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.

SH7720 Group, SH7721 Group

User's Manual: Hardware

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family/SH7700 Series

SH7720 Group	HD6417720
	HD6417320
SH7721 Group	R8A77210
	R8A77211

Renesas Electronics
www.renesas.com

Rev.4.00 2010.09

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.

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 SH7720 or SH7721 Group RISC (Reduced Instruction Set Computer) microcomputer includes 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 LSI in the design of application systems. Users of this manual 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 this LSI to the above users.
Refer to the SH-3/SH-3E/SH3-DSP 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
SH7720 Group	HD6417720, HD6417320
SH7721 Group	R8A77210, R8A77211

- 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-3/SH-3E/SH3-DSP 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'xx, 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/

SH7720 or SH7721 Group manuals:

Document Title	Document No.
SH7720/SH7721 Group Hardware Manual	This manual
SH-3/SH-3E/SH3-DSP Software Manual	REJ09B0317

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

ADC	Analog to Digital Converter
ALU	Arithmetic Logic Unit
ASE	Adaptive System Evaluator
ASID	Address Space Identifier
AUD	Advanced User Debugger
BCD	Binary Coded Decimal
bps	bit per second
BSC	Bus State Controller
CCN	Cache memory Controller
CMT	Compare Match Timer
CPG	Clock Pulse Generator
CPU	Central Processing Unit
DES	Data Encryption Standard
DMAC	Direct Memory Access Controller
etu	Elementary Time Unit
FIFO	First-In First-Out
Hi-Z	High Impedance
H-UDI	User Debugging Interface
INTC	Interrupt Controller
IrDA	Infrared Data Association
JTAG	Joint Test Action Group
LQFP	Low Profile QFP
LRU	Least Recently Used
LSB	Least Significant Bit
MMU	Memory Management Unit
MPX	Multiplex
MSB	Most Significant Bit
PC	Program Counter
PFC	Pin Function Controller
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer

ROM	Read Only Memory
RSA	Rivest Shamir Adleman
RTC	Real Time Clock
SCIF	Serial Communication Interface with FIFO
SDHI	SD Host Interface
SDRAM	Synchronous DRAM
SSL	Secure Socket Layer
TAP	Test Access Port
T.B.D	To Be Determined
TLB	Translation Lookaside Buffer
TMU	Timer Unit
TPU	Timer Pulse Unit
UART	Universal Asynchronous Receiver/Transmitter
UBC	User Break Controller
USB	Universal Serial Bus
WDT	Watchdog Timer

Contents

Figures

Section 1 Overview

Section 2 CPU

Section 3 DSP Operating Unit

Section 10 Direct Memory Access Controller (DMAC)

Section 11 Clock Pulse Generator (CPG)

Section 12 Watchdog Timer (WDT)

Section 13 Power-Down Modes

Section 14 Timer Unit (TMU)

Section 15 16-Bit Timer Pulse Unit (TPU)

Section 16 Compare Match Timer (CMT)

Section 17 Realtime Clock (RTC)

Section 18 Serial Communication Interface with FIFO (SCIF)

Section 19 Infrared Data Association Module (IrDA)

Section 20 I^2C Bus Interface (IIC)

Section 21 Serial I/O with FIFO (SIOF)

Section 22 Analog Front End Interface (AFEIF)

Section 23 USB Pin Multiplex Controller

Section 25 USB Function Controller (USBF)

Section 26 LCD Controller (LCDC)

Section 27 A/D Converter

Section 28 D/A Converter (DAC)

Section 29 PC Card Controller (PCC)

Section 30 SIM Card Module (SIM)

Section 31 MultiMediaCard Interface (MMCIF)

Section 33 User Break Controller (UBC)

Section 35 I/O Ports

Section 36 User Debugging Interface (H-UDI)

Section 38 Electrical Characteristics

Appendix

Tables

Section 10 Direct Memory Access Controller (DMAC)

Section 11 Clock Pulse Generator (CPG)

Section 13 Power-Down Modes

Section 34 Pin Function Controller (PFC)

Section 35 I/O Ports

Section 36 User Debugging Interface (H-UDI)

Section 38 Electrical Characteristics

Appendix

Section 1 Overview

1.1 Features

This LSI is a single-chip RISC microprocessor that integrates a 32-bit RISC-type Super H architecture CPU with a digital signal processing (DSP) extension as its core, together with a large-capacity 32-kbyte cache memory, a 16-kbyte X/Y memory, and an interrupt controller.

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. This LSI also supports a stereo audio recording and playback function, a USB host controller, a function controller, an LCD controller, a PCMCIA interface, an A/D converter, and a D/A converter.

The USB host controller and LCD controller have bus master functions, so that data supplied from an external memory (area 3) can be freely processed. Since the USB host controller, in particular, conforms to Open HCI standards, it is extremely easy to transfer data from the PC of a device driver or other devices. Also, low-power operation suitable for battery operation is possible because the LCD controller continues to display even in sleep mode.

A powerful built-in power management function keeps power consumption low, even during high-speed operation. This LSI is ideal for electronics devices, which require both high speed and low power consumption.

The SH7720 group integrates an SSL (Secure Socket Layer) accelerator that performs RSA (Rivest-Shamir-Adleman) operations and DES (Data Encryption Standard) and Triple-DES encryption/decryption, while the SH7721 group does not have the SSL accelerator. Each group consists of several models which includes or does not include an SD host interface (SDHI) to be suited to a variety of applications. See table 1.2 and 1.3, Product Lineup, for the models including (or not including) the SDHI.

Note: For the detailed specifications of the SDHI and SSL, contact the Renesas representatives in your region.

Table 1.1 shows the features of this LSI.

Table 1.1 SH7720/SH7721 Features

Item	Features
CPU	• Renesas Original SuperH architecture • Upper compatibility with SH-1, SH-2, and SH3-DSP at object code level • 32-bit internal data bus • General-register — Sixteen 32-bit general registers (eight 32-bit shadow registers) — Five 32-bit control registers — Four 32-bit system registers • RISC type instruction set — Instruction length: 16-bit fixed length for improved code efficiency — Load/store architecture — Delayed branch instruction — Instruction set based on C language • Instruction execution time: One instruction/cycle for basic instructions • Logical address space: 4 Gbytes • Space identifier ASID: 8 bits, 256 logical address spaces • Five-stage pipeline
DSP operating unit	• Mixture of 16-bit and 32-bit instructions • 32-/40-bit internal data bus • Multiplier, ALU, barrel shifter, and DSP register • 16-bit x 16-bit → 32-bit one cycle multiplier • Large-capacity DSP data register file — Six 32-bit data registers — Two 40-bit data registers • Extended Harvard architecture for DSP data buses — Two data buses — One instruction bus • Up to four parallel operations: ALU, multiply, two loads, and store • Two address units to generating addresses for two memory access • DSP data addressing modes: Increment, index register addition (with or without modulo addressing) • Zero-overhead repeat loop control • Conditional execution instructions • User DSP mode and privileged DSP mode

Item	Features
Memory management unit (MMU)	• 4-Gbyte address space, 256 address spaces (8-bit ASID) • Page unit sharing • Supports multiple page sizes: 1 kbyte or 4 kbytes • 128-entry, 4-way set associative TLB • Specifies replacement way by software and supports random replacement algorithm • Address assignment allows direct access to TLB contents
Cache memory	• 32-kbyte cache mixing instructions and data • 512-entry, 4-way set associative, 16-byte block length • Write-back, write-through, least recent used (LRU) replacement algorithm • Single-stage write-back buffer
X/Y memory	• User-selectable mapping mechanism — Fixed mapping for mission-critical realtime applications — Automatic mapping through TLB for easy to use • Three independent read/write ports — 8-/16-/32-bit access from CPU — Up to two 16-bit accesses from DSP — 8-/16-/32-bit access from DMAC • 8-kbyte RAM for X and Y memory individual (4 kbytes × 4)
Interrupt controller (INTC)	• Seven external interrupt pins (NMI, IRQ5 to IRQ0) — NMI: Fall/rise selectable — IRQ: Fall/rise/high level/low level selectable • On-chip peripheral interrupt: Sets priority for each module
Bus state controller (BSC)	• Physical address space is provided to support areas of up to 64 Mbytes and 32 Mbytes. • Each area allows independent setting of the following functions: — Bus size (8, 16, or 32 bits). An access wait cycle count with a different size to be supported is provided for each area. — Number of access wait cycles. Some areas can be inserted wait cycles independently in read access and write access. — Sets of idle wait cycle (for the same or different area) — Supports SRAM, page mode ROM, SDRAM, and pseudo SRAM (ready for page mode) by specifying memory to be connected to each area. — Outputs chip select signals to corresponding areas, such as $\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$, $\overline{CS5A}$/$\overline{CS5B}$, and $\overline{CS6A}$/$\overline{CS6B}$

Item	Features
Direct memory access controller (DMAC)	• Number of channels: Six channels (two channels support external requests) • Address space: 4 Gbytes on architecture • Data transfer length: Bytes, words (2 bytes), longwords (4 bytes), 16 bytes (longword × 4) • Maximum number of transfer times: 16,777,216 times • Address mode: Single address mode or dual address mode selectable • Transfer request: Selectable from external request, on-chip peripheral module request, and auto request • Bus mode: Selectable from cycle steal mode (normal mode and intermittent mode) and burst mode • Priority: Selectable from channel priority fixed mode and round robin mode • Interrupt request: Supports interrupt request to CPU at the end of data transfer • External request detection: Selectable from DREQ input low/high level detection and rising/falling detection • Transfer request acceptance signal: DACK and TEND can be set an active level
Clock pulse generator (CPG)	• Clock mode: Input clock selectable from external clock (EXTAL or CKIO) and crystal resonator • Generates three types of clocks — CPU clock: Maximum 133.34 MHz — Bus clock: Maximum 66.67 MHz — Peripheral clock: Maximum 33.34 MHz • Supports power-down mode — Sleep mode — Standby mode — Module standby mode (X/Y memory standby enabled) • One-channel watchdog timer
Watchdog timer (WDT)	• One-channel watchdog timer (WDT) • Interrupt request: WDT only
Timer unit (TMU)	• Internal three-channel 32-bit timer • Auto-reload type 32-bit down counter • Internal prescaler for Pϕ • Interrupt request

Item	Features
16-bit timer pulse unit (TPU)	• Four-channel 16-bit timer • PWM mode • Four types of counter input clocks • Phase counting mode (two channels)
Compare match timer (CMT)	• Internal six-channel 32-bit counter (16-/32-bit switchable) • Selectable prescaling for Pϕ • Internal full-channel compare match function • With interrupt request and DMAC request
Realtime clock (RTC)	• Built-in clock, calendar functions, and alarm functions • On-chip 32-kHz crystal oscillator circuit with a maximum resolution (cycle interrupt) of 1/256 second
Serial communication interface with FIFO (SCIF0, SCIF1)	• Includes a 64-byte FIFO for transmission and another for reception • Supports high-speed UART for Bluetooth • Internal prescaler for Pϕ • With interrupt request and DMAC request
Infrared data association module (IrDA)	• Conforms to the IrDA 1.0 system • Asynchronous serial communication • On-chip 64-stage FIFO buffers for transmission and reception
I^2C bus interface (IIC)	• Supports multi master transmission/reception
Serial I/O with FIFO (SIOF0, SIOF1)	• Includes a 64-byte FIFO for transmission and another for reception • Supports 8-/16-/16-bit stereo sound input/output • Sampling rate clock input selectable from Pϕ and external pin • Includes a prescaler for Pϕ • Interrupt requests and DMAC requests
Analog front end interface (AFEIF)	• STLC7550 can directly be connected • Data access arrangement function • 128-word transmit FIFO • 128-word receive FIFO

Item	Features
USB host controller (USBH)	• Conforms to OHCI Rev. 1.0 • USB Rev. 1.1 compatible • 127 endpoints • Support interrupt/bulk/control/isochronous mode • Bus master controller (can access area 3 and synchronous DRAM) • Two ports with analog transceiver (one is common with USB function controller) • External clock input function
USB function controller (USBF)	• Conforms to OHCI Rev. 1.0 • Six endpoints • Support interrupt/bulk/control/isochronous mode • One port with analog transceiver (common with USB function controller), 12 Mbps only • External clock input function
LCD controller (LCDC)	• From 16 × 1 to 1024 × 1024 pixels can be supported • 4/8/15/16 bpp (bit per pixel) color pallet • 1/2/4/6 bpp (bit per pixel) gray scale • 8-bit frame rate controller • TFT/DSTN/STN panels • Signal polarity setting function • Hardware panel rotation • Power control function • Selectable clock source (LCLK, Bclk, or Pclk)
A/D converter (ADC)	• 10 bits ± 4 LSB, four channels • Conversion time: 15 μs • Input range: 0 to AV_{CC} (max. 3.6 V)
D/A converter (DAC)	• 8 bits ± 4 LSB, two channels • Conversion time: 10 μs • Output range: 0 to AV_{CC} (max. 3.6 V)
PC card controller (PCC)	• Complies with the PCMCIA Rev.2.1/JEIDA Version 4.2 • Supports the IC memory card interface and I/O card interface

Item	Features
SIM card interface (SIM)	• Single channel ready for ISO7816-3 data protocol (T = 0, T = 1) • Asynchronous half-duplex character transmission protocol • Data length of 8 bits • Generates and checks a parity bit • Number of output clocks per 1 etu selectable • Direct convention/inverse convention selectable • Internal prescaler for Pϕ • Clock polarity changeable at idle time (low or high) • With interrupt request and DMAC request
MultiMedia Card interface (MMCIF)	• Complies with The MultiMedia Card System Specification Version 3.1 • Supports MMC mode • 16.5-Mbps bit rate (max) for the card interface (Pϕ = 33 MHz) • Incorporates sixty-four 16-bit data-transfer FIFOs • Interrupt and DMA request • Module standby function
SD host interface (SDHI) Note: Only for models with the SDHI	• Supports SDHC (SD High Capacity) and SDIO — Supports Part 1 Physical Layer Ver.1.01 to 2.0 of SD Specification, but not supported for High-Speed — Supports Part E1 SDIO Ver. 1.00 to 2.00 of SD Specification • SD memory/IO card interface (1 bit/4 bits SD bus) • SD clock frequency ≤ 1/2 peripheral clock frequency • Error check function: CRC7 (command/response), CRC16 (data) • MMC (MultiMedia Card) access • Interrupt request and DAMC transfer request (SD_BUF read/write) • Card detection function • Write protect
SSL accelerator (SSL) Note: SH7720 group only	• RSA encryption • Supported operations: addition, subtraction, multiplication, power operation • DES and Triple-DES encryption/decryption

Item	Features
User break controller (UBC)	• 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 • 1-kbyte on-chip memory for executing high-speed emulation program

Table 1.2 Product Lineup (SH7720 Group)

Model	Power Supply Voltage I/O	Power Supply Voltage Internal	Operating Frequency	Product Code	Package	SSL	SDHI
SH7720	3.3 V ±0.3V	1.5 V ±0.1V	133.34 MHz	HD6417720BP133C	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	O	—
				HD6417720BP133CV	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	O	—
				HD6417720BL133C	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	O	—
				HD6417720BL133CV	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	O	—
SH7320				HD6417320BP133C	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	O	O
				HD6417320BP133CV	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	O	O
				HD6417320BL133C	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	O	O
				HD6417320BL133CV	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	O	O

[Legend] O: Provided; —: Not provided

Table 1.3 Product Lineup (SH7721 Group)

Model	Power Supply Voltage I/O	Power Supply Voltage Internal	Operating Frequency	Product Code	Package	SSL	SDHI
SH7721	3.3 V ±0.3V	1.5 V ±0.1V	133.34 MHz	R8A77210C133BG	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	—	—
				R8A77210C133BGV	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	—	—
				R8A77210C133BA	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	—	—
				R8A77210C133BAV	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	—	—
				R8A77211C133BG	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	—	O
				R8A77211C133BGV	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	—	O
				R8A77211C133BA	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	—	O
				R8A77211C133BAV	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	—	O

[Legend] O: Provided; —: Not provided

1.2 Block Diagram

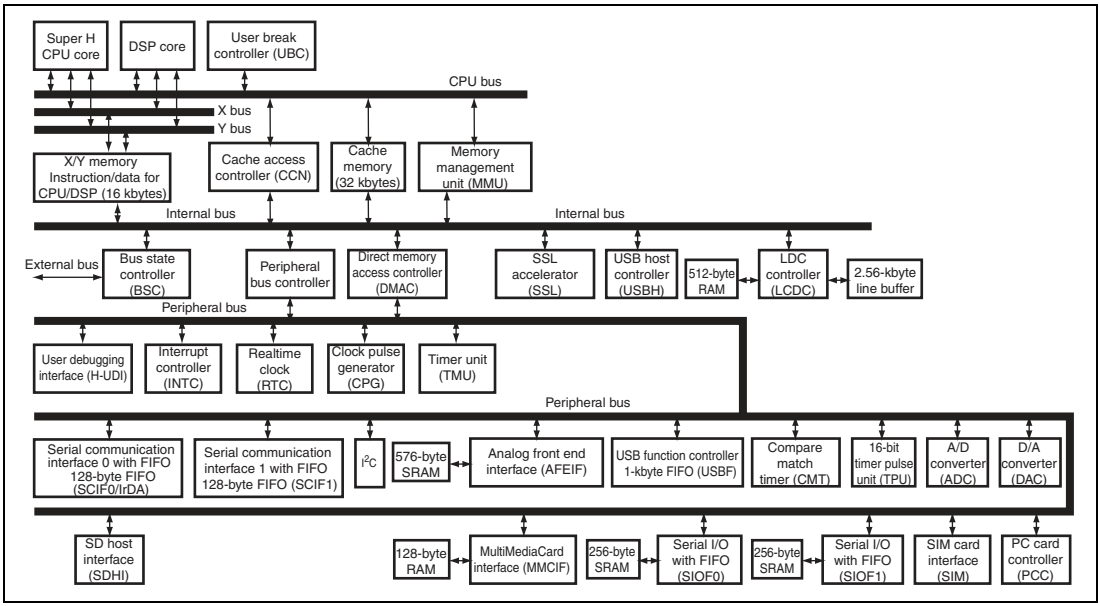


Figure 1.1 Block Diagram

1.3 Pin Assignments

1.3.1 Pin Assignments

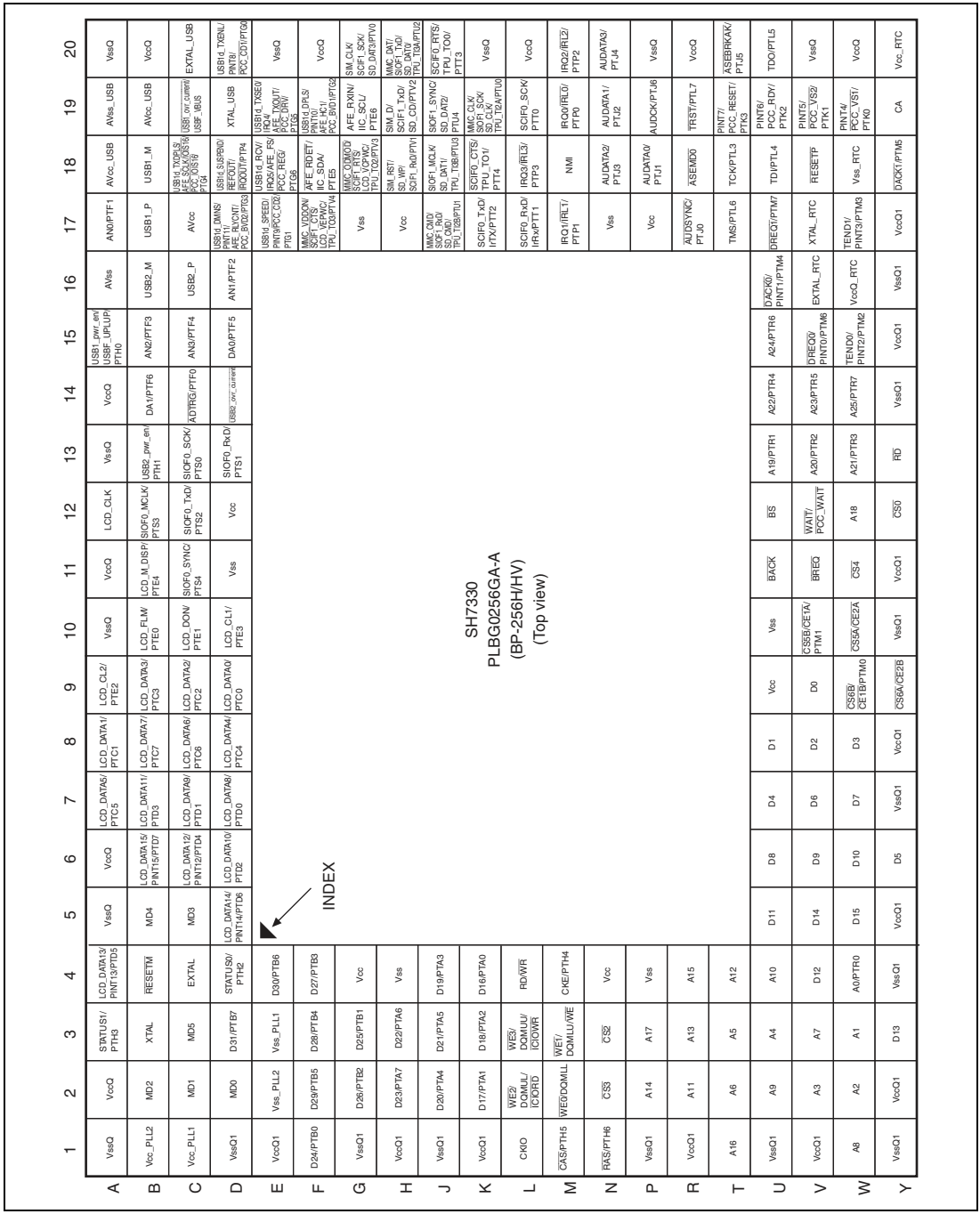


Figure 1.2 Pin Assignments (PLBG0256GA-A (BP-256H/HV))

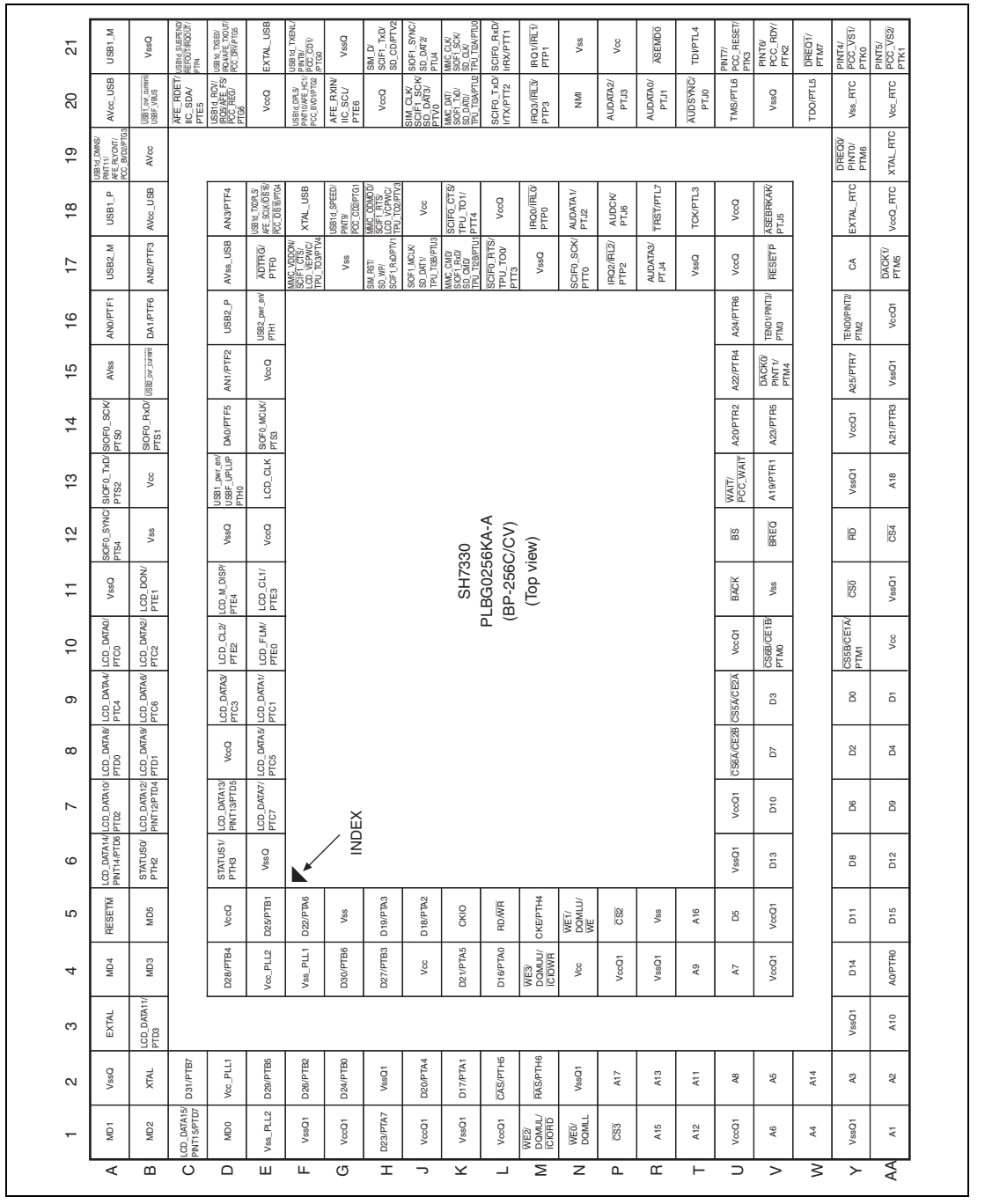


Figure 1.3 Pin Assignments (PLBG0256KA-A (BP-256C/CV))

Table 1.4 List of Pin Assignments

Pin No. (PLBG 0256 GA-A)	Pin No. (PLBG 0256 KA-A)	Pin Name	Function	I/O	I/O Buffer Power Supply
A1	A2	VssQ	I/O power supply (0V)		—
A2	D5	VccQ	I/O power supply (3.3 V)		—
A3	D6	STATUS1/PTH3	Status output/general-purpose port	O/IO	VccQ
A4	D7	LCD_DATA13/PINT13/ PTD5	LCD data/port interrupt/ general-purpose port	O/I/IO	VccQ
A5	E6	VssQ	I/O power supply (0V)		—
A6	D8	VccQ	I/O power supply (3.3 V)		—
A7	E8	LCD_DATA5/PTC5	LCD data/general-purpose port	O/IO	VccQ
A8	E9	LCD_DATA1/PTC1	LCD data/general-purpose port	O/IO	VccQ
A9	D10	LCD_CL2/PTE2	LCD shift clock 2/general-purpose port	O/IO	VccQ
A10	A11	VssQ	I/O power supply (0V)		—
A11	E12	VccQ	I/O power supply (3.3 V)		—
A12	E13	LCD_CLK	LCD clock source	I	VccQ
A13	D12	VssQ	I/O power supply (0V)		—
A14	E15	VccQ	I/O power supply (3.3 V)		—
A15	D13	USB1_pwr_en/ USBF_UPLUP/PTH0	USB1 power-enable/pull-up control/general-purpose port	O/O/IO	VccQ
A16	A15	AVss	Analog power supply (0V)		—
A17	A16	AN0/PTF1	ADC analog input/general-purpose port	I/I	AVcc
A18	B18	AVcc_USB	USB power supply (3.3 V)		—
A19	D17	AVss_USB	USB power supply (0 V)		—
A20	B21	VssQ	I/O power supply (0V)		—
B1	E4	Vcc_PLL2	PLL2 power supply (1.5 V)		—
B2	B1	MD2	Clock mode setting	I	VccQ
B3	B2	XTAL	Crystal	O	VccQ
B4	A5	$\overline{\text{RESETM}}$	Manual reset	I	VccQ
B5	A4	MD4	Bus width setting	I	VccQ

Pin No. (PLBG 0256 GA-A)	Pin No. (PLBG 0256 KA-A)	Pin Name	Function	I/O	I/O Buffer Power Supply
B6	C1	LCD_DATA15/PINT15/ PTD7	LCD data/port interrupt/ general-purpose port	O/I/IO	VccQ
B7	B3	LCD_DATA11/PTD3	LCD data/general-purpose port	O/IO	VccQ
B8	E7	LCD_DATA7/PTC7	LCD data/general-purpose port	O/IO	VccQ
B9	D9	LCD_DATA3/PTC3	LCD data/general-purpose port	O/IO	VccQ
B10	E10	LCD_FLM/PTE0	LCD line marker/general-purpose port	O/IO	VccQ
B11	D11	LCD_M_DISP/PTE4	LCD current-alternating signal/ general-purpose port	O/IO	VccQ
B12	E14	SIOF0_MCLK/PTS3	SIOF master clock/general-purpose port	I/IO	VccQ
B13	E16	USB2_pwr_en/PTH1	USB2 power-enable/ general-purpose port	O/IO	VccQ
B14	B16	DA1/PTF6	DAC analog output/general-purpose port	O/I	VccQ
B15	B17	AN2/PTF3	ADC analog input/general-purpose port	I/I	AVcc
B16	A17	USB2_M	USB D− port 2	IO	AVcc_ USB
B17	A18	USB1_P	USB D+ port 1	IO	AVcc_ USB
B18	A21	USB1_M	USB D− port 1	IO	AVcc_ USB
B19	A20	AVcc_USB	USB power supply (3.3 V)		—
B20	E20	VccQ	I/O power supply (3.3 V)		—
C1	D2	Vcc_PLL1	PLL1 power supply (1.5 V)		—
C2	A1	MD1	Clock mode setting	I	VccQ
C3	B5	MD5	Endian setting	I	VccQ
C4	A3	EXTAL	External clock	I	VccQ
C5	B4	MD3	Bus width setting	I	VccQ
C6	B7	LCD_DATA12/PINT12/ PTD4	LCD data/port interrupt/ general-purpose port	O/I/IO	VccQ
C7	B8	LCD_DATA9/PTD1	LCD data/general-purpose port	O/IO	VccQ

Pin No. (PLBG 0256 GA-A)	Pin No. (PLBG 0256 KA-A)	Pin Name	Function	I/O	I/O Buffer Power Supply
C8	B9	LCD_DATA6/PTC6	LCD data/general-purpose port	O/IO	VccQ
C9	B10	LCD_DATA2/PTC2	LCD data/general-purpose port	O/IO	VccQ
C10	B11	LCD_DON/PTE1	LCD display on signal/ general-purpose port	O/IO	VccQ
C11	A12	SIOF0_SYNC/PTS4	SIOF frame sync/general-purpose port	IO/IO	VccQ
C12	A13	SIOF0_TxD/PTS2	SIOF transmit data/general-purpose port	O/IO	VccQ
C13	A14	SIOF0_SCK/PTS0	SIOF serial clock/general-purpose port	IO/IO	VccQ
C14	E17	$\overline{\text{ADTRG}}$/PTF0	ADC external trigger/general-purpose port	I/I	VccQ
C15	D18	AN3/PTF4	ADC analog input/general-purpose port	I/I	AVcc
C16	D16	USB2_P	USB D+ port 2	IO	AVcc_ USB
C17	B19	AVcc	Analog power supply (3.3 V)		—
C18	E18	USB1d_TXDPLS/ AFE_SCLK/$\overline{\text{IOIS16}}$/ $\overline{\text{PCC\_IOIS16}}$/PTG4	D+ transmit output/AFE shift clock/16-bit IO/PCCI 6-bit IO/general-purpose port	O/I/I/I/ IO	VccQ
C19	B20	$\overline{\text{USB1\_ovr\_current}}$/ USBF_VBUS	USB1 overcurrent/monitor	I/I	VccQ
C20	E21	EXTAL_USB	USB external clock	I	VccQ
D1	F1	VssQ1	I/O power supply (0 V)		—
D2	D1	MD0	Clock mode setting	I	VccQ
D3	C2	D31/PTB7	Data bus/general-purpose port	IO/IO	VccQ1
D4	B6	STATUS0/PTH2	Status output/general-purpose port	O/IO	VccQ
D5	A6	LCD_DATA14/PINT14/ PTD6	LCD data/port interrupt/ general-purpose port	O/I/IO	VccQ
D6	A7	LCD_DATA10/PTD2	LCD data/general-purpose port	O/IO	VccQ
D7	A8	LCD_DATA8/PTD0	LCD data/general-purpose port	O/IO	VccQ
D8	A9	LCD_DATA4/PTC4	LCD data/general-purpose port	O/IO	VccQ

Pin No. (PLBG 0256 GA-A)	Pin No. (PLBG 0256 KA-A)	Pin Name	Function	I/O	I/O Buffer Power Supply
D9	A10	LCD_DATA0/PTC0	LCD data/general-purpose port	O/IO	VccQ
D10	E11	LCD_CL1/PTE3	LCD shift clock 1/general-purpose port	O/IO	VccQ
D11	B12	Vss	Internal power supply (0 V)		—
D12	B13	Vcc	Internal power supply (1.5 V)		—
D13	B14	SIOF0_RxD/PTS1	SIOF receive data/general-purpose port	I/IO	VccQ
D14	B15	$\overline{\text{USB2\_ovr\_current}}$	USB2 port overcurrent	I	VccQ
D15	D14	DA0/PTF5	DAC analog output/general-purpose port	O/I	VccQ
D16	D15	AN1/PTF2	ADC analog input/general-purpose port	I/I	AVcc
D17	A19	USB1d_DMNS/PINT11/ AFE_RLYCNT/ PCC_BVD2/PTG3	D- signal input/port interrupt/ AFE on-hook control/PCC buttery detection 2/general-purpose port	I/I/O/I/ IO	VccQ
D18	C21	USB1d_SUSPEND/ $\overline{\text{REFOUT}}$/$\overline{\text{IRQOUT}}$/ PTP4	Suspend state/bus request (refresh)/ bus request (interrupt)/ general-purpose port	O/O/O/ IO	VccQ
D19	F18	XTAL_USB	USB crystal	O	VccQ
D20	F21	USB1d_TXENL/PINT8/ $\overline{\text{PCC\_CD1}}$/PTG0	Driver output enable/port interrupt/ PCC card detection 1/ general-purpose port	O/I/I/IO	VccQ
E1	G1	VccQ1	I/O power supply (1.8/3.3 V)		—
E2	E1	Vss_PLL2	PLL2 power supply (0 V)		—
E3	F4	Vss_PLL1	PLL1 power supply (0 V)		—
E4	G4	D30/PTB6	Data bus/general-purpose port	IO/IO	VccQ1
E17	G18	USB1d_SPEED/PINT9/ $\overline{\text{PCC\_CD2}}$/PTG1	Speed control/port interrupt/ PCC card detection 2/ general-purpose port	O/I/I/IO	VccQ
E18	D20	USB1d_RCV/IRQ5/ AFE_FS/$\overline{\text{PCC\_REG}}$/ PTG6	Receive data/interrupt/ area indicate signal/ AFE frame synchronization/ PCC space indication/ general-purpose port	I/I/I/O/ IO	VccQ

Pin No. (PLBG 0256 GA-A)	Pin No. (PLBG 0256 KA-A)	Pin Name	Function	I/O	I/O Buffer Power Supply
E19	D21	USB1d_TXSE0/IRQ4/ AFE_TXOUT/ PCC_DRV/PTG5	SE0 state/interrupt/ AFE serial transmission/ PCC buffer control/ general-purpose port	O/I/O/ O/IO	VccQ
E20	G21	VssQ	I/O power supply (0V)		—
F1	G2	D24/PTB0	Data bus/general-purpose port	IO/IO	VccQ1
F2	E2	D29/PTB5	Data bus/general-purpose port	IO/IO	VccQ1
F3	D4	D28/PTB4	Data bus/general-purpose port	IO/IO	VccQ1
F4	H4	D27/PTB3	Data bus/general-purpose port	IO/IO	VccQ1
F17	F17	MMC_VDDON/ SCIF1_CTS/ LCD_VEPWC/ TPU_TO3/PTV4	MMC card power supply control/ SCIF transmit enable/LCD power supply control/ TPU compare-match output/general-purpose port	O/I/O/ O/IO	VccQ
F18	C20	AFE_RDET/IIC_SDA/ PTE5	AFE ringing/IIC data I/O /general-purpose port	I/IO/I	VccQ
F19	F20	USB1d_DPLS/PINT10/ AFE_HC1/PCC_BVD1/ PTG2	D+ transmit input/port interrupt/ AFE hardware control/ PCC battery detection 1/ general-purpose port	I/I/O/I/ IO	VccQ
F20	H20	VccQ	I/O power supply (3.3 V)		—
G1	H2	VssQ1	I/O power supply (0V)		—
G2	F2	D26/PTB2	Data bus/general-purpose port	IO/IO	VccQ1
G3	E5	D25/PTB1	Data bus/general-purpose port	IO/IO	VccQ1
G4	J4	Vcc	Internal power supply (1.5 V)		—
G17	G17	Vss	Internal power supply (0 V)		—
G18	H18	MMC_ODMOD/ SCIF1_RTS/ LCD_VCPWC/TPU_TO2/ PTV3	MMC open drain control/ SCIF transmit request/LCD power supply control/TPU compare-match output/general-purpose port	O/O/O/ O/IO	VccQ
G19	G20	AFE_RXIN/IIC_SCL/ PTE6	AFE serial receive/ IIC clock/general-purpose port	I/IO/I	VccQ
G20	J20	SIM_CLK/ SCIF1_SCK/ SD_DAT3/PTV0	SIM clock/SCIF serial clock/ SD data/general-purpose port	O/IO/ IO/IO	VccQ

Pin No. (PLBG 0256 GA-A)	Pin No. (PLBG 0256 KA-A)	Pin Name	Function	I/O	I/O Buffer Power Supply
H1	J1	VccQ1	I/O power supply (1.8/3.3 V)		—
H2	H1	D23/PTA7	Data bus/general-purpose port	IO/IO	VccQ1
H3	F5	D22/PTA6	Data bus/general-purpose port	IO/IO	VccQ1
H4	G5	Vss	Power-supply (0 V)		—
H17	J18	Vcc	Power-supply (1.5 V)		—
H18	H17	SIM_RST/SCIF1_RxD/ SD_WP/PTV1	SIM reset/SCIF receive data/ SD write protect/ general-purpose port	O/I/I/IO	VccQ
H19	H21	SIM_D/SCIF1_TxD/ SD_CD/PTV2	SIM data/SCIF transmit data/ SD card detection/ general-purpose port	IO/O/I/ IO	VccQ
H20	K20	MMC_DAT/SIOF1_TxD/ SD_DAT0/TPU_TI3A/ PTU2	MMC data/SIOF transmit data/ SD data/TPU clock input/ general-purpose port	IO/O/ IO/I/IO	VccQ
J1	K1	VssQ1	I/O power supply (0V)		—
J2	J2	D20/PTA4	Data bus/general-purpose port	IO/IO	VccQ1
J3	K4	D21/PTA5	Data bus/general-purpose port	IO/IO	VccQ1
J4	H5	D19/PTA3	Data bus/general-purpose port	IO/IO	VccQ1
J17	K17	MMC_CMD/ SIOF1_RxD/SD_CMD/ TPU_TI2B/PTU1	MMC command/SIOF receive data/SD command/TPU clock input/general-purpose port	IO/I/IO/ I/IO	VccQ
J18	J17	SIOF1_MCLK/SD_DAT1/ TPU_TI3B/PTU3	SIOF master clock/SD data/ TPU clock input/general-purpose port	I/IO/I/IO	VccQ
J19	J21	SIOF1_SYNC/SD_DAT2/ PTU4	SIOF frame sync/ SD data/general-purpose port	IO/IO/IO	VccQ
J20	L17	$\overline{\text{SCIF0\_RTS}}$/TPU_TO0/ PTT3	SCIF transmit request/TPU compare-match output/ general-purpose port	O/O/IO	VccQ
K1	L1	VccQ1	I/O power supply (1.8/3.3 V)		—
K2	K2	D17/PTA1	Data bus/general-purpose port	IO/IO	VccQ1

Pin No. (PLBG 0256 GA-A)	Pin No. (PLBG 0256 KA-A)	Pin Name	Function	I/O	I/O Buffer Power Supply
K3	J5	D18/PTA2	Data bus/general-purpose port	IO/IO	VccQ1
K4	L4	D16/PTA0	Data bus/general-purpose port	IO/IO	VccQ1
K17	L20	SCIF0_TxD/IrTx/PTT2	SCIF transmit data/ IrDA transmit data/general-purpose port	O/O/IO	VccQ
K18	K18	$\overline{\text{SCIF0\_CTS}}$/TPU_TO1/ PTT4	SCIF transmit enable/TPU compare-match output/ general-purpose port	I/O/IO	VccQ
K19	K21	MMC_CLK/SIOF1_SCK/ SD_CLK/TPU_TI2A/ PTU0	MMC clock/SIOF serial clock/ SD clock/TPU clock input/general-purpose port	O/IO/O/ I/IO	VccQ
K20	M17	VssQ	I/O power supply (0V)		—
L1	K5	CKIO	System clock	IO	VccQ1
L2	M1	$\overline{\text{WE2}}$/DQMUL/$\overline{\text{ICIORD}}$	Second-highest-byte write/ DQ mask UL/IO read	O/O/O	VccQ1
L3	M4	$\overline{\text{WE3}}$/DQMUU/$\overline{\text{ICIOWR}}$	Highest-byte write/ DQ mask UU/IO write	O/O/O	VccQ1
L4	L5	RD/$\overline{\text{WR}}$	Read/write signal	O	VccQ1
L17	L21	SCIF0_RxD/IrRx/PTT1	SCIF receive data/IrDA receive data/general-purpose port	I/I/IO	VccQ
L18	M20	IRQ3/$\overline{\text{IRL3}}$/PTP3	Interrupt/interrupt/general-purpose port	I/I/IO	VccQ
L19	N17	SCIF0_SCK/PTT0	SCIF serial clock/general-purpose port	IO/IO	VccQ
L20	L18	VccQ	I/O power supply (3.3 V)		—
M1	L2	$\overline{\text{CAS}}$/PTH5	Column address/general-purpose port	O/IO	VccQ1
M2	N1	$\overline{\text{WE0}}$/DQMLL	Lowest-byte write/DQ mask LL	O/O	VccQ1
M3	N5	$\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$	Second-lowest-byte write/ DQ mask LU/write enable	O/O/O	VccQ1
M4	M5	CKE/PTH4	Clock enable/general-purpose port	O/IO	VccQ1
M17	M21	IRQ1/$\overline{\text{IRL1}}$/PTP1	Interrupt/interrupt/ general-purpose port	I/I/IO	VccQ
M18	N20	NMI	NMI interrupt	I	VccQ

Pin No. (PLBG 0256 GA-A)	Pin No. (PLBG 0256 KA-A)	Pin Name	Function	I/O	I/O Buffer Power Supply
M19	M18	IRQ0/$\overline{\text{IRL0}}$/PTP0	Interrupt/interrupt/ general-purpose port	I/I/IO	VccQ
M20	P17	IRQ2/$\overline{\text{IRL2}}$/PTP2	Interrupt/interrupt/ general-purpose port	I/I/IO	VccQ
N1	M2	$\overline{\text{RAS}}$/PTH6	Row address/general-purpose port	O/IO	VccQ1
N2	P1	$\overline{\text{CS3}}$	Chip select	O	VccQ1
N3	P5	$\overline{\text{CS2}}$	Chip select	O	VccQ1
N4	N4	Vcc	Power-supply (1.5 V)		—
N17	N21	Vss	Power-supply (0 V)		—
N18	P20	AUDATA2/PTJ3	AUD data/general-purpose port	O/IO	VccQ
N19	N18	AUDATA1/PTJ2	AUD data/general-purpose port	O/IO	VccQ
N20	R17	AUDATA3/PTJ4	AUD data/general-purpose port	O/IO	VccQ
P1	N2	VssQ1	I/O power supply (0V)		—
P2	W2	A14	Address bus	O	VccQ1
P3	P2	A17	Address bus	O	VccQ1
P4	R5	Vss	Internal power supply (0 V)		—
P17	P21	Vcc	Internal power supply (1.5 V)	—	—
P18	R20	AUDATA0/PTJ1	AUD data/general-purpose port	O/IO	VccQ
P19	P18	AUDCK/PTJ6	AUD clock/general-purpose port	O/IO	VccQ
P20	T17	VssQ	I/O power supply (0V)		—
R1	P4	VccQ1	I/O power supply (1.8/3.3 V)		—
R2	T2	A11	Address bus	O	VccQ1
R3	R2	A13	Address bus	O	VccQ1
R4	R1	A15	Address bus	O	VccQ1
R17	T20	$\overline{\text{AUDSYNC}}$/PTJ0	AUD synchronous signal/ general-purpose port	O/IO	VccQ
R18	R21	$\overline{\text{ASEMD0}}$	ASE mode	I	VccQ
R19	R18	$\overline{\text{TRST}}$/PTL7	Test reset/general-purpose port	I/IO	VccQ
R20	U17	VccQ	I/O power supply (3.3 V)		—
T1	T5	A16	Address bus	O	VccQ1

Pin No. (PLBG 0256 GA-A)	Pin No. (PLBG 0256 KA-A)	Pin Name	Function	I/O	I/O Buffer Power Supply
T2	V1	A6	Address bus	O	VccQ1
T3	V2	A5	Address bus	O	VccQ1
T4	T1	A12	Address bus	O	VccQ1
T17	U20	TMS/PTL6	Test mode select/general-purpose port	I/IO	VccQ
T18	T18	TCK/PTL3	Test clock/general-purpose port	I/IO	VccQ
T19	U21	PINT7/PCC_RESET/ PTK3	Port interrupt/PCC reset/general-purpose port	I/O/IO	VccQ
T20	V18	$\overline{\text{ASEBRKAK}}$/PTJ5	ASE break mode acknowledge/ general-purpose port	O/IO	VccQ
U1	R4	VssQ1	I/O power supply (0 V)		—
U2	T4	A9	Address bus	O	VccQ1
U3	W1	A4	Address bus	O	VccQ1
U4	AA3	A10	Address bus	O	VccQ1
U5	Y5	D11	Data bus	IO	VccQ1
U6	Y6	D8	Data bus	IO	VccQ1
U7	AA8	D4	Data bus	IO	VccQ1
U8	AA9	D1	Data bus	IO	VccQ1
U9	AA10	Vcc	Internal power supply (1.5 V)		—
U10	V11	Vss	Internal power supply (0 V)		—
U11	U11	$\overline{\text{BACK}}$	Bus request acknowledge	O	VccQ1
U12	U12	$\overline{\text{BS}}$	Bus start	O	VccQ1
U13	V13	A19/PTR1	Address bus/general-purpose port	O/IO	VccQ1
U14	U15	A22/PTR4	Address bus/general-purpose port	O/IO	VccQ1
U15	U16	A24/PTR6	Address bus/general-purpose port	O/IO	VccQ1
U16	V15	$\overline{\text{DACK0}}$/PINT1/PTM4	DMA transfer request reception/ port interrupt/ general-purpose port	O/I/IO	VccQ1
U17	W21	$\overline{\text{DREQ1}}$/PTM7	DMA transfer request/ general-purpose port	I/IO	VccQ1
U18	T21	TDI/PTL4	Test data input/general-purpose port	I/IO	VccQ

Pin No. (PLBG 0256 GA-A)	Pin No. (PLBG 0256 KA-A)	Pin Name	Function	I/O	I/O Buffer Power Supply
U19	V21	PINT6/PCC_RDY/PTK2	Port interrupt/PCC ready/general-purpose port	I/I/IO	VccQ
U20	W20	TDO/PTL5	Test data output/general-purpose port	O/IO	VccQ
V1	U1	VccQ1	I/O power supply (1.8/3.3 V)		—
V2	Y2	A3	Address bus	O	VccQ1
V3	U4	A7	Address bus	O	VccQ1
V4	AA6	D12	Data bus	IO	VccQ1
V5	Y4	D14	Data bus	IO	VccQ1
V6	AA7	D9	Data bus	IO	VccQ1
V7	Y7	D6	Data bus	IO	VccQ1
V8	Y8	D2	Data bus	IO	VccQ1
V9	Y9	D0	Data bus	IO	VccQ1
V10	Y10	$\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$/PTM1	Chip select/chip select/ general-purpose port	O/O/IO	VccQ1
V11	V12	$\overline{\text{BREQ}}$	Bus request	I	VccQ1
V12	U13	$\overline{\text{WAIT}}$/$\overline{\text{PCC\_WAIT}}$	Wait/PCC wait	I/I	VccQ1
V13	U14	A20/PTR2	Address bus/general-purpose port	O/IO	VccQ1
V14	V14	A23/PTR5	Address bus/general-purpose port	O/IO	VccQ1
V15	Y19	$\overline{\text{DREQ0}}$/PINT0/PTM6	DMA transfer request/ port interrupt/general-purpose port	I/I/IO	VccQ1
V16	Y18	EXTAL_RTC	RTC external clock	I	VccQ_ RTC
V17	AA19	XTAL_RTC	RTC crystal	O	VccQ_ RTC
V18	V17	$\overline{\text{RESETP}}$	Power-on reset	I	VccQ_ RTC
V19	AA21	PINT5/$\overline{\text{PCC\_VS2}}$/PTK1	Port interrupt/ PCC voltage detection 2/ general-purpose port	I/I/IO	VccQ
V20	V20	VssQ	I/O power supply (0 V)		—

Pin No. (PLBG 0256 GA-A)	Pin No. (PLBG 0256 KA-A)	Pin Name	Function	I/O	I/O Buffer Power Supply
W1	U2	A8	Address bus	O	VccQ1
W2	AA2	A2	Address bus	O	VccQ1
W3	AA1	A1	Address bus	O	VccQ1
W4	AA4	A0/PTR0	Address bus/general-purpose port	O/IO	VccQ1
W5	AA5	D15	Data bus	IO	VccQ1
W6	V7	D10	Data bus	IO	VccQ1
W7	V8	D7	Data bus	IO	VccQ1
W8	V9	D3	Data bus	IO	VccQ1
W9	V10	$\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$/PTM0	Chip select/chip select/general-purpose port	O/O/IO	VccQ1
W10	U9	$\overline{\text{CS5A}}$/$\overline{\text{CE2A}}$	Chip select/chip select	O/O	VccQ1
W11	AA12	$\overline{\text{CS4}}$	Chip select	O	VccQ1
W12	AA13	A18	Address bus	O	VccQ1
W13	AA14	A21/PTR3	Address bus/general-purpose port	O/IO	VccQ1
W14	Y15	A25/PTR7	Address bus/general-purpose port	O/IO	VccQ1
W15	Y16	TEND0/PINT2/PTM2	DMA transfer end/port interrupt/general-purpose port	O/I/IO	VccQ1
W16	AA18	VccQ_RTC	RTC power supply (3.3 V)		—
W17	V16	TEND1/PINT3/PTM3	DMA transfer end/port interrupt/general-purpose port	O/I/IO	VccQ1
W18	Y20	Vss_RTC	RTC power supply (0 V)		—
W19	Y21	PINT4/$\overline{\text{PCC\_VS1}}$/PTK0	Port interrupt/PCC voltage detection 1/general-purpose port	I/I/IO	VccQ
W20	U18	VccQ	I/O power supply (3.3 V)		—
Y1	Y1	VssQ1	I/O power supply (0 V)		—
Y2	V5	VccQ1	I/O power supply (1.8/3.3 V)		—
Y3	V6	D13	Data bus	IO	VccQ1
Y4	Y3	VssQ1	I/O power supply (0 V)		—
Y5	V4	VccQ1	I/O power supply (1.8/3.3 V)		—
Y6	U5	D5	Data bus	IO	VccQ1

Pin No. (PLBG 0256 GA-A)	Pin No. (PLBG 0256 KA-A)	Pin Name	Function	I/O	I/O Buffer Power Supply
Y7	U6	VssQ1	I/O power supply (0 V)		—
Y8	U7	VccQ1	I/O power supply (1.8/3.3 V)		—
Y9	U8	$\overline{\text{CS6A}}$/$\overline{\text{CE2B}}$	Chip select/chip select	O/O	VccQ1
Y10	AA11	VssQ1	I/O power supply (0 V)		—
Y11	U10	VccQ1	I/O power supply (1.8/3.3 V)		—
Y12	Y11	$\overline{\text{CS0}}$	Chip select	O	VccQ1
Y13	Y12	$\overline{\text{RD}}$	Read strobe	O	VccQ1
Y14	Y13	VssQ1	I/O power supply (0 V)		VccQ1
Y15	Y14	VccQ1	I/O power supply (1.8/3.3 V)		—
Y16	AA15	VssQ1	I/O power supply (0 V)		—
Y17	AA16	VccQ1	I/O power supply (1.8/3.3 V)		—
Y18	AA17	$\overline{\text{DACK1}}$/PTM5	DMA transfer request reception/ general-purpose port	O/IO	VccQ1
Y19	Y17	CA	Chip active	I	VccQ_ RTC
Y20	AA20	Vcc_RTC	RTC power supply (1.5 V)		—

1.3.2 Pin Functions

Table 1.5 lists the pin functions.

Table 1.5 SH7720/SH7721 Pin Functions

Classification	Symbol	I/O	Name	Function
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.
	VccQ1	—	Power supply	Input/output power supply (1.8/3.3 V) pin.
	VssQ1	—	Ground	Input/output power supply (0 V) pin.
Clock	Vcc_PLL1	—	PLL1 power supply	Power supply for the on-chip PLL1 oscillator. (1.5 V)
	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. (1.5 V)
	Vss_PLL2	—	PLL2 ground	Ground pin for the on-chip PLL2 oscillator.
	EXTAL	I	External clock	For connection to a crystal resonator. An external clock signal may also be input.

Classification	Symbol	I/O	Name	Function
Clock	XTAL	O	Crystal	For connection to a crystal resonator.
	CKIO	I/O	System clock	Used as a pin to input external clock or output clock.
Operating mode control	MD5 to MD0	I	Mode set	Sets the operating mode. Do not change values on these pins during operation. MD2 to MD0 set the clock mode, MD3 and MD4 set the bus width of area 0 and MD5 sets the endian.
System control	$\overline{\text{RESETP}}$	I	Power-on reset	When low, the system enters the power-on reset state.
	$\overline{\text{RESETM}}$	I	Manual reset	When low, the system enters the manual reset state.
	STATUS1, STATUS0	O	Status output	Indicates the operating state.
	$\overline{\text{BREQ}}$	I	Bus request	Low when an external device requests the release of the bus mastership.
	$\overline{\text{BACK}}$	O	Bus request acknowledge	Indicates that the bus mastership has been released to an external device. Reception of the $\overline{\text{BACK}}$ signal informs the device which has output the $\overline{\text{BREQ}}$ signal that it has acquired the bus.
	CA	I	Chip active	High in normal operation, and low in hardware standby mode.
Interrupts	NMI	I	Non-maskable interrupt	Non-maskable interrupt request pin. Fix to high level when not in use.
	IRQ5 to IRQ0	I	Interrupt requests 5 to 0	Maskable interrupt request pins. Selectable as level input or edge input. The rising edge or falling edge is selectable as the detection edge. The low level or high level is selectable as the detection level.
	$\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$	I	Interrupt requests 3 to 0	Maskable interrupt request pin. Input a coded interrupt level.

Classification	Symbol	I/O	Name	Function
Interrupts	PINT15 to PINT0	I	Port interrupt requests 15 to 0	Port interrupt request pins
	$\overline{\text{REFOUT}}$	O	Bus request	Bus request signal for refreshing
	$\overline{\text{IRQOUT}}$	O	Bus request	Bus request signal for interrupt
Address bus	A25 to A0	O	Address bus	Outputs addresses.
Data bus	D31 to D0	I/O	Data bus	32-bit bidirectional data bus
Bus control	$\overline{\text{CS4}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$ $\overline{\text{CS6A}}$, $\overline{\text{CS6B}}$, $\overline{\text{CS5A}}$, $\overline{\text{CS5B}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$, $\overline{\text{CE1A}}$, $\overline{\text{CE1B}}$	O	Chip select	Chip-select signal for external memory or devices.
	$\overline{\text{RD}}$	O	Read strobe	Indicates reading of data from external devices.
	RD/$\overline{\text{WR}}$	O	Read/write signal	Read/write signal
	$\overline{\text{BS}}$	O	Bus start	Bus-cycle start signal pin
	$\overline{\text{BACK}}$	O	Bus request acknowledge	Indicates that the bus mastership has been released to an external device.
	$\overline{\text{BREQ}}$	I	Bus request	Low when an external device requests the release of the bus mastership.
	$\overline{\text{WE}}$	O	Write enable	Write enable pin for PCMCIA
	$\overline{\text{WE3}}$ ($\overline{\text{BE3}}$)	O	Highest-byte write	Indicates that bits 31 to 24 of the data in the external memory or device are being written.
	$\overline{\text{WE2}}$ ($\overline{\text{BE2}}$)	O	Second-highest-byte write	Indicates that bits 23 to 16 of the data in the external memory or device are being written.
	$\overline{\text{WE1}}$ ($\overline{\text{BE1}}$)	O	Second-lowest-byte write	Indicates that bits 15 to 8 of the data in the external memory or device are being written.
	$\overline{\text{WE0}}$ ($\overline{\text{BE0}}$)	O	Lowest-byte write	Indicates that bits 7 to 0 of the data in the external memory or device are being written.
	CKE	O	Clock enable	Clock enable (SDRAM)

Classification	Symbol	I/O	Name	Function
Bus control	$\overline{\text{CAS}}$	O	Column address	Connect to the $\overline{\text{CAS}}$ pin when the SDRAM is connected.
	DQMUU	O	DQ mask UU	Selects D31 to D24. (SDRAM)
	DQMUL	O	DQ mask UL	Selects D23 to D16. (SDRAM)
	DQMLU	O	DQ mask LU	Selects D15 to D8. (SDRAM)
	DQMLL	O	DQ mask LL	Selects D7 to D0. (SDRAM)
	$\overline{\text{RAS}}$	O	Row address	Connect to the $\overline{\text{RAS}}$ pin when the SDRAM is connected.
	$\overline{\text{WAIT}}$	I	Wait input	Inserts a wait cycle into the bus cycles during access to the external space.
	$\overline{\text{IOIS16}}$	I	16-bit IO	Indicates 16-bit I/O when PCMCIA is in use.
	$\overline{\text{ICIORD}}$	O	IO read	Indicates I/O read when PCMCIA is in use.
	$\overline{\text{ICIOWR}}$	O	IO write	Indicates I/O write when PCMCIA is in use.
Direct memory access controller (DMAC)	$\overline{\text{DREQ0}}$, $\overline{\text{DREQ1}}$	I	DMA-transfer request	Input pins for external requests for DMA transfer
	$\overline{\text{DACK0}}$, $\overline{\text{DACK1}}$	O	DMA transfer request reception	Indicates the acceptance of DMA transfer requests to external devices.
	TEND0, TEND1	O	DMA-transfer end	Transfer end output pins for DMAC
16-bit timer pulse unit (TPU)	TPU_TO3 to TPU_TO0	O	TPU compare-match output	TPU compare-match output pins
	TPU_TI3A to TPU_TI2A	I	TPU clock input	TPU clock input pins
	TPU_TI2B to TPU_TI3B	I	TPU clock input	TPU clock input pins
Analog front end interface (AFEIF)	AFE_RLYCNT	O	AFE on-hook control	On-hook control pin
	AFE_FS	I	AFE frame synchronization	AFE frame synchronization signal pin
	AFE_SCLK	I	AFE shift clock	AFE shift clock input pin

Classification	Symbol	I/O	Name	Function
Analog front end interface (AFEIF)	AFE_TXOUT	O	AFE serial transmission	AFE serial transmit data output pin
	$\overline{\text{AFE\_RDET}}$	I	AFE ringing signal	AFE ringing signal input pin
	AFE_HC1	O	AFE hardware control	AFE hardware control signal
	AFE_RXIN	I	AFE serial reception	AFE serial receive data
Serial communication interface with FIFO (SCIF)	SCIF0_TxD, SCIF1_TxD	O	SCIF transmit data	Transmit data pins
	SCIF0_RxD, SCIF1_RxD	I	SCIF receive data	Receive data pins
	SCIF0_SCK, SCIF1_SCK	I/O	SCIF serial clock	Clock input/output pins
	$\overline{\text{SCIF0\_RTS}}$, $\overline{\text{SCIF1\_RTS}}$	O	SCIF transmit request	Transmit request output pins
	$\overline{\text{SCIF0\_CTS}}$, $\overline{\text{SCIF1\_CTS}}$	I	SCIF transmit enable	Modem control pins
IrDA	IrTX	O	IrDA transmit data	IrDA transmit data output pin
	IrRX	I	IrDA receive data	IrDA receive data input pin
Serial I/O with FIFO (SIOF)	SIOF0_SYNC, SIOF1_SYNC	I/O	SIOF frame sync	SIOF frame synchronization signals
	SIOF0_TxD, SIOF1_TxD	O	SIOF transmit data	SIOF transmit data pin
	SIOF0_RxD, SIOF1_RxD	I	SIOF receive data	SIOF receive data pin
	SIOF0_SCK, SIOF1_SCK	I/O	SIOF serial clock	SIOF serial clock pins
	SIOF0_MCLK, SIOF1_MCLK	I	SIOF master clock	SIOF master clock input pins
I^2C bus interface (IIC)	IIC_SCL	I/O	IIC clock	I^2C serial clock pin
	IIC_SDA	I/O	IIC data	I^2C data input/output pin
Realtime clock (RTC)	VccQ_RTC	—	RTC power supply	Power supply pin for the RTC (3.3 V)

Classification	Symbol	I/O	Name	Function
Realtime clock (RTC)	Vcc_RTC	—	RTC power supply	Power supply pin for the RTC (1.5 V)
	Vss_RTC	—	RTC ground	Ground pin for the RTC.
	EXTAL_RTC	I	RTC external clock	Connects crystal resonator for the RTC. Also used to input external clock for the RTC.
	XTAL_RTC	O	RTC crystal	Connects crystal resonator for the RTC.
LCD controller (LCDC)	LCD_DATA15 to LCD_DATA0	O	LCD data	Data output pin for LCD panel
	LCD_CL1	O	LCD shift clock 1	LCD shift clock 1/ horizontal sync signal pin
	LCD_CL2	O	LCD shift clock 2	LCD shift clock 2/dot clock pin
	LCD_CLK	I	LCD clock source	LCD clock source input pin
	LCD_FLM	O	LCD line marker	First line marker/vertical sync signal pin
	LCD_DON	O	LCD display on	LCD display on signal pin
	LCD_VCPWC	O	LCD power control (VCC)	LCD module power control (VCC) pin
	LCD_VEPWC	O	LCD power control (VEE)	LCD module power control (VEE) pin
	LCD_M_DISP	O	LCD current alternating signal	LCD current alternating signal pin
PC card controller (PCC)	PCC_BVD1	I	PCC battery detection 1	Pin for buttery voltage detect 1/ card status change signal from PC card
	PCC_BVD2	I	PCC battery detection 2	Pin for buttery voltage detect 2/ digital sound signal pin from PC card
	PCC_RDY	I	PCC ready	Pin for ready signal/interrupt request signal form PC card
	$\overline{\text{PCC\_REG}}$	O	PCC space indication	Area indicate signal pin for PC card
	PCC_RESET	O	PCC reset	Reset signal pin for PC card
	$\overline{\text{PCC\_CD1}}$	I	PCC card detection 1	Pin for card detect 1 signal from PC card

Classification	Symbol	I/O	Name	Function
PC card controller (PCC)	$\overline{\text{PCC\_CD2}}$	I	PCC card detection 2	Pin for card detect 2 signal from PC card
	$\overline{\text{PCC\_WAIT}}$	I	PCC wait request	PCC hardware wait request signal pin
	$\overline{\text{PCC\_DRV}}$	O	PCC buffer control	PCC buffer control signal pin
	$\overline{\text{PCC\_VS1}}$	I	PCC voltage detection 1	Pin for voltage sense 1 signal from PC card
	$\overline{\text{PCC\_VS2}}$	I	PCC voltage detection 2	Pin for voltage sense 2 signal from PC card
	$\overline{\text{PCC\_IOIS16}}$	I	PCC16-bit IO	Pin for write protection signal/16-bit I/O signal from PC card
MultiMedia Card interface (MMCIF)	$\overline{\text{MMC\_ODMOD}}$	O	MMC open drain control	Open drain mode control pin
	MMC_VDDON	O	MMC card power control	MMC power control pin
	MMC_CLK	O	MMC clock	Clock output pin
	MMC_DAT	I/O	MMC data	Data input/output pin in MMC mode Response/data input pin in SPI mode This pin is connected to the Data out pin on the MMC side.
	MMC_CMD	I/O	MMC command	Command output/response input pin in MMC mode Command/data output pin in SPI mode This pin is connected to the Data in pin on the MMC side.
SD host interface (SDHI)	SD_CLK	O	SD clock	Clock output pin
	SD_CMD	I/O	SD command	Command output/response input pin
	SD_DAT0	I/O	SD data 0	Data input/output pin
	SD_DAT1	I/O	SD data 1	Data input/output pin
	SD_DAT2	I/O	SD data 2	Data input/output pin
	SD_DAT3	I/O	SD data 3	Data input/output pin

Classification	Symbol	I/O	Name	Function
SD host interface (SDHI)	SD_CD	I	SD card detection	Card detection pin
	SD_WP	I	SD write protect	Write protect pin
SIM card module (SIM)	SIM_RST	O	SIM reset	Smart card reset output pin
	SIM_CLK	O	SIM clock	Smart card clock output pin
	SIM_D	I/O	SIM data	Transmit/receive data input/output pin
A/D converter (ADC)	AN3 to AN0	I	ADC analog input	Analog input pin
	AVcc	—	Analog power supply	Power supply pin for the A/D or D/A converter. When the A/D or D/A converter is not in use, connect this pin to input/output power supply (VccQ).
	AVss	—	Analog ground	Ground pin for the A/D or D/A converter. Connect this pin to input/output power supply (VssQ).
	$\overline{\text{ADTRG}}$	I	ADC external trigger	External trigger signal for starting A/D conversion
D/A converter (DAC)	DA0	O	DAC analog output	Channel 0 analog output pin
	DA1	O	DAC analog output	Channel 1 analog output pin
USB	AVcc_USB	—	USB power supply	Power supply pin for USB
	AVss_USB	—	USB ground	Ground pin for USB
	EXTAL_USB	I	USB external clock	Connects crystal resonator for USB. Also used to input external clock for USB (48 MHz)
	XTAL_USB	O	USB crystal	Connects a crystal resonator for USB
	$\overline{\text{USB1\_ovr\_current}}$/ USBF_VBUS	I	USB1 over-current/ monitor	USB port 1 over-current detection/ USB cable connection monitor pin
	$\overline{\text{USB2\_ovr\_current}}$	I	USB2 over-current	USB port 2 over-current detection pin

Classification	Symbol	I/O	Name	Function
USB	USB1_pwr_en/ USBF_UPLUP	O	USB1 power enable/pull-up control	USB port 1 power enable control/ pull- up control output pin
	SUB2_pwer_en	O	USB2 power enable	USB port 2 power enable control pin
	USB1_P	I/O	USB D+ port 1	D+ port 1 transceiver pin for USB
	USB1_M	I/O	USB D− port 1	D− port 1 transceiver pin for USB
	USB2_P	I/O	USB D+ port 2	D+ port 2 transceiver pin for USB
	USB2_M	I/O	USB D− port 2	D− port 2 transceiver pin for USB
	USB1d_DMNS	I	D− signal input	Input pin to driver for D− signal from receiver
	USB1d_ SUSPEND	O	Suspend state	Transceiver suspend state output pin
	USB1d_RCV	I	Receive data	Input pin for receive data from differential receiver
	USB1d_TXENL	O	Driver output enable	Driver output enable pin
	USB1d_SPEED	O	Speed control	Transceiver speed control pin
	USB1d_TXSE0	O	SE0 state	SE0 state output pin
	USB1d_ TXDPLS	O	D+ transmit output	D+ transmit output pin to driver
	USB1d_DPLS	I	D+ transmit input	D+ transmit input pin to driver
I/O port	PTA7 to PTA0	I/O	General purpose port	8-bit general-purpose port pins
	PTB7 to PTB0	I/O	General purpose port	8-bit general-purpose port pins
	PTC7 to PTC0	I/O	General purpose port	8-bit general-purpose port pins
	PTD7 to PTD0	I/O	General purpose port	8-bit general-purpose port pins
	PTE6, PTE5	I	General purpose port	7-bit general-purpose port pins
	PTE4 to PTE0	I/O	General purpose port	
	PTF6 to PTF0	I	General purpose port	7-bit general-purpose port pins

Classification	Symbol	I/O	Name	Function
I/O port	PTG6 to PTG0	I/O	General purpose port	7-bit general-purpose port pins
	PTH6 to PTH0	I/O	General purpose port	7-bit general-purpose port pins
	PTJ6 to PTJ0	I/O	General purpose port	7-bit general-purpose port pins
	PTK3 to PTK0	I/O	General purpose port	4-bit general-purpose port pins
	PTL7 to PTL3	I/O	General purpose port	5-bit general-purpose port pins
	PTM7 to PTM0	I/O	General purpose port	8-bit general-purpose port pins
	PTP4 to PTP0	I/O	General purpose port	5-bit general-purpose port pins
	PTR7 to PTR0	I/O	General purpose port	8-bit general-purpose port pins
	PTS4 to PTS0	I/O	General purpose port	5-bit general-purpose port pins
	PTT4 to PTT0	I/O	General purpose port	5-bit general-purpose port pins
	PTU4 to PTU0	I/O	General purpose port	5-bit general-purpose port pins
	PTV4 to PTV0	I/O	General purpose port	5-bit general-purpose port pins
User debugging interface (H-UDI)	TCK	I	Test clock	Test-clock input pin
	TMS	I	Test mode select	Test-mode select signal input pin
	TDI	I	Test data input	Serial input pin for instructions and data
	TDO	O	Test data output	Serial output pin for instructions and data
	$\overline{\text{TRST}}$	I	Test reset	Initial-signal input pin

Classification	Symbol	I/O	Name	Function
Advanced user debugger (AUD)	AUDATA3 to AUDATA0	O	AUD data	Destination-address output pin in branch-trace mode
	AUDCK	O	AUD clock	Synchronous clock output pin in branch-trace mode
	$\overline{\text{AUDSYNC}}$	O	AUD synchronous signal	Data start-position acknowledge-signal output pin in branch-trace mode
E10A interface	$\overline{\text{ASEBRKAK}}$	O	ASE break mode acknowledge	Indicates that the E10A emulator has entered its break mode.
	$\overline{\text{ASEMD0}}$	I	ASE mode	Sets ASE mode.

Notes: 1. All Vcc/Vss/VccQ/VssQ/VccQ1/VssQ1/AVcc/AVss/AVcc_USB/AVss_USB/VccQ_RTC/Vcc_RTC/Vss_RTC/Vcc_PLL1/Vss_PLL1/Vcc_PLL2/Vss_PLL2 should be connected to the system power supply (so that power is supplied at all times.) In hardware standby mode, the power supply to other than Vcc_RTC and VccQ_RTC can be turned off (section 13.8).

2. Always supply power to the Vcc_RTC and VccQ_RTC, even if the RTC is not being used.
3. Always supply power to the Vcc_PLL1 and Vcc_PLL2, even if the PLL is not being used.
4. Drive $\overline{\text{ASEMD0}}$ high when using the user system alone, and not using an emulator or the H-UDI. When this pin is low or open, $\overline{\text{RESETP}}$ may be masked.
5. Drivability can be switched by the register settings of the pin function controller (PFC). When 3.3 V is applied to VccQ1, set the drivability low. When 1.8 V is applied to VccQ1, set the drivability high.
6. SDHI associated pins support only for the models including the SDHI.

Section 2 CPU

2.1 Processing States and Processing Modes

2.1.1 Processing States

This LSI supports four types of processing states: a reset state, an exception handling state, a program execution state, and a low-power consumption state, according to the CPU processing states.

(1) Reset State

In the reset state, the CPU is reset. The LSI supports two types of resets: power-on reset and manual reset. For details on resets, refer to section 7, Exception Handling.

In power-on reset, the registers and internal statuses of all LSI on-chip modules are initialized. In manual reset, the register contents of a part of the LSI on-chip modules are retained. For details, refer to section 37, List of Registers. The CPU internal statuses and registers are initialized both in power-on reset and manual reset. After initialization, the program branches to address H'A0000000 to pass control to the reset processing program to be executed.

(2) Exception Handling State

In the exception handling state, the CPU processing flow is changed temporarily by a general exception or interrupt exception processing. The program counter (PC) and status register (SR) are saved in the save program counter (SPC) and save status register (SSR), respectively. The program branches to an address obtained by adding a vector offset to the vector base register (VBR) and passes control to the exception processing program defined by the user to be executed. For details on reset, refer to section 7, Exception Handling.

(3) Program Execution State

The CPU executes programs sequentially.

(4) Low-Power Consumption State

The CPU stops operation to reduce power consumption. The power-down mode can be entered by executing the SLEEP instruction. For details on the power-down mode, refer to section 13, Power-Down Modes.

Figure 2.1 shows a status transition diagram.

2.1.2 Processing Modes

This LSI supports two processing modes: user mode and privileged mode. These processing modes can be determined by the processing mode bit (MD) in the status register (SR). If the MD bit is cleared to 0, the user mode is selected. If the MD bit is set to 1, the privileged mode is selected. The CPU enters the privileged mode by a transition to reset state or exception handling state. In the privileged mode, any registers and resources in address spaces can be accessed.

Clearing the MD bit in the SR to 0 puts the CPU in the user mode. In the user mode, some of the registers, including SR, and some of the address spaces cannot be accessed by the user program and system control instructions cannot be executed. This function effectively protects the system resources from the user program. To change the processing mode from user to privileged mode, a transition to exception handling state is required.

Note: To call a service routine used in privileged mode from user mode, the LSI supports an unconditional trap instruction (TRAPA). When a transition from user mode to privileged mode occurs, the contents of the SR and PC are saved. A program execution in user mode can be resumed by restoring the contents of the SR and PC. To return from an exception processing program, the LSI supports an RTE instruction.

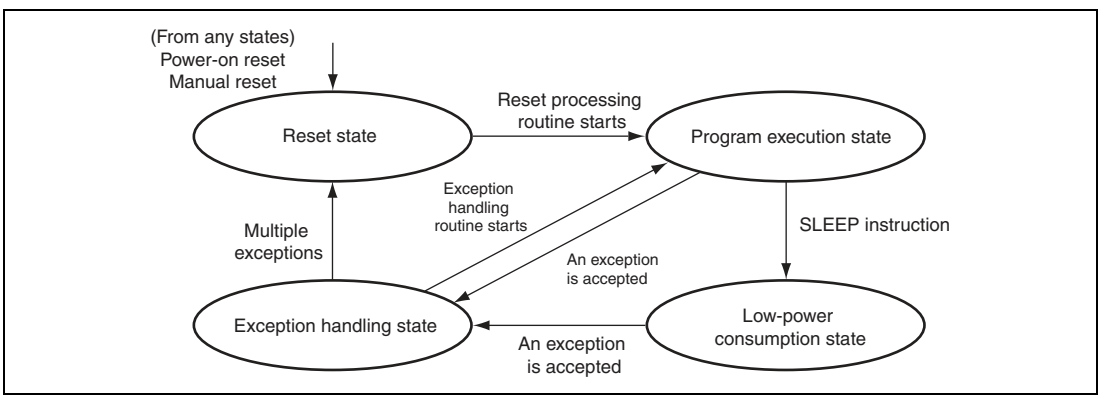


Figure 2.1 Processing State Transitions

2.2 Memory Map

2.2.1 Virtual Address Space

The LSI supports 32-bit virtual addresses and accesses system resources using the 4-Gbytes of virtual address space. User programs and data are accessed from the virtual address space. The virtual address space is divided into several areas as shown in table 2.1.

(1) P0/U0 Area

This area is called the P0 area when the CPU is in privileged mode and the U0 area when in user mode. For the P0 and U0 areas, access using the cache is enabled. The P0 and U0 areas are handled as address translatable areas.

If the cache is enabled, access to the P0 or U0 area is cached. If a P0 or U0 address is specified while the address translation unit is enabled, the P0 or U0 address is translated into a physical address based on translation information defined by the user.

If the CPU is in user mode, only the U0 area can be accessed. If P1, P2, P3, or P4 is accessed in user mode, a transition to an address error exception occurs.

(2) P1 Area

The P1 area is defined as a cacheable but non-address translatable area. Normally, programs executed at high speed in privileged mode, such as exception processing handlers, which are at the core of the operating system (OS), are assigned to the P1 area.

(3) P2 Area

The P2 area is defined as a non-cacheable but non-address translatable area. A reset processing program to be called from the reset state is described at the start address (H'A0000000) of the P2 area. Normally, programs such as system initialization routines and OS initiation programs are assigned to the P2 area. To access a part of an on-chip I/O, its corresponding program should be assigned to the P2 area.

(4) P3 Area

The P3 area is defined as a cacheable and address translatable area. This area is used if an address translation is required for a privileged program.

(5) P4 Area

The P4 area is defined as a control area which is non-cacheable and non-address translatable. This area can be accessed only in privileged mode. A part of the LSI's on-chip I/O is assigned to this area.

Table 2.1 Virtual Address Space

Address Range	Name	Mode	Description
H'00000000 to H'7FFFFFFF	P0/U0	Privileged/user mode	2-Gbyte physical space, cacheable, address translatable In user mode, only this address space can be accessed.
H'80000000 to H'9FFFFFFF	P1	Privileged mode	0.5-Gbyte physical space, cacheable
H'A0000000 to H'BFFFFFFF	P2	Privileged mode	0.5-Gbyte physical space, non-cacheable
H'C0000000 to H'DFFFFFFF	P3	Privileged mode	0.5-Gbyte physical space, cacheable, address translatable
H'E0000000 to H'FFFFFFFF	P4	Privileged mode	0.5-Gbyte control space, non-cacheable

2.2.2 External Memory Space

This LSI uses 29 bits of the 32-bit virtual address to access external memory. In this case, 0.5-Gbyte of external memory space can be accessed. The external memory space is managed in area units. Different types of memory can be connected to each area, as shown in figure 2.2. For details, please refer to section 9, Bus State Controller (BSC).

In addition, area 1 in the external memory space is used as an on-chip I/O space where most of this LSI's on-chip I/Os are mapped.

Normally, the upper three bits of the 32-bit virtual address are masked and the lower 29 bits are used for external memory addresses.*[2] For example, address H'00000100 in the P0 area, address H'80000100 in the P1 area, address H'A0000100 in the P2 area, and address H'C0000100 in the P3 area of the virtual address space are mapped into address H'00000100 of area 0 in the external memory space. The P4 area in the virtual address space is not mapped into the external memory address. If an address in the P4 area is accessed, an external memory cannot be accessed.

Notes: 1. To access an on-chip I/O mapped into area 1 in the external memory space, access the address from the P2 area which is not cached in the virtual address space.

2. If the address translation unit is enabled, arbitrary mapping in page units can be specified. For details, refer to section 4, Memory Management Unit (MMU).

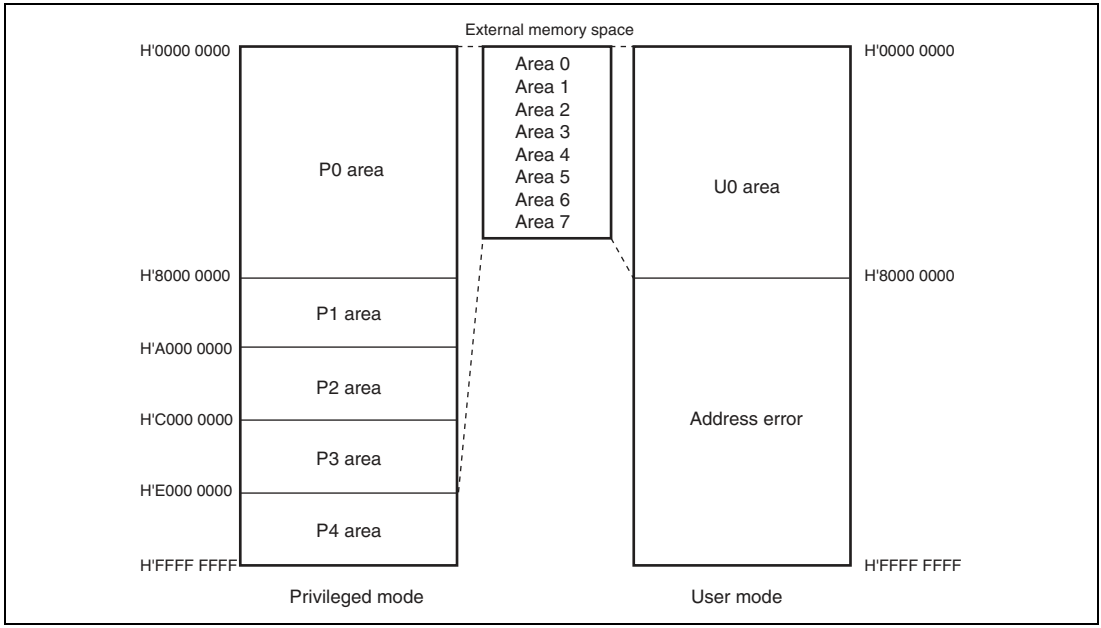


Figure 2.2 Virtual Address to External Memory Space Mapping

2.3 Register Descriptions

This LSI provides thirty-three 32-bit registers: 24 general registers, five control registers, three system registers, and one program counter.

(1) General Registers

This LSI incorporates 24 general registers: R0_BANK0 to R7_BANK0, R0_BANK1 to R7_BANK1 and R8 to R15. R0 to R7 are banked. The process mode and the register bank (RB) bit in the status register (SR) define which set of banked registers (R0_BANK0 to R7_BANK0 or R0_BANK1 to R7_BANK1) are accessed as general registers.

(2) System Registers

This LSI incorporates the multiply and accumulate registers (MACH/MACL) and procedure register (PR) as system registers. These registers can be accessed regardless of the processing mode.

(3) Program Counter

The program counter stores the value obtained by adding 4 to the current instruction address.

(4) Control Registers

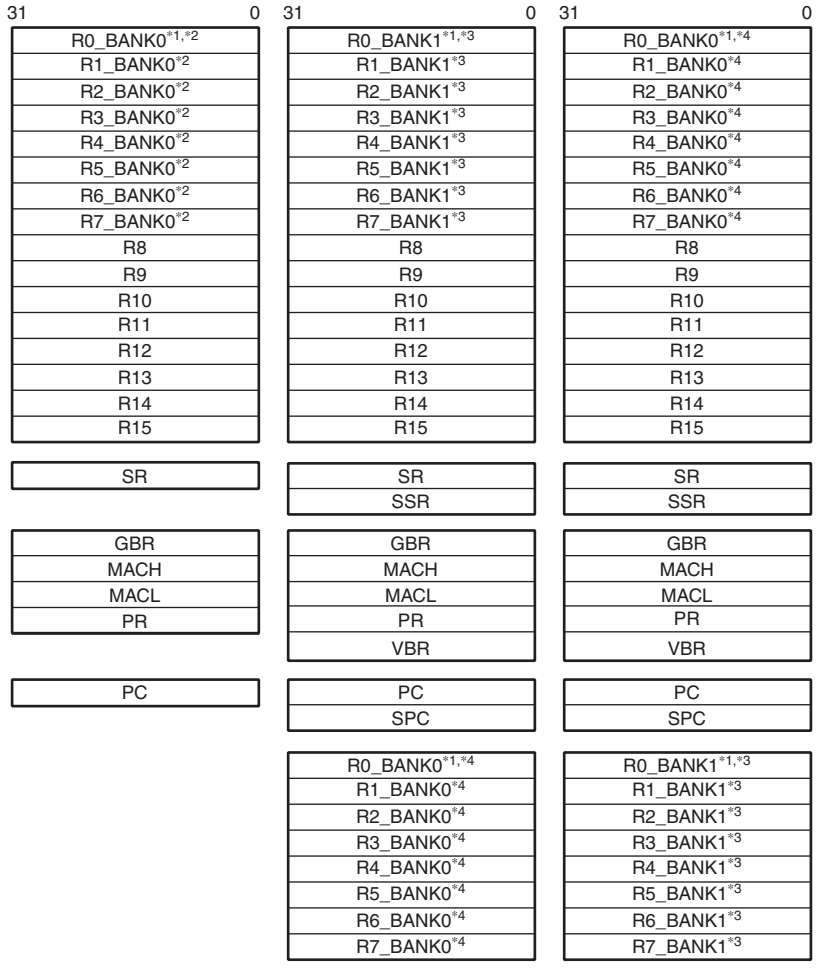
This LSI incorporates the status register (SR), global base register (GBR), save status register (SSR), save program counter (SPC), and vector base register as control register. Only the GBR can be accessed in user mode. Control registers other than the GBR can be accessed only in privileged mode.

Table 2.2 shows the register values after reset. Figure 2.3 shows the register configurations in each process mode.

Table 2.2 Register Initial Values

Register Type	Registers	Initial Values*
General registers	R0_BANK0 to R7_BANK0, R0_BANK1 to R7_BANK1, R8 to R15	Undefined
System registers	MACH, MACL, PR	Undefined
Program counter	PC	H'A0000000
Control registers	SR	MD bit = 1, RB bit = 1, BL bit = 1, I3 to I0 bits = H'F (1111), reserved bits = all 0, other bits = undefined
	GBR, SSR, SPC	Undefined
	VBR	H'00000000

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



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

Figure 2.3 Register Configuration in Each Processing Mode

2.3.1 General Registers

There are twenty-four 32-bit general registers: R0_BANK0 to R7_BANK0, R0_BANK1 to R7_BANK1, and R8 to R15. R0 to R7 are banked. The process mode and the register bank (RB) bit in the status register (SR) define which set of banked registers (R0_BANK0 to R7_BANK0 or R0_BANK1 to R7_BANK1) are accessed as general registers. R0 to R7 registers in the selected bank are accessed as R0 to R7. R0 to R7 in the non-selected bank is accessed as R0_BANK to R7_BANK by the control register load instruction (LDC) and control register store instruction (STC).

In user mode, bank 0 is selected regardless of the RB bit value. Sixteen registers: R0_BANK0 to R7_BANK0 and R8 to R15 are accessed as general registers R0 to R15. The R0_BANK1 to R7_BANK1 registers in bank 1 cannot be accessed.

In privileged mode that is entered by a transition to exception handling state, the RB bit is set to 1 to select bank 1. In privileged mode, sixteen registers: R0_BANK1 to R7_BANK1 and R8 to R15 are accessed as general registers R0 to R15. A bank is switched automatically when an exception handling state is entered, registers R0 to R7 need not be saved by the exception handling routine. The R0_BANK0 to R7_BANK0 registers in bank 0 can be accessed as R0_BANK to R7_BANK by the LDC and STC instructions.

In privileged mode, bank 0 can also be used as general registers by clearing the RB bit to 0. In this case, sixteen registers: R0_BANK0 to R7_BANK0 and R8 to R15 are accessed as general registers R0 to R15. The R0_BANK1 to R7_BANK1 registers in bank 1 can be accessed as R0_BANK to R7_BANK by the LDC and STC instructions.

The general registers R0 to R15 are used as equivalent registers for almost all instructions. In some instructions, the R0 register is automatically used or only the R0 register can be used as source or destination registers.

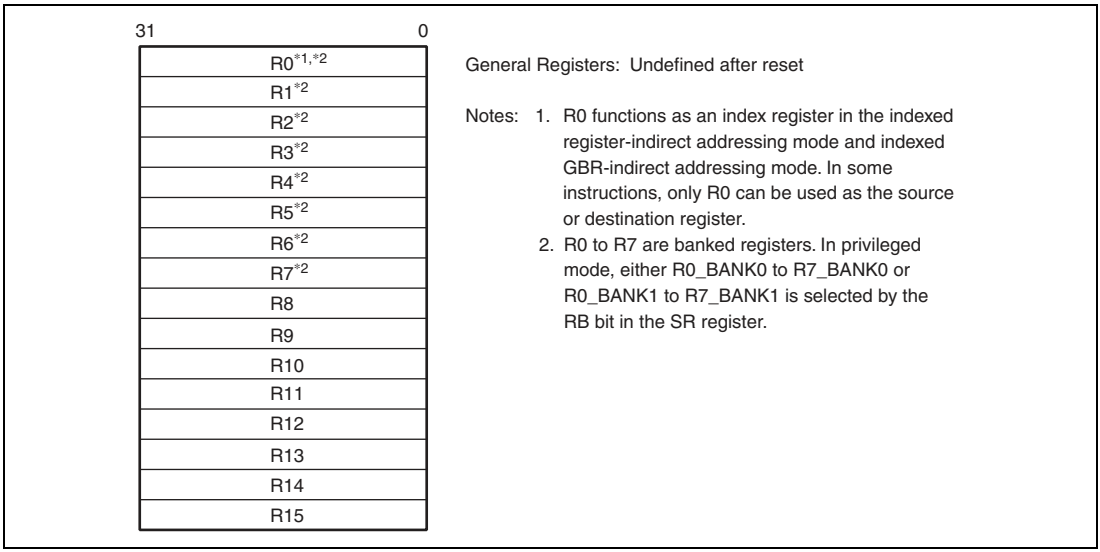


Figure 2.4 General Registers

2.3.2 System Registers

The system registers: multiply and accumulate registers (MACH/MACL) and procedure register (PR) as system registers can be accessed by the LDS and STS instructions.

(1) Multiply and Accumulate Registers (MACH/MACL)

The multiply and accumulate registers (MACH/MACL) store the results of multiplication and accumulation instructions or multiplication instructions. The MACH/MACL registers also store addition values for the multiplication and accumulations. After reset, these registers are undefined. The MACH and MACL registers store upper 32 bits and lower 32 bits, respectively.

(2) Procedure Register (PR)

The procedure register (PR) stores the return address for a subroutine call using the BSR, BSRF, or JSR instruction. The return address stored in the PR register is restored to the program counter (PC) by the RTS (return from the subroutine) instruction. After reset, this register is undefined.

2.3.3 Program Counter

The program counter (PC) stores the value obtained by adding 4 to the current instruction address. There is no instruction to read the PC directly. Before an exception handling state is entered, the PC is saved in the save program counter (SPC). Before a subroutine call is executed, the PC is saved in the procedure register (PR). In addition, the PC can be used for PC relative addressing mode.

Figure 2.5 shows the system register and program counter configurations.

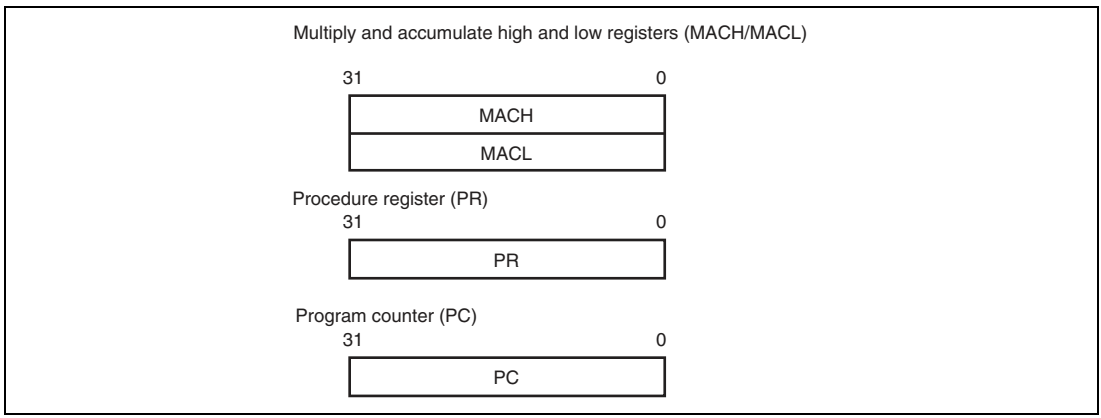


Figure 2.5 System Registers and Program Counter

2.3.4 Control Registers

The control registers (SR, SSR, SPC, GBR, and VBR) can be accessed by the LDC or STC instruction in privileged mode. The GBR register can be accessed in the user mode.

The control registers are described below.

(1) Status Register (SR)

The status register (SR) indicates the system status as shown below. The SR register can be accessed only in privileged mode.

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	MD	1	R/W	Processing Mode Indicates the CPU processing mode. 0: User mode 1: Privileged mode The MD bit is set to 1 in reset or exception handling state.
29	RB	1	R/W	Register Bank The general registers R0 to R7 are banked registers. 0: In this case, R0_BANK0 to R7_BANK0 and R8 to R15 are used as general registers. R0_BANK1 to R7_BANK1 can be accessed by the LDC or STR instruction. 1: In this case, R0_BANK1 to R7_BANK1 and R8 to R15 are used as general registers. R0_BANK0 to R7_BANK0 can be accessed by the LDC or STR instruction. The RB bit is set to 1 in reset or exception handling state.

Bit	Bit Name	Initial Value	R/W	Description
28	BL	1	R/W	Block Specifies whether an exception, interrupt, or user break is enabled or not. 0: Enables an exception, interrupt, or user break. 1: Disables an exception, interrupt, or user break. The BL bit is set to 1 in reset or exception handling state.
27 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	M	—	R/W	M Bit
8	Q	—	R/W	Q Bit These bits are used by the DIV0S, DIV0U, and DIV1 instructions. These bits can be changed even in user mode by using the DIV0S, DIV0U, and DIV1 instructions. These bits are undefined at reset. These bits do not change in an exception handling state.
7 to 4	I3 to I0	All 1	R/W	Interrupt Mask Indicates the interrupt mask level. These bits do not change even if an interrupt occurs. At reset, these bits are initialized to B'1111. These bits are not affected in an exception handling state.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	S	—	R/W	Saturation Mode Specifies the saturation mode for multiply instructions or multiply and accumulate instructions. This bit can be specified by the SETS and CLRS instructions in user mode. At reset, this bit is undefined. This bit is not affected in an exception handling state.

Bit	Bit Name	Initial Value	R/W	Description
0	T	—	R/W	T Bit Indicates true or false for compare instructions or carry or borrow occurrence for an operation instruction with carry or borrow. This bit can be specified by the SETT and CLRT instructions in user mode. At reset, this bit is undefined. This bit is not affected in an exception handling state.

Note: The M, Q, S, and T bits can be set/cleared by the user mode specific instructions. Other bits can be read or written in privileged mode.

(2) Save Status Register (SSR)

The save status register (SSR) can be accessed only in privileged mode. Before entering the exception, the contents of the SR register is stored in the SSR register. At reset, the SSR initial value is undefined.

(3) Save Program Counter (SPC)

The save program counter (SPC) can be accessed only in privileged mode. Before entering the exception, the contents of the PC is stored in the SPC. At reset, the SPC initial value is undefined.

(4) Global Base Register (GBR)

The global base register (GBR) is referenced as a base register in GBR indirect addressing mode. At reset, the GBR initial value is undefined.

(5) Vector Base Register (VBR)

The vector base register (VBR) can be accessed only in privileged mode. If a transition from reset state to exception handling state occurs, this register is referenced as a base address. For details, refer to section 7, Exception Handling. At reset, the VBR is initialized as H'00000000.

Figure 2.6 shows the control register configuration.

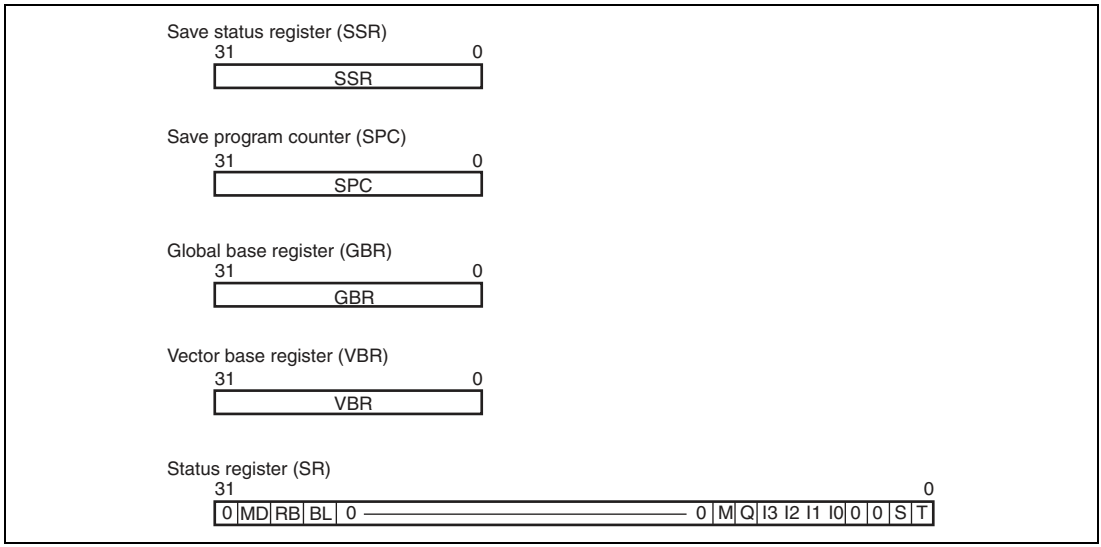
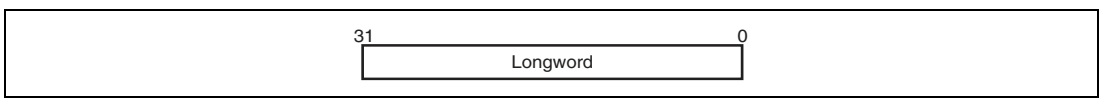


Figure 2.6 Control Register Configuration

2.4 Data Formats

2.4.1 Register Data Format

Register operands are always longwords (32 bits). When the 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.



2.4.2 Memory Data Formats

Memory data formats are classified into byte, word, and longword. Memory can be accessed in byte, word, and longword. When the 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.

An address error will occur if word data starting from an address other than 2n or longword data starting from an address other than 4n is accessed. In such cases, the data accessed cannot be guaranteed.

When a word or longword operand is accessed, the byte positions on the memory corresponding to the word or longword data on the register is determined to the specified endian mode (big endian or little endian).

Figure 2.7 shows a byte correspondence in big endian mode. In big endian mode, the MSB byte in the register corresponds to the lowest address in the memory, and the LSB the in the register corresponds to the highest address. For example, if the contents of the general register R0 is stored at an address indicated by the general register R1 in longword, the MSB byte of the R0 is stored at the address indicated by the R1 and the LSB byte of the R1 register is stored at the address indicated by the (R1 +3).

The on-chip device registers assigned to memory are accessed in big endian mode. Note that the available access size (byte, word, or long word) differs in each register.

Note: The CPU instruction codes of this LSI must be stored in word units. In big endian mode, the instruction code must be stored from upper byte to lower byte in this order from the word boundary of the memory.

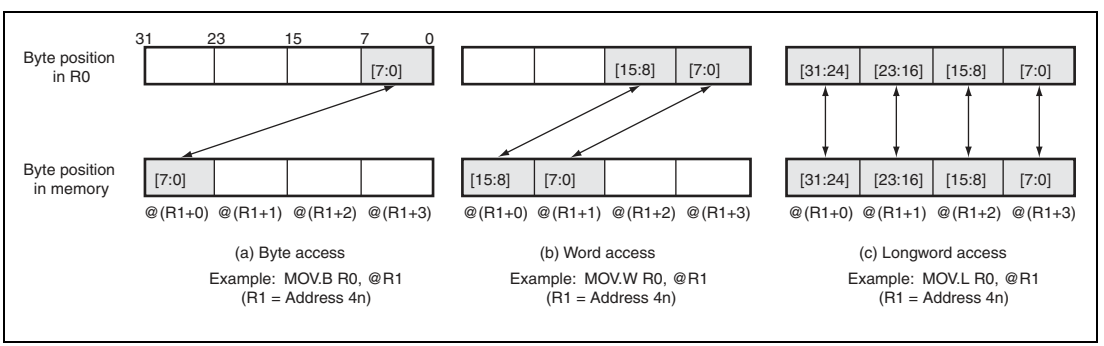


Figure 2.7 Data Format on Memory (Big Endian Mode)

The little endian mode can also be specified as data format. Either big-endian or little-endian mode can be selected according to the MD5 pin at reset. When MD5 is low at reset, the processor operates in big-endian mode. When MD5 is high at reset, the processor operates in little-endian mode. The endian mode cannot be modified dynamically.

In little endian mode, the MSB byte in the register corresponds to the highest address in the memory, and the LSB the in the register corresponds to the lowest address (figure 2.8). For example, if the contents of the general register R0 is stored at an address indicated by the general register R1 in longword, the MSB byte of the R0 is stored at the address indicated by the (R1+3) and the LSB byte of the R1 register is stored at the address indicated by the R1.

If the little endian mode is selected, the on-chip memory are accessed in little endian mode. However, the on-chip device registers assigned to memory are accessed in big endian mode. Note that the available access size (byte, word, or long word) differs in each register.

Note: The CPU instruction codes of this LSI must be stored in word units. In little endian mode, the instruction code must be stored from lower byte to upper byte in this order from the word boundary of the memory.

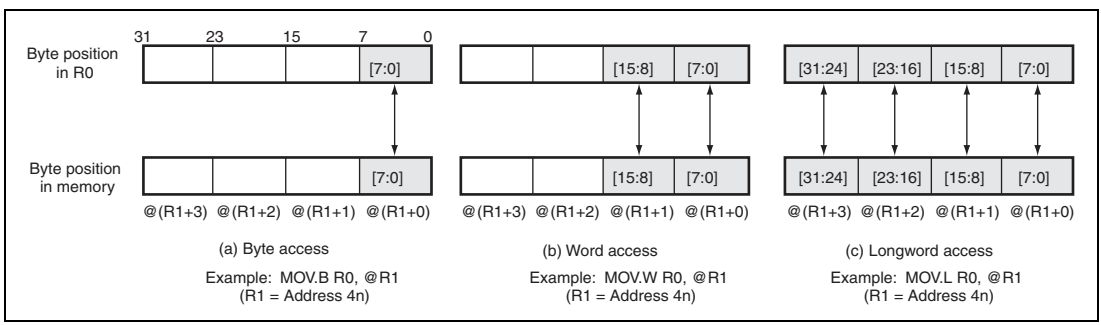


Figure 2.8 Data Format on Memory (Little Endian Mode)

2.5 Features of CPU Core Instructions

2.5.1 Instruction Execution Method

(1) Instruction Length

All instructions have a fixed length of 16 bits and are executed in the sequential pipeline. In the sequential pipeline, almost all instructions can be executed in one cycle. All data items are handles in longword (32 bits). Memory can be accessed in byte, word, or longword. In this case, Memory byte or word data is sign-extended and operated on as longword data. Immediate data is sign-extended to longword size for arithmetic operations (MOV, ADD, and CMP/EQ instructions) or zero-extended to longword size for logical operations (TST, AND, OR, and XOR instructions).

(2) Load/Store Architecture

Basic operations are executed between registers. In operations involving memory, data is first loaded into a register (load/store architecture). However, bit manipulation instructions such as AND are executed directly on memory.

(3) Delayed Branching

Unconditional branch instructions are executed as delayed branches. With a delayed branch instruction, the branch is made after execution of the instruction (called the slot instruction) immediately following the delayed branch instruction. This minimizes disruption of the pipeline when a branch is made.

This LSI supports two types of conditional branch instructions: delayed branch instruction or normal branch instruction.

```
Example:  BRA       TARGET
          ADD       R1, R0      ; ADD is executed before branching to
the                               TARGET
```

(4) T Bit

The result of a comparison is indicated by the T bit in the status register (SR), and a conditional branch is performed according to whether the result is True or False. Processing speed has been improved by keeping the number of instructions that modify the T bit to a minimum.

```
Example:  ADD       #1, R0      ; The T bit cannot be modified by the
ADD                               instruction
          CMP/EQ    #0, R0      ; The T bit is set to 1 if R0 is 0.
          BT        TARGET      ; Branch to TARGET if the T bit is set
to                        1 (R0=0).
```

(5) Literal Constant

Byte literal constant is placed inside the instruction code as immediate data. Since the instruction length is fixed to 16 bits, word and longword literal constant is not placed inside the instruction code, but in a table in memory. The table in memory is referenced with a MOV instruction using PC-relative addressing mode with displacement.

```
Example:  MOV.W     @(disp, PC), R0
```

(6) Absolute Addresses

When data is referenced by absolute address, the absolute address value is placed in a table in memory beforehand as well as word or longword literal constant. Using the method whereby immediate data is loaded when an instruction is executed, this value is transferred to a register and the data is referenced using register indirect addressing mode.

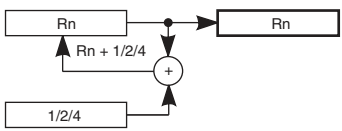
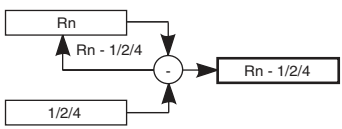
(7) 16-Bit/32-Bit Displacement

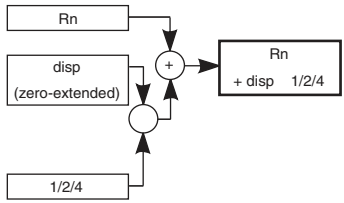
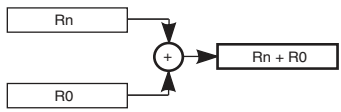
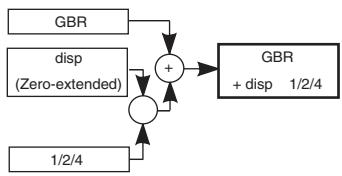
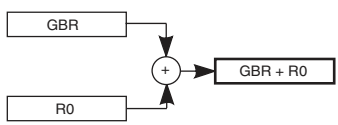
When data is referenced with a 16- or 32-bit displacement, the displacement value is placed in a table in memory beforehand. Using the method whereby word or longword immediate data is loaded when an instruction is executed, this value is transferred to a register and the data is referenced using indexed register indirect addressing mode.

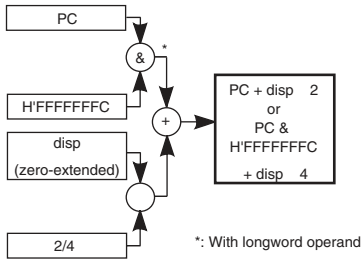
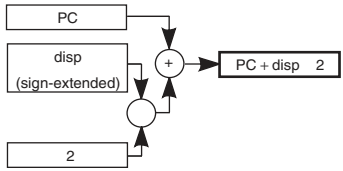
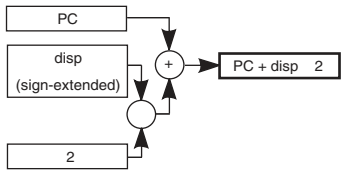
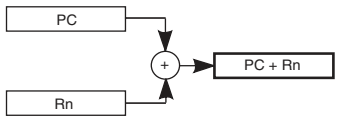
2.5.2 CPU Instruction Addressing Modes

The following table shows addressing modes and effective address calculation methods for instructions executed by the CPU core.

Table 2.3 Addressing Modes and Effective Addresses for CPU Instructions

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
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. 	Rn After instruction execution Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn
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. 	Byte: Rn – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn (Instruction executed with Rn after calculation)

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
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 Word: Rn + disp × 2 Longword: Rn + disp × 4
Indexed register indirect	@(R0, Rn)	Effective address is sum of register Rn and R0 contents. 	Rn + R0
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 Word: GBR + disp × 2 Longword: GBR + disp × 4
Indexed GBR indirect	@(R0, GBR)	Effective address is sum of register GBR and R0 contents. 	GBR + R0

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
PC-relative with displacement	@(disp:8, PC)	Effective address is PC 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 + disp × 2 Longword: PC&H'FFFFFFFC + disp × 4
PC-relative	disp:8	Effective address is PC with 8-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + disp × 2
	disp:12	Effective address is PC with 12-bit displacement disp added after being sign-extended and multiplied by 2 	PC + disp × 2
	Rn	Effective address is sum of PC and Rn. 	PC + Rn

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 addressing modes with displacement (disp) as shown below, the assembler description in this manual indicates the value before it is scaled (x1, x2, or x4) according to the operand size to clarify the LSI operation. For details on assembler description, refer to the description rules in each assembler.

@ (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

2.5.3 Instruction Formats

Table 2.4 shows the instruction formats, and the meaning of the source and destination operands, for instructions executed by the CPU core. The meaning of the operands depends on the instruction code. The following symbols are used in the table.

xxxx: Instruction code

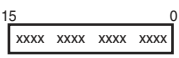
mmmm: Source register

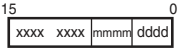
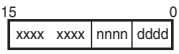
nnnn: Destination register

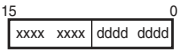
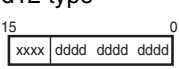
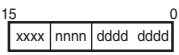
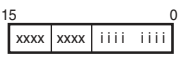
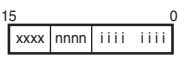
iiii: Immediate data

dddd: Displacement

Table 2.4 CPU Instruction Formats

Instruction Format	Source Operand	Destination Operand	Sample Instruction	
0 type 	—	—	NOP	
n type 	—	nnnn: register direct	MOVT	Rn
	Control register or system register	nnnn: register direct	STS	MACH,Rn
	Control register or system register	nnnn: pre-decrement register indirect	STC.L	SR,@-Rn
m type 	mmmm: register direct	Control register or system register	LDC	Rm,SR
	mmmm: post-increment register indirect	Control register or system register	LDC.L	@Rm+,SR
	mmmm: register indirect	—	JMP	@Rm
	PC-relative using Rm	—	BRAF	Rm

Instruction Format	Source Operand	Destination Operand	Sample Instruction
nm type 	mmmm: register direct	nnnn: register direct	ADD Rm,Rn
	mmmm: register indirect	nnnn: register indirect	MOV.L Rm,@Rn
	mmmm: post-increment register indirect (multiply-and-accumulate operation) nnnn: * post-increment register indirect (multiply-and-accumulate operation)	MACH, MACL	MAC.W @Rm+,@Rn+
	mmmm: post-increment register indirect	nnnn: register direct	MOV.L @Rm+,Rn
	mmmm: register direct	nnnn: pre-decrement register indirect	MOV.L Rm,@-Rn
	mmmm: register direct	nnnn: indexed register indirect	MOV.L Rm,@(R0,Rn)
md type 	mmmmdddd: register indirect with displacement	R0 (register direct)	MOV.B @(disp,Rm),R0
nd4 type 	R0 (register direct)	nnnndddd: register indirect with displacement	MOV.B R0,@(disp,Rn)
nmd type 	mmmm: register direct	nnnndddd: register indirect with displacement	MOV.L Rm,@(disp,Rn)
	mmmmdddd: register indirect with displacement	nnnn: register direct	MOV.L @(disp,Rm),Rn

Instruction Format	Source Operand	Destination Operand	Sample Instruction
d type 	dddddddd: GBR indirect with displacement	R0 (register direct)	MOV.L @(disp,GBR),R0
	R0 (register direct)	dddddddd: GBR indirect with displacement	MOV.L R0,@(disp,GBR)
	dddddddd: PC-relative with displacement	R0 (register direct)	MOVA @(disp,PC),R0
	dddddddd: PC-relative	—	BF label
d12 type 	ddddddddddddd: PC-relative	—	BRA label (label=disp+PC)
nd8 type 	dddddddd: PC-relative with displacement	nnnn: register direct	MOV.L @(disp,PC),Rn
i type 	iiiiiiii: immediate	Indexed GBR indirect	AND.B #imm,@(R0,GBR)
	iiiiiiii: immediate	R0 (register direct)	AND #imm,R0
	iiiiiiii: immediate	—	TRAPA #imm
ni type 	iiiiiiii: immediate	nnnn: register direct	ADD #imm,Rn

Note: * In multiply-and-accumulate instructions, nnnn is the source register.

2.6 Instruction Set

2.6.1 Instruction Set Based on Functions

Table 2.5 shows the instructions classified by function.

Table 2.5 CPU Instruction Types

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Data transfer instructions	5	MOV	Data transfer	39
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Upper/lower swap	
		XTRCT	Extraction of middle of linked registers	
Arithmetic operation instructions	21	ADD	Binary addition	33
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Signed division initialization	
		DIV0U	Unsigned division initialization	
		DMULS	Signed double-precision multiplication	
		DMULU	Unsigned double-precision multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply-and-accumulate, double-precision multiply-and-accumulate	
		MUL	Double-precision multiplication (32×32 bits)	

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Arithmetic operation instructions	21	MULS	Signed multiplication (16 × 16 bits)	33
		MULU	Unsigned multiplication (16 × 16 bits)	
		NEG	Sign inversion	
		NEGC	Sign inversion with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow	
Logic operation instructions	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit setting	
		TST	Logical AND and T bit setting	
		XOR	Exclusive logical OR	
Shift instructions	12	ROTCL	1-bit left shift with T bit	16
		ROTCR	1-bit right shift with T bit	
		ROTL	1-bit left shift	
		ROTR	1-bit right shift	
		SHAD	Arithmetic dynamic shift	
		SHAL	Arithmetic 1-bit left shift	
		SHAR	Arithmetic 1-bit right shift	
		SHLD	Logical dynamic shift	
		SHLL	Logical 1-bit left shift	
		SHLLn	Logical n-bit left shift	
		SHLR	Logical 1-bit right shift	
		SHLRn	Logical n-bit right shift	

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Branch instructions	9	BF	Conditional branch, delayed conditional branch (T = 0)	11
		BT	Conditional branch, delayed conditional branch (T = 1)	
		BRA	Unconditional branch	
		BRAF	Unconditional branch	
		BSR	Branch to subroutine procedure	
		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	
System control instructions	15	CLRMAC	MAC register clear	75
		CLRS	S bit clear	
		CLRT	T bit clear	
		LDC	Load into control register	
		LDS	Load into system register	
		LDTLB	PTEH/PTEL load into TLB	
		NOP	No operation	
		PREF	Data prefetch to cache	
		RTE	Return from exception handling	
		SETS	S bit setting	
		SETT	T bit setting	
		SLEEP	Transition to power-down mode	
		STC	Store from control register	
		STS	Store from system register	
		TRAPA	Trap exception handling	
Total:	68			188

The instruction code, operation, and number of execution states of the CPU instructions are shown in the following tables, classified by instruction type, using the format shown below.

Instruction	Instruction Code	Operation	Privilege	Execution States	T Bit
Indicated by mnemonic.	Indicated in MSB ↔ LSB order.	Indicates summary of operation.	Indicates a privileged instruction.	Value when no wait states are inserted*1	Value of T bit after instruction is executed
Explanation of Symbols OP.Sz SRC, DEST OP: Operation code Sz: Size SRC: Source DEST: Destination Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement	Explanation of Symbols mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 ..... 1111: R15 iiii: Immediate data dddd: Displacement*2	Explanation of Symbols →, ←: Transfer direction (xx): Memory operand M/Q/T: Flag bits in SR &: Logical AND of each bit |: Logical OR of each bit ^: Exclusive logical OR of each bit ~: Logical NOT of each bit <<n: n-bit left shift >>n: n-bit right shift		Explanation of Symbols —: No change	

Notes: 1. The table shows the minimum number of execution states. In practice, the number of instruction execution states will be increased in cases such as the following:

a. When there is a conflict between an instruction fetch and a data access

b. When the destination register of a load instruction (memory → register) is also used by the following instruction

2. Scaled (x1, x2, or x4) according to the instruction operand size, etc.

Table 2.6 Data Transfer Instructions

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
MOV	#imm,Rn	1110nnnniiiiiiii	imm → Sign extension → Rn	–	1	–
MOV.W	@(disp,PC),Rn	1001nnnndddddddd	(disp x 2+PC)→Sign extension → Rn	–	1	–
MOV.L	@(disp,PC),Rn	1101nnnndddddddd	(disp x 4+PC)→Rn	–	1	–
MOV	Rm,Rn	0110nnnnmmmm0011	Rm→Rn	–	1	–
MOV.B	Rm,@Rn	0010nnnnmmmm0000	Rm→(Rn)	–	1	–
MOV.W	Rm,@Rn	0010nnnnmmmm0001	Rm→(Rn)	–	1	–
MOV.L	Rm,@Rn	0010nnnnmmmm0010	Rm→(Rn)	–	1	–
MOV.B	@Rm,Rn	0110nnnnmmmm0000	(Rm)→Sign extension→Rn	–	1	–
MOV.W	@Rm,Rn	0110nnnnmmmm0001	(Rm)→Sign extension→Rn	–	1	–
MOV.L	@Rm,Rn	0110nnnnmmmm0010	(Rm)→Rn	–	1	–
MOV.B	Rm,@–Rn	0010nnnnmmmm0100	Rn–1→Rn, Rm→(Rn)	–	1	–
MOV.W	Rm,@–Rn	0010nnnnmmmm0101	Rn–2→Rn, Rm→(Rn)	–	1	–
MOV.L	Rm,@–Rn	0010nnnnmmmm0110	Rn–4→Rn, Rm→(Rn)	–	1	–
MOV.B	@Rm+,Rn	0110nnnnmmmm0100	(Rm)→Sign extension→Rn, Rm+1→Rm	–	1	–
MOV.W	@Rm+,Rn	0110nnnnmmmm0101	(Rm)→Sign extension→Rn, Rm+2→Rm	–	1	–
MOV.L	@Rm+,Rn	0110nnnnmmmm0110	(Rm)→Rn, Rm+4→Rm	–	1	–
MOV.B	R0,@(disp,Rn)	10000000nnnndddd	R0→(disp+Rn)	–	1	–
MOV.W	R0,@(disp,Rn)	10000001nnnndddd	R0→(disp x 2+Rn)	–	1	–
MOV.L	Rm,@(disp,Rn)	0001nnnnmmmmdddd	Rm→(disp x 4+Rn)	–	1	–
MOV.B	@(disp,Rm),R0	10000100mmmmdddd	(disp+Rm)→Sign extension→R0	–	1	–
MOV.W	@(disp,Rm),R0	10000101mmmmdddd	(disp x 2+Rm)→Sign extension→R0	–	1	–
MOV.L	@(disp,Rm),Rn	0101nnnnmmmmdddd	(disp x 4+Rm)→Rn	–	1	–
MOV.B	Rm,@(R0,Rn)	0000nnnnmmmm0100	Rm→(R0+Rn)	–	1	–
MOV.W	Rm,@(R0,Rn)	0000nnnnmmmm0101	Rm→(R0+Rn)	–	1	–
MOV.L	Rm,@(R0,Rn)	0000nnnnmmmm0110	Rm→(R0+Rn)	–	1	–

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
MOV.B	@(R0,Rm),Rn	0000nnnnmmmm1100	(R0+Rm)→Sign extension→Rn	–	1	–
MOV.W	@(R0,Rm),Rn	0000nnnnmmmm1101	(R0+Rm)→Sign extension→Rn	–	1	–
MOV.L	@(R0,Rm),Rn	0000nnnnmmmm1110	(R0+Rm)→Rn	–	1	–
MOV.B	R0,@(disp,GBR)	11000000dddddddd	R0→(disp+GBR)	–	1	–
MOV.W	R0,@(disp,GBR)	11000001dddddddd	R0→(disp x 2+GBR)	–	1	–
MOV.L	R0,@(disp,GBR)	11000010dddddddd	R0→(disp x 4+GBR)	–	1	–
MOV.B	@(disp,GBR),R0	11000100dddddddd	(disp+GBR)→Sign extension→R0	–	1	–
MOV.W	@(disp,GBR),R0	11000101dddddddd	(disp x 2+GBR)→Sign extension→R0	–	1	–
MOV.L	@(disp,GBR),R0	11000110dddddddd	(disp x 4+GBR)→R0	–	1	–
MOVA	@(disp,PC),R0	11000111dddddddd	disp x 4+PC→R0	–	1	–
MOVT	Rn	0000nnnn00101001	T→Rn	–	1	–
SWAP.B	Rm,Rn	0110nnnnmmmm1000	Rm→Swap lowest two bytes→Rn	–	1	–
SWAP.W	Rm,Rn	0110nnnnmmmm1001	Rm→Swap two consecutive words→Rn	–	1	–
XTRCT	Rm,Rn	0010nnnnmmmm1101	Rm: Middle 32 bits of Rn →Rn	–	1	–

Table 2.7 Arithmetic Operation Instructions

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
ADD	Rm,Rn	0011nnnnmmmm1100	Rn+Rm→Rn	–	1	–
ADD	#imm,Rn	0111nnnniiiiiiii	Rn+imm→Rn	–	1	–
ADDC	Rm,Rn	0011nnnnmmmm1110	Rn+Rm+T→Rn, Carry→T	–	1	Carry
ADDV	Rm,Rn	0011nnnnmmmm1111	Rn+Rm→Rn, Overflow→T	–	1	Overflow
CMP/EQ	#imm,R0	10001000iiiiiiii	If R0 = imm, 1 → T	–	1	Comparison result
CMP/EQ	Rm,Rn	0011nnnnmmmm0000	If Rn = Rm, 1 → T	–	1	Comparison result
CMP/HS	Rm,Rn	0011nnnnmmmm0010	If Rn ≥ Rm with unsigned data, 1 → T	–	1	Comparison result
CMP/GE	Rm,Rn	0011nnnnmmmm0011	If Rn ≥ Rm with signed data, 1 → T	–	1	Comparison result
CMP/HI	Rm,Rn	0011nnnnmmmm0110	If Rn > Rm with unsigned data, 1 → T	–	1	Comparison result
CMP/GT	Rm,Rn	0011nnnnmmmm0111	If Rn > Rm with signed data, 1 → T	–	1	Comparison result
CMP/PL	Rn	0100nnnn00010101	If Rn ≥ 0, 1 → T	–	1	Comparison result
CMP/PZ	Rn	0100nnnn00010001	If Rn > 0, 1 → T	–	1	Comparison result
CMP/STR	Rm,Rn	0010nnnnmmmm1100	If Rn and Rm have an equivalent byte, 1 → T	–	1	Comparison result
DIV1	Rm,Rn	0011nnnnmmmm0100	Single-step division (Rn/Rm)	–	1	Calculation result
DIV0S	Rm,Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	–	1	Calculation result
DIV0U		0000000000011001	0 → M/Q/T	–	1	0
DMULS.L	Rm,Rn	0011nnnnmmmm1101	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	–	2 (to 5)*	–
DMULU.L	Rm,Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	–	2 (to 5)*	–
DT	Rn	0100nnnn00010000	Rn – 1 → Rn, if Rn = 0, 1 → T, else 0 → T	–	1	Comparison result

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
EXTS.B	Rm,Rn	0110nnnnmmmm1110	A byte in Rm is sign-extended → Rn	–	1	–
EXTS.W	Rm,Rn	0110nnnnmmmm1111	A word in Rm is sign-extended → Rn	–	1	–
EXTU.B	Rm,Rn	0110nnnnmmmm1100	A byte in Rm is zero-extended → Rn	–	1	–
EXTU.W	Rm,Rn	0110nnnnmmmm1101	A word in Rm is zero-extended → Rn	–	1	–
MAC.L	@Rm+, @Rn+	0000nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC, Rn + 4 → Rn, Rm + 4 → Rm 32 × 32 + 64 → 64 bits	–	2 (to 5)*	–
MAC.W	@Rm+, @Rn+	0100nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC, Rn + 2 → Rn, Rm + 2 → Rm 16 × 16 + 64 → 64 bits	–	2 (to 5)*	–
MUL.L	Rm,Rn	0000nnnnmmmm0111	Rn × Rm → MACL 32 × 32 → 32 bits	–	2 (to 5) *	–
MULS.W	Rm,Rn	0010nnnnmmmm1111	Signed operation of Rn × Rm → MACL 16 × 16 → 32 bits	–	1(to 3)*	–
MULU.W	Rm,Rn	0010nnnnmmmm1110	Unsigned operation of Rn × Rm → MACL 16 × 16 → 32 bits	–	1(to 3)*	–
NEG	Rm,Rn	0110nnnnmmmm1011	0–Rm→Rn	–	1	–
NEGC	Rm,Rn	0110nnnnmmmm1010	0–Rm–T→Rn, Borrow→T	–	1	Borrow
SUB	Rm,Rn	0011nnnnmmmm1000	Rn–Rm→Rn	–	1	–
SUBC	Rm,Rn	0011nnnnmmmm1010	Rn–Rm–T→Rn, Borrow →T	–	1	Borrow
SUBV	Rm,Rn	0011nnnnmmmm1011	Rn–Rm→Rn, Underflow→T	–	1	Underflow

Note: * The number of execution cycles indicated within the parentheses ( ) are required when the operation result is read from the MACH/MACL register immediately after the instruction.

Table 2.8 Logic Operation Instructions

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
AND	Rm,Rn	0010nnnnmmmm1001	Rn & Rm → Rn	–	1	–
AND	#imm,R0	11001001iiiiiiii	R0 & imm → R0	–	1	–
AND.B	#imm,@(R0, GBR)	11001101iiiiiiii	(R0+GBR) & imm → (R0+GBR)	–	3	–
NOT	Rm,Rn	0110nnnnmmmm0111	~Rm → Rn	–	1	–
OR	Rm,Rn	0010nnnnmmmm1011	Rn | Rm → Rn	–	1	–
OR	#imm,R0	11001011iiiiiiii	R0 | imm → R0	–	1	–
OR.B	#imm,@(R0, GBR)	11001111iiiiiiii	(R0+GBR) | imm → (R0+GBR)	–	3	–
TAS.B	@Rn	0100nnnn00011011	If (Rn) is 0, 1 → T; 1 → MSB of (Rn)	–	4	Test result
TST	Rm,Rn	0010nnnnmmmm1000	Rn & Rm; if the result is 0, 1 → T	–	1	Test result
TST	#imm,R0	11001000iiiiiiii	R0 & imm; if the result is 0, 1 → T	–	1	Test result
TST.B	#imm,@(R0, GBR)	11001100iiiiiiii	(R0 + GBR) & imm; if the result is 0, 1 → T	–	3	Test result
XOR	Rm,Rn	0010nnnnmmmm1010	Rn ^ Rm → Rn	–	1	–
XOR	#imm,R0	11001010iiiiiiii	R0 ^ imm → R0	–	1	–
XOR.B	#imm,@(R0, GBR)	11001110iiiiiiii	(R0+GBR) ^ imm → (R0+GBR)	–	3	–

Table 2.9 Shift Instructions

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
ROTL	Rn	0100nnnn00000100	T←Rn←MSB	–	1	MSB
ROTR	Rn	0100nnnn00000101	LSB→Rn→T	–	1	LSB
ROTCL	Rn	0100nnnn00100100	T←Rn←T	–	1	MSB
ROTCR	Rn	0100nnnn00100101	T→Rn→T	–	1	LSB
SHAD	Rm, Rn	0100nnnnmmmm1100	Rn ≥ 0: Rn << Rm → Rn Rn < 0: Rn >> Rm → [MSB → Rn]	–	1	–
SHAL	Rn	0100nnnn00100000	T←Rn←0	–	1	MSB
SHAR	Rn	0100nnnn00100001	MSB→Rn→T	–	1	LSB
SHLD	Rm, Rn	0100nnnnmmmm1101	Rm ≥ 0: Rn << Rm → Rn Rm < 0: Rn >> Rm → [0 → Rn]	–	1	–
SHLL	Rn	0100nnnn00000000	T←Rn←0	–	1	MSB
SHLR	Rn	0100nnnn00000001	0→Rn→T	–	1	LSB
SHLL2	Rn	0100nnnn00001000	Rn<<2 → Rn	–	1	–
SHLR2	Rn	0100nnnn00001001	Rn>>2 → Rn	–	1	–
SHLL8	Rn	0100nnnn00011000	Rn<<8 → Rn	–	1	–
SHLR8	Rn	0100nnnn00011001	Rn>>8 → Rn	–	1	–
SHLL16	Rn	0100nnnn00101000	Rn<<16 → Rn	–	1	–
SHLR16	Rn	0100nnnn00101001	Rn>>16 → Rn	–	1	–

Table 2.10 Branch Instructions

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
BF	disp	10001011dddddddd	If T = 0, disp × 2 + PC → PC; if T = 1, nop	–	3/1*	–
BF/S	disp	10001111dddddddd	Delayed branch, if T = 0, disp × 2 + PC → PC; if T = 1, nop	–	2/1*	–
BT	disp	10001001dddddddd	If T = 1, disp × 2 + PC → PC; if T = 0, nop	–	3/1*	–
BT/S	disp	10001101dddddddd	Delayed branch, if T = 1, disp × 2 + PC → PC; if T = 0, nop	–	2/1*	–
BRA	disp	1010dddddddddddd	Delayed branch, disp × 2 + PC → PC	–	2	–
BRAF	Rm	0000mmmm00100011	Delayed branch,Rm + PC → PC	–	2	–
BSR	disp	1011dddddddddddd	Delayed branch, PC → PR, disp × 2 + PC → PC	–	2	–
BSRF	Rm	0000mmmm00000011	Delayed branch, PC → PR, Rm + PC → PC	–	2	–
JMP	@Rm	0100mmmm00101011	Delayed branch, Rm → PC	–	2	–
JSR	@Rm	0100mmmm00001011	Delayed branch, PC → PR, Rm → PC	–	2	–
RTS		0000000000001011	Delayed branch, PR → PC	–	2	–

Note: * One state when the branch is not executed.

Table 2.11 System Control Instructions

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
CLRMAC		0000000000101000	0→MACH,MACL	–	1	–
CLRS		0000000001001000	0→S	–	1	–
CLRT		0000000000001000	0→T	–	1	0
LDC	Rm,SR	0100mmmm00001110	Rm→SR	√	6	LSB
LDC	Rm,GBR	0100mmmm00011110	Rm→GBR	–	4	–
LDC	Rm,VBR	0100mmmm00101110	Rm→VBR	√	4	–
LDC	Rm,SSR	0100mmmm00111110	Rm→SSR	√	4	–
LDC	Rm,SPC	0100mmmm01001110	Rm→SPC	√	4	–
LDC	Rm,R0_BANK	0100mmmm10001110	Rm→R0_BANK	√	4	–
LDC	Rm,R1_BANK	0100mmmm10011110	Rm→R1_BANK	√	4	–
LDC	Rm,R2_BANK	0100mmmm10101110	Rm→R2_BANK	√	4	–
LDC	Rm,R3_BANK	0100mmmm10111110	Rm→R3_BANK	√	4	–
LDC	Rm,R4_BANK	0100mmmm11001110	Rm→R4_BANK	√	4	–
LDC	Rm,R5_BANK	0100mmmm11011110	Rm→R5_BANK	√	4	–
LDC	Rm,R6_BANK	0100mmmm11101110	Rm→R6_BANK	√	4	–
LDC	Rm,R7_BANK	0100mmmm11111110	Rm→R7_BANK	√	4	–
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm)→SR, Rm+4→Rm	√	8	LSB
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm)→GBR, Rm+4→Rm	–	4	–
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm)→VBR, Rm+4→Rm	√	4	–
LDC.L	@Rm+,SSR	0100mmmm00110111	(Rm)→SSR,Rm+4→Rm	√	4	–
LDC.L	@Rm+,SPC	0100mmmm01000111	(Rm)→SPC,Rm+4→Rm	√	4	–
LDC.L	@Rm+, R0_BANK	0100mmmm10000111	(Rm)→R0_BANK,Rm+4→Rm	√	4	–
LDC.L	@Rm+, R1_BANK	0100mmmm10010111	(Rm)→R1_BANK,Rm+4→Rm	√	4	–
LDC.L	@Rm+, R2_BANK	0100mmmm10100111	(Rm)→R2_BANK,Rm+4→Rm	√	4	–
LDC.L	@Rm+, R3_BANK	0100mmmm10110111	(Rm)→R3_BANK, Rm+4→Rm	√	4	–
LDC.L	@Rm+, R4_BANK	0100mmmm11000111	(Rm)→R4_BANK, Rm+4→Rm	√	4	–

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
LDC.L	@Rm+, R5_BANK	0100mmmm11010111	(Rm)→R5_BANK, Rm+4→Rm	√	4	–
LDC.L	@Rm+, R6_BANK	0100mmmm11100111	(Rm)→R6_BANK, Rm+4→Rm	√	4	–
LDC.L	@Rm+, R7_BANK	0100mmmm11110111	(Rm)→R7_BANK, Rm+4→Rm	√	4	–
LDS	Rm,MACH	0100mmmm00001010	Rm→MACH	–	1	–
LDS	Rm,MACL	0100mmmm00011010	Rm→MACL	–	1	–
LDS	Rm,PR	0100mmmm00101010	Rm→PR	–	1	–
LDS.L	@Rm+,MACH	0100mmmm00000110	(Rm)→MACH, Rm+4→Rm	–	1	–
LDS.L	@Rm+,MACL	0100mmmm00010110	(Rm)→MACL, Rm+4→Rm	–	1	–
LDS.L	@Rm+,PR	0100mmmm00100110	(Rm)→PR, Rm+4→Rm	–	1	–
LDTLB		0000000000111000	PTEH/PTEL→TLB	√	1	–
NOP		0000000000001001	No operation	–	1	–
PREF	@Rm	0000mmmm10000011	(Rm) → cache	–	1	–
RTE		0000000000101011	Delayed branch, SSR → SR, SPC → PC	√	5	–
SETS		0000000001011000	1→S	–	1	–
SETT		0000000000011000	1→T	–	1	1
SLEEP		0000000000011011	Sleep	√	4*[1]	–
STC	SR,Rn	0000nnnn00000010	SR→Rn	√	1	–
STC	GBR,Rn	0000nnnn00010010	GBR→Rn	–	1	–
STC	VBR,Rn	0000nnnn00100010	VBR→Rn	√	1	–
STC	SSR, Rn	0000nnnn00110010	SSR→Rn	√	1	–
STC	SPC,Rn	0000nnnn01000010	SPC→Rn	√	1	–
STC	R0_BANK,Rn	0000nnnn10000010	R0_BANK→Rn	√	1	–
STC	R1_BANK,Rn	0000nnnn10010010	R1_BANK→Rn	√	1	–
STC	R2_BANK,Rn	0000nnnn10100010	R2_BANK→Rn	√	1	–
STC	R3_BANK,Rn	0000nnnn10110010	R3_BANK→Rn	√	1	–
STC	R4_BANK,Rn	0000nnnn11000010	R4_BANK→Rn	√	1	–
STC	R5_BANK,Rn	0000nnnn11010010	R5_BANK→Rn	√	1	–
STC	R6_BANK,Rn	0000nnnn11100010	R6_BANK→Rn	√	1	–

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
STC	R7_BANK,Rn	0000nnnn11110010	R7_BANK→Rn	√	1	–
STC.L	SR,@–Rn	0100nnnn00000011	Rn–4→Rn, SR→(Rn)	√	1	–
STC.L	GBR,@–Rn	0100nnnn00010011	Rn–4→Rn, GBR→(Rn)	–	1	–
STC.L	VBR,@–Rn	0100nnnn00100011	Rn–4→Rn, VBR→(Rn)	√	1	–
STC.L	SSR,@–Rn	0100nnnn00110011	Rn–4→Rn, SSR→(Rn)	√	1	–
STC.L	SPC,@–Rn	0100nnnn01000011	Rn–4→Rn, SPC→(Rn)	√	1	–
STC.L	R0_BANK,@–Rn	0100nnnn10000011	Rn–4→Rn, R0_BANK→(Rn)	√	1	–
STC.L	R1_BANK,@–Rn	0100nnnn10010011	Rn–4→Rn, R1_BANK→(Rn)	√	1	–
STC.L	R2_BANK,@–Rn	0100nnnn10100011	Rn–4→Rn, R2_BANK→(Rn)	√	1	–
STC.L	R3_BANK,@–Rn	0100nnnn10110011	Rn–4→Rn, R3_BANK→(Rn)	√	1	–
STC.L	R4_BANK,@–Rn	0100nnnn11000011	Rn–4→Rn, R4_BANK→(Rn)	√	1	–
STC.L	R5_BANK,@–Rn	0100nnnn11010011	Rn–4→Rn, R5_BANK→(Rn)	√	1	–
STC.L	R6_BANK,@–Rn	0100nnnn11100011	Rn–4→Rn, R6_BANK→(Rn)	√	1	–
STC.L	R7_BANK,@–Rn	0100nnnn11110011	Rn–4→Rn, R7_BANK→(Rn)	√	1	–
STS	MACH,Rn	0000nnnn00001010	MACH→Rn	–	1	–
STS	MACL,Rn	0000nnnn00011010	MACL→Rn	–	1	–
STS	PR,Rn	0000nnnn00101010	PR→Rn	–	1	–
STS.L	MACH,@–Rn	0100nnnn00000010	Rn–4→Rn, MACH→(Rn)	–	1	–
STS.L	MACL,@–Rn	0100nnnn00010010	Rn–4→Rn, MACL→(Rn)	–	1	–
STS.L	PR,@–Rn	0100nnnn00100010	Rn–4→Rn, PR→(Rn)	–	1	–
TRAPA	#imm	11000011iiiiiiii	Unconditional trap exception occurs*2	–	8	–

Notes: The table shows the minimum number of clocks required for execution. In practice, the number of execution cycles will be increased in the following conditions.

a. If there is a conflict between an instruction fetch and a data access

b. If the destination register of a load instruction (memory → register) is also used by the following instruction.

For addressing modes with displacement (disp) as shown below, the assembler description in this manual indicates the value before it is scaled (x 1, x 2, or x 4) according to the operand size to clarify the LSI operation. For details on assembler description, refer to the description rules in each assembler.

@ (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

1. Number of states before the chip enters the sleep state.
2. For details, refer to section 7, Exception Handling.

2.6.2 Operation Code Map

Table 2.12 shows the operation code map.

Table 2.12 Operation Code Map

Instruction Code				Fx: 0000		Fx: 0001		Fx: 0010		Fx: 0011 to 1111	
MSB			LSB	MD: 00		MD: 01		MD: 10		MD: 11	
0000	Rn	Fx	0000								
0000	Rn	Fx	0001								
0000	Rn	00MD	0010	STC	SR, Rn	STC	GBR, Rn	STC	VBR, Rn	STC	SSR, Rn
0000	Rn	01MD	0010	STC	SPC, Rn						
0000	Rn	10MD	0010	STC	R0_BANK, Rn	STC	R1_BANK, Rn	STC	R2_BANK, Rn	STC	R3_BANK, Rn
0000	Rn	11MD	0010	STC	R4_BANK, Rn	STC	R5_BANK, Rn	STC	R6_BANK, Rn	STC	R7_BANK, Rn
0000	Rm	00MD	0011	BSRF	Rm			BRA	Rm		
0000	Rm	10MD	0011	PREF	@Rm						
0000	Rn	Rm	01MD	MOV.B	Rm, @(R0, Rn)	MOV.W	Rm, @(R0, Rn)	MOV.L	Rm,@(R0, Rn)	MUL.L	Rm, Rn
0000	0000	00MD	1000	CLRT		SETT		CLRMAC		LDTLB	
0000	0000	01MD	1000	CLRS		SETS					
0000	0000	Fx	1001	NOP		DIV0U					
0000	0000	Fx	1010								
0000	0000	Fx	1011	RTS		SLEEP		RTE			
0000	Rn	Fx	1000								
0000	Rn	Fx	1001					MOVT	Rn		
0000	Rn	Fx	1010	STS	MACH, Rn	STS	MACL, Rn	STS	PR, Rn		
0000	Rn	Fx	1011								
0000	Rn	Rm	11MD	MOV. B	@(R0, Rm), Rn	MOV.W	@(R0, Rm), Rn	MOV.L	@(R0, Rm), Rn	MAC.L	@Rm+,@Rn+
0001	Rn	Rm	disp	MOV.L	Rm, @(disp:4, Rn)						
0010	Rn	Rm	00MD	MOV.B	Rm, @Rn	MOV.W	Rm, @Rn	MOV.L	Rm, @Rn		
0010	Rn	Rm	01MD	MOV.B	Rm, @–Rn	MOV.W	Rm, @–Rn	MOV.L	Rm, @–Rn	DIV0S	Rm, Rn
0010	Rn	Rm	10MD	TST	Rm, Rn	AND	Rm, Rn	XOR	Rm, Rn	OR	Rm, Rn

Instruction Code				Fx: 0000		Fx: 0001		Fx: 0010		Fx: 0011 to 1111	
MSB			LSB	MD: 00		MD: 01		MD: 10		MD: 11	
0010	Rn	Rm	11MD	CMP/STR	Rm, Rn	XTRCT	Rm, Rn	MULU.W	Rm, Rn	MULSW	Rm, Rn
0011	Rn	Rm	00MD	CMP/EQ	Rm, Rn			CMP/HS	Rm, Rn	CMP/GE	Rm, Rn
0011	Rn	Rm	01MD	DIV1	Rm, Rn	DMULU.L	Rm,Rn	CMP/HI	Rm, Rn	CMP/GT	Rm, Rn
0011	Rn	Rm	10MD	SUB	Rm, Rn			SUBC	Rm, Rn	SUBV	Rm, Rn
0011	Rn	Rm	11MD	ADD	Rm, Rn	DMULS.L	Rm,Rn	ADDC	Rm, Rn	ADDV	Rm, Rn
0100	Rn	Fx	0000	SHLL	Rn	DT	Rn	SHAL	Rn		
0100	Rn	Fx	0001	SHLR	Rn	CMP/PZ	Rn	SHAR	Rn		
0100	Rn	Fx	0010	STS.L	MACH, @–Rn	STS.L	MACL, @–Rn	STS.L	PR, @–Rn		
0100	Rn	00MD	0011	STC.L	SR, @–Rn	STC.L	GBR, @–Rn	STC.L	VBR, @–Rn	STC.L	SSR, @–Rn
0100	Rn	01MD	0011	STC.L	SPC, @–Rn						
0100	Rn	10MD	0011	STC.L	R0_BANK, @–Rn	STC.L	R1_BANK, @–Rn	STC.L	R2_BANK, @–Rn	STC.L	R3_BANK, @–Rn
0100	Rn	11MD	0011	STC.L	R4_BANK, @–Rn	STC.L	R5_BANK, @–Rn	STC.L	R6_BANK, @–Rn	STC.L	R7_BANK, @–Rn
0100	Rn	Fx	0100	ROTL	Rn			ROTCL	Rn		
0100	Rn	Fx	0101	ROTR	Rn	CMP/PL	Rn	ROTCR	Rn		
0100	Rm	Fx	0110	LDS.L	@Rm+, MACH	LDS.L	@Rm+, MACL	LDS.L	@Rm+, PR		
0100	Rm	00MD	0111	LDC.L	@Rm+, SR	LDC.L	@Rm+, GBR	LDC.L	@Rm+, VBR	LDC.L	@Rm+, SSR
0100	Rm	01MD	0111	LDC.L	@Rm+, SPC						
0100	Rm	10MD	0111	LDC.L	@Rm+, R0_BANK	LDC.L	@Rm+, R1_BANK	LDC.L	@Rm+, R2_BANK	LDC.L	@Rm+, R3_BANK
0100	Rm	11MD	0111	LDC.L	@Rm+, R4_BANK	LDC.L	@Rm+, R5_BANK	LDC.L	@Rm+, R6_BANK	LDC.L	@Rm+, R7_BANK
0100	Rn	Fx	1000	SHLL2	Rn	SHLL8	Rn	SHLL16	Rn		
0100	Rn	Fx	1001	SHLR2	Rn	SHLR8	Rn	SHLR16	Rn		
0100	Rm	Fx	1010	LDS	Rm, MACH	LDS	Rm, MACL	LDS	Rm, PR		
0100	Rm/ Rn	Fx	1011	JSR	@Rm	TAS.B	@Rn	JMP	@Rm		
0100	Rn	Rm	1100	SHAD	Rm, Rn						
0100	Rn	Rm	1101	SHLD	Rm, Rn						

Instruction Code				Fx: 0000		Fx: 0001		Fx: 0010		Fx: 0011 to 1111	
MSB			LSB	MD: 00		MD: 01		MD: 10		MD: 11	
0100	Rm	00MD	1110	LDC	Rm, SR	LDC	Rm, GBR	LDC	Rm, VBR	LDC	Rm, SSR
0100	Rm	01MD	1110	LDC	Rm, SPC						
0100	Rm	10MD	1110	LDC	Rm, R0_BANK	LDC	Rm, R1_BANK	LDC	Rm, R2_BANK	LDC	Rm, R3_BANK
0100	Rm	11MD	1110	LDC	Rm, R4_BANK	LDC	Rm, R5_BANK	LDC	Rm, R6_BANK	LDC	Rm, R7_BANK
0100	Rn	Rm	1111	MAC.W	@Rm+, @Rn+						
0101	Rn	Rm	disp	MOV.L	@(disp:4, Rm), Rn						
0110	Rn	Rm	00MD	MOV.B	@Rm, Rn	MOV.W	@Rm, Rn	MOV.L	@Rm, Rn	MOV	Rm, Rn
0110	Rn	Rm	01MD	MOV.B	@Rm+, Rn	MOV.W	@Rm+, Rn	MOV.L	@Rm+, Rn	NOT	Rm, Rn
0110	Rn	Rm	10MD	SWAP.B	Rm, Rn	SWAP.W	Rm, Rn	NEGC	Rm, Rn	NEG	Rm, Rn
0110	Rn	Rm	11MD	EXTU.B	Rm, Rn	EXTU.W	Rm, Rn	EXTS.B	Rm, Rn	EXTS.W	Rm, Rn
0111	Rn	imm		ADD	# imm : 8, Rn						
1000	00MD	Rn	disp	MOV. B R0, @(disp: 4, Rn)		MOV. W R0, @(disp: 4, Rn)					
1000	01MD	Rm	disp	MOV.B @(disp:4, Rm), R0		MOV.W @(disp: 4, Rm), R0					
1000	10MD	imm/disp		CMP/EQ	#imm:8, R0	BT	disp: 8			BF	disp: 8
1000	11MD	imm/disp				BT/S	disp: 8			BF/S	disp: 8
1001	Rn	disp		MOV.W	@(disp : 8, PC), Rn						
1010	disp			BRA	disp: 12						
1011	disp			BSR	disp: 12						
1100	00MD	imm/disp		MOV.B R0, @(disp: 8, GBR)		MOV.W R0, @(disp: 8, GBR)		MOV.L R0, @(disp: 8, GBR)		TRAPA	#imm: 8
1100	01MD	disp		MOV.B @(disp: 8, GBR), R0		MOV.W @(disp: 8, GBR), R0		MOV.L @(disp: 8, GBR), R0		MOVA @(disp: 8, PC), R0	
1100	10MD	imm		TST	#imm: 8, R0	AND	#imm: 8, R0	XOR	#imm: 8, R0	OR	#imm: 8, R0
1100	11MD	imm		TST.B #imm: 8, @(R0, GBR)		AND.B #imm: 8, @(R0, GBR)		XOR.B #imm: 8, @(R0, GBR)		OR.B #imm: 8, @(R0, GBR)	
1101	Rn	disp		MOV.L	@(disp: 8, PC), Rn						
1110	Rn	imm		MOV	#imm:8, Rn						
1111	**********										

Note: For details, refer to the SH-3/SH-3H/SH3-DSP Software Manual.

Section 3 DSP Operating Unit

3.1 DSP Extended Functions

This LSI incorporates a DSP unit and X/Y memory directly connected to the DSP unit. This LSI supports the DSP extended function instruction sets needed to control the DSP unit and X/Y memory. The DSP extended function instructions are classified into four groups.

(1) Extended System Control Instructions for the CPU

If the DSP extended function is enabled, the following extended system control instructions can be used for the CPU.

- Repeat loop control instructions and repeat loop control register access instructions are added. Looped programs can be executed efficiently by using the zero-overhead repeat control unit. For details, refer to section 3.3, CPU Extended Instructions.
- Modulo addressing control instructions and control register access instructions are added. Function allows access to data with a circular structure. For details, refer to section 3.4, DSP Data Transfer Instructions.
- DSP unit register access instructions are added. Some of the DSP unit registers can be used in the same way as the CPU system registers. For details, refer to section 3.4, DSP Data Transfer Instructions.

(2) Data Transfer Instructions for Data Transfers between DSP Unit and On-Chip X/Y Memory

Data transfer instructions for data transfers between the DSP unit and on-chip X/Y memory are called double-data transfer instructions. Instruction codes for these double-transfer instructions are 16 bit codes as well as CPU instruction codes. These data transfer instructions perform data transfers between the DSP unit and on-chip X/Y memory that is directly connected to the DSP unit. These data transfer instructions can be described in combination with other DSP unit operation instructions. For details, refer to section 3.4, DSP Data Transfer Instructions.

(3) Data Transfer Instructions for Data Transfers between DSP Unit Registers and All Virtual Address Spaces

Data transfer instructions for data transfers between DSP unit registers and all virtual address spaces are called single-data transfer instructions. Instruction codes for the double-transfer instructions are 16 bit codes as well as CPU instruction codes. These data transfer instructions performs data transfers between the DSP unit registers and all virtual address spaces. For details, refer to section 3.4, DSP Data Transfer Instructions.

(4) DSP Unit Operation Instructions

DSP unit operation instructions are called DSP data operation instructions. These instructions are provided to execute digital signal processing operations at high speed using the DSP. Instruction codes for these instructions are 32 bits. The DSP data operation instruction fields consist of two fields: field A and field B. In field A, a function for double data transfer instructions can be described. In field B, ALU operation instructions and multiply instructions can be described. The instructions described in fields A and B can be executed in parallel. A maximum of four instructions (ALU operation, multiply, and two data transfers) can be executed in parallel. For details, refer to section 3.5, DSP Data Operation Instructions.

Notes: 1. 32-bit instruction codes are handled as two consecutive 16-bit instruction codes. Accordingly, 32-bit instruction codes can be assigned to a word boundary. 32-bit instruction codes must be stored in memory, upper word and lower word, in this order, in word units.

2. In little endian, the upper and lower words must be stored in memory as data to be accessed in word units.

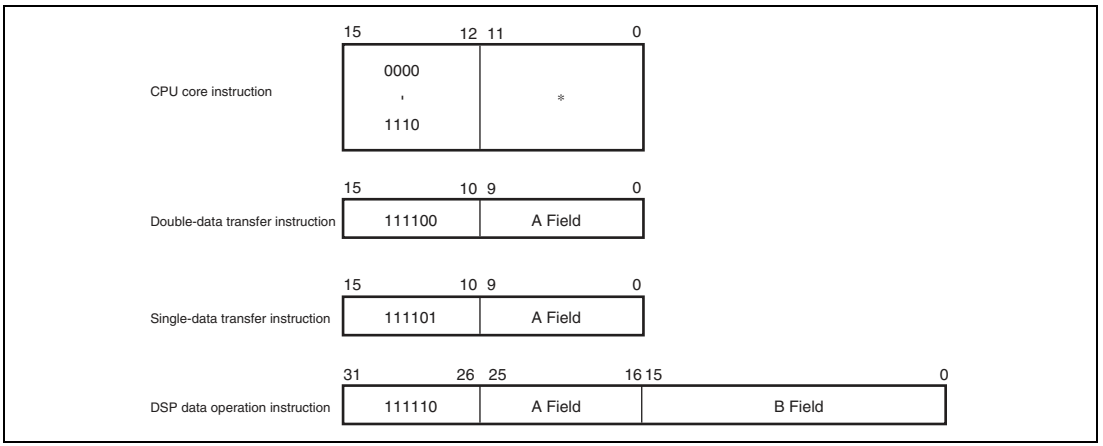


Figure 3.1 DSP Instruction Format

3.2 DSP Mode Resources

3.2.1 Processing Modes

The CPU processing modes can be extended using the mode bit (MD) and DSP bit (DSP) in the status register (SR), as shown below.

Table 3.1 CPU Processing Modes

			Description	
MD	DSP	Processing Mode	Access of Resources Protected in Privileged Mode or Privileged Instruction Execution	DSP Extended Functions
0	0	User mode	Prohibited	Invalid
0	1	User DSP mode	Prohibited	Valid
1	0	Privileged mode	Allowed	Invalid
1	1	Privileged DSP mode	Allowed	Valid

As shown above, the extension of the DSP function by the DSP bit can be specified independently of the control by the MD bit. Note, however, that the DSP bit can be modified only in privileged mode. Before the DSP bit is modified, a transition to privileged mode or privileged DSP mode is necessary.

3.2.2 DSP Mode Memory Map

In DSP mode, a part of the P2 area in the virtual address space can be accessed in user DSP mode. When this area is accessed in user DSP mode, this area is referred to as a Uxy area. X/Y memory is then assigned to this Uxy area. Accordingly, X/Y memory can also be accessed in user DSP mode.

Table 3.2 Virtual Address Space

Address Range	Name	Protection	Description
H'A5000000 to H'A5FFFFFF	P2/Uxy	Privileged or DSP	16-Mbyte physical address space, non-cacheable, non-address translatable Can be accessed in privileged mode, privileged DSP mode, and user DSP mode

3.2.3 CPU Register Sets

In DSP mode, the status register (SR) in the CPU unit is extended to add control bits and three control registers: a repeat start register (SR), repeat end register (RE), and modulo register (MOD) are added as control registers.

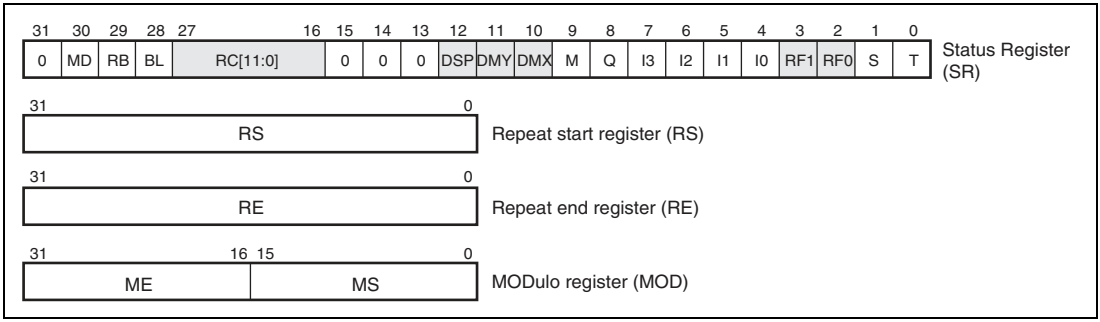


Figure 3.2 CPU Registers in DSP Mode

(1) Extension of Status Register (SR)

In DSP mode, the following control bits are added to the status register (SR). These added bits are called DSP extension bits. These DSP extension bits are valid only in DSP mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	—	—	For details, refer to section 2, CPU.
27 to 16	RC11 to RC0	All 0	R/W	Repeat Counter Holds the number of repeat times in order to perform loop control, and can be modified in privileged mode, privileged DSP mode, or user DSP mode. At reset, this bit is initialized to 0. This bit is not affected in the exception handling state.
15 to 13	—	—	—	For details, refer to section 2, CPU.
12	DSP	0	R/W	DSP Bit Enables or disables the DSP extended functions. If this bit is set to 1, the DSP extended functions are enabled. This bit can be modified in privileged mode, privileged DSP mode, or user DSP mode. At reset, this bit is initialized to 0. This bit is not affected in the exception handling state.
11	MDY	0	R/W	Modulo Control Bits
10	MDX	0	R/W	Enable or disable modulo addressing for X/Y memory access. These bits can be modified in privileged mode, privileged DSP mode, or user DSP mode. At reset, these bits are initialized to 0. These bits are affected in the exception handling state.
9 to 4	—	—	—	For details, refer to section 2, CPU.
3	FR1	0	R/W	Repeat Flag Bits
2	FR0	0	R/W	Used by repeat control instructions. These bits can be modified in privileged mode, privileged DSP mode, or user DSP mode. At reset, these bits are initialized to 0. These bits are affected in the exception handling state.
1, 0	—	—	—	For details, refer to section 2, CPU.

Note: When data is written to the SR register, 0 should be written to bits that are specified as 0.

(2) Repeat Start Register (RS)

The repeat start register (RS) holds the start address of a loop repeat module that is controlled by the repeat function. This register can be accessed in DSP mode. At reset, the initial value of this register is undefined. This register is not affected in the exception handling state.

(3) Repeat End Register (RE)

The repeat end register (RE) holds the end address of a loop repeat module that is controlled by the repeat function. This register can be accessed in DSP mode. At reset, this register is initialized to 0. This register is not affected in the exception handling state.

(4) Modulo Register (MOD)

The modulo register stores the modulo end address and modulo start address for modulo addressing in upper and lower 16 bits. The upper and lower 16 bits of the modulo register are referred to as the ME register and MS register, respectively. This register can be accessed in DSP mode. At reset, the initial value of this register is undefined. This register is not affected in the exception handling state.

The above registers can be accessed by the control register load instruction (LDC) and store instruction (STC). Note that the LDC and STC instructions for the RS, RE, and MOD registers can be used only in privileged DSP mode and user DSP mode. The LDC and STC instruction for the SR register can be executed only when the MD bit is set to 1 or in user DSP mode. Note, however, that the LDC and STC instructions can modify only the RC11 to RC0, RF1 to RF0, DMX, and DMY bits in the SR, as described below.

- In user mode, if the LCD and STC instructions are used for the RS, an illegal instruction exception occurs.
- In privileged and privileged DSP modes, all SR bits can be modified.
- In user DSP mode, the SR can be read by the STC instruction.
- In user DSP mode, the LDC instruction can be issued to the SR but only the DSP extension bits can be modified.

Table 3.3 Operation of SR Bits in Each Processing Mode

Field	Privileged Mode MD = 1 & DSP = 0	User Mode MD = 0 & DSP = 0	Privileged DSP Mode MD = 1 & DSP = 1	User DSP Mode MD = 0 & DSP = 1	Access to DSP-Related Bit with Dedicated Instruction	Initial Value after Reset
MD	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		1
RB	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		1
BL	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		1
RC [11:0]	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	R: OK, L: OK	SETRC instruction	000000000000
DSP	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		0
DMY	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	R: OK, L: OK		0
DMX	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	R: OK, L: OK		0
Q	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		x
M	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		x
I[3:0]	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		1111
RF[1:0]	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	R: OK, L: OK	SETRC instruction	x
S	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		x
T	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		x

[Legend]

S: STC instruction

L: LDC instruction

OK: STC/LDC operation is enabled.

Invalid instruction: Exception occurs when an invalid instruction is executed.

NG: Previous value is retained. No change.

x: Undefined

Before entering the exception handling state, all bits including the DSP extension bits of the SR registers are saved in the SSR. Before returning from the exception handling, all bits including the DSP extension bits of the SR must be restored. If the repeat control must be recovered before entering the exception handling state, the RS and RE registers must be recovered to the value that existed before exception handling. In addition, if it is necessary to recover modulo control before entering the exception handling state, the MOD register must be recovered to the value that existed before exception handling.

3.2.4 DSP Registers

The DSP unit incorporates eight data registers (A0, A1, X0, X1, Y0, Y1, M0, and M1) and a status register (DSR). Figure 3.3 shows the DSP register configuration. These are 32-bit width registers with the exception of registers A0 and A1. Registers A0 and A1 include 8 guard bits (fields A0G and A1G), giving them a total width of 40 bits. The DSR register stores the DSP data operation result (zero, negative, others). The DSP register has a DC bit whose function is similar to the T bit in the CPU register. For details on DSR bits, refer to section 3.5, DSP Data Operation Instructions.

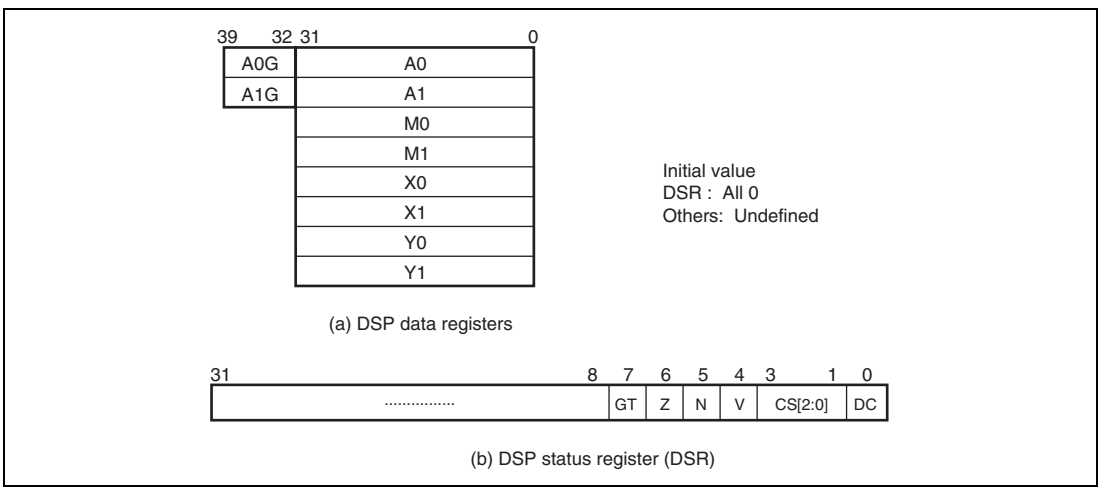


Figure 3.3 DSP Register Configuration

3.3 CPU Extended Instructions

3.3.1 DSP Repeat Control

In DSP mode, a specific function is provided to execute repeat loops efficiently. By using this function, loop programs can be executed without overhead caused by the compare and branch instructions.

(1) Examples of Repeat Loop Programs

Examples of repeat loop programs are shown below.

- Example 1: Repeat loop consisting of 4 or more instructions

```
            LDRS RptStart   ; Sets repeat start instruction address
                            to the RS register
            LDRE RptDtct +4         ; Sets (repeat detection instruction
                                    address + 4) to the RE register
            SETRC #4        ; Sets the number of repetitions (4) to
                            the RC[11:0] bits of the SR register
            Instr0          ; At least one instruction is required
                            from SETRC instruction to [Repeat start
                            instruction]
RptStart:   instr1          ; [Repeat start instruction]
            ... ...         ;
            ... ...         ;
RptDtct:    instr(N-3)     ;Three instruction prior to the repeat
                            end instruction is regarded as repeat
                            detection instruction
RptEnd2:    instr(N-2)      ;
RptEnd1:    instr(N-1)      ;
RptEnd:     instrN          ; [Repeat end instruction]
```

In the above program example, instructions from the RptStart address (instr1 instruction) to the RptEnd address (instrN instruction) are repeated four times. These repeated instructions in the program are called repeat loop. The start and end instructions of the repeat loop are called the repeat start instruction and repeat end instruction, respectively. The CPU sequentially executes instructions and starts repeat loop control if the CPU detects the completion of a specific instruction. This specific instruction is called the repeat detection instruction. In a repeat loop consisting of four or more instructions, an instruction three instructions prior to the repeat end instruction is regarded as the repeat detection instruction. In a repeat loop consisting of four or

more instructions, the same instruction is regarded as the RptStart instruction and RptDtct instruction.

To control the repeat loop, the DSP extended control registers, such as the RE register and RS register and the RC[11:0] and RF[1:0] bits of the SR register, are used. These registers can be specified by the LDRE, LDRS, and SETRC instructions.

- Repeat end register (RE)

  The RE register is specified by the LDRE instruction. The RE register specifies (repeat detection instruction address +4). In a repeat loop consisting of four or more instructions, an instruction three instructions prior to the repeat end instruction is regarded as the repeat detection instruction. A repeat loop consisting of three or less instructions is described later.
- Repeat start register (RS)

  The RE register is specified by the LDRS instruction. In a repeat loop consisting of 4 or more instructions, the RS register specifies the repeat start instruction address. In a repeat loop consisting of three or less instructions, a specific address is specified in the RS. This is described later.
- Repeat counter (RC[11:0] bits of the SR)

  The repeat counter is specifies the number of repetitions by the SETRC instruction. During repeat loop execution, the RC holds the remaining number of repetitions.
- Repeat flags (RF[1:0] bits of the SR)

  The repeat flags are automatically specified according to the RS and RE register values during SETRC instruction execution. The repeat flags store information on the number of instructions included in the repeat loop. Normally, the user cannot modify the repeat flag values.

The CPU always executes instructions by comparing the RE register to program counter values. Because the PC stores (the current instruction address +4), if the RE matches the PC during repeat instruction detection execution, a repeat detection instruction can be detected. If a repeat detection instruction is executed without branching and if RC[11:0] > 0, then repeat control is performed. If RC[11:0] ≥ 2 when the repeat end instruction is completed, the RC[11:0] is decremented by 1 and then control is passed to the address specified by the RS register.

Examples 2 to 4 show program examples of the repeat loop consisting of three instructions, two instructions, and one instruction, respectively. In these examples, an instruction immediately prior to the repeat start instruction is regarded as a repeat detection instruction. The RS register specifies the specific value that indicates the number of repeat instructions.

- Example 2: Repeat loop consisting of three instructions

```
        LDRS RptDtct +4     ; Sets (repeat detection instruction
                              address + 4) to the RS register
        LDRE RptDtct +4     ; Sets (repeat detection instruction
                              address + 4) to the RE register
        SETRC #4            ; Sets the number of repetitions (4) to
                              the RC[11:0] bits of the SR register
                            ; If RE-RS==0 during SETRC instruction
                              execution, the repeat loop is regarded
                              as three-instruction repeat.
RptDtct:  instr0            ; An instruction prior to the Repeat
                              start instruction is regarded as a
                              repeat detection instruction.
RptStart: instr1            ; [Repeat start instruction]
          Instr2            ;
RptEnd:   instr3            ; [Repeat end instruction]
```

- Example 3: Repeat loop consisting of two instructions

```
          LDRS RptDtct +6; Sets (repeat detection instruction
                            address + 6) to the RS register
          LDRE RptDtct +4       ; Sets (repeat detection instruction
                                  address + 4) to the RE register
          SETRC #4          ; Sets the number of repetitions (4) to
                              the RC[11:0] bits of the SR register
                            ; If RE-RS==-2 during SETRC instruction
                            execution, the repeat loop is regarded
                              as two-instruction repeat.
RptDtct:  instr0            ; An instruction prior to the Repeat
                              start instruction is regarded as a
                              repeat detection instruction.
RptStart: instr1            ; [Repeat start instruction]
RptEnd:   instr2            ; [Repeat end instruction]
```

- Example 4: Repeat loop consisting of one instruction

```
            LDRS RptDtct +8; Sets (repeat detection instruction
                             address + 8) to the RS register
            LDRE RptDtct +4       ; Sets (repeat detection instruction
                                    address + 4) to the RE register
            SETRC #4        ; Sets the number of repetitions (4) to
                              the RC[11:0] bits of the SR register
                            ; If RE-RS==-4 during SETRC instruction
                            execution, the repeat loop is regarded
                              as one-instruction repeat.
  RptDtct:  instr0          ; An instruction prior to the Repeat
                              start instruction is regarded as a
                              repeat detection instruction.
  RptStart:
  RptEnd:   instr1          ; [Repeat start instruction]==[Repeat end
                              instruction]
```

In repeat loops consisting of three instructions, two instructions and one instruction, specific addresses are specified in the RS register. RE – RS is calculated during SETRC instruction execution, and the number of instructions included in the repeat loop is determined according to the result. A value of 0, –2,and –4 in the result correspond to three instructions, two instructions, and one instruction, respectively.

If repeat instruction execution is completed without branching and if RC[11:0] > 0, an instruction following the repeat detection instruction is regarded as a repeat start instruction and instruction execution is repeated for the number of times corresponding to the recognized number of instructions. If RC[11:0] ≥ 2 when the repeat end instruction is completed, the RC[11:0] is decremented by 1 and then control is passed to the address specified by the RS register. If RC[11:0] ==1 (or 0) when the repeat end instruction is completed, the RC[11:0] is cleared to 0 and then the control is passed to the next instruction following the repeat end instruction.

Note: If RE – RS is a positive value, the CPU regards the repeat loop as a four-instruction repeat loop. (In a repeat loop consisting of four or more instructions, RE – RS is always a positive value. For details, refer to example 1 above.) If RE – RS is positive, or a value other than 0, –2,and –4, correct operation cannot be guaranteed.

Table 3.4 shows the addresses to be specified in the repeat start register (RS) and repeat end register (RE).

Table 3.4 RS and RE Setting Rule

	Number of Instructions in Repeat Loop			
	1	**2**	**3**	**≥4**
RS	RptStart0 + 8	RptStart0 + 6	RptStart0 + 4	RptStart
RE	RptStart0 + 4	RptStart0 + 4	RptStart0 + 4	RptEnd3 + 4

Note: The terms used above in table 3.2, are defined as follows.
RptStart: Address of the repeat start instruction
RptStart0: Address of the instruction one instruction prior to the repeat start instruction
RptEnd3: Address of the instruction three instructions prior to the repeat end instruction

(2) Repeat Control Instructions and Repeat Control Macros

To describe a repeat loop, the RS and RE registers must be specified appropriately by the LDRS and LDRS instructions and then the number of repetitions must be specified by the SERTC instruction. An 8-bit immediate data or a general register can be used as an operand of the SETRC instruction. To specify the RC as a value greater than 256, use SETRC Rm type instructions.

Table 3.5 Repeat Control Instructions

Instruction	Operation	Number of Execution States
LDRS @(disp,PC)	Calculates (disp x 2 + PC) and stores the result to the RS register	1
LDRE @(disp,PC)	Calculates (disp x 2 + PC) and stores the result to the RE register	1
SETRC #imm	Sets 8-bit immediate data imm to the RC[11:0] bits of the SR register and sets the information related to the number of repetitions to the RF[1:0] bits of the SR. RC[11:0] can be specified as 0 to 255.	1
SETRC Rm	Sets the[11:0] bits of the Rm register to the RC[11:0] bits of the SR register and sets the information related to the number of repetitions to the RF[1:0] bits of the SR. RC[11:0] can be specified as 0 to 4095.	1

The RS and RE registers must be specified appropriately according to the rules shown in table 3.4. The SH assembler supports control macros (REPEAT) as shown in table 3.6 to solve problems.

Table 3.6 Repeat Control Macros

Instruction	Operation	Number of Execution States
REPEAT RptStart, RptEnd, #imm	Specifies RptStart as repeat start instruction, RptEnd as repeat end instruction, and 8-bit immediate data #imm as number of repetitions. This macro is extended to three instructions: LDRS, LDRE, and SETRC which are converted correctly.	3
REPEAT RptStart, RptEnd, Rm	Specifies RptStart as repeat start instruction, RptEnd as repeat end instruction, and the [11:0] bits of Rm as number of repetitions. This macro is extended to three instructions: LDRS, LDRE, and SETRC which are converted correctly.	3

Using the repeat macros shown in table 3.4, examples 1 to 4 shown above can be simplified to examples 5 to 8 as shown below.

- Example 5: Repeat loop consisting of 4 or more instructions (extended to the instruction stream shown in example 1, above)

```
          REPEAT RptStart, RptEnd, #4
          Instr0          ;
RptStart: instr1          ; [Repeat start instruction]
          ... ...         ;
          ... ...         ;
          instr(N-3)    ;
          instr(N-2)    ;
          instr(N-1)    ;
Rptend:   instrN          ; [Repeat end instruction]
```

- Example 6: Repeat loop consisting of three instructions (extended to the instruction stream shown in example 2, above)

```
          REPEAT RptStart, RptEnd, #4
          instr0         ;
RptStart: instr1         ; [Repeat start instruction]
          instr2         ;
RptEnd:   instr3         ; [Repeat end instruction]
```

- Example 7: Repeat loop consisting of two instructions (extended to the instruction stream shown in example 3, above)

```
          REPEAT RptStart, RptEnd, #4
          instr0         ;
RptStart: instr1         ; [Repeat start instruction]
RptEnd:   instr2         ; [Repeat end instruction]
```

- Example 8: Repeat loop consisting of one instruction instructions (extended to the instruction stream shown in example 4, above)

```
          REPEAT RptStart, RptEnd, #4
          instr0         ;
RptStart:
RptEnd:   instr1         ; [Repeat start instruction]==[Repeat end
                           instruction]
```

In the DSP mode, the system control instructions (LDC and STC) that handle the RS and RE registers are extended. The RC[11:0] bits and RF[1:0] bits of the SR can be controlled by the LDC and STC instructions for the SR register. These instructions should be used if an exception is enabled during repeat loop execution. The repeat loop can be resumed correctly by storing the RS and RE register values and RC[11:0] bits and RF[1:0] bits of the SR register before exception handling and by restoring the stored values after exception handling. However, note that there are some restrictions on exception acceptance during repeat loop execution. For details refer to Restrictions on Repeat Loop Control in section 3.3.1, DSP Repeat Control and section 7, Exception Handling.

Table 3.7 DSP Mode Extended System Control Instructions

Instruction	Operation	Number of Execution States
STC RS, Rn	RS→Rn	1
STC RE, Rn	RE→Rn	1
STC.L RS, @-Rn	Rn-4→Rn, RS→(Rn)	1
STC.L RE, @-Rn	Rn-4→Rn, RE→(Rn)	1
LDC.L @Rn+, RS	(Rn)→RS, Rn+4→Rn	4
LDC.L @Rn+, RE	(Rn)→RE, Rn+4→Rn	4
LDC Rn,RS	Rn →RS	4
LDC Rn, RE	Rn→RE	4

(3) Restrictions on Repeat Loop Control

(a) Repeat control instruction assignment

The SETRC instruction must be executed after executing the LDRS and LDRE instructions. In addition, note that at least one instruction is required between the SETRC instruction and a repeat start instruction.

(b) Illegal instruction one or more instructions following the repeat detection instruction

If one of the following instructions is executed between an instruction following a repeat detection instruction to a repeat end instruction, an illegal instruction exception occurs.

- Branch instructions
  BRA, BSR, BT, BF, BT/S, BF/S, BSRF, RTS, BRAF, RTE, JSR, JMP, TRAPA
- Repeat control instructions
  SETRC, LDRS, LDRE
- Load instructions for SR, RS, and RE registers
  LDC Rn,SR, LDC @Rn+,SR, LDC Rn,RE, LDC @Rn+,RE, LDC Rn,RS, LDC @Rn+,RS

Note: This restriction applies to all instructions for a repeat loop consisting of one to three instructions and to three instructions including a repeat end instruction.

(c) Instructions prohibited during repeat loop (In a repeat loop consisting of four or more instructions)

The following instructions must not be placed between the repeat start instruction and repeat detection instruction in a repeat loop consisting of four or more instructions. Otherwise, the correct operation cannot be guaranteed.

- Repeat control instructions
  SETRC, LDRS, LDRE
- Load instructions for SR, RS, and RE registers
  LDC Rn,SR, LDC @Rn+,SR, LDC Rn,RE, LDC @Rn+,RE, LDC Rn,RS, LDC @Rn+,RS

Note: Multiple repeat loops cannot be guaranteed. Describe the inner loop by repeat control instructions, and the external loop by other instructions such as DT or BF/S.

(d) Branching to an instruction following the repeat detection instruction and restriction on an exception acceptance

Execution of a repeat detection instruction must be completed without any branch so that the CPU can recognize the repeat loop. Therefore, when the execution branches to an instruction following the repeat detection instruction, the control will not be passed to a repeat start instruction after executing a repeat end instruction because the repeat loop is not recognized by the CPU. In this case, the RC[11:0] bits of the SR register will not be changed.

- If a conditional branch instruction is used in the repeat loop, an instruction before a repeat detection instruction must be specified as a branch destination.
- If a subroutine call is used in the repeat loop, a delayed slot instruction of the subroutine call instruction must be placed before a repeat detection instruction.

Here, a branch includes a return from an exception processing routine. If an exception whose return address is placed in an instruction following the repeat detection instruction occurs, the repeat control cannot be returned correctly. Accordingly, an exception acceptance is restricted from the repeat detection instruction to the repeat end instruction. Exceptions such as interrupts that can be retained by the CPU are retained. For exceptions that cannot be retained by the CPU, a transition to an exception occurs but a program cannot be returned to the previous execution state correctly. For details, refer to section 7, Exception Handling.

Notes: 1. If a TRAPA instruction is used as a repeat detection instruction, an instruction following the repeat detection instruction is regarded as a return address. In this case, a control cannot be returned to the repeat control correctly. In a TRAPA instruction, an address of an instruction following the repeat detection address is regarded as return address. Accordingly, to return to the repeat control correctly, place a return address prior to the repeat detection instruction.

2. If a SLEEP instruction is placed following a repeat detection instruction, a transition to the low-power consumption state or an exception acceptance such as interrupts can be performed correctly. In this case, however, the repeat control cannot be returned correctly. To return to the repeat control correctly, the SLEEP instruction must be placed prior to the repeat detection instruction.

(e) Branch from a repeat detection instruction

If a repeat detection instruction is a delayed slot instruction of a delayed branch instruction or a branch instruction, a repeat loop can be acknowledged when a branch does not occur in a branch instruction. If a branch occurs in a branch instruction, a repeat control is not performed and a branch destination instruction is executed.

(f) Program counter during repeat control

If RC[11:0] ≥ 2, the program counter (PC) value is not correct for instructions two instructions following a repeat detection instruction. In a repeat loop consisting of one to three instructions, the PC indicates the correct value (instruction address + 4) for an instruction (repeat start instruction) following a repeat detect ion instruction but the PC continues to indicate the same address (repeat start instruction address) from the subsequent instruction to a repeat end instruction. In a repeat loop consisting of four or more instructions, the PC indicates the correct value (instruction address + 4) for an instruction following a repeat detect ion instruction, but PC indicates the RS and (RS +2) for instructions two and three instructions following the repeat detection instruction. Here, RS indicates the value stored in the repeat start register (RS). The correct operation cannot be guaranteed for the incorrect PC values.

Accordingly, PC relative addressing instructions placed two or more instructions following the repeat detection instruction cannot be executed correctly and the correct results cannot be obtained.

- PC relative addressing instructions
  MOV.A @(disp, PC), Rn
  MOV.W @(disp, PC), Rn
  MOV.L @(disp, PC), Rn
  (Including the case when the MOV #imm,Rn is extended to MOV.W @(disp, PC), Rn or MOV.L @(disp, PC), Rn)

Table 3.8 PC Value during Repeat Control (When RC[11:0] ≥ 2)

	Number of Instructions in Repeat Loop			
	1	**2**	**3**	**≥4**
RptDtct	RptDtct + 4	RptDtct + 4	RptDtct + 4	RptDtct +4
RptDtct1	RptDtct1 + 4	RptDtct1+ 4	RptDtct1 + 4	RptDtct1 + 4
RptDtct2	—	RptDtct1+ 4	RptDtct1 + 4	RS
RptDtct3	—	—	RptDtct1 + 4	RS + 2

Note: In table 3.8, the following labels are used.
RptDtct: An address of the repeat detection instruction
RptDtct1: An address of the instruction one instruction following the repeat start instruction (In a repeat loop consisting of one to three instructions, RptStart is a repeat start instruction)
RptDtct2: An address of the instruction two instruction following the repeat start instruction
RptDtct3: An address of the instruction three instruction following the repeat start instruction

(g) Repeat counter and repeat control

The CPU always executes a program with comparing the repeat end register (RE) and the program counter (PC). If the PC matches the RE while the RC[11:0] bits of the SR register are other than 0, the repeat control function is initiated.

- If RC ≥ 2, a control is passed to a repeat start instruction after a repeat end instruction has been executed. The RC is decremented by 1 at the completion of the repeat end instruction.
  In this case, restrictions (1) to (6) are also applied.
- If RC == 1, the RC is decremented to 0 at the completion of the repeat end instruction and a control is passed to the subsequent instruction. In this case, restrictions (1) to (6) are also applied.
- If RC == 0, the repeat control function is not initiated even if a repeat detection instruction is executed. The repeat loop is executed once as normal instructions and a control is not be passed to a repeat start instruction even if a repeat end instruction is executed.

3.4 DSP Data Transfer Instructions

In DSP mode, data transfer instructions are added for the DSP unit registers. The newly added instructions are classified into the following three groups.

1. Double data transfer instructions

   The DSP unit is connected to the X memory and Y memory via the specific buses called X bus and Y bus. By using the data transfer instructions using the X and Y buses, two data items can be transferred between the DSP unit and X/Y memories simultaneously. These instructions are called double data transfer instructions. These double data transfer instructions can be described in combination with the DSP operation instructions to execute data transfer and data operation in parallel,
2. Single data transfer instructions

   The DSP unit is also connected to the L bus that is used by the CPU. The DSP registers other than the DSR can access any virtual addresses generated by the CPU. In this case, the single data transfer instructions are used. The single data transfer instructions cannot be used in combination with the DSP operation instructions and can access only one data item at a time.
3. System control instructions

   Some of the DSP unit registers are handled as the CPU system registers. To control these system registers, the system control registers are supported. The DSP registers are connected to the CPU general registers via the data transfer bus (C bus).

In any DSP data transfer instructions, an address to be accessed is generated and output by the CPU. For DSP data transfer instructions, some of the CPU general registers are used for address generation and specific addressing modes are used.

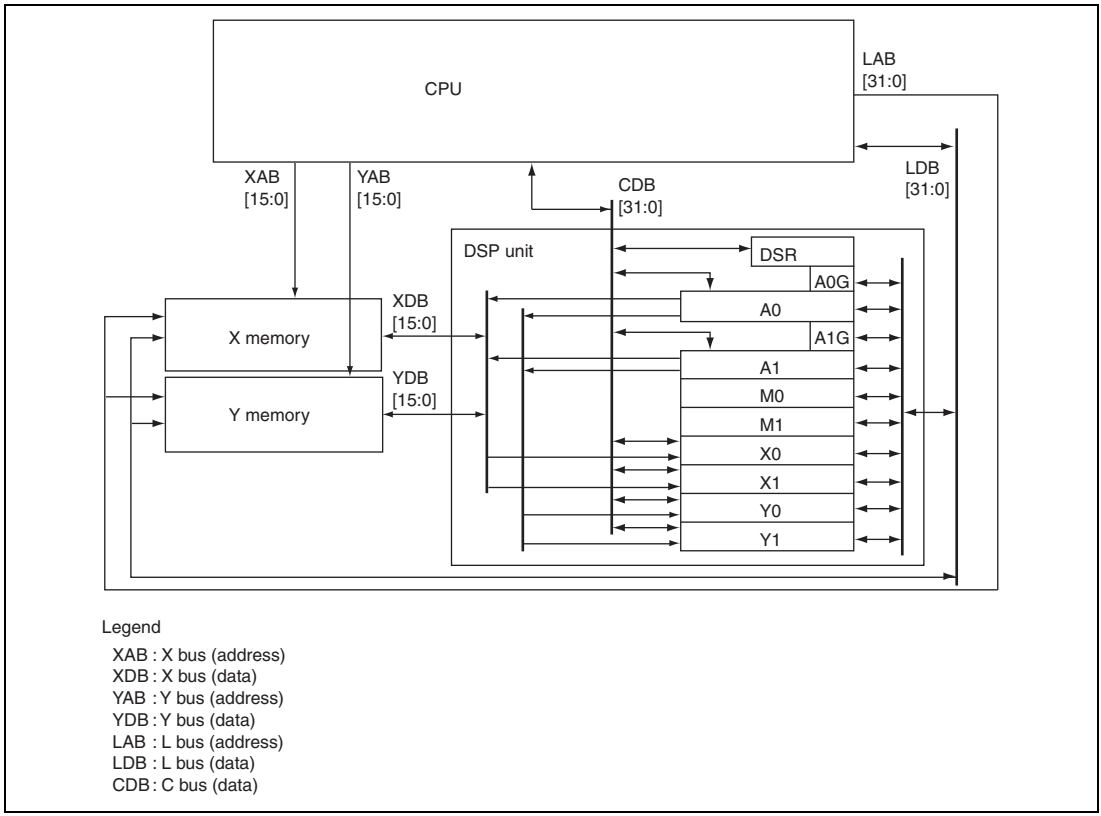


Figure 3.4 DSP Registers and Bus Connections

(1) Double data transfer instructions (MOVX.W, MOVY.W)

With double data transfer instructions, X memory and Y memory can be accessed in parallel.

In this case, the specific buses called X bus and Y bus are used to access X memory and Y memory, respectively. To fetch the CPU instructions, the L bus is used. Accordingly, no conflict occurs among X, Y, and L buses.

Load instructions for X memory specify the X0 or X1 register as the destination operand. Load instructions for Y memory specify the Y0 or Y1 register as the destination operand. Store registers for X or Y memory specify the A0 or A1 register as the source operand. These instructions use only word data (16 bits). When a word data transfer instruction is executed, the upper word of register operand is used. To load word data, data is loaded to the upper word of the destination register and the lower word of the destination register is automatically cleared to 0.

Double data transfer instructions can be described in parallel to the DSP operation instructions. Even if a conditional operation instruction is specified in parallel to a double data transfer instruction, the specified condition does not affect the data transfer operations. For details, refer to section 3.5, DSP Data Operation Instructions.

Double data transfer instructions can access only the X memory or Y memory and cannot access other memory space. The X bus and Y bus are 16 bits and support 64-byte address spaces corresponding to address areas H'A5000000 to H'A500FFFF and H'A5010000 to H'A501FFFF, respectively. Because these areas are included in the P2/Uxy area, they are not affected by the cache and address translation unit.

(2) Single data transfer instructions

The single data transfer instructions access any memory location. All DSP registers other than the DSR can be specified as source and destination operands.* Guard bit registers A0G and A1G can also be specified as two independent registers. Because these instructions use the L bus (LAB and LDB), these instructions can access any virtual space handled by the CPU. If these instructions access the cacheable area while the cache is enabled, the area accessed by these instructions are cached. The X memory and Y memory are mapped to the virtual address space and can also be accessed by the single data transfer instructions. In this case, bus conflict may occur between data transfer and instruction fetch because the CPU also uses the L bus for instruction fetches.

The single data transfer instructions can handle both word and longword data. In word data transfer, only the upper word of the operand register is valid. In word data load, word data is loaded into the upper word of the destination registers and the lower word of the destination is automatically cleared to 0. If the guard bits are supported, the sign bit is extended before storage. In longword data load, longword data is loaded into the upper and lower word of the destination register. If the guard bits are supported, the sign bit is extended before storage. When the guard register is stored, the sign bit is extended to the upper 24 bits of the LDB and are loaded onto the LDB bus.

Notes: * Since the DSR register is defined as the system register, it can be accessed by the LDS or STS instruction.

1. Any data transfer instruction is executed at the MA stage of the pipeline.
2. Any data transfer instruction does not modify the condition code bits of the DSR register.

(3) System control instructions

The DSR, A0, X0, X1, Y0, and Y1 registers in the DSP unit can also be used as the CPU system registers. Accordingly, data transfer operations between these DSP system registers and general registers or memory can be executed by the STS and LDS instructions. These DSP system registers can be treated as the CPU system register such as PR, MACH and MACL and can use the same addressing modes.

Table 3.9 Extended System Control Instructions in DSP Mode

Instruction		Operation	Execution States
STS	DSR,Rn	DSR → Rn	1
STS	A0,Rn	A0 → Rn	1
STS	X0,Rn	X0 → Rn	1
STS	X1,Rn	X1 → Rn	1
STS	Y0,Rn	Y0 → Rn	1
STS	Y1,Rn	Y1 → Rn	1
STS.L	DSR,@-Rn	Rn – 4 → Rn, DSR → (Rn)	1
STS.L	A0,@-Rn	Rn – 4 → Rn, A0 → (Rn)	1
STS.L	X0,@-Rn	Rn – 4 → Rn, X0 → (Rn)	1
STS.L	X1,@-Rn	Rn – 4 → Rn, X1 → (Rn)	1
STS.L	Y0,@-Rn	Rn – 4 → Rn, Y0 → (Rn)	1
STS.L	Y1,@-Rn	Rn – 4 → Rn, Y1 → (Rn)	1
LDS.L	@Rn+,DSR	(Rn) → DSR, Rn + 4 → Rn	1
LDS.L	@Rn+,A0	(Rn) → A0, Rn + 4 → Rn	1
LDS.L	@Rn+,X0	(Rn) → X0, Rn + 4 → Rn	1
LDS.L	@Rn+,X1	(Rn) → X1, Rn + 4 → Rn	1
LDS.L	@Rn+,Y0	(Rn) → Y0, Rn + 4 → Rn	1
LDS.L	@Rn+,Y1	(Rn) → Y1, Rn + 4 → Rn	1
LDS	Rn,DSR	Rn → DSR	1
LDS	Rn,A0	Rn → A0	1
LDS	Rn,X0	Rn → X0	1
LDS	Rn,X1	Rn → X1	1
LDS	Rn,Y0	Rn → Y0	1
LDS	Rn,Y1	Rn → Y1	1

3.4.1 General Registers

The DSP instructions 10 general registers in the 16 general registers are used as address pointers or index registers for double data transfers and single data transfers. In the following descriptions, another register function in the DSP instructions is also indicated within parentheses [ ].

- Double data transfer instructions (X memory and Y memory are accessed simultaneously)

  In double data transfers, X memory Y memory can be accessed simultaneously. To specify X and Y memory addresses, two address pointers are supported.

	Address Pointer	Index Register
X memory (MOVX.W)	R4,R5[Ax]	R8 [Ix]
Y memory (MOVY.W)	R6,R7[Ay]	R9 [Iy]

- Single data transfer instructions

  In single data transfer, any virtual address space can be accessed via the L bus. The following address pointers and index registers are used.

	Address Pointer	Index Register
Any virtual space (MOVS.W/L)	R4,R5, R2, R3[As]	R8 [Is]

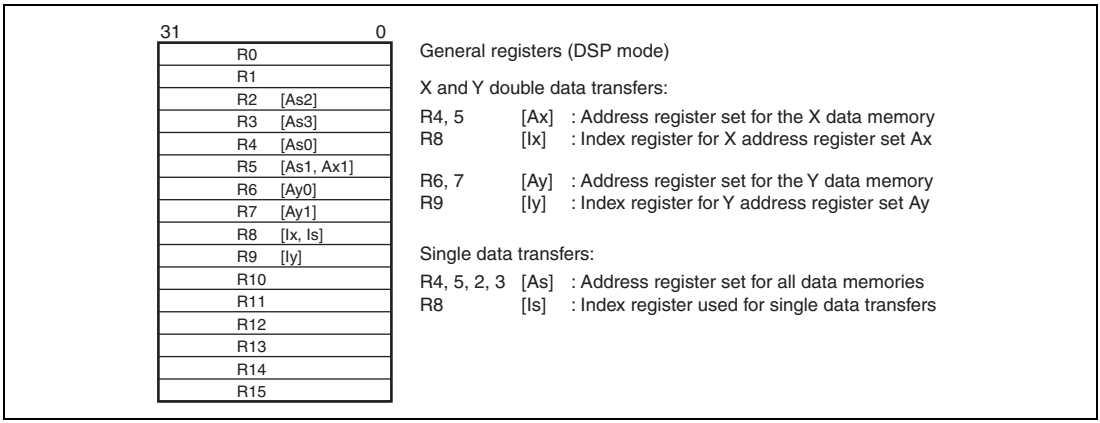


Figure 3.5 General Registers (DSP Mode)

In assembler, R0 to R9 are used as symbols. In the DSP data transfer instructions, the following register names (alias) can also be used. In assembler, described as shown below.

Ix: .REG (R8)

Ix indicates the alias of register 8. Other aliases are shown below.

Ax0: .REG (R4)

Ax1: .REG (R5)

Ix: .REG (R8)

Ay0: .REG (R6)

Ay1: .REG(R7)

Iy: .REG (R9)

As0: .REG (R4); This definition is used for if the alias is required in the single data transfer

As1: .REG (R5); This definition is used for if the alias is required in the single data transfer

As2: .REG (R2)

As3: .REG (R3)

Is: .REG (R8); This definition is used for if the alias is required in the single data transfer

3.4.2 DSP Data Addressing

Table 3.10 shows the relationship between the double data transfer instructions and single data transfer instructions.

Table 3.10 Overview of Data Transfer Instructions

	Double Data Transfer Instructions	Single Data Transfer Instructions
	MOVX.W MOVY.W	**MOVS.W, MOVS.L**
Address register	Ax: R4, R5 Ay: R6, R7	As: R2, R3, R4, R5
Index register	Ix: R8, Iy: R9	Is: R8
Addressing	Nop/Inc (+2)/index addition: post-increment	Nop/Inc (+2, +4)/index addition: post-increment
	—	Dec (–2, –4): pre-decrement
Modulo addressing	Possible	Not possible
Data bus	XDB, YDB	LDB
Data length	16 bits (word)	16/32 bits (word/longword)
Bus conflict	No	Yes
Memory	X/Y data memory	Entire memory space
Source register	Da: A0, A1	Ds: A0/A1, M0/M1, X0/X1, Y0/Y1, A0G, A1G
Destination register	Dx: X0/X1 Dy: Y0/Y1	Ds: A0/A1, M0/M1, X0/X1, Y0/Y1, A0G, A1G

(1) Addressing Mode for Double Data Transfer Instructions

The double data transfer instructions supports the following three addressing modes.

- Non-update address register addressing

  The Ax and Ay registers are address pointers. They are not updated.
- Increment address register addressing

  The Ax and Ay registers are address pointers. After a data transfer, they are each incremented by 2 (post-increment).
- Addition index register addressing

  The Ax and Ay registers are address pointers. After a data transfer, the value of the Ix or Iy register is added to each (post-increment). The double data transfer instructions do not supports decrement addressing mode. To perform decrement, –2 or –4 is set in the index register and addition index register addressing is specified.

When using X/Y data addressing, bit 0 of the address pointer is invalid; bits 0 and 1 of the address pointer are invalid in word access. Accordingly, bit 0 of the address pointer and index register must be cleared to 0 in X/Y data addressing.

When accessing X and Y memory using the X and Y buses, the upper word of Ax and Ay is ignored. The result of Ay+ or Ay+Iy is stored in the lower word of Ay, while the upper word retains its original value. The Ax and Ax +Ix operations are executed in longword (32 bits) and the upper word may be changed according to the result.

(2) Single Data Addressing

The following four kinds of addressing can be used with single data transfer instructions.

- Non-update address register addressing

  The As register is an address pointer. An access to @As is performed but As is not updated.
- Increment address register addressing:

  The As register is an address pointer. After an access to @As, the As register is incremented by 2 or 4 (post-increment).
- Addition index register addressing:

  The As register is an address pointer. After an access to @As, the value of the Is register is added to the As register (post-increment).
- Decrement address register addressing:

  The As register is an address pointer. Before a data transfer, –2 or –4 is added to the As register (i.e. 2 or 4 is subtracted) (pre-decrement).

In single data transfer instructions, all bits in 32-bit address are valid.

3.4.3 Modulo Addressing

In double data transfer instructions, a module addressing can be used. If the address pointer value reaches the preset modulo end address while a modulo addressing mode is specified,, the address pointer value becomes the modulo start address.

To control modulo addressing, the modulo register (MOD) extended in the DSP mode and the DMX and DMY bits of the SR register are used.

The MOD register is provided to set the start and end addresses of the modulo address area. The upper and lower words of the MOD register store modulo start address (MS) and modulo end address (ME), respectively. The LDC and STC instructions are extended for MOD register handling.

If the DMX bit in the SR register is set, the modulo addressing is specified for the X address register. If the DMY bit in the SR register is set, the modulo addressing is specified for the Y address register. Modulo addressing is valid for either the X or the Y address register, only; it cannot be set for both at the same time. Therefore, DMX and DMY cannot both be set simultaneously (if they are, the DMY setting will be valid). (In the future, this specification may be changed.) The MDX and MDY bits of the SR can be specified by the STC or LDC instruction for the SR register.

If an exception is accepted during modulo addressing, the MDX and MDY bits of the SR and MOD register must be saved. By restoring these register values, a control is returned to the modulo addressing after an exception handling.

Table 3.11 Modulo Addressing Control Instructions

Instruction	Operation	Execution States
`STC MOD,Rn`	MOD → Rn	1
`STC.L MOD,Rn`	Rn – 4 → Rn, MOD → (Rn)	1
`LDC.L @Rn+,MOD`	(Rn) → Rn, Rn + 4 → Rn	4
`LDC Rn,MOD`	Rn → MOD	4

An example of the use of modulo addressing is shown below.

```
MOV.L #H'70047000, R10
                ;Specify MS=H'7000 ME = H'7004
LDC R10,MOD     ;Specify ME:MS to MOD register
STC SR, R10     ;
MOV.L #H'FFFFF3FF, R11;
MOV.L #H'00000400, R12;
AND R11, R10    ;
OR R12, R10     ;
LDC R10, SR     ; Specify SR.MDX=1,
                  SR.MDY=0, and X modulo addressing mode
MOV.L #H'A5007000, R4
MOVX.W @R4+,X0  ; R4: H'A5007000→ H'A5007002
MOVX.W @R4+,X0  ; R4: H'A5007002→ H'A5007004
MOVX.W @R4+,X0  ; R4: H'A5007004→ H'A5007000
                (Matches to ME and MS is set)
MOVX.W @R4+,X0  ; R4: H'A5007000→ H'A5007002
```

The start and end addresses are specified in MS and ME, then the DMX or DMY bit is set to 1. When the X or Y data transfer instruction specified by the DMX or DMY is executed, the address register contents before updating are compared with ME*, and if they match, start address MS is stored in the address register as the value after updating.

When the addressing type of the X/Y data transfer instruction is no-update, the X/Y data transfer instruction is not returned to MS even if they match ME. When the addressing type of the X/Y data transfer instruction is addition index register addressing, the address pointed may not match the address pointer ME, and exceed it. In this case, the address pointer value does not become the modulo start address.

The maximum modulo size is 64 kbytes. This is sufficient to access the X and Y data memory.

Note: Not only with modulo addressing, but when X and Y data addressing is used, bit 0 is ignored. 0 must always be written to bit 0 of the address pointer, index register, MS, and ME.

3.4.4 Memory Data Formats

Memory data formats that can be used in the DSP instructions are classified into byte and longword. An address error will occur if word data starting from an address other than 2n or longword data starting from an address other than 4n is accessed by MOVS.L, LDS.L, or STS.L instruction. In such cases, the data accessed cannot be guaranteed

An address error will not occur if word data starting from an address other than 2n is accessed by the MOVX.W or MOVY.W instruction. When using the MOVX.W or MOVY.W instruction, an address must be specified on the boundary 2n. If an address is specified other than 2n, the data accessed cannot be guaranteed.

3.4.5 Instruction Formats of Double and Single Transfer Instructions

The format of double data transfer instructions is shown in tables 3.12 and that of single data transfer instructions in table 3.13.

Table 3.12 Double Data Transfer Instruction Formats

Type	Mnemonic	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X memory data transfer	NOPX	1	1	1	1	0	0	0		0		0		0	0		
	MOVX.W @Ax,Dx							Ax		Dx		0		0	1		
	MOVX.W @Ax+,Dx													1	0		
	MOVX.W @Ax+Ix,Dx													1	1		
	MOVX.W Da,@Ax									Da		1		0	1		
	MOVX.W Da,@Ax+													1	0		
	MOVX.W Da,@Ax+Ix													1	1		
Y memory data transfer	NOPY	1	1	1	1	0	0		0		0		0			0	0
	MOVY.W @Ay,Dy								Ay		Dy		0			0	1
	MOVY.W @Ay+,Dy															1	0
	MOVY.W @Ay+Iy,Dy															1	1
	MOVY.W Da,@Ay										Da		1			0	1
	MOVY.W Da,@Ay+															1	0
	MOVY.W Da,@Ay+Iy															1	1

Note: Ax: 0 = R4, 1 = R5
Ay: 0 = R6, 1 = R7
Dx: 0 = X0, 1 = X1
Dy: 0 = Y0, 1 = Y1
Da: 0 = A0, 1 = A1

Table 3.13 Single Data Transfer Instruction Formats

Type	Mnemonic	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Single data transfer	MOVS.W @-As,Ds	1	1	1	1	0	1	As		Ds		0:(*)		0	0	0	0
	MOVS.W @As,Ds							0:R4		1:(*)				0	1		
	MOVS.W @As+,Ds							1:R5		2:(*)				1	0		
	MOVS.W @As+Is,Ds							2:R2		3:(*)				1	1		
	MOVS.W Ds,@-As							3:R3		4:(*)				0	0	0	1
	MOVS.W Ds,@As									5:A1				0	1		
	MOVS.W Ds,@As+									6:(*)				1	0		
	MOVS.W Ds,@As+Is									7:A0				1	1		
	MOVS.L @-As,Ds									8:X0				0	0	1	0
	MOVS.L @As,Ds									9:X1				0	1		
	MOVS.L @As+,Ds									A:Y0				1	0		
	MOVS.L @As+Is,Ds									B:Y1				1	1		
	MOVS.L Ds,@-As									C:M0				0	0	1	1
	MOVS.L Ds,@As									D:A1G				0	1		
	MOVS.L Ds,@As+									E:M1				1	0		
	MOVS.L Ds,@As+Is									F:A0G				1	1		

Note: * Codes reserved for system use.

3.5 DSP Data Operation Instructions

3.5.1 DSP Registers

This LSI has eight data registers (A0, A1, X0, X1, Y0, Y1, M0 and M1) and one control register (DSR) as DSP registers (figure 3.3).

Four kinds of operation access the DSP data registers. The first is DSP data processing. When a DSP fixed-point data operation uses A0 or A1 as the source register, it uses the guard bits (bits 39 to 32). When it uses A0 or A1 as the destination register, guard bits 39 to 32 are valid. When a DSP fixed-point data operation uses a DSP register other than A0 or A1 as the source register, it sign-extends the source value to bits 39 to 32. When it uses one of these registers as the destination register, bits 39 to 32 of the result are discarded.

The second kind of operation is an X or Y data transfer operation, MOVX.W, MOVY.W. This operation accesses the X and Y memories through the 16-bit X and Y data buses (figure 3.4). The register to be loaded or stored by this operation always comprises the upper 16 bits (bits 31 to 16). X0 or X1 can be the destination of an X memory load and Y0 or Y1 can be the destination of a Y memory load, but no other register can be the destination register in this operation. When data is read into the upper 16 bits of a register (bits 31 to 16), the lower 16 bits of the register (bits 15 to 0) are automatically cleared. A0 and A1 can be stored in the X or Y memory by this operation, but no other registers can be stored.

The third kind of operation is a single-data transfer instruction, MOVS.W or MOVS.L. These instructions access any memory location through the LDB (figure 3.4). All DSP registers connect to the LDB and can be the source or destination register of the data transfer. These instructions have word and longword access modes. In word mode, registers to be loaded or stored by this instruction comprise the upper 16 bits (bits 31 to 16) for DSP registers except A0G and A1G. When data is loaded into a register other than A0G and A1G in word mode, the lower half of the register is cleared. When A0 or A1 is used, the data is sign-extended to bits 39 to 32 and the lower half is cleared. When A0G or A1G is the destination register in word mode, data is loaded into an 8-bit register, but A0 or A1 is not cleared. In longword mode, when the destination register is A0 or A1, it is sign-extended to bits 39 to 32.

The fourth kind of operation is system control instructions such as LDS, STS, LDS.L, or STS.L. The DSR, A0, X0, X1, Y0, and Y1 registers of the DSP register can be treated as system registers. For these registers, data transfer instructions between the CPU general registers and system registers or memory access instructions are supported.

Tables 3.14 and 3.15 show the data type of registers used in DSP instructions. Some instructions cannot use some registers shown in the tables because of instruction code limitations. For

example, PMULS can use A1 as the source register, but cannot use A0. These tables ignore details of register selectability.

Table 3.14 Destination Register in DSP Instructions

			Guard Bits	Register Bits		
Registers	**Instructions**		**39 32**	**31 16**	**15**	**0**
A0, A1	DSP operation	Fixed-point, PSHA, PMULS	Sign-extended	40-bit result		
		Integer, PDMSB	Sign-extended	24-bit result	Cleared	
		Logical, PSHL	Cleared	16-bit result	Cleared	
	Data transfer	MOVS.W	Sign-extended	16-bit data	Cleared	
		MOVS.L	Sign-extended	32-bit data		
A0G, A1G	Data transfer	MOVS.W	Data	No update		
		MOVS.L	Data	No update		
X0, X1 Y0, Y1 M0, M1	DSP operation	Fixed-point, PSHA, PMULS		32-bit result		
		Integer, logical, PDMSB, PSHL		16-bit result	Cleared	
	Data transfer	MOVX/Y.W, MOVS.W		16-bit result	Cleared	
		MOVS.L		32-bit data		

Table 3.15 Source Register in DSP Operations

			Guard Bits		Register Bits			
Registers	**Instructions**		**39**	**32**	**31**	**16**	**15**	**0**
A0, A1	DSP operation	Fixed-point, PDMSB, PSHA	40-bit data					
		Integer	24-bit data					
		Logical, PSHL, PMULS			16-bit data			
	Data transfer	MOVX/Y.W, MOVS.W			16-bit data			
		MOVS.L			32-bit data			
A0G, A1G	Data transfer	MOVS.W	Data					
		MOVS.L	Data					
X0, X1 Y0, Y1 M0, M1	DSP	Fixed-point, PDMSB, PSHA	Sign*		32-bit data			
		Integer	Sign*		16-bit data			
		Logical, PSHL, PMULS			16-bit data			
	Data transfer	MOVS.W			16-bit data			
		MOVS.L			32-bit data			

Note: * The data is sign-extended and input to the ALU.

The DSP unit incorporates one control register and DSP status register (DSR). The DSR register stores the DSP data operation result (zero, negative, others). The DSP register also has the DC bit whose function is similar to the T bit in the CPU register. The DC bit functions as status flag. Conditional DSP data operations are controlled based on the DC bit. These operation control affects only the DSP unit instructions. In other words, these operations control affects only the DSP registers and does not affect address register update and CPU instructions such as load and store instructions. A condition to be reflected on the DC bit should be specified to the DC status selection bits (CS[2:0]).

The unconditional DSP type data instructions other than PMULS, MOVX, MOVY, and MOVS change the condition flag and DC bit. However, the CPU instructions including the MAC instruction do not modify the DC bit. In addition, conditional DSP instructions do not modify the DSR.

Table 3.16 DSR Register Bits

Bits	Bit Name	Initial Value	R/W	Function
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	GT	0	R/W	Signed Greater Bit Indicates that the operation result is positive (except 0), or that operand 1 is greater than operand 2 1: Operation result is positive, or operand 1 is greater than operand 2
6	Z	0	R/W	Zero Bit Indicates that the operation result is zero (0), or that operand 1 is equal to operand 2 1: Operation result is zero (0), or operands are equal
5	N	0	R/W	Negative Bit Indicates that the operation result is negative, or that operand 1 is smaller than operand 2 1: Operation result is negative, or operand 1 is smaller than operand 2
4	V	0	R/W	Overflow Bit Indicates that the operation result has overflowed 1: Operation result has overflowed
3 to 1	CS	All 0	R/W	DC Bit Status Selection Designate the mode for selecting the operation result status to be set in the DC bit 000: Carry/borrow mode 001: Negative value mode 010: Zero mode 011: Overflow mode 100: Signed greater mode 101: Signed greater than or equal to mode 110: Reserved (setting prohibited) 111: Reserved (setting prohibited)

Bits	Bit Name	Initial Value	R/W	Function
0	DC	0	R/W	DSP Status Bit Sets the status of the operation result in the mode designated by the CS bits 0: Designated mode status has not occurred 1: Designated mode status has occurred Indicates the operation result by carry or borrow regardless of the CS bit status after the PADDC or PSUBC instruction has been executed.

The DSR is assigned to the system registers. For the DSR, the following load and store instructions are supported.

```
STS DSR,Rn;
STS.L DSR,@-Rn;
LDS Rn,DSR;
LDS.L @Rn+,DSR;
```

If the DSR is read by the STS instruction, upper bits (bits 31 to 16) are all 0.

