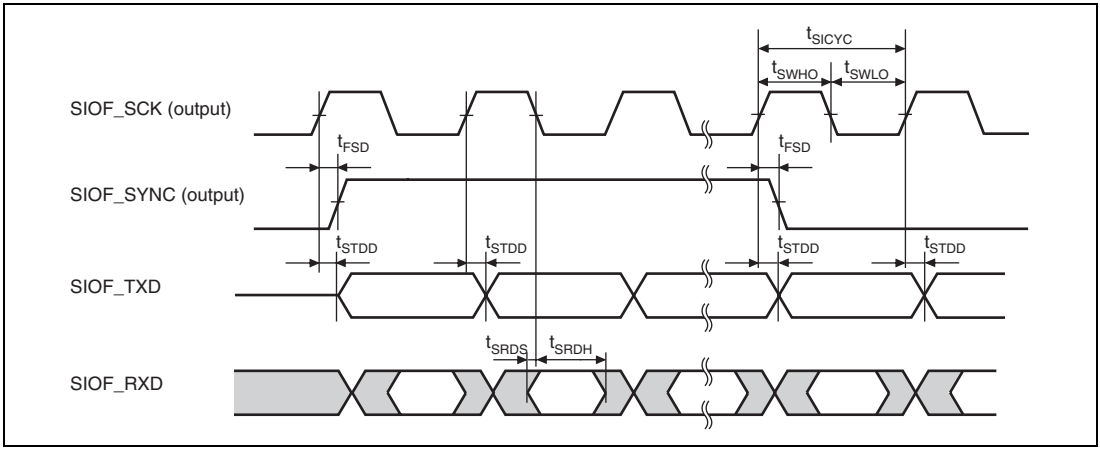


Figure 33.49 SIOF Transmission/Reception Timing (Master Mode 2, Fall Sampling)

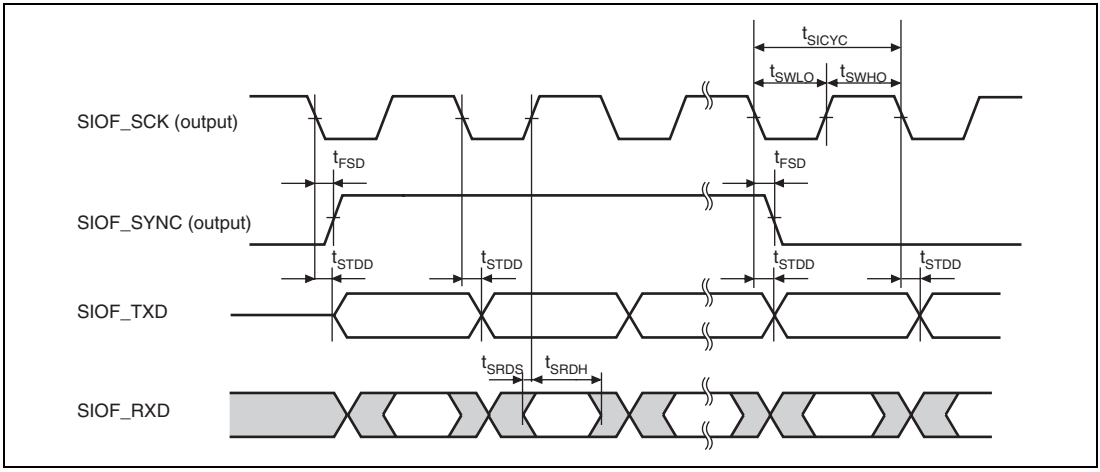


Figure 33.50 SIOF Transmission/Reception Timing (Master Mode 2, Rise Sampling)

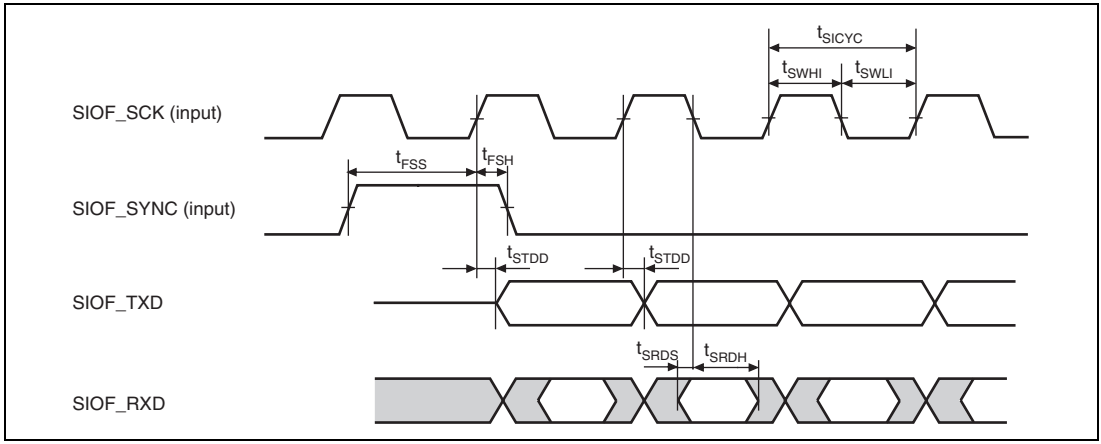


Figure 33.51 SIOF Transmission/Reception Timing (Slave Mode 1, Slave Mode 2)

33.4.13 SCIF/SCIFA Module Signal Timing

Table 33.15 SCIF/SCIFA Module Signal Timing (Asynchronous)

Item	Symbol	Min.	Max.	Unit	Figure
SCK input clock cycle	t_{SCYC}	$4 \times t_{pcyc}$	—	ns	33.52
SCK input clock high level width	t_{SCWH}	$0.4 \times t_{SCYC}$	$0.6 \times t_{SCYC}$	ns	
SCK input clock low level width	t_{SCWL}	$0.4 \times t_{SCYC}$	$0.6 \times t_{SCYC}$	ns	
SCK input clock rise time	t_{SCKr}	—	$1.5 \times t_{pcyc}$	ns	
SCK input clock fall time	t_{SCKf}	—	$1.5 \times t_{pcyc}$	ns	
TXD transmit data delay time	t_{TXD}	—	$3 \times t_{pcyc} + 50$	ns	33.53
RXD input data setup time	t_{RXS}	$2 \times t_{pcyc}$	—	ns	
RXD input data hold time	t_{RXH}	$2 \times t_{pcyc}$	—	ns	
RST delay time	t_{RTSD}	—	100	ns	
CTS setup time	t_{CTSS}	100	—	ns	
CTS hold time	t_{CTSH}	100	—	ns	

Note: t_{pcyc} is the cycle time of the peripheral clock (Pϕ).

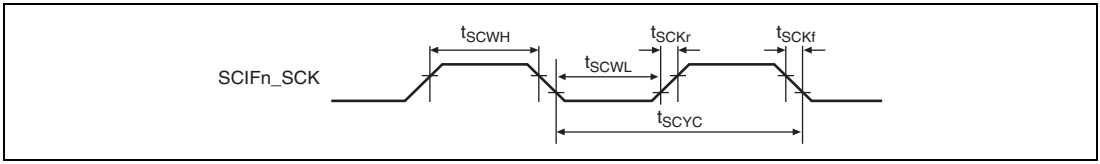


Figure 33.52 SCIF/SCIFA Module Signal Timing

Table 33.16 SCIF/SCIFA Module Signal Timing (Clock Synchronous)

Item	Symbol	Min.	Max.	Unit	Figure
SCK input/output clock cycle	t_{SCYC}	$12 \times t_{pcyc}$	—	ns	33.52
SCK input/output clock high level width	t_{SCWH}	$0.4 \times t_{SCYC}$	$0.6 \times t_{SCYC}$	ns	
SCK input/output clock low level width	t_{SCWL}	$0.4 \times t_{SCYC}$	$0.6 \times t_{SCYC}$	ns	
SCK input/output synchronous clock rise time	t_{SCKr}	—	$1.5 \times t_{pcyc}$	ns	
SCK input/output synchronous clock fall time	t_{SCKf}	—	$1.5 \times t_{pcyc}$	ns	
TXD output data delay time (SCK input)	t_{TXD}	—	$3 \times t_{pcyc} + 50$	ns	33.53
TXD output data delay time (SCK output)		—	50	ns	
RXD input data setup time (SCK input/output in common)	t_{RXS}	$4 \times t_{pcyc}$	—	ns	
RXD input data hold time (SCK input/output in common)	t_{RXH}	$4 \times t_{pcyc}$	—	ns	

Note: t_{pcyc} is the cycle time of the peripheral clock (Pϕ).

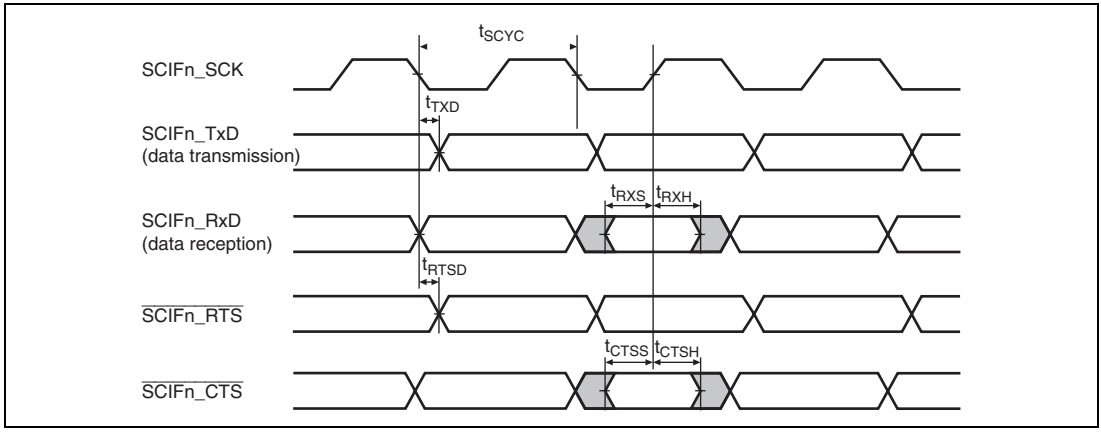


Figure 33.53 SCI Input/Output Timing in Clock Synchronous Mode

33.4.14 SIM Module Signal Timing

Table 33.17 SIM Module Signal Timing

Item	Symbol	Min.	Max.	Unit	Figure
SIM_CLK clock cycle	t_{SMCYC}	$2/t_{Pcyc}$	$16/t_{Pcyc}$	ns	33.54
SIM_CLK clock high level width	t_{SMCWH}	$0.4 \times t_{SMCYC}$	—	ns	
SIM_CLK clock low level width	t_{SMCWL}	$0.4 \times t_{SMCYC}$	—	ns	
SIM_RST reset output delay	t_{SMRD}	0	20	ns	

Note: t_{Pcyc} is the cycle time of the peripheral clock (Pϕ).

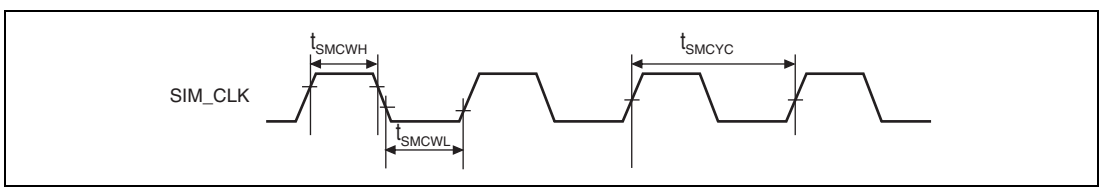


Figure 33.54 SIM Module Signal Timing

33.4.15 H-UDI Related Pin Timing

Table 33.18 H-UDI Related Pin Timing

Item	Symbol	Min.	Max.	Unit	Figure
TCK cycle time	t_{TCKcyc}	50	—	ns	33.55
TCK high pulse width	t_{TCKH}	12	—	ns	
TCK low pulse width	t_{TCKL}	12	—	ns	
TCK rise/fall fall time	t_{TCKf}	—	4	ns	
$\overline{TRST}$ setup time	t_{TRSTS}	12	—	ns	33.56
$\overline{TRST}$ hold time	t_{TRSTH}	50	—	t_{cyc}	
TDI setup time	t_{TDIS}	10	—	ns	33.57
TDI hold time	t_{TDIH}	10	—	ns	
TMS setup time	t_{TMSS}	10	—	ns	
TMS hold time	t_{TMSH}	10	—	ns	
TDO delay time	t_{TDOD}	—	16	ns	
$\overline{MPMD}$ setup time	t_{MPMDS}	12	—	ns	33.58
$\overline{MPMD}$ hold time	t_{MPMDH}	12	—	ns	

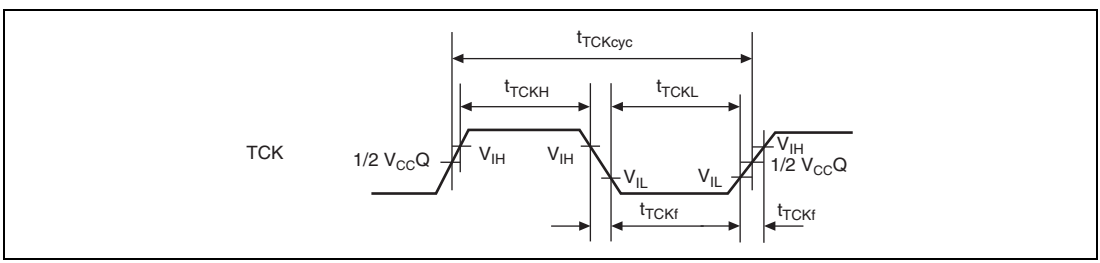


Figure 33.55 TCK Input Timing

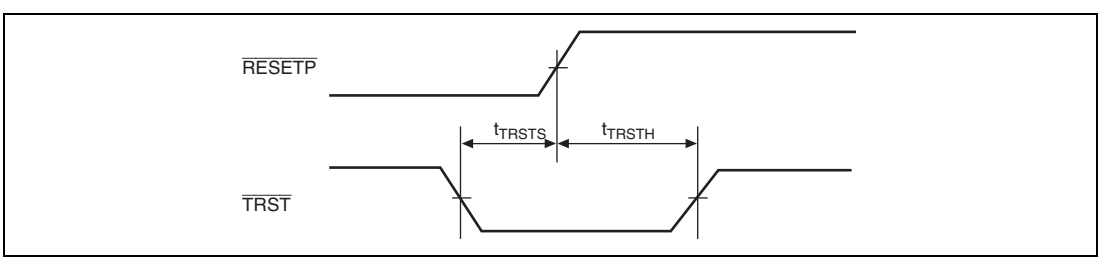


Figure 33.56 $\overline{TRST}$ Input Timing (Reset Hold)

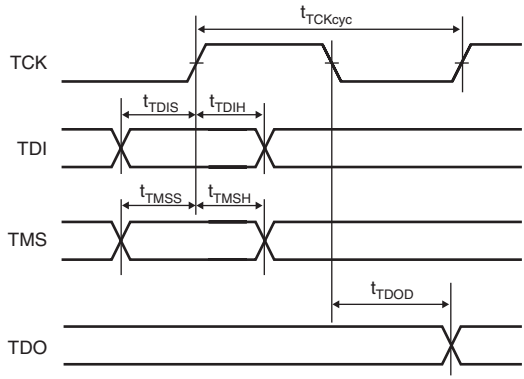


Figure 33.57 H-UDI Data Transfer Timing

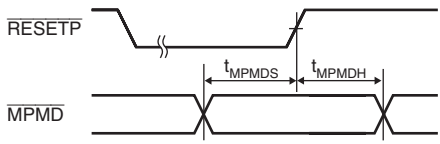


Figure 33.58 $\overline{\text{MPMD}}$ Input Timing

33.5 A/D Converter Characteristics

Table 33.19 lists the A/D converter characteristics.