3.5.2 DSP Operation Instruction Set

DSP operation instructions are instructions for digital signal processing performed by the DSP unit. These instructions have a 32-bit instruction code, and multiple instructions can be executed in parallel. The instruction code is divided into a field A and field B; a parallel data transfer instruction is specified in the field A, and a single or double data operation instruction in the field B. Instructions can be specified independently, and are also executed independently.

B-field data operation instructions are of three kinds: double data operation instructions, conditional single data operation instructions, and unconditional single data operation instructions. The formats of the DSP operation instructions are shown in table 3.17. The respective operands are selected independently from the DSP registers. The correspondence between DSP operation instruction operands and registers is shown in table 3.18.

Table 3.17 DSP Operation Instruction Formats

Type	Instruction Formats	
Double data operation instructions		ALUop. Sx, Sy, Du
		MLTop. Se, Df, Dg
Conditional single data operation instructions	DCT	ALUop. Sx, Sy, Dz
	DCF	ALUop. Sx, Sy, Dz
	DCT	ALUop. Sx, Dz
	DCF	ALUop. Sx, Dz
	DCT	ALUop. Sy, Dz
	DCF	ALUop. Sy, Dz
Unconditional single data operation instructions		ALUop. Sx, Sy, Dz
		ALUop. Sx, Dz
		ALUop. Sy, Dz
		MLTop. Se, Sf, Dg

Table 3.18 Correspondence between DSP Instruction Operands and Registers

	ALU Operations				Multiply Operations		
Register	**Sx**	**Sy**	**Dz**	**Du**	**Se**	**Sf**	**Dg**
A0	Yes		Yes	Yes			Yes
A1	Yes		Yes	Yes	Yes	Yes	Yes
M0		Yes	Yes				Yes
M1		Yes	Yes				Yes
X0	Yes		Yes	Yes	Yes	Yes	
X1	Yes		Yes		Yes		
Y0		Yes	Yes	Yes	Yes	Yes	
Y1		Yes	Yes			Yes	

When writing parallel instructions, the field-B instruction is written first, followed by the field-A instruction. A sample parallel processing program is shown in figure 3.6.

```
        PADD  A0, M0, A0    PMULS  X0, Y0, M0    MOVX.W @R4+,    X0    MOVY.W @R6+, Y0
DCF     PINC  M1, A1                             MOVX.W @R5+R8,  X0    MOVY.W @R7+, Y1
        PCMP  M1, M0                             MOVX.W @R4,     X1    [NOPY]
```

Figure 3.6 Sample Parallel Instruction Program

Square brackets mean that the contents can be omitted.

The no operation instructions NOPX and NOPY can be omitted. For details on the field B in DSP data operation instructions, refer to section 3.6.4, DSP Operation Instructions.

The DSR register condition code bit (DC) is always updated on the basis of the result of an unconditional ALU or shift operation instruction. Conditional instructions do not update the DC bit. Multiply instructions, also, do not update the DC bit. DC bit updating is performed by means of the CS[2:0] bits in the DSR register. The DC bit update rules are shown in table 3.19.

Table 3.19 DC Bit Update Definitions

CS [2:0]			Condition Mode	Description
0	0	0	Carry or borrow mode	The DC bit is set if an ALU arithmetic operation generates a carry or borrow, and is cleared otherwise. When a PSHA or PSHL shift instruction is executed, the last bit data shifted out is copied into the DC bit. When an ALU logical operation is executed, the DC bit is always cleared.
0	0	1	Negative value mode	When an ALU or shift (PSHA) arithmetic operation is executed, the MSB of the result, including the guard bits, is copied into the DC bit. When an ALU or shift (PSHL) logical operation is executed, the MSB of the result, excluding the guard bits, is copied into the DC bit.
0	1	0	Zero value mode	The DC bit is set if the result of an ALU or shift operation is all-zeros, and is cleared otherwise.
0	1	1	Overflow mode	The DC bit is set if the result of an ALU or shift (PSHA) arithmetic operation exceeds the destination register range, excluding the guard bits, and is cleared otherwise. When an ALU or shift (PSHL) logical operation is executed, the DC bit is always cleared.
1	0	0	Signed greater-than mode	This mode is similar to signed greater-or-equal mode, but DC is cleared if the result is all-zeros. DC = ~{(negative value ^ over-range) | zero value}; In case of arithmetic operation DC = 0; In case of logical operation
1	0	1	Signed greater-or-equal mode	If the result of an ALU or shift (PSHA) arithmetic operation exceeds the destination register range, including the guard bits (over-range), the definition is the same as in negative value mode. If the result is not over-range, the definition is the opposite of that in negative value mode. When an ALU or shift (PSHL) logical operation is executed, the DC bit is always cleared. DC = ~(negative value ^ over-range); In case of arithmetic operation DC = 0 ; In case of logical operation
1	1	0	Reserved (setting prohibited)	
1	1	1	Reserved (setting prohibited)	

- Conditional Operations and Data Transfer

  Some instructions belonging to this class can be executed conditionally, as described earlier. The specified condition is valid only for the B field of the instruction, and is not valid for data transfer instructions for which a parallel specification is made. Examples are shown in figure 3.7.

```
DCT PADD X0,Y0,A0  MOVX.W @R4+,X0  MOVY.W A0,@R6+R9

When condition is True

Before execution: X0=H'33333333, Y0=H'55555555, A0=H'123456789A,
                  R4=H'00008000, R6=H'00005000, R9=H'00000004
                  (R4)=H'1111, (R6)=H'2222
After execution:  X0=H'11110000, Y0=H'55555555, A0=H'0088888888,
                  R4=H'00008002, R6=H'00005004, R9=H'00000004
                  (R4)=H'1111, (R6)=H'3456

When condition is False

Before execution: X0=H'33333333, Y0=H'55555555, A0=H'123456789A,
                  R4=H'00008000, R6=H'00005000, R9=H'00000004
                  (R4)=H'1111, (R6)=H'2222
After execution:  X0=H'11110000, Y0=H'55555555, A0=H'123456789A,
                  R4=H'00008002, R6=H'00005004, R9=H'00000004
                  (R4)=H'1111, (R6)=H'3456
```

Figure 3.7 Examples of Conditional Operations and Data Transfer Instructions

- Assignment of NOPX and NOPY Instruction Codes

  When there is no data transfer instruction to be parallel-processed simultaneously with a DSP operation instruction, an NOPX or NOPY instruction can be written as the data transfer instruction, or the instruction can be omitted. The instruction code is the same whether an NOPX or NOPY instruction is written or the instruction is omitted. Examples of NOPX and NOPY instruction codes are shown in table 3.20.

Table 3.20 Examples of NOPX and NOPY Instruction Codes

Instruction			Code
PADD X0,Y0,A0	MOVX.W @R4+,X0	MOVY.W @R6+R9,Y0	1111100000001011
			1011000100000111
PADD X0,Y0,A0	NOPX	MOVY.W @R6+R9,Y0	1111100000000011
			1011000100000111
PADD X0,Y0,A0	NOPX	NOPY	1111100000000000
			1011000100000111
PADD X0,Y0,A0	NOPX		1111100000000000
			1011000100000111
PADD X0,Y0,A0			1111100000000000
			1011000100000111
	MOVX.W @R4+,X0	MOVY.W @R6+R9,Y0	1111000000001011
	MOVX.W @R4+,X0	NOPY	1111000000001000
	MOVS.W @R4+,X0		1111010010001000
	NOPX	MOVY.W @R6+R9,Y0	1111000000000011
		MOVY.W @R6+R9,Y0	1111000000000011
	NOPX	NOPY	1111000000000000
NOP			0000000000001001

3.5.3 DSP-Type Data Formats

This LSI has several different data formats that depend on the instruction. This section explains the data formats for DSP type instructions.

Figure 3.8 shows three DSP-type data formats with different binary point positions. A CPU-type data format with the binary point to the right of bit 0 is also shown for reference.

The DSP-type fixed point data format has the binary point between bit 31 and bit 30. The DSP-type integer format has the binary point between bit 16 and bit 15. The DSP-type logical format does not have a binary point. The valid data lengths of the data formats depend on the instruction and the DSP register.

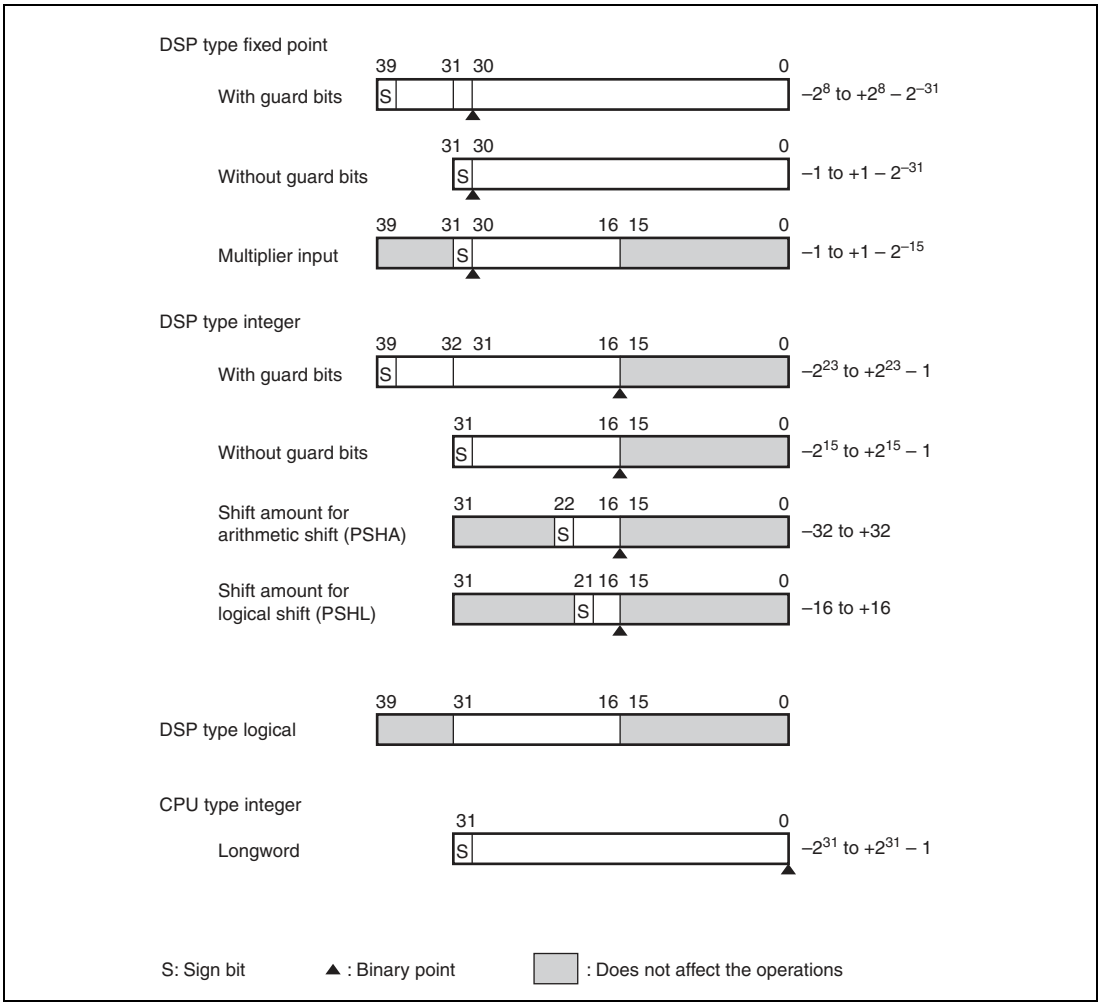


Figure 3.8 Data Formats

The shift amount for the arithmetic shift (PSHA) instruction has a 7-bit field that can represent values from –64 to +63, but –32 to +32 are valid numbers for the instruction. Also the shift amount for a logical shift operation has a 6-bit field, but –16 to +16 are valid numbers for the instruction.

3.5.4 ALU Fixed-Point Arithmetic Operations

Figure 3.9 shows the ALU arithmetic operation flow. Table 3.21 shows the variation of this type of operation and table 3.22 shows the correspondence between each operand and registers.

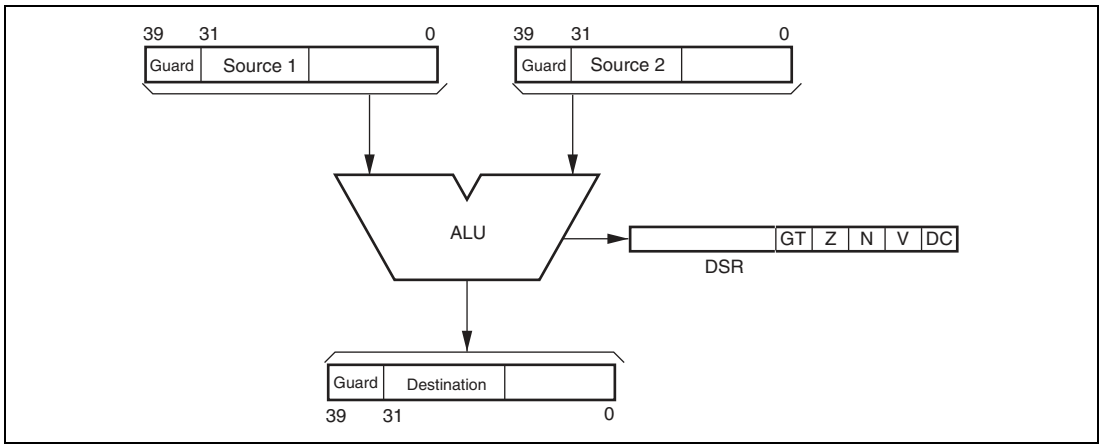


Figure 3.9 ALU Fixed-Point Arithmetic Operation Flow

Note: The ALU fixed-point arithmetic operations are basically 40-bit operation; 32 bits of the base precision and 8 bits of the guard-bit parts. So the signed bit is copied to the guard-bit parts when a register not providing the guard-bit parts is specified as the source operand. When a register not providing the guard-bit parts is specified as a destination operand, the lower 32 bits of the operation result are input into the destination register.

ALU fixed-point operations are executed between registers. Each source and destination operand are selected independently from one of the DSP registers. When a register providing guard bits is specified as an operand, the guard bits are activated for this type of operation. These operations are executed in the DSP stage, as shown in figure 3.10. The DSP stage is the same stage as the MA stage in which memory access is performed.

Table 3.21 Variation of ALU Fixed-Point Operations

Mnemonic	Function	Source 1	Source 2	Destination
PADD	Addition	Sx	Sy	Dz (Du)
PSUB	Subtraction	Sx	Sy	Dz (Du)
PADDC	Addition with carry	Sx	Sy	Dz
PSUBC	Subtraction with borrow	Sx	Sy	Dz
PCMP	Comparison	Sx	Sy	—
PCOPY	Data copy	Sx	All 0	Dz
		All 0	Sy	Dz
PABS	Absolute	Sx	All 0	Dz
		All 0	Sy	Dz
PNEG	Negation	Sx	All 0	Dz
		All 0	Sy	Dz
PCLR	Clear	All 0	All 0	Dz

Table 3.22 Correspondence between Operands and Registers

Register	Sx	Sy	Dz	Du
A0	Yes		Yes	Yes
A1	Yes		Yes	Yes
M0		Yes	Yes	
M1		Yes	Yes	
X0	Yes		Yes	Yes
X1	Yes		Yes	
Y0		Yes	Yes	Yes
Y1		Yes	Yes	

As shown in figure 3.10, data loaded from the memory at the MA stage, which is programmed at the same line as the ALU operation, is not used as a source operand for this operation, even though the destination operand of the data load operation is identical to the source operand of the ALU operation. In this case, previous operation results are used as the source operands for the ALU operation, and then updated as the destination operand of the data load operation.

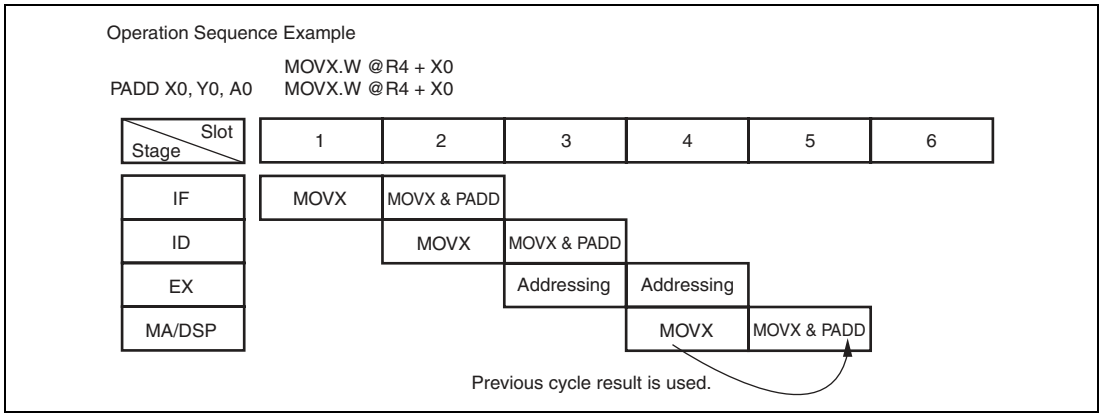


Figure 3.10 Operation Sequence Example

Every time an ALU arithmetic operation is executed, the DC, N, Z, V, and GT bits in DSR are basically updated in accordance with the operation result. However, in case of a conditional operation, they are not updated even though the specified condition is true and the operation is executed. In case of an unconditional operation, they are always updated in accordance with the operation result. The definition of a DC bit is selected by CS[2:0] (condition selection) bits in DSR. The DC bit result is as follows:

(1) Carry or Borrow Mode: CS[2:0] = B'000

The DC bit indicates that carry or borrow is generated from the most significant bit of the operation result, except the guard-bit parts. Some examples are shown in figure 3.11. This mode is the default condition. When the input data is negative in a PABS or PNEG instruction, carry is generated.

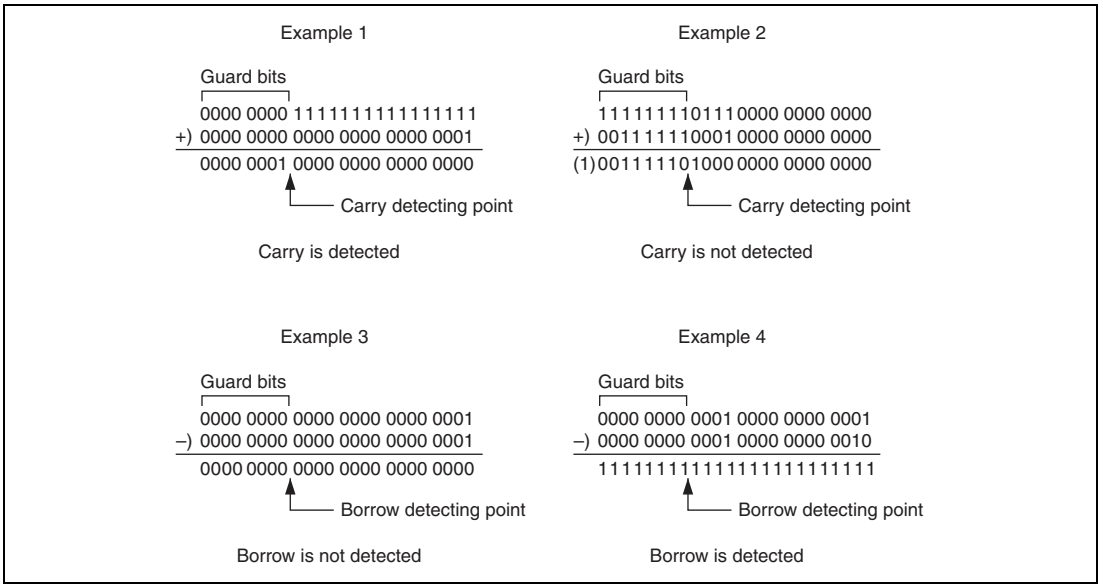


Figure 3.11 DC Bit Generation Examples in Carry or Borrow Mode

(2) Negative Value Mode: CS[2:0] = B'001

The DC flag indicates the same value as the MSB of the operation result. When the result is a negative number, the DC bit shows 1. When it is 0 or a positive number, the DC bit shows 0. The ALU always executes 40-bit arithmetic operation, so the sign bit to detect whether positive or negative is always got from the MSB of the operation result regardless of the destination operand. Some examples are shown in figure 3.12.

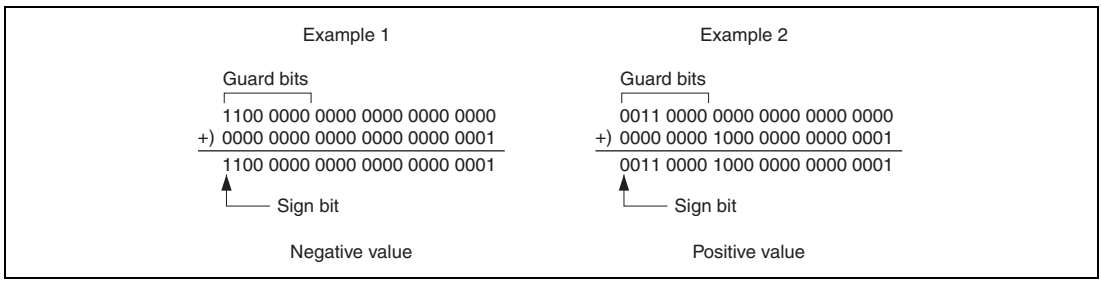


Figure 3.12 DC Bit Generation Examples in Negative Value Mode

(3) Zero Value Mode: CS[2:0] = B'010

The DC flag indicates whether the operation result is 0 or not. When the result is 0, the DC bit shows 1. When it is not 0, the DC bit shows 0.

(4) Overflow Mode: CS[2:0] = B'011

The DC bit indicates whether or not overflow occurs in the result. When an operation yields a result beyond the range of the destination register, except the guard-bit parts, the DC bit is set. Even though guard bits are provided in the destination register, the DC bit always indicates the result of when no guard bits are provided. So, the DC bit is always set to 1 if the guard-bit parts are used for large number representation. Some DC bit generation examples in overflow mode are shown in figure 3.13.

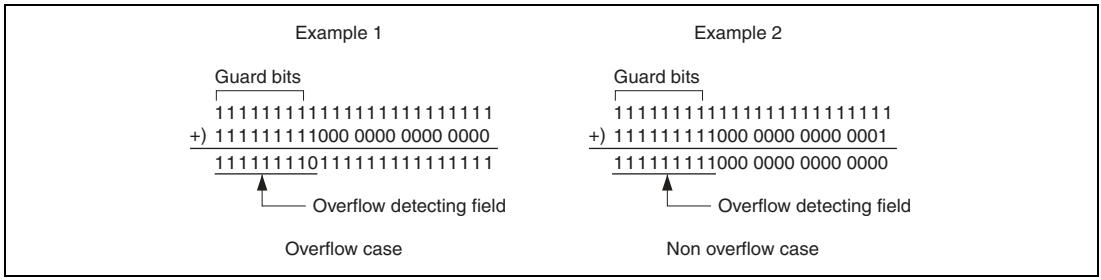


Figure 3.13 DC Bit Generation Examples in Overflow Mode

(5) Signed Greater Than Mode: CS[2:0] = B'100

The DC bit indicates whether or not the source 1 data (signed) is greater than the source 2 data (signed) as the result of compare operation PCMP. The PCMP operation should be executed before executing the conditional operation under this condition mode. This mode is similar to the Negative Value Mode described before, because the result of a compare operation is a positive value if the source 1 data is greater than the source 2 data. However, the signed bit of the result shows a negative value if the compare operation yields a result beyond the range of the destination operand, including the guard-bit parts (called “Over-range”), even though the source 1 data is greater than the source 2 data. The DC bit is updated concerning this type of special case in this condition mode. The equation below shows the definition of getting this condition:

DC = ~ {(Negative ^ Over-range) | Zero}

When the PCMP operation is executed under this condition mode, the result of the DC bit is the same as the T bit’s result of the CMP/GT operation of the CPU instruction.

(6) Signed Greater Than or Equal Mode: CS[2:0] = B'101

The DC bit indicates whether the source 1 data (signed) is greater than or equal to the source 2 data (signed) as the result of compare operation PCMP. This mode is similar to the Signed Greater Than Mode described before but the equal case is also included in this mode. The equation below shows the definition of getting this condition:

DC = ~ (Negative ^ Over-range)

When the PCMP operation is executed under this condition mode, the result of the DC bit is the same as the T bit’s result of a CMP/GE operation of the CPU instruction.

The N bit always indicates the same state as the DC bit set in negative value mode by the CS[2:0] bits. See the negative value mode part above. The Z bit always indicates the same state as the DC bit set in zero value mode by the CS[2:0] bits. See the zero value mode part above. The V bit always indicates the same state as the DC bit set in overflow mode by the CS[2:0] bits. See the overflow mode part above. The GT bit always indicates the same state as the DC bit set in signed greater than mode by the CS[2:0] bits. See the signed greater than mode part above.

Note: The DC bit is always updated as the carry flag for ‘PADDC’ and is always updated as the carry/borrow flag for ‘PSUBC’ regardless of the CS[2:0] state.

- Overflow Protection

  The S bit in SR is effective for any ALU fixed-point arithmetic operations in the DSP unit. See section 3.5.11, Overflow Protection, for details.

3.5.5 ALU Integer Operations

Figure 3.14 shows the ALU integer arithmetic operation flow. Table 3.23 shows the variation of this type of operation. The correspondence between each operand and registers is the same as ALU fixed-point operations as shown in table 3.22.

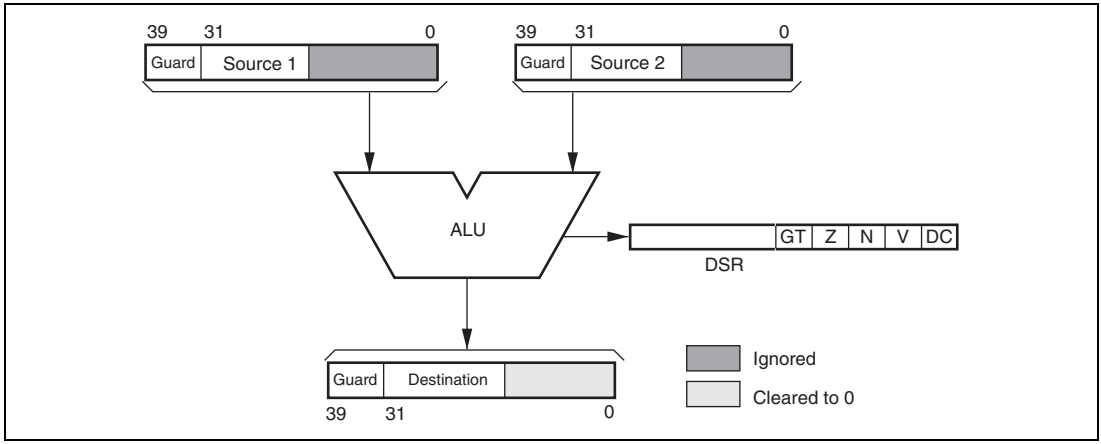


Figure 3.14 ALU Integer Arithmetic Operation Flow

Table 3.23 Variation of ALU Integer Operations

Mnemonic	Function	Source 1	Source 2	Destination
PINC	Increment by 1	Sx	+1	Dz
		+1	Sy	Dz
PDEC	Decrement by 1	Sx	–1	Dz
		–1	Sy	Dz

Note: The ALU integer operations are basically 24-bit operation, the upper 16 bits of the base precision and 8 bits of the guard-bits parts. So the signed bit is copied to the guard-bit parts when a register not providing the guard-bit parts is specified as the source operand. When a register not providing the guard-bit parts is specified as a destination operand, the upper word excluding the guard bits of the operation result are input into the destination register.

In ALU integer arithmetic operations, the lower word of the source operand is ignored and the lower word of the destination operand is automatically cleared. The guard-bit parts are effective in ALU integer arithmetic operations if they are supported. Others are basically the same operation as ALU fixed-point arithmetic operations. As shown in table 3.23, however, this type of operation provides two kinds of instructions only, so that the second operand is actually either +1 or –1. When a word data is loaded into one of the DSP unit's registers, it is input as an upper word data. When a register providing guard bits is specified as an operand, the guard bits are also activated. These operations, as well as fixed-point operations, are executed in the DSP stage, as shown in figure 3.10. The DSP stage is the same stage as the MA stage in which memory access is performed.

Every time an ALU arithmetic operation is executed, the DC, N, Z, V, and GT bits in DSR are basically updated in accordance with the operation result. This is the same as fixed-point operations but the lower word of each source and destination operand is not used in order to generate them. See section 3.5.4, ALU Fixed-Point Arithmetic Operations, for details.

In case of a conditional operation, they are not updated even though the specified condition is true and the operation is executed. In case of an unconditional operation, they are always updated in accordance with the operation result. See section 3.5.4, ALU Fixed-Point Arithmetic Operations, for details.

- Overflow Protection

  The S bit in SR is effective for any ALU integer arithmetic operations in DSP unit. See section 3.5.11, Overflow Protection, for details.

3.5.6 ALU Logical Operations

Figure 3.15 shows the ALU logical operation flow. Table 3.24 shows the variation of this type of operation. The correspondence between each operand and registers is the same as the ALU fixed-point operations as shown in table 3.21.

As shown in figure 3.15, this type of operation uses only the upper word of each operand. The lower word and guard-bit parts are ignored for the source operand and those of the destination operand are automatically cleared. These operations are also executed in the DSP stage, as shown in figure 3.10. The DSP stage is the same stage as the MA stage in which memory access is performed.

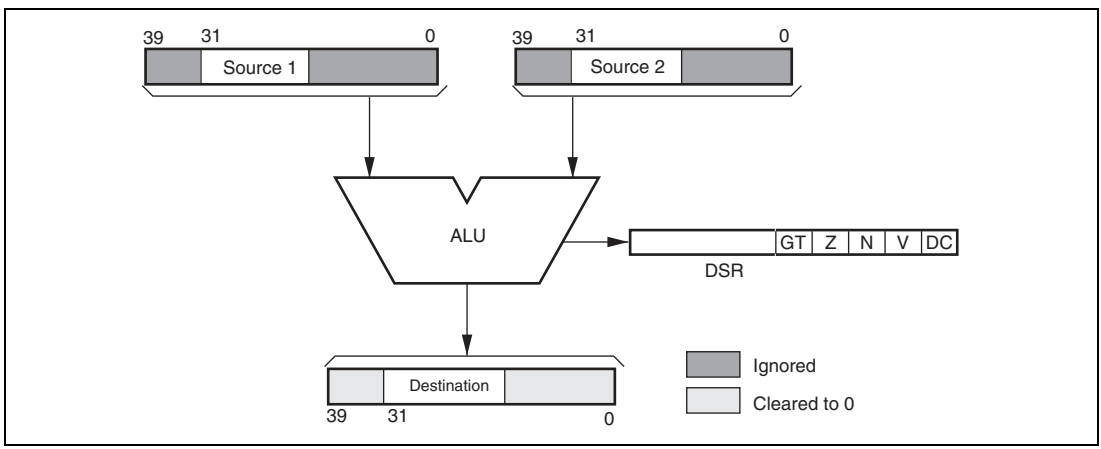


Figure 3.15 ALU Logical Operation Flow

Table 3.24 Variation of ALU Logical Operations

Mnemonic	Function	Source 1	Source 2	Destination
PAND	Logical AND	Sx	Sy	Dz
POR	Logical OR	Sx	Sy	Dz
PXOR	Logical exclusive OR	Sx	Sy	Dz

Every time an ALU logical operation is executed, the DC, N, Z, V, and GT bits in the DSR register are basically updated in accordance with the operation result. In case of a conditional operation, they are not updated even though the specified condition is true and the operation is executed. In case of an unconditional operation, they are always updated in accordance with the operation result. The definition of the DC bit is selected by the CS[2:0] (condition selection) bits in DSR. The DC bit result is:

(1) Carry or Borrow Mode: CS[2:0] = 000

The DC bit is always cleared.

(2) Negative Value Mode: CS[2:0] = 001

Bit 31 of the operation result is loaded into the DC bit.

(3) Zero Value Mode: CS[2:0] = 010

The DC bit is set when the operation result is zero; otherwise it is cleared.

(4) Overflow Mode: CS[2:0] = 011

The DC bit is always cleared.

(5) Signed Greater Than Mode: CS[2:0] = 100

The DC bit is always cleared.

(6) Signed Greater Than or Equal Mode: CS[2:0] = 101

The DC bit is always cleared.

The N bit always indicates the same state as the DC bit set in negative value mode by the CS[2:0] bits. See the negative value mode part above. The Z bit always indicates the same state as the DC bit set in zero value mode by the CS[2:0] bits. See the zero value mode part above. The V bit always indicates the same state as the DC bit set in overflow mode by the CS[2:0] bits. See the overflow mode part above. The GT bit always indicates the same state as the DC bit set in signed greater than mode by the CS[2:0] bits. See the signed greater than mode part above.

3.5.7 Fixed-Point Multiply Operation

Figure 3.16 shows the multiply operation flow. Table 3.25 shows the variation of this type of operation and table 3.26 shows the correspondence between each operand and registers. The multiply operation of the DSP unit is single-word signed single-precision multiplication. These operations are executed in the DSP stage, as shown in figure 3.10. The DSP stage is the same stage as the MA stage in which memory access is performed.

If a double-precision multiply operation is needed, the CPU standard double-word multiply instructions can be made of use.

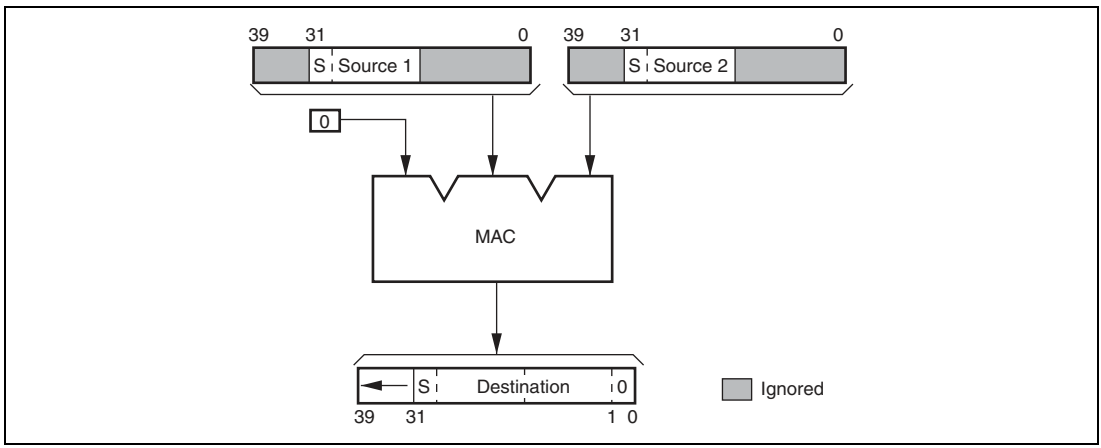


Figure 3.16 Fixed-Point Multiply Operation Flow

Table 3.25 Variation of Fixed-Point Multiply Operation

Mnemonic	Function	Source 1	Source 2	Destination
PMULS	Signed multiplication	Se	Sf	Dg

Table 3.26 Correspondence between Operands and Registers

Register	Se	Sf	Dg
A0	—	—	Yes
A1	Yes	Yes	Yes
M0	—	—	Yes
M1	—	—	Yes
X0	Yes	Yes	—
X1	Yes	—	—
Y0	Yes	Yes	—
Y1	—	Yes	—

Note: The multiply operations basically generate 32-bit operation results. So when a register providing the guard-bit parts are specified as a destination operand, the guard-bit parts will copy bit 31 of the operation result.

The multiply operation of the DSP unit side is not integer but fixed-point arithmetic operation. So, the upper words of each multiplier and multiplicand are input into a MAC unit as shown in figure 3.16. In the SH's standard multiply operations, the lower words of both source operands are input into a MAC unit. The operation result is also different from the SH's case. The SH's multiply operation result is aligned to the LSB of the destination, but the fixed-point multiply operation result is aligned to the MSB, so that the LSB of the fixed-point multiply operation result is always 0.

The fixed-point multiply operation is executed in one cycle. Multiply is always unconditional, but does not affect any condition code bits, DC, N, Z, V, and GT , in DSR.

- Overflow Protection

  The S bit in SR is effective for this multiply operation in the DSP unit. See section 3.5.11, Overflow Protection, for details.

  If the S bit is 0, overflow occurs only when H'8000*H'8000 ((-1.0)*(-1.0)) operation is executed as signed fixed-point multiply. The result is H'00 8000 0000 but it does not mean (+1.0). If the S bit is 1, overflow is prevented and the result is H'00 7FFF FFFF.

3.5.8 Shift Operations

Shift operations can use either register or immediate value as the shift amount operand. Other source and destination operands are specified by the register. There are two kinds of shift operations of arithmetic and logical shifts. Table 3.27 shows the variation of this type of operation. The correspondence between each operand and registers, except for immediate operands, is the same as the ALU fixed-point operations as shown in table 3.21.

Table 3.27 Variation of Shift Operations

Mnemonic	Function	Source 1	Source 2	Destination
PSHA Sx, Sy, Dz	Arithmetic shift	Sx	Sy	Dz
PSHL Sx, Sy, Dz	Logical shift	Sx	Sy	Dz
PSHA #Imm1, Dz	Arithmetic shift with immediate.	Dz	Imm1	Dz
PSHL #Imm2, Dz	Logical shift with immediate.	Dz	Imm2	Dz

–32 <= Imm1 <= +32, –16 <= Imm2 <= +16

(1) Arithmetic Shift

Figure 3.17 shows the arithmetic shift operation flow.

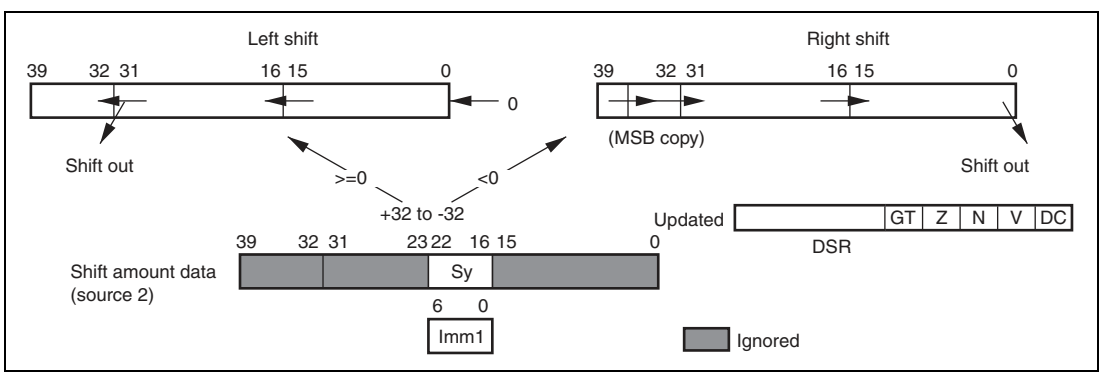


Figure 3.17 Arithmetic Shift Operation Flow

Note: The arithmetic shift operations are basically 40-bit operation, that is, the 32 bits of the base precision and eight bits of the guard-bit parts. So the signed bit is copied to the guard-bit parts when a register not providing the guard-bit parts is specified as the source operand. When a register not providing the guard-bit parts is specified as a destination operand, the lower 32 bits of the operation result are input into the destination register.

In this arithmetic shift operation, all bits of the source 1 and destination operands are activated. The shift amount is specified by the source 2 operand as an integer data. The source 2 operand can be specified by either a register or immediate operand. The available shift range is from –32 to +32. Here, a negative value means the right shift, and a positive value means the left shift. It is possible for any source 2 operand to specify from –64 to +63 but the result is unknown if an invalid shift value is specified. In case of a shift with an immediate operand instruction, the source 1 operand must be the same register as the destination's. This operation is executed in the DSP stage, as shown in figure 3.10 as well as in fixed-point operations. The DSP stage is the same stage as the MA stage in which memory access is performed.

Every time an arithmetic shift operation is executed, the DC, N, Z, V, and GT bits in DSR are basically updated in accordance with the operation result. In case of a conditional operation, they are not updated even though the specified condition is true and the operation is executed. In case of an unconditional operation, they are always updated in accordance with the operation result. The definition of the DC bit is selected by the CS[2:0] (condition selection) bits in DSR. The DC bit result is:

1. Carry or Borrow Mode: CS[2:0] = B'000
   The DC bit indicates the last shifted out data as the operation result.
2. Negative Value Mode: CS[2:0] = B'001
   The DC bit is set to 1 when the operation result is a negative value, and cleared to 0 when the operation result is zero or a positive value.
3. Zero Value Mode: CS[2:0] = B'010
   The DC bit is set when the operation result is zero; otherwise it is cleared.
4. Overflow Mode: CS[2:0] = B'011
   The DC bit is set to 1 when an overflow occurs.
5. Signed Greater Than Mode: CS[2:0] = B'100
   The DC bit is always cleared to 0.
6. Signed Greater Than or Equal Mode: CS[2:0] = B'101
   The DC bit is always cleared to 0.

The N bit always indicates the same state as the DC bit set in negative value mode by the CS[2:0] bits. See the negative value mode part above. The Z bit always indicates the same state as the DC bit set in zero value mode by the CS[2:0] bits. See the zero value mode part above. The V bit always indicates the same state as the DC bit set in overflow mode by the CS[2:0] bits. See the overflow mode part above. The GT bit always indicates the same state as the DC bit set in signed greater than mode by the CS[2:0] bits. See the signed greater than mode part above.

- Overflow Protection

  The S bit in SR is also effective for arithmetic shift operation in the DSP unit. See section 3.5.11, Overflow Protection, for details.

(2) Logical Shift

Figure 3.18 shows the logical shift operation flow.

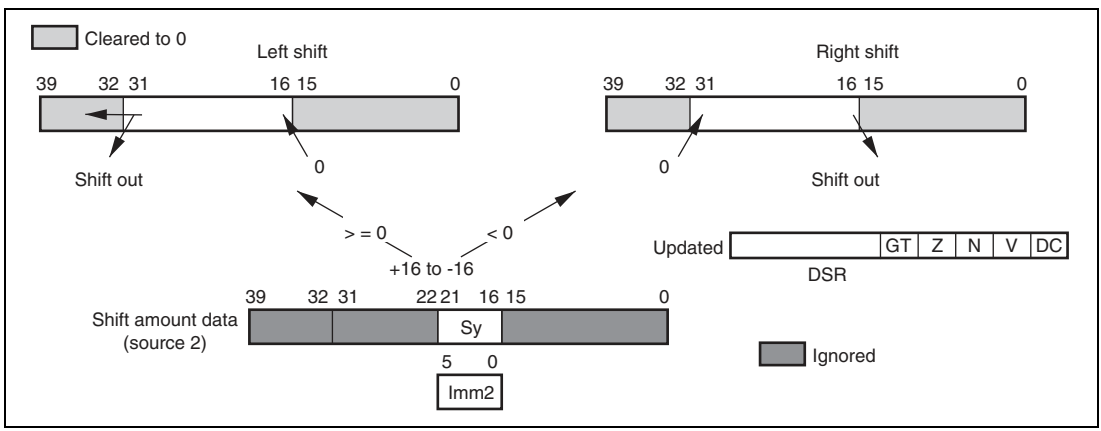


Figure 3.18 Logical Shift Operation Flow

As shown in figure 3.18, the logical shift operation uses the upper word of the source 1 operand and the destination operand. The lower word and guard-bit parts are ignored for the source operand and those of the destination operand are automatically cleared as in the ALU logical operations. The shift amount is specified by the source 2 operand as an integer data. The source 2 operand can be specified by either the register or immediate operand. The available shift range is from –16 to +16. Here, a negative value means the right shift, and a positive value means the left shift. It is possible for any source 2 operand to specify from –32 to +31, but the result is unknown if an invalid shift value is specified. In case of a shift with an immediate operand instruction, the source 1 operand must be the same register as the destination's. These operations are executed in the DSP stage, as shown in figure 3.10. The DSP stage is the same stage as the MA stage in which memory access is performed.

Every time a logical shift operation is executed, the DC, N, Z, V, and GT bits in DSR are basically updated in accordance with the operation result. In case of a conditional operation, they are not updated even though the specified condition is true and the operation is executed. In case of an unconditional operation, they are always updated in accordance with the operation result. The definition of the DC bit is selected by the CS[2:0] (condition selection) bits in DSR. The DC bit result is:

1. Carry or Borrow Mode: CS[2:0] = B'000
   The DC bit indicates the last shifted out data as the operation result.
2. Negative Value Mode: CS[2:0] = B'001
   Bit 31 of the operation result is loaded into the DC bit.
3. Zero Value Mode: CS[2:0] = B'010
   The DC bit is set to 1 when the operation result is zero; otherwise it is cleared to 0.
4. Overflow Mode: CS[2:0] = B'011
   The DC bit is always cleared to 0.
5. Signed Greater Than Mode: CS[2:0] = B'100
   The DC bit is always cleared to 0.
6. Signed Greater Than or Equal Mode: CS[2:0] = B'101
   The DC bit is always cleared.

The N bit always indicates the same state as the DC bit set in negative value mode by the CS[2:0] bits. See the negative value mode part above. The Z bit always indicates the same state as the DC bit set in zero value mode by the CS[2:0] bits. See the zero value mode part above. The V bit always indicates the same state as the DC bit set in overflow mode by the CS[2:0] bits, but it is always cleared in this operation. So is the GT bit.

3.5.9 Most Significant Bit Detection Operation

The PDMSB, most significant bit detection operation, is used to calculate the shift amount for normalization. Figure 3.19 shows the PDMSB operation flow and table 3.28 shows the operation definition. Table 3.29 shows the possible variations of this type of operation. The correspondence between each operand and registers is the same as for ALU fixed-point operations, as shown in table 3.21.

Note: The result of the MSB detection operation is basically 24 bits as well as ALU integer operation, the upper 16 bits of the base precision and eight bits of the guard-bit parts. When a register not providing the guard-bit parts is specified as a destination operand, the upper word of the operation result is input into the destination register.

As shown in figure 3.19, the PDMSB operation uses all bits as a source operand, but the destination operand is treated as an integer operation result because shift amount data for normalization should be integer data as described in section 3.5.8, Shift Operations. These operations are executed in the DSP stage, as shown in figure 3.10. The DSP stage is the same stage as the MA stage in which memory access is performed.

Every time a PDMSB operation is executed, the DC, N, Z, V, and GT bits in DSR are basically updated in accordance with the operation result. In case of a conditional operation, they are not updated, even though the specified condition is true, and the operation is executed. In case of an unconditional operation, they are always updated with the operation result.

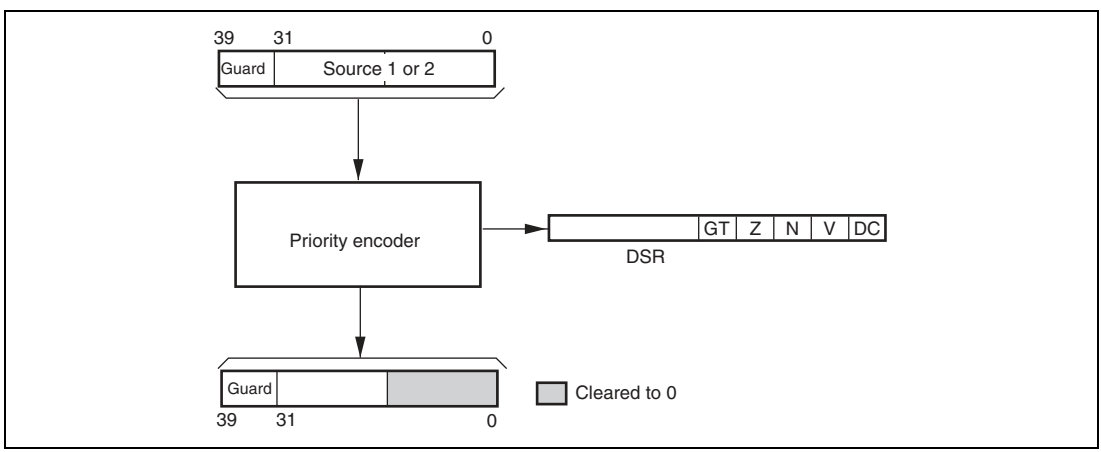


Figure 3.19 PDMSB Operation Flow

The definition of the DC bit is selected by the CS0 to CS2 (condition selection) bits in DSR. The DC bit result is

(1) Carry or Borrow Mode: CS[2:0] = B'000

The DC bit is always cleared to 0.

(2) Negative Value Mode: CS[2:0] = B'001

The DC bit is set when the operation result is a negative value, and cleared to 0 when the operation result is zero or a positive value.

(3) Zero Value Mode: CS[2:0] = B'010

The DC bit is set when the operation result is zero; otherwise it is cleared to 0.

(4) Overflow Mode: CS[2:0] = B'011

The DC bit is always cleared to 0.

(5) Signed Greater Than Mode: CS[2:0] = B'100

The DC bit is set to 1 when the operation result is a positive value; otherwise it is cleared to 0.

(6) Signed Greater Than or Equal Mode: CS[2:0] = B'101

The DC bit is set to 1 when the operation result is zero or a positive value; otherwise it is cleared to 0.

Table 3.28 Operation Definition of PDMSB

Source Data														Result for DST							
Guard Bit					Upper Word					Lower Word				Guard Bit	Upper Word						
39	38	…	33	32	31	30	29	28	…	3	2	1	0	39 to 32	31 to 22	21	20	19	18	17	16	Decimal
0	0	…	0	0	0	0	0	0	…	0	0	0	0	All 0	All 0	0	1	1	1	1	1	+31
0	0	…	0	0	0	0	0	0	…	0	0	0	1	All 0	All 0	0	1	1	1	1	0	+30
0	0	…	0	0	0	0	0	0	…	0	0	1	*	All 0	All 0	0	1	1	1	0	1	+29
0	0	…	0	0	0	0	0	0	…	0	1	*	*	All 0	All 0	0	1	1	1	0	0	+28
		:							:									:				
0	0	…	0	0	0	0	0	1	…	*	*	*	*	All 0	All 0	0	0	0	0	1	0	+2
0	0	…	0	0	0	0	1	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0	1	+1
0	0	…	0	0	0	1	*	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0	0	0
0	0	…	0	0	1	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1	1	–1
0	0	…	0	1	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1	0	–2
		:							:									:				
0	1	…	*	*	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	0	0	0	–8
1	0	…	*	*	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	0	0	0	–8
																		:				
1	1	…	1	0	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1	0	–2
1	1	…	1	1	0	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1	1	–1
1	1	…	1	1	1	0	*	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0	0	0
1	1	…	1	1	1	1	0	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0	1	+1
1	1	…	1	1	1	1	1	0	…	*	*	*	*	All 0	All 0	0	0	0	0	1	0	+2
		:							:									:				
1	1	…	1	1	1	1	1	1	…	1	0	*	*	All 0	All 0	0	1	1	1	0	0	+28
1	1	…	1	1	1	1	1	1	…	1	1	0	*	All 0	All 0	0	1	1	1	0	1	+29
1	1	…	1	1	1	1	1	1	…	1	1	1	0	All 0	All 0	0	1	1	1	1	0	+30
1	1	…	1	1	1	1	1	1	…	1	1	1	1	All 0	All 0	0	1	1	1	1	1	+31

Note: * means don't care.

Table 3.29 Variation of PDMSB Operation

Mnemonic	Function	Source	Source 2	Destination
PDMSB	MSB detection	Sx	—	Dz
		—	Sy	Dz

The N bit always indicates the same state as the DC bit set in negative value mode by the CS[2:0] bits. See the negative value mode part above. The Z bit always indicates the same state as the DC bit set in zero value mode by the CS[2:0] bits. See the zero value mode part above. The V bit is always cleared. The GT bit always indicates the same state as the DC bit set in signed greater than mode by the CS[2:0] bits. See the signed greater than mode part above.

3.5.10 Rounding Operation

The DSP unit provides the function that rounds from 32 bits to 16 bits. In case of providing guard-bit parts, it rounds from 40 bits to 24 bits. When a round instruction is executed, H'00008000 is added to the source operand data and then, the lower word is cleared. Figure 3.20 shows the rounding operation flow and figure 3.21 shows the operation definition. Table 3.30 shows the variation of this type of operation. The correspondence between each operand and registers is the same as ALU fixed-point operations as shown in table 3.21.

As shown in figure 3.21, the rounding operation uses full-size data for both source and destination operands. These operations are executed in the DSP stage as shown in figure 3.10. The DSP stage is the same stage as the MA stage in which memory access is performed.

The rounding operation is always executed unconditionally, so that the DC, N, Z, V, and GT bits in DSR are always updated in accordance with the operation result. The definition of the DC bit is selected by the CS0 to CS2 (condition selection) bits in DSR. The result of these condition code bits is the same as the ALU-fixed point arithmetic operations.

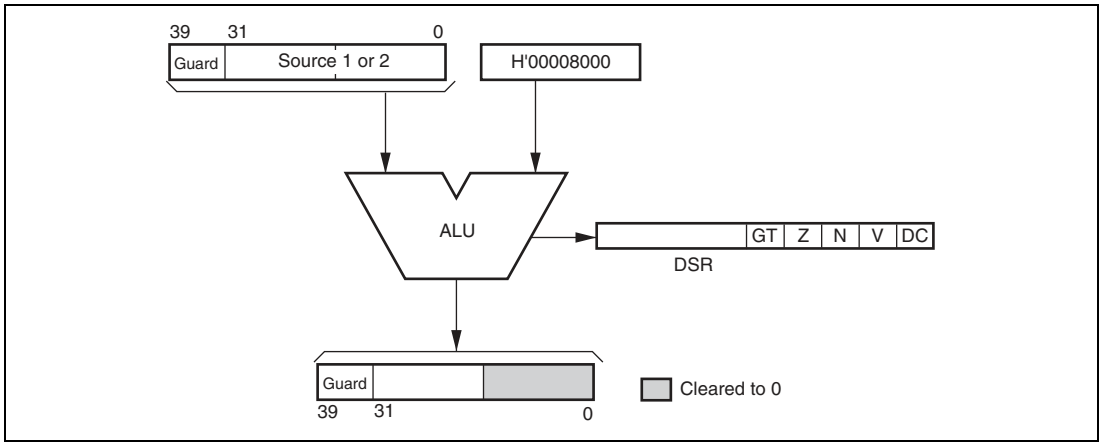


Figure 3.20 Rounding Operation Flow

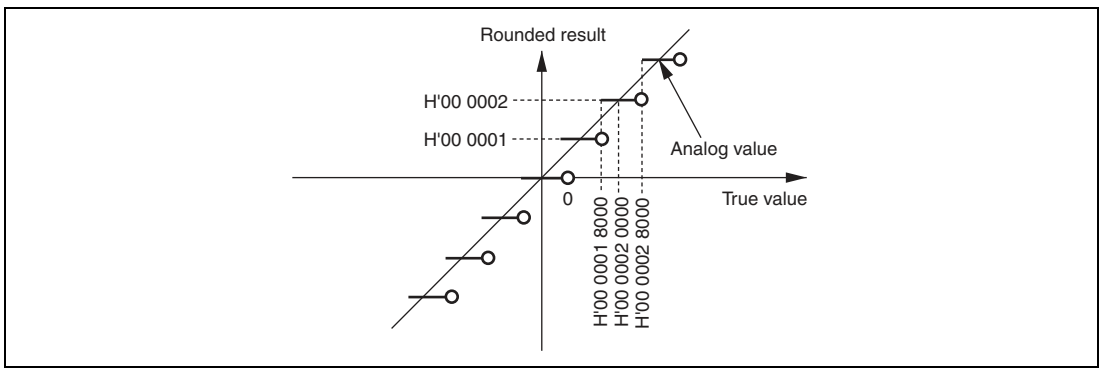


Figure 3.21 Definition of Rounding Operation

Table 3.30 Variation of Rounding Operation

Mnemonic	Function	Source 1	Source 2	Destination
PRND	Rounding	Sx	—	Dz
		—	Sy	Dz

- Overflow Protection

  The S bit in SR is effective for any rounding operations in the DSP unit. See section 3.5.11, Overflow Protection, for details.

3.5.11 Overflow Protection

The S bit in SR is effective for any arithmetic operations executed in the DSP unit, including the SH's standard multiply and MAC operations. The S bit in SR is used as the overflow protection enable bit. The arithmetic operation overflows when the operation result exceeds the range of two's complement representation without guard-bit parts. Table 3.31 shows the definition of overflow protection for fixed-point arithmetic operations, including fixed-point signed by signed multiplication described in section 3.5.7, Fixed-Point Multiply Operation. Table 3.32 shows the definition of overflow protection for integer arithmetic operations. The lower word of the saturation value of the integer arithmetic operation is don't care. Lower word value cannot be guaranteed.

When the overflow protection is effective, overflow never occurs. So, the V bit is cleared, and the DC bit is also cleared when the overflow mode is selected by the CS[2:0] bits.

Table 3.31 Definition of Overflow Protection for Fixed-Point Arithmetic Operations

Sign	Overflow Condition	Fixed Value	Hex Representation
Positive	Result > $1 - 2^{-31}$	$1 - 2^{-31}$	H'00 7FFF FFFF
Negative	Result < –1	–1	H'FF 8000 0000

Table 3.32 Definition of Overflow Protection for Integer Arithmetic Operations

Sign	Overflow Condition	Fixed Value	Hex Representation
Positive	Result > $2^{15} - 1$	$2^{15} - 1$	00 7FFF ****
Negative	Result < -2^{15}	-2^{15}	FF 8000 ****

Note: * means don't care.

3.5.12 Local Data Move Instruction

The DSP unit of this LSI provides additional two independent registers, MACL and MACH, in order to support CPU standard multiply/MAC operations. They can be also used as temporary storage registers by local data move instructions between MACH/L and other DSP registers. Figure 3.22 shows the flow of seven local data move instructions. Table 3.33 shows the variation of this type of instruction.

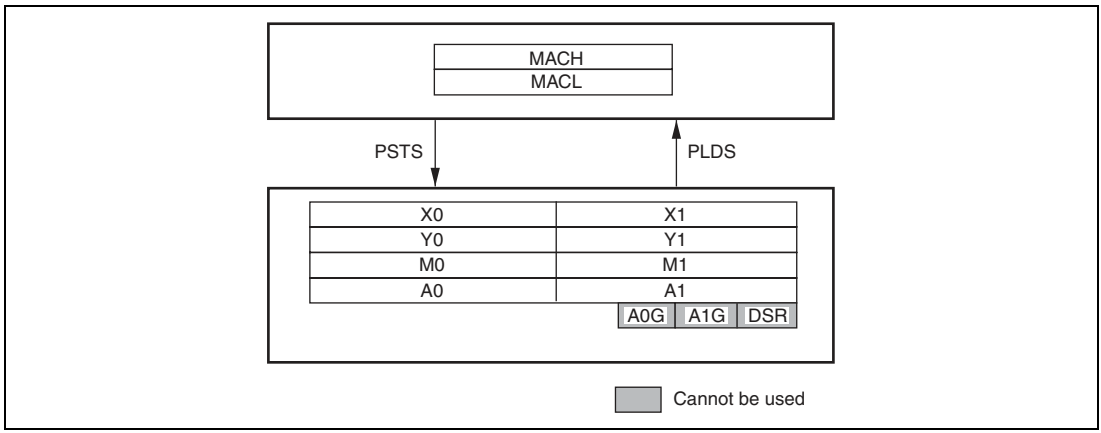


Figure 3.22 Local Data Move Instruction Flow

Table 3.33 Variation of Local Data Move Operations

Mnemonic	Function	Operand
PLDS	Data move from DSP register to MACL/MACH	Dz
PSTS	Data move from MACL/MACH to DSP register	Dz

This instruction is very similar to other transfer instructions. If either the A0 or A1 register is specified as the destination operand of PSTS, the signed bit is sign-extended and copied into the corresponding guard-bit parts, A0G or A1G. The DC bit in DSR and other condition code bits are not updated regardless of the instruction result. This instruction can operate as a conditional. This instruction can operate with MOVX and MOVY in parallel.

3.5.13 Operand Conflict

When an identical destination operand is specified with multiple parallel instructions, data conflict occurs. Table 3.34 shows the correspondence between each operand and registers.

Table 3.34 Correspondence between Operands and Registers

		X-Memory Load			Y-Memory Load			6-Instruction ALU			3-Instruction Multiply			3-Instruction ALU		
		Ax	Ix	Dx	Ay	Iy	Dy	Sx	Sy	Du	Se	Sf	Dg	Sx	Sy	Dz
DSP Registers	A0							*1		*2			*2	*1		*1
	A1							*1		*2	*1	*1	*2	*1		*1
	M0								*1				*1		*1	*1
	M1								*1				*1		*1	*1
	X0			*2				*1		*2	*1	*1		*1		*2
	X1			*2				*1			*1			*1		*2
	Y0						*2		*1	*2	*1	*1			*1	*2
	Y1						*2		*1			*1			*1	*2

Notes: 1. Registers available for operands
2. Registers available for operands (when there is operand conflict)

There are three cases of operand conflict problems.

- When ALU operation and multiply instructions specify the same destination operand (Du and Dg)
- When X-memory load and ALU operation specify the same destination operand (Dx and Du, or Dz)
- When Y-memory load and ALU operation specify the same destination operand (Dy and Du, or Dz)

In these cases above, the result is not guaranteed.

3.6 DSP Extended Function Instruction Set

3.6.1 CPU Extended Instructions

Table 3.35 DSP Mode Extended System Control Instructions

Instruction	Instruction Code	Operation	Execution States	T Bit
SETRC #imm	10000010iiiiiiii	imm → RC (of SR)	1	–
SETRC Rn	0100nnnn00010100	Rn[11:0] → RC(of SR)	1	–
LDRS @(disp,PC)	10001100dddddddd	(disp x 2 + PC) → RS	1	–
LDRE @(disp,PC)	10001110dddddddd	(disp x 2 + PC) → RE	1	–
STC MOD,Rn	0000nnnn01010010	MOD →Rn	1	–
STC RS,Rn	0000nnnn01100010	RS → Rn	1	–
STC RE,Rn	0000nnnn01110010	RE → Rn	1	–
STS DSR,Rn	0000nnnn01101010	DSR → Rn	1	–
STS A0,Rn	0000nnnn01111010	A0 → Rn	1	–
STS X0,Rn	0000nnnn10001010	X0 → Rn	1	–
STS X1,Rn	0000nnnn10011010	X1 → Rn	1	–
STS Y0,Rn	0000nnnn10101010	Y0 → Rn	1	–
STS Y1,Rn	0000nnnn10111010	Y1 → Rn	1	–
STS.L DSR,@-Rn	0100nnnn01100010	Rn-4 → Rn, DSR → (Rn)	1	–
STS.L A0,@-Rn	0100nnnn01110010	Rn-4 → Rn, A0 → (Rn)	1	–
STS.L X0,@-Rn	0100nnnn10000010	Rn-4 → Rn, X0 → (Rn)	1	–
STS.L X1,@-Rn	0100nnnn10010010	Rn-4 → Rn, X1 → (Rn)	1	–
STS.L Y0,@-Rn	0100nnnn10100010	Rn-4 → Rn, Y0 → (Rn)	1	–
STS.L Y1,@-Rn	0100nnnn10110010	Rn-4 → Rn, Y1 →(Rn)	1	–
STC.L MOD,@-Rn	0100nnnn01010011	Rn-4 → Rn, MOD → (Rn)	1	–
STC.L RS,@-Rn	0100nnnn01100011	Rn-4 → Rn, RS → (Rn)	1	–
STC.L RE,@-Rn	0100nnnn01110011	Rn-4 → Rn, RE → (Rn)	1	–
LDS.L @Rn + ,DSR	0100nnnn01100110	(Rn) → DSR, Rn + 4→Rn	1	–
LDS.L @Rn + ,A0	0100nnnn01110110	(Rn) → A0, Rn + 4 → Rn	1	–
LDS.L @Rn + ,X0	0100nnnn10000110	(Rn) → X0, Rn + 4 → Rn	1	–
LDS.L @Rn + ,X1	0100nnnn10010110	(Rn) → X1, Rn + 4 → Rn	1	–
LDS.L @Rn + ,Y0	0100nnnn10100110	(Rn) → Y0, Rn + 4 → Rn	1	–

Instruction	Instruction Code	Operation	Execution States	T Bit
LDS.L @Rn + ,Y1	0100nnnn10110110	(Rn) → Y1, Rn + 4 → Rn	1	–
LDC.L @Rn + ,MOD	0100nnnn01010111	(Rn) → MOD, Rn + 4 → Rn	4	–
LDC.L @Rn + ,RS	0100nnnn01100111	(Rn) → RS, Rn + 4 → Rn	4	–
LDC.L @Rn + ,RE	0100nnnn01110111	(Rn) → RE, Rn + 4 → Rn	4	–
LDS Rn,DSR	0100nnnn01101010	Rn → DSR	1	–
LDS Rn,A0	0100nnnn01111010	Rn → A0	1	–
LDS Rn,X0	0100nnnn10001010	Rn → X0	1	–
LDS Rn,X1	0100nnnn10011010	Rn → X1	1	–
LDS Rn,Y0	0100nnnn10101010	Rn → Y0	1	–
LDS Rn,Y1	0100nnnn10111010	Rn → Y1	1	–
LDC Rn,MOD	0100nnnn01011110	Rn → MOD	4	–
LDC Rn,RS	0100nnnn01101110	Rn → RS	4	–
LDC Rn,RE	0100nnnn01111110	Rn → RE	4	–

3.6.2 Double-Data Transfer Instructions

Table 3.36 Double Data Transfer Instruction

Instruction		Instruction Code	Operation	Execution States	DC
X memory data transfer	NOPX	1111000*0*0*00**	X memory no access	1	–
	MOVX.W @Ax,Dx	111100A*D*0*01**	(Ax) → MSW of Dx, 0 → LSW of Dx	1	–
	MOVX.W @Ax+,Dx	111100A*D*0*10**	(Ax) → MSW of Dx, 0 → LSW of Dx, Ax + 2 → Ax	1	–
	MOVX.W @Ax+Ix,Dx	111100A*D*0*11**	(Ax) → MSW of Dx, 0 → LSW of Dx, Ax + Ix → Ax	1	–
	MOVX.W Da,@Ax	111100A*D*1*01**	MSW of Da → (Ax)	1	–
	MOVX.W Da,@Ax+	111100A*D*1*10**	MSW of Da → (Ax), Ax + 2 → Ax	1	–
	MOVX.W Da,@Ax+Ix	111100A*D*1*11**	MSW of Da → (Ax), Ax + Ix → Ax	1	–
Y memory data transfer	NOPY	111100*0*0*0**00	Y memory no access	1	–
	MOVY.W @Ay,Dy	111100*A*D*0**01	(Ay) → MSW of Dy, 0 → LSW of Dy	1	–
	MOVY.W @Ay+,Dy	111100*A*D*0**10	(Ay) → MSW of Dy, 0 → LSW of Dy, Ay + 2 → Ay	1	–
	MOVY.W @Ay+Iy,Dy	111100*A*D*0**11	(Ay) → MSW of Dy, 0 → LSW of Dy, Ay + Iy → Ay	1	–
	MOVY.W Da,@Ay	111100*A*D*1**01	MSW of Da → (Ay)	1	–
	MOVY.W Da,@Ay+	111100*A*D*1**10	MSW of Da → (Ay), Ay + 2 → Ay	1	–
	MOVY.W Da,@Ay+Iy	111100*A*D*1**11	MSW of Da → (Ay), Ay + Iy → Ay	1	–

3.6.3 Single-Data Transfer Instructions

Table 3.37 Single Data Transfer Instructions

Instruction	Instruction Code	Operation	Execution States	DC	Category
MOVS.W @-As,Ds	111101AADDDD0000	As-2 → As, (As) → MSW of Ds, 0 → LSW of Ds	1	–	
MOVS.W @As,Ds	111101AADDDD0100	(As) → MSW of Ds, 0 → LSW of Ds	1	–	
MOVS.W @As+,Ds	111101AADDDD1000	(As) → MSW of Ds, 0 → LSW of Ds, As + 2 → As	1	–	
MOVS.W @As+Ix,Ds	111101AADDDD1100	(Asc) → MSW of Ds, 0 → LSW of Ds, As + Ix → As	1	–	
MOVS.W Ds,@-As	111101AADDDD0001	As-2 → As, MSW of Ds → (As)	1	–	*
MOVS.W Ds,@As	111101AADDDD0101	MSW of Ds → (As)	1	–	*
MOVS.W Ds,@As+	111101AADDDD1001	MSW of Ds → (As), As + 2 → As	1	–	*
MOVS.W Ds,@As+Ix	111101AADDDD1101	MSW of Ds → (As), As + Ix → As	1	–	*
MOVS.L @-As,Ds	111101AADDDD0010	As-4 → As, (As) → Ds	1	–	
MOVS.L @As,Ds	111101AADDDD0110	(As) → Ds	1	–	
MOVS.L @As+,Ds	111101AADDDD1010	(As) → Ds, As + 4 → As	1	–	
MOVS.L @As+Ix,Ds	111101AADDDD1110	(As) → Ds, As + Ix → As	1	–	
MOVS.L Ds,@-As	111101AADDDD0011	As-4 → As, Ds → (As)	1	–	
MOVS.L Ds,@As	111101AADDDD0111	Ds → (As)	1	–	
MOVS.L Ds,@As+	111101AADDDD1011	Ds → (As), As + 4 → As	1	–	
MOVS.L Ds,@As+Ix	111101AADDDD1111	Ds → (As), As + Ix → As	1	–	

Note: * If guard bit registers A0G and A1G are specified in source operand Ds, the data is output to the LDB[7:0] bus and the sign bit is copied into the upper bits, [31:8].

The correspondence between DSP data transfer operands and registers is shown in table 3.38.

Table 3.38 Correspondence between DSP Data Transfer Operands and Registers

Register		Ax	Ix	Dx	Ay	Iy	Dy	Da	As	Ds
SH register	R0									
	R1									
	R2 (As2)								Yes	
	R3 (As3)								Yes	
	R4 (Ax0)	Yes							Yes	
	R5 (Ax1)	Yes							Yes	
	R6 (Ay0)				Yes					
	R7 (Ay1)				Yes					
	R8 (Ix)		Yes							
	R9 (Iy)					Yes				
DSP register	A0							Yes		Yes
	A1							Yes		Yes
	M0									Yes
	M1									Yes
	X0			Yes						Yes
	X1			Yes						Yes
	Y0						Yes			Yes
	Y1						Yes			Yes
	A0G									Yes
	A1G									Yes

3.6.4 DSP Operation Instructions

Table 3.39 DSP Operation Instructions

Instruction		Instruction Code	Operation	Execution States	DC
	PMULS Se,Sf, Dg	111110********** 0100eeff0000gg00	Se*Sf → Dg (Signed)	1	–
	PADD Sx,Sy,Du PMULS Se,Sf,Dg	111110********** 0111eeffxxyygguu	Sx + Sy → Du Se*Sf → Dg (Signed)	1	*
	PSUB Sx,Sy,Du PMULS Se,Sf,Dg	111110********** 0110eeffxxyygguu	Sy-Sy → Du Se*Sf → Dg (Signed)	1	*
	PADD Sx,Sy,Dz	111110********** 10110001xxyyzzzz	Sx + Sy → Dz	1	*
DCT	PADD Sx,Sy,Dz	111110********** 10110010xxyyzzzz	If DC = 1, Sx + Sy → Dz If DC = 0, nop	1	–
DCF	PADD Sx,Sy,Dz	111110********** 10110011xxyyzzzz	If DC = 0, Sx + Sy → Dz If DC = 1, nop	1	–
	PSUB Sx,Sy,Dz	111110********** 10100001xxyyzzzz	Sx-Sy → Dz	1	*
DCT	PSUB Sx,Sy,Dz	111110********** 10100010xxyyzzzz	If DC = 1, Sx-Sy → Dz If DC = 0, nop	1	–
DCF	PSUB Sx,Sy,Dz	111110********** 10100011xxyyzzzz	If DC = 0, Sx-Sy → Dz If DC = 1, nop	1	–
	PSHA Sx,Sy,Dz	111110********** 10010001xxyyzzzz	If Sy >= 0, Sx<<Sy → Dz (arithmetic shift) If Sy < 0, Sx>>Sy → Dz	1	*
DCT	PSHA Sx,Sy,Dz	111110********** 10010010xxyyzzzz	If DC = 1 & Sy >= 0, Sx<<Sy → Dz (arithmetic shift) If DC=1 & Sy<0, Sx>>Sy → Dz If DC=0, nop	1	–
DCF	PSHA Sx,Sy,Dz	111110********** 10010011xxyyzzzz	If DC = 0 & Sy >= 0, Sx<<Sy → Dz (arithmetic shift) If DC = 0 & Sy < 0, Sx>>Sy → Dz If DC = 1, nop	1	–
	PSHL Sx,Sy,Dz	111110********** 10000001xxyyzzzz	If Sy >= 0, Sx<<Sy → Dz (logical shift) If Sy < 0, Sx>>Sy → Dz	1	*

Instruction		Instruction Code	Operation	Execution States	DC
DCT	PSHL Sx,Sy,Dz	111110********** 10000010xxyyzzzz	If DC = 1 & Sy >= 0, Sx<<Sy → Dz (logical shift) If DC = 1 & Sy < 0, Sx>>Sy → Dz If DC = 0, nop	1	–
DCF	PSHL Sx,Sy,Dz	111110********** 10000011xxyyzzzz	If DC = 0 & Sy >= 0, Sx<<Sy → Dz (logical shift) If DC = 0 & Sy < 0, Sx>>Sy → Dz If DC = 1, nop	1	–
	PCOPY Sx,Dz	111110********** 11011001xx00zzzz	Sx →Dz	1	*
	PCOPY Sy,Dz	111110********** 1111100100yyzzzz	Sy →Dz	1	*
DCT	PCOPY Sx,Dz	111110********** 11011010xx00zzzz	If DC = 1, Sx → Dz If DC = 0, nop	1	–
DCT	PCOPY Sy,Dz	111110********** 1111101000yyzzzz	If DC = 1, Sy → Dz If DC = 0, nop	1	–
DCF	PCOPY Sx,Dz	111110********** 11011011xx00zzzz	If DC = 0, Sx → Dz If DC = 1, nop	1	–
DCF	PCOPY Sy,Dz	111110********** 1111101100yyzzzz	If DC = 0, Sy → Dz If DC = 1, nop	1	–
	PDMSB Sx,Dz	111110********** 10011101xx00zzzz	Sx → Dz normalization count shift value	1	*
	PDMSB Sy,Dz	111110********** 1011110100yyzzzz	Sy → Dz normalization count shift value	1	*
DCT	PDMSB Sx,Dz	111110********** 10011110xx00zzzz	If DC = 1, normalization count shift value Sx → Dz If DC = 0, nop	1	–
DCT	PDMSB Sy,Dz	111110********** 1011111000yyzzzz	If DC = 1, normalization count shift value Sy → Dz If DC = 0, nop	1	–
DCF	PDMSB Sx,Dz	111110********** 10011111xx00zzzz	If DC = 0, normalization count shift value Sx → Dz If DC = 1, nop	1	–

Instruction		Instruction Code	Operation	Execution States	DC
DCF	PDMSB Sy,Dz	111110**********	If DC = 0, normalization count shift value Sy → Dz	1	–
		1011111100yyzzzz	If DC=1, nop		
	PINC Sx,Dz	111110**********	MSW of Sx + 1 → Dz	1	*
		10011001xx00zzzz			
	PINC Sy,Dz	111110**********	MSW of Sy + 1 → Dz	1	*
		1011100100yyzzzz			
DCT	PINC Sx,Dz	111110**********	If DC = 1, MSW of Sx + 1 → Dz	1	–
		10011010xx00zzzz	If DC = 0, nop		
DCT	PINC Sy,Dz	111110**********	If DC = 1, MSW of Sy + 1 → Dz	1	–
		1011101000yyzzzz	If DC = 0, nop		
DCF	PINC Sx,Dz	111110**********	If DC = 0, MSW of Sx + 1 → Dz	1	–
		10011011xx00zzzz	If DC = 1, nop		
DCF	PINC Sy,Dz	111110**********	If DC = 0, MSW of Sy + 1 → Dz	1	–
		1011101100yyzzzz	If DC = 1, nop		
	PNEG Sx,Dz	111110**********	0-Sx → Dz	1	*
		11001001xx00zzzz			
	PNEG Sy,Dz	111110**********	0-Sy → Dz	1	*
		1110100100yyzzzz			
DCT	PNEG Sx,Dz	111110**********	If DC = 1, 0-Sx → Dz	1	–
		11001010xx00zzzz	If DC = 0, nop		
DCT	PNEG Sy,Dz	111110**********	If DC = 1, 0-Sy → Dz	1	–
		1110101000yyzzzz	If DC = 0, nop		
DCF	PNEG Sx,Dz	111110**********	If DC = 0, 0-Sx → Dz	1	–
		11001011xx00zzzz	If DC = 1, nop		
DCF	PNEG Sy,Dz	111110**********	If DC = 0, 0-Sy → Dz	1	–
		1110101100yyzzzz	If DC = 1, nop		
	POR Sx,Sy,Dz	111110**********	Sx | Sy → Dz	1	*
		10110101xxyyzzzz			
DCT	POR Sx,Sy,Dz	111110**********	If DC = 1, Sx | Sy → Dz	1	–
		10110110xxyyzzzz	If DC = 0, nop		
DCF	POR Sx,Sy,Dz	111110**********	If DC = 0, Sx | Sy → Dz	1	–
		10110111xxyyzzzz	If DC = 1, nop		

Instruction		Instruction Code	Operation	Execution States	DC
	PAND Sx,Sy,Dz	111110**********	Sx & Sy → Dz	1	*
		10010101xxyyzzzz			
DCT	PAND Sx,Sy,Dz	111110**********	If DC = 1, Sx & Sy → Dz	1	–
		10010110xxyyzzzz	If DC = 0, nop		
DCF	PAND Sx,Sy,Dz	111110**********	If DC = 0, Sx & Sy → Dz	1	–
		10010111xxyyzzzz	If DC = 1, nop		
	PXOR Sx,Sy,Dz	111110**********	Sx ^ Sy → Dz	1	*
		10100101xxyyzzzz			
DCT	PXOR Sx,Sy,Dz	111110**********	If DC = 1, Sx ^ Sy → Dz	1	–
		10100110xxyyzzzz	If DC = 0, nop		
DCF	PXOR Sx,Sy,Dz	111110**********	If DC = 0, Sx ^ Sy → Dz	1	–
		10100111xxyyzzzz	If DC = 1, nop		
	PDEC Sx,Dz	111110**********	Sx [39:16]-1 → Dz	1	*
		10001001xx00zzzz			
DCT	PDEC Sx,Dz	111110**********	If DC = 1, Sx [39:16]-1 → Dz	1	–
		10001010xx00zzzz	If DC = 0, nop		
DCF	PDEC Sx,Dz	111110**********	If DC = 0, Sx [39:16]-1 → Dz	1	–
		10001011xx00zzzz	If DC = 1, nop		
	PDEC Sy,Dz	111110**********	Sy [31:16]-1 → Dz	1	*
		1010100100yyzzzz			
DCT	PDEC Sy,Dz	111110**********	If DC = 1, Sy [31:16]-1 → Dz	1	–
		1010101000yyzzzz	If DC = 0, nop		
DCF	PDEC Sy,Dz	111110**********	If DC = 0, Sy [31:16]-1 → Dz	1	–
		1010101100yyzzzz	If DC = 1, nop		
	PCLR Dz	111110**********	h'00000000 → Dz	1	*
		100011010000zzzz			
DCT	PCLR Dz	111110**********	If DC = 1, h'00000000 → Dz	1	–
		100011100000zzzz	If DC = 0, nop		
DCF	PCLR Dz	111110**********	If DC = 0, h'00000000 → Dz	1	–
		100011110000zzzz	If DC = 1, nop		
	PSHA #imm,Dz	111110**********	If imm>=0, Dz<<imm → Dz (arithmetic shift)	1	*
		00010iiiiiizzzz	If imm<0, Dz>>imm → Dz		

Instruction		Instruction Code	Operation	Execution States	DC
	PSHL #imm,Dz	111110********** 00000iiiiiizzzz	If imm>=0, Dz<<imm → Dz (logical shift) If imm<0, Dz>>imm → Dz	1	*
	PSTS MACH,Dz	111110********** 1100110100000zzzz	MACH → Dz	1	–
DCT	PSTS MACH,Dz	111110********** 1100111000000zzzz	If DC = 1, MACH → Dz	1	–
DCF	PSTS MACH,Dz	111110********** 1100111100000zzzz	If DC = 0, MACH → Dz	1	–
	PSTS MACL,Dz	111110********** 1101110100000zzzz	MACL → Dz	1	–
DCT	PSTS MACL,Dz	111110********** 1101111000000zzzz	If DC = 1, MACL → Dz	1	–
DCF	PSTS MACL,Dz	111110********** 1101111100000zzzz	If DC = 0, MACL → Dz	1	–
	PLDS Dz,MACH	111110********** 1110110100000zzzz	Dz → MACH	1	–
DCT	PLDS Dz,MACH	111110********** 1110111000000zzzz	If DC = 1, Dz → MACH	1	–
DCF	PLDS Dz,MACH	111110********** 1110111100000zzzz	If DC = 0, Dz → MACH	1	–
	PLDS Dz,MACL	111110********** 1111110100000zzzz	Dz → MACL	1	–
DCT	PLDS Dz,MACL	111110********** 1111111000000zzzz	If DC = 1, Dz → MACL	1	–
DCF	PLDS Dz,MACL	111110********** 1111111100000zzzz	If DC = 0, Dz → MACL	1	–
	PADDC Sx,Sy,Dz	111110********** 10110000xxyyzzzz	Sx + Sy + DC →Dz Carry → DC	1	Carry
	PSUBC Sx,Sy, Dz	111110********** 10100000xxyyzzzz	Sx-Sy-DC → Dz Borrow → DC	1	Borrow
	PCMP Sx,Sy	111110********** 10000100xxyy0000	Sx-Sy → DC update	1	*

Instruction	Instruction Code	Operation	Execution States	DC
PABS Sx,Dz	111110********** 10001000xx00zzzz	If Sx<0, 0-Sx → Dz If Sx>=0, Sx→ Dz	1	*
PABS Sy,Dz	111110********** 1010100000yyzzzz	If Sy<0, 0-Sy → Dz If Sy>=0, Sy → Dz	1	*
PRND Sx,Dz	111110********** 10011000xx00zzzz	Sx + h'00008000 → Dz LSW of Dz → h'0000	1	*
PRND Sy,Dz	111110********** 1011100000yyzzzz	Sy + h'00008000 → Dz LSW of Dz → h'0000	1	*

Note: * See table 3.19.

3.6.5 Operation Code Map in DSP Mode

Table 3.40 shows the operation code map including an instruction codes extended in the DSP mode.

Table 3.40 Operation Code Map

Instruction Code				Fx: 0000		Fx: 0001		Fx: 0010		Fx: 0011 to 1111	
MSB			LSB	MD: 00		MD: 01		MD: 10		MD: 11	
0000	Rn	Fx	0000								
0000	Rn	Fx	0001								
0000	Rn	00MD	0010	STC	SR, Rn	STC	GBR, Rn	STC	VBR, Rn	STC	SSR, Rn
0000	Rn	01MD	0010	STC	SPC, Rn	STC	MOD, Rn	STC	RS, Rn	STC	RE, Rn
0000	Rn	10MD	0010	STC	R0_BANK, Rn	STC	R1_BANK, Rn	STC	R2_BANK, Rn	STC	R3_BANK, Rn
0000	Rn	11MD	0010	STC	R4_BANK, Rn	STC	R5_BANK, Rn	STC	R6_BANK, Rn	STC	R7_BANK, Rn
0000	Rm	00MD	0011	BSRF	Rm			BRAF	Rm		
0000	Rm	10MD	0011	PREF	@Rm						
0000	Rn	Rm	01MD	MOV.B	Rm, @(R0, Rn)	MOV.W	Rm, @(R0, Rn)	MOV.L	Rm,@(R0, Rn)	MUL.L	Rm, Rn
0000	0000	00MD	1000	CLRT		SETT		CLRMAC		LDTLB	
0000	0000	01MD	1000	CLRS		SETS					
0000	0000	10MD	1000								
0000	0000	11MD	1000								
0000	0000	Fx	1001	NOP		DIV0U					
0000	0000	Fx	1010								
0000	0000	Fx	1011	RTS		SLEEP		RTE			
0000	Rn	Fx	1000								
0000	Rn	Fx	1001					MOVT	Rn		
0000	Rn	00MD	1010	STS	MACH, Rn	STS	MACL, Rn	STS	PR, Rn		
0000	Rn	01MD	1010					STS	DSR, Rn	STS	A0, Rn
0000	Rn	10MD	1010	STS	X0, Rn	STS	X1, Rn	STS	Y0, Rn	STS	Y1, Rn
0000	Rn	Fx	1011								

Instruction Code				Fx: 0000		Fx: 0001		Fx: 0010		Fx: 0011 to 1111	
MSB			LSB	MD: 00		MD: 01		MD: 10		MD: 11	
0000	Rn	Rm	11MD	MOV. B @(R0, Rm), Rn		MOV.W @(R0, Rm), Rn		MOV.L @(R0, Rm), Rn		MAC.L @Rm+,@Rn+	
0001	Rn	Rm	disp	MOV.L Rm, @(disp:4, Rn)							
0010	Rn	Rm	00MD	MOV.B	Rm, @Rn	MOV.W	Rm, @Rn	MOV.L	Rm, @Rn		
0010	Rn	Rm	01MD	MOV.B	Rm, @–Rn	MOV.W	Rm, @–Rn	MOV.L	Rm, @–Rn	DIV0S	Rm, Rn
0010	Rn	Rm	10MD	TST	Rm, Rn	AND	Rm, Rn	XOR	Rm, Rn	OR	Rm, Rn
0010	Rn	Rm	11MD	CMP/STR	Rm, Rn	XTRCT	Rm, Rn	MULU.W	Rm, Rn	MULSW	Rm, Rn
0011	Rn	Rm	00MD	CMP/EQ	Rm, Rn			CMP/HS	Rm, Rn	CMP/GE	Rm, Rn
0011	Rn	Rm	01MD	DIV1	Rm, Rn	DMULU.L	Rm,Rn	CMP/HI	Rm, Rn	CMP/GT	Rm, Rn
0011	Rn	Rm	10MD	SUB	Rm, Rn			SUBC	Rm, Rn	SUBV	Rm, Rn
0011	Rn	Rm	11MD	ADD	Rm, Rn	DMULS.L	Rm,Rn	ADDC	Rm, Rn	ADDV	Rm, Rn
0100	Rn	Fx	0000	SHLL	Rn	DT	Rn	SHAL	Rn		
0100	Rn	Fx	0001	SHLR	Rn	CMP/PZ	Rn	SHAR	Rn		
0100	Rn	Fx	0010	STS.L MACH, @–Rn		STS.L MACL, @–Rn		STS.L PR, @–Rn			
0100	Rn	00MD	0011	STC.L	SR, @–Rn	STC.L	GBR, @–Rn	STC.L VBR,	@–Rn	STC.L	SSR, @–Rn
0100	Rn	01MD	0011	STC.L	SPC, @–Rn	STC.L	MOD, @–Rn	STC.L RS,	@–Rn	STC.L	RE, @–Rn
0100	Rn	10MD	0011	STC.L R0_BANK, @–Rn		STC.L R1_BANK, @–Rn		STC.L R2_BANK, @–Rn		STC.L R3_BANK, @–Rn	
0100	Rn	11MD	0011	STC.L R4_BANK, @–Rn		STC.L R5_BANK, @–Rn		STC.L R6_BANK, @–Rn		STC.L R7_BANK, @–Rn	
0100	Rn	Fx	0100	ROTL	Rn	SETRC	Rn	ROTCL	Rn		
0100	Rn	Fx	0101	ROTR	Rn	CMP/PL	Rn	ROTCR	Rn		
0100	Rm	00MD	0110	LDS.L @Rm+, MACH		LDS.L @Rm+, MACL		LDS.L @Rm+, PR			
0100	Rm	01MD	0110					LDS.L @Rm+, DSR		LDS.L @Rm+, A0	
0100	Rm	10MD	0110	LDS.L	@Rm+, X0	LDS.L	@Rm+, X1	LDS.L	@Rm+, Y0	LDS.L	@Rm+, Y1

Instruction Code				Fx: 0000		Fx: 0001		Fx: 0010		Fx: 0011 to 1111	
MSB			LSB	MD: 00		MD: 01		MD: 10		MD: 11	
0100	Rm	00MD	0111	LDC.L	@Rm+, SR	LDC.L	@Rm+, GBR	LDC.L	@Rm+, VBR	LDC.L	@Rm+, SSR
0100	Rm	01MD	0111	LDC.L	@Rm+, SPC	LDC.L	@Rm+, MOD	LDC.L	@Rm+, RS	LDC.L	@Rm+, RE
0100	Rm	10MD	0111	LDC.L @Rm+, R0_BANK		LDC.L @Rm+, R1_BANK		LDC.L @Rm+, R2_BANK		LDC.L @Rm+, R3_BANK	
0100	Rm	11MD	0111	LDC.L @Rm+, R4_BANK		LDC.L @Rm+, R5_BANK		LDC.L @Rm+, R6_BANK		LDC.L @Rm+, R7_BANK	
0100	Rn	Fx	1000	SHLL2	Rn	SHLL8	Rn	SHLL16	Rn		
0100	Rn	Fx	1001	SHLR2	Rn	SHLR8	Rn	SHLR16	Rn		
0100	Rm	00MD	1010	LDS	Rm, MACH	LDS	Rm, MACL	LDS	Rm, PR		
0100	Rm	01MD	1010					LDS	Rm, DSR	LDS	Rm, A0
0100	Rm	10MD	1010	LDS	Rm, X0	LDS	Rm, X1	LDS	Rm, Y0	LDS	Rm, Y1
0100	Rm/Rn	Fx	1011	JSR	@Rm	TAS.B	@Rn	JMP	@Rm		
0100	Rn	Rm	1100	SHAD	Rm, Rn						
0100	Rn	Rm	1101	SHLD	Rm, Rn						
0100	Rm	00MD	1110	LDC	Rm, SR	LDC	Rm, GBR	LDC	Rm, VBR	LDC	Rm, SSR
0100	Rm	01MD	1110	LDC	Rm, SPC	LDC	Rm, MOD	LDC	Rm, RS	LDC	Rm, RE
0100	Rm	10MD	1110	LDC Rm, R0_BANK		LDC Rm, R1_BANK		LDC Rm, R2_BANK		LDC Rm, R3_BANK	
0100	Rm	11MD	1110	LDC Rm, R4_BANK		LDC Rm, R5_BANK		LDC Rm, R6_BANK		LDC Rm, R7_BANK	
0100	Rn	Rm	1111	MAC.W @Rm+, @Rn+							
0101	Rn	Rm	disp	MOV.L @ (disp:4, Rm), Rn							
0110	Rn	Rm	00MD	MOV.B	@Rm, Rn	MOV.W	@Rm, Rn	MOV.L	@Rm, Rn	MOV	Rm, Rn
0110	Rn	Rm	01MD	MOV.B	@Rm+, Rn	MOV.W	@Rm+, Rn	MOV.L	@Rm+, Rn	NOT	Rm, Rn
0110	Rn	Rm	10MD	SWAP.B	Rm, Rn	SWAP.W	Rm, Rn	NEGC	Rm, Rn	NEG	Rm, Rn
0110	Rn	Rm	11MD	EXTU.B	Rm, Rn	EXTU.W	Rm, Rn	EXTS.B	Rm, Rn	EXTS.W	Rm, Rn
0111	Rn	imm		ADD	#imm : 8, Rn						
1000	00MD	Rn	disp	MOV.B R0, @(disp: 4, Rn)		MOV.W R0, @(disp: 4, Rn)		SETRC	#imm		
		imm									

Instruction Code				Fx: 0000		Fx: 0001		Fx: 0010		Fx: 0011 to 1111	
MSB			LSB	MD: 00		MD: 01		MD: 10		MD: 11	
1000	01MD	Rm	disp	MOV.B @(disp:4, Rm), R0		MOV.W @(disp: 4, Rm), R0					
1000	10MD	imm/disp		CMP/EQ	#imm:8, R0	BT	disp: 8			BF	disp: 8
1000	11MD	imm/disp		LDRS	@(disp:8,PC)	BT/S	disp: 8	LDRE	@(disp:8,PC)	BF/S	disp: 8
1001	Rn	disp		MOV.W @ (disp : 8, PC), Rn							
1010	disp			BRA	disp : 12						
1011	disp			BSR	disp: 12						
1100	00MD	imm/disp		MOV.B R0, @(disp: 8, GBR)		MOV.W R0, @(disp: 8, GBR)		MOV.L R0, @(disp: 8, GBR)		TRAPA	#imm: 8
1100	01MD	disp		MOV.B @(disp: 8, GBR), R0		MOV.W @(disp: 8, GBR), R0		MOV.L @(disp: 8, GBR), R0		MOVA @(disp: 8, PC), R0	
1100	10MD	imm		TST	#imm: 8, R0	AND	#imm: 8, R0	XOR	#imm: 8, R0	OR	#imm: 8, R0
1100	11MD	imm		TST.B #imm: 8, @(R0, GBR)		AND.B #imm: 8, @(R0, GBR)		XOR.B #imm: 8, @(R0, GBR)		OR.B #imm: 8, @(R0, GBR)	
1101	Rn	disp		MOV.L @(disp: 8, PC), Rn							
1110	Rn	imm		MOV	#imm:8, Rn						
1111	00**	********		MOVX.W, MOVY.W	Double data transfer instruction						
1111	01**	********		MOVS.W, MOVS.L	Single data transfer instruction						
1111	10**	********		MOVX.W, MOVY.W	Double data transfer instruction, with DSP parallel operation instruction (32-bit instruction)						
1111	11**	********									

Notes: 1. For details, refer to the SH-3/SH-3E/SH3-DSP Software Manual.

2. Instructions in the hatched areas are DSP extended instructions. These instructions can be executed only when the DSP bit in the SR register is set to 1.

Section 4 Memory Management Unit (MMU)

This LSI has an on-chip memory management unit (MMU) that supports a virtual memory system. The on-chip translation look-aside buffer (TLB) caches information for user-created address translation tables located in external memory. It enables high-speed translation of virtual addresses into physical addresses. Address translation uses the paging system and supports two page sizes (1 kbyte or 4 kbytes). The access rights to virtual address space can be set for each of the privileged and user modes to provide memory protection.

4.1 Role of MMU

The MMU is a feature designed to make efficient use of physical memory. As shown in figure 4.1, if a process is smaller in size than the physical memory, the entire process can be mapped onto physical memory. However, if the process increases in size to the extent that it no longer fits into physical memory, it becomes necessary to partition the process and to map those parts requiring execution onto memory as occasion demands (figure 4.1 (1)). Having the process itself consider this mapping onto physical memory would impose a large burden on the process. To lighten this burden, the idea of virtual memory was born as a means of performing en bloc mapping onto physical memory (figure 4.1 (2)). In a virtual memory system, substantially more virtual memory than physical memory is provided, and the process is mapped onto this virtual memory. Thus a process only has to consider operation in virtual memory. Mapping from virtual memory to physical memory is handled by the MMU. The MMU is normally controlled by the operating system, switching physical memory to allow the virtual memory required by a process to be mapped onto physical memory in a smooth fashion. Switching of physical memory is performed via secondary storage, etc.

The virtual memory system that came into being in this way is particularly effective in a time-sharing system (TSS) in which a number of processes are running simultaneously (figure 4.1 (3)). If processes running in a TSS had to take mapping onto virtual memory into consideration while running, it would not be possible to increase efficiency. Virtual memory is thus used to reduce this load on the individual processes and so improve efficiency (figure 4.1 (4)). In the virtual memory system, virtual memory is allocated to each process. The task of the MMU is to perform efficient mapping of these virtual memory areas onto physical memory. It also has a memory protection feature that prevents one 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 occur that the relevant translation information is not recorded in the MMU, with the result that one process may inadvertently access the virtual memory allocated to another process. In this case, 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 also be implemented by software alone, the need for translation to be performed by software each time a process accesses physical memory would result in poor efficiency. For this reason, a buffer for address translation (translation look-aside buffer: TLB) is provided in hardware to hold frequently used address translation information. The TLB can be described as a cache for storing address translation information. Unlike cache memory, however, if address translation fails, that is, if an exception is generated, switching of address translation information is normally performed by software. This makes it possible for memory management to be performed flexibly by software.

The MMU has two methods of mapping from virtual memory to physical memory: a paging method using fixed-length address translation, and a segment method using variable-length address translation. With the paging method, the unit of translation is a fixed-size address space (usually of 1 to 64 kbytes) called a page.

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

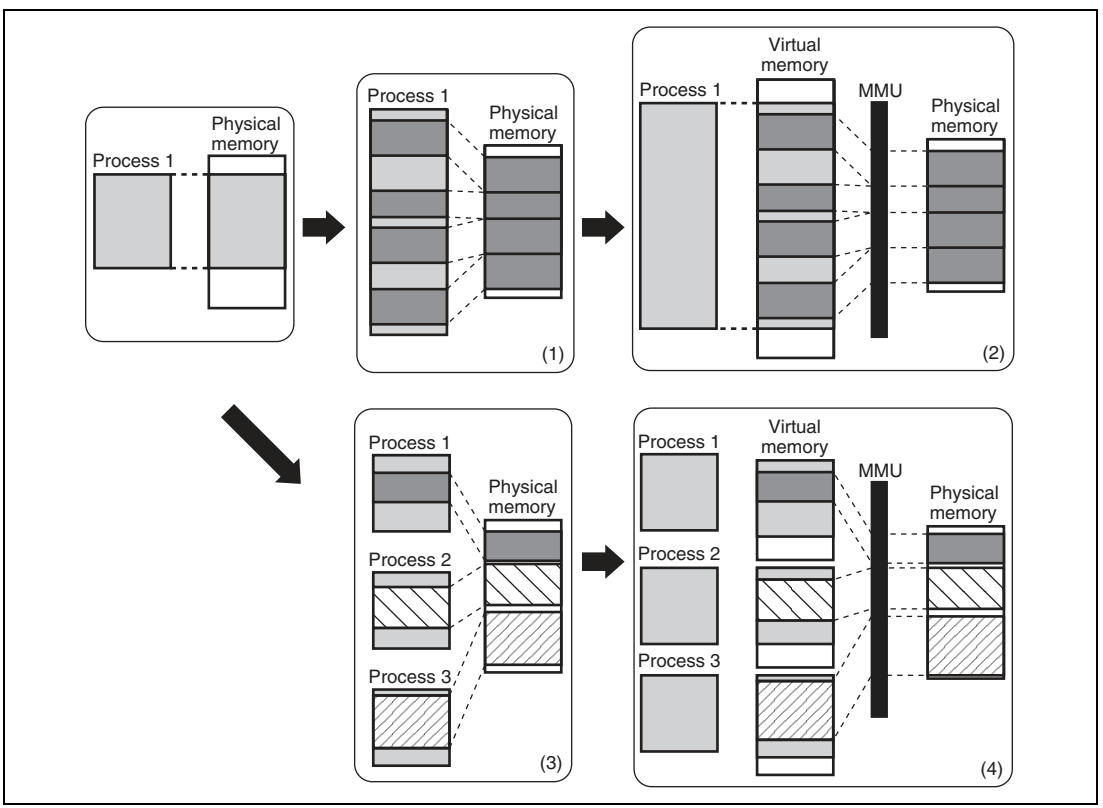


Figure 4.1 MMU Functions

4.1.1 MMU of This LSI

(1) Virtual Address Space

This LSI supports a 32-bit virtual address space that enables access to a 4-Gbyte address space. As shown in figures 4.2 and 4.3, the virtual address space is divided into several areas. In privileged mode, a 4-Gbyte space comprising areas P0 to P4 are accessible. In user mode, a 2-Gbyte space of U0 area is accessible, and a 16-Mbyte space of Uxy area is also accessible if the DSP bit in the SR register is set to 1. Access to any area (excluding the U0 area and Uxy area) in user mode will result in an address error.

If the MMU is enabled by setting the AT bit in the MMUCR register to 1, P0, P3, and U0 areas can be used as any physical address area in 1- or 4-kbyte page units. By using an 8-bit address space identifier, P0, P2, and U0 areas can be increased to up to 256 areas. Mapping from virtual address to 29-bit physical address can be achieved by the TLB.

(a) P0, P3, and U0 Areas

The P0, P3, and U0 areas can be address translated by the TLB and can be accessed through the cache. If the MMU is enabled, these areas can be mapped to any physical address space in 1- or 4-kbyte page units via the TLB. If the CE bit in the cache control register (CCR1) is set to 1 and if the corresponding cache enable bit (C bit) of the TLB entry is set to 1, access via the cache is enabled. If the MMU is disabled, replacing the upper three bits of an address in these areas with 0s creates the address in the corresponding physical address space. If the CE bit in the CCR1 register is set to 1, access via the cache is enabled. When the cache is used, either the copy-back or write-through mode is selected for write access via the WT bit in CCR1.

If these areas are mapped to the on-chip module control register area or on-chip memory area in area 1 in the physical address space via the TLB, the C bit of the corresponding page must be cleared to 0.

(b) P1 Area

The P1 area can be accessed via the cache and cannot be address-translated by the TLB. Whether the MMU is enabled or not, replacing the upper three bits of an address in these areas with 0s creates the address in the corresponding physical address space. Use of the cache is determined by the CE bit in the cache control register (CCR1). When the cache is used, either the copy-back or write-through mode is selected for write access by the CB bit in the CCR1 register.

(c) P2 Area

The P2 area cannot be accessed via the cache and cannot be address-translated by the TLB. Whether the MMU is enabled or not, replacing the upper three bits of an address in this area with 0s creates the address in the corresponding physical address space.

(d) P4 Area

The P4 area is mapped to the on-chip I/O of this LSI. This area cannot be accessed via the cache and cannot be address-translated by the TLB. Figure 4.4 shows the configuration of the P4 area.

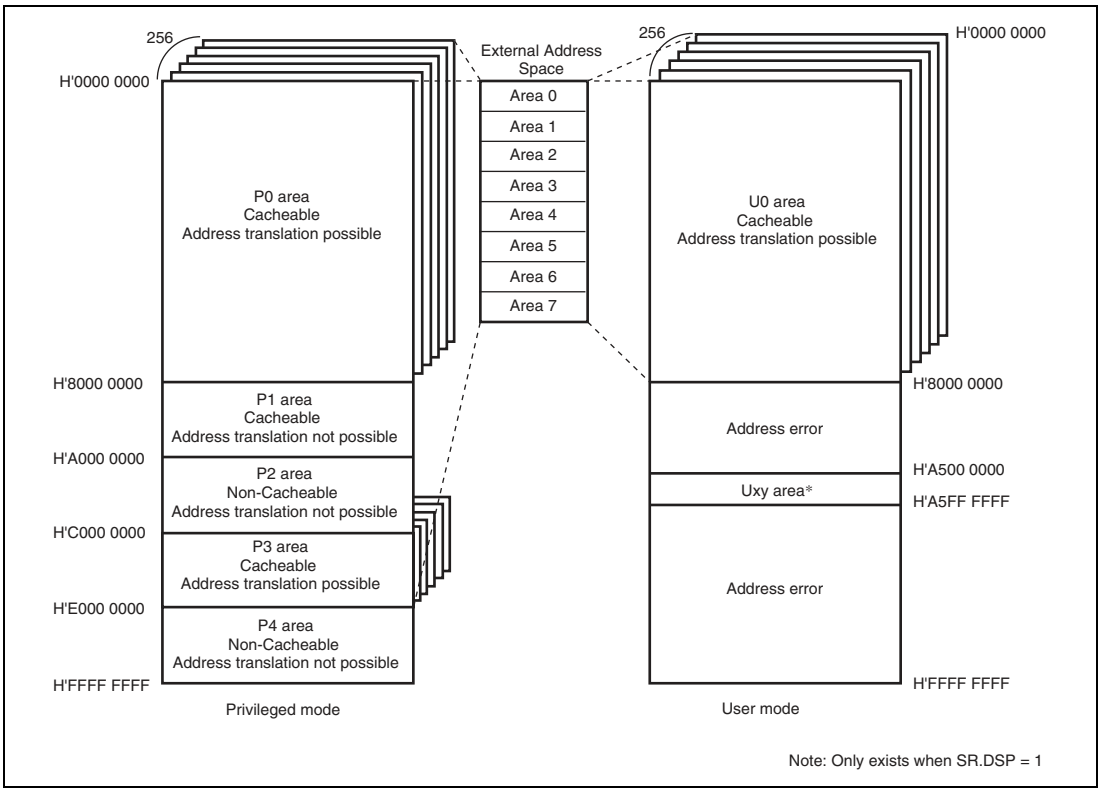


Figure 4.2 Virtual Address Space (MMUCR.AT = 1)

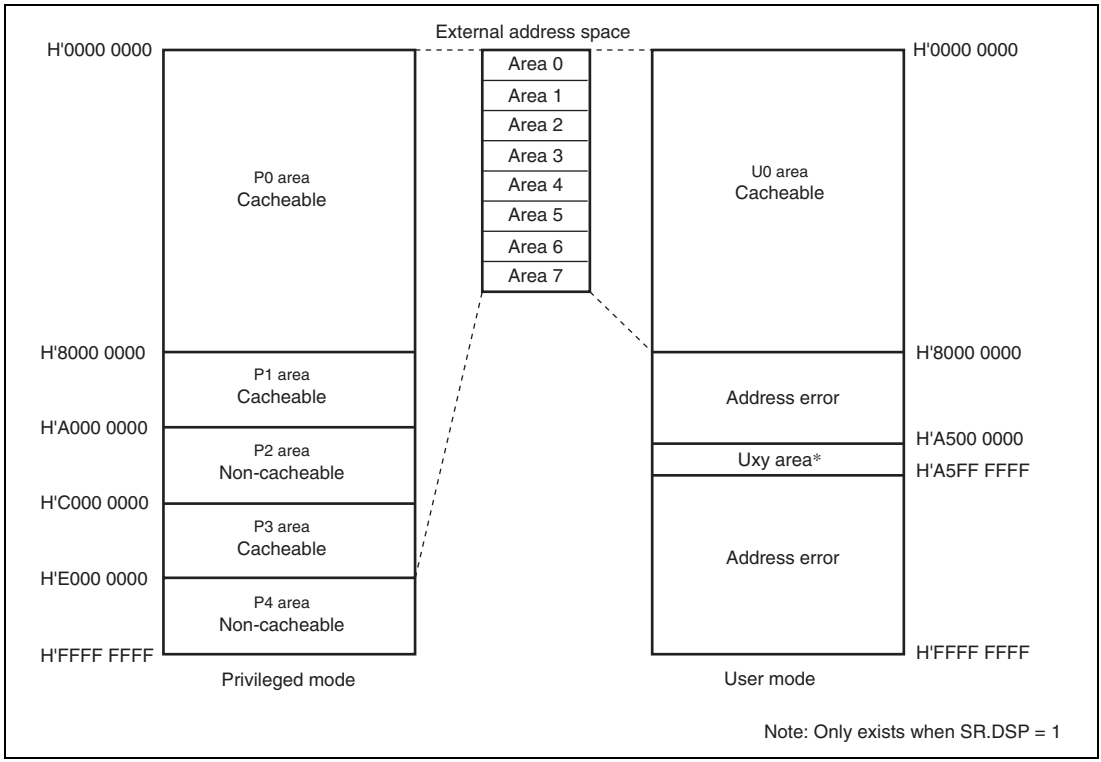


Figure 4.3 Virtual Address Space (MMUCR.AT = 0)

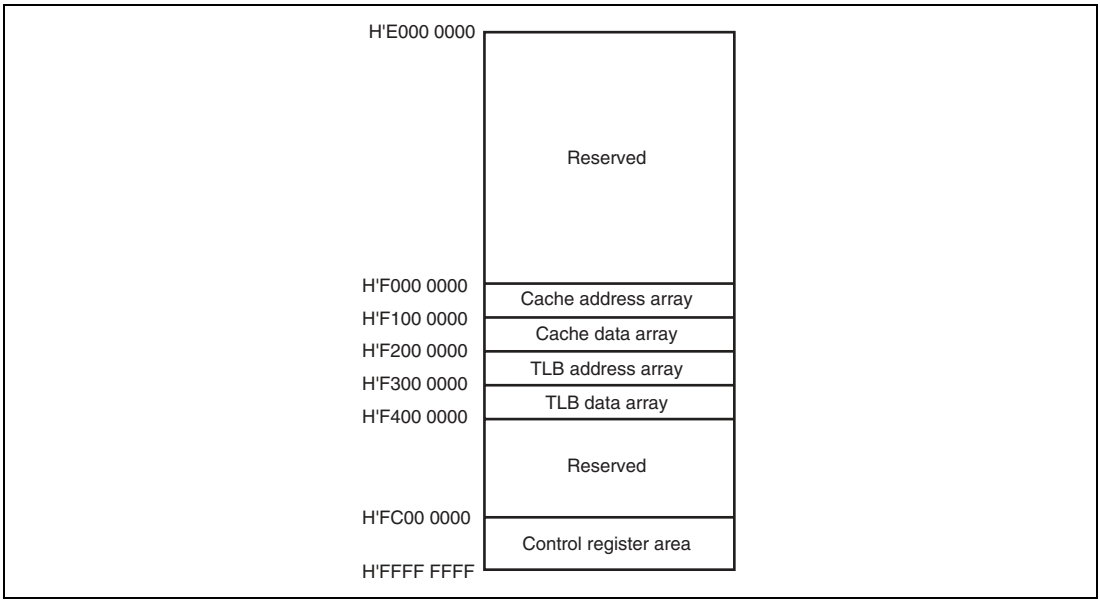


Figure 4.4 P4 Area

The area from H'F000 0000 to H'F0FF FFFF is for direct access to the cache address array. For more information, see section 5.4, Memory-Mapped Cache.

The area from H'F100 0000 to H'F1FF FFFF is for direct access to the cache data array. For more information, see section 5.4, Memory-Mapped Cache.

The area from H'F200 0000 to H'F2FF FFFF is for direct access to the TLB address array. For more information, see section 4.6, Memory-Mapped TLB.

The area from H'F300 0000 to H'F3FF FFFF is for direct access to the TLB data array. For more information, see section 4.6, Memory-Mapped TLB.

The area from H'FC00 0000 to H'FFFF FFFF is reserved for registers of the on-chip peripheral modules. For more information, see section 37, List of Registers.

(e) Uxy Area

The Uxy area is mapped to the on-chip memory of this LSI. This area is made usable in user mode when the DSP bit in the SR register is set to 1. In user mode, accessing this area when the DSP bit is 0 will result in an address error. This area cannot be accessed via the cache and cannot be address-translated by the TLB. For more information on the Uxy area, see section 6, X/Y Memory.

(2) Physical Address Space

This LSI supports a 29-bit physical address space. As shown in figure 4.5, the physical address space is divided into eight areas. Area 1 is mapped to the on-chip module control register area and on-chip memory area. Area 7 is reserved.

For details on physical address space, refer to section 9, Bus State Controller (BSC).

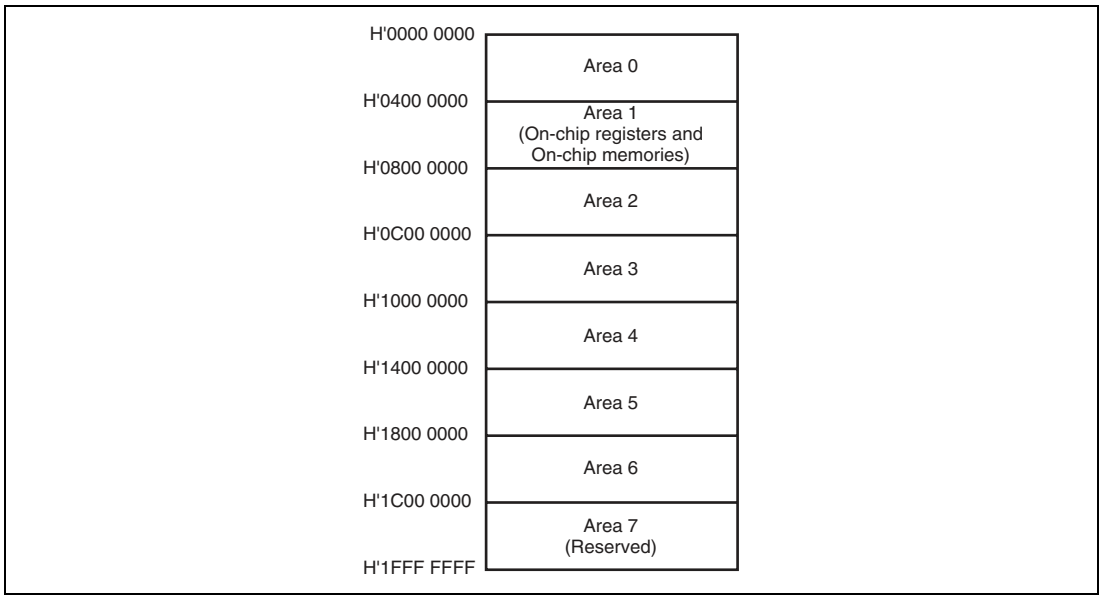


Figure 4.5 Physical Address Space

(3) Address Transition

When the MMU is enabled, the virtual address space is divided into units called pages. Physical addresses are translated in page units. Address translation tables in external memory hold information such as the physical address that corresponds to the virtual address and memory protection codes. When an access to area P1 or P2 occurs, there is no TLB access and the physical address is defined uniquely by hardware. If it belongs to area P0, P3 or U0, the TLB is searched by virtual address and, if that virtual address is registered in the TLB, the access hits the TLB. The corresponding physical address and the page control information are read from the TLB and the physical address is determined.

If the virtual address is not registered in the TLB, a TLB miss exception occurs and processing will shift to the TLB miss handler. In the TLB miss handler, the TLB address translation table in external memory is searched and the corresponding physical address and the page control

information are registered in the TLB. After returning from the handler, the instruction that caused the TLB miss is re-executed. When the MMU is enabled, address translation information that results in a physical address space of H'2000 0000 to H'FFFF FFFF should not be registered in the TLB.

When the MMU is disabled, masking the upper three bits of the virtual address to 0s creates the address in the corresponding physical address space. Since this LSI supports 29-bit address space as physical address space, the upper three bits of the virtual address are ignored as shadow areas. For details, refer to section 9, Bus State Controller (BSC). For example, address H'0000 1000 in the P0 area, address H'8000 1000 in the P1 area, address H'A000 1000 in the P2 area, and address H'C000 1000 in the P3 area are all mapped to the same physical memory. If these addresses are accessed while the cache is enabled, the upper three bits are always cleared to 0 to guarantee the continuity of addresses stored in the address array of the cache.

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

There are two virtual memory modes: single virtual memory mode and multiple virtual memory mode. In single virtual memory mode, multiple processes run in parallel using the virtual address space exclusively and the physical address corresponding to a given virtual address is specified uniquely. In multiple virtual memory mode, multiple processes run in parallel sharing the virtual address space, so a given virtual address may be translated into different physical addresses depending on the process. By the value set to the MMU control register (MMUCR), either single or multiple virtual mode is selected.

In terms of operation, the only difference between single virtual memory mode and multiple virtual memory mode is in the TLB address comparison method (see section 4.3.3, TLB Address Comparison).

(5) Address Space Identifier (ASID)

In multiple virtual memory mode, the address space identifier (ASID) is used to differentiate between processes running in parallel and sharing virtual address space. The ASID is eight bits in length and can be set by software setting of the ASID of the currently running process in page table entry register high (PTEH) within the MMU. When the process is switched using the ASID, the TLB does not have to be purged.

In single virtual memory mode, the ASID is used to provide memory protection for processes running simultaneously and using the virtual address space exclusively (see section 4.3.3, TLB Address Comparison).

4.2 Register Descriptions

There are four registers for MMU processing. These are all peripheral module registers, so they are located in address space area P4 and can only be accessed from privileged mode by specifying the address.

The MMU has the following registers. Refer to section 37, List of Registers, for more details on the addresses and access size of these registers.

- Page table entry register high (PTEH)
- Page table entry register low (PTEL)
- Translation table base register (TTB)
- MMU control register (MMUCR)

4.2.1 Page Table Entry Register High (PTEH)

The page table entry register high (PTEH) register residing at address H'FFFF FFF0, which consists of a virtual page number (VPN) and ASID. The VPN set is the VPN of the virtual address at which the exception is generated in case of an MMU exception or address error exception. When the page size is 4 kbytes, the VPN is the upper 20 bits of the virtual address, but in this case the upper 22 bits of the virtual address are set. The VPN can also be modified by software. As the ASID, software sets the number of the currently executing process. The VPN and ASID are recorded in the TLB by the LDTLB instruction.

A program that modifies the ASID in PTEH should be allocated in the P1 or P2 areas.

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	VPN	—	R/W	The Number of the Logical Page
9, 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	ASID	—	R/W	Address Space Identifier

4.2.2 Page Table Entry Register Low (PTEL)

The page table entry register low (PTEL) register residing at address H'FFFF FFF4, and used to store the physical page number and page management information to be recorded in the TLB by the LDTLB instruction. The contents of this register are only modified in response to a software command.

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
28 to 10	PPN	—	R/W	The Number of the Physical Page
9	—	0	R	Page Management Information For more details, see section 4.3, TLB Functions.
8	V	—	R/W	
7	—	0	R	
6, 5	PR	—	R/W	
4	SZ	—	R/W	
3	C	—	R/W	
2	D	—	R/W	
1	SH	—	R/W	
0	—	0	R	

4.2.3 Translation Table Base Register (TTB)

The translation table base register (TTB) residing at address H'FFFF FFF8, which points to the base address of the current page table. The hardware does not set any value in TTB automatically. TTB is available to software for general purposes. The initial value is undefined.

4.2.4 MMU Control Register (MMUCR)

The MMU control register (MMUCR) residing at address H'FFFF FFE0. Any program that modifies MMUCR should reside in the P1 or P2 area.

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SV	0	R/W	Single Virtual Memory Mode 0: Multiple virtual memory mode 1: Single virtual memory mode
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	RC	All 0	R/W	Random Counter A 2-bit random counter that is automatically updated by hardware according to the following rules in the event of an MMU exception. When a TLB miss exception occurs, all of TLB entry way corresponding to the virtual address at which the exception occurred are checked. If all ways are valid, 1 is added to RC; if there is one or more invalid way, they are set by priority from way 0, in the order way 0, way 1, way 2, way 3. In the event of an MMU exception other than a TLB miss exception, the way which caused the exception is set in RC.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	TF	0	R/W	TLB Flush Write 1 to flush the TLB (clear all valid bits of the TLB to 0). When they are read, 0 is always returned.
1	IX	0	R/W	Index Mode 0: VPN bits 16 to 12 are used as the TLB index number. 1: The value obtained by EX-ORing ASID bits 4 to 0 in PTEH and VPN bits 16 to 12 is used as the TLB index number.
0	AT	0	R/W	Address Translation Enables/disables the MMU. 0: MMU disabled 1: MMU enabled

4.3 TLB Functions

4.3.1 Configuration of the TLB

The TLB caches address translation table information located in the external memory. The address translation table stores the virtual page number and the corresponding physical number, the address space identifier, and the control information for the page, which is the unit of address translation. Figure 4.6 shows the overall TLB configuration. The TLB is 4-way set associative with 128 entries. There are 32 entries for each way. Figure 4.7 shows the configuration of virtual addresses and TLB entries.

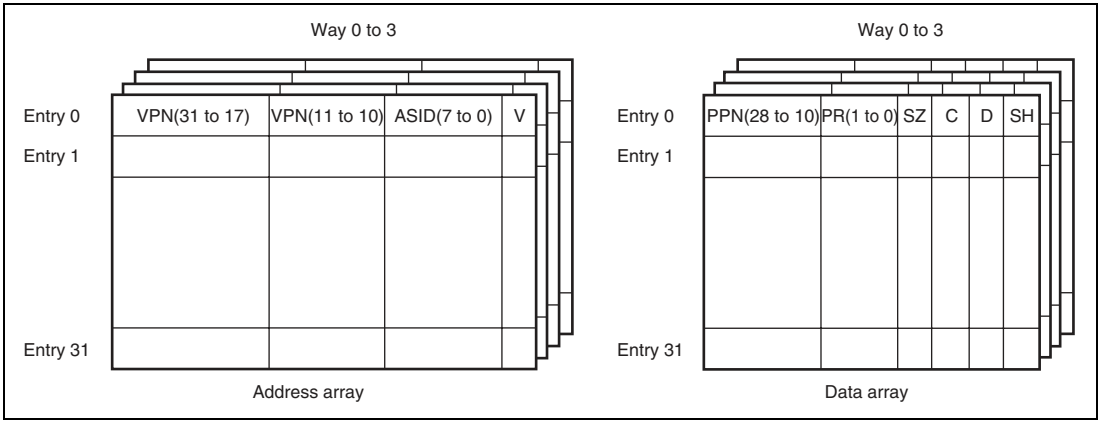
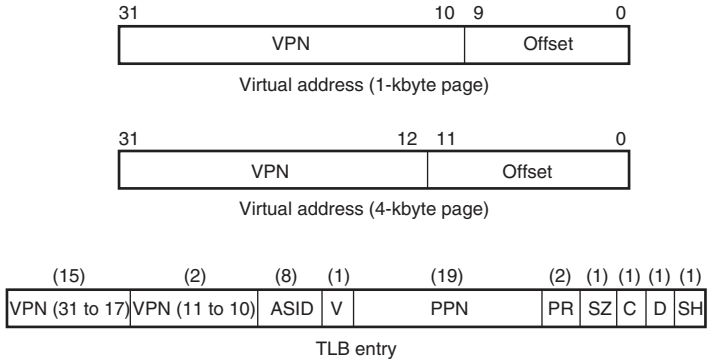


Figure 4.6 Overall Configuration of the TLB



[Legend]

VPN: Virtual page number
Upper 19 bits of virtual address for a 1-kbyte page, or upper 20 bits of logical address for a 4-kbyte page. Since VPN bits 16 to 12 are used as the index number, they are not stored in the TLB entry. Attention must be paid to the synonym problem (see section 4.4.4, Avoiding Synonym Problems).

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, the address is compared with the ASID in PTEH when address comparison is performed.

SH: Share status bit
0: Page not shared between processes
1: Page shared between processes

SZ: Page-size bit
0: 1-kbyte page
1: 4-kbyte page

V: Valid bit
Indicates whether 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 physical address. PPN bits 11 to10 are not used in case of a 4-kbyte page.

PR: Protection key field
2-bit field encoded to define the access rights to the page.
00: Reading only is possible in privileged mode.
01: Reading/writing is possible in privileged mode.
10: Reading only is possible in privileged/user mode.
11: Reading/writing is possible in privileged/user mode.

C: Cacheable bit
Indicates whether the page is cacheable.
0: Non-cacheable
1: Cacheable

D: Dirty bit
Indicates whether the page has been written to.
0: Not written to
1: Written to

Figure 4.7 Virtual address and TLB Structure

4.3.2 TLB Indexing

The TLB uses a 4-way set associative scheme, so entries must be selected by index. VPN bits 16 to 12 and ASID bits 4 to 0 in PTEH are used as the index number regardless of the page size. The index number can be generated in two different ways depending on the setting of the IX bit in MMUCR.

1. When IX = 1, VPN bits 16 to 12 are EX-ORed with ASID bits 4 to 0 to generate a 5-bit index number
2. When IX = 0, VPN bits 16 to 12 alone are used as the index number

The first method is used to prevent lowered TLB efficiency that results when multiple processes run simultaneously in the same virtual address space (multiple virtual memory) and a specific entry is selected by indexing of each process. In single virtual memory mode (MMUCR.SV = 1), IX bit should be set to 0. Figures 4.8 and 4.9 show the indexing schemes.

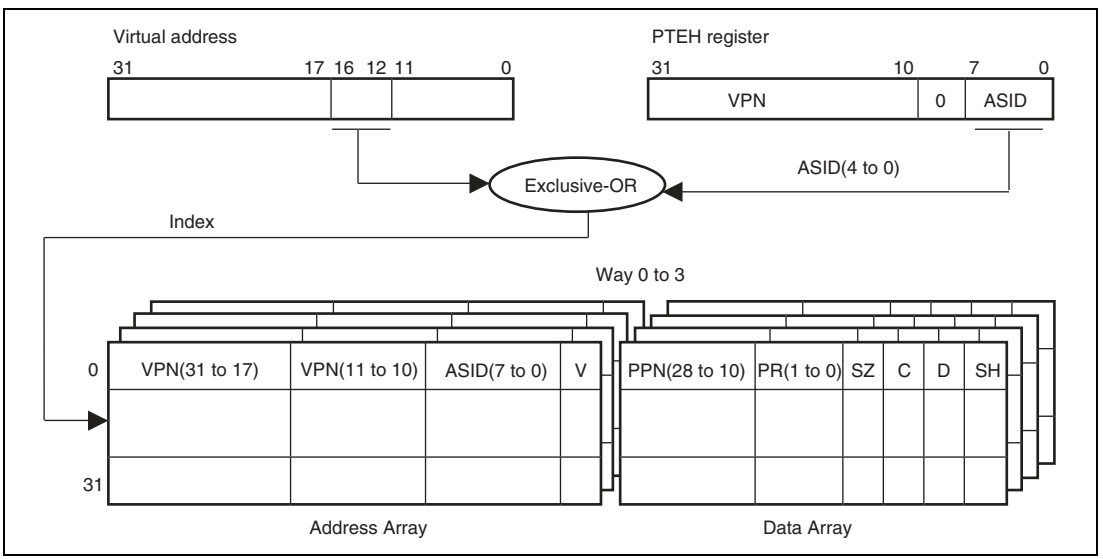


Figure 4.8 TLB Indexing (IX = 1)

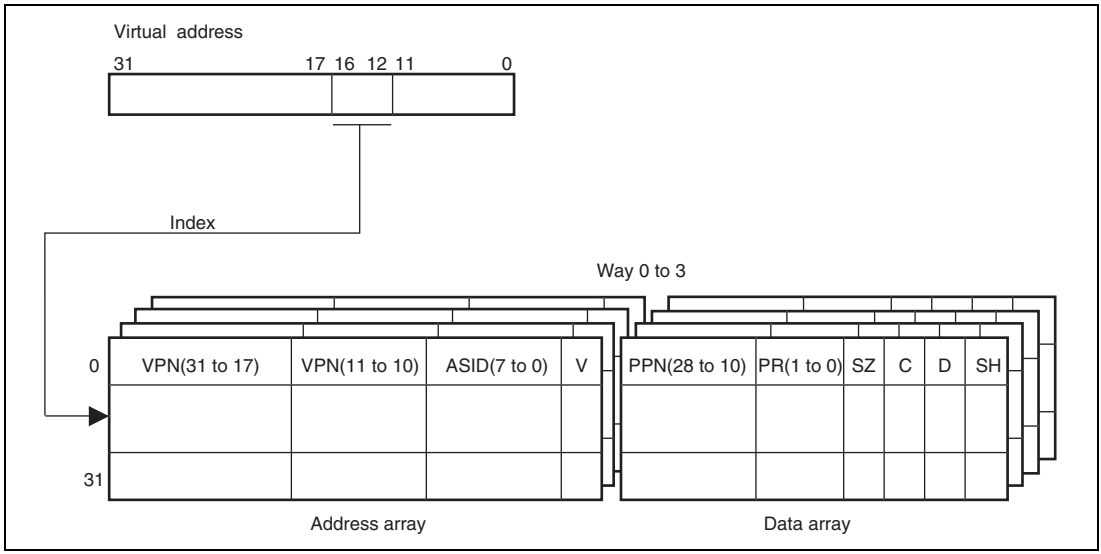


Figure 4.9 TLB Indexing (IX = 0)

4.3.3 TLB Address Comparison

The results of address comparison determine whether a specific virtual page number is registered in the TLB. The virtual page number of the virtual address that accesses external memory is compared to the virtual page number of the indexed TLB entry. The ASID within the PTEH is compared to the ASID of the indexed TLB entry. All four ways are searched simultaneously. If the compared values match, and the indexed TLB entry is valid (V bit = 1), the hit is registered.

It is necessary to have software ensure that TLB hits do not occur simultaneously in more than one way, as hardware operation is not guaranteed if this occurs. An example of setting which causes TLB hits to occur simultaneously in more than one way is described below. It is necessary to ensure that this kind of setting is not made by software.

1. If there are two identical TLB entries with the same VPN and a setting is made such that a TLB hit is made only by a process with ASID = H'FF when one is in the shared state (SH = 1) and the other in the non-shared state (SH = 0), then if the ASID in PTEH is set to H'FF, there is a possibility of simultaneous TLB hits in both these ways.
2. If several entries which have different ASID with the same VPN are registered in single virtual memory mode, there is the possibility of simultaneous TLB hits in more than one way when accessing the corresponding page in privileged mode. Several entries with the same VPN must not be registered in single virtual memory mode.

3. There is the possibility of simultaneous TLB hits in more than one way. These hits may occur depending on the contents of ASID in PTEH when a page to which SH is set 1 is registered in the TLB in index mode (MMUCR.IX = 1). Therefore a page to which SH is set 1 must not be registered in index mode. When memory is shared by several processings, different pages must be registered in each ASID.

The object compared varies depending on the page management information (SZ, SH) in the TLB entry. It also varies depending on whether the system supports multiple virtual memory or single virtual memory.

The page-size information determines whether VPN (11 to 10) is compared. VPN (11 to 10) is compared for 1-kbyte pages (SZ = 0) but not for 4-kbyte pages (SZ = 1).

The sharing information (SH) determines whether the PTEH.ASID and the ASID in the TLB entry are compared. ASIDs are compared when there is no sharing between processes (SH = 0) but not when there is sharing (SH = 1).

When single virtual memory is supported (MMUCR.SV = 1) and privileged mode is engaged (SR.MD = 1), all process resources can be accessed. This means that ASIDs are not compared when single virtual memory is supported and privileged mode is engaged. The objects of address comparison are shown in figure 4.10.

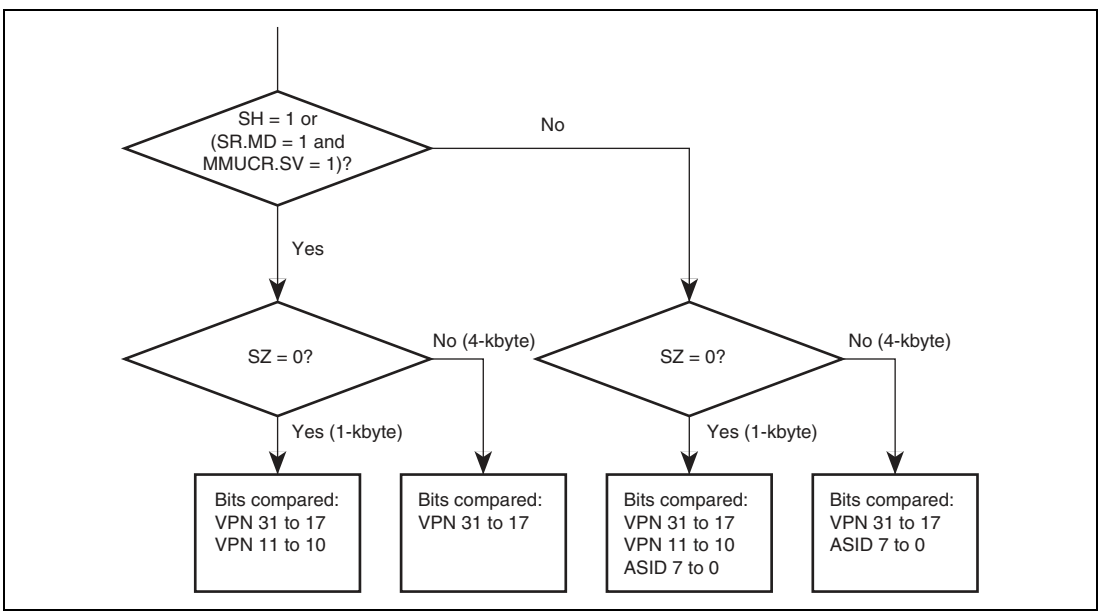


Figure 4.10 Objects of Address Comparison

4.3.4 Page Management Information

In addition to the SH and SZ bits, the page management information of TLB entries also includes D, C, and PR bits.

The D bit of a TLB entry indicates whether the page is dirty (i.e., has been written to). If the D bit is 0, an attempt to write to the page results in an initial page write exception. For physical page swapping between secondary memory and main memory, for example, pages are controlled so that a dirty page is paged out of main memory only after that page is written back to secondary memory. To record that there has been a write to a given page in the address translation table in memory, an initial page write exception is used.

The C bit in the entry indicates whether the referenced page resides in a cacheable or non-cacheable area of memory. When the control registers and on-chip memory in area 1 are mapped, set the C bit to 0. The PR field specifies the access rights for the page in privileged and user modes and is used to protect memory. Attempts at non-permitted accesses result in TLB protection violation exceptions.

Access states designated by the D, C, and PR bits are shown in table 4.1.

Table 4.1 Access States Designated by D, C, and PR Bits

		Privileged Mode		User Mode	
		Reading	**Writing**	**Reading**	**Writing**
D bit	0	Permitted	Initial page write exception	Permitted	Initial page write exception
	1	Permitted	Permitted	Permitted	Permitted
C bit	0	Permitted (no caching)	Permitted (no caching)	Permitted (no caching)	Permitted (no caching)
	1	Permitted (with caching)	Permitted (with caching)	Permitted (with caching)	Permitted (with caching)
PR bit	00	Permitted	TLB protection violation exception	TLB protection violation exception	TLB protection violation exception
	01	Permitted	Permitted	TLB protection violation exception	TLB protection violation exception
	10	Permitted	TLB protection violation exception	Permitted	TLB protection violation exception
	11	Permitted	Permitted	Permitted	Permitted

4.4 MMU Functions

4.4.1 MMU Hardware Management

There are two kinds of MMU hardware management as follows.

1. The MMU decodes the virtual address accessed by a process and performs address translation by controlling the TLB in accordance with the MMUCR settings.
2. In address translation, the MMU receives page management information from the TLB, and determines the MMU exception and whether the cache is to be accessed (using the C bit). For details of the determination method and the hardware processing, see section 4.5, MMU Exceptions.

4.4.2 MMU Software Management

There are three kinds of MMU software management, as follows.

1. MMU register setting

   MMUCR setting, in particular, should be performed in areas P1 and P2 for which address translation is not performed. Also, since SV and IX bit changes constitute address translation system changes, in this case, TLB flushing should be performed by simultaneously writing 1 to the TF bit also. Since MMU exceptions are not generated in the MMU disabled state with the AT bit cleared to 0, use in the disabled state must be avoided with software that does not use the MMU.
2. TLB entry recording, deletion, and reading

   TLB entry recording can be done in two ways by using the LDTLB instruction, or by writing directly to the memory-mapped TLB. For TLB entry deletion and reading, the memory allocation TLB can be accessed. See section 4.4.3, MMU Instruction (LDTLB), for details of the LDTLB instruction, and section 4.6, Memory-Mapped TLB, for details of the memory-mapped TLB.
3. MMU exception processing

   When an MMU exception is generated, it is handled on the basis of information set from the hardware side. See section 4.5, MMU Exceptions, for details.

When single virtual memory mode is used, it is possible to create a state in which physical memory access is enabled in the privileged mode only by clearing the share status bit (SH) to 0 to specify recording of all TLB entries. This strengthens inter-process memory protection, and enables special access levels to be created in the privileged mode only.

Recording a 1- or 4- kbyte page TLB entry may result in a synonym problem. See section 4.4.4, Avoiding Synonym Problems.

4.4.3 MMU Instruction (LDTLB)

The load TLB instruction (LDTLB) is used to record TLB entries. When the IX bit in MMUCR is 0, the LDTLB instruction changes the TLB entry in the way specified by the RC bit in MMUCR to the value specified by PTEH and PTEL, using VPN bits 16 to 12 specified in PTEH as the index number. When the IX bit in MMUCR is 1, the EX-OR of VPN bits 16 to 12 specified in PTEH and ASID bits 4 to 0 in PTEH are used as the index number.

Figure 4.11 shows the case where the IX bit in MMUCR is 0.

When an MMU exception occurs, the virtual page number of the virtual address that caused the exception is set in PTEH by hardware. The way is set in the RC bit in MMUCR for each exception according to the rules (see section 4.2.4, MMU Control Register (MMUCR)). Consequently, if the LDTLB instruction is issued after setting only PTEL in the MMU exception processing routine, TLB entry recording is possible. Any TLB entry can be updated by software rewriting of PTEH and the RC bits in MMUCR.

As the LDTLB instruction changes address translation information, there is a risk of destroying address translation information if this instruction is issued in the P0, U0, or P3 area. Make sure, therefore, that this instruction is issued in the P1 or P2 area. Also, an instruction associated with an access to the P0, U0, or P3 area (such as the RTE instruction) should be issued at least two instructions after the LDTLB instruction.

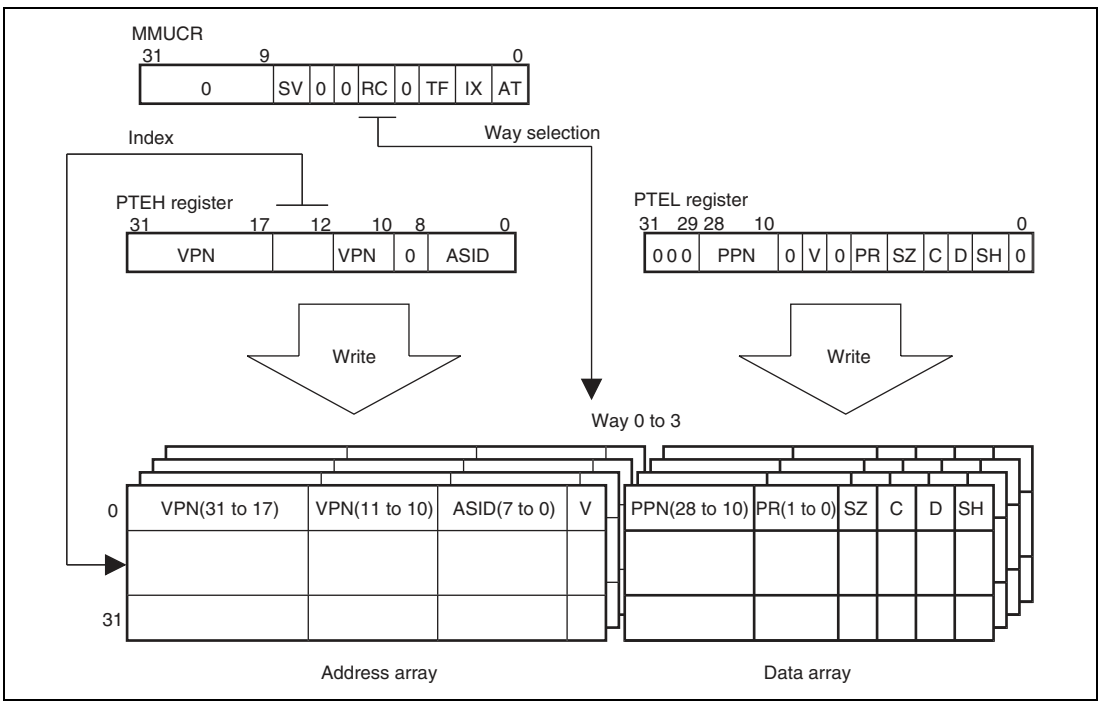


Figure 4.11 Operation of LDTLB Instruction

4.4.4 Avoiding Synonym Problems

When a 1- or 4-kbyte page is recorded in a TLB entry, a synonym problem may arise. If a number of virtual addresses are mapped onto a single physical address, the same physical address data will be recorded in a number of cache entries, and it will not be possible to guarantee data congruity. The reason that this problem occurs is explained below with reference to figure 4.12.

The relationship between bit n of the virtual address and cache size is shown in the following table. Note that no synonym problems occur in 4-kbyte page when the cache size is 16 kbytes.

Cache Size	Bit n in Virtual Address
16 kbytes	11
32 kbytes	12

To achieve high-speed operation of this LSI's cache, an index number is created using virtual address bits 12 to 4. When a 1-kbyte page is used, virtual address bits 12 to 10 is subject to address translation and when a 4-kbyte page is used, a virtual address bit 12 is subject to address translation. Therefore, the physical address bits 12 to 10 may not be the same as the virtual address bits 12 to 10.

For example, assume that, with 1-kbyte page TLB entries, TLB entries for which the following translation has been performed are recorded in two TLBs:

Virtual address 1 H'0000 0000 → physical address H'0000 0C00

Virtual address 2 H'000 00C00 → physical address H'0000 0C00

Virtual address 1 is recorded in cache entry H'000, and virtual address 2 in cache entry H'0C0. Since two virtual addresses are recorded in different cache entries despite the fact that the physical addresses are the same, memory inconsistency will occur as soon as a write is performed to either virtual address.

Consequently, the following restrictions apply to the recording of address translation information in TLB entries.

1. When address translation information whereby a number of 1-kbyte page TLB entries are translated into the same physical address is recorded in the TLB, ensure that the VPN bits 12 is the same.
2. When address translation information whereby a number of 4-kbyte page TLB entries are translated into the same physical address is recorded in the TLB, ensure that the VPN bit 12 is the same.

3. Do not use the same physical addresses for address translation information of different page sizes.

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

Note: When multiple items of address translation information use the same physical memory to provide for future SuperH RISC engine family expansion, ensure that the VPN bits 20 to 10 are the same.

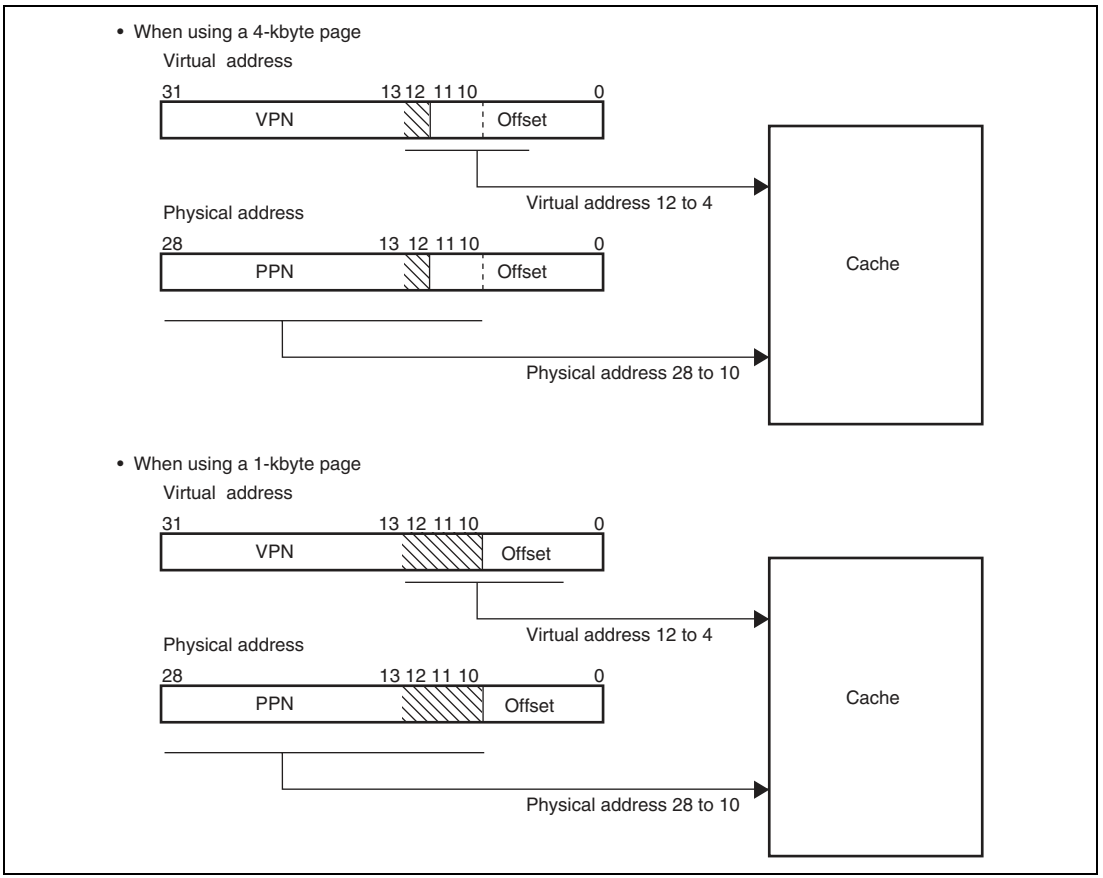


Figure 4.12 Synonym Problem (32-kbyte Cache)

4.5 MMU Exceptions

When the address translation unit of the MMU is enabled, occurrence of the MMU exception is checked following the CPU address error check. There are four MMU exceptions: TLB miss, TLB invalid, TLB protection violation, and initial page write, and these MMU exceptions are checked in this order.

4.5.1 TLB Miss Exception

A TLB miss results when the virtual address and the address array of the selected TLB entry are compared and no match is found. TLB miss exception processing includes both hardware and software operations.

- Hardware Operations

  In a TLB miss, this hardware executes a set of prescribed operations, as follows:

  A. The VPN field of the virtual address causing the exception is written to the PTEH register.
  B. The virtual address causing the exception is written to the TEA register.
  C. Either exception code H'040 for a load access, or H'060 for a store access, is written to the EXPEVT register.
  D. The PC value indicating the address of the instruction in which the exception occurred is written to the save program counter (SPC). If the exception occurred in a delay slot, the PC value indicating the address of the related delayed branch instruction is written to the SPC.
  E The contents of the status register (SR) at the time of the exception are written to the save status register (SSR).
  F. The mode (MD) bit in SR is set to 1 to place the privileged mode.
  G. The block (BL) bit in SR is set to 1 to mask any further exception requests.
  H. The register bank (RB) bit in SR is set to 1.
  I. The RC field in the MMU control register (MMUCR) is incremented by 1 when all entries indexed are valid. When some entries indexed are invalid, the smallest way number of them is set in RC. The setting priority is way0, way1, way2, and way3.
  J. Execution branches to the address obtained by adding the value of the VBR contents and H'0000 0400 to invoke the user-written TLB miss exception handler.

- Software (TLB Miss Handler) Operations

  The software searches the page tables in external memory and allocates the required page table entry. Upon retrieving the required page table entry, software must execute the following operations:

  A. Write the value of the physical page number (PPN) field and the protection key (PR), page size (SZ), cacheable (C), dirty (D), share status (SH), and valid (V) bits of the page table entry recorded in the address translation table in the external memory into the PTEL register.
  B. If using software for way selection for entry replacement, write the desired value to the RC field in MMUCR.
  C. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB.
  D. Issue the return from exception handler (RTE) instruction to terminate the handler routine and return to the instruction stream. Issue the RTE instruction after issuing two instructions from the LDTLB instruction.

4.5.2 TLB Protection Violation Exception

A TLB protection violation exception results when the virtual address and the address array of the selected TLB entry are compared and a valid entry is found to match, but the type of access is not permitted by the access rights specified in the PR field. TLB protection violation exception processing includes both hardware and software operations.

- Hardware Operations

  In a TLB protection violation exception, this hardware executes a set of prescribed operations, as follows:

  A. The VPN field of the virtual address causing the exception is written to the PTEH register.
  B. The virtual address causing the exception is written to the TEA register.
  C. Either exception code H'0A0 for a load access, or H'0C0 for a store access, is written to the EXPEVT register.
  D. The PC value indicating the address of the instruction in which the exception occurred is written into SPC (if the exception occurred in a delay slot, the PC value indicating the address of the related delayed branch instruction is written into SPC).
  E. The contents of SR at the time of the exception are written to SSR.
  F. The MD bit in SR is set to 1 to place the privileged mode.
  G. The BL bit in SR is set to 1 to mask any further exception requests.
  H. The RB bit in SR is set to 1.
  I. The way that generated the exception is set in the RC field in MMUCR.

J. Execution branches to the address obtained by adding the value of the VBR contents and H'0000 0100 to invoke the TLB protection violation exception handler.

- Software (TLB Protection Violation Handler) Operations

  Software resolves the TLB protection violation and issues the RTE (return from exception handler) instruction to terminate the handler and return to the instruction stream. Issue the RTE instruction after issuing two instructions from the LDTLB instruction.

4.5.3 TLB Invalid Exception

A TLB invalid exception results when the virtual address is compared to a selected TLB entry address array and a match is found but the entry is not valid (the V bit is 0). TLB invalid exception processing includes both hardware and software operations.

- Hardware Operations

  In a TLB invalid exception, this hardware executes a set of prescribed operations, as follows:

  A. The VPN field of the virtual address causing the exception is written to the PTEH register.
  B. The virtual address causing the exception is written to the TEA register.
  C. Either exception code H'040 for a load access, or H'060 for a store access, is written to the EXPEVT register.
  D. The PC value indicating the address of the instruction in which the exception occurred is written to the SPC. If the exception occurred in a delay slot, the PC value indicating the address of the delayed branch instruction is written to the SPC.
  E. The contents of SR at the time of the exception are written into SSR.
  F. The mode (MD) bit in SR is set to 1 to place the privileged mode.
  G. The block (BL) bit in SR is set to 1 to mask any further exception requests.
  H. The RB bit in SR is set to 1.
  I. The way number causing the exception is written to RC in MMUCR.
  J. Execution branches to the address obtained by adding the value of the VBR contents and H'0000 0100, and the TLB protection violation exception handler starts.

- Software (TLB Invalid Exception Handler) Operations

  The software searches the page tables in external memory and assigns the required page table entry. Upon retrieving the required page table entry, software must execute the following operations:

  A. Write the values of the physical page number (PPN) field and the values of the protection key (PR), page size (SZ), cacheable (C), dirty (D), share status (SH), and valid (V) bits of the page table entry recorded in the external memory to the PTEL register.
  B. If using software for way selection for entry replacement, write the desired value to the RC field in MMUCR.
  C. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB.
  D. Issue the RTE instruction to terminate the handler and return to the instruction stream. The RTE instruction should be issued after two instructions form the LDTLB instruction.

4.5.4 Initial Page Write Exception

An initial page write exception results in a write access when the virtual address and the address array of the selected TLB entry are compared and a valid entry with the appropriate access rights is found to match, but the D (dirty) bit of the entry is 0 (the page has not been written to). Initial page write exception processing includes both hardware and software operations.

- Hardware Operations

  In an initial page write exception, this hardware executes a set of prescribed operations, as follows:

  A. The VPN field of the virtual address causing the exception is written to the PTEH register.
  B. The virtual address causing the exception is written to the TEA register.
  C. Exception code H'080 is written to the EXPEVT register.
  D. The PC value indicating the address of the instruction in which the exception occurred is written to the SPC. If the exception occurred in a delay slot, the PC value indicating the address of the related delayed branch instruction is written to the SPC.
  E. The contents of SR at the time of the exception are written to SSR.
  F. The MD bit in SR is set to 1 to place the privileged mode.
  G. The BL bit in SR is set to 1 to mask any further exception requests.
  H. The RB bit in SR is set to 1.
  I. The way that caused the exception is set in the RC field in MMUCR.
  J. Execution branches to the address obtained by adding the value of the VBR contents and H'0000 0100 to invoke the user-written initial page write exception handler.

- Software (Initial Page Write Handler) Operations

  The software must execute the following operations:

  A. Retrieve the required page table entry from external memory.
  B. Set the D bit of the page table entry in the external memory to 1.
  C. Write the value of the PPN field and the PR, SZ, C, D, SH, and V bits of the page table entry in the external memory to the PTEL register.
  D. If using software for way selection for entry replacement, write the desired value to the RC field in MMUCR.
  E. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB.
  F. Issue the RTE instruction to terminate the handler and return to the instruction stream. The RTE instruction must be issued after two LDTLB instructions.

4.5.5 MMU Exception in Repeat Loop

If a CPU address error or MMU exception occurs in a specific instruction in the repeat loop, the SPC may indicate an illegal address or the repeat loop cannot be reexecuted correctly even if the SPC is correct. Accordingly, if a CPU address error or MMU exception occurs in a specific instruction in the repeat loop, this LSI generates a specific exception code to set the EXPEVT to H′070 for a TLB miss exception, TLB invalid exception, initial page write exception, and CPU address error and to H'0D0 for a TLB protection violation exception. In addition, a vector offset for TLB miss exception is H'100. For details, refer to section 7.4.3, Exception in Repeat Control Period.

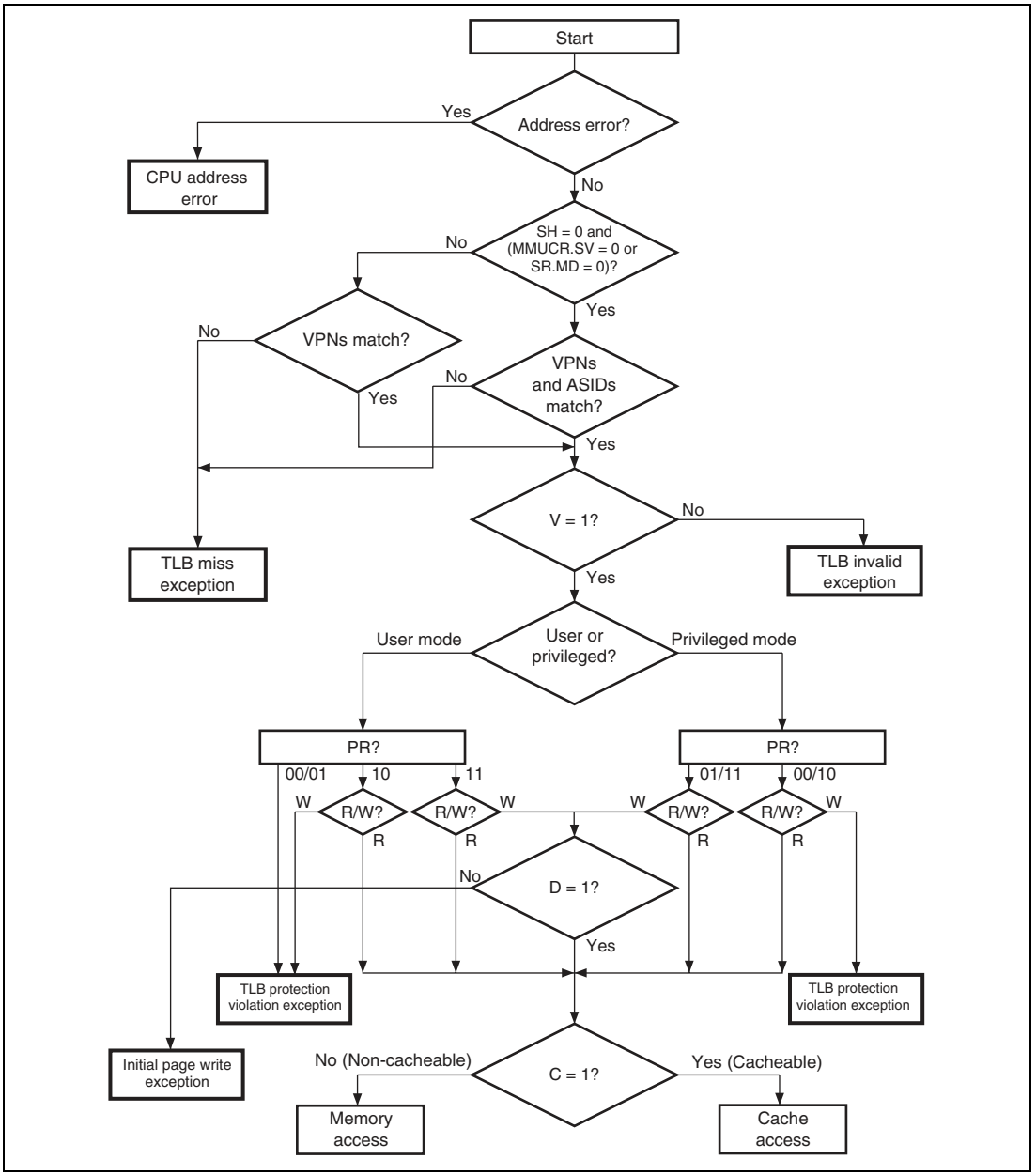


Figure 4.13 MMU Exception Generation Flowchart

4.6 Memory-Mapped TLB

In order for TLB operations to be managed by software, TLB contents can be read or written to in the privileged mode using the MOV instruction. The TLB is assigned to the P4 area in the virtual address space. The TLB address array (VPN, V bit, and ASID) is assigned to H'F200 0000 to H'F2FF FFFF, and the data array (PPN, PR, SZ, C, D, and SH bits) to H'F300 0000 to H'F3FF FFFF. The V bit in the address array can also be accessed from the data array. Only longword access is possible for both the address array and the data array. However, the instruction data cannot be fetched from both arrays.

4.6.1 Address Array

The address array is assigned to H'F200 0000 to H'F2FF FFFF. To access an address array, the 32-bit address field (for read/write operations) and 32-bit data field (for write operations) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the VPN, V bit and ASID to be written to the address array (figure 4.14 (1)).

In the address field, specify the entry address for selecting the entry (bits 16 to 12), W for selecting the way (bits 9 to 8) and H′F2 to indicate address array access (bits 31 to 24). The IX bit in MMUCR indicates whether an EX-OR is taken of the entry address and ASID.

The following two operations can be used on the address array:

1. Address array read

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

   The data specified in the data field are written to the TLB entry corresponding to the entry address and way set in the address field.

4.6.2 Data Array

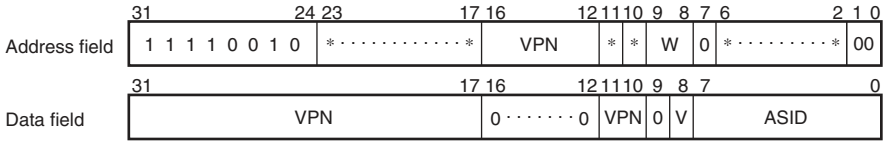
The data array is assigned to H'F300 0000 to H'F3FF FFFF. To access a data array, the 32-bit address field (for read/write operations), and 32-bit data field (for write operations) must be specified. The address section specifies information for selecting the entry to be accessed; the data section specifies the longword data to be written to the data array (figure 4.14 (2)).

In the address section, specify the entry address for selecting the entry (bits 16 to 12), W for selecting the way (bits 9 to 8), and H'F3 to indicate data array access (bits 31 to 24). The IX bit in MMUCR indicates whether an EX-OR is taken of the entry address and ASID.

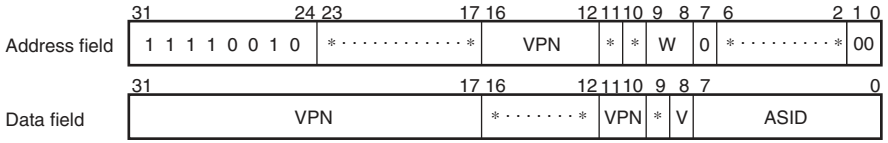
Both reading and writing use the longword of the data array specified by the entry address and way number. The access size of the data array is fixed at longword.

(1) TLB Address Array Access

• Read Access



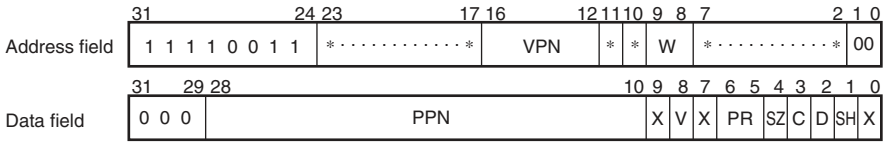
• Write Access



VPN: Virtual page number
V: Valid bit
W: Way (00: Way 0, 01: Way 1, 10: Way 2, 11: Way 3)
ASID: Address space identifier
*: Don't care bit

(2) TLB Data Array Access

• Read/Write Access



PPN: Physical page number
PR: Protection key field
C: Cacheable bit
SH: Share status bit
VPN: Virtual page number
X: 0 for read, don't care bit for write
W: Way (00: Way 0, 01: Way 1, 10: Way 2, 11: Way 3)
V: Valid bit
SZ: Page-size bit
D: Dirty bit
*: Don't care bit

Figure 4.14 Specifying Address and Data for Memory-Mapped TLB Access

4.6.3 Usage Examples

(1) Invalidating Specific Entries

Specific TLB entries can be invalidated by writing 0 to the entry's V bit. R0 specifies the write data and R1 specifies the address.

```
; R0=H'1547 381C  R1=H'F201 3000
; MMUCR.IX=0
; the V bit of way 0 of the entry selected by the VPN(16-12)=B'1 0011
; index is cleared to 0,achieving invalidation.
MOV.L  R0,@R1
```

(2) Reading the Data of a Specific Entry

This example reads the data section of a specific TLB entry. The bit order indicated in the data field in figure 4.17 (2) is read. R0 specifies the address and the data section of a selected entry is read to R1.

```
; R0=H'F300 4300   VPN(16-12)=B'00100   Way 3
; MOV.L @R0,R1
```

4.7 Usage Note

The following operations should be performed in the P1 or P2 area. In addition, when the P0, P3, or U0 area is accessed consecutively (this access includes instruction fetching), the instruction code should be placed at least two instructions after the instruction that executes the following operations.

1. Modification of SR.MD or SR.BL
2. Execution of the LDTLB instruction
3. Write to the memory-mapped TLB
4. Modification of MMUCR
5. Modification of PTEH.ASID

Section 5 Cache

5.1 Features

- Capacity: 16 or 32 kbytes
- Structure: Instructions/data mixed, 4-way set associative
- Locking: Way 2 and way 3 are lockable
- Line size: 16 bytes
- Number of entries: 256 entries/way in 16-kbyte mode to 512 entries/way in 32-kbyte mode
- Write system: Write-back/write-through is selectable for spaces P0, P1, P3, and U0
  Group 1 (P0, P3, and U0 areas)
  Group 2 (P1 area)
- Replacement method: Least-recently used (LRU) algorithm

Note: After power-on reset or manual reset, initialized as 16-kbyte mode (256 entries/way).

5.1.1 Cache Structure

The cache mixes instructions and data and uses a 4-way set associative system. It is composed of four ways (banks), and each of which is divided into an address section and a data section. Note that the following sections will be described for the 16-kbyte mode as an example. For other cache size modes, change the number of entries and size/way according to table 5.1. Each of the address and data sections is divided into 256 entries. The entry data is called a line. Each line consists of 16 bytes (4 bytes × 4). The data capacity per way is 4 kbytes (16 bytes × 256 entries) in the cache as a whole (4 ways). The cache capacity is 16 kbytes as a whole.

Table 5.1 Number of Entries and Size/Way in Each Cache Size

Cache Size	Number of Entries	Size/Way
16 kbytes	256	4 kbytes
32 kbytes	512	8 kbytes

Figure 5.1 shows the cache structure.

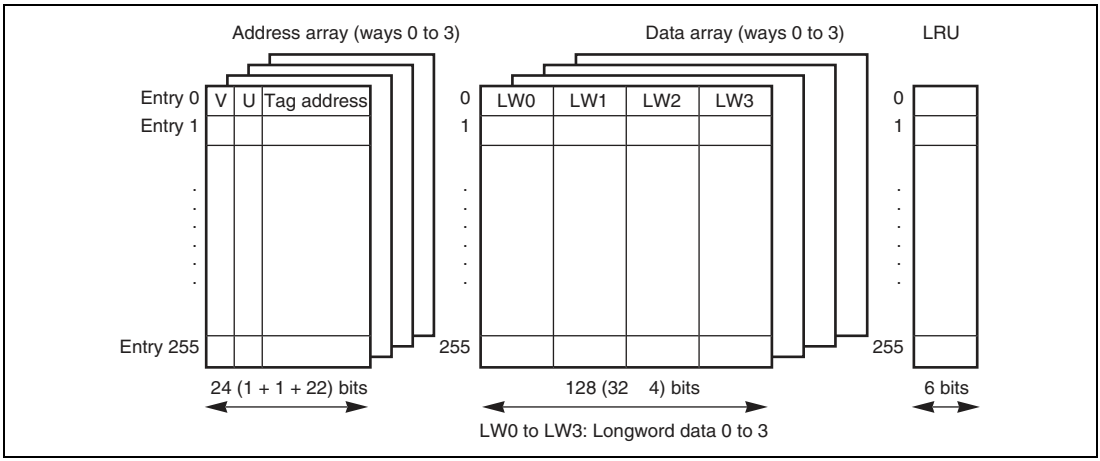


Figure 5.1 Cache Structure

(1) Address Array

The V bit indicates whether the entry data is valid. When the V bit is 1, data is valid; when 0, data is not valid. The U bit indicates whether the entry has been written to in write-back mode. When the U bit is 1, the entry has been written to; when 0, it has not. The tag address holds the physical address used in the external memory access. It is composed of 22 bits (address bits 31 to 10) used for comparison during cache searches.

In this LSI, the top three of 32 physical address bits are used as shadow bits (see section 9, Bus State Controller (BSC)), and therefore the top three bits of the tag address are cleared to 0.

The V and U bits are initialized to 0 by a power-on reset, but are not initialized by a manual reset. The tag address is not initialized by either a power-on or manual reset.

(2) Data Array

Holds a 16-byte instruction or data. Entries are registered in the cache in line units (16 bytes). The data array is not initialized by a power-on or manual reset.

(3) LRU

With the 4-way set associative system, up to four instructions or data with the same entry address can be registered in the cache. When an entry is registered, LRU shows which of the four ways it is recorded in. There are six LRU bits, controlled by hardware. A least-recently-used (LRU) algorithm is used to select the way.

Six LRU bits indicate the way to be replaced, when a cache miss occurs. Table 5.2 shows the relationship between the LRU bits and the way to be replaced when the cache locking mechanism is disabled. (For the relationship when the cache locking mechanism is enabled, refer to section 5.2.2, Cache Control Register 2 (CCR2).) If a bit pattern other than those listed in table 5.2 is set in the LRU bits by software, the cache will not function correctly. When modifying the LRU bits by software, set one of the patterns listed in table 5.2.

The LRU bits are initialized to H'000000 by a power-on reset, but are not initialized by a manual reset.

Table 5.2 LRU and Way Replacement (when Cache Locking Mechanism is Disabled)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000100, 010100, 100000, 110000, 110100	3
000001, 000011, 001011, 100001, 101001, 101011	2
000110, 000111, 001111, 010110, 011110, 011111	1
111000, 111001, 111011, 111100, 111110, 111111	0

5.2 Register Descriptions

The cache has the following registers. Refer to section 37, List of Registers, for more details on the addresses and access size of these registers.

- Cache control register 1 (CCR1)
- Cache control register 2 (CCR2)
- Cache control register 3 (CCR3)

5.2.1 Cache Control Register 1 (CCR1)

The cache is enabled or disabled using the CE bit in CCR1. CCR1 also has a CF bit (which invalidates all cache entries), and WT and CB bits (which select either write-through mode or write-back mode). Programs that change the contents of the CCR1 register should be placed in address space that is not cached.

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CF	0	R/W	Cache Flush Writing 1 flushes all cache entries (clears the V, U, and LRU bits of all cache entries to 0). This bit is always read as 0. Write-back to external memory is not performed when the cache is flushed.
2	CB	0	R/W	Write-Back Indicates the cache's operating mode for space P1. 0: Write-through mode 1: Write-back mode
1	WT	0	R/W	Write-Through Indicates the cache's operating mode for spaces P0, U0, and P3. 0: Write-back mode 1: Write-through mode
0	CE	0	R/W	Cache Enable Indicates whether the cache function is used. 0: The cache function is not used. 1: The cache function is used.

5.2.2 Cache Control Register 2 (CCR2)

The CCR2 register controls the cache locking mechanism in cache lock mode only. The CPU enters the cache lock mode when the DSP bit (bit 12) in the status register (SR) is set to 1 or the lock enable bit (bit 16) in the cache control register 2 (CCR2) is set to 1. The cache locking mechanism is disabled in non-cache lock mode (DSP bit = 0).

When a prefetch instruction (PREF@Rn) is issued in cache lock mode and a cache miss occurs, the line of data pointed to by Rn will be loaded into the cache, according to the setting of bits 9 and 8 (W3LOAD, W3LOCK) and bits 1 and 0 (W2LOAD, W2LOCK in CCR2).

Table 5.3 shows the relationship between the settings of bits and the way that is to be replaced when the cache is missed by a prefetch instruction.

On the other hand, when the cache is hit by a prefetch instruction, new data is not loaded into the cache and the valid entry is held. For example, a prefetch instruction is issued while bits W3LOAD and W3LOCK are set to 1 and the line of data to which Rn points is already in way 0, the cache is hit and new data is not loaded into way 3.

In cache lock mode, bits W3LOCK and W2LOCK restrict the way that is to be replaced, when instructions other than the prefetch instruction are issued. Table 5.4 shows the relationship between the settings of bits in CCR2 and the way that is to be replaced when the cache is missed by instructions other than the prefetch instruction.

Programs that change the contents of the CCR2 register should be placed in address space that is not cached.

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	LE	0	R/W	Lock enable (LE) Controls cache lock mode. 0: Enters cache lock mode when the DSP bit in the SR register is set to 1. 1: Enters cache lock mode regardless of the DSP bit value.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	W3LOAD	0	R/W	Way 3 Load (W3LOAD)
8	W3LOCK	0	R/W	Way 3 Lock (W3LOCK) When the cache is missed by a prefetch instruction while in cache lock mode and when bits W3LOAD and W3LOCK in CCR2 are set to 1, the data is always loaded into way 3. Under any other condition, the prefetched data is loaded into the way to which LRU points.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	W2LOAD	0	R/W	Way 2 Load (W2LOAD)
0	W2LOCK	0	R/W	Way 2 Lock (W2LOCK) When the cache is missed by a prefetch instruction while in cache lock mode and when bits W2LOAD and W2LOCK in CCR2 are set to 1, the data is always loaded into way 2. Under any other condition, the prefetched data is loaded into the way to which LRU points.

Note: W2LOAD and W3LOAD should not be set to 1 at the same time.

Table 5.3 Way Replacement when a PREF Instruction Misses the Cache

DSP Bit	W3LOAD	W3LOCK	W2LOAD	W2LOCK	Way to be Replaced
0	*	*	*	*	Determined by LRU (table 5.2)
1	*	0	*	0	Determined by LRU (table 5.2)
1	*	0	0	1	Determined by LRU (table 5.5)
1	0	1	*	0	Determined by LRU (table 5.6)
1	0	1	0	1	Determined by LRU (table 5.7)
1	0	*	1	1	Way 2
1	1	1	0	*	Way 3

Note: * Don't care
W3LOAD and W2LOAD should not be set to 1 at the same time.

Table 5.4 Way Replacement when Instructions other than the PREF Instruction Miss the Cache

DSP Bit	W3LOAD	W3LOCK	W2LOAD	W2LOCK	Way to be Replaced
0	*	*	*	*	Determined by LRU (table 5.2)
1	*	0	*	0	Determined by LRU (table 5.2)
1	*	0	*	1	Determined by LRU (table 5.5)
1	*	1	*	0	Determined by LRU (table 5.6)
1	*	1	*	1	Determined by LRU (table 5.7)

Note: * Don't care
W3LOAD and W2LOAD should not be set to 1 at the same time.

Table 5.5 LRU and Way Replacement (when W2LOCK = 1 and W3LOCK =0)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000001, 000100, 010100, 100000, 100001, 110000, 110100	3
000011, 000110, 000111, 001011, 001111, 010110, 011110, 011111	1
101001, 101011, 111000, 111001, 111011, 111100, 111110, 111111	0

Table 5.6 LRU and Way Replacement (when W2LOCK = 0 and W3LOCK =1)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000001, 000011, 001011, 100000, 100001, 101001, 101011	2
000100, 000110, 000111, 001111, 010100, 010110, 011110, 011111	1
110000, 110100, 111000, 111001, 111011, 111100, 111110, 111111	0

Table 5.7 LRU and Way Replacement (when W2LOCK = 1 and W3LOCK =1)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000001, 000011, 000100, 000110, 000111, 001011, 001111, 010100, 010110, 011110, 011111	1
100000, 100001, 101001, 101011, 110000, 110100, 111000, 111001, 111011, 111100, 111110, 111111	0

5.2.3 Cache Control Register 3 (CCR3)

The CCR3 register controls the cache size to be used. The cache size must be specified according to the LSI to be selected. If the specified cache size exceeds the size of cache incorporated in the LSI, correct operation cannot be guaranteed. Note that programs that change the contents of the CCR3 register should be placed in un-cached address space. In addition, note that all cache entries must be invalidated by setting the CF bit in the CCR1 to 1 before accessing the cache after the CCR3 is modified.

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 to 16	CSIZE7 to CSIZE0	H'01	R/W	Cache Size Specify the cache size as shown below. 0000 0001: 16-kbyte cache 0000 0010: 32-kbyte cache Settings other than above are prohibited.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

5.3 Operation

5.3.1 Searching the Cache

If the cache is enabled (the CE bit in CCR1 = 1), whenever instructions or data in spaces P0, P1, P3, and U0 are accessed the cache will be searched to see if the desired instruction or data is in the cache. Figure 5.2 illustrates the method by which the cache is searched. The cache is a physical cache and holds physical addresses in its address section. The example of operation in 16-kbyte mode is described below:

Entries are selected using bits 11 to 4 of the address (virtual) of the access to memory and the tag address of that entry is read. In parallel with reading the tag address, the virtual address is converted into the physical address. The virtual address of the access to memory and the physical address (tag address) read from the address array are compared. The address comparison uses all four ways. When the comparison shows a match and the selected entry is valid (V = 1), a cache hit occurs. When the comparison does not show a match or the selected entry is not valid (V = 0), a cache miss occurs. Figure 5.2 shows a hit on way 1.

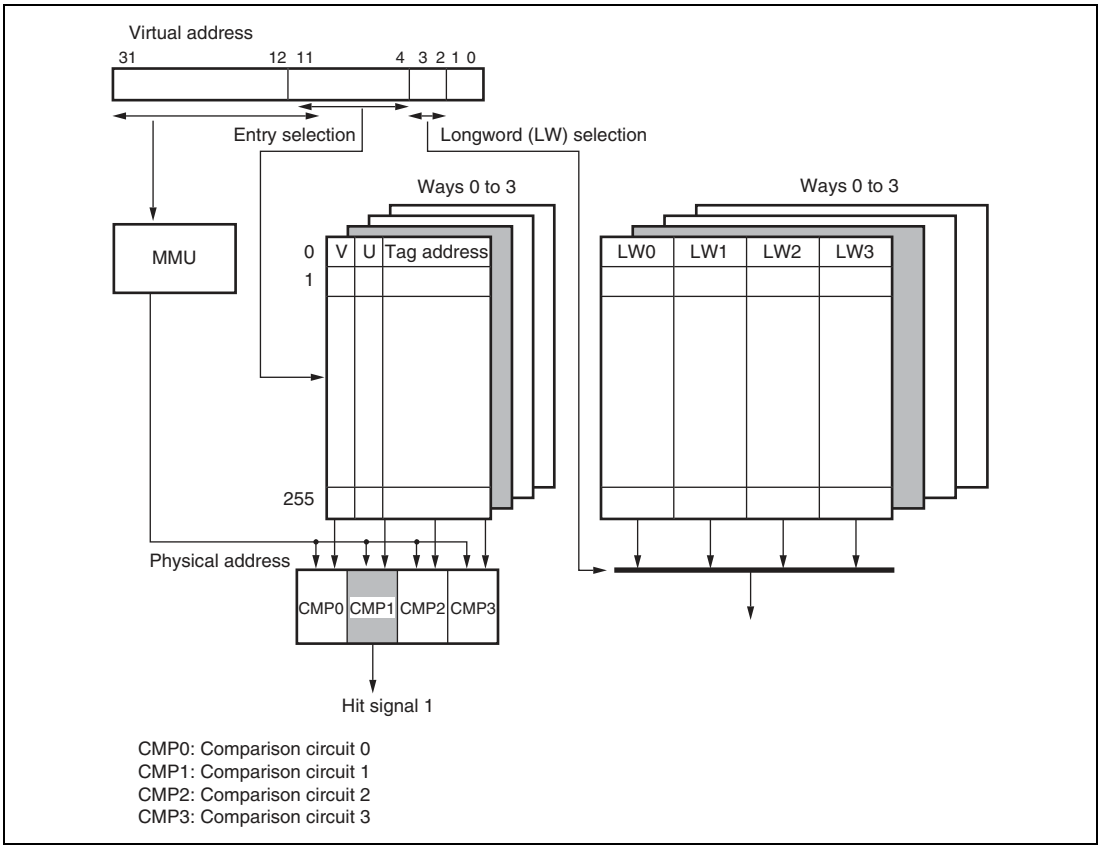


Figure 5.2 Cache Search Scheme

5.3.2 Read Access

(1) Read Hit

In a read access, instructions and data are transferred from the cache to the CPU. The LRU is updated to indicate that the hit way is the most recently hit way.

(2) Read Miss

An external bus cycle starts and the entry is updated. The way to be replaced is shown in table 5.4. Entries are updated in 16-byte units. When the desired instruction or data that caused the miss is loaded from external memory to the cache, the instruction or data is transferred to the CPU in parallel with being loaded to the cache. When it is loaded to the cache, the U bit is cleared to 0 and the V bit is set to 1 to indicate that the hit way is the most recently hit way. When the U bit for the entry which is to be replaced by entry updating in write-back mode is 1, the cache-update cycle starts after the entry is transferred to the write-back buffer. After the cache completes its update cycle, the write-back buffer writes the entry back to the memory. Transfer is in 16-byte units.

5.3.3 Prefetch Operation

(1) Prefetch Hit

The LRU is updated to indicate that the hit way is the most recently hit way. The other contents of the cache are not changed. Instructions and data are not transferred from the cache to the CPU.

(2) Prefetch Miss

Instructions and data are not transferred from the cache to the CPU. The way that is to be replaced is shown in table 5.3. The other operations are the same as those for a read miss.

5.3.4 Write Access

(1) Write Hit

In a write access in write-back mode, the data is written to the cache and no external memory write cycle is issued. The U bit of the entry that has been written to is set to 1, and the LRU is updated to indicate that the hit way is the most recently hit way. In write-through mode, the data is written to the cache and an external memory write cycle is issued. The U bit of the entry that has been written to is not updated, and the LRU is updated to indicate that the hit way is the most recently hit way.

(2) Write Miss

In write-back mode, an external write cycle starts when a write miss occurs, and the entry is updated. The way to be replaced is shown in table 5.4. When the U bit of the entry which is to be replaced by entry updating is 1, the cache-update cycle starts after the entry has been transferred to the write-back buffer. Data is written to the cache and the U bit and the V bit are set to 1. The LRU is updated to indicate that the replaced way is the most recently updated way. After the cache has completed its update cycle, the write-back buffer writes the entry back to the memory. Transfer is in 16-byte units. In write-through mode, no write to cache occurs in a write miss; the write is only to the external memory.

5.3.5 Write-Back Buffer

When the U bit of the entry to be replaced in write-back mode is 1, the entry must be written back to the external memory. To increase performance, the entry to be replaced is first transferred to the write-back buffer and fetching of new entries to the cache takes priority over writing back to the external memory. After the fetching of new entries to the cache completes, the write-back buffer writes the entry back to the external memory. During the write-back cycles, the cache can be accessed. The write-back buffer can hold one line of cache data (16 bytes) and its physical address. Figure 5.3 shows the configuration of the write-back buffer.

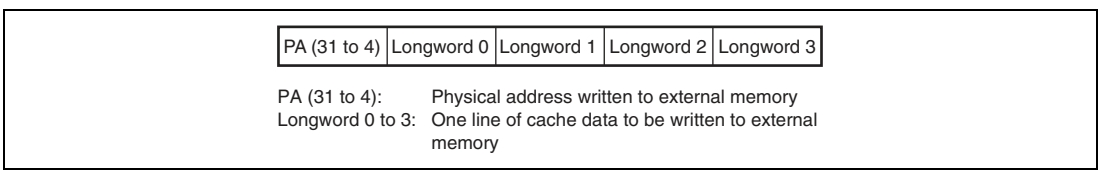


Figure 5.3 Write-Back Buffer Configuration

5.3.6 Coherency of Cache and External Memory

Use software to ensure coherency between the cache and the external memory. When memory shared by this LSI and another device is placed in an address space to which caching applies, use the memory-mapped cache to make the data invalid and written back, as required. Memory that is shared by this LSI's CPU and DMAC should also be handled in this way.

5.4 Memory-Mapped Cache

To allow software management of the cache, cache contents can be read and written by means of MOV instructions in privileged mode. The cache is mapped onto the P4 area in virtual address space. The address array is mapped onto addresses H'F0000000 to H'F0FFFFFF, and the data array onto addresses H'F1000000 to H'F1FFFFFF. Only longword can be used as the access size for the address array and data array, and instruction fetches cannot be performed.

5.4.1 Address Array

The address array is mapped onto H'F0000000 to H'F0FFFFFF. To access an address array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the tag address, V bit, U bit, and LRU bits to be written to the address array.

In the address field, specify the entry address for selecting the entry, W for selecting the way, A for enabling or disabling the associative operation, and H'F0 for indicating address array access. As for W, B'00 indicates way 0, B'01 indicates way 1, B'10 indicates way 2, and B'11 indicates way 3.

In the data field, specify the tag address, LRU bits, U bit, and V bit. Figure 5.4 shows the address and data formats in 16-byte mode. For other cache size modes, change the entry address and Was shown in table 5.8. The following three operations are available in the address array.

(1) Address-Array Read

Read the tag address, LRU bits, U bit, and V bit for the entry that corresponds to the entry address and way specified by the address field of the read instruction. In reading, the associative operation is not performed, regardless of whether the associative bit (A bit) specified in the address is 1 or 0.

(2) Address-Array Write (Non-Associative Operation)

Write the tag address, LRU bits, U bit, and V bit, specified by the data field of the write instruction, to the entry that corresponds to the entry address and way as specified by the address field of the write instruction. Ensure that the associative bit (A bit) in the address field is set to 0. When writing to a cache line for which the U bit = 1 and the V bit =1, write the contents of the cache line back to memory, then write the tag address, LRU bits, U bit, and V bit specified by the data field of the write instruction. Always clear the uppermost 3 bits (bits 31 to 29) of the tag address to 0. When 0 is written to the V bit, 0 must also be written to the U bit for that entry.

(3) Address-Array Write (Associative Operation)

When writing with the associative bit (A bit) of the address = 1, the addresses in the four ways for the entry specified by the address field of the write instruction are compared with the tag address that is specified by the data field of the write instruction. If the MMU is enabled in this case, a virtual address specified by data is translated into a physical address via the TLB before comparison. Write the U bit and the V bit specified by the data field of the write instruction to the entry of the way that has a hit. However, the tag address and LRU bits remain unchanged. When there is no way that receives a hit, nothing is written and there is no operation. This function is used to invalidate a specific entry in the cache. When the U bit of the entry that has received a hit is 1 at this point, writing back should be performed. However, when 0 is written to the V bit, 0 must also be written to the U bit of that entry.

5.4.2 Data Array

The data array is mapped onto H'F1000000 to H'F1FFFFFF. To access a data array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the longword data to be written to the data array.

In the address field, specify the entry address for selecting the entry, L for indicating the longword position within the (16-byte) line, W for selecting the way, and H'F1 for indicating data array access. As for L, B'00 indicates longword 0, B'01 indicates longword 1, B'10 indicates longword 2, and B'11 indicates longword 3. As for W, B'00 indicates way 0, B′01 indicates way 1, B'10 indicates way 2, and B′11 indicates way 3.

Since access size of the data array is fixed at longword, bits 1 and 0 of the address field should be set to B'00.

Figure 5.4 shows the address and data formats in 16-kbyte mode. For other cache size modes, change the entry address and W as shown in table 5.8.

The following two operations on the data array are available. The information in the address array is not affected by these operations.

(1) Data-Array Read

Read the data specified by L of the address filed, from the entry that corresponds to the entry address and the way that is specified by the address filed.

(2) Data-Array Write

Write the longword data specified by the data filed, to the position specified by L of the address field, in the entry that corresponds to the entry address and the way specified by the address field.

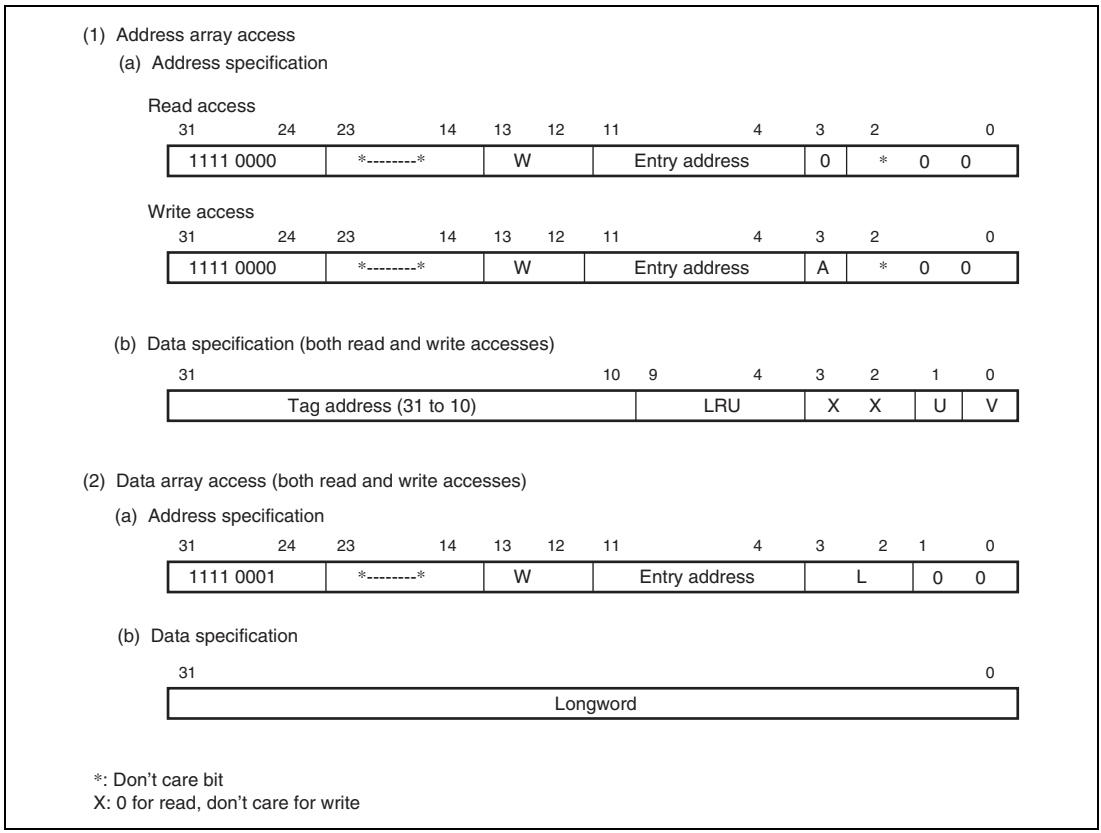


Figure 5.4 Specifying Address and Data for Memory-Mapped Cache Access (16-kbyte mode)

Table 5.8 Address Format Based on the Size of Cache to be Assigned to Memory

Cache Size	Entry Address Bits	W Bit
16 kbytes	11 to 4	13 and 12
32 kbytes	12 to 4	14 to 13

5.4.3 Usage Examples

(1) Invalidating Specific Entries

Specific cache entries can be invalidated by writing 0 to the entry's V bit in the memory-mapped cache access. When the A bit is 1, the tag address specified by the write data is compared to the tag address within the cache selected by the entry address, and a match is found, the entry is written back if the entry's U bit is 1 and the V bit and U bit specified by the write data are written. If no match is found, there is no operation. In the example shown below, R0 specifies the write data and R1 specifies the address.

```
; R0=H'01100010; VPN=B'0000 0001 0001 0000 0000 00, U=0, V=0
; R1=H'F0000088; address array access, entry=B'00001000, A=1
;
MOV.L R0,@R1
```

(2) Reading the Data of a Specific Entry

To read the data field of a specific entry is enabled by the memory-mapped cache access. The longword indicated in the data field of the data array in figure 5.4 is read into the register. In the example shown below, R0 specifies the address and R1 shows what is read.

```
; R0=H'F100 004C; data array access, entry=B'00000100
; Way = 0, longword address = 3
;
MOV.L @R0,R1 ; Longword 3 is read.
```

Section 6 X/Y Memory

This LSI has on-chip X-memory and Y-memory which can be used to store instructions or data.

6.1 Features

- Page

  There are four pages. The X memory is divided into two pages (pages 0 and 1) and the Y memory is divided into two pages (pages 0 and 1).
- Memory map

  The X/Y memory is located in the virtual address space, physical address space, and X-bus and Y-bus address spaces.

  In the virtual address space, this memory is located in the addresses shown in table 6.1. These addresses are included in space P2 (when SR.MD = 1) or Uxy (when SR.MD = 0 and SR.DSP = 1) according to the CPU operating mode.

Table 6.1 X/Y Memory Virtual Addresses

Page	Memory Size (Total Four Pages) 16 kbytes
Page 0 of X memory	H'A5007000 to H'A5007FFF
Page 1 of X memory	H'A5008000 to H'A5008FFF
Page 0 of Y memory	H'A5017000 to H'A5017FFF
Page 1 of Y memory	H'A5018000 to H'A5018FFF

On the other hand, this memory is located in a part of area 1 in the physical address space. When this memory is accessed from the physical address space, addresses in which the upper three bits are 0 in addresses shown in table 6.1 are used. In the X-bus and Y-bus address spaces, addresses in which the upper 16 bits are ignored in addresses of X memory and Y memory shown in table 6.1 are used.

- Ports

  Each page has three independent read/write ports and is connected to each bus. The X memory is connected to the I bus, X bus, and L bus. The Y memory is connected to the I bus, Y bus, and L bus. The L bus is used when this memory is accessed from the virtual address space. The I bus is used when this memory is accessed from the physical address space. The X bus and Y bus are used when this memory is accessed from the X-bus and Y-bus address spaces.

- Priority order

  In the event of simultaneous accesses to the same page from different buses, the accesses are processed according to the priority order. The priority order is: I bus > X bus > L bus in the X memory and I bus > Y bus > L bus in the Y memory.

6.2 Operation

6.2.1 Access from CPU

Methods for accessing by the CPU are directly via the L bus from the virtual addresses, and via the I bus after the virtual addresses are converted to be the physical addresses using the MMU. As long as a conflict on the page does not occur, access via the L bus is performed in one cycle. Several cycles are necessary for accessing via the I bus. According to the CPU operating mode, access from the CPU is as follows:

(1) Privileged mode and privileged DSP mode (SR. MD = 1)

The X/Y memory can be accessed by the CPU directly from space P2. The MMU can be used to map the virtual addresses in spaces P0 and P3 to this memory.

(2) User DSP mode (SR.MD = 0 and SR.DSP = 1)

The X/Y memory can be accessed by the CPU directly from space Uxy. The MMU can be used to map the virtual addresses in space U0 to this memory.

(3) User mode (SR.MD = 0 and SR.DSP = 0)

The MMU can be used to map the virtual addresses in space U0 to this memory.

6.2.2 Access from DSP

Methods for accessing from the DSP differ according to instructions.

With a X data transfer instruction and a Y data transfer instruction, the X/Y memory is always accessed via the X bus or Y bus. As long as a conflict on the page does not occur, access via the X bus or Y bus is performed in one cycle. The X memory access via the X bus and the Y memory access via the Y bus can be performed simultaneously.

In the case of a single data transfer instruction, methods for accessing from the DSP are directly via the L bus from the virtual addresses, and via the I bus after the virtual addresses are converted to be the physical addresses using the MMU. As long as a conflict on the page does not occur,

access via the L bus is performed in one cycle. Several cycles are necessary for accessing via the I bus. According to the CPU operating mode, access from the CPU is as follows:

(1) Privileged DSP mode (SR. MD = 1 and SR.DSP = 1)

The X/Y memory can be accessed by the DSP directly from space P2. The MMU can be used to map the virtual addresses in spaces P0 and P3 to this memory.

(2) User DSP mode (SR.MD = 0 and SR.DSP = 1)

The X/Y memory can be accessed by the DSP directly from space Uxy. The MMU can be used to map the virtual addresses in space U0 to this memory.

6.2.3 Access from Bus Master Module

The X/Y memory is always accessed by bus master modules such as the DMAC and USB host via the I bus, which is a physical address bus. Addresses in which the upper three bits are 0 in addresses shown in table 6.1 must be used.

6.3 Usage Notes

6.3.1 Page Conflict

In the event of simultaneous accesses to the same page from different buses, the conflict on the pages occurs. Although each access is completed correctly, this kind of conflict tends to lower X/Y memory accessibility. Therefore it is advisable to provide software measures to prevent such conflict as far as possible. For example, conflict will not arise if different memory or different pages are accessed by each bus.

6.3.2 Bus Conflict

The I bus is shared by several bus master modules. When the X/Y memory is accessed via the I bus, a conflict between the other I-bus master modules may occur on the I bus. This kind of conflict tends to lower X/Y memory accessibility. Therefore it is advisable to provide software measures to prevent such conflict as far as possible. For example, by accessing the X/Y memory by the CPU not via the I bus but directly from space P2 or Uxy, conflict on the I bus can be prevented.

6.3.3 MMU and Cache Settings

When the X/Y memory is accessed via the I bus using the cache from the CPU and DSP, correct operation cannot be guaranteed. If the X/Y memory is accessed while the cache is enabled (CCR1.CE = 1), it is advisable to access the X/Y memory via the L bus from space P2 or Uxy. If the X/Y memory is accessed from space P0, P3, or U0, it is advisable to access the X/Y memory via the I bus, which does not use the cache, with MMU setting enabled (MMUCR.AT = 1) and cache disabled (C bit = 0) as page attributes. Since access using the MMU occurs via the I bus, several cycles are necessary (the number of necessary cycles varies according to the ratio between the internal clock (Iϕ) and bus clock (Bϕ) or the operation state of the DMAC). In a program that requires high performance, it is advisable to access the X/Y memory from space P2 or Uxy. The relationship described above is summarized in table 6.2.

Table 6.2 MMU and Cache Settings

Setting		Virtual Address Space and Access Enabled or Disabled			
CCR1.CE	**MMUCR.AT**	**P0, U0**	**P1**	**P2, Uxy**	**P3**
0	0	B	B	A	B
0	1	B	B	A	B
1	0	X	X	A	X
1	1	C	X	A	C

Note: A: Accessible (recommended)
B: Accessible
C: Accessible (Note that MMU page attribute must be specified as cache disabled by clearing the C bit to 0.)
X: Not accessible

6.3.4 Sleep Mode

In sleep mode, I bus master modules such as the DMAC cannot access the X/Y memory.

Section 7 Exception Handling

Exception handling is separate from normal program processing, and is performed by a routine separate from the normal program. For example, if an attempt is made to execute an undefined instruction code or an instruction protected by the CPU processing mode, a control function may be required to return to the source program by executing the appropriate operation or to report an abnormality and carry out end processing. In addition, a function to control processing requested by LSI on-chip modules or an LSI external module to the CPU may also be required.

Transferring control to a user-defined exception processing routine and executing the process to support the above functions are called exception handling. This LSI has two types of exceptions: general exceptions and interrupts. The user can execute the required processing by assigning exception handling routines corresponding to the required exception processing and then return to the source program.

A reset input can terminate the normal program execution and pass control to the reset vector after register initialization. This reset operation can also be regarded as an exception handling. This section describes an overview of the exception handling operation. Here, general exceptions and interrupts are referred to as exception handling. For interrupts, this section describes only the process executed for interrupt requests. For details on how to generate an interrupt request, refer to section 8, Interrupt Controller (INTC).

7.1 Register Descriptions

There are five registers for exception handling. A register with an undefined initial value should be initialized by the software. Refer to section 37, List of Registers, for more details on the addresses and access size of these registers.

- TRAPA exception register (TRA)
- Exception event register (EXPEVT)
- Interrupt event register (INTEVT)
- Interrupt event register 2 (INTEVT2)
- Exception address register (TEA)

Figure 7.1 shows the bit configuration of each register.

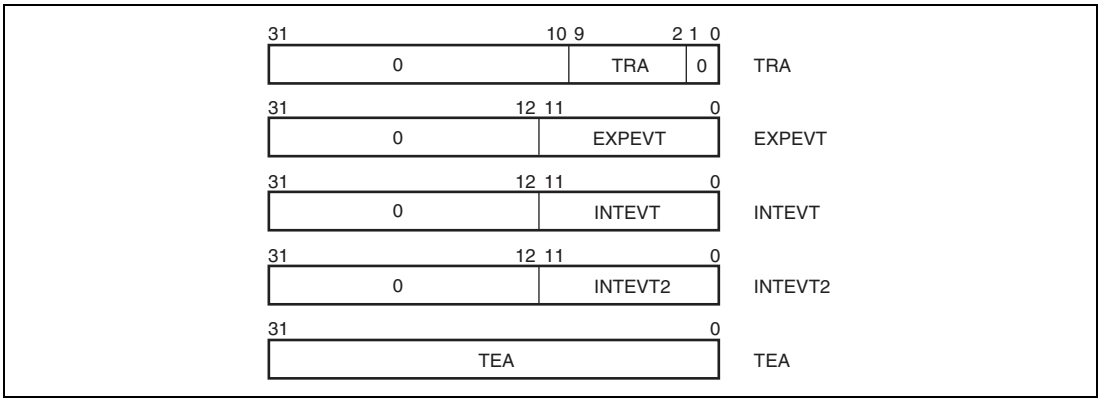


Figure 7.1 Register Bit Configuration

7.1.1 TRAPA Exception Register (TRA)

TRA is assigned to address H'FFFFFFD0 and consists of the 8-bit immediate data (imm) of the TRAPA instruction. TRA is automatically specified by the hardware when the TRAPA instruction is executed. Only bits 9 to 2 of the TRA can be re-written using the software.

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 2	TRA	—	R/W	8-bit Immediate Data
1, 0	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.

7.1.2 Exception Event Register (EXPEVT)

EXPEVT is assigned to address H'FFFFFFD4 and consists of a 12-bit exception code. Exception codes to be specified in EXPEVT are those for resets and general exceptions. These exception codes are automatically specified the hardware when an exception occurs. Only bits 11 to 0 of EXPEVT can be re-written using the software.

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	EXPEVT	*	R/W	12-bit Exception Code

Note: Initialized to H'000 at power-on reset and H'020 at manual reset.

7.1.3 Interrupt Event Register (INTEVT)

INTEVT is assigned to address H'FFFFFFD8 and stores an exception code or a code which indicates interrupt priority order. A code to be specified when an interrupt occurs is determined by an interrupt source. (For details, see section 8.4.6, Interrupt Exception Handling and Priority.) These exception and interrupt priority order codes are automatically specified by the hardware when an exception occurs. INTEVT can be modified using the software. Only bits 11 to 0 of INTEVT can be modified using the software.

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	INTEVT	—	R	12-bit Exception Code

7.1.4 Interrupt Event Register 2 (INTEVT2)

INTEVT2 is assigned to address H'A4000000 and consists of a 12-bit exception code. Exception codes to be specified in INTEVT2 are those for interrupt requests. These exception codes are automatically specified by the hardware when an exception occurs. INTEVT2 cannot be modified using the software.

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	INTEVT2	—	R	12-bit Exception Code

7.1.5 Exception Address Register (TEA)

TEA is assigned to address H'FFFFFFFC and the virtual address for an exception occurrence is stored in this register when an exception related to memory accesses occurs. TEA can be modified using the software.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TEA	All 0	R/W	The virtual address for an exception occurrence

7.2 Exception Handling Function

7.2.1 Exception Handling Flow

In exception handling, the contents of the program counter (PC) and status register (SR) are saved in the saved program counter (SPC) and saved status register (SSR), respectively, and execution of the exception handler is invoked from a vector address. By executing the return from exception handler (RTE) in the exception handler routine, it restores the contents of PC and SR, and returns to the processor state at the point of interruption and the address where the exception occurred.

A basic exception handling sequence consists of the following operations. If an exception occurs and the CPU accepts it, operations 1 to 8 are executed.

1. The contents of PC is saved in SPC.
2. The contents of SR is saved in SSR.
3. The block (BL) bit in SR is set to 1, masking any subsequent exceptions.
4. The mode (MD) bit in SR is set to 1 to place the privileged mode.
5. The register bank (RB) bit in SR is set to 1.
6. An exception code identifying the exception event is written to bits 11 to 0 of the exception event register (EXPEVT); an exception code identifying the interrupt request is written to bits 11 to 0 of the interrupt event register (INTEVT) or interrupt event register 2 (INTEVT2).
7. If a TRAPA instruction is executed, an 8-bit immediate data specified by the TRAPA instruction is set to TRA. For an exception related to memory accesses, the logic address where the exception occurred is written to TEA.*[1]
8. Instruction execution jumps to the designated exception vector address to invoke the handler routine.

The above operations from 1 to 8 are executed in sequence. During these operations, no other exceptions may be accepted unless multiple exception acceptance is enabled.

In an exception handling routine for a general exception, the appropriate exception handling must be executed based on an exception source determined by the EXPEVT. In an interrupt exception handling routine, the appropriate exception handling must be executed based on an exception source determined by the INTEVT or INTEVT2. After the exception handling routine has been completed, program execution can be resumed by executing an RTE instruction. The RTE instruction causes the following operations to be executed.

1. The contents of the SSR are restored into the SR to return to the processing state in effect before the exception handling took place.
2. A delay slot instruction of the RTE instruction is executed.*[2]
3. Control is passed to the address stored in the SPC.

The above operations from 1 to 3 are executed in sequence. During these operations, no other exceptions may be accepted. By changing the SPC and SSR before executing the RTE instruction, a status different from that in effect before the exception handling can also be specified.

Notes: 1. The MMU registers are also modified if an MMU exception occurs.
2. For details on the CPU processing mode in which RTE delay slot instructions are executed, please refer to section 7.5, Usage Notes.

7.2.2 Exception Vector Addresses

A vector address for general exceptions is determined by adding a vector offset to a vector base address. The vector offset for general exceptions other than the TLB miss exception is H'00000100. The vector offset for interrupts is H'00000600. The vector base address is loaded into the vector base register (VBR) using the software. The vector base address should reside in the P1 or P2 fixed physical address space.

7.2.3 Exception Codes

The exception codes are written to bits 11 to 0 of the EXPEVT (for reset or general exceptions) or the INTEVT and INTEVT2 (for interrupt requests) to identify each specific exception event. See section 8, Interrupt Controller (INTC), for details of the exception codes for interrupt requests. Table 7.1 lists exception codes for resets and general exceptions.

7.2.4 Exception Request and BL Bit (Multiple Exception Prevention)

The BL bit in SR is set to 1 when a reset or exception is accepted. While the BL bit is set to 1, acceptance of general exceptions is restricted as described below, making it possible to effectively prevent multiple exceptions from being accepted.

If the BL bit is set to 1, an interrupt request is not accepted and is retained. The interrupt request is accepted when the BL bit is cleared to 0. If the CPU is in low power consumption mode, an interrupt is accepted even if the BL bit is set to 1 and the CPU returns from the low power consumption mode.

A DMA error is not accepted and is retained if the BL bit is set to 1 and accepted when the BL bit is cleared to 0. User break requests generated while the BL bit is set are ignored and are not retained. Accordingly, user breaks are not accepted even if the BL bit is cleared to 0.

If a general exception other than a DMA address error or user break occurs while the BL bit is set to 1, the CPU enters a state similar to that in effect immediately after a reset, and passes control to the reset vector (H'A0000000) (multiple exception). In this case, unlike a normal reset, modules other than the CPU are not initialized, the contents of EXPEVT, SPC, and SSR are undefined, and this status is not detected by an external device.

To enable acceptance of multiple exceptions, the contents of SPC and SSR must be saved while the BL bit is set to 1 after an exception has been accepted, and then the BL bit must be cleared to 0. Before restoring the SPC and SSR, the BL bit must be set to 1.

7.2.5 Exception Source Acceptance Timing and Priority

(1) Exception Request of Instruction Synchronous Type and Instruction Asynchronous Type

Resets and interrupts are requested asynchronously regardless of the program flow. In general exceptions, a DMA address error and a user break under the specific condition are also requested asynchronously. The user cannot expect on which instruction an exception is requested. For general exceptions other than a DMA address error and a user break under a specific condition, each general exception corresponds to a specific instruction.

(2) Re-execution Type and Processing-completion Type Exceptions

All exceptions are classified into two types: a re-execution type and a processing-completion type. If a re-execution type exception is accepted, the current instruction executed when the exception is accepted is terminated and the instruction address is saved to the SPC. After returning from the exception processing, program execution resumes from the instruction where the exception was accepted. In a processing-completion type exception, the current instruction executed when the exception is accepted is completed, the next instruction address is saved to the SPC, and then the exception processing is executed.

During a delayed branch instruction and delay slot, the following operations are executed. A re-execution type exception detected in a delay slot is accepted before executing the delayed branch instruction. A processing-completion type exception detected in a delayed branch instruction or a delay slot is accepted when the delayed branch instruction has been executed. In this case, the acceptance of delayed branch instruction or a delay slot precedes the execution of the branch destination instruction. In the above description, a delay slot indicates an instruction following an unconditional delayed branch instruction or an instruction following a conditional delayed branch instruction whose branch condition is satisfied. If a branch does not occur in a conditional delayed branch, the normal processing is executed.

(3) Acceptance Priority and Test Priority

Acceptance priorities are determined for all exception requests. The priority of resets, general exceptions, and interrupts are determined in this order: a reset is always accepted regardless of the CPU status. Interrupts are accepted only when resets or general exceptions are not requested.

If multiple general exceptions occur simultaneously in the same instruction, the priority is determined as follows.

1. A processing-completion type exception generated at the previous instruction*
2. A user break before instruction execution (re-execution type)
3. An exception related to an instruction fetch (CPU address error and MMU related exceptions: re-execution type)
4. An exception caused by an instruction decode (General illegal instruction exceptions and slot illegal instruction exceptions: re-execution type, unconditional trap: processing-completion type)
5. An exception related to data access (CPU address error and MMU related exceptions: re-execution type)
6. Unconditional trap (processing-completion type)
7. A user break other than one before instruction execution (processing-completion type)
8. DMA address error (processing-completion type)

Note: * If a processing-completion type exception is accepted at an instruction, exception processing starts before the next instruction is executed. This exception processing executed before an exception generated at the next instruction is detected.

Only one exception is accepted at a time. Accepting multiple exceptions sequentially results in all exception requests being processed.

Table 7.1 Exception Event Vectors

Exception Type	Current Instruction	Exception Event	Priority*[1]	Exception Order	Process at BL=1	Vector Code	Vector Offset
Reset (asynchro-nous)	Aborted	Power-on reset	1	1	Reset	H'000	—
		Manual reset	1	2	Reset	H'020	—
General exception events (synchro-nous)	Re-executed	User break(before instruction execution)	2	0	Ignored	H'1E0	H'00000100
		CPU address error (instruction access) *[4]	2	1	Reset	H'0E0	H'00000100
		TLB miss (instruction access) *[4]*[5]	2	1-1	Reset	H'040	H'00000400
		TLB invalid (instruction access)*[4]*[5]	2	1-2	Reset	H'040	H'00000100
		TLB protection violation (instruction access)*[4]*[5]	2	1-3	Reset	H'0A0	H'00000100
		Illegal general instruction exception	2	2	Reset	H'180	H'00000100
		Illegal slot instruction exception	2	2	Reset	H'1A0	H'00000100
		CPU address error (data access)*[4]	2	3	Reset	H'0E0/ H'100	H'00000100
		TLB miss (data access)*[4]*[5]	2	3-1	Reset	H'040/ H'060	H'00000400
	Re-executed	TLB invalid (data access)*[4]*[5]	2	3-2	Reset	H'040/ H'060	H'00000100
		TLB protection violation (data access)*[4]*[5]	2	3-3	Reset	H'0A0/ H'0C0	H'00000100
		Initial page write (data access)*[4]*[5]	2	3-4	Reset	H'080	H'00000100
	Completed	Unconditional trap (TRAPA instruction)	2	4	Reset	H'160	H'00000100
		User breakpoint (After instruction execution, address)	2	5	Ignored	H'1E0	H'00000100

Exception Type	Current Instruction	Exception Event	Priority*1	Exception Order	Process at BL=1	Vector Code	Vector Offset
General exception events (asynchro-nous)	Completed	User breakpoint (Data break, I-BUS break)	2	5	Ignored	H'1E0	H'00000100
		DMA address error	2	6	Retained	H'5C0	H'00000100
General interrupt requests (asynchro-nous)	Completed	Interrupt requests	3	—*2	Retained	—*3	H'00000600

Notes: 1. Priorities are indicated from high to low, 1 being the highest and 3 the lowest.
A reset has the highest priority. An interrupt is accepted only when general exceptions are not requested.
2. For details on priorities in multiple interrupt sources, refer to section 8, Interrupt Controller (INTC).
3. If an interrupt is accepted, the exception event register (EXPEVT) is not changed. The interrupt source code is specified in the interrupt event registers (INTEVT and INTEVT2). For details, refer to section 8, Interrupt Controller (INTC).
4. If one of these exceptions occurs in a specific part of the repeat loop, a specific code and vector offset are specified.
5. These exception codes are valid when the MMU is used.

7.3 Individual Exception Operations

This section describes the conditions for specific exception handling, and the processor operations. This section describes resets and general exceptions. For interrupt operations, refer to section 8, Interrupt Controller (INTC).

7.3.1 Resets

(1) Power-On Reset

- Conditions
  Power-on reset is request
- Operations
  Set EXPEVT to H'000, initialize the CPU and on-chip peripheral modules, and branch to the reset vector H'A0000000. For details, refer to the register descriptions in the relevant sections.

(2) Manual Reset

- Conditions
  Manual reset is request
- Operations
  Set EXPEVT to H'020, initialize the CPU and on-chip peripheral modules, and branch to the reset vector H'A0000000. For details, refer to the register descriptions in the relevant sections.

7.3.2 General Exceptions

(1) CPU address error

- Conditions
  - Instruction is fetched from odd address (4n + 1, 4n + 3)
  - Word data is accessed from addresses other than word boundaries (4n + 1, 4n + 3)
  - Longword is accessed from addresses other than longword boundaries (4n + 1, 4n + 2, 4n + 3)
  - The area ranging from H'80000000 to H'FFFFFFFF in virtual space is accessed in user mode
- Types
  Instruction synchronous, re-execution type

- Save address

  Instruction fetch: An instruction address to be fetched when an exception occurred

  Data access: An instruction address where an exception occurs (a delayed branch instruction address if an instruction is assigned to a delay slot)
- Exception code

  An exception occurred during read: H'0E0

  An exception occurred during write: H'100
- Remarks

  The virtual address (32 bits) that caused the exception is set in TEA.

(2) Illegal general instruction exception

- Conditions
  - When undefined code not in a delay slot is decoded

    Delayed branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S

Note: For details on undefined code, refer to table 2.12. When an undefined code other than H'F000 to H'FFFF is decoded, operation cannot be guaranteed.

  - When a privileged instruction not in a delay slot is decoded in user mode

    Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP; instructions that access GBR with LDC/STC are not privileged instructions.
- Types

  Instruction synchronous, re-execution type
- Save address

  An instruction address where an exception occurs
- Exception code

  H'180
- Remarks

  None

(3) Illegal slot instruction

- Conditions
  - — When undefined code in a delay slot is decoded
    Delayed branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S
  - — When a privileged instruction in a delay slot is decoded in user mode
    Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP; instructions that access GBR with LDC/STC are not privileged instructions.
  - — When an instruction that rewrites PC in a delay slot is decoded
    Instructions that rewrite PC: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT, BF, BT/S, BF/S, TRAPA, LDC Rm, SR, LDC.L @Rm+, SR
- Types
  Instruction synchronous, re-execution type
- Save address
  A delayed branch instruction address
- Exception code
  H'1A0
- Remarks
  None

(4) Unconditional trap

- Conditions
  TRAPA instruction executed
- Types
  Instruction synchronous, processing-completion type
- Save address
  An address of an instruction following TRAPA
- Exception code
  H'160
- Remarks
  The exception is a processing-completion type, so an instruction after the TRAPA instruction is saved to SPC. The 8-bit immediate value in the TRAPA instruction is set in TRA[9:2].

(5) User break point trap

- Conditions

  When a break condition set in the user break controller is satisfied
- Types

  Break (L bus) before instruction execution: Instruction synchronous, re-execution type

  Operand break (L bus): Instruction synchronous, processing-completion type

  Data break (L bus): Instruction asynchronous, processing-completion type

  I bus break: Instruction asynchronous, processing-completion type
- Save address

  Re-execution type: An address of the instruction where a break occurs (a delayed branch instruction address if an instruction is assigned to a delay slot)

  Processing-completion type: An address of the instruction following the instruction where a break occurs (a delayed branch instruction destination address if an instruction is assigned to a delay slot)
- Exception code

  H'1E0
- Remarks

  For details on user break controller, refer to section 33, User Break Controller (UBC).

(6) DMA address error

- Conditions
  - Word data accessed from addresses other than word boundaries (4n + 1, 4n + 3)
  - Longword accessed from addresses other than longword boundaries (4n + 1, 4n + 2, 4n + 3)
- Types

  Instruction asynchronous, processing-completion type
- Save address

  An address of the instruction following the instruction where a break occurs (a delayed branch instruction destination address if an instruction is assigned to a delay slot)
- Exception code

  H'5C0

- Remarks

  An exception occurs when a DMA transfer is executed while an exception instruction address described above is specified in the DMAC. Since the DMA transfer is performed asynchronously with the CPU instruction operation, an exception is also requested asynchronously with the instruction execution. For details on DMAC, refer to section 10, Direct Memory Access Controller (DMAC).

7.3.3 General Exceptions (MMU Exceptions)

When the address translation unit of the memory management unit (MMU) is valid, MMU exceptions are checked after a CPU address error has been checked. Four types of MMU exceptions are defined: TLB miss exception, TLB invalid exception, TLB protection exception, initial page write exception. These exceptions are checked in this order.

A vector offset for a TLB miss exception is defined as H'00000400 to simplify exception source determination. For details on MMU exception operations, refer to section 4, Memory Management Unit (MMU).

(1) TLB miss exception

- Conditions

  Comparison of TLB addresses shows no address match.
- Types

  Instruction synchronous, re-execution type
- Save address

  Instruction fetch: An instruction address to be fetched when an exception occurred

  Data access: An instruction address where an exception occurs (a delayed branch instruction address if an instruction is assigned to a delay slot)
- Exception code

  An exception occurred during read: H'040

  An exception occurred during write: H'060
- Remarks
- The virtual address (32 bits) that caused the exception is set in TEA, and the MMU register is updated. The vector address for TLB miss exception is VBR + H'0400. To speed up TLB miss processing, the offset differs from other exceptions.

(2) TLB invalid exception

- Conditions

  Comparison of TLB addresses shows address match but V = 0.
- Types

  Instruction synchronous, re-execution type
- Save address

  Instruction fetch: An instruction address to be fetched when an exception occurred

  Data access: An instruction address where an exception occurs (a delayed branch instruction address if an instruction is assigned to a delay slot)
- Exception code

  An exception occurred during read: H'040

  An exception occurred during write: H'060
- Remarks

  The virtual address (32 bits) that caused the exception is set in TEA, and the MMU register is updated.

(3) TLB protection exception

- Conditions

When a hit access violates the TLB protection information (PR bits).

- Types

  Instruction synchronous, re-execution type
- Save address

  Instruction fetch: An instruction address to be fetched when an exception occurred

  Data access: An instruction address where an exception occurs (a delayed branch instruction address if an instruction is assigned to a delay slot)
- Exception code

  An exception occurred during read: H'0A0

  An exception occurred during write: H'0C0
- Remarks

  The virtual address (32 bits) that caused the exception is set in TEA, and the MMU register is updated.

(4) Initial page write exception

- Conditions

  A hit occurred to the TLB for a store access, but D = 0.
- Types

  Instruction synchronous, re-execution type
- Save address

  Instruction fetch: An instruction address to be fetched when an exception occurred

  Data access: An instruction address where an exception occurs (a delayed branch instruction address if an instruction is assigned to a delay slot)
- Exception code

  H'080
- Remarks

  The virtual address (32 bits) that caused the exception is set in TEA, and the MMU register is updated.

7.4 Exception Processing While DSP Extension Function is Valid

When the DSP extension function is valid (the DSP bit in SR is set to 1), some exception processing acceptance conditions or exception processing may be changed.

7.4.1 Illegal Instruction Exception and Illegal Slot Instruction Exception

In the DSP mode, a DSP extension instruction can be executed. If a DSP extension instruction is executed when the DSP bit in SR is cleared to 0 (in a mode other than the DSP mode), an illegal instruction exception occurs.

In the DSP mode, STC and LDC instructions for the SR register can be executed even in user mode. (Note, however, that only the RC[11:0], DMX, DMY, and RF[1:0] bits in the DSP extension bits can be changed.)

7.4.2 CPU Address Error

In the DSP mode, a part of the space P2 (Uxy area: H'A5000000 to H'A5FFFFFF) can be accessed in user mode and no CPU address error will occur even if the area is accessed.

7.4.3 Exception in Repeat Control Period

If an exception is requested or an exception is accepted during repeat control, the exception may not be accepted correctly or a program execution may not be returned correctly from exception processing that is different from the normal state. These restrictions may occur from repeat detection instruction to repeat end instruction while the repeat counter is 1 or more. In this section, this period is called the repeat control period.

The following shows program examples where the number of instructions in the repeat loop are 4 or more, 3, 2, and 1, respectively. In this section, a repeat detection instruction and its instruction address are described as RptDtct. The first, second, and third instructions following the repeat detection instruction are described as RptDtct1, RptDtct2, and RptDtct3. In addition, [A], [B], [C1], and [C2] in the following examples indicate instructions where a restriction occurs. Table 7.2 summarizes the instruction positions and restriction types.

Table 7.2 Instruction Positions and Restriction Types

Instruction Position	SPC*[1]	Illegal Instruction*[2]	Interrupt, Break*[3]	CPU Address Error*[4]
[A]				
[B]			Retained	
[C1]		Added	Retained	Instruction/data
[C2]	Illegal	Added	Retained	Instruction/data

Notes: 1. A specific address is specified in the SPC if an exception occurs while SR.RC[11:0] ≥ 2.
2. There are a greater number of instructions that can be illegal instructions while SR.RC[11:0] ≥ 1.
3. An interrupt, break or DMA address error request is retained while SR.RC[11:0] ≥1.
4. A specific exception code is specified while SR.RC[11:0] ≥1.

- Example 1: Repeat loop consisting of four or greater instructions

```
            LDRS RptStart   ; [A]
            LDRE RptDtct + 4      ; [A]
            SETRC #4        ; [A]
            instr0          ; [A]
RptStart:   instr1          ; [A] [Repeat start instruction]
            ……              ; [A]
            ……              ; [A]
RptDtct:    RptDtct         ; [B] A repeat detection
                                  instruction is an
                                  instruction three
                                  instructions before a
                                  repeat end instruction
            RptDtct1        ; [C1]
            RptDtct2        ; [C2]
RptEnd:     RptDtct3        ; [C2] [Repeat end instruction]
            instrNext       ; [A]
```

- Example 2: Repeat loop consisting of three instructions

```
            LDRS   RptDtct + 4     ; [A]
            LDRE   RptDtct + 4     ; [A]
            SETRC #4               ; [A]
RptDtct:    RptDtct                ; [B] A repeat detection
                                         instruction is an
                                         instruction prior to a
                                         repeat start instruction
RptStart:   RptDtct1               ; [C1][Repeat start instruction]
            RptDtct2               ; [C2]
RptEnd:     RptDtct3               ; [C2][Repeat end instruction]
            instrNext              ; [A]
```

- Example 3: Repeat loop consisting of two instructions

```
            LDRS   RptDtct + 6     ; [A]
            LDRE   RptDtct + 4     ; [A]
            SETRC #4               ; [A]
RptDtct:    RptDtct                ; [B] A repeat detection
                                         instruction is an
                                         instruction prior to a
                                         repeat start instruction
RptStart:   RptDtct1                 ; [C1][Repeat start instruction]
RptEnd:     RptDtct2               ; [C2][Repeat end instruction]
            instrNext              ; [A]
```

- Example 4: Repeat loop consisting of one instruction

```
            LDRS   RptDtct + 8     ; [A]
            LDRE   RptDtct + 4     ; [A]
            SETRC #4               ; [A]
RptDtct:    RptDtct                ; [B] A repeat detection
                                         instruction is an
                                         instruction prior to a
                                         repeat start instruction
RptStart:
RptEnd:     RptDtct1               ; [C1][Repeat start
                                     instruction]== [Repeat end
                                     instruction]
            instrNext              ; [A]
```

(1) SPC Saved by an Exception in Repeat Control Period

If an exception is accepted in the repeat control period while the repeat counter (RC[11:0]) in the SR register is two or greater, the program counter to be saved may not indicate the value to be returned correctly. To execute the repeat control after returning from an exception processing, the return address must indicate an instruction prior to a repeat detection instruction. Accordingly, if an exception is accepted in repeat control period, an exception other than re-execution type exception by a repeat detection instruction cannot return to the repeat control correctly.

Table 7.3 SPC Value When a Re-Execution Type Exception Occurs in Repeat Control (SR.RC[11:0]≥2)

Instruction Where an Exception Occurs	Number of Instructions in a Repeat Loop			
	1	2	3	4 or Greater
RptDtct	RptDtct	RptDtct	RptDtct	RptDtct
RptDtct1	RptDtct1	RptDtct1	RptDtct1	RptDtct1
RptDtct2	—	RptDtct1	RptDtct1	RS-4
RptDtct3	—	—	RptDtct1	RS-2

Note: The following labels are used here.

RptDtct: Repeat detection instruction address

RptDtct1: Instruction address immediately after the repeat detect instruction

RptDtct2: Second instruction address from the repeat detect instruction

RptDtct3: Third instruction address from the repeat detect instruction

RS: Repeat start instruction address

If a re-execution type exception is accepted at an instruction in the hatched areas above, a return address to be saved in the SPC is incorrect. If SR.RC[11:0] is 1 or 0, a correct return address is saved in the SPC.

(2) Illegal Instruction Exception in Repeat Control Period

If one of the following instructions is executed at the address following RptDtct1, a general illegal instruction exception occurs. For details on an address to be saved in the SPC, refer to SPC Saved by an Exception in Repeat Control Period in section 7.4.3, Exception in Repeat Control Period.

- Branch instructions
  BRA, BSR, BT, BF, BT/S, BF/S, BSRF, RTS, BRAF, RTE, JSR, JMP, TRAPA
- Repeat control instructions
  SETRC, LDRS, LDRE
- Load instructions for SR, RS, and RE
  LDC Rn,SR, LDC @Rn+,SR, LDC Rn,RE, LDC @Rn+,RE, LDC Rn,RS, LDC @Rn+, Rs

Note: An extension instruction of this LSI and is not disclosed to the user.

In a repeat loop consisting of one to three instructions, some restrictions apply to repeat detection instructions and all the remaining instructions. In a repeat loop consisting of four or more instructions, restrictions apply to only the three instructions that include a repeat end instruction.

(3) An Exception Retained in Repeat Control Period

In the repeat control period, an interrupt or some exception will be retained to prevent an exception acceptance at an instruction where returning from the exception cannot be performed correctly. For details, refer to repeat loop program examples 1 to 4. In the examples, exceptions generated at instructions indicated as [B], [C], ([C1], or [C2]), the following processing is executed.

- Interrupt, DMA address error
  An exception request is not accepted and retained at instructions [B] and [C]. If an instruction indicates as [A] is executed at the next time, an exception request is accepted.* As shown in examples 1 to 4, any interrupt or DMA address error cannot be accepted in a repeat loop consisting of four instructions or less.

Note: An interrupt request or a DMA address error exception request is retained in the interrupt controller (INTC) and the direct memory access controller (DMAC) until the CPU can accept a request.

- User break before instruction execution

  A user break before instruction execution is accepted at instruction [B], and an address of instruction [B] is saved in the SPC. This exception cannot be accepted at instruction [C] but the exception request is retained until an instruction [A] or [B] is executed at the next time. Then, the exception request is accepted before an instruction [A] or [B] is executed. In this case, an address of instruction [A] or [B] is saved in the SPC.
- User break after instruction execution

  A user break after instruction execution cannot be accepted at instructions [B] and [C] but the exception request is retained until an instruction [A] or [B] is executed at the next time. Then, the exception request is accepted before an instruction [A] or [B] is executed. In this case, an address of instruction [A] or [B] is saved in the SPC.

Table 7.4 Exception Acceptance in the Repeat Loop

Exception Type	Instruction [B]	Instruction [C]
Interrupt	Not accepted	Not accepted
DMA address error	Not accepted	Not accepted
User break before instruction execution	Accepted	Not accepted
User break after instruction execution	Not accepted	Not accepted

(4) CPU Address Error in Repeat Control Period

If a CPU address error occurs in the repeat control period, the exception is accepted but an exception code (H'070) indicating the repeat loop period is specified in the EXPEVT. If a CPU address error occurs in instructions following a repeat detection instruction to repeat end instruction, an exception code for instruction access or data access is specified in the EXPEVT.

The SPC is saved according to the description, SPC Saved by an Exception in Repeat Control Period in section 7.4.3, Exception in Repeat Control Period.

After the CPU address error exception processing, the repeat control cannot be returned correctly. To execute a repeat loop correctly, care must be taken not to generate a CPU address error in the repeat control period.

Note: In a repeat loop consisting of one to three instructions, some restrictions apply to repeat detection instructions and all the remaining instructions. In a repeat loop consisting of four or more instructions, restrictions apply to only the three instructions that include a repeat end instruction. The restriction occurs when SR.RC[11:0] ≥ 1.

Table 7.5 Instruction Where a Specific Exception Occurs When a Memory Access Exception Occurs in Repeat Control (SR.RC[11:0]≥1)

Instruction Where an Exception Occurs	Number of Instructions in a Repeat Loop			
	1	2	3	4 or Greater
RptDtct	—	—	—	—
RptDtct1	Instruction/data access	Instruction/data access	Instruction/data access	Instruction/data access
RptDtct2	—	Instruction/data access	Instruction/data access	Instruction/data access
RptDtct3	—	—	Instruction/data access	Instruction/data access

Note: The following labels are used here.
RptDtct: Repeat detection instruction address
RptDtct1: Instruction address immediately after the repeat detect instruction
RptDtct2: Second instruction address from the repeat detect instruction
RptDtct3: Third instruction address from the repeat detect instruction

(5) MMU Exception in Repeat Control Period

If an MMU exception occurs in the repeat control period, a specific exception code is generated as well as a CPU address error. For a TLB miss exception, TLB invalid exception, and initial page write exception, an exception code (H'070) indicating the repeat loop period is specified in the EXPEVT. For a TLB protection exception, an exception code (H'0D0) is specified in the EXPEVT. In a TLB miss exception, vector offset is specified as H'00000100.

An instruction where an exception occurs and the SPC value to be saved are the same as those for the CPU address error.

After this exception processing, the repeat control cannot be returned correctly. To execute a repeat loop correctly, care must be taken not to generate an MMU related exception in the repeat control period.

Note: In a repeat loop consisting of one to three instructions, some restrictions apply to repeat detection instructions and all the remaining instructions. In a repeat loop consisting of four or more instructions, restrictions apply to only the three instructions that include a repeat end instruction. The restriction occurs when SR.RC[11:0] ≥ 1.

7.5 Usage Notes

1. An instruction assigned at a delay slot of the RTE instruction is executed after the contents of the SSR is restored into the SR. An acceptance of an exception related to instruction access is determined according to the SR before restore. An acceptance of other exceptions is determined by processing mode of the SR after restore, and BL bit value. A processing-completion type exception is accepted before an instruction at the RTE branch destination address is executed. However, note that the correct operation cannot be guaranteed if a re-execution type exception occurs.
2. In an instruction assigned at a delay slot of the RTE instruction, a user break cannot be accepted.
3. If the MD and BL bits of the SR register are changed by the LDC instruction, an exception is accepted according to the changed SR value from the next instruction.* A processing-completion type exception is accepted before the next instruction is executed. An interrupt and DMA address error in re-execution type exceptions are accepted before the next instruction is executed.

Note: * If an LDC instruction is executed for the SR, the following instructions are re-fetched and an instruction fetch exception is accepted according to the modified SR value.

Section 8 Interrupt Controller (INTC)

The interrupt controller (INTC) determines the priority of interrupt sources and controls interrupt requests to the CPU. The INTC registers set the priority of each interrupt, allowing the user to process interrupt requests according to the user-set priority.

8.1 Features

- 16 levels of interrupt priority can be set

  By setting the interrupt-priority registers, the priorities of on-chip peripheral modules, and IRQ and PINT interrupts can be selected from 16 levels for individual request sources.
- NMI noise canceller function

  An NMI input-level bit indicates the NMI pin state. By reading this bit in the interrupt exception service routine, the pin state can be checked, enabling it to be used as a noise canceller.
- IRQ interrupts can be set

  Detection of low level, high level, rising edge, or falling edge
- Interrupt request signal can be externally output ($\overline{\text{IRQOUT}}$ pin)

  By notifying the external bus master that the external interrupt and on-chip peripheral module interrupt requests have been generated, the bus mastership can be requested.

Figure 8.1 shows a block diagram of the interrupt controller.

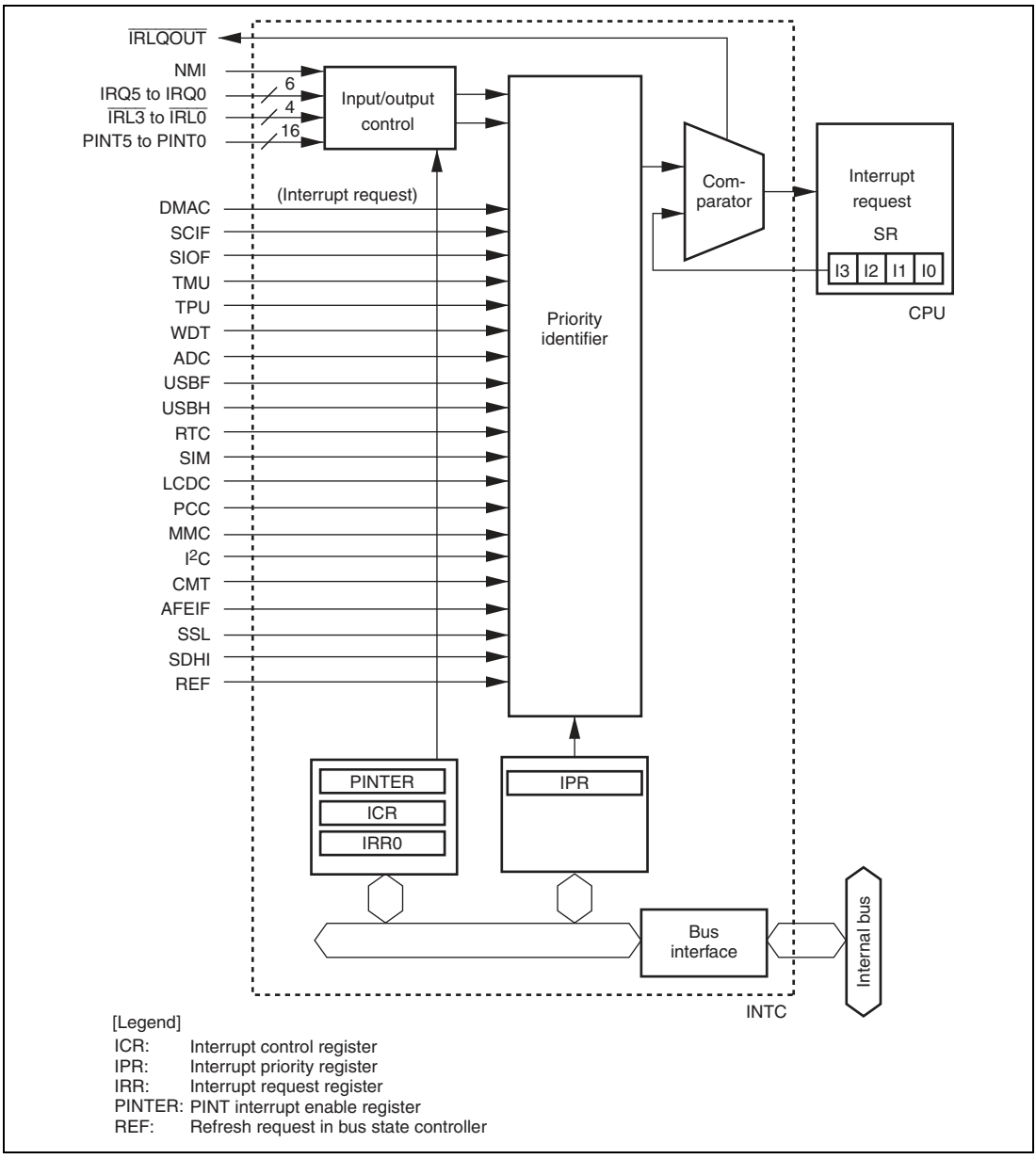


Figure 8.1 Block Diagram of INTC

8.2 Input/Output Pins

Table 8.1 shows the INTC pin configuration.

Table 8.1 Pin Configuration

Name	Abbreviation	I/O	Description
Nonmaskable interrupt input pin	NMI	Input	Input of interrupt request signal, not maskable by the interrupt mask bits in SR
Interrupt input pins	IRQ5 to IRQ0 $\overline{IRL3}$ to $\overline{IRL0}$*1	Input	Input of interrupt request signals
Port interrupt input pins	PINT15 to PINT0	Input	Input of port interrupt signals
Bus request signal pin	$\overline{IRQOUT}$*2	Output	Bus request signal for an interrupt

Notes: 1. $\overline{IRL3}$ to $\overline{IRL0}$ and IRQ3 to IRQ0 cannot be used simultaneously because these pins are multiplexed.

2. When the NMI or H-UDI interrupt requests are generated and the response time of CPU is short, this pin may not be asserted.

8.3 Register Descriptions

The INTC has the following registers. Refer to section 37, List of Registers, for more details on the addresses and access size of these registers.

- Interrupt control register 0 (ICR0)
- Interrupt control register 1 (ICR1)
- Interrupt control register 2 (ICR2)
- PINT interrupt enable register (PINTER)
- Interrupt priority register A (IPRA)
- Interrupt priority register B (IPRB)
- Interrupt priority register C (IPRC)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)
- Interrupt priority register H (IPRH)
- Interrupt priority register I (IPRI)
- Interrupt priority register J (IPRJ)
- Interrupt request register 0 (IRR0)
- Interrupt request register 1 (IRR1)
- Interrupt request register 2 (IRR2)
- Interrupt request register 3 (IRR3)
- Interrupt request register 4 (IRR4)
- Interrupt request register 5 (IRR5)
- Interrupt request register 6 (IRR6)
- Interrupt request register 7 (IRR7)
- Interrupt request register 8 (IRR8)
- Interrupt request register 9 (IRR9)

8.3.1 Interrupt Priority Registers A to J (IPRA to IPRJ)

IPRA to IPRJ are 16-bit readable/writable registers in which priority levels from 0 to 15 are set for on-chip peripheral module and IRQ interrupts.

Bit	Bit Name	Initial Value	R/W	Description
15	IPR15	0	R/W	These bits set the priority level for each interrupt source in 4-bit units. For details, see table 8.2.
14	IPR14	0	R/W	
13	IPR13	0	R/W	
12	IPR12	0	R/W	
11	IPR11	0	R/W	
10	IPR10	0	R/W	
9	IPR9	0	R/W	
8	IPR8	0	R/W	
7	IPR7	0	R/W	
6	IPR6	0	R/W	
5	IPR5	0	R/W	
4	IPR4	0	R/W	
3	IPR3	0	R/W	
2	IPR2	0	R/W	
1	IPR1	0	R/W	
0	IPR0	0	R/W	

Table 8.2 Interrupt Sources and IPRA to IPRJ

Register	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
IPRA	TMU0	TMU1	TMU2	RTC
IPRB	WDT	REF	SIM	Reserved*
IPRC	IRQ3	IRQ2	IRQ1	IRQ0
IPRD	Reserved*	TMU (TMU_SUNI)	IRQ5	IRQ4
IPRE	DMAC (1)	Reserved*	LCDC	SSL
IPRF	ADC	DMAC (2)	USBF	CMT
IPRG	SCIF0	SCIF1	Reserved*	Reserved*
IPRH	PINTA	PINTB	TPU	I^2C
IPRI	SIOF0	SIOF1	MMC	PCC
IPRJ	Reserved*	USBH	SDHI	AFEIF

Note: * Reserved. Always read as 0. The write value should always be 0. The SSL and SDHI-related bits are effective only for the models that include them. Reserved bits apply if they are not included.

As shown in table 8.2, on-chip peripheral module or IRQ interrupts are assigned to four 4-bit groups in each register. These 4-bit groups (bits 15 to 12, bits 11 to 8, bits 7 to 4, and bits 3 to 0) are set with values from H'0 (0000) to H'F (1111). Setting H'0 means priority level 0 (masking is requested); H'F means priority level 15 (the highest level).

8.3.2 Interrupt Control Register 0 (ICR0)

ICR0 is a register that sets the input signal detection mode of the external interrupt input pin NMI, and indicates the input signal level at the NMI pin.

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	0/1*	R	NMI Input Level Sets the level of the signal input at the NMI pin. This bit can be read from to determine the NMI pin level. This bit cannot be modified. 0: NMI input level is low 1: NMI input level is high
14 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
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
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * The initial value is 1 when NMI input is high, 0 when NMI input is low.

8.3.3 Interrupt Control Register 1 (ICR1)

ICR1 is a 16-bit register that specifies the detection mode for external interrupt input pins IRQ5 to IRQ0 individually: rising edge, falling edge, high level, or low level.

Bit	Bit Name	Initial Value	R/W	Description
15	MAI	0	R/W	All Interrupt Mask When this bit is set to 1, all interrupt requests are masked while low level is input to the NMI pin. The NMI interrupt is masked in standby mode. 0: When the NMI pin is low, all interrupt requests are not masked 1: When the NMI pin is low, all interrupt requests are masked
14	IRQLVL	1	R/W	Interrupt Request Level Detection Enables or disables the use of pins IRQ3 to IRQ0 as four independent interrupt pins. The IRQ4 and IRQ5 are not affected. 0: Use of pins IRQ3 to IRQ0 as four independent interrupt pins enabled 1: Use of pins $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ as encoded 15 level interrupt pins
13	BLMSK	0	R/W	BL Bit Mask When the BL bit in the SR register is set to 1, specifies whether the NMI interrupt is masked. 0: When the BL bit is set to 1, the NMI interrupt is masked 1: The NMI interrupt is accepted regardless of the BL bit setting
12	—	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
11	IRQ51S	0	R/W	IRQn Sense Select
10	IRQ50S	0	R/W	These bits select whether interrupt request signals corresponding to pins IRQ5 to IRQ0 are detected by a rising edge, falling edge, high level, or low level.
9	IRQ41S	0	R/W	
8	IRQ40S	0	R/W	
7	IRQ31S	0	R/W	
6	IRQ30S	0	R/W	
5	IRQ21S	0	R/W	
4	IRQ20S	0	R/W	
3	IRQ11S	0	R/W	
2	IRQ10S	0	R/W	
1	IRQ01S	0	R/W	
0	IRQ00S	0	R/W	

Bit 2n + 1	Bit 2n	
IRQn1S	**IRQn0S**	
0	0	Interrupt request is detected on falling edge of IRQn input
0	1	Interrupt request is detected on rising edge of IRQn input
1	0	Interrupt request is detected on low level of IRQn input
1	1	Interrupt request is detected on high level of IRQn input

[Legend] n= 0 to 5

8.3.4 Interrupt Request Register 0 (IRR0)

IRR0 is an 8-bit register that indicates interrupt requests from the TMU and IRQ0 to IRQ5.

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	TMU_ SUNIR	0	R/W	TMU_SUNI Interrupt Request Indicates whether the TMU_SUNI (TMU) interrupt request is generated. 0: TMU_SUNI interrupt request is not generated 1: TMU_SUNI interrupt request is generated
5	IRQ5R	0	R/W	IRQn Interrupt Request Indicates whether there is interrupt request input to the IRQn pin. When edge-detection mode is set for IRQn, an interrupt request is cleared by writing 0 to the IRQnR bit after reading IRQnR = 1. When level-detection mode is set for IRQn, these bits indicate whether an interrupt request is input. The interrupt request is set/cleared by only 1/0 input to the IRQn pin. IRQnR 0: No interrupt request input to IRQn pin 1: Interrupt request input to IRQn pin [Legend] n = 0 to 5
4	IRQ4R	0	R/W	
3	IRQ3R	0	R/W	
2	IRQ2R	0	R/W	
1	IRQ1R	0	R/W	
0	IRQ0R	0	R/W	

8.3.5 Interrupt Request Register 1 (IRR1)

IRR1 is an 8-bit register that indicates whether interrupt requests from the DMAC are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

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	DEI3R	0	R/W	DEI3 Interrupt Request Indicates whether the DEI3 (DMAC) interrupt is generated. 0: DEI3 interrupt request is not generated 1: DEI3 interrupt request is generated
2	DEI2R	0	R/W	DEI2 Interrupt Request Indicates whether the DEI2 (DMAC) interrupt request is generated. 0: DEI2 interrupt request is not generated 1: DEI2 interrupt request is generated
1	DEI1R	0	R/W	DEI1 Interrupt Request Indicates whether the DEI1 (DMAC) interrupt request is generated. 0: DEI1 interrupt request is not generated 1: DEI1 interrupt request is generated
0	DEI0R	0	R/W	DEI0 Interrupt Request Indicates whether the DEI0 (DMAC) interrupt request is generated. 0: DEI0 interrupt request is not generated 1: DEI0 interrupt request is generated

8.3.6 Interrupt Request Register 2 (IRR2)

IRR2 is an 8-bit register that indicates whether interrupt requests from the SSL and LCDC are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Note: On the models not having the SSL, the SSL-related bits are reserved. The write value should always be 0.

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	SSLIR	0	R/W	SSLI Interrupt Request Indicates whether the SSLI (SSL) interrupt request is generated. 0: SSLI interrupt request is not generated 1: SSLI interrupt request is generated Note: On the models not having the SSL, this bit is reserved and always read as 0. The write value should always be 0.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	LCDIR	0	R/W	LCDCI Interrupt Request Indicates whether the LCDCI (LCDC) interrupt request is generated. 0: LCDCI interrupt request is not generated 1: LCDCI interrupt request is generated

8.3.7 Interrupt Request Register 3 (IRR3)

IRR3 is an 8-bit register that indicates whether interrupt requests from the RTC and SIM are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	TENDIR	0	R/W	TEI Interrupt Request Indicates whether the TEI (SIM) interrupt is generated. 0: TEI interrupt request is not generated 1: TEI interrupt request is generated
6	TXIR	0	R/W	TXI Interrupt Request Indicates whether the TXI (SIM) interrupt request is generated. 0: TXI interrupt request is not generated 1: TXI interrupt request is generated
5	RXIR	0	R/W	RXI Interrupt Request Indicates whether the RXI (SIM) interrupt request is generated. 0: RXI interrupt request is not generated 1: RXI interrupt request is generated
4	ERIR	0	R/W	ERI Interrupt Request Indicates whether the ERI (SIM) interrupt request is generated. 0: ERI interrupt request is not generated 1: ERI interrupt request is generated
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	CUIR	0	R/W	CUI Interrupt Request Indicates whether the CUI (RTC) interrupt request is generated. 0: CUI interrupt request is not generated 1: CUI interrupt request is generated

Bit	Bit Name	Initial Value	R/W	Description
1	PRIR	0	R/W	PRI Interrupt Request Indicates whether the PRI (RTC) interrupt request is generated. 0: PRI interrupt request is not generated 1: PRI interrupt request is generated
0	ATIR	0	R/W	ATI Interrupt Request Indicates whether the ATI (RTC) interrupt request is generated. 0: ATI interrupt request is not generated 1: ATI interrupt request is generated

8.3.8 Interrupt Request Register 4 (IRR4)

IRR4 is an 8-bit register that indicates whether interrupt requests from the REF, WDT, and TMU are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit always read as 0. The write value should always be 0.
6	TUNI2R	0	R/W	TUNI2 Interrupt Request Indicates whether the TUNI2 (TMU) interrupt request is generated. 0: TUNI2 interrupt request is not generated 1: TUNI2 interrupt request is generated
5	TUNI1R	0	R/W	TUNI1Interrupt Request Indicates whether the TUNI1 (TMU) interrupt request is generated. 0: TUNI1 interrupt request is not generated 1: TUNI1 interrupt request is generated

Bit	Bit Name	Initial Value	R/W	Description
4	TUNI0R	0	R/W	TUNI0 Interrupt Request Indicates whether the TUNI0 (TMU) interrupt request is generated. 0: TUNI0 interrupt request is not generated 1: TUNI0 interrupt request is generated
3	ITIR	0	R/W	ITI Interrupt Request Indicates whether the ITI (WDT) interrupt request is generated. 0: ITI interrupt request is not generated 1: ITI interrupt request is generated
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RCMIR	0	R/W	RCMI Interrupt Request Indicates whether the RCMI (REF) interrupt request is generated. 0: RCMI interrupt request is not generated 1: RCMI interrupt request is generated

8.3.9 Interrupt Request Register 5 (IRR5)

IRR5 is an 8-bit register that indicates whether interrupt requests from the SCIF0, SCIF1, DMAC, and ADC are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ADCIR	0	R/W	ADI Interrupt Request Indicates whether the ADI (ADC) interrupt request is generated. 0: ADI interrupt request is not generated 1: ADI interrupt request is generated
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	DEI5R	0	R/W	DEI5 Interrupt Request Indicates whether the DEI5 (DMAC) interrupt request is generated. 0: DEI5 interrupt request is not generated 1: DEI5 interrupt request is generated
4	DEI4R	0	R/W	DEI4 Interrupt Request Indicates whether the DEI4 (DMAC) interrupt request is generated. 0: DEI4 interrupt request is not generated 1: DEI4 interrupt request is generated
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	SCIF1IR	0	R/W	SCIFI1 Interrupt Request Indicates whether the SCIFI1 (SCIF1) interrupt request is generated. 0: SCIFI1 interrupt request is not generated 1: SCIFI1 interrupt request is generated
0	SCIF0IR	0	R/W	SCIFI0 Interrupt Request Indicates whether the SCIFI0 (SCIF0) interrupt request is generated. 0: SCIFI0 interrupt request is not generated 1: SCIFI0 interrupt request is generated

8.3.10 Interrupt Request Register 6 (IRR6)

IRR6 is an 8-bit register that indicates whether interrupt requests from the PINT, SIOF0, and SIOF1 are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	SIOF1IR	0	R/W	SIOFI1 Interrupt Request Indicates whether the SIOFI1 (SIOF1) interrupt request is generated. 0: SIOFI1 interrupt request is not generated 1: SIOFI1 interrupt request is generated
4	SIOF0IR	0	R/W	SIOFI0 Interrupt Request Indicates whether the SIOFI0 (SIOF0) interrupt request is generated. 0: SIOFI0 interrupt request is not generated 1: SIOFI0 interrupt request is generated
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PINTBR	0	R/W	PINTB Interrupt Request Indicates whether the PINTB (PINT) interrupt request is generated. 0: PINTB interrupt request is not generated 1: PINTB interrupt request is generated
0	PINTAR	0	R/W	PINTA Interrupt Request Indicates whether the PINTA (PINT) interrupt request is generated. 0: PINTA interrupt request is not generated 1: PINTA interrupt request is generated

8.3.11 Interrupt Request Register 7 (IRR7)

IRR7 is an 8-bit register that indicates whether interrupt requests from the TPU and IIC are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized 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. The write value should always be 0.
4	IICIR	0	R/W	IICI Interrupt Request Indicates whether the IICI (IIC) interrupt request is generated. 0: IICI interrupt request is not generated 1: IICI interrupt request is generated
3	TPI3R	0	R/W	TPI3 Interrupt Request Indicates whether the TPI3 (TPU) interrupt request is generated. 0: TPI3 interrupt request is not generated 1: TPI3 interrupt request is generated
2	TPI2R	0	R/W	TPI2 Interrupt Request Indicates whether the TPI2 (TPU) interrupt request is generated. 0: TPI2 interrupt request is not generated 1: TPI2 interrupt request is generated
1	TPI1R	0	R/W	TPI1 Interrupt Request Indicates whether the TPI1 (TPU) interrupt request is generated. 0: TPI1 interrupt request is not generated 1: TPI1 interrupt request is generated
0	TPI0R	0	R/W	TPI0 Interrupt Request Indicates whether the TPI0 (TPU) interrupt request is generated. 0: TPI0 interrupt request is not generated 1: TPI0 interrupt request is generated

8.3.12 Interrupt Request Register 8 (IRR8)

IRR8 is an 8-bit register that indicates whether interrupt requests from the SDHI, MMC, and AFEIF are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Note: Note: On the models not having the SDHI, the SDHI-related bits are reserved. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7	MMCI3R	0	R/W	MMCI3 Interrupt Request Indicates whether the MMCI3 (MMC) interrupt request is generated. 0: MMCI3 interrupt request is not generated 1: MMCI3 interrupt request is generated
6	MMCI2R	0	R/W	MMCI2 Interrupt Request Indicates whether the MMCI2 (MMC) interrupt request is generated. 0: MMCI2 interrupt request is not generated 1: MMCI2 interrupt request is generated
5	MMCI1R	0	R/W	MMCI1 Interrupt Request Indicates whether the MMCI1 (MMC) interrupt request is generated. 0: MMCI1 interrupt request is not generated 1: MMCI1 interrupt request is generated
4	MMCI0R	0	R/W	MMCI0 Interrupt Request Indicates whether the MMCI0 (MMC) interrupt request is generated. 0: MMCI0 interrupt request is not generated 1: MMCI0 interrupt request is generated
3	AFECIR	0	R/W	AFECI Interrupt Request Indicates whether the AFECI (AFEIF) interrupt request is generated. 0: AFECI interrupt request is not generated 1: AFECI interrupt request is generated

Bit	Bit Name	Initial Value	R/W	Description
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SDIR	0	R/W	SDI Interrupt Request Indicates whether the SDI (SDHI) interrupt request is generated. 0: SDI interrupt request is not generated 1: SDI interrupt request is generated Note: On the models not having the SDHI, this bit is reserved and always read as 0. The write value should always be 0.

8.3.13 Interrupt Request Register 9 (IRR9)

IRR9 is an 8-bit register that indicates whether interrupt requests from the PCC, USBH, USBF, and CMT are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	PCCIR	0	R/W	PCCI Interrupt Request Indicates whether the PCCI (PCC) interrupt request is generated. 0: PCCI interrupt request is not generated 1: PCCI interrupt request is generated
6	USBHIR	0	R	USBHI Interrupt Request Indicates whether the USBHI (USBH) interrupt request is generated. 0: USBHI interrupt request is not generated 1: USBHI interrupt request is generated
5	—	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
4	CMIR	0	R/W	CMI Interrupt Request Indicates whether the CMI (CMT) interrupt request is generated. 0: CMI interrupt request is not generated 1: CMI interrupt request is generated
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	USBFI1R	0	R	USBFI1 Interrupt Request Indicates whether the USBFI1 (USBF) interrupt request is generated. 0: USBFI1interrupt request is not generated 1: USBFI1 interrupt request is generated
1	USBFI0R	0	R	USBFI0 Interrupt Request Indicates whether the USBFI0 (USBF) interrupt request is generated. 0: USBFI0 interrupt request is not generated 1: USBFI0 interrupt request is generated
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

8.3.14 PINT Interrupt Enable Register (PINTER)

PINTER is a 16-bit register which enables interrupt requests input to the external interrupt input pins PINT0 to PINT15. This register is initialized to H'0000 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15	PINT15E	0	R/W	PINTn Interrupt Enable
14	PINT14E	0	R/W	Select whether the interrupt requests input to the pins PINT15 to PINT0 is enabled.
13	PINT13E	0	R/W	
12	PINT12E	0	R/W	0: Disable PINTn input interrupt requests
11	PINT11E	0	R/W	1: Enable PINTn input interrupt requests
10	PINT10E	0	R/W	n = 0 to 15
9	PINT9E	0	R/W	
8	PINT8E	0	R/W	
7	PINT7E	0	R/W	
6	PINT6E	0	R/W	
5	PINT5E	0	R/W	
4	PINT4E	0	R/W	
3	PINT3E	0	R/W	
2	PINT2E	0	R/W	
1	PINT1E	0	R/W	
0	PINT0E	0	R/W	

8.3.15 Interrupt Control Register 2 (ICR2)

INCR2 is a 16-bit register which specifies low or high detection mode to the external interrupt input pins PINT0 to PINT15 individually. This register is initialized to H'0000 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15	PINT15S	0	R/W	PINTn Sense Select
14	PINT14S	0	R/W	Selects whether to detect an interrupt request signal for the pins PINT15 to PINT0 by a high-level or low-level.
13	PINT13S	0	R/W	
12	PINT12S	0	R/W	0: Detects interrupt request by PINTn input low
11	PINT11S	0	R/W	1: Detects interrupt request by PINTn input high
10	PINT10S	0	R/W	n = 0 to 15
9	PINT9S	0	R/W	
8	PINT8S	0	R/W	
7	PINT7S	0	R/W	
6	PINT6S	0	R/W	
5	PINT5S	0	R/W	
4	PINT4S	0	R/W	
3	PINT3S	0	R/W	
2	PINT2S	0	R/W	
1	PINT1S	0	R/W	
0	PINT0S	0	R/W	

8.4 Interrupt Sources

There are four types of interrupt sources: NMI, IRQ, IRL, and on-chip peripheral modules. Each interrupt has a priority level (0 to 16), with 1 the lowest and 16 the highest. Priority level 0 masks an interrupt, so the interrupt request is ignored.

8.4.1 NMI Interrupt

The NMI interrupt has the highest priority level of 16. When the BLMSK bit in the interrupt control register 1 (ICR1) is 1 or the BL bit in the status register (SR) is 0, NMI interrupts are accepted if the MAI bit in ICR1 is 0. NMI interrupts are edge-detected. In sleep or standby mode, the interrupt is accepted regardless of the BL setting. The NMI edge select bit (NMIE) in the interrupt control register 0 (ICR0) is used to select either rising or falling edge detection.

When using edge-input detection for NMI interrupts, a pulse width of at least two Pϕ cycles (peripheral clock) is necessary. NMI interrupt exception handling does not affect the interrupt mask bits (I3 to I0) in the status register (SR). When the BL bit is 1, only an NMI interrupt is accepted if the BLMSK bit in ICR1 is 1.

It is possible to wake the chip up from sleep mode or standby mode with an NMI interrupt.

8.4.2 IRQ Interrupts

IRQ interrupts are input by level or edge from pins IRQ0 to IRQ5. The priority level can be set by interrupt priority registers C and D (IPRC and IPRD) in a range from 0 to 15.

When using edge-sensing for IRQ interrupts, clear the interrupt source by having software read 1 from the corresponding bit in IRR0, then write 0 to the bit.

When ICR1 is rewritten, IRQ interrupts may be mistakenly detected, depending on the IRQ pin states. To prevent this, rewrite the register while interrupts are masked, then release the mask after clearing the illegal interrupt by reading the interrupt request register 0 (IRR0) and writing 0 to IRR0.

Edge input interrupt detection requires input of a pulse width of more than two cycles on a Pϕ clock basis.

When using level-sensing for IRQ interrupts, the pin levels must be retained until the CPU samples the pins. Therefore, the interrupt source must be cleared by the interrupt handler.

The interrupt mask bits (I3 to I0) in the status register (SR) are not affected by IRQ interrupt handling. IRQ interrupts specified for edge detection can be used to recover from a standby state

when the corresponding interrupt level is higher than that set in the I3 to I0 bits of the SR register. (However, when RTC is used, recovering from standby by using the clock for RTC is enabled.)

8.4.3 IRL interrupts

IRL interrupts are input by pins $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ as level. The priority level is the higher level that is indicated by $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ pins. When the values of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ pins are 0 (B'0000), it indicates the highest level interrupt request (interrupt priority level 15). When the values of the pins are 15 (B'1111), no interrupt is requested (interrupt priority level 0). Figure 8.2 shows an example of connection for IRL interrupt.

IRL interrupts are included with noise canceller function and detected when the sampled levels of each peripheral module clock keep same value for 2 cycles. This prevents sampling error level in $\overline{\text{IRL}}$ pin changing.

IRL interrupts priority level should be kept until interrupt is accepted and its handling is started. However, changing to higher level is enabled.

The interrupt mask bits I3 to I0 in the status register (SR) are not affected by the IRL interrupt handling.

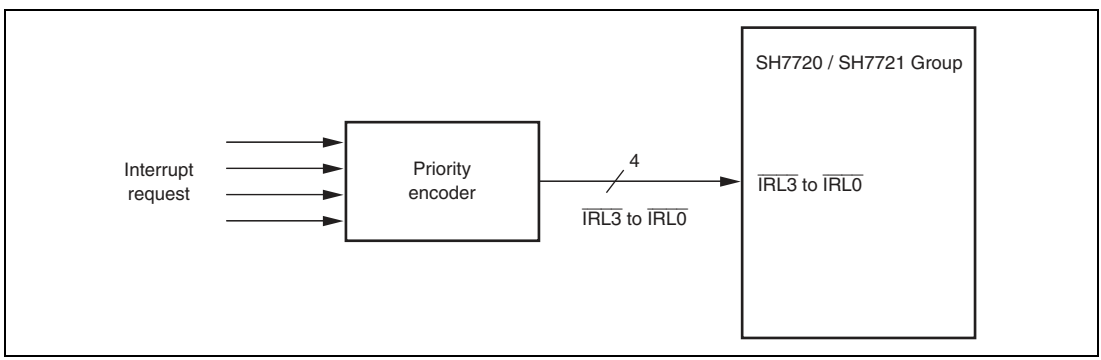


Figure 8.2 Example of IRL Interrupt Connection

8.4.4 PINT Interrupts

PINT interrupts are input by level from pins PINT0 to PINT15. The priority level of PINT0 to PINT7 (PINTA) and PINT8 to PINT15 (PINTB) can be set by the interrupt priority level register H (IPRH) in a range from 0 to 15. The PINT interrupt level should be retained until the interrupt processing starts after an 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.

While an RTC clock is supplied, recovery from a standby state on a PINT interrupt is possible if the interrupt level is higher than that set in the I3 to I0 bits of the SR register.

8.4.5 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are generated by the following modules:

- DMA controller (DMAC)
- I^2C bus interface (IIC)
- Smart card interface (SIM)
- Compare match timer (CMT)
- Timer unit (TMU)
- Timer pulse unit (TPU)
- Watchdog timer (WDT)
- User debugging interface (H-UDI)
- LCD controller (LCDC)
- Secure sockets layer (SSL)
- Analog front end interface (AFEIF)
- USB function controller (USBF)
- USB host controller (USBH)
- Bus state controller (BSC)
- Serial I/O with FIFO 0 (SIOF0)
- Serial I/O with FIFO 1 (SIOF1)
- Serial communication interface with FIFO 0 (SCIF0)
- Serial communication interface with FIFO 1 (SCIF1)
- MultiMediaCard interface (MMC)
- SD host interface (SDHI)
- Realtime clock (RTC)

- A/D converter (ADC)
- PC card controller (PCC)

Not every interrupt source is assigned a different interrupt vector. Sources are reflected in the interrupt event registers (INTEVT and INTEVT2). It is easy to identify sources by using the value of INTEVT or INTEVT2 as a branch offset.

A priority level (from 0 to 15) can be set for each module except H-UDI by writing to the interrupt priority registers A, B, and E to J (IPRA, IPRB, and IPRE to IPRJ). The priority level of the H-UDI interrupt is 15 (fixed).

The interrupt mask bits (I3 to I0) in the status register are not affected by on-chip peripheral module interrupt handling.

8.4.6 Interrupt Exception Handling and Priority

There are four types of interrupt sources: NMI, IRQ, IRL, and on-chip peripheral modules. The priority of each interrupt source is set within priority levels 0 to 16; level 16 is the highest and level 1 is the lowest. When the priority is set to level 0, that interrupt is masked and the interrupt request is ignored.

Tables 8.3 and 8.4 list the interrupt sources, the codes for the interrupt event registers (INTEVT and INTEVT2), and the interrupt priority.

Each interrupt source is assigned a unique code by INTEVT and INTEVT2. The start address of the exception handling routine is common for each interrupt source. This is why, for instance, the value of INTEVT or INTEVT2 is used as an offset at the start of the exception handling routine and branched to in order to identify the interrupt source.

IRQ interrupt and on-chip peripheral module interrupt priorities can be set freely between 0 and 15 for each module by setting interrupt priority registers A to J (IPRA to IPRJ). A reset assigns priority level 0 to IRQ and 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 order is the default priority order indicated at the right in tables 8.3 and 8.4.

Table 8.3 Interrupt Exception Handling Sources and Priority (IRQ Mode)

Interrupt Source		Interrupt Code *1	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
NMI		H'1C0*2	16	—	—	High
H-UDI		H'5E0*2	15	—	—	
IRQ	IRQ0	H'600*3	0 to 15 (0)	IPRC (3 to 0)	—	
	IRQ1	H'620*3	0 to 15 (0)	IPRC (7 to 4)	—	
	IRQ2	H'640*3	0 to 15 (0)	IPRC (11 to 8)	—	
	IRQ3	H'660*3	0 to 15 (0)	IPRC (15 to 12)	—	
	IRQ4	H'680*3	0 to 15 (0)	IPRD (3 to 0)	—	
	IRQ5	H'6A0*3	0 to 15 (0)	IPRD (7 to 4)	—	
TMU	TMU_SUNI	H'6C0*3	0 to 15 (0)	IPRD (11 to 8)	—	
DMAC (1)	DEI0	H'800*3	0 to 15 (0)	IPRE (15 to 12)	High	
	DEI1	H'820*3	0 to 15 (0)			
	DEI2	H'840*3	0 to 15 (0)			
	DEI3	H'860*3	0 to 15 (0)		Low	
LCDC	LCDCI	H'900*3	0 to 15 (0)	IPRE (7 to 4)	—	
SSL	SSLI	H'980*3	0 to 15 (0)	IPRE (3 to 0)	—	
USBF	USBFI0	H'A20*3	0 to 15 (0)	IPRF (7 to 4)	High	
	USBFI1	H'A40*3			Low	
USBH	USBHI	H'A60*3	0 to 15 (0)	IPRJ (11 to 8)	—	
DMAC (2)	DEI4	H'B80*3	0 to 15 (0)	IPRF (11 to 8)	High	
	DEI5	H'BA0*3			Low	
ADC	ADI	H'BE0*3	0 to 15 (0)	IPRF (15 to 12)	—	
SCIF0	SCIFI0	H'C00*3	0 to 15 (0)	IPRG (15 to 12)	—	
SCIF1	SCIFI1	H'C20*3	0 to 15 (0)	IPRG (11 to 8)	—	
PINT	PINTA	H'C80*3	0 to 15 (0)	IPRH (15 to 12)	—	
	PINTB	H'CA0*3	0 to 15 (0)	IPRH (11 to 8)	—	
SIOF0	SIOFI0	H'D00*3	0 to 15 (0)	IPRI (15 to 12)	—	
SIOF1	SIOFI1	H'D20*3	0 to 15 (0)	IPRI (11 to 8)	—	Low

Interrupt Source		Interrupt Code *1	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
TPU	TPI0	H'D80*3	0 to 15 (0)	IPRH (7 to 4)	High	High
	TPI1	H'DA0*3				
	TPI2	H'DC0*3				
	TPI3	H'DE0*3			Low	
IIC	IICI	H'E00*3	0 to 15 (0)	IPRH (3 to 0)	—	
MMC	MMCI0	H'E80*3	0 to 15 (0)	IPRI (7 to 4)	High	
	MMCI1	H'EA0*3				
	MMCI2	H'EC0*3				
	MMCI3	H'EE0*3			Low	
CMT	CMI	H'F00*3	0 to 15 (0)	IPRF (3 to 0)	—	
PCC	PCCI	H'F60*3	0 to 15 (0)	IPRI (3 to 0)	—	
SDHI	SDI	H'F80*3	0 to 15 (0)	IPRJ (7 to 4)	—	
AFEIF	AFECI	H'FE0*3	0 to 15 (0)	IPRJ (3 to 0)	—	
TMU0	TUNI0	H'400*2	0 to 15 (0)	IPRA (15 to 12)	—	
TMU1	TUNI1	H'420*2	0 to 15 (0)	IPRA (11 to 8)	—	
TMU2	TUNI2	H'440*2	0 to 15 (0)	IPRA (7 to 4)	—	
RTC	ATI	H'480*2	0 to 15 (0)	IPRA (3 to 0)	High	
	PRI	H'4A0*2				
	CUI	H'4C0*2			Low	
SIM	ERI	H'4E0*2	0 to 15 (0)	IPRB (7 to 4)	High	
	RXI	H'500*2				
	TXI	H'520*2				
	TEI	H'540*2			Low	
WDT	ITI	H'560*2	0 to 15 (0)	IPRB (15 to 12)	—	
REF	RCMI	H'580*2	0 to 15 (0)	IPRB (11 to 8)	—	Low

Notes: 1. INTEVT2 code.
2. The code set in INTEVT is as same as INTEVT2.
3. The code set in INTEVT indicates interrupt level H'200 to H'3C0. For the correspondence of interrupt level and INTEVT, see table 8.5.

Table 8.4 Interrupt Exception Handling Sources and Priority (IRL Mode)

Interrupt Source		Interrupt Code *[1]	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
NMI		H'1C0*[2]	16	—	—	High
H-UDI		H'5E0*[2]	15	—	—	
IRL	$\overline{IRL3}$ to $\overline{RL0}$=B'0000	H'200*[3]	15	—	—	
	$\overline{IRL3}$ to $\overline{IRL0}$=B'0001	H'220*[3]	14	—	—	
	$\overline{IRL3}$ to $\overline{IRL0}$=B'0010	H'240*[3]	13	—	—	
	$\overline{IRL3}$ to $\overline{IRL0}$=B'0011	H'260*[3]	12	—	—	
	$\overline{IRL3}$ to $\overline{IRL0}$=B'0100	H'280*[3]	11	—	—	
	$\overline{IRL3}$ to $\overline{IRL0}$=B'0101	H'2A0*[3]	10	—	—	
	$\overline{IRL3}$ to $\overline{IRL0}$=B'0110	H'2C0*[3]	9	—	—	
	$\overline{IRL3}$ to $\overline{IRL0}$=B'0111	H'2E0*[3]	8	—	—	
	$\overline{IRL3}$ to $\overline{IRL0}$=B'1000	H'300*[3]	7	—	—	
	$\overline{IRL3}$ to $\overline{IRL0}$=B'1001	H'320*[3]	6	—	—	
	$\overline{IRL3}$ to $\overline{IRL0}$=B'1010	H'340*[3]	5	—	—	
	$\overline{IRL3}$ to $\overline{IRL0}$=B'1011	H'360*[3]	4	—	—	
	$\overline{IRL3}$ to $\overline{IRL0}$=B'1100	H'380*[3]	3	—	—	
	$\overline{IRL3}$ to $\overline{IRL0}$=B'1101	H'3A0*[3]	2	—	—	
	$\overline{IRL3}$ to $\overline{IRL0}$=B'1110	H'3C0*[3]	1	—	—	
IRQ	IRQ4	H'680*[3]	0 to 15 (0)	IPRD (3 to 0)	—	
	IRQ5	H'6A0*[3]	0 to 15 (0)	IPRD (7 to 4)	—	
TMU	TMU_SUNI	H'6C0*[3]	0 to 15 (0)	IPRD (11 to 8)	—	
DMAC (1)	DEI0	H'800*[3]	0 to 15 (0)	IPRE (15 to 12)	High	
	DEI1	H'820*[3]	0 to 15 (0)			
	DEI2	H'840*[3]	0 to 15 (0)			
	DEI3	H'860*[3]	0 to 15 (0)		Low	
LCDC	LCDCI	H'900*[3]	0 to 15 (0)	IPRE (7 to 4)	—	
SSL	SSLI	H'980*[3]	0 to 15 (0)	IPRE (3 to 0)	—	Low

Interrupt Source		Interrupt Code *1	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
USBF	USBFI0	H'A20*3	0 to 15 (0)	IPRF (7 to 4)	High	High
	USBFI1	H'A40*3			Low	
USBH	USBHI	H'A60*3	0 to 15 (0)	IPRJ (11 to 8)	—	
DMAC (2)	DEI4	H'B80*3	0 to 15 (0)	IPRF (11 to 8)	High	
	DEI5	H'BA0*3			Low	
ADC	ADI	H'BE0*3	0 to 15 (0)	IPRF (15 to 12)	—	
SCIF0	SCIFI0	H'C00*3	0 to 15 (0)	IPRG (15 to 12)	—	
SCIF1	SCIFI1	H'C20*3	0 to 15 (0)	IPRG (11 to 8)	—	
PINT	PINTA	H'C80*3	0 to 15 (0)	IPRH (15 to 12)	—	
	PINTB	H'CA0*3	0 to 15 (0)	IPRH (11 to 8)	—	
SIOF0	SIOFI0	H'D00*3	0 to 15 (0)	IPRI (15 to 12)	—	
SIOF1	SIOFI1	H'D20*3	0 to 15 (0)	IPRI (11 to 8)	—	
TPU	TPI0	H'D80*3	0 to 15 (0)	IPRH (7 to 4)	High	
	TPI1	H'DA0*3				
	TPI2	H'DC0*3				
	TPI3	H'DE0*3			Low	
IIC	IICI	H'E00*3	0 to 15 (0)	IPRH (3 to 0)	—	
MMC	MMCI0	H'E80*3	0 to 15 (0)	IPRI (7 to 4)	High	
	MMCI1	H'EA0*3				
	MMCI2	H'EC0*3				
	MMCI3	H'EE0*3			Low	
CMT	CMI	H'F00*3	0 to 15 (0)	IPRF (3 to 0)	—	
PCC	PCCI	H'F60*3	0 to 15 (0)	IPRI (3 to 0)	—	
SDHI	SDI	H'F80*3	0 to 15 (0)	IPRJ (7 to 4)	—	
AFEIF	AFECI	H'FE0*3	0 to 15 (0)	IPRJ (3 to 0)	—	Low

Interrupt Source		Interrupt Code *1	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
TMU0	TUNI0	H'400*2	0 to 15 (0)	IPRA (15 to 12)	—	High
TMU1	TUNI1	H'420*2	0 to 15 (0)	IPRA (11 to 8)	—	
TMU2	TUNI2	H'440*2	0 to 15 (0)	IPRA (7 to 4)	—	
RTC	ATI	H'480*2	0 to 15 (0)	IPRA (3 to 0)	High	
	PRI	H'4A0*2				
	CUI	H'4C0*2			Low	
SIM	ERI	H'4E0*2	0 to 15 (0)	IPRB (7 to 4)	High	
	RXI	H'500*2				
	TXI	H'520*2				
	TEI	H'540*2			Low	
WDT	ITI	H'560*2	0 to 15 (0)	IPRB (15 to 12)	—	
REF	RCMI	H'580*2	0 to 15 (0)	IPRB (11 to 8)	—	Low

Notes: 1. INTEVT2 code.
2. The code set in INTEVT is as same as INTEVT2.
3. The code set in INTEVT indicates interrupt level H'200 to H'3C0. For the correspondence of interrupt level and INTEVT, see table 8.5.

Table 8.5 Interrupt Level and INTEVT Code

Interrupt Level	INTEVT Code
15	H'200
14	H'220
13	H'240
12	H'260
11	H'280
10	H'2A0
9	H'2C0
8	H'2E0
7	H'300
6	H'320
5	H'340
4	H'360
3	H'380
2	H'3A0
1	H'3C0

8.5 Operation

8.5.1 Interrupt Sequence

The sequence of interrupt operations is described below. Figure 8.3 is a flowchart of the operations.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest-priority interrupt from the interrupt requests sent, following the priority levels set in the interrupt priority registers A to J (IPRA to IPRJ). 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 table 8.3, Interrupt Exception Handling Sources and Priority (IRQ Mode) and table 8.4, Interrupt Exception Handling Sources and Priority (IRL Mode).
3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt mask bits (I3 to I0) in the status register (SR) of the CPU. If the request priority level is higher than the level in bits I3 to I0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. Detection timing: The INTC operates, and notifies the CPU of interrupt requests, in synchronization with the peripheral clock (Pϕ). The CPU receives an interrupt at a break in instructions.
5. The interrupt source code is set in the interrupt event registers (INTEVT and INTEVT2).
6. The status register (SR) and program counter (PC) are saved to SSR and SPC, respectively.
7. The block bit (BL), mode bit (MD), and register bank bit (RB) in SR are set to 1.
8. The CPU jumps to the start address of the interrupt handler (the sum of the value set in the vector base register (VBR) and H'00000600). This jump is not a delayed branch. The interrupt handler may branch with the INTEVT or INTEVT2 value as its offset in order to identify the interrupt source. This enables it to branch to the handling routine for the individual interrupt source.

Notes: 1. The interrupt mask bits (I3 to I0) in the status register (SR) are not changed by acceptance of an interrupt in this LSI.

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 after it has been cleared, and then clear the BL bit or execute an RTE instruction.

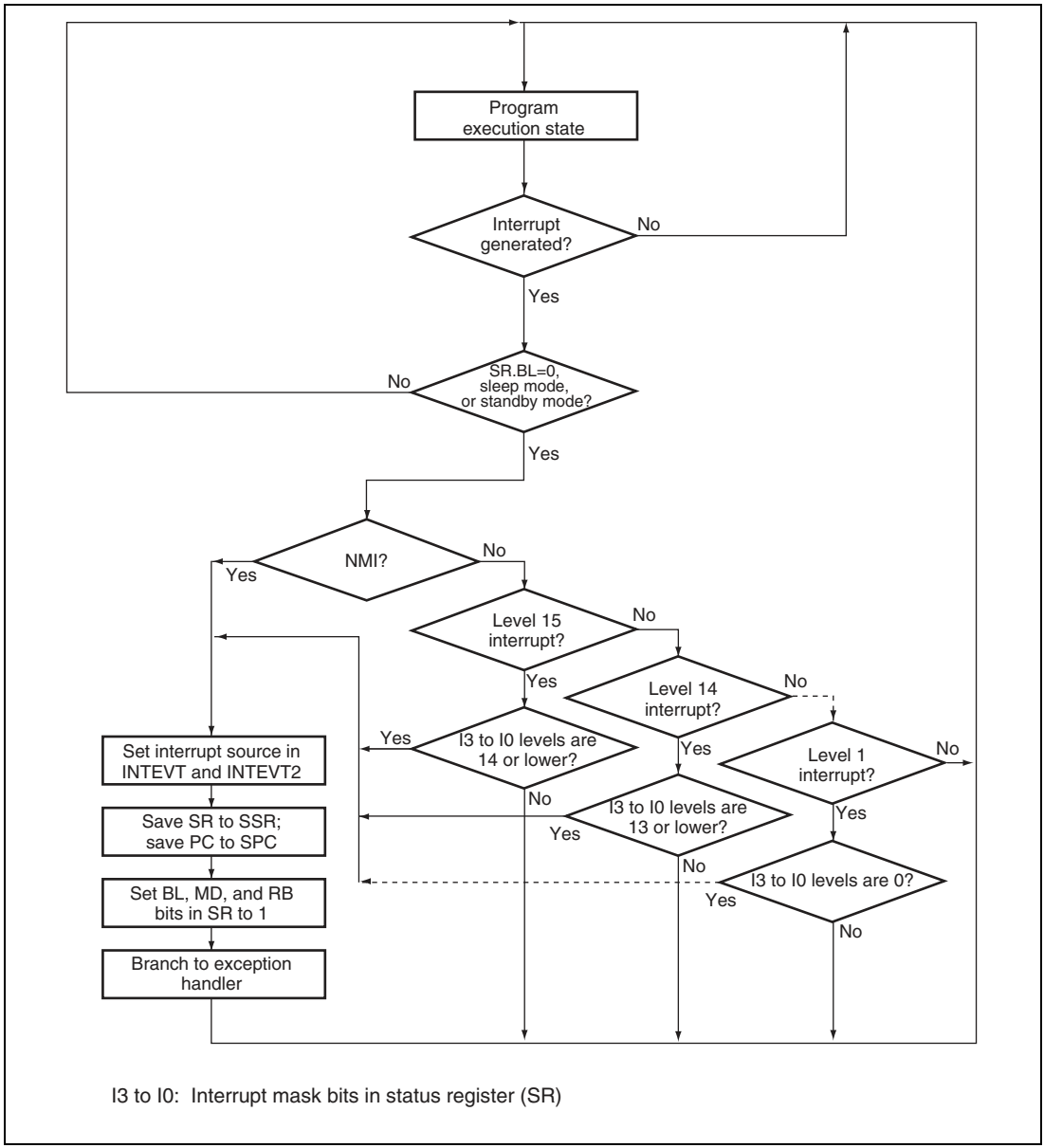


Figure 8.3 Interrupt Operation Flowchart

8.5.2 Multiple Interrupts

When handling multiple interrupts, an interrupt handler should include the following procedures:

1. To determine the interrupt source, branch to a specific interrupt handler corresponding to a code set in INTEVT or INTEVT2. The code in INTEVT or INTEVT2 can be used as an offset for branching to the specific handler.
2. Clear the interrupt source in each specific handler.
3. Save SSR and SPC to memory.
4. Clear the BL bit in SR, and set the accepted interrupt level in the interrupt mask bits in SR.
5. Handle the interrupt.
6. Execute the RTE instruction.

When these procedures are followed in order, an interrupt of higher priority than the one being handled can be accepted after clearing BL in step 4. Figure 8.3 shows a sample interrupt operation flowchart.

Section 9 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 directly with SRAM, SDRAM, and other memory storage devices, and external devices.

9.1 Features

The BSC has the following features:

(1) External address space

- A maximum 32 or 64 Mbytes for each of the eight areas, CS0, CS2 to CS4, CS5A, CS5B, CS6A and CS6B, totally 384 Mbytes (divided into eight areas).
- A maximum 64 Mbytes for each of the six areas, CS0, CS2 to CS4, CS5, and CS6, totally a total of 384 Mbytes (divided into six areas).
- Can specify the normal space interface, byte-selection SRAM, burst ROM (clock synchronous or asynchronous), SDRAM, PCMCIA for each address space.
- Can select the data bus width (8, 16, or 32 bits) for each address space.
- Controls the insertion of the wait state for each address space.
- Controls the insertion of the wait state for each read access and write access.
- Can set the independent idling cycle in the continuous access for five cases: read-write (in same space/different space), read-read (in same space/different space), or the first cycle is a write access.

(2) Normal space interface

- Supports the interface that can directly connect to the 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 direct interface

- Supports IC memory cards and I/O card interfaces defined in the JEIDA specifications Ver. 4.2 (PCMCIA2.1 Rev 2.1).
- 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) Burst ROM (clock synchronous) interface

- Can connect directly to a burst ROM of the clock synchronous type.

(8) Bus arbitration

- Shares all of the resources with other CPU and outputs the bus enable after receiving the bus request from external devices.

(9) 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).

(10) Interval timer using refresh counter

- Generates an interrupt request by a compare match.

Note: The PCMCIA direct interfaces supported by the BSC are only signals and bus protocols shown in table 9.1. For details on other control signals, see section 29, PC Card Controller (PCC) (external circuits and this LSI on-chip PC card controller).
Both area 5 and area 6 have the PCMCIA direct interface function which is common to the SH3. The on-chip PC card controller supports only area 6.

The block diagram of the BSC is shown in figure 9.1.

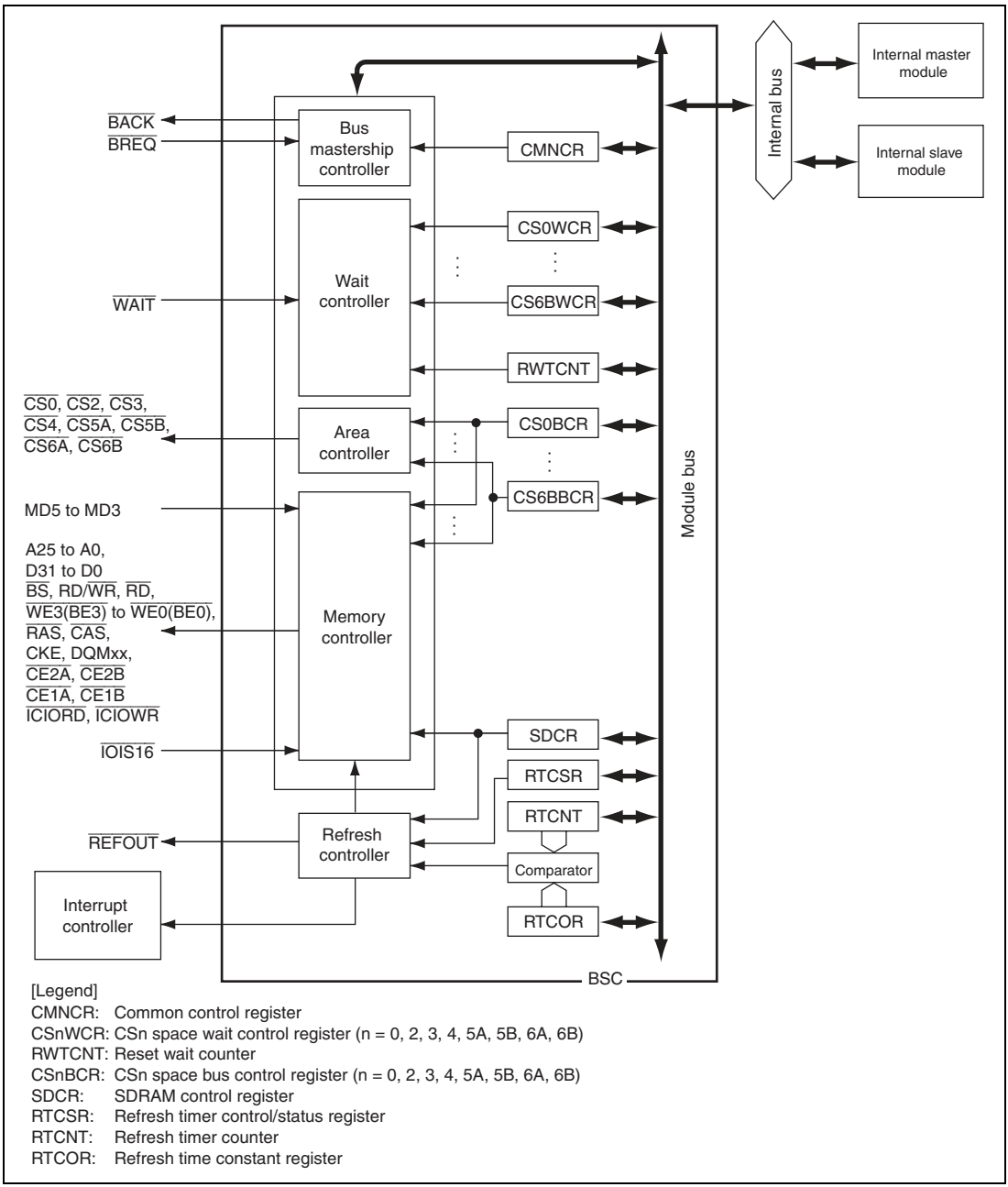


Figure 9.1 Block Diagram of BSC

9.2 Input/Output Pins

The configuration of pins in this module is shown in table 9.1.

Table 9.1 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 (clock synchronous/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 map 1 Corresponds to PCMCIA card select signals D15 to D8 when the PCMCIA is used.
$\overline{CS5B}$/$\overline{CE1A}$	O	Chip select 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 map 1 Corresponds to PCMCIA card select signals D15 to D8 when the PCMCIA is used.
$\overline{CS6B}$/$\overline{CE1B}$	O	Chip select Corresponds to PCMCIA card select signals D7 to D0 when the PCMCIA is used.
RD/$\overline{WR}$	O	Read/write signal Connects to $\overline{WE}$ pins when SDRAM or byte-selection SRAM is connected.
$\overline{RD}$	O	Read strobe (read data output enable signal) A strobe signal to indicate the memory read cycle when the PCMCIA is used.

Name	I/O	Function
WE3(BE3)/DQMUU/ 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.
WE2(BE2)/DQMUL/ 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.
WE1(BE1)/DQMLU/ 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 strobe signal when the PCMCIA is used.
WE0(BE0)/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.
RAS	O	Connects to RAS pin when SDRAM is connected.
CAS	O	Connects to CAS pin when SDRAM is connected.
CKE	O	Connects to CKE pin when SDRAM is connected.
IOIS16	I	PCMCIA 16-bit I/O signal Valid only in little endian mode. Pulled low in bit endian mode.
WAIT	I	External wait input (sampled at the falling edge of CKIO)
BREQ	I	Bus request input
BACK	O	Bus acknowledge output
MD5 to MD3	I	MD5: Selects data alignment (big endian or little endian) MD4 and MD3: Specify area 0 bus width (8/16/32 bits)
REFOUT	O	Bus mastership request signal for refreshing

9.3 Area Overview

9.3.1 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 5, Cache. The remaining 29 bits are used for division of the space into ten areas (address map 1) 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 9.2 and 9.3, this LSI can be connected directly to eight or six 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.

9.3.2 Shadow Area

The BSC decodes A28 to A25 of the physical address and generates chip select signals that correspond to areas 0, 2 to 4, 5A, 5B, 6A, and 6B. Address bits A31 to A29 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.

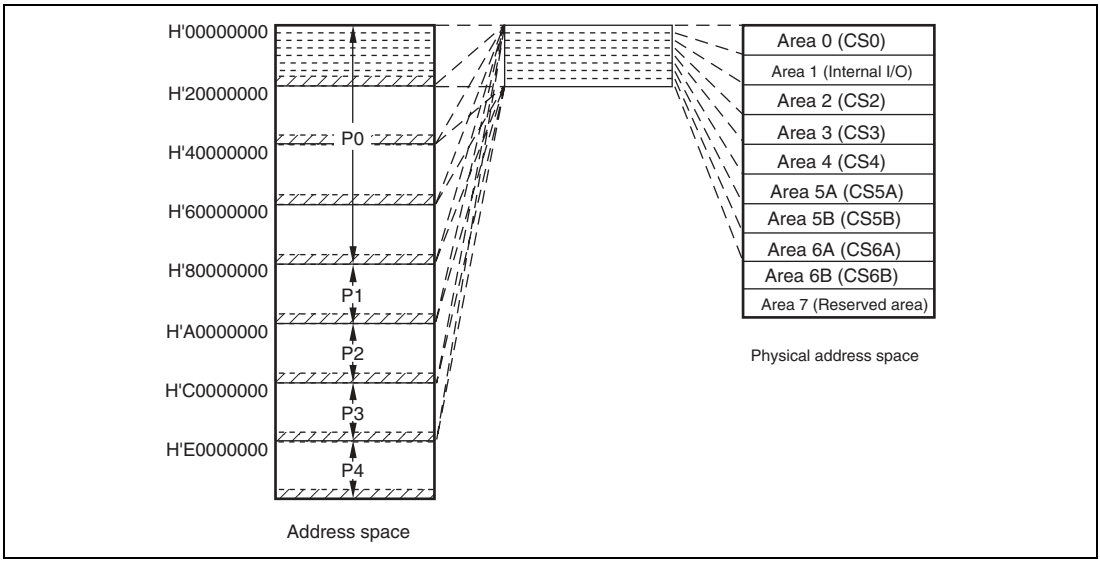


Figure 9.2 Address Space

9.3.3 Address Map

The external address space has a capacity of 384 Mbytes and is used by dividing eight partial spaces (address map 1) 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 listed below.

Table 9.2 Address Space Map 1 (CMNCR.MAP = 0)

Physical Address	Area	Memory to be Connected	Capacity
H'00000000 to H'03FFFFFF	Area 0	Normal memory Burst ROM (Asynchronous) Burst ROM (Synchronous)	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	64 Mbytes
H'0C000000 to H'0FFFFFFF	Area 3	Normal memory Byte-selection SRAM SDRAM	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.

Table 9.3 Address Space Map 2 (CMNCR.MAP = 1)

Physical Address	Area	Memory to be Connected	Capacity
H'00000000 to H'03FFFFFF	Area 0	Normal memory Burst ROM (Asynchronous) Burst ROM (Synchronous)	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	64 Mbytes
H'0C000000 to H'0FFFFFFF	Area 3	Normal memory Byte-selection SRAM SDRAM	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, CS5BBCR and CS5BWCR are valid.
For area 6, CS6BBCR and CS6BWCR are valid.
3. Set the top three bits of the address to 101 to allocate in the P2 space.

9.3.4 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 byte (8 bits), 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, MD4), and bus width is listed in the table below.

Table 9.4 Correspondence between External Pins (MD3 and MD4), Memory Type of CS0, and Memory Bus Width

MD4	MD3	Memory Type	Bus Width
0	0	Normal memory	Reserved (Setting prohibited)
	1		8 bits*
1	0		16 bits
	1		32 bits

Note: * The bus width must not be specified as eight bits if the burst ROM (clock synchronous) interface is selected.

9.3.5 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 9.5.

Table 9.5 Correspondence between External Pin (MD5) and Endians

MD5	Endian
0	Big endian
1	Little endian

9.4 Register Descriptions

The BSC has the following registers. Refer to section 37, List of Registers, for more details on the addresses and access size of these registers.

Do not access spaces other than CS0 until the termination of the setting the memory interface.

- Common control register (CMNCR)
- Bus control register for CS0 (CS0BCR)
- Bus control register for CS2 (CS2BCR)
- Bus control register for CS3 (CS3BCR)
- Bus control register for CS4 (CS4BCR)
- Bus control register for CS5A (CS5ABCR)
- Bus control register for CS5B (CS5BBCR)
- Bus control register for CS6A (CS6ABCR)
- Bus control register for CS6B (CS6BBCR)
- Wait control register for CS0 (CS0WCR)
- Wait control register for CS2 (CS2WCR)
- Wait control register for CS3 (CS3WCR)
- Wait control register for CS4 (CS4WCR)
- Wait control register for CS5A (CS5AWCR)
- Wait control register for CS5B (CS5BWCR)
- Wait control register for CS6A (CS6AWCR)
- Wait control register for CS6B (CS6BWCR)
- SDRAM control register (SDCR)
- Refresh timer control/status register (RTCSR)
- Refresh timer counter (RTCNT)
- Refresh time constant register (RTCOR)
- SDRAM mode register (SDMR2)
- SDRAM mode register (SDMR3)

9.4.1 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	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	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	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	MAP	0	R/W	Space Specification Selects the address map for the external address space. The address maps to be selected are shown in tables 9.2 and 9.3. 0: Selects address map 1 1: Selects address map 2
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	DPRTY1	0	R/W	DMA Burst Transfer Priority
9	DPRTY0	0	R/W	Specify the priority for a refresh request/bus mastership request during DMA burst transfer. 00: Accepts a refresh request and bus mastership request during DMA burst transfer 01: Accepts a refresh request but does not accept a bus mastership request during DMA burst transfer 10: Accepts neither a refresh request nor a bus mastership request during DMA burst transfer 11: Reserved (Setting prohibited)
8	DMAIW2	0	R/W	Wait States between Access Cycles when DMA Single Address is Transferred
7	DMAIW1	0	R/W	
6	DMAIW0	0	R/W	Specify the number of idle cycles to be inserted after an access to an external device with DACK when DMA single address transfer is performed. The method of inserting idle cycles depends on the contents of DMAIWA. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycled inserted 100: 6 idle cycled inserted 101: 8 idle cycle inserted 110: 10 idle cycles inserted 111: 12 idle cycled inserted
5	DMAIWA	0	R/W	Method of Inserting Wait States between Access Cycles when DMA Single Address is Transferred Specifies the method of inserting the idle cycles specified by the DMAIW1 and DMAIW0 bits. Clearing this bit will make this LSI insert the idle cycles when another device, which includes this LSI, drives the data bus after an external device with DACK drove it. Setting this bit will make this LSI insert the idle cycles even when the continuous accesses to an external device with DACK are performed. 0: Inserts the idle cycles when another device drives the data bus after an external device with DACK drove it. 1: Inserts the idle cycles every time when an external device with DACK is accessed.

Bit	Bit Name	Initial Value	R/W	Description
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, $\overline{BS}$, $\overline{CSn}$, RD/$\overline{WR}$, $\overline{WEn}$($\overline{BEn}$)/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
0	HIZCNT	0	R/W	High-Z Control Specifies the state in standby mode and bus released for CKIO, CKE, $\overline{RAS}$, and $\overline{CAS}$. 0: High impedance in standby mode and bus released for CKIO, CKE, $\overline{RAS}$, and $\overline{CAS}$. 1: Driven in standby mode and bus released for CKIO, CKE, $\overline{RAS}$, and $\overline{CAS}$.

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.

9.4.2 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	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	IWW2	0	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: No idle cycle 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
29	IWW1	1	R/W	
28	IWW0	1	R/W	

Bit	Bit Name	Initial Value	R/W	Description
27	IWRWD2	0	R/W	Idle Cycles for Another Space Read-Write
26	IWRWD1	1	R/W	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.
25	IWRWD0	1	R/W	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	IWRWS2	0	R/W	Idle Cycles for Read-Write in Same Space
23	IWRWS1	1	R/W	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.
22	IWRWS0	1	R/W	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	IWRRD2	0	R/W	Idle Cycles for Read-Read in Another Space
20	IWRRD1	1	R/W	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.
19	IWRRD0	1	R/W	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	IWRRS2	0	R/W	Idle Cycles for Read-Read in Same Space
17	IWRRS1	1	R/W	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.
16	IWRRS0	1	R/W	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	TYPE3	0	R/W	Memory Type
14	TYPE2	0	R/W	Specify the type of memory connected to a space.
13	TYPE1	0	R/W	0000: Normal space
12	TYPE0	0	R/W	0001: Burst ROM (clock asynchronous)
				0010: Reserved (setting prohibited)
				0011: Byte-selection SRAM
				0100: SDRAM
				0101: PCMCIA
				0110: Reserved (setting prohibited)
				0111: Burst ROM (clock synchronous)
				1000: Reserved (setting prohibited)
				1001: Reserved (setting prohibited)
				1010: Reserved (setting prohibited)
				1011: Reserved (setting prohibited)
				1100: Reserved (setting prohibited)
				1101: Reserved (setting prohibited)
				1110: Reserved (setting prohibited)
				1111: Reserved (setting prohibited)
				Note: Memory type for area 0 immediately after reset is normal space. The normal space, burst ROM (clock asynchronous), or burst ROM (clock synchronous) can be selected by these bits.
				For details on memory type in each area, see tables 9.2 and 9.3.
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	BSZ1	1*	R/W	Data Bus Width
9	BSZ0	1*	R/W	Specify the data bus width of spaces. 00: Reserved (setting prohibited) 01: 8-bit size 10: 16-bit size 11: 32-bit size Notes: 1. The data bus width for area 0 is specified by the external pin. The BSZ1 and BSZ0 bit settings in CS0BCR are 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.

9.4.3 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, Byte-Selection SRAM

- CS0WCR, CS6BWCR

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 $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) and RD/$\overline{\text{WR}}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal at the read/write timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal during the read/write access cycle and asserts the RD/$\overline{\text{WR}}$ signal at the write timing.
19 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 11	SW1 SW0	0 0	R/W R/W	Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) 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	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessary for read or write access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: 0 cycle 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: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Reserved (Setting prohibited)
6	WM	0	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	HW1	0	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) negation to Address, $\overline{CSn}$ negation
0	HW0	0	R/W	Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) 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	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 $\overline{WEn}$ ($\overline{BEn}$) and RD/$\overline{WR}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{WEn}$ ($\overline{BEn}$) signal at the read/write timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ ($\overline{BEn}$) signal during the read/write access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19 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	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessary for read or write access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: 0 cycle
				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: Reserved (setting prohibited)
				1110: Reserved (setting prohibited)
				1111: Reserved (setting prohibited)
6	WM	0	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 0	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0.

- CS4WCR

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 $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) and RD/$\overline{\text{WR}}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal at the read/write timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal during the read/write access cycle and asserts the RD/$\overline{\text{WR}}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18	WW2	0	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR3 to WR0 setting (read or write access wait) 001: 0 cycles 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
17	WW1	0	R/W	
16	WW0	0	R/W	
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	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to
11	SW0	0	R/W	$\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) Assertion- Specify the number of delay cycles from address and $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessary for
8	WR1	1	R/W	read or write access.
7	WR0	0	R/W	0000: 0 cycle 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: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	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	HW1 HW0	0 0	R/W R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) negation to Address, $\overline{CSn}$ negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) negation to address and $\overline{CSn}$ negation. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS5AWCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18 17 16	WW2 WW1 WW0	0 0 0	R/W R/W R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR3 to WR0 setting (read or write access wait) 001: 0 cycles 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	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to
11	SW0	0	R/W	$\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) Assertion Specify the number of delay cycles from address and $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessary for
8	WR1	1	R/W	read or write access.
7	WR0	0	R/W	0000: 0 cycle 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: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	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

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	HW1	0	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) negation to Address, $\overline{CSn}$ negation
0	HW0	0	R/W	Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) 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	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 $\overline{WEn}$ ($\overline{BEn}$) and RD/$\overline{WR}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{WEn}$ ($\overline{BEn}$) signal at the read/write timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ ($\overline{BEn}$) signal during the read/write access cycle and asserts the RD/$\overline{WR}$ 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	WW2	0	R/W	Number of Write Access Wait Cycles
17	WW1	0	R/W	Specify the number of cycles that are necessary for write access.
16	WW0	0	R/W	000: The same cycles as WR3 to WR0 setting (read or write access wait) 001: 0 cycles 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	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) Assertion
11	SW0	0	R/W	Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) 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	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessary for read or write access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: 0 cycle 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: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	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	HW1	0	R/W	Number of Delay Cycles from RD, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) negation to Address, $\overline{\text{CSn}}$ negation
0	HW0	0	R/W	Specify the number of delay cycles from $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) negation to address and $\overline{\text{CSn}}$ negation. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS6AWCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to
11	SW0	0	R/W	$\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) Assertion Specify the number of delay cycles from address and $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessary for
8	WR1	1	R/W	read or write access.
7	WR0	0	R/W	0000: 0 cycle 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: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	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

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	HW1	0	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) negation to Address, $\overline{CSn}$ negation
0	HW0	0	R/W	Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) 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 (Clock Asynchronous)

- CS0WCR

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	BEN	0	R/W	Burst Enable Specification Enables or disables 8-burst access for a 16-bit bus width or 16-burst access for an 8-bit bus width during 16-byte access. If this bit is set to 1, 2-burst access is performed four times when the bus width is 16 bits and 4-burst access is performed four times when the bus width is 8 bits. To use a device that does not support 8-burst access or 16-burst access, set this bit to 1. 0: Enables 8-burst access for a 16-bit bus width and 16-burst access for an 8-bit bus width. 1: Disables 8-burst access for a 16-bit bus width and 16-burst access for an 8-bit bus width.
19, 18	—	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
17	BW1	0	R/W	Number of Burst Wait Cycles
16	BW0	0	R/W	Specify the number of wait cycles to be inserted between the second or later access cycles in burst access. 00: 0 cycles 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	W3	1	R/W	Number of Access Wait Cycles
9	W2	0	R/W	Specify the number of wait cycles to be inserted in the first access cycle.
8	W1	1	R/W	0000: 0 cycles
7	W0	0	R/W	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: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
6	WM	0	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 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

- CS4WCR

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	BEN	0	R/W	Burst Enable Specification Enables or disables 8-burst access for a 16-bit bus width or 16- burst access for an 8-bit bus width during 16-byte access. If this bit is set to 1, 2-burst access is performed four times when the bus width is 16 bits and 4-burst access is performed four times when the bus width is 8 bits. To use a device that does not support 8-burst access or 16-burst access, set this bit to 1. 0: Enables 8-burst access for a 16-bit bus width and 16-burst access for an 8-bit bus width. 1: Disables 8-burst access for a 16-bit bus width and 16-burst access for an 8-bit bus width.
19, 18	—	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
17	BW1	0	R/W	Number of Burst Wait Cycles
16	BW0	0	R/W	Specify the number of wait cycles to be inserted between the second or later access cycles in burst access. 00: 0 cycles 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	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) assertion. These bits can be specified only in area 4. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
11	SW0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
10	W3	1	R/W	Number of Access Wait Cycles
9	W2	0	R/W	Specify the number of wait cycles to be inserted in the first access cycle.
8	W1	1	R/W	
7	W0	0	R/W	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: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	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	HW1	0	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) negation to Address, $\overline{CSn}$ negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) negation to address and $\overline{CSn}$ negation. These bits can be specified only in area 4. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
0	HW0	0	R/W	

(3) SDRAM

- CS2WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	A2CL1	1	R/W	CAS Latency for Area 2
7	A2CL0	0	R/W	Specify the CAS latency for area 2. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

- CS3WCR

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	TRP1	0	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
13	TRP0	0	R/W	
12	—	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
11	TRCD1	0	R/W	Number of Cycles from ACTV Command to READ(A)/WRIT(A) Command
10	TRCD0	1	R/W	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
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	A3CL1	1	R/W	CAS Latency for Area 3.
7	A3CL0	0	R/W	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 9.19.)
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TRWL1	0	R/W	Number of Cycles from WRITA/WRIT Command to Auto-Precharge/PRE Command
3	TRWL0	0	R/W	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	TRC1	0	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
0	TRC0	0	R/W	

Note: * If both areas 2 and 3 are specified as SDRAM, TRP1/0, TRCD0/1, TRWL1/0, and TRC1/0 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	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	SA1	0	R/W	Space Attribute Specification
20	SA0	0	R/W	Specify memory card interface or I/O card interface when the PCMCIA interface is selected. SA1 0: Specifies memory card interface when A25 = 1 1: Specifies I/O card interface when A25 = 1 SA0 0: Specifies memory card interface when A25 = 0 1: Specifies I/O card interface when A25 = 0 Note: When using the PC card controller, specifies the following settings. When the bit 4 (P0USE) in the PCC0GCR register of PCC is 1 and the bit 5 (P0PCCT) of the PCC0GCR register is 0, both SA1 and SA0 should be 0. When the bit 4 (P0USE) and the bit 5 (P0PCCT) in the PCC0GCR register of PCC are 1, both SA1 and SA0 should be 1.
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	TED3	0	R/W	Delay from Address to $\overline{RD}$ or $\overline{WE}$ Assert
13	TED2	0	R/W	Specify the delay time from address output to $\overline{RD}$ or $\overline{WE}$ assert in PCMCIA interface.
12	TED1	0	R/W	
11	TED0	0	R/W	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

Bit	Bit Name	Initial Value	R/W	Description
10	PCW3	1	R/W	Number of Access Wait Cycles
9	PCW2	0	R/W	Specify the number of wait cycles to be inserted.
8	PCW1	1	R/W	0000: 3 cycles
7	PCW0	0	R/W	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
6	WM	0	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.

Bit	Bit Name	Initial Value	R/W	Description
3	TEH3	0	R/W	Delay from $\overline{RD}$ or $\overline{WE}$ Negate to Address
2	TEH2	0	R/W	Specify the address hold time from $\overline{RD}$ or $\overline{WE}$ negate in the PCMCIA interface.
1	TEH1	0	R/W	
0	TEH0	0	R/W	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

(5) Burst ROM (Clock Synchronous)

- CS0WCR

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	BW1	0	R/W	Number of Burst Wait Cycles
16	BW0	0	R/W	Specify the number of wait cycles to be inserted between the second or later access cycles in burst access. 00: 0 cycles 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	W3	1	R/W	Number of Access Wait Cycles
9	W2	0	R/W	Specify the number of wait cycles to be inserted in the first access cycle.
8	W1	1	R/W	
7	W0	0	R/W	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: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
6	WM	0	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 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

9.4.4 SDRAM Control Register (SDCR)

SDCR specifies the method to refresh and access SDRAM, and the types of SDRAMs to be connected.

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	A2ROW1	0	R/W	Number of Bits of Row Address for Area 2
19	A2ROW0	0	R/W	Specify the number of bits of row address for area 2. 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
18	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
17	A2COL1	0	R/W	Number of Bits of Column Address for Area 2
16	A2COL0	0	R/W	Specify the number of bits of column address for area 2. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	DEEP	0	R/W	Deep Power-Down Mode This bit is valid for low-power SDRAM. If the RMODE bit is set to 1 while this bit is set to 1, the deep power-down entry command is issued and the low-power SDRAM enters the deep power-down mode. 0: Self-refresh mode 1: Deep power-down mode

Bit	Bit Name	Initial Value	R/W	Description
12	—	0	R	Reserved This bit is 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.

Bit	Bit Name	Initial Value	R/W	Description
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.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	A3ROW1	0	R/W	Number of Bits of Row Address for Area 3
3	A3ROW0	0	R/W	Specify the number of bits of the row address for area 3. 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	A3COL1	0	R/W	Number of Bits of Column Address for Area 3
0	A3COL0	0	R/W	Specify the number of bits of the column address for area 3. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)

9.4.5 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	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	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables a CMF interrupt request when the CMF bit of RTCSR is set to 1. 0: Disables the CMF interrupt request 1: Enables the CMF interrupt request
5	CKS2	0	R/W	Clock Select
4	CKS1	0	R/W	Select the clock input to count-up the refresh timer counter (RTCNT).
3	CKS0	0	R/W	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

Bit	Bit Name	Initial Value	R/W	Description
2	RRC2	0	R/W	Refresh Count
1	RRC1	0	R/W	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.
0	RRC0	0	R/W	000: Once 001: Twice 010: 4 times 011: 6 times 100: 8 times 101: Reserved (setting prohibited) 110: Reserved (setting prohibited) 111: Reserved (setting prohibited)

9.4.6 Refresh Timer Counter (RTCNT)

RTCNT is an 8-bit counter that increments using the clock selected by bits CKS2 to CKS0 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	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	—	All 0	R/W	8-bit Counter

9.4.7 Refresh Time Constant Register (RTCOR)

RTCOR is an 8-bit register. When 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.

If the CMIE bit of the RTCSR is set to 1, an interrupt is requested by this matching signal. This request is maintained until the CMF bit in RTCSR is cleared to 0. Clearing the CMF bit in RTCSR affects only interrupts and does not affect refresh requests. This makes it possible to count the number of refresh requests during refresh by interrupts, and to specify the refresh and interval timer interrupts simultaneously. When the RTCOR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

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	—	All 0	R/W	8-bit Counter

9.4.8 SDRAM Mode Registers 2, 3 (SDMR2 and SRMR3)

For the settings of SDRAM mode registers (SDMR2 and SDMR3), see table 9.19.

9.5 Operation

9.5.1 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 9.6 to 9.11 show the relationship between endian, device data width, and access unit.

Table 9.6 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	$\overline{\text{WE3(BE3)}}$, DQMUU	$\overline{\text{WE2(BE2)}}$, DQMUL	$\overline{\text{WE1(BE1)}}$, DQMLU	$\overline{\text{WE0(BE0)}}$, 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 9.7 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**	**$\overline{\text{WE3(BE3)}}$, DQMUU**	**$\overline{\text{WE2(BE2)}}$, DQMUL**	**$\overline{\text{WE1(BE1)}}$, DQMLU**	**$\overline{\text{WE0(BE0)}}$, 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 9.8 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**	**$\overline{\text{WE3}}$(BE3), DQMUU**	**$\overline{\text{WE2}}$(BE2), DQMUL**	**$\overline{\text{WE1}}$(BE1), DQMLU**	**$\overline{\text{WE0}}$(BE0), 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 9.9 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**	**$\overline{\text{WE3}}$($\overline{\text{BE3}}$), DQMUU**	**$\overline{\text{WE2}}$($\overline{\text{BE2}}$), DQMUL**	**$\overline{\text{WE1}}$($\overline{\text{BE1}}$), DQMLU**	**$\overline{\text{WE0}}$($\overline{\text{BE0}}$), 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 9.10 16-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	$\overline{\text{WE3(BE3)}}$, DQMUU	$\overline{\text{WE2(BE2)}}$, DQMUL	$\overline{\text{WE1(BE1)}}$, DQMLU	$\overline{\text{WE0(BE0)}}$, 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 9.11 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**	**$\overline{\text{WE3}}$(BE3), DQMUU**	**$\overline{\text{WE2}}$(BE2), DQMUL**	**$\overline{\text{WE1}}$(BE1), DQMLU**	**$\overline{\text{WE0}}$(BE0), 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

9.5.2 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 directly connected. When using SRAM with a byte-selection pin, see section 9.5.7, Byte-Selection SRAM Interface. Figure 9.3 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.

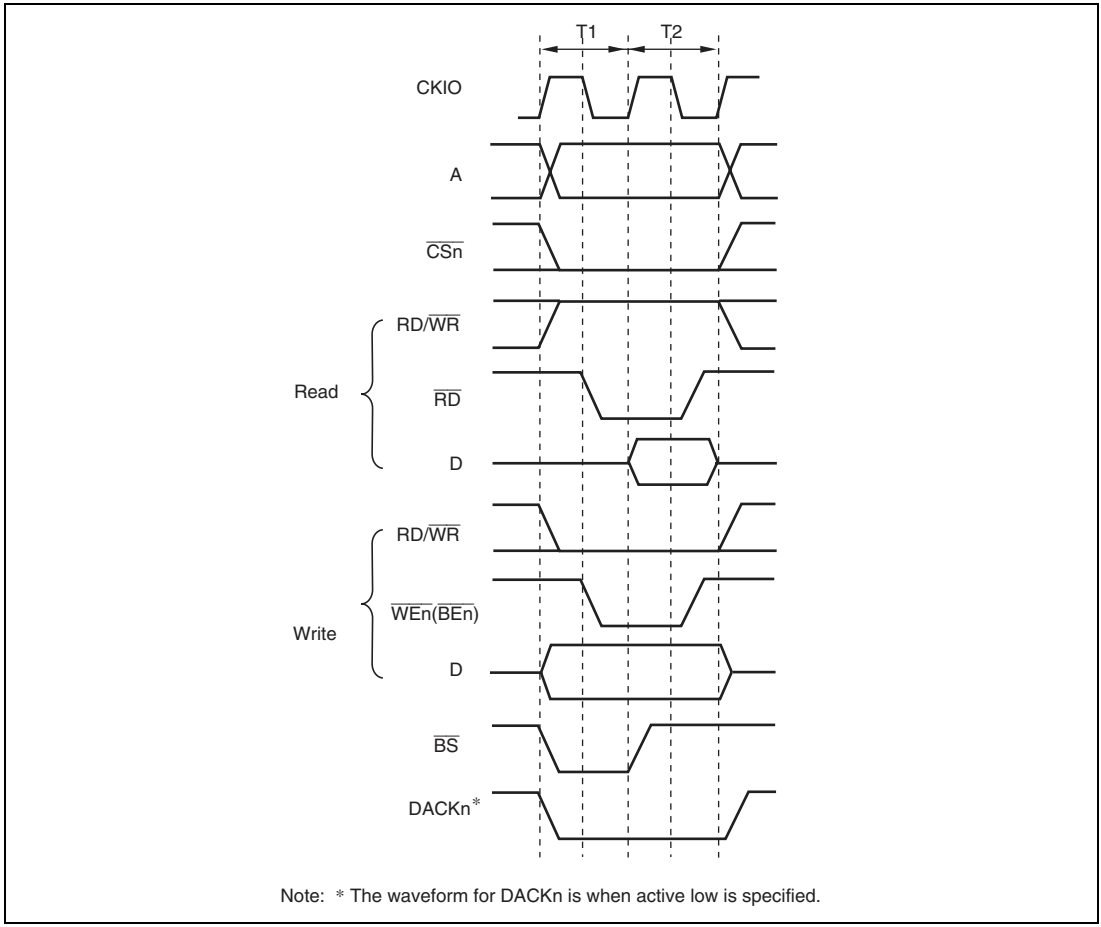


Figure 9.3 Normal Space Basic Access Timing (Access Wait 0)

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 $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) 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 RD/$\overline{\text{WR}}$ 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 9.4 and 9.5 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 9.4). If the WM bit of the CSnWCR is set to 1, external waits are ignored and no Tnop cycle is inserted (figure 9.5).

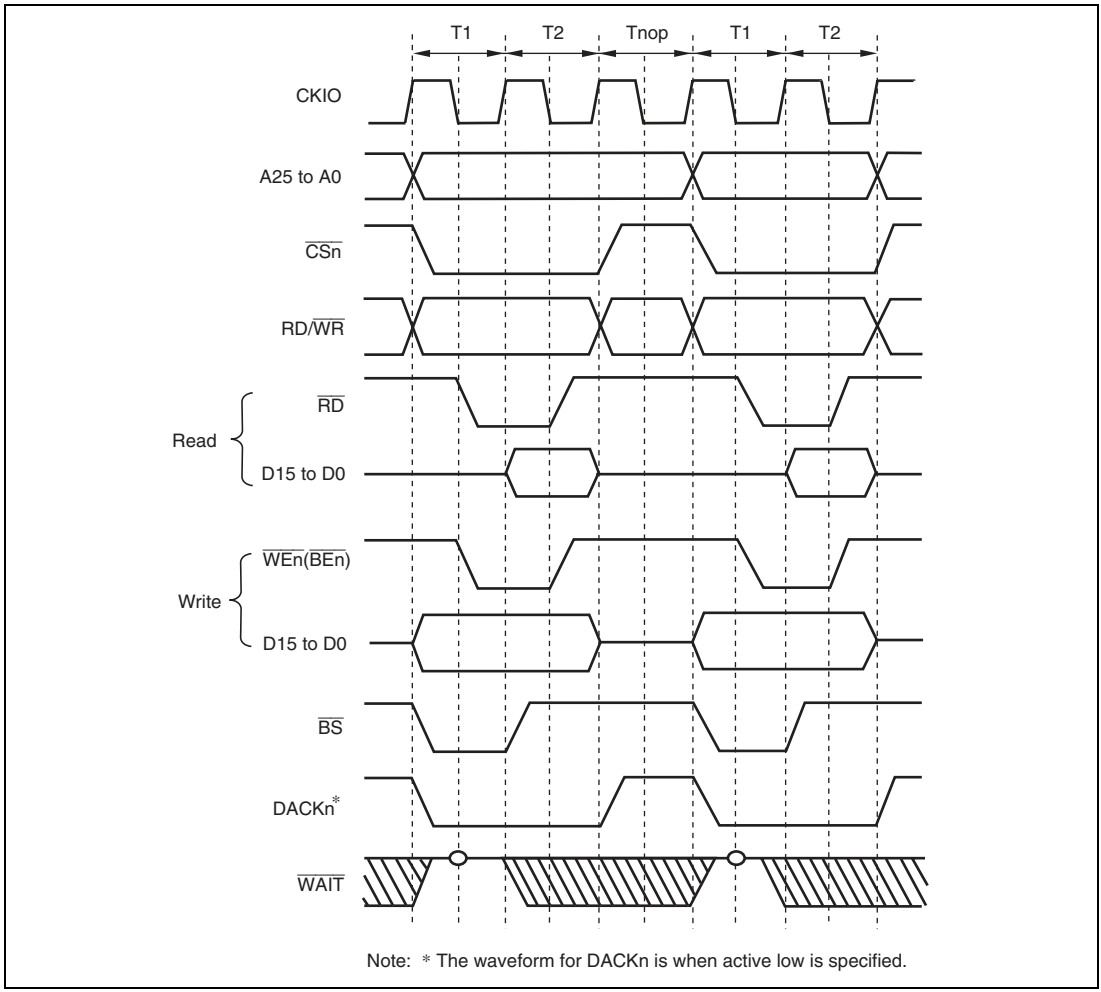


Figure 9.4 Continuous Access for Normal Space 1, Bus Width = 16 bits, Longword Access, CSnWCR.WM Bit = 0 (Access Wait = 0, Cycle Wait = 0)

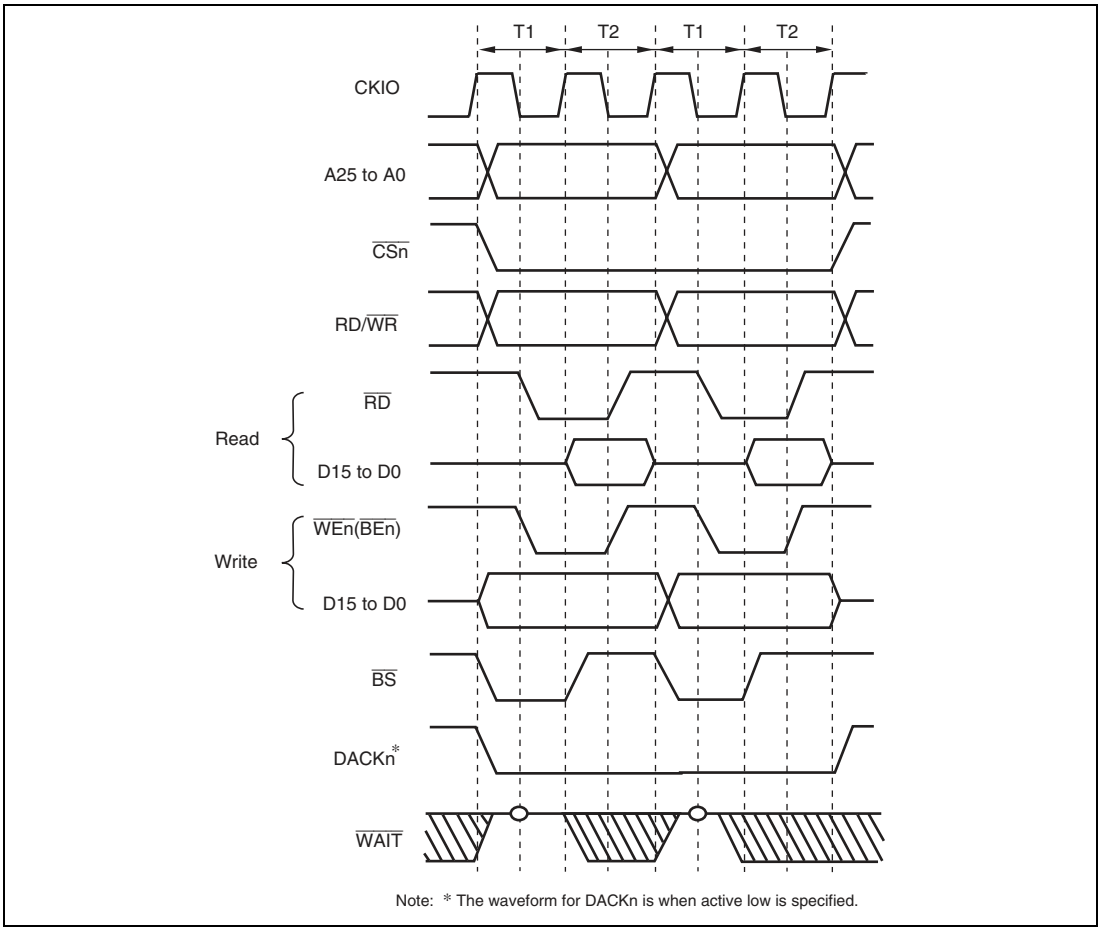


Figure 9.5 Continuous Access for Normal Space 2, Bus Width = 16 bits, Longword Access, CSnWCR.WM Bit = 1 (Access Wait = 0, Cycle Wait = 0)

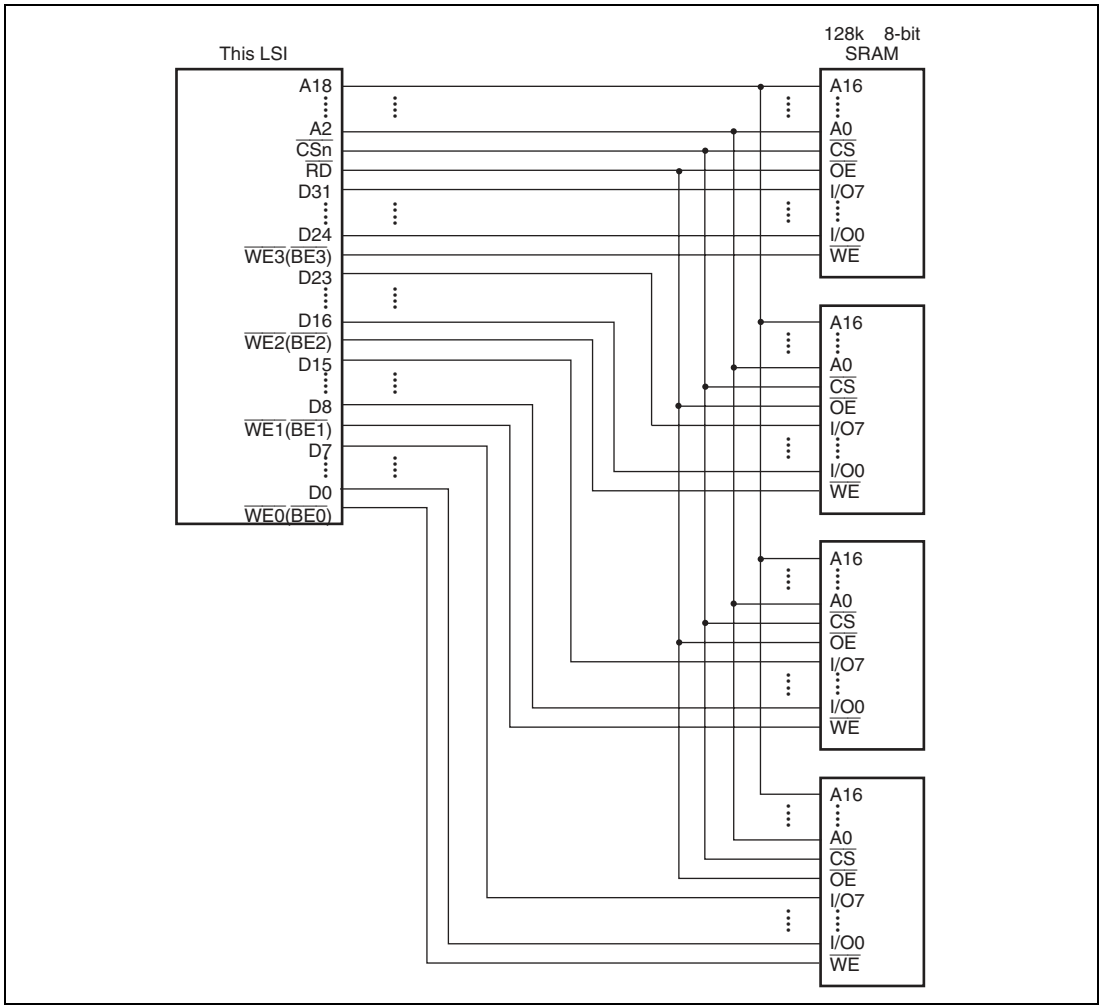


Figure 9.6 Example of 32-Bit Data-Width SRAM Connection

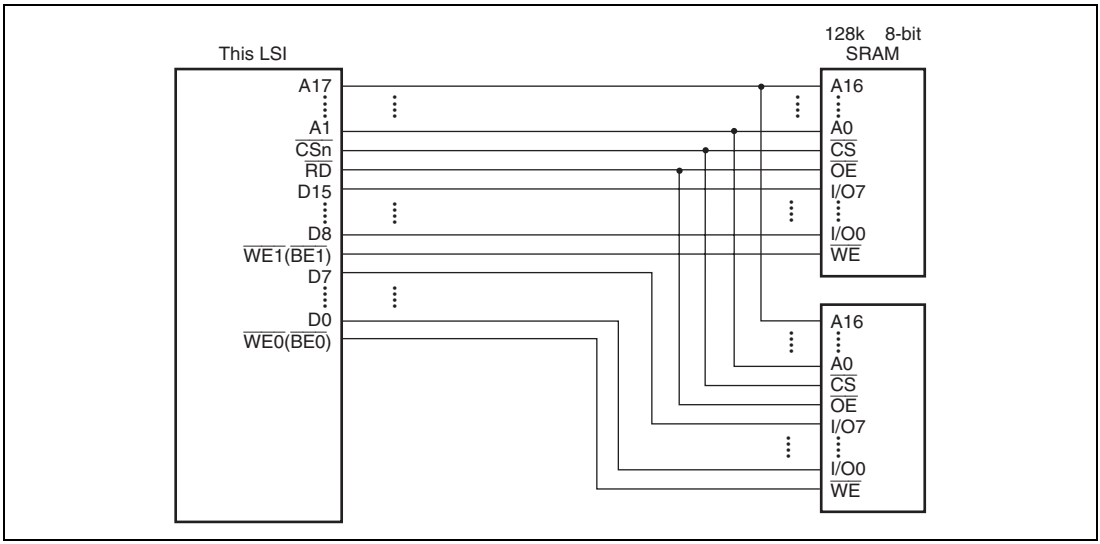


Figure 9.7 Example of 16-Bit Data-Width SRAM Connection

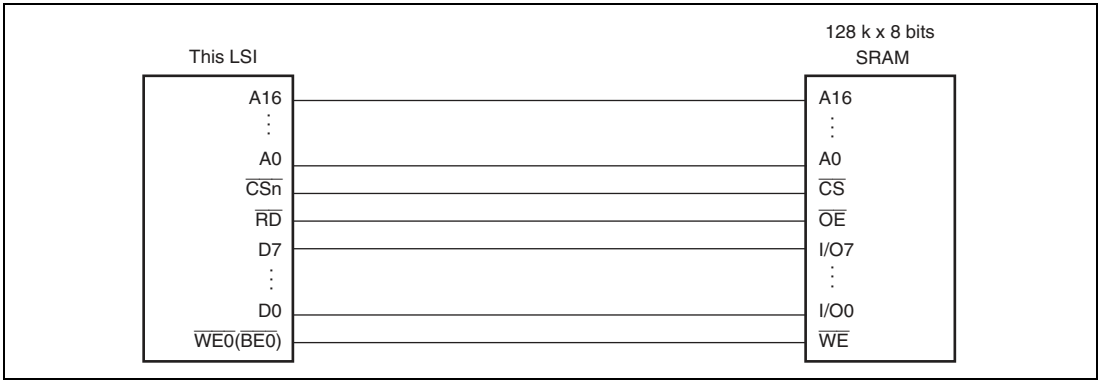


Figure 9.8 Example of 8-Bit Data-Width SRAM Connection

9.5.3 Access Wait Control

Wait cycle insertion on a normal space access can be controlled by the settings of bits WR3 to WR0 in CSnWCR. It is possible for areas 4, 5A, and 5B to insert wait cycles independently in read access and in write access. The areas other than 4, 5A, and 5B have common access wait for read cycle and write cycle. The specified number of Tw cycles is inserted as wait cycles in a normal space access shown in figure 9.9.

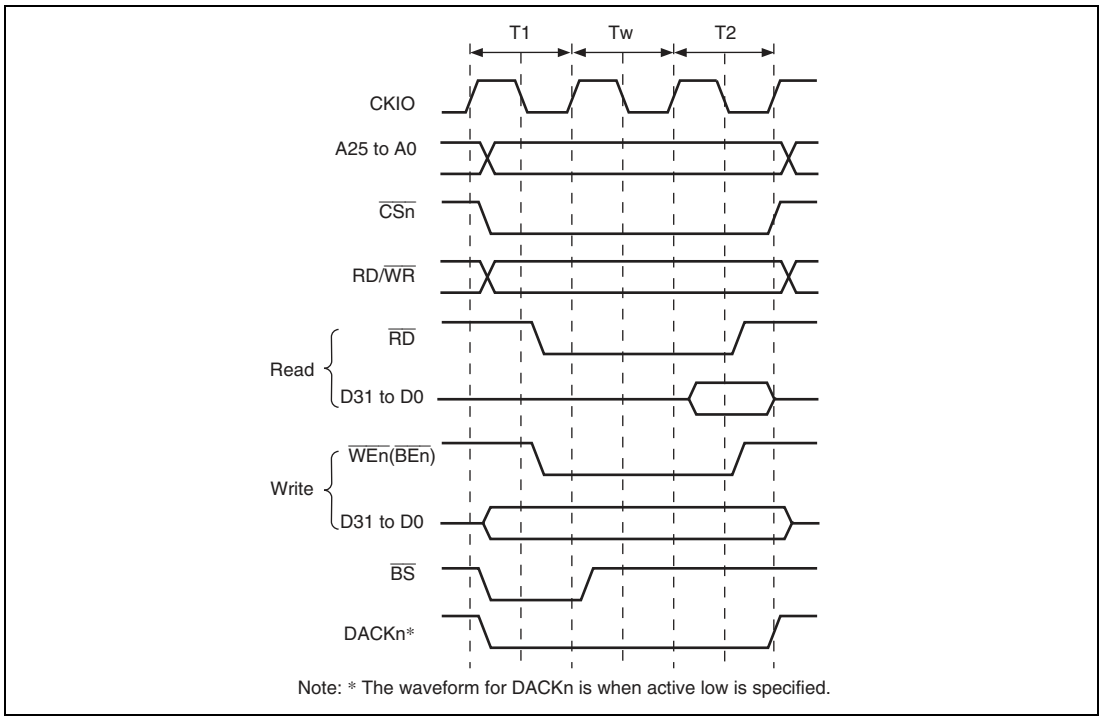


Figure 9.9 Wait Timing for Normal Space Access (Software Wait Only)

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 9.10. A 2-cycle wait is specified as a software wait. The $\overline{\text{WAIT}}$ signal is sampled on the falling edge of CKIO at the transition from the T1 or Tw cycle to the T2 cycle.

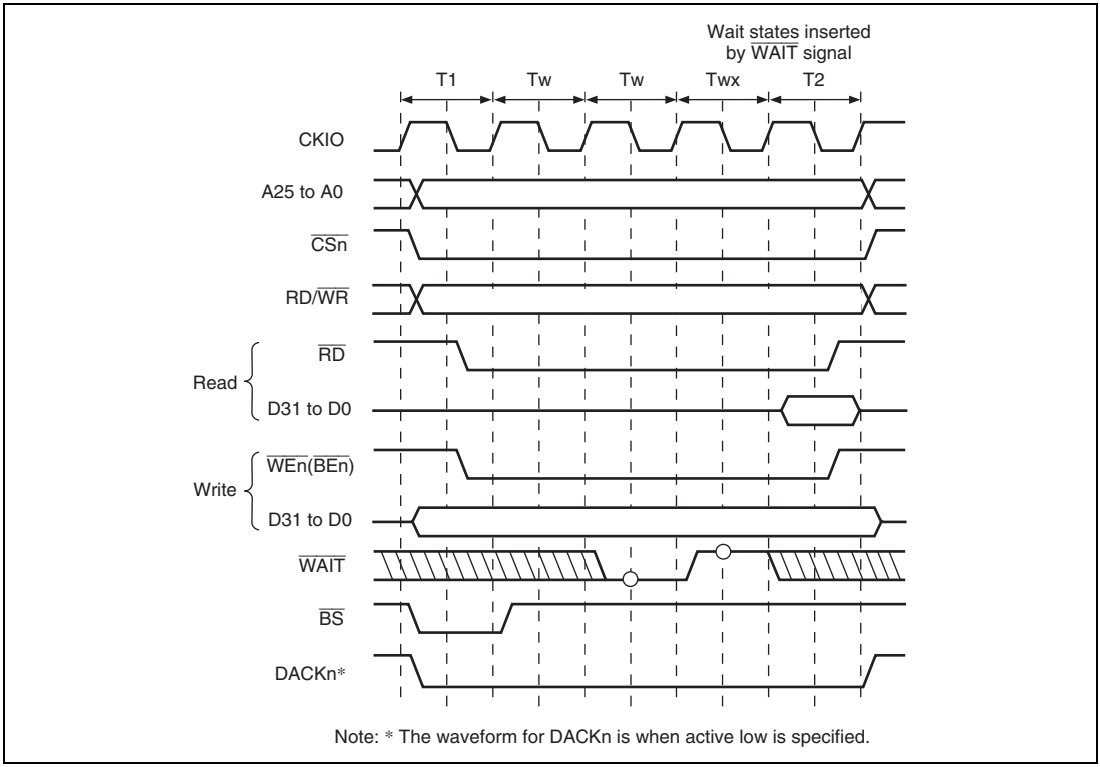


Figure 9.10 Wait State Timing for Normal Space Access (Wait State Insertion using $\overline{\text{WAIT}}$ Signal)

9.5.4 $\overline{\text{CSn}}$ Assert Period Expansion

The number of cycles from $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) assertion can be specified by setting bits SW1 and SW0 in CSnWCR. The number of cycles from $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) negation to $\overline{\text{CSn}}$ negation can be specified by setting bits HW1 and HW0. Therefore, a flexible interface to an external device can be obtained. Figure 9.11 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 $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) 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.

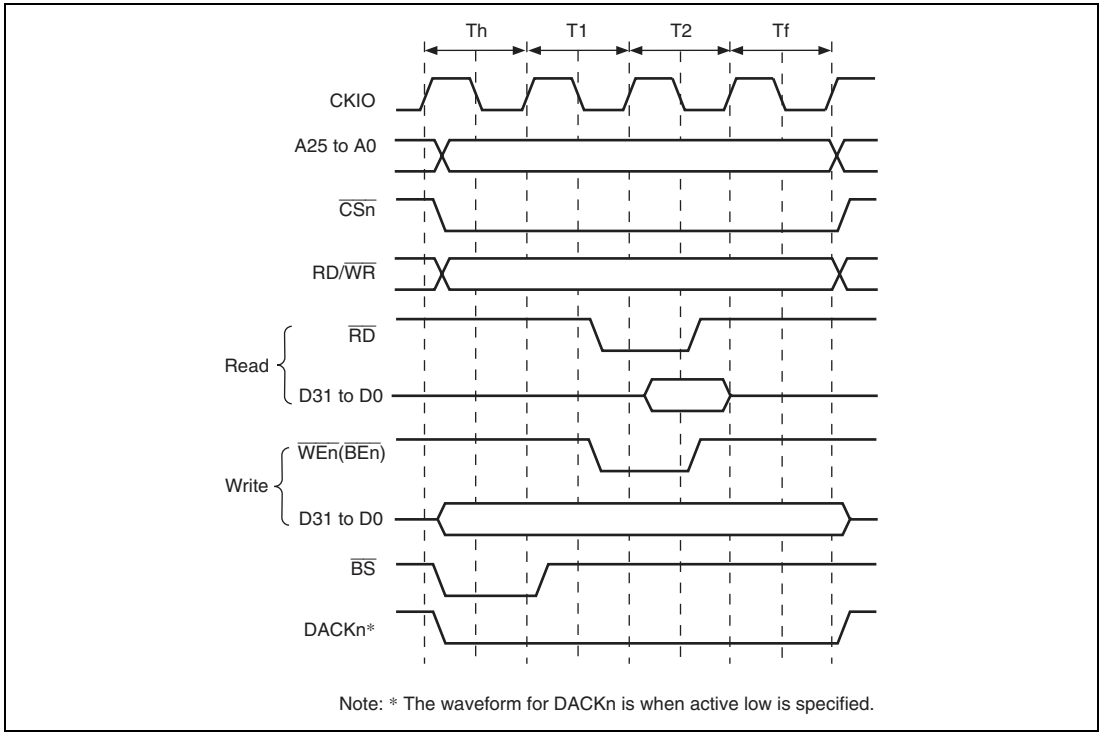


Figure 9.11 $\overline{\text{CSn}}$ Assert Period Expansion

9.5.5 SDRAM Interface

(1) SDRAM Direct 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 direct connection of SDRAM are $\overline{RAS}$, $\overline{CAS}$, RD/$\overline{WR}$, 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 $\overline{RAS}$, $\overline{CAS}$, RD/$\overline{WR}$, 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 9.5.1, Endian/Access Size and Data Alignment.

Figures 9.12 and 9.13 show examples of the connection of the SDRAM with the LSI.

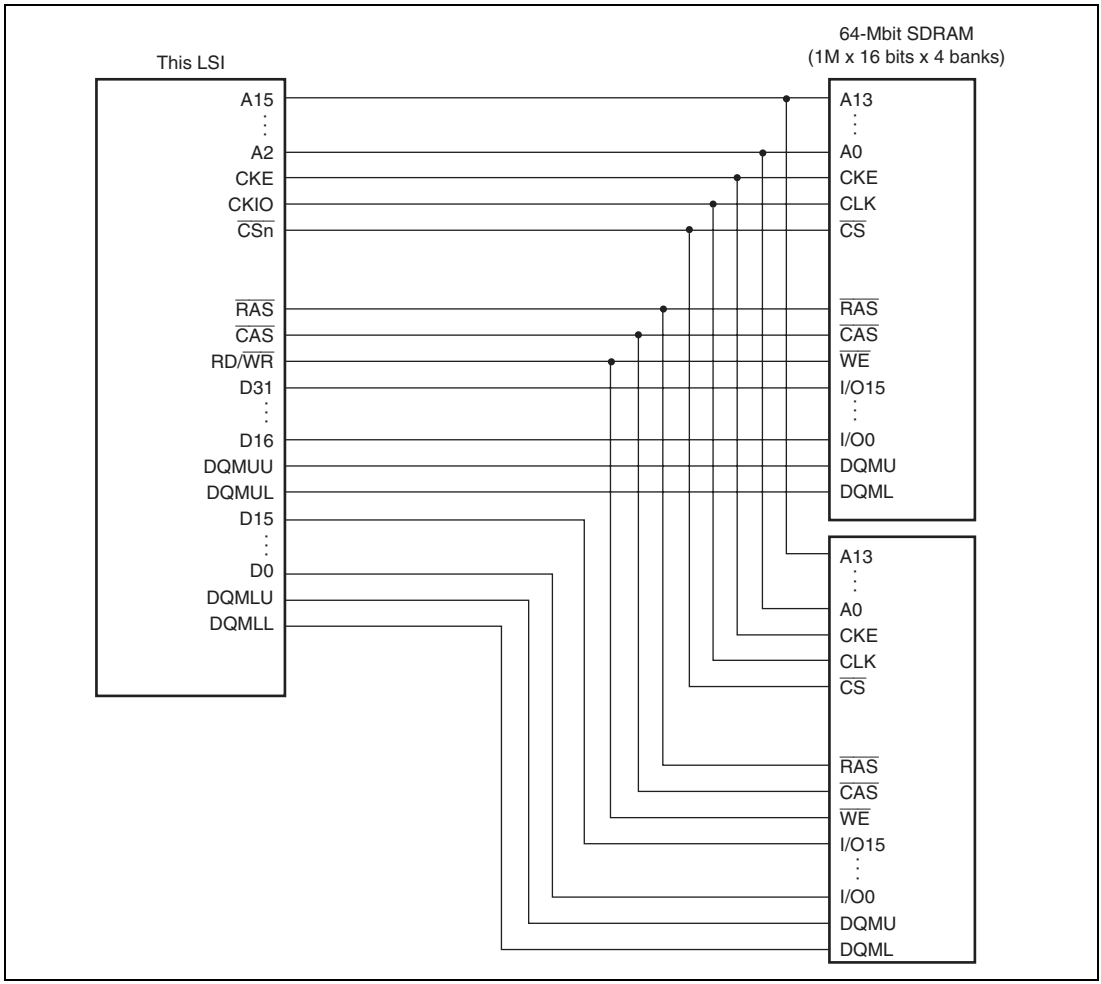


Figure 9.12 Example of 32-Bit Data-Width SDRAM Connection

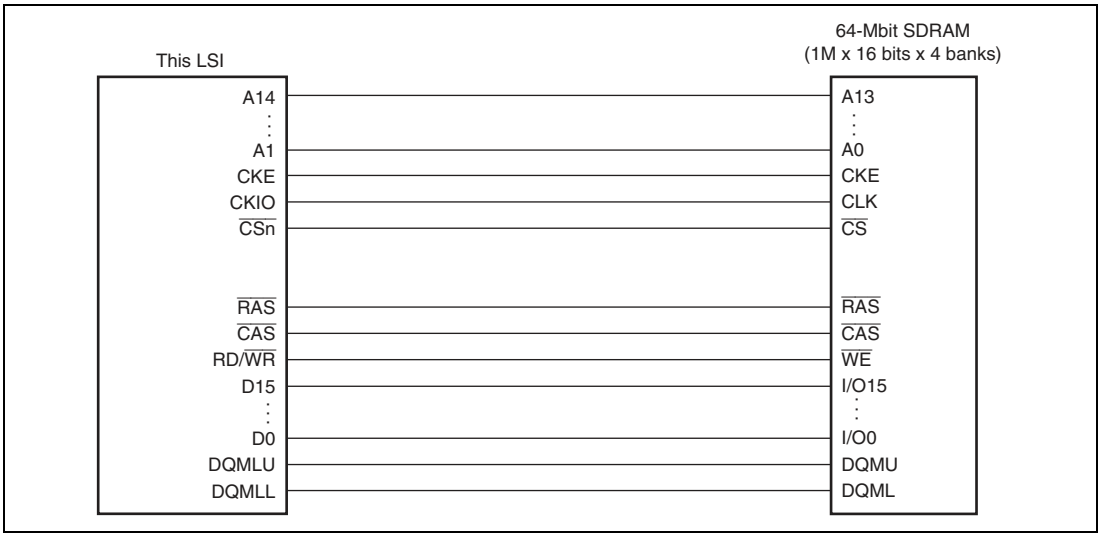


Figure 9.13 Example of 16-Bit Data-Width SDRAM Connection

(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 9.12 to 9.17 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 9.12 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 9.12 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 9.13 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 9.13 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 9.14 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 9.15 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 9.15 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 9.16 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 9.16 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 9.17 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 9.17 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

(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. 16-byte transfer in cache miss.
3. 16-byte transfer in DMAC or USDH(access to non-cacheable area)
4. 16- to 128-byte transfer by LCDC*

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 9.18 shows the relationship between the access size and the number of bursts.

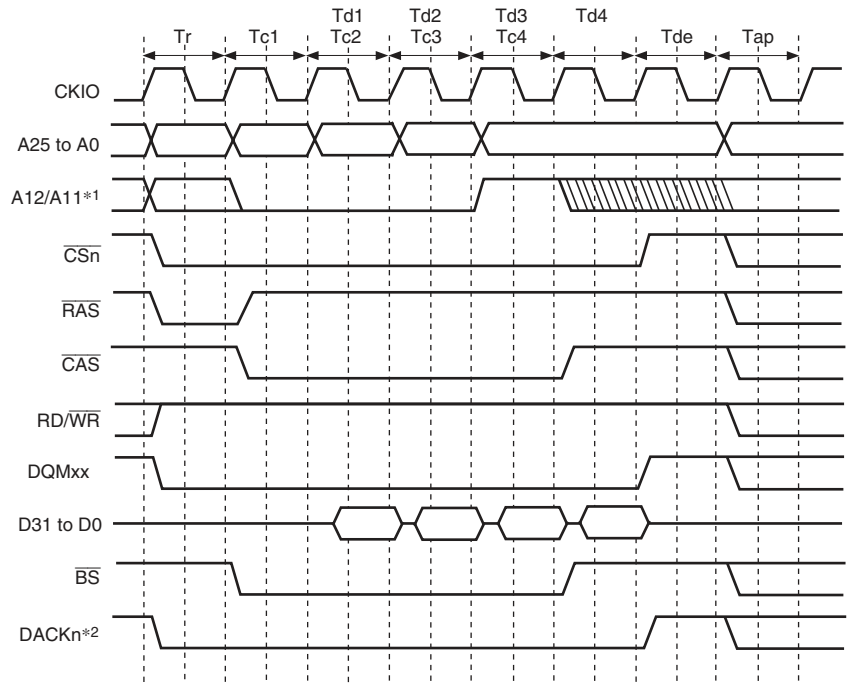
Note: * For details, see section 26, LCD Controller (LCDC).

Table 9.18 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
	16 bytes	8
	128 bytes	64
32 bits	8 bits	1
	16 bits	1
	32 bits	1
	16 bytes	4
	128 bytes	32

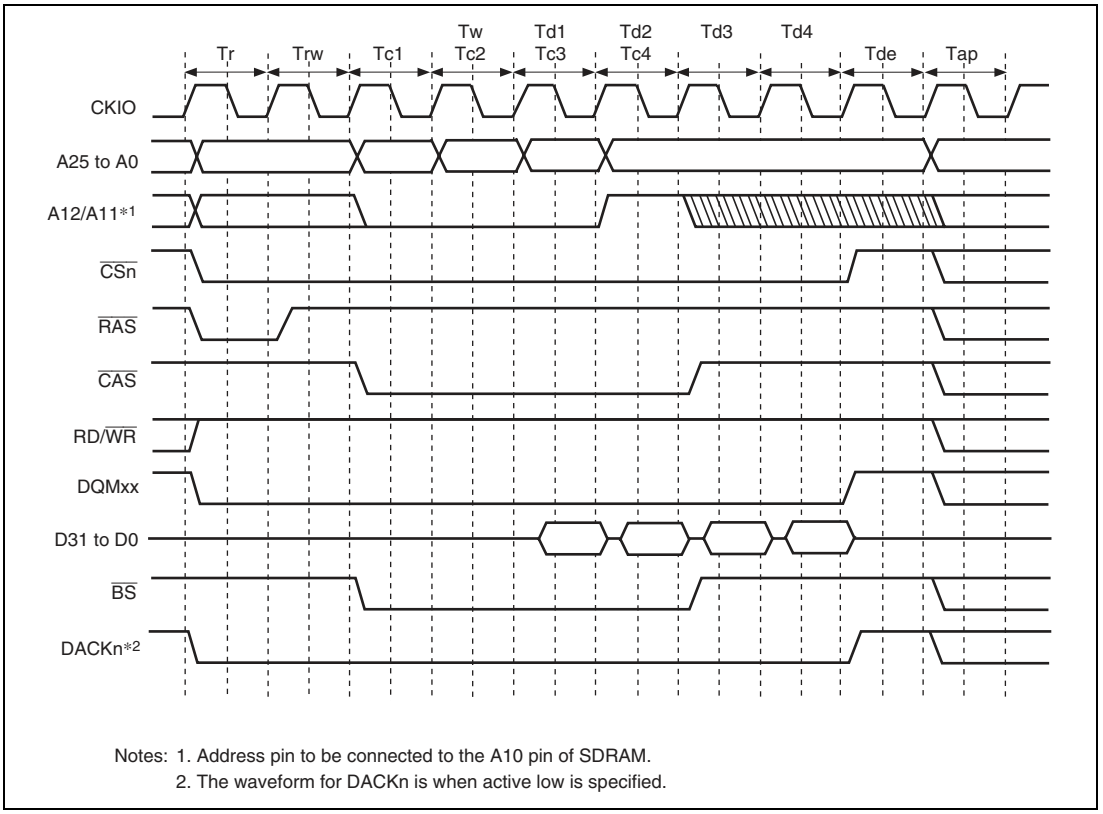
Figures 9.14 and 9.15 show a timing chart in burst read. In burst read, an ACTV command is output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, the READA command is issued in the Tc4 cycle, and the read data is received at the rising edge of the external clock (CKIO) in 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 9.15 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 READA 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 READA 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 9.14 Burst Read Basic Timing (Auto-Precharge)



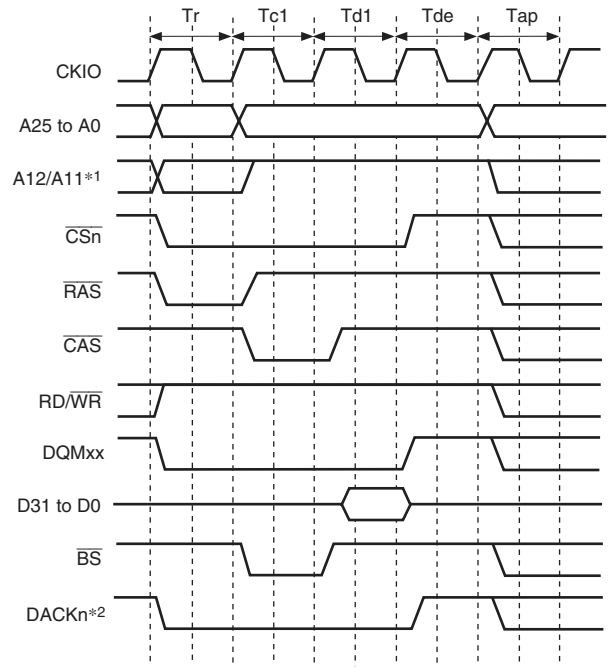
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.15 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 9.16 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 9.16 Basic Timing for Single Read (Auto-Precharge)

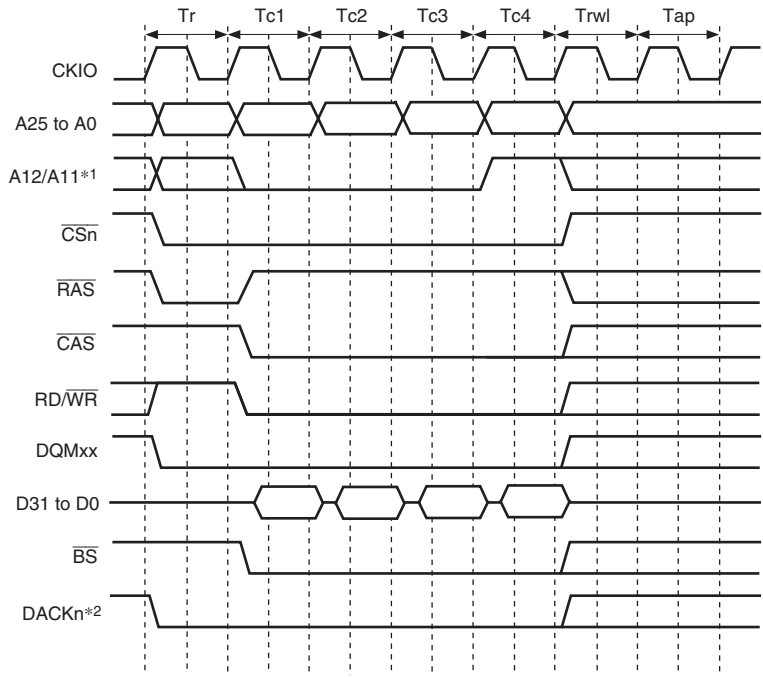
(5) Burst Write

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 9.18.

Figure 9.17 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.



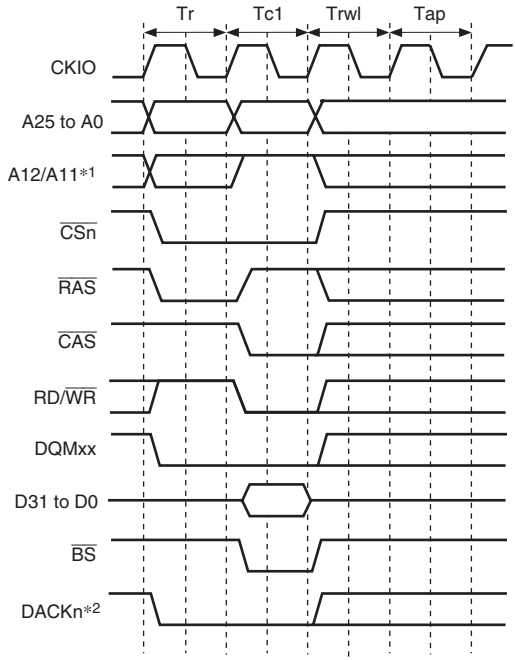
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.17 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 9.18 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 9.18 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 9.19, a burst read cycle for the same row address in figure 9.20, and a burst read cycle for different row addresses in figure 9.21. Similarly, a single write cycle without auto-precharge is shown in figure 9.22, a single write cycle for the same row address in figure 9.23, and a single write cycle for different row addresses in figure 9.24.

In figure 9.20, 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 9.19 or 9.22, followed by repetition of the cycle in figure 9.20 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 9.21 or 9.24 is executed instead of that in figure 9.20 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.

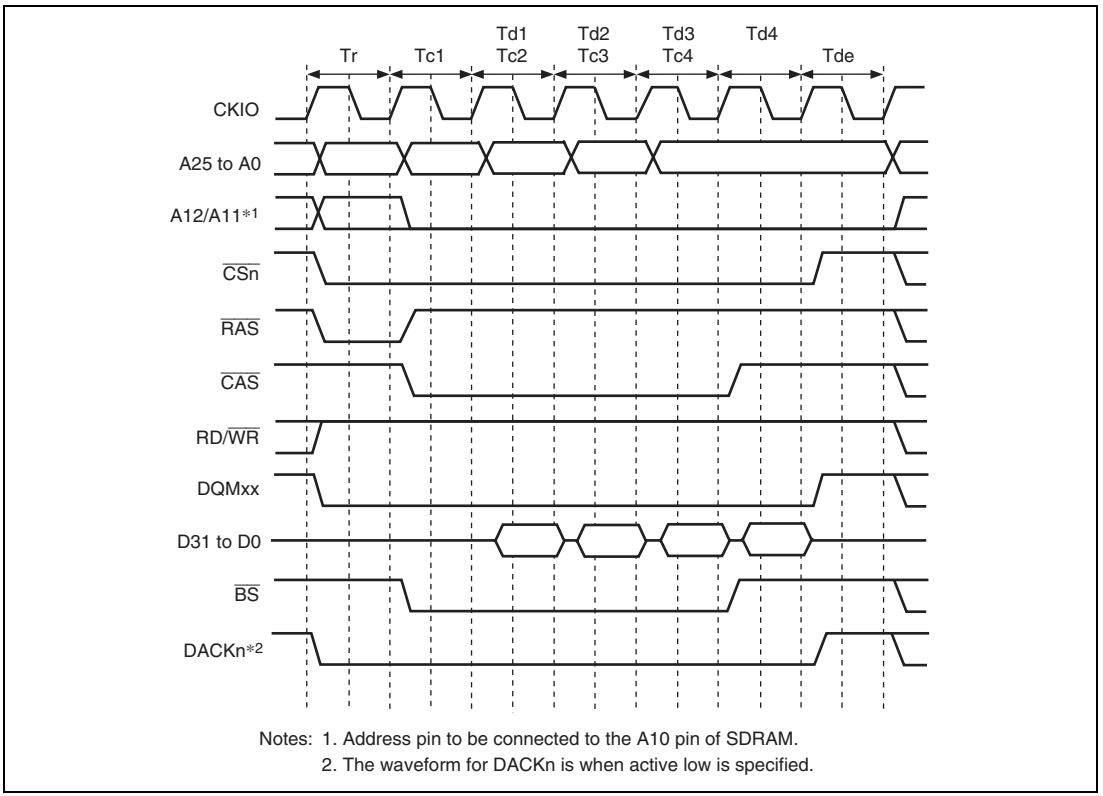
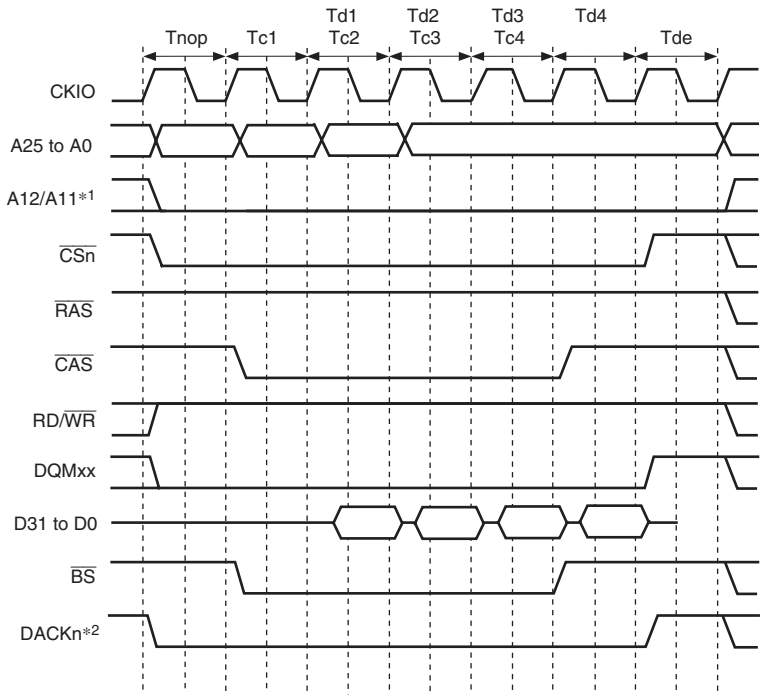
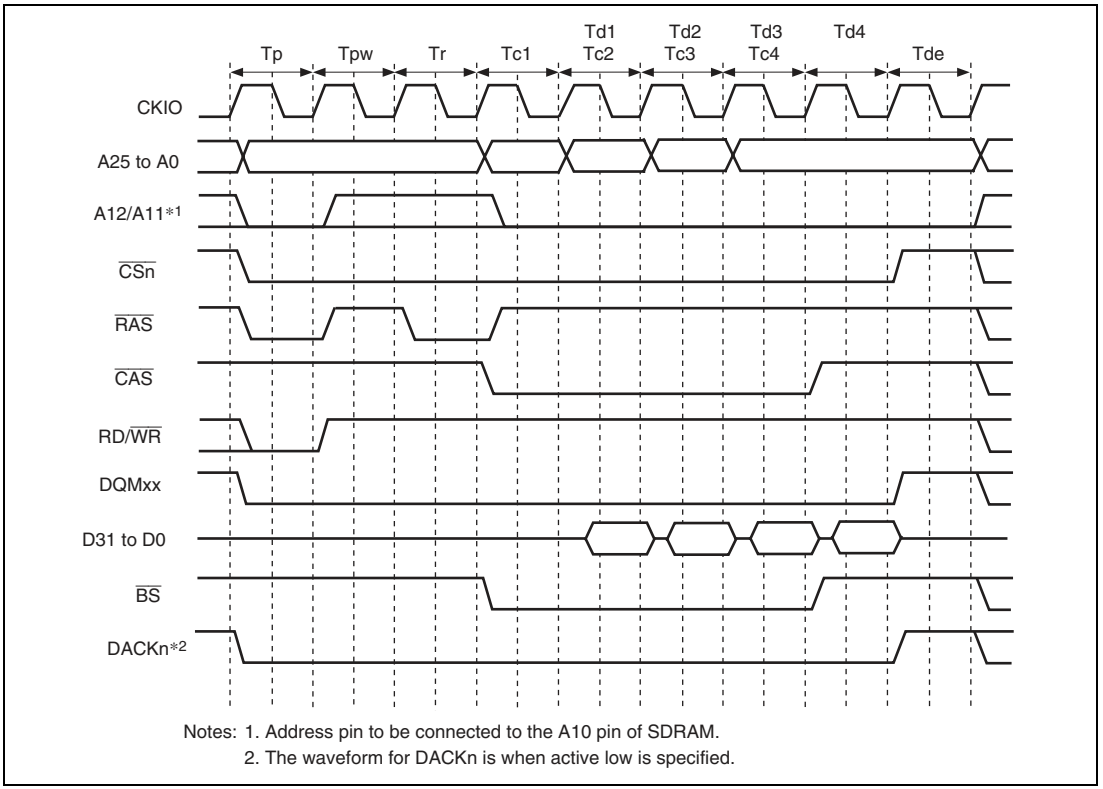


Figure 9.19 Burst Read Timing (No Auto-Precharge)



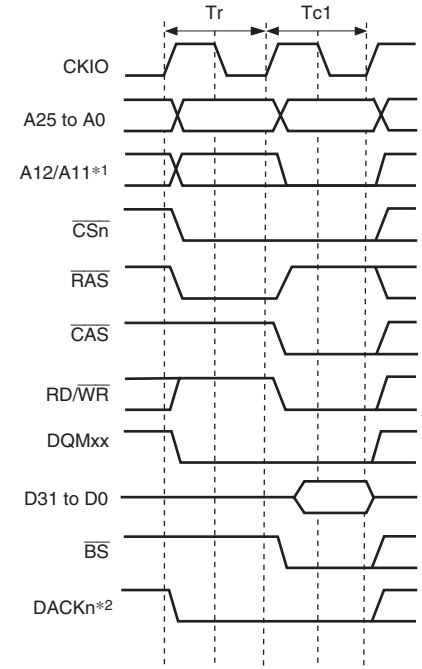
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.20 Burst Read Timing (Bank Active, Same Row Address)



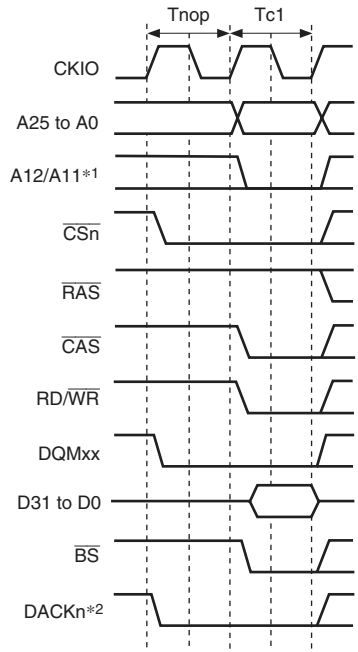
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.21 Burst Read Timing (Bank Active, Different Row Addresses)



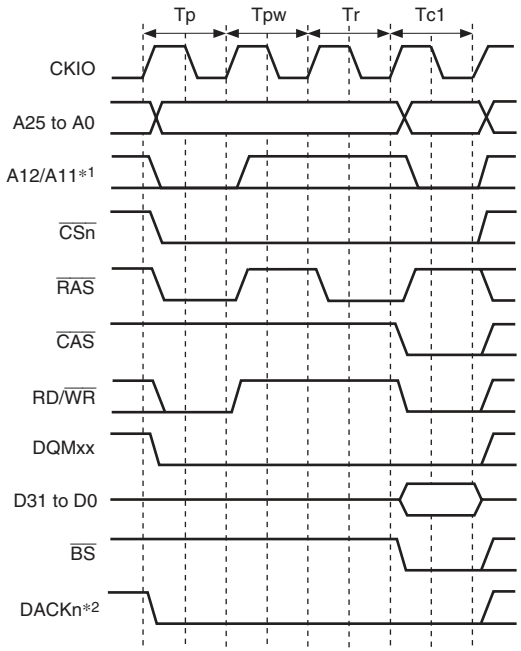
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.22 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 9.23 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 9.24 Single Write Timing (Bank Active, Different Row Addresses)

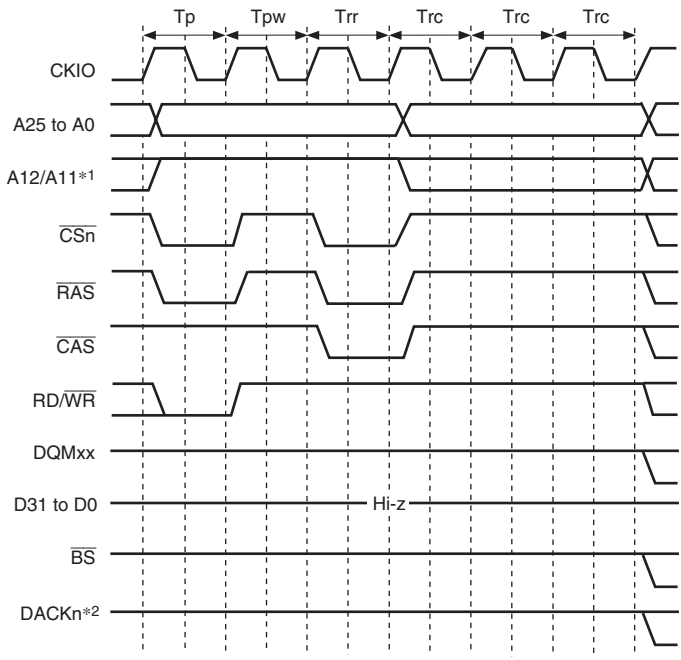
(8) Refreshing

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 9.25 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 9.25 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.

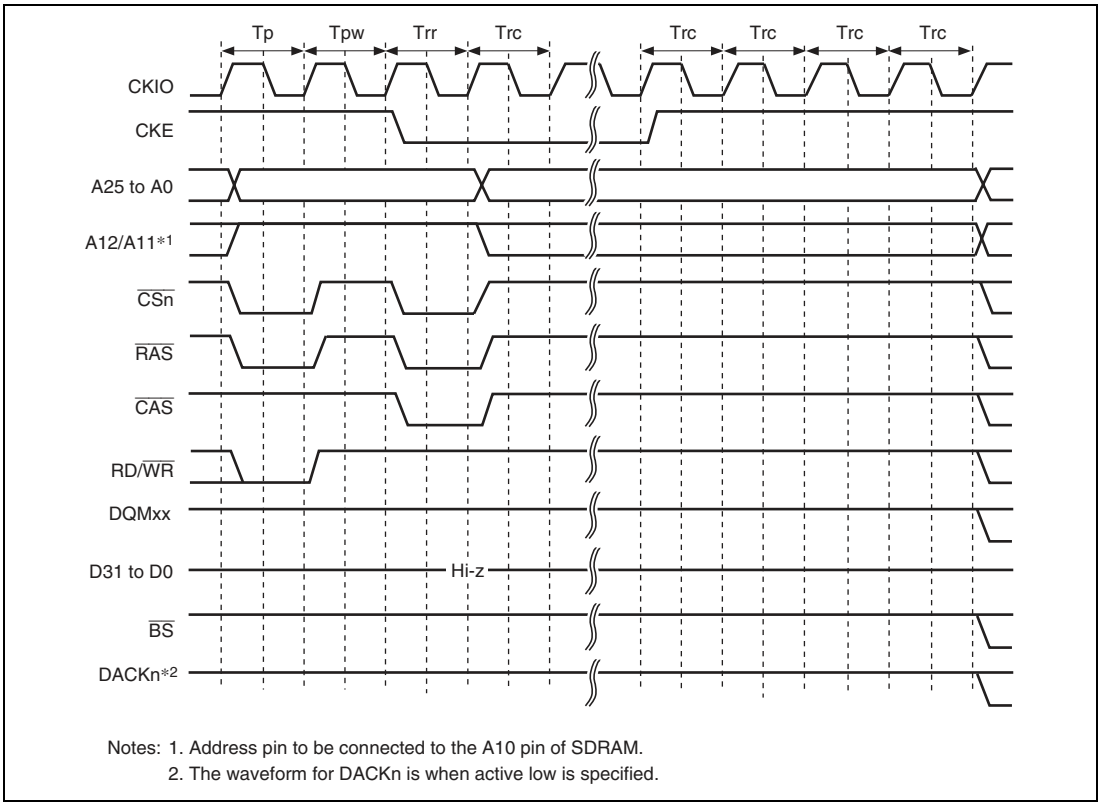
Self-refresh timing is shown in figure 9.26. Settings must be made so that self-refresh clearing and data retention are performed correctly, and auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the state in which auto-refreshing is set, or when exiting standby 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.



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 9.26 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.

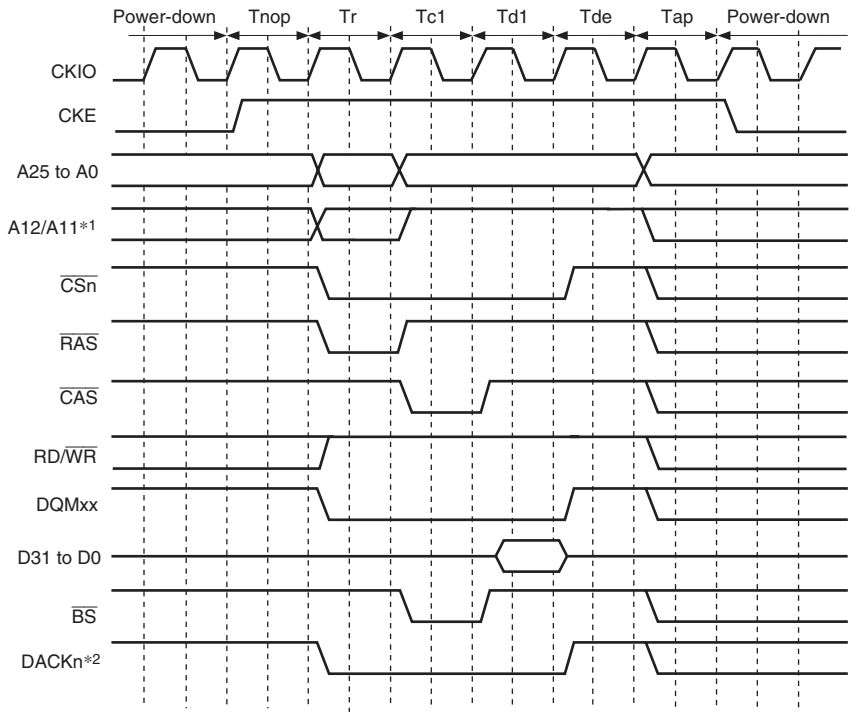
If a new refresh request occurs while waiting for the previous refresh request, the previous refresh request is deleted. To refresh correctly, a bus cycle longer than the refresh interval or the bus mastership occupation must be prevented from occurring.

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 9.27 shows the access timing in power-down mode.



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 9.27 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 RD/$\overline{\text{WR}}$ 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'A4FD4000 + X for area 2 SDRAM, and to address H'A4FD5000 + 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 to 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 9.19. In this time 0 is output at the external address pins of A12 or later.

Table 9.19 Access Address in SDRAM Mode Register Write

- Setting for Area 2 (SDMR2)

  Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'A4FD4440	H'0000440
	3	H'A4FD4460	H'0000460
32 bits	2	H'A4FD4880	H'0000880
	3	H'A4FD48C0	H'00008C0

  Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'A4FD4040	H'0000040
	3	H'A4FD4060	H'0000060
32 bits	2	H'A4FD4080	H'0000080
	3	H'A4FD40C0	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'A4FD5440	H'0000440
	3	H'A4FD5460	H'0000460
32 bits	2	H'A4FD5880	H'0000880
	3	H'A4FD58C0	H'00008C0

  Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'A4FD5040	H'0000040
	3	H'A4FD5060	H'0000060
32 bits	2	H'A4FD5080	H'0000080
	3	H'A4FD50C0	H'00000C0

Mode register setting timing is shown in figure 9.28. 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.

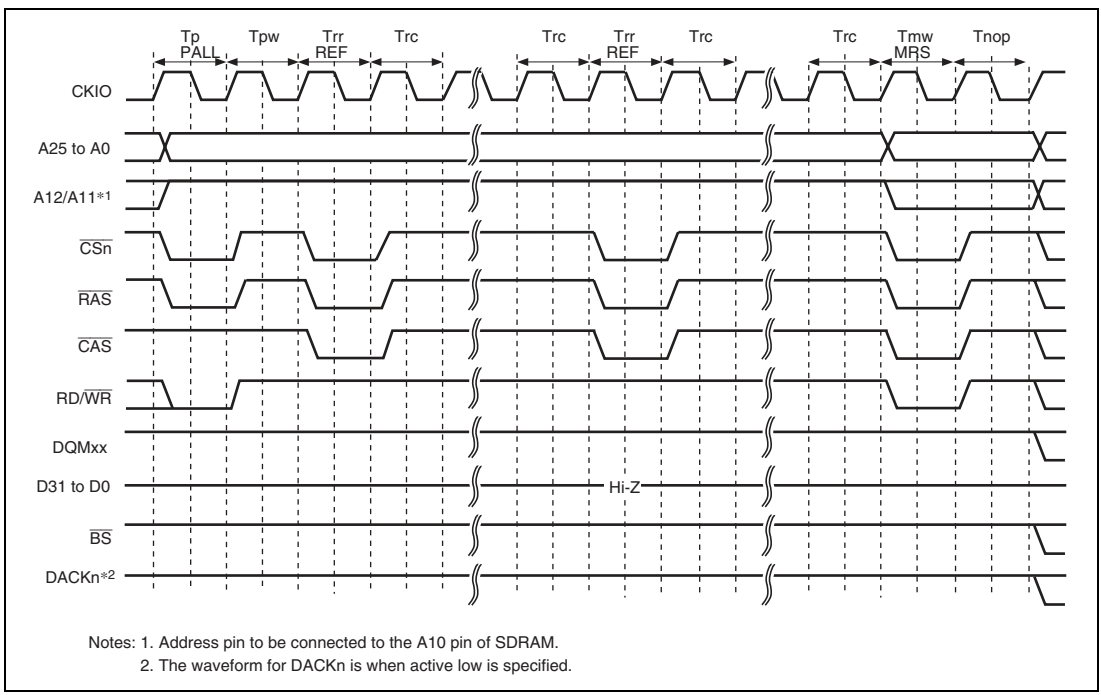


Figure 9.28 Write Timing for SDRAM Mode Register (Based on JEDEC)

(12) Low-Power SDRAM

The low-power SDRAM can be accessed using the same protocol as the normal SDRAM. The differences between the low-power SDRAM and normal SDRAM are that partial refresh takes place that puts only a part of the SDRAM in the self-refresh state during the self-refresh function, and that power consumption is low during refresh under user conditions such as the operating temperature. The partial refresh is effective in systems in which data in a work area other than the specific area can be lost without severe repercussions. For details, refer to the data sheet for the low-power SDRAM to be used.

The low-power SDRAM supports the extension mode register (EMRS) in addition to the mode registers as the normal SDRAM. This LSI supports issuing of the EMRS command.

The EMRS command is issued according to the conditions specified in table 9.20. For example, if data H'0YYYYYYY is written to address H'A4FD5XXX in long-word, the commands are issued to the CS3 space in the following sequence: PALL -> REF × 8 -> MRS -> EMRS. In this case, the MRS and EMRS issue addresses are H'0000XXX and H'YYYYYYY, respectively. If data H'1YYYYYYY is written to address H'A4FD5XXX in long-word, the commands are issued to the CS3 space in the following sequence: PALL -> MRS -> EMRS.

Table 9.20 Output Addresses when EMRS Command is Issued

Command to be Issued	Access Address	Access Data	Write Access Size	MRS Command Issue Address	EMRS Command Issue Address
CS2 MRS	H'A4FD4XXX	H'********	16 bits	H'0000XXX	—
CS3 MRS	H'A4FD5XXX	H'********	16 bits	H'0000XXX	—
CS2MRS +EMRS (with refresh)	H'A4FD4XXX	H'0YYYYYYY	32 bits	H'0000XXX	H'YYYYYYY
CS3 MRS +EMRS (with refresh)	H'A4FD5XXX	H'0YYYYYYY	32 bits	H'0000XXX	H'YYYYYYY
CS2 MRS +EMRS (without refresh)	H'A4FD4XXX	H'1YYYYYYY	32 bits	H'0000XXX	H'YYYYYYY
CS3 MRS +EMRS (without refresh)	H'A4FD5XXX	H'1YYYYYYY	32 bits	H'0000XXX	H'YYYYYYY

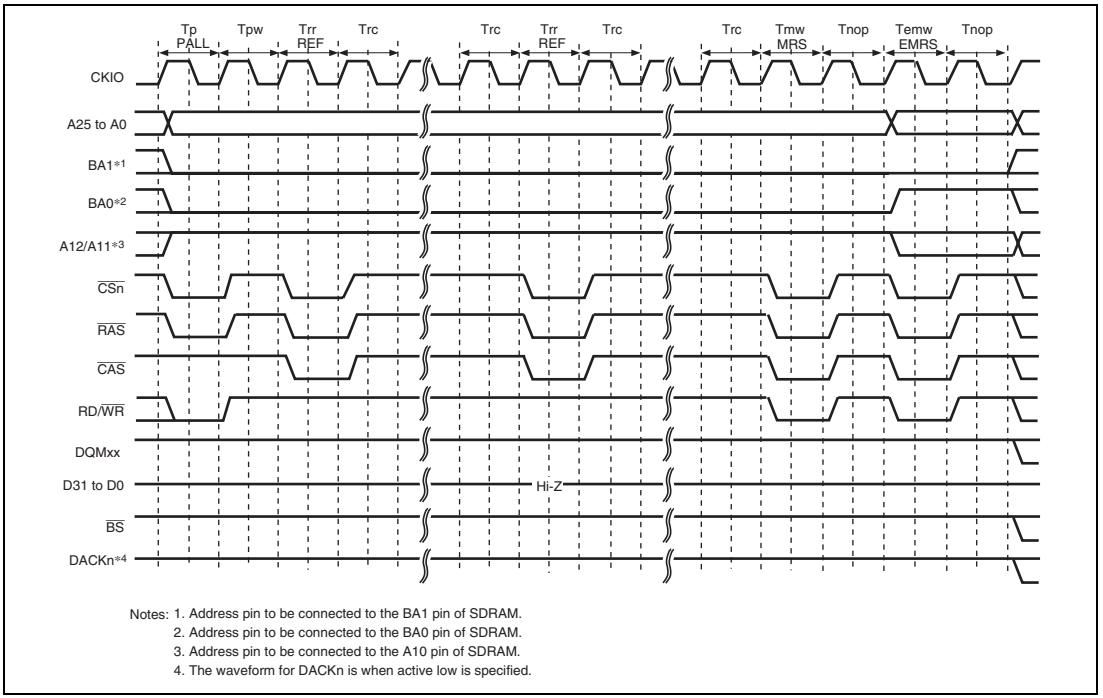
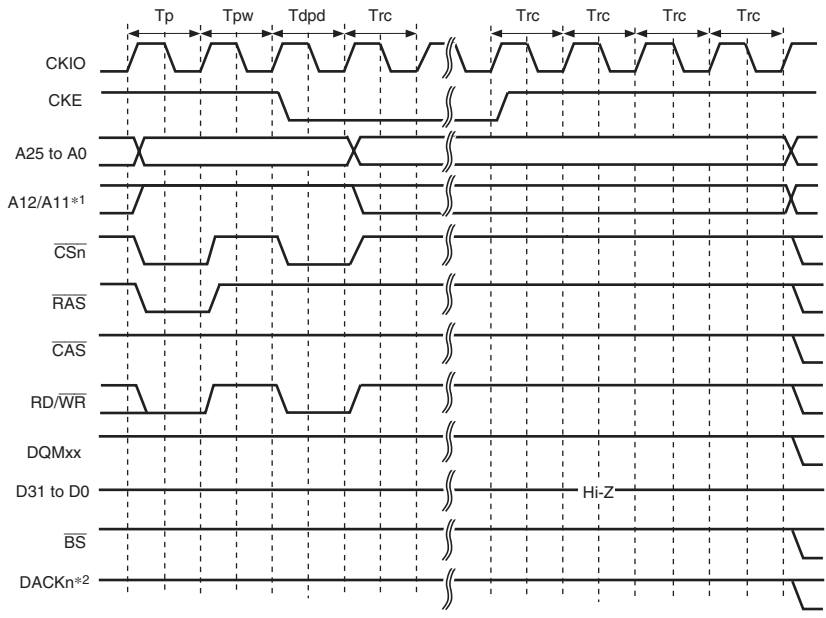


Figure 9.29 EMRS Command Issue Timing

- Deep power-down mode

  The low-power SDRAM supports the deep power-down mode as a low-power consumption mode. In the partial self-refresh function, self-refresh is performed on a specific area. In the deep power-down mode, self-refresh will not be performed on any memory area. This mode is effective in systems where all of the system memory areas are used as work areas.

If the RMODE bit of the SDCR is set to 1 while the DEEP and RFSH bits of the SDCR are set to 1, the low-power SDRAM enters the deep power-down mode. If the RMODE bit is cleared to 0, the CKE signal is pulled high to cancel the deep power-down mode. Before executing an access after returning from the deep power-down mode, the power-up sequence must be re-executed.



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 9.30 Transition Timing in Deep Power-Down Mode

9.5.6 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 CKIO.

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 9.21 lists a relationship between bus width, access size, and the number of bursts. Figure 9.31 shows a timing chart.

Table 9.21 Relationship between Bus Width, Access Size, and Number of Bursts

Bus Width	BEN Bit	Access Size	Number of Bursts	Number of Accesses
8 bits	Not affected	8 bits	1	1
	Not affected	16 bits	2	1
	Not affected	32 bits	4	1
	0	16 bytes	16	1
	1		4	4
16 bits	Not affected	8 bits	1	1
	Not affected	16 bits	1	1
	Not affected	32 bits	2	1
	0	16 bytes	8	1
	1		2	4
32 bits	Not affected	8 bits	1	1
	Not affected	16 bits	1	1
	Not affected	32 bits	1	1
	Not affected	16 bytes	4	1

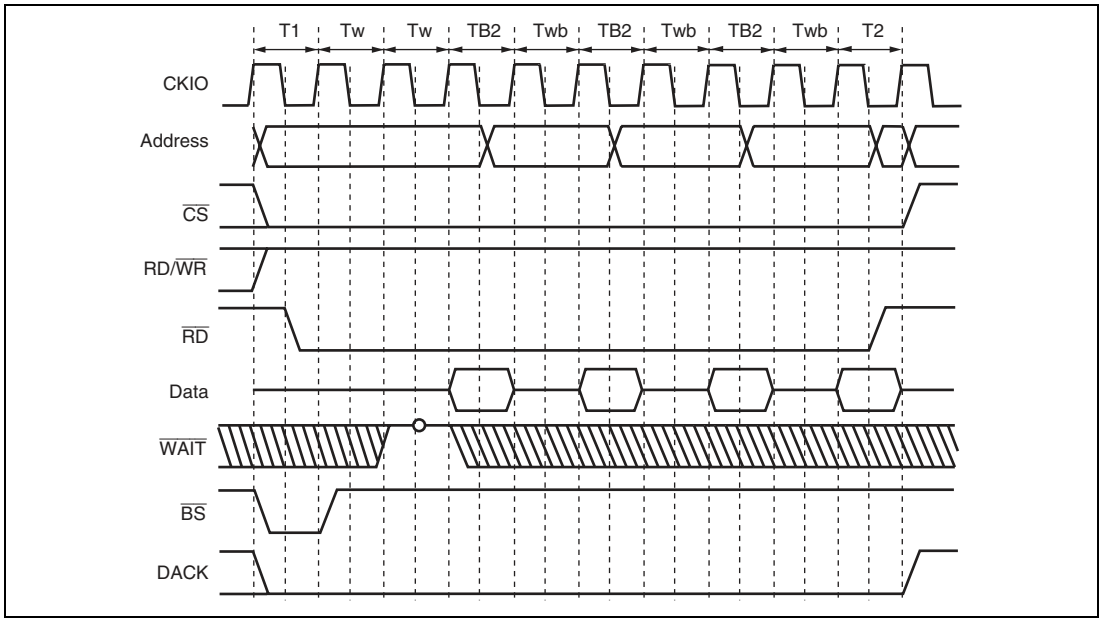


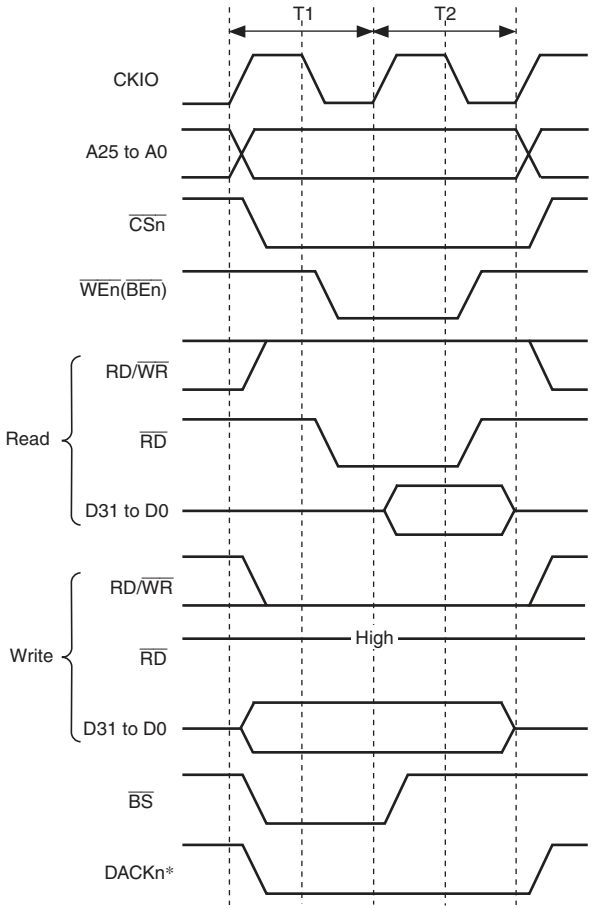
Figure 9.31 Burst ROM (Clock Asynchronous) Access (Bus Width = 32 Bits, 16-byte Transfer (Number of Bursts = 4), Access Wait for First Time = 2, Access Wait for 2nd Time and after = 1)

9.5.7 Byte-Selection SRAM Interface

The byte-selection SRAM interface is for access to an SRAM which has a byte-selection pin ($\overline{\text{WEn}}$ ($\overline{\text{BEn}}$)). 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 $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) pin, which is different from that for the normal space interface. The basic access timing is shown in figure 9.32. In write access, data is written to the memory according to the timing of the byte-selection pin ($\overline{\text{WEn}}$ ($\overline{\text{BEn}}$)). For details, refer to the data sheet for the corresponding memory.

If the BAS bit in CSnWCR is set to 1, the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) pin and RD/$\overline{\text{WR}}$ pin timings change. Figure 9.33 shows the basic access timing. In write access, data is written to the memory according to the timing of the write enable pin (RD/$\overline{\text{WR}}$). The data hold timing from RD/$\overline{\text{WR}}$ negation to data write must be acquired by setting the HW[1:0] bits in CSnWCR. Figure 9.34 shows the access timing when a software wait is specified.



Note: The waveform for DACKn is when active low is specified.

Figure 9.32 Basic Access Timing for Byte-Selection SRAM (BAS = 0)

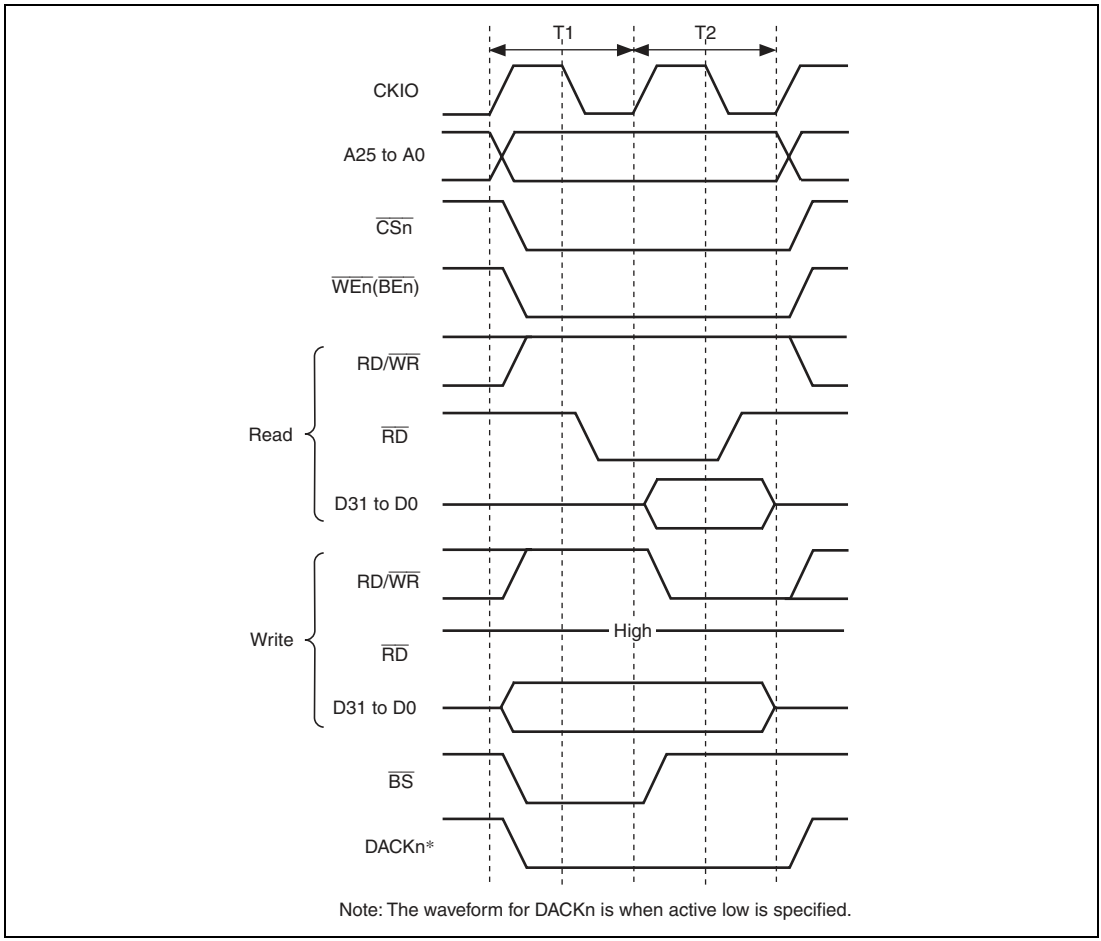


Figure 9.33 Basic Access Timing for Byte-Selection SRAM (BAS = 1)

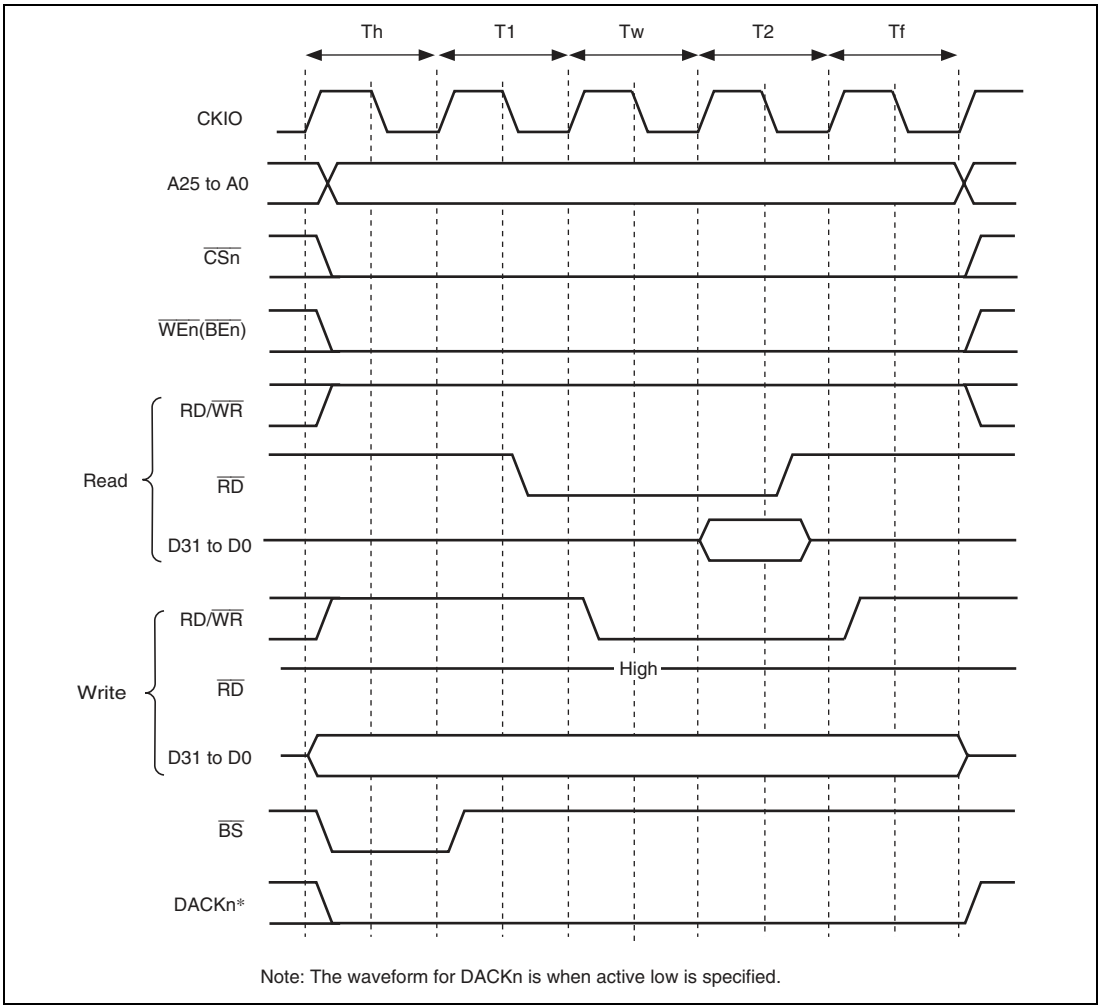


Figure 9.34 Wait Timing for Byte-Selection SRAM (BAS = 1) (Software Wait Only)

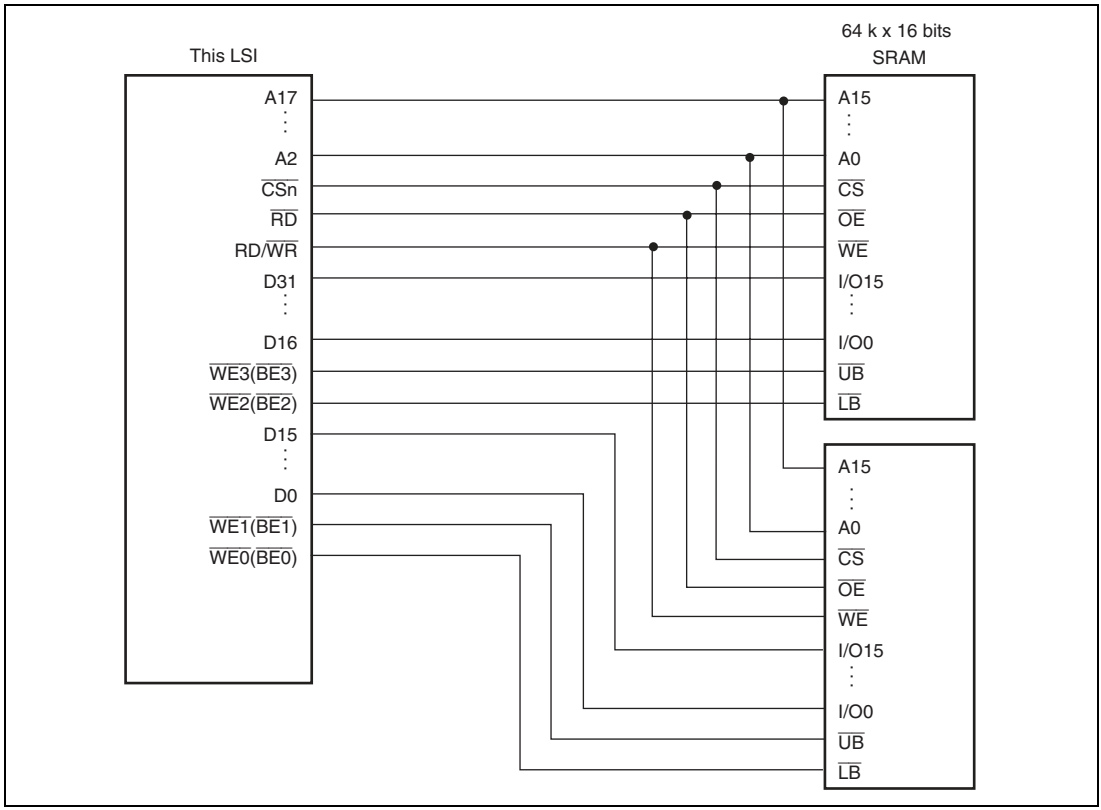


Figure 9.35 Example of Connection with 32-Bit Data-Width Byte-Selection SRAM

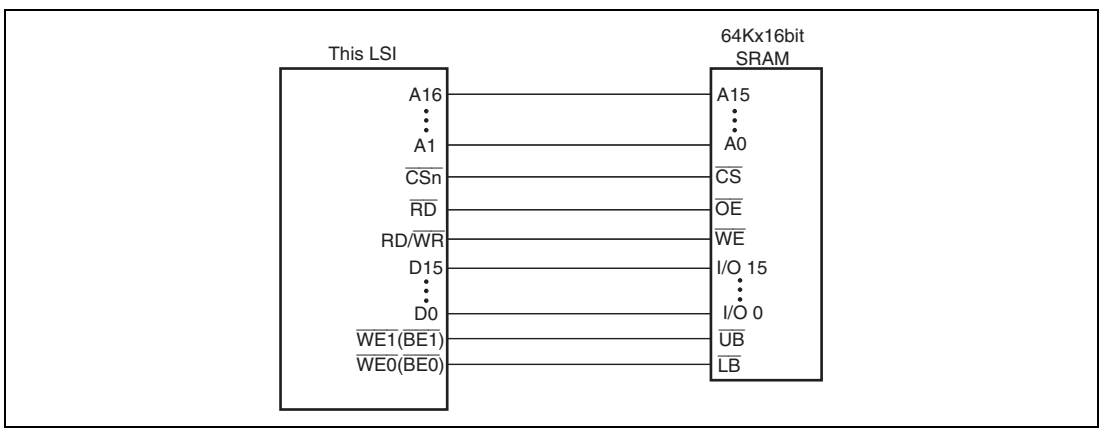


Figure 9.36 Example of Connection with 16-Bit Data-Width Byte-Selection SRAM

9.5.8 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 Rev. 2.1) 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 9.37 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.

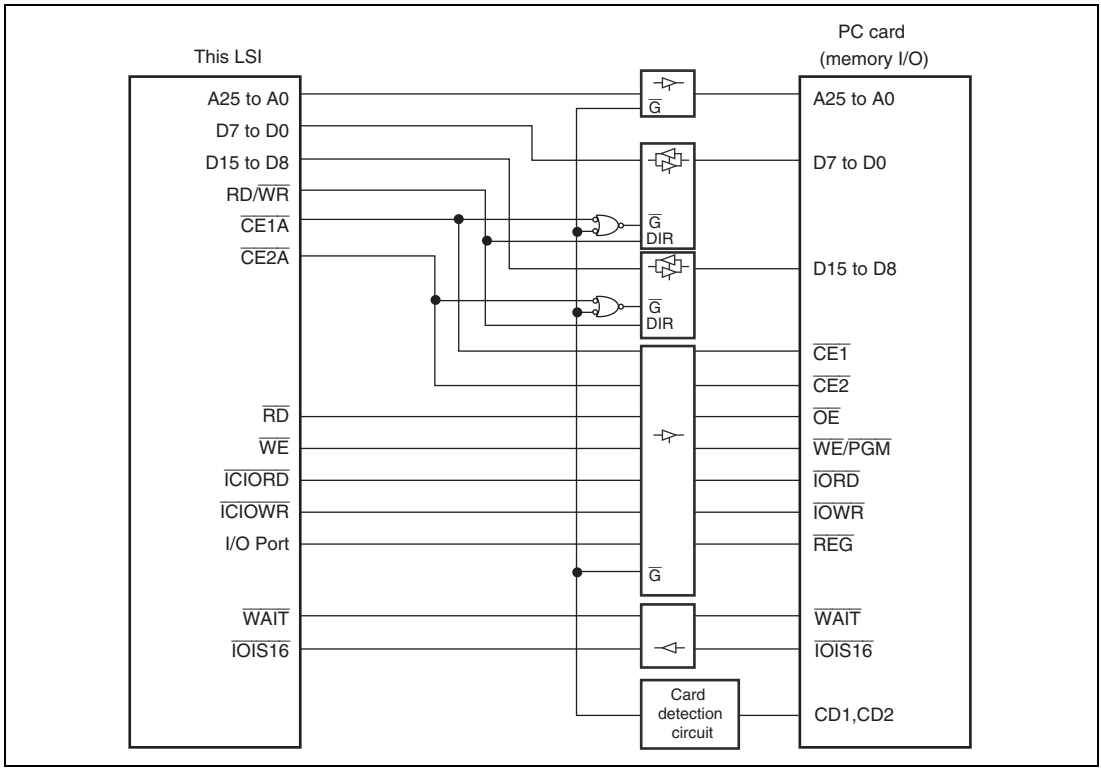


Figure 9.37 Example of PCMCIA Interface Connection

(1) Basic Timing for Memory Card Interface

Figure 9.38 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 (CKIO) 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 9.39 shows the PCMCIA memory bus wait timing.

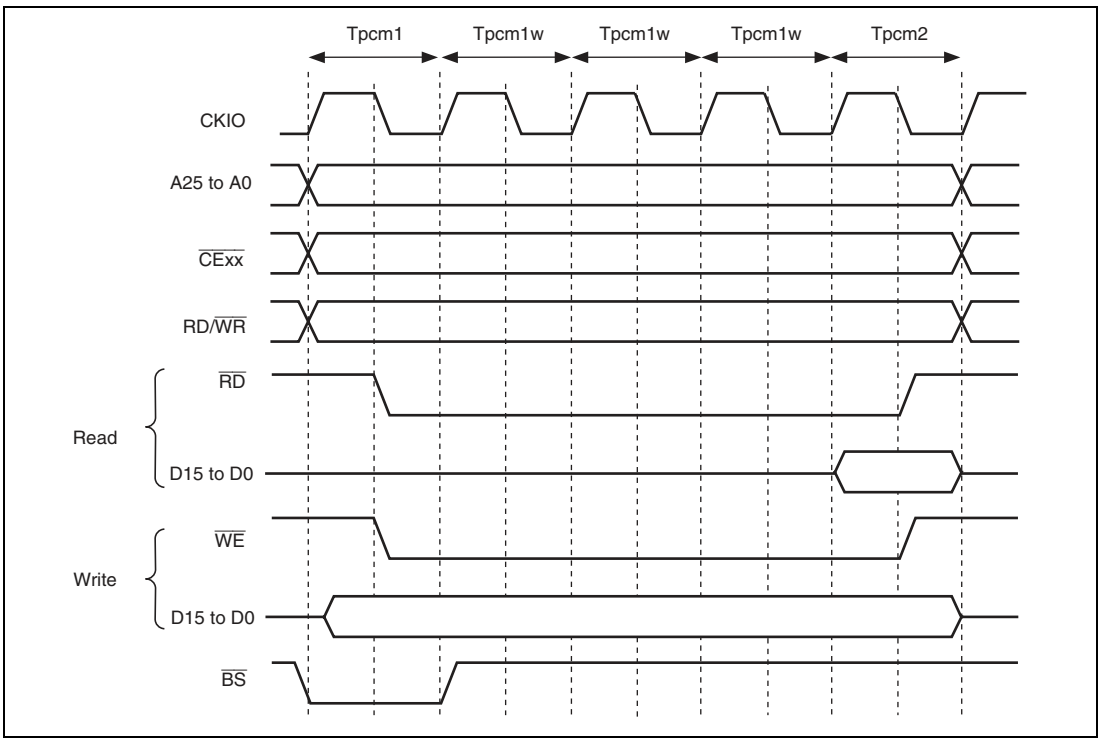


Figure 9.38 Basic Access Timing for PCMCIA Memory Card Interface

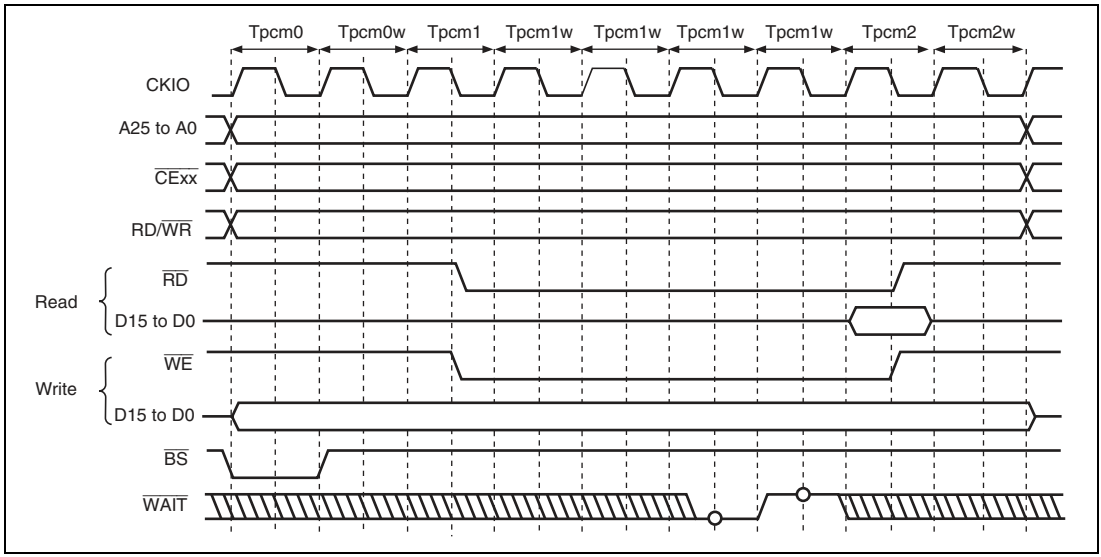


Figure 9.39 Wait Timing for PCMCIA Memory Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait = 1, Hardware Wait = 1)

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 an I/O 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.

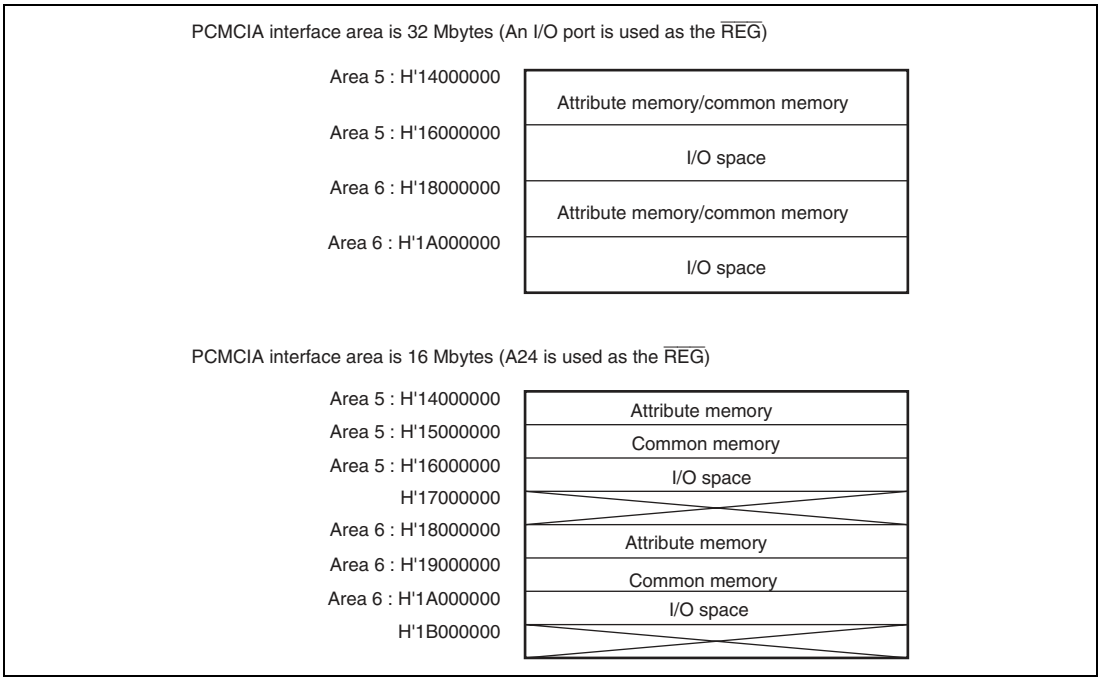


Figure 9.40 Example of PCMCIA Space Assignment (CS5BWCR.SA[1:0] = B'10, CS6BWCR.SA[1:0] = B'10)

(2) Basic Timing for I/O Card Interface

Figures 9.41 and 9.42 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 CKIO 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 $\overline{\text{CE2}}$ signal 1.5 cycles after the CKIO 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 $\overline{\text{CEn}}$.

Figure 9.43 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.

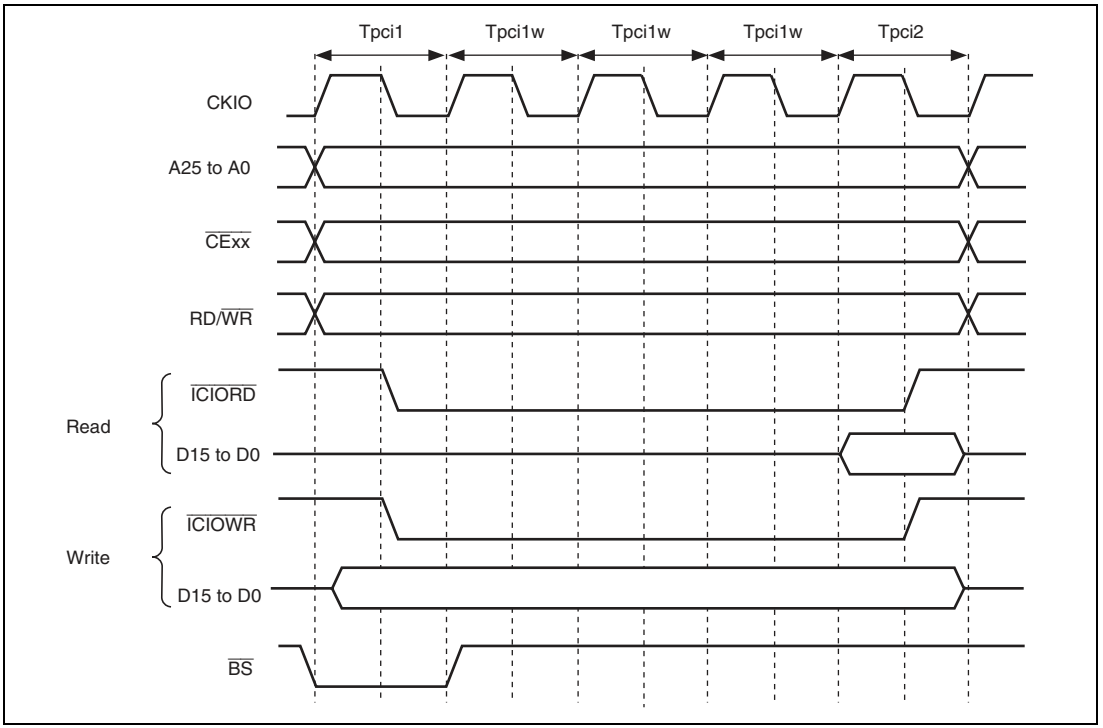


Figure 9.41 Basic Timing for PCMCIA I/O Card Interface

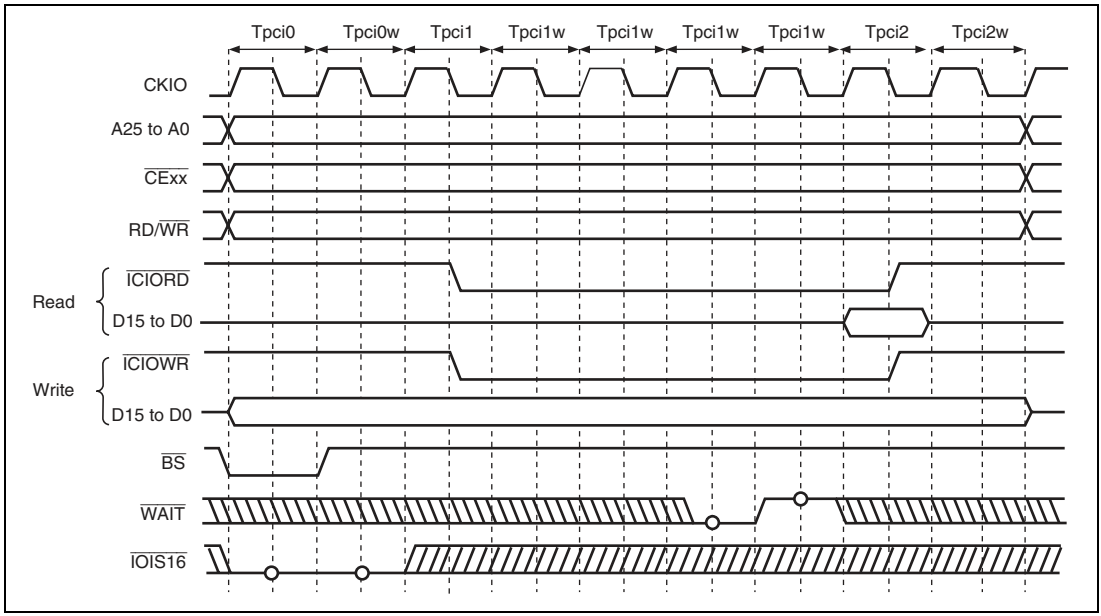


Figure 9.42 Wait Timing for PCMCIA I/O Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait = 1, Hardware Wait = 1)

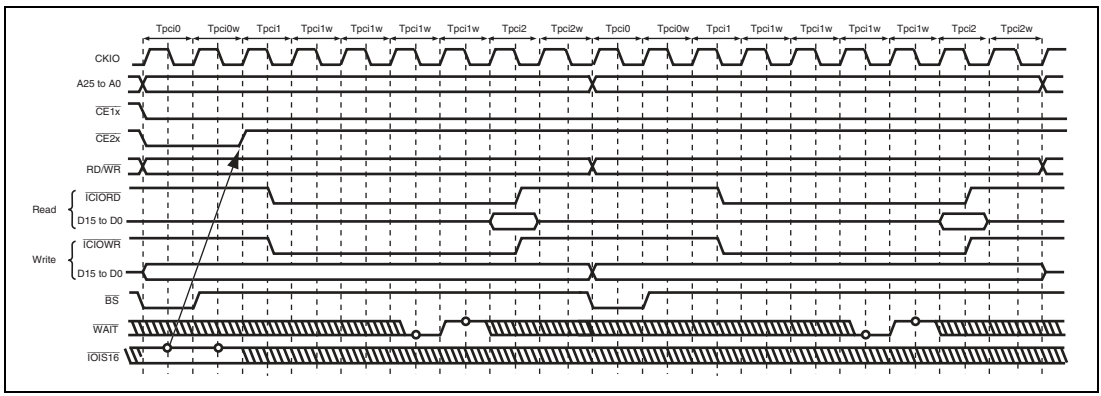


Figure 9.43 Timing for Dynamic Bus Sizing of PCMCIA I/O Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Waits = 3)

9.5.9 Burst ROM (Clock Synchronous) Interface

The burst ROM (clock synchronous) interface is supported to access a ROM with a synchronous burst function at high speed. The burst ROM interface accesses the burst ROM in the same way as a normal space. This interface is valid only for area 0.

In the first access cycle, wait cycles are inserted. In this case, the number of wait cycles to be inserted is specified by the W[3:0] bits of the CS0WCR. In the second and subsequent cycles, the number of wait cycles to be inserted is specified by the BW[1:0] bits of the CS0WCR.

While the burst ROM is accessed (clock synchronous), the $\overline{BS}$ signal is asserted only for the first access cycle and an external wait input is also valid for the first access cycle.