Table 33.19 A/D Converter Characteristics

Item	Min.	Typ.	Max.	Unit
Resolution	10	10	10	bits
Conversion time	15	—	—	μs
Analog input capacitance	—	—	20	pF
Permissible signal source (single source) impedance	—	—	5	kΩ
Nonlinearity error	—	—	±3.0	LSB
Offset error	—	—	±2.0	LSB
Full scale error	—	—	±2.0	LSB
Quantization error	—	—	±0.5	LSB
Absolute accuracy	—	—	±4.0	LSB

33.6 D/A Converter Characteristics

Table 33.20 lists D/A converter characteristics.

Table 33.20 D/A Converter Characteristics

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	10	10	10	bits	
Conversion time	—	—	10.0	μs	20 pF capacitive load
Absolute accuracy	—	±10	±16	LSB	2 MΩ resistance load
	—	±2.5	±4.0	LSB	No resistance load

33.7 AC Characteristic Test Conditions

Unless otherwise specified, the AC characteristic test conditions are as follows.

(1) Temperature

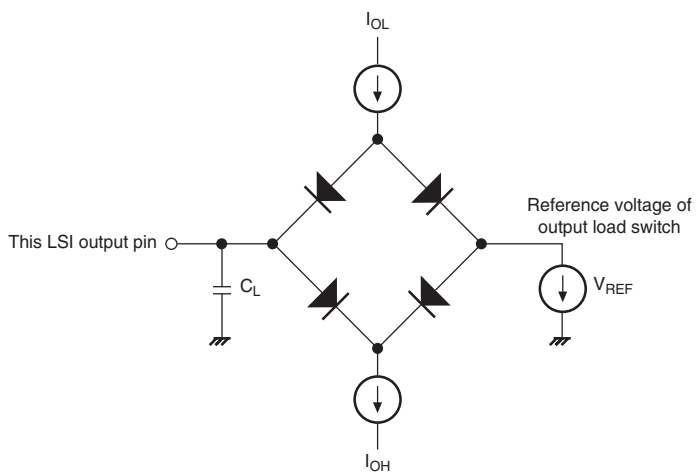
- Standard temperature range product: Ta = –20 to 75°C
- Extended temperature range product:Ta = –40 to 85°C

(2) Power Supply Voltage

- $V_{CC}Q$ = 3.0 to 3.6 V
- V_{CC} = V_{CC}_PLL1 = V_{CC}_PLL2 = 1.1 to 1.3 V
- AV_{CC} = 3.0 to 3.6 V

(3) Input Signals

- I/O signal reference level:
  $\frac{V_{CC}Q}{2}$, ($V_{CC}Q$ = 3.0 to 3.6 V, V_{CC} = 1.1 to 1.3 V)
- Input pulse level: V_{SS} to $V_{CC}Q$
- Input rise and fall times: 1 ns



Notes: 1. CL is the total value that includes the capacitance of measurement instruments, and is set as follows for each pin:
30 pF: CKO, $\overline{CS0}$, $\overline{CS2}$ to $\overline{CS6B}$
50 pF: All other pins
2. I_{OL} = 0.2 mA, I_{OH} = -0.2 mA

Figure 33.59 Output Load Circuit

Appendix

A. CPU Operation Mode Register (CPUOPM)

The CPUOPM is used to control the CPU operation mode. This register can be read from or written to the address H'FF2F0000 in P4 area or H'1F2F0000in area 7 as 32-bit size. The write value to the reserved bits should be the initial value. The operation is not guaranteed if the write value is not the initial value.

The CPUOPM register should be updated by the CPU store instruction not the access from SuperHyway bus master except CPU.

After the CPUOPM is updated, read CPUOPM once, and execute one of the following two methods.

1. Execute a branch using the RTE instruction.
2. Execute the ICBI instruction for any address (including non-cacheable area).

After one of these methods is executed, it is guaranteed that the CPU runs under the updated CPUOPM value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	RABD	—	INTMU	—	—	—
Initial value:	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved The write value must be the initial value.
9 to 6	—	1	R	Reserved The write value must be the initial value.
5	RABD	0	R/W	Speculative execution bit for subroutine return 0: Instruction fetch for subroutine return is issued speculatively. When this bit is set to 0, refer to appendix C, Speculative Execution for Subroutine Return. 1: Instruction fetch for subroutine return is not issued speculatively.
4	—	0	R	Reserved The write value must be the initial value.
3	INTMU	0	R/W	Interrupt mode switch bit 0: SR.IMASK is not changed when an interrupt is accepted. 1: SR.IMASK is changed to the accepted interrupt level.
2 to 0	—	All 0	R	Reserved The write value must be the initial value.

B. Instruction Prefetching and Its Side Effects

This LSI is provided with an internal buffer for holding pre-read instructions, and always performs pre-reading. Therefore, program code must not be located in the last 64-byte area of any memory space. If program code is located in these areas, a bus access for instruction prefetch may occur exceeding the memory areas boundary. A case in which this is a problem is shown below.

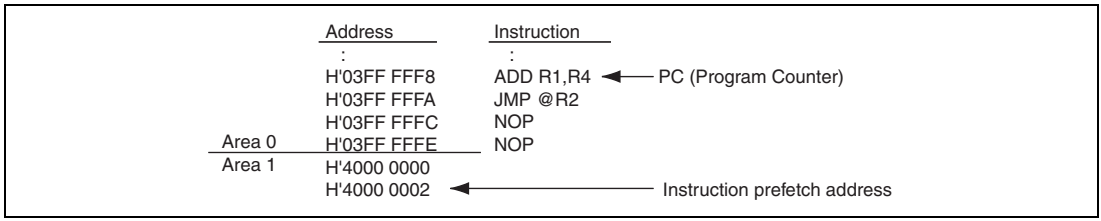


Figure B.1 Instruction Prefetch

Figure B.1 presupposes a case in which the instruction (ADD) indicated by the program counter (PC) and the address H'04000002 instruction prefetch are executed simultaneously. It is also assumed that the program branches to an area other than area 1 after executing the following JMP instruction and delay slot instruction.

In this case, a bus access (instruction prefetch) to area 1 may unintentionally occur from the programming flow.

(1) Instruction Prefetch Side Effects

1. It is possible that an external bus access caused by an instruction prefetch may result in misoperation of an external device, such as a FIFO, connected to the area concerned.
2. If there is no device to reply to an external bus request caused by an instruction prefetch, hang-up will occur.

(2) Remedies

1. These illegal instruction fetches can be avoided by using the MMU.
2. The problem can be avoided by not locating program code in the last 64 bytes of any area.