If the bus width is 16 bits, the burst length must be specified as 8. If the bus width is 32 bits, the burst length must be specified as 4. The burst ROM interface does not support the 8-bit bus width for the burst ROM. The burst ROM interface performs burst operations for all read accesses. For example, in a longword access over a 16-bit bus, valid 16-bit data is read two times and invalid 16-bit data is read six times.

These invalid data read cycles increase the memory access time and degrade the program execution speed and DMA transfer speed. To prevent this problem, a 16-byte read by cache fill or 16-byte read by the DMA should be used. The burst ROM interface performs write accesses in the same way as normal space access.

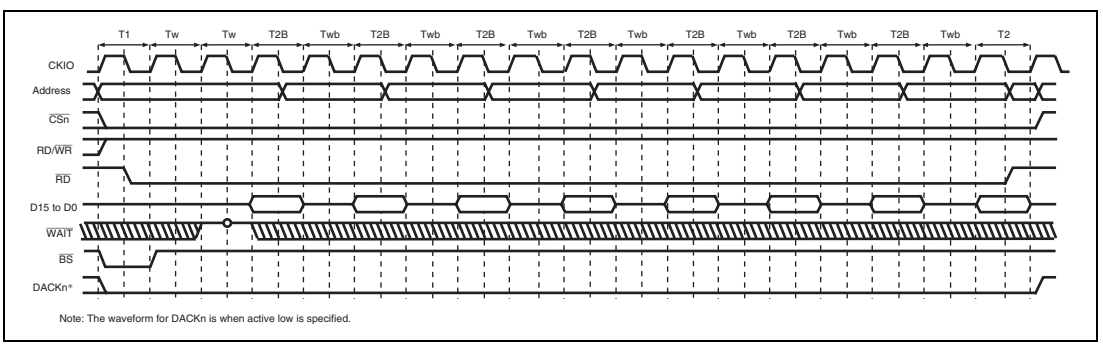


Figure 9.44 Burst ROM (Clock Synchronous) Access Timing (Burst Length = 8, Wait Cycles inserted in First Access = 2, Wait Cycles inserted in Second and Subsequent Accesses = 1)

9.5.10 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, and bits DMAIW[2:0] and DMAIWA in CMNCR. 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
6. Data output from an external device caused by DMA single transfer is followed by data output from another device that includes this LSI (DMAIWA = 0)
7. Data output from an external device caused by DMA single transfer is followed by any type of access (DMAIWA = 1)

9.5.11 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 $\overline{\text{CSn}}$ signal or other bus control signals. The states that do not allow bus mastership release are shown below.

1. 16-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 transfer by the DMAC or USBH
6. Setting the BLOCK bit in CMNCR to 1
7. 16 to 128-byte transfer by LCDC
8. Transfer by USBH

Bits DPRTY[1:0] in CMNCR can select whether or not the bus request is received during DMAC burst transfer.

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 CKIO. The bus mastership enable signal is asserted 0.5 cycles after the above timing, synchronized with the falling edge of CKIO. The bus control signals ($\overline{\text{BS}}$, $\overline{\text{CSn}}$, $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, DQMxx, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$), $\overline{\text{RD}}$, and RD/$\overline{\text{WR}}$) are placed in the high-impedance state at subsequent rising edges of CKIO. Bus request signals are sampled at the falling edge of CKIO.

The sequence for reclaiming the bus mastership from a slave is described below. 1.5 cycles after the negation of $\overline{\text{BREQ}}$ is detected at the falling edge of CKIO, the bus control signals are driven high. The $\overline{\text{BACK}}$ 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 CKIO where address and data signals are driven. Figure 9.45 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 $\overline{\text{REFOUT}}$ pin to request the bus mastership

while the SDRAM waits for the refresh. The LSI asserts the $\overline{\text{REFOUT}}$ pin until the bus mastership is received. If the slave releases the bus, the LSI acquires the bus mastership to execute the SDRAM refresh.

The bus release by the $\overline{\text{BREQ}}$ and $\overline{\text{BACK}}$ 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.

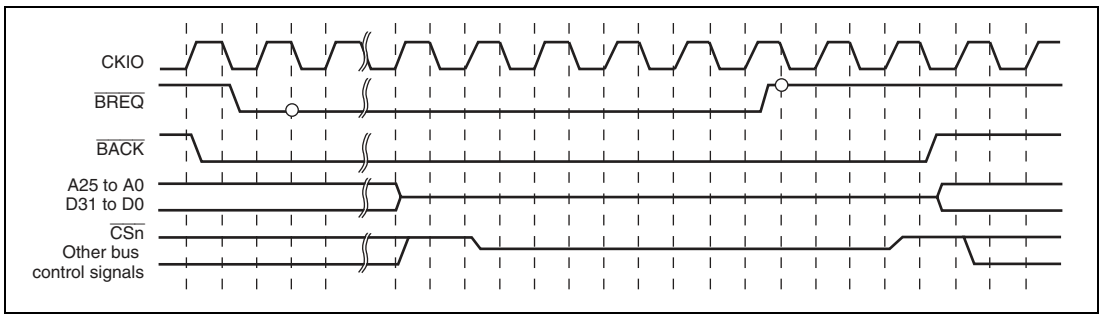


Figure 9.45 Bus Arbitration Timing

9.6 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 5-bit counter (RWTCNT). At power-on reset, the RWTCNT is cleared to 0. After power-on reset, RWTCNT is counted up synchronously together with CKIO and an external access will not be generated until RWTCNT is counted up to H′001F. At manual reset, RWTCNT is not cleared.

(2) Access from the Site of the LSI Internal Bus Master

There are three types of LSI internal buses: a cache bus, internal bus, and peripheral bus. The CPU and cache memory are connected to the cache bus. Internal bus masters other than the CPU and bus state controller are connected to the internal bus. Low-speed peripheral modules are connected to the peripheral bus. Internal memories other than the cache memory and debugging modules such as a UBC and AUD are connected bidirectionally to the cache bus and internal bus. Access from the cache bus to the internal bus is enabled but access from the internal bus to the cache bus is disabled. This gives rise to the following problems.

Internal bus masters such as DMAC other than the CPU can access on-chip memory other than the cache memory but cannot access the cache memory. If an on-chip bus master other than the CPU writes data to an external memory other than the cache, 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 on-chip bus master other than the CPU, the corresponding cache memory should be purged by software.

If the CPU initiates read access for the cache, the cache is searched. If the cache stores data, the CPU latches the data and completes the read access. If the cache does not store data, the CPU

performs four contiguous longword read cycles 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 performs four contiguous longword accesses to perform a cache fill operation on the external interface. For a cache-through area, the CPU performs access according to the actual access addresses. For an instruction fetch to an even word boundary (4n), the CPU performs longword access. For an instruction fetch to an odd word boundary (4n + 2), the CPU performs word access.

For a read cycle of a cache-through area or an on-chip peripheral module, the read cycle is first accepted and then read cycle is initiated. The read data is sent to the CPU via the cache bus.

In a write cycle for the cache area, 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, 16-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 16-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.

Since the bus state controller (BSC) incorporates a one-stage write buffer, the BSC can execute an access via the internal bus before the previous external bus cycle is completed in a write cycle. If the on-chip module is read or written after the external low-speed memory is written, the on-chip module can be accessed before the completion of the external low-speed memory write cycle.

In read cycles, the CPU 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) On-Chip Peripheral Module Access

To access an on-chip module register, two or more peripheral module clock (Pϕ) cycles are required. Care must be taken in system design.

(4) External Bus Priority Order

Access via an external bus is performed in the priority order below:

$\overline{\text{BREQ}}$ > Refresh > LCDC > USBH > DMAC > CPU

Note that next transfer is not performed until current transfer (e.g. burst transfer) has completed.

Section 10 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.

10.1 Features

- Six channels (two channels can receive an external request)
- 4-Gbyte physical address space
- Data transfer unit is selectable: Byte, word (2 bytes), longword (4 bytes), and 16 bytes (longword × 4)
- Maximum transfer count: 16,777,216 transfers
- Address mode: Dual address mode or single address mode can be selected.
- 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, SIOF1, MMC, CMT (channels 0 to 4), SIM, USBF, SIOF0, SIOF1, ADC, and SDHI
- 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 transfers end by the specified counts.
- External request detection: There are following four types of DREQ input detection.

  — Low level detection

  — High level detection

  — Rising edge detection

  — Falling edge detection
- Transfer request acknowledge signal:

  Active levels for DACK and TEND can be set independently.

Figure 10.1 shows the block diagram of the DMAC.

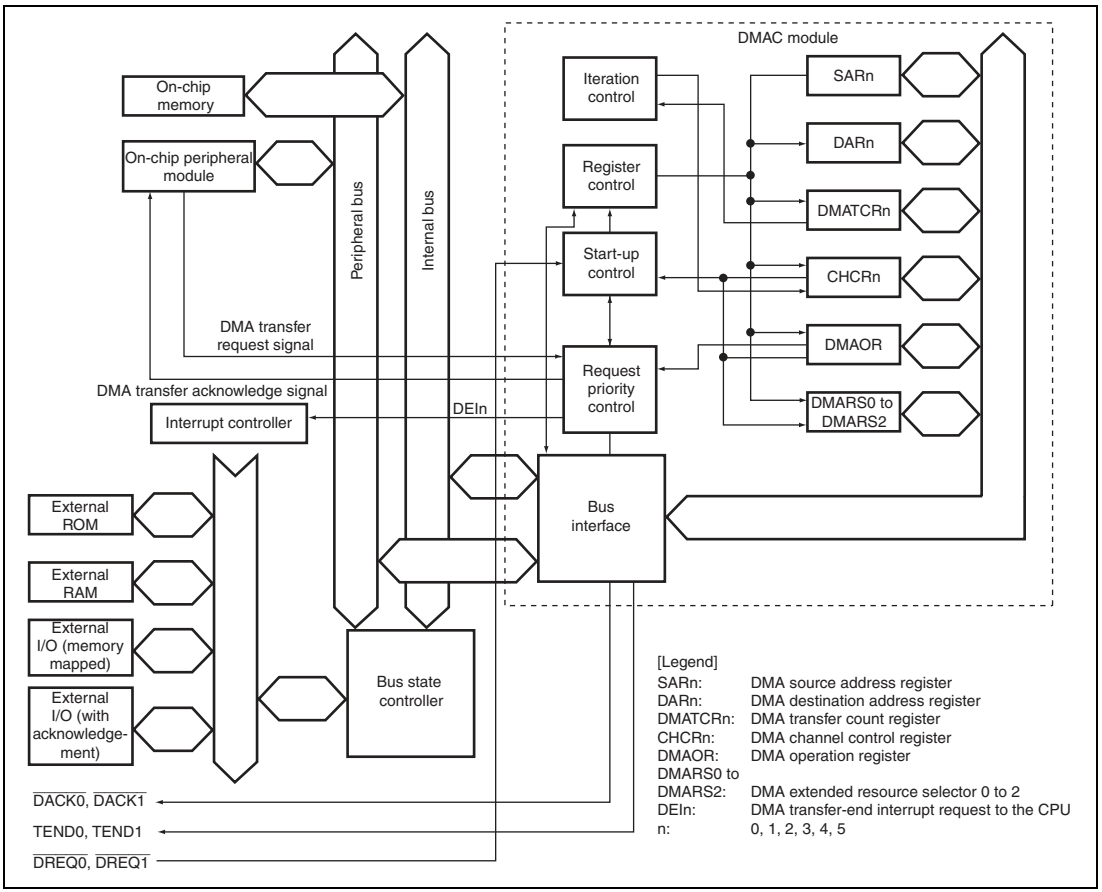


Figure 10.1 Block Diagram of DMAC

10.2 Input/Output Pins

The external pins for the DMAC are described below. Table 10.1 lists the configuration of the pins that are connected to external bus. The DMAC has pins for 2 channels (channels 0 and 1) for external bus use.

Table 10.1 Pin Configuration

Channel	Name	Pin Name	I/O	Function
0	DMA transfer request	$\overline{\text{DREQ0}}$	Input	DMA transfer request input from external device to channel 0
	DMA transfer request reception	$\overline{\text{DACK0}}$	Output	DMA transfer request acknowledge output from channel 0 to external device
	DMA transfer end	$\overline{\text{TEND0}}$	Output	DMA transfer end of DMAC channel 0 output of
1	DMA transfer request	$\overline{\text{DREQ1}}$	Input	DMA transfer request input from external device to channel 1
	DMA transfer request reception	$\overline{\text{DACK1}}$	Output	DMA transfer request acknowledge output from channel 1 to external device
	DMA transfer end	$\overline{\text{TEND1}}$	Output	DMA transfer end of DMAC channel 1 output

10.3 Register Descriptions

The DMAC has the following registers. Refer to section 37, List of Registers, for more details on the addresses and states of these registers in each operating mode. The SAR for channel 0 is expressed such as SAR_0.

(1) Channel 0

- DMA source address register_0 (SAR_0)
- DMA destination address register_0 (DAR_0)
- DMA transfer count register_0 (DMATCR_0)
- DMA channel control register_0 (CHCR_0)

(2) Channel 1

- DMA source address register_1 (SAR_1)
- DMA destination address register_1 (DAR_1)
- DMA transfer count register_1 (DMATCR_1)
- DMA channel control register _1 (CHCR_1)

(3) Channel 2

- DMA source address register_2 (SAR_2)
- DMA destination address register_2 (DAR_2)
- DMA transfer count register_2 (DMATCR_2)
- DMA channel control register_2 (CHCR_2)

(4) Channel 3

- DMA source address register_3 (SAR_3)
- DMA destination address register_3 (DAR_3)
- DMA transfer count register_3 (DMATCR_3)
- DMA channel control register_3 (CHCR_3)

(5) Channel 4

- DMA source address register_4 (SAR_4)
- DMA destination address register_4 (DAR_4)
- DMA transfer count register_4 (DMATCR_4)
- DMA channel control register_4 (CHCR_4)

(6) Channel 5

- DMA source address register_5 (SAR_5)
- DMA destination address register_5 (DAR_5)
- DMA transfer count register_5 (DMATCR_5)
- DMA channel control register_5 (CHCR_5)

(7) Common

- DMA operation register (DMAOR)
- DMA extended resource selector 0 (DMARS0)
- DMA extended resource selector 1 (DMARS1)
- DMA extended resource selector 2 (DMARS2)

10.3.1 DMA Source Address Registers (SAR_0 to SAR_5)

SAR 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. When the data is transferred from an external device with the DACK in single address mode, the SAR is ignored.

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit address boundary. When transferring data in 16-byte units, a 16-byte boundary must be set for the source address value. The initial value is undefined.

10.3.2 DMA Destination Address Registers (DAR_0 to DAR_5)

DAR 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. When the data is transferred from an external device with the DACK in single address mode, the DAR is ignored.

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit address boundary. When transferring data in 16-byte units, a 16-byte boundary must be set for the destination address value. The initial value is undefined.

10.3.3 DMA Transfer Count Registers (DMATCR_0 to DMATCR_5)

DMATCR 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 DMATCR are always read as 0, and the write value should always be 0. To transfer data in 16 bytes, one 16-byte transfer (128 bits) counts one. The initial value is undefined.

10.3.4 DMA Channel Control Registers (CHCR_0 to CHCR_5)

CHCR are 32-bit readable/writable registers that control the DMA transfer mode.

Bit	Bit Name	Initial Value	R/W	Descriptions
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23	DO	0	R/W	DMA Overrun Selects whether DREQ is detected by overrun 0 or by overrun 1. This bit is valid only in CHCR_0 and CHCR_1. This bit is always reserved and read as 0 in CHCR_2 to CHCR_5. The write value should always be 0. 0: Detects DREQ by overrun 0 1: Detects DREQ by overrun 1
22	TL	0	R/W	Transfer End Level Specifies whether the TEND signal output is high active or low active. This bit is valid only in CHCR_0 and CHCR_1. This bit is always reserved and read as 0 in CHCR2 to CHCR_5. The write value should always be 0. 0: Low-active output of TEND 1: High-active output of TEND
21 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	AM	0	R/W	Acknowledge Mode Selects whether DACK is output in data read cycle or in data write cycle in dual address mode. In single address mode, DACK is always output regardless of the specification by this bit. This bit is valid only in CHCR_0 and CHCR_1. This bit is always reserved and read as 0 in CHCR_2 to CHCR_5. The write value should always be 0. 0: DACK output in read cycle (dual address mode) 1: DACK output in write cycle (dual address mode)

Bit	Bit Name	Initial Value	R/W	Descriptions
16	AL	0	R/W	Acknowledge Level Specifies whether the DACK signal output is high active or low active. This bit is valid only in CHCR_0 and CHCR_1. This bit is always reserved and read as 0 in CHCR_2 to CHCR_5. The write value should always be 0. 0: Low-active output of DACK 1: High-active output of DACK
15	DM1	0	R/W	Destination Address Mode 1, 0
14	DM0	0	R/W	Specify whether the DMA destination address is incremented, decremented, or left fixed. (In single address mode, the DM1 and DM0 bits are ignored when data is transferred to an external device with DACK.) 00: Fixed destination address (setting prohibited in 16-byte transfer) 01: Destination address is incremented (+1 in byte-unit transfer, +2 in word-unit transfer, +4 in longword-unit transfer, +16 in 16-byte transfer) 10: Destination address is decremented (–1 in byte-unit transfer, –2 in word-unit transfer, –4 in longword-unit transfer; setting prohibited in 16-byte transfer) 11: Setting prohibited
13	SM1	0	R/W	Source Address Mode 1, 0
12	SM0	0	R/W	Specify whether the DMA source address is incremented, decremented, or left fixed. (In single address mode, SM1 and SM0 bits are ignored when data is transferred from an external device with DACK.) 00: Fixed source address (setting prohibited in 16-byte transfer) 01: Source address is incremented (+1 in byte-unit transfer, +2 in word-unit transfer, +4 in longword-unit transfer, +16 in 16-byte transfer) 10: Source address is decremented (–1 in byte-unit transfer, –2 in word-unit transfer, –4 in longword-unit transfer; setting prohibited in 16-byte transfer) 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Descriptions
11	RS3	0	R/W	Resource Select 3 to 0
10	RS2	0	R/W	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.
9	RS1	0	R/W	
8	RS0	0	R/W	

0	0	0	0	External request, dual address mode
0	0	0	1	Setting prohibited
0	0	1	0	External request, single address mode External address space → External device with DACK
0	0	1	1	External request, single address mode External device with DACK → External address space
0	1	0	0	Auto request
0	1	0	1	Setting prohibited
0	1	1	0	Setting prohibited
0	1	1	1	Setting prohibited
1	0	0	0	Selected by DMA extended resource selector
1	0	0	1	Setting prohibited
1	0	1	0	Setting prohibited
1	0	1	1	Setting prohibited
1	1	0	0	Setting prohibited
1	1	0	1	Setting prohibited
1	1	1	0	ADC
1	1	1	1	Setting prohibited

Note: External request specification is valid only in CHCR_0 and CHCR_1. None of the external request can 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 and the detecting level. These bits are valid only in CHCR_0 and CHCR_1. These bits are always reserved and read as 0 in CHCR_2 to CHCR_5. The write value should always be 0. 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	TS1	0	R/W	Transfer Size 1, 0
3	TS0	0	R/W	Specify the size of data to be transferred. Select the size of data to be transferred when the source or destination is an on-chip peripheral module register of which transfer size is specified. 00: Byte size 01: Word size (2 bytes) 10: Longword size (4 bytes) 11: 16-byte unit (four longword transfers)
2	IE	0	R/W	Interrupt Enable Specifies whether or not an interrupt request is generated to the CPU at the end of the DMA transfer. Setting this bit to 1 generates an interrupt request (DEI) to the CPU when the TE bit is set to 1. 0: Interrupt request is disabled. 1: Interrupt request is enabled.

Bit	Bit Name	Initial Value	R/W	Descriptions
1	TE	0	R/(W)*	Transfer End Flag Shows that DMA transfer ends. The TE bit is set to 1 when data transfer ends when DMATCR becomes to 0. The TE bit is not set to 1 in the following cases. • DMA transfer ends due to an NMI interrupt or DMA address error before DMATCR is cleared to 0. • DMA transfer is ended by clearing the DE bit and DME bit in the DMA operation register (DMAOR). To clear the TE bit, the TE bit should be written to 0 after reading 1. Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled. 0: During the DMA transfer or DMA transfer has been interrupted [Clearing condition] Writing 0 after TE = 1 read 1: DMA transfer ends by the specified count (DMATCR = 0)
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.

10.3.5 DMA Operation Register (DMAOR)

DMAOR is a 16-bit readable/writable register that specifies the priority level of channels at the DMA transfer. This register shows the DMA transfer status.

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	CMS1	0	R/W	Cycle Steal Mode Select 1, 0
12	CMS0	0	R/W	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. 00: Normal mode 01: Setting prohibited 10: Intermittent mode 16 Executes one DMA transfer in each of 16 clocks of an external bus clock. 11: Intermittent mode 64 Executes one DMA transfer in each of 64 clocks of an external bus clock.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PR1	0	R/W	Priority Mode 1, 0
8	PR0	0	R/W	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.

Bit	Bit Name	Initial Value	R/W	Description
2	AE	0	R/(W)*	Address Error Flag Indicates that an address error occurred during DMA transfer. 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. This bit can only be cleared by writing 0 after reading 1. 0: No DMAC address error [Clearing condition] Writing AE = 0 after AE = 1 read 1: DMAC address error occurs
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. This bit can only be cleared by writing 0 after reading 1. When the NMI is input, the DMA transfer in progress can be done in 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.

10.3.6 DMA Extended Resource Selectors 0 to 2 (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 SCIF0, SIOF1, MMC, CMT (channels 0 to 4), SIM, USBF, SIOF0, SIOF1, and SDHI.

When MID/RID other than the values listed in table 10.2 is set, the operation of this LSI is not guaranteed. The transfer request from DMARS is valid only when the resource select bits (RS3 to RS0) have 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	C1MID5	0	R/W	Transfer request module ID5 to ID0 for DMA channel 1 (MID) See table 10.2.
14	C1MID4	0	R/W	
13	C1MID3	0	R/W	
12	C1MID2	0	R/W	
11	C1MID1	0	R/W	
10	C1MID0	0	R/W	
9	C1RID1	0	R/W	Transfer request register ID1 and ID0 for DMA channel 1 (RID) See table 10.2.
8	C1RID0	0	R/W	
7	C0MID5	0	R/W	Transfer request module ID5 to ID0 for DMA channel 0 (MID) See table 10.2.
6	C0MID4	0	R/W	
5	C0MID3	0	R/W	
4	C0MID2	0	R/W	
3	C0MID1	0	R/W	
2	C0MID0	0	R/W	
1	C0RID1	0	R/W	Transfer request register ID1 and ID0 for DMA channel 0 (RID) See table 10.2.
0	C0RID0	0	R/W	

- DMARS1

Bit	Bit Name	Initial Value	R/W	Description
15	C3MID5	0	R/W	Transfer request module ID5 to ID0 for DMA channel 3 (MID) See table 10.2.
14	C3MID4	0	R/W	
13	C3MID3	0	R/W	
12	C3MID2	0	R/W	
11	C3MID1	0	R/W	
10	C3MID0	0	R/W	
9	C3RID1	0	R/W	Transfer request register ID1 and ID0 for DMA channel 3 (RID) See table 10.2.
8	C3RID0	0	R/W	
7	C2MID5	0	R/W	Transfer request module ID5 to ID0 for DMA channel 2 (MID) See table 10.2.
6	C2MID4	0	R/W	
5	C2MID3	0	R/W	
4	C2MID2	0	R/W	
3	C2MID1	0	R/W	
2	C2MID0	0	R/W	
1	C2RID1	0	R/W	Transfer request register ID1 and ID0 for DMA channel 2 (RID) See table 10.2.
0	C2RID0	0	R/W	

- DMARS2

Bit	Bit Name	Initial Value	R/W	Description
15	C5MID5	0	R/W	Transfer request module ID5 to ID0 for DMA channel 5 (MID) See table 10.2.
14	C5MID4	0	R/W	
13	C5MID3	0	R/W	
12	C5MID2	0	R/W	
11	C5MID1	0	R/W	
10	C5MID0	0	R/W	
9	C5RID1	0	R/W	Transfer request register ID1 and ID0 for DMA channel 5 (RID) See table 10.2.
8	C5RID0	0	R/W	
7	C4MID5	0	R/W	Transfer request module ID5 to ID0 for DMA channel 4 (MID) See table 10.2.
6	C4MID4	0	R/W	
5	C4MID3	0	R/W	
4	C4MID2	0	R/W	
3	C4MID1	0	R/W	
2	C4MID0	0	R/W	
1	C4RID1	0	R/W	Transfer request register ID1 and ID0 for DMA channel 4 (RID) See table 10.2.
0	C4RID0	0	R/W	

Table 10.2 Transfer Request Sources

Peripheral Module	Setting Value for One Channel (MID + RID)	MID	RID	Function
SCIF0	H'21	B'001000	B'01	Transmit
	H'22		B'10	Receive
SCIF1	H'29	B'001010	B'01	Transmit
	H'2A		B'10	Receive
CMT (channel 0)	H'03	B'000000	B'11	—
CMT (channel 1)	H'07	B'000001	B'11	—
CMT (channel 2)	H'0B	B'000010	B'11	—
CMT (channel 3)	H'0F	B'000011	B'11	—
CMT (channel 4)	H'13	B'000100	B'11	—
USBF	H'83	B'100000	B'11	Transmit
	H'80		B'00	Receive
SIM	H'A1	B'101000	B'01	Transmit
	H'A2		B'10	Receive
MMC	H'A8	B'101010	B'00	Transmit/receive
SIOF0	H'B1	B'101100	B'01	Transmit
	H'B2		B'10	Receive
SIOF1	H'B5	B'101101	B'01	Transmit
	H'B6		B'10	Receive
SDHI	H'C1	B'110000	B'01	Transmit
	H'C2		B'10	Receive

10.4 Operation

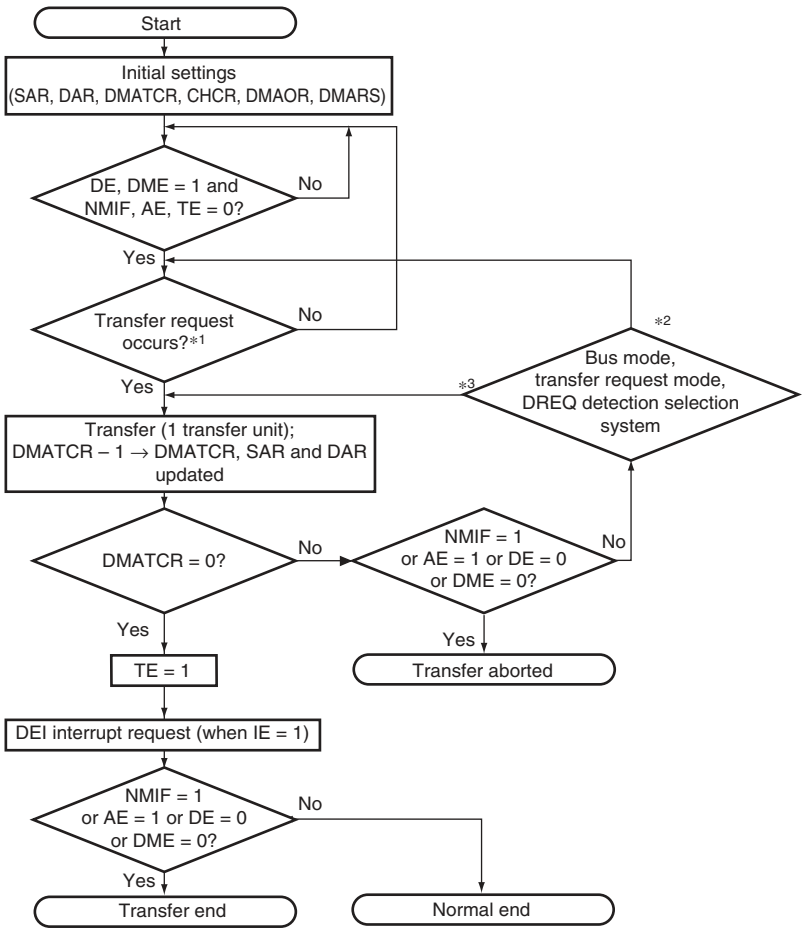
When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. In bus mode, burst mode or cycle steal mode can be selected.

10.4.1 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 (depending on the TS0 and TS1 settings). 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 have 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 10.2 shows a flowchart of this procedure.



Notes: 1. In auto-request mode, transfer begins when the NMIF, AE, and TE bits are all 0 and the DE and DME bits are set to 1.
2. DREQ = level detection in burst mode (external request) or cycle-steal mode.
3. DREQ = edge detection in burst mode (external request), or auto-request mode in burst mode.

Figure 10.2 DMA Transfer Flowchart

10.4.2 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 RS3 to RS0 bits in CHCR0 to CHCR3, and DMARS0 to DMARS2.

(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 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

In this mode, a transfer is performed at the request signals (DREQ0 and DREQ1) of an external device. This mode is valid only in channel 0 and channel 1. Choose one of the modes shown in table 10.3 according to the application system. 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 a request at the DREQ input.

Table 10.3 Selecting External Request Modes with RS Bits

RS3	RS2	RS1	RS0	Address Mode	Source	Destination
0	0	0	0	Dual address mode	Any	Any
		1	0	Single address mode	External memory, memory-mapped external device	External device with DACK
			1		External device with DACK	External memory, memory-mapped external device

Choose to detect DREQ by either the edge or level of the signal input with the DL bit and DS bit in CHCR_0 and CHCR_1 as shown in table 10.4. The source of the transfer request does not have to be the data transfer source or destination.

Table 10.4 Selecting External Request Detection with DL, DS Bits

CHCR_0 or 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 10.5 Selecting External Request Detection with DO Bit

CHCR_0 or 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 comprise the transmit data empty transfer request and receive data full transfer request from the ADC set by CHCR0 to CHCR5 and the SCIF0, SCIF1, MMC, USBF, SIM, SIOF0, SIOF1, and SDHI set by DMARS0/1/2, and the compare-match timer transfer request from the CMT (channels 0 to 4).

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 SCIF0 is set as the transfer request, the transfer destination must be the SCIF0's transmit data register. Likewise, when receive data full transfer request of the SCIF0 is set as the transfer request, the transfer source must be the SCIF0's receive data register. These conditions also apply to the SIOF1, MMC, USBF, SIM, SIOF0, SIOF1, and SDHI. When the ADC is set as the transfer request, the transfer source must be the A/D data register. Any address can be specified for data source and destination, when transfer request is generated by the CMT (channels 0 to 4).

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 10.6 Selecting On-Chip Peripheral Module Request Modes with RS3 to RS0 Bits

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Source	Destination	Bus Mode
RS[3:0]	MID	RID					
1000	001000	01	SCIF0 transmitter	TXI0 (transmit FIFO data empty interrupt)	Any	SCFTDR0	Cycle steal
		10	SCIF0 receiver	RXI0 (receive FIFO data full interrupt)	SCFRDR0	Any	Cycle steal
	001010	01	SCIF1transmitter	TXI1 (transmit FIFO data empty interrupt)	Any	SITDR	Cycle steal
		10	SCIF1 receiver	RXI1 (receive FIFO data full interrupt)	SCFRDR1	Any	Cycle steal
	000000	11	CMT (channel 0)	Compare-match transfer request	Any	Any	Cycle steal/ burst
	000001	11	CMT (channel 1)	Compare-match transfer request	Any	Any	Cycle steal/ burst
	000010	11	CMT (channel 2)	Compare-match transfer request	Any	Any	Cycle steal/ burst
	000011	11	CMT (channel 3)	Compare-match transfer request	Any	Any	Cycle steal/ burst
	000100	11	CMT (channel 4)	Compare-match transfer request	Any	Any	Cycle steal/ burst

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Source	Destination	Bus Mode
RS[3:0]	MID	RID					
1000	100000	11	USBF transmitter	Transmit data empty request	Any	EPDR2	Cycle steal
		00	USBF receiver	Transmit data full request	EPDR1	Any	Cycle steal
	101000	01	SIM transmitter	TXI (transmit data empty)	Any	SCTDR	Cycle steal
		10	SIM receiver	RXI (receive data full)	SCRDR	Any	Cycle steal
	101010	00	MMC transmitter	Receive data empty request	Any	Data register	Cycle steal
			MMC receiver	Receive data full request	Data register	Any	Cycle steal
	101100	01	SIOF0 transmitter	TXI0 (transmit FIFO data empty)	Any	SITDR0	Cycle steal
		10	SIOF0 receiver	RXI0 (receive FIFO data full)	SIRDR0	Any	Cycle steal
	101101	01	SIOF1 transmitter	TXI1 (transmit FIFO data empty)	Any	SITDR1	Cycle steal
		10	SIOF1 receiver	RXI1 (receive FIFO data full)	SIRDR0	Any	Cycle steal
	110000	01	SD transmitter	Transmit data empty request	Any	Data register	Cycle steal
		10	SD receiver	Receive data full request	Data register	Any	Cycle steal
1110	—	—	ADC	ADI (A/D conversion end)	ADDR	Any	Cycle steal

10.4.3 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 PR1 and PR0 bits 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 PR1 and the PR0 bits in DMAOR.

(2) Round-Robin Mode

In round-robin mode each time data of one transfer unit (word, byte, longword, or 16-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 10.3. The priority of round-robin mode is CH0 > CH1 > CH2 > CH3 > CH4 > CH5 immediately after a reset. When round-robin mode is specified, the same bus mode, either cycle steal mode or burst mode, must be specified for all of the channels.

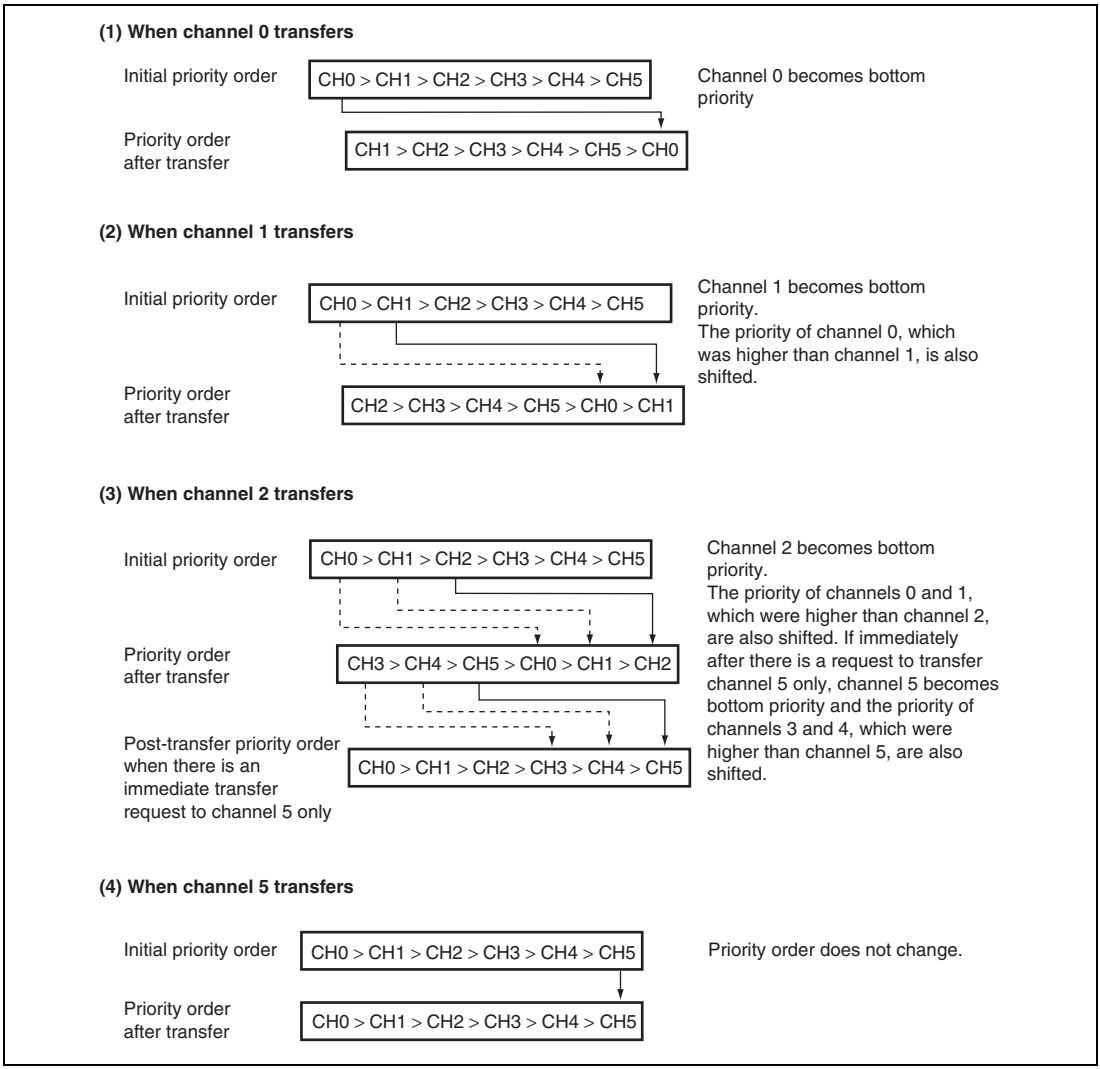


Figure 10.3 Round-Robin Mode

Figure 10.4 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.

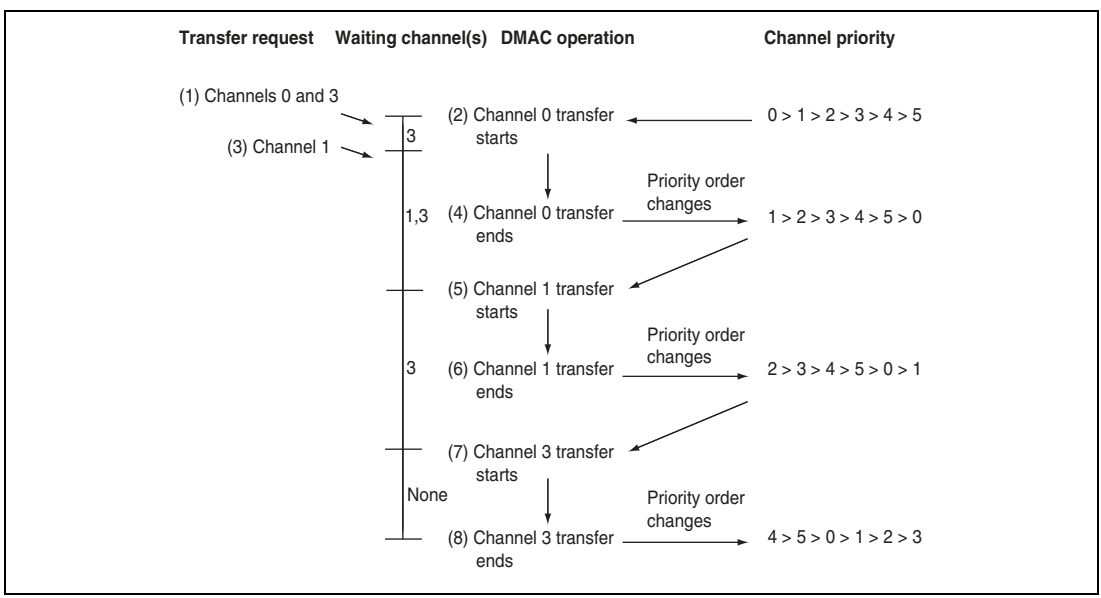


Figure 10.4 Changes in Channel Priority in Round-Robin Mode

10.4.4 DMA Transfer Types

DMA transfer has two types; single address mode transfer and 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 10.7.

Table 10.7 Supported DMA Transfers

	Destination				
Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Peripheral Module**	**X/Y Memory U Memory**
External device with DACK	Not available	Dual, single	Dual, single	Not available	Not available
External memory	Dual, single	Dual	Dual	Dual	Dual
Memory-mapped external device	Dual, single	Dual	Dual	Dual	Dual
On-chip peripheral module	Not available	Dual	Dual	Dual	Dual
X/Y memory	Not available	Dual	Dual	Dual	Dual

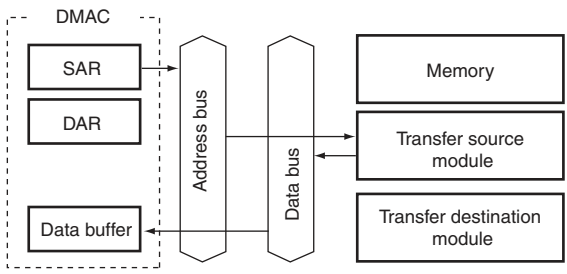
Notes: 1. Dual: Dual address mode
2. Single: Single address mode
3. For on-chip peripheral modules, 16-byte transfer is available only by registers which can be accessed in longword units.

(1) Address Modes

(a) Dual Address Mode

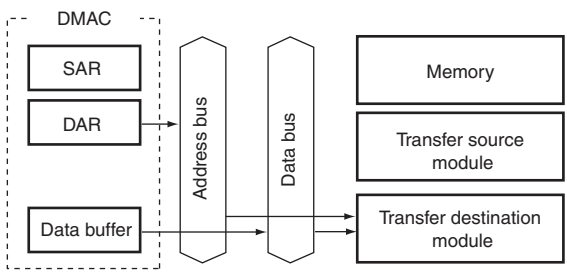
In dual address mode, both the transfer source and destination are accessed by an address. The source and destination can be located externally or internally.

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 10.5, 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 10.5 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. The channel control register (CHCR) can specify whether the DACK is output in read cycle or write cycle.

Figure 10.6 shows an example of DMA transfer timing in dual address mode.

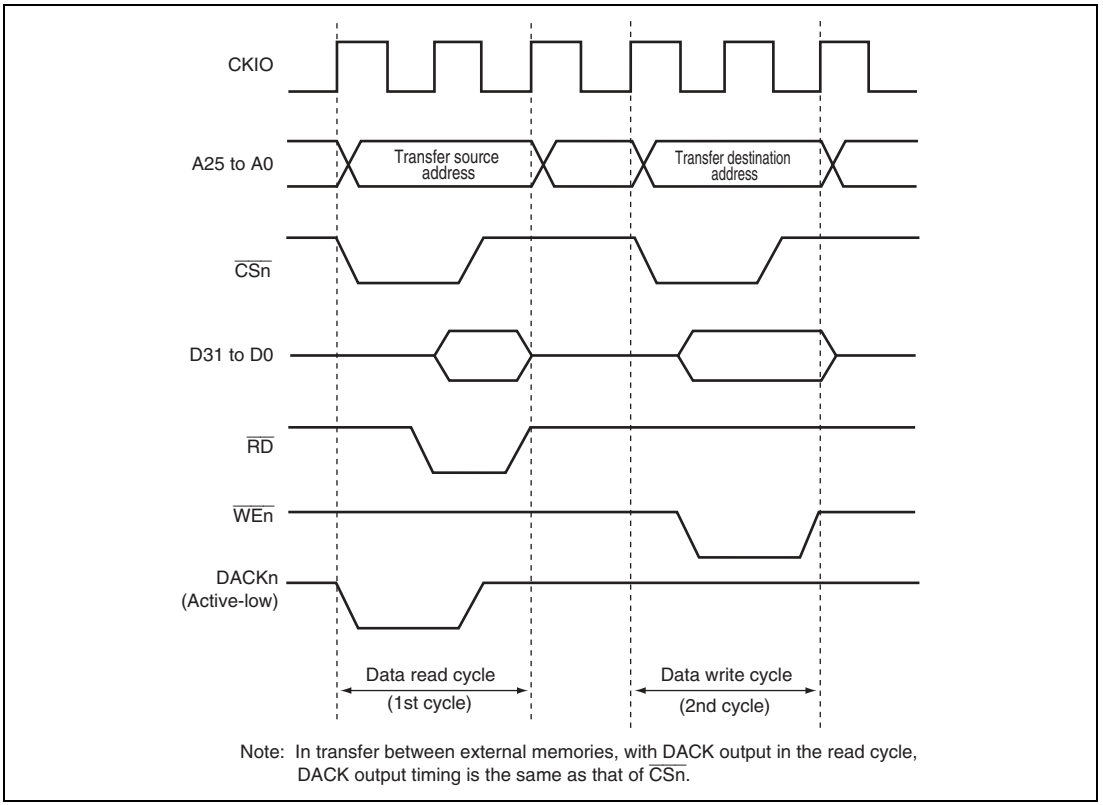


Figure 10.6 Example of DMA Transfer Timing in Dual Mode (Source: Ordinary Memory, Destination: Ordinary Memory)

(b) Single Address Mode

In single address mode, either the transfer source or transfer destination peripheral device is accessed (selected) by means of the DACK signal, and the other device is accessed by an address. In this mode, the DMAC performs one DMA transfer in one bus cycle, accessing one of the external devices by outputting the DACK transfer request acknowledge signal to it, and at the same time outputting an address to the other device involved in the transfer. For example, in the case of transfer between external memory and an external device with DACK shown in figure 10.7, when the external device outputs data to the data bus, that data is written to the external memory in the same bus cycle.

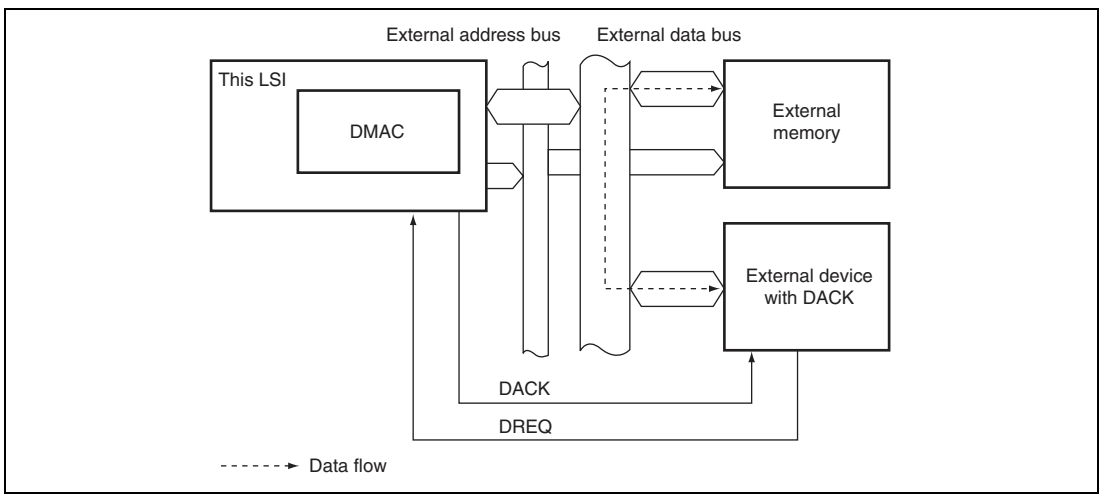


Figure 10.7 Data Flow in Single Address Mode

Two kinds of transfer are possible in single address mode: (1) transfer between an external device with DACK and a memory-mapped external device, and (2) transfer between an external device with DACK and external memory. In both cases, only the external request signal (DREQ) is used for transfer requests.

Figure 10.8 shows an example of DMA transfer timing in single address mode.

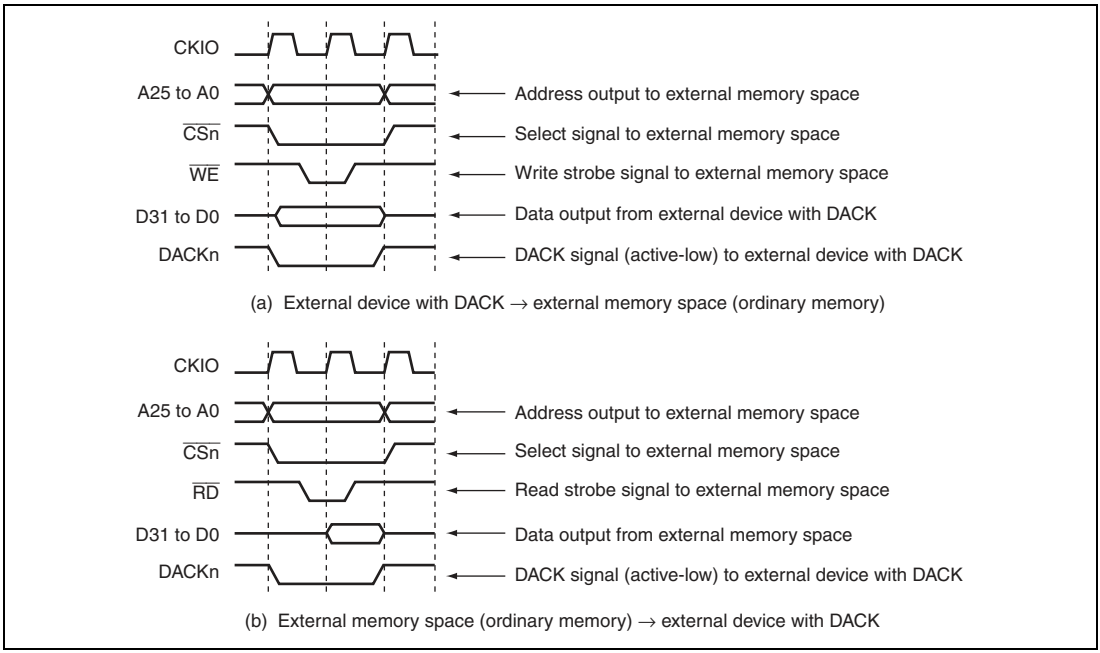


Figure 10.8 Example of DMA Transfer Timing in Single Address Mode

(2) Bus Modes

There are two bus modes: cycle steal mode and burst mode. Select the mode in the TB bits in the channel control register (CHCR).

(a) Cycle-Steal Mode

- Normal mode

  In cycle-steal normal mode, the bus mastership is given to another bus master after a one-transfer-unit (byte, word, longword, or 16-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 10.9 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

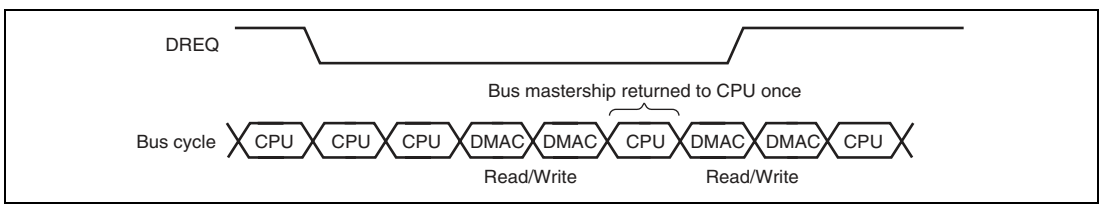


Figure 10.9 DMA Transfer Example in Cycle-Steal Normal Mode (Dual Address, DREQ Low Level Detection)

- Intermittent mode 16 and intermittent mode 64

  In intermittent mode of cycle steal, the DMAC returns the bus mastership to other bus master whenever a unit of transfer (byte, word, longword, or 16-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 or 64 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 10.10 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

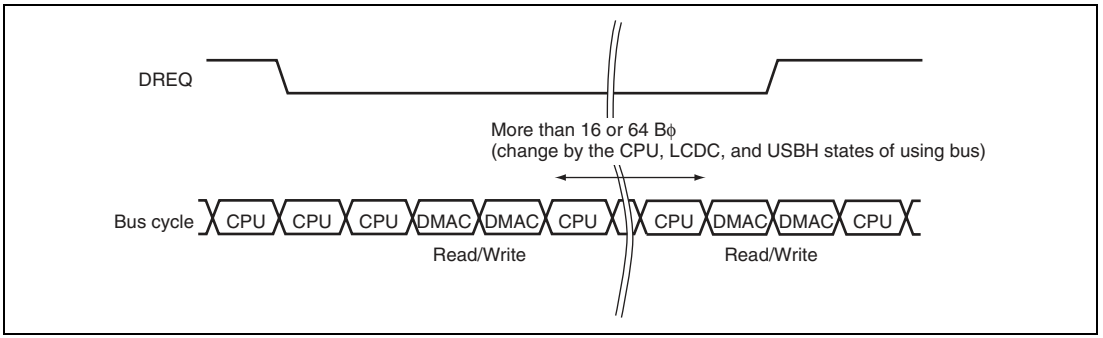


Figure 10.10 Example of DMA Transfer in Cycle Steal Intermittent Mode (Dual Address, DREQ Low Level Detection)

(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 for other than CMT (channels 0 to 4) when the on-chip peripheral module is the transfer request source.

Figure 10.11 shows DMA transfer timing in burst mode.

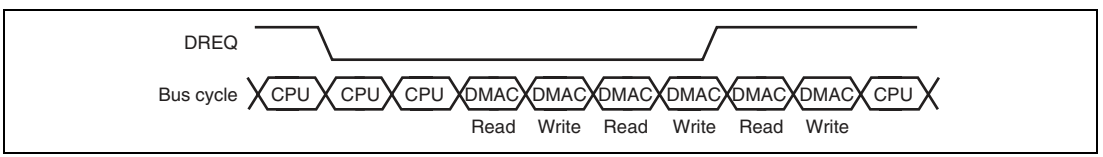


Figure 10.11 DMA Transfer Example in Burst Mode (Dual Address, DREQ Low Level Detection)

(3) Relationship between Request Modes and Bus Modes by DMA Transfer Category

Table 10.8 shows the relationship between request modes and bus modes by DMA transfer category.

Table 10.8 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	8/16/32/128	0,1
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128	0, 1
	External memory and external memory	All*1	B/C	8/16/32/128	0 to 5*5
	External memory and memory-mapped external device	All*1	B/C	8/16/32/128	0 to 5*5
	Memory-mapped external device and memory-mapped external device	All*1	B/C	8/16/32/128	0 to 5*5
	External memory and on-chip peripheral module	All*2	B/C*3	8/16/32/128*4	0 to 5*5
	Memory-mapped external device and on-chip peripheral module	All*2	B/C*3	8/16/32/128*4	0 to 5*5
	On-chip peripheral module and on-chip peripheral module	All*2	B/C*3	8/16/32/128*4	0 to 5*5
	X/Y memory and X/Y memory	All*1	B/C	8/16/32/128	0 to 5*5
	X/Y memory and memory-mapped external device	All*1	B/C	8/16/32/128	0 to 5*5
	X/Y memory and on-chip peripheral module	All*2	B/C*3	8/16/32/128*4	0 to 5*5
	X/Y memory and external memory	All*1	B/C	8/16/32/128	0 to 5*5
Single	External device with DACK and external memory	External	B/C	8/16/32	0, 1
	External device with DACK and memory-mapped external device	External	B/C	8/16/32	0, 1

B: Burst mode, C: Cycle steal mode

Notes: 1. External requests, auto requests, and on-chip peripheral module requests are all available. In the case of on-chip peripheral module requests, however, the CMT (channels 0 to 4) are only available.
2. External requests, auto requests, and on-chip peripheral module requests are all available. However, with the exception of the CMT (channels 0 to 4) as the transfer request source, the request source register must be designated as the transfer source or the transfer destination.
3. Only cycle steal except for the CMT (channels 0 to 4) as the transfer request source.
4. Access size permitted for the on-chip peripheral module register functioning as the transfer source or transfer destination.
5. 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), even though channel 1 is transferring in burst mode, if there is a transfer request to channel 0 which has 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 when the channel 0 transfer with a higher priority has completely finished.

If channel 0 is operating in cycle steal mode, immediately after channel 0 with a higher priority completes the transfer of one transfer unit, the channel 1 transfer will begin again without releasing the bus mastership. Transfer will then switch between the two in the order of channel 0, channel 1, channel 0, and channel 1. For the bus state, the CPU cycle after cycle steal mode transfer finishes is replaced with a burst mode transfer cycle (hereafter referred to as burst mode high-priority execution).

This example is illustrated in figure 10.12. If there are channels with conflicting burst transfers, transfer for the channel with the highest priority is performed first.

In DMA transfer for more than one channel, the DMAC does not give the bus mastership to the bus master until all conflicting burst transfers have finished.

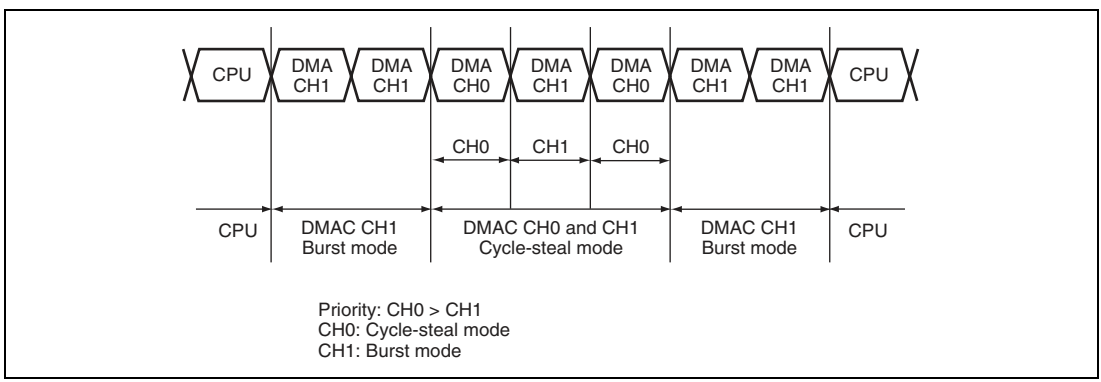


Figure 10.12 Bus State when Multiple Channels are Operating

In round-robin mode, the priority changes according to the specifications shown in figure 10.3. Note that a channel operating in cycle steal mode cannot be handled together with a channel operating in burst mode.

10.4.5 Number of Bus Cycle States and DREQ Pin Sampling Timing

(1) Number of Bus Cycle States

When the DMAC is the bus master, the number of bus cycle states is controlled by the bus state controller (BSC) in the same way as when the CPU is the bus master. For details, see section 9, Bus State Controller (BSC).

(2) $\overline{\text{DREQ}}$ Pin Sampling Timing

Figures 10.13, 10.14, 10.15, and 10.16 show the sample timing of the DREQ input in each bus mode, respectively.

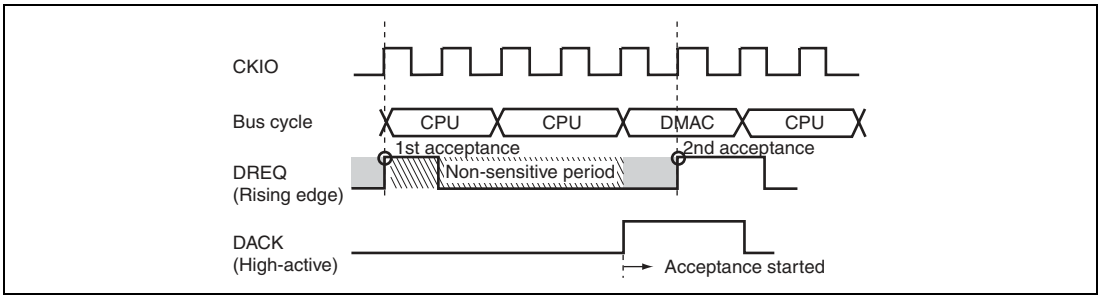


Figure 10.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection

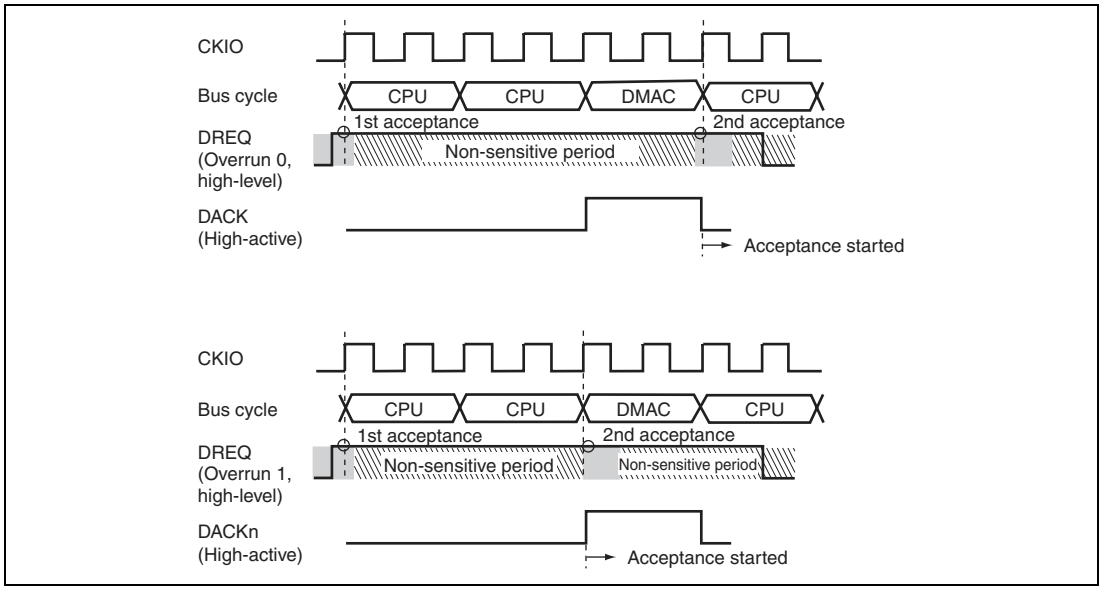


Figure 10.14 Example of DREQ Input Detection in Cycle Steal Mode Level Detection

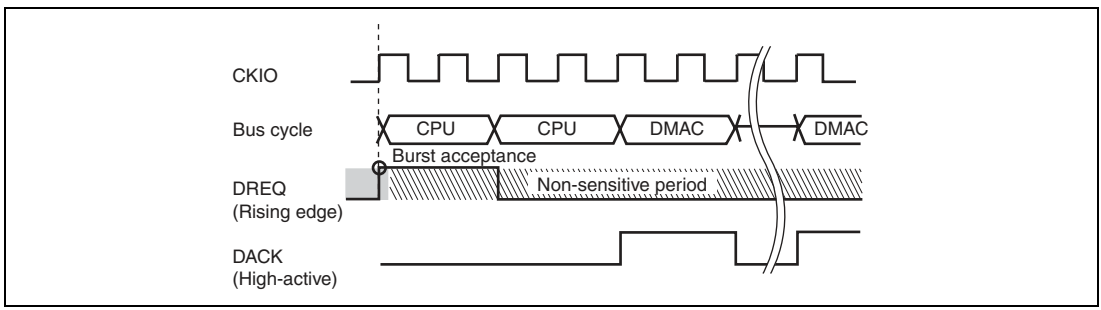


Figure 10.15 Example of DREQ Input Detection in Burst Mode Edge Detection

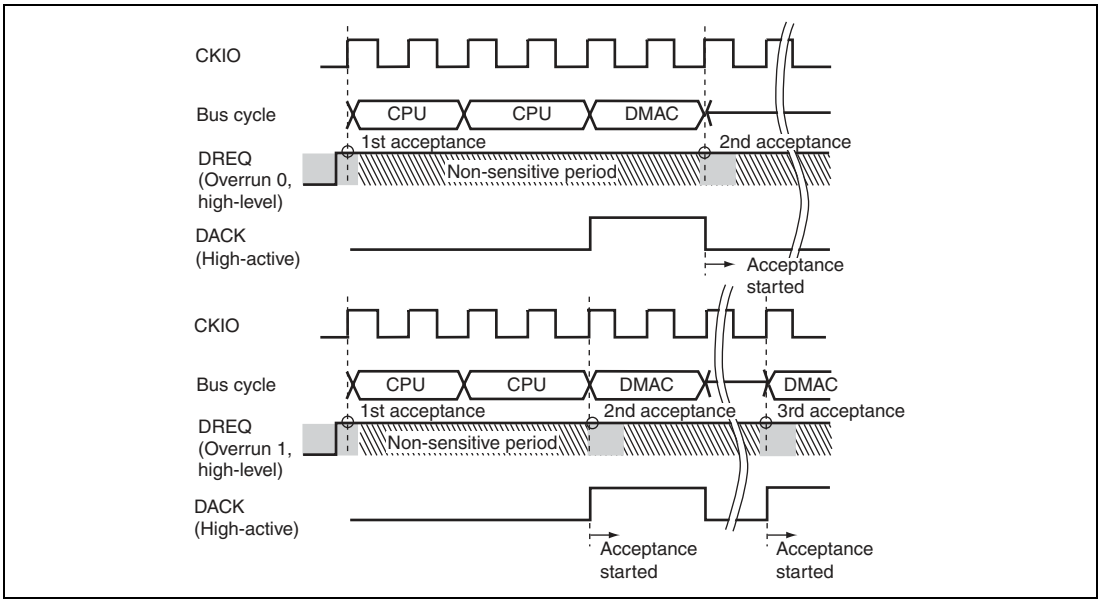


Figure 10.16 Example of DREQ Input Detection in Burst Mode Level Detection

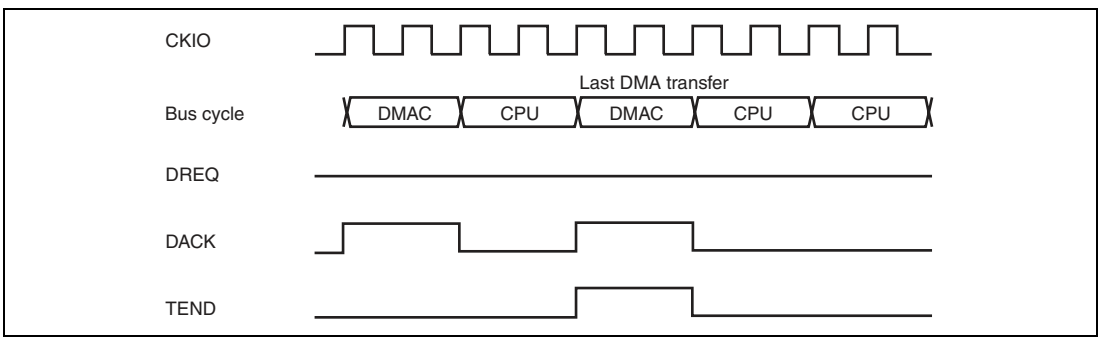
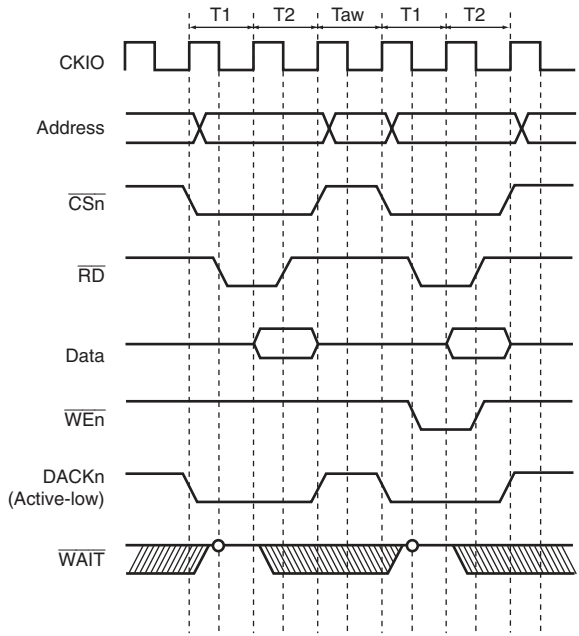


Figure 10.17 Example of DMA Transfer End in Cycle Steal Mode Level Detection

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 output is divided because of the data alignment. This example is illustrated in figure 10.18.



Note: The DACK is asserted for the last transfer unit of the DMA transfer. When the transfer unit is divided into several bus cycles and the CSn is negated between bus cycles, the DACK is also divided.

Figure 10.18 Example of BSC Ordinary Memory Access (No Wait, Idle Cycle 1, Longword Access to 16-Bit Device)

10.5 Usage Notes

Pay attentions to the following notes when the DMAC is used.

10.5.1 Notes on DACK Pin Output

When burst mode and cycle steal mode are simultaneously set in two or more channels, an additional DACK may be asserted at the end of burst transfer. This phenomenon will occur when all of the conditions described below are satisfied.

1. When the DMA transfer is simultaneously performed in two or more channels support both burst mode and cycle steal mode
2. When the channel to be used in burst mode is set to dual address mode, and DACK is output in data write cycle
3. When the DMAC cannot obtain the bus mastership consecutively even though a transfer demand of cycle steal has been received after the completion of burst transfer

This phenomenon is avoided by taking either of three measures shown below.

- Measure 1

  After confirming the completion of burst transfer (TE bit = 1), perform the DMA transfer of other cycle steal mode
- Measure 2

  The channel to be used in burst mode should not be set to output DACK in data write cycle
- Measure 3

  When the DMA transfer is simultaneously performed in two or more channels, set all of the channels to burst mode or cycle steal mode

10.5.2 Notes on the Cases When DACK is Divided

(1) Overview

When DACK is divided for output while the DMAC is accessing an external device, sampling of DREQ may be accepted once more during the access.

(2) Conditions and Phenomena

Conditions: In the cases when DACK is divided for output during external access, specifically, the following cases:

- 16-byte access
- 32-bit access in an 8-bit space
- 16-bit access in an 8-bit space
- 32-bit access in a 16-bit space,

Any one of the following inter-access idle cycle specifications has been made for that space:

- Idle between write cycles (IWW = 001 or more)
- Idle between read cycles in the same space (IWRRS = 001 or more)
- External wait masking (WM = 0)

Phenomena: For the access patterns above, the DREQ pin signal is detected with the timing shown in figures 10.19 and 10.21. For other access patterns, DREQ is detected normally as shown in figures 10.20 and 10.22.

(3) How to Avoid the Problem

For the external accesses under the conditions of 2 above, the problems can be avoided in the following way:

1. Detection of DREQ edges: During the bus cycle, input a DREQ edge (rising edge) only once at most.
2. When overrun-0 in DREQ level detection is specified: During the bus cycle, negate the DREQ input after detection of the first DACK output negation but before the second DACK output negation takes place.
3. When overrun-1 in DREQ level detection is specified: During the bus cycle, negate the DREQ input after detection of the first DACK output assertion but before the second DACK output assertion takes place.

(4) DREQ Pin Detection Timing Charts

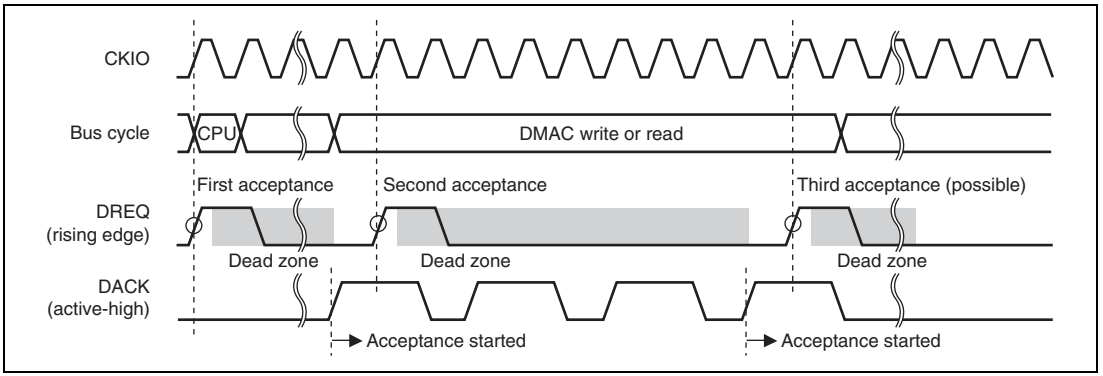


Figure 10.19 Timing of DREQ Input Detection by Edge Detection in Cycle Stealing Mode (DACK is Divided into Four due to Idle Cycle Insertion between Access Cycles and So DREQ Sampling is Accepted One Extra Time)

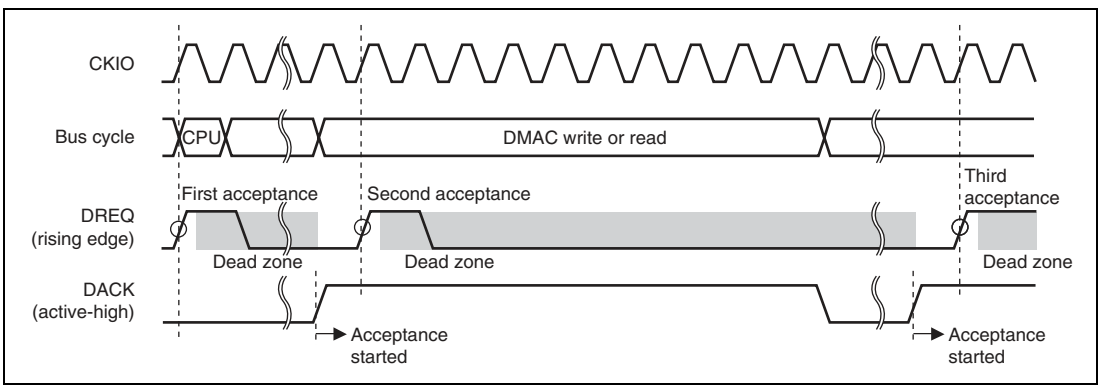


Figure 10.20 Timing of DREQ Input Detection by Edge Detection in Cycle Stealing Mode (DACK is Not Divided By Idle Cycle Insertion between Access Cycles and So DREQ Sampling is Accepted Normally)

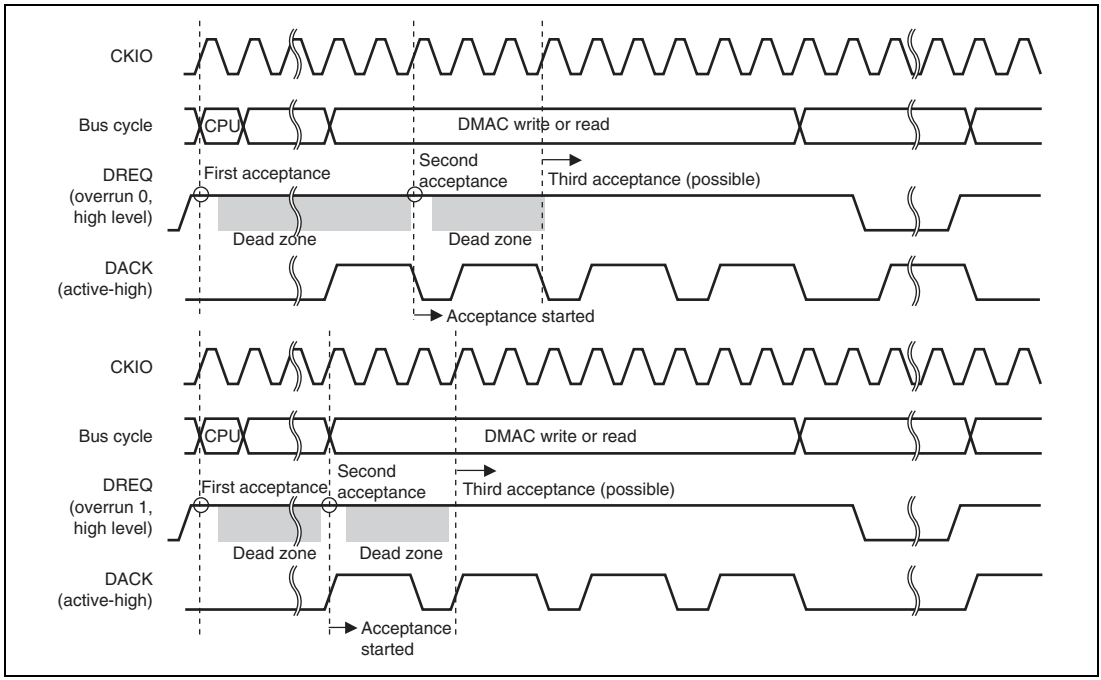


Figure 10.21 Timing of DREQ Input Detection by Level Detection in Cycle Stealing Mode (DACK is Divided into Four due to Idle Cycle Insertion between Access Cycles and So DREQ Sampling is Accepted One Extra Time)

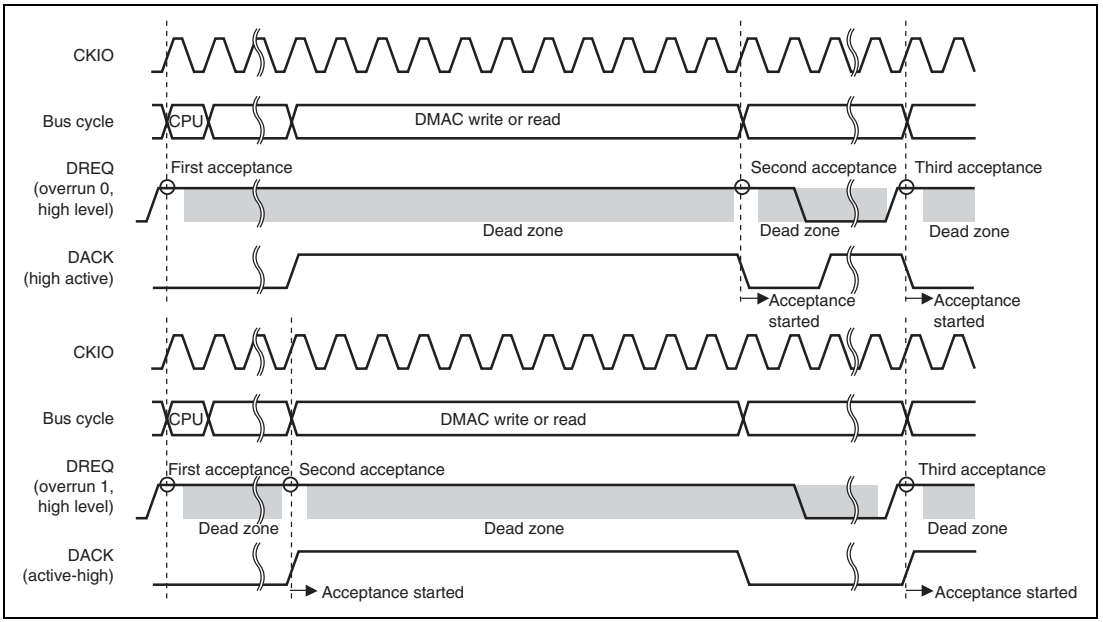


Figure 10.22 Timing of DREQ Input Detection by Edge Detection in Cycle Stealing Mode (DACK is Not Divided By Idle Cycle Insertion between Access Cycles and So DREQ Sampling is Accepted Normally)

10.5.3 Other Notes

1. Before making a transition to standby mode, either wait until DMA transfer finishes or suspend DMA transfer.
2. If an on-chip peripheral module whose clock supply is to be stopped by the module standby function is performing DMA transfer, either wait until DMA transfer finishes or suspend DMA transfer before making a transition to module standby mode.
3. Do not write to SAR, DAR, DMATCR, or DMARS during DMA transfer.

Concerning Above Notes 1 and 2:

DMA transfer end can be confirmed by checking whether the TE bit in CHCR is set to 1.

To suspend DMA transfer, clear the DE bit in CHCR to 0.

4. When reading these following flag bits 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.

   (1) DMA Channel Control Registers (CHCR_0 to CHCR_5) TE bit.

   (2) DMA Operation Register (DMAOR) AE bit and NMIF bit.

   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).

Section 11 Clock Pulse Generator (CPG)

This LSI has a clock pulse generator which generates an internal clock (Iϕ), a peripheral clock (Pϕ), and a bus clock (Bϕ). The clock pulse generator consists of oscillators, PLL circuits, and a divider.

11.1 Features

The CPG has the following features:

- Four clock modes: Selection of four clock modes according to the frequency range to be used and direct connection of crystal resonator or external clock input.
- Three clocks generated independently: An internal clock for the CPU and cache (Iϕ); a peripheral clock (Pϕ) for the on-chip supporting modules; and a bus clock (Bϕ=CKIO) for the external bus interface.
- Frequency change function: Internal and peripheral clock frequencies can be changed independently using the PLL circuit and divider circuit within the CPG. Frequencies are changed by software using frequency control register (FRQCR) settings.
- Power-down mode control: The clock can be stopped for sleep mode and standby mode and specific modules can be stopped using the module standby function.

A block diagram of the CPG is shown in figure 11.1.

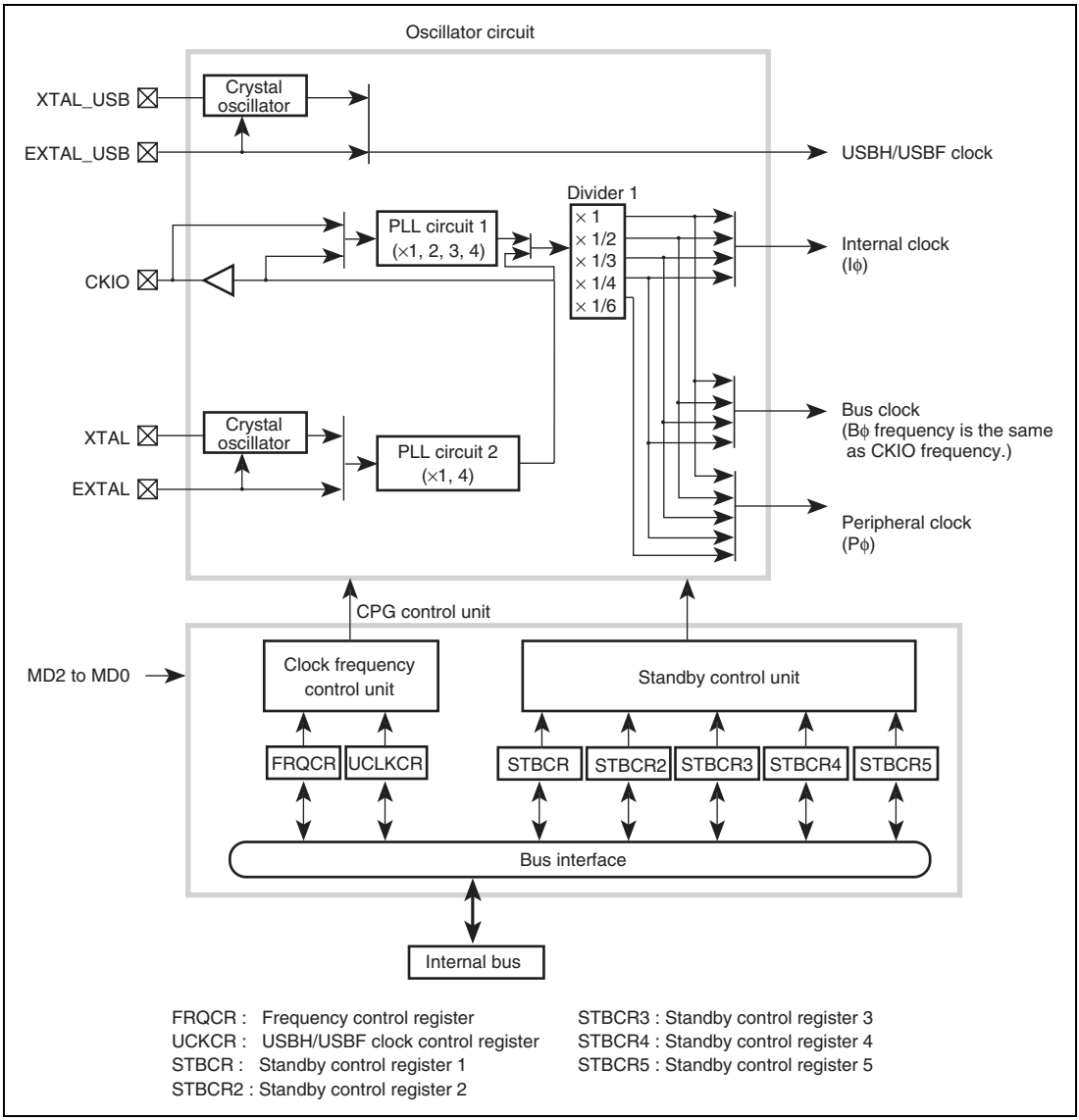


Figure 11.1 Block Diagram of CPG

The individual clock pulse generator blocks function as follows:

(1) PLL Circuit 1

PLL circuit 1 doubles, triples, quadruples, or leaves unchanged the input clock frequency from the CKIO terminal. The multiplication rate is set by the frequency control register. When this is done, the phase of the leading edge of the internal clock is controlled so that it will agree with the phase of the leading edge of the CKIO pin.

(2) PLL Circuit 2

PLL circuit 2 quadruples or leaves unchanged the input clock frequency from the crystal oscillator or EXTAL pin. The multiplication rate is set in the clock operating modes. The clock operating modes are set by pins MD0, MD1, and MD2. See table 11.2 for more information on clock operating modes.

(3) Crystal Oscillator

This oscillator is used when a crystal resonator is connected to the XTAL or EXTAL pin. It operates according to the clock operating mode setting.

(4) Divider 1

Divider 1 generates a clock at the operating frequency used by the internal or peripheral clock. The operating frequency of the internal clock (Iϕ) can be 1, 1/2, 1/3, or 1/4 times the output frequency of PLL circuit 1, as long as it stays at or above the clock frequency of the CKIO pin. The operating frequency of the peripheral clock (Pϕ) can be 1, 1/2, 1/3, 1/4, or 1/6 times the output frequency of PLL circuit 1 within 8.34 MHz $\leq$ Pϕ $\leq$ 33.34 MHz. The division ratio is set in the frequency control register.

(5) Clock Frequency Control Circuit

The clock frequency control circuit controls the clock frequency using the MD0, MD1, and MD2 pins and the frequency control register.

(6) Standby Control Circuit

The standby control circuit controls the state of the clock pulse generator and other modules during clock switching or in sleep or standby mode.

(7) Frequency Control Register

The frequency control register has control bits assigned for the following functions: clock output/non-output from the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the internal clock and the peripheral clock.

(8) Standby Control Register

The standby control register has bits for controlling the power-down modes. See section 13, Power-Down Modes, for more information.

(9) USBH/USBF Clock Control Register

The USBH/USBF clock control register specifies a signal source for generation of the USBH/USBF clock.

11.2 Input/Output Pins

Table 11.1 lists the CPG pins and their functions.

Table 11.1 Pin Configuration

Pin Name	Abbreviation	I/O	Description
Mode control pins	MD0	Input	Set the clock operating mode
	MD1	Input	Set the clock operating mode
	MD2	Input	Set the clock operating mode
Crystal I/O pins (clock input pins)	XTAL	Output	Connects a crystal resonator
	EXTAL	Input	Connects a crystal resonator. Also used to input an external clock.
Clock I/O pin	CKIO	I/O	Inputs or outputs an external clock
External clock pin for USBH/USBF	EXTAL_USB	Input	Inputs an external clock to USBH/USBF (48 MHz)
	XTAL_USB	Output	

Note: To prevent device malfunction, the value of the mode control pin is sampled only upon a power-on reset.

11.3 Clock Operating Modes

Table 11.2 shows the relationship between the mode control pins (MD2 to MD0) combinations and the clock modes. Table 11.3 shows the available combinations of the values of the clock modes and frequency control register (FRQCR).

Table 11.2 Clock Operating Modes

	Pin Values			Clock I/O		PLL2	PLL1	CKIO
Mode	MD2	MD1	MD0	Source	Output	On/Off	On/Off	Frequency
0	0	0	0	EXTAL	CKIO	ON (x 1)	ON (x 1, 2, 3, 4)	(EXTAL)
1	0	0	1	EXTAL	CKIO	ON (x 4)	ON (x 1, 2, 3, 4)	(EXTAL) x 4
2	0	1	0	Crystal resonator	CKIO	ON (x 4)	ON (x 1, 2, 3, 4)	(Crystal) x 4
7	1	1	1	CKIO	—	OFF	ON (x 1, 2, 3, 4)	(CKIO)

Mode 0: The LSI is supplied with a clock that is wave-formed by PLL circuit 2 after receiving an external clock from the EXTAL pin. The frequency of CKIO ranges from 24.00 to 66.67 MHz, because the input clock frequency ranges from 24.00 to 66.67 MHz.

Mode 1: The clock supplied to the internal circuitry in the LSI is generated by PLL circuit 2 quadrupling the frequency after receiving an external clock from the EXTAL pin. Therefore, the frequency of a clock generated outside the LSI can be lower. The frequency of CKIO ranges from 40.00 to 66.67 MHz, because an input clock with a frequency range of 10.00 to 16.67 MHz is used.

Mode 2: The clock is generated by an on-chip crystal oscillator, and its frequency is quadrupled by PLL circuit 2. Therefore, the frequency of a clock generated outside the LSI can be lower. The frequency of CKIO ranges from 40.00 to 66.67 MHz, because a crystal oscillator with a frequency range of 10.00 to 16.67 MHz is used.

Mode 7: The CKIO pin works as an input pin in this mode. The frequency of the external clock supplied to the LSI is multiplied by the setting ratio after the external clock is input via the CKIO pin and wave-formed by PLL circuit 1. This mode is suitable for connecting a synchronous DRAM, because the change in the load in the CKIO pin is controlled by PLL circuit 1.

Table 11.3 Possible Combination of Clock Mode and FRQCR Values

Mode	FRQCR Value	PLL Circuit 1	PLL Circuit 2	Clock Ratio* (I:B:P)	Frequency Range of Input Clock and Crystal Resonator	Frequency Range of CKIO Pin
0	1000	on (×1)	on (×1)	1:1:1	33.34 MHz	33.34 MHz
	1001	on (×1)	on (×1)	1:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1003	on (×1)	on (×1)	1:1:1/4	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1101	on (×2)	on (×1)	2:1:1	33.34 MHz	33.34 MHz
	1103	on (×2)	on (×1)	2:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1111	on (×2)	on (×1)	1:1:1	33.34 MHz	33.34 MHz
	1113	on (×2)	on (×1)	1:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1202	on (×3)	on (×1)	3:1:1	33.34 MHz	33.34 MHz
	1204	on (×3)	on (×1)	3:1:1/2	33.34 MHz to 44.45 MHz	33.34 MHz to 44.45 MHz
	1222	on (×3)	on (×1)	1:1:1	33.34 MHz	33.34 MHz
	1224	on (×3)	on (×1)	1:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1303	on (×4)	on (×1)	4:1:1	33.34 MHz	33.34 MHz
	1313	on (×4)	on (×1)	2:1:1	33.34 MHz	33.34 MHz
	1333	on (×4)	on (×1)	1:1:1	33.34 MHz	33.34 MHz
1, 2	1001	on (×1)	on (×4)	4:4:2	10.00 MHz to 16.67 MHz	40.00 MHz to 66.67 MHz
	1003	on (×1)	on (×4)	4:4:1	10.00 MHz to 16.67 MHz	40.00 MHz to 66.67 MHz
	1103	on (×2)	on (×4)	8:4:2	10.00 MHz to 16.67 MHz	40.00 MHz to 66.67 MHz
	1113	on (×2)	on (×4)	4:4:2	10.00 MHz to 16.67 MHz	40.00 MHz to 66.67 MHz
	1204	on (×3)	on (×4)	12:4:2	10.00 MHz to 16.67 MHz	40.00 MHz to 66.67 MHz
	1224	on (×3)	on (×4)	4:4:2	10.00 MHz to 16.67 MHz	40.00 MHz to 66.67 MHz

Mode	FRQCR Value	PLL Circuit 1	PLL Circuit 2	Clock Ratio* (I:B:P)	Frequency Range of Input Clock and Crystal Resonator	Frequency Range of CKIO Pin
7	1000	on (×1)	OFF	1:1:1	33.34 MHz	33.34 MHz
	1001	on (×1)	OFF	1:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1003	on (×1)	OFF	1:1:1/4	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1101	on (×2)	OFF	2:1:1	33.34 MHz	33.34 MHz
	1103	on (×2)	OFF	2:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1111	on (×2)	OFF	1:1:1	33.34 MHz	33.34 MHz
	1113	on (×2)	OFF	1:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1202	on (×3)	OFF	3:1:1	33.34 MHz to 44.45 MHz	33.34 MHz to 44.45 MHz
	1204	on (×3)	OFF	3:1:1/2	33.34 MHz to 44.45 MHz	33.34 MHz to 44.45 MHz
	1222	on (×3)	OFF	1:1:1	33.34 MHz	33.34 MHz
	1224	on (×3)	OFF	1:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1303	on (×4)	OFF	4:1:1	33.34 MHz	33.34 MHz
	1313	on (×4)	OFF	2:1:1	33.34 MHz	33.34 MHz
	1333	on (×4)	OFF	1:1:1	33.34 MHz	33.34 MHz

Notes: * The input clock is 1.
Maximum frequency: Iϕ = 133.34 MHz, Bϕ (CKIO) = 66.67 MHz, Pϕ = 33.34 MHz

1. Use the CKIO frequency within 33.34 MHz ≤ CKIO ≤ 66.67 MHz.
2. The input to divider 1 is the output of PLL circuit 1.
3. Use the internal clock frequency within 33.34 MHz ≤ Iϕ ≤ 133.34 MHz.
   The internal clock frequency is the product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1 selected by the STC bit in FRQCR, and the division ratio selected by the IFC bit in FRQCR.
   Do not set the internal clock frequency lower than the CKIO pin frequency.
4. Use the peripheral clock frequency within 8.34 MHz ≤ Pϕ ≤ 33.34 MHz.
   The peripheral clock frequency is the product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1 selected by the STC bit in FRQCR, and the division ratio selected by the PFC bit in FRQCR.
   Do not set the peripheral clock frequency higher than the frequency of the CKIO pin.
5. × 1, × 2, × 3, or × 4 can be used as the multiplication ratio of PLL circuit 1. × 1, × 1/2, × 1/3, or × 1/4can be selected as the division ratio of an internal clock. × 1, × 1/2, × 1/3, × 1/4, or
   × 1/6 can be selected as the division ratio of a peripheral clock. Set the rate in FRQCR.

11.4 Register Descriptions

The CPG has the following registers. Refer to section 37, List of Registers, for more details on the addresses and access size of these registers.

- Frequency control register (FRQCR)
- USBH/USBF clock control register (UCLKCR)

11.4.1 Frequency Control Register (FRQCR)

The frequency control register (FRQCR) is a 16-bit readable/writable register used to specify whether a clock is output from the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the internal clock and the peripheral clock.

Only word access can be used on the FRQCR register. FRQCR is initialized by a power-on reset, but not initialized by a power-on reset at the WDT overflow. FRQCR retains its value in a manual reset and in standby mode.

The write values to bits 14, 13, 11, 10, 7, 6, and 3 should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
15	PLL2EN	0	R/W	PLL2 Enable PLL2EN specifies whether make the PLL circuit 2 ON in clock operating mode 7. PLL circuit 2 is ON in clock operating modes other than mode 7 regardless of the PLL2EN setting. 0: PLL circuit 2 is OFF 1: PLL circuit 2 is ON
14, 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	CKOEN	1	R/W	Clock Output Enable CKOEN specifies whether a clock is output from the CKIO pin or the CKIO pin is placed in the level-fixed state in the standby mode, CKIO pin is fixed at low during STATUS 1 = L, and STATUS0 = H, when CKOEN is set to 0. Therefore, the malfunction of an external circuit because of an unstable CKIO clock in releasing the standby mode can be prevented. The CKIO pin becomes to input pin regardless of the value of the CKOEN bit in clock operating mode 7. 0: CKIO pin goes to low level state in standby mode 1: Clock is output from CKIO pin
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	STC1	0	R/W	Frequency Multiplication Ratio of PLL Circuit 1
8	STC0	0	R/W	00: × 1 time 01: × 2 times 10: × 3 times 11: × 4 times
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	IFC1	0	R/W	Internal Clock Frequency Division Ratio
4	IFC0	0	R/W	These bits specify the frequency division ratio of the internal clock (Iϕ) with respect to the output frequency of PLL circuit 1. 00: × 1 time 01: × 1/2 time 10: × 1/3 time 11: × 1/4 time
3	—	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
2	PFC2	0	R/W	Peripheral Clock Frequency Division Ratio
1	PFC1	1	R/W	These bits specify the division ratio of the peripheral clock (Pϕ) frequency with respect to the output frequency of PLL circuit 1.
0	PFC0	1	R/W	000: × 1 time 001: × 1/2 time 010: × 1/3 time 011: × 1/4 time 100: × 1/6 time Other than above: Reserved (setting prohibited)

11.4.2 USBH/USBF Clock Control Register (UCLKCR)

The USBH/USBF clock control register is an 8-bit readable/writable register. UCLKCR is initialized to H'60 by a power-on reset. Word-size access is used to write to this register. This writing should be performed with H'A5 in the upper byte and the write data in the lower byte.

Bit	Bit Name	Initial Value	R/W	Description
7	USSCS2	0	R/W	Source Clock Select
6	USSCS1	1	R/W	These bits select the source clock.
5	USSCS0	1	R/W	000: Clock stopped
				001: Setting prohibited
				010: Setting prohibited
				011: Initial value
				(To run the USBH/USB, however, change the setting to "110: EXTAL_USB" or "111: USB crystal resonator".)
				100: Setting prohibited
				101: Setting prohibited
				110: EXTAL_USB
				111: USB crystal resonator
4	USSTB	0	R/W	Standby USB Crystal
				Specifies stop or operation of the USB crystal oscillator in standby mode.
				0: USB crystal oscillator stops in standby mode when the STBXTL bit (bit 4) in the STBCR register is 0.
				1: USB crystal oscillator continues operating in standby mode
3 to 0	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0.

11.5 Changing Frequency

The frequency of the internal clock and peripheral clock can be changed either by changing the multiplication rate of PLL circuit 1 or by changing the division rates of divider 1. All of these are controlled by software through FRQCR. The methods are described below.

11.5.1 Changing Multiplication Rate

A PLL settling time is required when the multiplication rate of PLL circuit 1 is changed. The on-chip WDT counts the settling time.

1. In the initial state, the multiplication rate of PLL circuit 1 is 1.
2. Set a value that will become the specified oscillation settling time in the WDT and stop the WDT. The following must be set:
   TME bit in WTCSR = 0: WDT stops
   CKS2 to CKS0 bits in WTCSR: Division ratio of WDT count clock
   WTCNT: Initial counter value
3. Set the desired value in the STC1 and STC0 bits. The division ratio can also be set in the IFC1 and IFC0 bits and PFC2 to PFC0 bits.
4. The processor pauses internally and the WDT starts incrementing. The internal and peripheral clocks both stop and the WDT is supplied with the clock. The clock will continue to be output at the CKIO pin.
5. Supply of the clock that has been set begins at WDT count overflow, and the processor begins operating again. The WDT stops after it overflows.

11.5.2 Changing Division Ratio

The WDT will not count unless the multiplication rate is changed simultaneously.

1. In the initial state, IFC1 and IFC0 = 00 and PFC2 to PFC0 = 011.
2. Set the IFC1, IFC0, and PFC2 to PFC0 bits to the new division ratio. The values that can be set are limited by the clock mode and the multiplication rate of PLL circuit 1. Note that if the wrong value is set, the processor will malfunction.
3. The clock is immediately supplied at the new division ratio.

11.6 Usage Notes

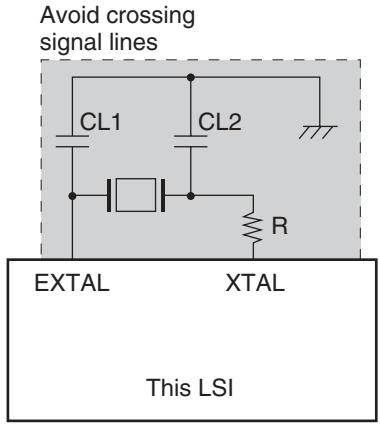
Note the following when using the USBH and USBF.

1. When the USBH and USBF are not used, it is recommended that UCLKCR should be cleared to H'00 to halt the clock.
2. Halt the USBH and USBF modules before changing the value of UCLKCR. This is done by selecting the "Clock stopped" setting with the module stop bit 31 (USBH module stop) and module stop bit 30 (USBF module stop) in STBCR3.
3. UCLKCR is initialized only by a power-on reset. In a manual reset, it retains its current set values.
4. When using the USBH/USBF, be sure to set the peripheral clock (Pϕ) to a frequency higher than 13 MHz.
5. When using the USBH, be sure to set the bus clock (Bϕ) to a frequency higher than 32 MHz.

11.7 Notes on Board Design

(1) When Using an External Crystal Resonator

Place the crystal resonator, capacitors CL1 and CL2, and damping resistor R close to the EXTAL and XTAL pins. To prevent induction from interfering with correct oscillation, use a common grounding point for the capacitors connected to the resonator, and do not locate a wiring pattern near these components.



Note: The values for CL1, CL2, and the damping resistance should be determined after consultation with the crystal manufacturer.

Figure 11.2 Points for Attention when Using Crystal Resonator

(2) Bypass Capacitors

Insert a laminated ceramic capacitor as a bypass capacitor for each V_{SS}/V_{CC} pair. Mount the bypass capacitors to the power supply pins, and use components with a frequency characteristic suitable for the operating frequency of the LSI, as well as a suitable capacitance value.

(3) 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.

Connect the EXTAL pin to V_{CC} or V_{SS}Q and make the XTAL pin open in clock mode 7.

The analog power supply system of the PLL is sensitive to a noise. Therefore the system malfunction may occur by the intervention with other power supply. Do not supply the analog power supply with the same resource as the digital power supply of V_{CC} and V_{CC}Q.

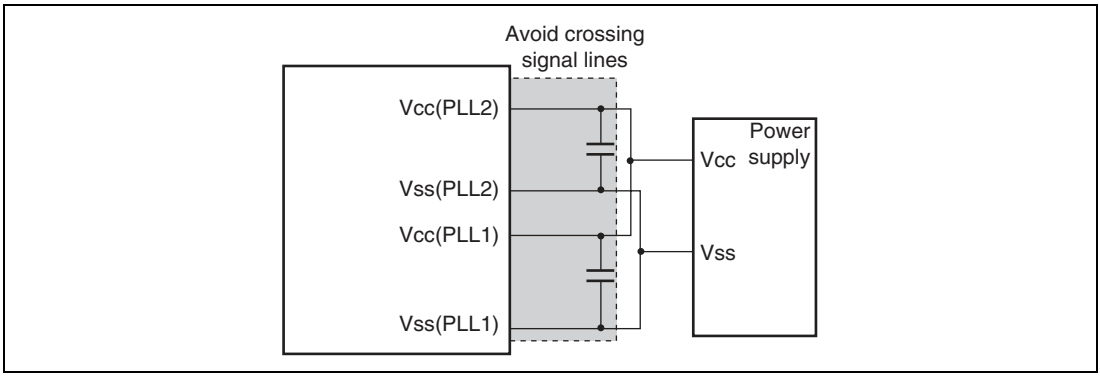


Figure 11.3 Points for Attention when Using PLL Oscillator Circuit

Section 12 Watchdog Timer (WDT)

This LSI includes the watchdog timer (WDT).

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 WDT is a single-channel timer that uses a peripheral clock as an input and counts the clock settling time when clearing software standby mode and temporary standbys, such as frequency changes. It can also be used as an interval timer.

12.1 Features

The WDT has the following features:

- Can be used to ensure the clock settling time: Use the WDT to cancel software standby mode and the temporary standbys which occur when the clock frequency is changed.
- Can switch between watchdog timer mode and interval timer mode.
- Generates internal resets in watchdog timer mode: Internal resets occur after counter overflow.
- An interrupt is generated in interval timer mode
  An interval timer interrupt is generated when the counter overflows.
- Choice of eight counter input clocks
  Eight clocks (×1 to ×1/4096) that are obtained by dividing the peripheral clock can be chosen.
- Choice of two resets
  Power-on reset and manual reset are available.

Figures 12.1 shows a block diagram of the WDT.

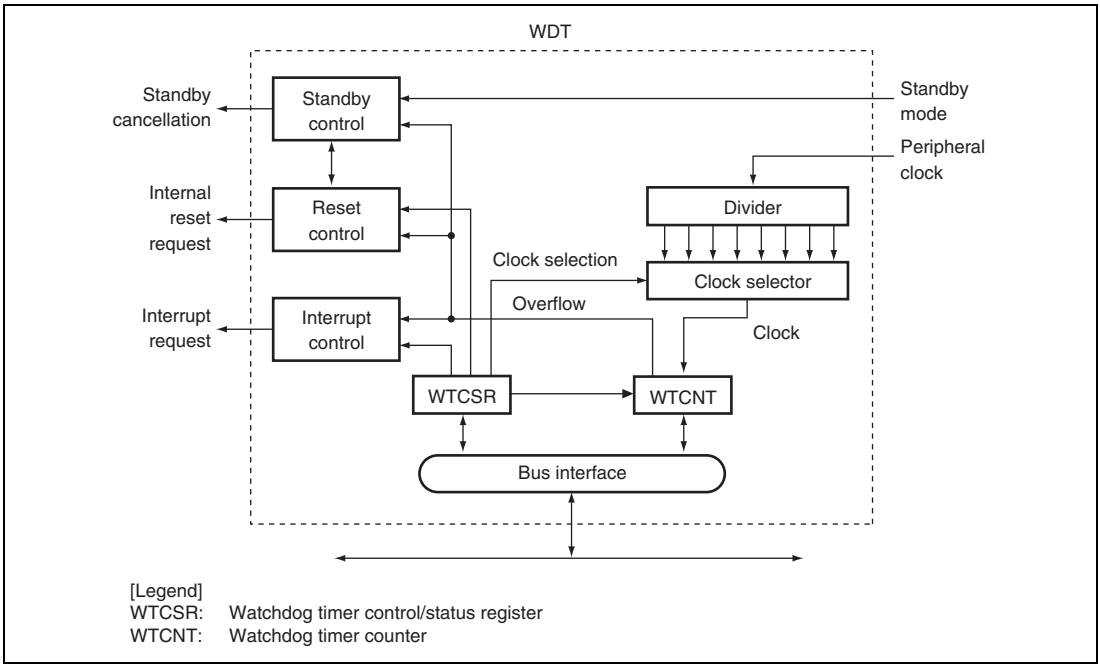


Figure 12.1 Block Diagram of WDT

12.2 Register Descriptions for WDT

The WDT has the following two registers. Refer to section 37, List of Registers, for more details on the addresses and states of these registers in each operating mode.

- Watchdog timer counter (WTCNT)
- Watchdog timer control/status register (WTCSR)

12.2.1 Watchdog Timer Counter (WTCNT)

WTCNT is an 8-bit readable/writable register that increments on the selected clock. When an overflow occurs, it generates a reset in watchdog timer mode and an interrupt in interval time mode. The WTCNT counter is not initialized by an internal reset due to the WDT overflow. The WTCNT counter is initialized to H'00 only by a power-on reset. Use a word access to write to the WTCNT counter, with H'5A in the upper byte. Use a byte access to read WTCNT.

Note: WTCNT differs from other registers in that it is more difficult to write to. See section 12.2.3, Notes on Register Access, for details.

12.2.2 Watchdog Timer Control/Status Register (WTCSR)

WTCSR is an 8-bit readable/writable register composed of bits to select the clock used for the count, bits to select the timer mode, and overflow flags. WTCSR holds its value in an internal reset due to the WDT overflow. WTCSR is initialized to H'00 only by a power-on reset.

When used to count the clock settling time for canceling a software standby, it retains its value after counter overflow. Use a word access to write to WTCSR, with H'A5 in the upper byte. Use a byte access to read WTCSR.

Note: WTCSR differs from other registers in that it is more difficult to write to. See section 12.2.3, Notes on Register Access, for details.

Bit	Bit Name	Initial Value	R/W	Description
7	TME	0	R/W	Timer Enable Starts and stops timer operation. Clear this bit to 0 when using the WDT in software standby mode or when changing the clock frequency. 0: Timer disabled: Count-up stops and WTCNT value is retained 1: Timer enabled
6	WT/IT	0	R/W	Timer Mode Select Selects whether to use the WDT as a watchdog timer or an interval timer. 0: Interval timer mode 1: Watchdog timer mode Note: If WT/IT is modified when the WDT is operating, the up-count may not be performed correctly.
5	RSTS	0	R/W	Reset Select Selects the type of reset when the WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored. 0: Power-on reset 1: Manual reset
4	WOVF	0	R/W	Watchdog Timer Overflow Indicates that the WTCNT has overflowed in watchdog timer mode. This bit is not set in interval timer mode. 0: No overflow 1: WTCNT has overflowed in watchdog timer mode
3	IOVF	0	R/W	Interval Timer Overflow Indicates that the WTCNT has overflowed in interval timer mode. This bit is not set in watchdog timer mode. 0: No overflow 1: WTCNT has overflowed in interval timer mode

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	These bits select the clock to be used for the WTCNT count from the eight types obtainable by dividing the peripheral clock (Pϕ). The overflow period that is shown inside the parenthesis in the table is the value when the peripheral clock (Pϕ) is 15 MHz.
0	CKS0	0	R/W	000: Pϕ (17 μs) 001: Pϕ /4 (68 μs) 010: Pϕ /16 (273 μs) 011: Pϕ /32 (546 μs) 100: Pϕ /64 (1.09 ms) 101: Pϕ /256 (4.36 ms) 110: Pϕ /1024 (17.48 ms) 111: Pϕ /4096 (69.91 ms) Note: If bits CKS2 to CKS0 are modified when the WDT is operating, the up-count may not be performed correctly. Ensure that these bits are modified only when the WDT is not operating.

12.2.3 Notes on Register Access

The watchdog timer counter (WTCNT) and watchdog timer control/status register (WTCSR) are more difficult to write to than other registers. The procedure for writing to these registers is given below.

- Writing to WTCNT and WTCSR

  These registers must be written by a word transfer instruction. They cannot be written by a byte or longword transfer instruction. When writing to WTCNT, set the upper byte to H'5A and transfer the lower byte as the write data, as shown in figure 12.3. When writing to WTCSR, set the upper byte to H'A5 and transfer the lower byte as the write data. This transfer procedure writes the lower byte data to WTCNT or WTCSR.

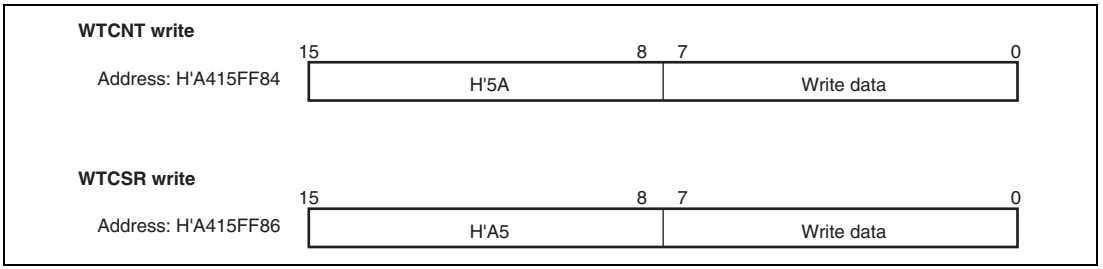


Figure 12.2 Writing to WTCNT and WTCSR

12.3 WDT Operation

12.3.1 Canceling Software Standbys

The WDT can be used to cancel software standby mode with an NMI interrupt or external interrupt (IRQ). The procedure is described below. (The WDT does not run when resets are used for canceling, so keep the $\overline{\text{RESETP}}$ pin low until the clock stabilizes.)

1. Before transition to software standby mode, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS2 to CKS0 bits in WTCSR and the initial values for the counter in the WTCNT counter. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. Move to software standby mode by executing a SLEEP instruction to stop the clock.
4. The WDT starts counting by detecting the edge change of the NMI signal.
5. When the WDT count overflows, the CPG starts supplying the clock and the processor resumes operation. The WOVF flag in WTCSR is not set when this happens.
6. Since the WDT continues counting from H'00, set the STBY bit in STBCR to 0 in the interrupt processing program and this will stop the WDT. When the STBY bit remains 1, the LSI again enters software standby mode when the WDT has counted up to H'80. This software standby mode can be canceled by a power-on reset.

12.3.2 Changing Frequency

To change the frequency used by the PLL, use the WDT. When changing the frequency only by switching the divider, do not use the WDT.

1. Before changing the frequency, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS2 to CKS0 bits in WTCSR and the initial values for the counter in the WTCNT counter. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. When the frequency control register (FRQCR) is written, the processor stops temporarily. The WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and the processor resumes operation. The WOVF flag in WTCSR is not set when this happens.
5. The counter stops at the values H'00.
6. Before changing WTCNT after the execution of the frequency change instruction, always confirm that the value of WTCNT is H'00 by reading WTCNT.

12.3.3 Using Watchdog Timer Mode

1. Set the WT/IT bit in WTCSR to 1, set the reset type in the RSTS bit, set the type of count clock in the CKS2 to CKS0 bits, and set the initial value of the counter in the WTCNT counter.
2. Set the TME bit in WTCSR to 1 to start the count in watchdog timer mode.
3. While operating in watchdog timer mode, rewrite the counter periodically to H'00 to prevent the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WTCSR to 1 and generates the type of reset specified by the RSTS bit. The counter then resumes counting.

12.3.4 Using Interval Timer Mode

When operating in interval timer mode, interval timer interrupts are generated at every overflow of the counter. This enables interrupts to be generated at set periods.

1. Clear the WT/IT bit in WTCSR to 0, set the type of count clock in the CKS2 to CKS0 bits, and set the initial value of the counter in the WTCNT counter.
2. Set the TME bit in WTCSR to 1 to start the count in interval timer mode.
3. When the counter overflows, the WDT sets the IOVF flag in WTCSR to 1 and an interval timer interrupt request (ITI) is sent to the INTC. The counter then resumes counting.

Section 13 Power-Down Modes

This LSI has four types of power-down modes: Sleep mode, software standby mode, module standby function, and hardware standby mode.

13.1 Features

- Supports sleep/software standby/module standby/hardware standby.

13.1.1 Power-Down Modes

This LSI has the following power-down modes and function:

- Sleep mode
- Software standby mode
- Module standby function (DSP, cache, TLB, X/Y memory, UBC, DMAC, H-UDI, and on-chip peripheral module)
- Hardware standby mode

Table 13.1 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 13.1 States of Power-Down Modes

		State						
Mode	**Transition Conditions**	**CPG**	**CPU**	**CPU Register**	**On-Chip Memory**	**On-Chip Peripheral Modules**	**External Memory**	**Canceling Procedure**
Sleep mode	Execute SLEEP instruction with STBY bit in STBCR cleared to 0	Runs	Halts	Held	Halts (contents remained)	Run	Auto-refreshing	• Interrupt • Reset
Software Standby mode	Execute SLEEP instruction with STBY bit in STBCR set to 1	Halts	Halts	Held	Halts (contents remained)	Halt*	Self-refreshing	• Interrupt (NMI, IRQ (edge detection), RTC, TMU, PINT • Reset
Module standby function	Set MSTP bit in STBCR to 1	Runs	Runs/ halts	Held	Specified module halts (contents remained)	Specified module halts	Auto-refreshing	• Clear MSTP bit to 0 • Power-on reset
Hardware standby mode	Set CA pin to low	Halts	Halts	Held	Held	Halt*	Self-refreshing	• Power-on reset

Note: * The RTC operates when the START bit in RCR2 is set to 1. For details, see section 17, Realtime Clock (RTC).

13.1.2 Reset

Resetting occurs when power is supplied, and when execution is started again from an initialized state. There are two types of reset: A power-on reset and a manual reset. In a power-on reset, all processing in execution is suspended, all unprocessed events are canceled, and reset processing starts immediately. On the other hand, processing to retain the contents of external memory is continued in a manual reset. The conditions for generating power-on and manual resets are as follows.

(1) Power-On Reset

1. Driving the $\overline{\text{RESETP}}$ pin low.
2. While the WT/IT bit in WTCSR is set to 1 and the RSTS bit is cleared to 0, the WDT starts counting and continues until it overflows.
3. Generation of the H-UDI reset (for details on the H-UDI reset, refer to section 36, User Debugging Interface (H-UDI)).

(2) Manual Reset

1. Driving the $\overline{\text{RESETM}}$ pin low.
2. While the WT/IT bit in WTCSR and the RSTS bit are set to 1, the WDT starts counting and continues until it overflows.

13.2 Input/Output Pins

Table 13.2 lists the pin configuration related to power-down modes.

Table 13.2 Pin Configuration

Pin Name	Abbreviation	I/O	Description
Status 1 output	STATUS1	Output	Operating state of the processor. HH: Reset HL: Sleep mode LH: Standby mode LL: Normal operation
Status 0 output	STATUS0		
Power-on reset input	$\overline{\text{RESETP}}$	Input	Power-on reset occurs at low-level.
Manual-reset input	$\overline{\text{RESETM}}$	Input	Manual reset occurs at low-level.
Chip active	CA	Input	Hardware standby mode entered at low-level.

13.3 Register Descriptions

There are following five registers related to power-down modes. Refer to section 37, List of Registers, for more details on the addresses and states of these registers in each operating mode.

- Standby control register (STBCR)
- Standby control register 2 (STBCR2)
- Standby control register 3 (STBCR3)
- Standby control register 4 (STBCR4)
- Standby control register 5 (STBCR5)

13.3.1 Standby Control Register (STBCR)

STBCR is an 8-bit readable/writable register that specifies the state of power-down modes.

Bit	Bit Name	Initial Value	R/W	Description
7	STBY	0	R/W	Standby Specifies transition to software standby mode. 0: Executing SLEEP instruction enters chip into sleep mode 1: Executing SLEEP instruction enters chip into software standby mode
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	STBXTL	0	R/W	Standby Crystal Specifies halt/operation of a crystal oscillator in standby mode. 0: Crystal oscillator is halted in standby mode 1: Crystal oscillator is operated continuously in standby mode

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	MSTP2	0	R/W	Module Stop Bit 2 When the MSTP2 bit is set to 1, the supply of the clock to the TMU is halted. 0: TMU operates 1: Clock supply to TMU halted
1	MSTP1	0	R/W	Module Stop Bit 1 When the MSTP1 bit is set to 1, the supply of the clock to the RTC is halted. 0: RTC operates 1: Clock supply to RTC halted
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

13.3.2 Standby Control Register 2 (STBCR2)

STBCR2 is an 8-bit readable/writable register that controls the operation of modules in power-down mode.

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP10	0	R/W	Module Stop Bit 10 When the MSTP10 bit is set to 1, the supply of the clock to the H-UDI is halted. 0: H-UDI operates 1: Clock supply to H-UDI halted
6	MSTP9	0	R/W	Module Stop Bit 9 When the MSTP9 bit is set to 1, the supply of the clock to the UBC is halted. 0: UBC operates 1: Clock supply to UBC halted

Bit	Bit Name	Initial Value	R/W	Description
5	MSTP8	0	R/W	Module Stop Bit 8 When the MSTP8 bit is set to 1, the supply of the clock to the DMAC is halted. 0: DMAC operates 1: Clock supply to DMAC halted
4	MSTP7	0	R/W	Module Stop Bit 7 When the MSTP7 bit is set to 1, the supply of the clock to the DSP is halted. 0: DSP operates 1: Clock supply to DSP halted
3	MSTP6	0	R/W	Module Stop Bit 6 When the MSTP6 bit is set to 1, the supply of the clock to the TLB is halted. 0: TLB operates 1: Clock supply to TLB halted
2	MSTP5	0	R/W	Module Stop Bit 5 When the MSTP5 bit is set to 1, the supply of the clock to the cache memory is halted. 0: Cache memory operates 1: Clock supply to cache memory halted
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	MSTP3	0	R/W	Module Stop Bit 3 When the MSTP3 bit is set to 1, the supply of the clock to the X/Y memory is halted. 0: X/Y memory operates 1: Clock supply to X/Y memory halted

13.3.3 Standby Control Register 3 (STBCR3)

STBCR3 is an 8-bit readable/writable register that controls the operation of modules in power-down mode.

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP37	0	R/W	Module Stop Bit 37 When the MSTP37 bit is set to 1, the supply of the clock to the SIOF1 is halted. 0: SIOF1 operates 1: Clock supply to SIOF1 halted
6	MSTP36	0	R/W	Module Stop Bit 36 When the MSTP36 bit is set to 1, the supply of the clock to the SIOF0 is halted. 0: SIOF0 operates 1: Clock supply to SIOF0 halted
5	MSTP35	0	R/W	Module Stop Bit 35 When the MSTP35 bit is set to 1, the supply of the clock to the CMT is halted. However, count-up operation is continued when the channel 5 is in the operation. 0: CMT operates 1: Clock supply to CMT halted
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	MSTP33	0	R/W	Module Stop Bit 33 When the MSTP33 bit is set to 1, the supply of the clock to the ADC is halted. 0: ADC operates 1: Clock supply to ADC halted
2	MSTP32	0	R/W	Module Stop Bit 32 When the MSTP32 bit is set to 1, the supply of the clock to the DAC is halted. 0: DAC operates 1: Clock supply to DAC halted

Bit	Bit Name	Initial Value	R/W	Description
1	MSTP31	0	R/W	Module Stop Bit 31 When the MSTP31 bit is set to 1, the supply of the clock to the USBH is halted. 0: USBH operates 1: Clock supply to USBH halted
0	MSTP30	0	R/W	Module Stop Bit 30 When the MSTP30 bit is set to 1, the supply of the clock to the USBF is halted. 0: USBF operates 1: Clock supply to USBF halted

13.3.4 Standby Control Register 4 (STBCR4)

STBCR4 is an 8-bit readable/writable register that controls the operation of modules in power-down mode.

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	MSTP45	0	R/W	Module Stop Bit 45 When the MSTP45 bit is set to 1, the supply of the clock to the PCC is halted. 0: PCC operates 1: Clock supply to PCC halted
4	MSTP44	0	R/W	Module Stop Bit 44 When the MSTP44 bit is set to 1, the supply of the clock to the I^2C is halted. 0: I^2C operates 1: Clock supply to I^2C halted

Bit	Bit Name	Initial Value	R/W	Description
3	MSTP43	0	R/W	Module Stop Bit 43 When the MSTP43 bit is set to 1, the supply of the clock to the MMC is halted. 0: MMC operates 1: Clock supply to MMC halted
2	MSTP42	0	R/W	Module Stop Bit 42 When the MSTP42 bit is set to 1, the supply of the clock to the SIM is halted. 0: SIM operates 1: Clock supply to SIM halted
1	MSTP41	0	R/W	Module Stop Bit 41 When the MSTP41 bit is set to 1, the supply of the clock to the SCIF1 is halted. 0: SCIF1 operates 1: Clock supply to SCIF1 halted
0	MSTP40	0	R/W	Module Stop Bit 40 When the MSTP40 bit is set to 1, the supply of the clock to the SCIF0 is halted. 0: SCIF0 operates 1: Clock supply to SCIF0 halted

13.3.5 Standby Control Register 5 (STBCR5)

STBCR5 is an 8-bit readable/writable register that controls the operation of modules in power-down 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	MSTP56	0	R/W	Module Stop Bit 56 When the MSTP56 bit is set to 1, the supply of the clock to the SDHI is halted. 0: Clock supply to SDHI halted 1: SDHI operates Note: On the models not having the SDHI, this bit is reserved and is always read as 0. The write value should always be 0.
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	MSTP54	0	R/W	Module Stop Bit 54 When the MSTP54 bit is set to 1, the supply of the clock to the TPU is halted. 0: TPU operates 1: Clock supply to TPU halted
3	—	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
2	MSTP52	0	R/W	Module Stop Bit 52 When the MSTP52 bit is set to 1, the supply of the clock to the SSL is halted. 0: SSL operates 1: Clock supply to SSL halted Note: On the models not having the SSL, this bit is reserved. The write value should always be 1.
1	MSTP51	0	R/W	Module Stop Bit 51 When the MSTP51 bit is set to 1, the supply of the clock to the AFEIF is halted. 0: AFEIF operates 1: Clock supply to AFEIF halted
0	MSTP50	0	R/W	Module Stop Bit 50 When the MSTP50 bit is set to 1, the supply of the clock to the LCDC is halted. 0: LCDC operates 1: Clock supply to LCDC halted

13.4 Sleep Mode

13.4.1 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. 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 CKIO pin. In sleep mode the output of the STATUS0 pin and STATUS1 pin go high and low, respectively.

13.4.2 Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, IRQ, IRL, PINT, and on-chip peripheral module) or reset. Interrupts are accepted in sleep mode even when the BL bit in SR is 1. If necessary, save SPC and SSR to the stack before executing the SLEEP instruction.

(1) Canceling with Interrupt

When an NMI, IRQ, IRL, PINT, or on-chip peripheral module interrupt occurs, sleep mode is canceled and interrupt exception handling is executed. A code indicating the interrupt source is set in INTEVT and INTEVT2.

(2) Canceling with Reset

Sleep mode is canceled by a power-on reset or a manual reset.

13.5 Software Standby Mode

13.5.1 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 CKIO pin also halts.

The contents of the CPU and cache registers remain unchanged. Some registers of the on-chip peripheral modules are, however, initialized. Refer to section 37, List of Registers, for the register states of the on-chip peripheral modules in software standby mode. The procedure for a transition to software standby mode is as follows.

1. Clear the TME bit in the WDT's timer control register (WTCSR) to 0 to stop the WDT.
2. Clear the WDT's timer counter (WTCNT) to 0 and set the CKS2 to CKS0 bits in WTCSR to appropriate values to secure the specified oscillation settling time.
3. After the STBY bit in STBCR is set to 1, the SLEEP instruction is executed.
4. Software standby mode is entered and the clocks within the chip are halted. The output of the STATUS0 pin and STATUS1 pin go high and low, respectively.

13.5.2 Canceling Software Standby Mode

Software standby mode is canceled by interrupts (NMI, IRQ (edge detection), RTC, TMU, and PINT) or a reset.

(1) Canceling with Interrupt

The on-chip WDT can be used for hot starts. When the chip detects an NMI, IRQ (edge detection)*[1], RTC*[1], TMU*[1], or PINT*[1] interrupt, the clock will be supplied to the entire chip and software standby mode will be canceled after the time set in the WDT's timer control/status register has elapsed. Both STATUS1 and STATUS0 pins go low. Interrupt exception handling then begins and a code indicating the interrupt source is set in INTEVT and INTEVT2. After the branch to the interrupt handling routine, clear the STBY bit in STBCR. WDT stops automatically. If the STBY bit is not cleared, WDT continues operation and a transition is made to software standby mode*[2] when WTCNT reaches H'80. Note that a manual reset is not accepted until the STBY bit is cleared. Interrupts are accepted in software standby mode even when the BL bit in SR is 1. If necessary, save SPC and SSR to the stack before executing the SLEEP instruction.

Immediately after an interrupt is detected, the phase of the clock output of the CKIO pin may be unstable, until software standby mode is canceled.

Notes: 1. Only when the RTC is used, software standby mode can be canceled by IRQ (edge detection), RTC, TMU, or PINT interrupt.
2. Cancel this software standby mode by a power-on reset.

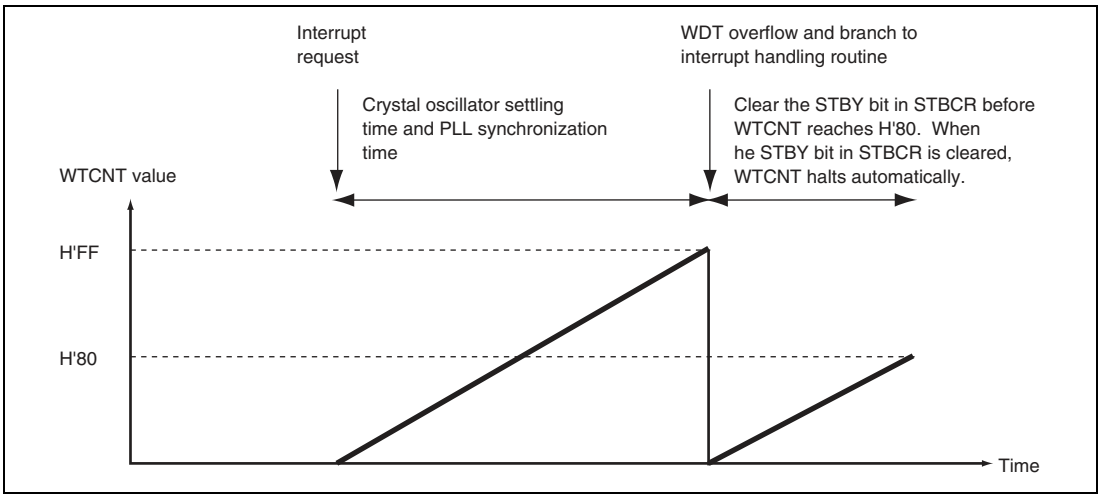


Figure 13.1 Canceling Standby Mode with STBY Bit in STBCR

(2) Canceling with Reset

Software standby mode is canceled by a reset with the $\overline{\text{RESETP}}$ pin and $\overline{\text{RESETM}}$ pin. Keep the $\overline{\text{RESETP}}$ pin and $\overline{\text{RESETM}}$ pin low until the clock oscillation settles in clock operating mode to use PLL. The internal clock will continue to be output to the CKIO pin.

13.6 Module Standby Function

13.6.1 Transition to Module Standby Function

Setting the MSTP bits in the standby control register 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. When changing the setting of an MSTP bit to use the module standby function, be sure to halt operation of the module whose clock supply is to be stopped before setting the corresponding MSTP bit to 1.

In the module standby state, the states of the external pins of the on-chip peripheral modules differ depending on the on-chip peripheral module and I/O port settings. Register state is as same as in standby mode. When changing the setting of an MSTP bit to use the module standby function, be sure to halt operation of the module whose clock supply is to be stopped before setting the corresponding MSTP bit to 1.

13.6.2 Canceling Module Standby Function

The module standby function can be canceled by clearing the MSTP bits to 0, or by a power-on reset.

When canceling the module standby function by clearing the corresponding MSTP bit, be sure to read the relevant MSTP bit to confirm that it has been cleared to 0.

13.7 STATUS Pin Change Timing

The STATUS1 and STATUS0 pin change timings are shown below.

13.7.1 Reset

(1) Power-on reset

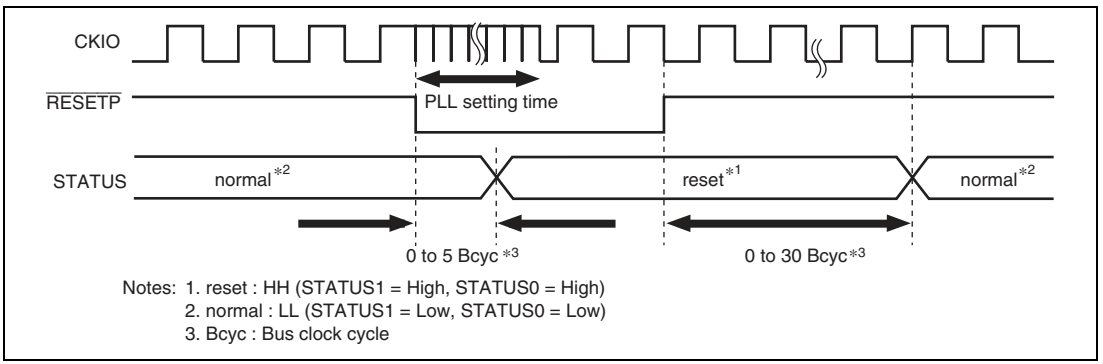


Figure 13.2 STATUS Output at Power-on Reset

(2) Manual reset

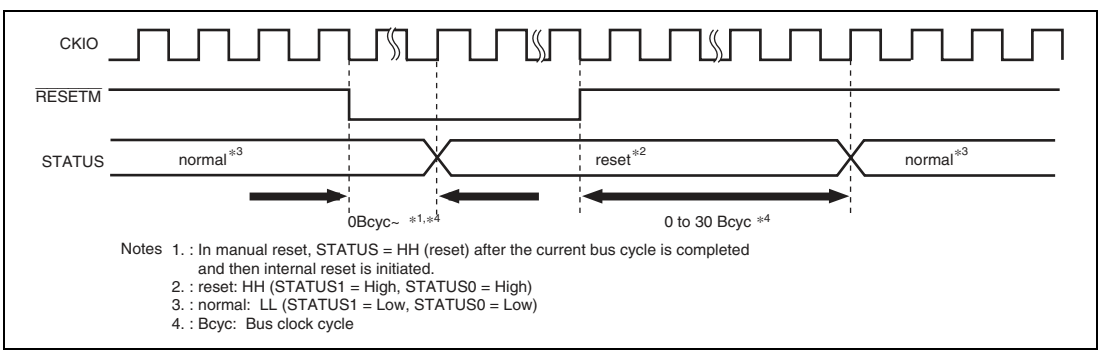
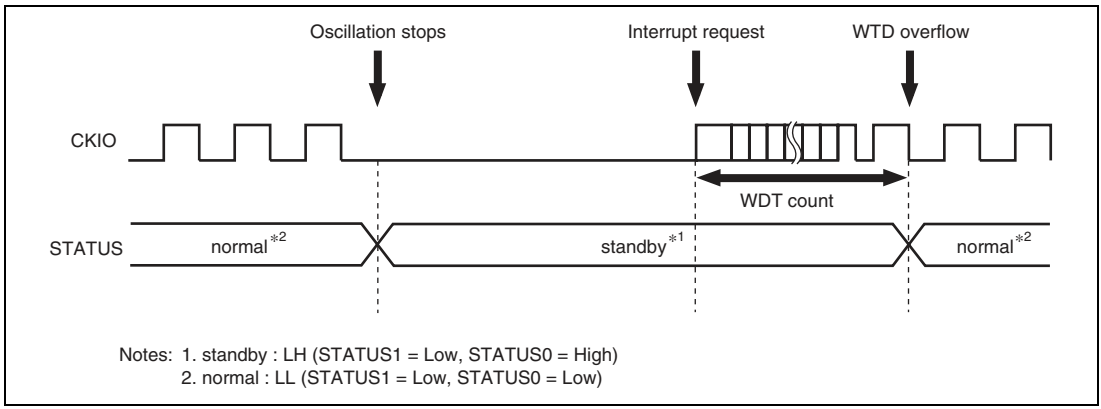


Figure 13.3 STATUS Output at Manual Reset

13.7.2 Software Standby Mode

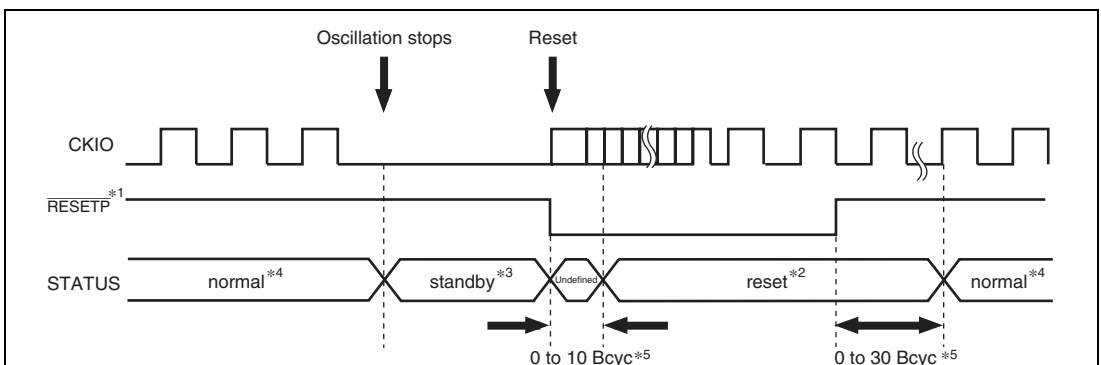
(1) Canceling software standby mode by an interrupt



Notes: 1. standby : LH (STATUS1 = Low, STATUS0 = High)
2. normal : LL (STATUS1 = Low, STATUS0 = Low)

Figure 13.4 STATUS Output when Software Standby Mode is Canceled by an Interrupt

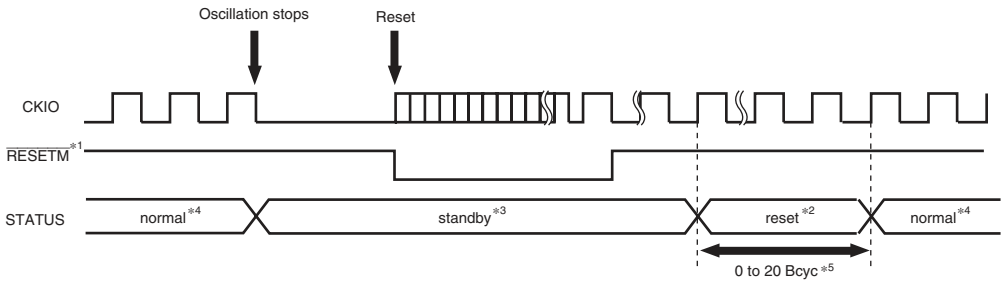
(2) Canceling software standby mode by a power-on reset



Notes: 1. If a standby mode is canceled by a power on reset, the WDT stops counting.
RESETP must be kept low for the PLL oscillation stabilization time.
2. reset : HH (STATUS1 = High, STATUS0 = High)
3. standby : LH (STATUS1 = Low, STATUS0 = High)
4. normal : LL (STATUS1 = Low, STATUS0 = Low)
5. Bcyc : Bus clock cycle

Figure 13.5 STATUS Output When Software Standby Mode is Canceled by a Power-on Reset

(3) Canceling software standby mode by a manual reset

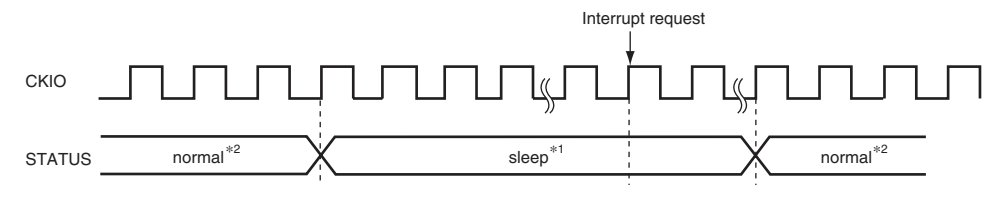


Notes: 1. If a standby mode is canceled by a manual reset, the WDT stops counting. RESETM must be kept low for the PLL oscillation stabilization time.
2. reset : HH (STATUS1 = High, STATUS0 = High)
3. standby : LH (STATUS1 = Low, STATUS0 = High)
4. normal : LL (STATUS1 = Low, STATUS0 = Low)
5. Bcyc : Bus clock cycle

Figure 13.6 STATUS Output When Software Standby Mode is Canceled by a Manual Reset

13.7.3 Sleep Mode

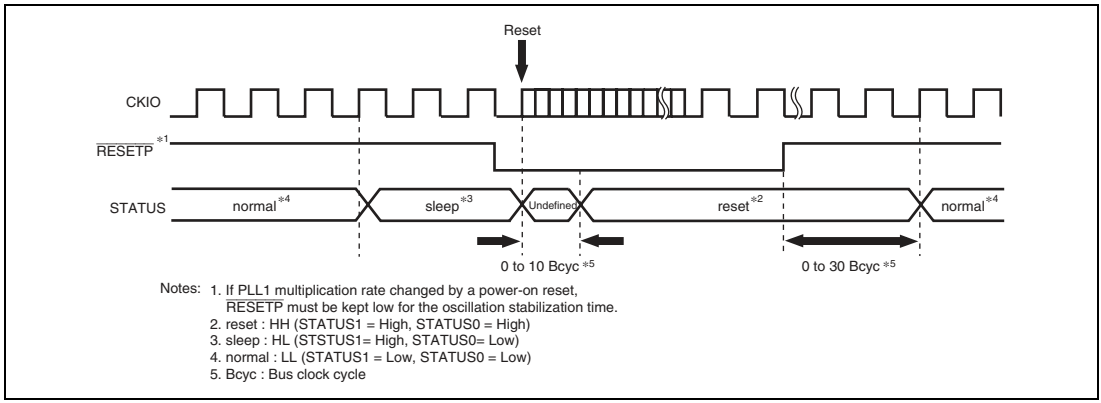
(1) Canceling sleep mode by an interrupt



Notes: 1. sleep : HL (STATUS1 = High, STATUS0 = Low)
2. normal : LL (STATUS1 = Low, STATUS0 = Low)

Figure 13.7 STATUS Output when Sleep Mode is Canceled by an Interrupt

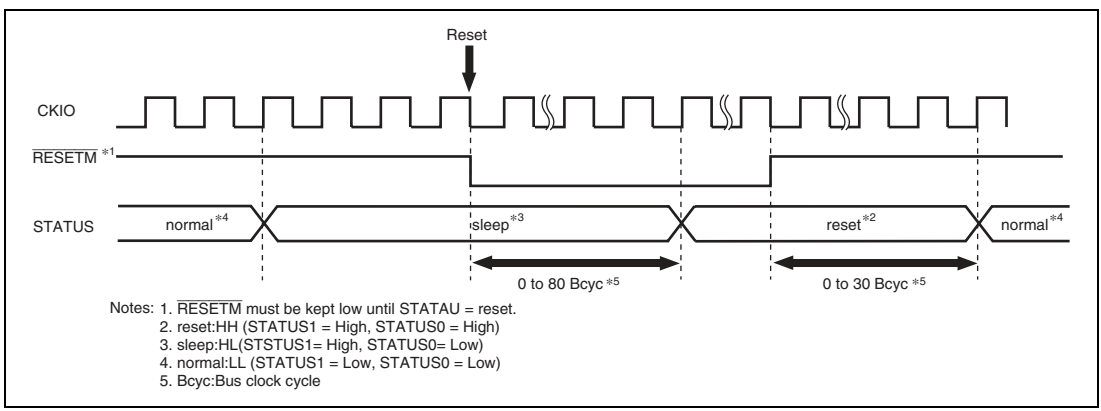
(2) Canceling sleep mode by a power-on reset



Notes: 1. If PLL1 multiplication rate changed by a power-on reset, $\overline{\text{RESETP}}$ must be kept low for the oscillation stabilization time.
2. reset : HH (STATUS1 = High, STATUS0 = High)
3. sleep : HL (STSTUS1= High, STATUS0= Low)
4. normal : LL (STATUS1 = Low, STATUS0 = Low)
5. Bcyc : Bus clock cycle

Figure 13.8 STATUS Output When Sleep Mode is Canceled by a Power-on Reset

(3) Canceling sleep mode by a manual reset



Notes: 1. $\overline{\text{RESETM}}$ must be kept low until STATAU = reset.
2. reset:HH (STATUS1 = High, STATUS0 = High)
3. sleep:HL(STSTUS1= High, STATUS0= Low)
4. normal:LL (STATUS1 = Low, STATUS0 = Low)
5. Bcyc:Bus clock cycle

Figure 13.9 STATUS Output When Sleep Mode is Canceled by a Manual Reset

13.8 Hardware Standby Mode

13.8.1 Transition to Hardware Standby Mode

This LSI enters hardware standby mode by driving the CA pin low. In hardware standby mode as well as a standby mode entered by the SLEEP instruction, all modules stop other than the modules that operate using the RTC clock.

Hardware standby mode differs from standby mode as follows:

1. Interrupts and manual reset cannot be accepted.
2. TMU does not operate.
3. RTC operates without power supply to the power supply pins other than that of RTC.

If the CA pin goes low, the operation differs according to the CPG status.

- During standby mode
  Hardware standby mode is entered with the clock stopped.
  Interrupts and manual reset cannot be accepted and TMU halts.
- During WDT operation while the standby mode is canceled by an interrupt
  After standby mode is canceled and the CPU restarts operating, a transition to hardware standby mode occurs.
- Sleep mode
  After sleep mode is canceled and the CPU restarts operating, a transition to hardware standby mode occurs.

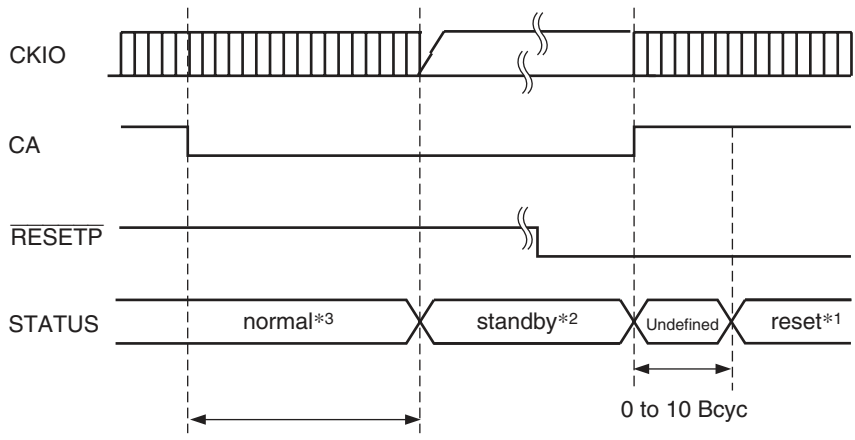
Note that the CA pin must be brought low during hardware standby mode.

13.8.2 Canceling the Hardware Standby Mode

Hardware standby mode is canceled by power-on or reset. If the CA pin is pulled high while the $\overline{\text{RESETP}}$ signal is low, clock oscillation starts. In this case, $\overline{\text{RESETP}}$ must be kept low until clock oscillation has settled. If $\overline{\text{RESETP}}$ is then pulled high, the CPU initiates the power-on reset processing. If an interrupt or manual reset is input, correct operation cannot be guaranteed.

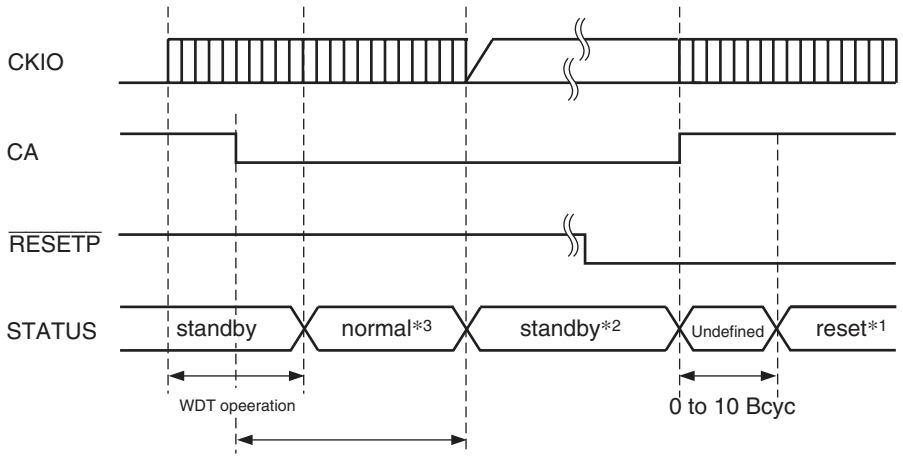
13.8.3 Hardware Standby Mode Timing

Figures 13.10 and 13.11 show signal timings in hardware standby mode. Since the signal on the CA pin is sampled at the timing of EXTAL_RTC, clock should be input to the EXTAL_RTC pin when hardware standby mode is entered. In hardware standby mode, the CA pin must be kept low. The clock oscillation starts if the CA pin is pulled high after the $\overline{\text{RESETP}}$ pin is brought low.



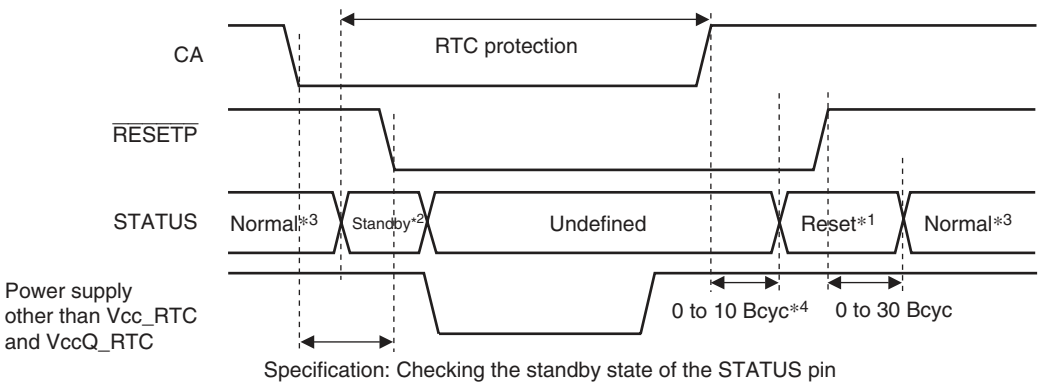
Notes: 1. reset : HH (STATUS1 = High, STATUS0 = High)
2. standby : LHLH (STATUS1 = Low, STATUS0 = High)
3. normal : LL (STATUS1 = Low, STATUS0 = Low)
4. Bcyc : Bus clock cycle

Figure 13.10 Hardware Standby Mode Timing (CA is pulled low in normal operation)



Notes: 1. reset : HH (STATUS1 = High, STATUS0 = High)
2. standby : LH(STATUS1 = Low, STATUS0 = High)
3. normal : LL (STATUS1 = Low, STATUS0 = Low)
4. Bcyc : Bus clock cycle

Figure 13.11 Hardware Standby Mode Timing
(CA is pulled low while WDT operates after the standby mode is canceled)



Notes: *1 Reset: HH (STATUS1 = High, STATUS0 = High)
*2 Standby: LH (STATUS1 = Low, STATUS0 = High)
*3 Normal operation: LL (STATUS1 = Low, STATUS0 = Low)
*4 Bcyc: Bus clock cycle

Figure 13.12 Timing When Power of Pins other than V_{cc}_RTC and V_{cc}Q_RTC is Off

Section 14 Timer Unit (TMU)

This LSI includes a three-channel 32-bit timer unit (TMU).

14.1 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 RTC output clock (16 kHz)
- Allows channels operate when this LSI is in standby mode

  Even when this LSI is in standby mode, channels can operate when the RTC output clock is used as a counter input clock.

Figure 14.1 shows a block diagram of the TMU.

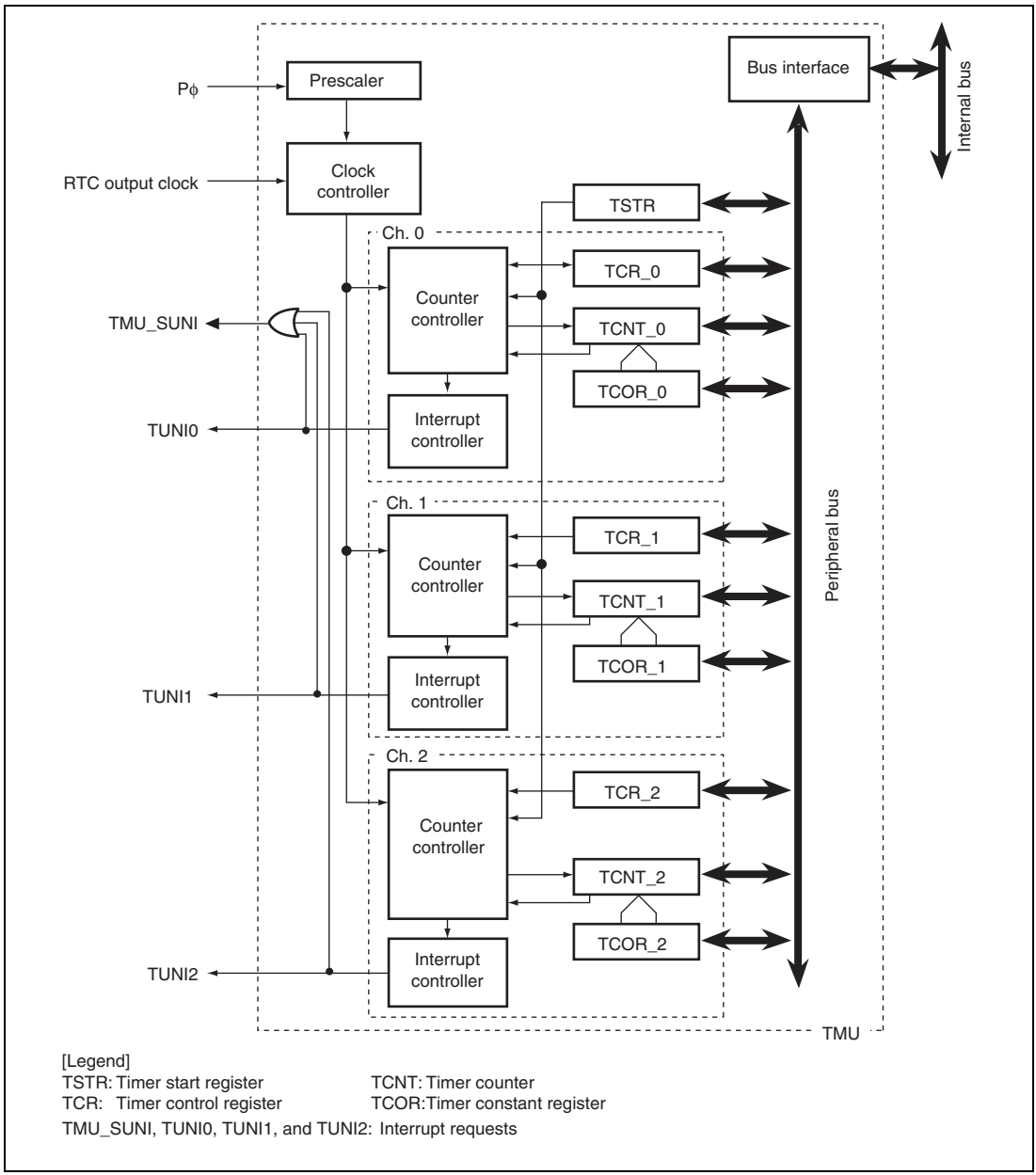


Figure 14.1 Block Diagram of TMU

14.2 Register Descriptions

The TMU has the following registers. Refer to section 37, List of Registers, for more details on the addresses and states of these registers in each operating mode. Notation for the CMT registers takes the form XXX_N, where XXX including the register name and N indicating the channel number. For example, TCOR_0 denotes the TCOR for channel 0.

(1) Common

- Timer start register (TSTR)

(2) Channel 0

- Timer constant register_0 (TCOR_0)
- Timer counter_0 (TCNT_0)
- Timer control register_0 (TCR_0)

(3) Channel 1

- Timer constant register_1 (TCOR_1)
- Timer counter_1 (TCNT_1)
- Timer control register_1 (TCR_1)

(4) Channel 2

- Timer constant register_2 (TCOR_2)
- Timer counter_2 (TCNT_2)
- Timer control register_2 (TCR_2)

14.2.1 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects whether to operate or halt the timer counters (TCNT).

TSTR is initialized to H'00 at a power-on reset, manual reset, or in module stop mode. It is retained in sleep mode and standby mode.

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

14.2.2 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 indicating timer counter (TCNT) underflow has been set to 1, and also carry out counter clock selection.

TCR are initialized to H'0000 at a power-on reset or manual reset. They are retained in standby or sleep mode.

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

Bit	Bit Name	Initial Value	R/W	Description
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	TPSC2	0	R/W	Timer Prescaler 2 to 0
1	TPSC1	0	R/W	Select the TCNT count clock.
0	TPSC0	0	R/W	000: Count on Pϕ/4 001: Count on Pϕ/16 010: Count on Pϕ/64 011: Count on Pϕ/256 101: Count on RTC output clock (16 kHz) Others are setting prohibited.

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

14.2.3 Timer Constant Registers (TCOR)

TCOR set the value to be set in TCNT when TCNT underflows.

TCOR are 32-bit readable/writable registers.

TCOR are initialized to H'FFFFFFFF at a power-on reset or manual reset. They are retained in standby or sleep mode.

14.2.4 Timer Counters (TCNT)

TCNT count down upon input of a clock. The clock input is selected using the TPSC2 to TPSC0 bits in the timer control register (TCR).

When a TCNT count-down results in an underflow (H'00000000 → H'FFFFFFFF), the underflow flag (UNF) in the timer control register (TCR) of the relevant channel is set. The TCOR value is simultaneously set in TCNT itself and the count-down continues from that value.

TCNT are initialized to H'FFFFFFFF at a power-on reset or manual reset. They are retained in standby or sleep mode.

14.3 Operation

Each of the three channels has a 32-bit timer counter (TCNT) and a 32-bit timer constant register (TCOR). TCNT counts down. The auto-reload function enables synchronized counting.

14.3.1 Counter Operation

When the STR0 to STR2 bits in the timer start register (TSTR) are set to 1, the corresponding timer counter (TCNT) starts counting. When TCNT underflows, the UNF flag in the corresponding timer control register (TCR) is set. At this time, if the UNIE bit in TCR is 1, an interrupt request is sent to the CPU. Also at this time, the value is copied from TCOR to TCNT and the down-count operation is continued.

(1) Count Operation Setting Procedure

An example of the procedure for setting the count operation is shown in figure 14.2.

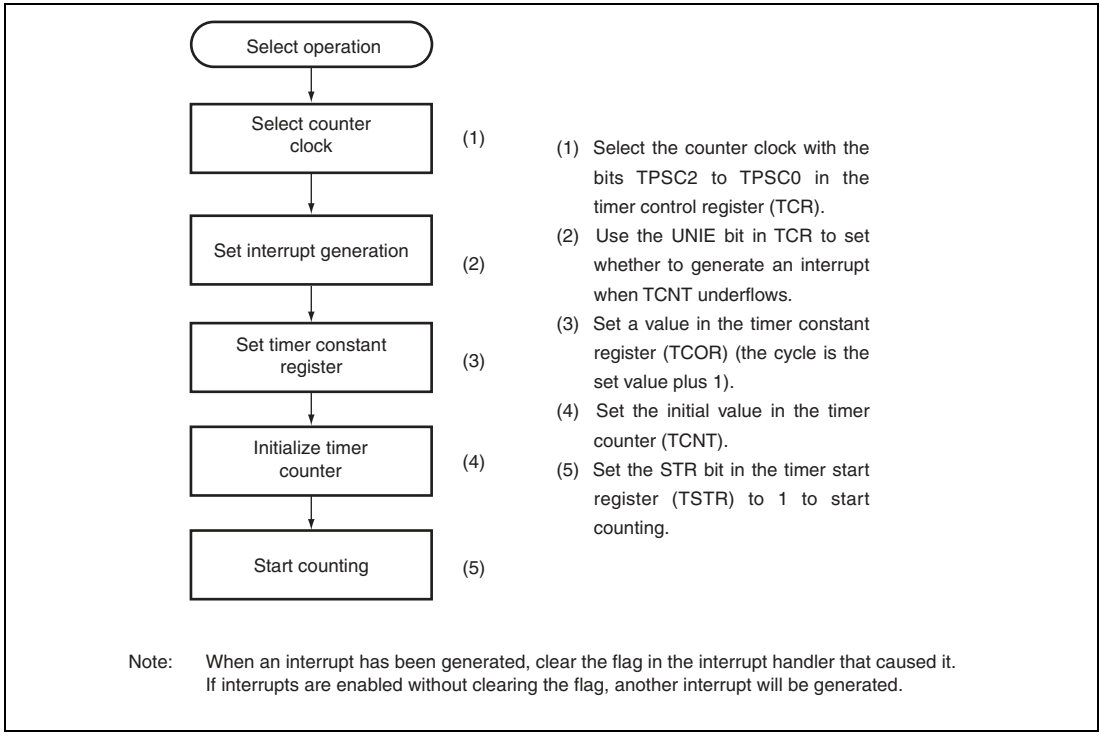


Figure 14.2 Setting Count Operation

(2) Auto-Reload Count Operation

Figure 14.3 shows the TCNT auto-reload operation.

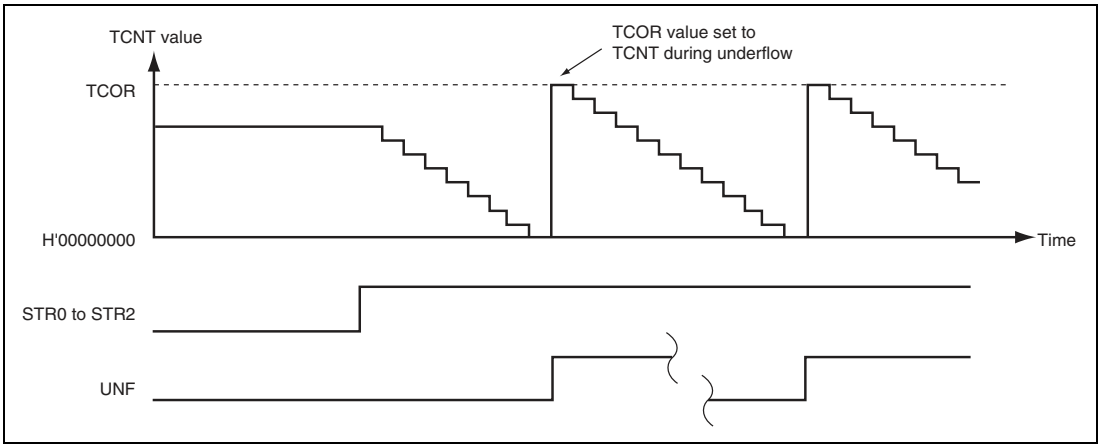


Figure 14.3 Auto-Reload Count Operation

(3) TCNT Count Timing

1. Internal clock

   Set the bits TPSC2 to TPSC0 in TCR to select one of the four internal clocks created by dividing a peripheral module clock (Pϕ/4, Pϕ/16, Pϕ/64, Pϕ/256). Figure 14.4 shows the timing.

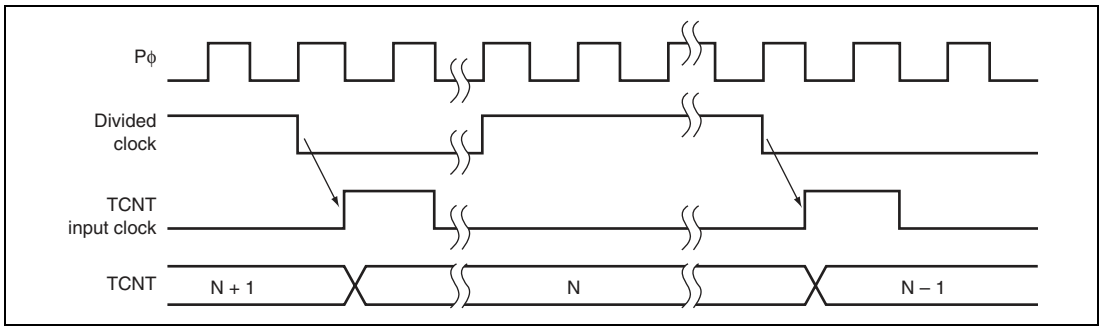


Figure 14.4 Count Timing when Internal Clock is Operating

2. Internal RTC clock

   Set the bits TPSC2 to TPSC0 in TCR to select the RTC output clock as a clock for timer. Figure 14.5 shows the timing.

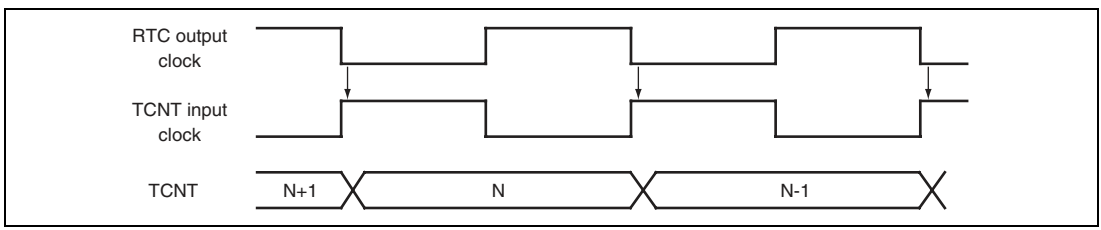


Figure 14.5 Count Timing when RTC Clock is Operating

14.4 Interrupts

There is one source of TMU interrupts: underflow interrupts (TUNI).

14.4.1 Status Flag Set Timing

The UNF bit is set to 1 when TCNT underflows. Figure 14.6 shows the timing.

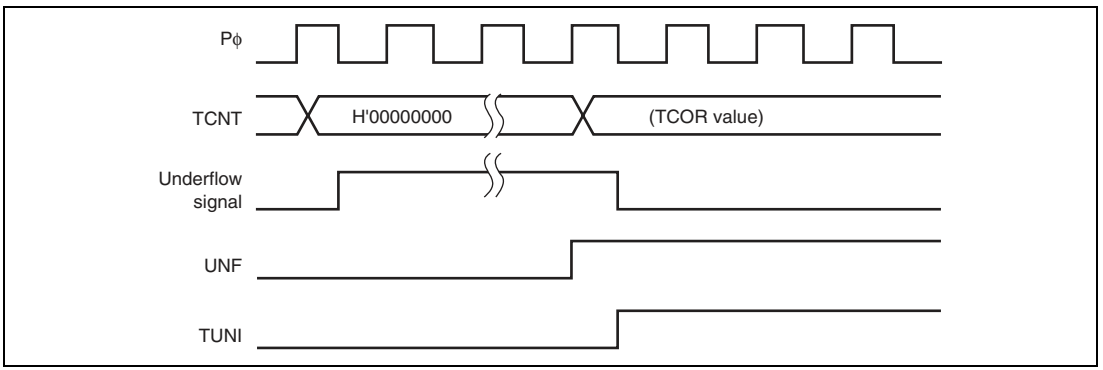


Figure 14.6 UNF Set Timing

14.4.2 Status Flag Clear Timing

The status flag can be cleared by writing 0 from the CPU. Figure 14.7 shows the timing.

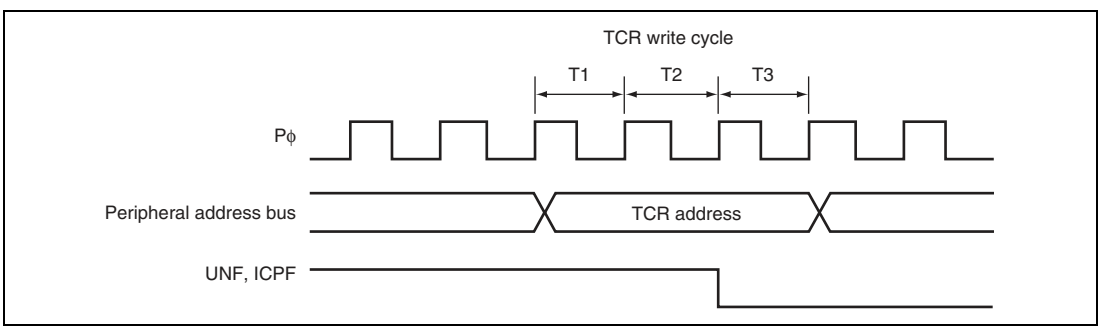


Figure 14.7 Status Flag Clear Timing

14.4.3 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. Codes are set in the interrupt event register (INTEVT, INTEVT2) for these interrupts and interrupt processing must be executed according to the codes.

The relative priorities between channels can be changed using the interrupt controller. For details, refer to section 7, Exception Handling, and section 8, Interrupt Controller (INTC). Table 14.1 lists TMU interrupt sources.

Table 14.1 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

Software standby mode can be cancelled by TSU_SUNI which is OR of an underflow interrupt for each TMU channel. (It is available when the RTC output clock is selected as a counter input clock.)

TMU_SUNI is processed as an interrupt which differs from an underflow interrupt for each channel by the interrupt controller (INTC). Therefore, an underflow interrupt for each channel and TMU_SUNI should be used differently.

When canceling software standby mode, set the bits 11 to 8 in interrupt priority register D (IPRD) of INTC to any value and bits 15 to 4 in interrupt priority register A (IPRA) of INTC to H'000 so that only the TMU_SUNI is accepted. In the TMU_SUNI interrupt routine, clear both the under flow flag (UNF) in the timer control register (TCR) and the TMU_SUNI interrupt request bit (TMU_SUNIR) in the interrupt request register 0 of INTC.

In the normal operating state, set the bits 11 to 8 in IPRD to H'0 and bits 15 to 4 in IPRA to any value so that an underflow interrupt can be accepted for each channel. For details, see section 8, Interrupt Controller (INTC).

14.5 Usage Notes

14.5.1 Writing to Registers

Synchronization processing is not performed for timer counting during register writes. When writing to registers, always clear the appropriate start bits for the channel (STR2 to STR0) in the timer start register (TSTR) to halt timer counting.

14.5.2 Reading Registers

Synchronization processing is performed for timer counting during register reads. When timer counting and register read processing are performed simultaneously, the register value before TCNT counting down with synchronization processing is read.

Section 15 16-Bit Timer Pulse Unit (TPU)

This LSI has an on-chip 16-bit timer pulse unit (TPU) that comprises four 16-bit timer channels.

15.1 Features

- Maximum 4-pulse output
  - A total of 16 timer general registers (TGRA to TGRD × 4 ch.) are provided (four each for channels). TGRA can be set as an output compare register.
  - TGRB, TGRC, and TGRD for each channel can also be used as timer counter clearing registers. TGRC and TGRD can also be used as buffer registers.
- Selection of four counter input clocks for channels 0 and 1, and of six counter input clocks for channels 2 and 3.
- The following operations can be set for each channel:
  - Waveform output at compare match: Selection of 0, 1, or toggle output
  - Counter clear operation: Counter clearing possible by compare match
  - PWM mode: Any PWM output duty can be set
    Maximum of 4-phase PWM output possible
- Buffer operation settable for each channel
  - Automatic rewriting of output compare register possible
- Phase counting mode settable independently for each of channels 2, and 3
  - Two-phase encoder pulse up/down-count possible
- An interrupt request for each channel (TPIn (n = 0, 1, 2, 3))
  - For channels 0 and 1, compare match interrupts and overflow interrupts can be requested independently
  - For channels 2, and 3, compare match interrupts, overflow interrupts, and underflow interrupts can be requested independently

Table 15.1 lists the functions of the TPU.

Table 15.1 TPU Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3
Count 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 TPU_TI2A TPU_TI2B	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 TPU_TI3A TPU_TI3B
General registers		TGR0A TGR0B	TGR1A TGR1B	TGR2A TGR2B	TGR3A TGR3B
General registers/ buffer registers		TGR0C TGR0D	TGR1C TGR1D	TGR2C TGR2D	TGR3C TGR3D
Output pins		TPU_TO0	TPU_TO1	TPU_TO2	TPU_TO3
Counter clear function		TGR compare match	TGR compare match	TGR compare match	TGR compare match
Compare match output	0 output		◯	◯	◯
	1 output	◯	◯	◯	◯
	Toggle output	◯	◯	◯	◯
PWM mode		◯	◯	◯	◯
Phase counting mode		—	—	◯	◯
Buffer operation		◯	◯	◯	◯
Interrupt sources		5 sources • Compare match • Overflow	5 sources • Compare match • Overflow	6 sources • Compare match • Overflow • Underflow	6 sources • Compare match • Overflow • Underflow

[Legend]

◯: Possible

—: Not possible

Note: TPU_TI2B and TPU_TI3B are used as count clocks only in phase counting mode.

Figure 15.1 shows a block diagram of the TPU.

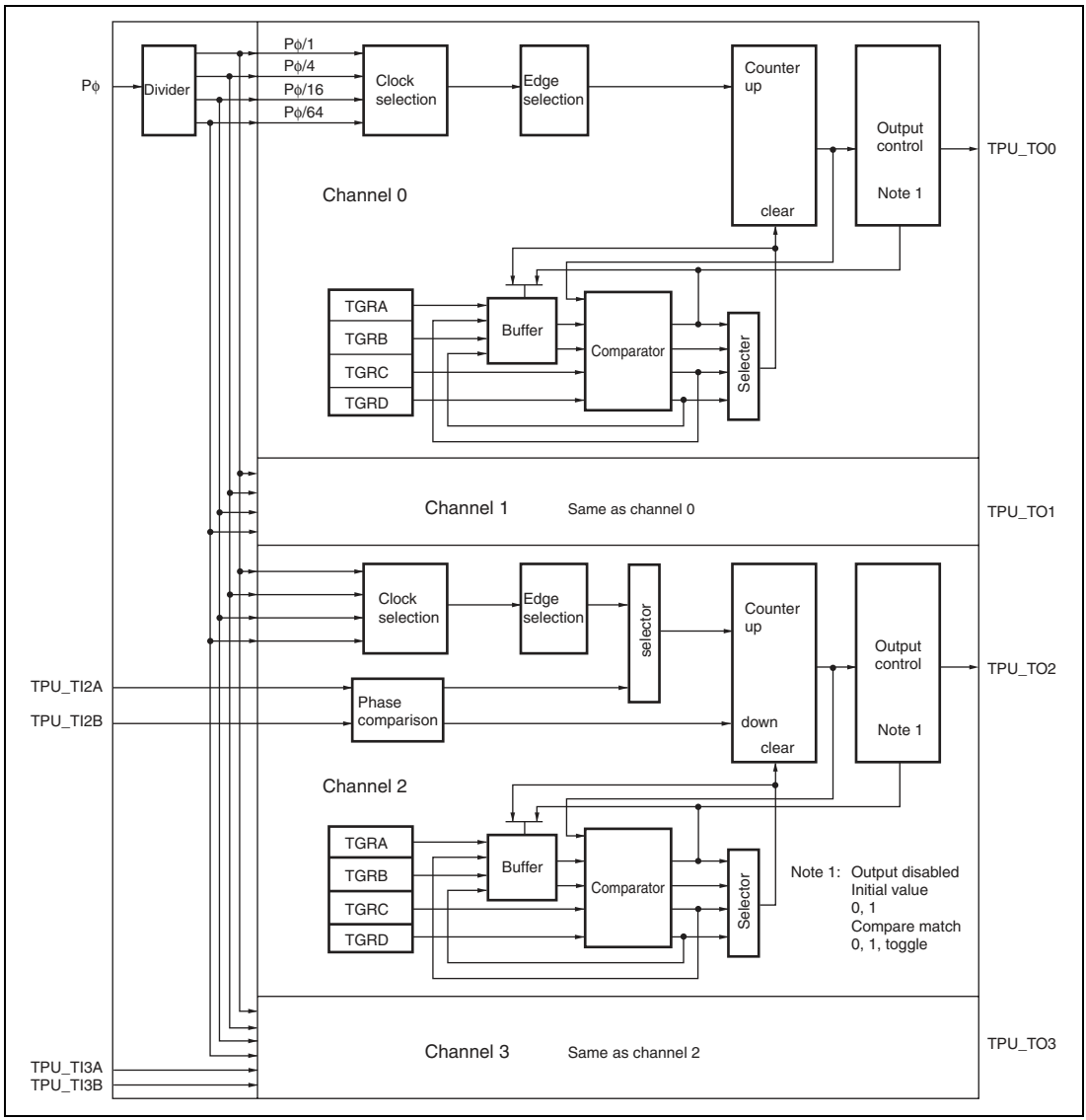


Figure 15.1 Block Diagram of TPU

15.2 Input/Output Pins

Table 15.2 summarizes the TPU related external pins.

Table 15.2 TPU Pin Configurations

Channel	Name	Pin Name	I/O	Function
0	TPU compare match output 0	TPU_TO0	Output	TGR0A output compare output/PWM output pin
1	TPU compare match output 1	TPU_TO1	Output	TGR1A output compare output/PWM output pin
2	TPU compare match output 2A	TPU_TO2	Output	TGR2A output compare output/PWM output pin
	TPU clock input 2A	TPU_TI2A	Input	External clock channel 2A input pin /channel 2 counting mode A phase input
	TPU clock input 2B	TPU_TI2B	Input	Channel 2 counting mode B phase input
3	TPU compare match output 3A	TPU_TO3	Output	TGR3A output compare output/PWM output pin
	TPU clock input 3A	TPU_TI3A	Input	External clock channel 3A input pin /channel 3 counting mode A phase input
	TPU clock input 3B	TPU_TI3B	Input	Channel 3 counting mode B phase input

15.3 Register Descriptions

The TPU has the following registers. Refer to section 37, List of Registers, for more details on the addresses and states of these registers in each operating mode.

Channel 0:

- Timer control register_0 (TCR_0)
- Timer mode register_0 (TMDR_0)
- Timer I/O control register_0 (TIOR_0)
- Timer interrupt enable register_0 (TIER_0)
- Timer status register_0 (TSR_0)
- Timer counter_0 (TCNT_0)
- Timer general register A_0 (TGRA_0)
- Timer general register B_0 (TGRB_0)
- Timer general register C_0 (TGRC_0)
- Timer general register D_0 (TGRD_0)

Channel 1:

- Timer control register_1 (TCR_1)
- Timer mode register_1 (TMDR_1)
- Timer I/O control register_1 (TIOR_1)
- Timer interrupt enable register_1 (TIER_1)
- Timer status register_1 (TSR_1)
- Timer counter_1 (TCNT_1)
- Timer general register A_1 (TGRA_1)
- Timer general register B_1 (TGRB_1)
- Timer general register C_1 (TGRC_1)
- Timer general register D_1 (TGRD_1)

Channel 2:

- Timer control register_2 (TCR_2)
- Timer mode register_2 (TMDR_2)
- Timer I/O control register_2 (TIOR_2)
- Timer interrupt enable register_2 (TIER_2)
- Timer status register_2 (TSR_2)

- Timer counter_2 (TCNT_2)
- Timer general register A_2 (TGRA_2)
- Timer general register B_2 (TGRB_2)
- Timer general register C_2 (TGRC_2)
- Timer general register D_2 (TGRD_2)

Channel 3:

- Timer control register_3 (TCR_3)
- Timer mode register_3 (TMDR_3)
- Timer I/O control register_3 (TIOR_3)
- Timer interrupt enable register_3 (TIER_3)
- Timer status register_3 (TSR_3)
- Timer counter_3 (TCNT_3)
- Timer general register A_3 (TGRA_3)
- Timer general register B_3 (TGRB_3)
- Timer general register C_3 (TGRC_3)
- Timer general register D_3 (TGRD_3)
- Timer start register (TSTR)

15.3.1 Timer Control Registers (TCR)

The TCR registers are 16-bit registers that control the TCNT channels. The TPU has four TCR registers, one for each of channels 0 to 3. The TCR registers are initialized to H'0000 by a reset, but not initialized in standby mode, sleep mode, or module standby.

TCR register settings should be made only when TCNT operation is stopped.

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	CCLR2	0	R/W	Counter Clear 2, 1, and 0
6	CCLR1	0	R/W	These bits select the TCNT counter clearing source.
5	CCLR0	0	R/W	000: TCNT clearing disabled 001: TCNT cleared by TGRA compare match 010: TCNT cleared by TGRB compare match 011: Reserved (setting prohibited) 100: TCNT clearing disabled 101: TCNT cleared by TGRC compare match 110: TCNT cleared by TGRD compare match 111: Reserved (setting prohibited)
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. ϕ/4 both edges = ϕ/2 rising edge). If phase counting mode is used this setting is ignored. 00: Count at rising edge 01: Count at falling edge 1X: Count at both edges* [Legend] X: Don't care Note: * If Pϕ/1 is selected for the input clock, operation is disabled.
2	TPSC2	0	R/W	Time Prescaler 2, 1, and 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The clock source can be selected independently for each channel. Table 15.3 shows the clock sources that can be set for each channel. For more in formation on count clock selection, see table 15.4.
0	TPSC0	0	R/W	

Table 15.3 TPU Clock Sources

	Internal Clock				External Clock	
Channel	Pϕ/1	Pϕ/4	Pϕ/16	Pϕ/64	TPU_TI2A	TPU_TI3A
0	○	○	○	○		
1	○	○	○	○		
2	○	○	○	○	○	
3	○	○	○	○		○

[Legend]
○ : Setting
Blank : No setting

Table 15.4 TPSC2 to TPSC0 (1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
0	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	*	*	Reserved (setting prohibited)	

Table 15.4 TPSC2 to TPSC0 (2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
1	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	*	*	Reserved (setting prohibited)	

Table 15.4 TPSC2 to TPSC0 (3)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	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	0	0	External clock: counts on TPU_TI2A pin input
			1	Reserved (setting prohibited)
		1	*	

Table 15.4 TPSC2 to TPSC0 (4)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
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	0	0	External clock: counts on TPU_TI3A pin input
			1	Reserved (setting prohibited)
		1	*	

Note: * Don't care

15.3.2 Timer Mode Registers (TMDR)

The TMDR registers are 16-bit readable/writable registers that are used to set the operating mode for each channel. The TPU has four TMDR registers, one for each channel. The TMDR registers are initialized to H'0000 by a reset, but not initialized in standby mode, sleep mode, or module standby.

TMDR register settings should be made only when TCNT operation is stopped.

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 TGRA and TGRB update timing when TGRC and TGRD are used as a compare match buffer. When TGRC and TGRD are not used as a compare match buffer register, this bit does not function. 0: TGRA and TGRB are rewritten at compare match of each register. 1: TGRA and TGRB are rewritten in counter clearing.
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. 0: TGRB operates normally 1: TGRB and TGRD used together for buffer operation*
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. 0: TGRA operates normally 1: TGRA and TGRC used together for buffer operation
3	—	0	R	Reserved This bit is always read as 0 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
2	MD2	0	R/W	Modes 2 to 0
1	MD1	0	R/W	These bits are used to set the timer operating mode.
0	MD0	0	R/W	000: Normal operation
				001: Reserved (setting prohibited)
				010: PWM mode
				011: Reserved (setting prohibited)
				100: Phase counting mode 1
				101: Phase counting mode 2
				110: Phase counting mode 3
				111: Phase counting mode 4

Note: * Operation when setting (BFWT, BFB, BFA) = (1, 1, 0) is the same as when setting (BFWT, BFB, BFA) = (1, 0, 1). However, when the BFB bit is set to 1 (TGRB and TGRD used together for buffer operation), the setting of (BFWT, BFB, BFA) = (1, 1, 1) should be made. In this case, the value set in TGRA should also be set in TGRC because TGRA and TGRC are also used together for buffer operation.

15.3.3 Timer I/O Control Registers (TIOR)

The TIOR registers are 16-bit registers that control the TPU_TO pin. The TPU has four TIOR registers, one for each channel. The TIOR registers are initialized to H'0000 by a reset, but not initialized in standby mode, sleep mode, or module standby.

TIOR register settings should be made only when TCNT operation is halted.

Care is required since TIOR is affected by the TMDR setting.

If the counting operation is halted, the initial value set by this register is output from the TPU_TO pin.

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	IOA2	0	R/W	I/O Control A2 to A0
1	IOA1	0	R/W	Bits IOA3 to IOA0 specify the functions of TGRA and the TPU_TO pin. For details, see table 15.5.
0	IOA0	0	R/W	

Table 15.5 IOA2 to IOA0

Channels	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description	
0 to 3	0	0	0	Always 0 output (Initial value)	
			1	Initial output is 0 output for TPU_TO pin	0 output at TGRA compare match*
		1	0		1 output at TGRA compare match
			1		Toggle output TGRA at compare match*
	1	0	0	Always 1 output	
			1	Initial output is 1 output for TPU_TO pin	0 output at TGRA compare match
		1	0		1 output at TGRA compare match*
			1		Toggle output at TGRA compare match*

Note: * This setting is invalid in PWM mode.

15.3.4 Timer Interrupt Enable Registers (TIER)

The TIER registers are 16-bit registers that control enabling or disabling of interrupt requests for each channel. The TPU has four TIER registers, one for each channel. The TIER registers are initialized to H'0000 by a reset, but not initialized in standby mode, sleep mode or module standby.

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	0	R	Reserved These bits are always read as 0 and cannot be modified.
5	TC1EU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests by the TCFU bit when the TCFU bit in TSR is set to 1 in phase counting mode of channels 2, and 3 (TCNT underflow). In channels 0 and 1, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests by TCFU disabled 1: Interrupt requests by TCFU enabled
4	TC1EV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests by the TCFV bit when the TCFV bit in TSR is set to 1 (TCNT overflow). 0: Interrupt requests by TCFV disabled 1: Interrupt requests by TCFV enabled
3	TG1ED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests by the TGFD bit when the TGFD bit in TSR is set to (TCNT and TGRD compare match). 0: Interrupt requests by TGFD disabled 1: Interrupt requests by TGFD enabled
2	TG1EC	0	R/W	TGR Interrupt Enable C Enables or disables interrupt requests by the TGFC bit when the TGFC bit in TSR is set to 1 (TCNT and TGRC compare match). 0: Interrupt requests by TGFC disabled 1: Interrupt requests by TGFC enabled

Bit	Bit Name	Initial Value	R/W	Description
1	TG1EB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests by the TGFB bit when the TGFB bit in TSR is set to 1 (TCNT and TGRB compare match). 0: Interrupt requests by TGFB disabled 1: Interrupt requests by TGFB enabled
0	TG1EA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests by the TGFA bit when the TGFA bit in TSR is set to 1 (TCNT and TGRA compare match). 0: Interrupt requests by TGFA disabled 1: Interrupt requests by TGFA enabled

15.3.5 Timer Status Registers (TSR)

The TSR registers are 16-bit registers that indicate the status of each channel. The TPU has four TSR registers, one for each channel. The TSR registers are initialized to H'0000 by a reset, but not initialized in standby mode, sleep mode or module standby mode.

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	TCFD	0	R	Count Direction Flag Status flag that shows the direction in which TCNT counts in phase counting mode of channels 2, and 3. In channels 0 and 1, bit 7 is reserved. It is always read as 0 and cannot be modified. 0: TCNT counts down 1: TCNT counts up
6	—	0	R	Reserved This bit is always read as 0 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
5	TCFU	0	R/(W)*	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 2, and 3 are set to phase counting mode. In channels 0 and 1, bit 5 is reserved. It is always read as 0 and cannot be modified. [Clearing condition] (Initial value) When 0 is written to TCFU after reading TCFU = 1 [Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)
4	TCFV	0	R/(W)*	Overflow Flag Status flag that indicates that TCNT overflow has occurred. [Clearing condition] When 0 is written to TCFV after reading TCFV = 1 [Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)
3	TGFD	0	R/(W)*	Compare Flag D Status flag that indicates the occurrence of TGRD compare match. [Clearing conditions] When 0 is written to TGFD after reading TGFD = 1 [Setting conditions] When TCNT = TGRD
2	TGFC	0	R/(W)*	Compare Flag C Status flag that indicates the occurrence of TGRC compare match. [Clearing conditions] When 0 is written to TGFC after reading TGFC = 1 [Setting conditions] When TCNT = TGRC

Bit	Bit Name	Initial Value	R/W	Description
1	TGFB	0	R/(W)*	Compare Flag B Status flag that indicates the occurrence of TGRB compare match. [Clearing conditions] When 0 is written to TGFB after reading TGFB = 1 [Setting conditions] When TCNT = TGRB
0	TGFA	0	R/(W)*	Output Compare Flag A Status flag that indicates the occurrence of TGRA compare match. [Clearing conditions] When 0 is written to TGFA after reading TGFA = 1 [Setting conditions] When TCNT = TGRA

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

15.3.6 Timer Counters (TCNT)

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

The TCNT counters are initialized to H'0000 by a reset.

The TCNT counters are not initialized in standby mode, sleep mode, or module standby.

15.3.7 Timer General Registers (TGR)

The TGR registers are 16-bit registers. The TPU has 16 TGR registers, four each for channels 0 and 3. TGRC and TGRD can also be designated for operation as buffer registers*. The TGR registers are initialized to H'FFFF by a reset. These registers are not initialized in standby mode, sleep mode, or module standby.

Note: * TGR buffer register combinations are TGRA—TGRC and TGRB—TGRD.

15.3.8 Timer Start Register (TSTR)

TSTR is a 16-bit readable/writable register that selects TCNT operation/stoppage for channels 0 to 3. TSTR is initialized to H'0000 by a reset, but not initialized in standby mode, sleep mode, or module standby.

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 3 to 0
2	CST2	0	R/W	These bits select operation or stoppage for TCNT.
1	CST1	0	R/W	0: TCNTn count operation is stopped)
0	CST0	0	R/W	1: TCNTn performs count operation n=3 to 0

15.4 Operation

15.4.1 Overview

Operation in each mode is outlined below.

(1) Normal Operation

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

(2) Buffer Operation

When a compare match occurs, the value in the buffer register for the relevant channel is transferred to TGR. For update timing from a buffer register, rewriting on compare match occurrence or on counter clearing can be selected.

(3) PWM Mode

In this mode, a PWM waveform is output. The output level can be set by means of TIOR. A PWM waveform with a duty of between 0% and 100% can be output, according to the setting of each TGR register.

(4) Phase Counting Mode

In this mode, TCNT is incremented or decremented by detecting the phases of two clocks input from the external clock input pins (TPU_TI2A and TPU_TI2B, or TPU_TI3A and TPU_TI3B) in channels 2, and 3. When phase counting mode is set, the corresponding TI pin functions as the clock pin, and TCNT performs up/down-counting.

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

15.4.2 Basic Functions

(1) Counter Operation

When one of bits CST0 to CST3 is set to 1 in TSTR, the TCNT counter for the corresponding channel starts counting. TCNT can operate as a free-running counter, periodic counter, and so on.

(a) Example of count operation setting procedure

Figure 15.2 shows an example of the count operation setting procedure.

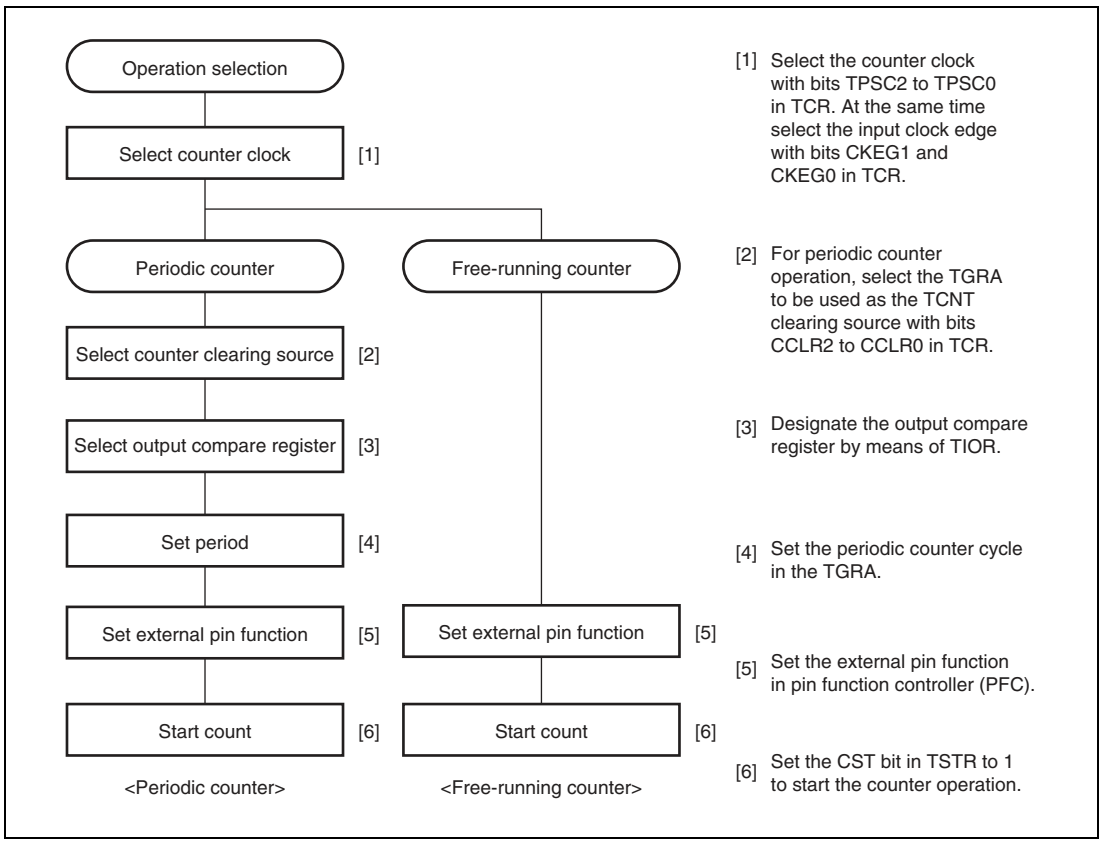


Figure 15.2 Example of Counter Operation Setting Procedure

(b) Free-running count operation and periodic count operation

Immediately after a reset, the TPU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. After overflow, TCNT starts counting up again from H'0000.

Figure 15.3 illustrates free-running counter operation.

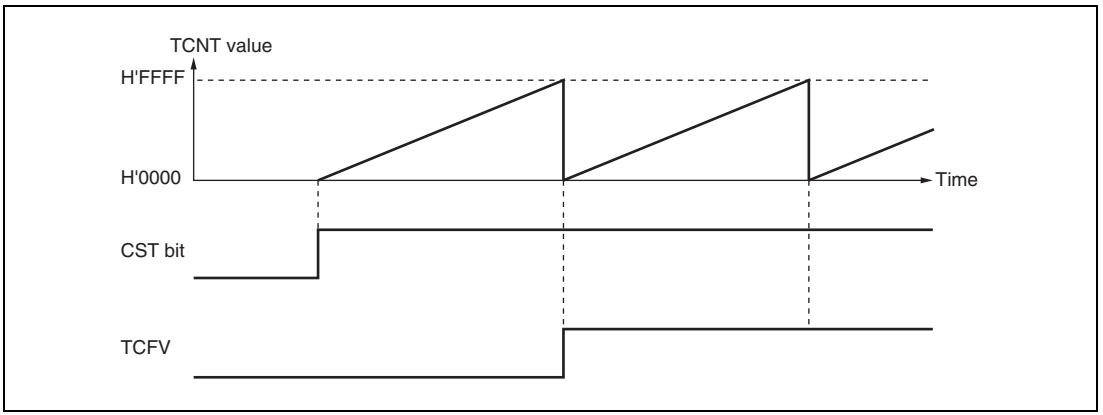


Figure 15.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

After a compare match, TCNT starts counting up again from H'0000.

Figure 15.4 illustrates periodic counter operation.

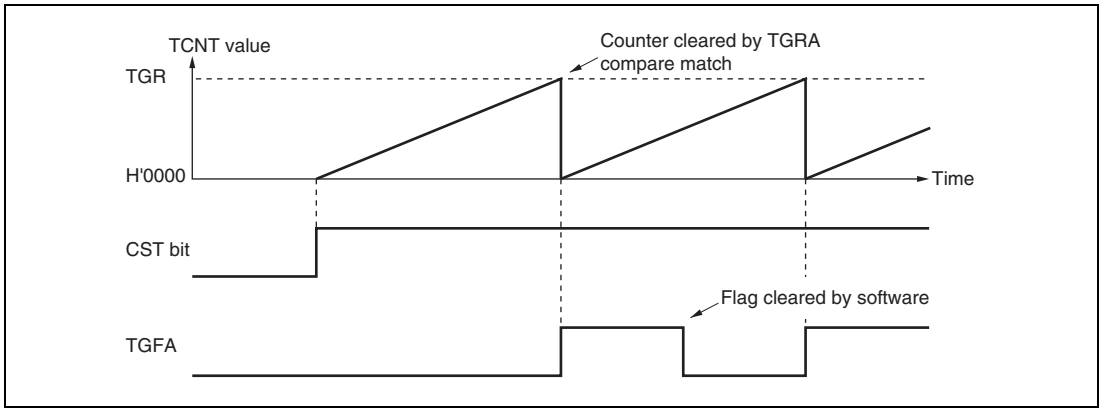


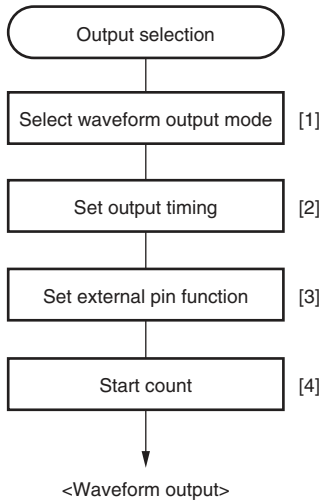
Figure 15.4 Periodic Counter Operation

(2) Waveform Output by Compare Match

The TPU can perform 0, 1, or toggle output from the corresponding output pin (TPU_TO pin) using TGRA compare match.

(a) Example of setting procedure for waveform output by compare match

Figure 15.5 shows an example of the setting procedure for waveform output by compare match.



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

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

[3] Set the external pin function in pin function controller (PFC).

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

Figure 15.5 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of waveform output operation

Figure 15.6 shows an example of 0 output/1 output.

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

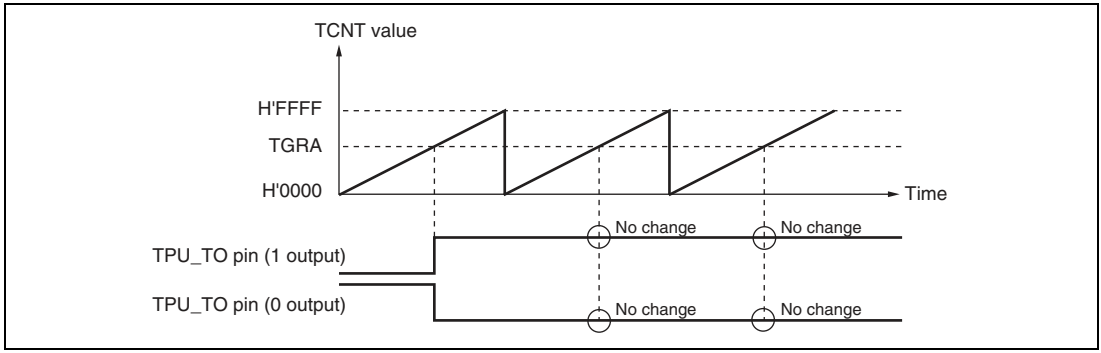


Figure 15.6 Example of 0 Output/1 Output Operation

Figure 15.7 shows an example of toggle output.

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

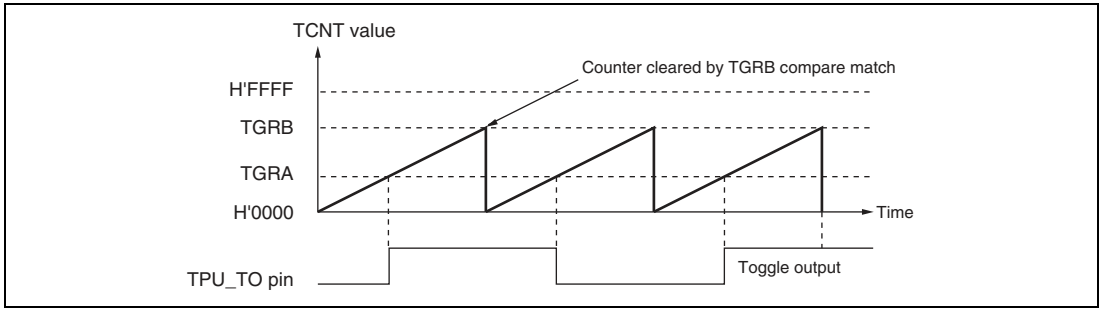


Figure 15.7 Example of Toggle Output Operation

15.4.3 Buffer Operation

Buffer operation, enables TGRC and TGRD to be used as buffer registers.

Table 15.6 shows the register combinations used in buffer operation.

Table 15.6 Register Combinations in Buffer Operation

Timer General Register	Buffer Register
TGRA	TGRC
TGRB	TGRD

When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register. For update timing from a buffer register, rewriting on compare match occurrence or on counter cleaning can be selected.

This operation is illustrated in figure 15.8.

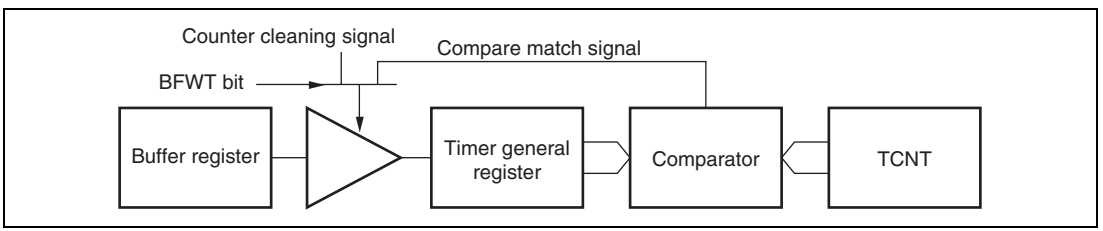


Figure 15.8 Compare Match Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 15.9 shows an example of the buffer operation setting procedure.

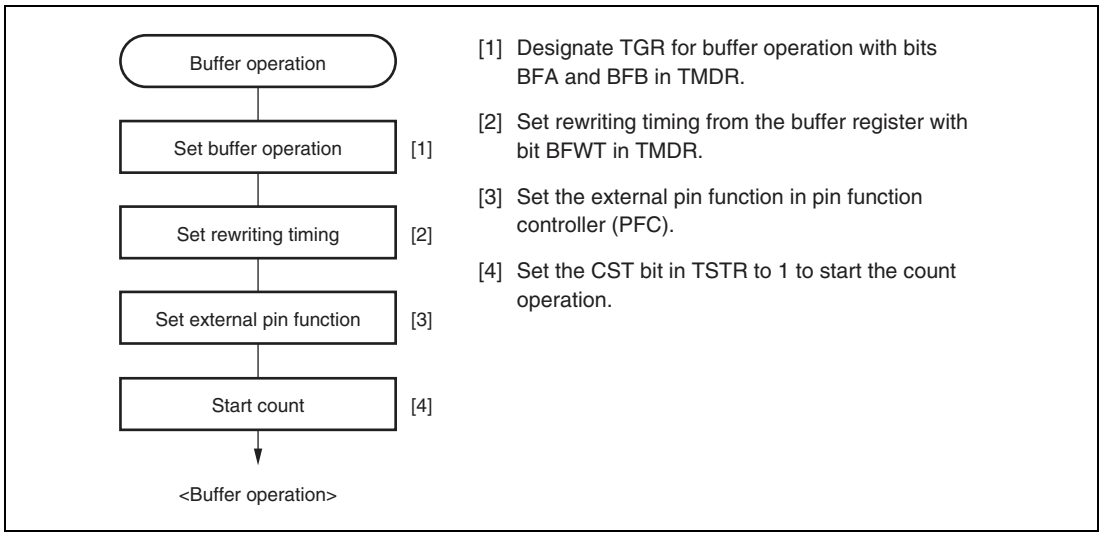


Figure 15.9 Example of Buffer Operation Setting Procedure

(2) Example of Buffer Operation

Figure 15.10 shows an operation example in which PWM mode has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A (TPU_TO pin), and 0 output at counter clearing. Rewriting timing from the buffer register is set at counter clearing.

As buffer operation has been set, when compare match A occurs the output changes. When counter clearing occurs by TGRB, the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

For details of PWM modes, see section 15.4.4, PWM Modes.

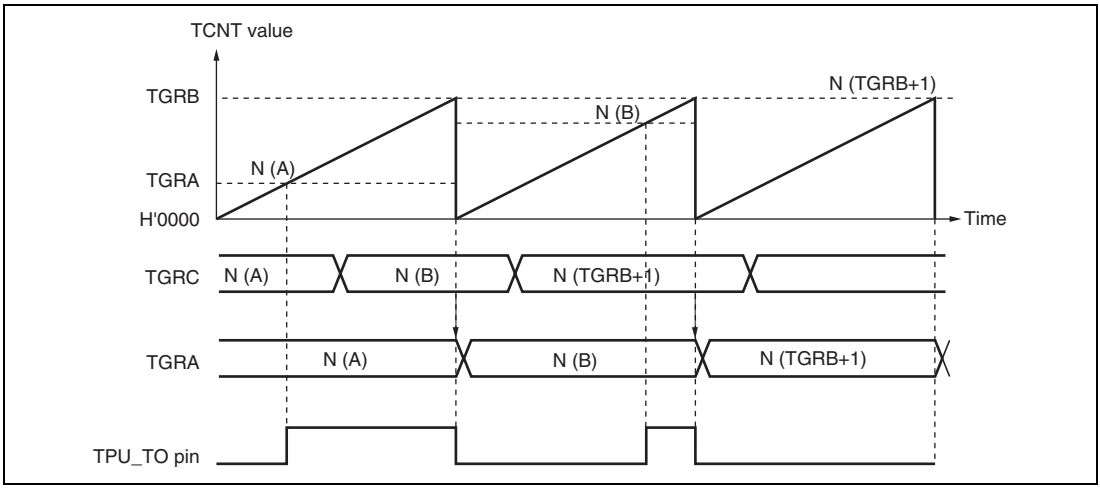


Figure 15.10 Example of Buffer Operation

15.4.4 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 TGRA.

Designating TGRB compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently.

PWM output is generated from the TPU_TO pin using TGRB as the period register and TGRA as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a period register compare match, the output value of each pin is the initial value set in TIOR. Set TIOR so that the initial output and an output value by compare match are different. If the same levels or toggle outputs are selected, operation is disabled.

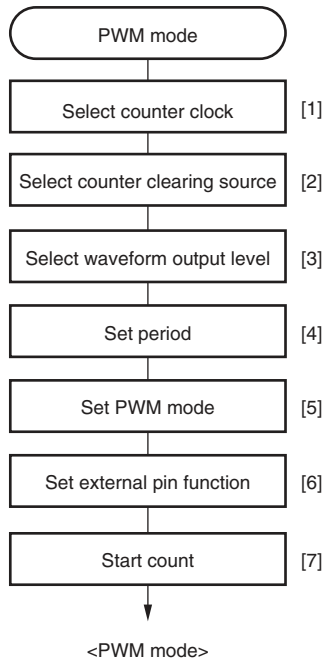
Conditions of duty 0% and 100% are shown below.

- Duty 0%: The set value of the duty register (TGRA) is TGRB + 1 for the period register(TGRB).
- Duty 100%: The set value of the duty register (TGRA) is 0.

In PWM mode 1, a maximum 4-phase PWM output is possible.

(1) Example of PWM Mode Setting Procedure

Figure 15.11 shows an example of the PWM mode setting procedure.



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

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

[3] Use TIOR to select the initial value and output value.

[4] Set the period in TGRB, and set the duty in TGRA.

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

[6] Set the external pin function in pin function controller (PFC).

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

Figure 15.11 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 15.12 shows an example of PWM mode operation.

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

In this case, the value set in TGRB is used as the period, and the value set in TGRA as the duty.

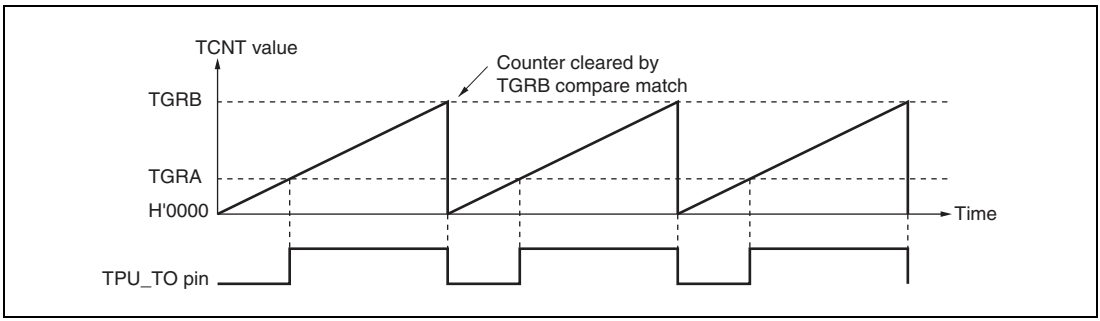


Figure 15.12 Example of PWM Mode Operation (1)

Figure 15.13 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

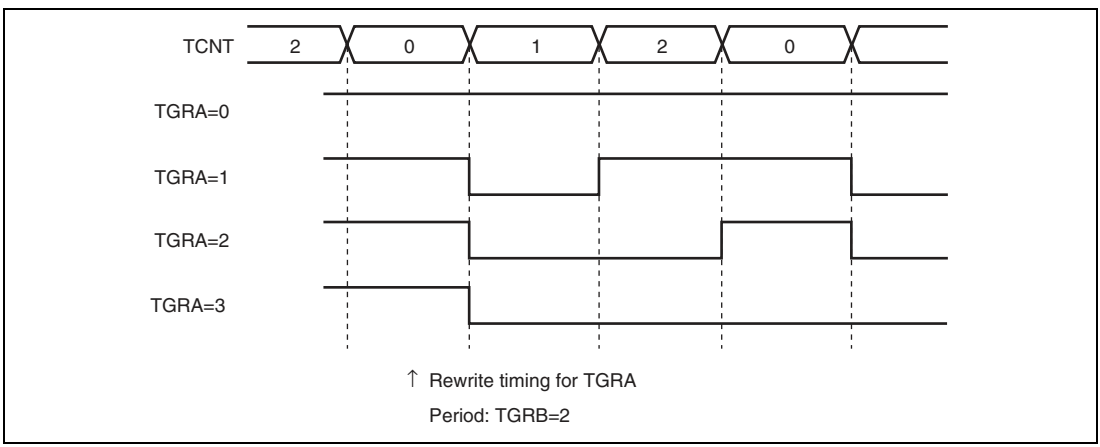


Figure 15.13 Examples of PWM Mode Operation (2)

15.4.5 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 2, and 3.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in TCR. However, the functions of bits CCLR1 and CCLR0 in TCR, and of TIOR, TIER, and TGR are valid, and compare match and interrupt functions can be used.

The previous set value (initial output value set before the timer was started in phase counting mode) is output from the TPU_TO pin in TIOR.

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

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.

Table 15.7 shows the correspondence between external clock pins and channels.

Table 15.7 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 2 is set to phase counting mode	TPU_TI2A	TPU_TI2B
When channel 3 is set to phase counting mode	TPU_TI3A	TPU_TI3B

(1) Example of Phase Counting Mode Setting Procedure

Figure 15.14 shows an example of the phase counting mode setting procedure.

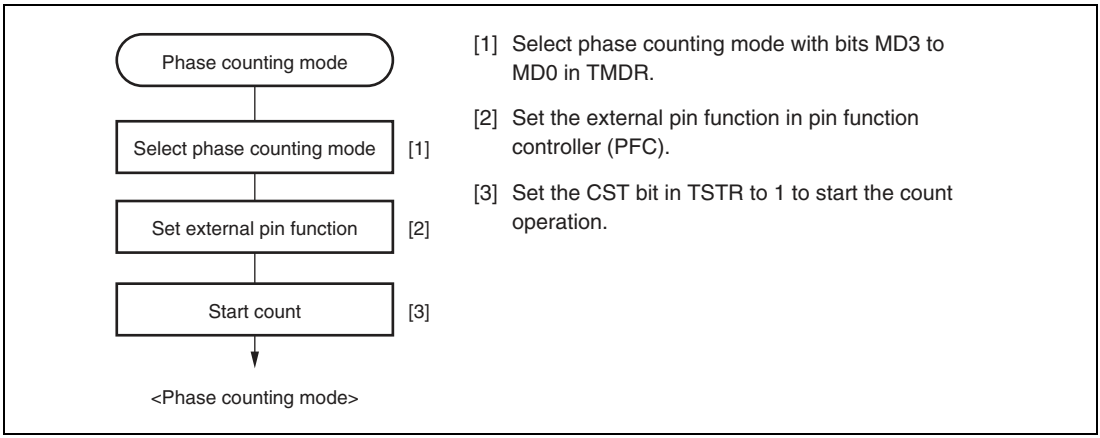


Figure 15.14 Example of Phase Counting Mode Setting Procedure

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

(a) Phase counting mode 1

Figure 15.15 shows an example of phase counting mode 1 operation, and table 15.8 summarizes the TCNT up/down-count conditions.

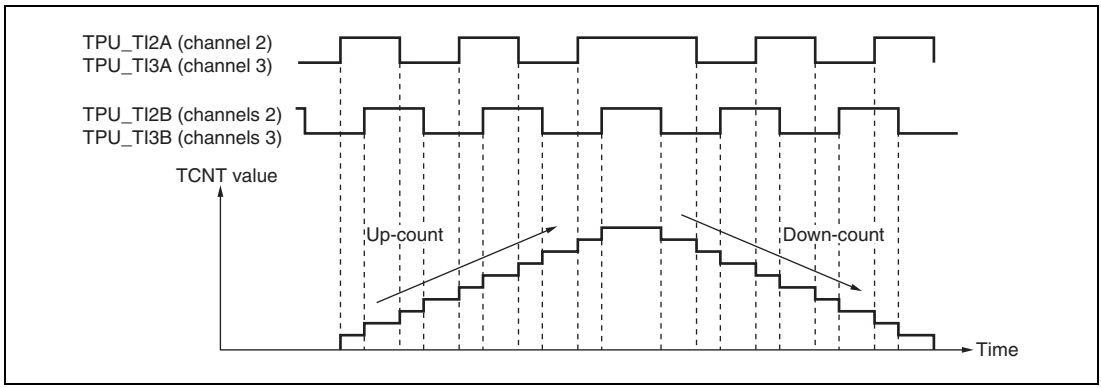


Figure 15.15 Example of Phase Counting Mode 1 Operation

Table 15.8 Up/Down-Count Conditions in Phase Counting Mode 1

TPU_TI2A (Channel 2) TPU_TI3A (Channel 3)	TPU_TI2B (Channel 2) TPU_TI3B (Channel 3)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	
↓	Low level	

[Legend]
↑ : Rising edge
↓ : Falling edge

(b) Phase counting mode 2

Figure 15.16 shows an example of phase counting mode 2 operation, and table 15.9 summarizes the TCNT up/down-count conditions.

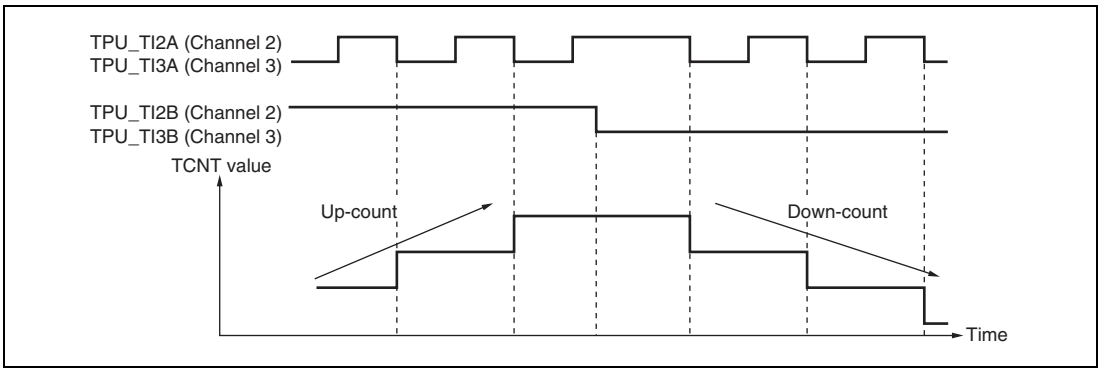


Figure 15.16 Example of Phase Counting Mode 2 Operation

Table 15.9 Up/Down-Count Conditions in Phase Counting Mode 2

TPU_TI2A (Channel 2) TPU_TI3A (Channel 3)	TPU_TI2B (Channel 2) TPU_TI3B (Channel 3)	Operation
High level	↑	Don't care
Low level	↓	
↑	Low level	
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	
↑	High level	
↓	Low level	Down-count

[Legend]
↑ : Rising edge
↓ : Falling edge

(c) Phase counting mode 3

Figure 15.17 shows an example of phase counting mode 3 operation, and table 15.10 summarizes the TCNT up/down-count conditions.

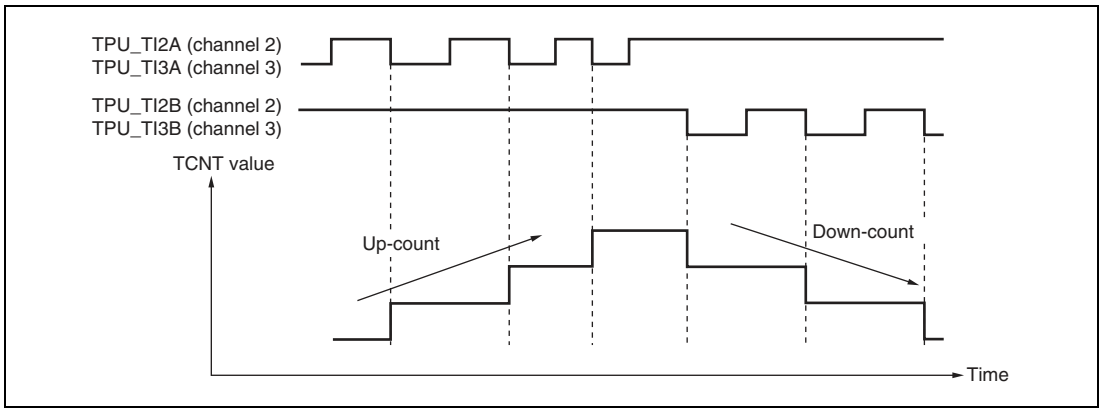


Figure 15.17 Example of Phase Counting Mode 3 Operation

Table 15.10 Up/Down-Count Conditions in Phase Counting Mode 3

TPU_TI2A (Channel 2) TPU_TI3A (Channel 3)	TPU_TI2B (Channel 2) TPU_TI3B (Channel 3)	Operation
High level	↑	Don't care
Low level	↓	
↑	Low level	
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	
↓	Low level	

[Legend]
↑ : Rising edge
↓ : Falling edge

(d) Phase counting mode 4

Figure 15.18 shows an example of phase counting mode 4 operation, and table 15.11 summarizes the TCNT up/down-count conditions.

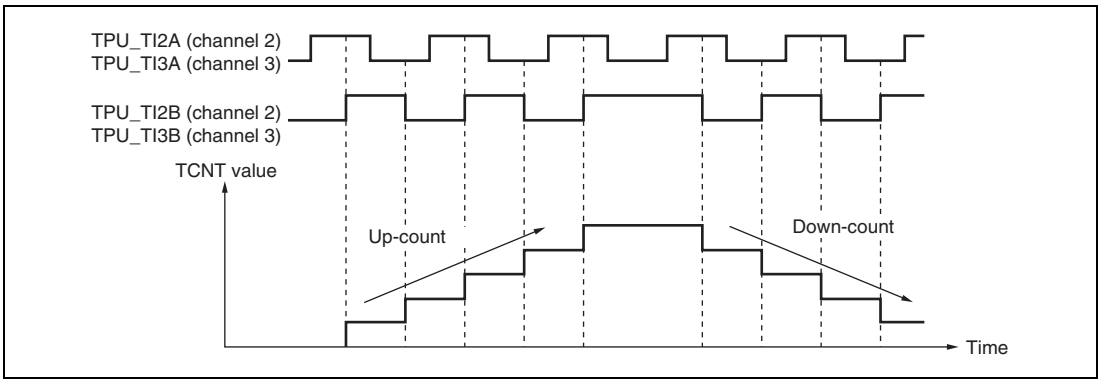


Figure 15.18 Example of Phase Counting Mode 4 Operation

Table 15.11 Up/Down-Count Conditions in Phase Counting Mode 4

TPU_TI2A (Channel 2) TPU_TI3A (Channel 3)	TPU_TI2B (Channel 2) TPU_TI3B (Channel 3)	Operation
High level	⌐↑	Up-count
Low level	⌐↓	
⌐↑	Low level	Don't care
⌐↓	High level	
High level	⌐↓	Down-count
Low level	⌐↑	
⌐↑	High level	Don't care
⌐↓	Low level	

[Legend]
⌐↑ : Rising edge
⌐↓ : Falling edge

15.5 Usage Notes

Note that the kinds of operation and contention described below can occur during TPU operation.

(1) Input Clock Restrictions

The input clock pulse width must be at least 2 states in the case of single-edge detection, and at least 3 states in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 2 states, and the pulse width must be at least 3 states. Figure 15.19 shows the input clock conditions in phase counting mode.

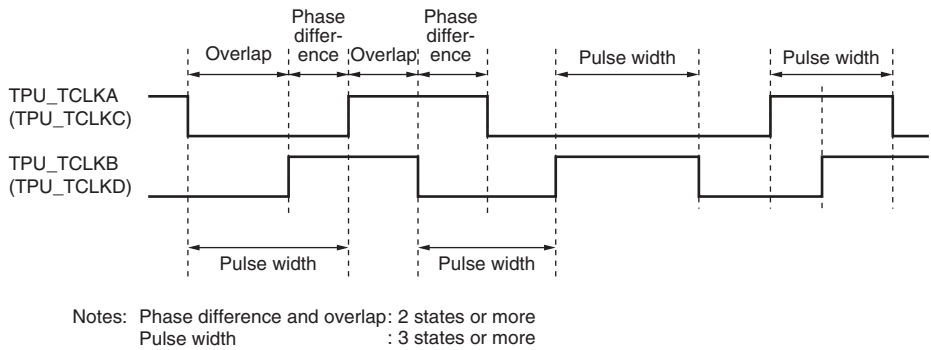


Figure 15.19 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

Section 16 Compare Match Timer (CMT)

This LSI includes a 32-bit compare match timer (CMT) of five channels (channel 0 to channel 4).

16.1 Features

- 16 bits/32 bits can be selected.
- Each channel is provided with an auto-reload up counter.
- All channels are provided with 32-bit constant registers and 32-bit up counters that can be written or read at any time.
- Allows selection among three counter input clocks for channel 0 to channel 4:
  - 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.
  A common interrupt vector (CMI) is assigned to each interrupt source.
- Generate a DMA transfer request when compare match or overflow occurs in channels 0 to 4.
- Module standby mode can be set.

Figure 16.1 shows a block diagram of the CMT.

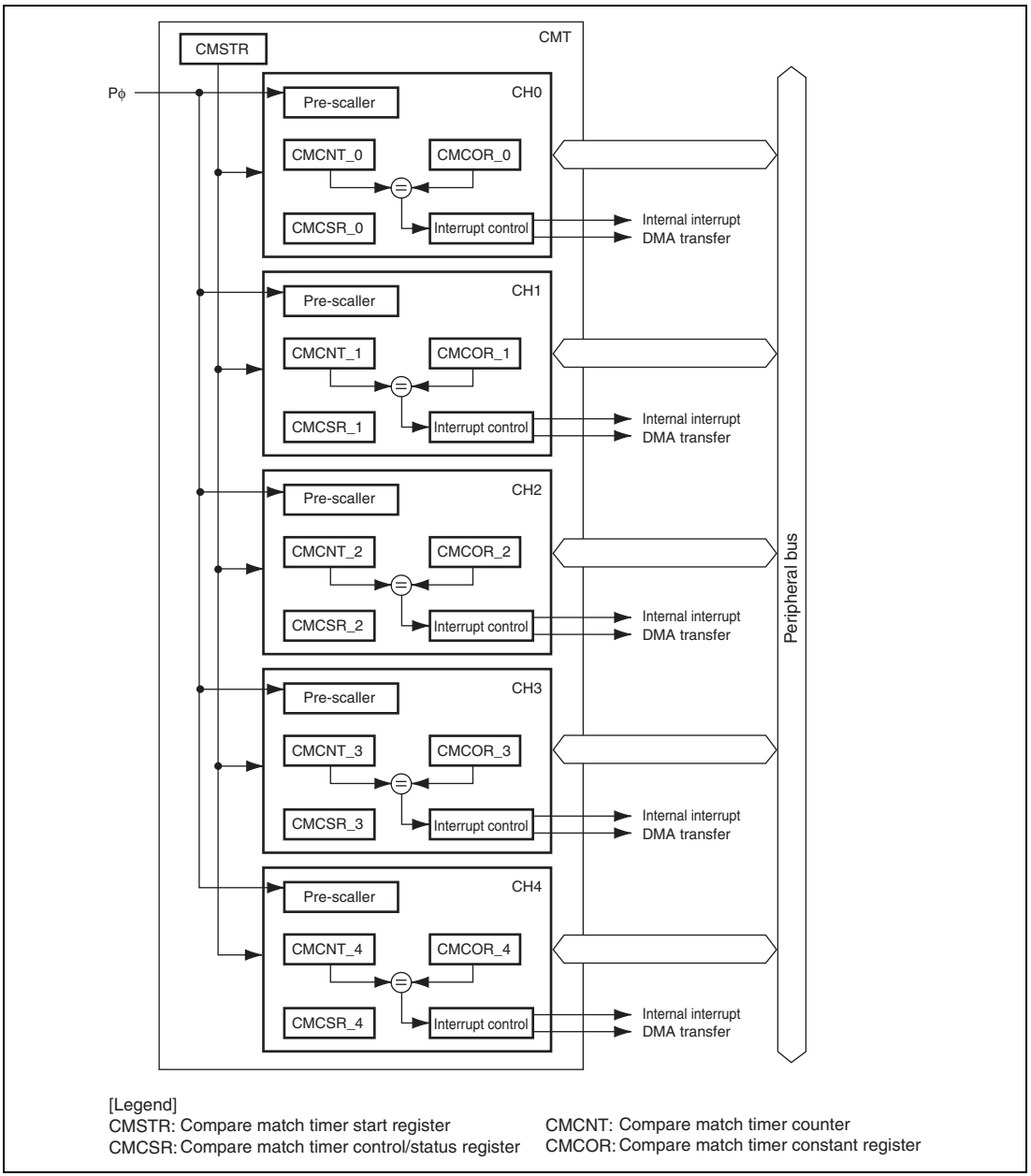


Figure 16.1 Block Diagram of CMT

16.2 Register Descriptions

The CMT has the following registers. Refer to section 37, List of Registers, for more details on the addresses and states of these registers in each operating mode. Notation for the CMT registers takes the form XXX_N, where XXX including the register name and N indicating the channel number. For example, CMCSR_0 denotes the CMCSR for channel 0.

(1) Common

- Compare match timer start register (CMSTR)

(2) Channel 0

- Compare match timer control/status register_0 (CMCSR_0)
- Compare match timer counter_0 (CMCNT_0)
- Compare match timer constant register_0 (CMCOR_0)

(3) Channel 1

- Compare match timer control/status register_1 (CMCSR_1)
- Compare match timer counter_1 (CMCNT_1)
- Compare match timer constant register_1 (CMCOR_1)

(4) Channel 2

- Compare match timer control/status register_2 (CMCSR_2)
- Compare match timer counter_2 (CMCNT_2)
- Compare match timer constant register_2 (CMCOR_2)

(5) Channel 3

- Compare match timer control/status register_3 (CMCSR_3)
- Compare match timer counter_3 (CMCNT_3)
- Compare match timer constant register_3 (CMCOR_3)

(6) Channel 4

- Compare match timer control/status register_4 (CMCSR_4)
- Compare match timer counter_4 (CMCNT_4)
- Compare match timer constant register_4 (CMCOR_4)

16.2.1 Compare Match Timer Start Register (CMSTR)

CMSTR is a 16-bit register that selects whether the compare match timer counter (CMCNT) is operated or halted.

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	STR4	0	R/W	Count Start 4 to 0 Selects whether to operate or halt the compare match timer counter for each channel (CMCNT_4 to CMCNT_0). 0: CMCNTn count operation halted 1: CMCNTn count operation n: 4 to 0 (corresponds to each channel)
3	STR3	0	R/W	
2	STR2	0	R/W	
1	STR1	0	R/W	
0	STR0	0	R/W	

16.2.2 Compare Match Timer Control/Status Register (CMCSR)

CMCSR is a 16-bit register that indicates the occurrence of compare matches, enables interrupts and DMA transfer request, and sets the counter input clocks.

Do not change bits other than bits CMF and OVF during the compare match timer counter (CMCNT) operation.

Bit	Bit Name	Initial Value	R/W	Description
15	CMF	0	R/(W)*[1]	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] • Write 0 to CMF after reading CMF=1 1: CMCNT and CMCOR values have matched
14	OVF	0	R/(W)*[1]	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] • Write 0 to OVF after reading 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.

Bit	Bit Name	Initial Value	R/W	Description
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
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	CMR1	0	R/W	Compare Match Request 1, 0
4	CMR0	0	R/W	Selects enable or disable for a DMA transfer request or internal interrupt request in a compare match. 00: Disables a DMA transfer request and internal interrupt request 01: Enables DMA transfer request 10: Enables an internal interrupt request 11: Setting prohibited
3	—	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
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	These bits select the clock input to CMCNT. When the STRn (n: 4 to 0) bit in CMSTR is set to 1, CMCNT begins incrementing with the clock selected by these bits.
0	CKS0	0	R/W	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.

16.2.3 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.

16.2.4 Compare Match Timer Constant Register (CMCOR)

CMCOR is a 32-bit register that sets the compare match period with CMCNT for each channel.

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.

16.3 Operation

16.3.1 Counter Operation

The CMT starts the operation of the counter by writing a 1 to the STRn bit in CMSTR of a channel that has been selected for operation. Complete all of the settings before starting the 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 setting 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 bits CMF and OVF in CMCSR are set to 1.

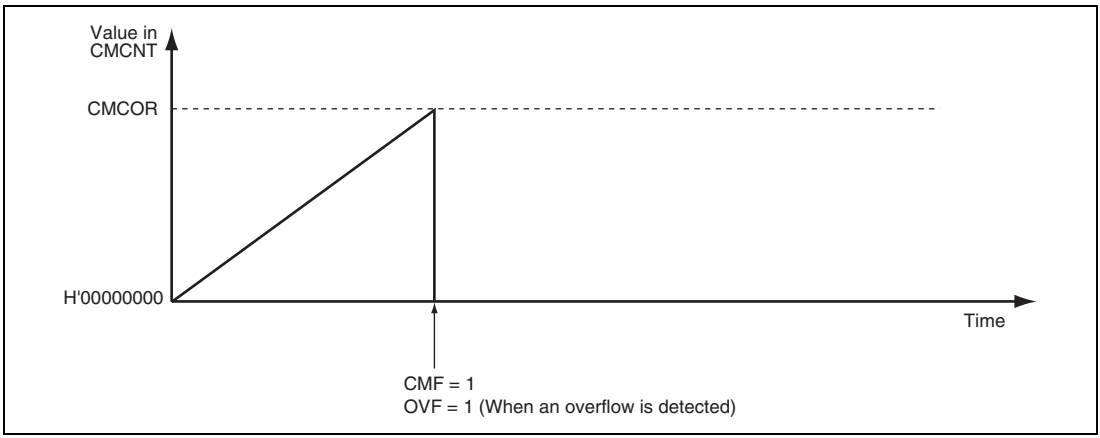


Figure 16.2 Counter Operation (One-Shot Operation)

- 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 bits CMF and OVF in CMCSR are set to 1.

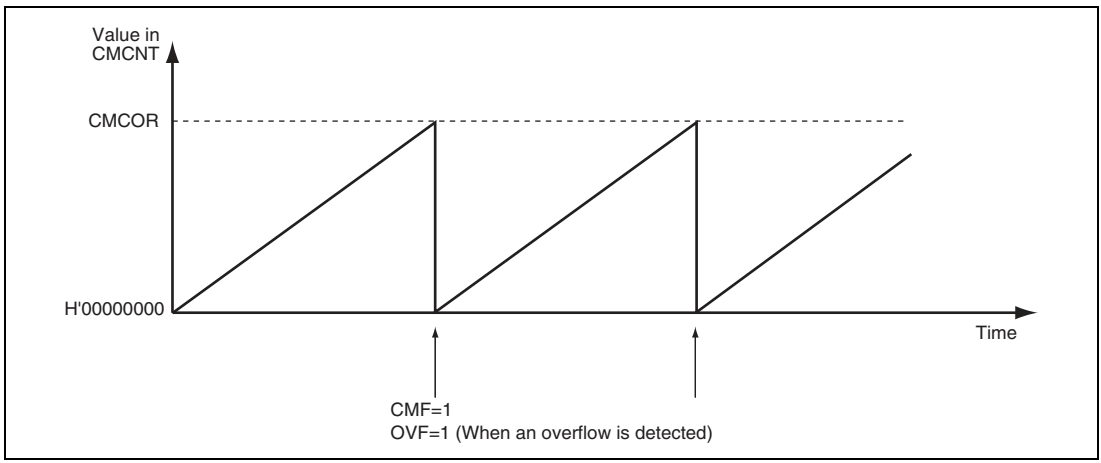


Figure 16.3 Counter Operation (Free-Running Operation)

16.3.2 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.

16.3.3 Timing for Counting by CMCNT

In this module, the clock for the counter can be selected from among the following:

- For channels 0 to 4:
  — Peripheral clock (Pϕ): 1/8, 1/32, or 1/128

The clock for the counter is selected by bits CKS2 to CKS0 in CMCSR. CMCNT is incremented at the rising edge of the selected clock.

16.3.4 DMA Transfer Requests and Internal Interrupt Requests to CPU

The setting of bits CMR1 and CMR0 in CMCSR selects the sending of a request for a DMA transfer or for an internal interrupt to the CPU at a compare match.

A DMA transfer request has different specifications according to the CMT channel as described below.

1. For channels 0 and 1, a single DMA transfer request is output at a compare match.
2. For channels 2 to 4, a DMA transfer request continues until the amount of data transferred has reached the value set in the DMAC, and the output of the request then automatically stops.

To clear the interrupt request, the CMF bit should be set to 0. Set the CMF bit to 0 in the handling routine for the CMT interrupt.

16.3.5 Compare Match Flag Set Timing (All Channels)

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 16.4 shows the set timing of the CMF bit.

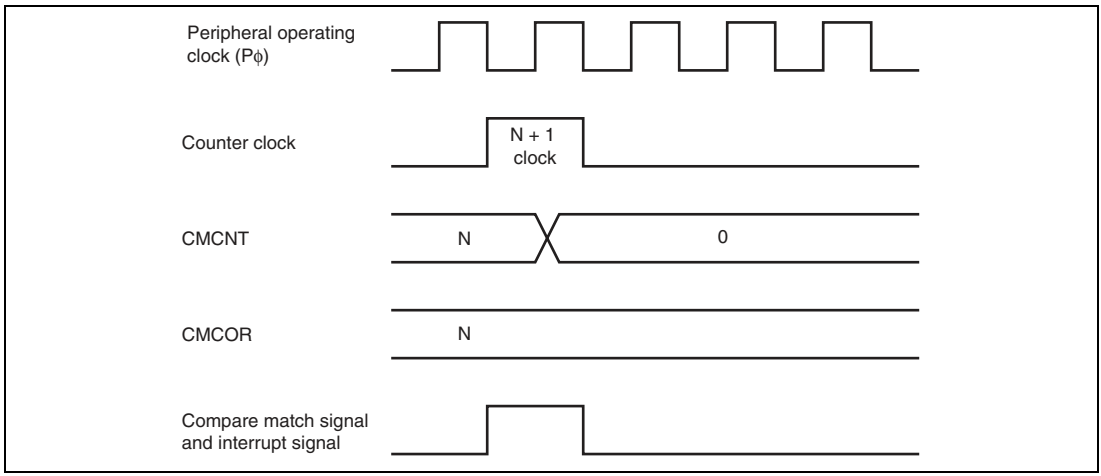


Figure 16.4 CMF Set Timing

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 (ATI): 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 (PRI): 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 (CUI): a carry interrupt indicates when a carry occurs during a counter read
- Automatic leap year adjustment

Figure 17.1 shows the block diagram of RTC.

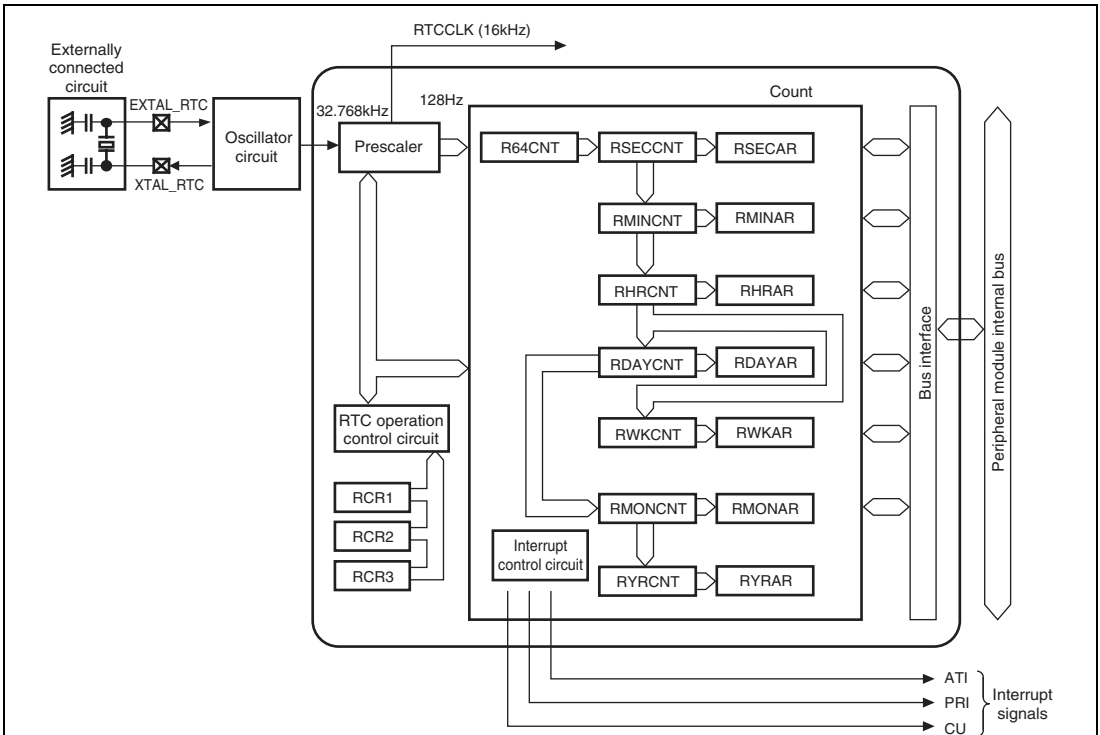


Figure 17.1 RTC Block Diagram

17.2 Input/Output Pin

Table 17.1 shows the RTC pin configuration.

Table 17.1 Pin Configuration

Name	Abbreviation	I/O	Description
RTC external clock	EXTAL_RTC	Input	Connects crystal resonator for RTC. Also used to input external clock for RTC.
RTC crystal	XTAL_RTC	Output	Connects crystal resonator for RTC.
RTC power supply	V_{CC}_RTC	—	Power-supply pin for RTC (1.5 V)*
RTC GND	V_{SS}_RTC	—	GND pin for RTC*
RTC power supply	V_{CC}Q_RTC	—	Power-supply pin for RTC (3.3 V)*

Note: * Power-supply pins for RTC should be power supplied even when the RTC is not used.

17.3 Register Descriptions

The RTC has the following registers. Refer to section 37, List of Registers, for more details on the addresses and access size of these registers.

- 64-Hz counter (R64CNT)
- Second counter (RSECCNT)
- Minute counter (RMINCNT)
- Hour counter (RHRCNT)
- Day of week counter (RWKCNT)
- Date counter (RDAYCNT)
- Month counter (RMONCNT)
- Year counter (RYRCNT)
- Second alarm register (RSECAR)
- Minute alarm register (RMINAR)
- Hour alarm register (RHRAR)
- Day of week alarm register (RWKAR)
- Date alarm register (RDAYAR)
- Month alarm register (RMONAR)
- Year alarm register (RYRAR)
- RTC control register 1 (RCR1)
- RTC control register 2 (RCR2)
- RTC control register 3 (RCR3)

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	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	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.

Though 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 coincide, 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 coincide, 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	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 coincide, 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	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	—	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 coincide, 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	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 coincide, 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	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	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, 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	Bit Name	Initial Value	R/W	Description
7	CF	Undefined	R/W	Carry Flag 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.

Bit	Bit Name	Initial Value	R/W	Description
4	CIE	0	R/W	Carry Interrupt Enable Flag 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 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
2, 1	—	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 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	Bit Name	Initial Value	R/W	Description
7	PEF	0	R/W	Periodic Interrupt Flag Indicates interrupt generation with the period designated by the PES2 to PES0 bits. When set to 1, PEF generates periodic interrupts. 0: Interrupts not generated with the period designated by the bits PES2 to PES0. [Clearing condition] When 0 is written to PEF 1: Interrupts generated with the period designated by the PES2 to PES0 bits. [Setting condition] When an interrupt is generated with the period designated by the bits PES0 to PES2 or when 1 is written to the PEF flag
6	PES2	0	R/W	Interrupt Enable Flags 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
5	PES1	0	R/W	
4	PES0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
3	RTCEN	1	R/W	Crystal Oscillator Control Controls the operation of the crystal oscillator for the RTC. 0: Halts the crystal oscillator for the RTC. 1: Runs the crystal oscillator for the RTC.
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.
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	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.

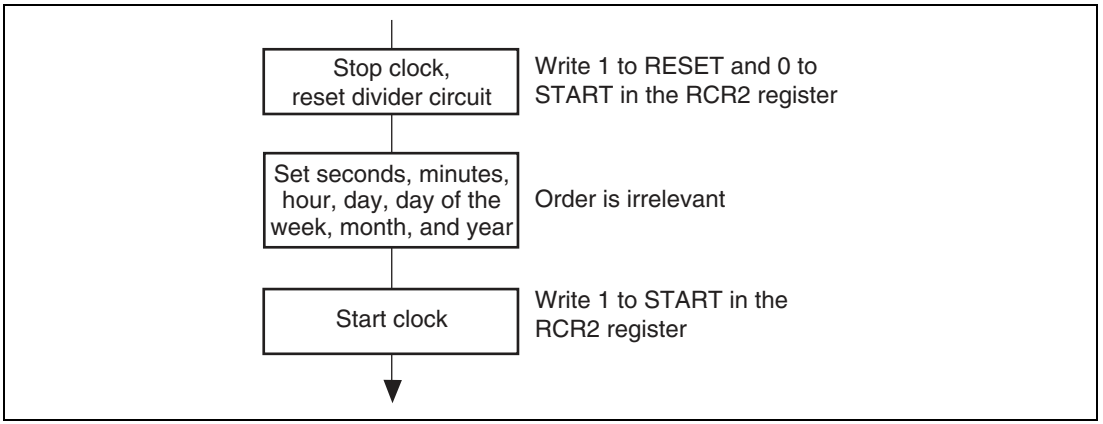


Figure 17.2 Setting Time

17.4.3 Reading Time

Figure 17.3 shows how to read the time.

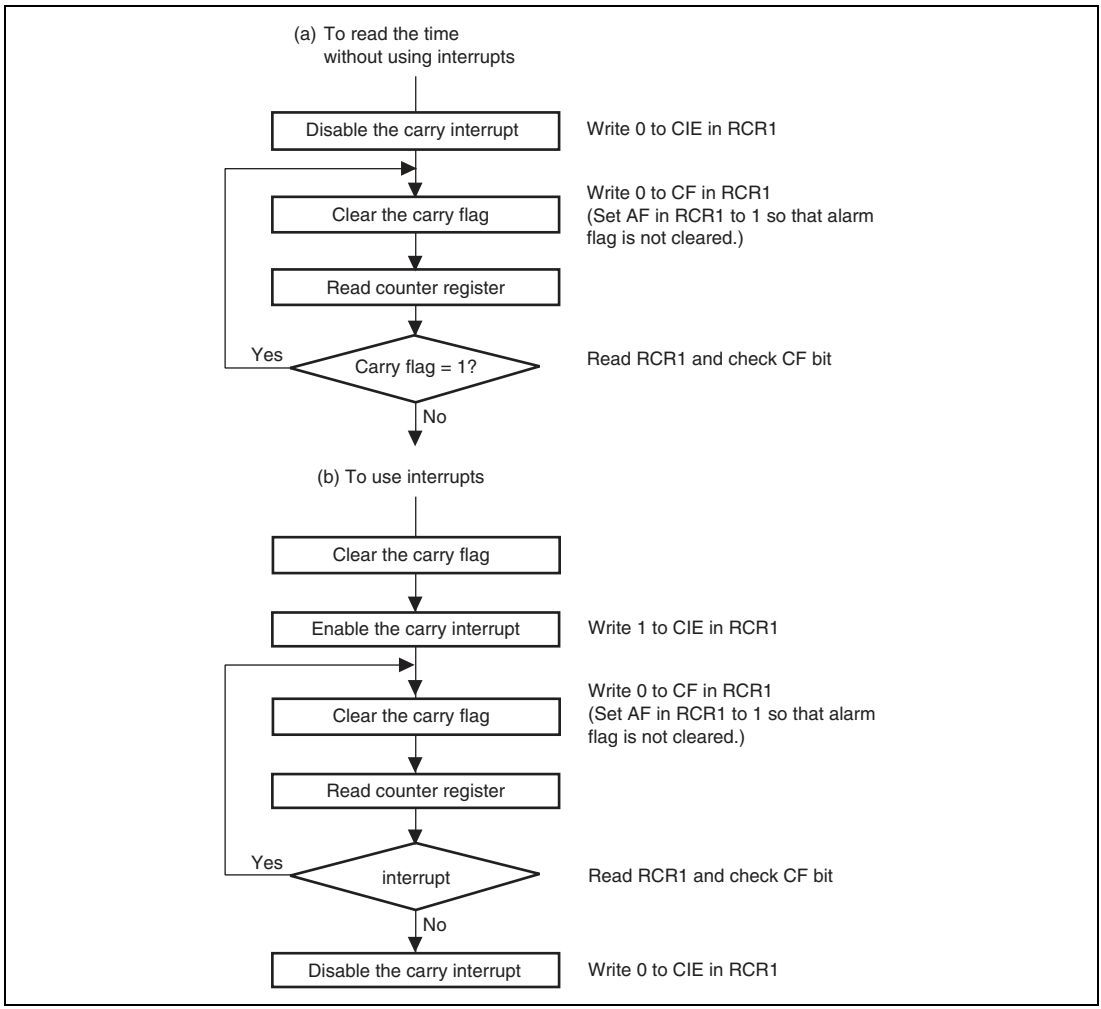


Figure 17.3 Reading Time

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.

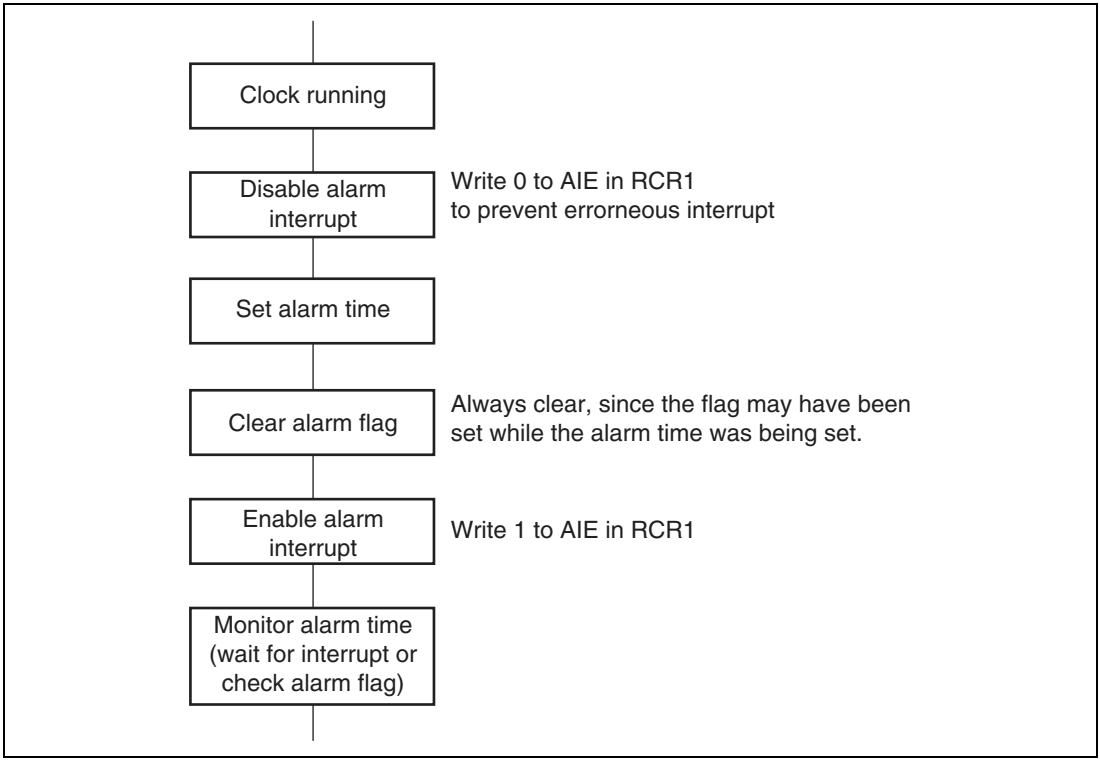


Figure 17.4 Using Alarm Function

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 PES0 to PES2 in RCR2. When the time set by the PES0 to PES2 has elapsed, the PEF is set to 1.

The PEF is cleared to 0 upon periodic interrupt generation or when the flags PES0 to PES2 are set. Periodic interrupt generation can be confirmed by reading this bit, but normally the interrupt function is used.

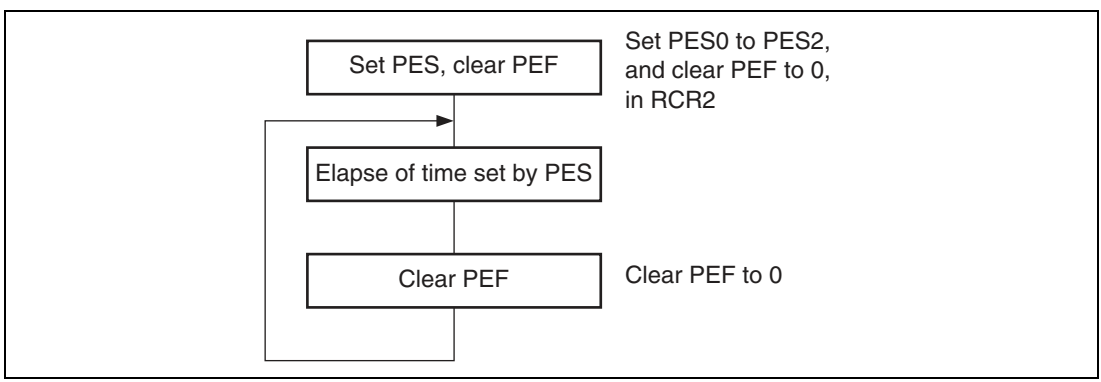


Figure 17.5 Using Periodic Interrupt Function

17.5.3 Transition to Standby Mode after Setting Register

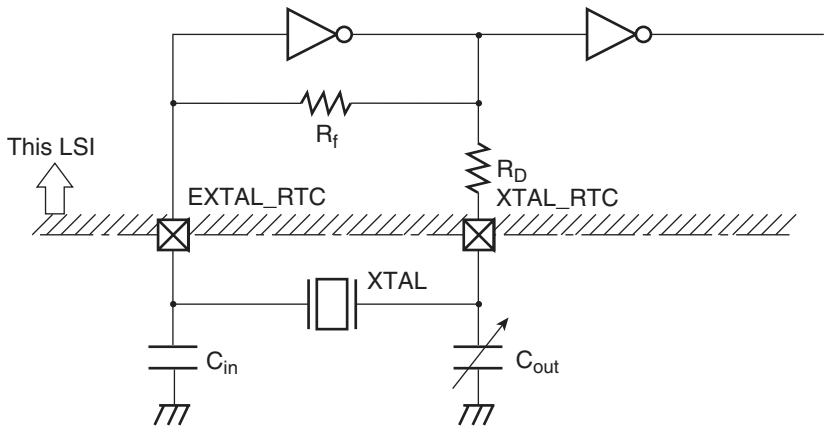
When a transition to standby mode is made after registers in the RTC are set, sometimes counting is not performed correctly. In case the registers are set, be sure to make a transition to standby mode after waiting for two RTC clocks or more.

17.5.4 Crystal Oscillator Circuit

Crystal oscillator circuit constants (recommended values) are shown in table 17.2, and the RTC crystal oscillator circuit in figure 17.5.

Table 17.2 Recommended Oscillator Circuit Constants (Recommended Values)

f_{osc}	C_{in}	C_{out}
32.768 kHz	10 to 22 pF	10 to 22 pF



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

Section 18 Serial Communication Interface with FIFO (SCIF)

This LSI has single-channel serial communication interface with FIFO (SCIF) that supports asynchronous serial communication. The SCIF can perform asynchronous and synchronous serial communication. It also has 64-stage FIFO registers for both transmission and reception that enable this LSI efficient high-speed continuous communication. Channel 0 operates as an IrDA interface while optional module IrDA is used.

18.1 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 (SCIFIn (n = 0, 1)) (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 (SCIFIn (n = 0, 1)) (synchronous mode)

  A common interrupt vector is assigned to each interrupt source.
- The direct memory access controller (DMAC) can be activated to execute a data transfer by a transmit-FIFO-data-empty or receive-FIFO-data-full interrupt.
- On-chip modem control functions (CTS and RTS)
- Transmit data stop function is available
- While the SCIF is not used, it can be stopped by stopping the clock for it to reduce power consumption.
- The number of data 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.
- Channel 0 operates as an IrDA interface.
- 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 18.1 shows the block diagram of SCIF.

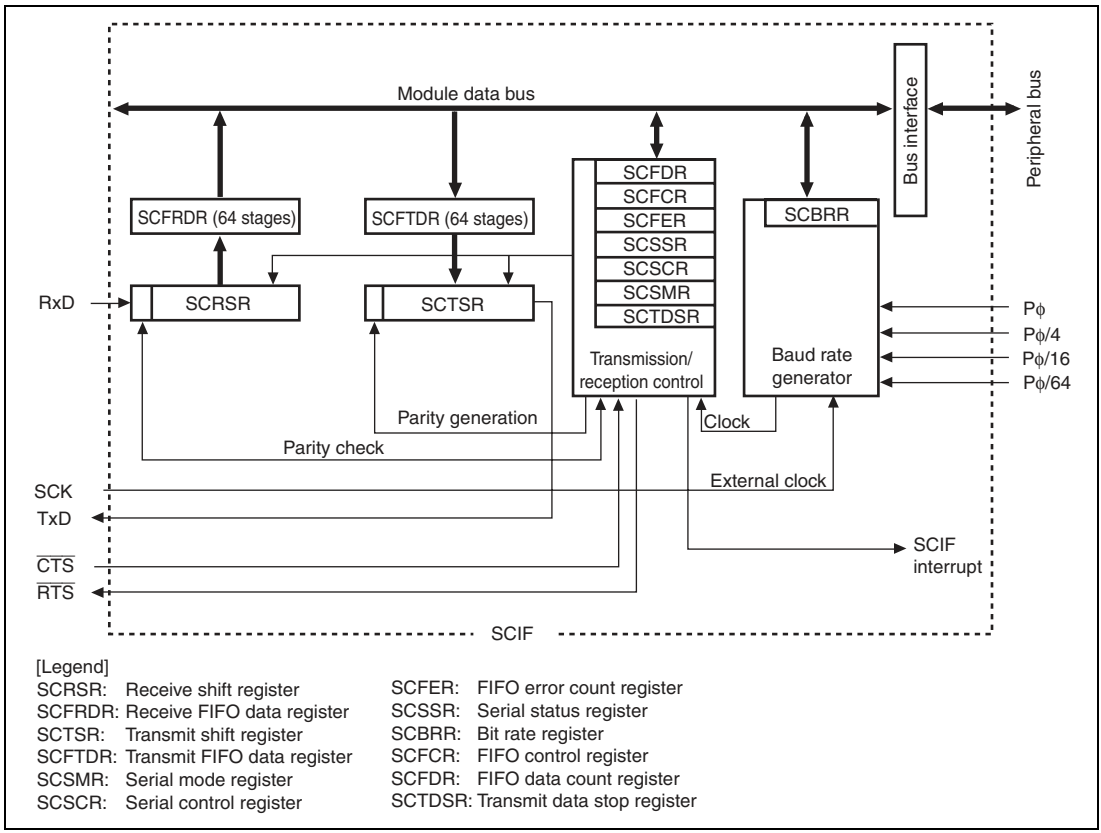


Figure 18.1 Block Diagram of SCIF

18.2 Input/Output Pins

Table 18.1 shows the pin configuration of SCIF.

Table 18.1 Pin configuration

Channel	Pin Name	Abbreviation*1	I/O	Function
0	SCIF0_SCK	SCK	Input*3	Clock input/output
	SCIF0_RxD	RxD	Input	Receive data input
	SCIF0_TxD	TxD	Output	Transmit data output
	$\overline{\text{SCIF0\_CTS}}$	$\overline{\text{CTS}}$*2	Input	Clear to send
	$\overline{\text{SCIF0\_RTS}}$	$\overline{\text{RTS}}$*2	Output	Request to send
1	SCIF1_SCK	SCK	Input*3	Clock input/output
	SCIF1_RXD	RxD	Input	Receive data input
	SCIF1_TXD	TxD	Output	Transmit data output
	$\overline{\text{SCIF1\_CTS}}$	$\overline{\text{CTS}}$*2	Input	Clear to send
	$\overline{\text{SCIF1\_RTS}}$	$\overline{\text{RTS}}$*2	Output	Request to send

Notes: 1. Pin names SCK, RxD, TxD, $\overline{\text{CTS}}$, and $\overline{\text{RTS}}$ are used in this manual for all channels, omitting the channel designation.

2. These pins are used as serial pins by setting the SCIF with the TE and RE bits in SCIF and the MCE bit in SCFCR.

3. The SCK pin can be set as input (input enabled or disabled).

18.3 Register Descriptions

SCIF has the following registers. Refer to section 37, List of Registers, for more details on the addresses and states of these registers in each operating mode. Note that the channel number of each register is omitted.

(1) Channel 0

- Receive shift register_0 (SCRSR_0)
- Receive FIFO data register_0 (SCFRDR_0)
- Transmit shift register_0 (SCTSR_0)
- Transmit FIFO data register_0 (SCFTDR_0)
- Serial mode register_0 (SCSMR_0)
- Serial control register_0 (SCSCR_0)
- FIFO error count register_0 (SCFER_0)
- Serial status register_0 (SCSSR_0)
- Bit rate register_0 (SCBRR_0)
- FIFO control register_0 (SCFCR_0)
- FIFO data count register_0 (SCFDR_0)
- Transmit data stop register_0 (SCTDSR_0)

(2) Channel 1

- Receive shift register_1 (SCRSR_1)
- Receive FIFO data register_1 (SCFRDR_1)
- Transmit shift register_1 (SCTSR_1)
- Transmit FIFO data register_1 (SCFTDR_1)
- Serial mode register_1 (SCSMR_1)
- Serial control register_1 (SCSCR_1)
- FIFO error count register_1 (SCFER_1)
- Serial status register_1 (SCSSR_1)
- Bit rate register_1 (SCBRR_1)
- FIFO control register_1 (SCFCR_1)
- FIFO data count register_1 (SCFDR_1)
- Transmit data stop register_1 (SCTDSR_1)

18.3.1 Receive Shift Register (SCRSR)

SCRSR receives serial data. Data input at the RxD pin is loaded into the 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 the SCFRDR, which is a receive FIFO data register. The CPU cannot read from or write to the SCRSR directly.

18.3.2 Receive FIFO Data Register (SCFRDR)

The 64-byte receive FIFO data register (SCFRDR) stores serial receive data. The SCIF completes the reception of one byte of serial data by moving the received data from the receive shift register (SCRSR) into the SCFRDR for storage. Continuous receive is enabled until 64 bytes are stored.

The CPU can read but not write the SCFRDR. When data is read without received data in the SCFRDR, 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	SCFRD7 to SCFRD0	Undefined	R	FIFO Data Registers for Serial Receive Data

18.3.3 Transmit Shift Register (SCTSR)

SCTSR transmits serial data. The SCIF loads transmit data from the transmit FIFO data register (SCFTDR) into the SCTSR, 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 the SCFTDR into the SCTSR and starts transmitting again. The CPU cannot read or write the SCTSR directly.

18.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is a 64-byte 8-bit-length FIFO register that stores data for serial transmission. When the SCIF detects that the transmit shift register (SCTSR) is empty, it moves transmit data written in the SCFTDR into the SCTSR and starts serial transmission. Continuous serial transmission is performed until the transmit data in the SCFTDR becomes empty. The CPU can always write to the SCFTDR.

When the transmit data in the SCFTDR is full (64 bytes), next data cannot be written. If attempted to write, the data is ignored.

Bit	Bit Name	Initial value	R/W	Description
7 to 0	SCFTD7 to SCFTD0	Undefined	R	FIFO Data Registers for Serial Transmit Data

18.3.5 Serial Mode Register (SCSMR)

SCSMR is a 16-bit register that specifies the SCIF serial communication format and selects the clock source for the baud rate generator and the sampling rate.

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.
10	SRC2	0	R/W	Sampling Control 2 to 0
9	SRC1	0	R/W	Select sampling rate.
8	SRC0	0	R/W	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	C/A	0	R/W	Communication Mode Selects whether the SCI operates in the asynchronous or synchronous mode. 0: Asynchronous mode 1: Synchronous mode

Bit	Bit Name	Initial Value	R/W	Description
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 SCFTDR is not transmitted.
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 (O/E) setting. Receive data parity is checked according to the even/odd (O/E) mode setting.

Bit	Bit Name	Initial Value	R/W	Description
4	O/E	0	R/W	Parity Mode Selects even or odd parity when parity bits are added and checked. The O/E setting is used only when the PE is set to 1 to enable parity addition and check. The O/E 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	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	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: In synchronous mode, bits other than CKS1 and CKS0 are fixed 0.

18.3.6 Serial Control Register (SCSCR)

SCSCR 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	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 SCFCR is enabled and the TSF flag in SCSSR 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 SCSSR 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 SCSSR 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 SCSSR 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 SCSSR 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 than the specified number of transmission triggers to SCFTDR 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 SCSSR 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 SCIF serial transmitter. 0: Transmitter disabled 1: Transmitter enabled* Note: * The serial mode register (SCSMR) and FIFO control register (SCFCR) 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 SCIF 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. The serial mode register (SCSMR) and FIFO control register (SCFCR) 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	CKE1	0	R/W	Clock Enable 1and 0
0	CKE0	0	R/W	These bits select the SCIF clock source. The bits CKE1 and CKE0 should be set before selecting the SCIF operating mode by SCSMR. 00: Internal clock, SCK pin used for input pin (input signal is ignored)*1 01: Internal clock, SCK pin used for synchronous clock output*2 10: External clock, SCK pin used for clock input*3 11: External clock, SCK pin used for clock input*3 Notes: 1. When the data sampling is executed using on-chip baud rate generator, CKE1 and CKE0 should be set to 00. 2. In synchronous mode, a clock with a frequency equal to the bit rate is output. When the channel 0 is used as the IrDA interface, CKE1 and CKE0 should be set to 01. 3. In asynchronous mode, input the clock which is appropriate for the sampling rate. For example, when the sampling rate is 1/16, input the clock frequency 8 times the bit rate. When the external clock is not input, CKE1 and CKE0 should be set to 00. When the SCK pin is set as an I/O port pin, CKE1 and CKE0 should be set to 00.

18.3.7 FIFO Error Count Register (SCFER)

SCFER is a 16-bit read-only register that indicates the number of receive data errors (framing error/parity error).

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	PER5	0	R	Parity Error Indicates the number of data, in which parity errors are generated, in receive data stored in the receive FIFO data register (SCFRDR) in asynchronous mode. Bits 13 to 8 indicate the number of data with parity errors after the ER bit in SCSSR is set. If all 64-byte receive data in SCFRDR have parity errors, bits PER5 to PER0 indicate 0s.
12	PER4	0	R	
11	PER3	0	R	
10	PER2	0	R	
9	PER1	0	R	
8	PER0	0	R	
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	FER5	0	R	Framing Error Indicates the number of data, in which framing errors are generated, in receive data stored in the receive FIFO data register (SCFRDR) in asynchronous mode. Bits 5 to 0 indicate the number of data with framing errors after the ER bit in SCSSR is set. If all 64-byte receive data in SCFRDR have framing errors, bits FER5 to FER0 indicate 0s.
4	FER4	0	R	
3	FER3	0	R	
2	FER2	0	R	
1	FER1	0	R	
0	FER0	0	R	

18.3.8 Serial Status Register (SCSSR)

SCSSR is a 16-bit readable/writable register that indicates SCIF 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	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 SCSCR is cleared to 0, the ORER flag is not affected and retains its previous state. 2. SCFRDR 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 matches the value set in SCTDSR. 0: Transmit data number does not match the value set in SCTDSR [Clearing conditions] • Power-on reset, manual reset • Writing 0 after reading TSF = 1 1: Transmit data number matches the value set in SCTDSR
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 O/$\overline{E}$ bit in the SCSMR. Notes: 1. Indicates clearing the RE bit to 0 in SCSCR does not affect the ER bit, which retains its previous value. Even if a receive error occurs, the received data is transferred to SCFRDR 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 SCSSR. 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, the SCFTDR did not contain valid data, so transmission has ended. 0: Transmission is in progress [Clearing condition] Data is written to SCFTDR. 1: End of transmission [Setting condition] SCFTDR 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 the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), the number of data in SCFTDR becomes less than the number of transmission triggers specified by the TTRG1 and TTRG0 bits in the FIFO control register (SCFCR), and writing the transmit data to SCFTDR is enabled. 0: The number of transmit data written to SCFTDR is greater than the specified number of transmission triggers [Clearing condition] Data exceeding the specified number of transmission triggers is written to SCFTDR, software reads TDFE after it has been set to 1, then writes 0 to TDFE. 1: The number of transmission data in SCFTDR becomes less than the specified number of transmission triggers [Setting conditions] • Power-on reset, manual reset • The number of transmission data in SCFTDR becomes less than the specified number of transmission triggers as a result of transmission* Note: * Since SCFTDR is a 64-byte FIFO register, the maximum number of data which can be written when TDFE is 1 is "64 minus the specified number of transmission triggers". If attempted to write additional data, the data is ignored. The number of data in SCFTDR is indicated by SCFDR.

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 SCFRDR 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 the receive FIFO data register (SCFRDR) in asynchronous mode. 0: No framing error occurred in the data read from SCFRDR [Clearing conditions] • Power-on reset, manual reset • No framing error is present in the data read from SCFRDR 1: A framing error occurred in the data read from SCFRDR [Setting condition] • A framing error is present in the data read from SCFRDR

Bit	Bit Name	Initial Value	R/W	Description
2	PER	0	R	Parity Error Indicates a parity error in the data read from the receive FIFO data register (SCFRDR) in asynchronous mode. 0: No parity error occurred in the data read from SCFRDR [Clearing conditions] • Power-on reset, manual reset • No parity error is present in the data read from SCFRDR 1: A parity error occurred in the data read from SCFRDR [Setting condition] • A parity error is present in the data read from SCFRDR

Bit	Bit Name	Initial Value	R/W	Description
1	RDF	0	R/(W)*	Receive FIFO Data Full Indicates that received data is transferred to the receive FIFO data register (SCFRDR), the number of data in SCFRDR becomes more than the number of receive triggers specified by the RTRG1 and RTRG0 bits in SCFCR. 0: The number of transmit data written to SCFRDR is less than the specified number of receive triggers [Clearing conditions] • Power-on reset, manual reset • SCFRDR is read until the number of receive data in SCFRDR becomes less than the specified number of receive triggers, and RDF is read as 1, then written to with 0. 1: The number of receive data in SCFRDR is more than the specified number of receive triggers [Setting condition] The number of receive data which is greater than the specified number of receive triggers is being stored to SCFRDR.* Note: * Since SCFTDR is a 64-byte FIFO register, the maximum number of data which can be read when RDF is 1 is the specified number of receive triggers. If attempted to read after all data in SCFRDR have been read, the data is undefined. The number of receive data in SCFRDR is indicated by the lower bits of SCFTDR.

Bit	Bit Name	Initial Value	R/W	Description
0	DR	0	R/(W)*	Receive Data Ready Indicates that the receive FIFO data register (SCFRDR) stores the data which is less than the specified number of receive triggers, 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 SCFRDR after the receive ended normally. [Clearing conditions] (Initial value) • Power-on reset, manual reset • All receive data in SCFRDR is read, and DR is read as 1, then written to with 0. 1: Next receive data is not received [Setting condition] SCFRDR stores the data which is less than the specified number of receive triggers, 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.

18.3.9 Bit Rate Register (SCBRR)

SCBRR is an eight-bit readable/writable register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SCSMR), determines the serial transmit/receive bit rate.

Bit	Bit Name	Initial value	R/W	Description
7 to 0	SCBRD7 to SCBRD0	H'FF	R/W	Bit Rate Set

The SCBRR 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 1/11

$$N = \frac{P\phi}{22 \times 2^{2n-1} \times B} \times 10^6 - 1$$

4. When sampling rate is 1/13

$$N = \frac{P\phi}{26 \times 2^{2n-1} \times B} \times 10^6 - 1$$

5. 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: SCBRR 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.)

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

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 1/11

$$\text{Error (\%)} = \left(\frac{P\phi \times 10^6}{(1+N) \times B \times 22 \times 2^{2n-1}} - 1 \right) \times 100$$

4. When sampling rate is 1/13

$$\text{Error (\%)} = \left(\frac{P\phi \times 10^6}{(1+N) \times B \times 26 \times 2^{2n-1}} - 1 \right) \times 100$$

5. When sampling rate is 1/27

$$\text{Error (\%)} = \left(\frac{P\phi \times 10^6}{(1+N) \times B \times 58 \times 2^{2n-1}} - 1 \right) \times 100$$

18.3.10 FIFO Control Register (SCFCR)

SCFCR is a 16-bit readable/writable register that resets the number of data in the transmit and receive FIFO registers, sets the number of trigger data, and contains an enable bit for the loop back test.

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.
13 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	RSTRG2	0	R/W	Trigger of the $\overline{\text{RTS}}$ Output Active 2 to 0
9	RSTRG1	0	R/W	The $\overline{\text{RTS}}$ signal goes to high, when the number of receive data count stored in the receive FIFO data register (SCFRDR) is increased more than the number of setting triggers listed below.
8	RSTRG0	0	R/W	000: 63 001: 1 010: 8 011: 16 100: 32 101: 48 110: 54 111: 60

Bit	Bit Name	Initial Value	R/W	Description
7	RTRG1	0	R/W	Trigger of the Number of Receive FIFO Data 1, 0
6	RTRG0	0	R/W	Set the number of receive data which sets the receive data full (RDF) flag in the serial status register (SCSSR). These bits set the RDF flag when the number of receive data stored in the receive FIFO data register (SCFRDR) is increased more than the number of setting triggers listed below. 00: 1 01: 16 10: 32 11: 48
5	TTRG1	0	R/W	Trigger of the Number of Transmit FIFO Data 1, 0
4	TTRG0	0	R/W	Set the number of remaining transmit data which sets the transmit FIFO data register empty (TDFE) flag in the serial status register (SCSSR). These bits set the TDFE flag when the number of transmit data in the transmit FIFO data register (SCFTDR) is decreased less than the number of setting triggers 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 SCFTDR 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.

Bit	Bit Name	Initial Value	R/W	Description
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.
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

18.3.11 FIFO Data Count Register (SCFDR)

SCFDR is a 16-bit register which indicates the number of data stored in the receive FIFO data register (SCFRDR). The SCFDR is always read from the CPU.

The bits 14 to 8 of this register indicate the number of transmit data items stored in the SCFTDR that have not yet been transmitted.

The bits 6 to 0 of this register indicate the number of receive data items stored in the SCFRDR.

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	T6	0	R	These bits indicate the number of non-transmitted data stored in the SCFTDR. The H'00 means no transmit data, and the H'40 means that the full of transmit data are stored in the SCFTDR.
13	T5	0	R	
12	T4	0	R	
11	T3	0	R	
10	T2	0	R	
9	T1	0	R	
8	T0	0	R	
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	R6	0	R	These bits indicate the number of receive data stored in the SCFRDR. The H'00 means no receive data, and the H'40 means that the full of receive data are stored in the SCFRDR.
5	R5	0	R	
4	R4	0	R	
3	R3	0	R	
2	R2	0	R	
1	R1	0	R	
0	R0	0	R	

18.3.12 Transmit Data Stop Register (SCTDSR)

SCTDSR is an 8-bit readable/writable register that sets the number of data to be transmitted. This register is available when the TSE bit in the FIFO control register (SCFCR) 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.

18.4 Operation

For serial communication, the SCIF has asynchronous mode in which characters are synchronized individually and synchronous mode in which synchronization is achieved with clock pulses. The SCIF has the 64-byte FIFO buffer for both transmission and reception, reduces an overhead of the CPU, and enables continuous high-speed communication.

18.4.1 Asynchronous Mode

Operation in asynchronous mode is described below.

The transmission and reception format is selected in the serial mode register (SCSMR), as listed in table 18.2. The clock source of SCIF is determined by the combination of CKE1 and CKE0 bits in the serial control register (SCSCR).

- 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: SCIF operates using the on-chip baud rate generator
  - — External clock: The clock appropriate for the sampling rate should be input. For example, when the sampling rate is 1/16, input the clock frequency 8 times the bit rate. (The internal baud rate generator should not be used.)

Table 18.2 SCSMR Settings and SCIF Transmit/Receive

SCSMR Settings				SCIF Transmit/Receive			
Bit 6	Bit 5	Bit 3					
CHR	PE	STOP	Mode	Data Length	Multi-processor Bit	Parity Bit	Stop Bit Length
0	0	0	Asynchro-nous mode	8-bit data	Not set	Not set	1 bit
		1					2 bits
	1	0				Set	1 bit
		1					2 bits
1	0	0		7-bit data		Not set	1 bit
		1					2 bits
	1	0				Set	1 bit
		1					2 bits

18.4.2 Serial Operation

(1) Transmit/Receive Formats

Table 18.3 lists eight communication formats that can be selected. The format is selected by settings in the serial mode register (SCSMR).

Table 18.3 Serial Transmit/Receive Formats

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	

(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 SCI's serial clock, according to the setting of the CKE bit in the serial control register (SCSCR).

When an external clock is input at the SCK pin, the clock appropriate for the sampling rate should be input. For example, when the sampling rate is 1/16, the clock frequency should be 8 times the bit rate used.

(3) Transmitting and Receiving Data (SCIF Initialization)

Before transmitting or receiving, clear the TE and RE bits to 0 in SCSCR, then initialize the SCIF 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 the transmit shift register (SCTSR). Clearing TE and RE to 0, however, does not initialize the serial status register (SCSSR), transmit FIFO data register (SCFTDR), or receive FIFO data register (SCFRDR), which retain their previous contents.

Clear TE to 0 after all transmit data are transmitted and the TEND bit in the SCSSR 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 SCFCR to 1 and reset the SCFTDR before TE is set again to start transmission.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCIF operation becomes unreliable if the clock is stopped.

Figure 18.2 is a sample flowchart for initializing the SCIF.

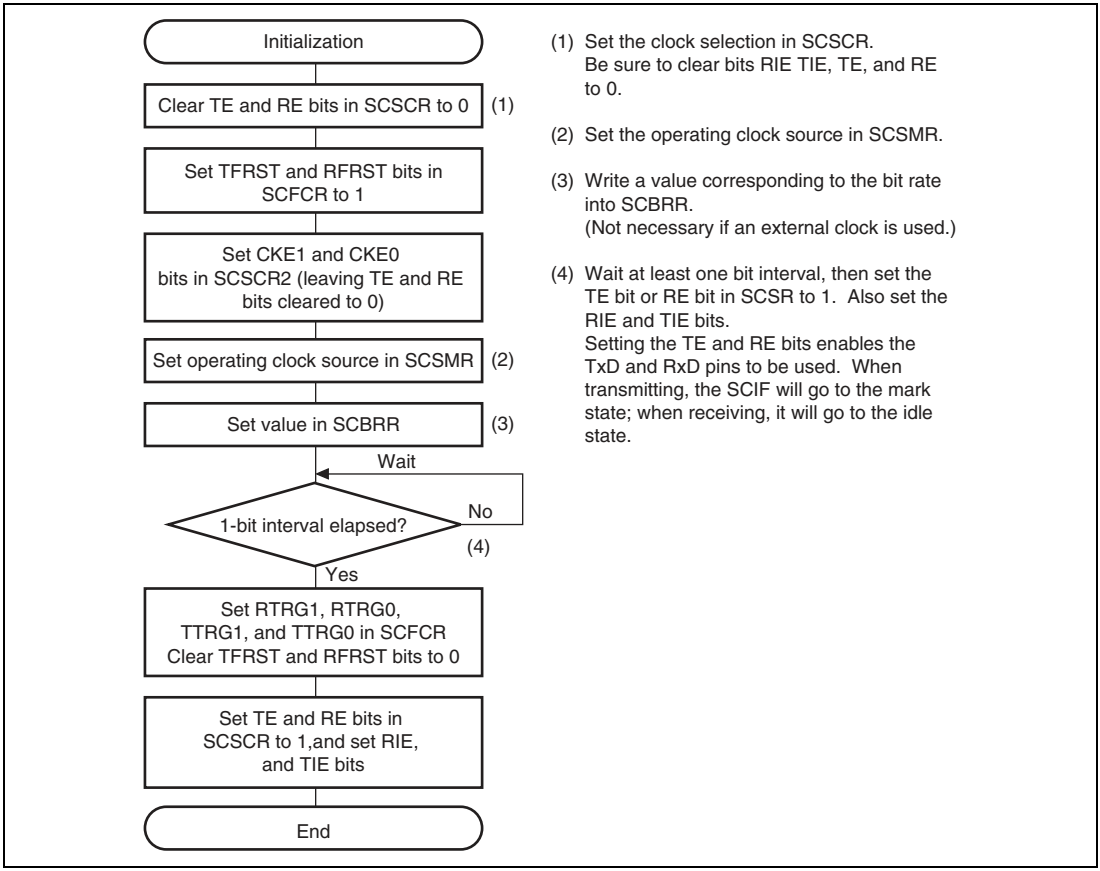
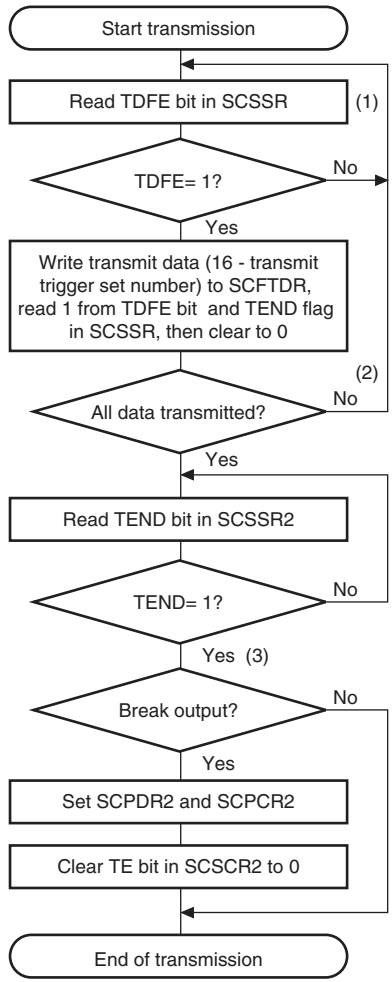


Figure 18.2 Sample SCIF Initialization Flowchart

(4) Transmitting and Receiving Data (Serial data transmission)

Figure 18.3 shows a sample serial transmission flowchart. After SCIF transmission is enabled, use the following procedure to perform serial data transmission.



(1) SCIF status check and transmit data write:
Read serial status register (SCSSR) and check that the TDFE flag is set to 1, then write transmit data to the transmit FIFO data register (SCFTDR), 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 set number).

(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 SCFTDR2, 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 SC data register (SCPDR) and port SC control register (SCPCR), then clear the TE bit to 0 in the serial control register (SCSCR).
In steps 1 and 2, it is possible to ascertain the number of data bytes that can be written from the number of transmit data bytes in SCFTDR indicated by the upper 8 bits of the FIFO data count set register 2 (SCFDR).

Figure 18.3 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into the transmit FIFO data register (SCFTDR), the SCIF transfers the data from SCFTDR to the transmit shift register (SCTSR) and starts transmitting. Confirm that the TDFE flag in the serial status register (SCSSR) is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is (64 – 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 the FIFO control register (SCFCR), the TDFE flag is set. If the TIE bit in the serial control register (SCSR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt request is generated. When the number of transmit data matches the data set in the transmit data stop register (SCTDSR) while the transmit data stop function is used, the transmit operation is stopped and the TSF flag in the serial status register (SCSSR) is set. When the TSIE bit in the serial control register (SCSCR) 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 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.

   If there is no transmit data, the TEND flag in the serial status register (SCSSR) 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 18.4 shows an example of the operation for transmission in asynchronous mode.

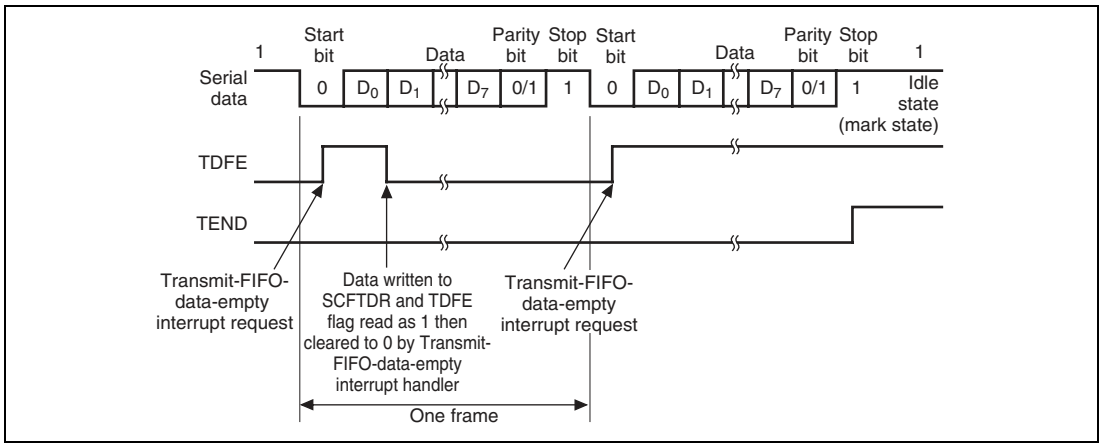


Figure 18.4 Example of Transmit Operation (Example with 8-Bit Data, Parity, One Stop Bit)

- Transmit data stop function

  When the value of the SCTDSR register and the number of transmit data match, transmit operation stops. Setting the TSIE bit (interrupt enable bit) allows the generation of an interrupt and activation of DMAC.

  Figure 18.5 shows an example of the operation for transmit data stop function.

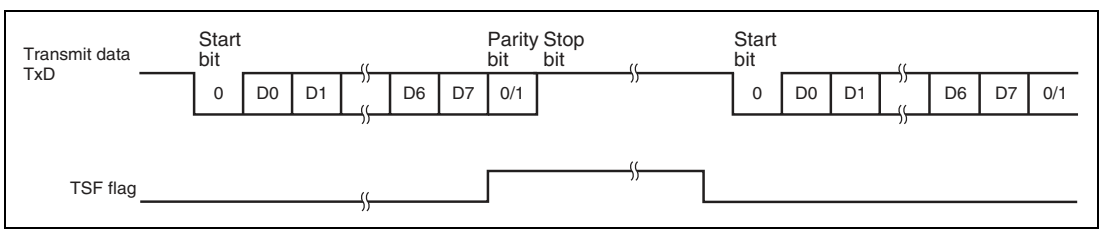
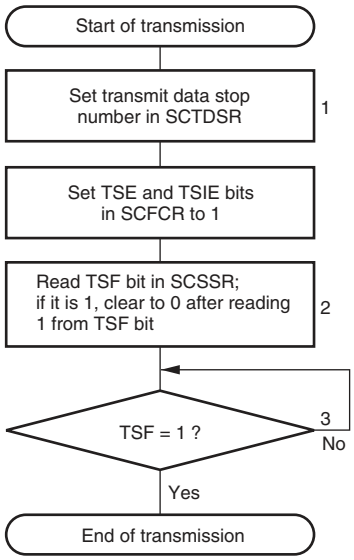


Figure 18.5 Example of Transmit Data Stop Function

Figure 18.6 shows the transmit data stop function flowchart.



1. Set the transmit data stop number in SCTDSR, then set the TSE bit in SCFCR to 1.When an interrupt is enabled, also set the TSIE bit to 1.
2. If the TSF bit in SCSSR is set to 1, clear it to 0 after reading 1. When transmit data is written to SCFTDR in this state, transmit operation is started.
3. If the TSF bit is set to 1 (transmit data stop number is matched with transmit data number), transmit operation is stopped. If the TSIE bit is set to 1, an interrupt is generated.

Serial transmission continuation procedure:
Set the TCRST bit in SCFCR to 1, clear transmit count, and clear the TCRST bit to 0. Then follow steps 1, 2, and 3.

Figure 18.6 Transmit Data Stop Function Flowchart

(5) Transmitting and Receiving Data (Serial data reception)

Figures 18.7 and 18.8 show sample serial reception flowcharts. After SCIF reception is enabled, use the following procedure to perform serial data reception.

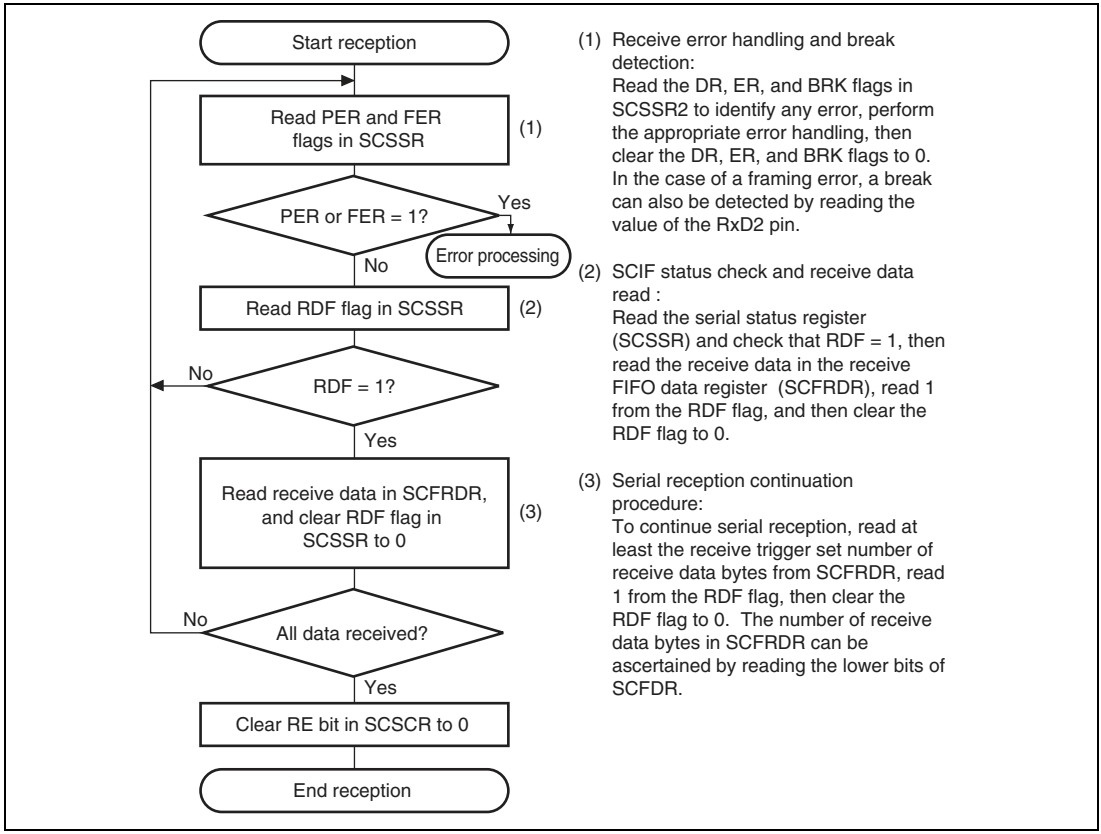
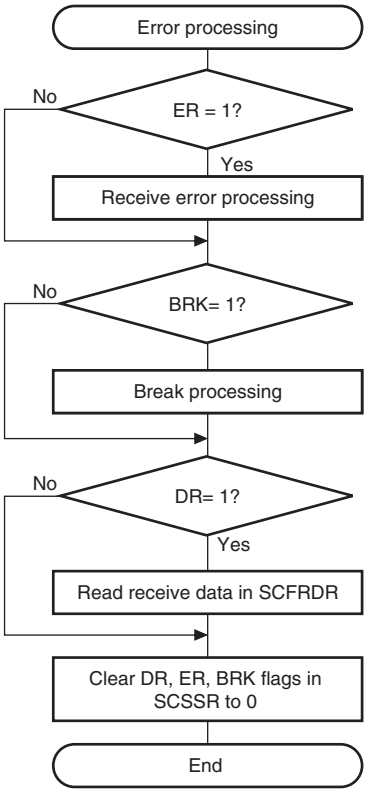


Figure 18.7 Sample Serial Reception Flowchart (1)



1. Whether a framing error or parity error has occurred in the receive data read from SCFRDR can be ascertained from the FER and PER bits in SCSSR.
2. 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 18.8 Sample Serial Reception Flowchart (2)

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. 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 the receive error (framing error/parity error) is generated, receive operation is continued.

4. If the RIE bit in SCSCR 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 SCSCR is set to 1 when the ER flag changes to 1, a receive-error interrupt request is generated.
   If the BRIE bit in SCSCR is set to 1 when the BRK flag changes to 1, a break reception interrupt request is generated.
   If the DRIE bit in SCSCR 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 18.9 shows an example of the operation for reception.

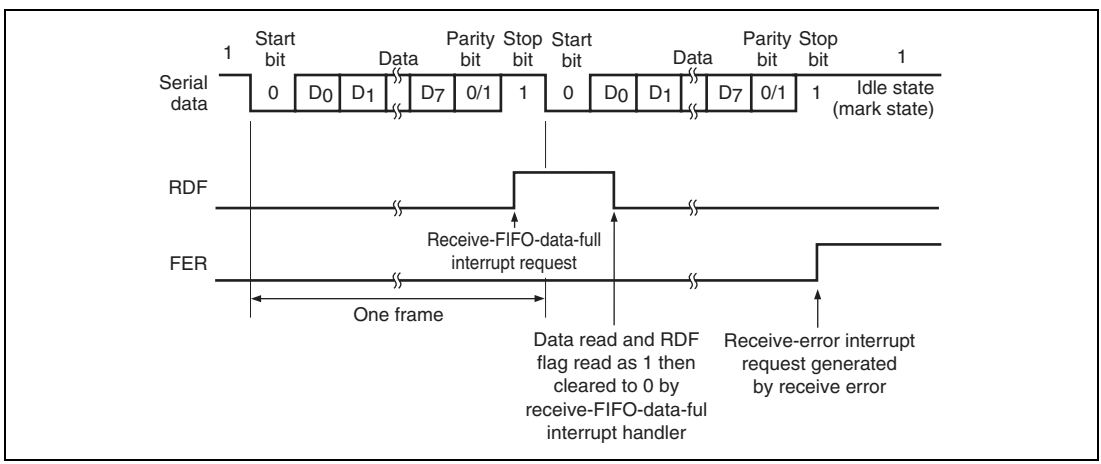


Figure 18.9 Example of SCIF Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit)

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 18.10 shows an example of the operation when modem control is used.

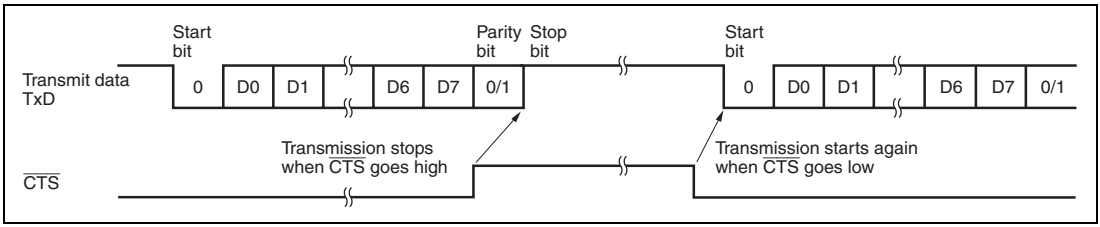


Figure 18.10 Example of $\overline{\text{CTS}}$ Control Operation

When modem control is enabled, the $\overline{\text{RTS}}$ signal goes high after the number of receive FIFO (SCFRDR) has exceeded the number of $\overline{\text{RTS}}$ output triggers.

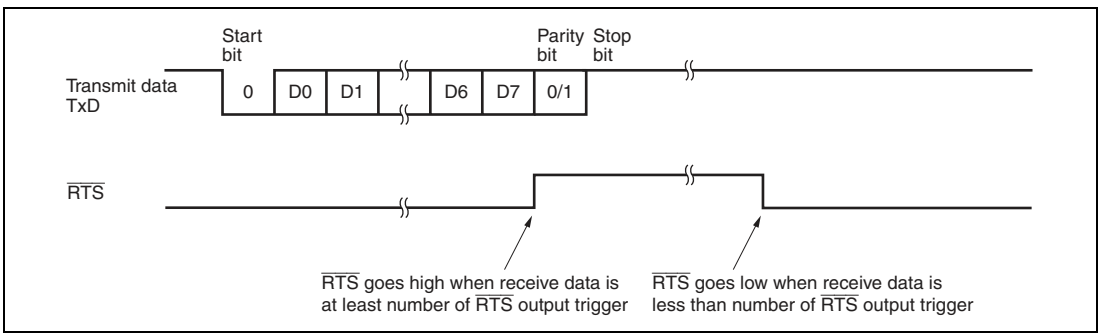


Figure 18.11 Example of $\overline{\text{RTS}}$ Control Operation

18.4.3 Synchronous Mode

Operation in synchronous mode is described below.

The SCIF 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 the serial mode register (SCSMR). The SCIF clock source is determined by the CKE1 and CKE0 bits in the serial control register (SCSCR).

- 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 SCIF clock source

  When the internal clock is selected:
  The SCIF operates on the baud rate generator clock and outputs a serial clock from SCK pin.

When the external clock is selected:
The SCIF operates on the external clock input through the SCK pin.

18.4.4 Serial Operation in Synchronous Mode

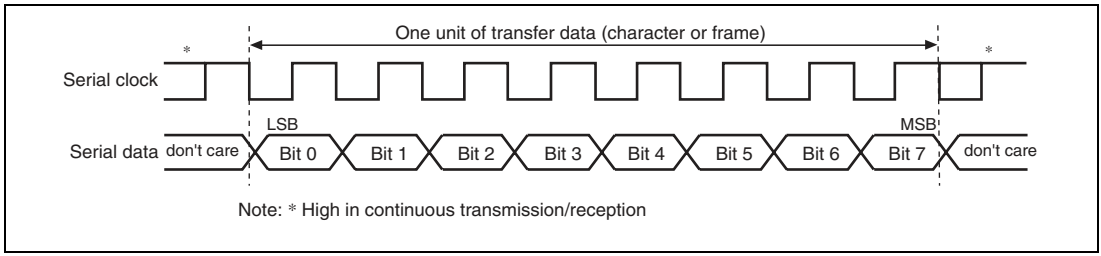


Figure 18.12 Data Format in Synchronous Communication

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 SCIF 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 SCIF, according to the setting of the CKE1 and CKE0 bits in SCSCR.

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 the transmit FIFO data register (SCFTDR) 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 (SCIF Initialization)

Before transmitting and receiving data, it is necessary to clear the TE and RE bits in SCSCR to 0, then initialize the SCIF 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, the transmit shift register (SCTSR) is initialized. Note that clearing the TE and RE bits to 0 does not change the contents of SCSSR, SCFTDR, or SCFRDR. The TE bit should be cleared to 0 after all transmit data has been sent and the TEND bit in SCSSR 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 SCFCR should first be set to 1 to reset SCFTDR.

Figure 18.13 shows sample SCIF initialization flowcharts.

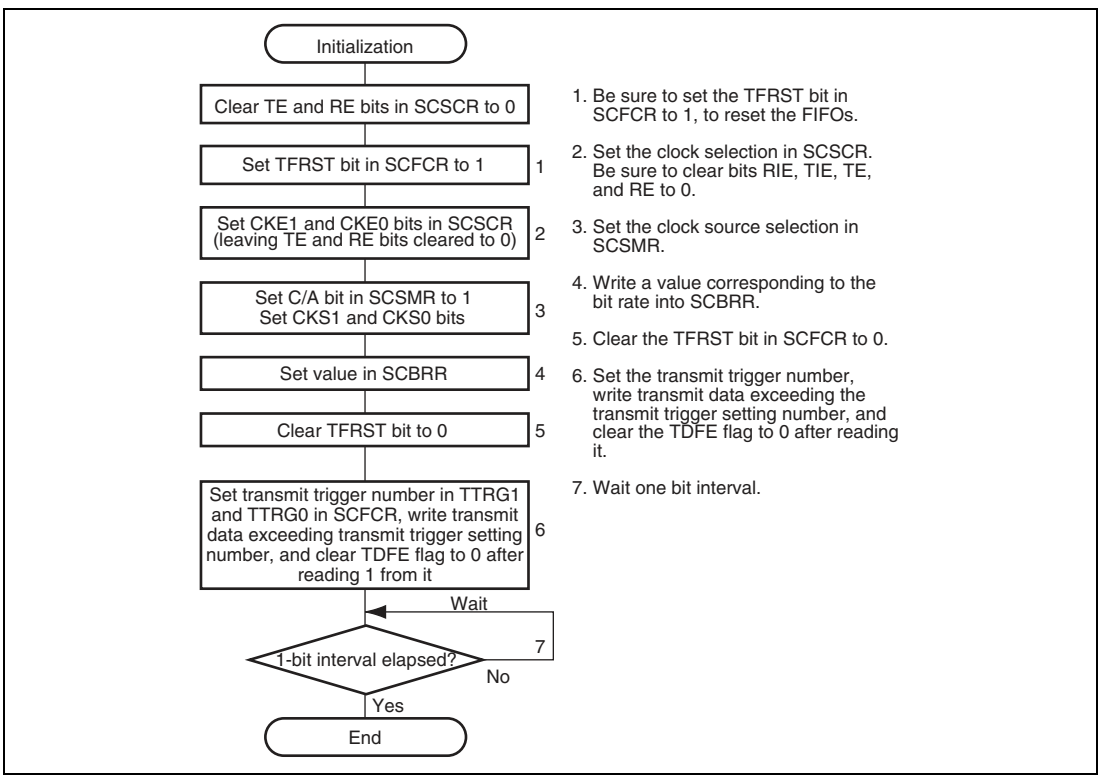


Figure 18.13 Sample SCIF Initialization Flowchart (1) (Transmission)

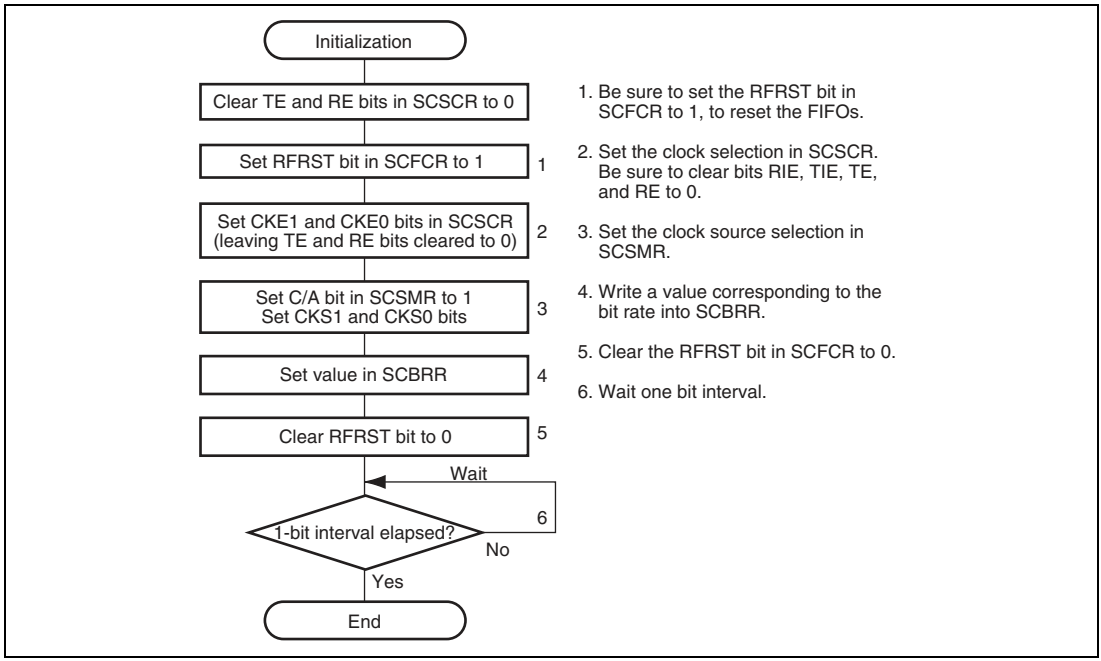


Figure 18.13 Sample SCIF Initialization Flowchart (2) (Reception)

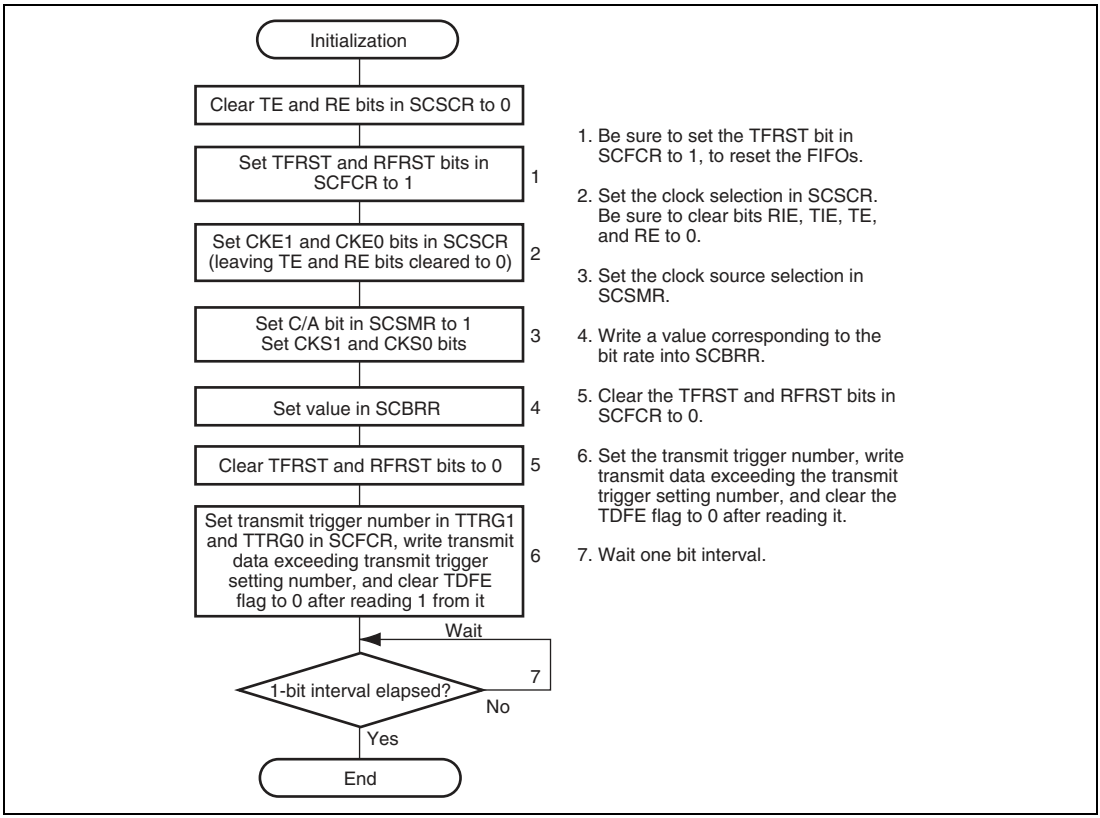


Figure 18.13 Sample SCIF Initialization Flowchart (3)
(Simultaneous Transmission and Reception)

(4) Data Transfer Operations (Serial Data Transmission)

Figure 18.14 shows sample flowcharts for serial transmission.

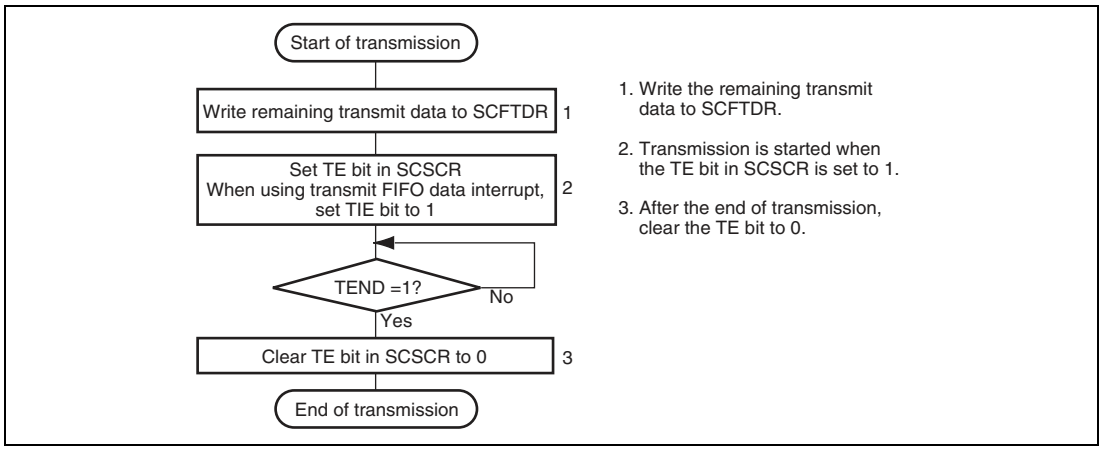


Figure 18.14 Sample Serial Transmission Flowchart (1)
(First Transmission after Initialization)

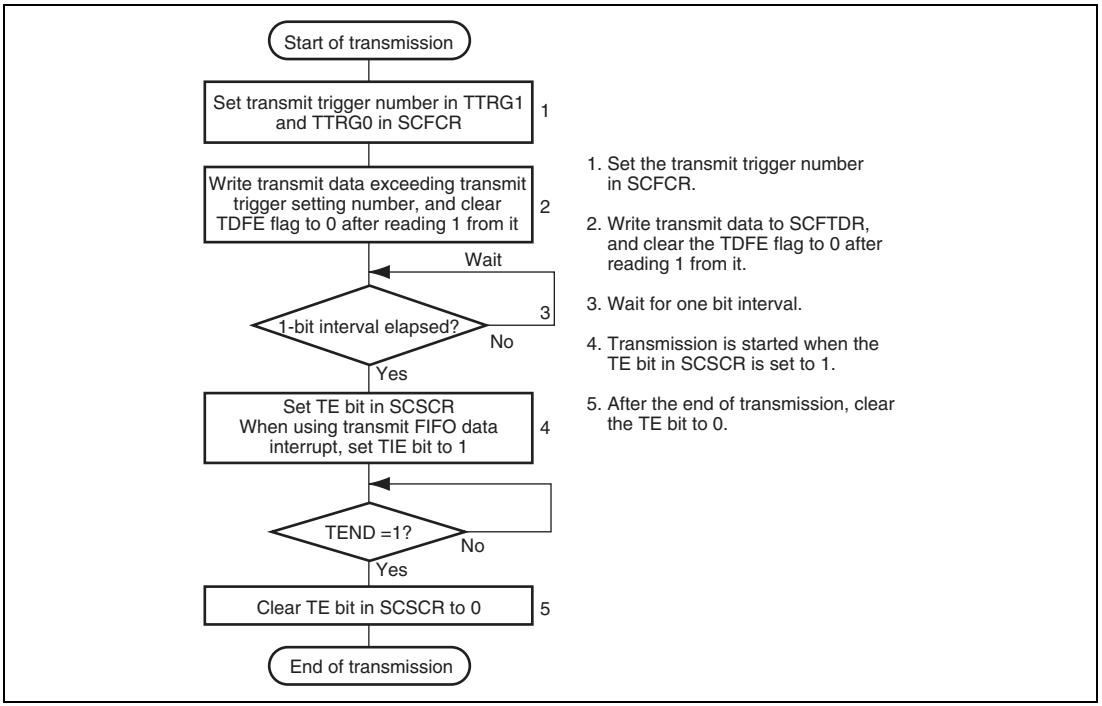


Figure 18.14 Sample Serial Transmission Flowchart (2)
(Second and Subsequent Transmission)

(5) Data Transfer Operations (Serial Data Reception)

Figure 18.15 shows sample flowcharts for serial reception.

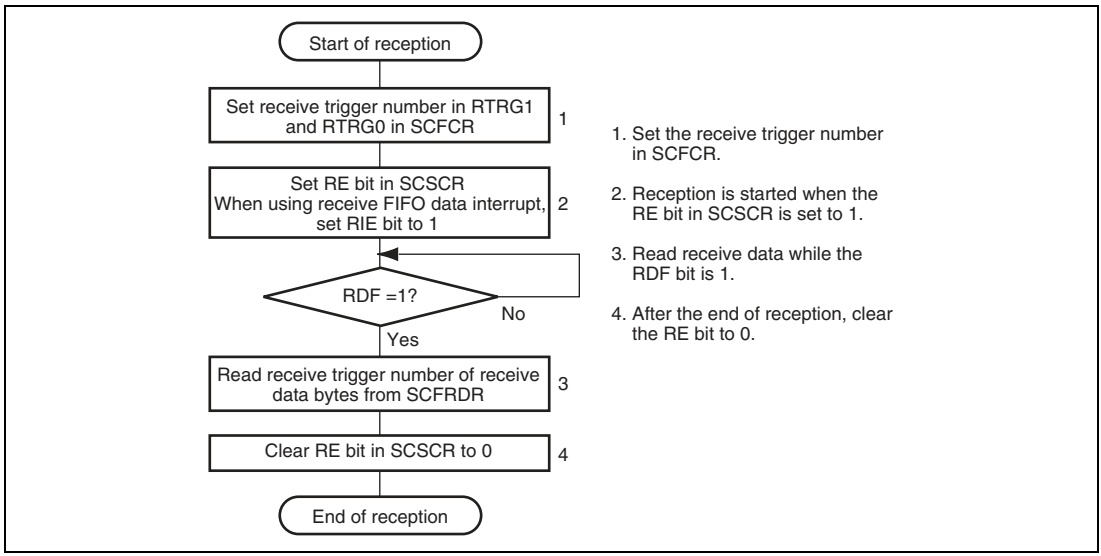


Figure 18.15 Sample Serial Reception Flowchart (1)
(First Reception after Initialization)

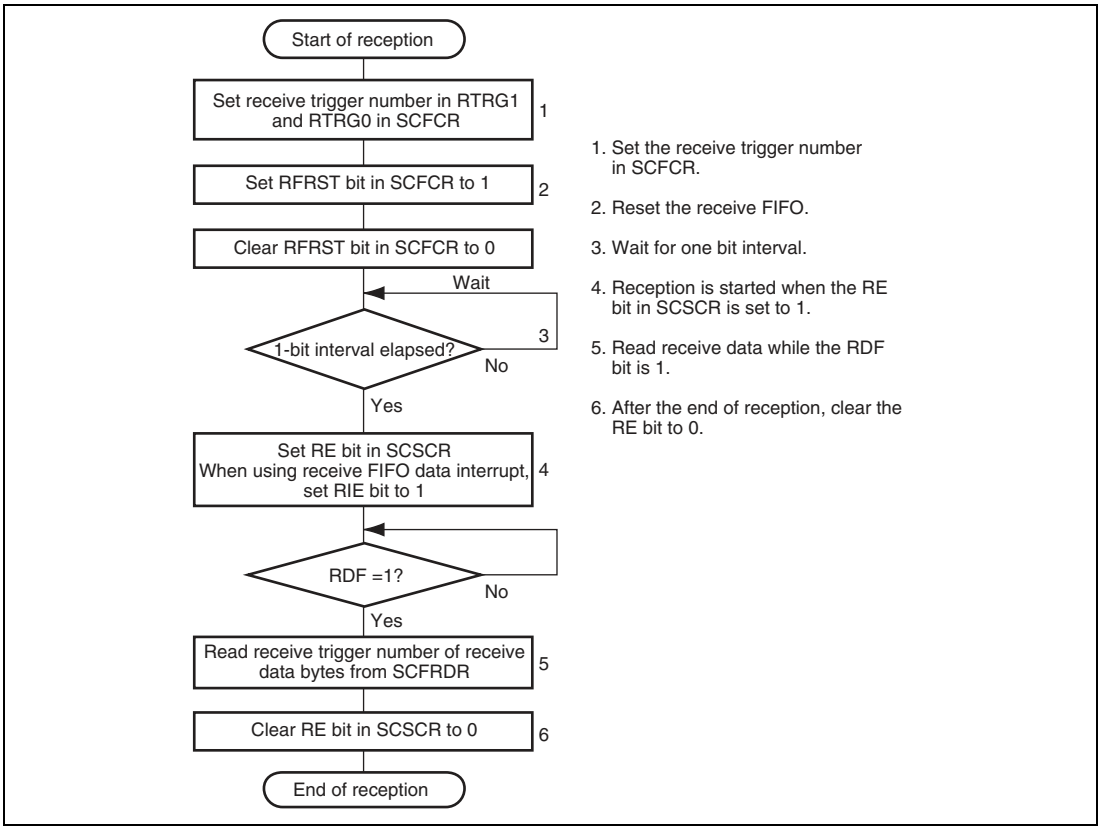


Figure 18.15 Sample Serial Reception Flowchart (2)
(Second and Subsequent Reception)

(6) Data Transfer Operations (Simultaneous Serial Data Transmission and Reception)

Figure 18.16 shows sample flowcharts for simultaneous serial transmission and reception.

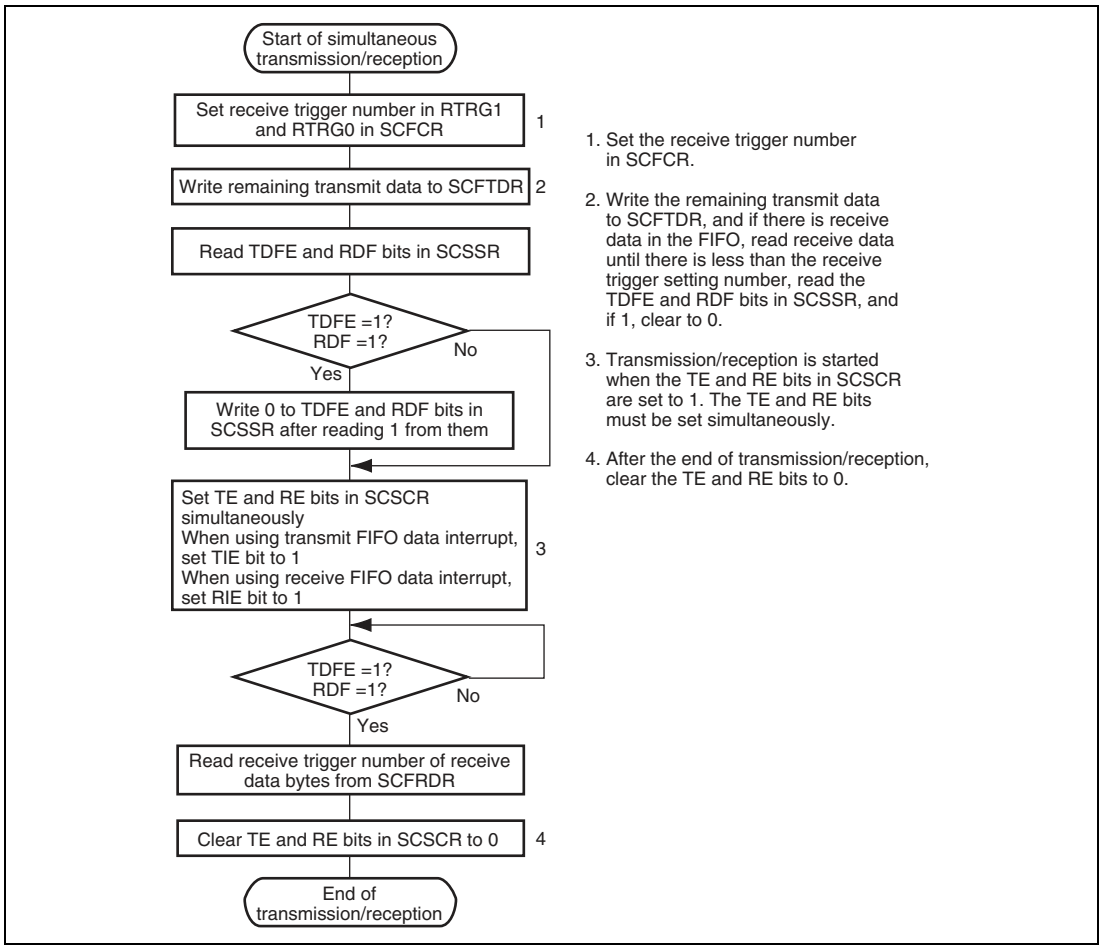


Figure 18.16 Sample Simultaneous Serial Transmission and Reception Flowchart (1) (First Transfer after Initialization)

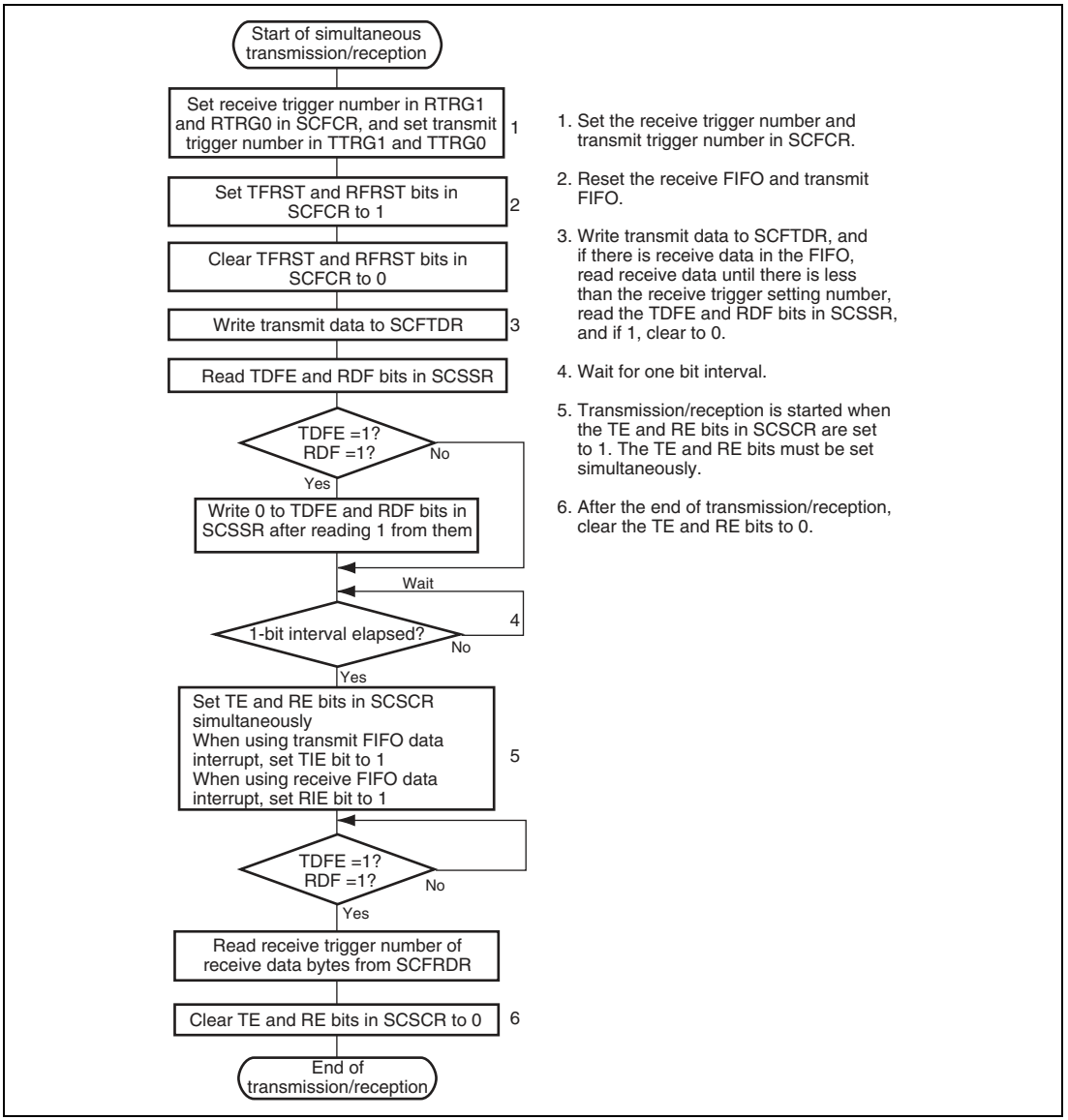


Figure 18.16 Sample Simultaneous Serial Transmission and Reception Flowchart (2) (Second and Subsequent Transfer)

18.5 Interrupt Sources and DMAC

In asynchronous mode, the SCIF 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 SCIF supports two interrupts: transmit-FIFO-data-empty and receive-FIFO-data-full.

Table 18.4 shows the interrupt sources. The interrupt sources are enabled or disabled by means of the TIE, RIE, ERIE, BRIE, DRIE, and TSIE bits in SCSCR.

When the TDFE flag in SCSSR is set to 1, the transmit-FIFO-data-empty interrupt request is generated. When the TSF flag in SCSSR 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 and data stop interrupt requests. The DMAC transfer request is automatically cleared when the number of data written to SCFTDR by the DMAC is increased more than that of setting transmit triggers.

When the RDF flag in SCSSR 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 SCFRDR by the DMAC until the number of receive data in SCFRDR is decreased less than that of receive triggers.

When executing the data transmission and reception, set the DMAC, and then set SCIF after entered in the enabled state. The completion of the DMA transfer is the completion of transmission and reception. For the DMAC setting procedure, see section 10, Direct Memory Access Controller (DMAC).

An interrupt request is generated when the ER flag in SCSSR is set to1; the BRK flag in SCSSR is set to 1; the DR flag in SCSSR is set to 1; or the TSF flag in SCSSR 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. When activating the DMAC, carry out the following procedure.

- Set the interrupt enable bits (TIE, RIE) that correspond to the interrupt sources used for activation of the DMAC. Clear the other interrupt enable bits (TSIE, ERIE, BRIE, and DRIE) to 0.

Table 18.4 SCIF Interrupt Sources

Interrupt Source	DMAC Activation
Interrupt initiated by receive error (ER) or break (BRK)	Not possible
Interrupt initiated by receive FIFO data full flag (RDF) or data ready flag (DR)	Possible*[1]
Interrupt initiated by receive FIFO data empty flag (TDFE) or transmit data stop flag (TSF)	Possible*[2]

Notes: 1. DMAC can be activated only by the receive-FIFO-data-full interrupt request.
2. DMAC can be activated only by the transmit-FIFO-data-empty interrupt request.

See section 7, Exception Handling, for priorities and the relationship with non-SCIF interrupts.

18.6 Usage Notes

(1) SCFTDR Writing and the TDFE Flag

The TDFE flag in the serial status register (SCSSR) is set when the number of transmit data bytes written in the transmit FIFO data register (SCFTDR) has fallen below the transmit trigger number set by bits TTRG1 and TTRG0 in the FIFO control register (SCFCR). After TDFE 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 less than or equal to the transmit 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 transmit trigger number has been written to SCFTDR.
The number of transmit data bytes in SCFTDR can be found in the bits 14 to 8 of the FIFO data count set register (SCFDR).

(2) SCFRDR Reading and the RDF Flag

The RDF flag in the serial status register (SCSSR) is set when the number of receive data bytes in the receive FIFO data register (SCFRDR) has become equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in the FIFO control register (SCFCR). After RDF 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 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 SCFRDR can be found in the bits 6 to 0 of the FIFO data count set register (SCFDR).

(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 SCFRDR is halted in the break state, the SCIF receiver continues to operate.

(4) Receive Data Sampling Timing and Receive Margin

An example with a sampling rate 1/16 is given. The SCIF operates on a base clock with a frequency of 8 times the transfer 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. The timing is shown in figure 18.17.

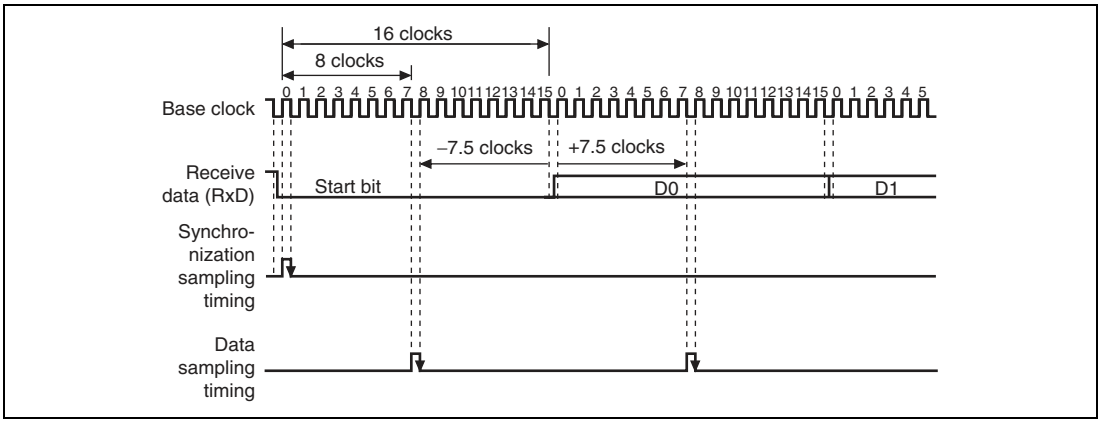


Figure 18.17 Receive Data Sampling Timing in Asynchronous Mode

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:

$$M = (0.5 - 1/(2 \times 16)) \times 100\%$$
$$= 46.875\% \quad \cdots\cdots (2)$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

Section 19 Infrared Data Association Module (IrDA)

This LSI has an on-chip Infrared Data Association (IrDA) interface that is based on the IrDA 1.0 system and can perform infrared communication.

The IrDA is an optional module used for modulation and demodulation of signals for the SCIF_0 module, and it must always be used together with the SCIF_0 module.

19.1 Features

- Conforms to the IrDA 1.0 system
- Asynchronous serial communication
  - Data length: 8 bits
  - Stop bit length: 1 bit
  - Parity bit: None
- On-chip 64-stage FIFO buffers for both transmit and receive operations
- On-chip baud rate generator with selectable bit rates
- Guard functions to protect the receiver during transmission
- Clock supply halted to reduce power consumption when not using the IrDA interface

Figure 19.1 shows a block diagram of the IrDA.

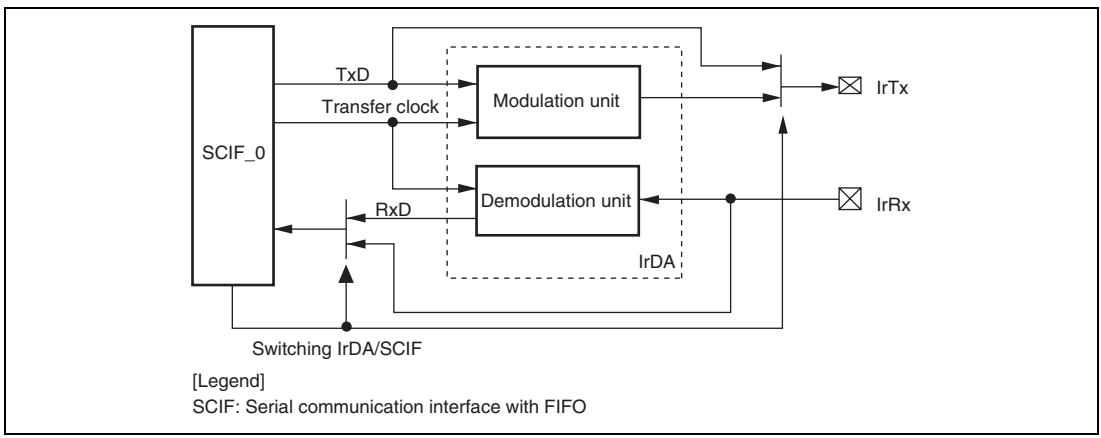


Figure 19.1 Block Diagram of IrDA

19.2 Input/Output Pins

Table 19.1 shows the IrDA pin configuration.

Table 19.1 Pin Configuration

Name	Pin Name	I/O	Function
IrDA receive data	IrRX	Input	Receive data input
IrDA transmit data	IrTX	Output	Transmit data output

Note: Clock input from the serial clock pin cannot be set in IrDA mode.

19.3 Register Description

The IrDA has the following internal registers. Refer to section 37, List of Registers, for more details on the addresses and states of these registers in each operating mode.

- IrDA mode register (SCIMR)

19.3.1 IrDA Mode Register (SCIMR)

SCIMR selects IrDA or SCIF mode and selects the IrDA output pulse width.

IrDA operates when the IRMOD bit is set to 1. When the IRMOD bit is cleared to 0, IrDA can operate as an SCIF.

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	IRMOD	0	R/W	IrDA Mode Selects whether this module operates as an IrDA serial communication interface or as an SCIF. 0: Operates as an SCIF 1: Operates as an IrDA

Bit	Bit Name	Initial Value	R/W	Description
6 to 3	ICK3 to ICK0	All 0	R/W	Output Pulse Division Ratio 3 to 0*
				Specifies the ratio for dividing the peripheral clock (Pϕ) to generate the IRCLK clock pulse to be used for IrDA. IRCLK is obtained as follows:
				IRCLK = 1/(2N + 2) × Pϕ
				N = Value set by ICK3 to ICK0
2	PSEL	0	R/W	Output Pulse Width Select
				Selects an IrDA output pulse width that is 3/16 of the bit length for 115 kbps or 3/16 of the bit length for the selected baud rate.
				0: Pulse width is 3/16 of the bit length
				1: Pulse width is 3/16 of 115 kbps bit length for the baud rate selected by ICK3 to ICK0
1, 0	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0.

Note: * Recommended value of IrDA

For example, when the transfer rate in 115.2kbps, set the value (Nb+1):(Ni+1) = 2:1. Setting (Nb+1):(Ni+1) = 2:1 (115.2 kbps) allows operation in synchronization while the SCIF module and IrDA module perform asynchronous operation. Synchronous operation equalizes errors in the IR frame when such bit rate errors occur. Use bit 2 (PSEL) in SCIMR as PSEL = 1 to adjust transfer and receive data.

[Legend] Nb: The baud rate value in SCIF (SCBRD7 to SCBRD0 in SCBRR)

Ni : The baud rate value in IrDA (ICK3 to ICK0 in SCIMR)

B : Bit rate (bits/s)

Setting Example as Pϕ =33.1776MHz:

Nb	Nb+1	B	Ni	Ni+1
17	18	115.2	8	9
35	36	57.6	8	9
53	54	38.4	8	9
107	108	19.2	8	9
215	2196	9.6	8	9

19.4 Operation

The IrDA module can perform infrared communication conforming to IrDA 1.0 by connecting infrared transmit/receive units. The serial communication interface unit includes a buffer in the transmit unit and the receive unit, allowing CPU overhead to be reduced and continuous high-speed communication to be performed.

The IrDA module modifies IrTx/IrRx transmit/receive data waveforms to satisfy the IrDA 1.0 specification for infrared communication.

In the IrDA 1.0 specification, communication is first performed at a speed of 9600 bps, and the communication speed is changed. However, the communication rate cannot be automatically changed in this module, so the communication speed should be confirmed, and the appropriate speed set for this module by software.

19.4.1 Transmitting

The waveforms of a serial output signal (UART frame) from the SCIF are modified and the signal is converted into the IR frame serial output signal by the IrDA module, as shown in figure 19.2.

When serial data is 0, a pulse of 3/16 the IR frame bit width is generated and output. When serial data is 1, no pulse is output.

19.4.2 Receiving

Received 3/16 IR frame bit-width pulses are demodulated and converted to a UART frame, as shown in figure 19.2.

Demodulation to 0 is performed for pulse output, and demodulation to 1 is performed for no pulse output.

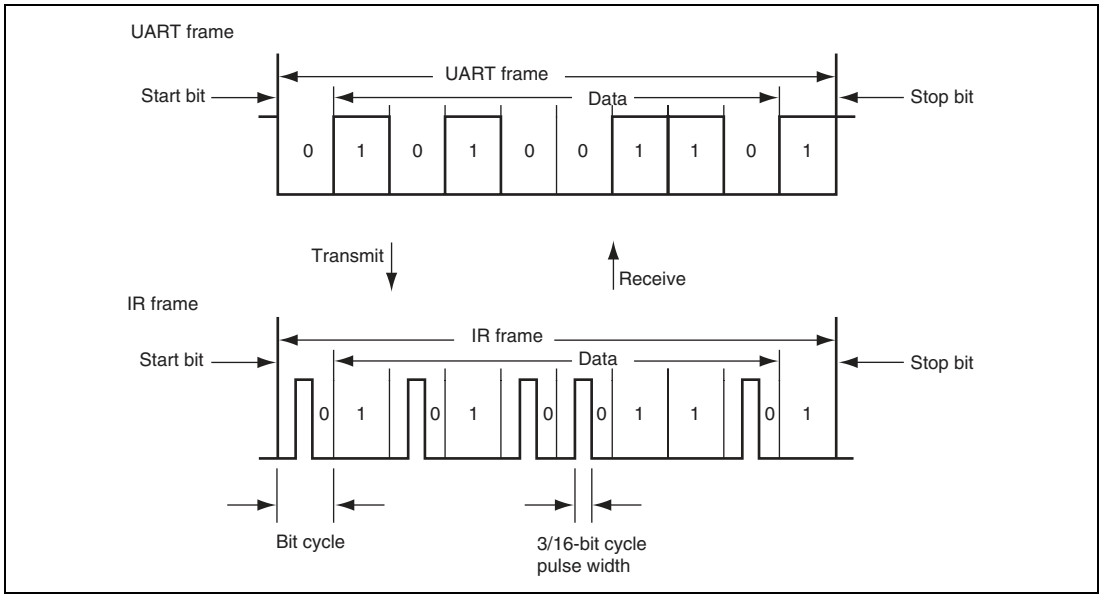


Figure 19.2 Transmit/Receive Operation

19.4.3 Data Format Specification

The data format of UART frames used for IrDA communication must be specified by the SCIF_0 registers. The UART frame has eight data bits, no parity bit, and one stop bit.

IrDA communication is performed in asynchronous mode, and this mode must also be specified by the SCIF_0 registers. The sampling rate must be set to 1/16.

When using IrDA, set the SCIF_0 operating clock by setting the CKE1 and CKE0 bits in the serial control register to 01.

The IrDA communication rate is the same as the SCIF_0 bit rate, which is specified by the SCIF_0 registers.

For details on SCIF_0 registers, refer to section 18, Serial Communication Interface with FIFO (SCIF).

Section 20 I²C Bus Interface (IIC)

The I²C bus interface supports and provides a subset of the Philips I²C bus (inter-IC bus) interface functions. The register configuration that controls the I²C bus differs partly from the Philips configuration, however.

Figure 20.1 shows a block diagram of the I²C bus interface.

Figure 20.2 shows an example of I/O pin connections to external circuits.

20.1 Features

- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independent from each other, the continuous transmission/reception can be performed.
- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization/wait function

  In master mode, the state of 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.

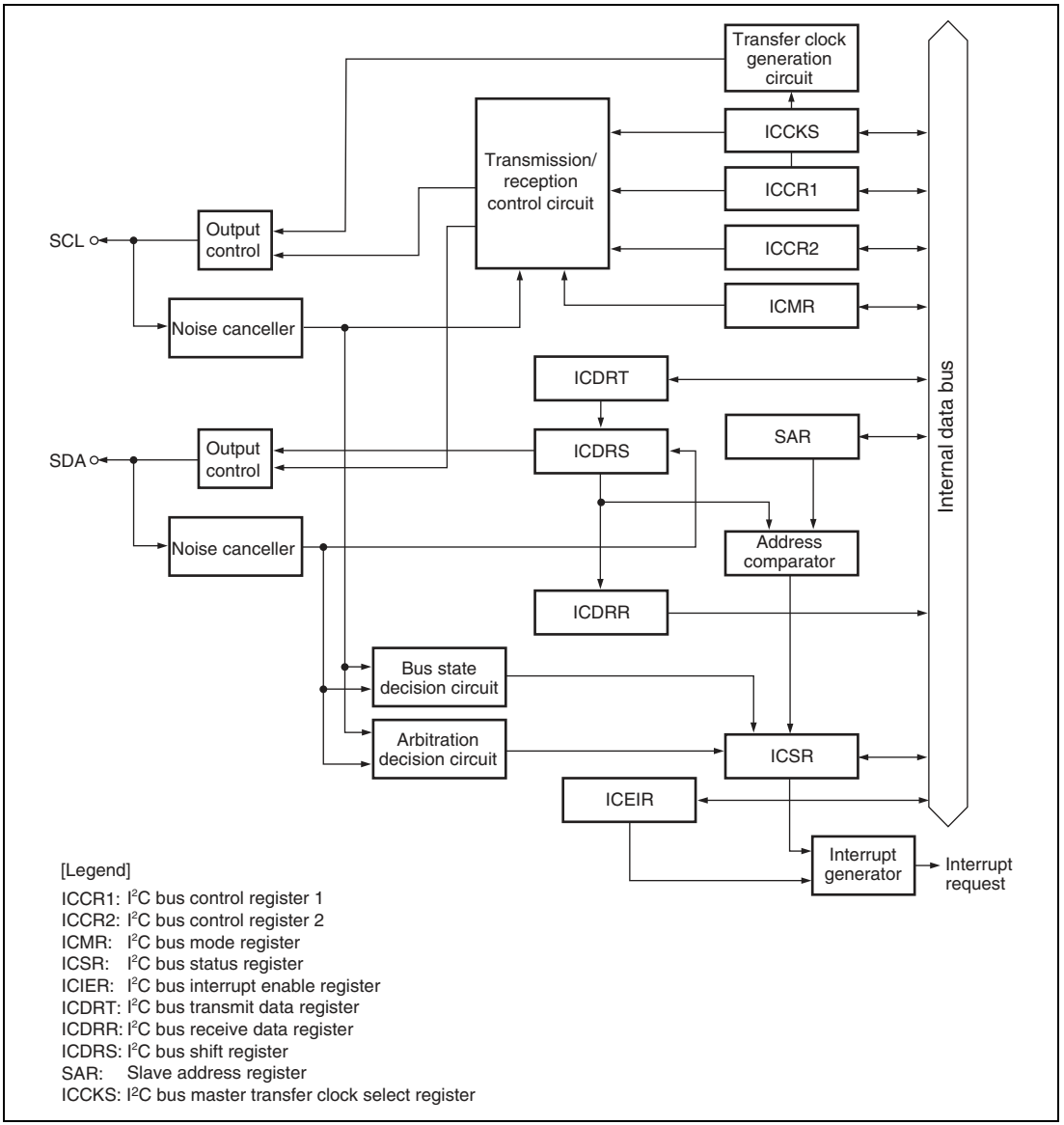


Figure 20.1 Block Diagram of I²C Bus Interface

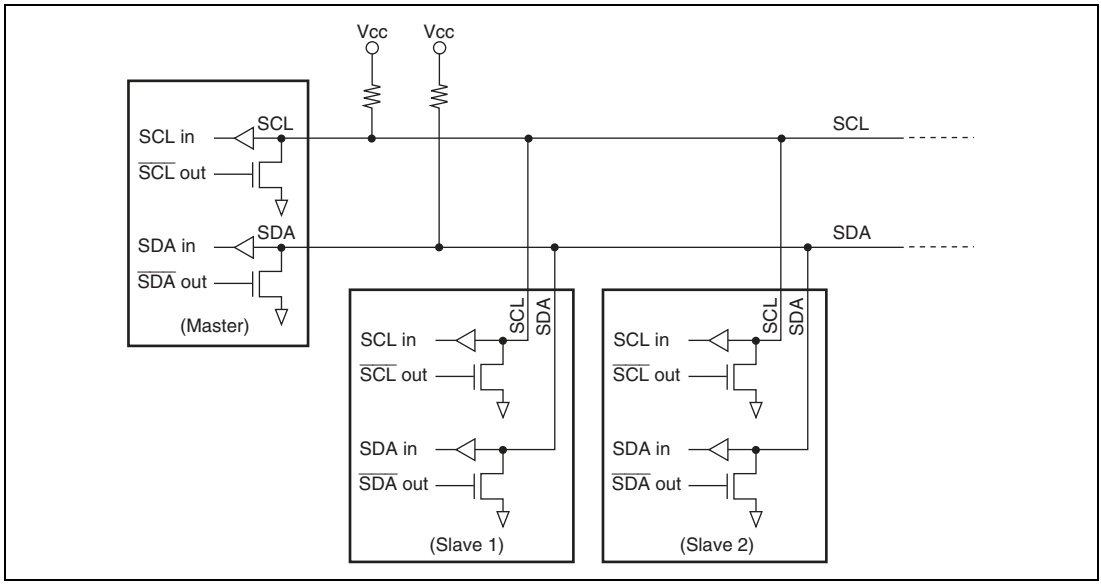


Figure 20.2 External Circuit Connections of I/O Pins

20.2 Input/Output Pins

Table 20.1 summarizes the input/output pins used by the I²C bus interface.

Table 20.1 I²C Bus Interface Pins

Name	Pin Name	Abbreviation	I/O	Function
IIC clock	IIC_SCL	SCL	I/O	IIC serial clock input/output
IIC data I/O	IIC_SDA	SDA	I/O	IIC serial data input/output

20.3 Register Descriptions

The I²C bus interface has the following registers:

- I²C bus control register 1 (ICCR1)
- I²C bus control register 2 (ICCR2)
- I²C bus mode register (ICMR)
- I²C bus interrupt enable register (ICIER)
- I²C bus status register (ICSR)
- Slave address register (SAR)
- I²C bus transmit data register (ICDRT)
- I²C bus receive data register (ICDRR)
- I²C bus shift register (ICDRS)
- I²C bus master transfer clock select register (ICCKS)

20.3.1 I²C Bus Control Register 1 (ICCR1)

ICCR1 enables or disables the I²C bus interface, controls transmission or reception, and selects master or slave mode, transmission or reception, and transfer clock frequency in master mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface Enable 0: This module is halted. 1: This bit is enabled for transfer operations.
6	RCVD	0	R/W	Reception Disable This bit 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. After data receive has been started in slave receive mode, when the first seven bits of the receive data agree with the slave address that is set to SAR and the eighth bit is 1, TRS is automatically set to 1. Operating modes are described below according to MST and TRS combination. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

20.3.2 I²C Bus Control Register 2 (ICCR2)

ICCR1 issues start/stop conditions, manipulates the SDA pin, monitors the SCL pin, and controls reset in the control part of the I²C bus interface.

Bit	Bit Name	Initial Value	R/W	Description
7	BBSY	0	R/W	Bus Busy This bit 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. To issue start/stop conditions, use the MOV instruction.
6	SCP	1	W	Start/Stop Issue Condition Disable The SCP bit 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. If 1 is written, the data is not stored.
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).

Bit	Bit Name	Initial Value	R/W	Description
4	SDAOP	1	R/W	SDAO Write Protect This bit 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 by the MOV instruction. This bit is always read as 1.
3	SCLO	1	R	This bit monitors SCL output level. When SCLO is 1, SCL pin outputs high. When SCLO is 0, SCL pin outputs low.
2	—	1	—	Reserved This bit is always read as 1.
1	IICRST	0	R/W	IIC Control Part Reset This bit resets the control part except for I²C registers. If this bit is set to 1 when hang-up occurs because of communication failure during I²C operation, I²C control part can be reset without setting ports and initializing registers.
0	—	1	—	Reserved This bit is always read as 1.

20.3.3 I²C Bus Mode Register (ICMR)

ICMR selects whether the MSB or LSB is transferred first, performs master mode wait control, and selects the transfer bit count.

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	—	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	—	All 1	—	Reserved These bits are always read as 1.
3	BCWP	1	R/W	BC Write Protect This bit controls the BC2 to BC0 modifications. When modifying BC2 to BC0, this bit should be cleared to 0 and use the MOV instruction. 0: When writing, values of BC2 to BC0 are set. 1: When reading, 1 is always read. When writing, settings of BC2 to BC0 are invalid.
2	BC2	0	R/W	Bit Counter 2 to 0 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. Bit BC2 to BC0 settings should be made during an interval between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL pin is low. The value returns to 000 at the end of a data transfer, including the acknowledge bit. I²C Bus Format 000: 9 bits 001: 2 bits 010: 3 bits 011: 4 bits 100: 5 bits 101: 6 bits 110: 7 bits 111: 8 bits
1	BC1	0	R/W	
0	BC0	0	R/W	

20.3.4 I²C Bus Interrupt Enable Register (ICIER)

ICIER enables or disables interrupt sources and acknowledge bits, sets acknowledge bits to be transferred, and confirms acknowledge bits to be received.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, this bit enables or disables the transmit data empty interrupt (TXI). 0: Transmit data empty interrupt request (TXI) is disabled. 1: Transmit data empty interrupt request (TXI) is enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable This bit 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 This bit enables or disables the receive data full interrupt request (RXI) when a 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.
4	NAKIE	0	R/W	NACK Receive Interrupt Enable This bit enables or disables the NACK receive interrupt request (NAKI) when the NACKF and AL bits in ICSR are set to 1. NAKI can be canceled by clearing the NACKF, OVE, or NAKIE bit to 0. 0: NACK receive interrupt request (NAKI) is disabled. 1: NACK receive interrupt request (NAKI) is enabled.

Bit	Bit Name	Initial Value	R/W	Description
3	STIE	0	R/W	Stop Condition Detection Interrupt Enable 0: Stop condition detection interrupt request (STPI) is disabled. 1: Stop condition detection interrupt request (STPI) is enabled.
2	ACKE	0	R/W	Acknowledge Bit Judgement 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. 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 performs confirmation of interrupt request flags and status.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	0	R/W	Transmit Data Register Empty [Setting condition] • When data is transferred from ICDRT to ICDRS and ICDRT becomes empty • When TRS is set • When a start condition (including re-transfer) has been issued • When transmit mode is entered from receive mode in slave mode [Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1 • When data is written to ICDRT with an instruction
6	TEND	0	R/W	Transmit End [Setting conditions] • When the ninth clock of SCL rises with the I²C bus format while the TDRE flag is 1 [Clearing conditions] • When 0 is written in TEND after reading TEND = 1 • When data is written to ICDRT with an instruction
5	RDRF	0	R/W	Receive Data Register Full [Setting condition] • When a receive data is transferred from ICDRS to ICDRR [Clearing conditions] • When 0 is written in RDRF after reading RDRF = 1 • When ICDRR is read with an instruction

Bit	Bit Name	Initial Value	R/W	Description
4	NACKF	0	R/W	No Acknowledge Detection Flag [Setting condition] • When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1 [Clearing condition] • When 0 is written in NACKF after reading NACKF = 1
3	STOP	0	R/W	Stop Condition Detection Flag [Setting conditions] • In master mode: when a stop condition is detected after frame transfer is completed • In slave mode: when a stop condition is detected after the address set in SAR matches the salve address that comes as the first byte after the detection of a start condition [Clearing condition] • When 0 is written in STOP after reading STOP = 1
2	AL/OVE	0	R/W	Arbitration Lost Flag/Overrun Error Flag This 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 taken by another master. [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 [Clearing condition] • When 0 is written in AL/OVE after reading AL/OVE = 1

Bit	Bit Name	Initial Value	R/W	Description
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR. [Setting conditions] • When the slave address is detected in slave receive mode • When the general call address is detected in slave receive mode. [Clearing condition] • When 0 is written in AAS after reading AAS = 1
0	ADZ	0	R/W	General Call Address Recognition Flag This bit is valid in I²C bus format slave receive mode. [Setting condition] • When the general call address is detected in slave receive mode [Clearing conditions] • When 0 is written in ADZ after reading ADZ = 1

20.3.6 Slave Address Register (SAR)

SAR selects the communication format and sets the slave address. When the chip is in slave mode with the I²C bus format, if the upper 7 bits of SAR match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA6 to SVA0	All 0	R/W	Slave Address 6 to 0 These bits set a unique address in bits SVA6 to SVA0, differing form the addresses of other slave devices connected to the I²C bus.
0	—	0	R	Reserved These bits are 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 during transferring data of ICDRS, continuous transfer is possible. If the MLS bit of ICMR is set to 1 and when the data is written to ICDRT, the MSB/LSB inverted data is read. The initial value of ICDRT is H'FF.

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. The initial value of ICDRR is H'FF.

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.

20.3.10 I²C Bus Master Transfer Clock Select Register (ICCKS)

ICCKS is enabled in master mode and selects a transfer clock used in master mode. Specify ICCKS according to the required transfer rate. For transfer rate, see table 20.2.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 1. The write value should always be 0.
4	CKS4	0	R/W	Master Transfer Clock Select 4 to 0 Specify these bits according to the required transfer rate in master mode. In slave mode, these bits are used to ensure the data setup time in transmit mode.
3	CKS3	0	R/W	
2	CKS2	0	R/W	
1	CKS1	0	R/W	
0	CKS0	0	R/W	

Table 20.2 Transfer Rate

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		Transfer Rate					
CKS4	CKS3	CKS2	CKS1	CKS0	Peripheral Clock	Pϕ = 10 MHz	Pϕ = 16 MHz	Pϕ = 20 MHz	Pϕ = 25 MHz	Pϕ = 30 MHz	Pϕ = 32 MHz
0	0	0	0	0	Pϕ/28	357kHz	—	—	—	—	—
0	0	0	0	1	Pϕ/40	250kHz	400kHz	—	—	—	—
0	0	0	1	0	Pϕ/48	208kHz	333kHz	—	—	—	—
0	0	0	1	1	Pϕ/64	156kHz	250kHz	313kHz	391kHz	—	—
0	0	1	0	0	Pϕ/80	125kHz	200kHz	250kHz	313kHz	375kHz	400kHz
0	0	1	0	1	Pϕ/100	100kHz	160kHz	200kHz	250kHz	300kHz	320kHz
0	0	1	1	0	Pϕ/112	89kHz	143kHz	179kHz	223kHz	268kHz	286kHz
0	0	1	1	1	Pϕ/128	78kHz	125kHz	156kHz	195kHz	234 kHz	250kHz
0	1	0	0	0	Pϕ/56	179kHz	286kHz	357kHz	446kHz	536kHz	571kHz
0	1	0	0	1	Pϕ/80	125kHz	200kHz	250kHz	313kHz	375kHz	400kHz
0	1	0	1	0	Pϕ/96	104kHz	167kHz	208kHz	260kHz	313kHz	333kHz
0	1	0	1	1	Pϕ/128	78kHz	125kHz	156kHz	195kHz	234 kHz	250kHz
0	1	1	0	0	Pϕ/160	63kHz	100kHz	125kHz	156kHz	188kHz	200kHz
0	1	1	0	1	Pϕ/200	50kHz	80kHz	100kHz	125kHz	150kHz	160kHz
0	1	1	1	0	Pϕ/224	45kHz	71kHz	89kHz	112kHz	134 kHz	143kHz
0	1	1	1	1	Pϕ/256	39kHz	63kHz	78kHz	98kHz	117kHz	125kHz
1	0	0	0	0	Pϕ/112	89kHz	143kHz	179kHz	223kHz	268kHz	286kHz
1	0	0	0	1	Pϕ/160	63kHz	100kHz	125kHz	156kHz	188kHz	200kHz
1	0	0	1	0	Pϕ/192	52kHz	83kHz	104 kHz	130kHz	156kHz	167kHz
1	0	0	1	1	Pϕ/256	39kHz	63kHz	78kHz	98kHz	117kHz	125kHz
1	0	1	0	0	Pϕ/320	31kHz	50kHz	63kHz	78kHz	94kHz	100kHz
1	0	1	0	1	Pϕ/400	25kHz	40kHz	50kHz	63kHz	75kHz	80kHz
1	0	1	1	0	Pϕ/448	22kHz	36kHz	45kHz	56kHz	67kHz	71kHz
1	0	1	1	1	Pϕ/512	20kHz	31kHz	39kHz	49kHz	59kHz	63kHz
1	1	0	0	0	Pϕ/224	45kHz	71kHz	89kHz	112kHz	134kHz	143kHz
1	1	0	0	1	Pϕ/320	31kHz	50kHz	63kHz	78kHz	94kHz	100kHz
1	1	0	1	0	Pϕ/384	26kHz	42kHz	52kHz	65kHz	78kHz	83kHz
1	1	0	1	1	Pϕ/512	20kHz	31kHz	39kHz	49kHz	59kHz	63kHz
1	1	1	0	0	Pϕ/640	16kHz	25kHz	31kHz	39kHz	47kHz	50kHz

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		Transfer Rate					
CKS4	CKS3	CKS2	CKS1	CKS0	Peripheral Clock	Pϕ = 10 MHz	Pϕ = 16 MHz	Pϕ = 20 MHz	Pϕ = 25 MHz	Pϕ = 30 MHz	Pϕ = 32 MHz
1	1	1	0	1	Pϕ/800	13kHz	20kHz	25kHz	31kHz	38kHz	40kHz
1	1	1	1	0	Pϕ/896	11kHz	18kHz	22kHz	28kHz	33kHz	36kHz
1	1	1	1	1	Pϕ/1024	10kHz	16kHz	20kHz	24kHz	29kHz	31kHz

Note: In master mode, a transfer rate of 300 kHz or lower should be used.
In slave mode, a transfer rate of 400 kHz or lower should be used.

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 8 bits.

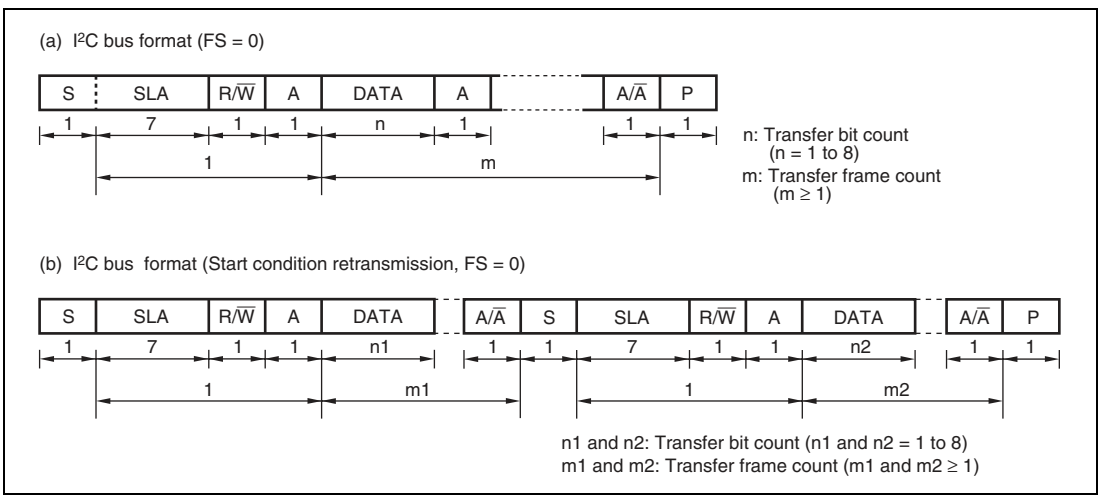


Figure 20.3 I²C Bus Formats

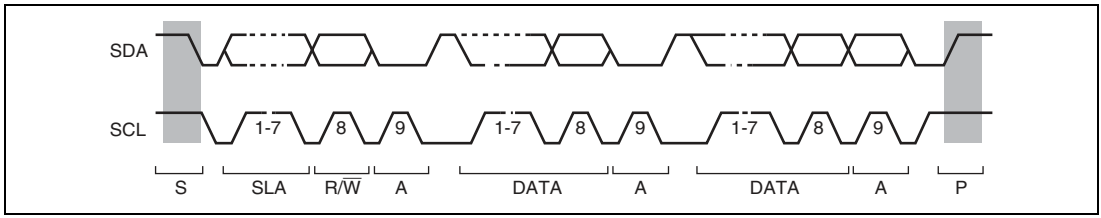


Figure 20.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

R/$\overline{W}$: Indicates the direction of data transfer: from the slave device to the master device when R/$\overline{W}$ is 1, or from the master device to the slave device when R/$\overline{W}$ is 0.

A: Acknowledge. The receive device drives SDA to low.

DATA: Transfer data

P: Stop condition. The master device drives SDA from low to high while SCL is high.

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. Set the MLS bit in ICMR and the CKS4 to CKS0 bits in ICCKS to 1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is free. Set the MST and TRS bits in ICCR1 to select master transmit mode. Then, write 1 to BBSY and 0 to SCP using MOV instruction. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first byte data show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cleared to 0, 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 using MOV instruction. SCL is fixed low until the transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.

6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the end of last byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) from the receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TEND or NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mode.

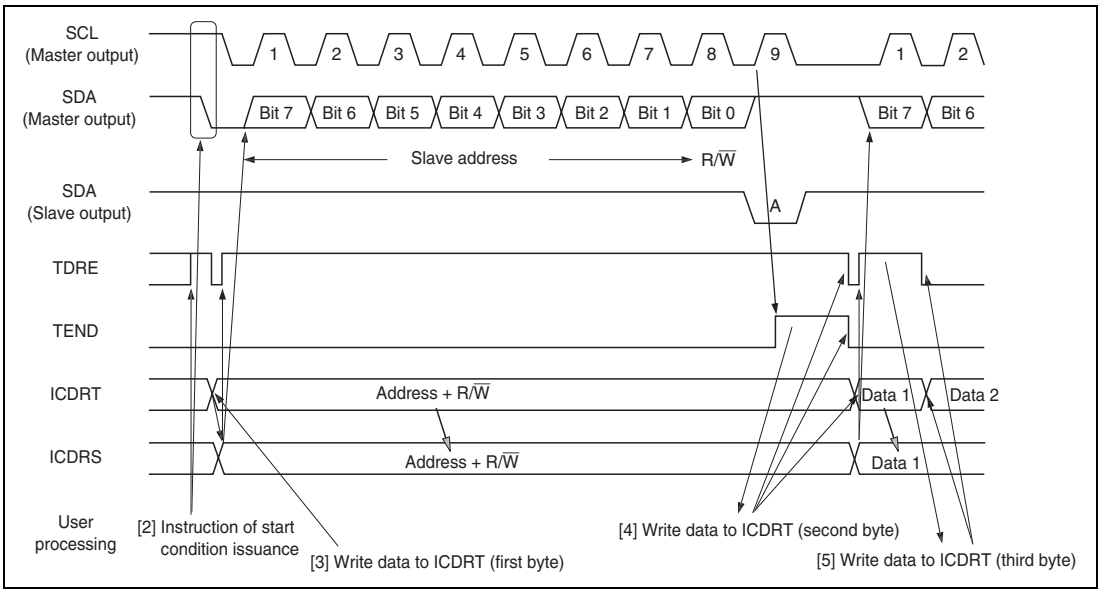


Figure 20.5 Master Transmit Mode Operation Timing (1)

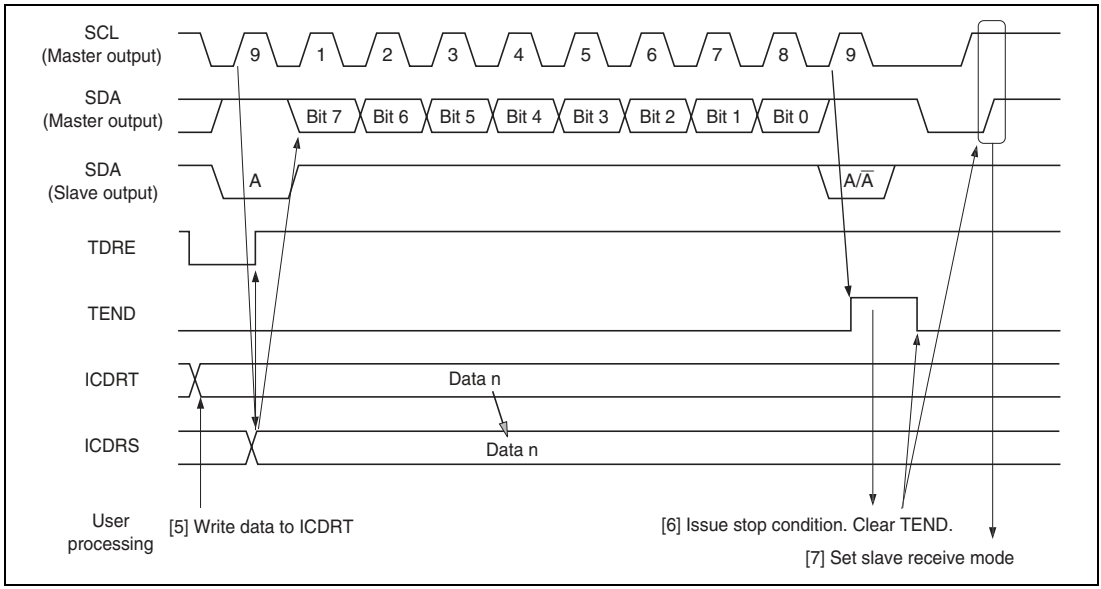


Figure 20.6 Master Transmit Mode Operation Timing (2)

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 ICST 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.

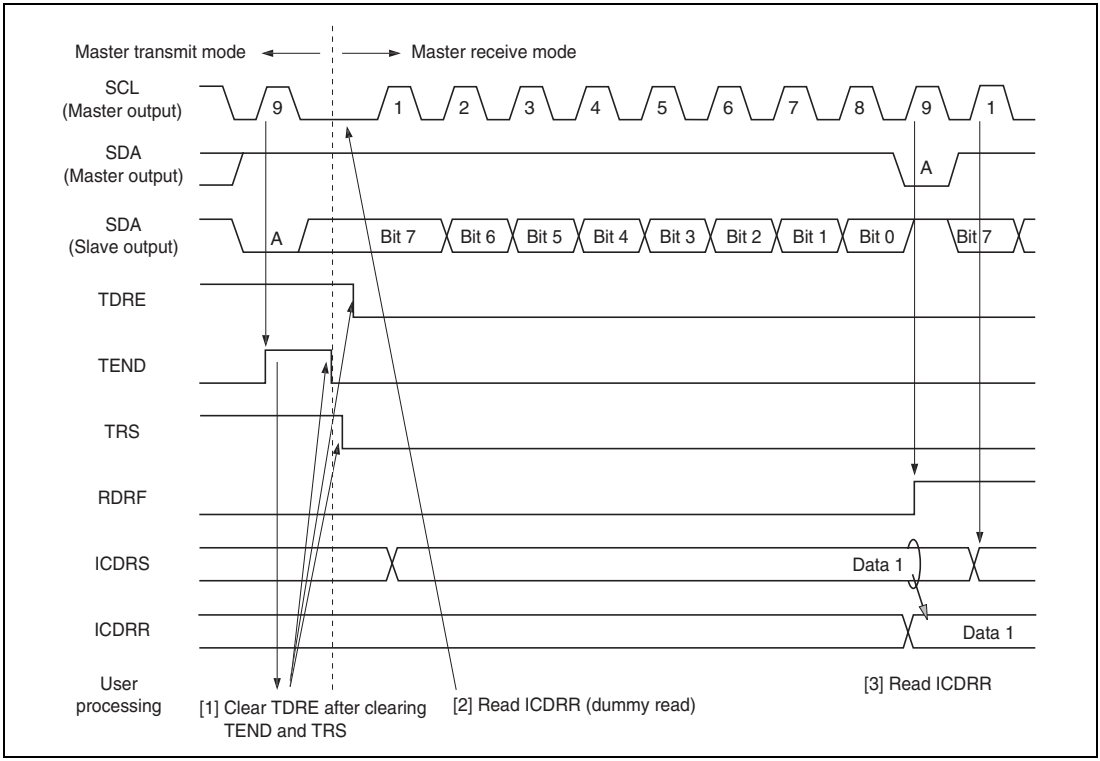


Figure 20.7 Master Receive Mode Operation Timing (1)

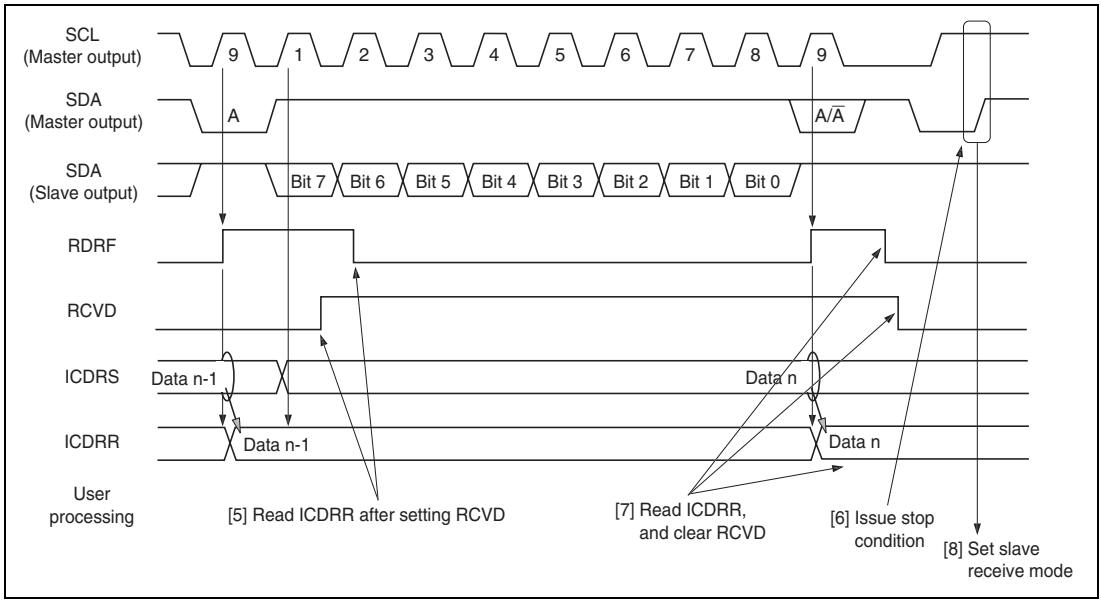


Figure 20.8 Master Receive Mode Operation Timing (2)

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 the MLS bit in ICMR and the CKS4 to CKS0 bits in ICCKS1 to 1. (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/$\overline{W}$) is 1, the TRS and ICSR bits in ICCR1 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 free.
5. Clear TDRE.

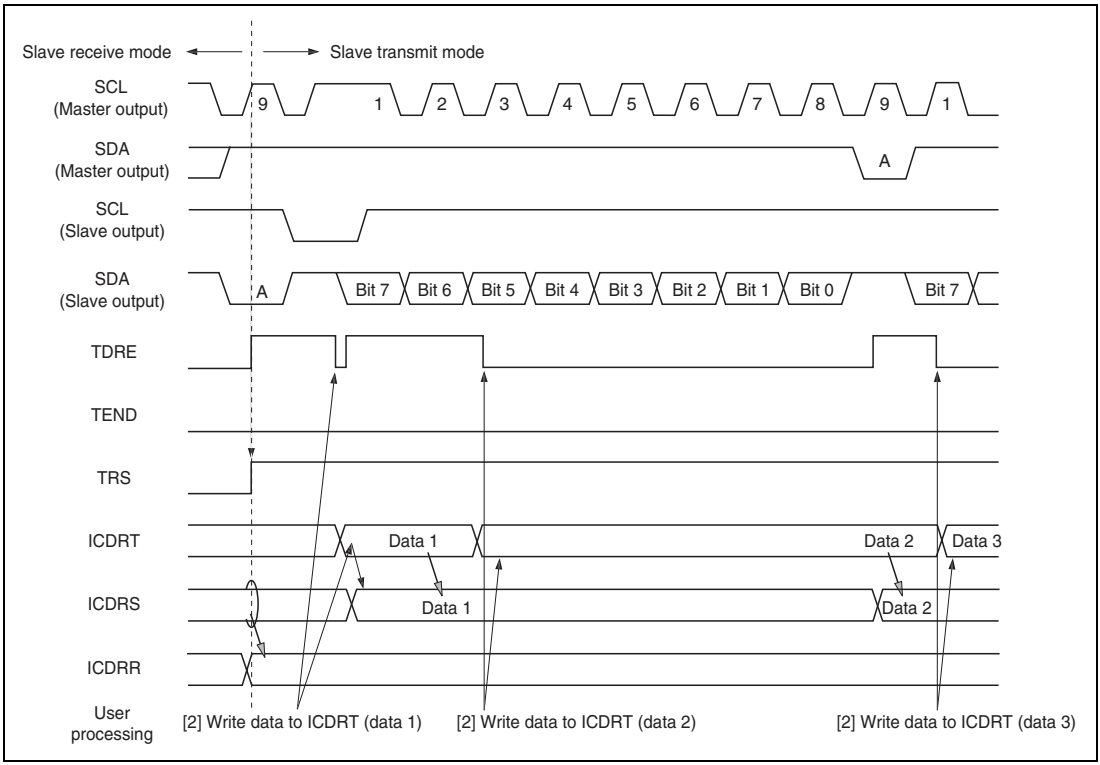


Figure 20.9 Slave Transmit Mode Operation Timing (1)

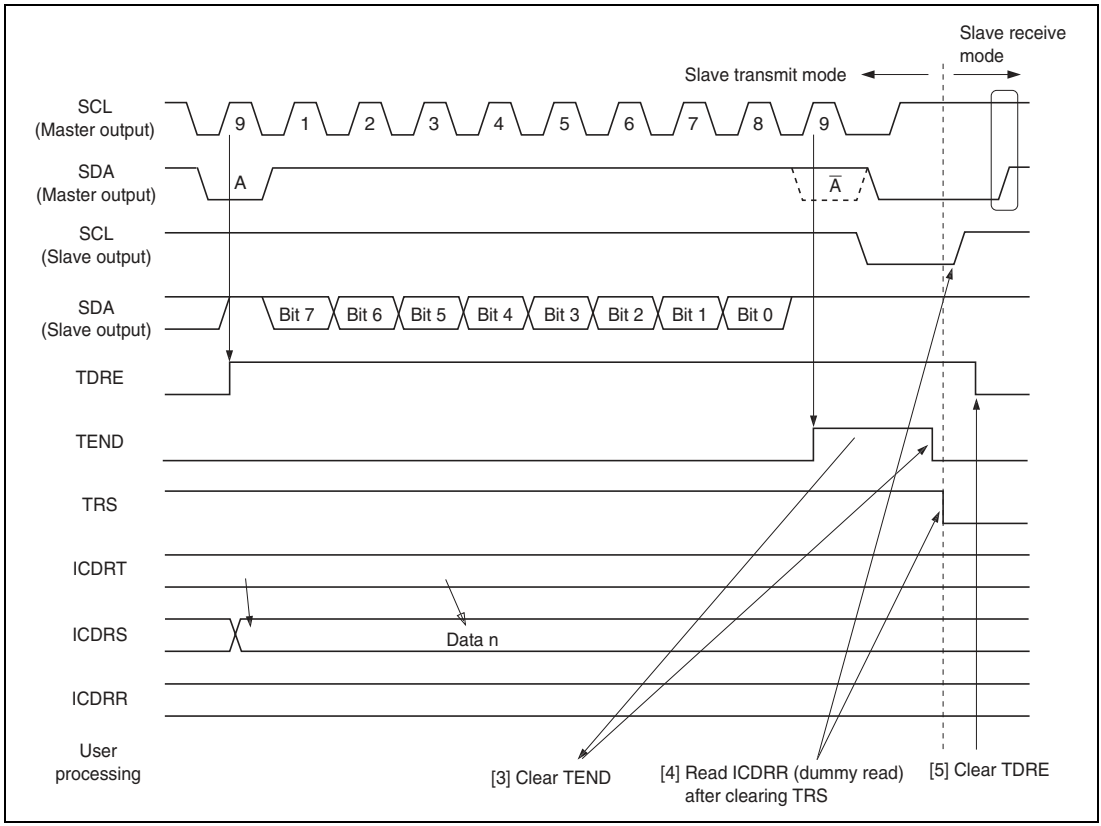


Figure 20.10 Slave Transmit Mode Operation Timing (2)

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 the MLS bit in ICMR and the CKS4 to CKS0 bits in ICCKS1 to 1. (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/$\overline{W}$, 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.

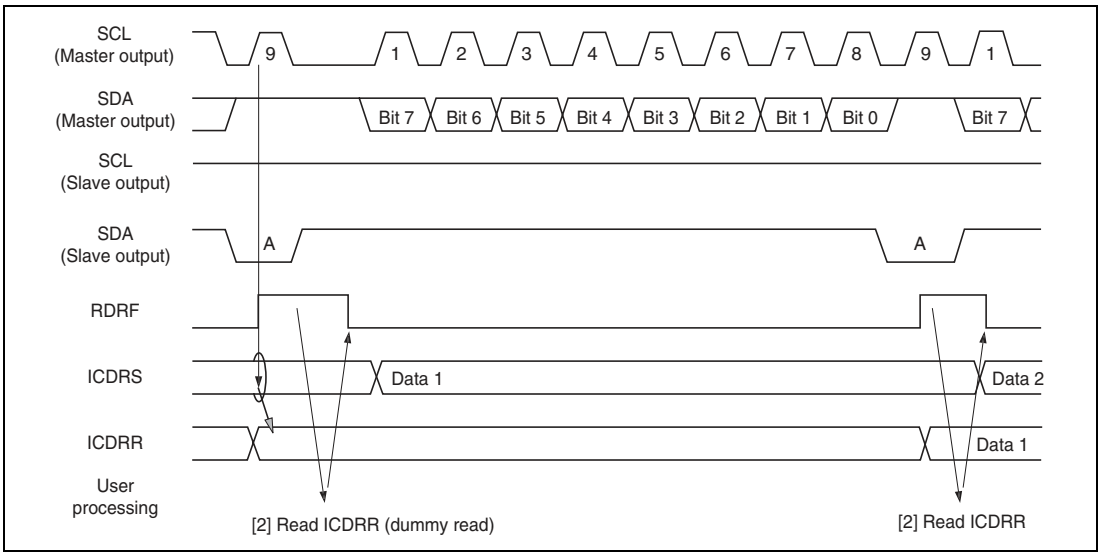


Figure 20.11 Slave Receive Mode Operation Timing (1)

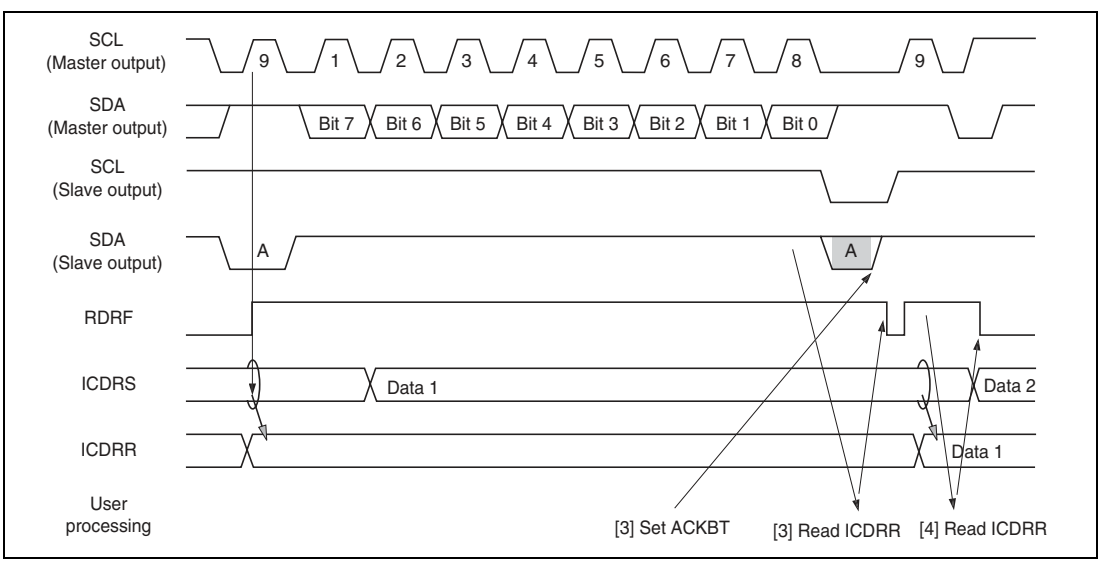


Figure 20.12 Slave Receive Mode Operation Timing (2)

20.4.6 Noise Canceller

The logic levels at the SCL and SDA pins are routed through noise cancellers before being latched internally. Figure 20.16 shows a block diagram of the noise canceller circuit.

The noise canceller consists of two cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the peripheral clock, but is not passed forward to the next circuit unless the outputs of both latches agree. If they do not agree, the previous value is held.

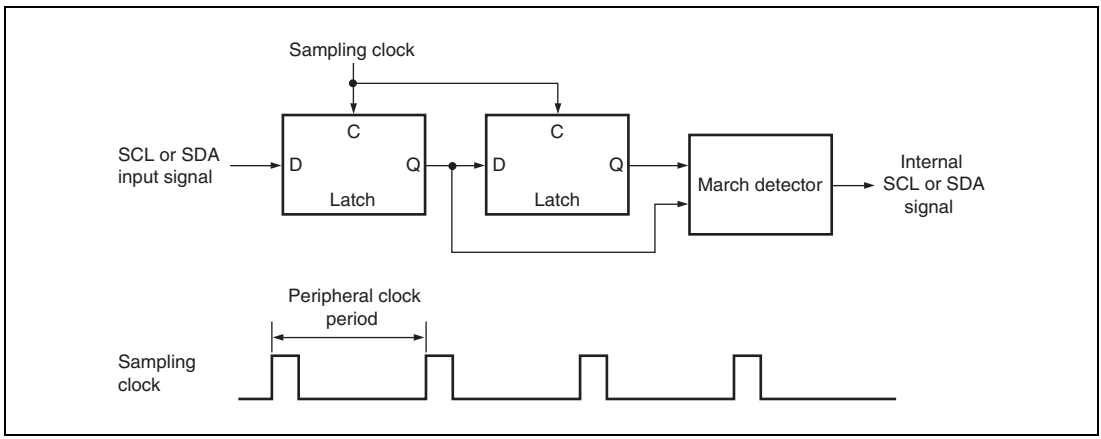


Figure 20.13 Block Diagram of Noise Conceller

20.4.7 Example of Use

Flowcharts in respective modes that use the I²C bus interface are shown in figures 20.17 to 20.20.

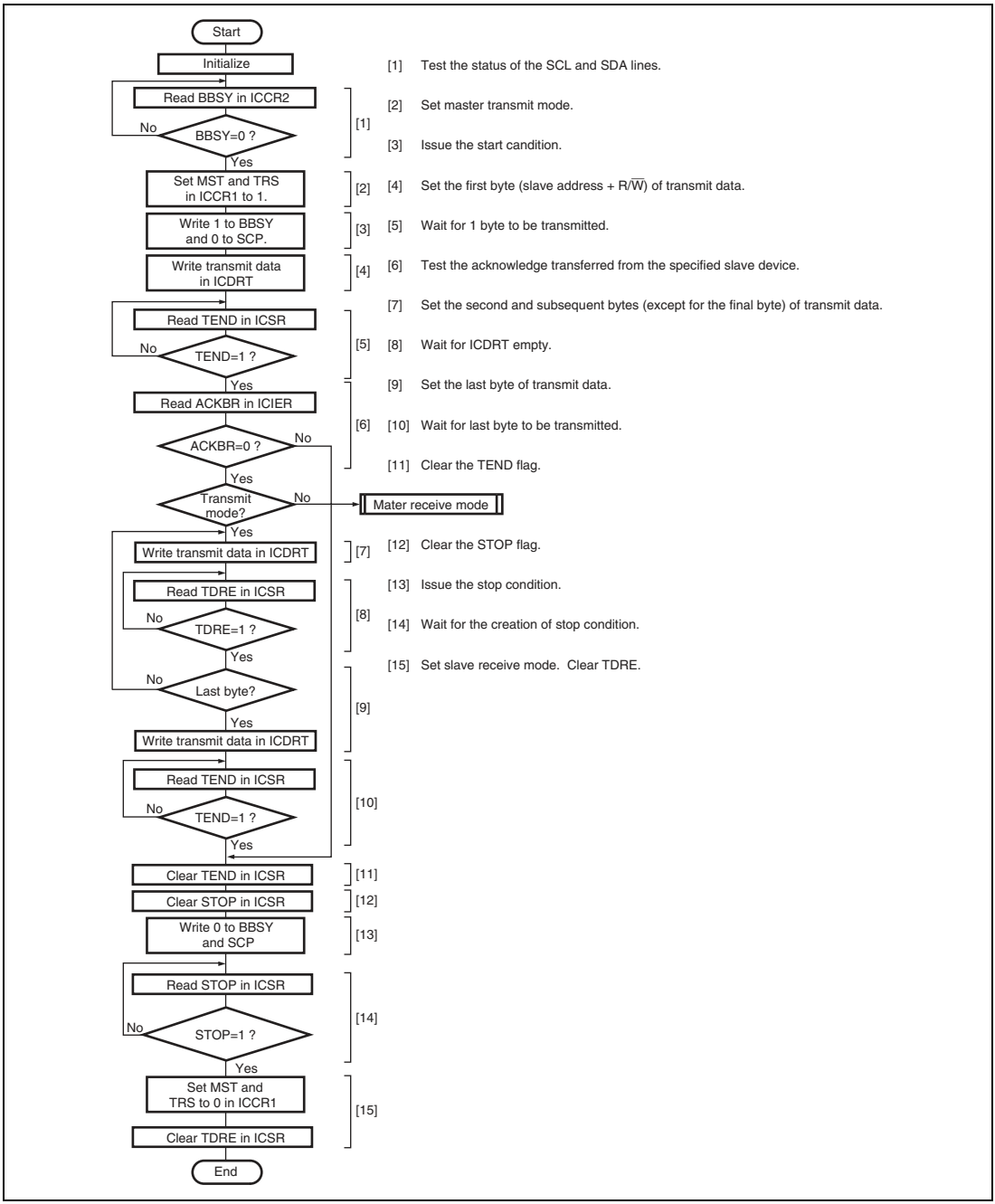
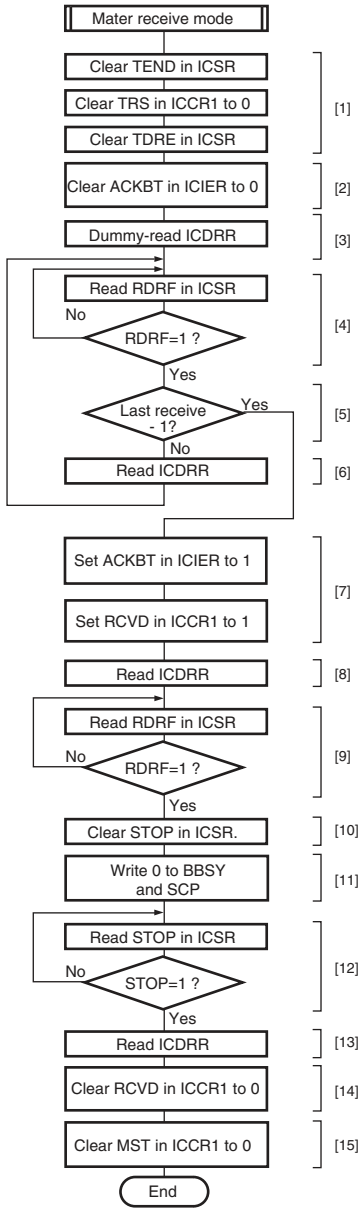


Figure 20.14 Sample Flowchart for Master Transmit Mode



[1] Clear TEND, select master receive mode, and then clear TDRE.*1*2

[2] Set acknowledge to the transmit device.*1

[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 last.

[7] Set acknowledge of the final byte. Disable continuous reception (RCVD = 1).

[8] Read the (final byte - 1) of receive 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.

Note: 1. Do not activate an interrupt during the execution of steps [1] to [3].
2. When one byte is received, steps [2] to [6] are skipped; step [7] is executed after step [1]. Setp [8] is ICDRR dummy read.

Figure 20.15 Sample Flowchart for Master Receive Mode

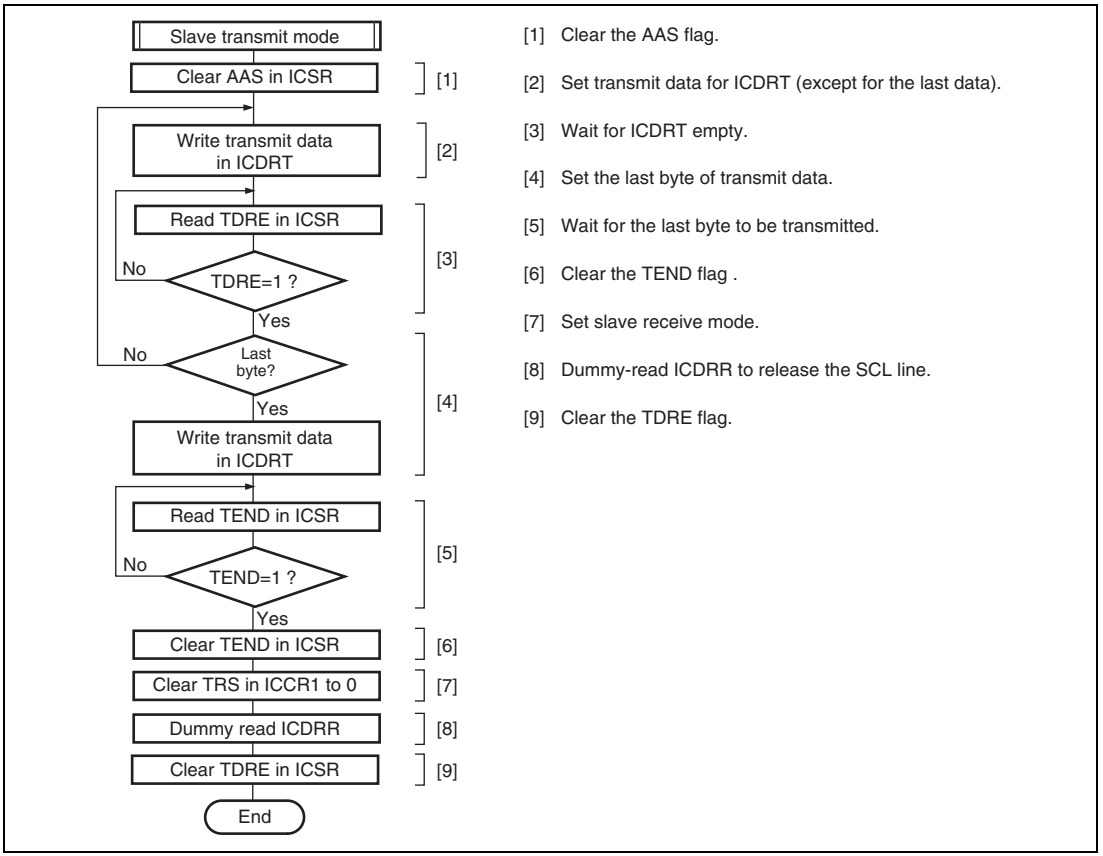


Figure 20.16 Sample Flowchart for Slave Transmit Mode

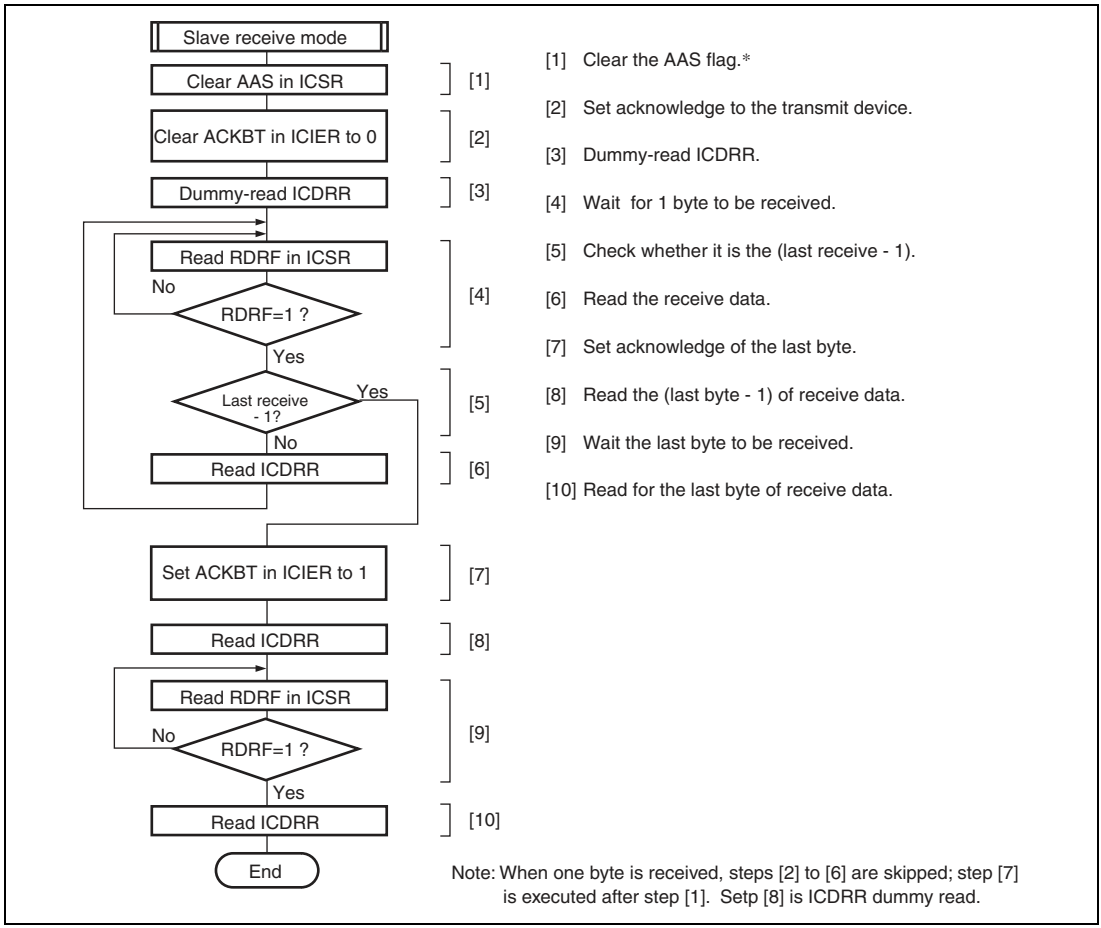


Figure 20.17 Sample Flowchart for Slave Receive Mode

20.5 Interrupt Request

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK receive, STOP recognition, and arbitration lost/overrun error. Table 20.3 shows the contents of each interrupt request.

A common interrupt vector (IICI) is assigned to each interrupt source.

Table 20.3 Interrupt Requests

Interrupt Request	Abbreviation	Interrupt Condition
Transmit Data Empty	TXI	(TDRE=1) • (TIE=1)
Transmit End	TEI	(TEND=1) • (TEIE=1)
Receive Data Full	RXI	(RDRF=1) • (RIE=1)
STOP Recognition	STPI	(STOP=1) • (STIE=1)
NACK Receive	NAKI	{(NACKF=1)+(AL=1)} • (NAKIE=1)
Arbitration Lost/Overrun Error		

When interrupt conditions described in table 20.3 are 1 and the I bit in CCR is 0, the CPU executes an interrupt exception processing. Interrupt sources should be cleared in the exception processing. TDRE and TEND are automatically cleared to 0 by writing the transmit data to ICDRT. RDRF are automatically cleared to 0 by reading ICDRR. TDRE is set to 1 again at the same time when transmit data is written to ICDRT. When TDRE 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 20.21 shows the timing of the bit synchronous circuit and table 20.4 shows the time when SCL output changes from low to Hi-Z then SCL is monitored.

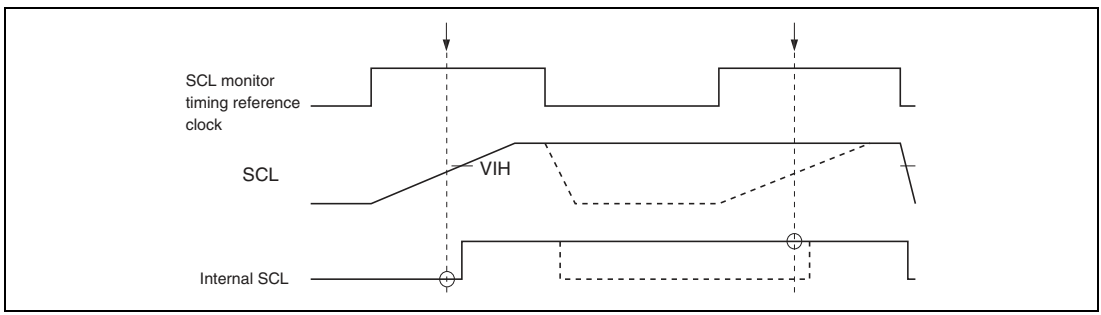


Figure 20.18 The Timing of the Bit Synchronous Circuit

Table 20.4 Time for Monitoring SCL

CKS3	CKS2	Time for Monitoring SCL
0	0	7.5 tpcyc
	1	19.5 tpcyc
1	0	17.5 tpcyc
	1	41.5 tpcyc

20.7 Usage Notes

A stop condition or retransmit start condition should be issued after the falling edge of the ninth clock is recognized. The falling edge of the ninth clock is recognized by checking the SCLO bit in the I²C bus control register 2 (ICCR2).

A stop condition or retransmit start condition may not be output normally if issuance of a stop or retransmit start condition is attempted with a certain timing under either of the following cases. There is no problem in uses under conditions other than the blow.

1. When the rising speed of SCL is lowered due to the load of the SCL line (load capacitance or pull-up resistance) exceeding the time defined in section 20.6, Bit Synchronous Circuit.
2. When the bit synchronous circuit works because the low-level period between the eighth and ninth clock pulses is extended by the slave device.

Section 21 Serial I/O with FIFO (SIOF)

This LSI includes a clock-synchronized serial I/O module with FIFO (SIOF) that comprises two channels. The functions of SIOF_0 and SIOF_1 are the same.

21.1 Features

- Serial transfer
  - 16-stage 32-bit FIFOs (independent transmission and reception)
  - Supports 8-bit data/16-bit data/16-bit stereo audio input and 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 (SIOFIn (n = 0, 1))
- DMA transfer
  - Supports DMA transmission and reception by a transfer request for transmission and reception

Figure 21.1 shows a block diagram of the SIOF.

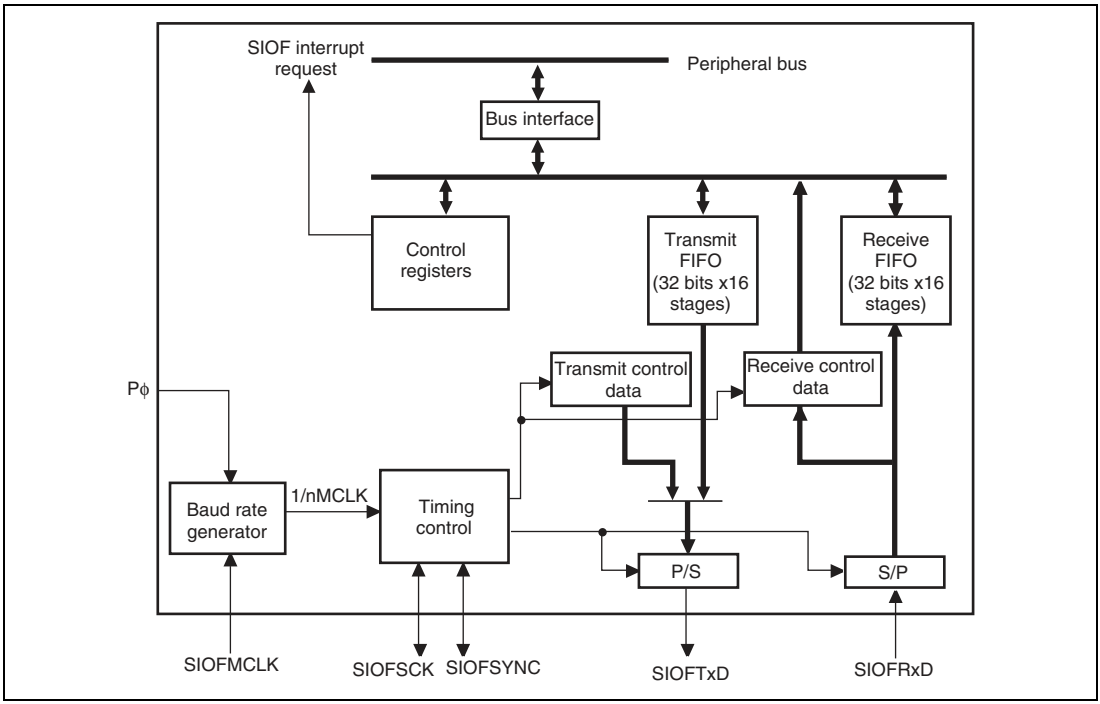


Figure 21.1 Block Diagram of SIOF

21.2 Input/Output Pins

The pin configuration in this module is shown in table 21.1.

Table 21.1 Pin Configuration

Channel	Pin Name	Abbreviation*	I/O	Function
0	SIOF0_MCLK	SIOFMCLK	Input	Master clock input
	SIOF0_SCK	SIOFSCK	I/O	Serial clock (common to transmission/reception)
	SIOF0_SYNC	SIOFSYNC	I/O	Frame synchronous signal (common to transmission/reception)
	SIOF0_TxD	SIOFTxD	Output	Transmit data
	SIOF0_RxD	SIOFRxD	Input	Receive data
1	SIOF1_MCLK	SIOFMCLK	Input	Master clock input
	SIOF1_SCK	SIOFSCK	I/O	Serial clock (common to transmission/reception)
	SIOF1_SYNC	SIOFSYNC	I/O	Frame synchronous signal (common to transmission/reception)
	SIOF1_TxD	SIOFTxD	Output	Transmit data
	SIOF1_RxD	SIOFRxD	Input	Receive data

Note: * The pins for channel 0 and channel 1 are collectively called SIOFMCLK, SIOFSCK, SIOFSYNC, SIOFTxD, and SIOFRxD in the following descriptions.

21.3 Register Descriptions

The SIOF has the following registers. Refer to section 37, List of Registers, for more details on the addresses and states of these registers in each operating mode. In the register descriptions following this section, channel numbers are omitted.

(1) Channel 0

- Mode register_0 (SIMDR_0)
- Control register_0 (SICTR_0)
- Transmit data register_0 (SITDR_0)
- Receive data register_0 (SIRDR_0)
- Transmit control data register_0 (SITCR_0)
- Receive control data register_0 (SIRCR_0)
- Status register_0 (SISTR_0)
- Interrupt enable register_0 (SIIER_0)
- FIFO control register_0 (SIFCTR_0)
- Clock select register_0 (SISCR_0)
- Transmit data assign register_0 (SITDAR_0)
- Receive data assign register_0 (SIRDAR_0)
- Control data assign register_0 (SICDAR_0)

(2) Channel 1

- Mode register_1 (SIMDR_1)
- Control register_1 (SICTR_1)
- Transmit data register_1 (SITDR_1)
- Receive data register_1 (SIRDR_1)
- Transmit control data register_1 (SITCR_1)
- Receive control data register_1 (SIRCR_1)
- Status register_1 (SISTR_1)
- Interrupt enable register_1 (SIIER_1)
- FIFO control register_1 (SIFCTR_1)
- Clock select register_1 (SISCR_1)
- Transmit data assign register_1 (SITDAR_1)
- Receive data assign register_1 (SIRDAR_1)
- Control data assign register_1 (SICDAR_1)

21.3.1 Mode Register (SIMDR)

SIMDR is a 16-bit readable/writable register that sets the SIOF operating mode.

Bit	Bit Name	Initial Value	R/W	Description
15	TRMD1	1	R/W	Transfer Mode 1, 0
14	TRMD0	0	R/W	Select transfer mode. For details, see table 21.2. 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 (The SIOFTxD signal is transmitted at the rising edge of SIOFSCK.) 1: The SIOFRxD signal is sampled at the rising edge of SIOFSCK (The SIOFTxD signal is transmitted at the falling edge of SIOFSCK.) Note: This bit is valid only in master mode.

Bit	Bit Name	Initial Value	R/W	Description
11	FL3	0	R/W	Frame Length 3 to 0
10	FL2	0	R/W	00xx: Data length is 8 bits and frame length is 8 bits.
9	FL1	0	R/W	0100: Data length is 8 bits and frame length is 16 bits.
8	FL0	0	R/W	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 synchronous pulse in master mode. 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 or reception 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 21.2 Operation in Each Transfer Mode

Transfer Mode	Master/Slave	SIOFSYNC	Bit Delay	Control Data Method*1
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*2	Not supported

Notes: *1 The control data method is valid only when the FL bit is specified as 1xxx. (x: Don't care.)

*2 Depending on the timing to start SYNC signal output in master mode 2, the SYNC signal of the head frame in the high period can be extended to I bit. For details, see section 21.5, Usage Notes.

21.3.2 Control Register (SICTR)

SICTR is a 16-bit readable/writable register that sets the SIOF operating state.

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. This bit is initialized in module stop mode.
14	FSE	0	R/W	Frame Synchronous 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. This bit is initialized in module stop mode.
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. This bit is initialized in module stop mode.
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. This bit is initialized in module stop mode.
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. This bit should be cleared to 0 before setting the register to be initialized. • 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 data register and transmit-related status. The following are initialized. — SITDR — SITCR — Transmit FIFO write pointer and read pointer — TCRDY, TFEMP, and TDREQ bits in SISTR — TXE bit
0	RXRST	0	R/W	Receive Reset 0: Does not reset receive operation 1: Resets receive operation • This bit setting becomes valid immediately. This bit should be cleared to 0 before setting the register to be initialized. • When the 1 setting for this bit becomes valid, the SIOF immediately disables reception from the SIOFRxD pin, and initializes the receive data register and receive-related status. The following are initialized. — SIRDR — SIRCR — Receive FIFO write pointer and read pointer — RCRDY, RFFUL, and RDREQ bits in SISTR — RXE bit

21.3.3 Transmit Data Register (SITDR)

SITDR is a 32-bit write-only register that specifies the SIOF transmit data.

SITDR is initialized by the conditions specified in section 37, List of Registers, or by a transmit reset caused by the TXRST bit in SICTR.

SITDR is initialized in module stop mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SITDL 15 to 0	All 0	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 to 0	All 0	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 and is initialized by the conditions specified in section 37, List of Registers, or by a receive reset caused by the RXRST bit in SICTR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SIRDL 15 to 0	All 0	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 to 0	All 0	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 in module stop mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SITC0 15 to 0	All 0	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 to 0	All 0	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 to 0	All 0	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 to 0	All 0	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 read-only 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.

SISTR is initialized in module stop mode.

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 space in the transmit FIFO does not exceed the size specified by the TFWM bit in SIFCTR. 1: Indicates that the size of empty space in the transmit FIFO exceeds the size specified by the TFWM bit in SIFCTR. A transmit data transfer request is issued when the empty space 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 space 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 space in the receive FIFO does not exceed the size specified by the RFWM bit in SIFCTR. 1: Indicates that the size of valid space in the receive FIFO exceeds the size specified by the RFWM bit in SIFCTR. A receive data transfer request is issued when the valid space 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 space 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	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	Bit Name	Initial Value	R/W	Description
15	TFWM2	0	R/W	Transmit FIFO Watermark
14	TFWM1	0	R/W	000: Issue a transfer request when 16 stages of the transmit FIFO are empty.
13	TFWM0	0	R/W	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 TDREQ bit in SISTR. • The transmit FIFO is always used as 16 stages of the FIFO regardless of these bit settings.
12	TFUA4	1	R	Transmit FIFO Usable Area
11	TFUA3	0	R	Indicate the number of words that can be transferred by the CPU or DMAC as B'00000 (full) to B'10000 (empty).
10	TFUA2	0	R	
9	TFUA1	0	R	
8	TFUA0	0	R	

Bit	Bit Name	Initial Value	R/W	Description
7	RFWM2	0	R/W	Receive FIFO Watermark
6	RFWM1	0	R/W	000: Issue a transfer request when 1 stage or more of the receive FIFO are valid.
5	RFWM0	0	R/W	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 RDREQ bit in SISTR. • The receive FIFO is always used as 16 stages of the FIFO regardless of these bit settings.
4	RFUA4	0	R	Receive FIFO Usable Area
3	RFUA3	0	R	Indicate the number of words that can be transferred by the CPU or DMAC as B'00000 (empty) to B'10000 (full).
2	RFUA2	0	R	
1	RFUA1	0	R	
0	RFUA0	0	R	

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 TRMD1 and TRMD0 bits in SIMDR are specified as B'10 or B'11.

Bit	Bit Name	Initial Value	R/W	Description
15	MSSEL	1	R/W	Master Clock Source Selection 0: Uses the input signal of the SIOFMCLK pin as the master clock 1: Uses Pϕ as the master clock The master clock is the clock input to the baud rate generator.
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	BRPS4	0	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).
11	BRPS3	0	R/W	
10	BRPS2	0	R/W	
9	BRPS1	0	R/W	
8	BRPS0	0	R/W	
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	BRDV2	0	R/W	Baud rate generator's Division Ratio Setting
1	BRDV1	0	R/W	Set the frequency division ratio for the output stage of the baud rate generator.
0	BRDV0	0	R/W	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*
				The final frequency division ratio of the baud rate generator is determined by BRPS × BRDV (maximum 1/1024).
				Note: *This setting is valid only when the BRPS4 to BRPS0 bits are set to B'00000.

21.3.11 Transmit Data Assign Register (SITDAR)

SITDAR is a 16-bit readable/writable register that specifies the position of the transmit data in a frame (slot number).

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
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	TDLA3	0	R/W	Transmit Left-Channel Data Assigns 3 to 0
10	TDLA2	0	R/W	Specify the position of left-channel data in a transmit frame as B'0000 (0) to B'1110 (14).
9	TDLA1	0	R/W	
8	TDLA0	0	R/W	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	TDRA3	0	R/W	Transmit Right-Channel Data Assigns 3 to 0
2	TDRA2	0	R/W	Specify the position of right-channel data in a transmit frame as B'0000 (0) to B'1110 (14).
1	TDRA1	0	R/W	
0	TDRA0	0	R/W	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 (slot number).

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	RDLA3	0	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.
10	RDLA2	0	R/W	
9	RDLA1	0	R/W	
8	RDLA0	0	R/W	
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	RDRA3	0	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.
2	RDRA2	0	R/W	
1	RDRA1	0	R/W	
0	RDRA0	0	R/W	

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 (slot number). SICDAR 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
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	CD0A3	0	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.
10	CD0A2	0	R/W	
9	CD0A1	0	R/W	
8	CD0A0	0	R/W	
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	CD1A3	0	R/W	Control Channel 1 Data Assigns 3 to 0
2	CD1A2	0	R/W	Specify the position of control channel 1 data in a receive or transmit frame as B'0000 (0) to B'1110 (14).
1	CD1A1	0	R/W	1111: Setting prohibited
0	CD1A0	0	R/W	• 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 two 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.

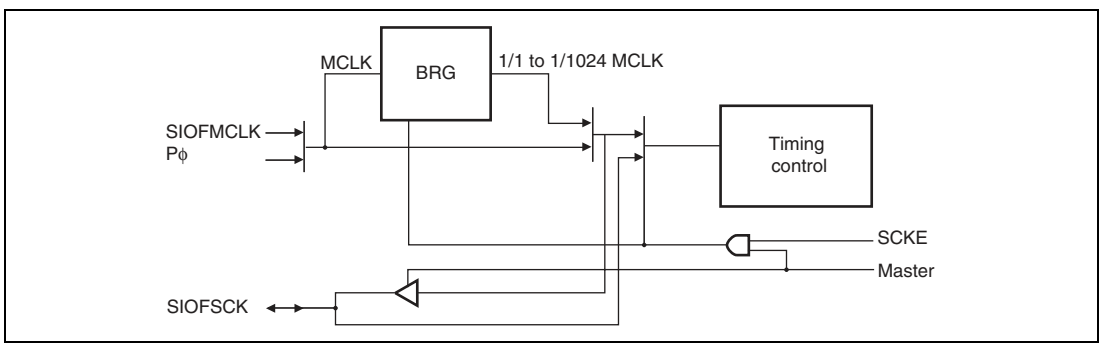


Figure 21.2 Serial Clock Supply

Table 21.3 shows an example of serial clock frequency.

Table 21.3 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 synchronous 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.

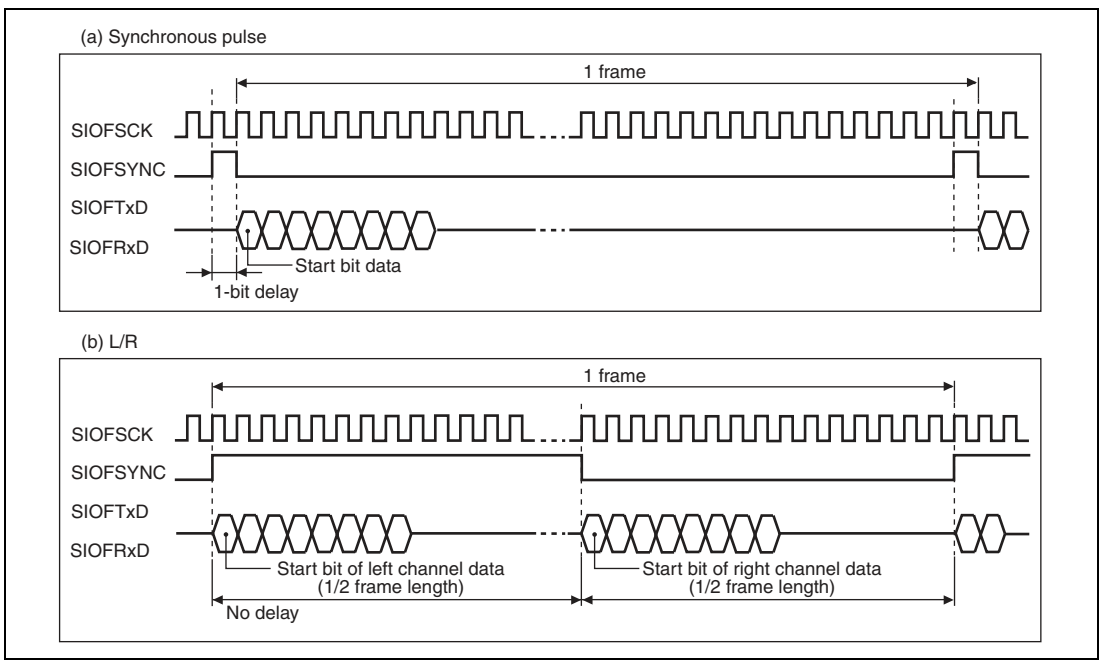


Figure 21.3 Serial Data Synchronization Timing

(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.

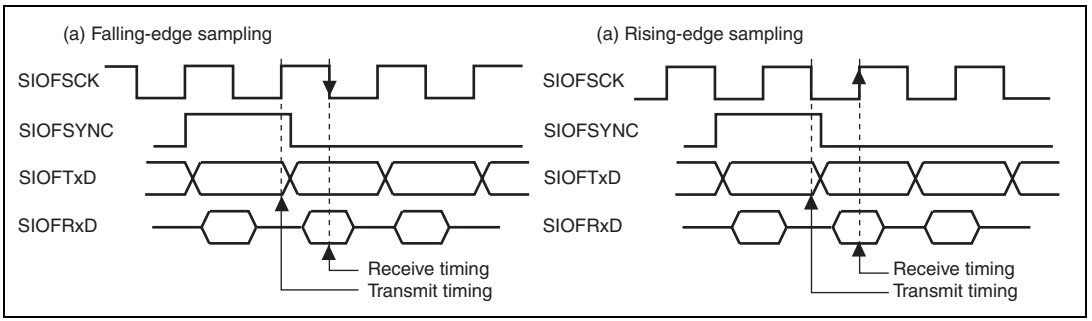


Figure 21.4 SIOF Transmit/Receive Timing

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 21.4. The transfer mode can be specified by the TRMD1 and TRMD0 bits in SIMDR.

Table 21.4 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

Note: * Depending on the timing of SYNC signal output, bit delay may be generated in head frame. For details, see section 21.5, Usage Notes.

(2) Frame Length

The length of the frame to be transferred by the SIOF is specified by the FL3 to FL0 bits in SIMDR. Table 21.5 shows the relationship between the FL3 to FL0 bit settings and frame length.

Table 21.5 Frame Length

FL3 to FL0	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

Note: 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 are 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.

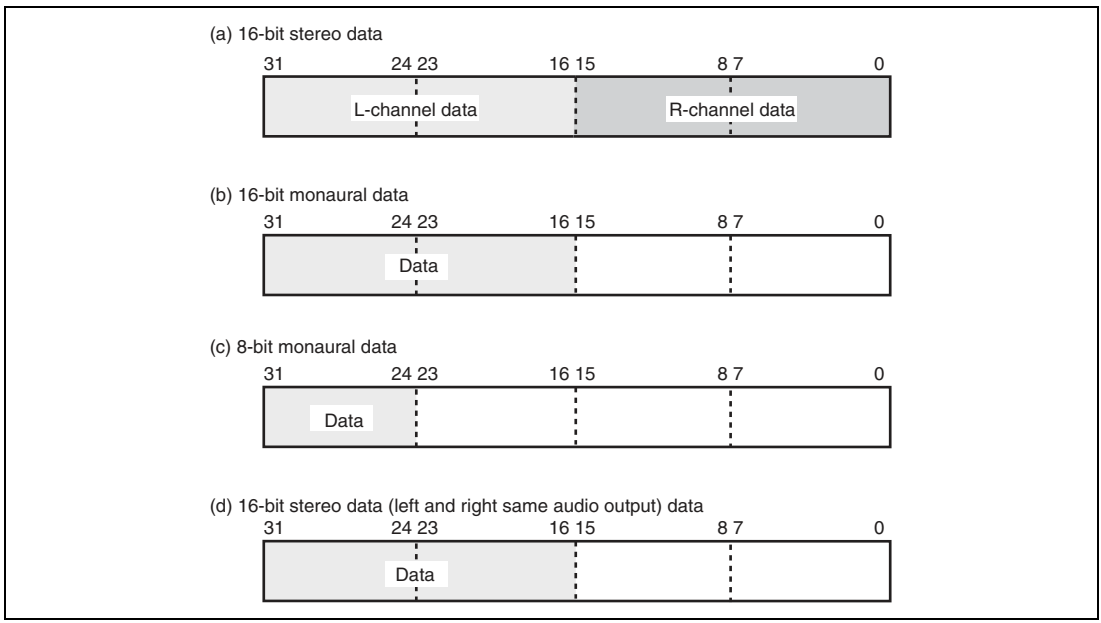


Figure 21.5 Transmit/Receive Data Bit Alignment

Note: In the figure, only the shaded areas are transmitted or received as valid data. 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. Tables 21.6 and 21.7 show the audio mode specification for transmit data and that for receive data, respectively.

Table 21.6 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

Note: x: Don't care

Table 21.7 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.