C. Speculative Execution for Subroutine Return

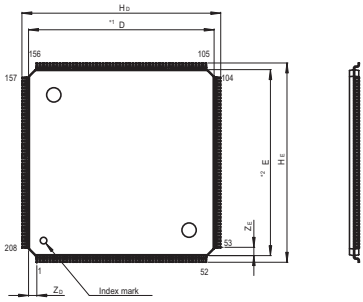
This LSI has the mechanism to issue an instruction fetch speculatively when returning from subroutine. By issuing an instruction fetch speculatively, the execution cycles to return from subroutine may be shortened.

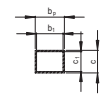
This function is enabled by setting 0 to the bit 5 (RABD) of CPU Operation Mode register (CPUOPM). But this speculative instruction fetch may issue the access to the address that should not be accessed from the program. Therefore a bus access to an unexpected area or an internal instruction address error may cause a problem. As for the effect of this bus access to unexpected memory area, refer to appendix B, Instruction Prefetching and Its Side Effects

Usage Condition: When the speculative execution for subroutine return is enabled, the RTS instruction should be used to return to the address set in PR by the JSR, BSR or BSRF instructions. It can prevent the access to unexpected address and avoid the problem.

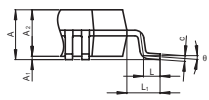
D. Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP208-28x28-0.50	PLQP0208KB-A	—	2.7g





Terminal cross section



Detail F

NOTE)
1. DIMENSIONS"*1"AND"*2" DO NOT INCLUDE MOLD FLASH
2. DIMENSION"*3"DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	27.9	28.0	28.1
E	27.9	28.0	28.1
A_2	—	1.40	—
H_D	29.8	30.0	30.2
H_E	29.8	30.0	30.2
A	—	—	1.70
A_1	0.05	0.10	0.15
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.095	0.145	0.195
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.50	0.65
L_1	—	1.00	—

Figure D.1 Package Dimensions

E. Pin State During Reset and Power-Down Mode and Handling of Pins Not in Use

Table E.1 Pin State and Handling of Pins Not in Use

Pin No.	Pin Name	Input/ Output	Power-On Reset: $\overline{\text{RESETP}}$= Low Level, $\overline{\text{RESETOUT}}$ Low Level	Power-On Reset: $\overline{\text{RESETP}}$= High Level, $\overline{\text{RESETOUT}}$ Low Level	Software Standby	Bus Mastership Release	Handling of Pin When Not in Use
1	AN2/PTM0	I/I	Z	Z	I/K	Z	Pull-up*[7]
2	AN3/PTM1	I/I	Z	Z	I/K	Z	Pull-up*[7]
3	VccQ	—	—	—	—	—	Connect to the power supply (3.3 V)
4	XTAL_RTC	O	O	O	O	O	Open
5	EXTAL_RTC	I	I	I	I	I	Pull-up*[7]
6	VssQ	—	—	—	—	—	Connect to the ground (0 V)
7	Test0_VccQ	I	I	I	—	—	Pull-up*[7]
8	IRQ0/$\overline{\text{IRL0}}$/PTH0	I/I/I	Z	IU	I/I/K	IU	Open*[1]
9	IRQ1/$\overline{\text{IRL1}}$/PTH1	I/I/I	Z	IU	I/I/K	IU	Open*[1]
10	IRQ2/$\overline{\text{IRL2}}$/PTH2	I/I/I	Z	IU	I/I/K	IU	Open*[1]
11	IRQ3/$\overline{\text{IRL3}}$/PTH3	I/I/I	Z	IU	I/I/K	IU	Open*[1]
12	IRQ4/PTH4	I/I	Z	IU	I/K	IU	Open*[1]
13	D31/PTB7	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]
14	D30/PTB6	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]
15	D29/PTB5	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]
16	D28/PTB4	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]
17	D27/PTB3	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]
18	D26/PTB2	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]
19	VssQ	—	—	—	—	—	Connect to the ground (0 V)
20	D25/PTB1	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]
21	VccQ	—	—	—	—	—	Connect to the power supply (3.3 V)
22	D24/PTB0	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]

Pin No.	Pin Name	Input/ Output	Power-On Reset: RESETP= Low Level, RESETOUT Low Level	Power-On Reset: RESETP= High Level, RESETOUT Low Level	Software Standby	Bus Mastership Release	Handling of Pin When Not in Use
23	D23/PTA7	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]
24	D22/PTA6	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]
25	D21/PTA5	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]
26	D20/PTA4	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]
27	Vss	—	—	—	—	—	Connect to the ground (0 V)
28	D19/PTA3	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]
29	Vcc	—	—	—	—	—	Connect to the power supply (1.2 V)
30	D18/PTA2	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]
31	D17/PTA1	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]
32	D16/PTA0	IO/IO	Z	Z	Z/K	Z	Pull-up*[7]
33	VssQ	—	—	—	—	—	Connect to the ground (0 V)
34	D15	IO	Z	Z	Z	Z	Pull-up*[7]
35	VccQ	—	—	—	—	—	Connect to the power supply (3.3 V)
36	D14	IO	Z	Z	Z	Z	Pull-up*[7]
37	D13	IO	Z	Z	Z	Z	Pull-up*[7]
38	D12	IO	Z	Z	Z	Z	Pull-up*[7]
39	D11	IO	Z	Z	Z	Z	Pull-up*[7]
40	D10	IO	Z	Z	Z	Z	Pull-up*[7]
41	D9	IO	Z	Z	Z	Z	Pull-up*[7]
42	D8	IO	Z	Z	Z	Z	Pull-up*[7]
43	D7	IO	Z	Z	Z	Z	Pull-up*[7]
44	D6	IO	Z	Z	Z	Z	Pull-up*[7]
45	VssQ	—	—	—	—	—	Connect to the ground (0 V)
46	D5	IO	Z	Z	Z	Z	Pull-up*[7]
47	VccQ	—	—	—	—	—	Connect to the power supply (3.3 V)
48	D4	IO	Z	Z	Z	Z	Pull-up*[7]

Pin No.	Pin Name	Input/ Output	Power-On Reset: RESETP= Low Level, RESETOUT Low Level	Power-On Reset: RESETP= High Level, RESETOUT Low Level	Software Standby	Bus Mastership Release	Handling of Pin When Not in Use
49	D3	IO	Z	Z	Z	Z	Pull-up※7
50	D2	IO	Z	Z	Z	Z	Pull-up※7
51	D1	IO	Z	Z	Z	Z	Pull-up※7
52	D0	IO	Z	Z	Z	Z	Pull-up※7
53	A0/PTT0	O/IO	L	L	R/K	Z	Open
54	A1	O	L	L	R	Z	Open
55	A2	O	L	L	R	Z	Open
56	A3	O	L	L	R	Z	Open
57	VssQ	—	—	—	—	—	Connect to the ground (0 V)
58	A4	O	L	L	R	Z	Open
59	VccQ	—	—	—	—	—	Connect to the power supply (3.3 V)
60	A5	O	L	L	R	Z	Open
61	A6	O	L	L	R	Z	Open
62	A7	O	L	L	R	Z	Open
63	A8	O	L	L	R	Z	Open
64	A9	O	L	L	R	Z	Open
65	A10	O	L	L	R	Z	Open
66	A11	O	L	L	R	Z	Open
67	A12	O	L	L	R	Z	Open
68	A13	O	L	L	R	Z	Open
69	VssQ	—	—	—	—	—	Connect to the ground (0 V)
70	A14	O	L	L	R	Z	Open
71	VccQ	—	—	—	—	—	Connect to the power supply (3.3 V)
72	A15	O	L	L	R	Z	Open
73	A16	O	L	L	R	Z	Open
74	A17	O	L	L	R	Z	Open
75	A18	O	L	L	R	Z	Open