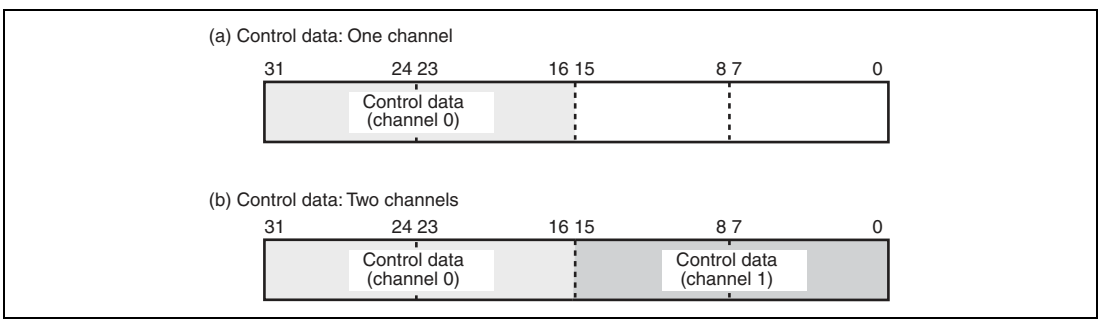


Figure 21.6 Control Data Bit Alignment

The number of channels in control data is specified by the CD0E and CD1E bits in SICDAR. Table 21.8 shows the relationship between the number of channels in control data and bit settings.

Table 21.8 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.

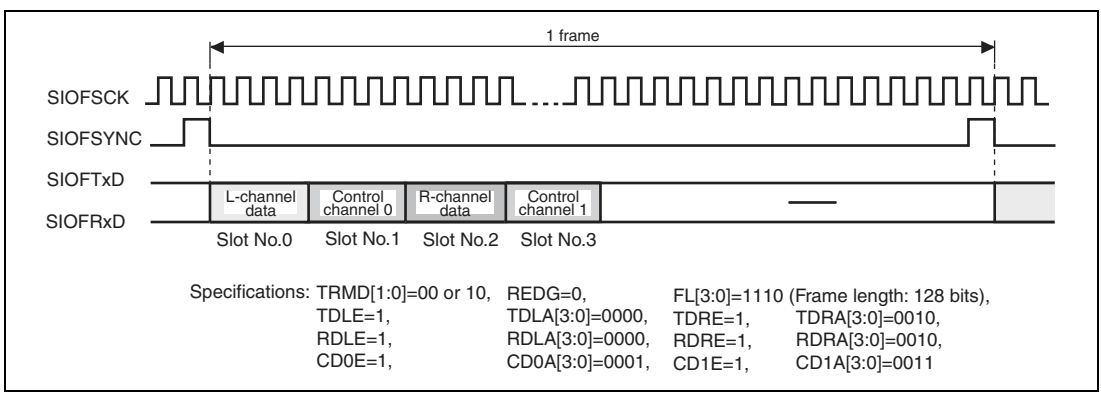


Figure 21.7 Control Data Interface (Slot Position)

(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.

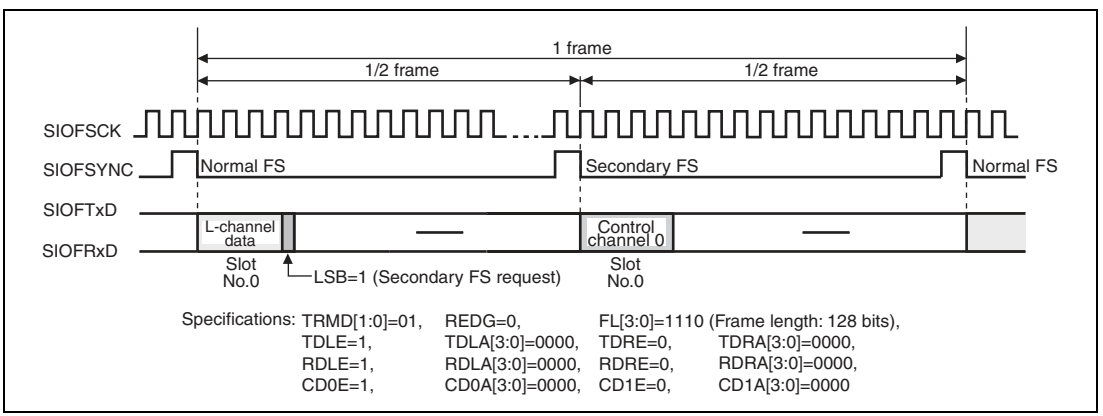


Figure 21.8 Control Data Interface (Secondary FS)

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 TFWM2 to TFWM0 bits and RFWM2 to RFWM0 bits in SIFCTR, respectively. Tables 21.9 and 21.10 summarize the conditions specified by SIFCTR.

Table 21.9 Conditions to Issue Transmit Request

TFWM2 to TFWM0	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 21.10 Conditions to Issue Receive Request

RFWM2 to RFWM0	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.

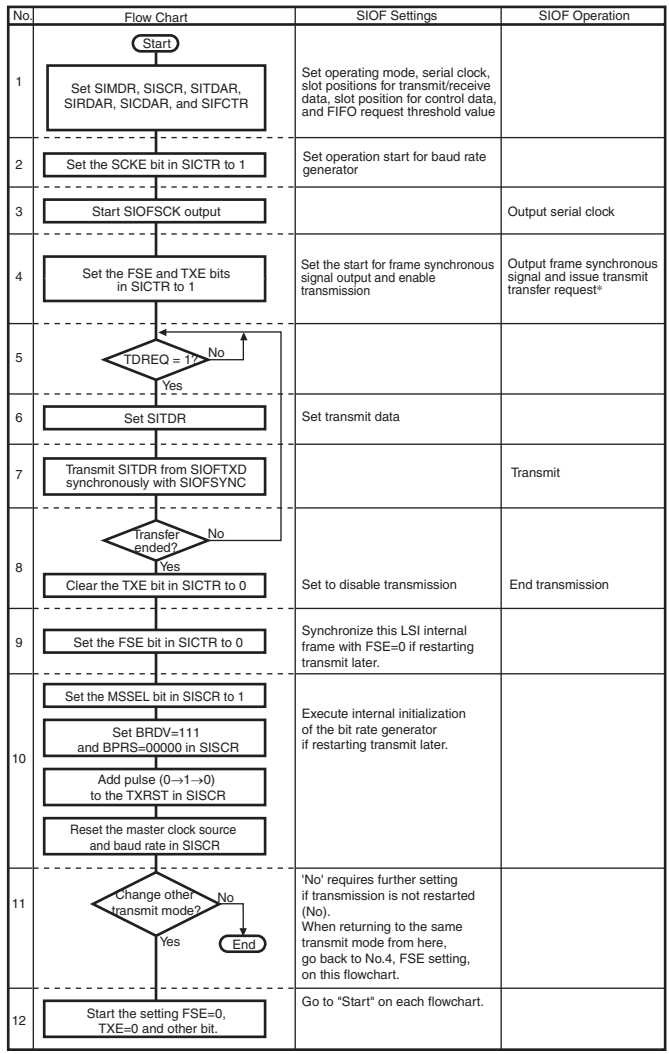
- Transmit FIFO: The number of empty FIFO stages is indicated by the TFUA4 to TFUA0 bits in SIFCTR.
- Receive FIFO: The number of valid data stages is indicated by the RFUA4 to RFUA0 bits in SIFCTR.

  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.

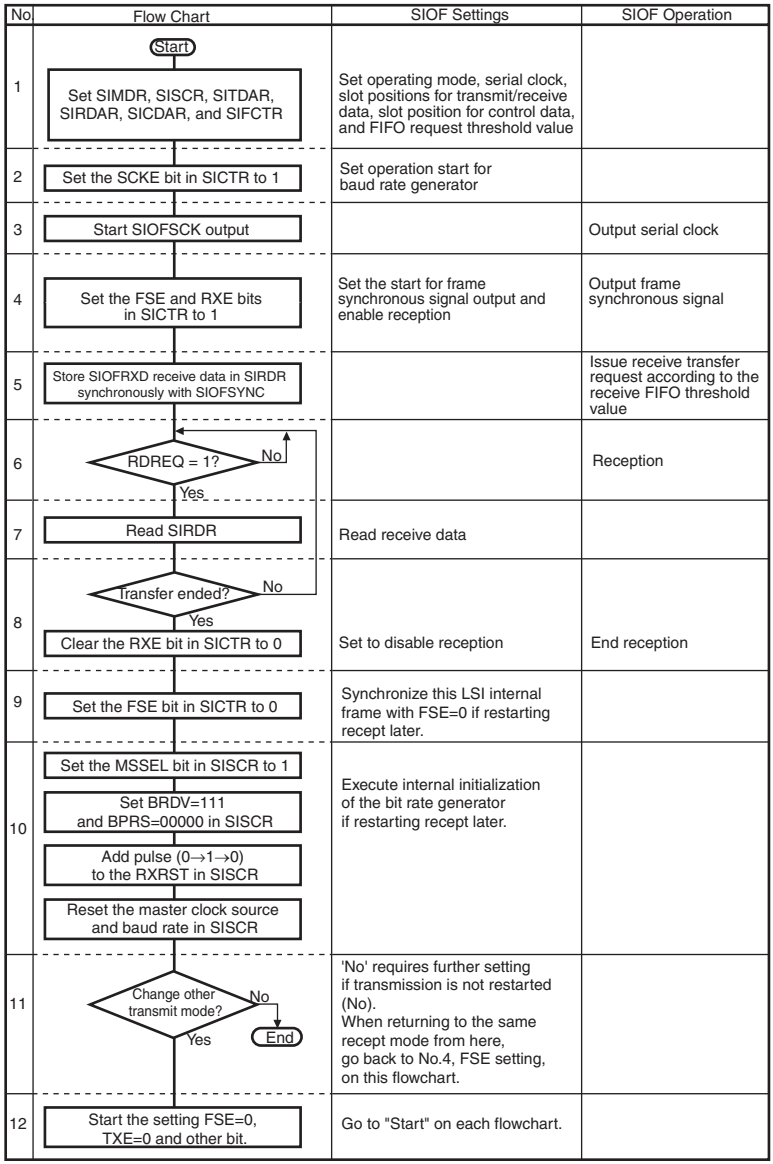


Figure 21.10 Example of Receive Operation in Master Mode

(3) Transmission in Slave Mode

Figure 21.11 shows an example of settings and operation for slave mode transmission.

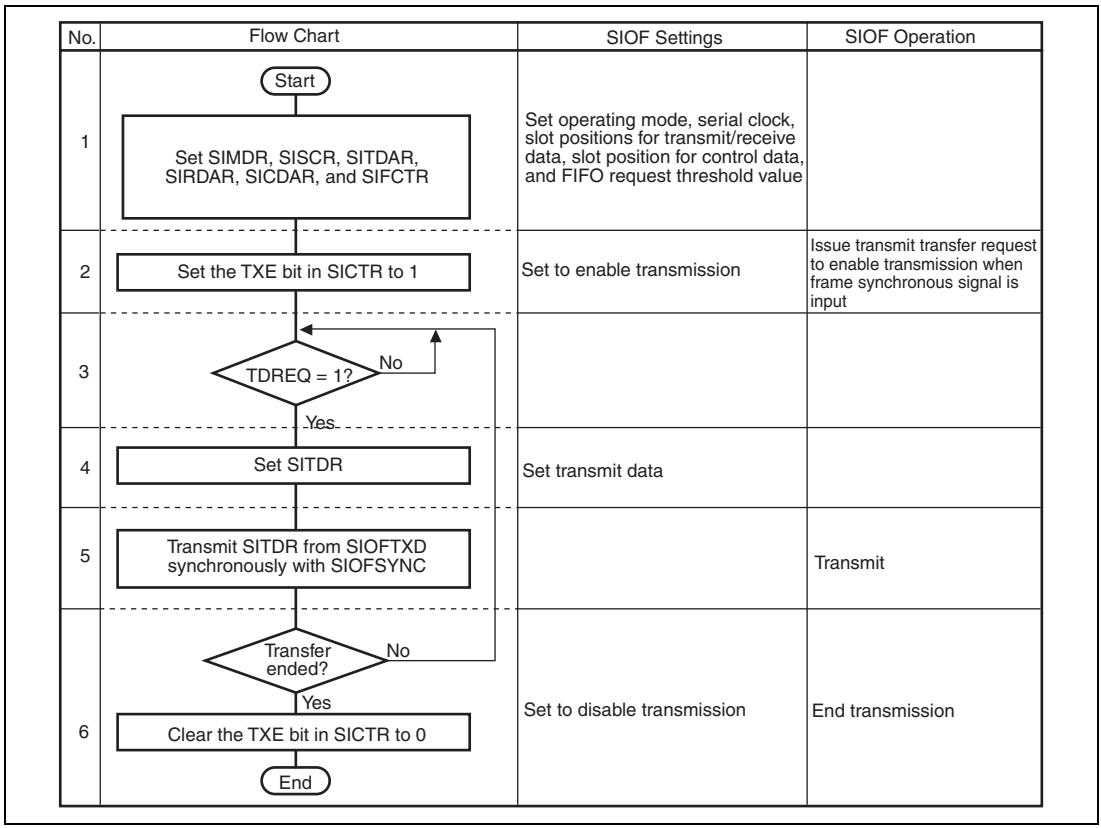


Figure 21.11 Example of Transmit Operation in Slave Mode

(4) Reception in Slave Mode

Figure 21.12 shows an example of settings and operation for slave mode reception.

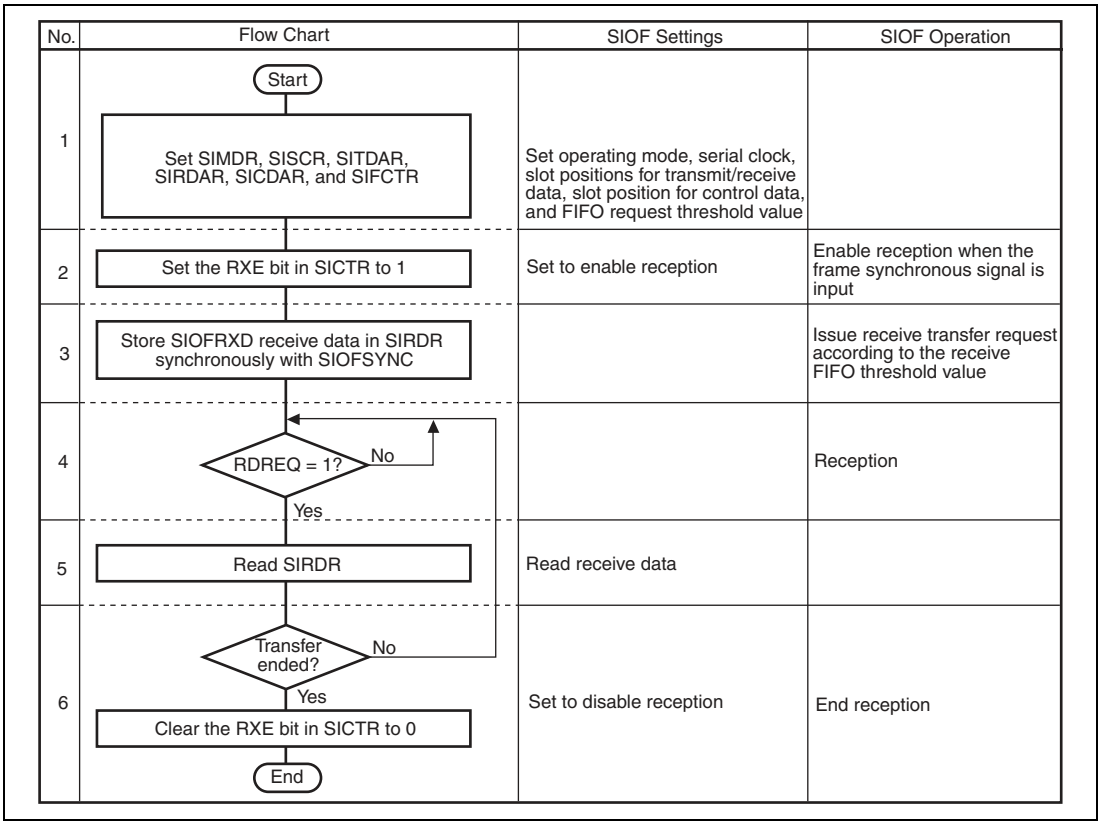


Figure 21.12 Example of Receive Operation in Slave Mode

(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 21.11 shows the details of initialization upon transmit or receive reset.

Table 21.11 Transmit and Receive Reset

Type	Objects Initialized
Transmit reset	SITDR
	Transmit FIFO write pointer and read pointer
	TCRDY, TFEMP, and TDREQ bits in SISTR
	TXE bit in SICTR
Receive reset	SIRDR
	Receive FIFO write pointer and read 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).

(6) Module Stop Mode

The SIOF stops the transmit/receive operation in module stop mode. Then the following contents are initialized.

- SITDR
- SITCR
- Read pointer of transmit/receive FIFO
- Write pointer of transmit/receive FIFO
- SISTR
- SICTR

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 21.12 lists the SIOF interrupt sources.

Table 21.12 SIOF Interrupt Sources

No.	Classification	Bit Name	Function Name	Description
1	Transmission	TDREQ	Transmit FIFO transfer request	The transmit FIFO stores data of specified size or more.
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.

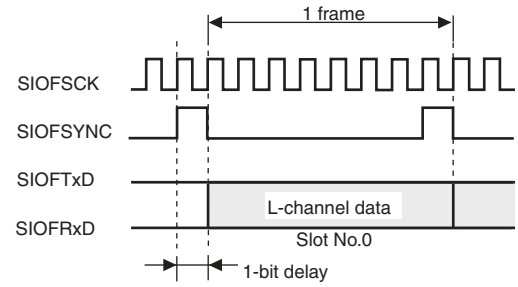
- Transmit FIFO underflow (TFUDF)
  The immediately preceding transmit data is again transmitted.
- Transmit FIFO overflow (TFOVF)
  The contents of the transmit FIFO are protected, and the write operation causing the overflow is ignored.
- Receive FIFO overflow (RFOVF)
  Data causing the overflow is 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 FSYN 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 21.19.

(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



Specifications: TRMD[1:0]=00 or 10, REDG=0, FL[3:0]=0000 (frame length: 8 bits)
TDLE=1, TDLA[3:0]=0000, TDRE=0, TDRA[3:0]=0000,
RDLE=1, RDLA[3:0]=0000, RDRE=0, RDRA[3:0]=0000,
CD0E=0, CD0A[3:0]=0000, CD1E=0, CD1A[3:0]=0000

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

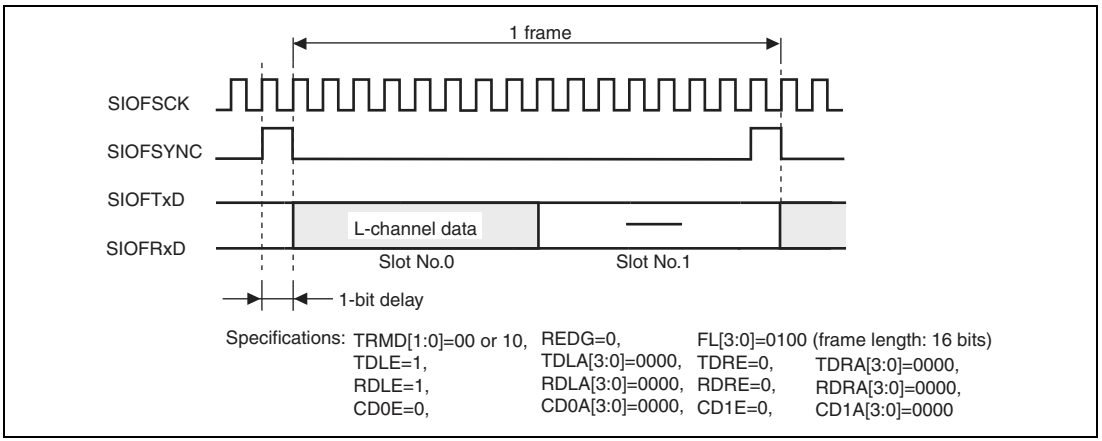


Figure 21.14 Transmit and Receive Timing (8-Bit Monaural Data (2))

(3) 16-bit Monaural Data (1)

Synchronous pulse method, falling edge sampling, slot No.0 used for transmit and receive data, and frame length = 64 bits

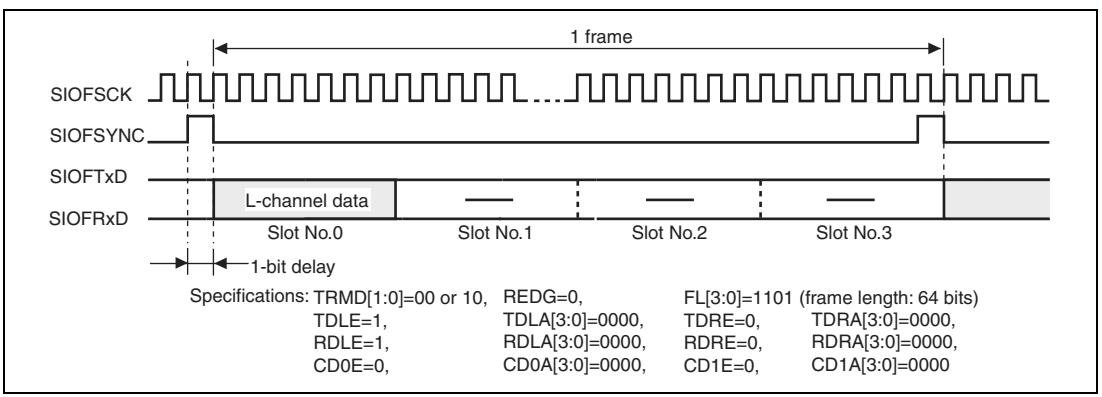


Figure 21.15 Transmit and Receive Timing (16-Bit Monaural Data (1))

(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

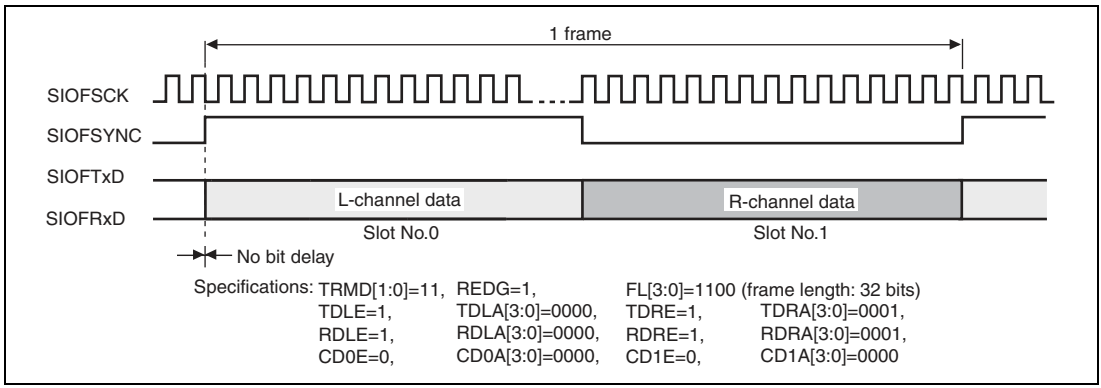


Figure 21.16 Transmit and Receive Timing (16-Bit Stereo Data (1))

(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

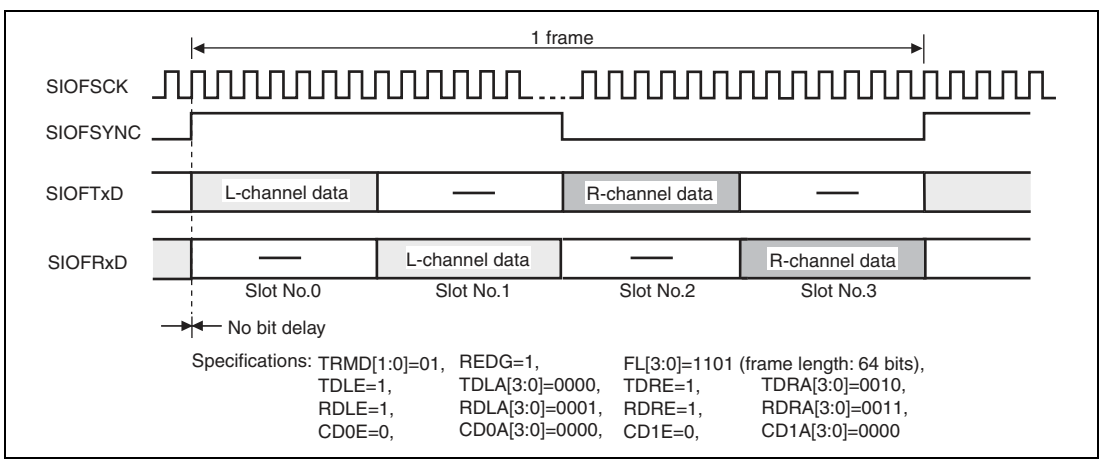


Figure 21.17 Transmit and Receive Timing (16-Bit Stereo Data (2))

(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 channel 0 data, slot No.3 used for control channel 1 data, and frame length = 128 bits

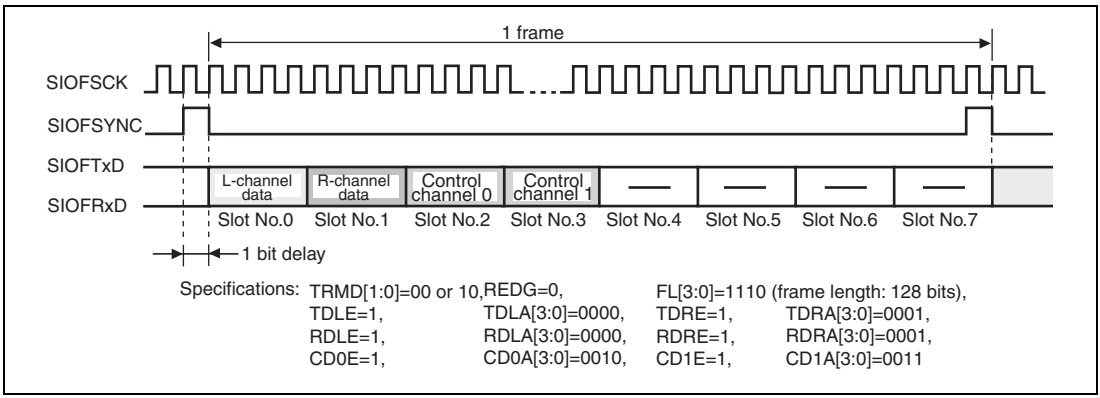


Figure 21.18 Transmit and Receive Timing (16-Bit Stereo Data (3))

(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 channel 0 data, slot No.3 used for control channel 1 data, and frame length = 128 bits

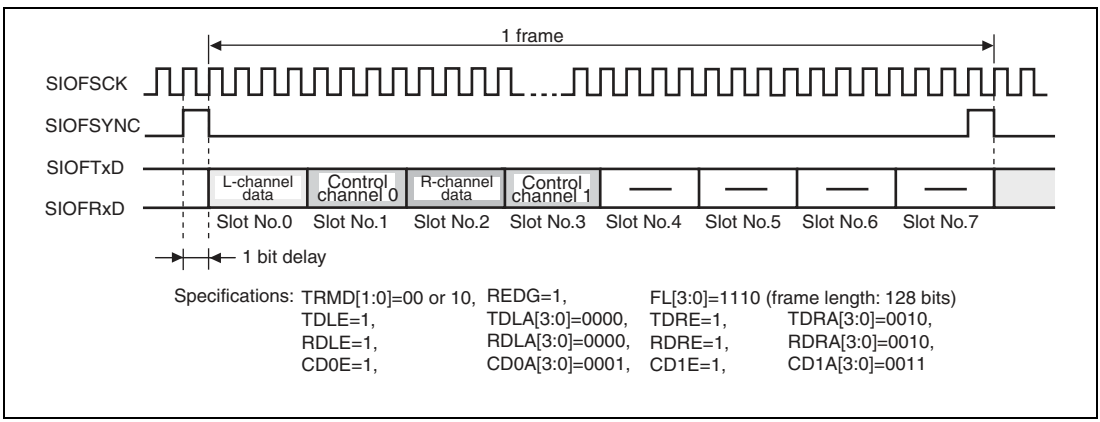


Figure 21.19 Transmit and Receive Timing (16-Bit Stereo Data (4))

(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 channel 0 data, slot No.3 used for control channel 1 data, and frame length = 128 bits

In this mode, valid data must be set to slot No. 0.

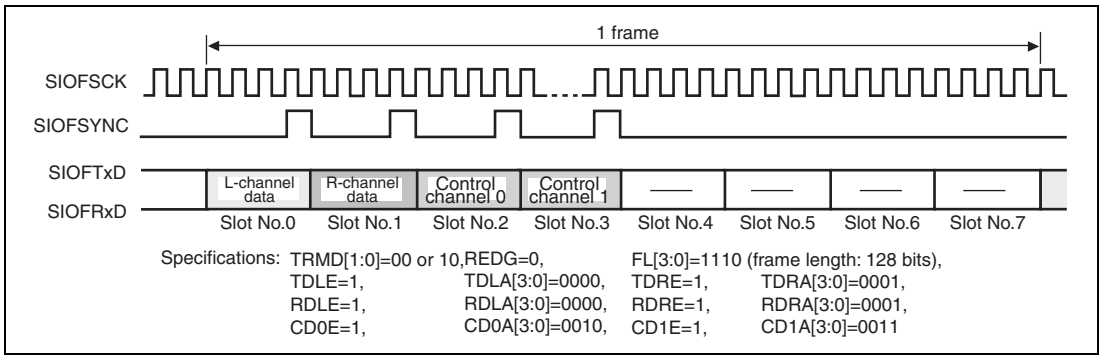


Figure 21.20 Transmit and Receive Timing (16-Bit Stereo Data)

21.5 Usage Notes

21.5.1 Regarding SYNC Signal High Width when Restarting Transmission in Master Mode 2

(1) Problem

If SYNC signal output is enabled (FSE bit = 1), while output of the SYNC signal is disabled by clearing the SICTR.FSE bit in master mode 2 to 0, the High period of the SYNC signal may more quickly become 1 bit long with the rising edge of the SYNC signal in the head frame. However, this period will not be generated after the second frame.

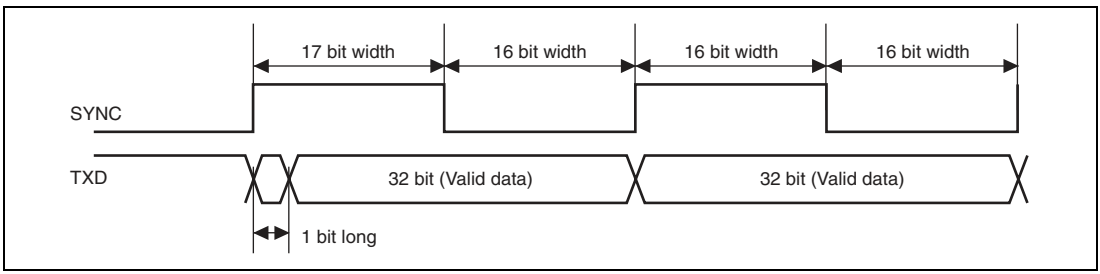


Figure 21.21 Frame Length (32-Bit)

(2) How to Avoid the Problem

To avoid this problem, either counter-measure (a) or (b) is recommended.

(a) When outputting data to the head frame, write dummy data to the transmission FIFO and write valid data after the second frame. The data of the head frame should be read and omitted at the receive side.

(b) Use a configuration that does not occur malfunction, even if the period of the SYNC signal becomes 1 bit longer than that of the value set in the head frame.

Section 22 Analog Front End Interface (AFEIF)

This LSI has an AFE interface that supports softwaremodem. This AFE interface can efficiently execute the modem processing, because it includes 128 stages of FIFO for each of transmission and reception. This AFE interface also includes the interface to data access arrangement (DAA) such as dial pulse generator circuit and ringing detection. Therefore, it is possible to establish a modem system with a minimum of hardware.

22.1 Features

- Serial interface with FIFO
- Clock synchronized serial interface
- Transmit/receive FIFO size is 16 bits (maximum) × 128 words
- Transmit/receive interrupt threshold size is programmable
- Dial pulse generator circuit is included
- Ringing detection (calling signal) function is included

Figure 22.1 shows a block diagram of AFEIF.

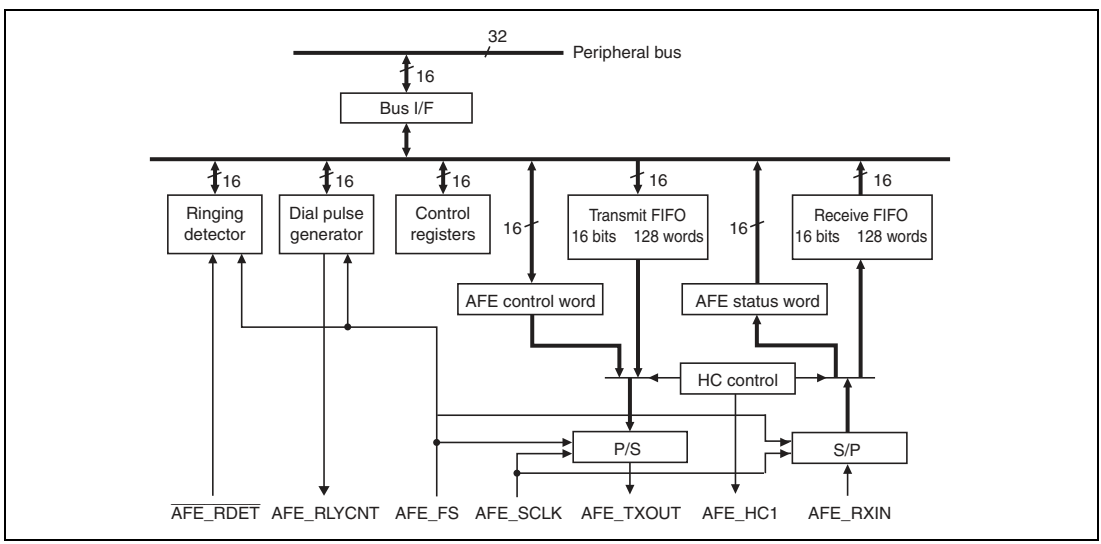


Figure 22.1 Block Diagram of AFE Interface

22.2 Input/Output Pins

Table 22.1 shows the pins for AFE interface.

Table 22.1 Pin Configuration

Pin Name	I/O	Function
$\overline{\text{AFE\_RDET}}$	Input	Ringing signal input
AFE_RLYCNT	Output	On-hook control signal
AFE_SCLK	Input	Shift clock
AFE_FS	Input	Frame synchronization signal
AFE_RXIN	Input	Serial receive data
AFE_HC1	Output	AFE hardware control signal
AFE_TXOUT	Output	Serial transmit data

22.3 Register Configuration

Registers for AFEIF are shown below. Byte access registers to these is inhibited.

- AFEIF control register 1 (ACTR1)
- AFEIF control register 2 (ACTR2)
- AFEIF status register 1 (ASTR1)
- AFEIF status register 2 (ASTR2)
- Make ratio count register (MRCR)
- Minimum pose count register (MPCR)
- Dial number queue (DPNQ)
- Ringing pulse counter (RCNT)
- AFE control data register (ACDR)
- AFE status data register (ASDR)
- Transmit data FIFO port (TDFP)
- Receive data FIFO port (RDFP)

22.3.1 AFEIF Control Register 1 and 2 (ACTR1, ACTR2)

ACTR is the control register for AFEIF and is composed of ACTR1 and ACTR2. ACTR1 is mainly used for FIFO control commands. ACTR2 is used for AFE control commands and DAA control commands.

- ACTR1

Bit	Bit Name	Initial Value	R/W	Description
15	HC	0	R/W	AFE Hardware Control This bit controls AFE. AFE_HC1 signal is made to high directly often the next serial transmit data transfer, when this bit is written to 1. Then ACDR data (AFE control word) is transferred by founding the second AFE.FS. AFEIF module automatically makes AFE_HC1 signal to low and HC bit to 0, directly after transferring the AFE control word. See section 22.4.2, AFE Interface for more detail about AFE control sequences.
14 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	DLB	0	R/W	FIFO Digital Loop Back 0: Normal operation 1: Digital loop back between transmit FIFO and receive FIFO is performed. In this time the transmit data is output to AFE_TXOUT, too.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	FFSZ2	0	R/W	FIFO Interrupt Size Set 2 to 0
3	FFSZ1	0	R/W	Specifies the size of FIFO. FIFO size to generate interrupt (TFE, RFF, THE, and RHF) is assigned as listed in table 22.2.
2	FFSZ0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
1	TE	0	R/W	Transmit Enable 0: Transmit operation is disabled. The READ pointer of FIFO is stacked to the first address. WRITE pointer is reset when 0 is written to this bit. TFEM and THEM bits in ASTR1 is set to 1 at that time. 1: Transmit operation is enabled.
0	RE	0	R/W	Receive Enable 0: Receive operation is disabled. The READ /WRITE pointer is fixed to the first address. Bits RFFM and RHFM in ASTR1 are set to 1 at that time. 1: Receive operation is enabled

Table 22.2 FIFO Interrupt Size

Bit 4: FFSZ2	Bit 3: FFSZ1	Bit 2: FFSZ0	Description			
			FIFO Size	TFE/RFF	THE/RHF	
0	0	0	128	128 empty/full	64 empty/full	(Initial value)
		1	64	64 empty/full	32 empty/full	
	1	0	32	32 empty/full	16 empty/full	
		1	16	16 empty/full	8 empty/full	
1	0	0	8	8 empty/full	4 empty/full	
		1	4	4 empty/full	2 empty/full	
	1	0	2	2 empty/full	1 empty/full	
		1	96	96 empty/full	48 empty/full	

- ACTR2

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	DPST	0	R/W	Dial Pulse Start Start bit of dial pulse. Dial number within the DPNQ register is output to AFE_RLYCNT as specified by PPS, MRCR and MPCR. After all dial number is output, DPE interrupt is generated to modify the DPST bit to 0. See section 22.4.3, DAA Interface for more detail about dial pulse output sequence. Take care that AFE_RLYCNT must be "H" to enable dial pulse generating circuit
3	PPS	0	R/W	Dial Pulse Duration Set 0: 10 PPS 1: 20 PPS
2	RCEN	0	R/W	Ringing Counter Enable 0: Stop Ringing Counter 1: Start Ringing Counter Note: See section 22.4.3, DAA Interface for more detail about how to count.
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	RLYC	0	R/W	Relay Control The signal controls Hook Relay. 0: On hook state. AFE_RLYCNT goes Low Level. 1: Off hook state. AFE_RLYCNT goes High Level.

22.3.2 Make Ratio Count Register (MRCR)

MRCR is the counter that specifies make ratio of dial pulse. Make interval is specified with AFE_FS as base clock of 9,600 Hz.

Pulse signal is not output when an invalid data (a data that is greater than 1E0H in case of PPS = 1 (20 pps), or a data that is greater than 3C0H in case of PPS = 0 (10 pps)) was input.

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	MRCR9 to MRCR0	0	R/W	Specifies make ratio of dial pulse.

22.3.3 Minimum Pause Count Register (MPCR)

MPCR is a counter that sets the dial number interval of the dial pulse. The interval is specified with AFE_FS as base clock of 9600 Hz.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	MPCR15 to MPCR0	0	R/W	Sets the dial number interval of the dial pulse.

22.3.4 AFEIF Status Register 1 and 2 (ASTR1, ASTR2)

ASTR is the control register for AFEIF, and composed of ASTR1 and ASTR2. ASTR1 is mainly used for transmit/receive FIFO interrupt control commands. ASTR2 is used for DAA interrupt control commands. See section 22.4.1, Interrupt Timing for more detail about interrupt handling.

(1) AFEIF Status Register 1 (ASTR1)

ASTR1 is composed by interrupt status flags (4 bits) relating transmit/receive FIFO and mask flags (4 bits) for transmit/receive FIFO interrupt signal. Status flag displays full/empty interrupt status of transmit/receive FIFO and half size interrupt status for FIFO. FIFO empty (TFE) and FIFO half size interrupt (THE) shows "1" as initial value, because transmit FIFO is empty after power on reset. These interrupt flags are to be cleared with the data write / read action to FIFO from CPU.

Each interrupt mask flag is able to prohibit interrupt generation of each interrupt that indicated in interrupt status flag. Every mask bits are automatically set when TE or RE bit are modified to 1. TFEM and THEM are 1 when TE = 0. RFFM and RHFM are 1 when RE = 0. Each mask bit is reset as 1.

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	TFEM	1	R/W	Transmit FIFO Empty Interrupt Mask 0: TFE Interrupt enable 1: TFE interrupt masked
10	RFFM	1	R/W	Receive FIFO Full Interrupt Mask 0: RFF Interrupt enable 1: RFF Interrupt masked
9	THEM	1	R/W	Threshold of Transmit FIFO Empty Interrupt Mask 0: THE Interrupt enable 1: THE Interrupt masked
8	RHFM	1	R/W	Threshold of Receive FIFO Full Interrupt Mask 0: RHF Interrupt enable 1: RHF Interrupt masked
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	TFE	1	R	Transmit FIFO Empty Interrupt 0: Normal state [Clearing condition] • Data are written into FIFO 1: TxFIFO empty interrupt [Setting conditions] • Reset • No effective data in area of FIFO • TE bit (ACTR1) is set to 0 (TFEM bit is set to 1)

Bit	Bit Name	Initial Value	R/W	Description
2	RFF	0	R	Receive FIFO Full Interrupt 0: Normal state [Clearing conditions] • Reset • Number of data in FIFO becomes smaller than the size that is indicated with FFSZ (ACTR1) • RE bit (ACTR1) is set to 0 1: Rx FIFO full interrupt [Setting condition] • Specified size with FFSZ (ACTR1) of receive data is accumulated into FIFO
1	THE	1	R	Transmit FIFO Half Size Empty 0: Normal state [Clearing condition] • Number of valid data in FIFO becomes greater than the half of the size that is indicated by FFSZ 1: Tx FIFO Half Size Interrupt [Setting conditions] • Reset • Number of valid data in FIFO becomes smaller than the half of the size that is indicated with FFSZ • TE bit (ACTR1) is set to 0 (THEM bit is set to 1)

Bit	Bit Name	Initial Value	R/W	Description
0	RHF	0	R	Receive FIFO Half Size Full 0: Normal state [Clearing conditions] • Reset • Number of data in FIFO becomes smaller than the half of the size that is indicated by FFSZ • RE bit (ACTR1) is set to 0 1: Rx FIFO half size interrupt [Setting condition] • The half of specified size with FFSZ (ACTR1) of receive data is accumulated into FIFO

(2) AFEIF Status Register 2 (ASTR2)

ASTR2 is the register that is composed of interrupt status flag (2 bits) relating DAA control and mask flag (2 bits) of interrupt signals for DAA control. Status flags shows statuses of ringing detect interrupt, end of dial pulse output interrupt. Interrupt flags are cleared by 0 write after read action of this register. Each Interrupt signal can be masked by each interrupt masks.

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	DPEM	1	R/W	Dial Pulse End Interrupt Mask 0: Interrupt enable 1: Interrupt mask
8	RDETM	1	R/W	Ringing Detect Mask 0: Ringing interrupt enable 1: Ringing interrupt mask
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	DPE	0	R/W	Dial Pulse End 0: Normal state [Clearing conditions] • Reset • Interrupt status 1 is read and then 0 is written to this bit 1: Dial pulse end interrupt [Setting conditions] • Output of all of dial pulse sequences completed or end command 0H detected • Illegal end (unspecified dial number and DPST set when RLYC bit (ACTR2) is low level)
0	RDEF	0	R/W	Ringing Detect 0: Normal state [Clearing conditions] • Reset • Interrupt status 1 is read and then 0 is written to this bit 1: Ringing waveform detect [Setting condition] • Ringing waveform is input to $\overline{\text{AFE\_RDET}}$ pin (Latched at rising edge)

22.3.5 Dial Pulse Number Queue (DPNQ)

This is the dial pulse number queue up to 4 digits which has 4-bits registers. This queue generates dial pulse according to the following table in the order of dial pulse number.

A dial-pulse-end interrupt is sent out after DN3 is output or if 0H or a value other than the corresponding data is detected.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	DN03 to DN00	All 0	R/W	DN0
11 to 8	DN13 to DN10	All 0	R/W	DN1
7 to 4	DN23 to DN20	All 0	R/W	DN2
3 to 0	DN33 to DN30	All 0	R/W	DN3

Table 22.3 Telephone Number and Data

TEL No.	Corresponding Data
0	AH
1	1H
2	2H
3	3H
4	4H
5	5H
6	6H
7	7H
8	8H
9	9H
Pause	FH
End	0H

22.3.6 Ringing Pulse Counter (RCNT)

The result of counting 1 cycle of ringing waveform with AFE_FS is shown here.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	RCNT15 to RCNT0	All 0	R/W	Ringing Counter Value The result of counting 1 cycle of input ringing waveform with AFE_FS (output of AFE). See section 22.4.3, DAA Interface for more detail about the ringing detect sequence.

22.3.7 AFE Control Data Register (ACDR)

ACDR is the register to store the AFE control word. After 1 is written to HC bit (ACTR1), data is transferred to AFE at the timing of 3rd FS.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	ACDR15 to ACDR0	All 0	R/W	Store the AFE control word.

22.3.8 AFE Status Data Register (ASDR)

ASDR is the register to store the AFE status word. After 1 is written to HC bit (ACTR2), data is transferred to ASDR from AFE at the timing of 3rd FS.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	ASDR15 to ASDR0	All 0	R	Store the AFE control word.

22.3.9 Transmit Data FIFO Port (TDFP)

TDFP is the write only port for transmit FIFO. Transmit FIFO has 128 stages (maximum), and can generate interrupt of the data empty as well as of the threshold size specified by FFSZ (ACTR1). Directly after the reset and when TE (ACTR1) bit is 0, the pointer of FIFO is set to the first address and data becomes empty. The interrupt will occur when the TE bit (ACTR1) is written to 1 at that state. In normal case, TE bit should be changed after writing data into transmit FIFO.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TDFP15 to TDFP0	All 0	W	Write only port for transmit FIFO.

22.3.10 Receive Data FIFO Port (RDFP)

RDFP is the read only register for receive FIFO. Receive FIFO has 128 stages (maximum), and can generate interrupt of the data full as well as of the threshold size specified by FFSZ (ACTR1). Directly after the reset and when RE bit (ACTR1) is 0, the pointer of FIFO is fixed at the first address and data from RDFP becomes undetermined.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	RDFP15 to RDFP0	Undefined	R	Read only register for receive FIFO.

22.4 Operation

22.4.1 Interrupt Timing

AFE interface module generates 3 types of interrupt: FIFO data transfer, ringing detect, and dial pulse transmit end. The timing of each interruption is described below.

(1) FIFO Interrupt Timing

Figure 22.2 shows interrupt timing of data transfer FIFO. Transmit FIFO generates the TFE and THE interrupts after the last data is transfer red shift register. Receive FIFO generates the RFF and RHF interrupt after the last data or specified word is transferred from shift register to FIFO.

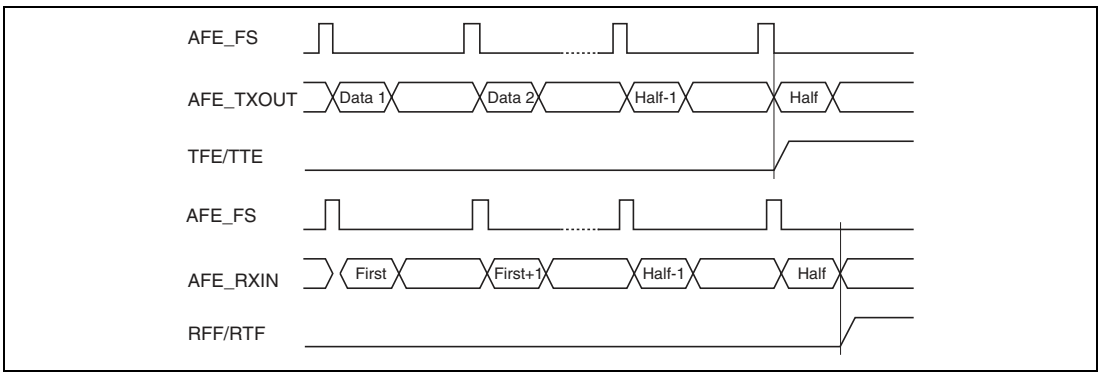


Figure 22.2 FIFO Interrupt Timing

(2) Ringing Interrupt Timing

As the figure 22.3 shows, the ringing signal from the line is transformed to rectangular wave and then input to AFEIF. The interrupt is generated at the falling edge of input wave in AFEIF module.

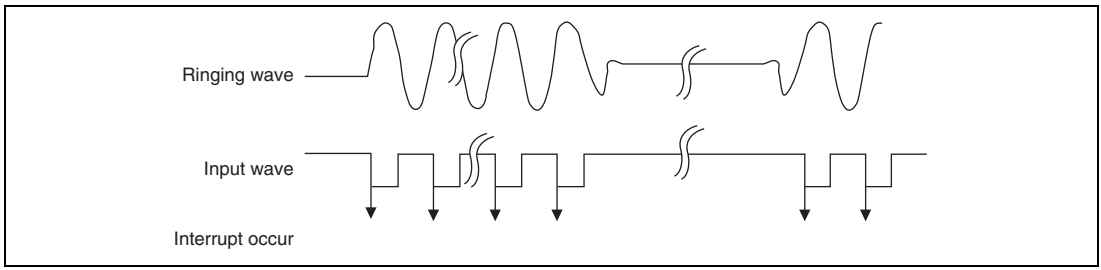


Figure 22.3 Ringing Interrupt Occurrence Timing

(3) Dial Pulse Interrupt Timing

Dial pulse interrupt is generated in the dial pulse transmit sequence when AFEIF reads 0H (end) data from DPNQ register or all of 4 digits are output. Refer to section 22.4.3, DAA Interface about dial pulse sequence.

(4) Interrupt Generator Circuit

Interrupt is generated as is shown in figure 22.4. That is, AFEIFI signal is generated by performing OR operation on the four signals from ASTR1 in FIFO interrupt control and the two signals from ASTR2 in DAA interrupt control, and then sent out to INTC as one interrupt signal (AFECI).

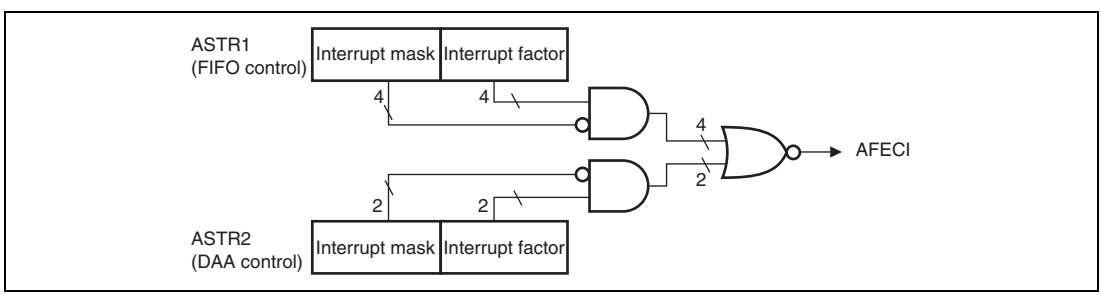


Figure 22.4 Interrupt Generator

22.4.2 AFE Interface

(1) Serial Data Transfer Specification

The specification for serial data transfer is base on that of STLC7550, which is an AFE manufactured by ST microelectronics. STLC7550 has a self-oscillation mode, and flame synchronous signal AFE_FS used for serial transfer and serial bit clock AFE_SCLK are supplied by AFE. Figure 22.5 shows the serial transfer interface. After outputting the valid data, AFE_TXOUT holds the value of LSB.

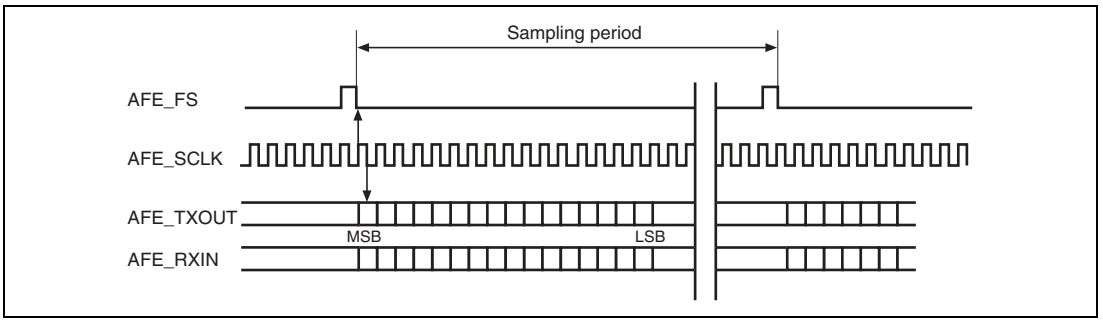
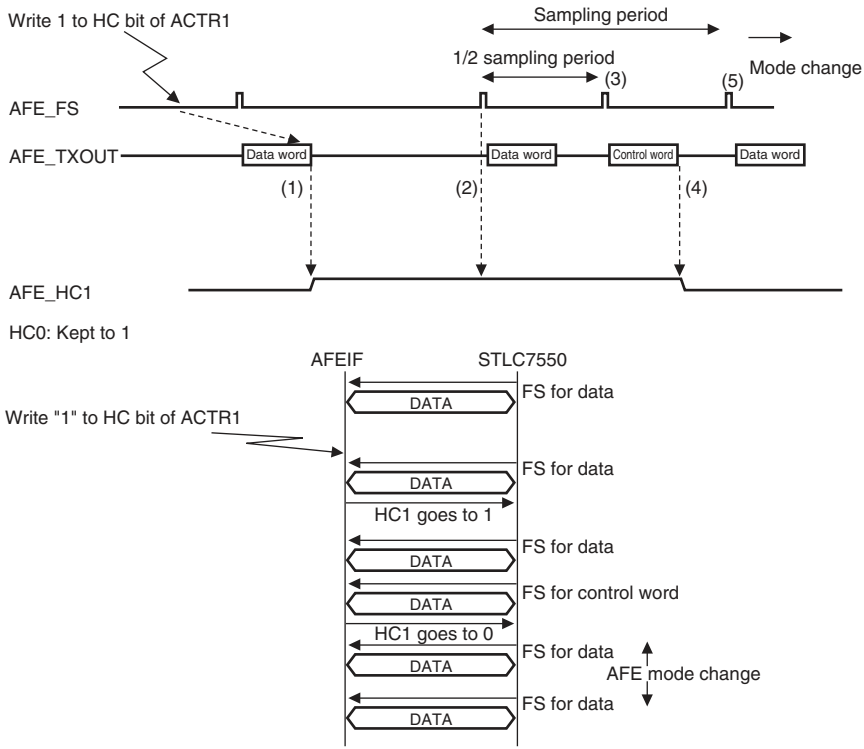


Figure 22.5 AFE Serial Interface

(2) HC Control Sequence

AFEIF module supports hardware control STLC7550 that is an AFE manufactured by ST microelectronics. Figure 22.6 shows the AFE control sequence.



1. If the CPU write "1" to the HC bit of ACTR1, the AEFIF drives AFE_HC1 to "H" right after transmit next data.
2. AFE fetches the HC1's status of "H" at rising edgge of next AFE_FS.
3. AFE output the FS at the next 1/2 sampling period and then AFEIF transfers the control word in synchronization with AFE_FS.
4. AFEIF keeps AFE_HC1 to "H" for 2nd AFE_FS and return to "L" after transmit the control word.
5. AFE fetches the AFE_HC1's status of "L" and changes the mode of itself.

Figure 22.6 AFE Control Sequence

22.4.3 DAA Interface

Figure 22.7 shows the blocks diagram of DAA circuit. Ringing detect and dial pulse sending sequence are described below.

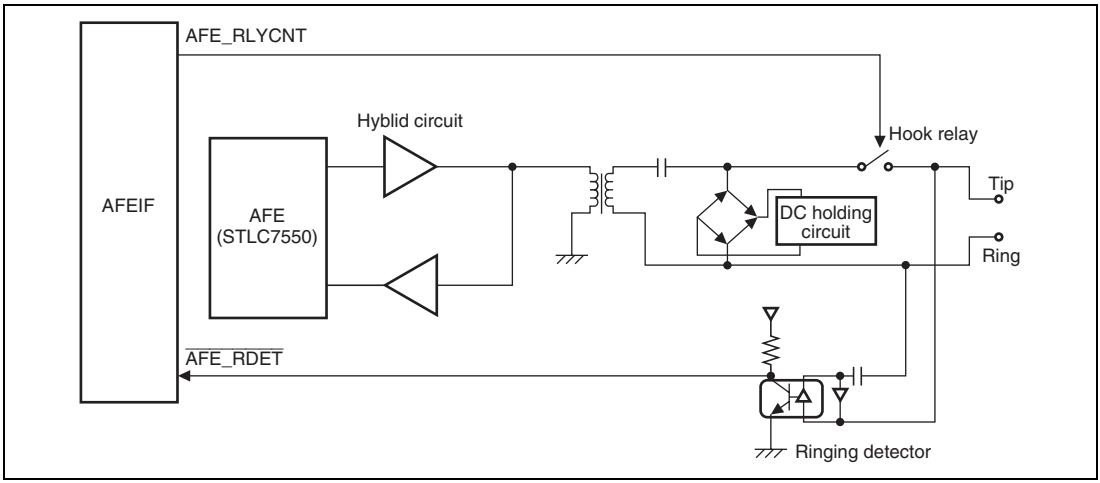


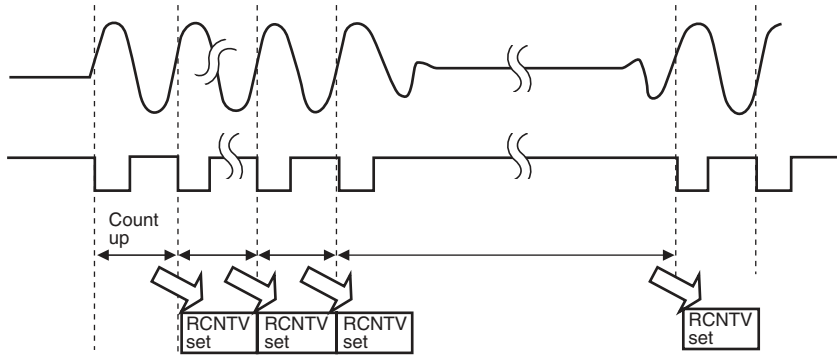
Figure 22.7 DAA Block Diagram

(1) Ringing Detect Sequence

After the first ringing interrupt occurs, counting starts with writing 1 into RCEN bit of CTR2. AFE must be operating before counting, because periodic counter counts AFE_FS from falling edge to next falling edge.

The value of RCNTV register is effective only after 2nd interrupt generation, because the value of RCNTV register is transferred from counter with a trigger of ending of 1st period cycle.

RCNTV will be 258 H (600 in decimal) if ringing cycle is 16 Hz and counted by 9600 Hz which is default value of AFE_FS. Figure 22.8 shows detecting sequence of ringing.



1. First INT occur. RCEN (ACTR2) turns on.
2. From 2nd INT, read the RCNTV.
3. After acknowledge the ringing, RCEN (ACTR) turns off and goes to off hook operation.

Figure 22.8 Ringing Detect Sequence

(2) Dial Pulse Sending Sequence

A dial pulse is generated according to the conditions that are specified in ACTR2, and is sent out to AFE_RLYCNT.

As the basic clock for generating the dial pulse is AFE_FS that is input from AFE, it is necessary to make AFE in operating state. An example of control sequence for dial pulse sending is shown below.

Note that this sequence cannot be operated when RLYC bit (ACTR2) is low.

[Conditions]
Make ratio: 33%
Pulse interval: 20 PPS
Minimum pause: 600 ms
Dial number: 0,1234567 ("," means pause)

[Control sequence]
1. Set PPS (ACTR2) → "1", MKR → "9EH1", MNRPCNT → "1680H"
2. Set DPNQ → "AF12H".
3. Set RLYC → "H". (Off Hook)
4. Detect dial tone or wait specific period. (Controlled by software)
5. Write "1" to DPST (ACTR2). (Start sending dial pulse)

6. After 4 digits of dial pulses are sent, interrupt is generated. (DPST is reset to "0")
7. Set DPNQ1 → "3456H".
8. Write "1" to DPST (ACTR2).
9. After 4 digits of dial pulses are sent, interrupt is generated. (DPST is reset to "0")
10. Set DPNQ2 → "70XXH".
11. Write "1" to DPST (ACTR2).
12. After 1 digit of dial pulse is sent, interrupt is generated. (DPST is reset to "0", and finish sending)

22.4.4 Wake up Ringing Interrupt

System wake up function by the ringing signal from telephone line is realized by inputting $\overline{\text{AFE\_RDET}}$ signal, which is an input signal for ringing, to PINT pin.

Section 23 USB Pin Multiplex Controller

23.1 Features

The USB multiplex controller controls the data path to USB transceiver from USB host controller port 1 or USB function controller.

Both USB host port 1 and USB function controller are connected to USB transceiver 1 via multiplexer that is controlled by UTRCTL register. The USB host controller port 2 and USB transceiver 2 are connected one-to-one. USB transceiver 1 can be connected to USB host controller or USB function controller, while USB transceiver 2 can only be connected to the USB host controller. Because these ports and transceivers are controlled individually, USB transceiver 2 can be connected to either the USB host controller or the USB function controller regardless its status. The signals to USB transceiver are used as external pins USB1d _**** which are multiplexed with pins 113 to 123.

Figure 23.1 shows the connections between the on-chip USB host controller of this LSI, the USB function controller, and the on-chip 2-port USB transceiver.

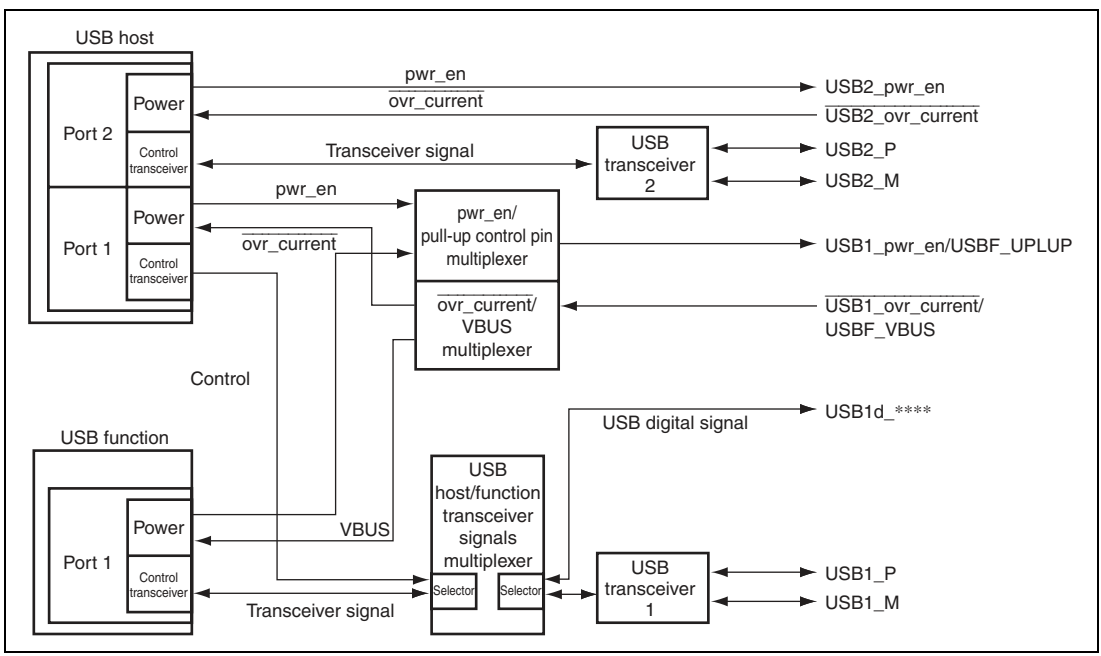


Figure 23.1 Block Diagram of USB PIN Multiplexer

23.2 Input/Output Pins

USB pin multiplexer controller has pins that are shown in tables 23.1, 23.2, and 23.3

Table 23.1 Pin Configuration (Digital Transceiver Signal)

Name	Pin Name	I/O	Description
RCV pin	USB1d_RCV	Input	Input pin for receive data from differential receiver
DPLS pin	USB1d_DPLS	Input	Input pin for D+ signal from receiver
DMNS pin	USB1d_DMNS	Input	Input pin for D– signal from receiver
TXDPLS pin	USB1d_TXDPLS	Output	D+ transmit output pin
TXENL pin	USB1d_TXENL	Output	Driver output enable pin
SUSPEND pin	USB1d_SUSPEND	Output	Transceiver suspend state output pin
SPEED pin	USB1d_SPEED	Output	Transceiver speed control pin
TXSE0 pin	USB1d_TXSE0	Output	SE0 state output pin

Note: The pins shown in table 23.1 are used for connecting an external USB transceiver, and cannot be used when the on-chip USB transceiver is connected.

Table 23.2 Pin Configuration (Analog Transceiver Signal)

Name	Pin Name	I/O	Description
1P pin	USB1_P	I/O	D+ port1 transceiver pin
1M pin	USB1_M	I/O	D– port1 transceiver pin
2P pin	USB2_P	I/O	D+ port2 transceiver pin
2M pin	USB2_M	I/O	D– port2 transceiver pin

Note: The pins shown in table 23.2 can be used as two ports USB host controller pins, or one port USB host controller pins and one port USB function controller pins. Make these pins open, when they are not used.

Table 23.3 Pin Configuration (Power Control Signal)

Name	Pin Name	I/O	Description
USB1 power enable/pull-up control pin	USB1_pwr_en/ USBF_UPLUP	Output	USB port 1 power enable control*/ pull- up control output
USB2 power enable pin	USB2_pwr_en	Output	USB port 2 power enable control
USB1 over current /monitor pin	$\overline{\text{USB1\_ovr\_current}}$/ USBF_VBUS	Input	USB port 1 over-current detect/ USB cable connection monitor pin*
USB2 over current pin	$\overline{\text{USB2\_ovr\_current}}$	Input	USB port 2 over-current detect

Note: The pins shown in table 23.3 can be used for power control of USB. Pins for port 1 (pins with *) have the functions that are multiplexed functions of USB controller and USB function controller.

Table 23.4 Pin Configuration (Clock Signal)

Name	Pin Name	I/O	Description
USB external clock	EXTAL_USB	Input	Connects a crystal resonator for USB. Also used to input an external clock for USB (48 MHz input).
USB crystal	XTAL_USB	Output	Connects a USB crystal resonator for USB.

23.3 Register Descriptions

The USB pin multiplexer controller has the following register.

- USB transceiver control register (UTRCTL)

23.3.1 USB Transceiver Control Register (UTRCTL)

UTRCTL controls the selection of transceiver function and signal source related to the USB port 1.

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R/W	Reserved These bits are always read as all 0s. The write values should always be all 0s.
8	DRV	0	R/W	See section 34, Pin Function Controller (PFC).
7 to 2	—	All 0	R/W	Reserved These bits are always read as all 0s. The write values should always be all 0s.
1	USB_TRANS	0	W	USB Port 1 Transceiver Select 0: USB transceiver is enabled 1: USB digital signals output is enabled
0	USB_SEL	1	W	USB Port 1 Signal Source Select 0: Port 1 of USB host controller is used 1: Port 1 of USB function controller is used

23.4 Examples of External Circuit

23.4.1 Example of the Connection between USB Function Controller and Transceiver

Figures 23.2 and 23.3 show example connections of USB function controller and transceiver. Figures 23.2 shows connections when using the on-chip USB transceiver. Figures 23.3 shows connections when not using the on-chip USB transceiver. When using the USB function controller, the signals must be input to the cable connection monitor pin UJBF_VBUS. The USBF_VBUS pin is multiplexed with the $\overline{\text{USB1\_ovr\_current}}$ pin, and writing 1 to bit 0 (USB_SEL) of UTRCTL selects the USBF_VBUS pin functions. According to the status of the USBF_VBUS pin, the USB function controller recognizes whether the cable is connected/disconnected. Also, pin D+ must be pulled up in order to notify the USB host/hub that the connection is established. The sample circuits in figures 23.2 and 23.3 use the USB1_pwr_en pin for pull-up control.

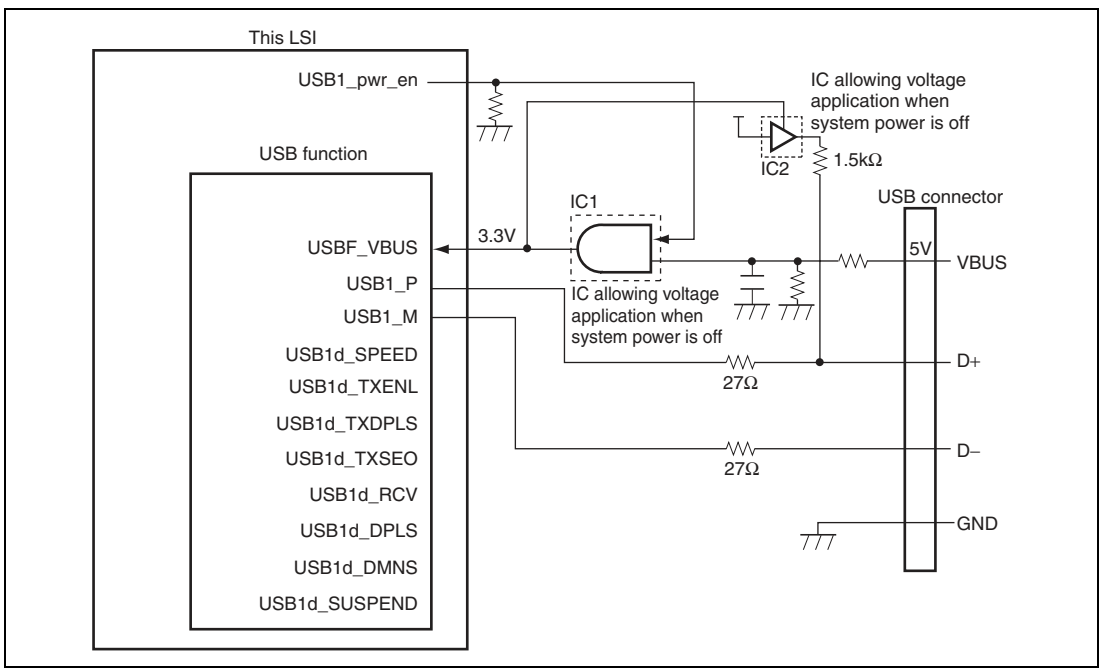


Figure 23.2 Example 1 of Transceiver Connection for USB Function Controller (On-Chip Transceiver is Used)

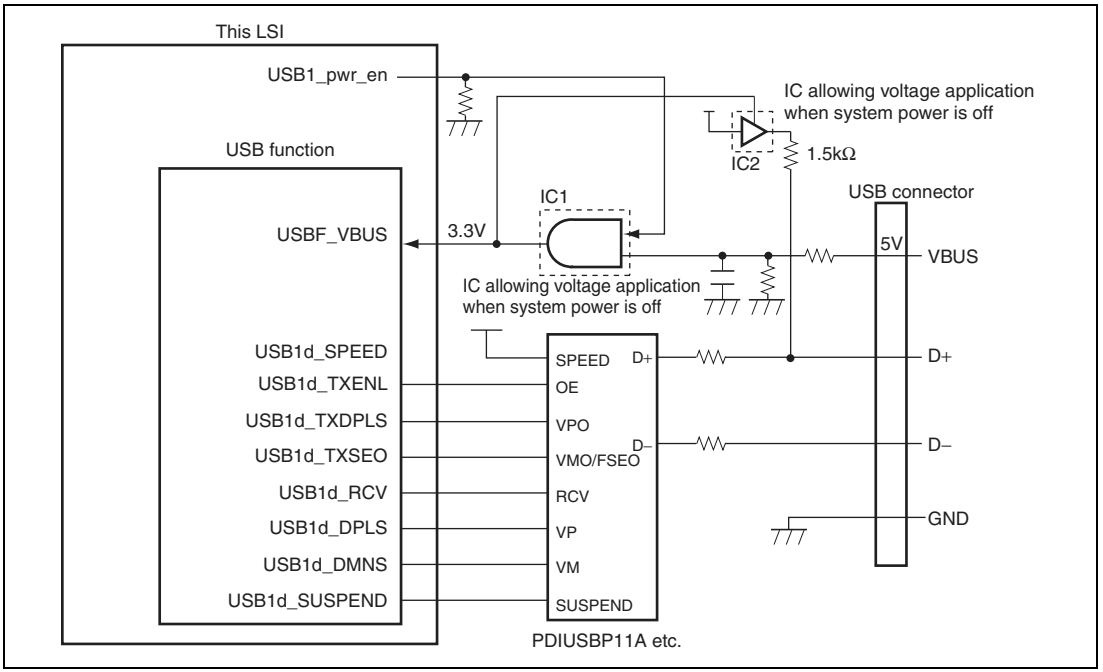


Figure 23.3 Example 2 of Transceiver Connection for USB function Controller (On-Chip Transceiver is not Used)

- D+ Pull-up Control

  Control D+ pull-up by using USB1_pwr_en pin in the system when the connection—notification (D+ pull-up) to USB host or hub is wished to be inhibited (i.e., during high-priority processing or initialization processing).

  The D+ pull-up control signal and USBF_VBUS pin input signal should be controlled by using the USB1_pwr_en pin and the USB cable VBUS (AND circuit) as is shown in examples of figures 23.2 and 23.3

  D+ pull-up is inhibited when the USB1_pwr_en pin is low in examples of figures 23.3 and 23.5. Use an IC such that allows voltage application when system power is off (for example, HD74LV1G126A) for the pull-up control IC (IC2 in figures 23.2 to 23.5).

  (The UDC core in this LSI holds the powered state when USBF_VBUS pin is low, regardless of the D+/D− state.)

- Detection of USB Cable Connection/Disconnection

  As USB function controller in this LSI manages the state by hardware, USB_VBUS signal is necessary to recognize connection or disconnection of the USB cable. The power supply signal (VBUS) in the USB cable is used for USBF_VBUS. However, if the cable is connected to the USB host or hub when the power of USB function controller (this LSI—installed system) is off, a voltage of 5 V will be applied from the USB host or hub.

  Therefore, use an IC such that allows voltage application when system power is off (for example, HD74LV1G08A) for the IC1 in figures 23.2 to 23.5.

  To recover from the standby state with the USB cable connected, the IRQ pin should be connected to the USB cable. (Recovery from the software standby state cannot be performed by a USB connection/disconnection interrupt.)

23.4.2 Example of the Connection between USB Host Controller and Transceiver

Figures 23.4 and 23.5 show example connections of the USB host controller and transceiver. Figure 23.4 shows an example connection using the built-in transceiver 1. By using the $\overline{\text{USB2\_ovr\_current}}$, USB2_pwr_en, USB2_P, and USB2_M pins in an external circuit similar to that in figure 23.4, you can also use built-in USB transceiver 2. Figure 23.5 shows an example connection when not using the built-in USB transceiver. When using the USB host controller, a separate LSI must be used for USB power bus control (equivalent to the USB power control LSIs in figures 23.4 and 23.5). Make sure the LSI has the power supply capacity to satisfy the USB standard, and select one that has an overcurrent protection function. Configure the system so that the input to the $\overline{\text{USB1\_ovr\_current}}$ pin is Low on detection of an overcurrent.

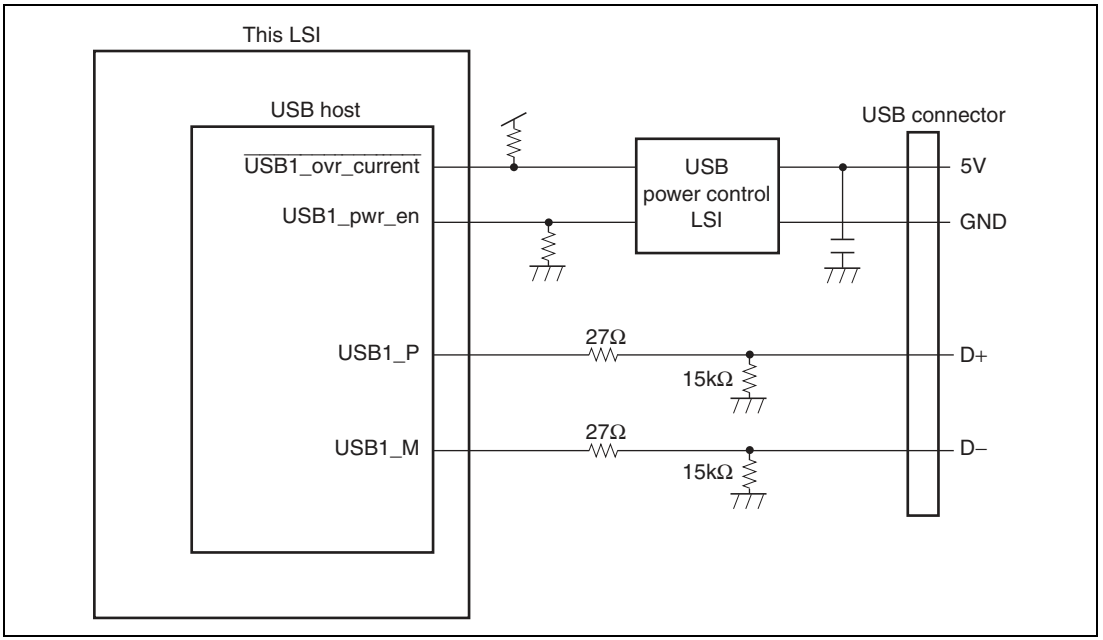


Figure 23.4 Example 1 of Transceiver Connection for USB Host Controller (On-Chip Transceiver is Used)

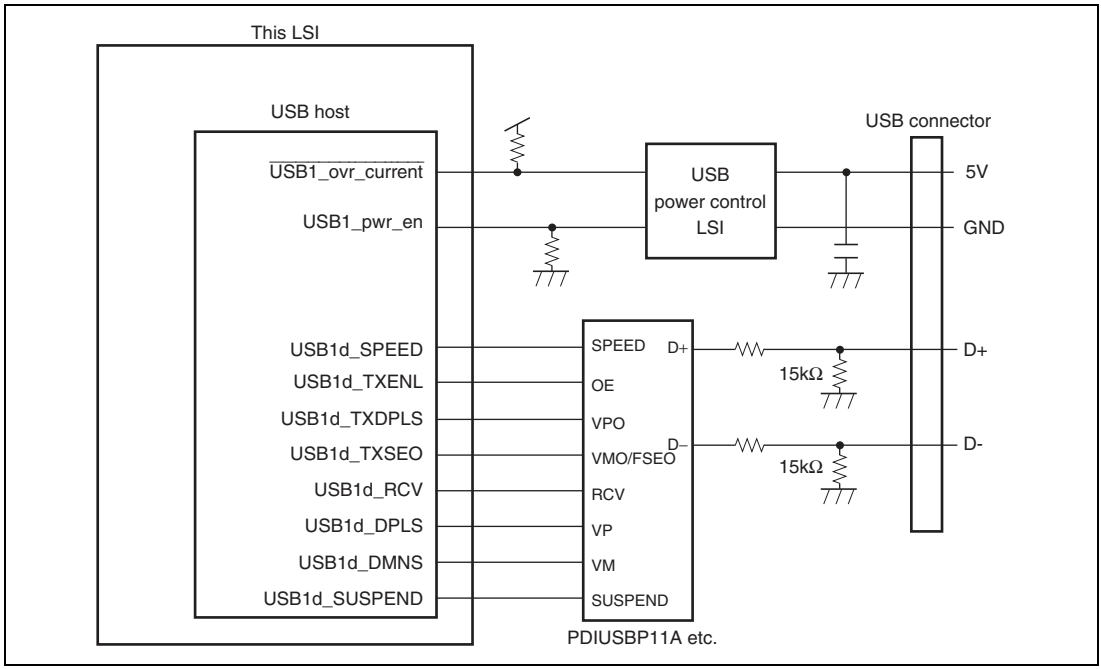


Figure 23.5 Example 2 of Transceiver Connection for USB Host Controller (On-Chip Transceiver is not Used)

23.5 Usage Notes

23.5.1 About the USB Transceiver

USB transceiver is included in this LSI. It is also possible to connect an external transceiver according to the setting in EXPFC register (see figures 23.3 and 23.5). In this case, ask the manufacturer of the transceiver about the recommended circuit that is used between the USB transceiver and USB connectors.

23.5.2 About the Examples of External Circuit

These examples of transceiver connection in this chapter are for reference only, therefore proper operation is not guaranteed with these circuit examples. If system countermeasures are required for external surges and ESD noise, use a protective diode, etc.

Section 24 USB Host Controller (USBH)

The USB Host Controller module incorporated in this LSI supports Open Host Controller Interface (Open HCI) Specification for USB as well as the Universal Serial Bus specification ver.1.1.

The Open HCI Specification for the USB is a register-level description of Host Controller for the USB, which in turn is described by the USB specification.

It is necessary to refer Open HCI specification to develop drivers for this USB Host Controller and hardware.

24.1 Features

- Support open HCI standard ver.1.0 register set
- Support Universal Serial Bus standard ver.1.1
- Root Hub function
- Support Full speed (12 Mbps) mode and Low speed (1.5 Mbps) mode
- Support Overcurrent detection
- Support 127 endpoints control in maximum
- Possible to use only the SDRAM area of area 3 as transmit data and descriptor.

24.2 Input/Output Pins

Pin configuration of the USB Host Controller is shown in table 24.1.

For the detailed method for setting each pin, see section 23, USB Pin Multiplex Controller.

Table 24.1 Pin Configuration

Pin Name	Pin Name	I/O	Function
USB1 power enable/pull-up control pin	USB1_pwr_en	Output	USB port 1 power enable control
USB2 power enable pin	USB2_pwr_en	Output	USB port 2 power enable control
USB1 overcurrent/monitor pin	$\overline{\text{USB1\_ovr\_current}}$/ USBF_VBUS	Input	USB port 1 over-current detect/ USB cable connection monitor pin
USB2 overcurrent pin	$\overline{\text{USB2\_ovr\_current}}$	Input	USB port 2 over-current detect
1P pin	USB1_P	I/O	D+ port 1 transceiver pin
1M pin	USB1_M	I/O	D− port 1 transceiver pin
2P pin	USB2_P	I/O	D+ port 2 transceiver pin
2M pin	USB2_M	I/O	D− port 2 transceiver pin
SPEED pin	USB1d_SPEED	Output	Transceiver speed control pin
USB external clock	EXTAL_USB	Input	Connect a crystal resonator for USB. Alternatively, an external clock may be input for USB (48 MHz).
USB crystal	XTAL_USB	Output	Connect a crystal resonator for USB.

24.3 Register Descriptions

The USBH has the following registers.

- Hc Revision register (USBHR)
- Hc Control register (USBHC)
- Hc Command Status register (USBHCS)
- Hc Interrupt Status register (USBHIS)
- Hc Interrupt Enable register (USBHIE)
- Hc Interrupt Disable register (USBHID)
- Hc HCCA register (USBHHCCA)
- Hc Period Current ED register (USBHPCED)
- Hc Control Head ED register (USBHCHED)
- Hc Control Current ED register (USBHCCED)
- Hc Bulk Head ED register (USBHBHED)
- Hc Bulk Current ED register (USBHBCED)
- Hc Done Head ED register (USBHDHED)
- Hc Fm Interval register (USBHFI)
- Hc Fm Remaining register (USBHFR)
- Hc Fm Number register (USBHFN)
- Hc Periodic Start register (USBHPS)
- Hc LS Threshold register (USBHLST)
- Hc Rh Descriptor A register (USBHRDA)
- Hc Rh Descriptor B register (USBHRDB)
- Hc Rh Status register (USBHRS)
- Hc Rh Port Status 1 register (USBHRPS1)
- Hc Rh Port Status 2 register (USBHRPS2)

24.3.1 Hc Revision Register (USBHR)

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	Rev7	0	R	Revision
6	Rev6	0	R	These read only bits include the BCD expression of the HCI specification version implemented for the host controller. The value H'10 corresponds to version 1.0. All HCI implementation complying with this specification have the value of H'10.
5	Rev5	0	R	
4	Rev4	1	R	
3	Rev3	0	R	
2	Rev2	0	R	
1	Rev1	0	R	
0	Rev0	0	R	

24.3.2 Hc Control Register (USBHC)

The Hc Control register defines the operation mode for the host controller. The bits of this register are amended only by the host controller driver (HCD) other than HCFS and RWC.

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	RWE	0	R/W	Remote Wakeup Enable This bit is set by HCD to enable/disable the remote wakeup function at the same time as the detection of an upstream resume signal. This function is not supported. Be sure to write 0.

Bit	Bit Name	Initial Value	R/W	Description
9	RWC	0	R/W	Remote Wakeup Connected This bit indicates whether the host controller supports a remote wakeup signal or not. When the remote wakeup is supported and used in the system, the host controller must set this bit between POST in the system firmware. The host controller clears the bit at the same time of the hardware reset, however, does not change at the same time as the software reset. This function is not supported. Be sure to write 0.
8	IR	0	R/W	Interrupt Routing This bit determines the routing of interrupts generated by the event registered in USBHIS. HCD clears this bit at the same time as the hardware reset, however, does not clear at the same time as the software reset. HCD uses this bit as a tag to indicate the ownership of the host controller. 0: All interrupts are routed to normal host bus interrupt mechanism 1: Interrupts are routed to SMI
7	HCFS1	0	R/W	Host Controller Functional State
6	HCFS0	0	R/W	HCD determines whether the host controller has started to route SOF after having read the SF bit of USBHIS. This bit can be changed by the host controller only in the UsbSuspend state. The host controller can move from the UsbSuspend state to the UsbResume state after having detected the resume signal from the downstream port. In the host controller, UsbSuspend is entered after the software reset so that UsbReset is entered after the hardware reset. The former resets the route hub. 00: USB Reset 01: USB Resume 10: USB Operational 11: USB Suspend

Bit	Bit Name	Initial Value	R/W	Description
5	BLE	0	R/W	Bulk List Enable This bit is set to enable the processing of the bulk list in the next frame. When this bit is cleared by HCD, the processing of the bulk list is not carried out after next SOF. The host controller checks this bit when processing this list. When disabling, HCD can correct the list. When USBHBCED indicates ED to be deleted, HCD should hasten the pointer by updating USBHBCED before re-enabling the list processing. 0: Bulk list processing is not carried out 1: Bulk list processing is carried out
4	CLE	0	R/W	Control List Enable This bit is set to enable the processing of the control list in the next frame. If cleared by HCD, the processing of the control list is not carried out after next SOF. The host controller must check this bit whenever the list will be processed. When disabling, HCD can correct the list. When USBHCCED indicates ED to be deleted, HCD should hasten the pointer by updating USBHCCED before re-enabling the list processing. 0: Control list processing is not carried out 1: Control list processing is carried out
3	IE	0	R/W	Isochronous Enable This bit is used by HCD to enable/disable the processing of isochronous ED. While processing the periodic list, HC will check the status of this bit when it finds an isochronous ED (F =1). If set (enabled), the host controller continues to process ED. If cleared (disabled), the host controller stops the processing of the periodic list (currently includes only isochronous ED) and starts to process the bulk/control list. Setting this bit is guaranteed to be valid in the next frame (not in the current frame). 0: Processes isochronous ED 1: Processes the bulk/control list

Bit	Bit Name	Initial Value	R/W	Description
2	PLE	0	R/W	Periodic List Enable This bit is set to enable the processing of the periodic list. If cleared by HCD, no periodic list processing is carried out after next SOF. HC must check this bit before HC starts to process the list. 0: The periodic list processing is not carried out after next SOF 1: The periodic list processing is carried out after next SOF
1	CBSR1	0	R/W	Control Bulk Service Ratio
0	CBSR0	0	R/W	This bit specifies the service ration of the control and bulk ED. The host controller must compare the ratio specified by the internal calculation whether it has processed several non-vacant control ED in determining whether another control ED is continued to be supplied or switched to bulk ED before any a periodic list is processed. In case of reset, HCD is responsible for restoring this value. 00: 1:1 01: 2:1 10: 3:1 11: 4:1

24.3.3 Hc Command Status Register (USBHCS)

The host controller uses USBHCS not only for reflecting the current status of the host controller, but also for receiving a command issued by HCD. A write is for setting HCD. The host controller must guarantee that the bit to which 1 is written to be set and the bit to which 0 is written to is unchanged. HCD must distribute multiple clear commands to the host controller by a previously issued command. The host controller driver can read all bits normally.

The SOC bit indicates the number of the frame that has detected the Scheduling Overrun error by the host controller. This occurs when the periodic list has not completed before EOF. When the Scheduling Overrun error is detected, the host controller increments the counter and sets SO bit in the USBHIS register.

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	SOC1	0	R/W	Scheduling Overrun Count
16	SOC0	0	R/W	These bits are incremented in each SchedulingOverrun error. These bits are initially set to B'00 and returned to B'11. These bits are incremented when SchedulingOverrun is detected even though the SO bit in USBHIS is set. These bits are used by HCD to monitor any continuous scheduling problem.
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	OCR	0	R/W	Ownership Change Request This bit is set by OS HCD to request the change of the control of the host controller. When this bit is set, the host controller sets the OC bit in USBHIS. After a change, this bit is cleared and remains until the next request from OS HCD. 0: After a change, this bit is cleared and remains until the next request from OS HCD 1: Set the OC bit in USBHIS
2	BLF	0	R/W	Bulk List Filled This bit is used to indicate that there are some TDs in the list. This bit is set by HCD to the list when TD is added to ED. When the host controller starts to process the head of the list, it checks this bit. As long as this bit is 0, the host controller does not start to process the list. When this bit is 1, the host controller starts to process the list to set BF to 0. When the host controller detects TD in the list, the host controller sets this bit to 1. When TD is never found in the list and HCD does not set this bit, the host controller completes the processing of the list. This bit is still 0 when the size list processing is stopped. 0: The list is not processed 1: The list is processed

Bit	Bit Name	Initial Value	R/W	Description
1	CLF	0	R/W	Control List Filled This bit is used to indicate that there are some TDs in the control list. This bit is set by HCD when TD is added to ED in the control list. When the host controller starts to process the head of the control list, it checks this bit. As long as this bit is 0, the host controller does not start to process the control list. If this bit is 1, the host controller starts to process the control list and this bit is set to 0. When the host controller finds TD in the list, the host controller sets this bit to 1. When TD is never detected in the control list and HCD does not set this bit, the host controller completes the processing of the control list. This bit is still 0 when the control list processing is stopped. 0: The list is not processed 1: The list is processed
0	HCR	0	R/W	Host Controller Reset This bit is set by HCD to initiate the software reset of the host controller. The system is moved to the UsbSuspend state in which most of the operational registers are reset except for the next state regardless of the functional state of the host controller. For example, an access to the IR bit in the USBHC register and without host bus is allowed. The host controller upon completion of the reset operation clears this bit. This bit does not cause any reset to the route hub and the next reset signal is not issued to the downstream port. 0: Cleared by the host controller at the completion of the reset control 1: UsbSuspend state

24.3.4 Hc Interrupt Status Register (USBHIS)

This register indicates the status in various events that cause hardware interrupts. When an event occurs, the host controller sets the corresponding bit in this register. When the bit is set to 1, a hardware interrupt is generated while an interrupt is enabled and the MIE bit is set in USBHIE (section 24.3.5, Hc Interrupt Enable Register (USBHIE)). HCD clears a specified bit in this register by writing 1 in the bit position to be cleared. The host controller driver cannot set any bit of these bits. The host controller never clears bits.

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	OC	0	R/W	Ownership Change This bit is set by the host controller when the OCR bit in USBHCS is set. This event generates a system management interrupt (SMI) at once when not masked. When there is no SMI pin, this bit is set to 0. 0: The OCR bit in USBHCS is not set 1: The OCR bit in USBHCS is set
29 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	RHSC	0	R/W	Root Hub Status Change This bit is set when the content of USBHRS or the content of any USBHRPS 1, 2 register has changed. 0: The content of USBHRS or USBHRPS is not changed 1: The content of USBHRS or USBHRPS is changed
5	FNO	0	R/W	Frame Number Overflow This bit is set when MSB (bit 15) in USBHFN changes value from 0 to 1 or from 1 to 0 or the Hcca Frame Number bit is updated. 0: MSB or the Hcca Frame Number bit in USBHFN is not updated 1: MSB or the Hcca Frame Number bit in USBHFN is updated

Bit	Bit Name	Initial Value	R/W	Description
4	UE	0	R/W	Unrecoverable Error This bit is set when the host controller detects a system error that is not related to USB. HCD clears this bit after the host controller is reset. 0: System error is not generated 1: System error is detected
3	RD	0	R/W	Resume Detected This bit is set when the host controller detects that a device of USB issues a resume signal. This bit is not set when HCD sets USB Resume state. 0: The resume signal is not detected 1: The resume signal is detected
2	SF	0	R/W	Start of Frame This bit is set by the host controller when each frame starts and after the Hcca Frame Number is updated. The host controller simultaneously generates the SOF token. 0: Each frame has not initiated or Hcca Frame Number is not updated 1: Initiation of each frame and updating of Hcca Frame Number
1	WDH	0	R/W	Write-back Done Head This bit is set immediately after the host controller has written Hc Done Head to Hcca Done Head. Hcca Done Head is not updated until this bit is cleared. HCD should clear this bit only after the content of Hcca Done Head has been stored. 0: When cleared after set to 1 1: When Hc Done Head is written to Hcca Done head
0	SO	0	R/W	Scheduling Overrun This bit is set when the USB schedule has overrun after Hcca Frame Number has updated. SchedulingOverrun also increments the SOC bit in USBHCS. 0: The USB schedule has not overrun 1: The USB schedule has overrun

24.3.5 Hc Interrupt Enable Register (USBHIE)

Each enable bit in USBHIE corresponds to the related interrupt bit in USBHIS. USBHIE is used to control an event to generate a hardware interrupt. A hardware interrupt is requested to the CPU when a bit in USBHIE is set, a corresponding bit in USBHIE is set, and the MIE bit is set. As a result, the USBHI bit in the interrupt request register 9 (IRR9) of the interrupt controller (INTC) is set (the USBHI bit is used in common regardless of the content of the interrupt generation event). Therefore, the USBHI bit can be used when an interrupt generation is detected by HCD.

Writing 1 in this register sets the corresponding bit, while writing 0 leaves the bit. When read, the current value of this register is returned.

Bit	Bit Name	Initial Value	R/W	Description
31	MIE	0	R/W	Master Interrupt Enable Setting this bit to 0 is ignored by the host controller. When this bit is set to 1, an interrupt generation by the event specified in another bit in this register is enabled. This is used by HDC that the master interrupt is enabled. When an interrupt is detected by HCD, use the USBIH bit of the interrupt controller (INTC). 0: Ignored 1: Interrupt generation due to the specified event enabled
30	OC	0	R/W	Ownership Change Enable 0: Ignored 1: Interrupt generation due to Ownership Change enabled
29 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	RHSC	0	R/W	Root Hub Status Change Enable 0: Ignored 1: Interrupt generation due to Root Hub Status Change enabled
5	FNO	0	R/W	Frame Number Overflow Enable 0: Ignored 1: Interrupt generation due to Frame Number Overflow enabled

Bit	Bit Name	Initial Value	R/W	Description
4	UE	0	R/W	Unrecoverable Error Enable 0: Ignored 1: Interrupt generation due to unrecoverable error enabled
3	RD	0	R/W	Resume Detected Enable 0: Ignored 1: Interrupt generation due to Resume Detected enabled
2	SF	0	R/W	Start of Frame Enable 0: Ignored 1: Interrupt generation due to Start of Frame enabled
1	WDH	0	R/W	Write-back Done Head Enable 0: Ignored 1: Interrupt generation due to WritebackDoneHead enabled
0	SO	0	R/W	Scheduling Overrun Enable 0: Ignored 1: Interrupt generation due to Scheduling Overrun enabled

24.3.6 Hc Interrupt Disable Register (USBHID)

Each disable bit in USBHID corresponds to the related interrupt bit in USBHIS. USBHID is related to USBHIE. Therefore, writing a 1 to a bit in this register clears the corresponding bit in USBHIE, while writing a 0 to a bit leaves the corresponding bit in USBHIE. When read, the current value of USBHIE is returned.

Bit	Bit Name	Initial Value	R/W	Description
31	MIE	0	R/W	Master Interrupt Enable 0: Ignored 1: Interrupt generation due to the specified event disabled
30	OC	0	R/W	Ownership Change Enable 0: Ignored 1: Interrupt generation due to OwnershipChange disabled

Bit	Bit Name	Initial Value	R/W	Description
29 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	RHSC	0	R/W	Root Hub Status Change Enable 0: Ignored 1: Interrupt generation due to RootHubStatusChange disabled
5	FNO	0	R/W	Frame Number Overflow Enable 0: Ignored 1: Interrupt generation due to FrameNumberOverflow disabled
4	UE	0	R/W	Unrecoverable Error Enable 0: Ignored 1: Interrupt generation due to UnrecoverableError disabled
3	RD	0	R/W	Resume Detected Enable 0: Ignored 1: Interrupt generation due to ResumeDetected disabled
2	SF	0	R/W	Start of Frame Enable (SF) 0: Ignored 1: Interrupt generation due to StartofFrame disabled
1	WDH	0	R/W	Write-back Done Head Enable (WDH) 0: Ignored 1: Interrupt generation due to WritebackDoneHead disabled
0	SO	0	R/W	Scheduling Overrun Enable (SO) 0: Ignored 1: Interrupt generation due to SchedulingOverrun disabled

24.3.7 HCCA Register (USBHHCCA)

USBHHCCA includes physical addresses of the host controller communication area. The host controller driver determines the alignment limitation by writing 1 to all bits in USBHHCCA and by reading the content of USBHHCCA. Alignment is evaluated by checking the number of 0 in the lower bits. The minimum alignment is 256 bytes. Consequently, bits 0 to 7 must be always returned to 0 when they are read. This area is used to retain the control structure and interrupt table that are accessed by the host controller and host controller driver.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	HCCA23 to HCCA0	All 0	R/W	HCCA Physical addresses of the host controller communication area
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.3.8 Hc Period Current ED Register (USBHPCED)

USBHPCED includes a physical address of current Isochronous ED or Interrupt ED.

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	PCED27 to PCED0	All 0	R	PCED Physical address of current Isochronous ED or Interrupt ED
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.3.9 Hc Control Head ED Register (USBHCHED)

USBHCHED includes a physical address of first ED in the control list.

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	CHED27 to CHED0	All 0	R/W	CHED Physical address of first ED in the control list
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.3.10 Hc Control Current ED Register (USBHCCED)

USBHCCED register includes a physical address of current ED in the control list.

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	CCED27 to CCED0	All 0	R/W	CCED Physical address of current ED in the control list
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.3.11 Hc Bulk Head ED Register (USBHBHED)

USBHBHED includes a physical address of first ED in the Bulk List.

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	BHED27 to BHED0	All 0	R/W	BHED Physical address of first ED in the Bulk List
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.3.12 Hc Bulk Current ED Register (USBHBCED)

USBHBCED includes a physical address of current ED in the Bulk List. When the bulk list is supplied by the round robin method, endpoints are ordered to the list according to these insertions.

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	BCED27 to BCED0	All 0	R/W	BCED Physical address of current ED in the Bulk List
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.3.13 Hc Done Head ED Register (USBHDHED)

USBHDHED includes a physical address of finally completed TD added to Done queue. The host controller driver needs not read this register so that the content is written to HCCA periodically in normal operation.

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	DH27 to DH0	All 0	R	DH Physical address of finally completed TD added to Done queue
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.3.14 Hc Fm Interval Register (USBHFI)

USBHFI includes a 14-bit value indicating the bit time interval of the frame (i.e., between two serial SOFs) and a 15-bit value indicating the maximum packet size at a full speed that is transmitted and received by the host controller without causing scheduling overrun. The host controller driver adjusts the frame interval minutely by writing a new value over the current value in each SOF.

Bit	Bit Name	Initial Value	R/W	Description
31	FIT	0	R/W	Frame Interval Toggle This bit is toggled by HCD whenever it loads a new value into FrameInterval.
30 to 16	FSMPS14 to FSMPS0	All 0	R/W	FS Largest Data Packet This field specifies a value, which is loaded into the Largest Data Packet Counter at the beginning of each frame. The counter value expresses the largest data amount of the bit that can be transmitted and received in one transaction by the host controller at any given time without causing scheduling overrun. The field value is calculated by HCD.
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	FI13	1	R/W	Frame Interval These bits specify the interval between two serial SOFs with bit times. The nominal value is set to 11999. HCD must store the current value of this field before resetting the host controller. With this procedure, this bit is reset to its nominal value by the host controller by setting the HCR bit in USBHCS. HCD can select to restore the stored value at the completion of the reset sequence.
12	FI12	0	R/W	
11	FI11	1	R/W	
10	FI10	1	R/W	
9	FI9	1	R/W	
8	FI8	0	R/W	
7	FI7	1	R/W	
6	FI6	1	R/W	
5	FI5	0	R/W	
4	FI4	1	R/W	
3	FI3	1	R/W	
2	FI2	1	R/W	
1	FI1	1	R/W	
0	FI0	1	R/W	

24.3.15 Hc Frame Remaining Register (USBHFR)

USBHFR is a 14-bit down counter indicating the bit time remaining in the current frame.

Bit	Bit Name	Initial Value	R/W	Description
31	FRT	0	R/W	Frame Remaining Toggle This bit is always loaded from the FIT bit in Hc Fm interval when FR reaches 0. This bit is used by HCD for the synchronization between FI and FR.
30 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13 to 0	FR13 to FR0	All 0	R/W	Frame Remaining This counter is decremented at each bit time. When this counter reaches 0, this counter is reset by loading the value of the FI bit specified in USBHFI at the next bit time boundary. When the host controller transits to the UsbOperational state, it read the FI bit in USBHFI again and uses the updated value from the next SOF.

24.3.16 Hc Fm Number b Register (USBHFN)

USBHFN is a 16-bit counter. It indicates the reference of timing between events occurring in the host controller and host controller driver. The host controller driver uses a 16-bit value specified in this register and generates a 32-bit frame number without necessity for a frequent access to the register.

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 0	FN15 to FN0	All 0	R/W	Frame Number These bits are incremented when USBHFN is reloaded. The count will return to H'0 after H'FFFF. When the host controller transits to the UsbOperational state, these bits are automatically incremented. After the host controller increments the FN bit and sends SOF in each frame boundary, the content is written to HCCA before the host controller reads first ED in the frame. After writing to HCCA, the host controller sets the SF bit in USBHIS.

24.3.17 Hc Periodic Start Register (USBHPS)

USBHPS has a 14-bit programmable value, which determines the earliest time when the host controller should start to process the periodic list.

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 to 0	PS13 to PS0	All 0	R/W	Periodic Start This field is cleared after the hardware has reset. Then this field is set by HCD while the host controller performs initial settings. The value is roughly calculated as the value of the USBHFI minus 10%. When USBHFR reaches the specified value, the processing of the periodic list has a higher priority than the control/bulk processing. Consequently, the host controller starts to process the interrupt list after the completion of the current control/bulk transaction.

24.3.18 Hc LS Threshold Register (USBHLST)

USBHLST includes an 11-bit value that is used by the host controller to determine whether or not to authorize the transfer of the LS packed 8 bytes in maximum before EOF. The host controller and host controller driver cannot change this value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	LST11	0	R/W	LS Threshold This field contains a value to be compared with the FR bit prior to the beginning of low-speed transaction. The transaction is started only when the FR bit value is beyond the value of the list. The value is calculated by HCD considering the transmission and set-up overhead.
10	LST10	1	R/W	
9	LST9	1	R/W	
8	LST8	0	R/W	
7	LST7	0	R/W	
6	LST6	0	R/W	
5	LST5	1	R/W	
4	LST4	0	R/W	
3	LST3	1	R/W	
2	LST2	0	R/W	
1	LST1	0	R/W	
0	LST0	0	R/W	

24.3.19 Hc Rh Descriptor A Register (USBHRDA)

USBHRDA is the first register of two registers describing the features of the root hub. The reset value is implementation specific. The descriptor length (11), descriptor type (TBD), and the hub controller current bit (0) of Class Descriptor of the hub are emulated by HCD. All other bits are placed in USBHRDA and USBHRDB.

Bit	Bit Name	Initial Value	R/W	Description
31	POTPGT7	0	R/W	Power On To Power Good Time
30	POTPGT6	0	R/W	These bits specify the time required for waiting before accessing the power-on port of the root hub. These bits are implementation specific. The unit of time is 2 ms. The time is calculated as POTPGT × 2 ms.
29	POTPGT5	0	R/W	
28	POTPGT4	0	R/W	
27	POTPGT3	0	R/W	
26	POTPGT2	0	R/W	
25	POTPGT1	1	R/W	
24	POTPGT0	0	R/W	
23 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	NOCP	1	R/W	No Over Current Protection This bit selects how the over-current status of the root hub is reported. When this bit is cleared, the OCPM bit specifies global report or report at each port. 0: Over-current status is collectively reported for all downstream ports 1: Over-current protection is not supported
11	OCPM	0	R/W	Over Current Protection Mode This bit selects how the over-current status in the root-hub port is reported. At reset, this bit reflects the same mode of PowerSwitchingMode. When the NOCP bit is cleared, this bit is valid. 0: Over-current status is collectively reported for all downstream ports 1: Over-current protection is not supported
10	DT	0	R	Device Type This bit indicates that the USB Host Controller is not a compound device. Always set this bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
9	NPS	1	R/W	No Power Switching This bit selects whether the power switching is supported or ports are always power-supplied. This bit is implementation specific. When this bit is cleared, the PSM bit specifies the global/port switching. 0: Ports can be power-switched 1: Ports are always powered on when the host controller is powered on Note: Since the initial value is 1, first clear this bit (write 0 with the HCD) to enable power switching of the port.
8	PSM	0	R/W	Power Switching Mode This bit specifies how the power switching of the root-hub port is controlled. This bit is implementation specific. This bit is valid only when the NPS bit is cleared. 0: All ports are simultaneously power-supplied 1: Each port is power-supplied individually. In this mode, the port power is controlled with either of global/port switching. When the PPCM bit in USBHRDB is set, the port is reacted only to the port-power command (set/clear port power). When the port mask is cleared, the port is controlled only by the global power-switch (set/clear global power).
7	NDP7	0	R	Number Down stream Ports These bits specify the number of downstream ports supported by the root hub. These bits are implementation specific. In this LSI, their value is H'2.
6	NDP6	0		
5	NDP5	0		
4	NDP4	0		
3	NDP3	0		
2	NDP2	0		
1	NDP1	1		
0	NDP0	0		

24.3.20 Hc Rh Descriptor B Register (USBHRDB)

USBHRDB is the second register of two registers describing the features of the root hub. These bits are written during the initial setting so as to correspond to the system implementation. The reset value is implementation specific.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	PPCM15 to PPCM0	All 0	R/W	Port Power Control Mask This bit indicates that the port is influenced by the global power-control command when the PSM bit in the USBHRDA register is set. When this bit is set, the power state of the port is affected by the power control at each port (set/clear port power). When this bit is cleared, the port is controlled by the global power switch (set/clear global power). If the device is placed in the global switching mode (PSM = 0), this bit is not valid. Bit 31: Port#15 power mask : Bit 18: Port#2 power mask Bit 17: Port#1 power mask Bit 16: Reserved Note: Clear the NPS of the USBHRDA register so that the power to all ports is OFF (Port Power Status = 0), then set this bit.
15 to 0	DR15 to DR0	All 0	R/W	Device Removable These bits are dedicated to the ports of the root hub. When these bits are cleared, the set device becomes removable. When these bits are set, do not remove the set device. Bit 15: Device affixed to Port#15 : Bit 2: Device affixed to Port#2 Bit 1: Device affixed to Port#1 Bit 0: Reserved

24.3.21 Hc Rh Status Register (USBHRS)

USBHRS is divided into two parts. The lower word of a long word indicates the hub status bits and the upper word indicates the hub status change bit. Reserved bits should be set to 0.

Bit	Bit Name	Initial Value	R/W	Description
31	CRWE	0	W	Clear Remote Wakeup Enable Writing a 1 to this bit clears DeviceRemoteWakeupEnable. Writing 0 to this bit has no effect.
30 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	OCIC	0	R/W	Over Current Indicator Change This bit is set when the OCI bit changes. Writing 1 clears this bit. Writing 0 has no effect.
16	LPSC	0	R/W	(Read) Local Power Status Change The root hub does not support the local power status function. Therefore, this bit is always read as 0. (Write) Set Global Power This bit is written to 1 to power on (clears the PPS bit in USBHRPS) all ports in global power mode (PSM bit in USBHRDA = 0). This bit sets the PPS bit only to the port in which the PPCM bit is not set in power mode at each port. When a 0 is written to, this bit is not cleared.
15	DRWE	0	R/W	(Read) Device Remote Wakeup Enable This bit enables the CSC bit as a resume event and generates the state transition from USBHSUSPEND1 to USBRESUME and ResumeDetected interrupt. 0: ConnectStatusChange is not the remote wakeup event 1: ConnectStatusChange is the remote wakeup event. (Write) Set Remote Wakeup Enable Writing a 1 sets DeviceRemoteWakeupEnable. Writing a 0 has no effect.
14 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	OCI	0	R	Over Current Indicator This bit reports the over-current condition. When this bit is set, an over-current condition exists. When this bit is cleared, all power operations are normal. This bit is always 0 when the over-current protection at each port is carried out. 0: All power operations are normal 1: An over-current condition exists
0	LPS	0	R/W	(Read) Local Power Status The root hub does not support the local power status function. Therefore, the bit is always read 0. (Write) Clear Global Power This bit is written to 1 to power on (the PPS bit in USBHRPS is cleared) all ports in global power mode (PSM in USBHRDA = 0). In the power mode at each port, the PPS bit is cleared to the port in which the PPCM bit is not set. Writing a 0 has no effect.

24.3.22 Hc Rh Port Status 1 and Hc Rh Port Status 2 Registers (USBHRPS1, USBHRPS2)

USBHRPS 1 and USBHRPS 2 registers are used for base-controlling each port and to report the port event. The lower word is used to reflect the port status while the upper word reflects the status change. Some status bits have special writing (see below). If an attempt to write to a bit indicating a change in port status occurs when a transaction in which a token is passed via a handshake is in progress, the writing to the bit is delayed until the transaction is completed. Always write reserved bits to 0.

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	PRSC	0	R/W	Port Reset Status Change This bit is set when the 10 ms port reset signal has completed. Writing a 1 clears this bit; writing a 0 has no effect. 0: Port reset is not complete 1: Port reset is complete
19	OCIC	0	R/W	Port Over Current Indicator Change This bit is valid when an over-current condition is reported on the base of each port. This bit is set when the root hub changes the POCI bit. Writing a 1 clears this bit. Writing a 0 has no effect. 0: PortOverCurrentIndicator not changed 1: PortoverCurrentIndicator changed
18	PSSC	0	R/W	Port Suspend Status Change This bit is set when all resume sequences have completed. These sequences include 20 ms resume pulse, LS EOP, and 3 ms resychronization delay. Writing a 1 clears this bit. Writing a 0 has no effect. This bit is cleared also when the PRSC bit is set. 0: Port resume not completed 1: Port resume completed

Bit	Bit Name	Initial Value	R/W	Description
17	PESC	0	R/W	Port Enable Status Change This bit is set when the PES bit is cleared due to a hardware event. This bit is not set by the change of writing of HCD. Writing a 1 clears this bit. Writing a 0 has no effect. 0: PortEnableStatus not changed 1: PortEnableStatus changed
16	CSC	0	R/W	Connect Status Change This bit is set whenever the connection or disconnection event occurs. Writing a 1 clears this bit. Writing a 0 has no effect. If the CCS bit is cleared when SetPortReset, SetPortEnable, or SetPortSuspend is written to, writing when the power supply of the port is disconnected does not occur, so this bit is set to enforce the driver to re-evaluate the connection status. 0: CurrentConnectionStatus not changed 1: CurrentConnectionStatus changed Note: If the DR bit in USBHRDB is set, this bit is set only after the root hub reset to inform that the system that a device can be attached.
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	LSDA	0	R/W	(Read) Low Speed Device Attached This bit indicates the speed of the device attached to this port. When this bit is set, a low-seed device is attached to this port. When this bit is cleared, a full-speed device is attached to this port. This bit is valid only when the CCS bit is set. 0: A full-speed device is set 1: A low-speed device is set (Write) Clear Port Power Writing a 1 clears the PPS bit. Writing a 0 has no effect.
8	PPS	1	R/W	(Read) Port Power Status This bit reflects the power state of the port regardless of the power-switching mode to be executed. However, because the initial value of the NPS bit of the USBHRDA is 1, this bit is first fixed to 1. The NPS bit must first be cleared before the power is switched, as shown below. When an over-current condition is detected, this bit is cleared. Writing SetPortPower or SetGlovalPower sets this bit. Writing ClearPortPower or ClearGlobalPower clears this bit. The PSM bit in USBHRDA and the PPCM bit in USBHRDB determine which power control switch can be used. Only Set/ClearGlobalPower controls this bit in global switching mode (PSM= 0). If the PPCM bit of that port is set in power switching mode (PSM = 1), only the Set/ClearPortPower command is enabled. If the mask is not set, the Set/ClearGlovalPowerCommand is enabled. When the port power is disabled, the CCS, PES, PSS, and PRS are reset. 0: Port power is off 1: Port power is on Note: If power switching is not supported, this bit is always read as 1. (Write) Set Port Power Writing a 1 sets the PPS bit. Writing a 0 has no effect.
7 to 5	—	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
4	PRS	0	R/W	(Read) Port Reset Status When this bit is set by writing to SetPortReset, the port reset signal is output. This bit is cleared when PRSC is set upon completion of a reset. When the CCS is cleared, this bit is not set. 0: Port reset signal is not active 1: Port reset signal is active (Write) Set Port Reset Writing a 1 sets PortReset signal. Writing a 0 has no effect. When the CCS bit is cleared, this write does not set the PRS bit, instead, sets the CSC bit. This reports a reset of the power disconnection port to the driver.
3	POCI	0	R/W	(Read) Port Over Current Indicator This bit is valid only when a root hub is placed in such a way that an over-current condition is reported on the base of each port. If the over-current report at each port is not supported, this bit is cleared to 0. If this bit is cleared, all power controls are normal in this port. If this bit is set, an over-current status exists in this port. This bit always reflects an over-current input signal. 0: No over-current condition 1: Over-current condition is detected (Write) Clear Suspend Status Writing a 1 initiates a resume. Writing a 0 has no effect. If the PSS bit is set, a resume is initiated.

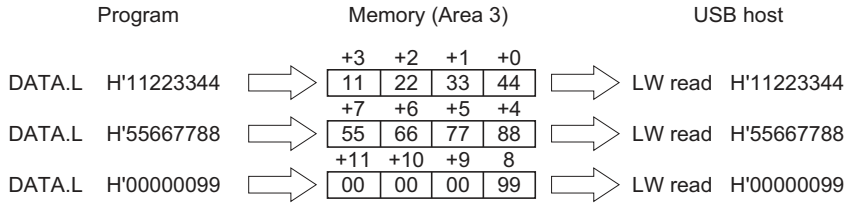
Bit	Bit Name	Initial Value	R/W	Description
2	PSS	0	R/W	(Read) Port Suspend Status This bit indicates that the port is suspended or during the resume sequence. Writing SetSuspendState sets this bit and setting PSSC clears this bit at the end of the resume interval. If the CCS bit is cleared, this bit cannot be set. When the PRSC bit is set upon completion of the port reset or HC is placed in the UsbResume state, this bit is cleared. If an upstream resume is in progress, it is transmitted to the host controller. 0: Port is not suspended 1: Port is selectively suspended (Write) Set Port Suspend Writing a 1 sets PortSuspendStatus. Writing a 0 has no effect. In addition, when the CCS bit is cleared, the PSS bit is not set by this writing. Instead, the CSC bit is set. This reports the suspended state of the power disconnection to the driver.
1	PES	0	R/W	(Read) Port Enable Status This bit indicates whether the port is enabled or disabled. The root hub clears this bit when the over-current condition and an operational bus error such as disconnect event, power-off switch, or babble is detected. The PESC is set by this change. This bit is set by writing SetPortEnable and cleared by writing ClearPortEnable. This bit cannot be set when the CCS bit is cleared. In addition, this bit is set upon completion of the port reset by which the PRSCtatusChange is set, or uponcompletion of the port suspend by which the PSSC is set. 0: Port disabled 1: Port enabled (Write) Set Port Enable Writing a 1 sets the PES bit. Writing a 0 has no effect. If the CCS bit is cleared, this writing does not set the PES bit, instead, sets the CS. This reports the driver that the power disconnection port has been tried to be enabled.

Bit	Bit Name	Initial Value	R/W	Description
0	CCS	0	R/W	(Read) Current Connect Status This bit indicates the status of the downstream port. 0: No device connected 1: Device connected Note: If DeviceRemoveable is set (not removable) this bit is always read as 1. (Write) Clear Port Enable Writing a 1 clears the PES bit. Writing a 0 has no effect. The CCS bit is not affected by any writing.

24.4 Data Storage Format which Required by USB Host Controller

24.4.1 Storage Format of the Transferred Data

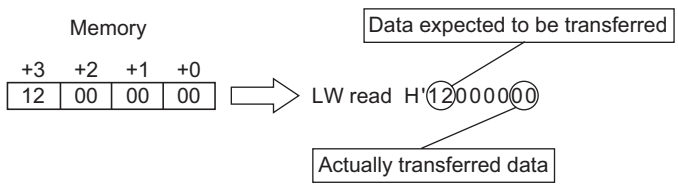
USB Host Controller expects that data is compiled from lower address to upper address regardless endian setting of the CPU. Below figure shows data read operation, which is done by USB Host Controller.



The correspondence between data in memory and data read by USB Host Controller must be equal. When USB Host Controller reads data from external memory, USB Host Controller reads data by long word read operation every time regardless of endian. USB Host Controller uses data in byte from lower address in long word which it reads regardless the endian mode. Even endian mode is set as big or little, set the data from down addresses.

Below program flow is the example of failure.

- In program, set transfer address A to register R0 at big endian
  In program, "MOV.B #H'12,@R0"
- In program, set transfer start address A to USB Host Controller, and set 1byte as transfer size.



This example shows above operation transfers expected data #H'12.

Data is filled from the lower bits of the memory in writing so that the data is read/written in bi-direction consistently regardless of the endian type. That is, the data is always aligned with the little endian specification.

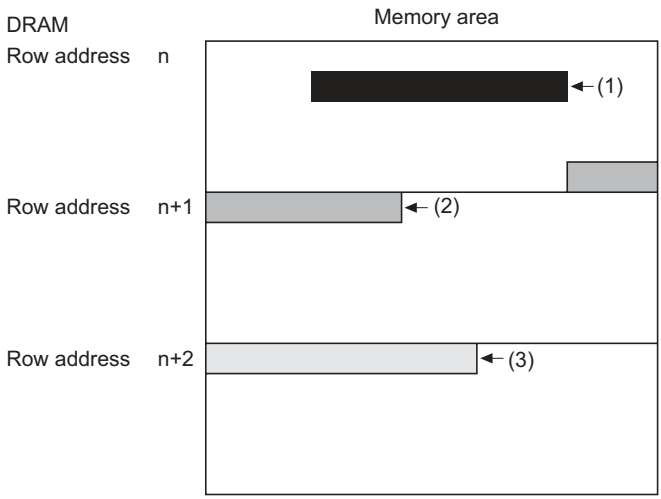
24.4.2 Storage Format of the Descriptor

ED (endpoint descriptor) and TD (transfer descriptor) that define each transfer transaction of USB Host Controller must be aligned so that each Dword corresponds to the long-word boundary (addresses 4n to 4n + 3) of the memory.

24.5 Data Alignment Restriction of USB Host Controller

24.5.1 Restriction on the Line Boundary of the Synchronous DRAM

The transferred data is stored in shared system memory with CPU. The data alignment in system memory are restricted depends on SDRAM specification which is used as system memory.



In above figure, transfer data 1 and 3 are able to be read or written by USB Host Controller. But transfer data 2 are possibly unable to be read or written by USB Host controller. Any data, which have possibility to be accessed by USB Host Controller, must be aligned in SDRAM not to cross row address alignment.

24.5.2 Restriction on the Memory Access Address

MPS in ED, CBP in General TD, and BP0 and OFFSET0 to 7 in Ischoronous TD must be set in multiples of 4 (4n). In the OpenHCI standard, 1 packet is transferred by ITD in General TD and 1 packet by 1 offset in Ischronous TD during IN transfer. In addition, when the amount of the data specified by TD during OUT transfer exceeds MAXPACKETSIZE (MPS), a packet transmission is carried out in MAXPACKETSIZE. Therefore, the setting value can be made as above. This restriction is due to the difference between the specifications of the HCI interface which is the standard of the IP bus interface of USB and of the bus interface of this LSI. Data might be correctly written to if data is transferred from addresses other than 4n address. For example, when a two-byte transfer is carried out from the address that terminates at 1, a long-word transfer is carried out and an unexpected data is written to starting address 0.

24.6 Accessing External Address from the USB Host

Accessing the external address from the USB Host is carried out as follows:

- When reading, 4, 8, 12, or 16-byte transfer in longword units.
- When writing, 1 to 16-byte transfer.

24.7 Usage Notes

1. When using the USB host controller, the bus clock (Bϕ) must be set to 32 MHz or higher. The peripheral clock (Pϕ) must also be set to a higher frequency than 13 MHz.
2. Usage notes on Resume operation

   (1) Phenomenon

   While the USB host is providing an output of a Resume (*1) signal, suppose that (a) PortPower is turned off or that (b) OverCurrent is produced. In this case, the Resume signal should ordinarily be stopped so that the idle (*2) state will be established. Actually, however, the result is that an idle signal is output.

   *1: In FullSpeed, D+ = Low and D- = High. In LowSpeed, D+ = High and D- = Low.

   *2: In FullSpeed, D+ = High and D- = Low. In LowSpeed, D+ = Low and D- = High.

   (2) Conditions when the above phenomenon occurs

   While a Resume (*1) signal is being output, (a) PortPower is turned off or (b) OverCurrent is produced.

   (3) Conditions when the above phenomenon does not occur

   The above phenomenon will not occur if there is no Resume operation, that is, Suspend operation has not been done.

   (4) Problem avoidance by software

   If the above phenomenon occurs, Resume is interrupted and then an idle signal is output. However, turning on PortPower enables device recognition. The above phenomenon is removed by the subsequent Port Reset for the device. Normal operation is thus recovered. Note, however, the above phenomenon will not be removed by USB Reset, which is generated by the HCFS1 and HCFS0 bits in the Hc Control (USBHC) register. For this reason, if you are using software that issues USB Reset by the HCFS1 and HCFS0 bits in the Hc Control (USBHC) register, modify the software so that it issues USB Reset (Port Reset) by setting the PRS bit in the Hc Rh Port Status 1 or Hc Rh Port Status 2 (USBHRPS1 or USBHRPS2) register.

   However, there is no need to take corrective action if Port Rest has already been issued by the PRS bit before the recognition of USB device connection.

Section 25 USB Function Controller (USBF)

This LSI incorporates an USB function controller (USBF).

25.1 Features

- UDC (USB device controller) conforming to USB1.1 processes incorporated USB protocol automatically.

  Automatic processing of USB standard commands for endpoint 0 (some commands and class/vendor commands require decoding and processing by firmware)
- Transfer speed: Full-speed
- Endpoint configuration: An arbitrary endpoint configuration can be set

  The arbitrary endpoint can be configured by setting the correspondence between the endpoint (the endpoint number used by the USB host) and the EP FIFO number that is provided by this USB function controller (the transfer method and direction are fixed).

EP FIFO Number	Abbreviation	Transfer Type	Maximum Packet Size	FIFO Buffer Capacity (Byte)	DMA Transfer
Endpoint 0	EP0s	Setup	8	8	—
	EP0i	Control-in	8	8	—
	EP0o	Control-out	8	8	—
Endpoint 1	EP1	Bulk-out	64	128	Possible
Endpoint 2	EP2	Bulk-in	64	128	Possible
Endpoint 3	EP3	Interrupt	8	8	—
Endpoint 4	EP4	Isochronous-out	64	128	—
Endpoint 5	EP5	Isochronous-in	64	128	—

- Interrupt requests: generates various interrupt signals necessary for USB transmission/reception
- Clock: External input (48 MHz)

- Power-down mode

  Power consumption can be reduced by stopping UDC internal clock when USB cable is disconnected

  Automatic transition to/recovery from suspend state
- Can be connected to a Philips PDIUSBP11 Series transceiver or compatible product (when using a compatible product, carry out evaluation and investigation with the manufacturer supplying the transceiver beforehand)

Figure 25.1 shows the block diagram of USBF.

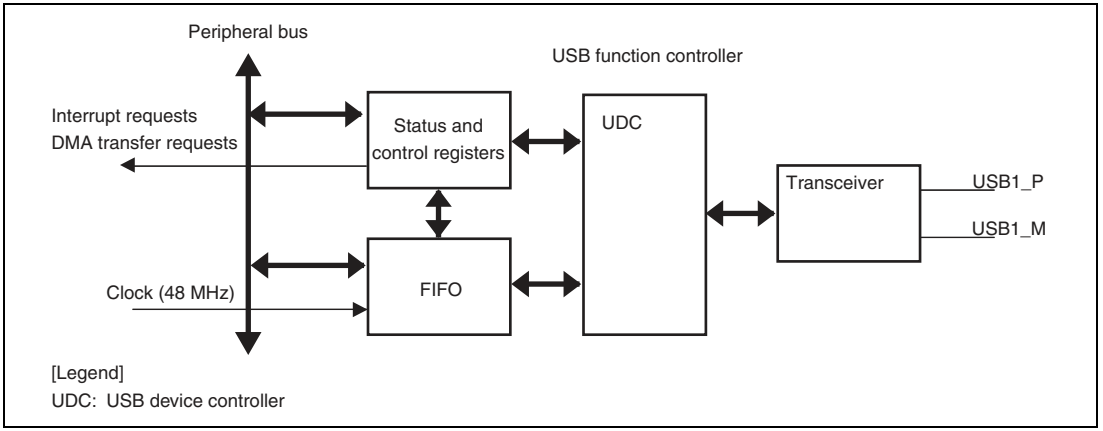


Figure 25.1 Block Diagram of USBF

25.2 Input/Output Pins

Table 25.1 lists the pin configuration of USBF.

Table 25.1 Pin Configuration and Functions

Name	Pin Name	I/O	Function
RCV pin	USB1d_RCV	Input	Input pin for receive data from differential receiver
DPLS pin	USB1d_DPLS	Input	Input pin to driver for D+ signal from receiver
DMNS pin	USB1d_DMNS	Input	Input pin to driver for D– signal from receiver
TXDPLS pin	USB1d_TXDPLS	Output	D+ transmit output pin to driver
TXSE0 pin	USB1d_TXSE0	Output	SE0 output pin
TXENL pin	USB1d_TXENL	Output	Driver output enable pin
USB1 overcurrent/monitor pin	$\overline{\text{USB1\_ovr\_current}}$/ USBF_VBUS	Input	USB port 1 over-current detection/ USB cable connection monitor pin
SUSPEND pin	USB1d_SUSPND	Output	Transceiver suspend state output pin
USB external clock	EXTAL_USB	Input	Connect a crystal resonator for USB. Alternatively, an external clock may be input for USB (48 MHz).
USB crystal	XTAL_USB	Output	Connect a crystal resonator for USB.
USB1 power enable/pull-up control pin	USB1_pwr_en/USBF_UPLUP	Output	USB port 1 power enable control/ Pull-up control output pin
1P pin	USB1_P	I/O	D+
1M pin	USB1_M	I/O	D–

25.3 Register Descriptions

USB has following registers. Refer to section 37, List of Registers, for more details on the addresses and states of these registers in each operating mode.

- Interrupt flag register 0 (IFR0)
- Interrupt flag register 1 (IFR1)
- Interrupt flag register 2 (IFR2)
- Interrupt flag register 3 (IFR3)
- Interrupt flag register 4 (IFR4)
- Interrupt select register 0 (ISR0)
- Interrupt select register 1 (ISR1)
- Interrupt select register 2 (ISR2)
- Interrupt select register 3 (ISR3)
- Interrupt select register 4 (ISR4)
- Interrupt enable register 0 (IER0)
- Interrupt enable register 1 (IER1)
- Interrupt enable register 2 (IER2)
- Interrupt enable register 3 (IER3)
- Interrupt enable register 4 (IER4)
- EP0i data register (EPDR0i)
- EP0o data register (EPDR0o)
- EP0s data register (EPDR0s)
- EP1 data register (EPDR1)
- EP2 data register (EPDR2)
- EP3 data register (EPDR3)
- EP4 data register (EPDR4)
- EP5 data register (EPDR5)
- EP0o receive data size register (EPSZ0o)
- EP1 receive data size register (EPSZ1)
- EP4 receive data size register (EPSZ4)
- Trigger register (TRG)
- Data status register (DASTS)
- FIFO clear register 0 (FCLR0)
- FIFO clear register 1 (FCLR1)

- DMA transfer setting register (DMA)
- Endpoint stall register 0 (EPSTL0)
- Endpoint stall register 1 (EPSTL1)
- Configuration value register (CVR)
- Time stamp register H (TSRH)
- Time stamp register L (TSRL)
- Control register 0 (CTLR0)
- Control register 1 (CTLR1)
- Endpoint information register (EPIR)
- Timer register H (TMRH)
- Timer register L (TMRL)
- Set time out register H (STOH)
- Set time out register L (STOL)

25.3.1 Interrupt Flag Register 0 (IFR0)

IFR0 is an interrupt flag register for EP0i, EP0o, EP1, EP2, bus reset, and setup command reception. When each flag is set to 1 and the interrupt is enabled in the corresponding bit of IER0, an interrupt request is generated as specified by the corresponding bit in ISR0. Clearing is performed by writing 0 to the bit to be cleared. Writing 1 is not valid and nothing is changed.

EP2 EMPTY and EP1 FULL are status bits that indicate the FIFO states of EP1 and EP2, respectively. Therefore, EP2 EMPTY and EP1 FULL cannot be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7	BRST	0	R/W	Bus Reset [Setting condition] When a bus reset signal is detected on the USB bus. [Clearing conditions] • When reset • When 0 is written to by CPU
6	EP1 FULL	0	R	EP1 (Bulk-out) FIFO Full [Setting condition] The FIFO buffer of EP1 has a dual-buffer configuration, and this bit is set when at least one of the FIFO buffer is full. [Setting conditions] • When reset • When both FIFO buffers are empty. Note: EP1 FULL is a status bit, and cannot be cleared.
5	EP2 TR	0	R/W	EP2 (Bulk-in) Transfer Request [Setting condition] When an IN token is received from the host to EP2 and both of FIFO buffers are empty. [Clearing conditions] • When reset • When 0 is written to by CPU

Bit	Bit Name	Initial Value	R/W	Description
4	EP2 EMPTY	1	R	EP2 (Bulk-in) FIFO Empty [Setting conditions] • When reset • The FIFO buffer of EP2 has a dual-buffer configuration, and this bit is set when at least one of the FIFO buffer is empty. [Clearing condition] When both of FIFO buffers are not empty. Note: EP2 EMPTY is a status bit, and cannot be cleared.
3	SETUP TS	0	R/W	Setup Command Receive Complete [Setting condition] When 8-byte data that decodes the command by the function is normally received from the host to EP0s and an ACK handshake is returned to the host from the function. [Clearing conditions] • When reset • When 0 is written to by CPU
2	EP0o TS	0	R/W	EP0o Receive Complete [Setting condition] When data is normally received from the host to EP0o and an ACK handshake is returned from the function to the host. [Clearing conditions] • When reset • When 0 is written to by CPU
1	EP0i TR	0	R/W	EP0i Transfer Request [Setting condition] When IN token is issued from the host to EP0i and the FIFO buffer is empty. [Clearing conditions] • When reset • When 0 is written to by CPU

Bit	Bit Name	Initial Value	R/W	Description
0	EP0i TS	0	R/W	EP0i Transmit Complete [Setting condition] When data to be transmitted to the host is written to EP0i, then data is normally transferred from the function to the host, and an ACK handshake is returned. [Clearing conditions] • When reset • When 0 is written to by CPU

25.3.2 Interrupt Flag Register 1 (IFR1)

IFR1 is an interrupt flag register for VBUS and EP3. When each flag is set to 1 and the interrupt is enabled in the corresponding bit of IER1, an interrupt request is generated as specified by the corresponding bit in ISR1. Clearing is performed by writing 0 to the bit to be cleared. Writing 1 is not valid and nothing is changed.

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	VBUS MN	0	R	USB Connection Status Status bit to monitor the USBF_VBUS pin state. Reflects the state of the USBF_VBUS pin. 0: Disconnected 1: Connected
2	EP3 TR	0	R/W	EP3 (Interrupt) Transfer Request [Setting condition] When an IN token is issued from the host to EP3 and the FIFO buffer is empty. [Clearing conditions] • When reset • When 0 is written to by CPU

Bit	Bit Name	Initial Value	R/W	Description
1	EP3 TS	0	R/W	EP3 (Interrupt) Transmit Complete [Setting condition] When data to be transmitted to the host is written to EP3, then data is normally transferred from the host to the function, and an ACK handshake is returned. [Clearing conditions] • When reset • When 0 is written to by CPU
0	VBUSF	0	R/W	USB Disconnection Detection The USBF_VBUS pin of this module is used for detecting connection/disconnection. [Setting condition] When the function is connected to the USB bus or disconnected from it. [Clearing conditions] • When reset • When 0 is written to by CPU.

25.3.3 Interrupt Flag Register 2 (IFR2)

IFR2 is an interrupt flag register for SURSS, SURSF, CFDN, SOF, SETC, and SETI. When each flag is set to 1 and an interrupt is enabled in the corresponding bit of IER2, an interrupt occurs as specified by the corresponding bit in ISR2. Clearing is performed by writing 0 to the bit to be cleared. Writing 1 is not valid and nothing is changed.

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	SURSS	0	R	Suspend/Resume Status Status bit indicating the state of the bus 0: Normal state 1: Suspend state

Bit	Bit Name	Initial Value	R/W	Description
4	SURSF	0	R/W	Suspend/Resume Detection [Setting condition] When the bus transits from the normal state to the suspend state or from the suspend state to the normal state. [Clearing conditions] • When reset • When 0 is written to by CPU
3	CFDN	0	R/W	End Point Information Load Complete [Setting condition] When the end point information written in EPIR is completed to be set (loaded) in this controller. Note: This controller operates normally as USB after the setting of the end point information is completed. [Clearing conditions] • When reset • When 0 is written to by CPU
2	SOF	0	R/W	SOF Packet [Setting condition] When the valid SOF packet is detected. [Clearing conditions] • When reset • When 0 is written to by CPU
1	SETC	0	R/W	Set Configuration Command Detection [Setting condition] When the valid Set Configuration command is detected. [Clearing conditions] • When reset • When 0 is written to by CPU

Bit	Bit Name	Initial Value	R/W	Description
0	SETI	0	R/W	Set Interface Command Detection [Setting condition] When the valid Set Interface command is detected. [Clearing conditions] • When reset • When 0 is written to by CPU

25.3.4 Interrupt Flag Register 3 (IFR3)

IFR1 is an interrupt flag register for EP4 TS, EP4 TF, EP5 TS, and EP5 TR. When each flag is set to 1 and the interrupt is enabled in the corresponding bit of IER3, an interrupt request is generated as specified by the corresponding bit in ISR3. Clearing is performed by writing 0 to the bit to be cleared. Writing 1 is not valid and nothing is changed.

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	EP5 TR	0	R/W	EP5 (Isochronous-in) Transmit Request Flag indicating the FIFO state of EP5. After the SOF packet is received, the FIFO buffer is switched automatically. The FIFO buffer which has transmitted data to the host in the previous frame (before SOF reception) can be written to by the CPU. This bit indicates the transmit state in the previous frame. [Setting condition] The FIFO buffer to be transmitted is empty when an IN token is issued from the host to EP5. [Clearing conditions] • When reset • When 0 is written to by CPU

Bit	Bit Name	Initial Value	R/W	Description
2	EP5 TS	0	R/W	EP5 (Isochronous-in) Normal Transmission Flag indicating the FIFO state of EP5. After the SOF packet is received, the FIFO buffer is switched automatically. The FIFO buffer which has transmitted data to the host in the previous frame (before SOF reception) can be written to by the CPU. This bit indicates the transmit state in the previous frame. [Setting condition] When a transmission was carried out normally in the previous frame. [Clearing conditions] • When reset • When 0 is written to by CPU
1	EP4 TF	0	R/W	EP4 (Isochronous-out) Abnormal Reception Flag indicating the FIFO state of EP4. Indicates the state of the FIFO buffer that was readable after the data reception is completed and the next SOF packet is received. [Setting condition] When the transfer data from the host is abnormally received (packet error) by EP4. [Clearing conditions] • When reset • When 0 is written to by CPU
0	EP4 TS	0	R/W	EP4 (Isochronous-out) Normal Reception Flag indicating the FIFO state of EP4. Indicates the state of the FIFO buffer that was readable after the data reception is completed and the next SOF packet is received. [Setting condition] When the transfer data from the host is normally received by EP4. [Clearing conditions] • When reset • When 0 is written to by CPU

25.3.5 Interrupt Flag Register 4 (IFR4)

IFR4 is an interrupt flag register for TMOUT. When each flag is set to 1 and the interrupt is enabled in the corresponding bit of IER4, an interrupt request is generated as specified by the corresponding bit in ISR4. Clearing is performed by writing 0 to the bit to be cleared. Writing 1 is not valid and nothing is changed.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TMOUT	0	R/W	Time Out [Setting condition] When the value of the timer register (TMR) is reached to that of the set time out register (STO). [Clearing conditions] • When reset • When 0 is written to by CPU

25.3.6 Interrupt Select Register 0 (ISR0)

ISR0 selects the interrupt requests to the INTC to be indicated in interrupt flag register 0. When a bit in ISR0 is cleared to 0, the corresponding interrupt is requested as a USBFI0 interrupt. When a bit is set to 1, the corresponding interrupt is requested as a USBFI1 interrupt. With the initial value, each of the interrupt source flags in the interrupt flag register 0 is selected as a USBFI0 interrupt.

Bit	Bit Name	Initial Value	R/W	Description
7	BRST IS	0	R/W	BRST Interrupt Select
6	EP1 FULL IS	0	R/W	EP1 FULL Interrupt Select
5	EP2 TR IS	0	R/W	EP2 TR Interrupt Select
4	EP2 EMPTY IS	0	R/W	EP2 EMPTY Interrupt Select
3	SETUP TS IS	0	R/W	SETUP Interrupt Select
2	EP0o TS IS	0	R/W	EP0o TS Interrupt Select
1	EP0i TR IS	0	R/W	EP0i TR Interrupt Select
0	EP0i TS IS	0	R/W	EP0i TS Interrupt Select

25.3.7 Interrupt Select Register 1 (ISR1)

ISR1 selects the interrupt requests to the INTC to be indicated in interrupt flag register 1. When a bit in ISR1 is cleared to 0, the corresponding interrupt is requested as a USBFI0 interrupt. When a bit is set to 1, the corresponding interrupt is requested as a USBFI1 interrupt. With the initial value, each of the interrupt source flags in the interrupt flag register 1 is selected as a USBFI0 interrupt.

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	EP3 TR IS	1	R/W	EP3 TR Interrupt Select
1	EP3 TS IS	1	R/W	EP3 TS Interrupt Select
0	VBUSF IS	1	R/W	VBUSF Interrupt Select

25.3.8 Interrupt Select Register 2 (ISR2)

ISR2 selects the interrupt requests to the INTC to be indicated in interrupt flag register 2. When a bit in ISR2 is cleared to 0, the corresponding interrupt is requested as a USBFI0 interrupt. When a bit is set to 1, the corresponding interrupt is requested as a USBFI1 interrupt. With the initial value, each of the interrupt source flags in the interrupt flag register 2 is selected as a USBFI0 interrupt.

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	SURSE IS	1	R/W	SURSE Interrupt Select
3	CFDN IS	1	R/W	CFDN Interrupt Select
2	SOFE IS	1	R/W	SOFE Interrupt Select
1	SETCE IS	1	R/W	SETCE Interrupt Select
0	SETIE IS	1	R/W	SETIE Interrupt Select

25.3.9 Interrupt Select Register 3 (ISR3)

ISR3 selects the interrupt requests to the INTC to be indicated in interrupt flag register 3. When a bit in ISR3 is cleared to 0, the corresponding interrupt is requested as a USBFI0 interrupt. When a bit is set to 1, the corresponding interrupt is requested as a USBFI1 interrupt. With the initial value, each of the interrupt source flags in the interrupt flag register 3 is selected as a USBFI0 interrupt.

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	EP5 TR IS	0	R/W	EP5 TR Interrupt Select
2	EP5 TS IS	0	R/W	EP5 TS Interrupt Select
1	EP4 TF IS	0	R/W	EP4 TF Interrupt Select
0	EP4 TS IS	0	R/W	EP4 TS Interrupt Select

25.3.10 Interrupt Select Register 4 (ISR4)

ISR4 selects the interrupt requests to the INTC to be indicated in interrupt flag register 4. When a bit in ISR4 is cleared to 0, the corresponding interrupt is requested as a USBFI0 interrupt. When a bit is set to 1, the corresponding interrupt is requested as a USBFI1 interrupt. With the initial value, each of the interrupt source flags in the interrupt flag register 4 is selected as a USBFI0 interrupt.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TMOUT IS	0	R/W	TMOUT Interrupt Select

25.3.11 Interrupt Enable Register 0 (IER0)

IER0 enables the interrupt requests of the interrupt flag register 0. When an interrupt flag is set to 1 while the corresponding bit of each interrupt is set to 1, the interrupt request set in the interrupt select register 0 is issued.

When an interrupt flag is set to 1 while the corresponding bit of each interrupt is set to 1, the INTN pin set in the interrupt select register 0 is asserted low and an interrupt request is issued.

Bit	Bit Name	Initial Value	R/W	Description
7	BRST IE	0	R/W	BRST Interrupt Enable
6	EP1 FULL IE	0	R/W	EP1 FULL Interrupt Enable
5	EP2 TR IE	0	R/W	EP2 TR Interrupt Enable
4	EP2 EMPTY IE	0	R/W	EP2 EMPTY Interrupt Enable
3	SETUP TS IE	0	R/W	SETUP TS Interrupt Enable
2	EP0o TS IE	0	R/W	EP0o TS Interrupt Enable
1	EP0i TR IE	0	R/W	EP0i TR Interrupt Enable
0	EP0i TS IE	0	R/W	EP0i TS Interrupt Enable

25.3.12 Interrupt Enable Register 1 (IER1)

IER1 enables the interrupt requests of the interrupt flag register 1. When an interrupt flag is set to 1 while the corresponding bit of each interrupt is set to 1, the interrupt request set in the interrupt select register 1 is issued.

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	EP3 TR IE	0	R/W	EP3 TR Interrupt Enable
1	EP3 TS IE	0	R/W	EP3 TS Interrupt Enable
0	VBUSF IE	0	R/W	VBUSF Interrupt Enable

25.3.13 Interrupt Enable Register 2 (IER2)

IER2 enables the interrupt requests of the interrupt flag register 2. When an interrupt flag is set to 1 while the corresponding bit of each interrupt is set to 1, the interrupt request set in the interrupt select register 2 is issued.

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	SURSE IE	0	R/W	SURSE Interrupt Enable
3	CFDN IE	0	R/W	CFDN Interrupt Enable
2	SOFE IE	0	R/W	SOFE Interrupt Enable
1	SETCE IE	0	R/W	SETCE Interrupt Enable
0	SETIE IE	0	R/W	SETIE Interrupt Enable

25.3.14 Interrupt Enable Register 3 (IER3)

IER3 enables the interrupt requests of the interrupt flag register 3. When an interrupt flag is set to 1 while the corresponding bit of each interrupt is set to 1, the interrupt request set in the interrupt select register 3 is issued.

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	EP5 TR IE	0	R/W	EP5 TR Interrupt Enable
2	EP5 TS IE	0	R/W	EP5 TS Interrupt Enable
1	EP4 TF IE	0	R/W	EP4 TF Interrupt Enable
0	EP4 TS IE	0	R/W	EP4 TS Interrupt Enable

25.3.15 Interrupt Enable Register 4 (IER4)

IER4 enables the interrupt requests of the interrupt flag register 4. When an interrupt flag is set to 1 while the corresponding bit of each interrupt is set to 1, the interrupt request set in the interrupt select register 4 is issued.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TMOUT IE	0	R/W	TMOUT Interrupt Enable

25.3.16 EP0i Data Register (EPDR0i)

EPDR0i is an 8-byte transmit FIFO buffer for endpoint 0. EPDR0i holds one packet of transmit data for control-in. Transmit data is fixed by writing one packet of data and setting EP0iPKTE in the trigger register. When an ACK handshake is returned from the host after the data has been transmitted, EP0iTS in interrupt flag register 0 is set. This FIFO buffer can be initialized by means of EP0iCLR in the FCLR0 register.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for control-in transfer

25.3.17 EP0o Data Register (EPDR0o)

EPDR0o is an 8-byte receive FIFO buffer for endpoint 0. EPDR0o holds endpoint 0 receive data other than setup commands. When data is received normally, EP0oTS in interrupt flag register 0 is set, and the number of receive bytes is indicated in the EP0o receive data size register. After the data has been read, setting EP0oRDFN in the trigger register enables the next packet to be received. This FIFO buffer can be initialized by means of BP0oCLR in the FCLR0 register.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	R	Data register for control-out transfer

25.3.18 EP0s Data Register (EPDR0s)

EPDR0s is a data register specifically for endpoint 0 setup command. EPDR0s holds 8-byte command data sent in the setup stage. However, only the command to be processed by a microprocessor (firmware) is received. The command data to be processed automatically by this module is not stored.

Since the setup command mast be received, previous data in the buffer is over written with new data. In other words, when the reception of data in the setup stage starts during read, reception has priority and read data is invalid.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	R	Data register for storing the setup command at the control-out transfer

Note: The EPDR0s register should be read in 8-byte units. If reading is stopped before it completes, data received in the subsequent setup stage is not read successfully.

25.3.19 EP1 Data Register (EPDR1)

EPDR1 is a 128-byte receive FIFO buffer for endpoint 1. EPDR1 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. The number of receive byte is displayed in the EP1 receive data size register. The buffer on read side can be received again by writing EP1RDFN in the trigger register to 1 after data is read. The receive data of this FIFO buffer can be transferred by DMA. This FIFO buffer can be initialized by means of EP1CLR in the FCLR0 register.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	R	Data register for interrupt transfer

25.3.20 EP2 Data Register (EPDR2)

EPDR2 is a 128-byte transmit FIFO buffer for endpoint 2. EPDR2 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. When transmit data is written to this FIFO buffer and EP2PKTE in the trigger register is set, one packet of transmit data is fixed, and the dual-FIFO buffer is switched over. Transmit data for this FIFO buffer can be transferred by DMA. This FIFO buffer can be initialized by means of EP2CLR in the FCLR0 register.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for endpoint 2 transfer

25.3.21 EP3 Data Register (EPDR3)

EPDR3 is an 8-byte transmit FIFO buffer for endpoint 3. EPDR4 holds one packet of transmit data for the interrupt transfer of endpoint 3. Transmit data is fixed by writing one packet of data and setting EP3PKTE in the trigger register. When an ACK handshake is returned from the host after the data has been transmitted, EP3TS in interrupt flag register 1 is set. This FIFO buffer can be initialized by means of EP3CLR in the FCLR0 register.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for endpoint 3 transfer

25.3.22 EP4 Data Register (EPDR4)

EPDR4 is a 128-byte receive FIFO buffer for endpoint 4. EPDR4 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. The number of receive byte is displayed in the EP4 receive data size register. The receive data is fixed when an SOF packet is received. Accordingly, all receive data must be read until the next SOF packet is received. When the next SOF packet is received, the FIFO side is automatically switched over, and the previous data will not be possible to be read. This FIFO buffer can be initialized by means of EP4CLR in the FCLR1 register.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	R	Data register for endpoint 4 transfer

25.3.23 EP5 Data Register (EPDR5)

EPDR5 is a 128-byte transmit FIFO buffer for endpoint 5. EPDR5 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. When transmit data is written to this FIFO buffer and an SOF packet is received, one packet of transmit data is fixed, and the dual-FIFO buffer is switched over. This FIFO buffer can be initialized by means of EP5CLR and EP5CCLR in the FCLR1 register. (EP5CLR initializes both FIFOs and EP5CCLR initializes one FIFO which is connected to the CPU.)

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for endpoint 5 transfer

25.3.24 EP0o Receive Data Size Register (EPSZ0o)

EPSZ0o is a receive data size resister for endpoint 0o. EPSZ0o indicates the number of bytes received from the host.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Number of receive data for endpoint 0

25.3.25 EP1 Receive Data Size Register (EPSZ1)

EPSZ1 is a receive data size resister for endpoint 1. EPSZ1 indicates the number of bytes received from the host. FIFO of endpoint 1 has a dual-buffer configuration. The size of the received data indicated by this register is the size of the currently selected side (can be read by CPU).

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Number of received bytes for endpoint 1

25.3.26 EP4 Receive Data Size Register (EPSZ4)

EPSZ4 is a receive data size resister for endpoint 4. EPSZ4 indicates the number of bytes received from the host. FIFO of endpoint 4 has a dual-buffer configuration. The size of the received data indicated by this register is the size of the currently selected side (can be read by CPU).

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Number of received bytes for endpoint 4

25.3.27 Trigger Register (TRG)

TRG generates one-shot triggers FIFO for each endpoint of EP0s, EP0i, EP0o, EP1, EP2, and EP3. The packet enable trigger for the IN FIFO register and read complete trigger for the OUT FIFO register are triggers to be given.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	W	Reserved The write value should always be 0.
6	EP3 PKTE	0	W	EP3 Packet Enable
5	EP1 RDFN	0	W	EP1 Read Complete
4	EP2 PKTE	0	W	EP2 Packet Enable
3	—	0	W	Reserved The write value should always be 0.
2	EP0s RDFN	0	W	EP0s Read Complete
1	EP0o RDFN	0	W	EP0o Read Complete
0	EP0i PKTE	0	W	EP0i Packet Enable

25.3.28 Data Status Register (DASTS)

DASTS indicates whether the IN FIFO data register contains valid data. DASTS is set to 1 when data written to IN FIFO is enabled by writing PKTE in TRG to 1, and cleared when all data has been transmitted to the host. In case of a dual-configuration FIFO for endpoint 2, this bit is cleared to 0 when both sides are empty.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0.
5	EP3 DE	0	R	EP3 Data Enable
4	EP2 DE	0	R	EP2 Data Enable
3 to 1	—	All 0	R	Reserved These bits are already read as 0.
0	EP0iDE	0	R	EP0i data enable

25.3.29 FIFO Clear Register 0 (FCLR0)

FCLR is a one shot register to clear the FIFO buffers for endpoints 0 to 3. Writing 1 to a bit clears the data in the corresponding FIFO buffer.

In case of reception FIFO, by writing data in the FIFO buffer, the data by which PKTE in TRG is not written to 1 and the data enabled by writing 1 can be cleared. In case of OUT FIFO, the data of which reception has not been completed can be cleared.

Both sides of the dual-configuration FIFO buffers (EP1 or EP3) can be cleared.

The corresponding interrupt flag is not cleared by this clear instruction. Do not clear a FIFO buffer during transmission and reception.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	W	Reserved The write value should always be 0.
6	EP3 CLR	—	W	EP3 Clear
5	EP1 CLR	—	W	EP1 Clear
4	EP2 CLR	—	W	EP2 Clear

Bit	Bit Name	Initial Value	R/W	Description
3, 2	—	—	W	Reserved The write value should always be 0.
1	EP0o CLR	—	W	EP0o Clear
0	EP0i CLR	—	W	EP0i Clear

25.3.30 FIFO Clear Register 1 (FCLR1)

FCLR is a one shot register to clear the FIFO buffers for endpoints 4 and 5. Writing 1 to a bit clears the data in the corresponding FIFO buffer.

The corresponding interrupt flag is not cleared by this clear instruction. Do not clear a FIFO buffer during transmission and reception.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	—	W	Reserved The write value should always be 0.
4	EP5 CCLR	—	W	EP5 CPU Clear
3, 2	—	—	W	Reserved The write value should always be 0.
1	EP5 CLR	—	W	EP5 Clear
0	EP4 CLR	—	W	EP4 Clear

25.3.31 DMA Transfer Setting Register (DMA)

DMA is set when the dual address transfer is used to the data register for endpoints 1 and 2 to which transfer is possible by DMA. The USB1_pwr_en pin level can be controlled by the bit 2.

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	PULLUP E	0	R/W	Pull-up Enable Controls connection notification to USB host/hub. 0: USB1_pwr_en pin goes high 1: USB1_pwr_en pin goes low

Bit	Bit Name	Initial Value	R/W	Description
1	EP2 DMAE	0	R/W	EP2DMA Enable Enables DMA transfer for EP2.
0	EP1 DMAE	0	R/W	EP1DMAE Enable Enables DMA transfer for EP1.

25.3.32 Endpoint Stall Register 0 (EPSTL0)

EPSTL stalls each endpoint. The endpoint in which the stall bit is set to 1 returns a stall handshake to the host from the next transfer when 1 is written to. The stall bit for endpoint 0 is cleared automatically on reception of 8 byte command data for which decoding is performed by the function and the EP0 STL bit is cleared. When the SETUPTS flag bit in the IFR0 register is set to 1, a write of the EP0 STL bit to 1 is ignored. For detailed operation, see section 25.8, Stall Operations.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	EP3 STL	0	R/W	EP3 Stall Sets EP3 stall
2	EP2 STL	0	R/W	EP2 Stall Sets EP2 stall
1	EP1 STL	0	R/W	EP1 Stall Sets EP1 stall
0	EP0 STL	0	R/W	EP0 Stall Sets EP0 stall

25.3.33 Endpoint Stall Register 1 (EPSTL1)

EPSTL stalls each endpoint. The endpoint in which the stall bit is set to 1 returns a stall handshake to the host from the next transfer when 1 is written to. For detailed operation, see section 25.8, Stall Operations.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	EP5 STL	0	R/W	EP5 Stall Sets EP5 stall
0	EP4 STL	0	R/W	EP4 Stall Sets EP4 stall

25.3.34 Configuration Value Register (CVR)

CVR is a register to store the Configuration/Interface/ value to be set when the Set Configuration/Set Interface command is normally received.

Bit	Bit Name	Initial Value	R/W	Description
7	CNFV1	0	R	Configuration Value
6	CNFV0	0	R	The configuration setting value is stored when the Set Configuration command has been received. CNFV is updated when the SETC bit in the interrupt flag register is set to 1.
5	INTV1	0	R	Interface Value
4	INTV0	0	R	The interface setting value is stored when the Set Interface command has been received. INTV is updated when the SETI bit in the interrupt flag register is set to 1.
3	—	0	R	Reserved This bit is always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
2	ALTV2	0	R	Alternate Value
1	ALTV1	0	R	The alternate setting value is stored when the Set interface command has been received.
0	ALTV0	0	R	ALTV is updated when the SETI bit in the interrupt flag register is set to 1.

25.3.35 Time Stamp Register (TSRH/TSRL)

TSR is a register to store the current time stamp value. The time stamp is updated when the SOF bit in IFR0 is set to 1. The value of the time stamp when the SOF mark function is enabled and the SOF packet is broken remains as previous one.

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved. This bit is always read as 0.
10	D10	0	R	Time Stamp Data
9	D9	0	R	
8	D8	0	R	
7	D7	0	R	
6	D6	0	R	
5	D5	0	R	
4	D4	0	R	
3	D3	0	R	
2	D2	0	R	
1	D1	0	R	
0	D0	0	R	

Note: The time stamp register is used as a 16-bit register which consists of upper byte TSRH and lower TSRL in USBF. TSRH can be read directly, but TSRL is read via an 8-bit temporary register. Therefore, the registers should be accessed in the order, TSRH and TSRL, in byte units. TSRL cannot be read singly.

25.3.36 Control Register 0 (CTLR0)

CTLR0 sets functions of ASCE, PWMD, RSME, and RWUP.

Bit	Bit name	Initial value	R/W	Description
7 to 5	c	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RWUPS	0	R	Remote Wakeup Status Status bit to indicate that the remote wakeup from the host is enabled/disabled. Indicates 0 when the remote wakeup is disabled with Device Remote Wakeup by the Set Feature/Clear Feature request and indicates 1 when it is enabled.
3	RSME	0	R/W	Resume Enable Bit to clear the suspend state (performs the remote wakeup) When this bit is written to 1, a resume register is set. When this bit will be used, be sure to hold to 1 for one clock or more at 12 MHz in minimum and then clear to 0 again.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	ASCE	0	R/W	Automatic Stall Clear Enable When this bit is set to 1, the stall handshake is returned to the host and the stall setting bit (EPSTLR/EPXSTL) of the returned endpoint is automatically cleared. Control in a unit of endpoint is disabled as this bit is common for all endpoints. When this bit is set to 0, be sure to clear the stall setting bit of each endpoint by using software. This bit should be set to 1 before each stall bit in EPSTL is set to 1.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

25.3.37 Control Register 1 (CTLR1)

CTLR1 makes settings of internal timer which is used in the isochronous transfer.

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	TMR ACLR	1	R/W	Timer Auto Clear Selects method to clear TMR (timer register). 0: Not cleared. When clearing TMR, write 0 to TMR by CPU. 1: Automatically cleared every time when SOF is received.
0	TMR EN	0	R/W	Timer Enable TMR EN is TMR (timer register) enable bit. 0: Timer operation is disabled 1: Timer operation is enabled

25.3.38 Endpoint Information Register (EPIR)

EPIR is a register to set the configuration information for each endpoint. 5 bytes of the information are required for one endpoint and the formats are listed in tables 25.3 and 25.4. Write the data in order from endpoint 0. Do not write more than 5 (bytes) × 10 (endpoints) = 50 bytes. Write this information once at power-on reset. Do not write it again afterwards.

Write data of one endpoint is described below. EPIR writes data in the same address in order. Therefore though there is only one EPIR register, write data for registration number N (N is from 0 to 9) is listed as EPIRN0 to EPIRN4 (EPIR [registration number] [write order]) for the purpose of explaining. Write data in order from EPIR00.

- EPIRN0:

Bit	Bit Name	Initial value	R/W	Description
7 to 4	D7 to D4	Undefined	W	Endpoint Number Settable range: 0 to 5
3	D3	Undefined	W	Configuration Number to which Endpoint Belongs
2	D2			Settable range: 0 or 1
1	D1	Undefined	W	Interface Number to which Endpoint Belongs
0	D0			Settable range: 0 to 3

- EPIRN1:

Bit	Bit Name	Initial value	R/W	Description
7	D7	Undefined	W	Alternate Number to which Endpoint Belongs
6	D6			Settable range: 0 or 1
5	D5	Undefined	W	Transfer Method of Endpoint
4	D4			Settable range: 0: Control 1: Isochronous 2: Bulk 3: Interrupt
3	D3	Undefined	W	Transfer Direction of Endpoint Settable range: 0: Out 1: In
2 to 0	D2 to D0	Undefined	W	Reserved The write value should always be 0.

- EPIRN2:

Bit	Bit Name	Initial value	R/W	Description
7 to 1	D7 to D1	Undefined	W	Maximum Packet Size of Endpoint Settable range: 0 to 64
0	D0	Undefined	W	Reserved The write value should always be 0.

- EPIRN3:

Bit	Bit Name	Initial value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Reserved The write value should always be 0.

- EPIRN4:

Bit	Bit Name	Initial value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Endpoint FIFO Number Settable range: 0 to 5

An endpoint number is an endpoint number used by the USB host. The endpoint FIFO number corresponds to the endpoint number which is described in this manual. When each endpoint number and endpoint FIFO number corresponds to each other, transfer can be performed between the USB host and the endpoint FIFO. Note that the setting values are limited as described below.

- Since each endpoint FIFO is optimized by a dedicated hardware corresponding to each transfer method, transfer direction, and maximum packet size, set the endpoint FIFO with a transfer method, transfer direction, and maximum packet size shown in the table below.

  Example: Endpoint FIFO number 1 cannot be set as other than bulk transfer, OUT, and maximum packet size (64 bytes). Although endpoint FIFO number 4 cannot be set as other than isochronous transfer and OUT, maximum packet size can be set in the range of 0 to 64 bytes.
- Endpoint 0 and endpoint FIFO number 0 must correspond.
- The maximum packet size of endpoint FIFO number 0 can be set to 8 bytes only.
- The setting value of endpoint FIFO number 0 can be set to the maximum packet size only and the rest data is all 0.
- The maximum packet size of endpoint FIFO numbers 1 and 2 can be set to 64 only.
- The maximum packet size of endpoint FIFO numbers 3 can be set to 8 only.
- The maximum packet size of endpoint FIFO numbers 4 and 5 can be set in the range of 0 to 64.
- When the isochronous transfer is set, Alternate can be used in the range of 0 and 1 for the same endpoint. Be sure to allocate the Alternate to the same endpoint FIFO number.
- Endpoint information can be set up to 10 in maximum.
- Endpoint information of 10 pieces must be written.
- All information of endpoints which are not used must be written as 0.

A list of restrictions of settable transfer method, transfer direction, and maximum packet size is described in table 25.2.

Table 25.2 Restrictions of Settable Values

Endpoint FIFO No.	Maximum Packet Size	Transfer Method	Transfer Direction
0	8 bytes	Control	—
1	64 bytes	Bulk	OUT
2	64 bytes	Bulk	IN
3	8 bytes	Interrupt	IN
4	0 to 64 bytes	Isochronous	OUT
5	0 to 64 bytes	Isochronous	IN

- Example of Setting

  This is an example when endpoint 4 and 5 used for the isochronous transfer are allocated with Alternate value.

Table 25.3 Example of Endpoint Configuration

EP No.	Conf.	Int.	Alt.	Transfer Method	Transfer Direction	Maximum Packet Size	EP FIFO No.
0	—	—	—	Control	IN/OUT	8 bytes	0
1	1	0	0	Bulk	OUT	64 bytes	1
2	1	0	0	Bulk	IN	64 bytes	2
3	1	0	0	Interrupt	IN	8 bytes	3
—	1	1	0	—	—	—	—
—	1	1	1	—	—	—	—
4	1	2	0	Isochronous	OUT	0 bytes	4
4	1	2	1	Isochronous	OUT	64 bytes	4
5	1	3	0	Isochronous	IN	0 bytes	5
5	1	3	1	Isochronous	IN	64 bytes	5

Table 25.4 Example of Setting of Endpoint Configuration Information

N	EPIR[N]0	EPIR[N]1	EPIR[N]2	EPIR[N]3	EPIR[N]4
0	00	00	10	00	00
1	14	20	80	00	01
2	24	28	80	00	02
3	34	38	10	00	03
4	00	00	00	00	00
5	00	00	00	00	00
6	46	10	00	00	04
7	46	50	80	00	04
8	67	18	00	00	05
9	57	58	80	00	05

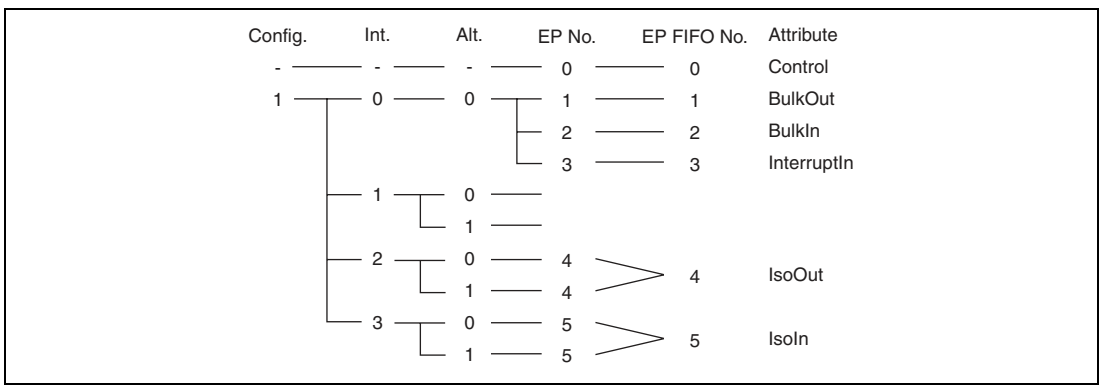


Figure 25.2 Example of Endpoint Configuration

25.3.39 Timer Register (TMRH/TMRL)

TMRHWMRL is a 16-bit timer which is operated with a peripheral clock ϕ. Measuring the SOF packet reception interval enables the SOF packet break to be detected.

The timer is operated, stopped, and cleared according to the settings of the control register 1 (CTLR1).

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	D15 to D0	0	R/W	Count Value

Note: The timer register is used as a 16-bit register which consists of upper byte TMRH and lower TMRL in USBF. TMRH can be read directly, but TMRL is read via an 8-bit temporary register. Therefore, the registers should be read in the order, TMRH and TMRL, in byte units. TMRL cannot be read singly.

25.3.40 Set Time Out Register (STOH/STOL)

STOH/STOL specifies the time out value of the timer register. When the count value of the timer register reaches the specified time out value, the time out interrupt flag in the interrupt flag register 4 is set.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	D15 to D0	0	R/W	Specified Time Out Value

Note: The timer register is used as a 16-bit register which consists of upper byte STOH and lower STOL in USBF. STOH can be read directly, but STOL is read via an 8-bit temporary register. Therefore, the registers should be read in the order, STOH and STOL, in byte units. TMRL cannot be read singly.

25.4 Operation

25.4.1 Cable Connection

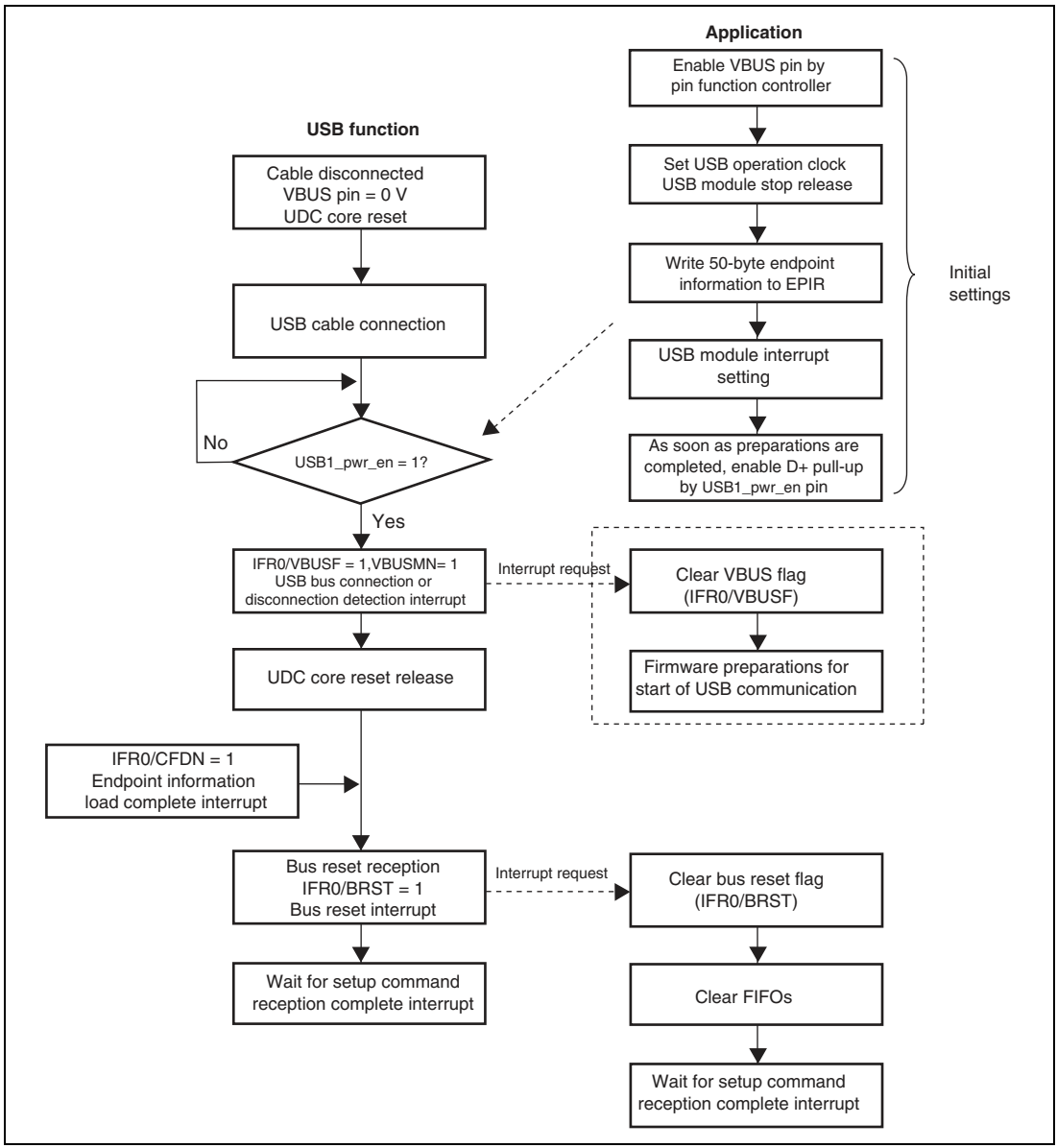


Figure 25.3 Cable Connection Operation

In applications that do not require USB cable connection to be detected, processing by the USB connection or disconnection detection interrupt is not necessary. Preparations should be made with the bus reset interrupt.

Also, in applications that require connection detection regardless of D+ pull-up control, detection should be carried out using IRQ or a general input port.

25.4.2 Cable Disconnection

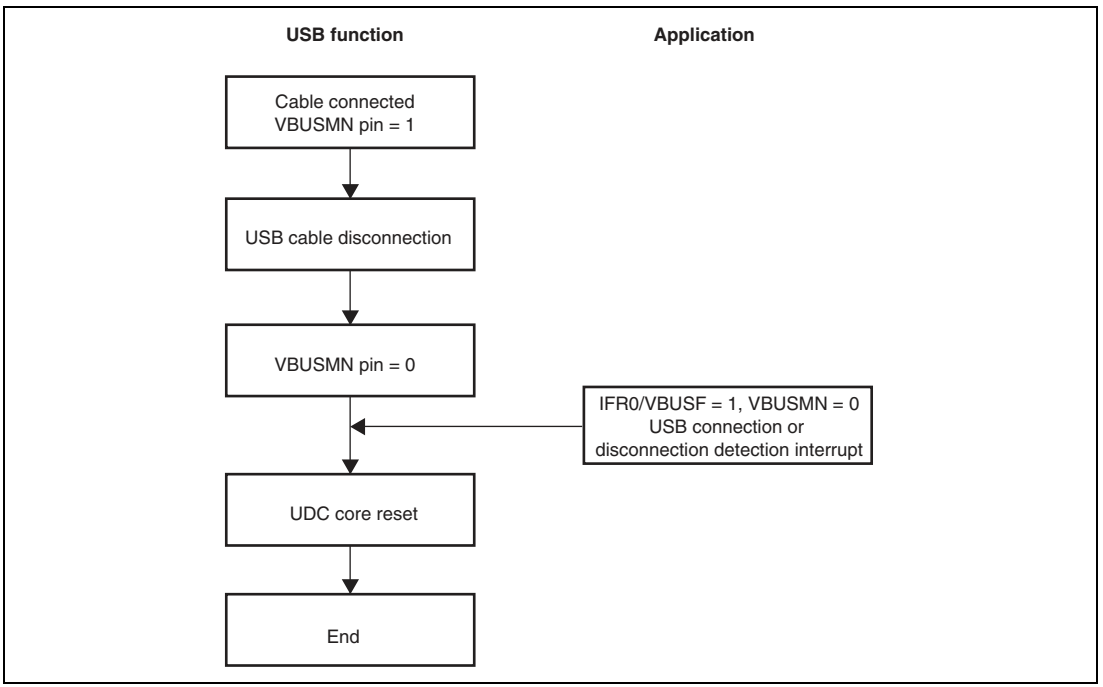


Figure 25.4 Cable Disconnection Operation

In applications that require connection/disconnection detection regardless of D+ pull-up control, detection should be carried out using IRQ or a general input port.

25.4.3 Control Transfer

Control transfer consists of three stages: setup, data (not always included), and status (figure 25.6). The data stage comprises several bus transactions. Operation flowcharts for each stage are shown below.

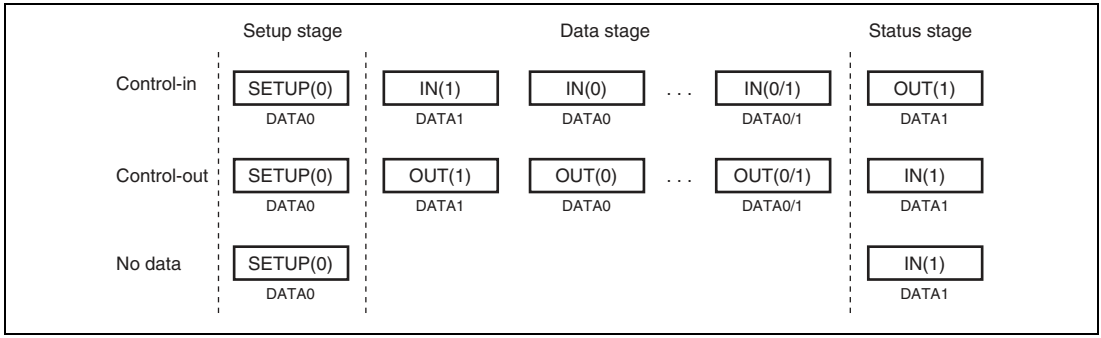
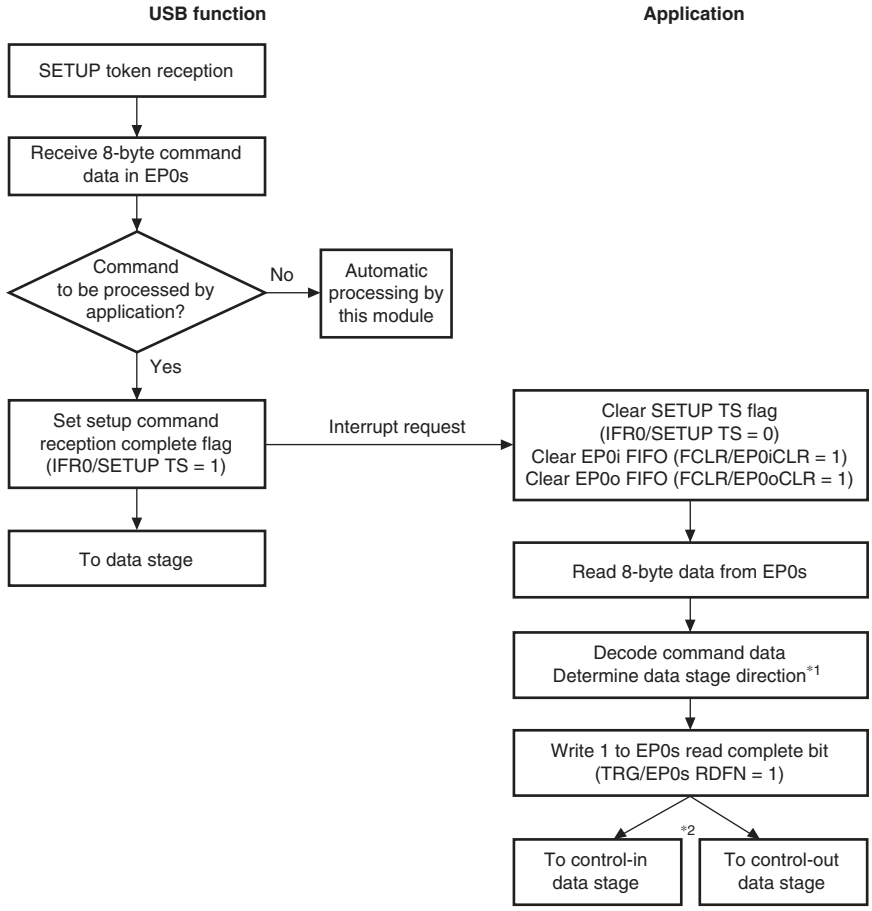


Figure 25.5 Transfer Stages in Control Transfer

- Setup Stage



Notes: 1 In the setup stage, the application analyzes command data from the host requiring processing by the application, and determines the subsequent processing (for example, data stage direction, etc.).
2 When the transfer direction is control-out, the EP0i transfer request interrupt required in the status stage should be enabled here. When the transfer direction is control-in, this interrupt is not required and should be disabled.

Figure 25.6 Setup Stage Operation

- Data Stage (Control-In)

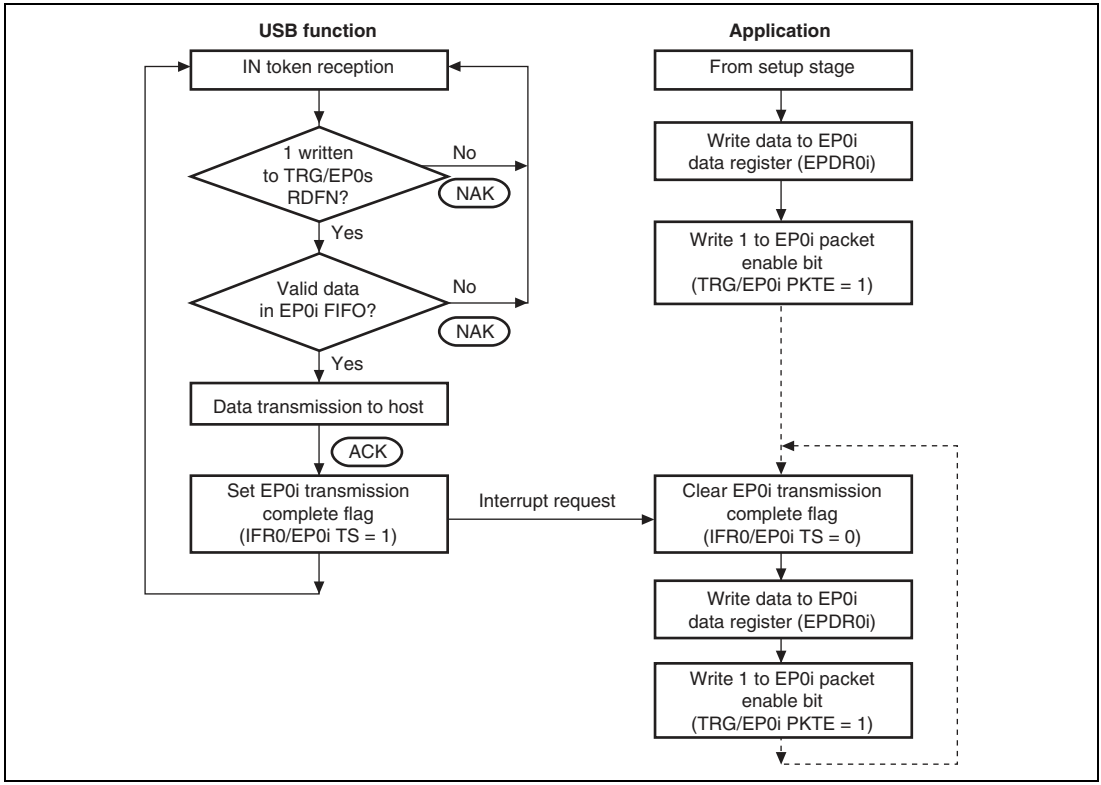


Figure 25.7 Data Stage (Control-In) Operation

The application first analyzes command data from the host in the setup stage, and determines the subsequent data stage direction. If the result of command data analysis is that the data stage is in-transfer, one packet of data to be sent to the host is written to the FIFO. If there is more data to be sent, this data is written to the FIFO after the data written first has been sent to the host (IFR0/EP0i TS = 1).

The end of the data stage is identified when the host transmits an OUT token and the status stage is entered.

Note: If the size of the data transmitted by the function is smaller than the data size requested by the host, the function indicates the end of the data stage by returning to the host a packet shorter than the maximum packet size. If the size of the data transmitted by the function is an integral multiple of the maximum packet size, the function indicates the end of the data stage by transmitting a zero-length packet.

- Data Stage (Control-Out)

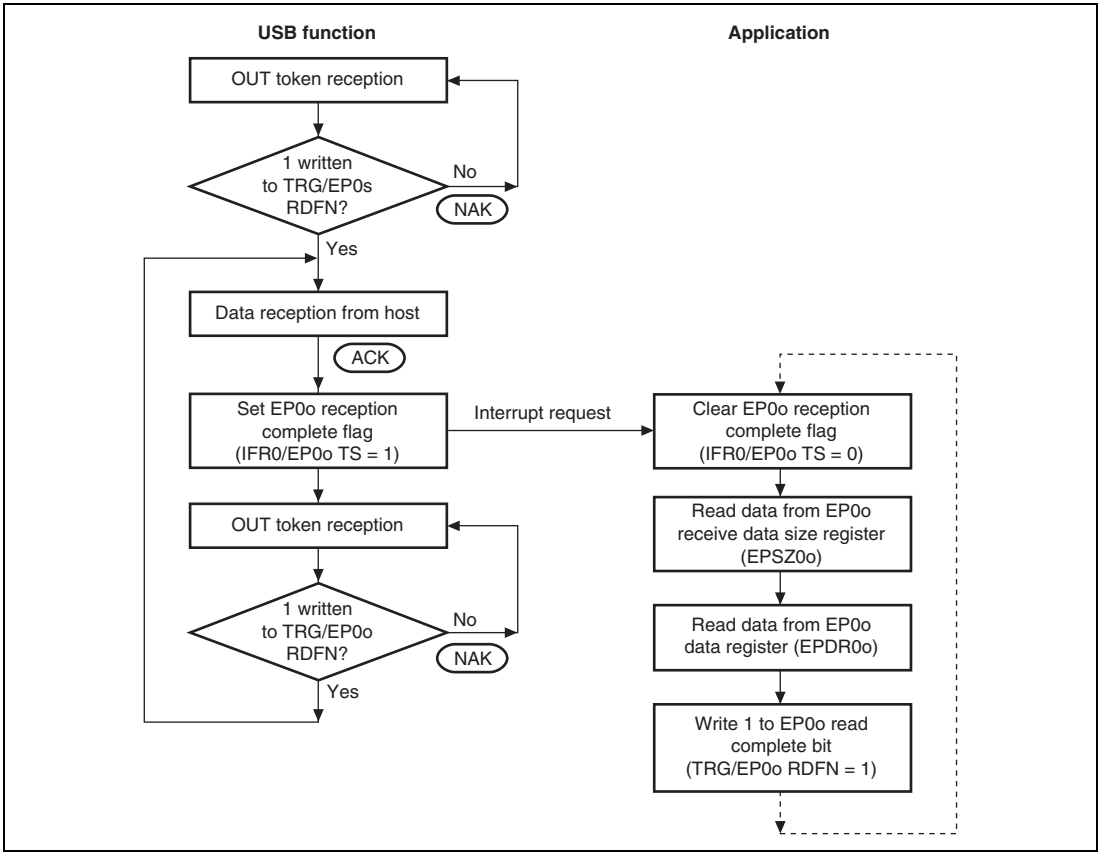


Figure 25.8 Data Stage (Control-Out) Operation

The application first analyzes command data from the host in the setup stage, and determines the subsequent data stage direction. If the result of command data analysis is that the data stage is out-transfer, the application waits for data from the host, and after data is received (IFR0/EP0o TS = 1), reads data from the FIFO. Next, the application writes 1 to the EP0o read complete bit, empties the receive FIFO, and waits for reception of the next data.

The end of the data stage is identified when the host transmits an IN token and the status stage is entered.

- Status Stage (Control-In)

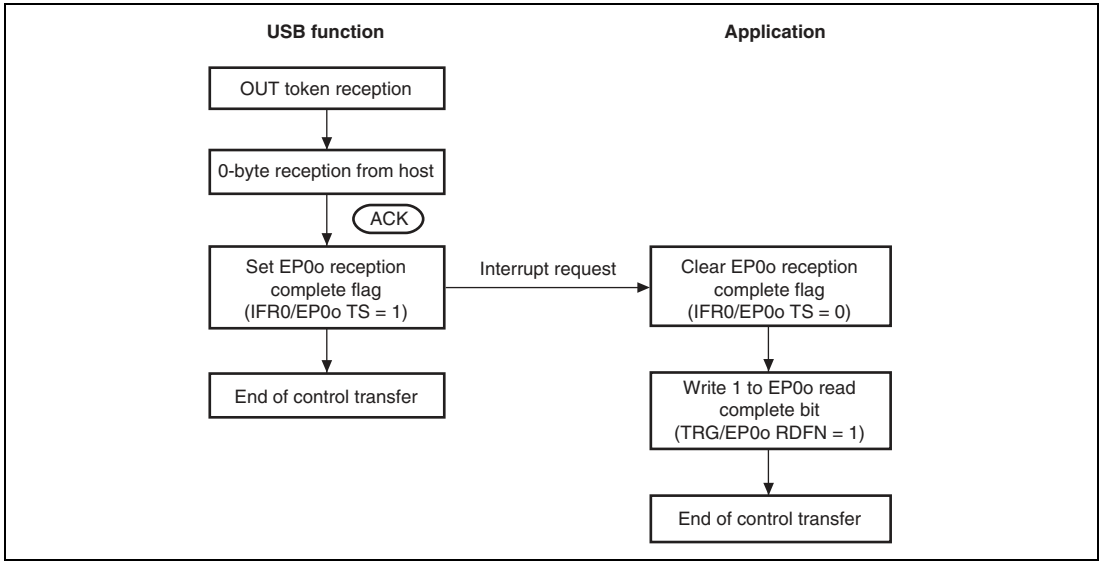


Figure 25.9 Status Stage (Control-In) Operation

The control-in status stage starts with an OUT token from the host. The application receives 0-byte data from the host, and ends control transfer.

- Status Stage (Control-Out)

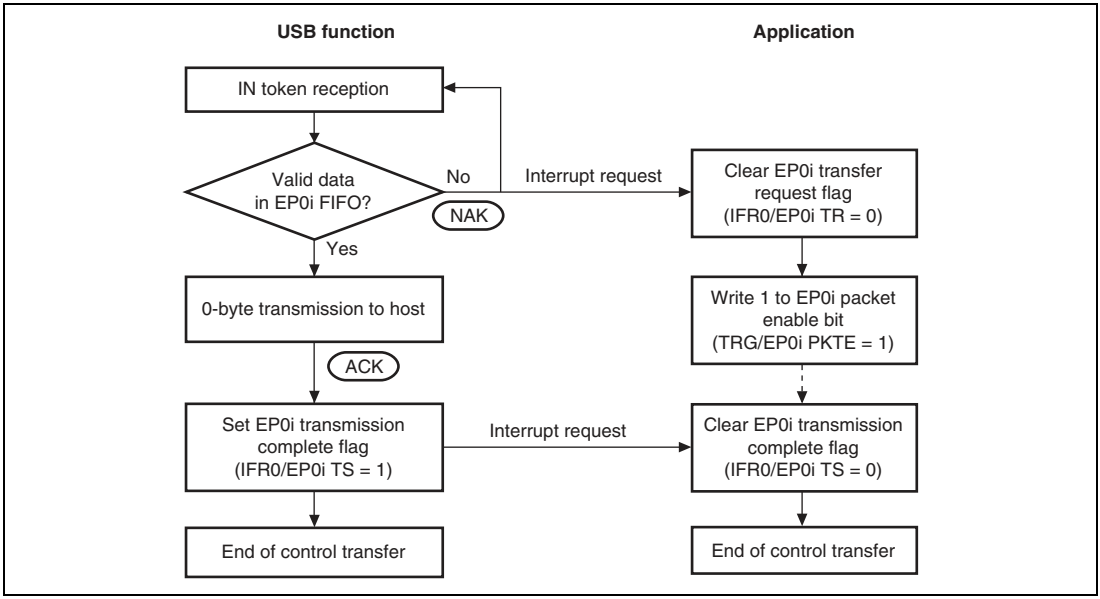


Figure 25.10 Status Stage (Control-Out) Operation

The control-out status stage starts with an IN token from the host. When an IN-token is received at the start of the status stage, there is not yet any data in the EP0i FIFO, and so an EP0i transfer request interrupt is generated. The application recognizes from this interrupt that the status stage has started. Next, in order to transmit 0-byte data to the host, 1 is written to the EP0i packet enable bit but no data is written to the EP0i FIFO. As a result, the next IN token causes 0-byte data to be transmitted to the host, and control transfer ends.

After the application has finished all processing relating to the data stage, 1 should be written to the EP0i packet enable bit.

25.4.4 EP1 Bulk-Out Transfer (Dual FIFOs)

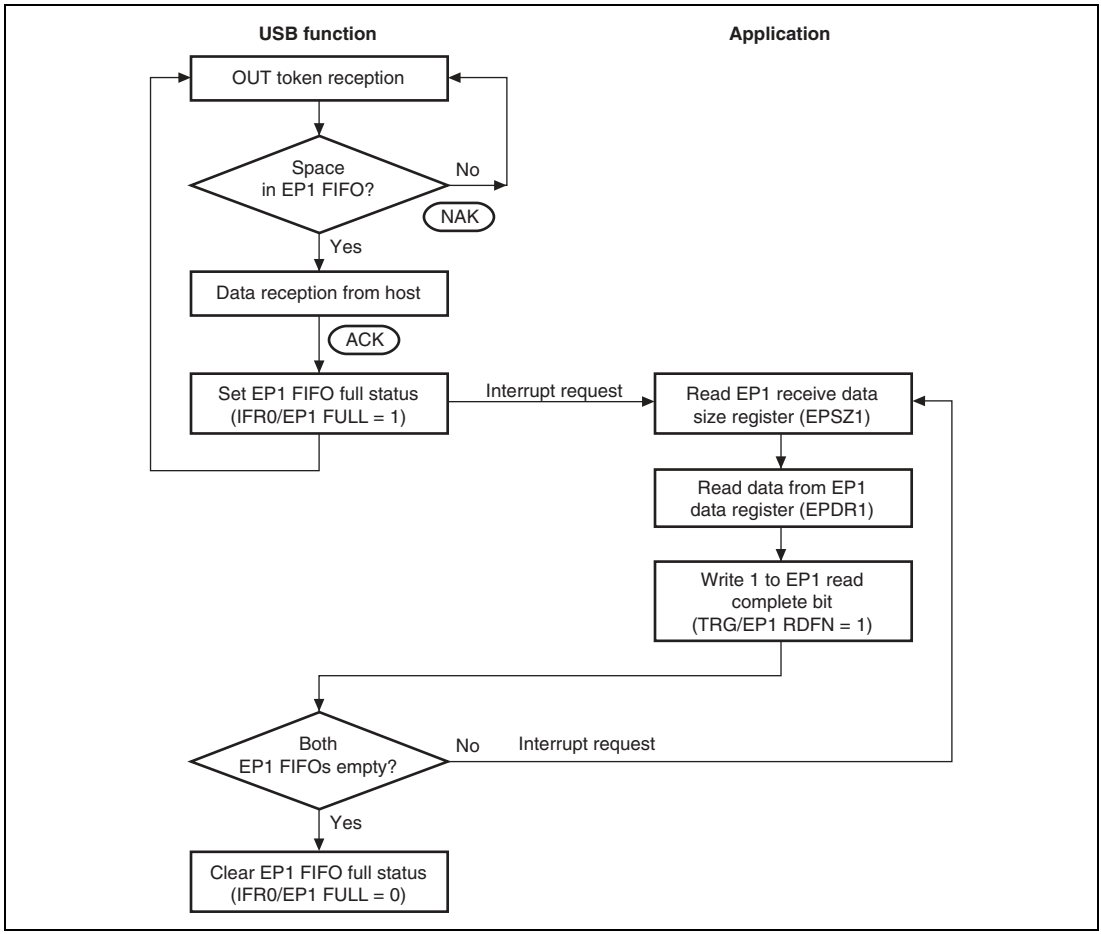


Figure 25.11 EP1 Bulk-Out Transfer Operation

EP1 has two 64-byte FIFOs, but the user can perform data reception and receive data reads without being aware of this dual-FIFO configuration.

When one FIFO is full after reception is completed, the IFR0/EP1 FULL bit is set. After the first receive operation into one of the FIFOs when both FIFOs are empty, the other FIFO is empty, and so the next packet can be received immediately. When both FIFOs are full, NACK is returned to the host automatically. When reading of the receive data is completed following data reception, 1 is written to the TRG/EP1 RDFN bit. This operation empties the FIFO that has just been read, and makes it ready to receive the next packet.

25.4.5 EP2 Bulk-In Transfer (Dual FIFOs)

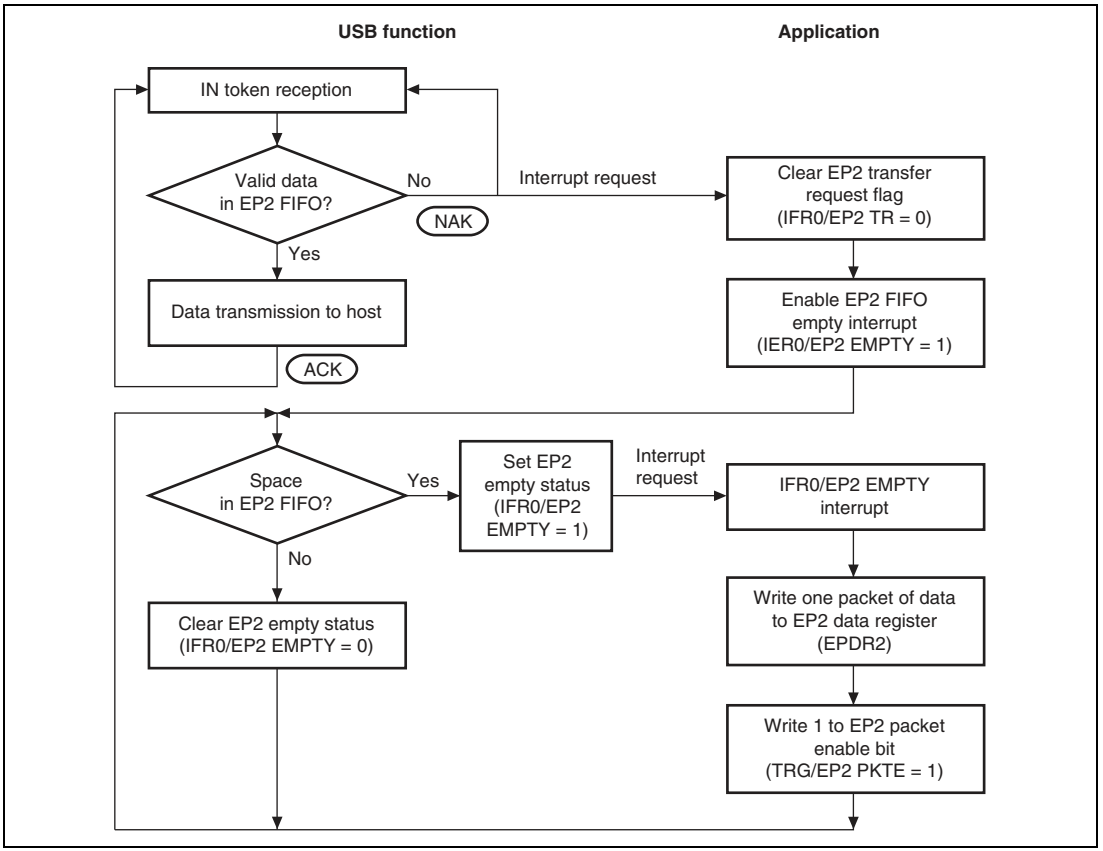


Figure 25.12 EP2 Bulk-In Transfer Operation

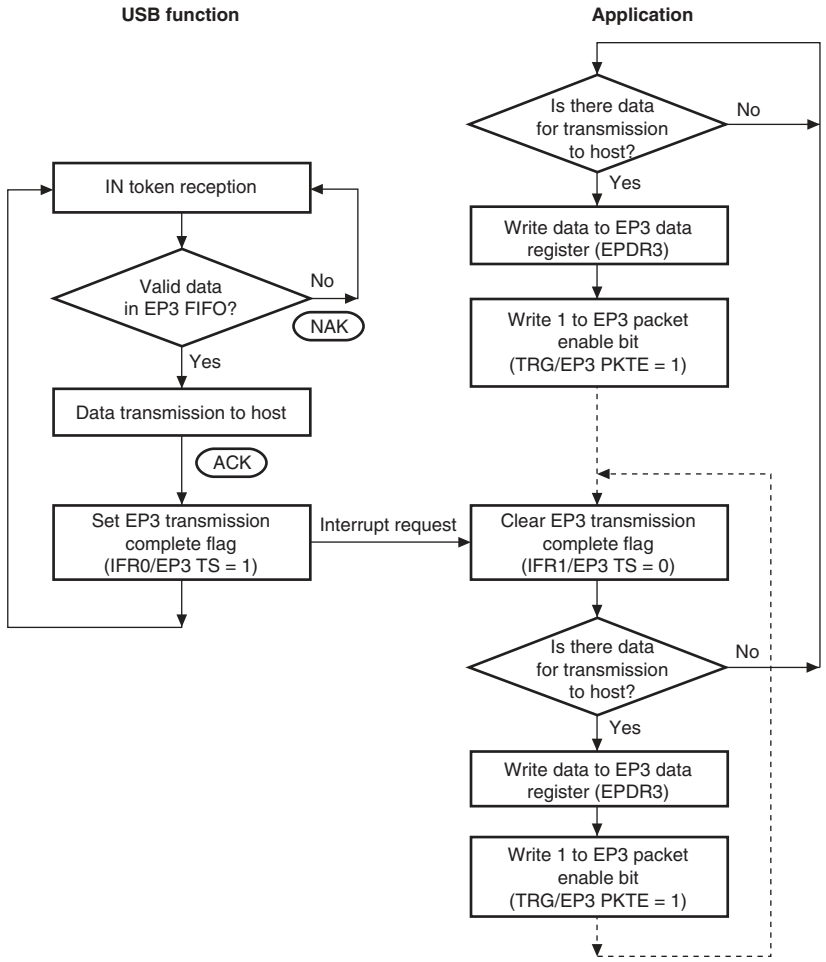
EP2 has two 64-byte FIFOs, but the user can perform data transmission and transmit data writes without being aware of this dual-FIFO configuration. However, one data write is performed for one FIFO. For example, even if both FIFOs are empty, it is not possible to perform EP2/PKTE at one time after consecutively writing 128 bytes of data. EP2/PKTE must be performed for each 64-byte write.

When performing bulk-in transfer, as there is no valid data in the FIFOs on reception of the first IN token, an IFR0/EP2 TR interrupt is requested. With this interrupt, 1 is written to the IER0/EP2 EMPTY bit, and the EP2 FIFO empty interrupt is enabled. At first, both EP2 FIFOs are empty, and so an EP2 FIFO empty interrupt is generated immediately.

The data to be transmitted is written to the data register using this interrupt. After the first transmit data write for one FIFO, the other FIFO is empty, and so the next transmit data can be written to the other FIFO immediately. When both FIFOs are full, EP2 EMPTY is cleared to 0. If at least one FIFO is empty, IFR0/EP2 EMPTY is set to 1. When ACK is returned from the host after data transmission is completed, the FIFO used in the data transmission becomes empty. If the other FIFO contains valid transmit data at this time, transmission can be continued.

When transmission of all data has been completed, write 0 to IER0/EP2 EMPTY and disable interrupt requests.

25.4.6 EP3 Interrupt-In Transfer



Note: This flowchart shows just one example of interrupt transfer processing. Other possibilities include an operation flow in which, if there is data to be transferred, the EP3 DE bit in the USB data status register is referenced to confirm that the FIFO is empty, and then data is written to the FIFO.

Figure 25.13 EP3 Interrupt-In Transfer Operation

25.5 EP4 Isochronous-Out Transfer

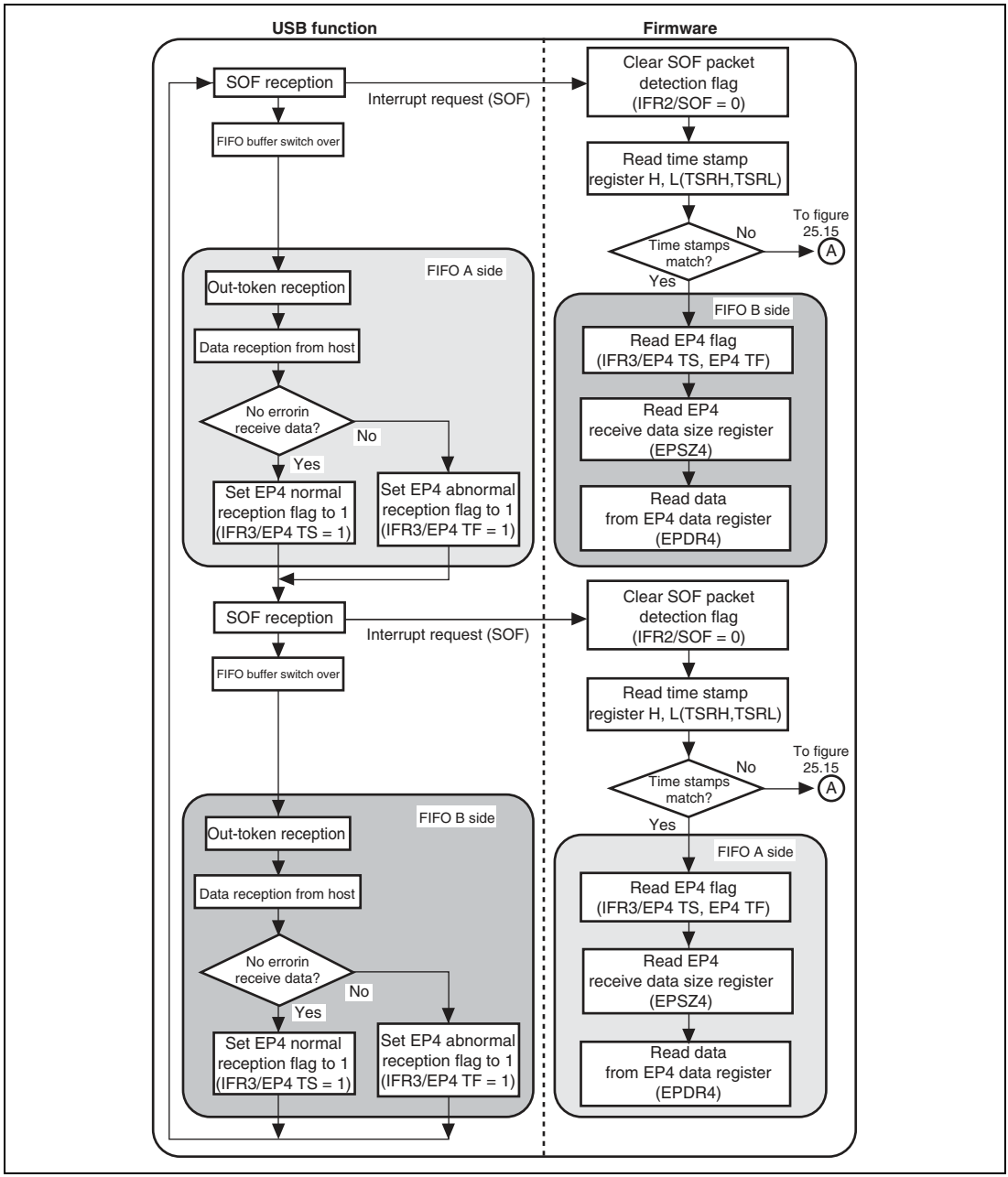


Figure 25.14 EP4 Isochronous-Out Transfer Operation (SOF is Normal)

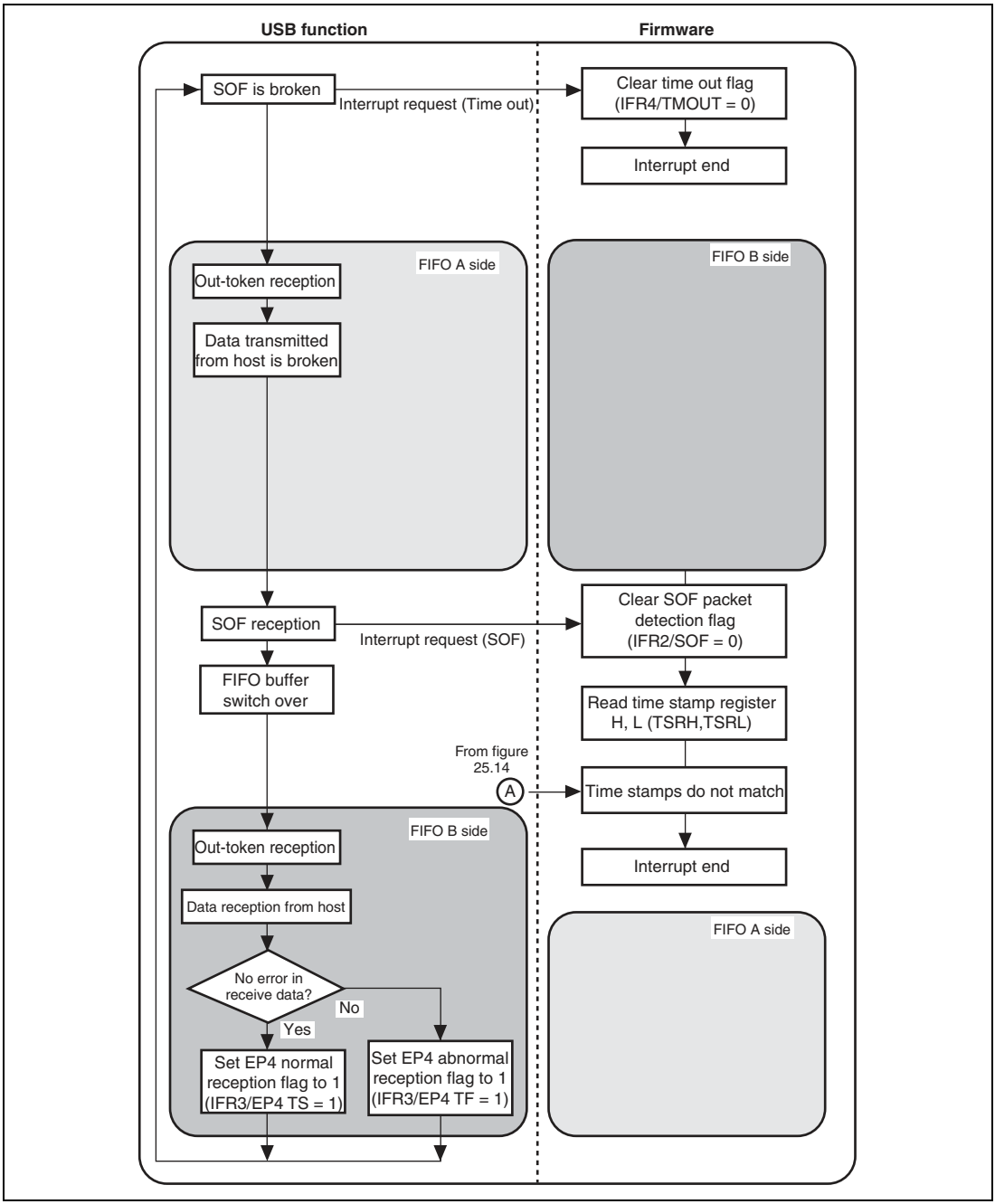


Figure 25.15 EP4 Isochronous-Out Transfer Operation (SOF is Broken)

Figure 25.14 shows the normal operation of the USB function and firmware in isochronous-out transfer.