Pin No.	Pin Name	Input/ Output	Power-On Reset: RESETP= Low Level, RESETOUT Low Level	Power-On Reset: RESETP= High Level, RESETOUT Low Level	Software Standby	Bus Mastership Release	Handling of Pin When Not in Use
76	A19/PTT1	O/IO	L	L	R/K	Z	Open
77	A20/PTT2	O/IO	L	L	R/K	Z	Open
78	A21/PTT3	O/IO	L	L	R/K	Z	Open
79	Vss	—	—	—	—	—	Connect to the ground (0 V)
80	A22/PTT4	O/IO	L	L	R/K	Z	Open
81	Vcc	—	—	—	—	—	Connect to the power supply (1.2 V)
82	A23/PTT5	O/IO	L	L	R/K	Z	Open
83	VssQ	—	—	—	—	—	Connect to the ground (0 V)
84	A24/PTT6	O/IO	L	L	R/K	Z	Open
85	VccQ	—	—	—	—	—	Connect to the power supply (3.3 V)
86	A25/PTT7	O/IO	L	L	R/K	Z	Open
87	$\overline{BS}$/PTK4	O/IO	H	H	R/K	Z	Open
88	$\overline{RD}$	O	H	H	R	Z	Open
89	$\overline{WE0}$/DQMLL	O/O	H	H	R/R	Z	Open
90	$\overline{WE1}$/DQMLU	O/O	H	H	R/R	Z	Open
91	$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK6	O/O/O/IO	H	H	R/R/R/K	Z	Open
92	$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK7	O/O/O/IO	H	H	R/R/R/K	Z	Open
93	$\overline{RDWR}$	O	H	H	R	Z	Open
94	SCIF3_RXD/ PTE7	I/IO	Z	IU	Z/K	IU	Open*[1]
95	VssQ	—	—	—	—	—	Connect to the ground (0 V)
96	$\overline{CS0}$	O	H	H	R	Z	Open
97	VccQ	—	—	—	—	—	Connect to the power supply (3.3 V)
98	$\overline{CS2}$/PTK0	O/IO	H	H	R/K	Z	Open
99	$\overline{CS3}$/PTK1	O/IO	H	H	R/K	Z	Open

Pin No.	Pin Name	Input/ Output	Power-On Reset: RESETP= Low Level, RESETOUT Low Level	Power-On Reset: RESETP= High Level, RESETOUT Low Level	Software Standby	Bus Mastership Release	Handling of Pin When Not in Use
100	$\overline{CS4}$/PTK2	O/IO	H	H	R/K	Z	Open
101	$\overline{CS5B}$/$\overline{CE1A}$/PTK3	O/O/IO	H	H	R/R/K	Z	Open
102	$\overline{CS6B}$/$\overline{CE1B}$/PTM3	O/O/IO	H	H	R/R/K	Z	Open
103	$\overline{CS5A}$/$\overline{CE2A}$/PTE4	O/O/IO	H	H	R/R/K	Z	Open
104	$\overline{CS6A}$/$\overline{CE2B}$/PTE5	O/O/IO	H	H	R/R/K	Z	Open
105	CKE/PTK5	O/IO	Z	H	R/K	R*9	Open
106	$\overline{RAS}$/PTJ0	O/IO	H	H	R/K	R*9	Open
107	TEND1/PTJ1	O/IO	Z	IU	O/K	IU	Open*1
108	$\overline{CAS}$/PTJ2	O/IO	H	H	R/K	R*9	Open
109	VssQ	—	—	—	—	—	Connect to the ground (0 V)
110	TEND0/PTJ3	O/IO	Z	IU	O/K	IU	Open*1
111	VccQ	—	—	—	—	—	Connect to the power supply (3.3 V)
112	IRQ6/PTJ4	I/I	Z	IU	I/K	IU	Open*1
113	IRQ7/PTJ5	I/I	Z	IU	I/K	IU	Open*1
114	DACK0/PTD5	O/IO	Z	IU	O/K	IU	Open*1
115	DACK1/PTD7	O/IO	Z	IU	O/K	IU	Open*1
116	SCIF4_RXD/PTE6	I/I	Z	IU	Z/K	IU	Open*1
117	SCIF5_RXD/PTE3	I/I	Z	IU	Z/K	IU	Open*1
118	SCIF4_SCK/PTE2	IO/IO	Z	IU	Z/K	IU	Open*1
119	SCIF5_SCK/PTE1	IO/IO	Z	IU	Z/K	IU	Open*1
120	TDO	O	Z*8	Z*8	H	Z	Open
121	$\overline{BACK}$/PTN0	O/IO	H	H	O/K	L	Open
122	$\overline{BREQ}$/PTN1	I/I	I	I	I/K	K	Pull-up*3*7
123	$\overline{WAIT}$/PTN2	I/I	I	I	I/K	Z	Pull-up*3*7
124	AUDCK/PTG5	O/IO	O	O	O/K	L	Open
125	IIC1_SDA/$\overline{ADTRG}$/ PTH5	IO/I/I	Z	I	Z/I/K	Z	Pull-up*7

Pin No.	Pin Name	Input/ Output	Power-On Reset: RESETP= Low Level, RESETOUT Low Level	Power-On Reset: RESETP= High Level, RESETOUT Low Level	Software Standby	Bus Mastership Release	Handling of Pin When Not in Use
126	IIC1_SCL/IOIS16/ PTN3	IO/I/I	Z	I	Z/I/K	Z	Pull-up*7
127	MPMD	I	IU	IU	IU	IU	Pull-up*4*7
128	ASEBRK/BRKACK	I/O	IU	IU	R	IU	Open
129	AUDSYNC/PTG4	O/IO	O	O	O/K	H	Open
130	AUDATA3/PTG3	O/IO	O	O	O/K	L	Open
131	AUDATA2/PTG2	O/IO	O	O	O/K	L	Open
132	Vss	—	—	—	—	—	Connect to the ground (0 V)
133	AUDATA1/PTG1	O/IO	O	O	O/K	L	Open
134	Vcc	—	—	—	—	—	Connect to the power supply (1.2 V)
135	AUDATA0/PTG0	O/IO	O	O	O/K	L	Open
136	TRST	I	IU	IU	IU	IU	Connect to the ground (0 V)
137	TMS	I	IU	IU	IU	IU	Open*1
138	TDI	I	IU	IU	IU	IU	Open*1
139	TCK	I	IU	IU	IU	IU	Open*1
140	TPU0_TO3/PINTB3/ PTF3	O/I/IO	Z	ID	O/I/K	ID	Open*2
141	TPU0_TO2/PINTB2/ PTF2	O/I/IO	Z	ID	O/I/K	ID	Open*2
142	TPU0_TO1/PINTB1/ PTF1	O/I/IO	Z	ID	O/I/K	ID	Open*2
143	TPU0_TO0/PINTB0/ PTF0	O/I/IO	Z	ID	O/I/K	ID	Open*2
144	PTN4	IO	Z	IU	K	IU	Open*1
145	Vcc_PLL1	—	—	—	—	—	Connect to the power supply (1.2 V)*6
146	Test1_VssQ	I	I	I	—	—	Pull-down
147	Vss_PLL1	—	—	—	—	—	Connect to the ground (0 V)*6

Pin No.	Pin Name	Input/ Output	Power-On Reset: RESETP= Low Level, RESETOUT Low Level	Power-On Reset: RESETP= High Level, RESETOUT Low Level	Software Standby	Bus Mastership Release	Handling of Pin When Not in Use
148	Vss_PLL2	—	—	—	—	—	Connect to the ground (0 V)*6
149	Test2_VccQ	I	IU	IU	—	—	Pull-up*7
150	Vcc_PLL2	—	—	—	—	—	Connect to the power supply (1.2 V)*6
151	PTH6	I	Z	IU	K	IU	Open*1
152	Vss	—	—	—	—	—	Connect to the ground (0 V)
153	Test3_VccQ	I	I	I	—	—	Pull-up*7
154	Vcc	—	—	—	—	—	Connect to the power supply (1.2 V)
155	XTAL	O	O	O	O	O	Open
156	EXTAL	I	I	I	I	I	Pull-up*7
157	STATUS0/PTJ6	O/IO	Z	L	H/K	L	Open
158	TPU1_TO1/PTJ7	O/IO	Z	IU	O/K	IU	Open*1
159	TPU1_TO0/PTH7	O/IO	Z	IU	O/K	IU	Open*1
160	IRQOUT/REFOUT/ PTQ7	O/O/IO	Z	O	Z/Z/K	H	Open
161	VssQ	—	—	—	—	—	Connect to the ground (0 V)
162	CKO	O	O	O	O	R*9	Open
163	VccQ	—	—	—	—	—	Connect to the power supply (3.3 V)
164	SCIF0_TXD/IRDA0_ TXD/PTQ2	O/O/IO	Z	IU	O/O/K	IU	Open*1
165	SCIF0_SCK/PTQ0	IO/IO	Z	IU	Z/K	IU	Open*1
166	SCIF1_TXD/IRDA1_ TXD/PTR2	O/O/IO	Z	IU	O/O/K	IU	Open*1
167	SCIF1_SCK/PTR0	IO/IO	Z	IU	Z/K	IU	Open*1
168	SCIF2_TXD/SIOF_ TXD/PTS2	O/O/IO	Z	IU	O/O/K	IU	Open*1
169	SCIF2_SCK/SIOF_ SCK/PTS0	IO/IO/IO	Z	IU	Z/R/K	IU	Open*1

Pin No.	Pin Name	Input/ Output	Power-On Reset: RESETP= Low Level, RESETOUT Low Level	Power-On Reset: RESETP= High Level, RESETOUT Low Level	Software Standby	Bus Mastership Release	Handling of Pin When Not in Use
170	SCIF2_RTS/SIOF_ SYNC/PTS4	O/IO/IO	Z	IU	O/R/K	IU	Open*1
171	SCIF0_RXD/IRDA0_ RXD/PTQ1	I/I/I	Z	IU	Z/Z/K	IU	Open*1
172	SCIF1_RXD/IRDA1_ RXD/PTR1	I/I/I	Z	IU	Z/Z/K	IU	Open*1
173	Vss	—	—	—	—	—	Connect to the ground (0 V)
174	SCIF2_RXD/SIOF_ RXD/PTS1	I/I/I	Z	IU	Z/Z/K	IU	Open*1
175	Vcc	—	—	—	—	—	Connect to the power supply (1.2 V)
176	SCIF2_CTS/SIOF_MCK/ IRQ5/PTS3	I/I/I/I	Z	IU	Z/Z/I/K	IU	Open*1
177	SCIF5_RTS/PINTA7/ PTC7	O/I/IO	Z	ID	O/I/K	ID	Open*2
178	SCIF5_CTS/PINTA6/ PTC6	I/I/IO	Z	ID	Z/I/K	ID	Open*2
179	SCIF4_RTS/PINTA5/ PTC5	O/I/IO	Z	ID	O/I/K	ID	Open*2
180	SCIF4_CTS/PINTA4/ PTC4	I/I/IO	Z	ID	Z/I/K	ID	Open*2
181	VssQ	—	—	—	—	—	Connect to the ground (0 V)
182	SCIF3_TXD/SIM_D/PTD3	O/IO/IO	Z	IU	O/Z/K	IU	Open*1
183	VccQ	—	—	—	—	—	Connect to the power supply (3.3 V)
184	RESETOUT/PTD2	O/IO	L	L	H/K	H	Open
185	PINTA3/PTC3	I/IO	Z	ID	I/K	ID	Open*2
186	SCIF3_RTS/SIM_RST/ PINTA2/PTC2	O/O/I/IO	Z	ID	O/O/I/K	ID	Open*2
187	SCIF3_CTS/PINTA1/PTC1	I/I/IO	Z	ID	Z/I/K	ID	Open*2

Pin No.	Pin Name	Input/ Output	Power-On Reset: RESETP= Low Level, RESETOUT Low Level	Power-On Reset: RESETP= High Level, RESETOUT Low Level	Software Standby	Bus Mastership Release	Handling of Pin When Not in Use
188	SCIF3_SCK/SIM_CLK/ PINTA0/PTC0	IO/O/I/IO	Z	ID	Z/O/I/K	ID	Open*2
189	SCIF5_TXD/PTD1	O/IO	Z	IU	O/K	IU	Open*1
190	SCIF4_TXD/PTD0	O/IO	Z	IU	O/K	IU	Open*1
191	DREQ0/PTD4	I/I	Z	IU	I/K	IU	Open*1
192	DREQ1/PTD6	I/I	Z	IU	I/K	IU	Open*1
193	$\overline{\text{RESETP}}$	I	I	I	I	I	Must be used
194	NMI	I	I	I	I	I	Pull-up*7
195	MD3	I	I	I	I	I	Must be used
196	Test4_VssQ	I	I	I	—	—	Pull-down
197	MD5	I	I	I	I	I	Must be used
198	VssQ	—	—	—	—	—	Connect to the ground (0 V)
199	IIC0_SDA/PTL0	I/I	Z	I	Z/K	Z	Pull-up*7
200	IIC0_SCL/PTL1	I/I	Z	I	Z/K	Z	Pull-up*7
201	MD0	I	I	I	I	I	Must be used
202	MD1	I	I	I	I	I	Must be used
203	DA1/PTL4	O/I	Z	Z	K/K	Z	Pull-up*7
204	DA0/PTL5	O/I	Z	Z	K/K	Z	Pull-up*7
205	AVcc	—	—	—	—	—	Connect to the power supply (3.3 V)
206	AN0/PTL6	I/I	Z	Z	I/K	Z	Pull-up*7
207	AN1/PTL7	I/I	Z	Z	I/K	Z	Pull-up*7
208	AVss	—	—	—	—	—	Connect to the ground (0 V)

[Legend]

I: Input

IU: Input (Pull-up MOS ON)

ID: Input (Pull-down MOS ON)

O: Output

H: High level output

L: Low level output

Z: Hi-Z

R: Depends on the pin state or the setting of registers.

K: Input is fixed; output pins and pull-up/pull-down MOS retain their states.

Notes:
1. Using an external pull-up resistor is safer. The resistance is greater, if the on-chip pull-up resistance alone is used. Check to ensure that there is no problem with the system.
2. Using an external pull-down resistor is safer. The resistance is greater, if the on-chip pull-up resistance alone is used. Check to ensure that there is no problem with the system.
3. Even if this pin is to be used as a port pin, keep the level on it high until it has been switched to the port function.
4. Set this pin to the high level when an emulator or the H-UDI is not in use, i.e. in stand-alone usage of the user system.
5. TRST is pulled up inside the chip. A little electric current thus flows while the pin is externally connected to ground. This current has no effect on the operation of the chip but does consume power unnecessarily. TRST pull-up can be turned off by the PULCR register of the PFC. See section 29, Pin Function Controller (PFC) for details.
6. See section 13.7, Notes on Board Design.
7. Connect a pull-up resistor to VccQ (3.3V).
8. Varies according to the state of the TAP controller.
9. Depends on the setting of the CMNCR in BSC. See Table E.2.