EP4 has two up to 64-byte FIFOs, but the user can perform data transmission and read receive data without being aware of this dual-FIFO configuration.

In isochronous transfer, transfer occurs only once per one frame (1 ms). So, when SOF is received, the FIFO buffer is switched automatically with hardware.

FIFO buffers are switched over by the SOF reception. Therefore, the FIFO buffer in which the USB function receives the data from the host and the FIFO buffer in which the firmware reads the receive data have different buffers, and a read and write of FIFO buffer are not competed. Accordingly, the data read by the firmware is the data received in one frame before. The buffers of FIFOs are switched over automatically by the SOF reception, so reading of data must be completed within the frame.

The USB function receives data from the host after an out-token is received. If there is an error in the data, set the internal TF flag to 1. If there is no error in the data, set the internal TS flag to 1.

In firmware, first, the processing routine of the isochronous transfer is called by SOF interrupt to check the time stamp. Then data is read from the FIFO buffer. The flag information (TS, TF) is read and decided if the data has an error. The flag information at this time represents the status of the currently readable FIFO buffer.

SOF happens to be broken because of external cause during transmission from the host. In this case, an operation flow is different from that in figure 25.14. As an example, figure 25.14 shows the operation flow of a broken frame and a subsequent frame when SOF is broken once. When SOF is broken, the FIFO buffer is not switched in current frame, and a time out interrupt occurs after time set by user has been elapsed. The USB function controller discards the data which has been transmitted to the frame from the host.

The firmware detects the SOF break by the time out interrupt. In this case, the FIFO buffer connected to the CPU does not read data since data has already been read. When the SOF interrupt occurs in the subsequent frame, the processing routine of the isochronous transfer is called and the time stamps are compared. The time stamps do not much since the SOF break occurred in the previous frame. Data is not read since the data in FIFO is not current one.

25.6 EP5 Isochronous-In Transfer

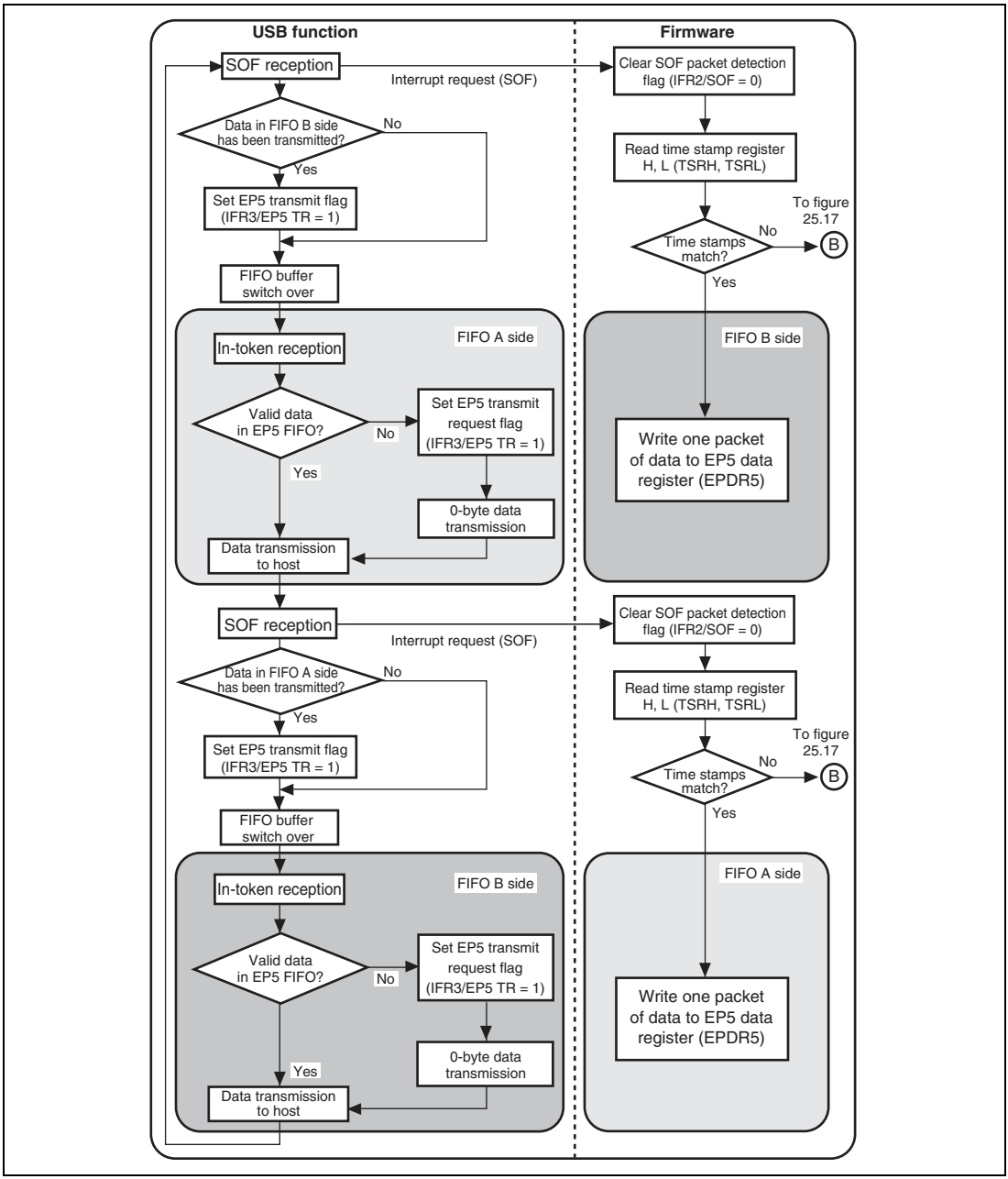


Figure 25.16 EP5 Isochronous-In Transfer Operation (SOF is Normal)

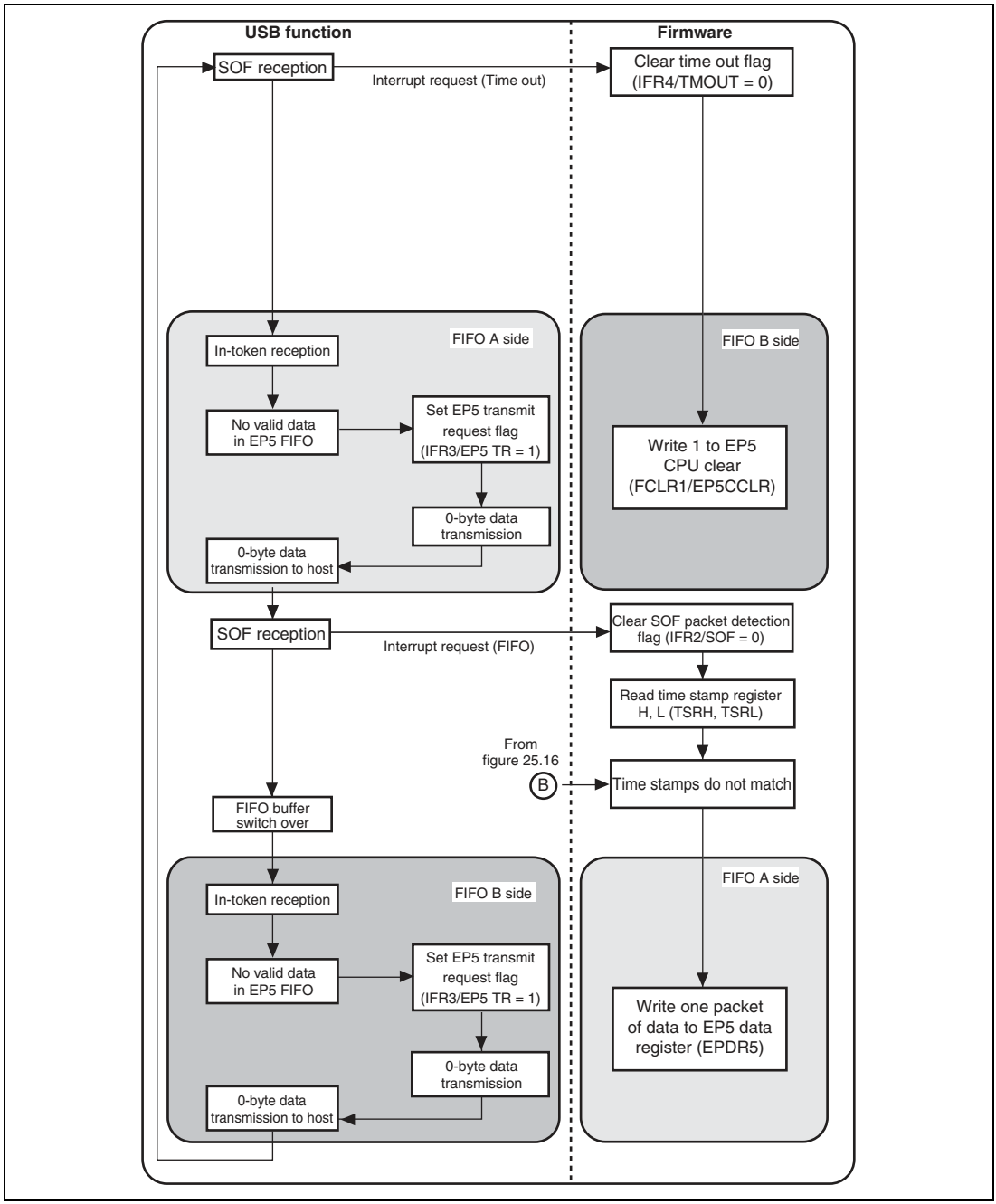


Figure 25.17 EP5 Isochronous-In Transfer Operation (SOF in Broken)

Figure 25.16 shows the normal operation of the USB function and firmware in isochronous-in transfer.

EP5 has two up to 64-byte FIFOs, but the user can perform data transmission and write transmit data without being aware of this dual-FIFO configuration.

In isochronous transfer, transfer occurs only once per one frame (1 ms). So, when SOF is received, the FIFO buffer is switched automatically with hardware.

FIFO buffers are switched over by the SOF reception. Therefore, the FIFO buffer in which the USB function transmits the data and the FIFO buffer in which the firmware writes the transmit data have different buffers, and a read and write of FIFO buffer are not competed. Accordingly, the data written by the firmware is the data transmitted in one frame after. The buffers of FIFOs are switched over automatically by the SOF reception, so writing of data must be completed within the frame.

The USB function transmits data to the host, and the internal TR flag is set to 1, when data to be transmitted to the host exists in FIFO after an in-token is received. If there is no data in the FIFO buffer, set the internal TR flag to 1 and transmit 0-byte data to the host.

In firmware, first, the processing routine of the isochronous transfer is called by SOF interrupt to check the time stamp. Then one packet data is written to FIFO. This written data is transmitted to the host in the next frame.

SOF happens to be broken because of external cause during transmission from the host. In this case, an operation flow is different from that in figure 25.16. As an example, figure 25.17 shows the operation flow of a broken frame and a subsequent frame when SOF is broken once. When SOF is broken, the FIFO buffer is not switched in corresponding frame, and a time out interrupt occurs after time set by user has been elapsed.

The firmware detects the SOF break by the time out interrupt. In this case, the FIFO buffer connected to the CPU has the data to be transmitted in the current frame. If this data is transmitted in the next frame, the data which is not current one is transmitted. Therefore, the firmware writes the EP5 CPU clear (FCLR1/EP5 CCLR) to 1. When the SOF interrupt occurs in the subsequent frame, the processing routine of the isochronous transfer is called and the time stamps are compared. The time stamps do not much since the SOF break occurred in the previous frame. One packet of data is written by the firmware according to the transmitted time stamp.

In the frame in which the SOF is broken, the FIFO buffer is not switched and there in no data to be transmitted to the host. Therefore, USB function controller transmits 0-byte data to the host. Since the data to be transmitted is cleared by firmware, 0-byte data is transmitted to the host.

25.7 Processing of USB Standard Commands and Class/Vendor Commands

25.7.1 Processing of Commands Transmitted by Control Transfer

A command transmitted from the host by control transfer may require decoding and execution of command processing on the application side. Whether command decoding is required on the application side is indicated in table 25.5 below.

Table 25.5 Command Decoding on Application Side

Decoding not Necessary on Application Side	Decoding Necessary on Application Side
Clear feature	Get descriptor
Get configuration	Class/Vendor command
Get interface	Synch frame
Get status	Set descriptor
Set address	
Set configuration	
Set feature	
Set interface	

If decoding is not necessary on the application side, command decoding and data stage and status stage processing are performed automatically. No processing is necessary by the user. An interrupt is not generated in this case.

If decoding is necessary on the application side, this module stores the command in the EP0s FIFO. After normal reception is completed, the IFR0/SETUP TS flag is set and an interrupt request is generated. In the interrupt routine, 8 bytes of data must be read from the EP0s data register (EPDR0S) and decoded by firmware. The necessary data stage and status stage processing should then be carried out according to the result of the decoding operation.

25.8 Stall Operations

25.8.1 Overview

This section describes stall operations in this module. There are two cases in which the USB function controller stall function is used:

- When the application forcibly stalls an endpoint for some reason
- When a stall is performed automatically within the USB function controller due to a USB specification violation

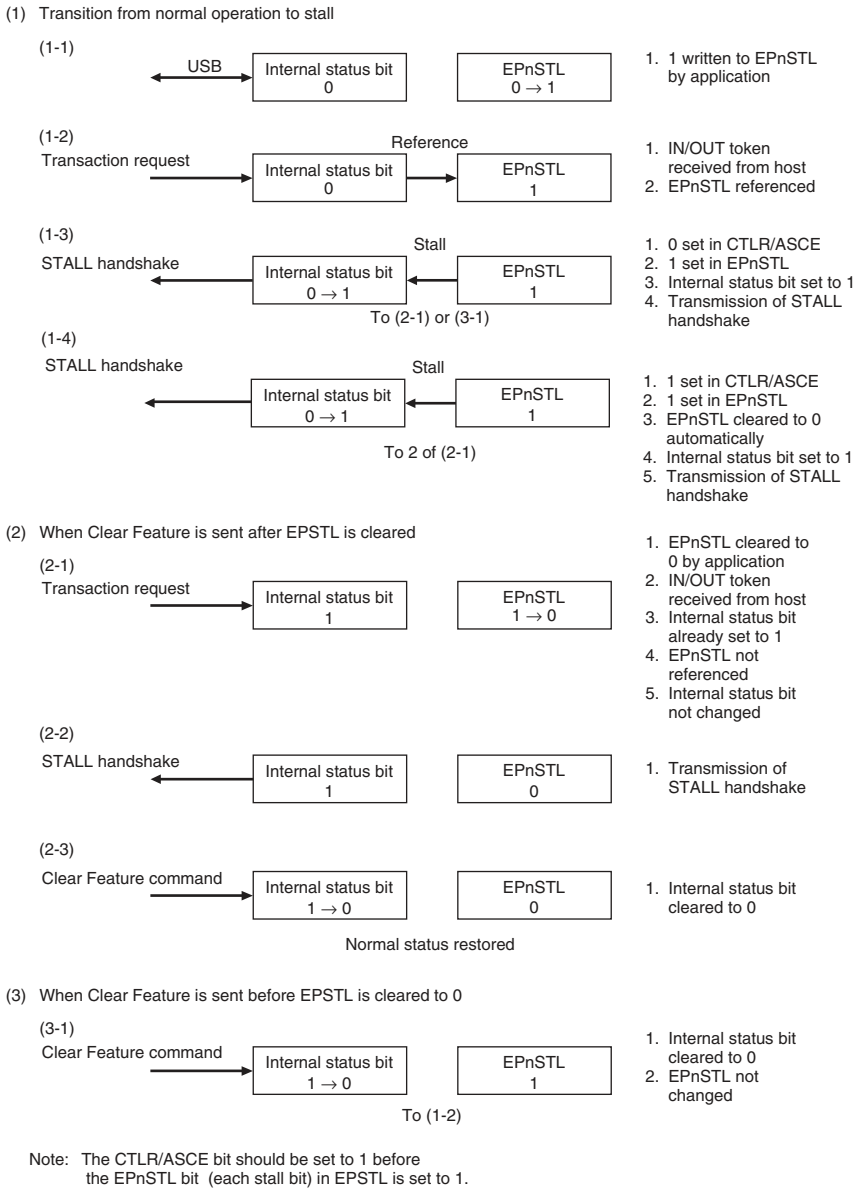
The USB function controller has internal status bits that hold the status (stall or non-stall) of each endpoint. When a transaction is sent from the host, the module references these internal status bits and determines whether to return a stall to the host. These bits cannot be cleared by the application; they must be cleared with a Clear Feature command from the host.

However, the internal status bit to EP0 is automatically cleared only when the setup command is received.

25.8.2 Forcible Stall by Application

The application uses the EPSTL register to issue a stall request for the USB function controller. When the application wishes to stall a specific endpoint, it sets the corresponding bit in EPSTL (1-1 in figure 25.16). The internal status bits are not changed at this time. When a transaction is sent from the host for the endpoint for which the EPSTL bit was set, the USB function controller references the internal status bit, and if this is not set, references the corresponding bit in EPSTL (1-2 in figure 25.16). If the corresponding bit in USBEPSTL is set, the USB function controller sets the internal status bit and returns a stall handshake to the host (1-3 in figure 25.16). In this time, if the CTLR/ASCE bit is set to 1, the corresponding bit in EPSTL is automatically cleared to 0 and a stall handshake is returned to the host (1-4 in figure 25.16). If the corresponding bit in EPSTL is not set, the internal status bit is not changed and the transaction is accepted.

Once an internal status bit is set, it remains set until cleared by a Clear Feature command from the host, without regard to the EPSTL register. Even after a bit is cleared by the Clear Feature command (3-1 in figure 25.16), the USB function controller continues to return a stall handshake while the bit in EPSTL is set, since the internal status bit is set each time a transaction is executed for the corresponding endpoint (1-2 in figure 25.16). To clear a stall, therefore, it is necessary for the corresponding bit in EPSTL to be cleared automatically when a stall is returned from the USB controller while the CTLR/ASCE bit is set to 1, or to be cleared by the application, and also for the internal status bit to be cleared with a Clear Feature command (2-1, 2-2, and 2-3 in figure 25.16).



Note: The CTLR/ASCE bit should be set to 1 before the EPnSTL bit (each stall bit) in EPSTL is set to 1.

Figure 25.18 Forcible Stall by Application

25.8.3 Automatic Stall by USB Function Controller

When a stall setting is made with the Set Feature command, or in the event of a USB specification violation, the USB function controller automatically sets the internal status bit for the relevant endpoint without regard to the corresponding bit in EPSTL, and returns a stall handshake (1-1 in figure 25.19).

Once an internal status bit is set, it remains set until cleared by a Clear Feature command from the host, without regard to the corresponding bit in EPSTL. After a bit is cleared by the Clear Feature command, the corresponding bit in EPSTL is referenced (3-1 in figure 25.19). The USB function controller continues to return a stall handshake while the internal status bit is set, since the internal status bit is set even if a transaction is executed for the corresponding endpoint (2-1 and 2-2 in figure 25.19). To clear a stall, therefore, the internal status bit must be cleared with a Clear Feature command (3-1 in figure 25.19). In this time, if set by the application, the corresponding bit in EPSTL should also be cleared (2-1 in figure 25.19).

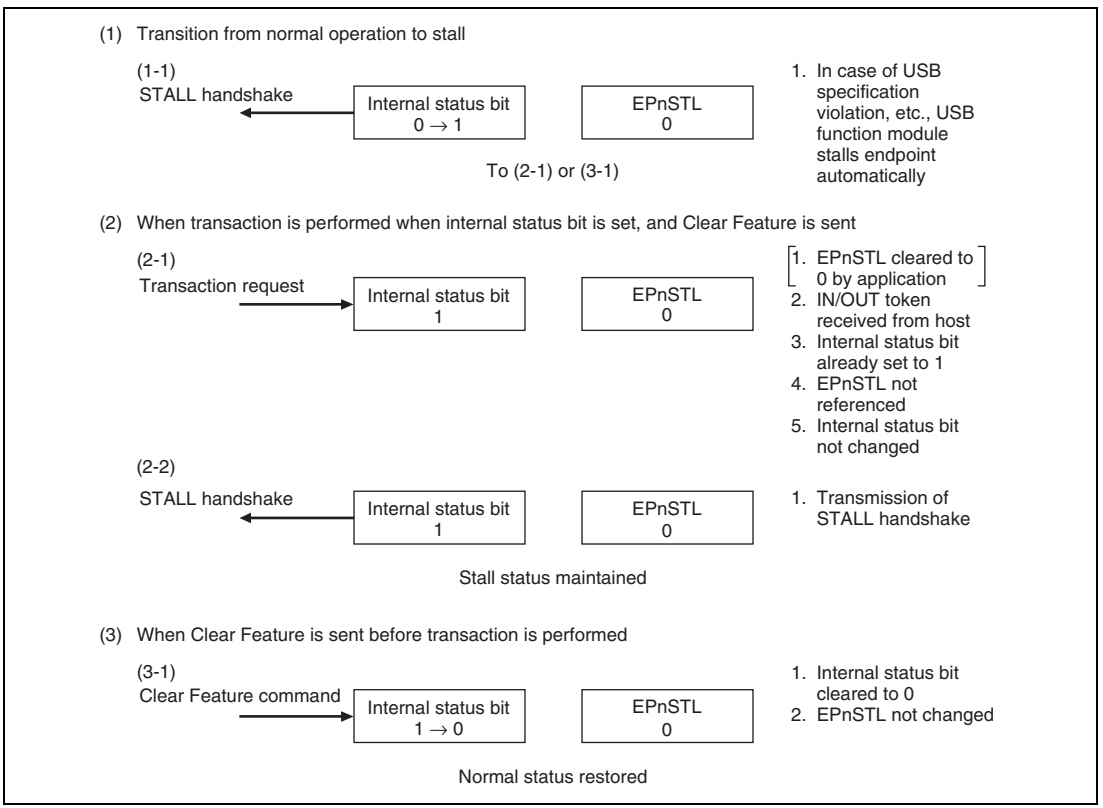


Figure 25.19 Automatic Stall by USB Function Controller

25.9 Usage Notes

25.9.1 Setup Data Reception

The following points should be noted on the EP0s data register (EPDR0s) in which reception of 8-byte setup data is performed.

1. Since the setup command must be received in the USB, writing from the USB bus side is prior to reading from the CPU side. While the CPU reads data after completion of reception and reception of the next setup command is started, reading from the CPU side is forcibly invalid. Therefore a value to be read after starting reception is undefined.
2. EPDR0s must be read in 8-byte units. If reading is suspended while it is in progress, data received in the next setup cannot be read successfully.

25.9.2 FIFO Clear

When the USB cable is disconnected during communication, data which is receiving or transmitting may remain in the FIFO. Therefore the FIFO must be cleared immediately after connecting the USB cable again.

Note that the FIFO in which data is receiving from the host or transmitting to the host must not be cleared.

25.9.3 Overreading/Overwriting of Data Register

The following points should be noted when the data register of the USBF is read from or written to.

(1) Receive Data Register

The receive data register must not read data which is more than valid receive data bytes. That is, data which is more than bytes indicated in the receive data size register must not be read. In case of the receive data register which has the dual FIFO buffer, the maximum number of data which can be read in a single time is maximum packet size. Write 1 to TRG after data in the current valid buffer is read. This writing switches the FIFO buffer. Then, the new number of bytes is reflected in the receive data size and the next data can be read.

(2) Transmit Data Register

The transmit data register must not write data which is more than maximum packet size. In case of the transmit data register which has the dual FIFO buffer, the maximum number of data which can be written in a single time is maximum packet size. Write 1 to TRG/PKTE after data is written. This writing switches the FIFO buffer. Then, the next data can be written to another buffer. Therefore data must not be written in both buffers in a single time.

25.9.4 Assigning EP0 Interrupt Sources

The EP0 interrupt sources assigned to IFR0 (bits 0, 1, and 2) must be assigned to the same interrupt pins by ISR0. The other interrupt sources have no restrictions.

25.9.5 FIFO Clear when DMA Transfer is Set

When the DMA transfer is enabled in endpoint 1, the data register cannot be cleared. Cancel the DMA transfer before clearing the data register.

25.9.6 Note on Using TR Interrupt

The bulk-in transfer has a transfer request interrupt (TR interrupt). The following points should be noted when using a TR interrupt.

When the IN token is sent from the USB host and there is no data in the corresponding EP FIFO, the TR interrupt flag is set. However, the TR interrupt is generated continuously at the timing as shown in figure 20.18. In this case, note that erroneous operation should not occur.

Note: When the IN token is received and there is no data in the corresponding EP FIFO, an NAK is determined. However, the TR interrupt flag is set after an NAK handshake is transmitted. Therefore when the next IN token is received before TRG/PKTE is written, the TR interrupt flag is set again.

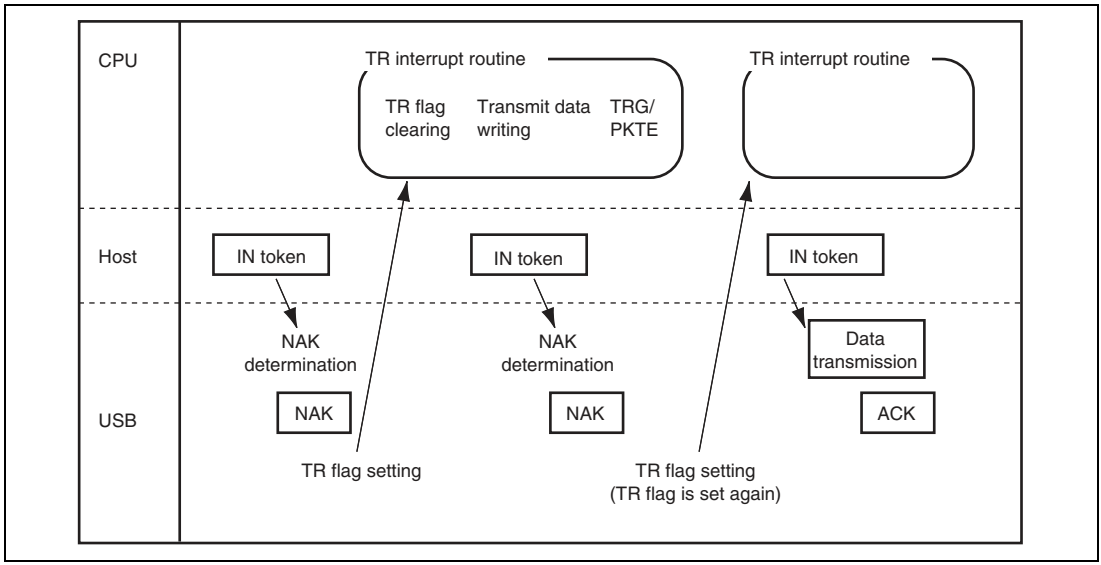


Figure 25.20 Set Timing of TR Interrupt Flag

25.9.7 Note on Clock Frequency

When using the USBF, be sure to set the peripheral clock (Pϕ) at a frequency higher than 13 MHz.

Section 26 LCD Controller (LCDC)

A unified memory architecture is adopted for the LCD controller (LCDC) so that the image data for display is stored in system memory. The LCDC module reads data from system memory, uses the palette memory to determine the colors, then puts the display on the LCD panel. It is possible to connect the LCDC to the LCD module* other than microcomputer bus interface types and NTSC/PAL types and those that apply the LVDS interface.

Note: * LCD module can be connected to the LVDS interface by using the LSI with LVDS conversion LSI.

26.1 Features

The LCDC has the following features.

- Panel interface
  - Serial interface method
  - Supports data formats for STN/dual-STN/TFT panels (8/12/16/18-bit bus width)*[1]
- Supports 4/8/15/16-bpp (bits per pixel) color modes
- Supports 1/2/4/6-bpp grayscale modes
- Supports LCD-panel sizes from 16 × 1 to 1024 × 1024*[2]
- 24-bit color palette memory (16 of the 24 bits are valid; R:5/G:6/B:5)
- STN/DSTN panels are prone to flicker and shadowing. The controller applies 65536-color control by 24-bit space-modulation FRC (Frame Rate Controller) with 8-bit RGB values for reduced flicker.
- Dedicated display memory is unnecessary using part of the synchronous DRAM (area 3) as the VRAM to store display data of the LCDC.
- The display is stable because of the large 2.4-kbyte line buffer
- Supports the inversion of the output signal to suit the LCD panel's signal polarity
- Supports the selection of data formats (the endian setting for bytes, packed pixel method) by register settings
- An interrupt can be generated at the user specified position (controlling the timing of VRAM update start prevents flicker)
- A hardware-rotation mode is included to support the use of landscape-format LCD panels as portrait-format LCD panels (the horizontal width of the panel before rotation must be within 320 pixels (see table 26.4.)

Notes: 1. When connecting the LCDC to a TFT panel with an unwired 18-bit bus, the lower bit lines should be connected to GND or to the lowest bit from which data is output.

2. For details, see section 26.4.1, LCD Module Sizes which can be Displayed in this LCDC.

Figure 26.1 shows a block diagram of LCDC.

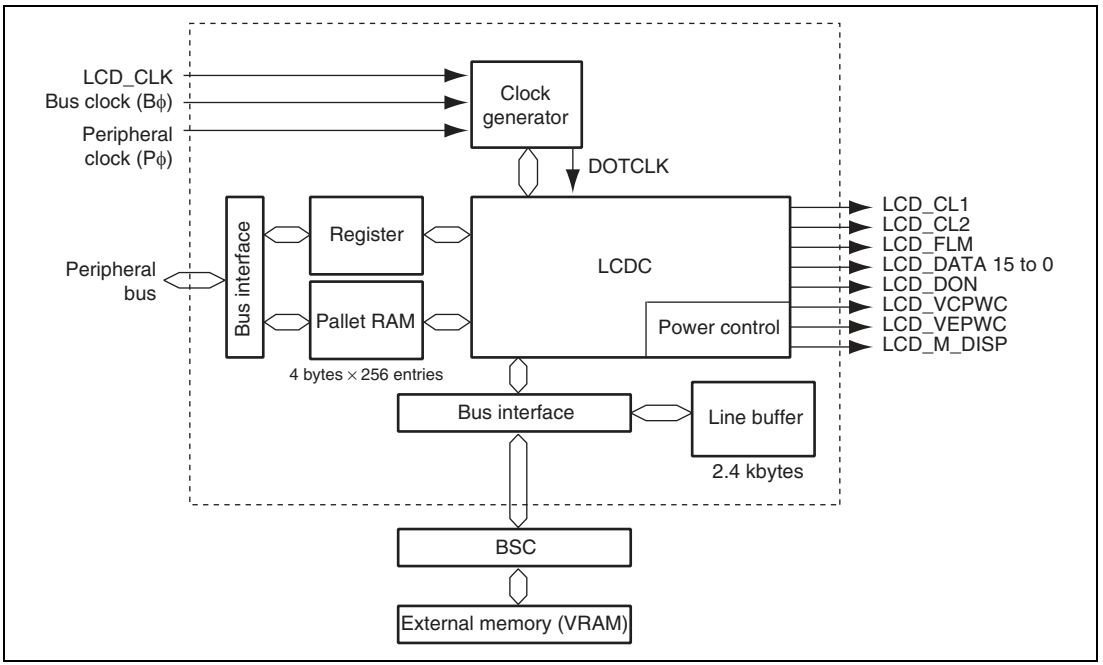


Figure 26.1 LCDC Block Diagram

26.2 Input/Output Pins

Table 26.1 summarizes the LCDC's pin configuration.

Table 26.1 Pin Configuration

Pin Name	I/O	Function
LCD_DATA15 to 0	Output	Data for LCD panel
LCD_DON	Output	Display-on signal (DON)
LCD_CL1	Output	Shift-clock 1 (STN/DSTN)/horizontal sync signal (HSYNC) (TFT)
LCD_CL2	Output	Shift-clock 2 (STN/DSTN)/dot clock (DOTCLK) (TFT)
LCD_M_DISP	Output	LCD current-alternating signal/DISP signal
LCD_FLM	Output	First line marker/vertical sync signal (VSYNC) (TFT)
LCD_VCPWC	Output	LCD-module power control (VCC)
LCD_VEPWC	Output	LCD-module power control (VEE)
LCD_CLK	Input	LCD clock-source input

Note: Check the LCD module specifications carefully in section 26.5, Clock and LCD Data Signal Examples, before deciding on the wiring specifications for the LCD module.

26.3 Register Configuration

The LCDC includes the following registers. Refer to section 37, List of Registers, for more details on the addresses and states of these registers in each operating mode.

- LCDC input clock register (LDICKR)
- LCDC module type register (LDMTR)
- LCDC data format register (LDDFR)
- LCDC scan mode register (LDSMR)
- LCDC data fetch start address register for upper display panel (LDSARU)
- LCDC data fetch start address register for lower display panel (LDSARL)
- LCDC fetch data line address offset register for display panel (LDLAOR)
- LCDC palette control register (LDPALCR)
- LCDC palette data register 00 to FF (LDPR00 to LDPRFF)
- LCDC horizontal character number register (LDHCNR)
- LCDC horizontal synchronization signal register (LDHSYNR)
- LCDC vertical displayed line number register (LDVDLNR)
- LCDC vertical total line number register (LDVTLNR)
- LCDC vertical synchronization signal register (LDVSYNR)
- LCDC AC modulation signal toggle line number register (LDACLNR)
- LCDC interrupt control register (LDINTR)
- LCDC power management mode register (LDPMMR)
- LCDC power supply sequence period register (LDPSPR)
- LCDC control register (LDCNTR)
- LCDC user specified interrupt control register (LDUINTR)
- LCDC user specified interrupt line number register (LDUINTLNR)
- LCDC memory access interval number register (LDLIRNR)

26.3.1 LCDC Input Clock Register (LDICKR)

This LCDC can select the bus clock (Bϕ), the peripheral clock (Pϕ), or the external clock (LCD_CLK) as its operation clock source. The selected clock source can be divided using an internal divider into a clock of 1/1 to 1/32 and be used as the LCDC operating clock (DOTCLK). The clock output from the LCDC is used to generate the synchronous clock output (LCD_CL2) for the LCD panel from the operating clock selected in this register. For a TFT panel, LCD_CL2 = DOTCLK, and for an STN or DSTN monochrome panel, LCD_CL2 = a clock with a frequency of (DOTCLK/data bus width of output to LCD panel). For a color panel, LCD_CL2 = a clock with a frequency of (3 × DOTCLK/data bus width of output to LCD panel). The LDICKR must be set so that the clock input to the LCDC is 66 MHz or less regardless of the LCD_CL2.

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	ICKSEL1	0	R/W	Input Clock Select
12	ICKSEL0	0	R/W	Set the clock source for DOTCLK. 00: Bus clock is selected (Bϕ) 01: Peripheral clock is selected (Pϕ) 10: External clock is selected (LCD_CLK) 11: Setting prohibited
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	DCDR5	0	R/W	Clock Division Ratio
4	DCDR4	0	R/W	Set the input clock division ratio. For details on the setting, refer to table 26.2.
3	DCDR3	0	R/W	
2	DCDR2	0	R/W	
1	DCDR1	0	R/W	
0	DCDR0	1	R/W	

Table 26.2 I/O Clock Frequency and Clock Division Ratio

DCDR[5:0]	Clock Division Ratio	I/O Clock Frequency (MHz)		
		50.000	60.000	66.000
000001	1/1	50.000	60.000	66.000
000010	1/2	25.000	30.000	33.000
000011	1/3	16.667	20.000	22.000
000100	1/4	12.500	15.000	16.500
000110	1/6	8.333	10.000	11.000
001000	1/8	6.250	7.500	8.250
001100	1/12	4.167	5.000	5.500
010000	1/16	3.125	3.750	4.125
011000	1/24	2.083	2.500	2.750
100000	1/32	1.563	1.875	2.063

Note: Any setting other than above is handled as a clock division ratio of 1/1 (initial value).

26.3.2 LCDC Module Type Register (LDMTR)

LDMTR sets the control signals output from this LCDC and the polarity of the data signals, according to the polarity of the signals for the LCD module connected to the LCDC.

Bit	Bit Name	Initial Value	R/W	Description
15	FLMPOL	0	R/W	FLM (Vertical Sync Signal) Polarity Select Selects the polarity of the LCD_FLM (vertical sync signal, first line marker) for the LCD module. 0: LCD_FLM pulse is high active 1: LCD_FLM pulse is low active
14	CL1POL	0	R/W	CL1 (Horizontal Sync Signal) Polarity Select Selects the polarity of the LCD_CL1 (horizontal sync signal) for the LCD module. 0: LCD_CL1 pulse is high active 1: LCD_CL1 pulse is low active

Bit	Bit Name	Initial Value	R/W	Description
13	DISPPOL	0	R/W	DISP (Display Enable) Polarity Select Selects the polarity of the LCD_M_DISP (display enable) for the LCD module. 0: LCD_M_DISP is high active 1: LCD_M_DISP is low active
12	DPOL	0	R/W	Display Data Polarity Select Selects the polarity of the LCD_DATA (display data) for the LCD module. This bit supports inversion of the LCD module. 0: LCD_DATA is high active, transparent-type LCD panel 1: LCD_DATA is low active, reflective-type LCD panel
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	MCNT	0	R/W	M Signal Control Sets whether or not to output the LCD's current-alternating signal of the LCD module. 0: M (AC line modulation) signal is output 1: M signal is not output
9	CL1CNT	0	R/W	CL1 (Horizontal Sync Signal) Control Sets whether or not to enable CL1 output during the vertical retrace period. 0: CL1 is output during vertical retrace period 1: CL1 is not output during vertical retrace period
8	CL2CNT	1	R/W	CL2 (Dot Clock of LCD Module) Control Sets whether or not to enable CL2 output during the vertical and horizontal retrace period. 0: CL2 is output during vertical and horizontal retrace period 1: CL2 is not output during vertical and horizontal retrace period
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	MIFTYP5	0	R/W	Module Interface Type Select
4	MIFTYP4	0	R/W	Set the LCD panel type and data bus width to be output to the LCD panel. There are three LCD panel types: STN, DSTN, and TFT. There are four data bus widths for output to the LCD panel: 4, 8, 12, and 16 bits. When the required data bus width for a TFT panel is 16 bits or more, connect the LCDC and LCD panel according to the data bus size of the LCD panel. Unlike in a TFT panel, in an STN or DSTN panel, the data bus width setting does not have a 1:1 correspondence with the number of display colors and display resolution, e.g., an 8-bit data bus can be used for 16 bpp, and a 12-bit data bus can be used for 4 bpp. This is because the number of display colors in an STN or DSTN panel is determined by how data is placed on the bus, and not by the number of bits. For data specifications for an STN or DSTN panel, see the specifications of the LCD panel used. The output data bus width should be set according to the mechanical interface specifications of the LCD panel. If an STN or DSTN panel is selected, display control is performed using a 24-bit space-modulation FRC (Frame Rate Controller) consisting of the 8-bit R, G, and B included in the LCDC, regardless of the color and gradation settings. Accordingly, the color and gradation specified by DSPCOLOR is selected from 16 million colors in an STN or DSTN panel. If a palette is used, the color specified in the palette is displayed. 000000: STN monochrome 4-bit data bus module 000001: STN monochrome 8-bit data bus module 001000: STN color 4-bit data bus module 001001: STN color 8-bit data bus module 001010: STN color 12-bit data bus module 001011: STN color 16-bit data bus module 010001: DSTN monochrome 8-bit data bus module 010011: DSTN monochrome 16-bit data bus module 011001: DSTN color 8-bit data bus module 011010: DSTN color 12-bit data bus module 011011: DSTN color 16-bit data bus module 101011: TFT color 16-bit data bus module Settings other than above: Setting prohibited
3	MIFTYP3	1	R/W	
2	MIFTYP2	0	R/W	
1	MIFTYP1	0	R/W	
0	MIFTYP0	1	R/W	

26.3.3 LCDC Data Format Register (LDDFR)

LDDFR sets the bit alignment for pixel data in one byte and selects the data type and number of colors used for display so as to match the display driver software specifications.

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	PABD	0	R/W	Byte Data Pixel Alignment Sets the pixel data alignment type in one byte of data. The contents of aligned data per pixel are the same regardless of this bit's setting. For example, data H'05 should be expressed as B'0101 which is the normal style handled by a MOV instruction of the this CPU, and should not be selected between B'0101 and B'1010. 0: Big endian for byte data 1: Little endian for byte data
7	—	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
6	DSPCOLOR6	0	R/W	Display Color Select
5	DSPCOLOR5	0	R/W	Set the number of display colors for the display (0 is written to upper bits of 4 to 6 bpp). For display colors to which the description (via palette) is added below, the color set by the color palette is actually selected by the display data and displayed.
4	DSPCOLOR4	0	R/W	
3	DSPCOLOR3	1	R/W	
2	DSPCOLOR2	1	R/W	
1	DSPCOLOR1	0	R/W	The number of colors that can be selected in rotation mode is restricted by the display resolution. For details, see table 26.4.
0	DSPCOLOR0	0	R/W	
				0000000: Monochrome, 2 grayscales, 1 bpp (via palette)
				0000001: Monochrome, 4 grayscales, 2 bpp (via palette)
				0000010: Monochrome, 16 grayscales, 4 bpp (via palette)
				0000100: Monochrome, 64 grayscales, 6 bpp (via palette)
				0001010: Color, 16 colors, 4 bpp (via palette)
				0001100: Color, 256 colors, 8 bpp (via palette)
				0011101: Color, 32k colors (RGB: 555), 15 bpp
				0101101: Color, 64k colors (RGB: 565), 16 bpp
				Settings other than above: Setting prohibited

26.3.4 LCDC Scan Mode Register (LDSMR)

LDSMR selects whether or not to enable the hardware rotation function that is used to rotate the LCD panel, and sets the burst length for the VRAM (synchronous DRAM in area 3) used for display.

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	ROT	0	R/W	Rotation Module Select Selects whether or not to rotate the display by hardware. Note that the following restrictions are applied to rotation. • An STN or TFT panel must be used. A DSTN panel is not allowed. • The maximum horizontal (internal scan direction of the LCD panel) width of the LCD panel is 320. • Set a binary exponential that exceeds the display size in LDLAOR. (For example, 256 must be selected when a 320 × 240 panel is rotated to be used as a 240 × 320 panel and the horizontal width of the image is 240 bytes.) 0: Not rotated 1: Rotated 90 degrees rightwards (left side of image is displayed on the upper side of the LCD module)
12 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	AU1	0	R/W	Access Unit Select
8	AU0	0	R/W	Select access unit of VRAM. This bit is enabled when ROT = 1 (rotate the display). When ROT = 0, 16-burst memory read operation is carried out whatever the AU setting is. 00: 4-burst 01: 8-burst 10: 16-burst 11: 32-burst Notes: 1. Above burst lengths are used for 32-bit bus. For 16-bit bus, the burst lengths are twice the lengths of 32-bit bus. 2. When displaying a rotated image, the burst length is limited depending on the number of column address bits and bus width of connected SDRAM. For details, see tables 26.3 and 26.4.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

26.3.5 LCDC Start Address Register for Upper Display Data Fetch (LDSARU)

LDSARU sets the start address from which data is fetched by the LCDC for display of the LCDC panel. When a DSTN panel is used, this register specifies the fetch start address for the upper side of the panel.

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27, 26	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
25 to 4	SAU25 to SAU4	All 0	R/W	Start Address for Upper Display Data Fetch The start address for data fetch of the display data must be set within the synchronous DRAM area of area 3.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. The minimum alignment unit of LDSARU is 512 bytes when the hardware rotation function is not used. Write 0 to the lower nine bits. When using the hardware rotation function, set the LDSARU value so that the upper-left address of the image is aligned with the 512-byte boundary.

2. When the hardware rotation function is used (ROT = 1), set the upper-left address of the image, which can be calculated from the display image size in this register. The equation below shows how to calculate the LDSARU value when the image size is 240 × 340 and LDLAOR = 256. The LDSARU value is obtained not from the panel size but from the memory size of the image to be displayed. Note that LDLAOR must be a binary exponential at least as large as the horizontal width of the image. Calculate backwards using the LDSARU value (LDSARU − 256 (LDLAOR value) × (320 − 1)) to ensure that the upper-left address of the image is aligned with the 512-byte boundary.
LDSARU = (upper-left address of image) + 256 (LDLAOR value) × 319 (line)

26.3.6 LCDC Start Address Register for Lower Display Data Fetch (LDSARL)

When a DSTN panel is used, LDSARL specifies the fetch start address for the lower side of the panel.

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27, 26	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
25 to 4	SAL25 to SAL4	All 0	R/W	Start Address for Lower Panel Display Data Fetch The start address for data fetch of the display data must be set within the synchronous DRAM area of area 3. STN and TFT: Cannot be used DSTN: Start address for fetching display data corresponding to the lower panel
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

26.3.7 LCDC Line Address Offset Register for Display Data Fetch (LDLAOR)

LDLAOR sets the address width of the Y-coordinates increment used for LCDC to read the image recognized by the graphics driver. This register specifies how many bytes the address from which data is to be read should be moved when the Y coordinates have been incremented by 1. This register does not have to be equal to the horizontal width of the LCD panel. When the memory address of a point (X, Y) in the two-dimensional image is calculated by Ax + By+ C, this register becomes equal to B in this equation.

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	LAO15 to LAO10	All 0	R/W	Line Address Offset The minimum alignment unit of LDLAOR is 16 bytes. Because the LCDC handles these values as 16-byte data, the values written to the lower four bits of the register are always treated as 0. The lower four bits of the register are always read as 0. The initial values (× resolution = 640) will continuously and accurately place the VGA (640 × 480 dots) display data without skipping an address between lines. For details, see tables 26.3 and 26.4. A binary exponential at least as large as the horizontal width of the image is recommended for the LDLAOR value while taking into consideration the software operation speed. When the hardware rotation function is used, the LDLAOR value should be a binary exponential (in this example, 256) at least as large as the horizontal width of the image (after rotation, it becomes 240 in a 240 × 320 panel) instead of the horizontal width of the LCD panel (320 in a 320 × 240 panel).
9	LAO9	1	R/W	
8	LAO8	0	R/W	
7	LAO7	1	R/W	
6 to 0	LAO6 to LAO0	All 0	R/W	

26.3.8 LCDC Palette Control Register (LDPALCR)

LDPALCR selects whether the CPU or LCDC accesses the palette memory. When the palette memory is being used for display operation, display mode should be selected. When the palette memory is being written to, color-palette setting mode should be selected.

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits always read as 0. The write value should always be 0.
4	PALS	0	R	Palette State Indicates the access right state of the palette. 0: Display mode: LCDC uses the palette 1: Color-palette setting mode: The host (CPU) uses the palette
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PALEN	0	R/W	Palette Read/Write Enable Requests the access right to the palette. 0: Request for transition to normal display mode 1: Request for transition to color palette setting mode

26.3.9 Palette Data Registers 00 to FF (LDPR00 to LDPRFF)

LDPR registers are for accessing palette data directly allocated (4 bytes x 256 addresses) to the memory space. To access the palette memory, access the corresponding register among this register group (LDPR00 to LDPRFF). Each palette register is a 32-bit register including three 8-bit areas for R, G, and B. For details on the color palette specifications, see section 26.4.3, Color Palette Specification.

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	Reserved
23 to 0	PALDnn23 to PALDnn0	—	R/W	Palette Data Bits 18 to 16, 9, 8, and 2 to 0 are reserved within each RGB palette and cannot be set. However, these bits can be extended according to the upper bits.

Note: nn = H'00 to H'FF

26.3.10 LCDC Horizontal Character Number Register (LDHCNR)

LDHCNR specifies the LCD module's horizontal size (in the scan direction) and the entire scan width including the horizontal retrace period.

Bit	Bit Name	Initial Value	R/W	Description
15	HDCN7	0	R/W	Horizontal Display Character Number
14	HDCN6	1	R/W	Set the number of horizontal display characters (unit: character = 8 dots).
13	HDCN5	0	R/W	
12	HDCN4	0	R/W	Specify to the value of (the number of display characters) -1.
11	HDCN3	1	R/W	
10	HDCN2	1	R/W	Example: For a LCD module with a width of 640 pixels. HDCN = (640/8) -1 = 79 = H'4F
9	HDCN1	1	R/W	
8	HDCN0	1	R/W	
7	HTCN7	0	R/W	Horizontal Total Character Number
6	HTCN6	1	R/W	Set the number of total horizontal characters (unit: character = 8 dots).
5	HTCN5	0	R/W	
4	HTCN4	1	R/W	Specify to the value of (the number of total characters) - 1.
3	HTCN3	0	R/W	
2	HTCN2	0	R/W	However, the minimum horizontal retrace period is three characters (24 dots).
1	HTCN1	1	R/W	
0	HTCN0	0	R/W	Example: For a LCD module with a width of 640 pixels. HTCN = [(640/8)-1] +3 = 82 = H'52 In this case, the number of total horizontal dots is 664 dots and the horizontal retrace period is 24 dots.

Notes: 1. The values set in HDCN and HTCN must satisfy the relationship of HTCN ≥ HDCN.
2. Set HDCN according to the display resolution as follows:
1 bpp: (multiplex of 16) – 1 [1 line is multiplex of 128 pixel]
2 bpp: (multiplex of 8) – 1 [1 line is multiplex of 64 pixel]
4 bpp: (multiplex of 4) – 1 [1 line is multiplex of 32 pixel]
6 bpp/8 bpp: (multiplex of 2) – 1 [1 line is multiplex of 16 pixel]

26.3.11 LCDC Horizontal Sync Signal Register (LDHSYNR)

LDHSYNR specifies the timing of the generation of the horizontal (scan direction) sync signals for the LCD module.

Bit	Bit Name	Initial Value	R/W	Description
15	HSYNW3	0	R/W	Horizontal Sync Signal Width
14	HSYNW2	0	R/W	Set the width of the horizontal sync signals (CL1 and Hsync) (unit: character = 8 dots).
13	HSYNW1	0	R/W	
12	HSYNW0	0	R/W	Specify to the value of (the number of horizontal sync signal width) -1. Example: For a horizontal sync signal width of 8 dots. HSYNW = (8 dots/8 dots/character) -1 = 0 = H'0
11 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	HSYNP7	0	R/W	Horizontal Sync Signal Output Position
6	HSYNP6	1	R/W	Set the output position of the horizontal sync signals (unit: character = 8 dots).
5	HSYNP5	0	R/W	
4	HSYNP4	1	R/W	Specify to the value of (the number of horizontal sync signal output position) -1.
3	HSYNP3	0	R/W	
2	HSYNP2	0	R/W	Example: For a LCD module with a width of 640 pixels. HSYNP = [(640/8) +1] -1 = 80 = H'50 In this case, the horizontal sync signal is active from the 648th through the 655th dot.
1	HSYNP1	0	R/W	
0	HSYNP0	0	R/W	

Note: The following conditions must be satisfied:
HTCN ≥ HSYNP+HSYNW+1
HSYNP ≥ HDCN+1

26.3.12 LCDC Vertical Display Line Number Register (LDVDLNR)

LDVDLNR specifies the LCD module's vertical size (for both scan direction and vertical direction). For a DSTN panel, specify an even number at least as large as the LCD panel's vertical size regardless of the size of the upper and lower panels, e.g. 480 for a 640 x 480 panel.

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	VDLN10	0	R/W	Vertical Display Line Number Set the number of vertical display lines (unit: line). Specify to the value of (the number of display line) - 1. Example: For an 480-line LCD module VDLN = 480-1 = 479 = H'1DF
9	VDLN9	0	R/W	
8	VDLN8	1	R/W	
7	VDLN7	1	R/W	
6	VDLN6	1	R/W	
5	VDLN5	0	R/W	
4	VDLN4	1	R/W	
3	VDLN3	1	R/W	
2	VDLN2	1	R/W	
1	VDLN1	1	R/W	
0	VDLN0	1	R/W	

26.3.13 LCDC Vertical Total Line Number Register (LDVTLNR)

LDVTLNR specifies the LCD panel's entire vertical size including the vertical retrace period.

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	VTLN10	0	R/W	Vertical Total Line Number Set the total number of vertical display lines (unit: line). Specify to the value of (the number of total line) -1. The minimum for the total number of vertical lines is 2 lines. The following conditions must be satisfied: VTLN>=VDLN, VTLN>=1. Example: For an 480-line LCD module and a vertical period of 0 lines. VTLN = (480+0) –1 = 479 = H'1DF
9	VTLN9	0	R/W	
8	VTLN8	1	R/W	
7	VTLN7	1	R/W	
6	VTLN6	1	R/W	
5	VTLN5	0	R/W	
4	VTLN4	1	R/W	
3	VTLN3	1	R/W	
2	VTLN2	1	R/W	
1	VTLN1	1	R/W	
0	VTLN0	1	R/W	

26.3.14 LCDC Vertical Sync Signal Register (LDVSYNR)

LDVSYNR specifies the vertical (scan direction and vertical direction) sync signal timing of the LCD module.

Bit	Bit Name	Initial Value	R/W	Description
15	VSYNW3	0	R/W	Vertical Sync Signal Width
14	VSYNW2	0	R/W	Set the width of the vertical sync signals (FLM and Vsync) (unit: line).
13	VSYNW1	0	R/W	
12	VSYNW0	0	R/W	Specify to the value of (the vertical sync signal width) -1. Example: For a vertical sync signal width of 1 line. VSYNW = (1-1) = 0 = H'0
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	VSYNP10	0	R/W	Vertical Sync Signal Output Position
9	VSYNP9	0	R/W	Set the output position of the vertical sync signals (FLM and Vsync) (unit: line).
8	VSYNP8	1	R/W	
7	VSYNP7	1	R/W	Specify to the value of (the number of vertical sync signal output position) -2.
6	VSYNP6	1	R/W	
5	VSYNP5	0	R/W	DSTN should be set to an odd number value. It is handled as (setting value+1)/2.
4	VSYNP4	1	R/W	
3	VSYNP3	1	R/W	Example: For an 480-line LCD module and a vertical retrace period of 0 lines (in other words, VTLN=479 and the vertical sync signal is active for the first line):
2	VSYNP2	1	R/W	
1	VSYNP1	1	R/W	
0	VSYNP0	1	R/W	• Single display VSYNP = [(1-1)+VTLN]mod(VTLN+1) = [(1-1)+479]mod(479+1) = 479mod480 = 479 =H'1DF • Dual displays VSYNP = [(1-1)×2+VTLN]mod(VTLN+1) = [(1-1)×2+479]mod(479+1) = 479mod480 = 479 =H'1DF

26.3.15 LCDC AC Modulation Signal Toggle Line Number Register (LDACLNR)

LDACLNR specifies the timing to toggle the AC modulation signal (LCD current-alternating signal) of the LCD module.

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	ACLN4	0	R/W	AC Line Number Set the number of lines where the LCD current-alternating signal of the LCD module is toggled (unit: line). Specify to the value of (the number of toggle line) - 1. Example: For toggling every 13 lines. ACLN = 13-1 = 12= H'0C
3	ACLN3	1	R/W	
2	ACLN2	1	R/W	
1	ACLN1	0	R/W	
0	ACLN0	0	R/W	

Note: When the total line number of the LCD panel is even, set an even number so that toggling is performed at an odd line.

26.3.16 LCDC Interrupt Control Register (LDINTR)

LDINTR specifies where to control the Vsync interrupt of the LCD module. See also 26.3.20, LCDC user specified interrupt control register (LDUINTR) and 26.3.21, LCDC user specified interrupt line number register (LDUINTLNR) for interrupts. Note that operations by this register setting and LCDC user specified interrupt control register (LDUINTR) setting are independent.

Bit	Bit Name	Initial Value	R/W	Description
15	MINTEN	0	R/W	Memory Access Interrupt Enable Enables or disables an interrupt generation at the start point of each vertical retrace line period for VRAM access by LCDC. 0: Disables an interrupt generation at the start point of each vertical retrace line period for VRAM access 1: Enables an interrupt generation at the start point of each vertical retrace line period for VRAM access
14	FINTEN	0	R/W	Frame End Interrupt Enable Enables or disables the generation of an interrupt after the last pixel of a frame is output to LDC panel. 0: Disables an interrupt generation when the last pixel of the frame is output 1: Enables an interrupt generation when the last pixel of the frame is output
13	VSINTEN	0	R/W	Vsync Starting Point Interrupt Enable Enables or disables the generation of an interrupt at the start point of LCDC's Vsync. 0: Interrupt at the start point of the Vsyncl is disabled 1: Interrupt at the start point of the Vsync is enabled
12	VEINTEN	0	R/W	Vsync Ending Point Interrupt Enable Enables or disables the generation of an interrupt at the end point of LCDC's Vsync. 0: Interrupt at the end point of the Vsync signal is disabled 1: Interrupt at the end point of the Vsync signal is enabled

Bit	Bit Name	Initial Value	R/W	Description
11	MINTS	0	R/W	Memory Access Interrupt State Indicates the memory access interrupt handling state. This bit indicates 1 when the LCDC memory access interrupt is generated (set state). During the memory access interrupt handling routine, this bit should be cleared by writing 0. 0: LCDC did not generate a memory access interrupt or has been informed that the generated memory access interrupt has completed 1: LCDC has generated a memory access end interrupt and not yet been informed that the generated memory access interrupt has completed
10	FINTS	0	R/W	Flame End Interrupt State Indicates the flame end interrupt handling state. This bit indicates 1 at the time when the LCDC flame end interrupt is generated (set state). During the flame end interrupt handling routine, this bit should be cleared by writing 0. 0: LCDC did not generate a flame end interrupt or has been informed that the generated flame end interrupt has completed 1: LCDC has generated a flame end interrupt and not yet been informed that the generated flame end interrupt has completed
9	VSINTS	0	R/W	Vsync Start Interrupt State Indicates the LCDC's Vsync start interrupt handling state. This bit is set to 1 at the time a Vsync start interrupt is generated. During the Vsync start interrupt handling routine, this bit should be cleared by writing 0 to it. 0: LCDC did not generate a Vsync start interrupt or has been informed that the generated Vsync start interrupt has completed 1: LCDC has generated a Vsync start interrupt and has not yet been informed that the generated Vsync start interrupt has completed

Bit	Bit Name	Initial Value	R/W	Description
8	VEINTS	0	R/W	Vsync End Interrupt State Indicates the LCDC's Vsync end interrupt handling state. This bit is set to 1 at the time a Vsync end interrupt is generated. During the Vsync end interrupt handling routine, this bit should be cleared by writing 0. 0: LCDC did not generate a Vsync end interrupt or has been informed that the generated Vsync end interrupt has completed 1: LCDC has generated a Vsync end interrupt and has not yet been informed that the generated Vsync interrupt has completed
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

26.3.17 LCDC Power Management Mode Register (LDPMMR)

LDPMMR controls the power supply circuit that provides power to the LCD module. The usage of two types of power-supply control pins, LCD_VCPWC and LCD_VEPWC, and turning on or off the power supply function are selected.

Bit	Bit Name	Initial Value	R/W	Description
15	ONC3	0	R/W	LCDC Power-On Sequence Period
14	ONC2	0	R/W	Set the period from LCD_VEPWC assertion to LCD_DON assertion in the power-on sequence of the LCD module in frame units.
13	ONC1	0	R/W	
12	ONC0	0	R/W	Specify to the value of (the period) -1.
				This period is the (c) period in figures 26.4 to 26.7, Power-Supply Control Sequence and States of the LCD Module. For details on setting this register, see table 26.5, Available Power-Supply Control-Sequence Periods at Typical Frame Rates. (The setting method is common for ONA, ONB, OFFD, OFFE, and OFFF.)
11	OFFD3	0	R/W	LCDC Power-Off Sequence Period
10	OFFD2	0	R/W	Set the period from LCD_DON negation to LCD_VEPWC negation in the power-off sequence of the LCD module in frame units.
9	OFFD1	0	R/W	
8	OFFD0	0	R/W	Specify to the value of (the period) -1.
				This period is the (d) period in figures 26.4 to 26.7, Power-Supply Control Sequence and States of the LCD Module.
7	—	0	R	Reserved
				This bit is always read as 0. The write value should always be 0.
6	VCPE	0	R/W	LCD_VCPWC Pin Enable
				Sets whether or not to enable a power-supply control sequence using the LCD_VCPWC pin.
				0: Disabled: LCD_VCPWC pin is masked and fixed low
				1: Enabled: LCD_VCPWC pin output is asserted and negated according to the power-on or power-off sequence

Bit	Bit Name	Initial Value	R/W	Description
5	VEPE	0	R/W	LCD_VEPWC Pin Enable Sets whether or not to enable a power-supply control sequence using the LCD_VEPWC pin. 0: Disabled: LCD_VEPWC pin is masked and fixed low 1: Enabled: LCD_VEPWC pin output is asserted and negated according to the power-on or power-off sequence
4	DONE	1	R/W	LCD_DON Pin Enable Sets whether or not to enable a power-supply control sequence using the LCD_DON pin. 0: Disabled: LCD_DON pin is masked and fixed low 1: Enabled: LCD_DON pin output is asserted and negated according to the power-on or power-off sequence
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	LPS1	0	R	LCD Module Power-Supply Input State
0	LPS0	0	R	Indicates the power-supply input state of the LCD module when using the power-supply control function. 0: LCD module power off 1: LCD module power on

26.3.18 LCDC Power-Supply Sequence Period Register (LDPSPR)

LDPSPR controls the power supply circuit that provides power to the LCD module. The timing to start outputting the timing signals to the LCD_VEPWC and LCD_VCPWC pins is specified.

Bit	Bit Name	Initial Value	R/W	Description
15	ONA3	1	R/W	LCDC Power-On Sequence Period
14	ONA2	1	R/W	Set the period from LCD_VCPWC assertion to starting output of the display data (LCD_DATA) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) in the power-on sequence of the LCD module in frame units.
13	ONA1	1	R/W	
12	ONA0	1	R/W	
				Specify to the value of (the period)-1.
				This period is the (a) period in figures 26.4 to 26.7, Power-Supply Control Sequence and States of the LCD Module.
11	ONB3	0	R/W	LCDC Power-On Sequence Period
10	ONB2	1	R/W	Set the period from starting output of the display data (LCD_DATA) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) to the LCD_VEPWC assertion in the power-on sequence of the LCD module in frame units.
9	ONB1	1	R/W	
8	ONB0	0	R/W	
				Specify to the value of (the period)-1.
				This period is the (b) period in figures 26.4 to 26.7, Power-Supply Control Sequence and States of the LCD Module.
7	OFFE3	0	R/W	LCDC Power-Off Sequence Period
6	OFFE2	0	R/W	Set the period from LCD_VEPWC negation to stopping output of the display data (LCD_DATA) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) in the power-off sequence of the LCD module in frame units.
5	OFFE1	0	R/W	
4	OFFE0	0	R/W	
				Specify to the value of (the period)-1.
				This period is the (e) period in figures 26.4 to 26.7, Power-Supply Control Sequence and States of the LCD Module.

Bit	Bit Name	Initial Value	R/W	Description
3	OFFF3	1	R/W	LCDC Power-Off Sequence Period
2	OFFF2	1	R/W	Set the period from stopping output of the display data (LCD_DATA) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) to LCD_VCPWC negation to in the power-off sequence of the LCD module in frame units.
1	OFFF1	1	R/W	
0	OFFF0	1	R/W	
				Specify to the value of (the period)-1.
				This period is the (f) period in figures 26.4 to 26.7, Power-Supply Control Sequence and States of the LCD Module.

26.3.19 LCDC Control Register (LDCNTR)

LDCNTR specifies start and stop of display by the LCDC.

When 1s are written to the DON2 bit and the DON bit, the LCDC starts display. Turn on the LCD module following the sequence set in the LDPMMR and LDCNTR. The sequence ends when the LPS[1:0] value changes from B'00 to B'11. Do not make any action to the DON bit until the sequence ends.

When 0 is written to the DON bit, the LCDC stops display. Turn off the LCD module following the sequence set in the LDPMMR and LDCNTR. The sequence ends when the LPS[1:0] value changes from B'11 to B'00. Do not make any action to the DON bit until the sequence ends.

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	DON2	0	R/W	Display On 2
				Specifies the start of the LCDC display operation.
				0: LCDC is being operated or stopped
				1: LCDC starts operation
				When this bit is read, always read as 0. Write 1 to this bit only when starting display. If a value other than 0 is written when starting display, the operation is not guaranteed. When 1 is written to, it resumes automatically to 0. Accordingly, this bit does not need to be cleared by writing 0.

Bit	Bit Name	Initial Value	R/W	Description
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DON	0	R/W	Display On Specifies the start and stop of the LCDC display operation. The control sequence state can be checked by referencing the LPS[1:0] of LDPMMR. 0: Display-off mode: LCDC is stopped 1: Display-on mode: LCDC operates

Notes: 1. Write H'0011 to LDCNTR when starting display and H'0000 when completing display. Data other than H'0011 and H'0000 must not be written to.
2. Setting bit DON2 to 1 makes the contents of the palette RAM undefined. Before writing to the palette RAM, set bit DON2 to 1.

26.3.20 LCDC User Specified Interrupt Control Register (LDUINTR)

LDUINTR sets whether the user specified interrupt is generated, and indicates its processing state. This interrupt is generated at the time when image data which is set by the line number register (LDUINTLNR) in LCDC is read from VRAM.

This LCDC issues the interrupts (LCDCI): user specified interrupt by this register, memory access interrupt by the LCDC interrupt control register (LDINTR), and OR of Vsync interrupt output. This register and LCDC interrupt control register (LDINTR) settings affect the interrupt operation independently.

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	UINTEN	0	R/W	User Specified Interrupt Enable Sets whether generate an LCDC user specified interrupt. 0: LCDC user specified interrupt is not generated 1: LCDC user specified interrupt is generated

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
0	UINTS	0	R/W	User Specified Interrupt State This bit is set to 1 at the time an LCDC user specified interrupt is generated (set state). During the user specified interrupt handling routine, this bit should be cleared by writing 0 to it. 0: LCDC did not generate a user specified interrupt or has been informed that the generated user specified interrupt has completed 1: LCDC has generated a user specified interrupt and has not yet been notified that the generated user specified interrupt has completed

Note: Interrupt processing flow:

1. Interrupt signal is input
2. LDINTR is read
3. If MINTS, FINTS, VSINTS, or VEINTS is 1, a generated interrupt is memory access interrupt, flame end interrupt, Vsync rising edge interrupt, or Vsync falling edge interrupt. Processing for each interrupt is performed.
4. If MINTS, FINTS, VSINTS, or VEINTS is 0, a generated interrupt is not memory access interrupt, flame end interrupt, Vsync rising edge interrupt, or Vsync falling edge interrupt.
5. UINTS is read.
6. If UINTS is 1, a generated interrupt is a user specified interrupt. Process for user specified interrupt is carried out.
7. If UINTS is 0, a generated interrupt is not a user specified interrupt. Other processing is performed.

26.3.21 LCDC User Specified Interrupt Line Number Register (LDUINTLNR)

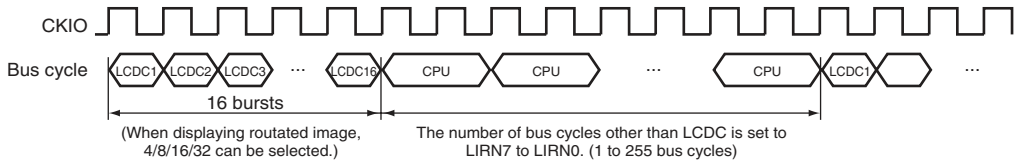
LDUINTLNR sets the point where the user specified interrupt is generated. Setting is done in horizontal line units.

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	UINTLN10	0	R/W	User Specified Interrupt Generation Line Number Specifies the line in which the user specified interrupt is generated (line units). Set (the number of lines in which interrupts are generated) −1 Example: Generate the user specified interrupt in the 80th line. UINTLN = 160/2 − 1 = 79 = H'04F
9	UINTLN9	0	R/W	
8	UINTLN8	0	R/W	
7	UINTLN7	0	R/W	
6	UINTLN6	1	R/W	
5	UINTLN5	0	R/W	
4	UINTLN4	0	R/W	
3	UINTLN3	1	R/W	
2	UINTLN2	1	R/W	
1	UINTLN1	1	R/W	
0	UINTLN0	1	R/W	

Notes: 1. When using the LCD module with STN/TFT display, the setting value of this register should be equal to lower than the vertical display line number (VDLN) in LDVDLNR.
2. When using the LCD module with DSTN display, the setting value of this register should be equal to or lower than half the vertical display line number (VDLN) in LDVDLNR. The user specified interrupt is generated at the point when the LCDC read the specified piece of image data in lower display from VRAM.

26.3.22 LCDC Memory Access Interval Number Register (LDLIRNR)

LDLIRNR controls the bus cycle interval when the LCDC reads VRAM. When LDLIRNR is set to other than H′00, the LCDC does not access VRAM until the specified number of bus cycles (accessing the external memory or on-chip registers) has been performed by the CPU/DMAC/USBH. When LDLIRNR is set to H'00 (initial value), the LCDC accesses the VRAM, the CPU/DMAC/USBH performs one bus cycle, and then the LCDC accessed VRAM.



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	LIRN7 to LIRN0	All 0	R/W	VRAM Read Bus Cycle Interval Specifies the number of the CPU/DMAC/USBH bus cycles which can be performed during burst bus cycles to read VRAM by LCDC. H'00: one bus cycle H'01: one bus cycle : H'FF: 255 bus cycles

26.4 Operation

26.4.1 LCD Module Sizes which can be Displayed in this LCDC

This LCDC is capable of controlling displays with up to 1024 × 1024 dots and 16 bpp (bits per pixel). The image data for display is stored in VRAM, which is shared with the CPU. This LCDC should read the data from VRAM before display.

This LSI has a maximum 32-burst memory read operation and a 2.4-kbyte line buffer, so although a complete breakdown of the display is unlikely, there may be some problems with the display depending on the combination. A recommended size at the frame rate of 60 Hz is 320 × 240 dots in 16 bpp or 640 × 480 dots in 8 bpp.

As a rough standard, the bus occupation ratio shown below should not exceed 40%.

$$\text{Bus occupation ratio (\%)} = \frac{\text{Overhead coefficient x Total number of display pixels ((HDCN + 1) x 8 x (VDLN + 1)) x Frame rate (Hz) x Number of colors (bpp)}}{\text{CKIO (Hz) x Bus width (bit)}} \times 100$$

The overhead coefficient becomes 1.375 when the CL2 SDRAM is connected to a 32-bit data bus and 1.188 when connected to a 16-bit data bus.

Figure 26.2 shows the valid display and the retrace period.

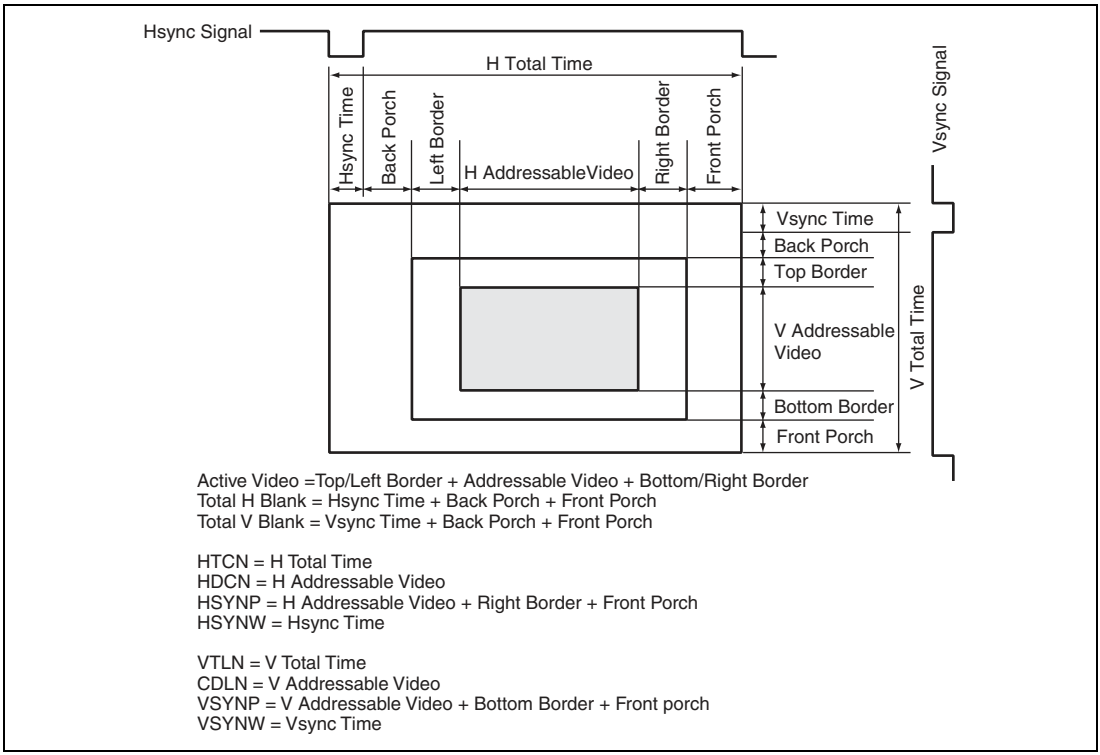


Figure 26.2 Valid Display and the Retrace Period

26.4.2 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (SDRAM)

This LCDC is capable of displaying a landscape-format image on a LCD module by rotating a portrait format image for display by 90 degrees. Only the numbers of colors for each resolution are supported as shown in tables 26.3 and 26.4. The size of the SDRAM (the number of column address bits) and its burst length are limited to read the SDRAM continuously.

The number of colors for display, SDRAM column addresses, and LCDC burst length are shown table 26.3 and 26.4.

A monochromatic LCD module is necessary for the display of images in the above monochromatic formats. A color LCD module is necessary for the display of images in the above color formats.

Table 26.3 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (32-bit SDRAM)

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
240 × 320	320 × 240	Monochrome	4 bpp (packed)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			4 bpp (unpacked)	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			6 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
		Color	8 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			16 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts
234 × 320	320 × 234	Monochrome	6 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
		Color	16 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
80 × 160	160 × 80	Monochrome	2 bpp	8 bits	—
				9 bits	—
				10 bits	—
			4 bpp (packed)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			6 bpp	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
		Color	4 bpp (packed)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			8 bpp	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			16 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
64 × 128	128 × 64	Monochrome	1 bpp	8 bits	—
				9 bits	—
				10 bits	—
			2 bpp	8 bits	—
				9 bits	—
				10 bits	—
			4 bpp (packed)	8 bits	—
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			6 bpp	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
		Color	4 bpp (packed)	8 bits	—
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			8 bpp	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—

Note: * Specify the data of the number of line specified as burst length that can be stored in address of SDRAM same as that of ROW.

Table 26.4 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (16-bit SDRAM)

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
240 × 320	320 × 240	Monochrome	4 bpp (packed)	8 bits	Not more than 4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			4 bpp (unpacked)	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts
			6 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts
		Color	8 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts
			16 bpp	8 bits	Unusable
				9 bits	Unusable
				10 bits	4 bursts
234 × 320	320 × 234	Monochrome	6 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts
		Color	16 bpp	8 bits	Unusable
				9 bits	Unusable
				10 bits	4 bursts

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
80 × 160	160 × 80	Monochrome	2 bpp	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			4 bpp (packed)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			4 bpp (unpacked)	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			6 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
		Color	4 bpp (packed)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			4 bpp (unpacked)	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			8 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			16 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
64 × 128	128 × 64	Monochrome	1 bpp	8 bits	—
				9 bits	—
				10 bits	—
			2 bpp	8 bits	—
				9 bits	—
				10 bits	—
			4 bpp (packed)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			6 bpp	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
		Color	4 bpp (packed)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			8 bpp	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—

Note: * Set the data of the number of line specified as burst length that can be stored in address of SDRAM same as that of ROW.

26.4.3 Color Palette Specification

(1) Color Palette Register

This LCDC has a color palette which outputs 24 bits of data per entry and is able to simultaneously hold 256 entries. The color palette thus allows the simultaneous display of 256 colors chosen from among 16-M colors.

The procedure below may be used to set up color palettes at any time.

1. The PALEN bit in the LDPALCR is 0 (initial value); normal display operation
2. Access LDPALCR and set the PALEN bit to 1; enter color-palette setting mode after three cycles of peripheral clock.
3. Access LDPALCR and confirm that the PALS bit is 1.
4. Access LDPR00 to LDPRFF and write the required values to the PALD00 to PALDFF bits.
5. Access LDPALCR and clear the PALEN bit to 0; return to normal display mode after a cycle of peripheral clock.

A 0 is output on the LCDC display data output (LCD_DATA) while the PALS bit in LDPALCR is set to 1.

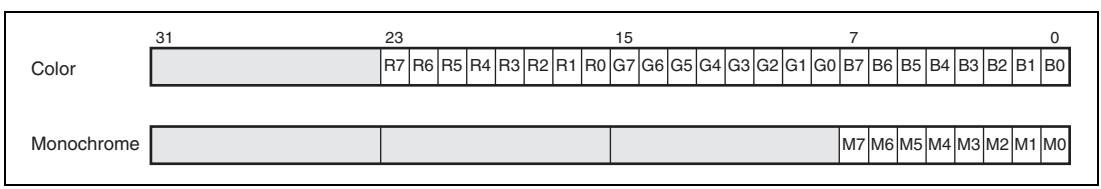


Figure 26.3 Color-Palette Data Format

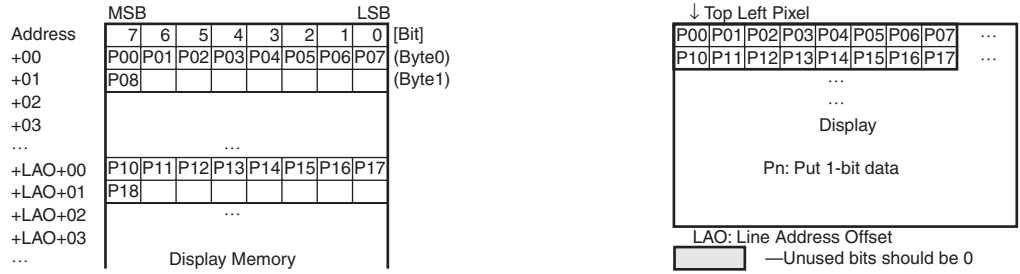
PALDnn color and gradation data should be set as above.

For a color display, PALDnn[23:16], PALDnn[15:8], and PALDnn[7:0] respectively hold the R, G, and B data. Although the bits PALDnn[18:16], PALDnn[9:8], and PALDnn[2:0] exist, no memory is associated with these bits. PALDnn[18:16], PALDnn[9:8], and PALDnn[2:0] are thus not available for storing palette data. The numbers of valid bits are thus R: 5, G: 6, and B: 5. A 24-bit (R: 8 bits, G: 8 bits, and B: 8 bits) data should, however, be written to the palette-data registers. When the values for PALDnn[23:19], PALDnn[15:10], or PALDnn[7:3] are not 0, 1 or 0 should be written to PALDnn[18:16], PALDnn[9:8], or PALDnn[2:0], respectively. When the values of PALDnn[23:19], PALDnn[15:10], or PALDnn[7:3] are 0, 0s should be written to PALDnn[18:16], PALDnn[9:8], or PALDnn[2:0], respectively. Then 24 bits are extended.

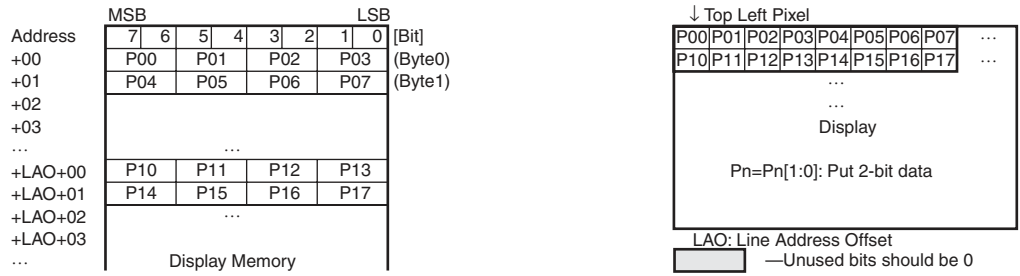
Grayscale data for a monochromatic display should be set in PALDnn[7:3]. PALDnn[23:8] are all "don't care". When the value in PALDnn[7:3] is not 0, 1s should be written to PALDnn[2:0]. When the value in PALDnn[7:3] is 0, 0s should be written to PALDnn[2:0]. Then 8 bits are extended.

26.4.4 Data Format

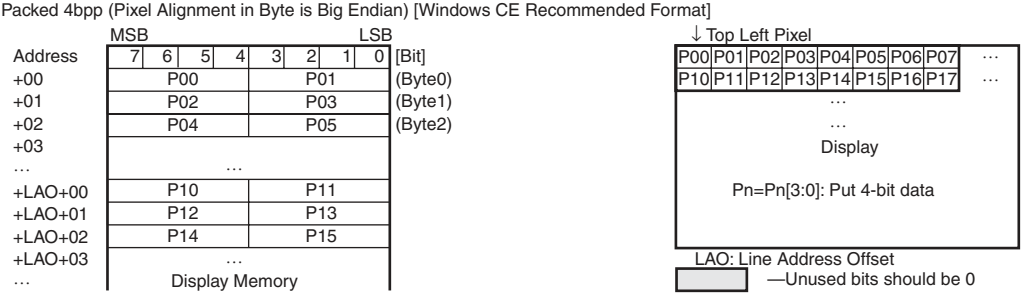
1. Packed 1bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]



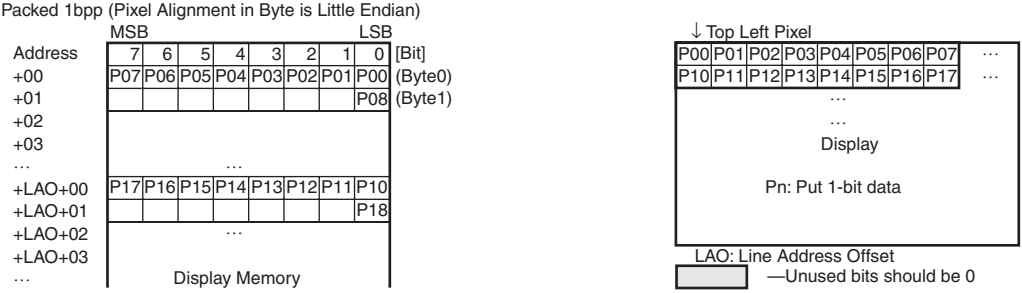
2. Packed 2bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]



3. Packed 4bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]

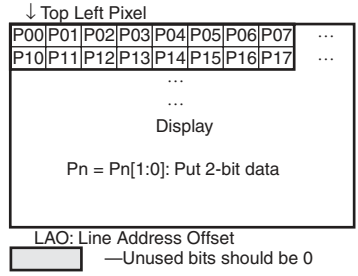


4. Packed 1bpp (Pixel Alignment in Byte is Little Endian)



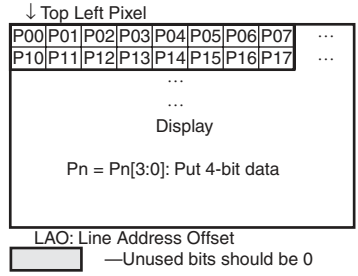
5. Packed 2bpp (Pixel Alignment in Byte is Little Endian)

Address	MSB 7	6	5	4	3	2	1	0 LSB	[Bit]
+00	P03		P02		P01		P00		(Byte0)
+01	P07		P06		P05		P04		(Byte1)
+02									
+03									
…			…						
+LAO+00	P13		P12		P11		P10		
+LAO+01	P17		P16		P15		P14		
+LAO+02			…						
+LAO+03									
…	Display Memory								



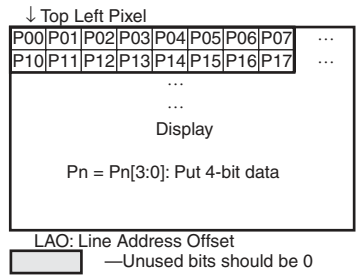
6. Packed 4bpp (Pixel Alignment in Byte is Little Endian)

Address	MSB 7	6	5	4	3	2	1	0 LSB	[Bit]
+00	P01				P00				(Byte0)
+01	P03				P02				(Byte1)
+02	P05				P04				(Byte2)
+03									
…			…						
+LAO+00	P11				P10				
+LAO+01	P13				P12				
+LAO+02	P15				P14				
+LAO+03			…						
…	Display Memory								



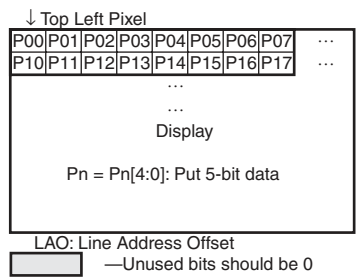
7. Unpacked 4bpp [Windows CE Recommended Format]

Address	MSB 7	6	5	4	3	2	1	0 LSB	[Bit]
+00					P00				(Byte0)
+01					P01				(Byte1)
+02					P02				(Byte2)
+03									
…			…						
+LAO+00					P10				
+LAO+01					P11				
+LAO+02					P12				
+LAO+03			…						
…	Display Memory								

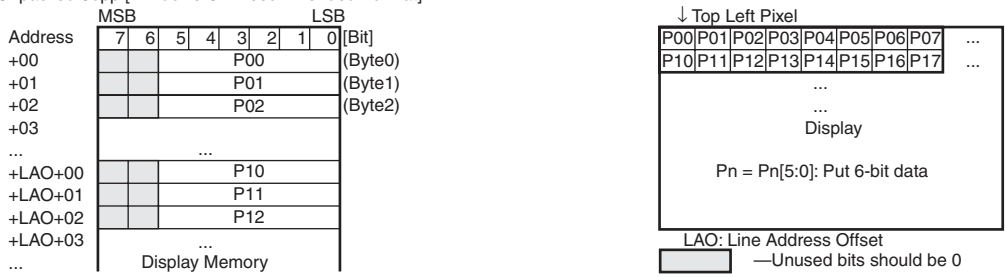


8. Unpacked 5bpp [Windows CE Recommended Format]

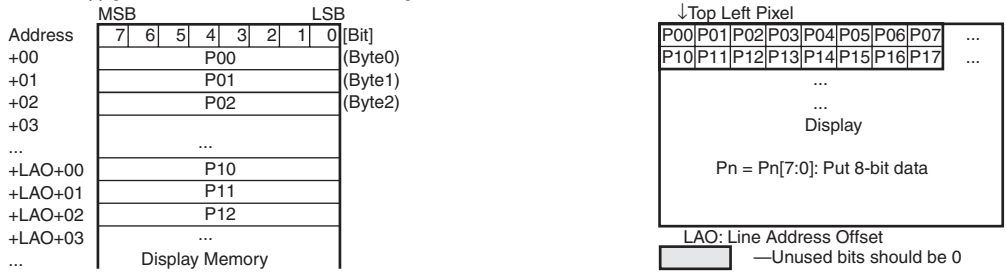
Address	MSB 7	6	5	4	3	2	1	0 LSB	[Bit]
+00				P00					(Byte0)
+01				P01					(Byte1)
+02				P02					(Byte2)
+03									
…			…						
+LAO+00				P10					
+LAO+01				P11					
+LAO+02				P12					
+LAO+03			…						
…	Display Memory								



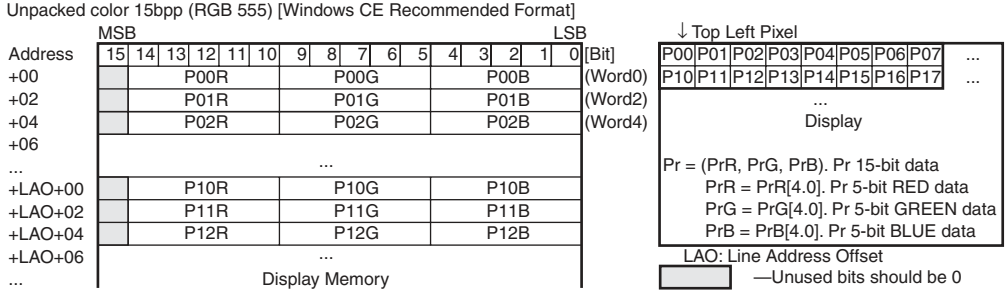
9. Unpacked 6bpp [Windows CE Recommended Format]



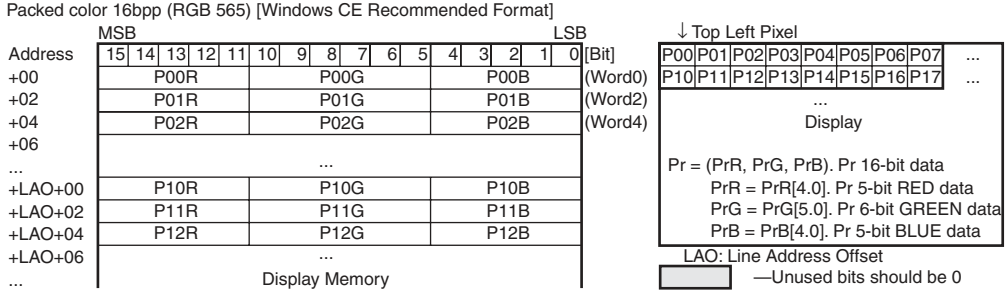
10. Packed 8bpp [Windows CE Recommended Format]



11. Unpacked color 15bpp (RGB 555) [Windows CE Recommended Format]



12. Packed color 16bpp (RGB 565) [Windows CE Recommended Format]



26.4.5 Setting the Display Resolution

The display resolution is set up in LDHCNR, LDHSYNR, LDVDLNR, LDVTLNR, and LDVSYNR. The LCD current-alternating period for an STN or DSTN display is set by using the LDACLNR. The initial values in these registers are typical settings for VGA (640 × 480 dots) on an STN or DSTN display.

The clock to be used is set with the LDICKR. The LCD module frame rate is determined by the display interval + retrace line interval (non-display interval) for one screen set in a size related register and the frequency of the clock used.

This LCDC has a Vsync interrupt function so that it is possible to issue an interrupt at the beginning of each vertical retrace line period (to be exact, at the beginning of the line after the last line of the display). This function is set up by using the LDINTR.

26.4.6 Power Management Registers

An LCD module normally requires a specific sequence for processing to do with the cutoff of the input power supply. Settings in LDPMMR, LDPSPR, and LDCNTR, in conjunction with the LCD power-supply control pins (LCD_VCPWC, LCD_VEPWC, and LCD_DON), are used to provide processing of power-supply control sequences that suits the requirements of the LCD module.

Figures 26.4 to 26.7 are summary timing charts for power-supply control sequences and table 26.5 is a summary of available power-supply control sequence periods.

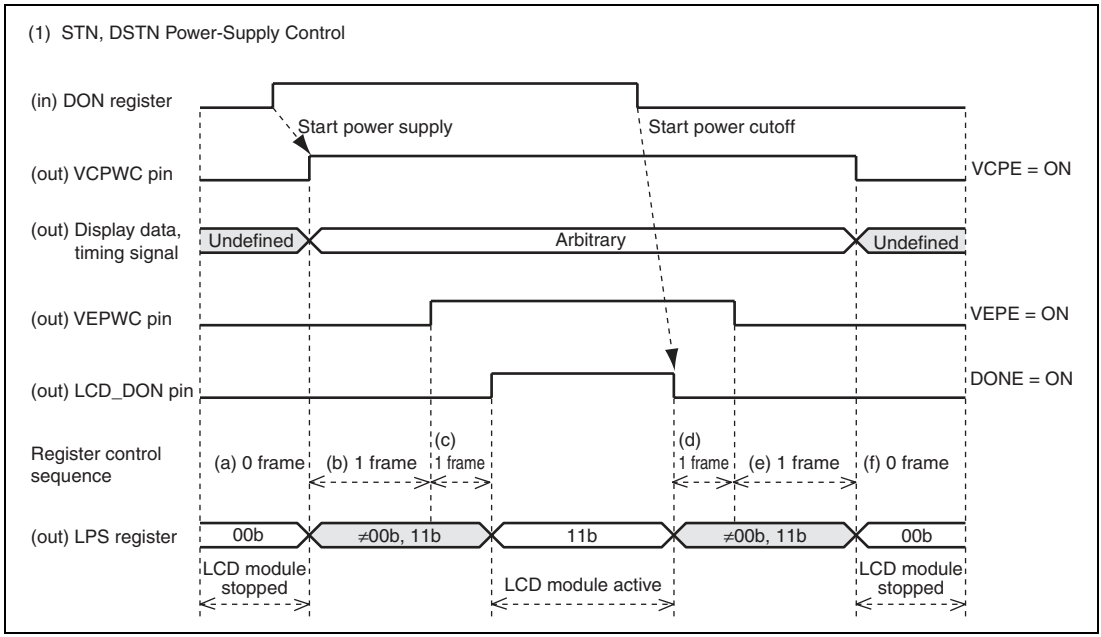


Figure 26.4 Power-Supply Control Sequence and States of the LCD Module

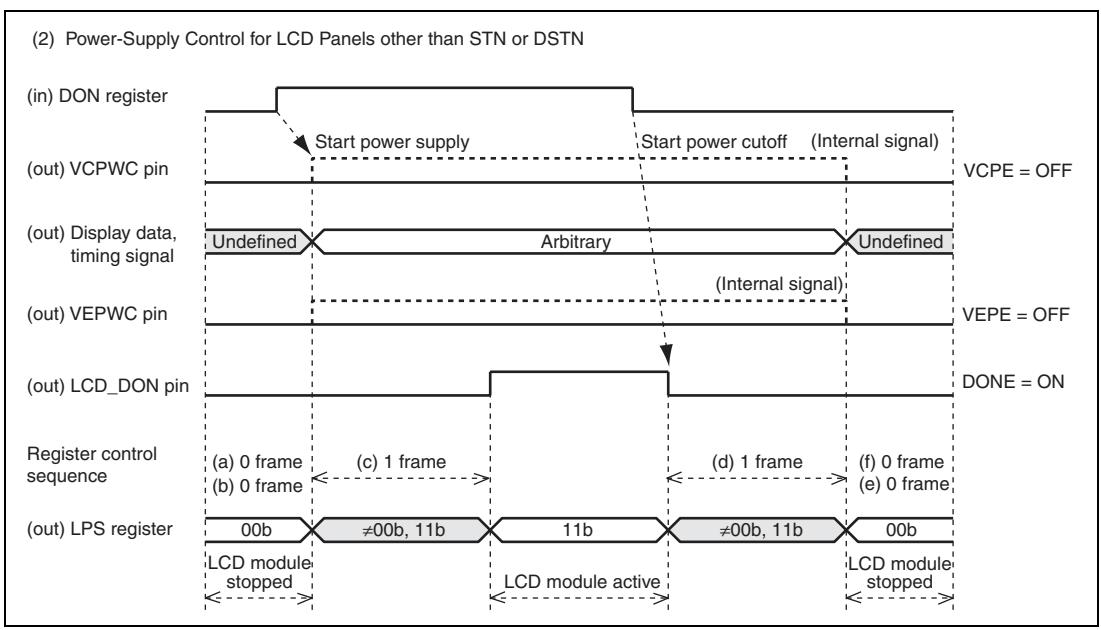


Figure 26.5 Power-Supply Control Sequence and States of the LCD Module

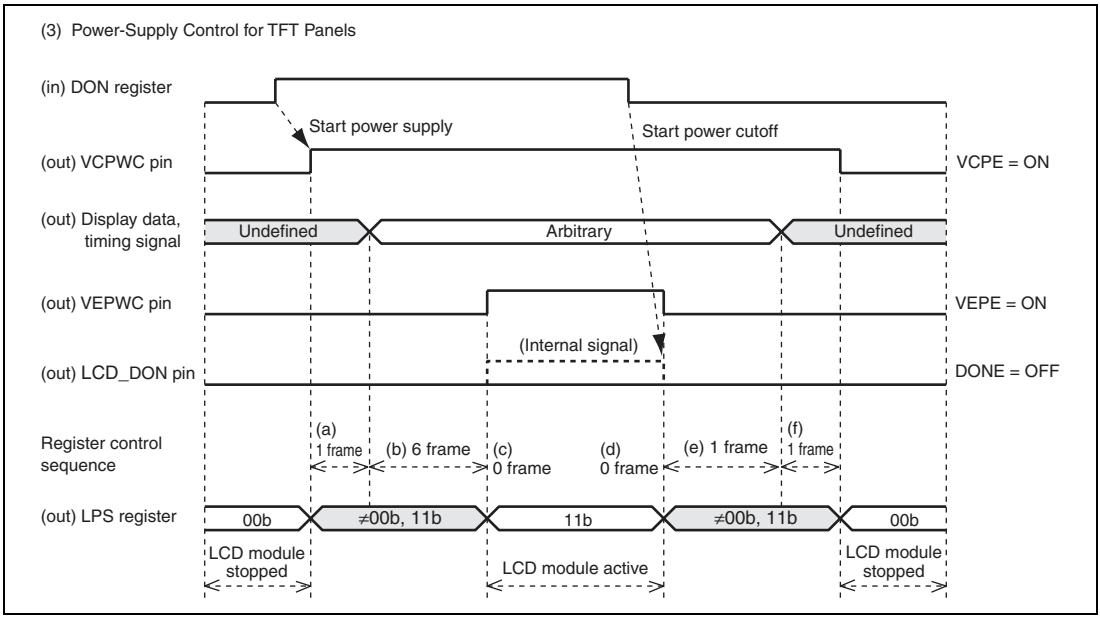


Figure 26.6 Power-Supply Control Sequence and States of the LCD Module

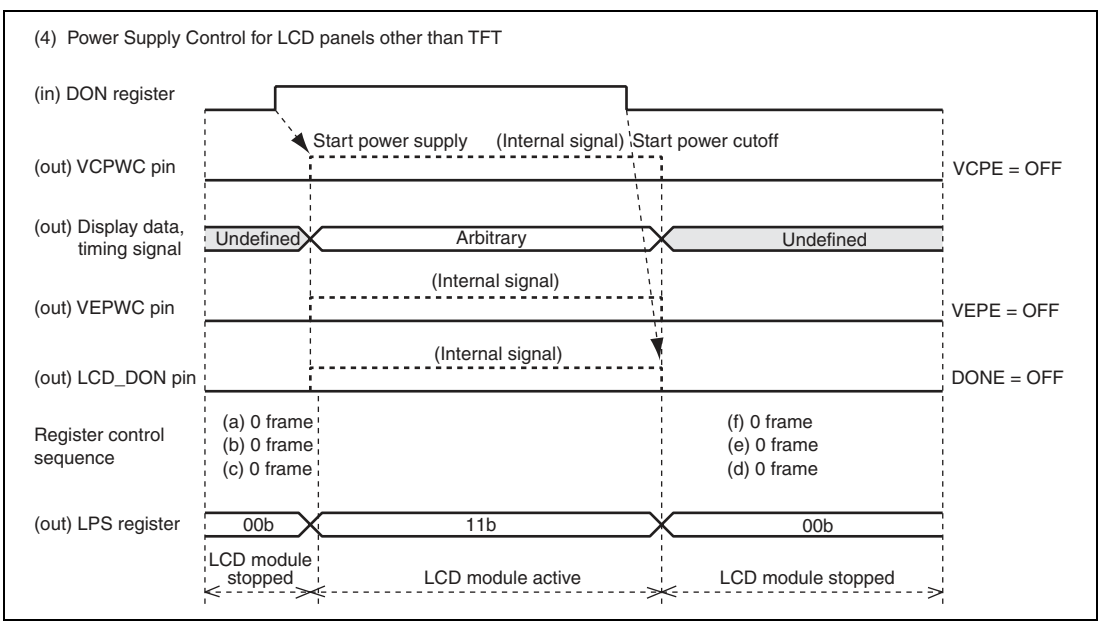


Figure 26.7 Power-Supply Control Sequence and States of the LCD Module

Table 26.5 Available Power-Supply Control-Sequence Periods at Typical Frame Rates

ONX, OFFX Register Value	Frame Rate	
	120 Hz	60 Hz
H'F	(–1+1)/120 = 0.00 (ms)	(–1+1)/60 = 0.00 (ms)
H'0	(0+1)/120 = 8.33 (ms)	(0+1)/60 = 16.67 (ms)
H'1	(1+1)/120 = 16.67 (ms)	(1+1)/60 = 33.33 (ms)
H'2	(2+1)/120 = 25.00 (ms)	(2+1)/60 = 50.00 (ms)
H'3	(3+1)/120 = 33.33 (ms)	(3+1)/60 = 66.67 (ms)
H'4	(4+1)/120 = 41.67 (ms)	(4+1)/60 = 83.33 (ms)
H'5	(5+1)/120 = 50.00 (ms)	(5+1)/60 = 100.00 (ms)
H'6	(6+1)/120 = 58.33 (ms)	(6+1)/60 = 116.67 (ms)
H'7	(7+1)/120 = 66.67 (ms)	(7+1)/60 = 133.33 (ms)
H'8	(8+1)/120 = 75.00 (ms)	(8+1)/60 = 150.00 (ms)
H'9	(9+1)/120 = 83.33 (ms)	(9+1)/60 = 166.67 (ms)
H'A	(10+1)/120 = 91.67 (ms)	(10+1)/60 = 183.33 (ms)
H'B	(11+1)/120 = 100.00 (ms)	(11+1)/60 = 200.00 (ms)
H'C	(12+1)/120 = 108.33 (ms)	(12+1)/60 = 216.67 (ms)
H'D	(13+1)/120 = 116.67 (ms)	(13+1)/60 = 233.33 (ms)
H'E	(14+1)/120 = 125.00 (ms)	(14+1)/60 = 250.00 (ms)

ONA, ONB, ONC, OFFD, OFFE, and OFFF are used to set the power-supply control-sequence periods, in units of frames, from 0 to 15. 1 is subtracted from each register. H'0 to H'E settings select from 1 to15 frames. The setting H'F selects 0 frames.

Actual sequence periods depend on the register values and the frame frequency of the display. The following table gives power-supply control-sequence periods for display frame frequencies used by typical LCD modules.

- When ONB is set to H'6 and display's frame frequency is 120 Hz
  The display's frame frequency is 120 Hz. 1 frame period is thus 8.33 (ms) = 1/120 (sec).
  The power-supply input sequence period is 7 frames because ONB setting is subtracted by 1.
  As a result, the sequence period is 58.33 (ms) = 8.33 (ms) × 7.

Table 26.6 LCDC Operating Modes

Mode		Function
Display on (LCDC active)	Register setting: DON = 1	Fixed resolution, the format of the data for display is determined by the number of colors, and timing signals are output to the LCD module.
Display off (LCDC stopped)	Register setting: DON = 0	Register access is enabled. Fixed resolution, the format of the data for display is determined by the number of colors, and timing signals are not output to the LCD module.

Table 26.7 LCD Module Power-Supply States

(STN, DSTN module)

State	Power Supply for Logic	Display Data, Timing Signal	Power Supply for High-Voltage Systems	DON Signal
Control Pin	LCD_VCPWC	LCD_CL2, LCD_CL1, LCD_FLM, LCD_M_DISP, LCD_DATA	LCD_VEPWC	LCD_DON
Operating State	Supply	Supply	Supply	Supply
(Transitional State)	Supply	Supply	Supply	
	Supply	Supply		
	Supply			
Stopped State				

(TFT module)

State	Power Supply for Logic	Display Data, Timing Signal	Power Supply for High-Voltage Systems
Control Pin	LCD_VCPWC	LCD_CL2, LCD_CL1, LCD_FLM, LCD_M_DISP, LCD_DATA	LCD_VEPWC
Operating State	Supply	Supply	Supply
(Transitional State)	Supply	Supply	
	Supply		
Stopped State			

The table above shows the states of the power supply, display data, and timing signals for the typical LCD module in its active and stopped states. Some of the supply voltages described may not be necessary, because some modules internally generate the power supply required for high-voltage systems from the logic-level power-supply voltage.

Notes on display-off mode (LCDC stopped):

If LCD module power-supply control-sequence processing is in use by the LCDC or the supply of power is cut off while the LCDC is in its display-on mode, normal operation is not guaranteed. In the worst case, the connected LCD module may be damaged.

26.4.7 Operation for Hardware Rotation

Operation in hardware-rotation mode is described below. Hardware-rotation mode can be thought of as using a landscape-format LCD panel instead of a portrait-format LCD panel by placing the landscape-format LCD panel as if it were a portrait-format panel. Whether the panel is intended for use in landscape or portrait format is thus no problem. The panel must, however, be within 320 pixels wide.

When making settings for hardware rotation, the following five differences from the setting for no hardware rotation must be noted. (The following example is for a display at 8 bpp. At 16 bpp, the amount of memory per dot will be doubled. The image size and register values used for rotation will thus be different.)

1. The image data must be prepared for display in the rotated panel. (If 240 × 320 pixels will be required after rotation, 240 × 320 pixel image data must be prepared.)
2. The register settings for the address of the image data must be changed (LDSARU and LDLAOR).

3. LDLAOR should be power of 2 (when the horizontal width after rotation is 240 pixels, LDLAOR should be set to 256).
4. Graphics software should be set up for the number 3 setting.
5. LDSARU should be changed to represent the address of the data for the lower-left pixel of the image rather than of the data for the upper-left pixel of the image.

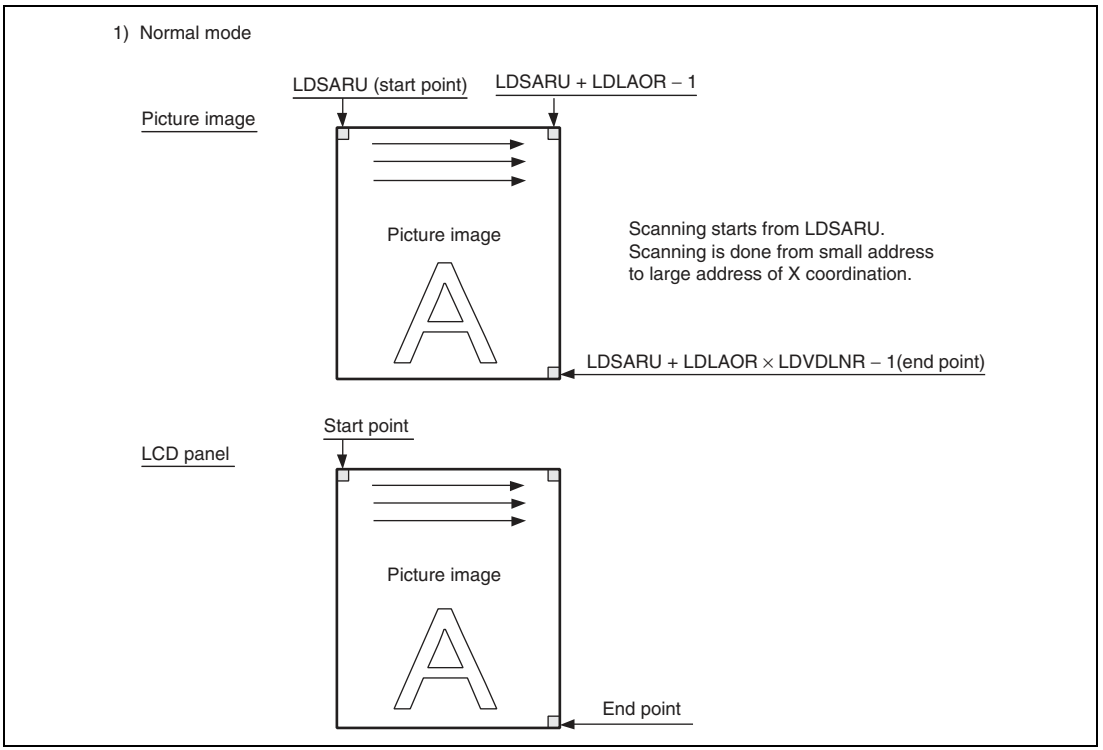


Figure 26.8 Operation for Hardware Rotation (Normal Mode)

For example, the registers have been set up for the display of image data in landscape format (320 × 240), which starts from LDSARU = 0x0c001000, on a 320 × 240 LCD panel. The graphics driver software is complete. Some changes are required to apply hardware rotation and use the panel as a 240 × 320 display. If LDLAOR is 512, the graphics driver software uses this power of 2 as the offset for the calculation of the addresses of Y coordinates in the image data. Before setting ROT to 1, the image data must be redrawn to suit the 240 × 320 LCD panel. LDLAOR will then be 256 because the size has changed and the graphics driver software must be altered accordingly. The point that corresponds to LDSARU moves from the upper left to the lower left of the display, so LDSARU should be changed to 0x0c001000 + 256 * 319.

Note: Hardware rotation allows the use of an LCD panel that has been rotated by 90 degrees. The settings in relation to the LCD panel should match the settings for the LCD panel before rotation. Rotation is possible regardless of the drawing processing carried out by the graphics driver software. However, the sizes in the image data and address offset values which are managed by the graphics driver software must be altered.

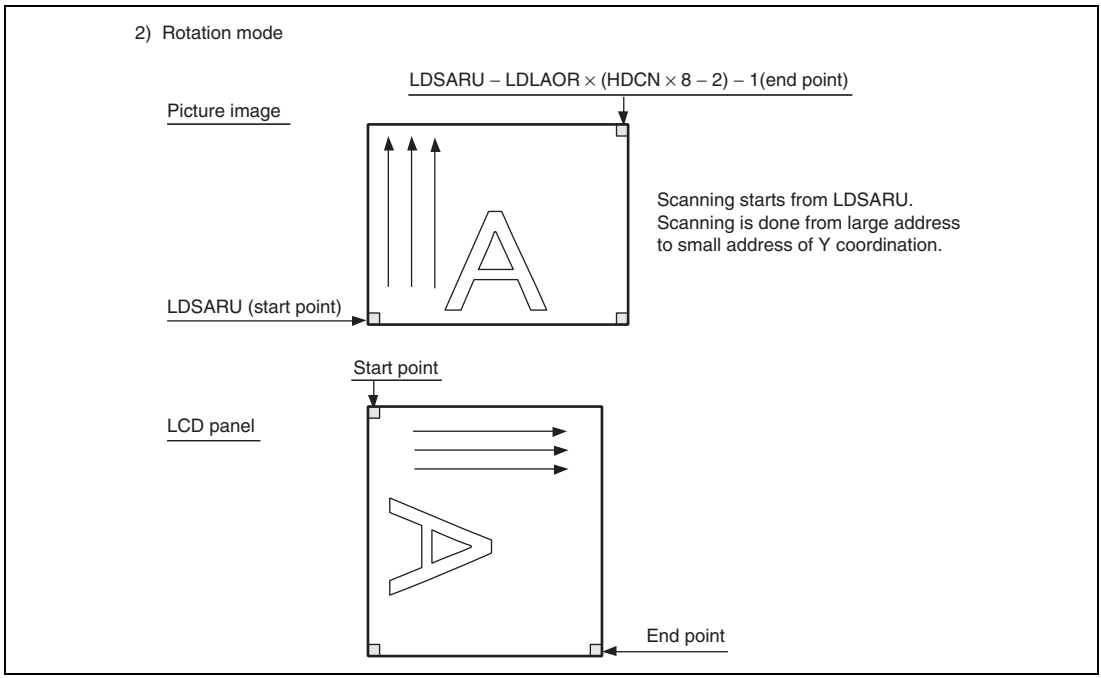


Figure 26.9 Operation for Hardware Rotation (Rotation Mode)

26.5 Clock and LCD Data Signal Examples

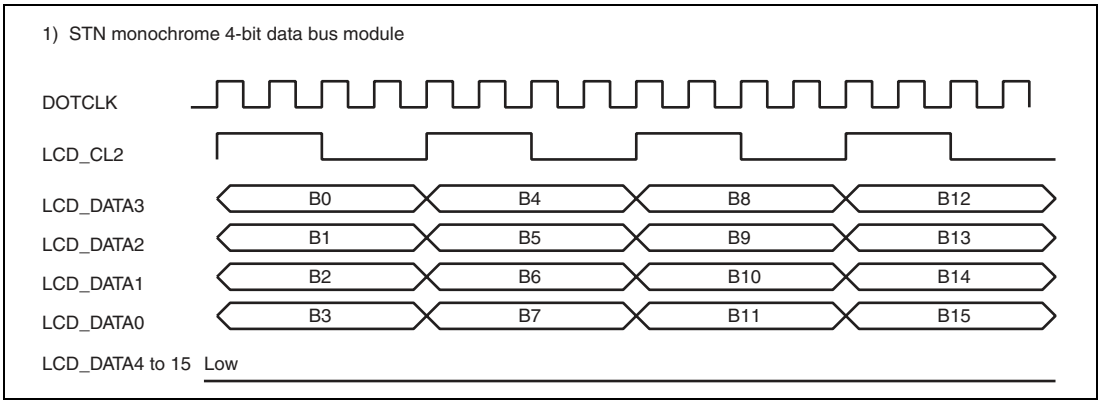


Figure 26.10 Clock and LCD Data Signal Example

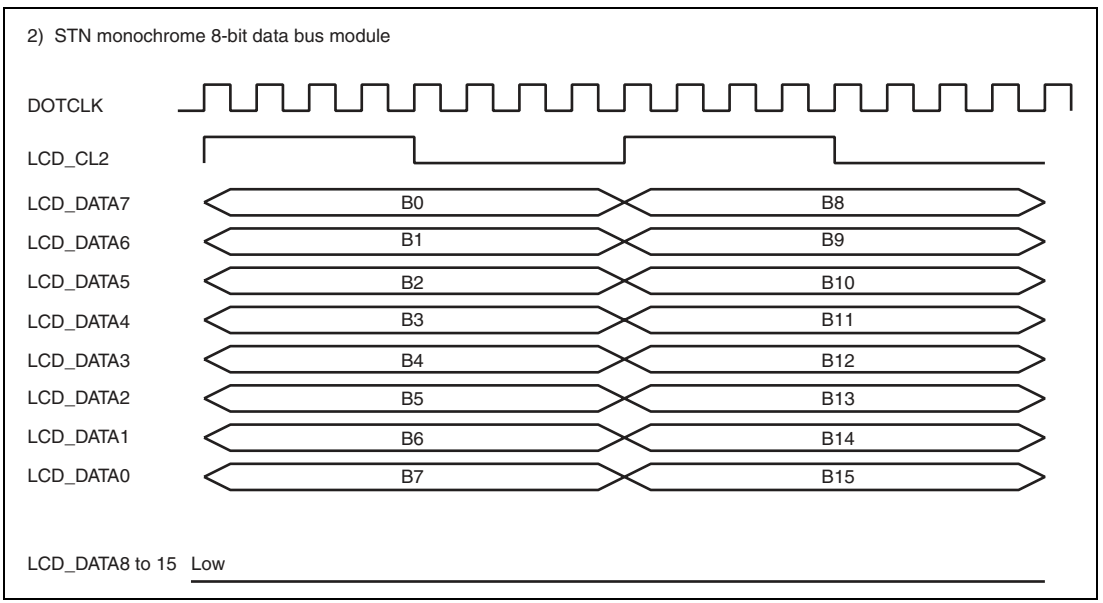


Figure 26.11 Clock and LCD Data Signal Example (STN Monochrome 8-Bit Data Bus Module)

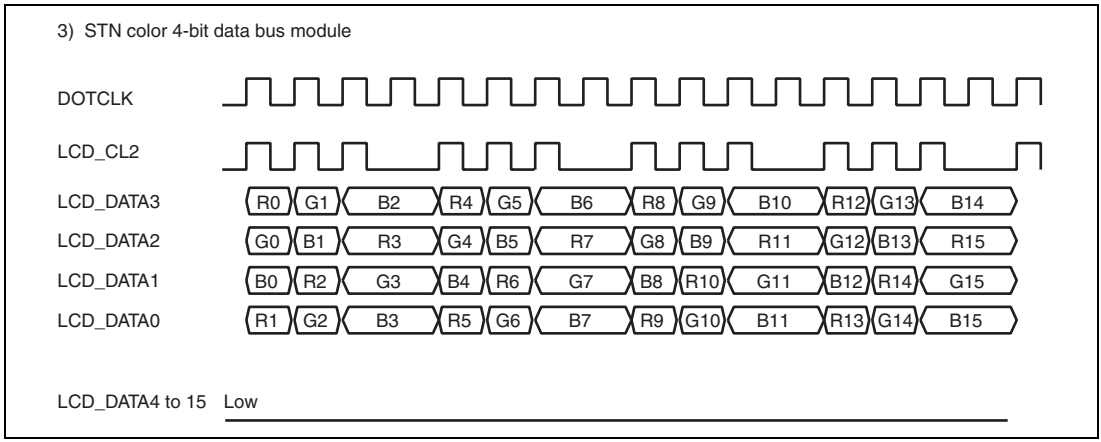


Figure 26.12 Clock and LCD Data Signal Example (STN Color 4-Bit Data Bus Module)

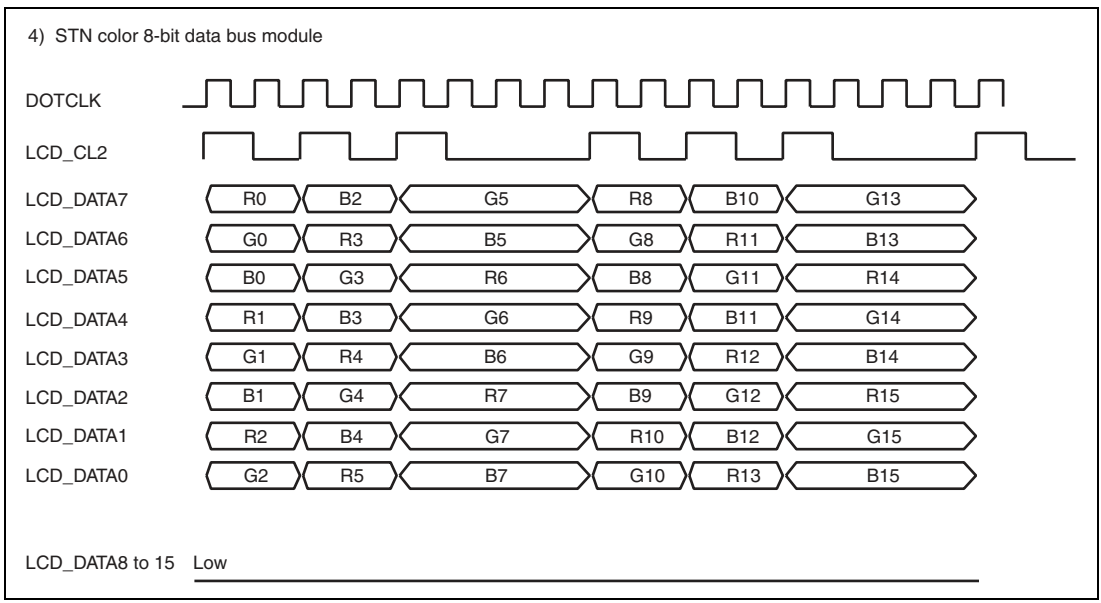


Figure 26.13 Clock and LCD Data Signal Example (STN Color 8-Bit Data Bus Module)

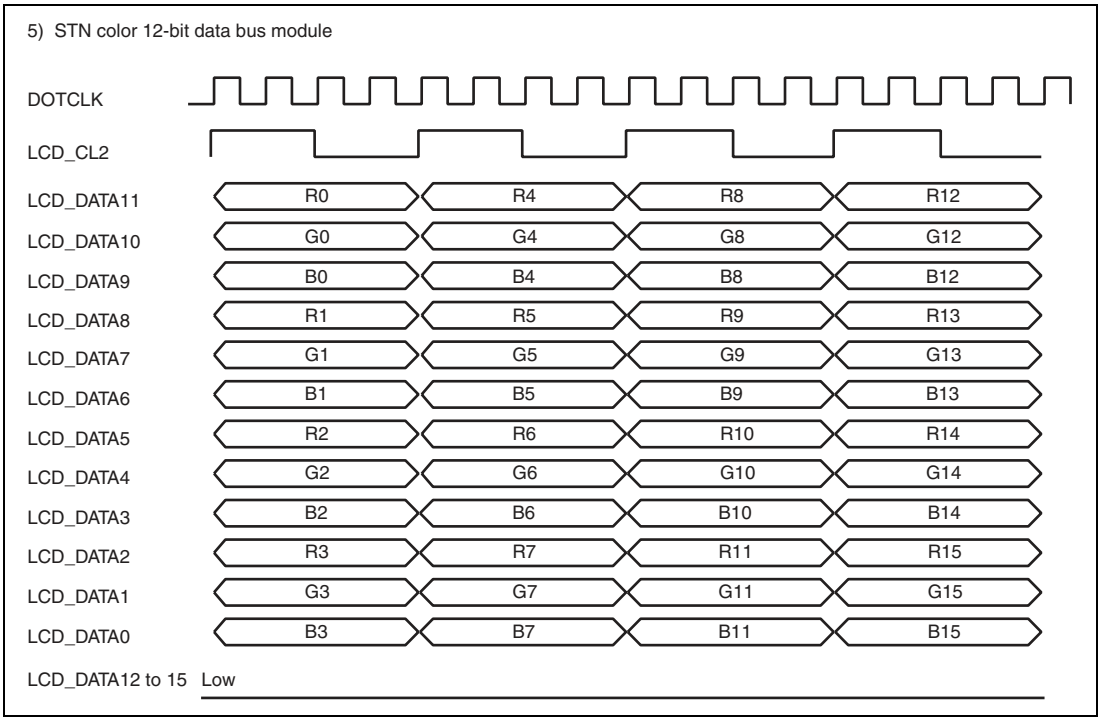


Figure 26.14 Clock and LCD Data Signal Example (STN Color 12-Bit Data Bus Module)

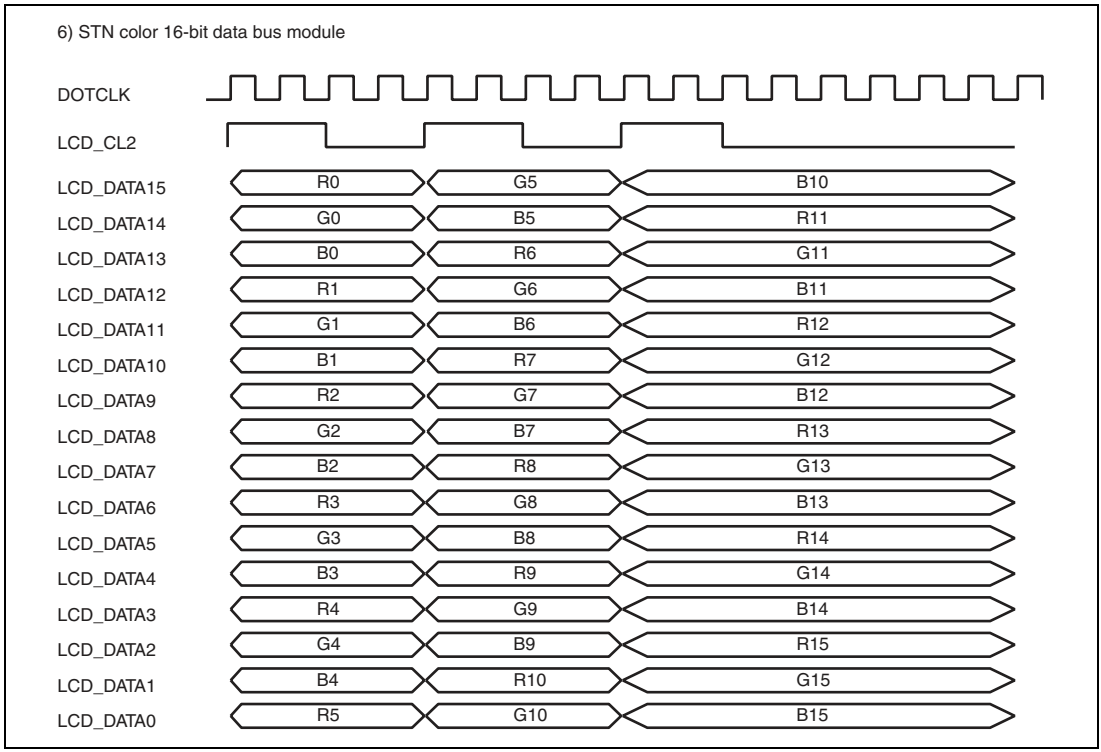


Figure 26.15 Clock and LCD Data Signal Example (STN Color 16-Bit Data Bus Module)

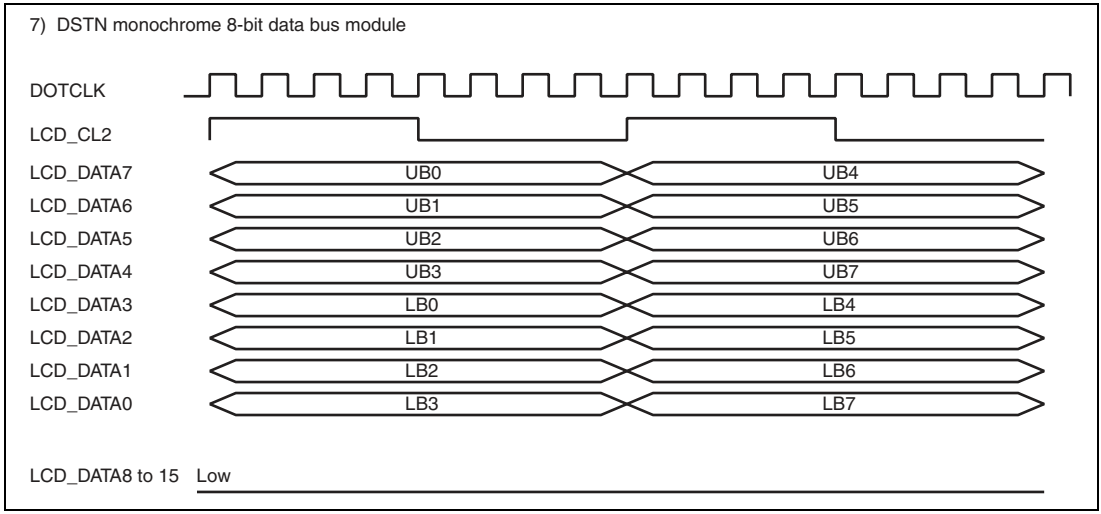


Figure 26.16 Clock and LCD Data Signal Example (DSTN Monochrome 8-Bit Data Bus Module)

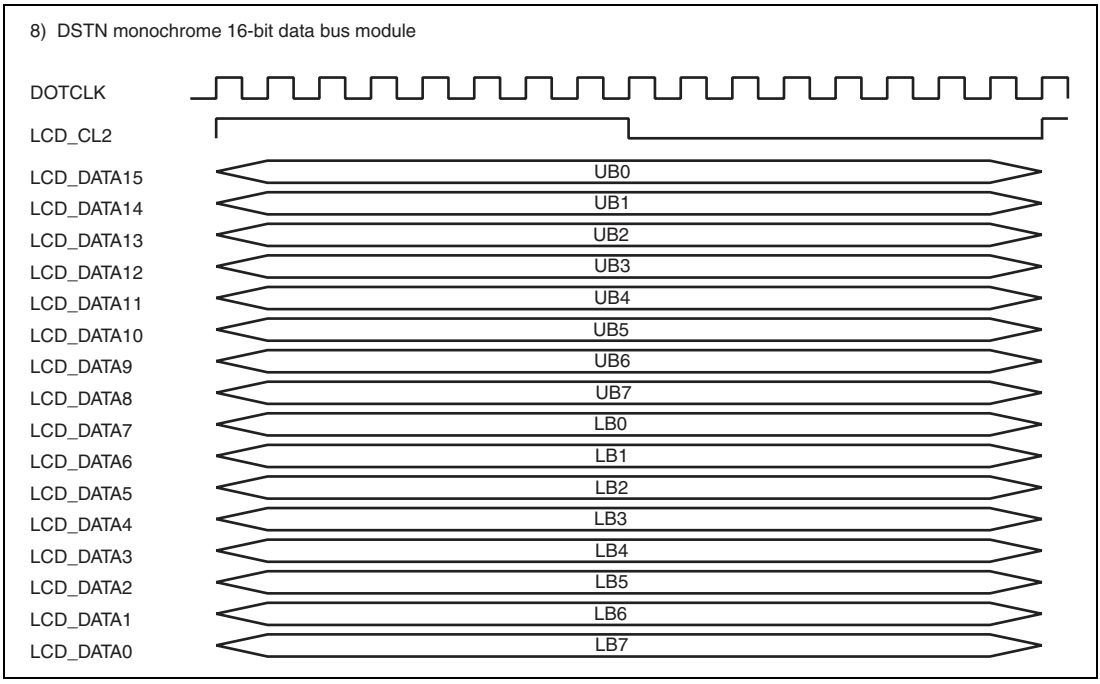


Figure 26.17 Clock and LCD Data Signal Example (DSTN Monochrome 16-Bit Data Bus Module)

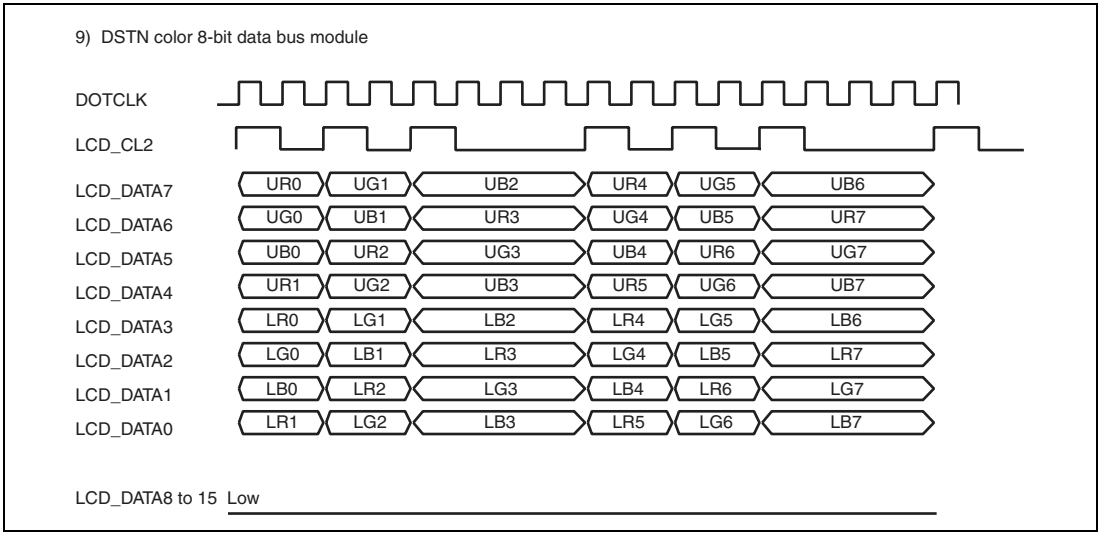


Figure 26.18 Clock and LCD Data Signal Example (DSTN Color 8-Bit Data Bus Module)

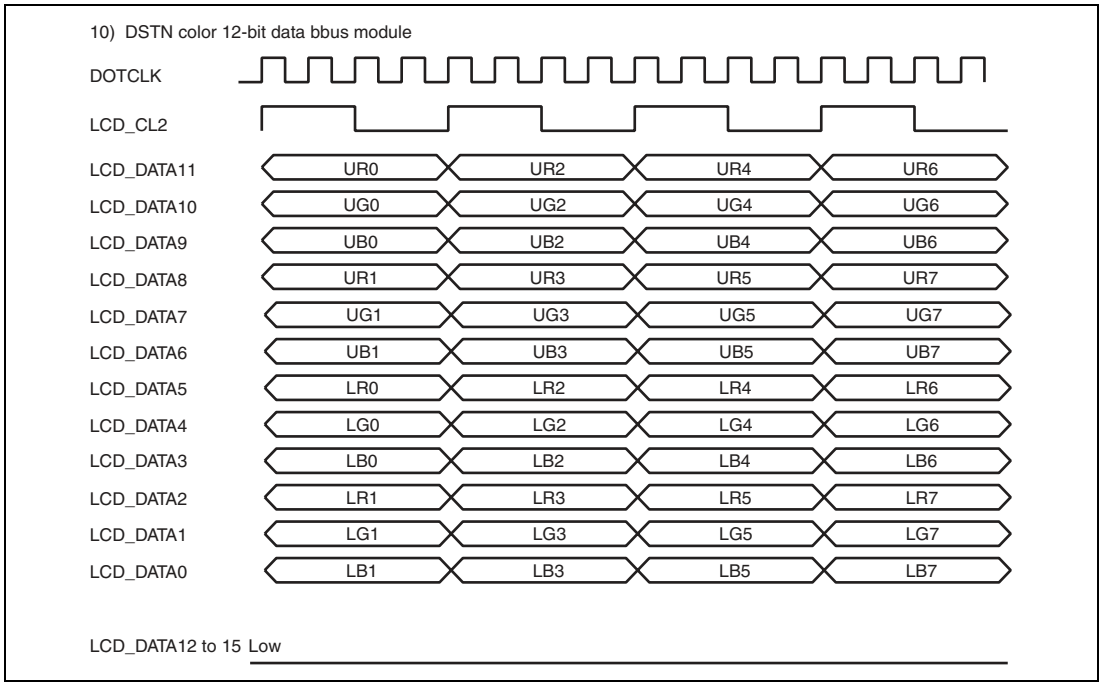


Figure 26.19 Clock and LCD Data Signal Example (DSTN Color 12-Bit Data Bus Module)

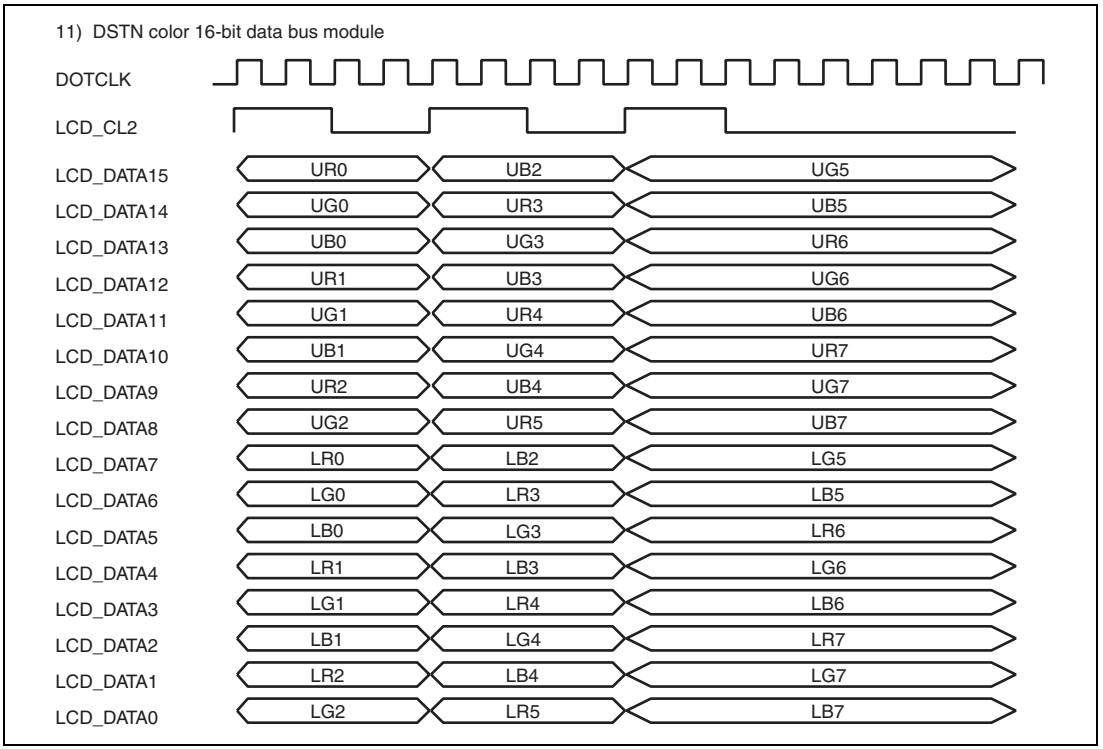


Figure 26.20 Clock and LCD Data Signal Example (DSTN Color 16-Bit Data Bus Module)

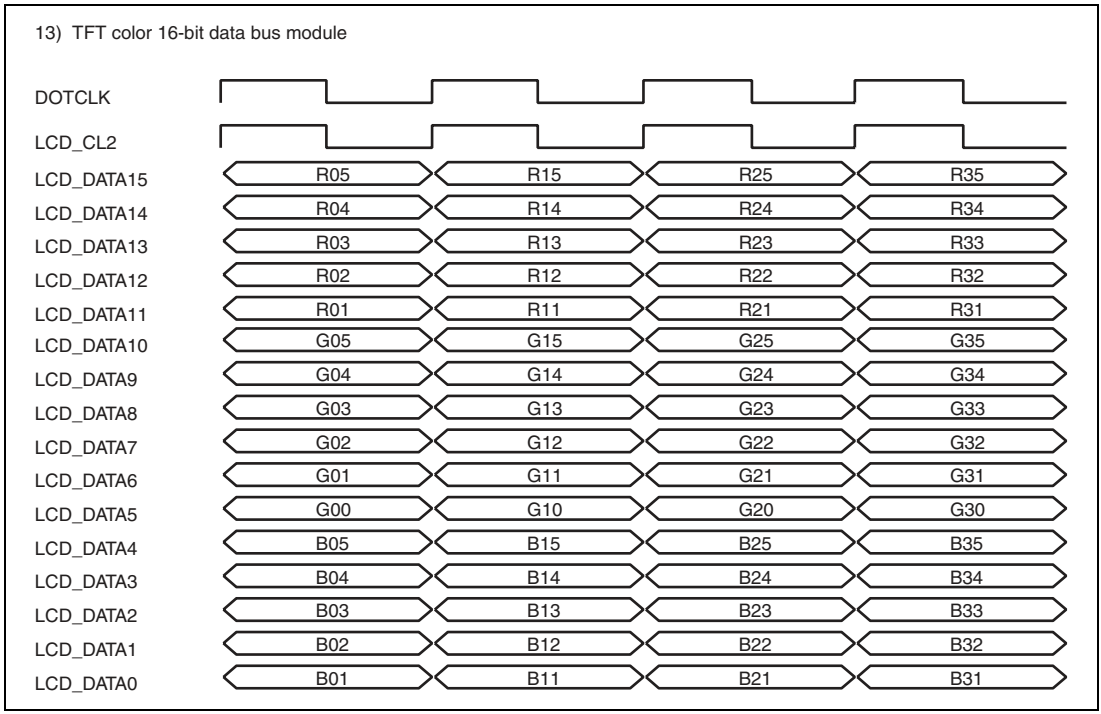


Figure 26.21 Clock and LCD Data Signal Example (TFT Color 16-Bit Data Bus Module)

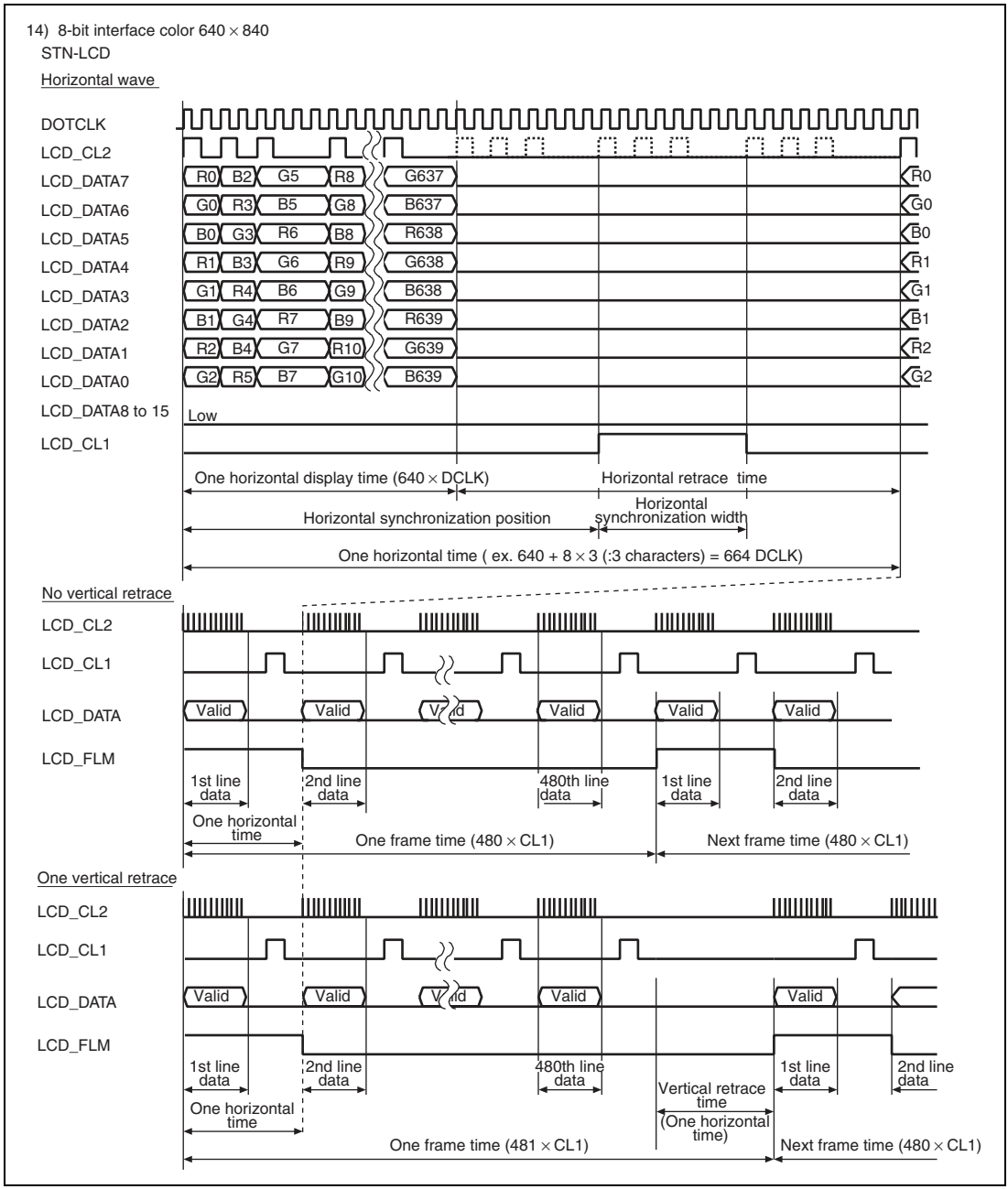


Figure 26.22 Clock and LCD Data Signal Example (8-Bit Interface Color 640 × 480)

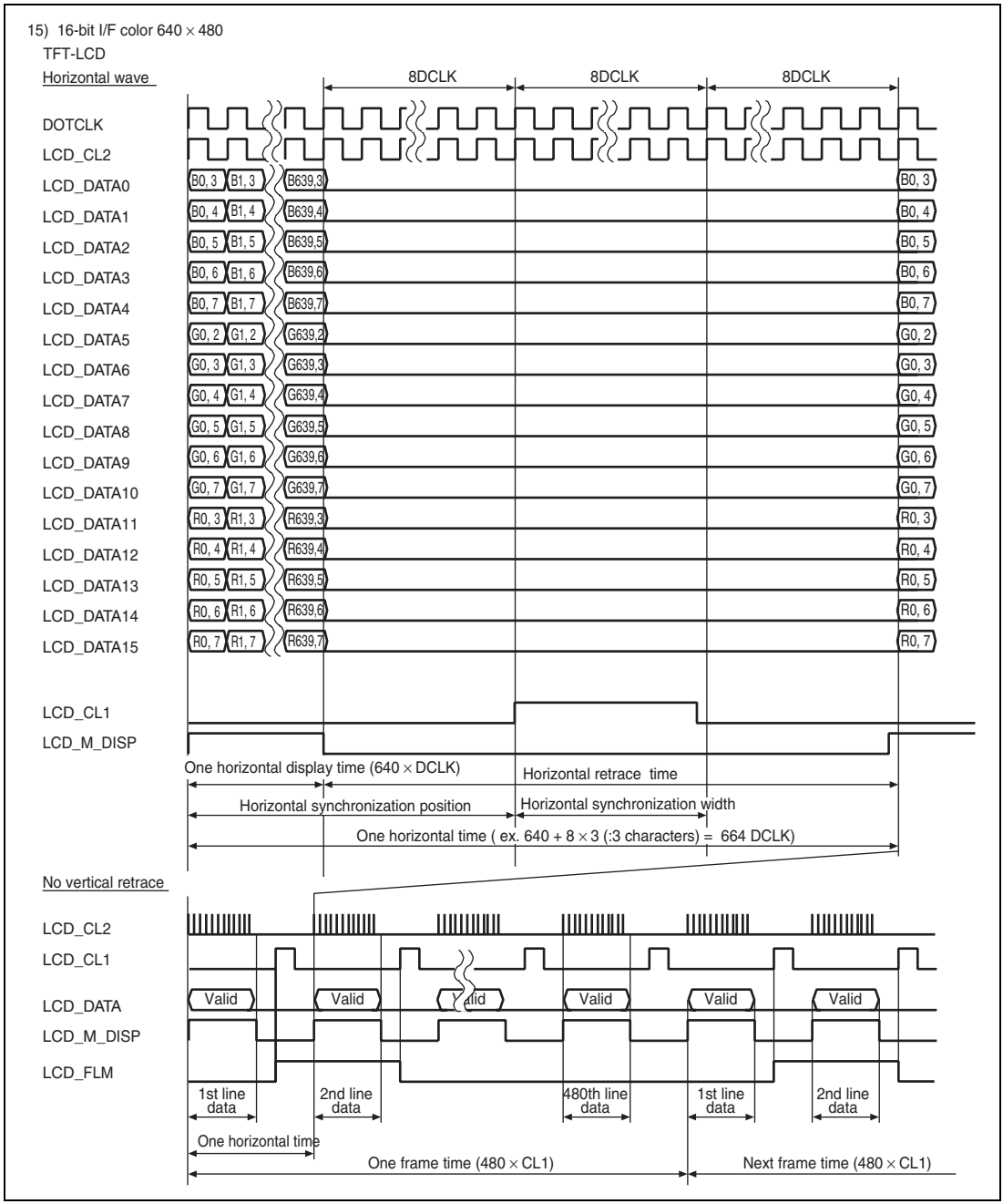


Figure 26.23 Clock and LCD Data Signal Example (16-Bit Interface Color 640 × 480)

26.6 Usage Notes

26.6.1 Procedure for Halting Access to Display Data Storage VRAM (Synchronous DRAM in Area 3)

Follow the procedure below to halt access to VRAM for storing display data (synchronous DRAM in area 3).

- Procedure for Halting Access to Display Data Storage VRAM
  A. Confirm that the LPS1 and LPS0 bits in LDPMMR are currently set to 1.
  B. Clear the DON bit in LDCNTR to 0 (display-off mode).
  C. Confirm that the LPS1 and LPS0 bits in LDPMMR have changed to 0.
  D. Wait for the display time for a single frame to elapse.

This halting procedure is required before selecting self-refreshing for the display data storage VRAM (synchronous DRAM in area 3) or making a transition to standby mode or module standby mode.

Section 27 A/D Converter

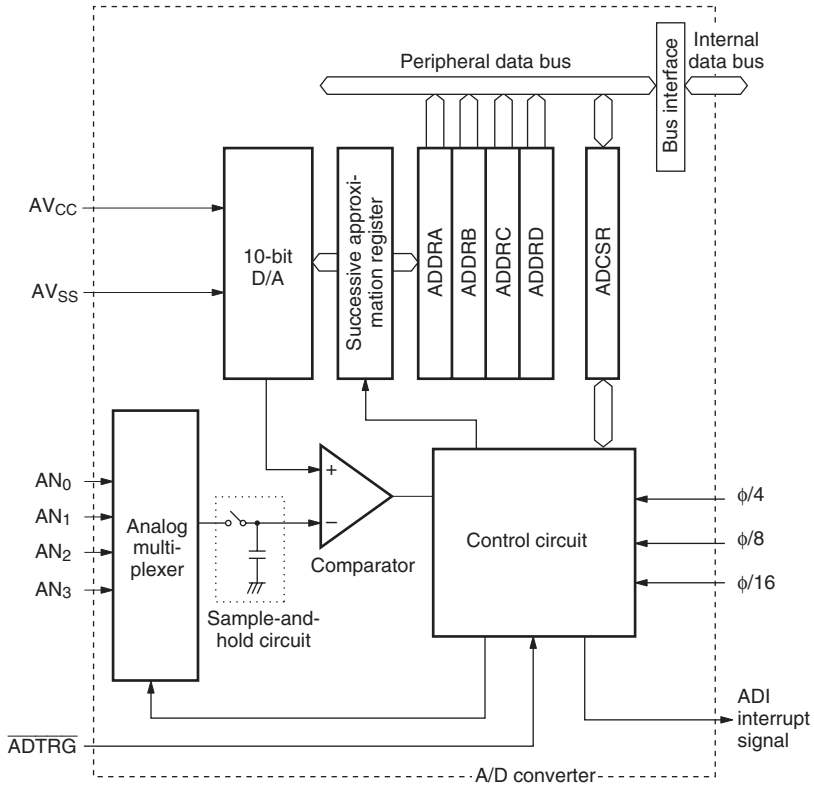
This LSI includes a 10-bit successive-approximation A/D converter allowing selection of up to four analog input channels.

27.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Four input channels
- High-speed conversion
  - Minimum conversion time: 15 μs per channel
- 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 27.1 shows a block diagram of the A/D converter.



[Legend]
ADCR: A/D control register
ADCSR: A/D control/status register
ADDRA: A/D data register A
ADDRB: A/D data register B
ADDRC: A/D data register C
ADDRD: A/D data register D

Figure 27.1 Block Diagram of A/D Converter

27.2 Input Pins

Table 27.1 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 27.1 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
ADC analog input pin 0	AN0	Input	Analog inputs
ADC analog input pin 1	AN1	Input	
ADC analog input pin 2	AN2	Input	
ADC analog input pin 3	AN3	Input	
ADC external trigger pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

27.3 Register Descriptions

The A/D converter has the following registers.

- A/D data register A (ADDRA)
- A/D data register B (ADDRB)
- A/D data register C (ADDRC)
- A/D data register D (ADDRC)
- A/D control/status register (ADCSR)

27.3.1 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 27.2 indicates the pairings of analog input channels and A/D data registers.

Each ADDR is initialized to H'0000 by a reset and the module standby function and in standby mode.

Table 27.2 Analog Input Channels and A/D Data Registers

Analog Input Channel	A/D Data Register
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD

27.3.2 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 the module standby function and in standby mode.

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	TRGE1	0	R	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 ($\overline{\text{ADTRG}}$)
10	TRGE0	0	R	
9	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	—	0	R	

Bit	Bit Name	Initial Value	R/W	Description
7	CKS1	0	R/W	Clock Select
6	CKS0	1	R/W	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	MULTI1	0	R/W	Selects single mode, multi mode, or scan mode.
4	MULTI0	0	R/W	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	CH2	0	R/W	Channel Select 2 to 0 (CH2 to CH0)
1	CH1	0	R/W	These bits and the MULTI bit select the analog input channels. Clear the ADST bit to 0 before changing the channel selection.
0	CH0	0	R/W	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.

27.4 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.

27.4.1 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 27.2 shows a timing diagram for this example.

1. Start the clock supply to the ADC module (clear the MSTP33 bit in STBCR3 to 0) to run the ADC module.
2. Single mode is selected (MULTI = 0), input channel AN1 is selected (CH2 = CH1 = 0, CH0 = 1), 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. When ADF = 1, ADIE = 1, and DMASL = 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. Stop the clock supply to the ADC module (set the MSTP33 bit in STBCR3 to 1) to place the ADC in the module standby state.

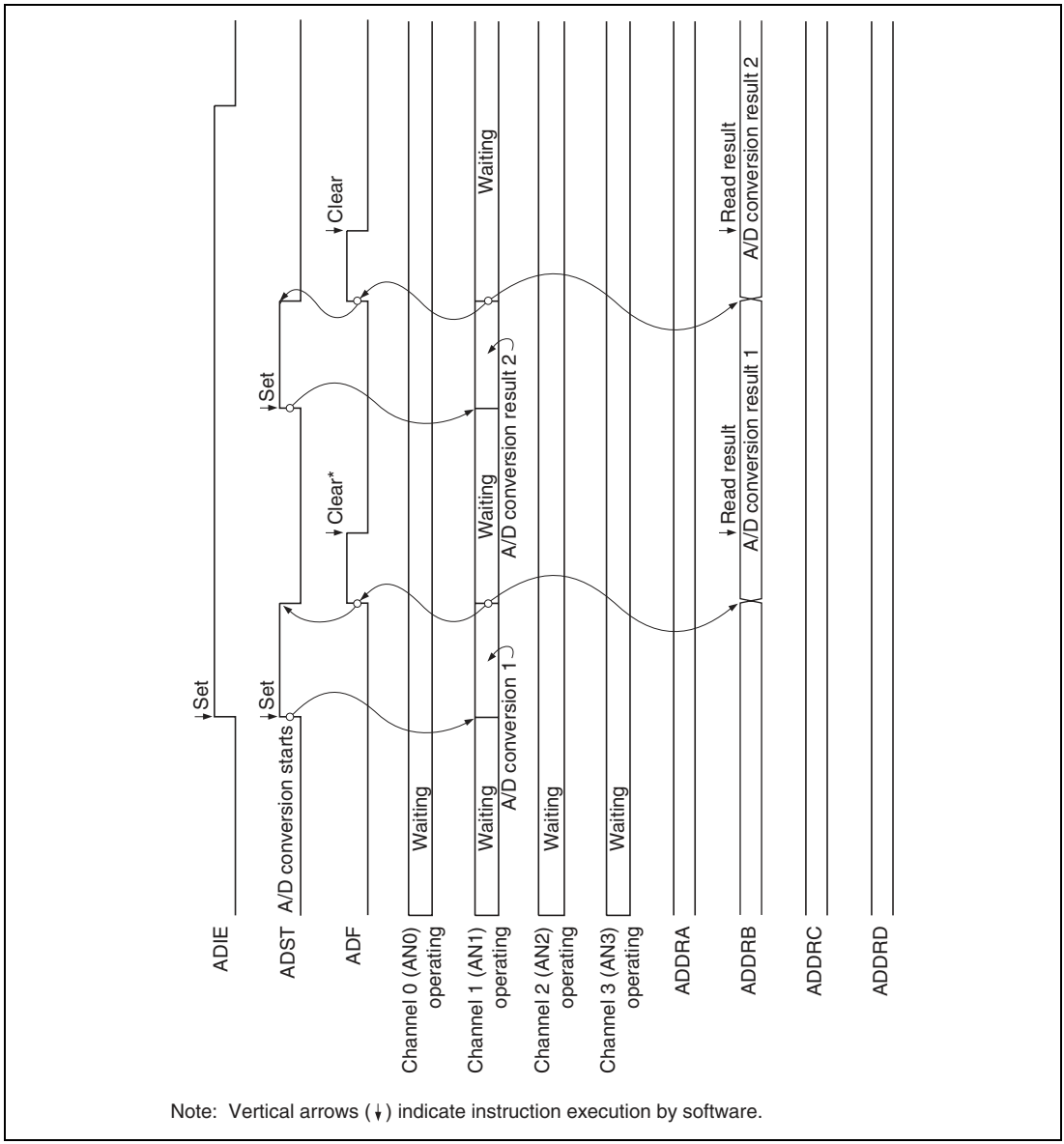


Figure 27.2 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

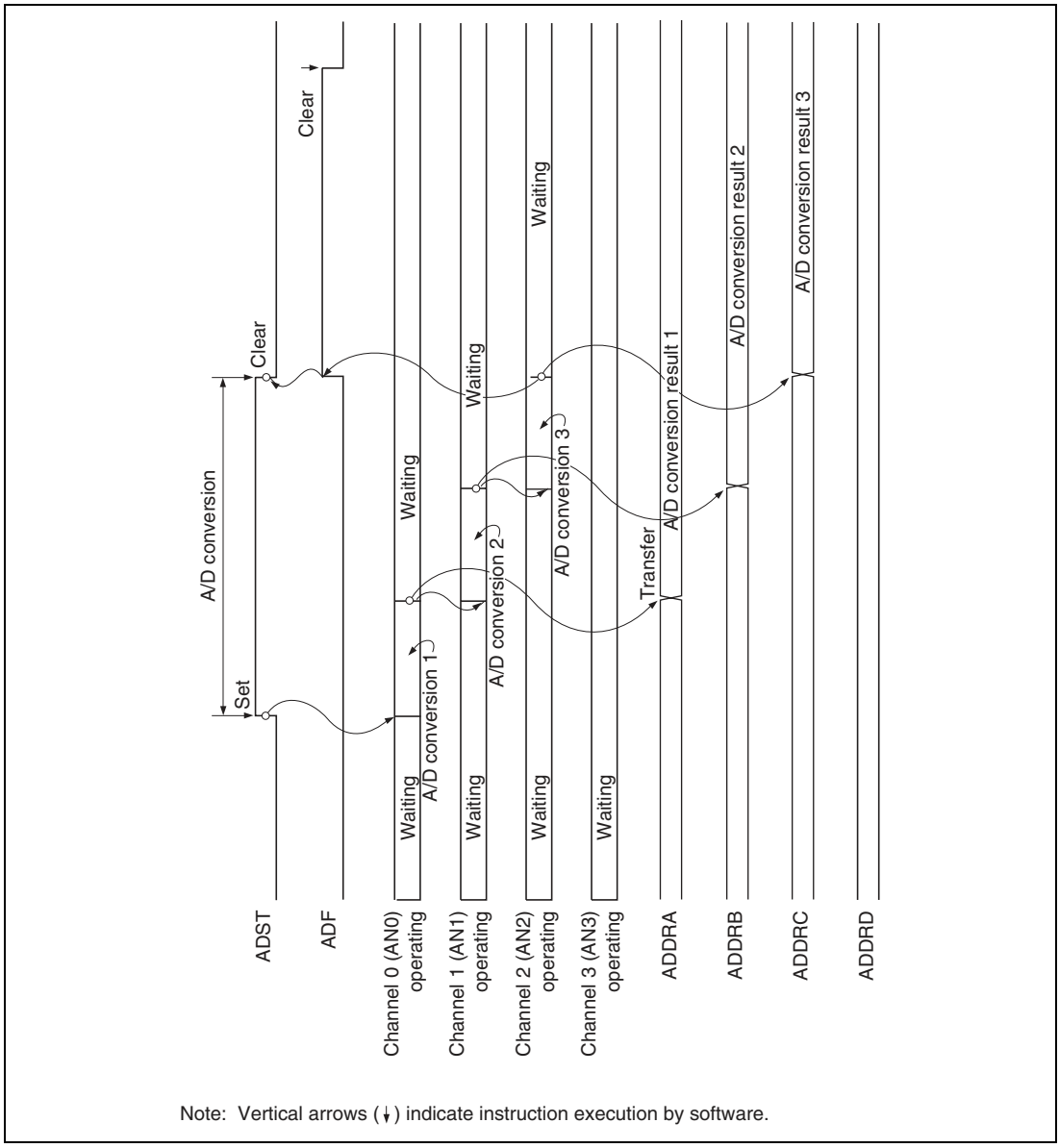
27.4.2 Multi Mode

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 27.3 shows a timing diagram for this example.

1. Start the clock supply to the ADC module (clear the MSTP33 bit in STBCR3 to 0) to run the ADC module.
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. Stop the clock supply to the ADC module (set the MSTP33 bit in STBCR3 to 1) to place the ADC in the module standby state.



Note: Vertical arrows (↓) indicate instruction execution by software.

Figure 27.3 Example of A/D Converter Operation (Multi Mode, Channels AN0 to AN2 Selected)

27.4.3 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 27.4 shows a timing diagram for this example.

1. Start the clock supply to the ADC module (clear the MSTP33 bit in STBCR3 to 0) to run the ADC module.
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.
8. Stop the clock supply to the ADC module (set the MSTP33 bit in STBCR3 to 1) to place the ADC in the module standby state.

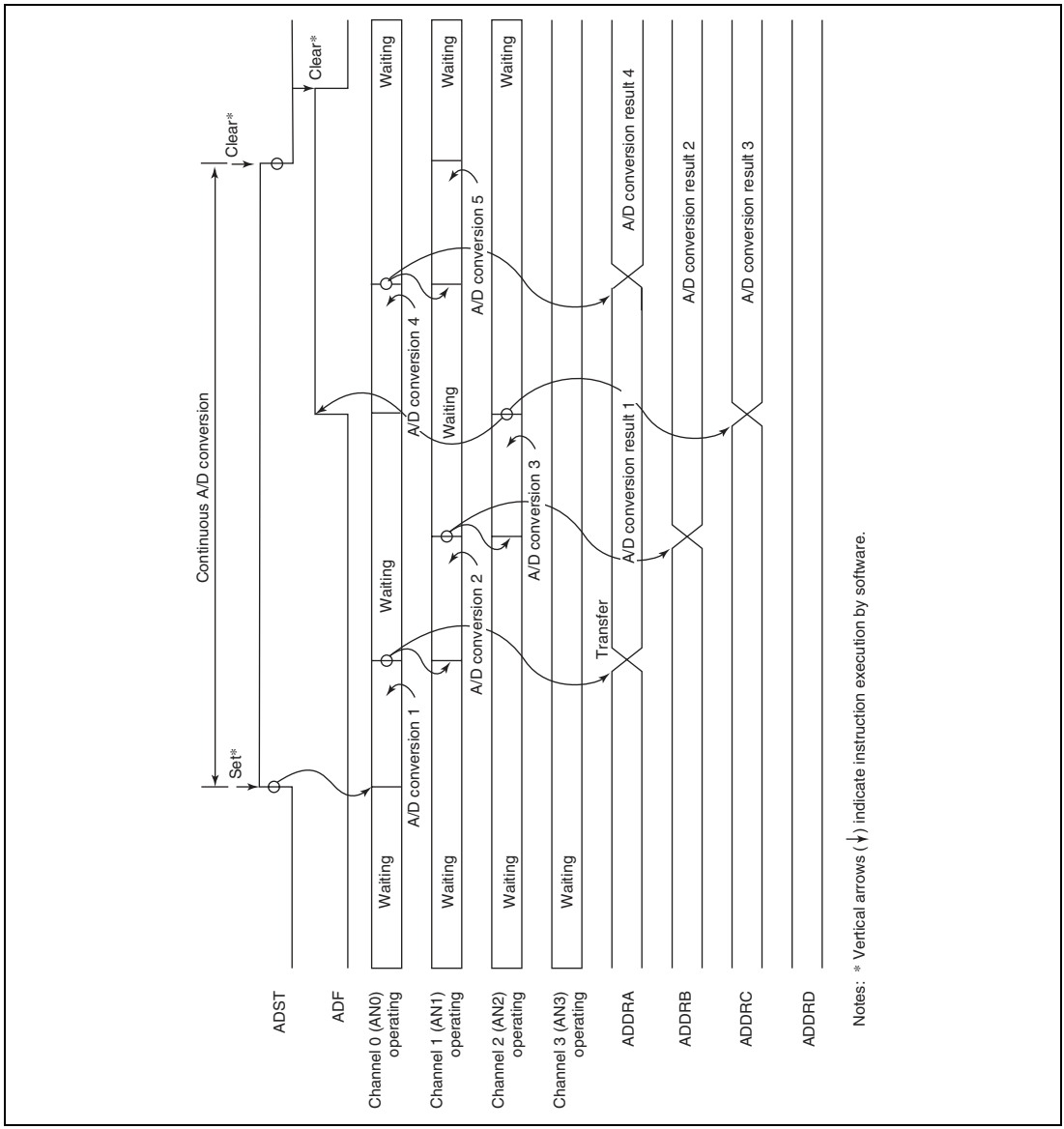


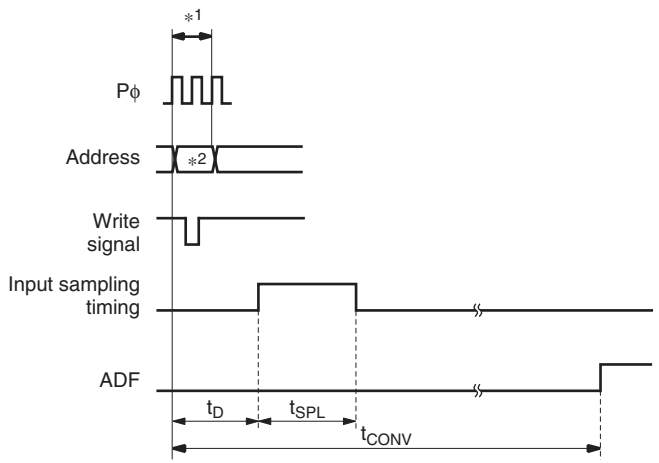
Figure 27.4 Example of A/D Converter Operation (Scan Mode, Channels AN0 to AN2 Selected)

27.4.4 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 27.5 shows the A/D conversion timing. Table 27.3 indicates the A/D conversion time.

As indicated in figure 27.5, 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 27.3.

In multi mode and scan mode, the values given in table 27.3 apply to the first conversion. In the second and subsequent conversions the conversion the conversion time is fixed at 512 states (fixed) when CKS1 = 1 and CKS0 = 0, 256 states (fixed) when CKS1 = 0 and CKS0 = 1, and 128 states (fixed) when CKS1 = 0 and CKS0 = 0.



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 27.5 A/D Conversion Timing

Table 27.3 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}) for Pϕ.

27.4.5 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 27.6 shows the timing.

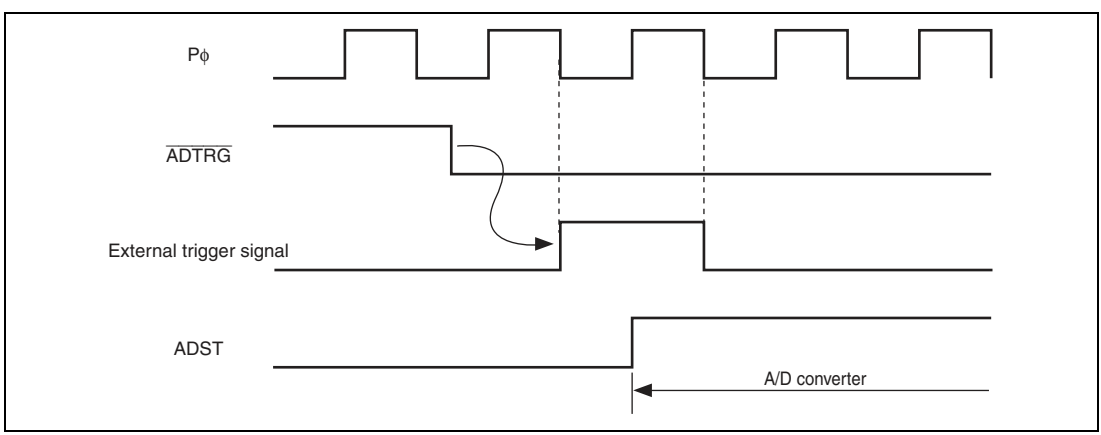


Figure 27.6 External Trigger Input Timing

27.5 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.

27.6 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 27.7. 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 27.7, 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 27.7, item (2)). Quantization error is the intrinsic error of the A/D converter and is expressed as 1/2 LSB (figure 27.7, item (3)). Nonlinearity error is the deviation between actual and ideal A/D conversion characteristics between zero voltage and full-scale voltage (figure 27.7, item (4)). Note that it does not include offset, full-scale, or quantization error.

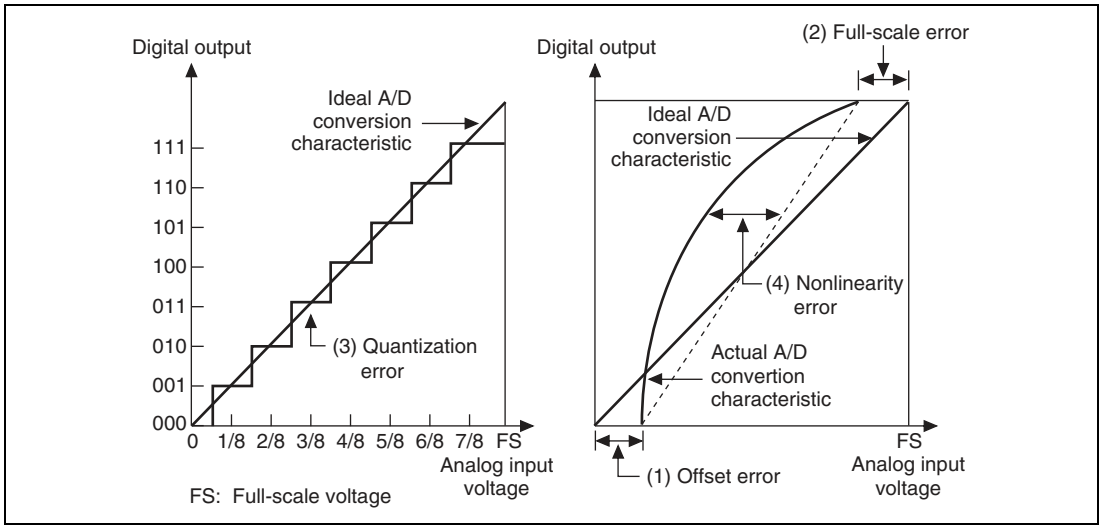


Figure 27.7 Definitions of A/D Conversion Accuracy

27.7 Usage Notes

27.7.1 Notes on A/D Conversion

(1) Notes on Clearing the ADF Bit in the ADCSR Register

Problem: Even though the ADCSR.ADF bit has been read as 1 and 0 was then written to the ADF bit, the ADF bit has not been cleared to 0.

Condition: This problem arises when reading of the ADF bit coincides with setting of the bit to 1 upon the end of A/D conversion.

Avoiding the Problem: Follow any of procedures (a), (b), or (c) below.

(a) Ensure that setting of the ADF bit to 1 upon the end of A/D conversion does not coincide with reading of the ADF bit. For example, read the ADF bit as 1 and then write 0 to the bit during processing of the A/D conversion end interrupt (ADI) that is generated at the end of A/D conversion (when the ADF is set to 1).

(b) If the ADF bit has not been cleared, repeat the operation of reading it as 1 and then writing 0 to it.

(c) Initialize the ADC and clear the ADF bit by placing the ADC in the module standby state.

(2) Notes on A/D Conversion in Scan Mode

Problem: A/D conversion in scan mode is not stopped by clearing the ADCSR.ADST bit (to 0).

Condition: This problem arises when 0 is written to the ADST bit in ADCSR to stop A/D conversion while A/D conversion in scan mode is in progress.

Avoiding the Problem: Place the ADC in module standby state after clearing the ADST bit (to 0). Placing the ADC in the module standby state initializes the ADC and stops A/D conversion. When further A/D conversion is required, restart A/D conversion after releasing the ADC from the module standby state.

(3) Notes on Transferring the Result of A/D Conversion by the DMAC

Problem: An incorrect superfluous DMA transfer is included before DMA transfer of the correct result of A/D conversion.

Condition: The problem arises in the following cases. Also see the table below:

(a) In Single/Multi Mode

The problem arises when A/D conversion is started while the setting of the ADCSR.DMASL bit is 1, after having proceeded while the setting of the DMASL bit was 0 and then stopped.

(b) In Scan Mode

The problem arises when A/D conversion is started while the setting of the ADCSR.DMASL bit is 1, after having proceeded and stopped.

Table 27.4 Conditions for the Method of Transferring Results of A/D Conversion and Inclusion of Superfluous DMA

The Next Conversion / Current Conversion		In Single/Multi Mode		In Scan Mode	
		DMASL = 0	DMASL = 1	DMASL = 0	DMASL = 1
Single Mode/ Multi Mode	DMASL = 0	Normal	Faulty	Normal	Faulty
	DMASL = 1	Normal	Normal	Normal	Normal
Scan Mode	DMASL = 0	Normal	Faulty	Normal	Faulty
	DMASL = 1	Normal	Faulty	Normal	Faulty

Avoiding the Problem: Follow either of procedures (a) or (b) below.

(a) After A/D conversion has stopped, initialize the ADC by placing it in the module standby state. Start the next round of A/D conversion after releasing the ADC from the module standby state.

(b) Operation under the following conditions ensures that the problem will not arise.

- **In Single/Multi Mode**

  Transfer when DMASL = 0 → Transfer when DMASL = 0
  Transfer when DMASL = 1 → Transfer when DMASL = 1

- **In Scan Mode**

  Transfer when DMASL = 0 → Transfer when DMASL = 0

27.7.2 Notes on A/D Conversion-End Interrupt and DMA Transfer

Generation of an interrupt or activation of the DMAC upon the end of A/D conversion is only allowed once per end of A/D conversion.

The conditions for the end of A/D conversion are the same as the setting conditions of the ADF bit of ADCSR.

According to the table below, A/D conversion value should be transferred by DMA transfer (in cycle steal mode), with the corresponding conversion mode and number of channels for conversion.

Conversion Mode	Number of Channels for Conversion	Data Size	Transfer Size for DMAC
Single mode	1	1 word	Word
Multi mode or Scan mode	1	1 word	Word
	2	2 words	Longword
	3	3 words	16 bytes
	4	4 words	16 bytes

27.7.3 Notes on the ADCSR.ADST

1. When the ADCSR.ADST bit is cleared to stop the A/D conversion in the scan mode or the multi mode, please restart the next A/D conversion after waiting of necessary time for the A/D conversion of one channel, or initialize the ADC and clear the ADF bit by placing the ADC in the module standby state.
   And restart A/D conversion after releasing the ADC from the module standby state.
2. The A/D conversion cannot be stopped by clearing the ADCSR.ADST bit at a single mode on the way.
   When the A/D conversion ends, the ADCSR.ADST bit is cleared automatically.
3. Except for above-mentioned 1. or 2., it is necessary to wait longer than one cycle of the clock selected in the ADCSR.CKS1-0 bit, to change the ADCSR.ADST bit, or initialize the ADC and clear the ADF bit by placing the ADC in the module standby state.
   And restart A/D conversion after releasing the ADC from the module standby state.

27.7.4 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 27.8). When converting high-speed analog signals or converting in scan mode, insert a low-impedance buffer.

27.7.5 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.

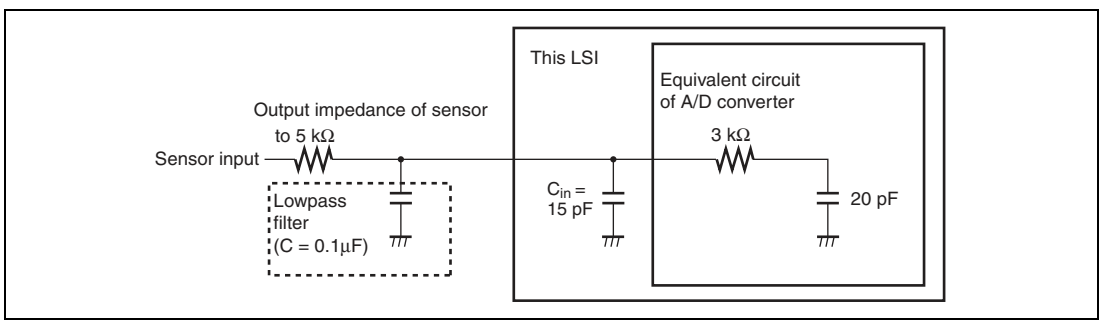


Figure 27.8 Analog Input Circuit Example

27.7.6 Setting Analog Input Voltage

Operating the chip in excess of the following voltage range may result in damage to chip reliability.

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).

27.7.7 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.

27.7.8 Notes on Countermeasures to Noise

Connect a protective circuit between AVcc and AVss, as shown in figure 27.9, 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.

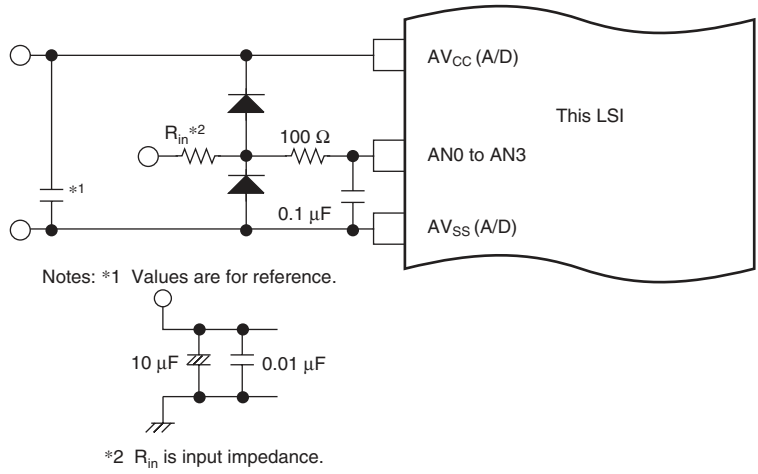


Figure 27.9 Example of Analog Input Protection Circuit

Table 27.5 Analog Input Pin Ratings

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Allowable signal-source impedance	—	5	kΩ

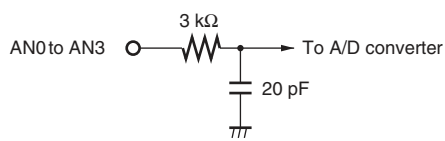


Figure 27.10 Analog Input Pin Equivalent Circuit

Section 28 D/A Converter (DAC)

This LSI incorporates a two-channel D/A converter (DAC) with the following features.

28.1 Features

- 8-bit resolution
- Two output channels
- Conversion time: Max. 10 μs (when load capacitance is 20 pF)
- Output voltage: 0 V to AVcc (analog power supply)

Figure 28.1 shows the block diagram for the DAC.

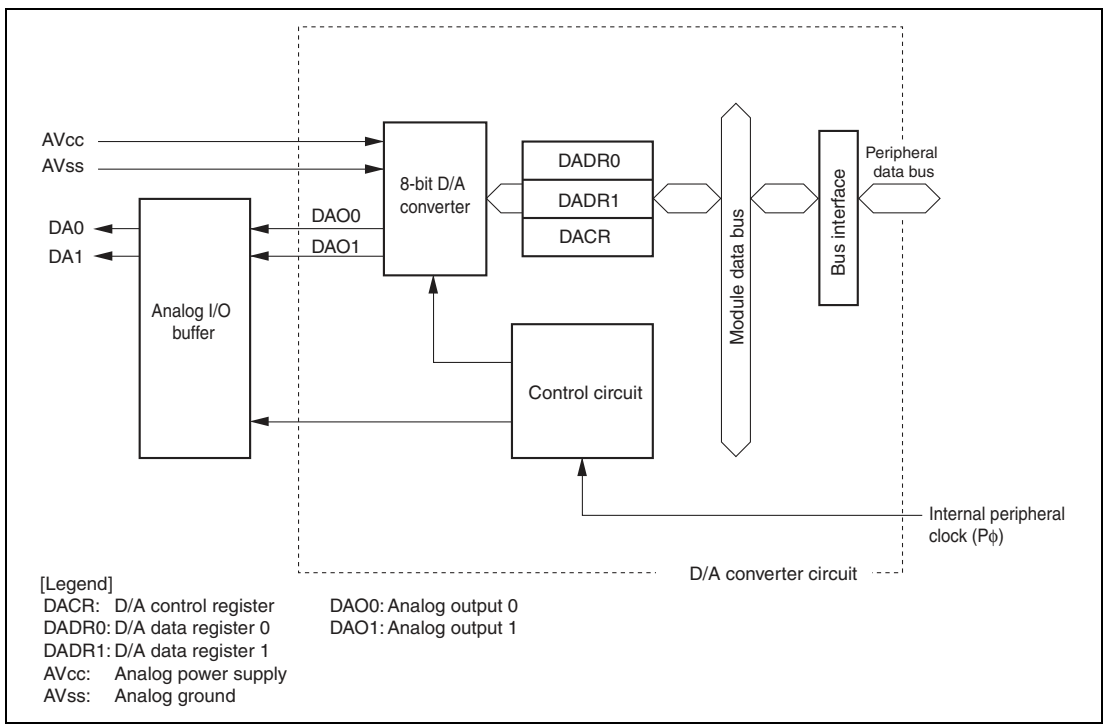


Figure 28.1 Block Diagram of D/A Converter

28.2 Input/Output Pins

Table 28.1 summarizes the input/output pins used by the D/A converter.

Table 28.1 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

28.3 Register Descriptions

The D/A converter has the following registers. Refer to section 37, List of Registers, for more details on the addresses and states of these registers in each operating mode.

- D/A data register 0 (DADR0)
- D/A data register 1 (DADR1)
- D/A control register (DACR)

28.3.1 D/A Data Registers 0 and 1 (DADR0, DADR1)

DADR0 and DADR1 are 8-bit readable/writable registers that store data for D/A conversion. When 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'00 at reset. Note that the D/A data register is not initialized upon entering the software standby, module standby, or hardware standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	H'00	R/W	8-bit registers that store data for D/A conversion.

28.3.2 D/A Control Register (DACR)

The DACR register is an 8-bit readable/writable register that controls D/A converter operation. The DACR is initialized to H'3F at reset. Note that the DACR is not initialized in software standby, module standby, or hardware standby mode.

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 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1. If 0 is written to these bits, correct operation cannot be guaranteed.

28.4 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). If the D/A data registers (DADR0 and DADR1) are modified, the D/A converter immediately initiates the new data conversion. When the DAOE1 and DAOE0 bits in the DACR register are set to 1, D/A conversion results are output.

An example of D/A conversion for channel 0 is shown below. The operation timing is shown in figure 28.2.

1. Write conversion data to DADR0.
2. When the DAOE0 bit in DACR is set to 1, D/A conversion starts. The results are output after the conversion has ended. The output value will be (DADR0 contents/255) × 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).

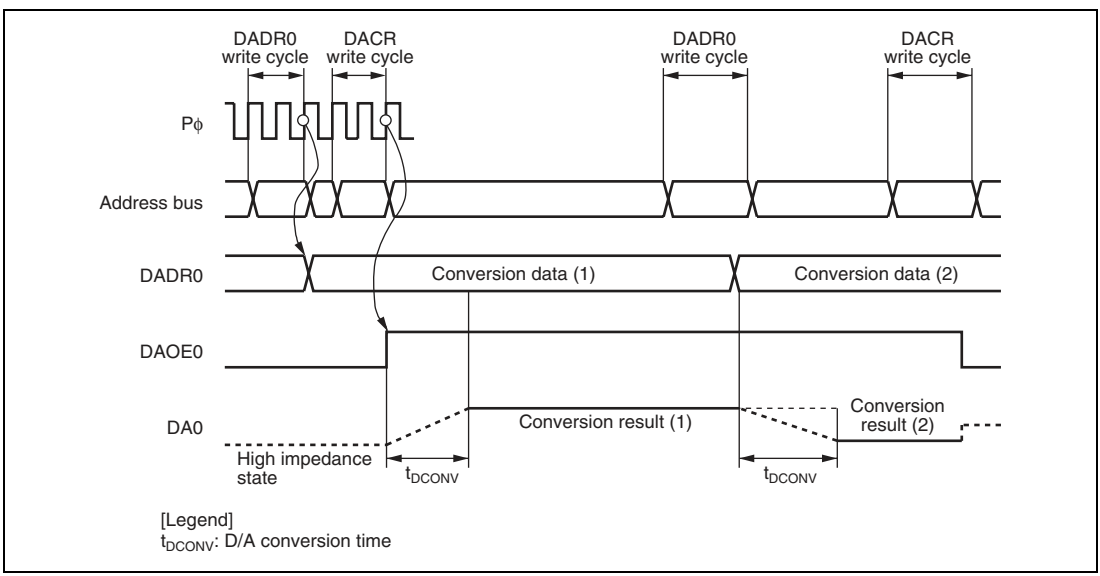


Figure 28.2 D/A Converter Operation Example

Section 29 PC Card Controller (PCC)

The PC card controller (PCC) controls the external buffer, interrupts, and exclusive ports of the PC card interface to be connected to this LSI. Using the PCC enables two slots of PC cards that conform to the PCMCIA Rev. 2.1/JEIDA Ver. 4.2 standard to be easily connected to this LSI.

29.1 Features

- As a PC card interface to be connected to physical area 6, an IC memory card interface and an I/O card interface are supported.
- Outputs control signals for the external buffer ($\overline{\text{PCC\_DRV}}$).
- Supports a preemptive operating system by switching attribute memory, common memory, and I/O space by using addresses.
- Provides a segment bit (an address bit for the PC card) for common memory, enabling access to a 64-Mbyte space fully conforming to PCMCIA specifications.
- Disables the PCC operation and supports only a bus interface of a PC card interface (by using the P0USE bit of PCC0GCR).

Figure 29.1 shows a block diagram of the PC card controller.

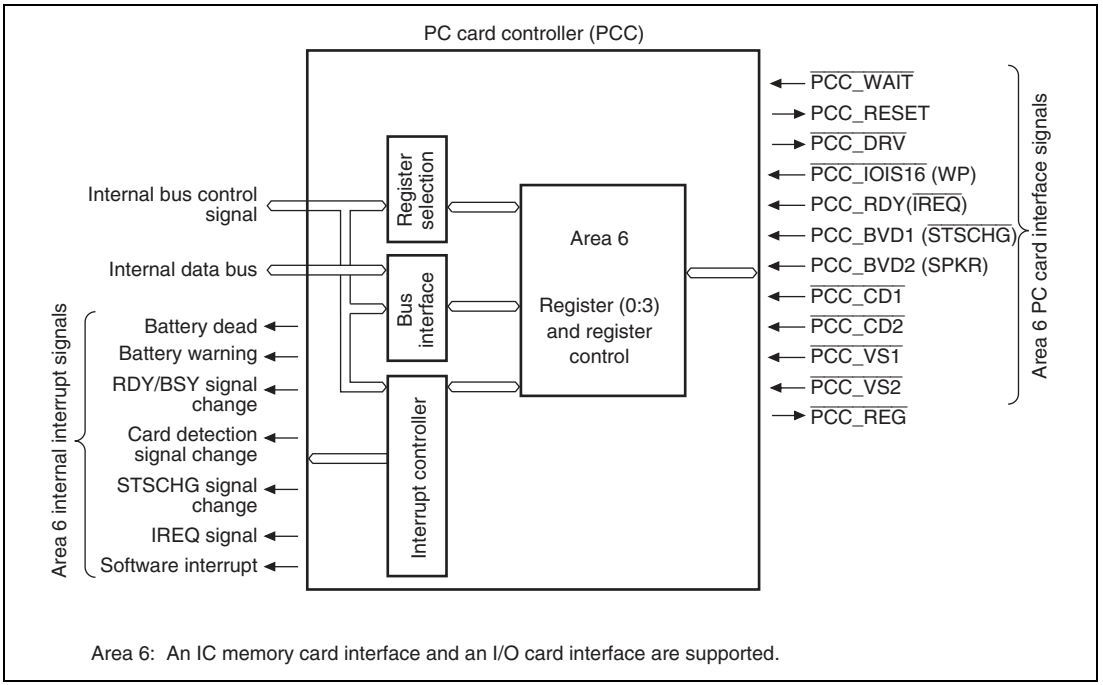


Figure 29.1 PC Card Controller Block Diagram

29.1.1 PCMCIA Support

This LSI supports an interface based on PCMCIA specifications for physical areas 6. Interfaces supported are the IC memory card interface and I/O card interface defined in the PCMCIA Rev. 2.1/JEIDA Ver. 4.2 standard. Both the IC memory card interface and I/O card interface are supported in area 6.

Table 29.1 Features of the PCMCIA Interface

Item	Feature
Access	Random access
Data bus	8/16 bits
Memory type	Masked ROM, OTPROM, EPROM, EEPROM, flash memory, SRAM
Common memory capacity	Maximum 64 Mbytes (Supports full PCMCIA specifications by using a segment bit (an address bit for the PC card))
Attribute memory capacity	Maximum 32 Mbytes
I/O space capacity	Maximum 32 Mbytes
Others	Dynamic bus sizing for I/O bus width* The PCMCIA interface can be accessed from the address-conversion region and non-address-conversion region.

Note: * Dynamic bus sizing for the I/O bus width is supported only in little-endian mode.

This LSI can directly access 32- and 64-Mbyte physical areas in a 64-Mbyte memory space and an I/O space of the PC card (continuous 32/16-Mbyte area mode). This LSI provides a segment bit (an address bit for the PC card) in the general control register for area 6 to support a common memory space with full PCMCIA specifications (64 Mbytes).

(1) Continuous 32-Mbyte Area Mode

Setting 0 (initial value) in bit 3 (P0MMOD) of the general control register enables the continuous 32-Mbyte area mode. In this mode, the attribute memory space and I/O memory space are 32 Mbytes and the common memory space is 64 Mbytes. In the common memory space, set 1 in bit 2 (P0PA25) of the general control register to access an address of more than 32 Mbytes. By this operation, 1 is output to A25 pin, enabling an address space of more than 32 Mbytes to be accessed. When an address of 32 Mbytes or less is accessed, set 0 in POPA25. This bit does not affect access to attribute memory space or I/O memory space.

Figure 29.2 shows the relationship between the memory space of this LSI and the memory and I/O spaces of the PC card in the continuous 32-Mbyte area mode. Memory space and I/O space are supported in area 6.

In area 6, set 1 in bit 0 (P0REG) of the general control register to access the common memory space of the PC card, and set 0 in bit 0 to access the attribute memory space (initial value: 0). By this operation, the set value is output to $\overline{\text{PCC\_REG}}$ pin, enabling any space to be accessed. When the I/O space is accessed in area 6, the output of $\overline{\text{PCC\_REG}}$ pin is always 0 regardless of the value of bit 0 (P0REG).

See the register descriptions in section 29.3, Register Descriptions for details of register settings.

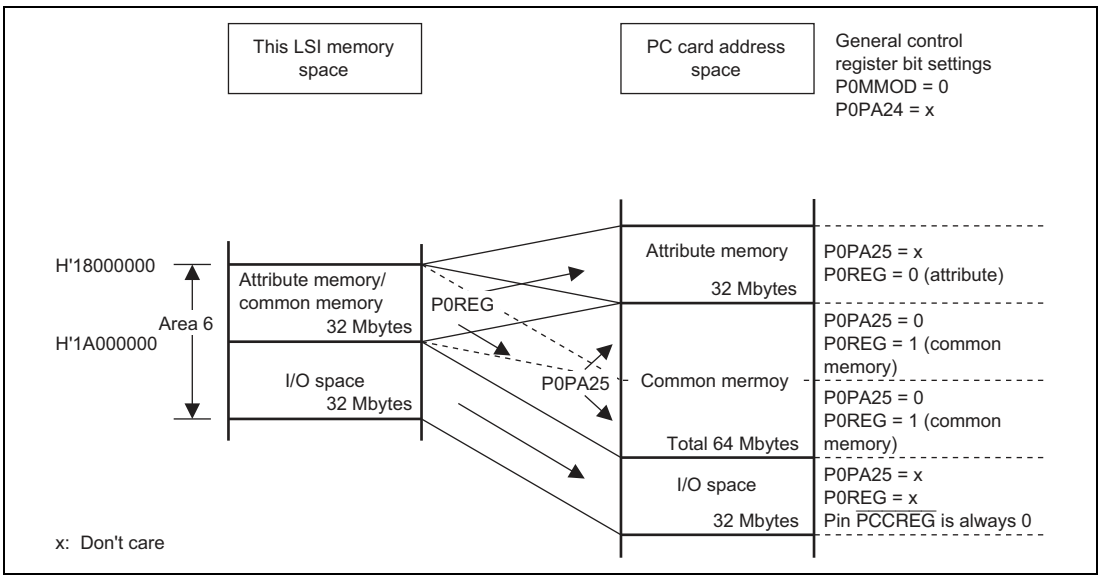


Figure 29.2 Continuous 32-Mbyte Area Mode

(2) Continuous 16-Mbyte Area Mode

Setting 1 in bit 3 (P0MMOD) of the general control register enables the continuous 16-Mbyte area mode. In this mode, the attribute memory space and I/O memory space are 16 Mbytes, and the common memory space is 64 Mbytes. In the common memory space, set the PC card address in bit 2 (P0PA25) and bit 1 (P0PA24) of the general control register to access each address of 16 Mbytes unit. By this operation, values are output to A25 and A24 pins, enabling an address space of more than 16 Mbytes to be accessed (initial value: 0 for P0PA25). When an address of 16 Mbytes or less is accessed, no settings are required. This bit does not affect access to attribute memory space or I/O memory space.

Figures 29.3 shows the relationship between the memory space of this LSI and the memory and I/O spaces of the PC card in the continuous 16-Mbyte area mode. Memory space and I/O space are supported in area 6.

The attribute memory space, common memory space, and I/O space of the PC card are provided as 16-Mbyte physical spaces in this mode. Therefore, this LSI automatically controls $\overline{\text{PCC\_REG}}$ pin (the value of bit 0 (P0REG) in the general control register is ignored). In area 6, the output of $\overline{\text{PCC\_REG}}$ pin is 0 when the attribute memory space or I/O space is accessed, and 1 when the common memory space is accessed.

See the register descriptions in section 29.3, Register Descriptions for details of register settings.

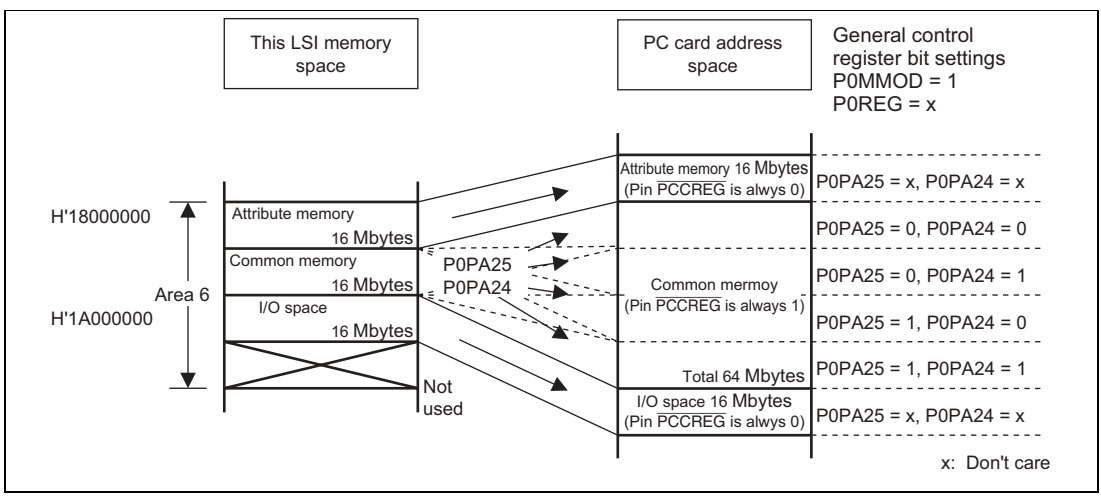


Figure 29.3 Continuous 16-Mbyte Area Mode (Area 6)

29.2 Input/Output Pins

PCC related external pins are listed below.

Table 29.2 PCC Pin Configuration

Pin Name	Abbreviation	I/O	Description
PCC wait request	$\overline{\text{PCC\_WAIT}}$	Input	Hardware wait request signal
PCC 16-bit input/output	$\overline{\text{PCC\_IOIS16}}$	Input	Write protection signal from PC card when IC memory interface is connected Signal to indicate 16-bit I/O from PC card when I/O card interface is connected
PCC ready	PCC_RDY	Input	Ready/busy signal form PC card when IC memory interface is connected Interrupt request signal from PC card when I/O card interface is connected
PCC battery detection 1	PCC_BVD1	Input	Buttery voltage detect 1 signal from PC card when IC memory interface is connected Card status change signal from PC card when I/O card interface is connected
PCC battery detection 2	PCC_BVD2	Input	Buttery voltage detect 2 signal from PC card when IC memory interface is connected Digital sound signal from PC card when I/O card interface is connected
PCC card detection 1	$\overline{\text{PCC\_CD1}}$	Input	Card detect 1 signal from PC card
PCC card detection 2	$\overline{\text{PCC\_CD2}}$	Input	Card detect 2 signal from PC card
PCC voltage detection 1	$\overline{\text{PCC\_VS1}}$	Input	Voltage sense 1 signal from PC card
PCC voltage detection 2	$\overline{\text{PCC\_VS2}}$	Input	Voltage sense 2 signal from PC card
PCC space indication	$\overline{\text{PCC\_REG}}$	Output	Area indicate signal for PC card
PCC buffer control	$\overline{\text{PCC\_DRV}}$	Output	Buffer control signal
PCC reset	PCC_RESET	Output	Reset signal for PC card

29.3 Register Descriptions

PCC has the following registers.

- Area 6 interface status register (PCC0ISR)
- Area 6 general control register (PCC0GCR)
- Area 6 card status change register (PCC0CSCR)
- Area 6 card status change interrupt enable register (PCC0CSCIER)

29.3.1 Area 6 Interface Status Register (PCC0ISR)

PCC0ISR is an 8-bit read-only register, which is used to read the status of the PC card connected to area 6. The initial value of PCC0ISR depends on the PC card status.

Bit	Bit Name	Initial Value	R/W	Description
7	P0RDY/ IREQ	Undefined*	R	PCC0 Ready The value on the RDY/$\overline{\text{BSY}}$ pin of the PC card connected to area 6 is read when the IC memory card interface is connected. The value of $\overline{\text{IREQ}}$ pin of the PC card connected to area 6 is read when the I/O card interface is connected. This bit cannot be written to. 0: Indicates that the value of RDY/$\overline{\text{BSY}}$ pin is 0 when the PC card connected to area 6 is an IC memory card interface type. The value of $\overline{\text{IREQ}}$ pin is 0 when the PC card connected to area 6 is the I/O card interface type. 1: Indicates that the value of RDY/$\overline{\text{BSY}}$ pin is 1 when the PC card connected to area 6 is the IC memory card interface type. The value of $\overline{\text{IREQ}}$ pin is 1 when the PC card connected to area 6 is the I/O card interface type.

Bit	Bit Name	Initial Value	R/W	Description
6	P0MWP	Undefined*	R	PCC0 Write Protect The value of WP pin of the PC card connected to area 6 is read when the IC memory card interface is connected. 0 is read when the I/O card interface is connected. This bit cannot be written to. 0: Indicates that the value of WP pin is 0 when the PC card connected to area 6 uses the IC memory card interface type. The value of bit 6 is always 0 when the PC card connected to area 6 is the I/O card interface type. 1: Indicates that the value of WP pin is 1 when the PC card connected to area 6 is the IC memory card interface type.
5	P0VS2	Undefined*	R	PCC0 Voltage Sense 2 The value of $\overline{VS2}$ pin of the PC card connected to area 6 is read. This bit cannot be written to. 0: The value of $\overline{VS2}$ pin of the PC card connected to area 6 is 0 1: The value of $\overline{VS2}$ pin of the PC card connected to area 6 is 1
4	P0VS1	Undefined*	R	PCC0 Voltage Sense 1 The value of $\overline{VS1}$ pin of the PC card connected to area 6 is read. This bit cannot be written to. 0: The value of $\overline{VS1}$ pin of the PC card connected to area 6 is 0 1: The value of $\overline{VS1}$ pin of the PC card connected to area 6 is 1
3	P0CD2	Undefined*	R	PCC0 Card Detect 2 The value of $\overline{CD2}$ pin of the PC card connected to area 6 is read. This bit cannot be written to. 0: The value of $\overline{CD2}$ pin of the PC card connected to area 6 is 0 1: The value of $\overline{CD2}$ pin of the PC card connected to area 6 is 1

Bit	Bit Name	Initial Value	R/W	Description
2	P0CD1	Undefined*	R	PCC0 Card Detect 1 The value of $\overline{CD1}$ pin of the PC card connected to area 6 is read. This bit cannot be written to. 0: The value of $\overline{CD1}$ pin of the PC card connected to area 6 is 0 1: The value of $\overline{CD1}$ pin of the PC card connected to area 6 is 1
1	P0BVD2/ P0SPKR	Undefined*	R	PCC0 Battery Voltage Detect 2 and 1 The values of BVD1 and BVD2 pin of the PC card connected to area 6 are read when the IC memory card interface is connected. The values of $\overline{STSCHG}$ and $\overline{SPKR}$ pin of the PC card connected to area 6 are read when the I/O card interface is connected. These bits cannot be written to. (1) The following applies to the IC memory interface. 11: The battery voltage of the PC card connected to area 6 is normal (Battery Good) 01: The battery must be changed although data is guaranteed for the PC card connected to area 6 (Battery Warning) x0: The battery voltage is abnormal and data is not guaranteed for the PC card connected to area 6 (Battery Dead) (2) The values of bits 1 and 0 for the I/O card interface are as follows: 0: The value of $\overline{STSCHG}$ or $\overline{SPKR}$ of the PC card connected to area 6 is 0 1: The value of $\overline{STSCHG}$ or $\overline{SPKR}$ of the PC card connected to area 6 is 1
0	P0BVD1/ P0STSCHG	Undefined*	R	

Note: * Differs depending on the state of the PC card.

29.3.2 Area 6 General Control Register (PCC0GCR)

PCC0GCR is an 8-bit readable/writable register, which controls the external buffer, resets, address A25 and A24 pins, and $\overline{\text{REG}}$ pin, and sets the PC card type for the PC card connected to area 6. PCC0GCR is initialized by a power-on reset but retains its value in a manual reset and in software standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	P0DRVE	0	R/W	PCC0 Buffer Control Controls the external buffer for the PC card connected to area 6. 0: High-level setting for control $\overline{\text{PCC\_DRV}}$ pin of the external buffer for the PC card connected to area 6 1: Low-level setting for control $\overline{\text{PCC\_DRV}}$ pin of the external buffer for the PC card connected to area 6
6	P0PCCR	0	R/W	PCC0 Card Reset Controls resets for the PC card connected to area 6. 0: Low-level setting for reset PCC_RESET pin for the PC card connected to area 6 1: High-level setting for reset PCC_RESET pin for the PC card connected to area 6
5	P0PCCT	0	R/W	PCC0 Card Type Specifies the type of the PC card connected to area 6. Cleared to 0 when the PC card is the IC memory card interface type; set to 1 when the PC card is the I/O card interface type. 0: The PC card connected to area 6 is handled as the IC memory card interface type 1: The PC card connected to area 6 is handled as the I/O card interface type

Bit	Bit Name	Initial Value	R/W	Description
4	P0USE	0	R/W	PCC0 Use/Not Use Specifies that the PC Card Controller to be worked or not worked. 0: PC Card Controller doesn't work 1: PC Card Controller works Note: When setting P0USE to 1, following settings are required. When P0USE is set to 1 and P0PCCT is set to 0, bits 21 and 20 (SA1 and SA0) in the CS6BWCR register of BSC should be set to 0. When P0USE and P0PCCT are set to 1, bits 21 and 20 (SA1 and SA0) in the CS6BWCR register of BSC should be set to 1. Before P0USE is set to 1, bits 15 to 12 (TYPE3 to TYPE0) in CS6BBCR of BSC should be set to 0101.
3	P0MMOD	0	R/W	PCC0 Mode Controls $\overline{\text{PCC\_REG}}$ and A24 pins for the PC card connected to area 6. Specifies either A24 of the address to be accessed or bit P0REG for outputting to $\overline{\text{PCC\_REG}}$ pin. When the common memory space is accessed, specifies either A24 of the address to be accessed or bit P0PA24 for outputting to A24 pin. By this operation, continuous 32 or 16 Mbytes can be selected for the address area of the common memory space of the PC card. 0: Bit P0REG is output to $\overline{\text{PCC\_REG}}$ pin, and A24 of address to be accessed is output to A24 pin (continuous 32-Mbyte area mode) 1: A24 of address to be accessed is output to $\overline{\text{PCC\_REG}}$ pin. When the common memory space is accessed, P0PA24 is output to A24 pin (continuous 16-Mbyte area mode)

Bit	Bit Name	Initial Value	R/W	Description
2	P0PA25	0	R/W	PC Card Address Controls A25 pin for the PC card connected to area 6. When the common memory space is accessed for the PC card connected to area 6, this bit is output to A25 pin. When the attribute memory space or I/O space is accessed, this bit is meaningless. 0: When the common memory space is accessed for the PC card connected to area 6, 0 is output to A25 pin 1: When the common memory space is accessed for the PC card connected to area 6, 1 is output to A25 pin
1	P0PA24	0	R/W	PC Card Address Controls A24 pin for the PC card connected to area 6. When bit P0MMOD is 1 and the common memory space is accessed for the PC card connected to area 6, this bit is output to A24 pin. When bit P0MMOD is 0 or the attribute memory space or I/O space is accessed, this bit is meaningless. 0: When bit P0MMOD is 1 and the common memory space is accessed for the PC card connected to area 6, 0 is output to A24 pin 1: When bit P0MMOD is 1 and the common memory space is accessed for the PC card connected to area 6, 1 is output to A24 pin
0	P0REG	0	R/W	PCC0REG Space Indication Controls $\overline{\text{PCC\_REG}}$ pin for the PC card connected to area 6. When bit P0MMOD is 0, this bit is output to $\overline{\text{PCC\_REG}}$ pin for the PC card connected to area 6. When bit P0MMOD is 1 or the I/O card interface is accessed, this bit is meaningless. 0: When bit P0MMOD is 0 and the PC card connected to area 6 is accessed, 0 is output to $\overline{\text{PCC\_REG}}$ pin 1: When bit P0MMOD is 0 and the PC card connected to area 6 is accessed, 1 is output to $\overline{\text{PCC\_REG}}$ pin

29.3.3 Area 6 Card Status Change Register (PCC0CSCR)

PCC0CSCR is an 8-bit readable/writable able register. PCC0CSCR bits are set to 1 by interrupt sources of the PC card connected to area 6 (only bit 7 can be set to 1 as required). PCC0CSCR is initialized by a power-on reset but retains its value in a manual reset and in software standby mode.

A common interrupt vector (PCCI) is assigned to each interrupt source.