Table E.2 Pin States of CKO, CKE, RAS, and CAS pins in Bus Mastership Release Mode

CMNCR			Pin State		
HIZCNT	CKODRV	CKOSTP	CKO	CKE	RAS, CAS
0	0	0	Hi-Z	Hi-Z	Hi-Z
		1			
	1	0	Output	Output	
		1	Low level		
1	0	0	Output		Output
		1	Low level		
	1	0	Output		
		1	Low level		

Main Revisions for This Edition

Revised items due to the version upgrade from Rev.3.00 to Rev.4.00

Item	Page	Revision (See Manual for Details)
1.1 Features of This LSI Table 1.1 Features of This LSI	6	Table amended

Item	Features
Clock pulse generator (CPG)	• Clock mode: Input clock selectable from external input (EXTAL) and crystal resonator • Output clock: Bus clock (Bφ) • Generates four types of system clocks CPU clock (Iφ): Maximum 266.7 MHz (266 MHz version) Maximum 200 MHz (200 MHz version) SH (SuperHyway) clock (Sφ): Maximum 133.4 MHz Bus clock (Bφ): Maximum 66.7 MHz Peripheral clock (Pφ): Maximum 33.4 MHz • Supports power-down mode Sleep mode Software standby mode Module standby mode

1.3.1 Pin Function Table 1.3 Pin Functions	26	Note added Note: Pins Test0_VccQ, Test1_VccQ, Test2_VccQ, Test3_VccQ, and Test4_VssQ are used for testing the chip prior to shipment from the factory. In ordinary use, they should be pulled up or pulled down to fix their electric potential.					
1.4 Product Lineup Table 1.4 Product Lineup	27	Table amended					

Abbreviation	Parts No.	Power Supply Voltage I/O	Power Supply Voltage Internal	Operating Frequency	Operating Temperature	Package
R8A77301	R8A77301C266FPV	3.3 ± 0.3 V	1.2 ± 0.1 V	266.7 MHz	Standard temperature range product -20 to 75 °C	208 Pin Plastic LQFP (PLQP0208KB-A)
	R8A77301C200FPV			200 MHz		
	R8A77301D266FPV			266.7 MHz	Extended temperature range product -40to 85 °C	
	R8A77301D200FPV			200 MHz		

6.7 Notes on Use of SH7780 FPU Operation Instructions	146	News added

Item	Page	Revision (See Manual for Details)
11.4.1 Common Control Register (CMNCR)	296	Register amended

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	CKO STP	CKO DRV	—	—	—	—	—	—	—	DM STP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BSD	MAP[1:0]		BLOCK	—	—	—	—	—	—	—	END IAN	—	HIZ MEM	HIZ CNT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0/1*	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R/W	R/W

Table amended

Bit	Bit Name	Initial Value	R/W	Description
10 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

11.4.6 Refresh Timer Control/Status Register (RTCSR)	332	Register amended

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CMF	—	CKS[2:0]			RRC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

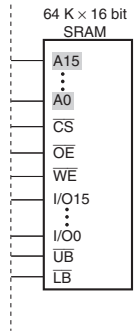
Table amended

Bit	Bit Name	Initial Value	R/W	Description
6	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.

11.4.8 Refresh Time Constant Register (RTCOR)	335	Description amended

When the RFSH bit in SDCR is 1, a memory refresh request is issued by this matching signal. This request is maintained until the refresh operation is performed. If the request is not processed when the next matching occurs, the previous request is ignored.

Clearing the CMF bit does not clear refresh requests. When the RTCOR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

Item	Page	Revision (See Manual for Details)
11.5.5 SDRAM Interface (2) Address Multiplexing Table 11.21 Relationship between A3BSZ[1:0], A3ROW[1:0], A3COL[1:0], and Address Multiplex Output (7)	365	Table amended Setting A3 BSZ [1:0]: 11 (32 bits) A3 ROW [1:0]: 10 (13 bits) A3 COL [1:0]: 10 (10 bits) Output Pin of This LSI: Row Address Output / Column Address Output Synchronous DRAM Pin Function
11.5.7 Byte-Selection SRAM Interface Figure 11.34 Example of Connection with 16-Bit Data-Width Byte-Selection SRAM	393	Figure amended 
11.5.10 Bus Arbitration	403	Description deleted Note that refresh requests and bus mastership requests are also accepted during DMA burst transfers. This LSI has the bus mastership until a bus request is received from another device. Upon acknowledging the assertion (low level) of the external bus request signal $\overline{\text{BREQ}}$, the LSI releases the bus at the completion of the current bus cycle and asserts the $\overline{\text{BACK}}$ signal. After the LSI acknowledges the negation (high level) of the $\overline{\text{BREQ}}$ signal that indicates the slave has released the bus, it negates the $\overline{\text{BACK}}$ signal and resumes the bus usage.
12.5.7 Restarting a DMA Transfer After Interruption	456	News added

Item	Page	Revision (See Manual for Details)
13.3 Clock Operating Modes Table 13.2 Clock Operating Modes	462	Table amended

Clock Mode	Pin Setting MD1	Pin Setting MD0	Register Initial Value FRQCR	Register Initial Value PLLCR	Clock Source	PLL (Multiplication Ratio)	Initial Clock Ratio Iϕ	Initial Clock Ratio Sϕ	Initial Clock Ratio Bϕ	Initial Clock Ratio Pϕ
0	0	0	H'0755 5558	H'0000 4000	EXTAL※1	ON (×8)	2	2	2	1
1	0	1	H'0700 0000	H'0000 0000	EXTAL※1	OFF	1/2	1/2	1/2	1/2
2	1	0	H'0755 5558	H'0000 4000	Crystal oscillator※2	ON (×8)	2	2	2	1
3	1	1	Setting prohibited							

21.4.7 Transmit and Receive Procedures (1) Transmission in Master Mode Figure 21.9 Example of Transmit Operation in Master Mode	663	Figure Replaced
(2) Reception in Master Mode Figure 21.10 Example of Receive Operation in Master Mode	664	Figure Replaced
26.4.4 Input Sampling and A/D Conversion Time Table 26.5 A/D Conversion Time (Single Mode)	886	Note added Note: Values in the table are numbers of states (t_{PVR}). Make settings such that the minimum conversion time is satisfied.
28.6.1 Port E Data Register (PEDR) Table 28.7 Port E Data Register (PEDR) Read/Write Operations • PE3DT, PE6DT	909	Table amended

PECR State PEnMD1	PECR State PEnMD0	Pin State	Read	Write
0	0	Other function	PEDR value	The value is written to PEDR, but does not affect the pin state.
	1	Output	PEDR value	The write value is output from the pin.
1	0	Input (Pull-up MOS on)	Pin state	The value is written to PEDR, but does not affect the pin state.
	1	Input (Pull-up MOS off)	Pin state	The value is written to PEDR, but does not affect the pin state.

33.1 Absolute Maximum Ratings Table 33.1 Absolute Maximum Ratings	1071	Table amended

Item		Symbol	Value	Unit
Operating temperature※1	Standard temperature range product	T_{opr}	–20 to 75	°C
	Extended temperature range product		–40 to 85	

Item	Page	Revision (See Manual for Details)
33.1 Absolute Maximum Ratings Table 33.1 Absolute Maximum Ratings	1071	Note added Note: 1. For a list of the operating temperatures of individual product versions, see table 1.4, Product Lineup.
33.3 DC Characteristics Table 33.4 DC Characteristics (1) [Common]	1075	Conditions amended Conditions: T_a = –20°C to +75°C (Standard temperature range product), T_a = –40°C to +85°C (Extended temperature range product)

Table amended

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Supply current*[3]	Normal operation	I_{CC}	—	135	275	mA	V_{CC} = 1.2V Iϕ = 266.7 MHz
			—	110	220	mA	V_{CC} = 1.2V Iϕ = 200 MHz
		$I_{CC}Q$	—	30	40	mA	$V_{CC}Q$ = 3.3 V Bϕ = 33.4 MHz
	Sleep mode	I_{CC}	—	30	50	mA	When sleep mode is entered after a power-on reset: $V_{CC}Q$ = 3.3 V Bϕ = 33.4 MHz
		$I_{CC}Q$	—	20	25	mA	
	Standby mode	I_{CC}	—	0.4	3	mA	T_a = 25°C $V_{CC}Q$ = 3.3 V V_{CC} = 1.2 V
		$I_{CC}Q$	—	15	30	µA	

Table 33.4 DC Characteristics (2-a) [Except for I^2C Related Pins]	1076	Conditions amended Conditions: T_a = –20°C to +75°C (Standard temperature range product), T_a = –40°C to +85°C (Extended temperature range product)					

Table amended

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	$\overline{RESETP}$, MD0, MD1, MD3, MD5, $\overline{TRST}$, MPMD, EXTAL, NMI	V_{IH}	$V_{CC}Q \times 0.9$	—	$V_{CC}Q + 0.3$	V	
	PTM0, PTM1, PTL4, PTL5, PTL6, PTL7		2.0	—	AV_{CC} + 0.3		
	Other input pins		2.0	—	$V_{CC}Q + 0.3$		
Input low voltage	$\overline{RESETP}$, MD0, MD1, MD3, MD5, $\overline{TRST}$, MPMD, EXTAL, NMI	V_{IL}	-0.3	—	$V_{CC}Q \times 0.1$	V	
	PTM0, PTM1, PTL4, PTL5, PTL6, PTL7		-0.3	—	$AV_{CC} \times 0.2$		
	Other input pins		-0.3	—	$V_{CC}Q \times 0.2$		

Item	Page	Revision (See Manual for Details)
33.3 DC Characteristics Table 33.4 DC Characteristics (2-b) [I^2C Related Pins*]	1077	Conditions amended Conditions: T_a = –20°C to +75°C (Standard temperature range product), T_a = –40°C to +85°C (Extended temperature range product)
Table 33.5 Permissible Output Current Values	1077	Conditions amended Conditions: T_a = –20°C to +75°C (Standard temperature range product), T_a = –40°C to +85°C (Extended temperature range product)
33.4 AC Characteristics Table 33.6 Maximum Operating Frequencies	1078	Conditions deleted
33.4.1 Clock Timing Table 33.7 Clock Timing	1078	Conditions deleted
33.4.2 Control Signal Timing Table 33.8 Control Signal Timing	1080	Conditions deleted
33.4.3 AC Bus Timing Table 33.9 Bus Timing	1083	Conditions deleted Table amended

Item	Symbol	Min.	Max.	Unit	Figure
Address delay time 1	t_{AD1}	1	10	ns	33.8 to 33.38
Address delay time 2	t_{AD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10	ns	33.15
$\overline{BS}$ delay time	t_{BSD}		10	ns	33.8 to 33.34
$\overline{CS}$ delay time 1	t_{CSD1}	1	10	ns	33.8 to 33.38
Read/write delay time 1	t_{RWD1}	1	10	ns	33.8 to 33.38
Read strobe delay time	t_{RSD}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10	ns	33.8 to 33.15, 33.35, 33.36
Read data setup time 1	t_{RDS1}	$1/2t_{cyc}$ + 7		ns	33.8 to 33.14, 33.33 to 33.38
Read data setup time3	t_{RDS3}	$1/2t_{cyc}$ + 7		ns	33.15
Write enable delay time 1	t_{WED1}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 10	ns	33.8 to 33.13, 33.37, 33.38
Write enable delay time 2	t_{WED2}		10	ns	33.14
Write data delay time 1	t_{WDD1}		10	ns	33.8 to 33.14, 33.35 to 33.38
Write data delay time 2	t_{WDD2}		10	ns	33.20 to 33.23, 33.27 to 33.29, 33.33, 33.34

Item	Page	Revision (See Manual for Details)
33.4.3 AC Bus Timing Table 33.9 Bus Timing	1084	Table amended

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{RAS}$ delay time 1	t_{RASD1}	1	10	ns	33.16 to 33.34
$\overline{CAS}$ delay time 1	t_{CASD1}	1	10	ns	33.16 to 33.34
DQM delay time 1	t_{DQMD1}	1	10	ns	33.16 to 33.34
CKE delay time 1	t_{CKED1}	1	10	ns	33.31 to 33.34
DACK delay time	t_{DACD}	—	13	ns	33.8 to 33.33
$\overline{ICIORD}$ delay time	t_{ICRSD}	—	$1/2t_{cyc}$ + 10	ns	33.37, 33.38
$\overline{ICIOWR}$ delay time	t_{ICWSD}	—	$1/2t_{cyc}$ + 10	ns	33.37, 33.38
$\overline{REFOUT}$, $\overline{IRQOUT}$ delay time	t_{REFOD}	—	$1/2t_{cyc}$ + 10	ns	33.39

33.4.8 Peripheral Module Signal Timing Table 33.10 Peripheral Module Signal Timing	1116	Conditions deleted
33.4.9 16-Bit Timer Pulse Unit (TPU) Table 33.11 16-Bit Timer Pulse Unit	1117	Conditions deleted
33.4.10 RTC Signal Timing Table 33.12 RTC Signal Timing	1118	Conditions deleted
33.4.11 I^2C Bus Interface Timing Table 33.13 I^2C Bus Interface Timing	1118	Conditions deleted
33.4.12 SIOF Module Signal Timing Table 33.14 SIOF Module Signal Timing	1120	Conditions deleted
33.4.14 SIM Module Signal Timing Table 33.17 SIM Module Signal Timing	1125	Conditions deleted
33.4.15 H-UDI Related Pin Timing Table 33.18 H-UDI Related Pin Timing	1126	Conditions deleted

Item	Page	Revision (See Manual for Details)
33.5 A/D Converter Characteristics Table 33.19 A/D Converter Characteristics	1128	Conditions deleted
33.6 D/A Converter Characteristics Table 33.20 D/A Converter Characteristics	1128	Conditions deleted
33.7 AC Characteristic Test Conditions	1129	Description added Unless otherwise specified, the AC characteristic test conditions are as follows. (1) Temperature • Standard temperature range product: Ta = –20 to 75°C • Extended temperature range product:Ta = –40 to 85°C (2) Power Supply Voltage • $V_{CC}Q$ = 3.0 to 3.6 V • V_{CC} = V_{CC}_PLL1 = V_{CC}_PLL2 = 1.1 to 1.3 V • AV_{CC} = 3.0 to 3.6 V (3) Input Signals • I/O signal reference level:

Index

G

H

I

L

M

N

O

P

R

S

T

U

V

W

Renesas 32-Bit RISC Microcomputer
SH7730 Group
User's Manual: Hardware

Publication Date: Rev.1.00, September 19, 2007
Rev.4.00, March 19, 2012
Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.1

SH7730 Group
User's Manual: Hardware

Renesas Electronics Corporation

R01UH0346EJ0400
(Previous Number: REJ09B0359-0300)