Bit	Bit Name	Initial Value	R/W	Description
7	P0SCDI	0	R/W	PCC0 Software Card Detect Change Interrupt A PCC0 software card detect change interrupt can be generated by writing 1 to this bit. When this bit is set to 1, the same interrupt as the PCC0 card detect change interrupt (bit 3 set status) occurs if bit 3 (PCC0 card detect change enable) in the area 6 card status change interrupt enable register (PCC0CSCIER) is set to 1. If bit 3 is cleared to 0, no interrupt occurs. 0: No software card detect change interrupt occurs for the PC card connected to area 6 1: Software card detect change interrupt occurs for the PC card connected to area 6
6	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.
5	P0IREQ	0	R/W	PCC0IREQ Request Indicates the interrupt request for the $\overline{\text{IREQ}}$ pin of the PC card when the PC card connected to area 6 is the I/O card interface type. The P0IREQ bit is set to 1 when an interrupt request signal in pulse mode or level mode is input to the $\overline{\text{IREQ}}$. The mode is selected by bits 5 and 6 (PCC0IREQ interrupt enable bits) in the area 6 card status change interrupt enable register (PCC0CSCIER). This bit can be cleared to 0 only in pulse mode. Write 0 to bit 5 to clear the bit to 0. This bit is not changed if 1 is written. In level mode, bit 5 is a read-only bit, which reflects the $\overline{\text{IREQ}}$ state (if the $\overline{\text{IREQ}}$ is low, 1 is read). This bit always reads 0 on the IC memory card interface. 0: No interrupt request on the $\overline{\text{IREQ}}$ of the PC card when the PC card is on the I/O card interface 1: An interrupt request on the $\overline{\text{IREQ}}$ of the PC card has occurred when the PC card is on the I/O card interface

Bit	Bit Name	Initial Value	R/W	Description
4	P0SC	0	R/W	PCC0 Status Change Indicates a change in the value of the $\overline{\text{STSCHG}}$ of the PC card when the PC card connected to area 6 is the I/O card interface type. When the $\overline{\text{STSCHG}}$ is changed from 1 to 0, the P0SC bit is set to 1. When $\overline{\text{STSCHG}}$ is not changed, the P0SC bit remains at 0. Write 0 to bit 4 when this bit is set to 1 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the IC memory card interface. 0: $\overline{\text{STSCHG}}$ of the PC card is not changed when the PC card is on the I/O card interface 1: $\overline{\text{STSCHG}}$ of the PC card is changed from 1 to 0 when the PC card is on the I/O card interface
3	P0CDC	0	R/W	PCC0 Card Detect Change Indicates a change in the value of the $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ in the PC card connected to area 6. When the $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ values are changed, the P0CDC bit is set to 1. When the values are not changed, the P0CDC bit remains at 0. Write 0 to bit 3 in order to clear this bit to 0. This bit is not changed if 1 is written. 0: $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ in the PC card are not changed 1: $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ in the PC card are changed
2	P0RC	0	R/W	PCC0 Ready Change Indicates a change in the value of the RDY/$\overline{\text{BSY}}$ of the PC card when the PC card connected to area 6 is the IC memory card interface type. When the RDY/$\overline{\text{BSY}}$ is changed from 0 to 1, the P0RC bit is set to 1. When the RDY/ $\overline{\text{BSY}}$ is not changed, the P0RC bit remains at 0. Write 0 to bit 2 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the I/O card interface. 0: RDY/$\overline{\text{BSY}}$ in the PC card is not changed when the PC card is on the IC memory card interface 1: RDY/$\overline{\text{BSY}}$ in the PC card is changed from 0 to 1 when the PC card is on the IC memory card interface

Bit	Bit Name	Initial Value	R/W	Description
1	P0BW	0	R/W	PCC0 Battery Warning Indicates whether the BVD2 and BVD1 of the PC card are in the state in which "the battery must be changed although the data is guaranteed" when the PC card connected to area 6 is on the IC memory card interface. When the BVD2 and BVD1 are 0 and 1, respectively, the P0BW bit is set to 1; in other cases, the P0BW bit remains at 0. This bit is updated when the BVD2 and BVD1 are changed. Write 0 to bit 1 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the I/O card interface. 0: BVD2 and BVD1 of the PC card are not in the battery warning state when the PC card is in the IC memory card interface 1: BVD2 and BVD1 of the PC card are in the battery warning state and "the battery must be changed although the data is guaranteed" when the PC card is on the IC memory card interface
0	P0BD	0	R/W	PCC0 Battery Dead Indicates whether the BVD2 and BVD1 of the PC card are in the state in which "the battery must be changed since the data is not guaranteed" when the PC card connected to area 6 is on the IC memory card interface. When the BVD2 and BVD1 are 1 and 0 or 0 and 0, the P0BD bit is set to 1; in other cases, the P0BD bit remains at 0. This bit is updated when the BVD2 and BVD1 are changed. Write 0 to bit 0 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the I/O card interface. 0: BVD2 and BVD1 of the PC card are not in the state in which "the battery must be changed since the data is not guaranteed" when the PC card is on the IC memory card interface 1: BVD2 and BVD1 of the PC card are in the state in which "the battery must be changed since the data is not guaranteed" when the PC card is on the IC memory card interface

29.3.4 Area 6 Card Status Change Interrupt Enable Register (PCC0CSCIER)

The area 6 card status change interrupt enable register (PCC0CSCIER) is an 8-bit readable/writable register. PCC0CSCIER enables or disables interrupt requests for interrupt sources for the PC card connected to area 6. When a PCC0CSCIER is set to 1, the corresponding interrupt is enabled, and when the bit is cleared to 0, the interrupt is disabled. PCC0CSCIER is initialized by a power-on reset but retains its value in a manual reset and in software standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	P0CRE	0	R/W	PCC0 Card Reset Enable When this bit is set to 1, and when the $\overline{CD1}$ and $\overline{CD2}$ detect that a PC card is connected to area 6, the area 6 general control register (PCC0GCR) is initialized. 0: The area 6 general control register (PCC0GCR) is not initialized even if a PC card is detected in area 6 1: The area 6 general control register (PCC0GCR) is initialized when a PC card is detected connected to area 6

Bit	Bit Name	Initial Value	R/W	Description
6	IREQE1	0	R/W	PCC0IREQ Request Enable
5	IREQE0	0	R/W	These bits enable or disable $\overline{\text{IREQ}}$ interrupt requests and select the interrupt mode when the PC card connected to area 6 is the I/O card interface type. Note that bit 5 (P0IREQ) in the area 6 card status change register (PCC0CSCR) is cleared if the values in bits 6 and 5 in this register are changed. These bits have no meaning on the IC memory card interface. 00: IREQ requests are not accepted for the PC card connected to area 6. Bit 5 in the status change register (PCC0CSCR) functions as a read-only bit that indicates the inverse of the $\overline{\text{IREQ}}$ signal. 01: The level-mode IREQ interrupt request signal is accepted for the PC card connected to area 6. In level mode, an interrupt occurs when level 0 of the signal input from the $\overline{\text{IREQ}}$ is detected. 10: The pulse-mode IREQ interrupt request signal is accepted for the PC card connected to area 6. In pulse mode, an interrupt occurs when a falling edge from 1 to 0 of the signal input from the $\overline{\text{IREQ}}$ is detected. 11: The pulse-mode IREQ interrupt request signal is accepted for the PC card connected to area 6. In pulse mode, an interrupt occurs when a rising edge from 0 to 1 of the signal input from the $\overline{\text{IREQ}}$ is detected.
4	P0SCE	0	R/W	PCC0 Status Change Enable When the PC card connected to area 6 is on the I/O card interface, bit 4 enables or disables the interrupt request when the value of the BVD1 ($\overline{\text{STSCHG}}$) is changed. This bit has no meaning in the IC memory card interface. 0: No interrupt occurs for the PC card connected to area 6 regardless of the value of the BVD1 ($\overline{\text{STSCHG}}$) 1: An interrupt occurs for the PC card connected to area 6 when the value of the BVD1 ($\overline{\text{STSCHG}}$) is changed from 1 to 0

Bit	Bit Name	Initial Value	R/W	Description
3	P0CDE	0	R/W	PCC0 Card Detect Change Enable Bit 3 enables or disables the interrupt request when the values of the $\overline{CD1}$ and $\overline{CD2}$ are changed. 0: No interrupt occurs for the PC card connected to area 6 regardless of the values of the $\overline{CD1}$ and $\overline{CD2}$ 1: An interrupt occurs for the PC card connected to area 6 when the values of the $\overline{CD1}$ and $\overline{CD2}$ are changed
2	P0RE	0	R/W	PCC0 Ready Change Enable When the PC card connected to area 6 is on the IC memory card interface, bit 2 enables or disables the interrupt request when the value of the RDY/$\overline{BSY}$ is changed. This bit has no meaning on the I/O card interface. 0: No interrupt occurs for the PC card connected to area 6 regardless of the value of the RDY/$\overline{BSY}$ 1: An interrupt occurs for the PC card connected to area 6 when the value of the RDY/$\overline{BSY}$ is changed from 0 to 1
1	P0BWE	0	R/W	PCC0 Battery Warning Enable When the PC card connected to area 6 is on the IC memory card interface, bit 1 enables or disables the interrupt request when the BVD2 or BVD1 are in the state in which "the battery must be changed although the data is guaranteed". This bit has no meaning on the I/O card interface. 0: No interrupt occurs when the BVD2 or BVD1 are in the state in which "the battery must be changed although the data is guaranteed" 1: An interrupt occurs when the BVD2 or BVD1 are in the state in which "the battery must be changed although the data is guaranteed"

Bit	Bit Name	Initial Value	R/W	Description
0	P0BDE	0	R/W	PCC0 Battery Dead Enable When the PC card connected to area 6 is on the IC memory card interface, bit 0 enables or disables the interrupt request when the BVD2 and BVD1 are in the state in which "the battery must be changed since the data is not guaranteed". This bit has no meaning on the I/O card interface. 0: No interrupt occurs when the BVD2 and BVD1 are in the state in which "the battery must be changed since the data is not guaranteed" 1: An interrupt occurs when the BVD2 and BVD1 are in the state in which "the battery must be changed since the data is not guaranteed"

29.4 Operation

29.4.1 PC card Connection Specification (Interface Diagram, Pin Correspondence)

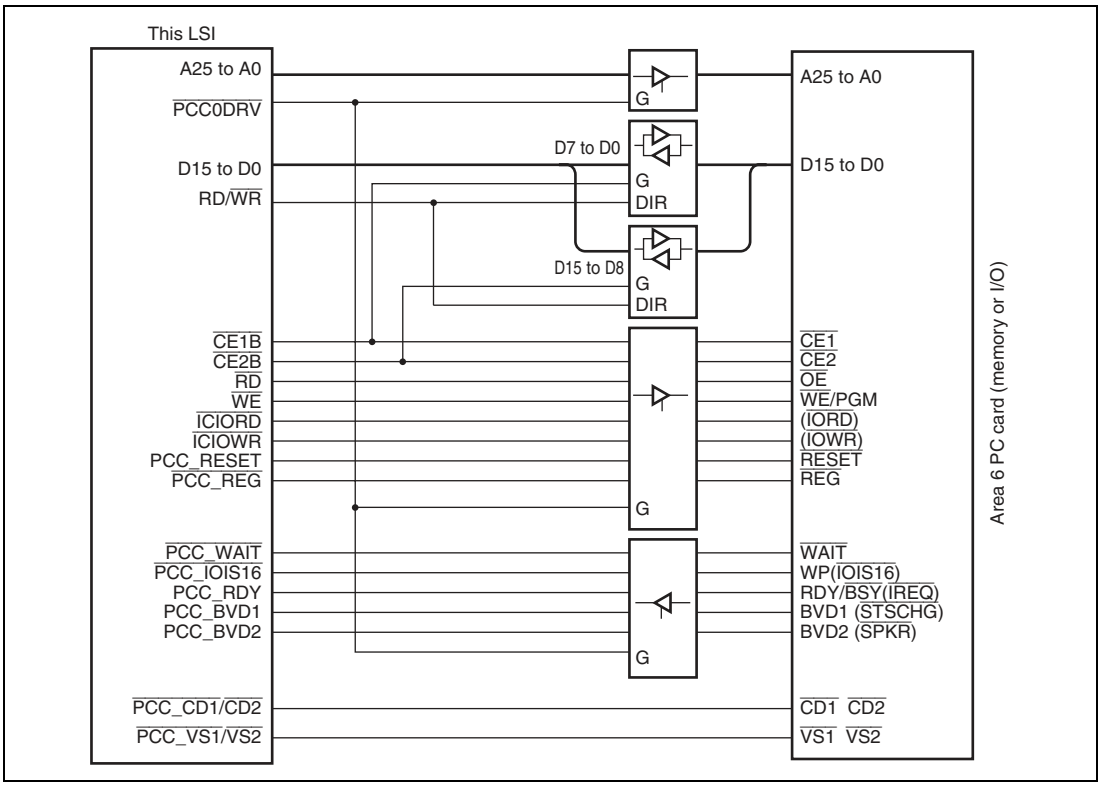


Figure 29.4 Interface

Table 29.3 PCMCIA Support Interface

	IC Memory Card Interface			I/O Card Interface			This LSI Corresponding Pin
Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	
1	GND		Ground	GND		Ground	—
2	D3	I/O	Data	D3	I/O	Data	D3
3	D4	I/O	Data	D4	I/O	Data	D4
4	D5	I/O	Data	D5	I/O	Data	D5
5	D6	I/O	Data	D6	I/O	Data	D6
6	D7	I/O	Data	D7	I/O	Data	D7
7	$\overline{CE1}$	I	Card enable	$\overline{CE1}$	I	Card enable	$\overline{CE1B}$
8	A10	I	Address	A10	I	Address	A10
9	$\overline{OE}$	I	Output enable	$\overline{OE}$	I	Output enable	$\overline{RD}$
10	A11	I	Address	A11	I	Address	A11
11	A9	I	Address	A9	I	Address	A9
12	A8	I	Address	A8	I	Address	A8
13	A13	I	Address	A13	I	Address	A13
14	A14	I	Address	A14	I	Address	A14
15	$\overline{WE}$/$\overline{PGM}$	I	Write enable	$\overline{WE}$/$\overline{PGM}$	I	Write enable	$\overline{WE}$
16	RDY/$\overline{BSY}$	O	Ready/busy	$\overline{IREQ}$	O	Interrupt request	PCC_RDY
17	VCC		Power supply	VCC		Power supply	—
18	VPP1		Programming power supply	VPP1		Programming and peripheral power supply	—
19	A16	I	Address	A16	I	Address	A16
20	A15	I	Address	A15	I	Address	A15
21	A12	I	Address	A12	I	Address	A12
22	A7	I	Address	A7	I	Address	A7
23	A6	I	Address	A6	I	Address	A6
24	A5	I	Address	A5	I	Address	A5
25	A4	I	Address	A4	I	Address	A4

	IC Memory Card Interface			I/O Card Interface			This LSI Corresponding Pin
Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	
26	A3	I	Address	A3	I	Address	A3
27	A2	I	Address	A2	I	Address	A2
28	A1	I	Address	A1	I	Address	A1
29	A0	I	Address	A0	I	Address	A0
30	D0	I/O	Data	D0	I/O	Data	D0
31	D1	I/O	Data	D1	I/O	Data	D1
32	D2	I/O	Data	D2	I/O	Data	D2
33	WP	O	Write protect	$\overline{\text{IOIS16}}$	O	16-bit I/O port	$\overline{\text{PCC\_IOIS16}}$
34	GND		Ground	GND		Ground	—
35	GND		Ground	GND		Ground	—
36	$\overline{\text{CD1}}$	O	Card detection	$\overline{\text{CD1}}$	O	Card detection	$\overline{\text{PCC\_CD1}}$
37	D11	I/O	Data	D11	I/O	Data	D11
38	D12	I/O	Data	D12	I/O	Data	D12
39	D13	I/O	Data	D13	I/O	Data	D13
40	D14	I/O	Data	D14	I/O	Data	D14
41	D15	I/O	Data	D15	I/O	Data	D15
42	$\overline{\text{CE2}}$	I	Card enable	$\overline{\text{CE2}}$	I	Card enable	$\overline{\text{CE2B}}$
43	$\overline{\text{VS1}}$	O	Voltage sense	$\overline{\text{VS1}}$	O	Voltage sense	$\overline{\text{PCC\_VS1}}$
44	RFU		Reserved	$\overline{\text{IORD}}$	I	I/O read	$\overline{\text{ICIORD}}$
45	RFU		Reserved	$\overline{\text{IOWR}}$	I	I/O write	$\overline{\text{ICIOWR}}$
46	A17	I	Address	A17	I	Address	A17
47	A18	I	Address	A18	I	Address	A18
48	A19	I	Address	A19	I	Address	A19
49	A20	I	Address	A20	I	Address	A20
50	A21	I	Address	A21	I	Address	A21
51	VCC		Power supply	VCC		Power supply	—

	IC Memory Card Interface			I/O Card Interface			This LSI Corresponding Pin
Pin	**Signal Name**	**I/O**	**Function**	**Signal Name**	**I/O**	**Function**	
52	VPP2		Programming power supply	VPP2		Programming and peripheral power supply	—
53	A22	I	Address	A22	I	Address	A22
54	A23	I	Address	A23	I	Address	A23
55	A24	I	Address	A24	I	Address	A24
56	A25	I	Address	A25	I	Address	A25
57	$\overline{\text{VS2}}$	O	Voltage sense	$\overline{\text{VS2}}$	O	Voltage sense	$\overline{\text{PCC\_VS2}}$
58	RESET	I	Reset	RESET	I	Reset	PCC_RESET
59	$\overline{\text{WAIT}}$	O	Wait request	$\overline{\text{WAIT}}$	O	Wait request	$\overline{\text{PCC\_WAIT}}$
60	RFU		Reserved	$\overline{\text{INPACK}}$	O	Input acknowledge	—
61	$\overline{\text{REG}}$	I	Attribute memory space select	$\overline{\text{REG}}$	I	Attribute memory space select	$\overline{\text{PCC\_REG}}$
62	BVD2	O	Battery voltage detection	$\overline{\text{SPKR}}$	O	Digital sound signal	PCC_BVD2
63	BVD1	O	Battery voltage detection	$\overline{\text{STSCHG}}$	O	Card status change	PCC_BVD1
64	D8	I/O	Data	D8	I/O	Data	D8
65	D9	I/O	Data	D9	I/O	Data	D9
66	D10	I/O	Data	D10	I/O	Data	D10
67	$\overline{\text{CD2}}$	O	Card detection	$\overline{\text{CD2}}$	O	Card detection	$\overline{\text{PCC\_CD2}}$
68	GND		Ground	GND		Ground	—

29.4.2 PC Card Interface Timing

(1) Memory Card Interface Timing

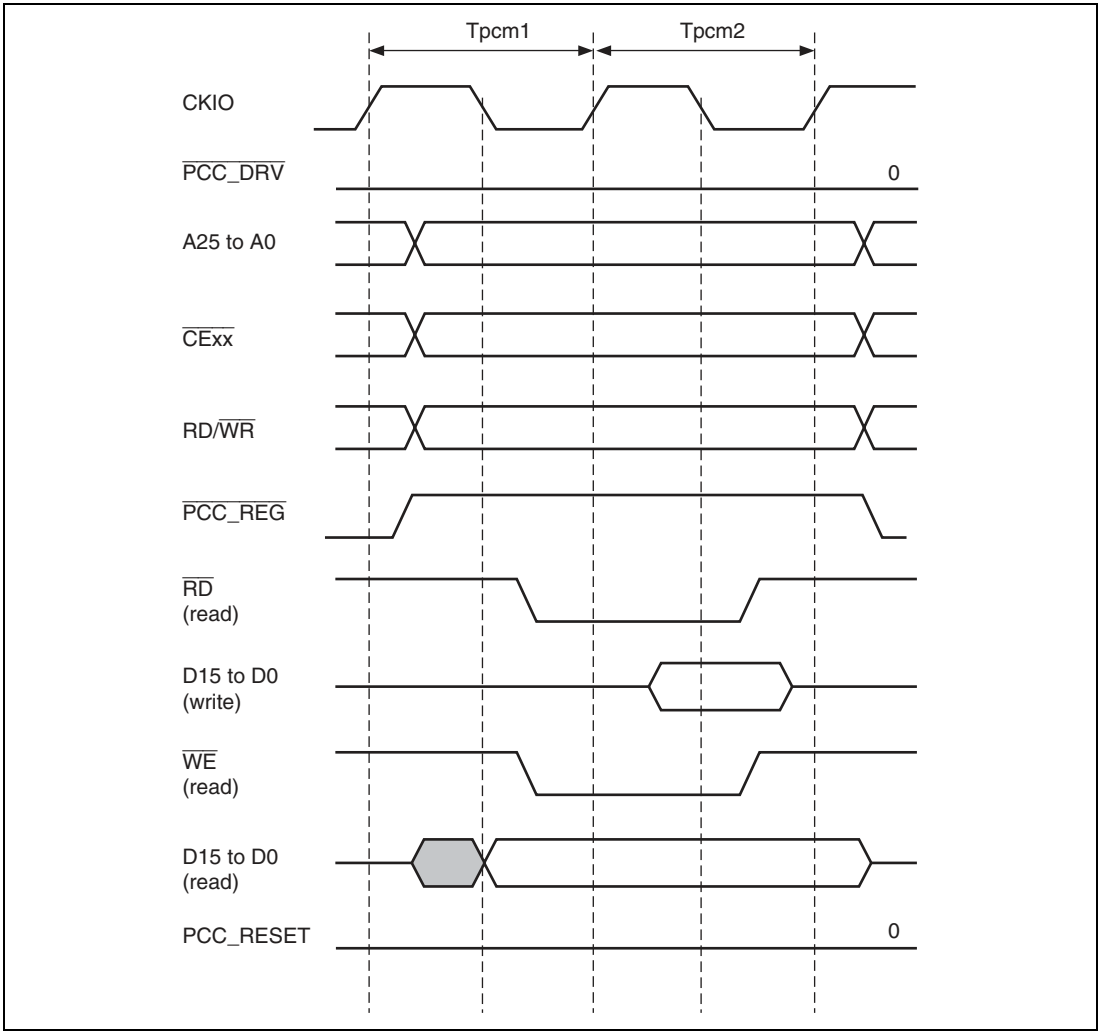


Figure 29.5 PCMCIA Memory Card Interface Basic Timing

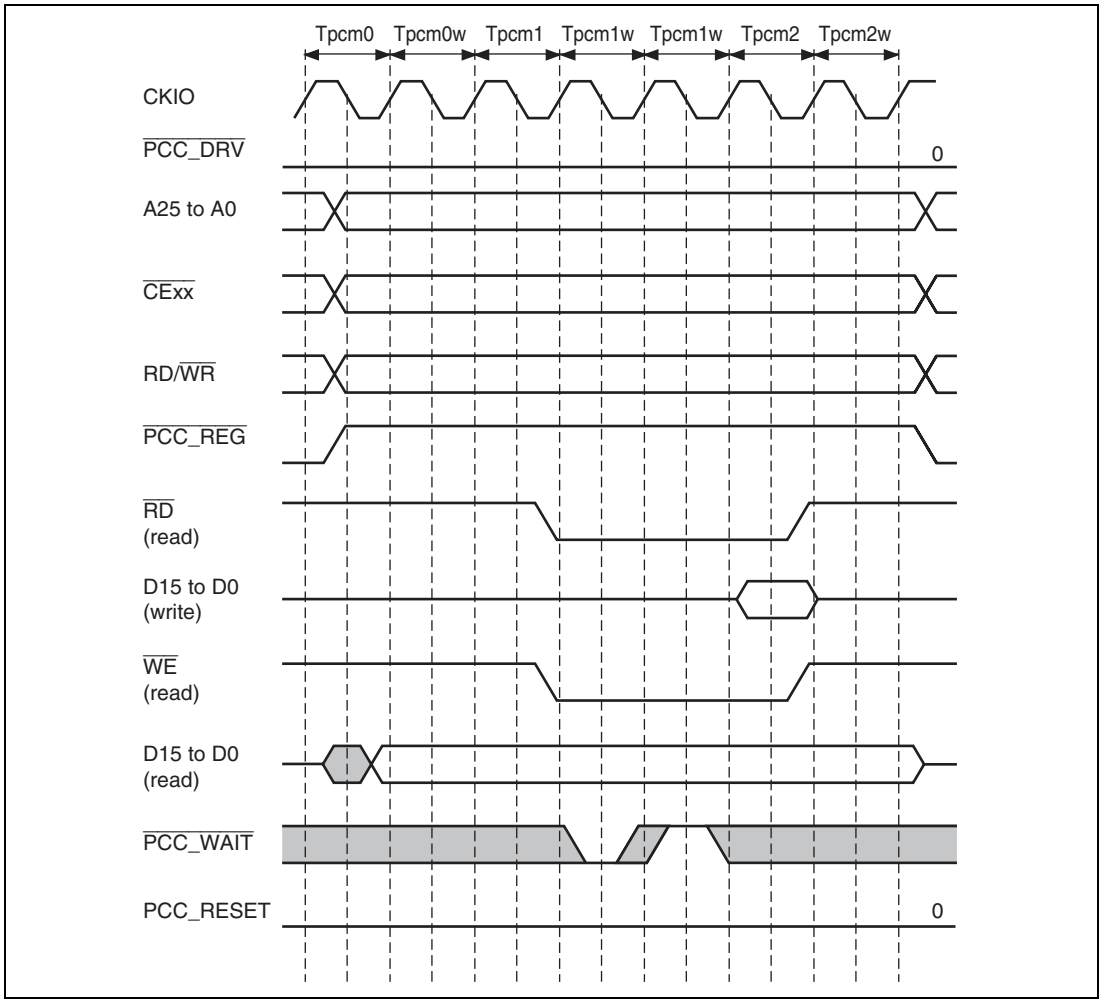


Figure 29.6 PCMCIA Memory Card Interface Wait Timing

(2) I/O Card Interface Timing

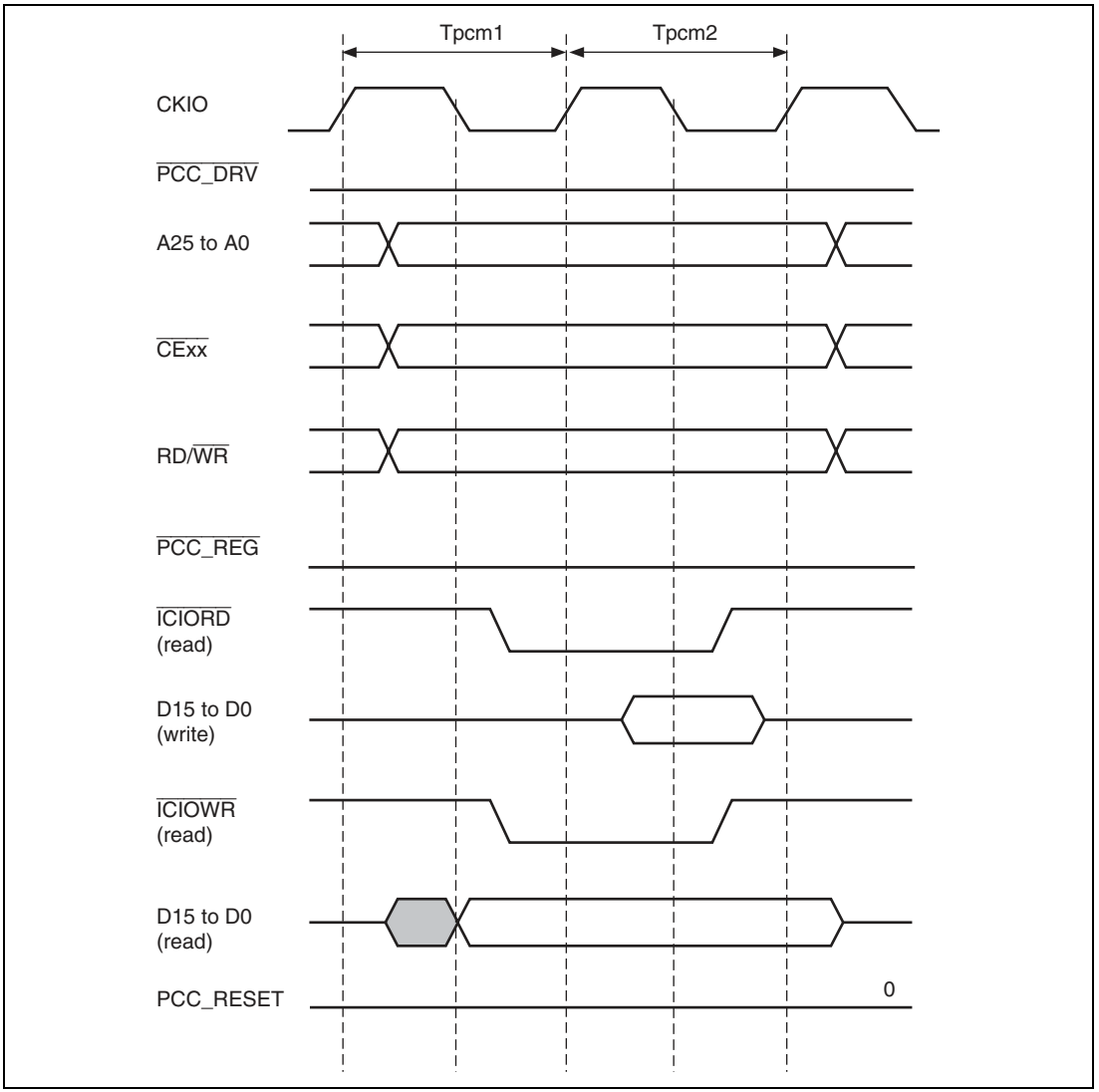


Figure 29.7 PCMCIA I/O Card Interface Basic Timing

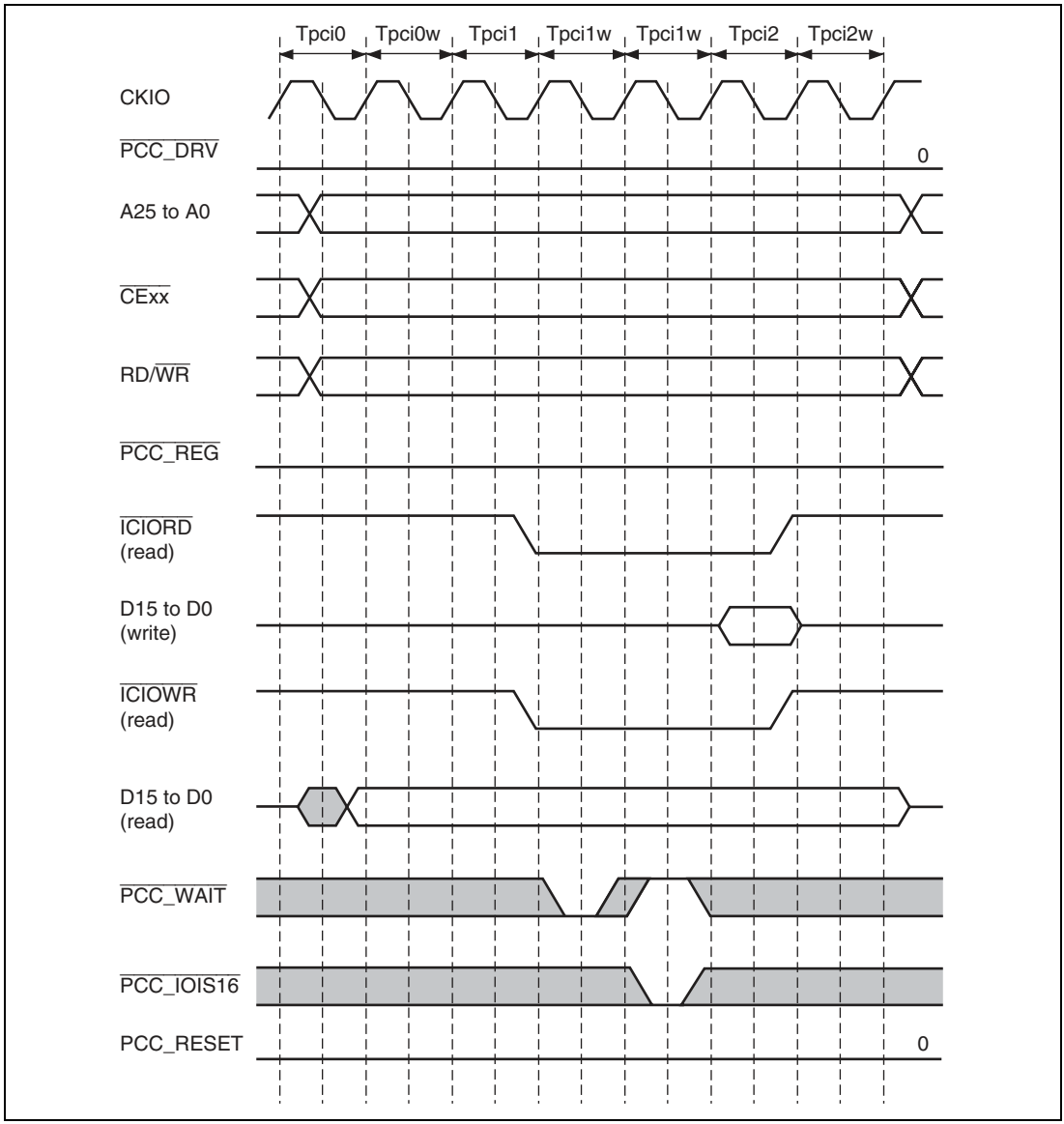


Figure 29.8 PCMCIA I/O Card Interface Wait Timing

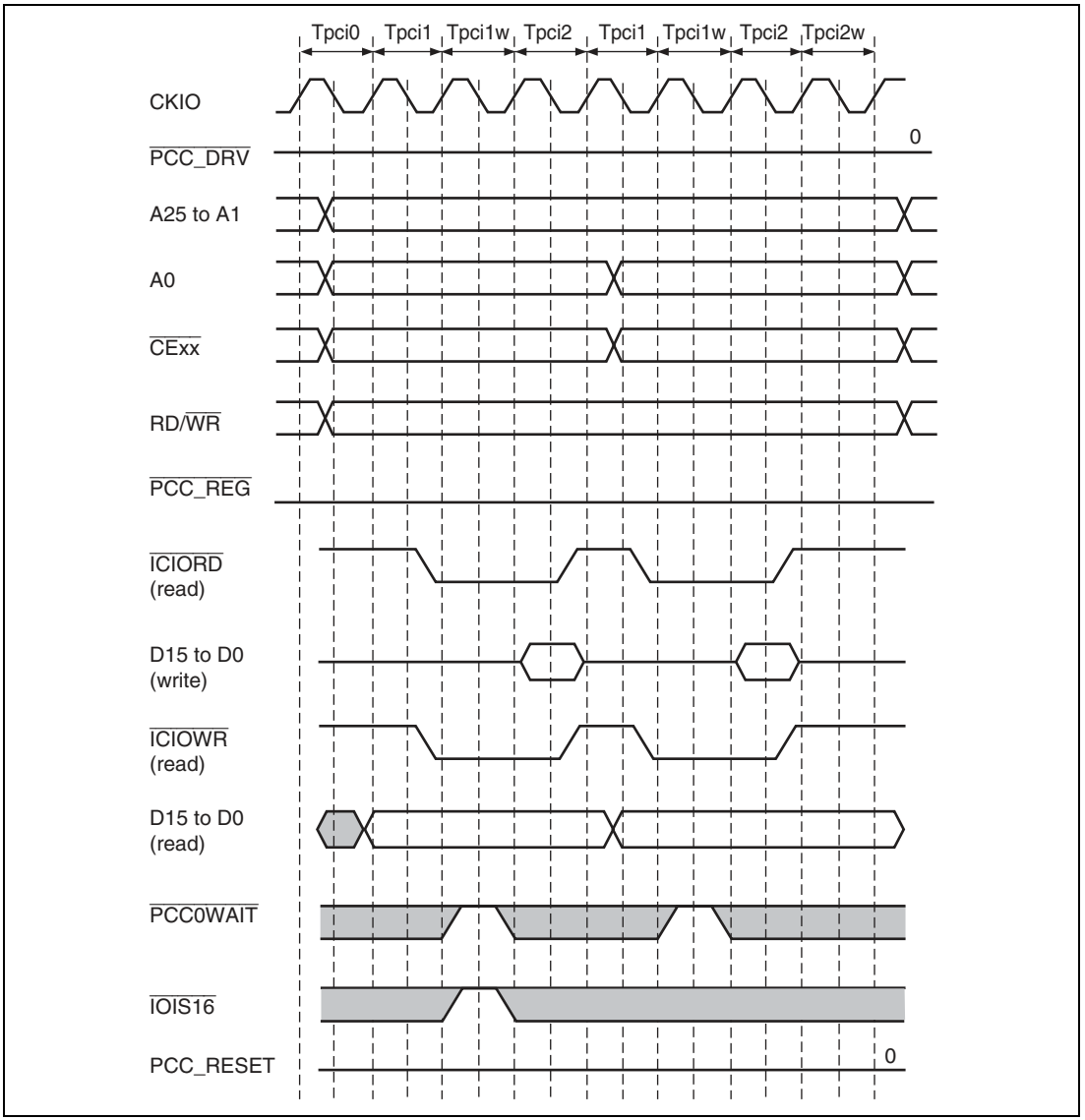


Figure 29.9 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface

29.5 Usage Notes

(1) External Bus Frequency Limit when Using PC Card

According to the PC card standard, the attribute memory access time is specified as 600 ns (3.3 V)/300 ns (5 V). Therefore, when this LSI accesses attribute memory, the bus cycle must be coordinated with the PC card interface timing. In this LSI, the timing can be adjusted by setting the TED, TEH, and PCW values in the CS6BWCR register, allowing a PC card to be used within the above frequency ranges.

The common memory access time and I/O access time (based on the $\overline{\text{IORD}}$ and $\overline{\text{IOWR}}$ signals) are also similarly specified (see table below), and a PC card must be used within the above ranges in order to satisfy all these specifications.

PC Card Space	Access Time (5 V Operation)	Access Time (3.3 V Operation)
Attribute memory	300 ns	600 ns
Common memory	250 ns	600 ns
I/O space (pulse width of $\overline{\text{IORD}}$ and $\overline{\text{IOWR}}$)	165 ns	165 ns

(2) Pin Function Control and Card Type Switching

When setting pin function controller pin functions to dedicated PC card use ("other function"), the disabled state should first be set in the card status change interrupt enable register (PCC0CSCIER). Also, the card status change register (PCC0CSCR) must be cleared after the setting has been made. However, this restriction does not apply to the card detection pins ($\overline{\text{CD1}}$ and $\overline{\text{CD2}}$).

When changing the card type bit (P0PCCT) in the area 6 general control register (PCC0GCR), the disabled state should first be set in the card status change interrupt enable register (PCC0CSCIER). Also, the card status change register (PCC0CSCR) must be cleared after the setting has been made.

Reason: When PC card controller settings are modified, the functions of PC card pins that generate various interrupts change, with the result that unnecessary interrupts may be generated.

(3) Setting Procedure when Using PC Card Controller

The following steps should be followed when using a card controller:

1. Set bit 12 (MAP) in the common control register (CMNCR) of bus state controller to 1.
2. Set bits 15 to 12 (TYPE3 to TYPE0) in the bus control register for CS6B (CS6BBCR) of the bus state controller to B'0101.
3. Set bit 4 (P0USE) in the area 6 general control register in the PC card controller to 1.
4. Set the pin function controller to custom PC card pin functions ("other functions").

Section 30 SIM Card Module (SIM)

The smart card interface supports IC cards (smart cards) conforming to the ISO/IEC 7816-3 (Identification Card) specification.

30.1 Features

- Communication 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
  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 etus (N = 255) when T = 1 (etu: Elementary Time Unit)
  Selectable direct convention/inverse convention
  Output clock can be fixed in high or low state
- Freely selectable bit rate by on-chip baud rate generator
- Four types of interrupt source
  Transmit data empty, receive data full, transmit/receive error, transmit complete
- 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 30.1 shows a block diagram of the smart card interface.

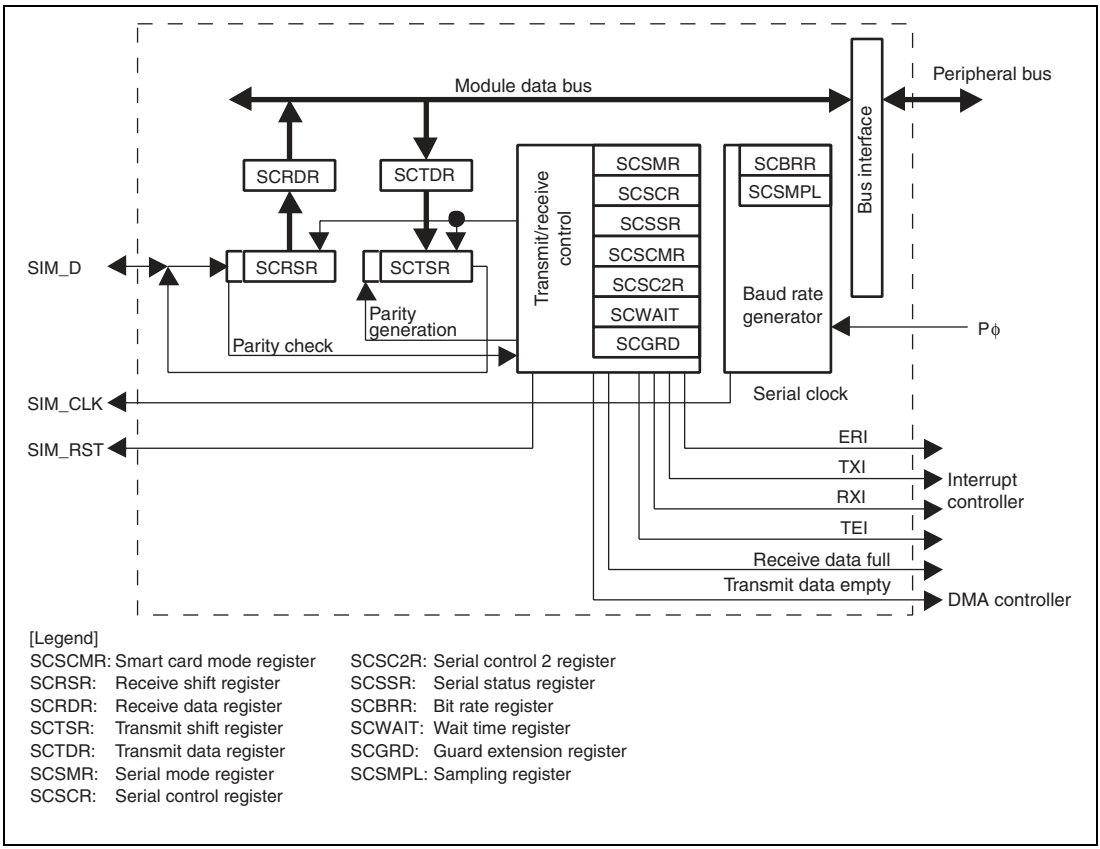


Figure 30.1 Smart Card Interface

30.2 Input/Output Pins

The pin configuration of the smart card interface is shown in table 30.1.

Table 30.1 Pin Configuration

Name	Abbreviation	I/O	Function
SIM data	SIM_D*	I/O	Transmit/receive data input/output
SIM clock	SIM_CLK	Output	Clock output
SIM reset	SIM_RST	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.

30.3 Register Descriptions

The SIM card module has the following registers. Refer to section 37, List of Registers, for more details on the addresses and states of these registers in each operating mode.

- Serial mode register (SCSMR)
- Bit rate register (SCBRR)
- Serial control register (SCSCR)
- Transmit shift register (SCTSR)
- Transmit data register (SCTDR)
- Serial status register (SCSSR)
- Receive shift register (SCRSR)
- Receive data register (SCRDR)
- Smart card mode register (SCSCMR)
- Serial control 2 register (SCSC2R)
- Guard extension register (SCGRD)
- Wait time register (SCWAIT)
- Sampling register (SCSMPL)

30.3.1 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	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.

30.3.2 Bit Rate Register (SCBRR)

SCBRR is an 8-bit readable/writable register that sets the transmit/receive bit rate.

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	BRR2	1	R/W	Set the transmit/receive bit rate 2 to 0.
1	BRR1	1	R/W	
0	BRR0	1	R/W	

The SCBRR setting can be determined from the following formula.

$$\text{sck\_frequency} = \frac{P\phi}{2\,(\text{brr} + 1)}$$

The units of Pϕ (peripheral clock frequency) and sck_frequency are MHz.

30.3.3 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	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, ERS, 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
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.

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	Clock Enable 1, 0
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 at low 01: Clock output as the output pin 10: Fix the output pin at high 11: Clock output as the output pin

30.3.4 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 is completed, transmit data is automatically transferred from SCTDR 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.

Direct reading and writing of SCTSR from the CPU or DMAC is not possible.

30.3.5 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	Bit Name	Initial Value	R/W	Description
7 to 0	SCTD7 to SCTD0	All 1	R/W	Transmit Data Store data for serial transmission.

30.3.6 Serial Status Register (SCSSR)

SCSSR is an 8-bit readable/writable register that indicates the operating state of the smart card interface.

Bit	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 CCSCR 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 transmission is ended. The TEND flag is read-only, and cannot be written. 0: Indicates that transmission is in progress [Clearing condition] When transmit data is transferred from SCTDR to SCTSR, and serial transmission is initiated. 1: Indicates that transmission 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 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. Notes: 1. Even if the RE bit in SCSCR is cleared to 0, the WAIT_ER flag is unaffected, and the previous state is retained. 2. In T = 0 mode, even if the setting condition for the WAIT_ER flag is satisfied when the RE bit is set to 1, the WAIT_ER flag may not be set to 1. In this case, the RE bit has been set to 1, then the WAIT_ER flag is set to 1 after 60 × (SCWAIT + n) etu (n ≥ 0: depending on the timing for setting the RE bit to 1) since the last transmission or reception. 3. In T = 0 mode, if the WAIT_ER flag does not need to be set to 1 after 60 × (SCWAIT + n) etu since the last transmission or reception, the mode should be changed from T = 0 to T = 1, and changed to T = 0 again by the PB bit in SCSCMR. In T = 1 mode, if the WAIT_ER flag does not need to be set to 1 after (SCWAIT) etu since the last reception, the mode should be changed from T = 1 to T = 0, and changed to T = 1 again by the PB bit in SCSCMR.

Bit	Bit Name	Initial Value	R/W	Description
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

30.3.7 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. The CPU or DMAC cannot directly read from or write to SCRSR.

30.3.8 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. SCRDR cannot be written to by the CPU or DMAC.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SCRD7 to SCRD0	All 0	R	Receive Data Store received serial data.

30.3.9 Smart Card Mode Register (SCSCMR)

SCSCMR is an 8-bit readable/writable register that selects functions of the smart card interface.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
6	LCB	0	R/W	Last Character When this bit is set to 1, the character protection time is 2 etus, and the setting of the guard extension register is invalid. 0: The character protection time is determined by the value of the guard extension register. 1: The character protection time is 2 etus.
5	PB	0	R/W	Protocol Selects the T = 0 or T = 1 protocol. 0: The smart card interface operates according to the T = 0 protocol. 1: The smart card interface operates according to the T = 1 protocol.
4	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
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.

Bit	Bit Name	Initial Value	R/W	Description
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.
1	RST	0	R/W	Smart Card Reset Controls the output of the SIM_RST pin of the smart card interface. 0: The SIM_RST pin of the smart card interface outputs low level. 1: The SIM_RST pin of the smart card interface outputs high level.
0	SMIF	1	R/W	Smart Card Interface Mode Select This bit is always read as 1. The write value should always be 1.

30.3.10 Serial Control 2 Register (SCSC2R)

SCSC2R is an 8-bit readable/writable register that enables or disables receive data full interrupt (RXI) requests.

Bit	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. Receive data full interrupt (RXI) requests are determined by the RIE bit setting.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

30.3.11 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	SCGRD7 to SCGRD0	All 0	R/W	Guard Extension Indicate the time added for character protection after transmitting a character to the smart card. The interval between the start of two successive characters is 12 etus (no addition) when the value of this register is H'00, is 13 etus when the value is H'01, and so on, up to 266 etus for H'FE. If the value of this register is H'FF, the interval between the start of two successive characters is 11 etus in T = 1 mode and is 12 etus in T = 0 mode.

30.3.12 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	SCWAIT15 to SCWAIT0	All 0	R/W	Wait Time Register • T = 0 In this 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 etus. • T = 1 In this 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.

30.3.13 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	SCSMPL10 to SCSMPL0	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.

30.4 Operation

30.4.1 Overview

The main functions of the smart card interface are as follows.

- One frame consists of 8-bit data and one 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 etus after the start bit.
- During transmission in T = 0 mode, if an error signal is sampled, after 2 etus or more have elapsed, the same data is automatically transmitted.
- Only asynchronous communication functions are supported; there is no clocked synchronous communication function.

30.4.2 Data Format

Figure 30.2 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.

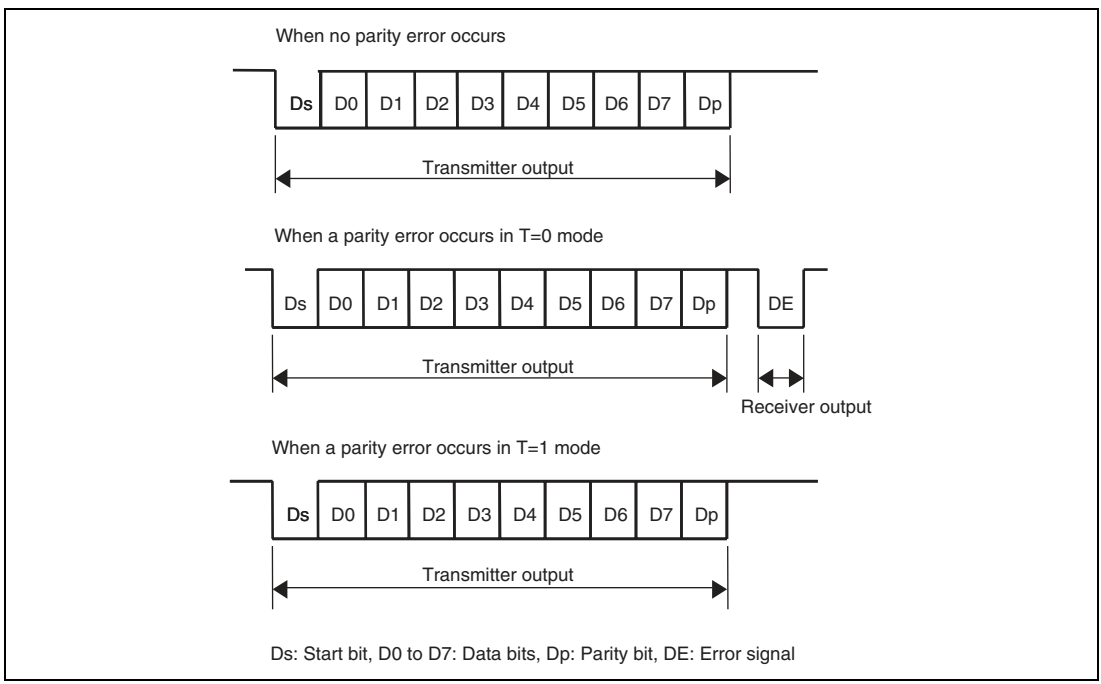


Figure 30.2 Data Format Used by Smart Card Interface

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.

30.4.3 Register Settings

Table 30.2 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 30.2 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	PE	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	SCTD7	SCTD6	SCTD5	SCTD4	SCTD3	SCTD2	SCTD1	SCTD0
SCSSR	TDRE	RDRF	ORER	ERS	PER	TEND	WAIT_ER	0
SCRDR	SCRD7	SCRD6	SCRD5	SCRD4	SCRD3	SCRD2	SCRD1	SCRD0
SCSCMR	0	LCB	PB	0	SDIR	SINV	RST	1
SCSC2R	EIO	0	0	0	0	0	0	0
SCWAIT	SCWAIT15 to SCWAIT0							
SCGRD	SCGRD7 to SCGRD0							
SCSMPL	SCSMPL10 to SCSMPL0, bits 11 to 15 are 0							

- 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.
- Bit rate register (SCBRR) setting

  Sets the bit rate. For the method of computing settings, refer to section 30.4.4, Clocks.
- 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 30.4.4, Clocks.
- 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. The SMIF bit is always set to 1.

  Figure 30.3 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 30.3 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, the only 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.

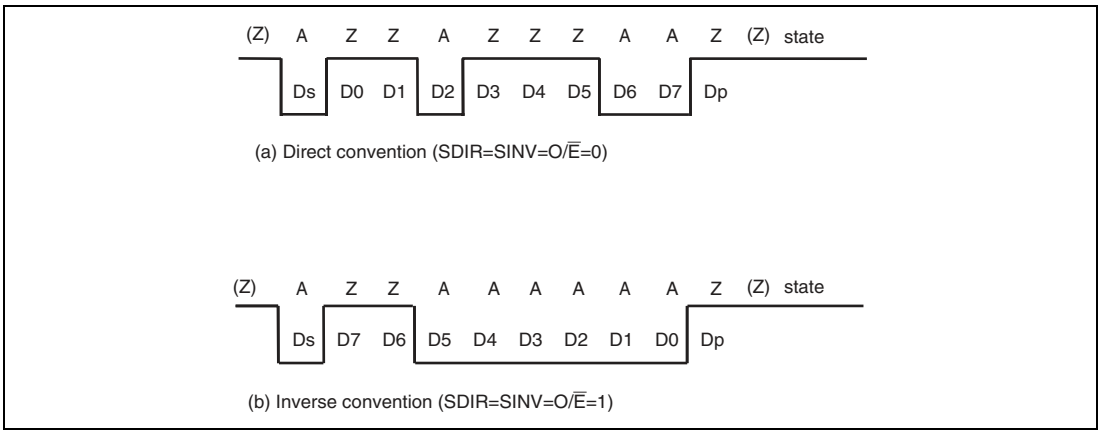


Figure 30.3 Examples of Start Character Waveforms

30.4.4 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 30.3

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 30.3 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.

30.4.5 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 30.4.

Step (1) to step (7) of figure 30.4 correspond to the following operation.

1. Clear the TE and RE bits in the serial control register (SCSCR) to 0.
2. Clear the error flags PER, ORER, ERS, and WAIT_ER in the serial status register (SCSSR) to 0.
3. Set the parity bit (O/$\overline{E}$ bit) in the serial mode register (SCSMR).
4. Set the LCB, PB, SMIF, SDIR, and SINV bits in the smart card mode register (SCSCMR).
5. Set the value corresponding to the bit rate to the bit rate register (SCBRR).
6. 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.
7. After waiting at least 1 etu, set 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.

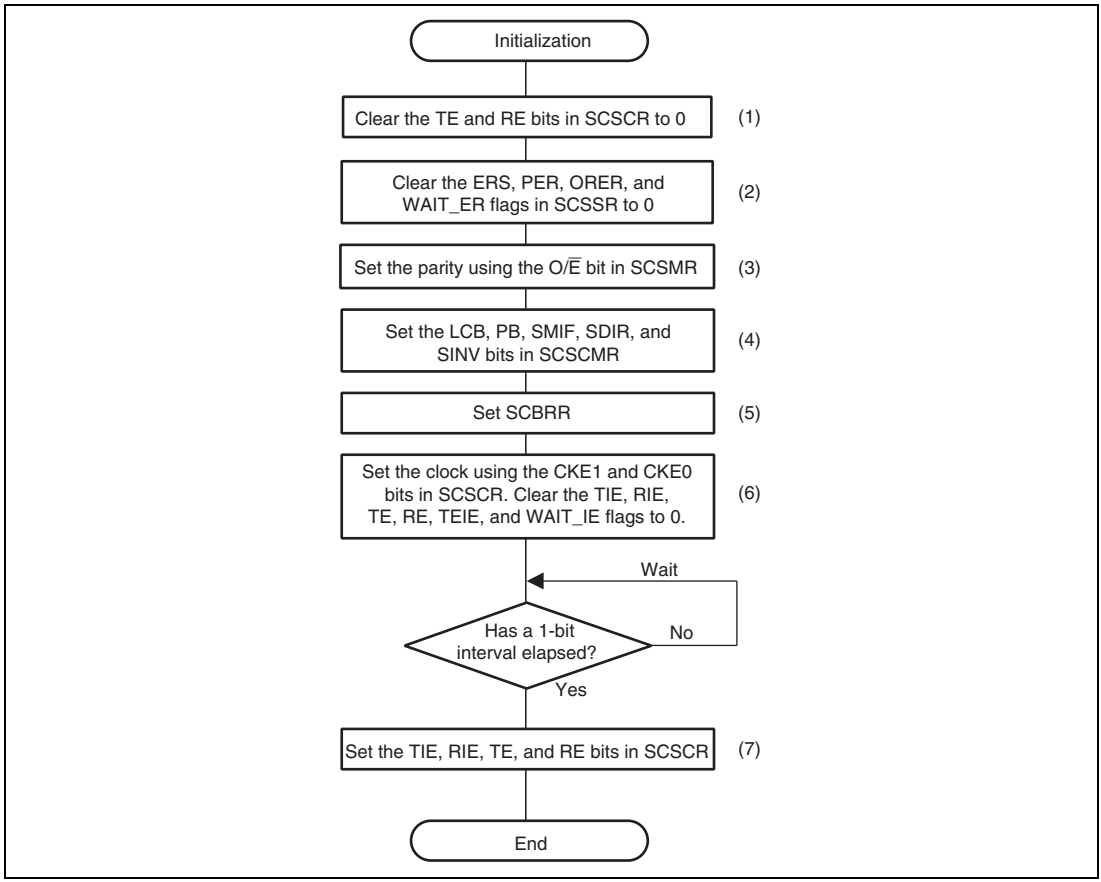


Figure 30.4 Example of Initialization Flow

(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 30.5.

Step (1) to step (6) of figure 30.5 correspond to the following operation.

1. Follow the initialization procedure above to initialize the smart card interface.
2. Confirm that the ERS bit (error flag) in SCSSR is cleared to 0.
3. Repeat steps (2) and (3) until it can be confirmed that the TDRE flag in SCSSR is set to 1.
4. 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.
5. When performing continuous data transmission, return to step (2).
6. 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 30.4.5, Data Transmit/Receive Operation.

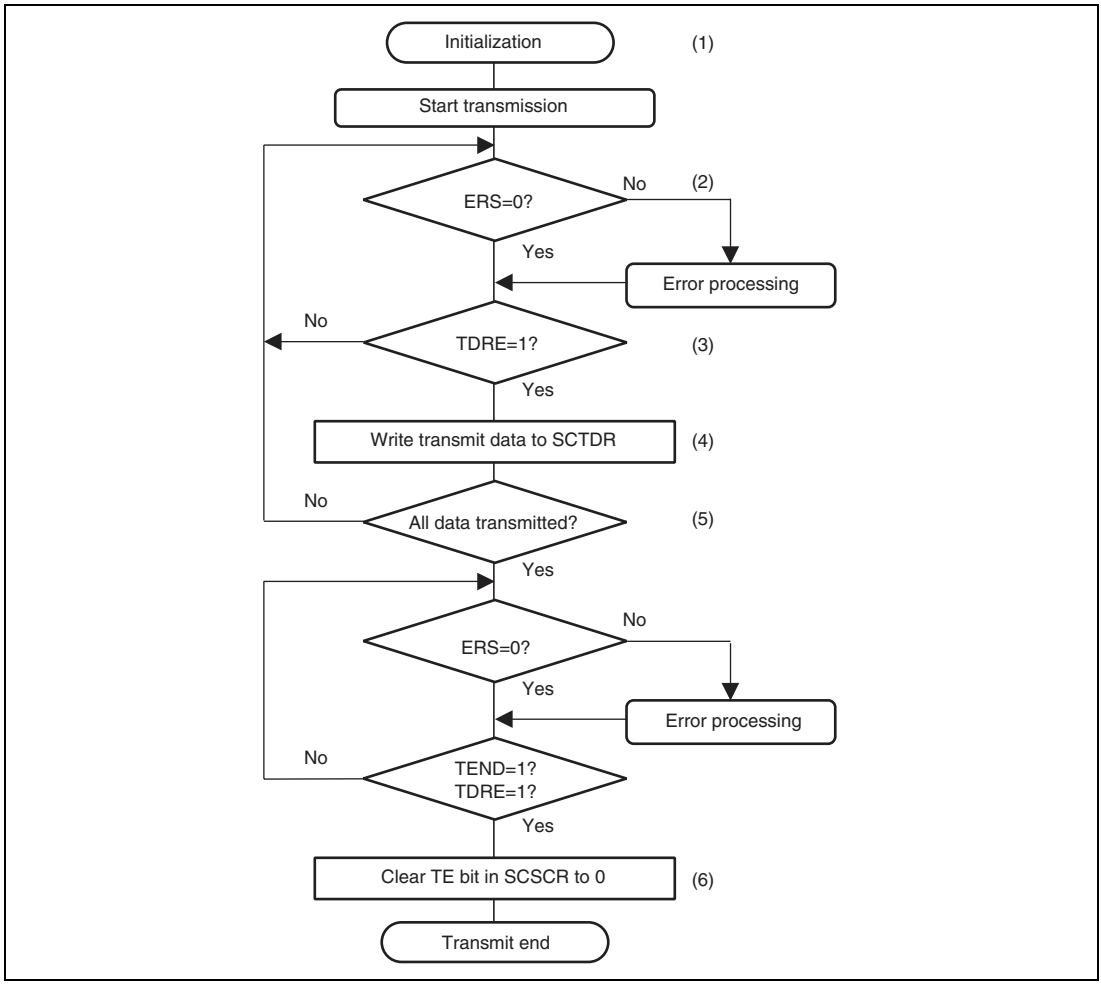


Figure 30.5 Example of Transmit Processing

(3) Serial Data Reception

An example of data receive processing in smart card mode is shown in figure 30.6.

Step (1) to step (6) of figure 30.6 correspond to the following operation.

1. Follow the initialization procedure above to initialize the smart card interface.
2. 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.
3. Repeat steps (2) and (3) in the figure until it can be confirmed that the RDRF flag is set to 1.
4. Read received data from SCRDR.
5. When receiving data continuously, return to step (2).
6. 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 30.4.5, 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.

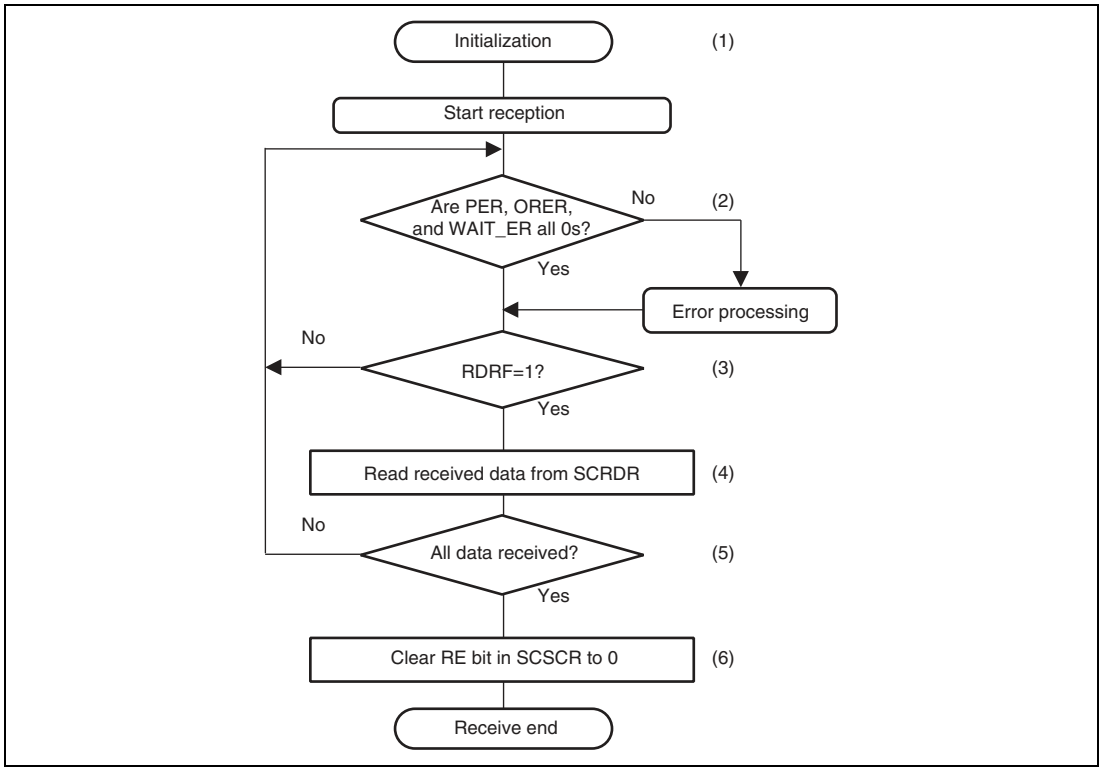


Figure 30.6 Example of Receive Processing

(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 30.4 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 30.4 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.

In transmission, when the TDRE flag in SCSSR is set to 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 receive operation, when the RDRF flag in SCSSR is set to 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 and the EIO 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.

30.5 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 30.7.

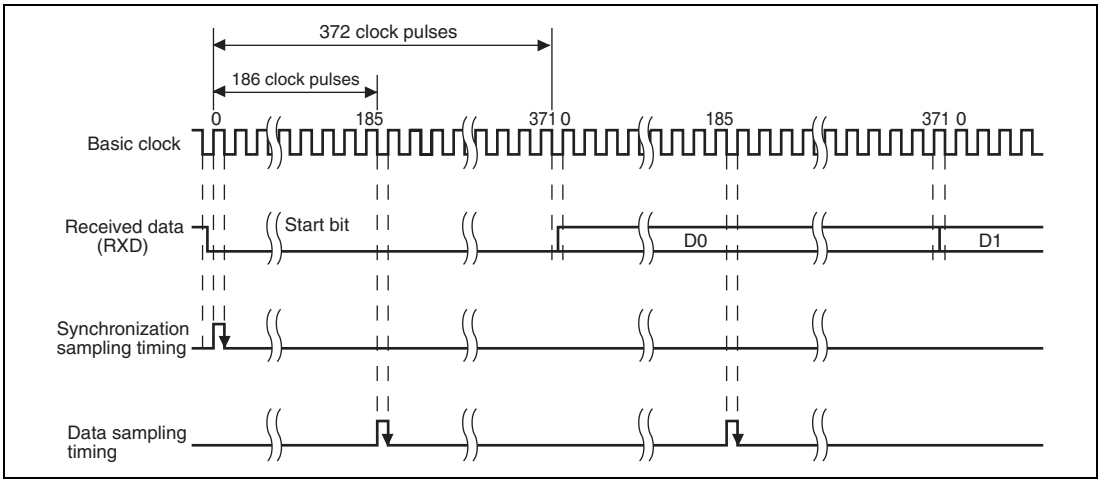


Figure 30.7 Receive Data Sampling Timing in Smart Card Mode

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.

(a) Retransmission when the smart card interface is in receive mode (T = 0)

Figure 30.8 shows retransmit operations when the smart card interface is in receive mode. Step (1) to step (5) of figure 30.8 correspond to the following operation.

1. 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.
2. The RDRF bit in SCSSR is not set for frames in which a parity error occurs.
3. If no error is detected as a result of checking the received parity bit, the PER bit in SCSSR is not set.
4. 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.

5. If a normal frame is received, the pin retains its high-impedance state at the timing for transmission of error signals.

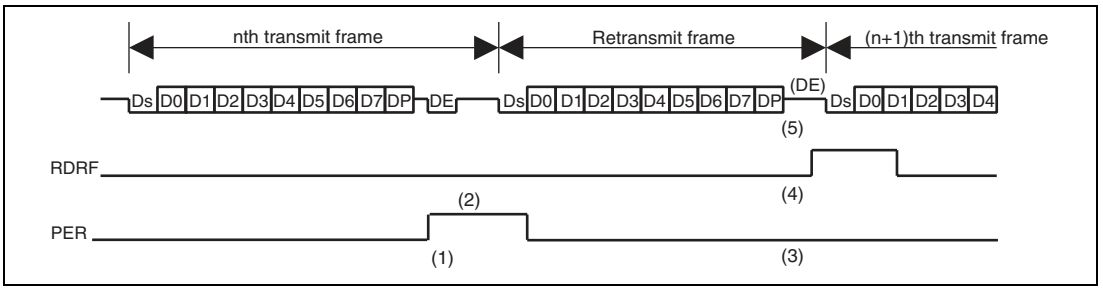


Figure 30.8 Retransmission when Smart Card Interface is in Receive Mode

(b) Retransmission when the smart card interface is in transmit mode (T = 0)

Figure 30.9 shows retransmit operations when the smart card interface is in transmit mode. Step (1) to step (4) of figure 30.9 correspond to the following operation

1. 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.
2. In T = 0 mode, the TEND bit in SCSSR is not set for a frame when an error signal indicating an error is received.
3. If no error signal is returned from the receive side, the ERS bit in SCSSR is not set.
4. If no error signal is returned from the receive side, it is assumed that transmission of one frame, including retransmission, is completed, and the TEND bit in SCSSR is set to 1. At this time, if the TIE bit in SCSCR is set to enable, a TEI interrupt request is issued.

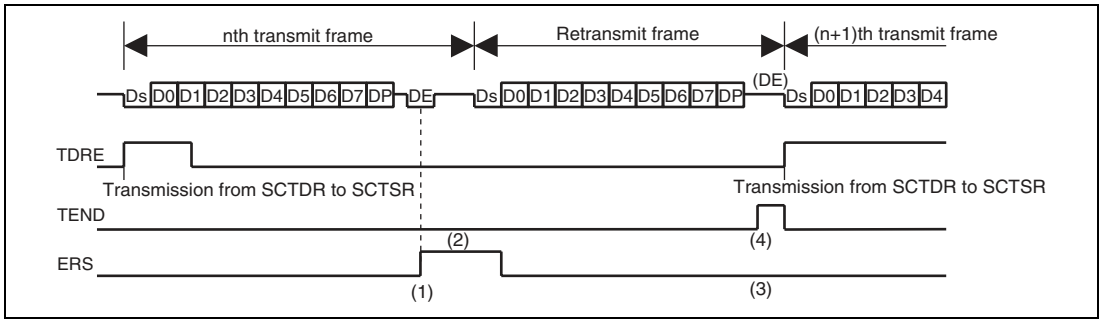


Figure 30.9 Retransmit Standby Mode (Clock Stopped) when Smart Card Interface is in Transmit Mode

(3) Standby Mode Switching

When switching between smart card interface mode and standby mode, in order to retain the clock duty, the following switching procedure should be used. Step (1) to step (7) of figure 30.10 correspond to the following operation.

- 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.

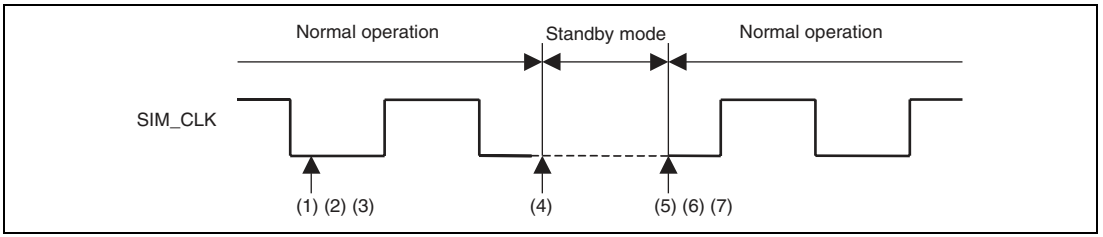


Figure 30.10 Procedure for Stopping Clock and Restarting

(4) Power-On and Clock Output

In order to retain the clock duty from power-on, the following switching procedure should be used.

1. The initial state is set to port-input with high impedance. 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.

(5) Pin Connections

An example of pin connections for the smart card interface is shown in figure 30.11.

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} 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.

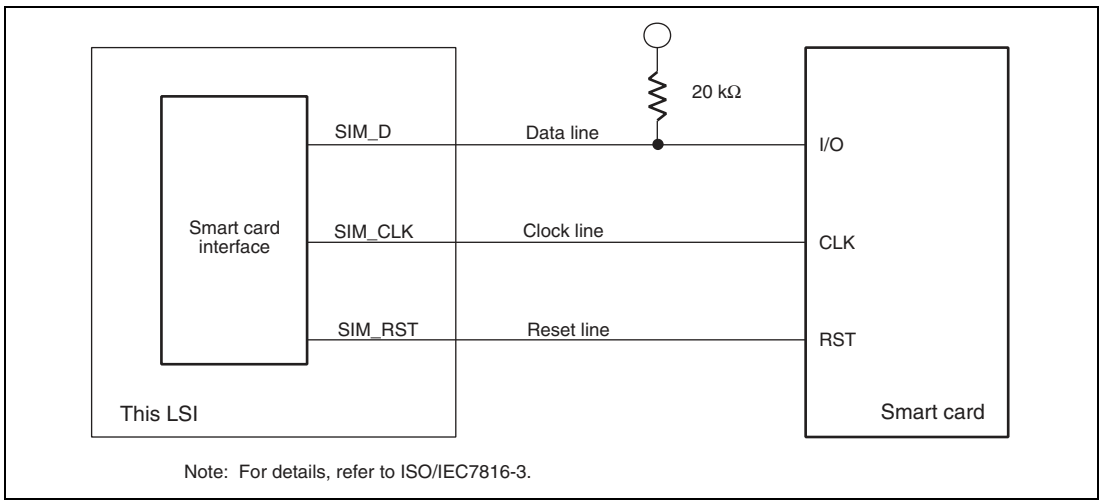


Figure 30.11 Example of Pin Connections in Smart Card Interface

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.

(6) Transmit End Interrupt

In continuous transmission, when the TEIE bit is always set to 1, the TEND bit is set to 1 at a transmit end. Therefore, the unnecessary transmit end interrupt (TEI) request occurs.

When SCTSR starts transmitting after the last transmit data is written to SCTDR, the TEIE bit in SCSCR should be set to 1 so that the occurrence of the unnecessary TEI interrupt request can be prevented.

The waveform of the timing to set the TEIE bit to 1 is shown in figure 30.12.

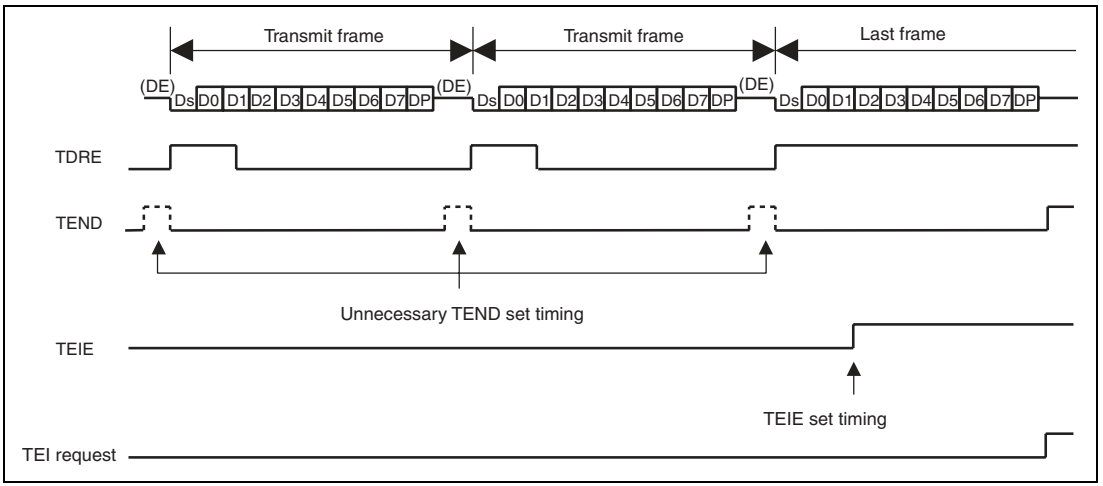


Figure 30.12 TEIE Set Timing

Section 31 MultiMediaCard Interface (MMCIF)

This LSI includes a MultiMediaCard interface (MMCIF). The MMCIF has MMC mode. The MMCIF is a clock-synchronous serial interface that transmits/receives data that is distinguished in terms of command and response. A number of command/responses are predefined in the MultiMediaCard. As the MMCIF specifies a command code and command type/response type upon the issuance of a command, commands extended by the secure MultiMediaCard (Secure-MMC) and additional commands can be supported in future within the range of combinations of currently defined command types/response types.

31.1 Features

- Interface that complies with 'The MultiMediaCard System Specification Version 3.1'
- Supports MMC mode
- For the card interface, 16.5-Mbps bit rate (max) at a peripheral-module operating clock of 33 MHz
- Incorporates sixty-four 16-bit data-transfer FIFOs
- DMA transfer request can be issued
- Four interrupt sources
  FIFO empty/full, command/response/data transfer complete, transfer error, and FIFO ready
- MMC mode
  Interface via the CLK output (transfer clock output) pin, CMD input/output (command transmission/response reception) pin, and DAT input/output (data transmission/reception) pin

A block diagram of the MMCIF is shown in figure 31.1.

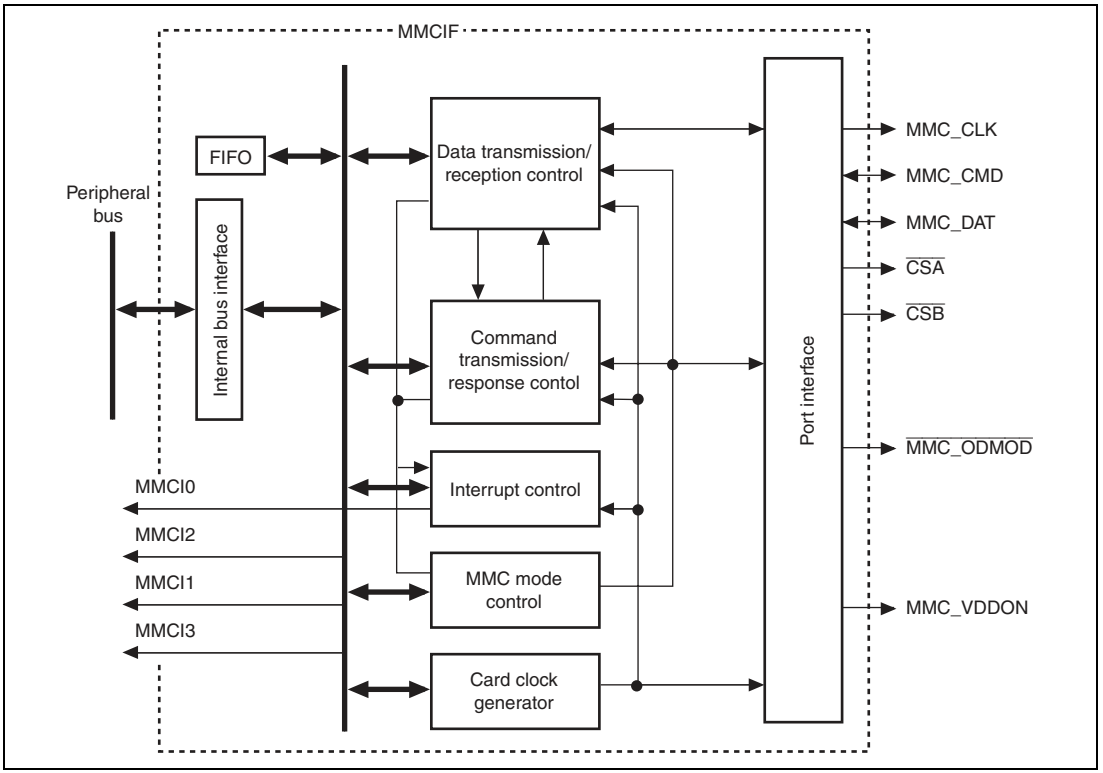


Figure 31.1 Block Diagram of MMCIF

31.2 Input/Output Pins

Table 31.1 summarizes the pins of the MMCIF.

Table 31.1 Pin Configuration

Pin Name	Abbreviation (MMC)	I/O	Function
MMC_CLK	CLK	Output	Clock output pin
MMC_CMD	CMD	I/O	Command output/response input pin
MMC_DAT	DAT	I/O	Data input/output pin
MMC_VDDON	MMC_VDDON	Output	MMC power control
$\overline{\text{MMC\_ODMOD}}$	MMC_ODMOD	Output	Open drain mode control (active-low signal)

Note: To describe transmission and reception operation, the data-transmission and data-reception sides as MCTXD and MCRXD, respectively. To insert/detach a card or for signals for switching open-drain/CMOS mode, use ports of this LSI.

31.3 Register Descriptions

The MMCIF has the following registers. Refer to section 37, List of Registers, for more details on the addresses and states of these registers in each operating mode.

- Mode register (MODER)
- Command type register (CMDTYR)
- Response type register (RSPTYR)
- Transfer byte number count register (TBCR)
- Transfer block number counter (TBNCR)
- Command registers 0 to 5 (CMDR0 to CMDR5)
- Response registers 0 to 16 (RSPR0 to RSPR16)
- Response register D (RSPRD)
- Command start register (CMDSTRT)
- Operation control register (OPCR)
- Command timeout control register (CTOCR)
- Data timeout register (DTOUTR)
- Card status register (CSTR)
- Interrupt control registers 0 and 1 (INTCR0 and INTCR1)
- Interrupt status registers 0 and 1 (INTSTR0 and INTSTR1)
- Pin mode control register (IOMCR)
- Transfer clock control register (CLKON)
- VDD/open drain control register (VDCNT)
- Data register (DR)
- FIFO pointer clear register (FIFOCLR)
- DMA control register (DMACR)
- Interrupt control register 2 (INTCR2)
- Interrupt status register 2 (INTSTR2)

31.3.1 Mode Register (MODER)

MODER specifies the MMCIF operating mode. The MMCIF has an operating mode: MMC mode.

Three signals, clock, command, and data signals, are used as the interfaces between the host system and the MMC in MMC mode. The clock signal is used to make the host system and the MMC synchronize each other. The command signal is used to issue a command from the host system to the MMC and send a response from the MMC to the host system. The data signal is used to write data to and read data from the MMC. The command and data signals are bidirectional buses.

The following sequence should be repeated when the MMCIF uses the MMC: Send a command, wait for the end of the command sequence and the end of the data busy state, and send the next command.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
0	—	0	R/W	Reserved The write value should always be 0.

31.3.2 Command Type Register (CMDTYR)

CMDTYR specifies the command format in conjunction with RSPTYR. Bits TY1 and TY0 specify the existence and direction of transfer data, and bits TY6 to TY2 specify the additional settings. All of bits TY6 to TY2 should be cleared to 0 or only one of them should be set to 1. Bits TY6 to TY2 can only be set to 1 if the corresponding settings in bits TY1 and TY0 allow that setting. If this register is not set correctly, operation cannot be guaranteed.

To perform single-block transfer, bits TY1 and TY0 should be set to 01 or 10 and bits TY6 to TY2 to 0.

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	TY6	0	R/W	Specifies the pre-defined multiblock transfer. Bits TY1 and TY0 should be set to 01 or 10. When the command which specifies this bit is used, the transfer block size and the number of transfer blocks should be specified in TBCR and TBNCR, respectively.
5	TY5	0	R/W	Specifies the multiblock transfer while the secure MMC is used. Bits TY1 and TY0 should be set to 01 or 10. When the command which specifies this bit is used, the transfer block size and the number of transfer blocks should be specified in TBCR and TBNCR, respectively.
4	TY4	0	R/W	This bit is set to 1 when the CMD12 command is issued. Bits TY1 and TY0 should be set to 00.
3	TY3	0	R/W	Specifies the stream transfer. Bits TY1 and TY0 should be set to 01 or 10. The stream transfer can be used only in MMC mode. The command sequence of the stream transfer specified by this bit ends when it is stopped by the CMD12 command.
2	TY2	0	R/W	Specifies the open-ended multiblock transfer. Bits TY1 and TY0 should be set to 01 or 10. The command sequence of the stream transfer specified by this bit ends when it is stopped by the CMD12 command.
1	TY1	0	R/W	Specify the existence and direction of transfer data.
0	TY0	0	R/W	00: A command without data transfer 01: A command with read data reception 10: A command with write data transmission 11: Setting prohibited

Table 31.2 summarizes the correspondence between the commands described in the MultiMediaCard System Specification Version 3.1 and the settings of CMDTYR and RSPTYR.

31.3.3 Response Type Register (RSPTYR)

RSPTYR specifies command format in conjunction with CMDTYR. Bits RTY2 to RTY0 are used to specify the number of response bytes, and bits RTY5 and RTY4 are used to make additional settings.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
5	RTY5	0	R/W	This bit is set when a command with R1b response is issued.
4	RTY4	0	R/W	Makes settings so that the command response (other than R2 response) CRC is checked by CRC7. Bits RTY2 to RTY0 should be set to 100.
3	RTY3	0	R/W	Reserved
2	RTY2	0	R/W	These bits specify the number of command response bytes.
1	RTY1	0	R/W	000: A command needs no command response.
0	RTY0	0	R/W	001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: A command needs a 6-byte command response. Specified by R1, R1b, R3, R4, and R5 responses. 101: A command needs a 17-byte command response. Specified by R2 response. 110: Setting prohibited 111: Setting prohibited

Note: Checking CRC by RTY4 is not checking the command response CRC error bit but checking the command response CRC. This checking is not performed for the CRC of the R2 command response in MMC mode.

Table 31.2 summarizes the correspondence between the commands described in The MultiMediaCard System Specification Version 3.1 and the settings of CMDTYR and RSPTYR.

Table 31.2 Correspondence between Commands and Settings of CMDTYR and RSPTYR

- MMC Mode

CMD INDEX	Abbreviation	resp	CMDTYR 6	5	4	3	2	1 to 0	RSPTYR 6	5	4	2 to 0
CMD0	GO_IDLE_STATE	—						00				000
CMD1	SEND_OP_COND	R3						00				100
CMD2	ALL_SEND_CID	R2						00				101
CMD3	SET_RELATIVE_ADDR	R1						00			*	100
CMD4	SET_DSR	—						00				000
CMD7	SELECT/DESELECT_CARD	R1b						00		1	*	100
CMD9	SEND_CSD	R2						00				101
CMD10	SEND_CID	R2						00				101
CMD11	READ_DAT_UNTIL_STOP	R1				1		01			*	100
CMD12	STOP_TRANSMISSION	R1b			1			00		1	*	100
CMD13	SEND_STATUS	R1						00			*	100
CMD15	GO_INACTIVE_STATE	—						00				000
CMD16	SET_BLOCKLEN	R1						00			*	100
CMD17	READ_SINGLE_BLOCK	R1		*				01			*	100
CMD18	READ_MULTIPLE_BLOCK	R1	*1				*1	01			*	100
CMD20	WRITE_DAT_UNTIL_STOP	R1				1		10			*	100
CMD23	SET_BLOCK_COUNT	R1						00			*	100
CMD24	WRITE_BLOCK	R1		*				10			*	100
CMD25	WRITE_MULTIPLE_BLOCK	R1	*1				*1	10			*	100
CMD26	PROGRAM_CID	R1						10			*	100
CMD27	PROGRAM_CSD	R1						10			*	100
CMD28	SET_WRITE_PROT	R1b						00		1	*	100
CMD29	CLR_WRITE_PROT	R1b						00		1	*	100
CMD30	SEND_WRITE_PROT	R1						01			*	100
CMD32*	TAG_SECTOR_START	R1						00			*	100
CMD33*	TAG_SECTOR_END	R1						00			*	100
CMD34*	UNTAG_SECTOR	R1						00			*	100
CMD35	TAG_ERASE_GROUP_START	R1						00			*	100

CMD INDEX	Abbreviation	resp	CMDTYR 6	5	4	3	2	1 to 0	RSPTYR 6	5	4	2 to 0
CMD36	TAG_ERASE_GROUP_END	R1						00			*	100
CMD37*	UNTAG_ERASE_GROUP	R1						00			*	100
CMD38	ERASE	R1b						00		1	*	100
CMD39	FAST_IO	R4						00			*	100
CMD40	GO_IRQ_STATE	R5						00			*	100
CMD42	LOCK_UNLOCK	R1b						10		1	*	100
CMD55	APP_CMD	R1						00			*	100
CMD56	GEN_CMD	R1b						*2		1	*	100

Notes:

* of CMD INDEX: These commands are not support by more developed MMC than MMCA ver 3.1.

*[1] of TY2 andTY6 in CMDTYR: When specify the number of blocks in advance, set TY6; set TY2 when the number of blocks is not specified.

* of TY5 bit in CMDTYR: Set to perform multi block transfer using secure MMC.

* of RTY4 in RSPTYR: Set to 1 after checking CRC in the command response other than R2. (CRC of the R2 command response cannot be checked.)

*[2] of CMD56: When reading, write 01; 10 when writing.

Blank: Set 0.

31.3.4 Transfer Byte Number Count Register (TBCR)

TBCR is an 8-bit readable/writable register that specifies the number of bytes to be transferred (block size) for each single block transfer command. TBCR specifies the number of data block bytes not including the start and end bytes and CRC.

The multiblock transfer command corresponds to the number of bytes of each data block. This setting is ignored by the stream transfer command in MMC mode stream.

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	C3	0	R/W	Transfer data block size
2	C2	0	R/W	0000: 1 byte
1	C1	0	R/W	0001: 2 bytes
0	C0	0	R/W	0010: 4 bytes
				0011: 8 bytes
				0100: 16 bytes
				0101: 32 bytes
				0110: 64 bytes
				0111: 128 bytes
				1000: 256 bytes
				1001: 512 bytes
				1010: 1024 bytes
				1011: 2048 bytes
				1100 to 1111: Setting prohibited

31.3.5 Transfer Block Number Counter (TBNCR)

TBNCR sets the number of blocks to be transferred when multiblock transfer is specified by bits TY5 and TY6 in CMDTYR. The contents of TBNCR is decremented for every 1-block transfer completion. When the contents of TBNCR is 0, the command sequence is terminated, and an interrupt is generated.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TBNCR	All 0	R/W	Transfer Block Number Counter [Clearing condition] When the specified number of blocks are transferred and 0 is written to TBNCR.

31.3.6 Command Registers 0 to 5 (CMDR0 to CMDR5)

CMDR are six 8-bit registers. A command is written to CMDR as shown in table 31.3, and a command is transmitted by setting the START bit in CMDSTRT to 1.

Table 31.3 CMDR Configuration

Register	Contents	Operation
CMDR0	Start bit, Host bit, and command index	Command index writing Sets the Start bit to 0, and the Host bit to 1.
CMDR1 to CMDR4	Command argument	Command argument writing
CMDR5	CRC, End bit	Setting of CRC is unnecessary (automatic calculation) Setting of end bit is unnecessary (end bit is set to 1)

- CMDR0

Bit	Bit Name	Initial Value	R/W	Description
7	Start	0	R/W	Start bit (This bit should be set to 0)
6	Host	0	R/W	Transmission bit (This bit should be set to 1)
5 to 0	INDEX	All 0	R/W	Command indexes

- CMDR1 to CMDR4

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	CMDR1 to CMDR4	All 0	R/W	Command arguments See specifications for the MMC.

- CMDR5

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	CRC	All 0	—	This bit is unnecessary to be set, and is always read as 0.
0	End	0	—	This bit is unnecessary to be set, and is always read as 0.

31.3.7 Response Registers 0 to 16 and D (RSPR0 to RSPR16 and RSPRD)

RSPR0 to RSPR16 are seventeen 8-bit command response registers. RSPRD is a 5-bit data register.

The number of command response bytes differs according to the command. The number of command response bytes can be specified by the response type register (RSPTYR) in the MMCIF. The command response is shifted-in from the bit 0 in RSPR16, and shifted to the number of command response bytes × 8 bits. Table 31.4 summarizes the correspondence between the number of command response bytes and valid RSPR.

Table 31.4 Correspondence between Command Response Byte Number and RSPR

	MMC Mode Response	
RSPR Registers	**6 Bytes (R1, R1b, R3, R4, R5)**	**17 Bytes (R2)**
RSPR0	—	1st byte
RSPR1	—	2nd byte
RSPR2	—	3rd byte
RSPR3	—	4th byte
RSPR4	—	5th byte
RSPR5	—	6th byte
RSPR6	—	7th byte
RSPR7	—	8th byte
RSPR8	—	9th byte
RSPR9	—	10th byte
RSPR10	—	11th byte
RSPR11	1st byte	12th byte
RSPR12	2nd byte	13th byte
RSPR13	3rd byte	14th byte
RSPR14	4th byte	15th byte
RSPR15	5th byte	16th byte
RSPR16	6th byte	17th byte

RSPR0 to RSPR16 are simple shift registers. A command response that has been shifted in is not automatically cleared, and it is continuously shifted until it is shifted out from the bit 7 in RSPR0. To clear unnecessary bytes to H'00, write arbitrary values to each RSPR.

- RSPR0 to RSPR16

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	RSPR	All 0	R/W	These bits are cleared to H'00 by writing an arbitrary value. RSPR0 to RSPR16 are continuous 17-byte shift registers. Command response is stored.

- RSPRD

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	RSPRD	All 0	R/W	These bits are cleared to H'00 by writing an arbitrary value. Command response is stored.

31.3.8 Command Start Register (CMDSTRT)

CMDSTRT triggers the start of command transmission, representing the start of a command sequence. The following operations should be completed before the command sequence starts.

Command transmission:

- Analysis of prior command response, clearing the command response register write if necessary
- Analyze/transfer receive data of prior command if necessary
- Preparation of transmission data of the next command if necessary
- Setting of CMDTYR, RSPTYR, TBCR, and TBNCR
  CMDR0 to CMDR4, CMDTYR, RSPTYR, TBCR, and TBNCR should not be changed until command transmission has ended (the CWRE flag in CSTR has been set to 1).
- Setting of CMDR0 to CMDR4

The command sequences are controlled by the sequencers in each MMCIF side and MMC side. Normally, these operate synchronously, however, these may become temporarily unsynchronized when an error occurs or when a command is aborted. Take care to set the CMDOFF bit in OPCR, to issue the CMD12 command, and to process an error in MMC mode. A new command sequence should be started after confirming that the command sequences on both the MMCIF and MMC sides have ended.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	START	0	R/W	Starts command transmission when 1 is written. This bit is cleared by hardware.

31.3.9 Operation Control Register (OPCR)

OPCR controls command operation abort, and suspends or continues data transfer.

Bit	Bit Name	Initial Value	R/W	Description
7	CMDOFF	0	R/W	Command Off Aborts all command operations (MMCIF command sequence) when 1 is written after a command is transmitted. This bit is then cleared by hardware. Write enable period: from command transmission completion to command sequence end Writes 0: Operation is not affected. Writes 1: Command sequence is forcibly aborted.
6	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.
5	RD_CONTI	0	R/W	Read Continue After 1 is written, this bit is cleared by hardware when MMCIF resumes reading data. Resumes read data reception when the sequence is halted according to FIFO full or termination of block reading in multiblock read. Write enable period: while MCCLK for read data reception is halted Writes 0: Operation is not affected. Writes 1: Resumes MCCLK output and read data reception.

Bit	Bit Name	Initial Value	R/W	Description
4	DATAEN	0	R/W	Data Enable Starts write data transmission by a command with write data. Resumes write data transmission when the transfer clock is halted according to FIFO empty or one block writing is terminated in multiblock write. Write enable period: (1) after reception of a command response with write data, (2) while transfer clock is halted according to FIFO empty, (3) when one block writing in multiblock write is terminated Writes 0: Operation is not affected. Writes 1: Starts or resumes transfer clock output and write data transmission.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

For write data transmission, the contents of the command response and data response should be analyzed, and then transmission should be triggered. In addition, write data transmission should be temporarily halted according to FIFO full/empty, and it should be resumed when the preparation has been completed.

For multiblock transfer, the transfer clock output should be temporarily halted for every block break to select either to continue to the next block or to abort the multiblock transfer command by issuing the CMD12 command, and the transfer clock output should be resumed. To continue to the next block, the RD_CONTI and DATAEN bits should be set to 1. To issue the CMD12 command, the CMDOFF bit should be set to 1 to abort the command sequence on the MMCIF side. Setting RD_CONTI or DATAE bit between blocks, can be omitted when auto mode is used in pre-define multi block transfer.

31.3.10 Command Timeout Control Register (CTOCR)

CTOCR specifies a cycle to generate a timeout for the command response.

When receiving the command response, CTOUTC continues counting the transfer clock, and enters the command timeout error state when the number of transfer clock reaches the number specified in CTOCR. When the CTERIE bit in INTCR1 is set to 1, the CTERI flag in INTSTR1 is set. To perform command timeout error handling, the command sequence should be aborted by setting the CMDOFF bit to 1, and then the CTERI flag should be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
0	CTSEL0	1	R/W	0: 128 transfer clocks from command transmission completion to response reception completion 1: 256 transfer clocks from command transmission completion to response reception completion

Note: When R2 response (17-byte command response) is required, a timeout is generated during response reception if the CTSEL0 bit is set to 0. Therefore, set the CTSEL0 bit to 1.

31.3.11 Data Timeout Register (DTOUTR)

DTOUTR specifies a cycle to generate a data timeout. The 16-bit counter (DTOUTC) and a prescaler count the peripheral clock to monitor the data timeout. The prescaler always counts the peripheral clock, and outputs a count pulse for every 10000 peripheral clocks. The initial value of DTOUTC is 0, and DTOUTC starts counting the prescaler output from the start of the command sequence. DTOUTC is cleared when the command sequence has ended, or when the command sequence has been aborted by setting the CMDOFF bit to 1, after which DTOUTC stops counting the prescaler output.

When the command sequence does not end, DTOUTC continues counting the prescaler output, and enters the data timeout error states when the number of prescaler output reaches the number specified in DTOUTR. When the DTERIE bit in INTCR1 is set to 1, the DTERI flag in INTSTR1 is set. To perform data timeout error handling, the command sequence should be aborted by setting the CMDOFF bit to 1, and then the DTERI flag should be cleared to prevent extra-interrupt generation.

For a command with data busy state, as the command sequence is terminated before entering the data busy state, data timeout cannot be monitored. Timeout in the data busy state should be monitored by firmware. When DTOUTR is set to 0, a data timeout is generated immediately after the command sequence has started.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	DTOUTR	All 1	R/W	Data timeout time/10000 Data timeout time is determined by peripheral clock cycle × DTOUTR setting value × 10000.

31.3.12 Card Status Register (CSTR)

CSTR indicates the MMCIF status during command sequence execution.

Bit	Bit Name	Initial Value	R/W	Description
7	BUSY	0	R	Command Busy Indicates command execution state. When the CMDOFF bit in OPCR is set to 1, this bit is cleared to 0 because the MMCIF command sequence is aborted. 0: Idle state waiting for a command, or data busy state 1: Command sequence execution in progress
6	FIFO_ FULL	0	R	FIFO Full When read data is received, this bit is set to 1 after FIFO has been full. This bit is cleared to 0 when RD_CONTI is set to 1 or command sequence is ended. 0: The FIFO is empty 1: The FIFO is full
5	FIFO_ EMPTY	0	R	FIFO Empty When write data is transmitted, this bit is set to 1 after FIFO has been empty. This bit is cleared to 0 when DATAEN is set to 1 or command sequence is ended. 0: The FIFO includes data 1: The FIFO is empty
4	CWRE	0	R	Command Register Write Enable Indicates whether the CMDR command is being transmitted or has been transmitted. 0: The CMDR command has been transmitted, or the START bit in CMDSTRT has not been set yet, so the new command can be written. 1: The CMDR command is waiting for transmission or is being transmitted. If the new command is written, a malfunction will result.

Bit	Bit Name	Initial Value	R/W	Description
3	DTBUSY	0	R	Data Busy Indicates command execution status. Indicates that the MMC is in the busy state during or after the command sequence of a command without data transfer, which includes the busy state in the response, or of a command with write data has been ended. 0: Idle state waiting for a command, or command sequence execution in progress. 1: MMC indicates data busy after command sequence ends.
2	DTBUSY_TU	Undefined	R	Data Busy Pin State Monitors level of the DAT pin and DO pin. This bit is monitored to confirm whether the card is in busy state by deselecting the card in busy state, and then selecting the card, again. 0: Indicates that the card is in busy state. 1: Idle state waiting for command.
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	REQ	0	R	Interrupt Request Indicates whether an interrupt is requested. An interrupt request is the logical sum of the INTSTR0, INTSTR1, and INTSTR2 flags. The INTSTR0, INTSTR1, and INTSTR2 flags are set by the enable bits in INTCR0, INTCR1, and INTCR2. 0: No interrupts requested 1: An interrupt is requested

31.3.13 Interrupt Control Registers 0 and 1 (INTCR0 and INTCR1)

INTCR enable or disable each flag set of INTSTR0 and INTSTR1 and interrupts.

- INTCR0

Bit	Bit Name	Initial Value	R/W	Description
7	FEIE	0	R/W	FIFO Empty Flag Enable 0: Disables FIFO empty flag setting 1: Enables FIFO empty flag setting
6	FFIE	0	R/W	FIFO Full Flag Enable 0: Disables FIFO full flag setting 1: Enables FIFO full flag setting
5	DRPIE	0	R/W	Data Response End Flag Enable 0: Disables data response end flag setting 1: Enables data response end flag setting
4	DTIE	0	R/W	Data Transfer End Flag Enable 0: Disables data transfer end flag setting 1: Enables data transfer end flag setting
3	CRPIE	0	R/W	Command Response End Flag Enable 0: Disables command response end flag setting 1: Enables command response end flag setting
2	CMDIE	0	R/W	Command Output End Flag Enable 0: Disables command output end flag setting 1: Enables command output end flag setting
1	DBSYIE	0	R/W	Data Busy End Flag Enable 0: Disables data busy end flag setting 1: Enables data busy end flag setting
0	BTIE	0	R/W	Multiblock Transfer End Flag Enable 0: Disables multiblock transfer end flag setting 1: Enables multiblock transfer end flag setting

- INTCR1

Bit	Bit Name	Initial Value	R/W	Description
7	INTRQ2E	0	R/W	MMCI0 Interrupt Enable 0: Disables int_err_n interrupt 1: Enables int_err_n interrupt
6	INTRQ1E	0	R/W	MMCI1 Interrupt Enable 0: Disables int_tran_n interrupt 1: Enables int_tran_n interrupt
5	INTRQ0E	0	R/W	MMCI2 Interrupt Enable 0: Disables int_fstat_n interrupt 1: Enables int_fstat_n interrupt
4	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.
3	WRERIE	0	R/W	Write Error Flag Enable 0: Disables write error flag setting 1: Enables write error flag setting
2	CRCERIE	0	R/W	CRC Error Flag Enable 0: Disables CRC error flag setting 1: Enables CRC error flag setting
1	DTERIE	0	R/W	Data Timeout Error Flag Enable 0: Disables data timeout error flag setting 1: Enables data timeout error flag setting
0	CTERIE	0	R/W	Command Timeout Error Flag Enable 0: Disables command timeout error flag setting 1: Enables command timeout error flag setting

31.3.14 Interrupt Status Registers 0 and 1 (INTSTR0 and INTSTR1)

INTSTR enable or disable MMCIF interrupts.

- INTSTR0

Bit	Bit Name	Initial Value	R/W	Description
7	FEI	0	R/(W)*	FIFO Empty Flag [Setting condition] When FIFO becomes empty while FEIE = 1 and write data is transmitted. (When the FIFO_EMPTY bit in CSTR is set.) [Clearing condition] Write 0 after reading FEI = 1.
6	FFI	0	R/(W)*	FIFO Full Flag [Setting condition] When FIFO becomes full while FFIE = 1 and read data is received. (When the FIFO_FULL bit in CSTR is set.) [Clearing condition] Write 0 after reading FFI = 1.
5	DRPI	0	R/(W)*	Data Response Flag [Setting condition] When the CRC status is received while DRPIE = 1. [Clearing condition] Write 0 after reading DRPI = 1.
4	DTI	0	R/(W)*	Data Transfer End Flag [Setting condition] When the number of bytes of data transfer specified in TBCR ends while DTIE = 1. [Clearing condition] Write 0 after reading DTI = 1.
3	CRPI	0	R/(W)*	Command Response Reception End Flag [Setting condition] When command response reception ends while CRPIE = 1. [Clearing condition] Write 0 after reading CRPI = 1.

Bit	Bit Name	Initial Value	R/W	Description
2	CMDI	0	R/(W)*	Command Transmit End Flag [Setting condition] When command transmission ends while CMDIE = 1. [Clearing condition] Write 0 after reading CMDI = 1.
1	DBSYI	0	R/(W)*	Data Busy End Flag [Setting condition] When data busy state ends while DBSYIE = 1. [Clearing condition] Write 0 after reading DBSYI = 1.
0	BTI	0	R/(W)*	Multiblock Transfer End Flag [Setting condition] When the number of bytes of data transfer specified by TBCR after TBNCR has been decremented to 0 ends while BTIE = 1. [Clearing condition] Write 0 after reading BTI = 1.

Note: * Cleared by writing 0 after reading 1.

- INTSTR1

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
3	WRERI	0	R/(W)*	Write Error Flag [Setting condition] When a status error for transmit data response (write error) is detected while WREIE = 1. [Clearing condition] Write 0 after reading WREI = 1. Note: When the write error occurs, halt the command sequence by setting the CMDOFF bit to 1.
2	CRCERI	0	R/(W)*	CRC Error Flag [Setting condition] When a CRC error for command response or receive data, and CRC status error for transmission data response are detected while CRCERIE = 1. For any non-R2 command response, CRC is checked when the RTY4 in RSPTYR is set for enabling. For the R2 command response, CRC is not checked; therefore, this flag is not set. [Clearing condition] Write 0 after reading CRCERI = 1. Note: When the CRC error occurs, halt the command sequence by setting the CMDOFF bit to 1.
1	DTERI	0	R/(W)*	Data Timeout Error Flag [Setting condition] When a data timeout error specified in DTOUTR occurs while DTERIE = 1. [Clearing condition] Write 0 after reading DTERI = 1. Note: When the data timeout error occurs, clear the DTERI flag after halting the command sequence by setting the CMDOFF bit to 1.

Bit	Bit Name	Initial Value	R/W	Description
0	CTERI	0	R/(W)*	Command Timeout Error Flag [Setting condition] When a command timeout error specified in TOCR occurs while CTERIE = 1. [Clearing condition] Write 0 after reading CTERI = 1. Note: When the command timeout error occurs, clear the CTERI flag after halting the command sequence by setting the CMDOFF bit to 1.

Note: * Cleared by writing 0 after reading 1.

31.3.15 Transfer Clock Control Register (CLKON)

CLKON controls the transfer clock frequency and clock on/off.

The 33-MHz peripheral clock is needed, and bits CSEL3 to CSEL0 should be set to 0001 for a 16.5-Mbps transfer clock of the MMCIF. At this time, transfer should be performed by sufficiently slow transfer clock in the open drain state.

In the command sequence, do not perform clock on/off or frequency modification.

Bit	Bit Name	Initial Value	R/W	Description
7	CLKON	0	R/W	Clock On 0: Stops the transfer clock output from the CLK/SCLK pin. 1: Outputs the transfer clock from the CLK/SCLK pin.
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CSEL3	0	R/W	Transfer Clock Frequency Select
2	CSEL2	0	R/W	0000: Setting prohibited
1	CSEL1	0	R/W	0001: Uses the 1/2-divided peripheral clock as a transfer clock.
0	CSEL0	0	R/W	0010: Uses the 1/4-divided peripheral clock as a transfer clock. 0011: Uses the 1/8-divided peripheral clock as a transfer clock. 0100: Uses the 1/16-divided peripheral clock as a transfer clock. 0101: Uses the 1/32-divided peripheral clock as a transfer clock. 0110: Uses the 1/64-divided peripheral clock as a transfer clock. 0111: Uses the 1/128-divided peripheral clock as a transfer clock. 1000: Uses the 1/256-divided peripheral clock as a transfer clock. 1001 to 1111: Setting prohibited

Note: The maximum operating frequency of the peripheral clock is 33.34 MHz.

31.3.16 VDD/Open-Drain Control Register (VDCNT)

VDCNT can use $\overline{\text{MMC\_ODMOD}}$ signal to control open drain. The MMC_VDDON signal output can be used to control the MMC power supply (VDD) on/off.

Bit	Bit Name	Initial Value	R/W	Description
7	VDDON	0	R/W	Specifies MMC_VDDON signal to be used as a MMC power supply (VDD) control signal. 0: MMC_VDDON is low signal output 1: MMC_VDDON is high signal output
6	ODMOD	0	R/W	Specifies $\overline{\text{MMC\_ODMOD}}$ signal to be used to control CMD output open drain in MMC mode. 0: $\overline{\text{MMC\_ODMOD}}$ signal is low signal output 1: $\overline{\text{MMC\_ODMOD}}$ signal is high signal output
5 to 0	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.

31.3.17 Data Register (DR)

DR is a register for reading/writing FIFO data.

Word/byte access is enabled to addresses of this register.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0 (7 to 0)	DR	Undefined	R/W	Register for reading/writing FIFO data. Word/byte access is enabled. However, byte access is disabled to address 2n+1.

31.3.18 FIFO Pointer Clear Register (FIFOCLR)

The FIFO write/read pointer is cleared by writing any value to FIFOCLR.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	FIFOCLR	—	W	The FIFO pointer is cleared by writing any value to this register.

31.3.19 DMA Control Register (DMACR)

DMACR sets DMA request signal output. DMAEN enables/disables a DMA request signal. The DMA request signal is output by a value that has been set to bits SET2 to SET0.

Set this register before executing a multiblock transfer command (CMD18 or CMD25). Auto mode cannot be used for open-ended multiblock transfer.

Bit	Bit Name	Initial Value	R/W	Description
7	DMAEN	0	R/W	0: Disables output of DMA request signal. (Initial value) 1: Enables output of DMA request signal.
6	AUTO	0	R/W	This bit is set when the pre-defined multiblock transfer using DMA transfer is performed in auto mode. 0: Auto mode is not used. 1: Auto mode is used.
5 to 3	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	SET2	0	R/W	Sets DMA request signal assert condition.
1	SET1	0	R/W	000: Not output (Initial value)
0	SET0	0	R/W	001: FIFO remained data is 1/4 or less of FIFO capacity.
				010: FIFO remained data is 1/2 or less of FIFO capacity.
				011: FIFO remained data is 3/4 or less of FIFO capacity.
				100: FIFO remained data is at least 1 byte.
				101: FIFO remained data is 1/4 or more of FIFO capacity.
				110: FIFO remained data is 1/2 or more of FIFO capacity.
				111: FIFO remained data is 3/4 or more of FIFO capacity.

31.3.20 Interrupt Control Register 2 (INTCR2)

The INTCR2 enables or disables an interrupt.

Bit	Bit Name	Initial Value	R/W	Description
7	INTRQ3E	0	R/W	MMCI3 Interrupt Enable
				0: Interrupt disabled
				1: Interrupt enabled
6 to 1	—	All 0	—	Reserved
				These bits are always read as 0. The write value should always be 0.
0	FRDYIE	0	R/W	FIFO Preparation End Flag Enable
				0: Disables FIFO preparation end flag set
				1: Enables FIFO preparation end flag set

31.3.21 Interrupt Status Register 2 (INTSTR2)

The INTSTR2 controls the MMCIF interrupt output.

If setting condition is satisfied, FRDYI is set even though it has been cleared. Disable flag setting by FRDYIE in INTCR2 before clearing FRDYI.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
1	FRDY_TU	1	R	When FRDYI setting condition is satisfied. Read value 0: When FIFO remained data is less than data set as assert condition by DMACR 1: When FIFO remained data is other than data set as assert condition by DMACR
0	FRDYIE	0	R/(W)*	FIFO Preparation End Flag Enable [Setting condition] When FIFO remained data is less than data set as assert condition by DMACR while FRDYIE = 1 and the DMAEN bit is set. [Clearing condition] Write 0 after reading FRDYI = 1.

Note: * Cleared by writing 0 after reading 1.

31.4 Operation

The MultiMediaCard is an external storage media that can be easily disconnected. The MMCIF controls data transfer with the MultiMediaCard, and operates in MMC mode.

Insert the MMC and turn on the power supply. Then operate the MMCIF by applying transfer clocks after setting an appropriate transfer clock frequency. The MMC_VDDON signal and $\overline{\text{MMC\_ODMOD}}$ signal can be used for MMC power control and open drain mode control, respectively.

The series of operations from command sending, command response reception, data transmission/reception, and data response reception is called the command sequence. The command sequence starts from sending a command by setting the START bit in CMDSTRT to 1, and ends when all necessary data transmission/reception and response reception has been completed. The MMC supports the data busy state in which only specific command is accepted to program/erase the flash memory in the MMC during command sequence execution and after command sequence execution has ended. The data busy state is indicated by a 0 output from the MMC side to the DAT pin in MMC mode.

Note: Do not connect or disconnect the MMC during command sequence or data busy.

31.4.1 Operations in MMC Mode

MMC mode is an operating mode in which the transfer clock is output from the MMC_CLK pin, command transmission/response receive occurs via the MMC_CMD pin, and data is transmitted/received via the MMC_DAT pin. In this mode the next command can be issued while data is being transmitted/received. This feature is applied to the multiblock transfer and stream transfer. In this case, the next command is the CMD12 command, which aborts the current command sequence.

In MMC mode, a broadcast command that simultaneously issues a command to multiple MMCs is supported. After the information for the MMC that is inserted by using the broadcast command is acknowledged, a relative address is given to each MMC. One MMC is selected by the relative address, other MMCs are deselected, and various commands are issued to the selected MMC.

Commands in MMC mode is basically classified into three types: broadcast, relative address, and flash memory operation commands. The MMC is operated by issuing a command according to card status.

(1) Operation of Broadcast Commands

The CMD0, CMD1, CMD2, and CMD4 are broadcast commands. The sequence assigning relative addresses to individual MMCs consists of these commands and the CMD3 command. In this sequence, the CMD output format is open drain, and the command response is wired-OR. In this case, the transfer clock frequency should be set sufficiently slow.

- All MMCs are initialized to the idle state by the CMD0.
- The operation condition register (OCR) of all MMCs is read via wired-OR, and MMCs that cannot operate are deactivated by the CMD1.

  The deactivated MMCs enter the ready state.
- The card identification (CID) of all MMCs in the ready state is read via wired-OR by the CMD2.

  The individual MMC compares its CID and data on the MMC_CMD, and if different, aborts CID output. A single MMC in which the CID can be entirely output enters the acknowledge state.
- A relative address (RCA) is given to the MMC in the acknowledge state by the CMD3.

  The MMC to which the RCA is given enters the standby state.
- CMD2 and CMD3 are repeated, assigning RCAs to all MMCs in the ready state, entering each into the standby state.

Note: When the R2 response (17-byte command response) is required, the CTSEL0 bit should be set to 1 since a timeout is generated during response reception if the CTSEL0 bit is set to 0.

(2) Operation of Relative Address Commands

The CMD7, CMD9, CMD10, CMD13, CMD15, CMD39, and CMD55 are relative address commands that address the MMC by RCA. The relative address commands are used to read MMC administration information and original information, and to change the specific card status.

The CMD7 sets one addressed MMC to the transfer state, and other MMCs to the standby state. Only the MMC in the transfer state can execute a flash-memory operation command other than the broadcast and relative address commands.

(3) Operation of Commands that do not Require Command Response

Some broadcast commands do not require command response.

Figure 31.2 shows an example of the command sequence for commands that do not require command response.

Figure 31.3 shows the operational flow for commands that do not require command response.

- The settings needed to issue a command are made.
- The START bit in CMDSTRT is set to 1 to start command transmission.
- The end of command sequence is detected by poling the BUSY flag in CSTR or by the command output end interrupt (CMDI).

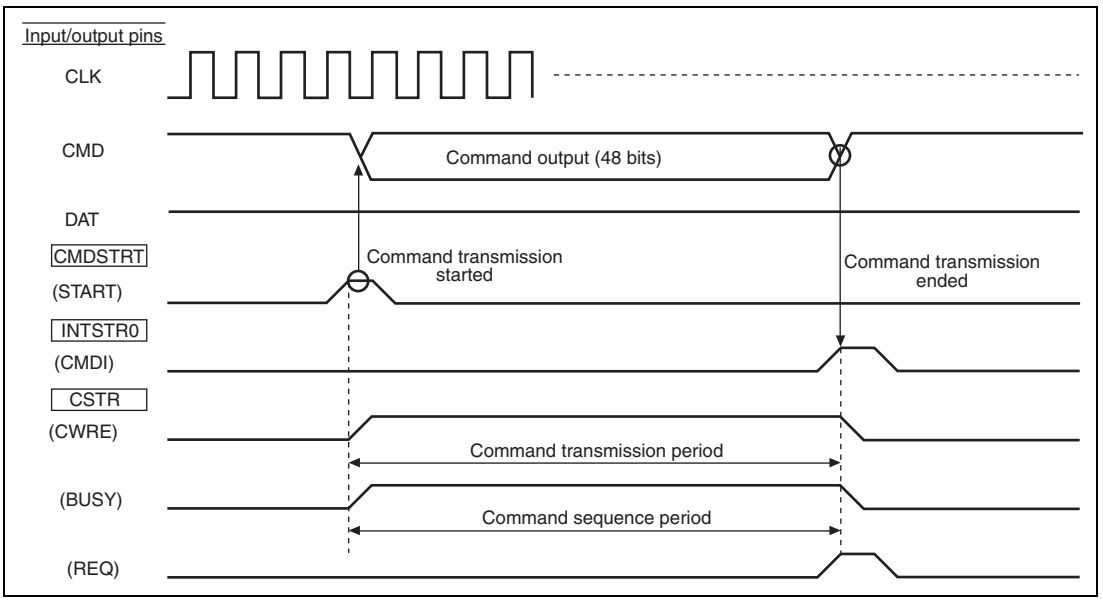


Figure 31.2 Example of Command Sequence for Commands that do not Require Command Response

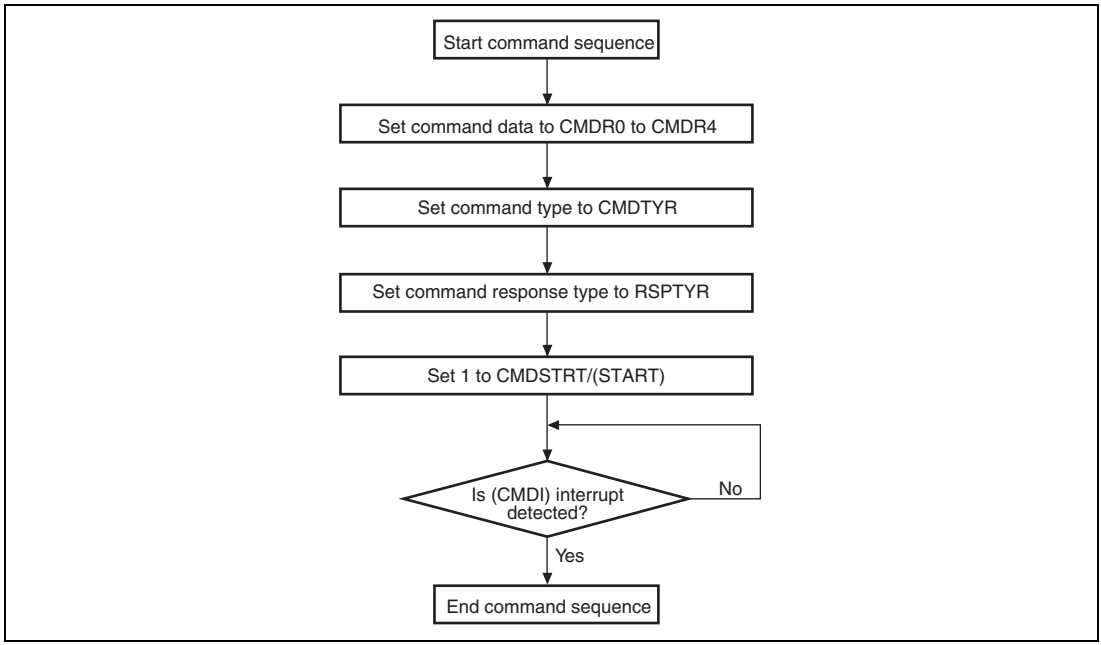


Figure 31.3 Operational Flow for Commands that do not Require Command Response

(4) Operation of Commands without Data Transfer

The broadcast, relative address, and flash memory operation commands include a number of commands that do not include data transfer. Such commands execute the desired data transfer using command arguments and command responses. For a command that is related to time-consuming processing such as flash memory write/erase, the MMC indicates the data busy state via the MMC_DAT.

Figures 31.4 and 31.5 show examples of the command sequence for commands without data transfer.

Figure 31.6 shows the operational flow for commands without data transfer.

- Settings needed to issue a command are made.
- The START bit in CMDSTRT is set to start command transmission.
- Command transmission complete can be confirmed by the command output end interrupt (CMDI).
- A command response is received from the MMC.
- If the MMC does not return the command response, the command response is detected by the command timeout error (CTERI).
- The end of a command sequence is detected by poling the BUSY flag in CSTR or by the command response end interrupt (CRPI).
- Whether the data busy state is entered or not is determined by the DTBUSY bit in CSTR. If the data busy state is entered, the end of the data busy state is detected by the data busy end interrupt (DBSYI).
- When the CRC error (CRCERI) or command timeout error (CTERI) occurs, write 1 to the CMDOFF bit.

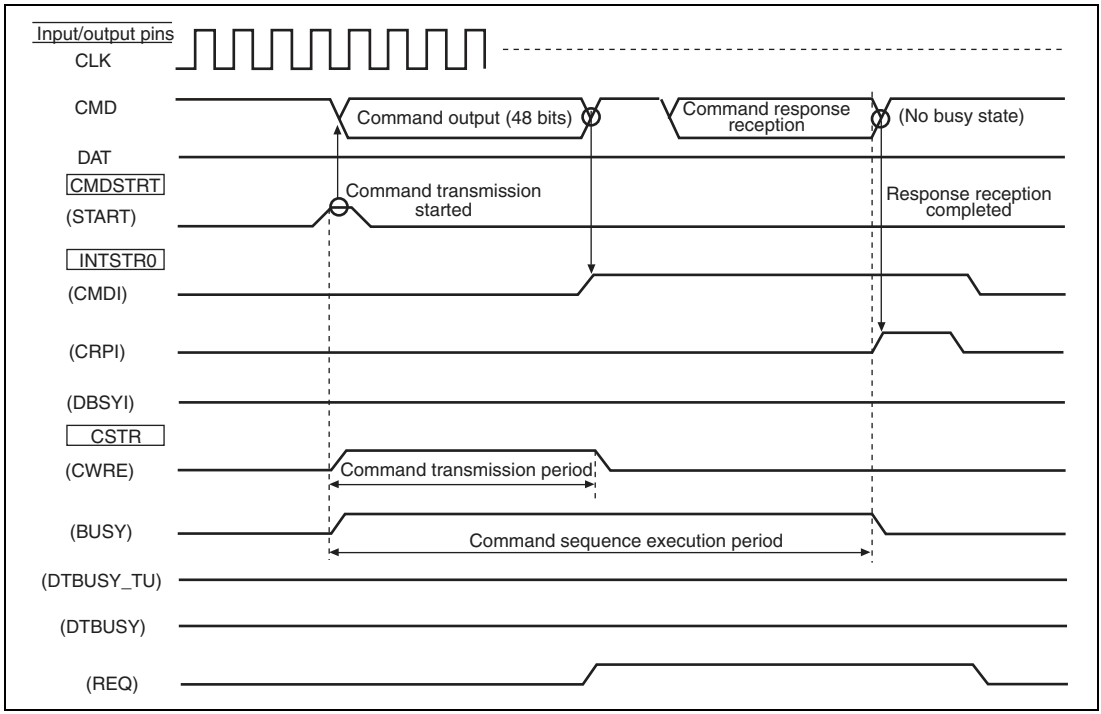


Figure 31.4 Example of Command Sequence for Commands without Data Transfer (No Data Busy State)

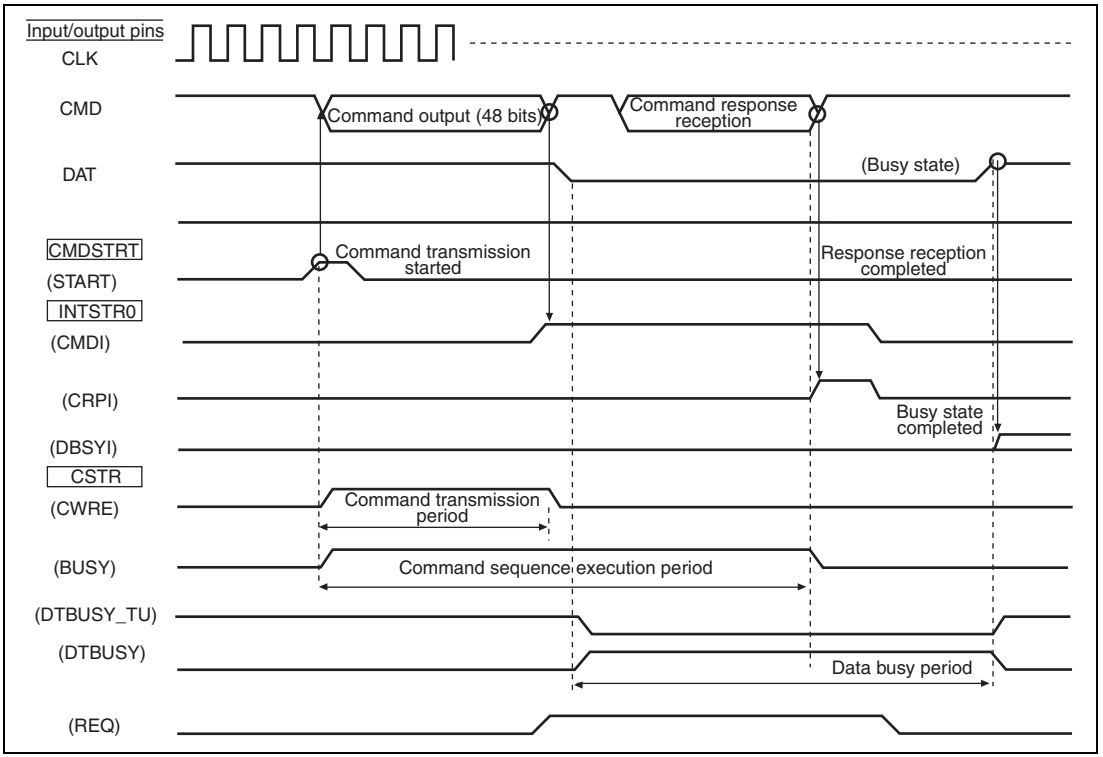
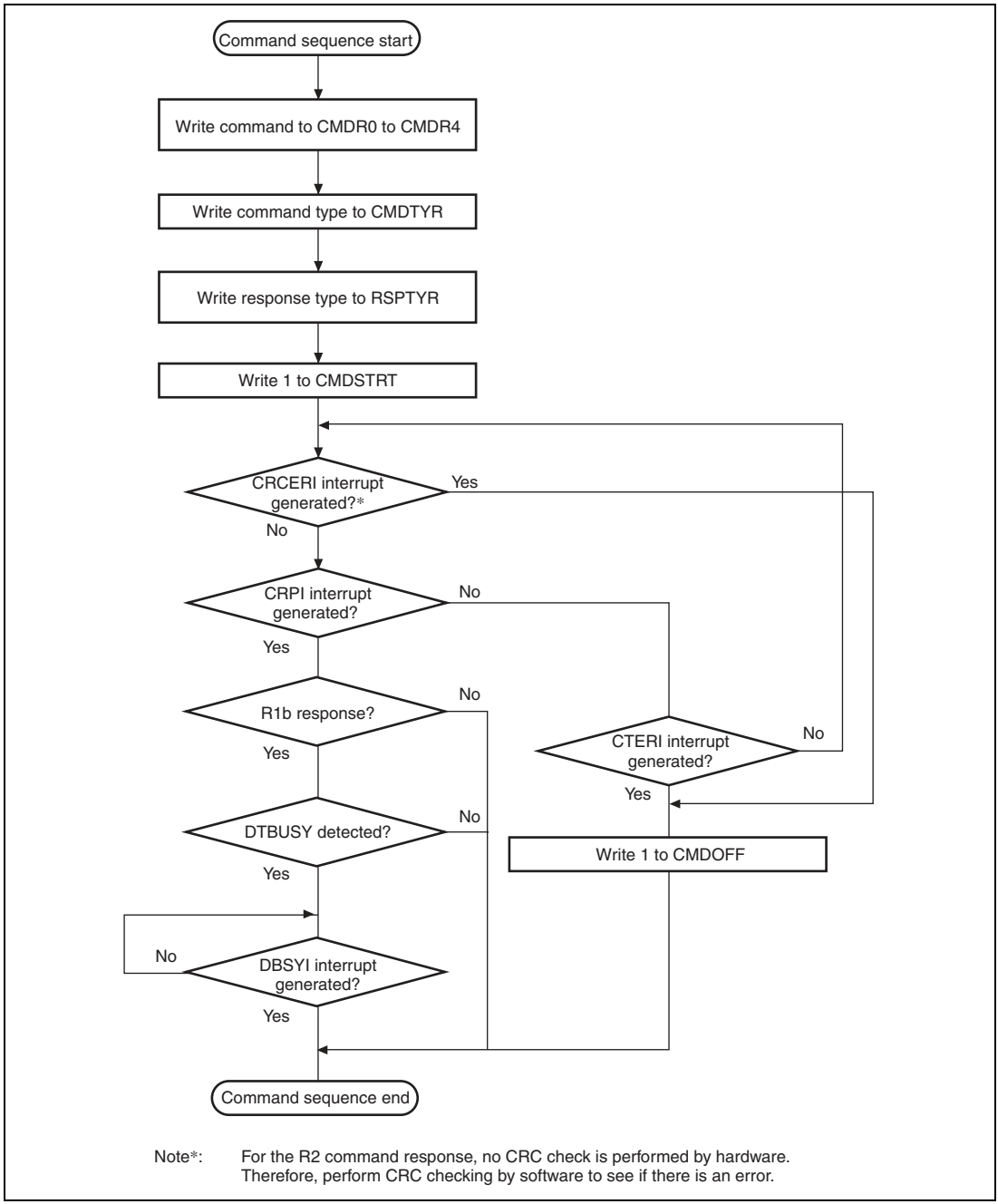


Figure 31.5 Example of Command Sequence for Commands without Data Transfer (with Data Busy State)



Note*: For the R2 command response, no CRC check is performed by hardware. Therefore, perform CRC checking by software to see if there is an error.

Figure 31.6 Operational Flowchart for Commands without Data Transfer

(5) Commands with Read Data

Flash memory operation commands include a number of commands involving read data. Such commands confirm the card status by the command argument and command response, and receive card information and flash memory data from the DAT pin.

For multiblock transfer, there are two methods. One is the open-ended method in which the instruction for continuing/suspending the command sequence is made by suspending the transfer for every block. Another one is the pre-defined method in which the transfer is performed after setting the number of blocks to be transferred.

The command sequence is suspended when FIFO is full between the block transfers. When the command sequence is suspended, data in the receive data FIFO is processed, if necessary, and the command sequence is then continued.

Figures 31.7 to 31.10 show the examples of the command sequence for commands with read data.

Figures 31.11 to 31.14 show the operational flowcharts for commands with read data.

- Settings needed to issue a command are made. FIFO is cleared.
- The START bit in CMDSTRT is set to 1 to start command transmission.
- Command transmission complete can be confirmed by the command output end interrupt (CMDI).
- A command response is received from the MMC.
- If the MMC does not return the command response, the command response is detected by the command timeout error (CTERI).
- Read data from the MMC is received.
- The suspension inter-blocks in multiblock transfer and suspension according to the FIFO full are detected by the data transfer end interrupt (DTI) and FIFO full interrupt (FFI), respectively.

  To continue the command sequence, the RD_CONTI bit in OPCR should be set to 1. To end the command sequence, the CMDOFF bit in OPCR should be set to 1, and the CMD12 should be issued. Note that the CMD12 is not required other than when the sequence is suspended in pre-defined multiblock transfer.
- The end of the command sequence is detected by polling the BUSY flag in CSTR or by the data transfer end flag (DTI) or the multiblock transfer (pre-defined) end flag (BTI).
- When the CRC error (CRCERI) or command timeout error (CTERI) occurs during command response reception, write 1 to the CMDOFF bit.
- When the CRC error (CRCERI) or data timeout error (DTERI) occurs during the read data reception, write 1 to the CMDOFF bit to clear the FIFO.

Note: In multiblock transfer, if you terminate the command sequence (by writing 1 in the CMDOFF bit) before the command response reception is completed (CRPI = 1), the command response cannot be received correctly. To receive a command response, continue the command sequence (by setting the RD_CONTI bit to 1) until the reception of the command response is completed.

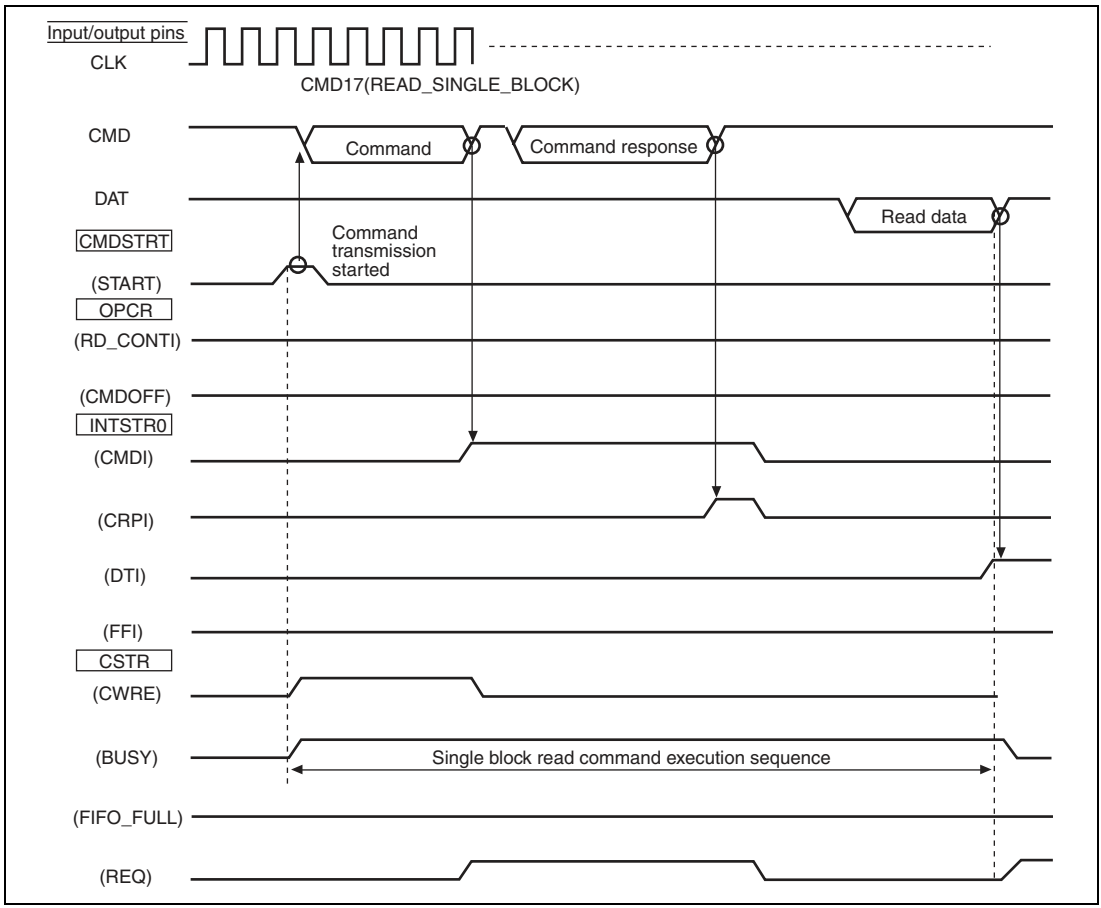


Figure 31.7 Example of Command Sequence for Commands with Read Data (Block Size ≤ FIFO Size)

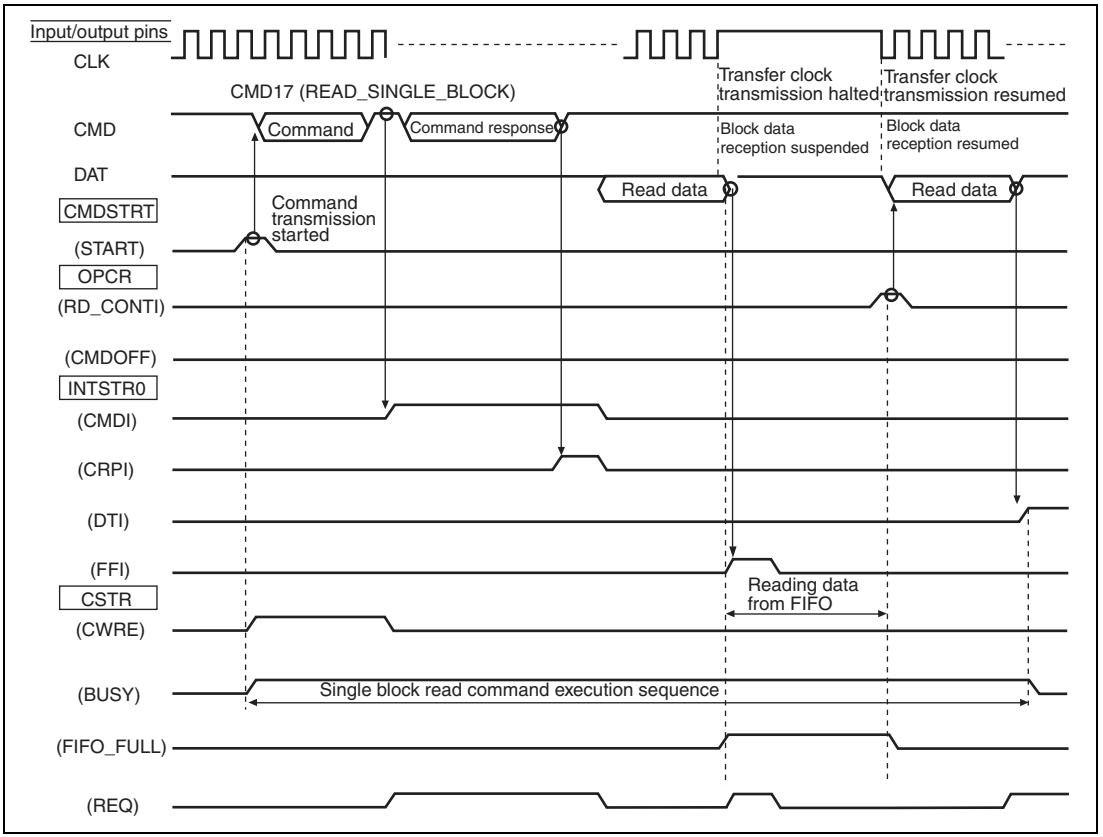


Figure 31.8 Example of Command Sequence for Commands with Read Data (Block Size > FIFO Size)

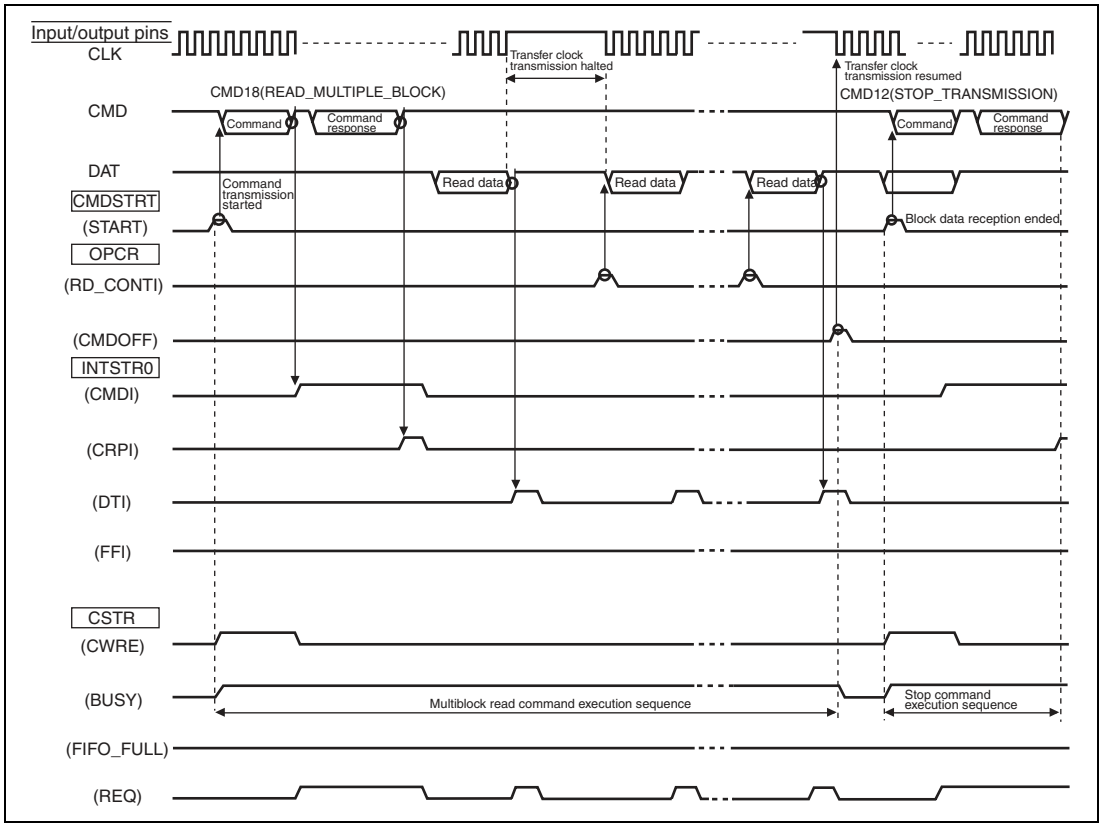


Figure 31.9 Example of Command Sequence for Commands with Read Data (Multiblock Transfer)

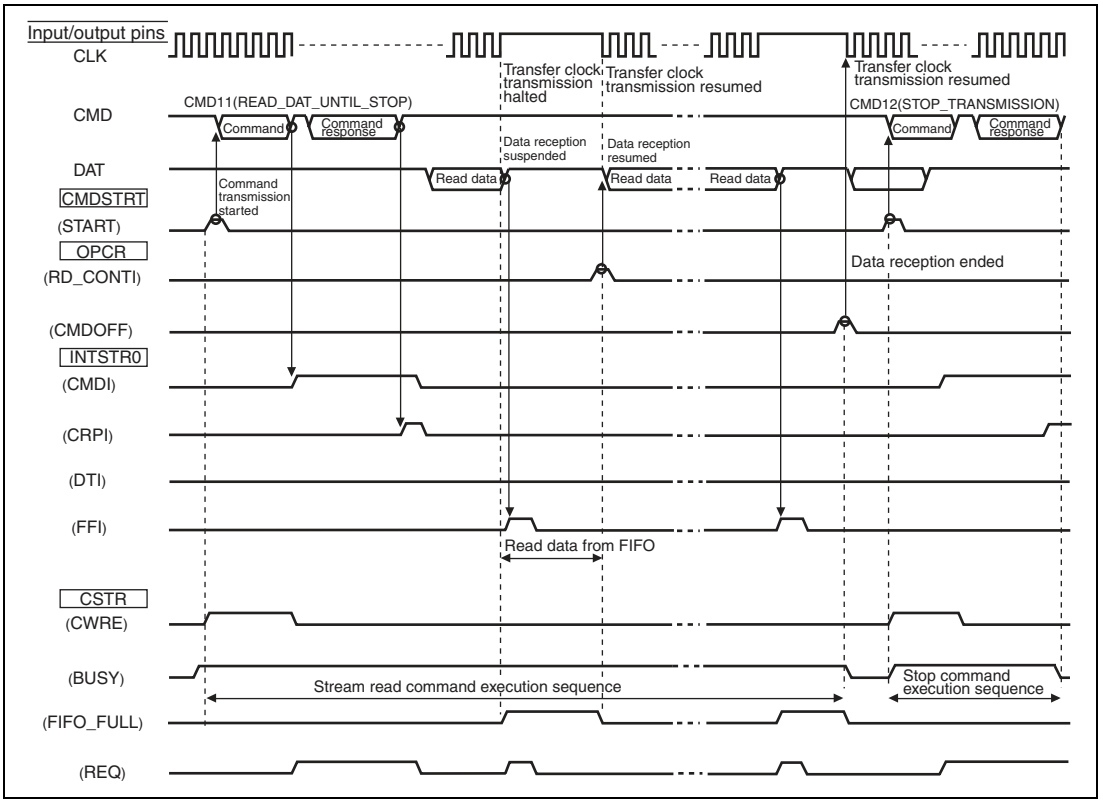


Figure 31.10 Example of Command Sequence for Commands with Read Data (Stream Transfer)

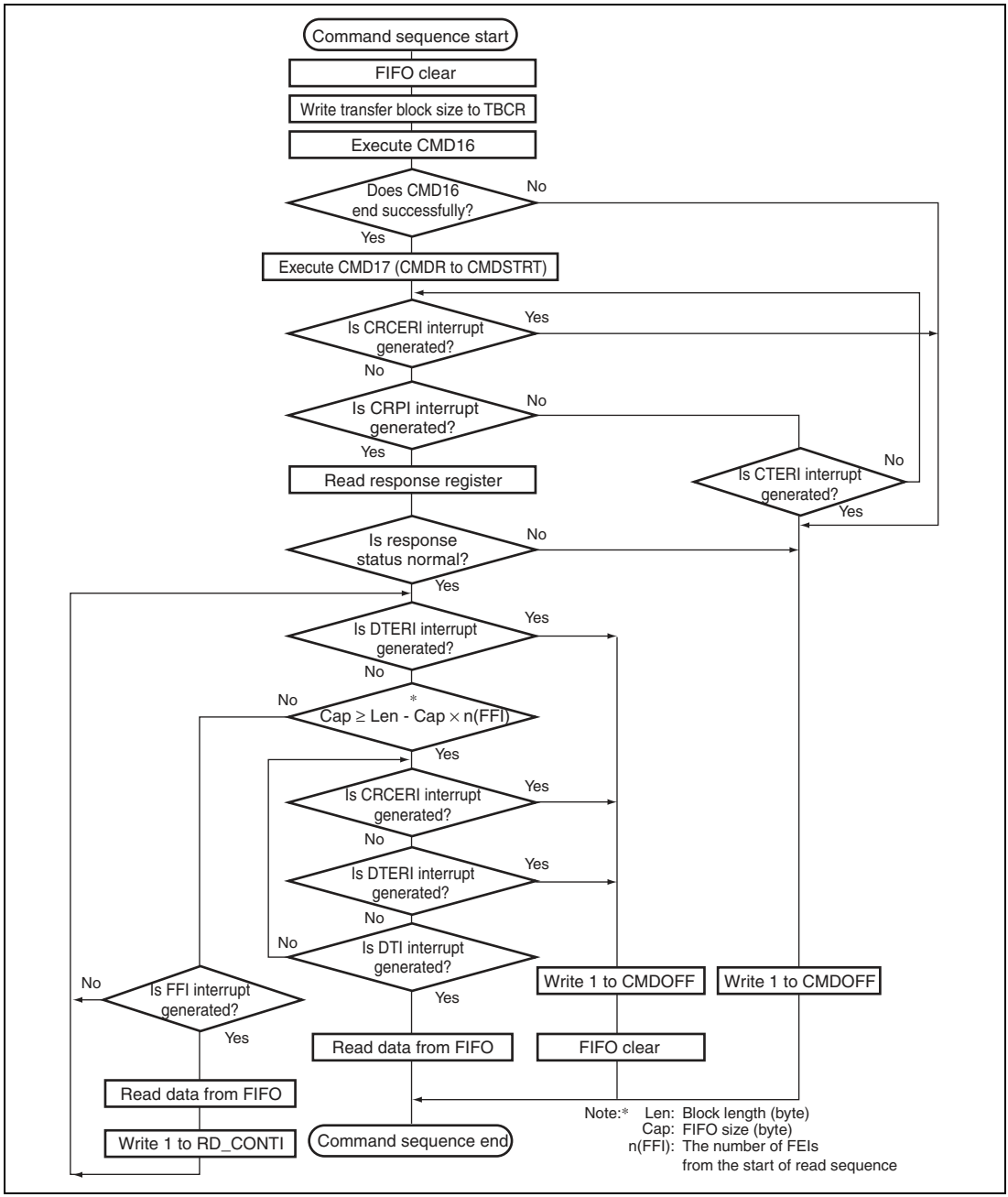


Figure 31.11 Operational Flowchart for Commands with Read Data (Single Block Transfer)

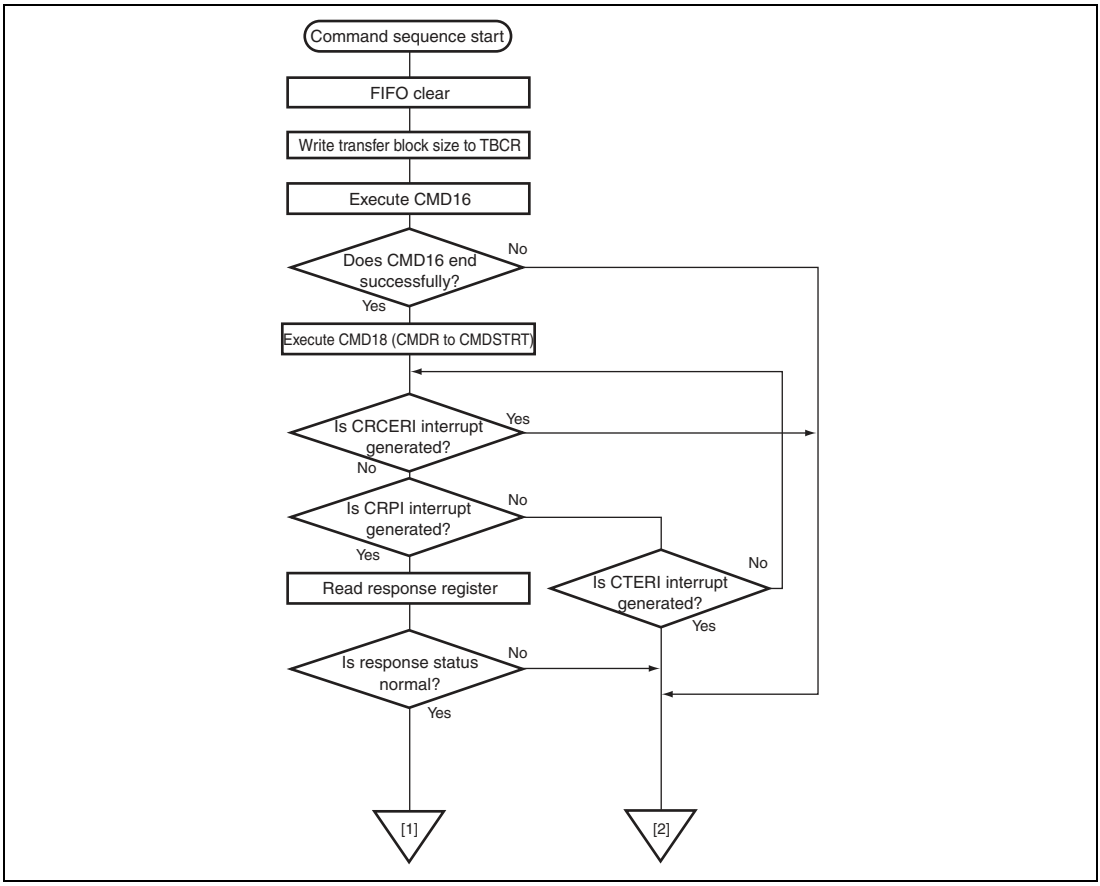


Figure 31.12 Operational Flowchart for Commands with Read Data (Open-ended Multiblock Transfer) (1)

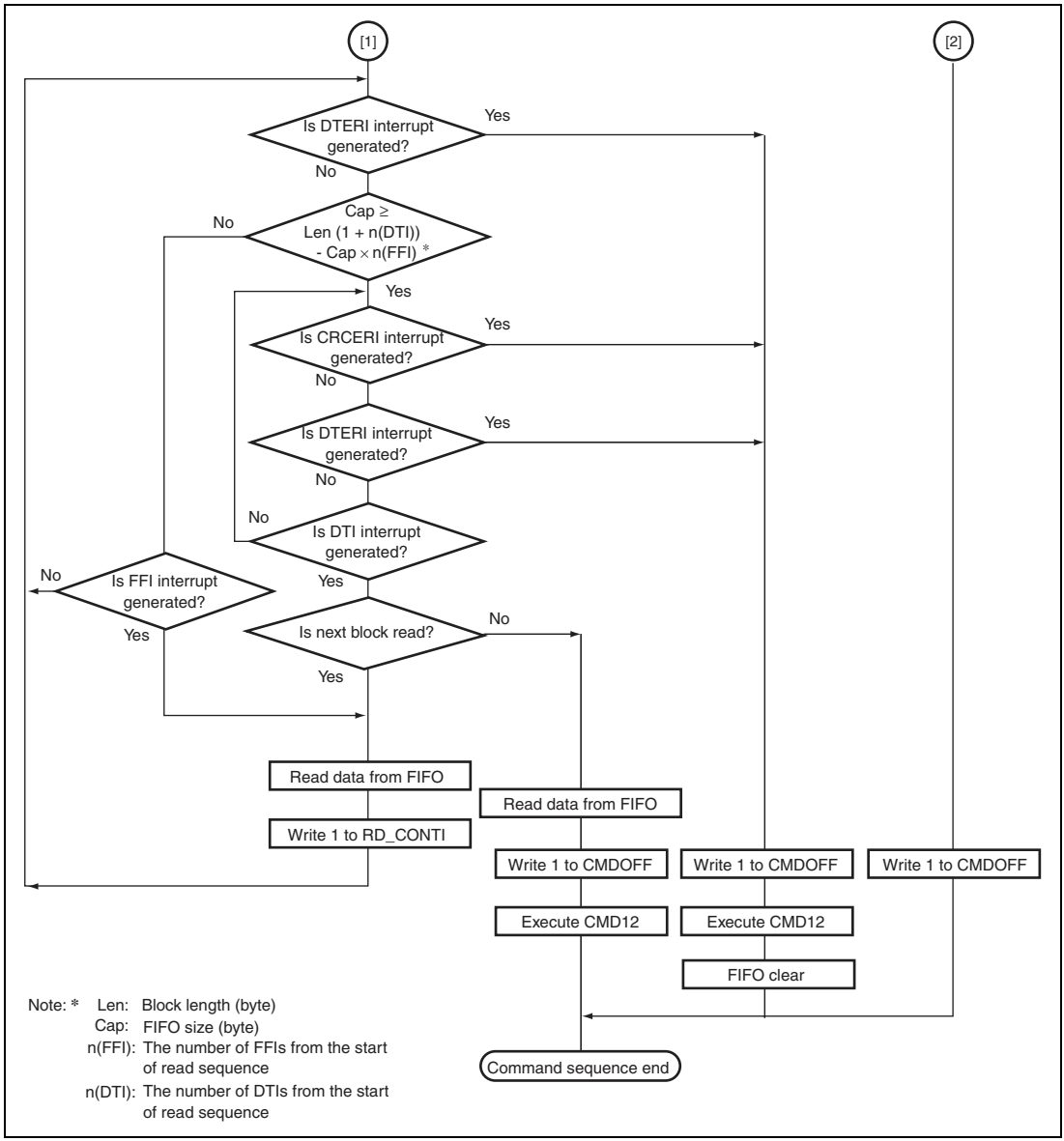


Figure 31.12 Operational Flowchart for Commands with Read Data (Open-ended Multiblock Transfer) (2)

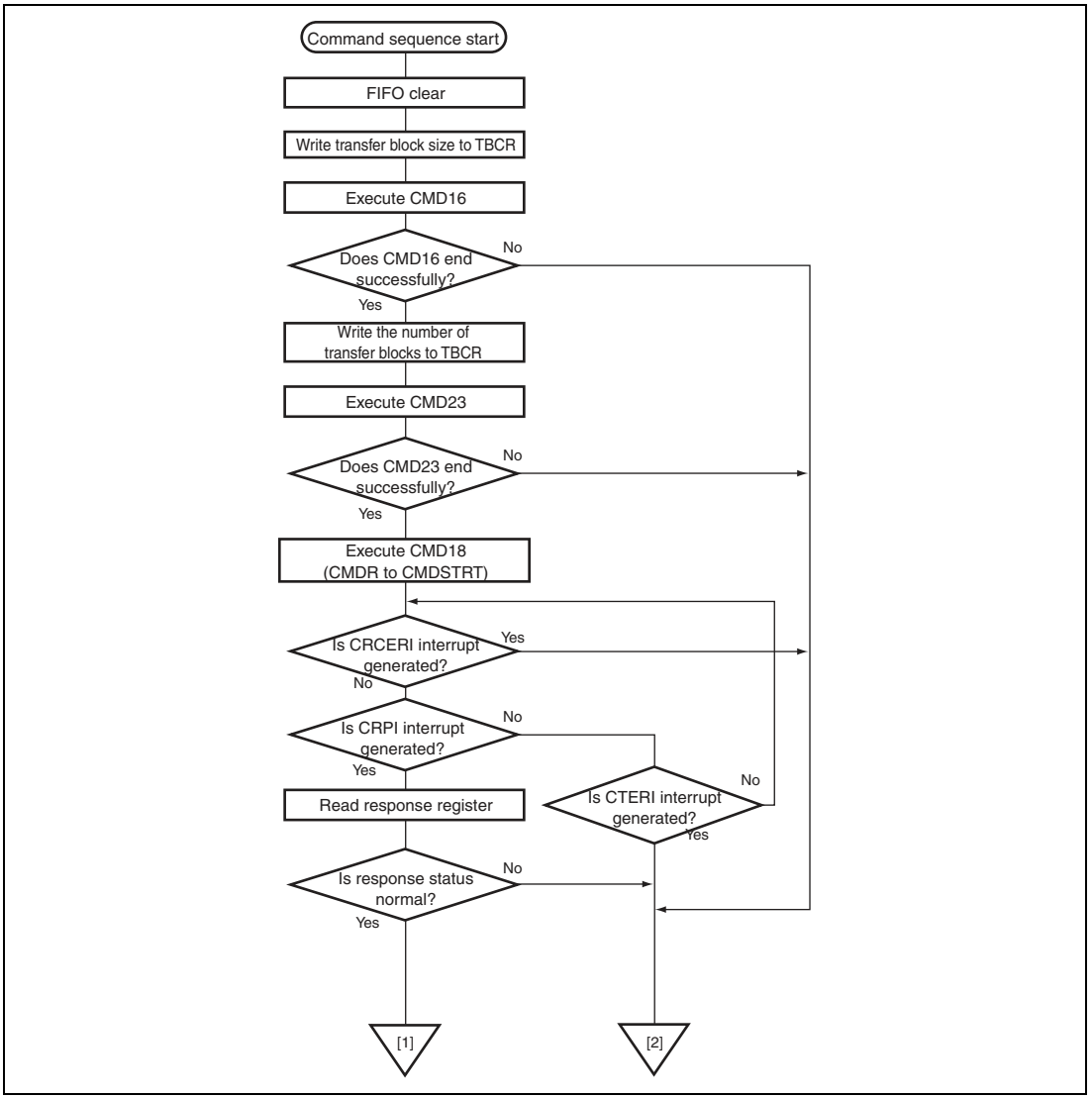


Figure 31.13 Operational Flowchart for Commands with Read Data (Pre-defined Multiblock Transfer) (1)

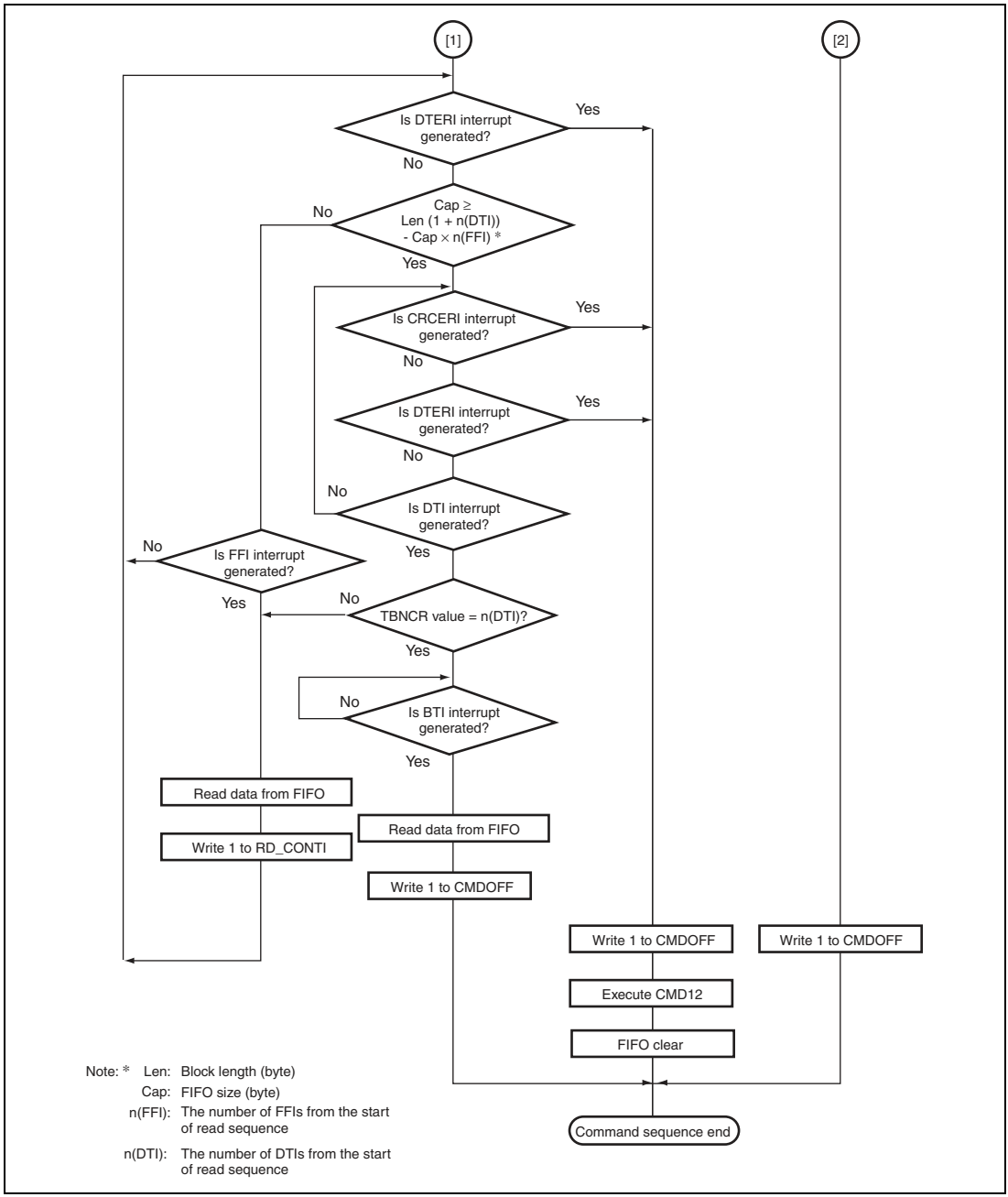


Figure 31.13 Operational Flowchart for Commands with Read Data (Pre-defined Multiblock Transfer) (2)

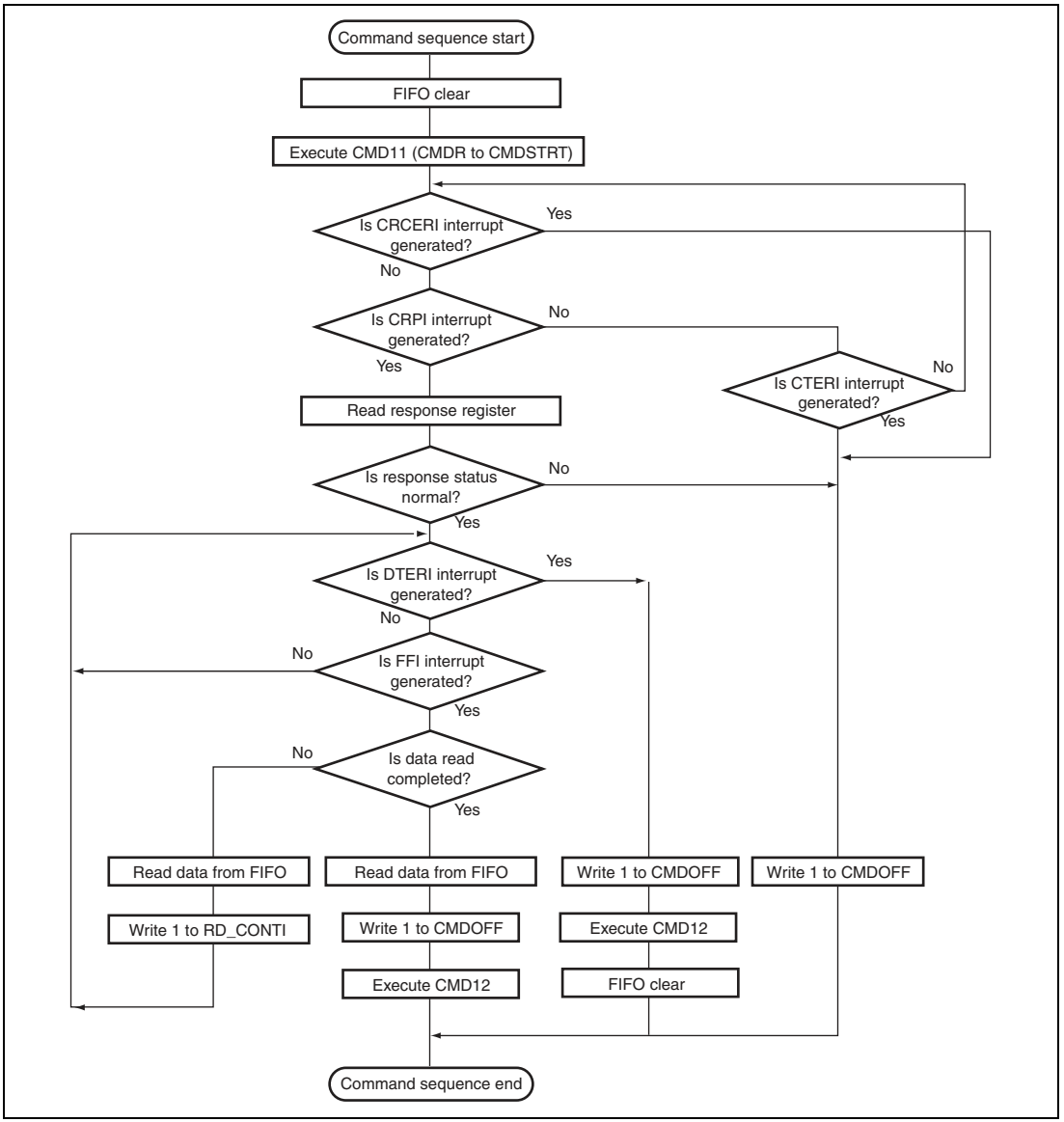


Figure 31.14 Operational Flowchart for Commands with Read Data (Stream Transfer)

(6) Commands with Write Data

Flash memory operation commands include a number of commands involving write data. Such commands confirm the card status by the command argument and command response, and transmit card information and flash memory data from the DAT pin. For a command that is related to time-consuming processing such as flash memory programming, the MMC indicates the data busy state via the DAT pin.

For multiblock transfer, there are two methods. One is the open-ended method in which the instruction for continuing/suspending the command sequence is made by suspending the transfer for every block. Another one is the pre-defined method in which the transfer is performed after setting the number of blocks to be transferred.

The command sequence is suspended when FIFO is full between the block transfers. When the command sequence is suspended, data in the receive data FIFO is processed, if necessary, and the command sequence is then continued.

Figures 31.15 to 31.18 show examples of the command sequence for commands with write data.

Figures 31.19 to 31.22 show the operational flowcharts for commands with write data.

- Settings needed to issue a command are made. The FIFO is cleared.
- The START bit in CMDSTRT is set to 1 to start command transmission.
- A command response is received from the MMC.
- If the MMC does not return the command response, the command response is detected by the command timeout error (CTERI).
- Write data is set to the FIFO.
- The DATAEN bit in OPCR is set to start write data transmission.
- Suspension inter-blocks in multiblock transfer and suspension according to the FIFO empty are detected by the data response end interrupt flag (DRPI) and FIFO empty interrupt flag (FEI), respectively. To continue the command sequence, data should be written to the FIFO, and the DATAEN bit in OPCR should be set to 1. To end the command sequence, the CMDOFF bit in OPCR should be set to 1, and the CMD12 should be issued. Note that the CMD12 is not required other than when the sequence is suspended in pre-defined multiblock transfer.
- The end of the command sequence is detected by poling the BUSY flag in CSTR, data response end flag (DPRI), or multiblock transfer (pre-defined) end flag (BTI).
- In addition, after the end of data transfer (after DRPI is detected), whether the data busy state is entered or not is determined by the DTBUSY bit in CSTR. If the data busy state is entered, cancellation of the data busy state is detected by the data busy end interrupt (DBSYI).

- When the CRC error (CRCERI) or command timeout error (CTERI) occurs during command response reception, write 1 to the CMDOFF bit.
- When the CRC error (CRCERI) or data timeout error (DTERI) occurs during the write data transmission, write 1 to the CMDOFF bit.

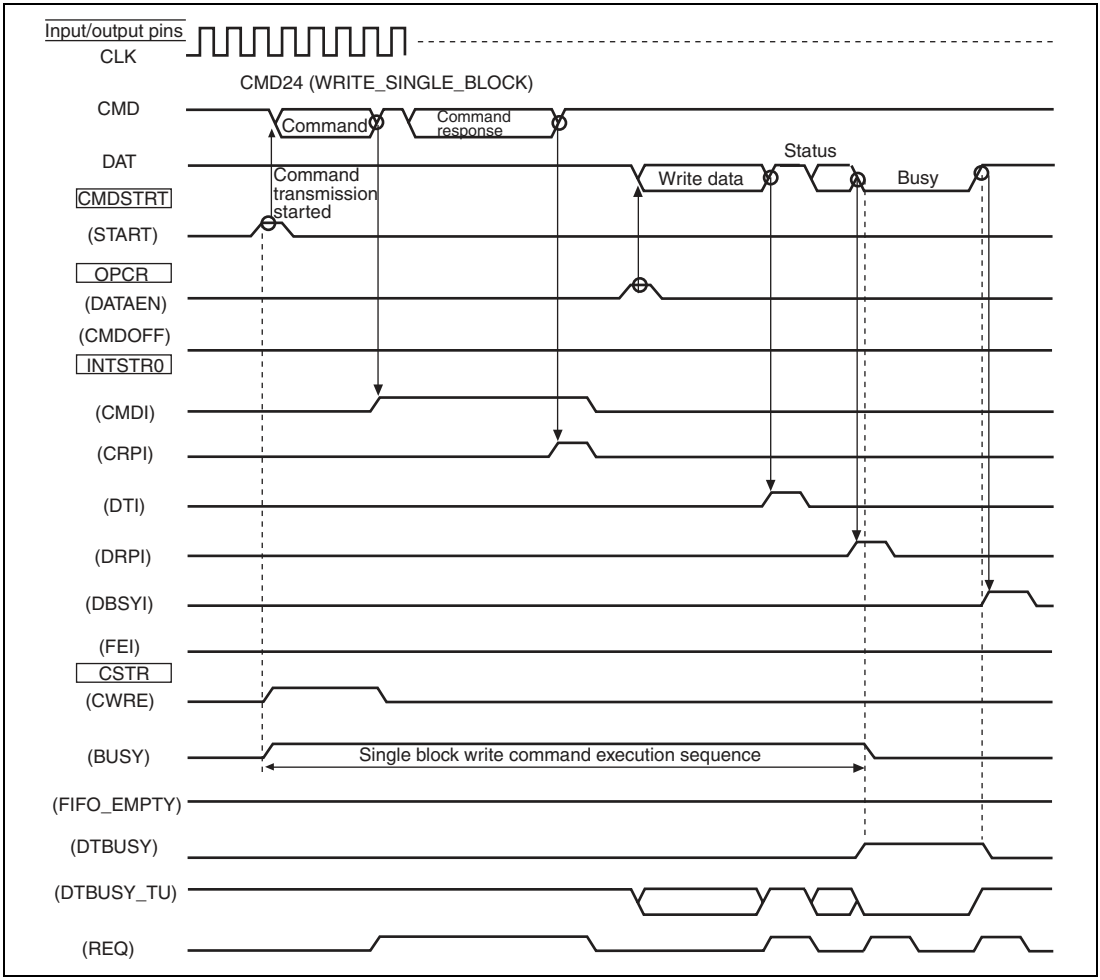


Figure 31.15 Example of Command Sequence for Commands with Write Data (Block Size ≤ FIFO Size)

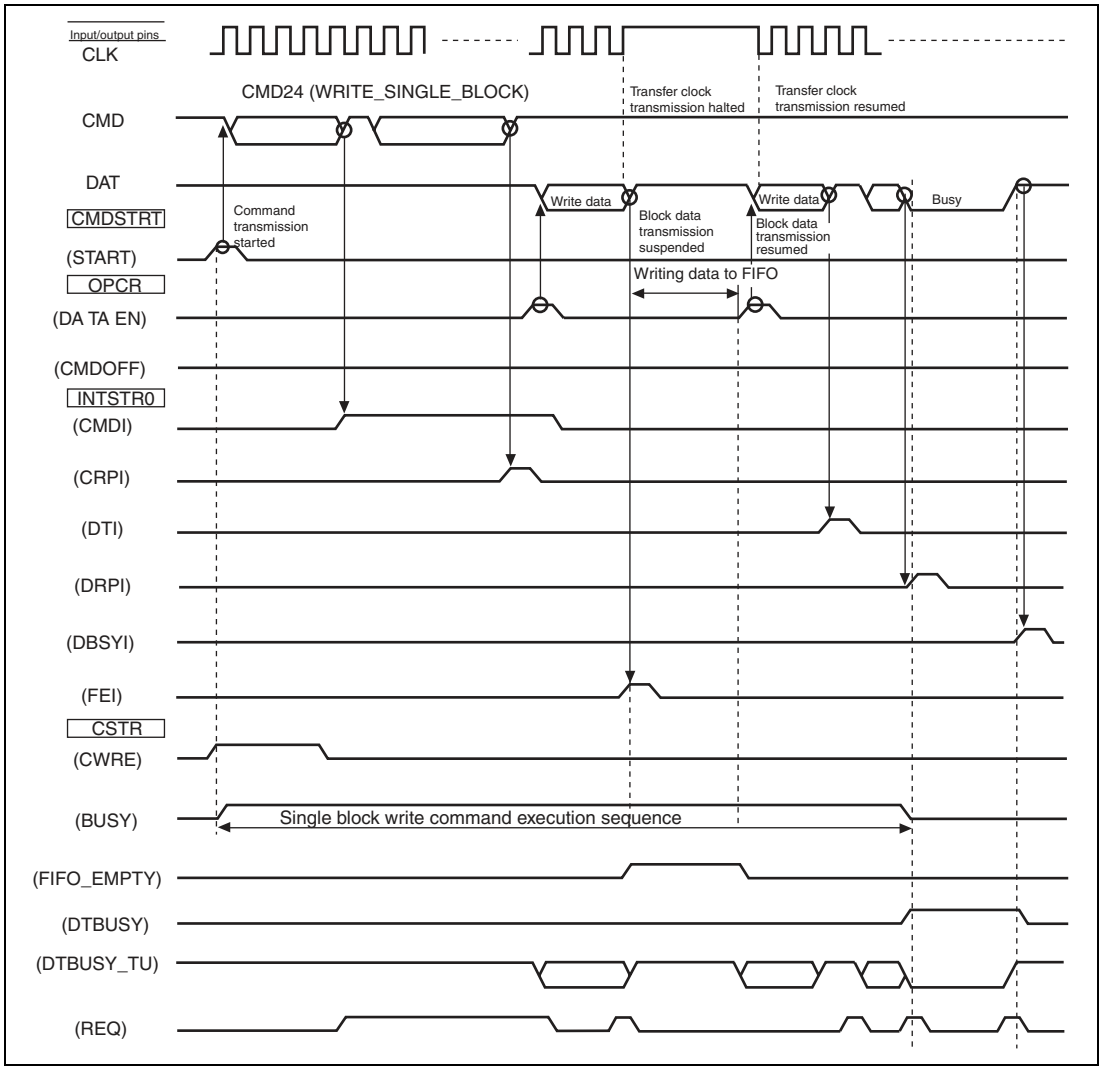


Figure 31.16 Example of Command Sequence for Commands with Write Data (Block Size > FIFO Size)

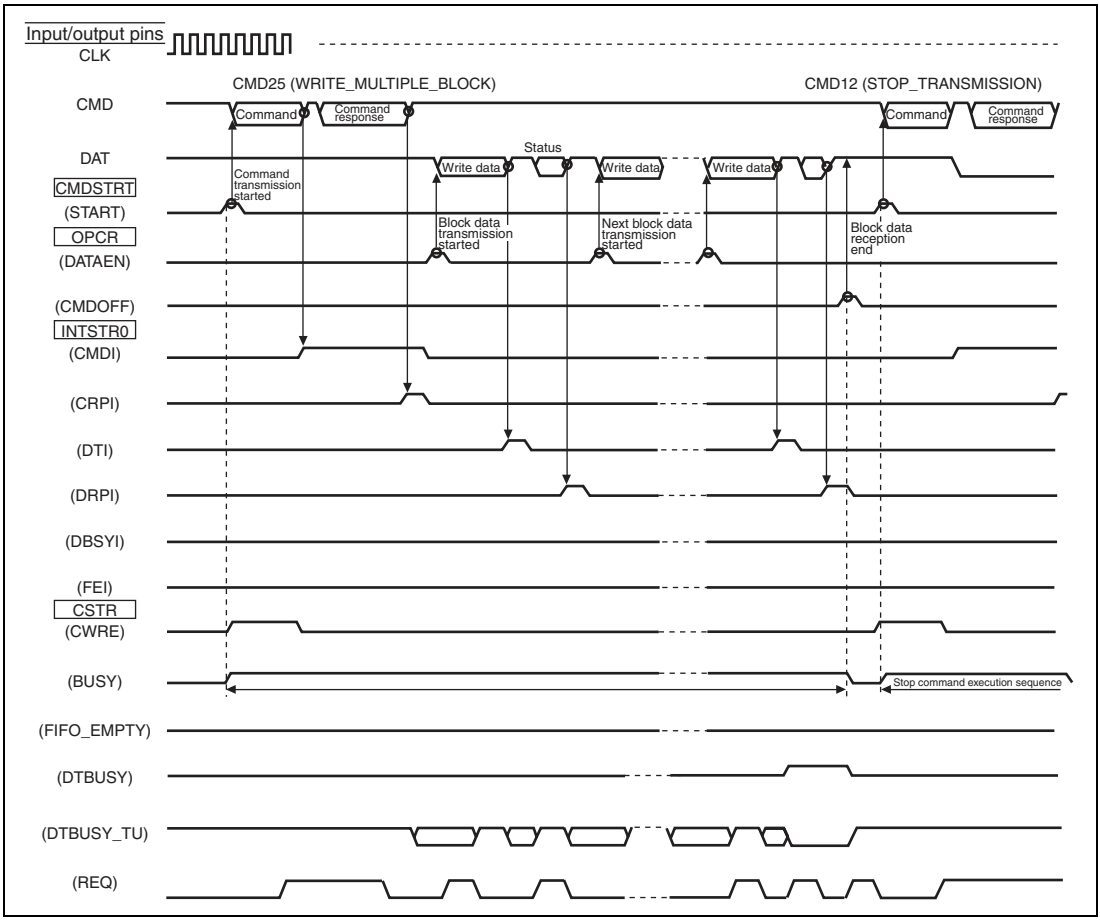


Figure 31.17 Example of Command Sequence for Commands with Write Data (Multiblock Transfer)

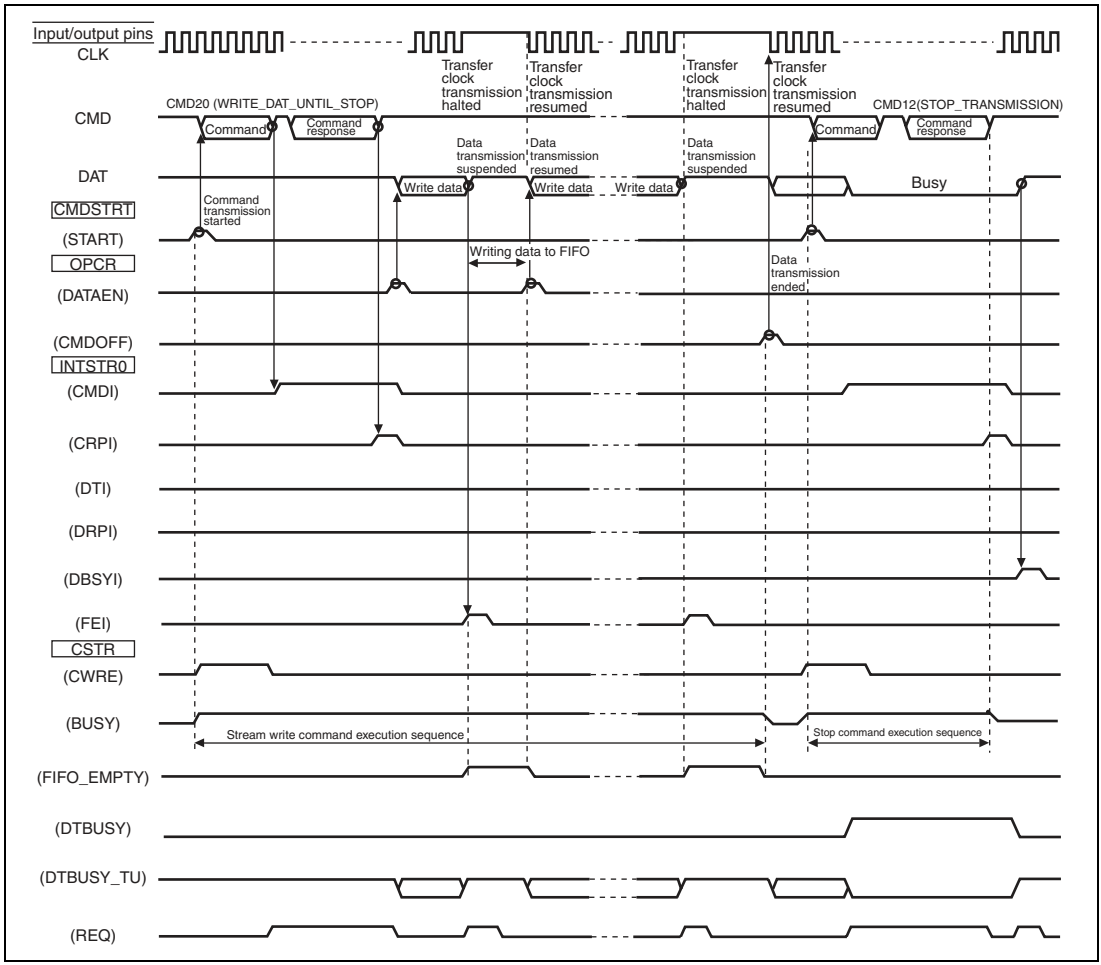


Figure 31.18 Example of Command Sequence for Commands with Write Data (Stream Transfer)

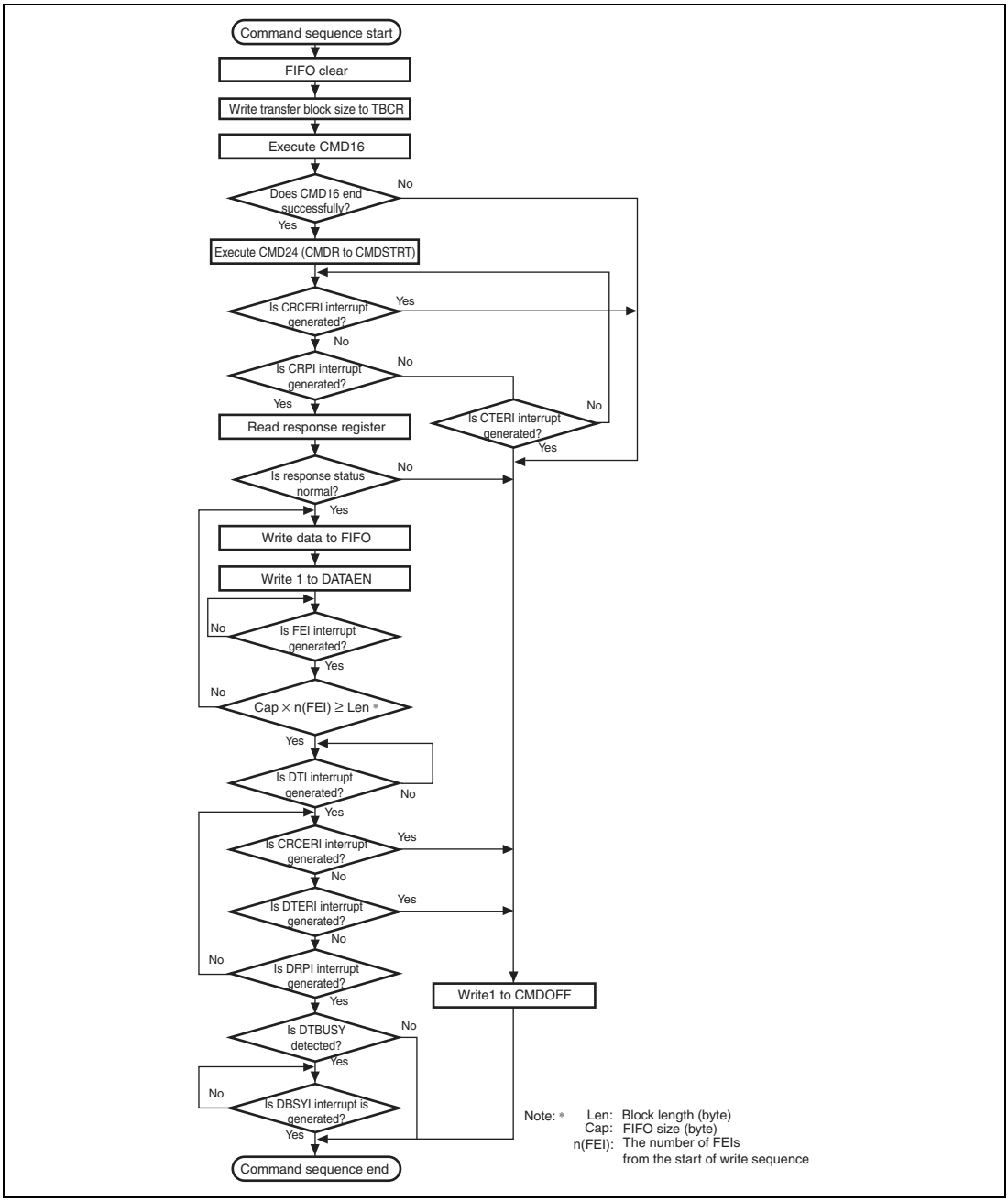


Figure 31.19 Operational Flowchart for Commands with Write Data (Single Block Transfer)

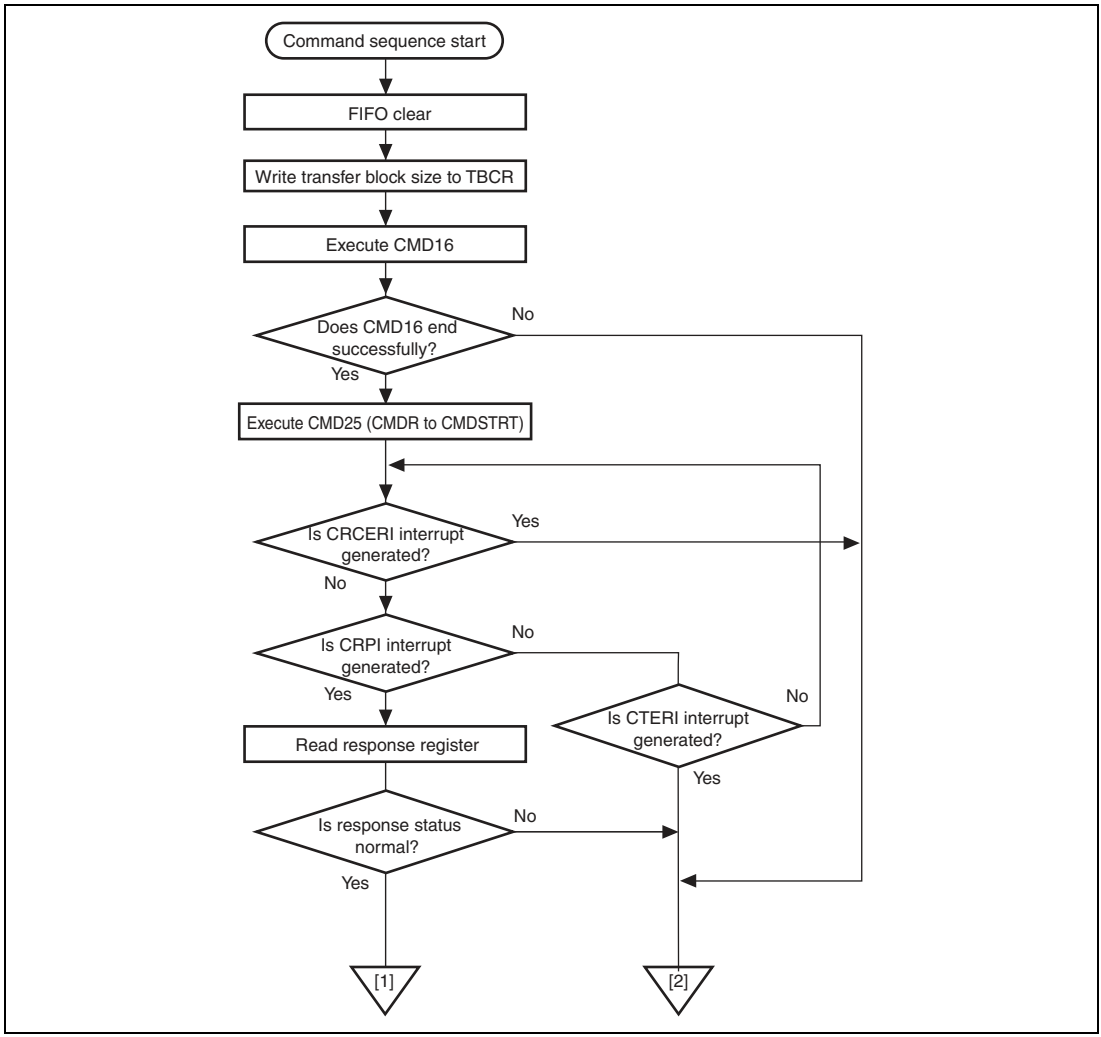


Figure 31.20 Operational Flowchart for Commands with Write Data (Open-ended Multiblock Transfer) (1)

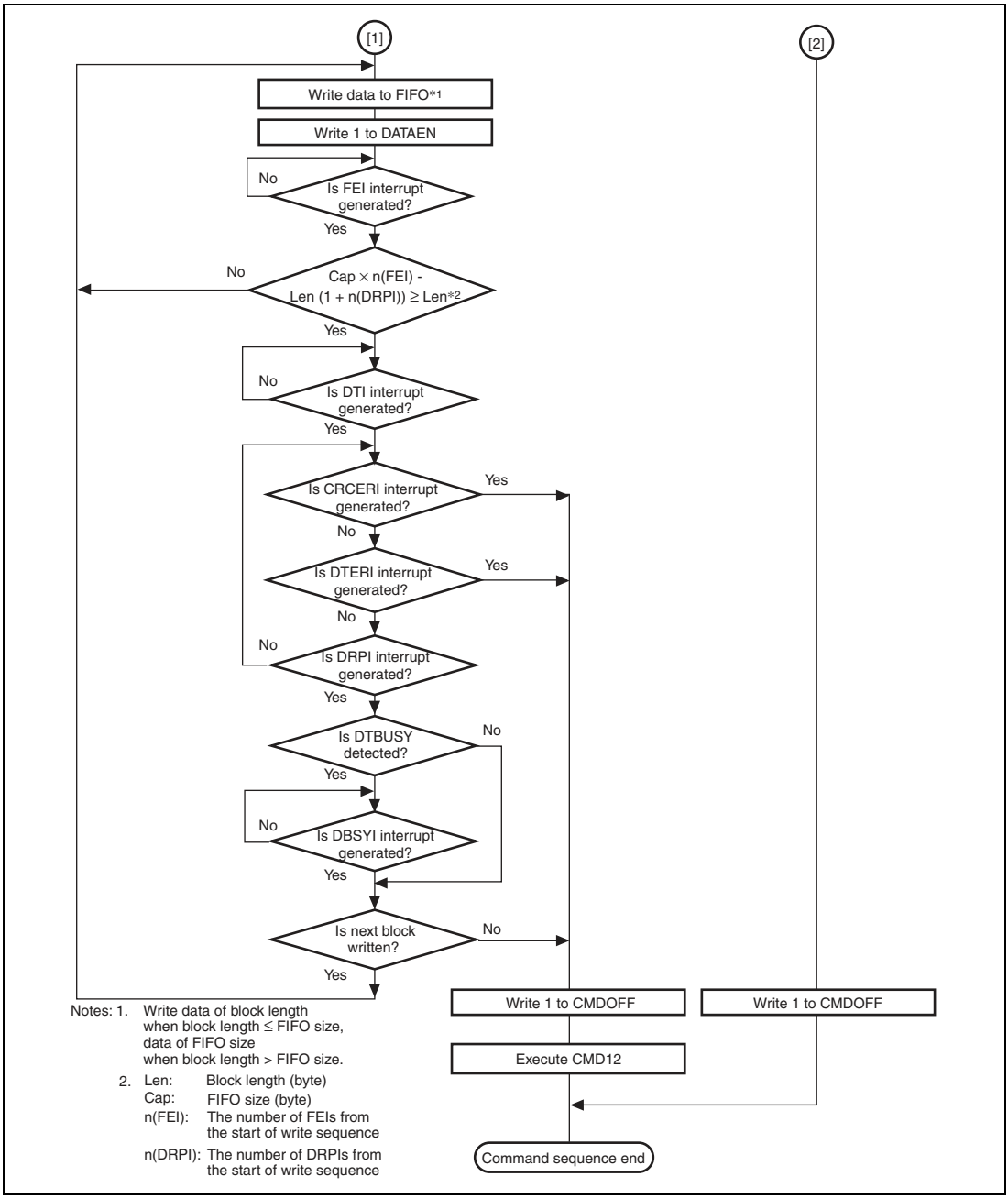


Figure 31.20 Operational Flowchart for Commands with Write Data (Open-ended Multiblock Transfer) (2)

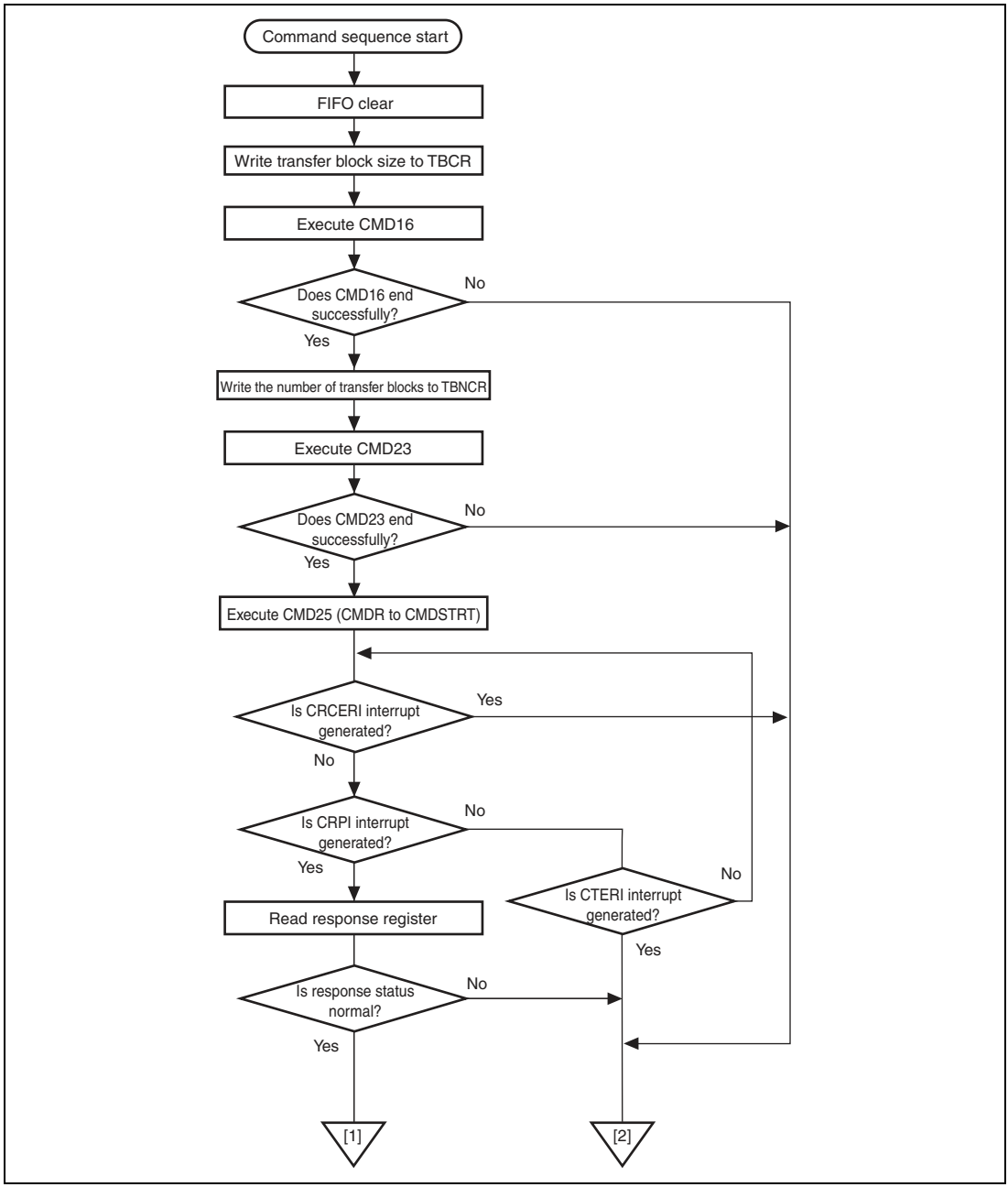


Figure 31.21 Operational Flowchart for Commands with Write Data (Pre-defined Multiblock Transfer) (1)

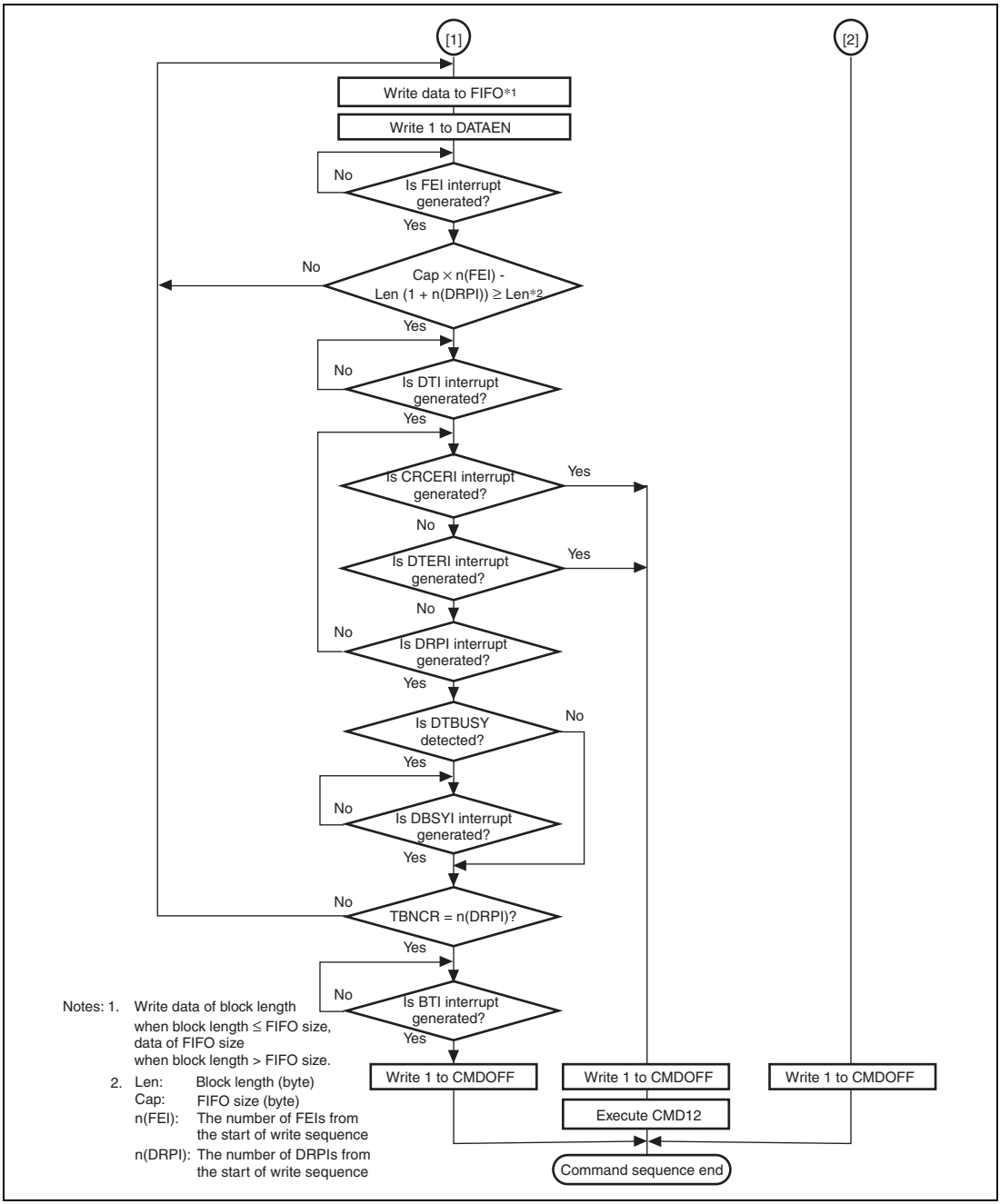


Figure 31.21 Operational Flowchart for Commands with Write Data (Pre-defined Multiblock Transfer) (2)

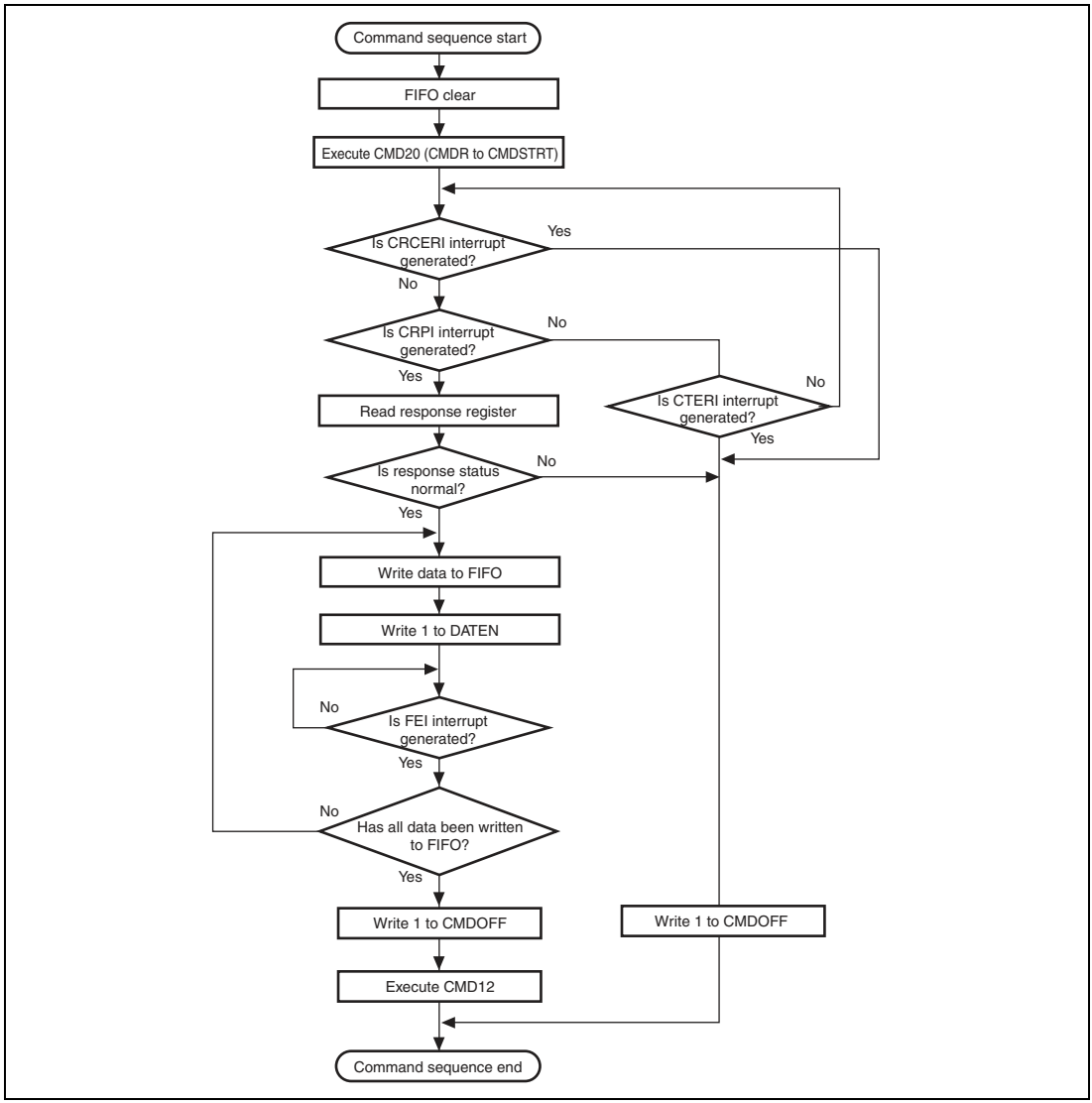


Figure 31.22 Operational Flowchart for Commands with Write Data (Stream Transfer)

31.5 Operations Using DMAC

31.5.1 Operation of Read Sequence

For transfer with DMAC, set MMCIF (DMACR) after setting DMAC. Transmit the read command after setting the DMACR.

Figures 31.23 to 31.26 show the operational flowcharts for read sequence.

- FIFO is cleared and DMACR is set.
- Read command transmission is started.
- Read data is received from the MMC.
- After read sequence, FIFO includes data. If necessary, 100 is written to the SET2 to SET0 bits in DMACR to read every data in FIFO.
- The end of the transfer with the DMAC is confirmed, and 0 is set to the DMAEN bit in DMACR.
- When the CRC error (CRCERI) or the command timeout error (CTERI) occurs during command response reception, write 1 to the CMDOFF bit and set DMACR to H'00.
- When the CRC error (CRCERI) or the data timeout error (DTERI) occurs during read data reception, write 1 to the CMDOFF bit and set DMACR to H'00 to clear the FIFO.

When the DMA is in use, and detected that reading is completed successfully after block transfer in pre-defined multiblock transfer, reading the next block is automatically performed again by setting the AUTO bit in DMACR to 1. Figures 31.27 shows the operational flowchart for pre-defined multi read sequences in MMC mode.

- FIFO is cleared.
- The number of blocks is set to TBNCR.
- DMACR is set.
- Read command transmission is started.
- Command response and read data are received from the MMC.
- If the MMC does not return the command response, it detected by the command timeout error flag (CTERI).
- Command sequence end is detected by polling the BUSY flag in CSTR or the multiblock transfer (pre-defined) end flag (BTI).
- Errors during command sequence (data reception) are detected by the CRC error flag and the data timeout error flag. When these flags are detected, set the CMDOFF bit in OPCR to 1 to issue the CMD12 command and suspend the command sequence.

- After read sequence, FIFO includes data. If necessary, 100 is written to the SET2 to SET0 bits in DMACR to read every data in FIFO.
- The end of the transfer with the DMAC is confirmed, and 0 is set to the DMAEN bit in DMACR.
- When the CRC error (CRCERI) or the command timeout error (CTERI) occurs during command response reception, write 1 to the CMDOFF bit and set DMACR to H'00.
- When the CRC error (CRCERI) or the data timeout error (DTERI) occurs during read data reception, write 1 to the CMDOFF bit and set DMACR to H'00 to clear the FIFO.

Notes: 1. Access from the DMAC the FIFO should be performed by byte or longword data.

2. In multiblock transfer, no normal command response can be received if you terminate the command sequence (by writing 1 in the CMDOFF bit) before the command response end interrupt (CRPI). To receive a normal command response, you need to continue the command sequence (by setting the RD_CONTI bit to 1) until the reception of the command response is completed.

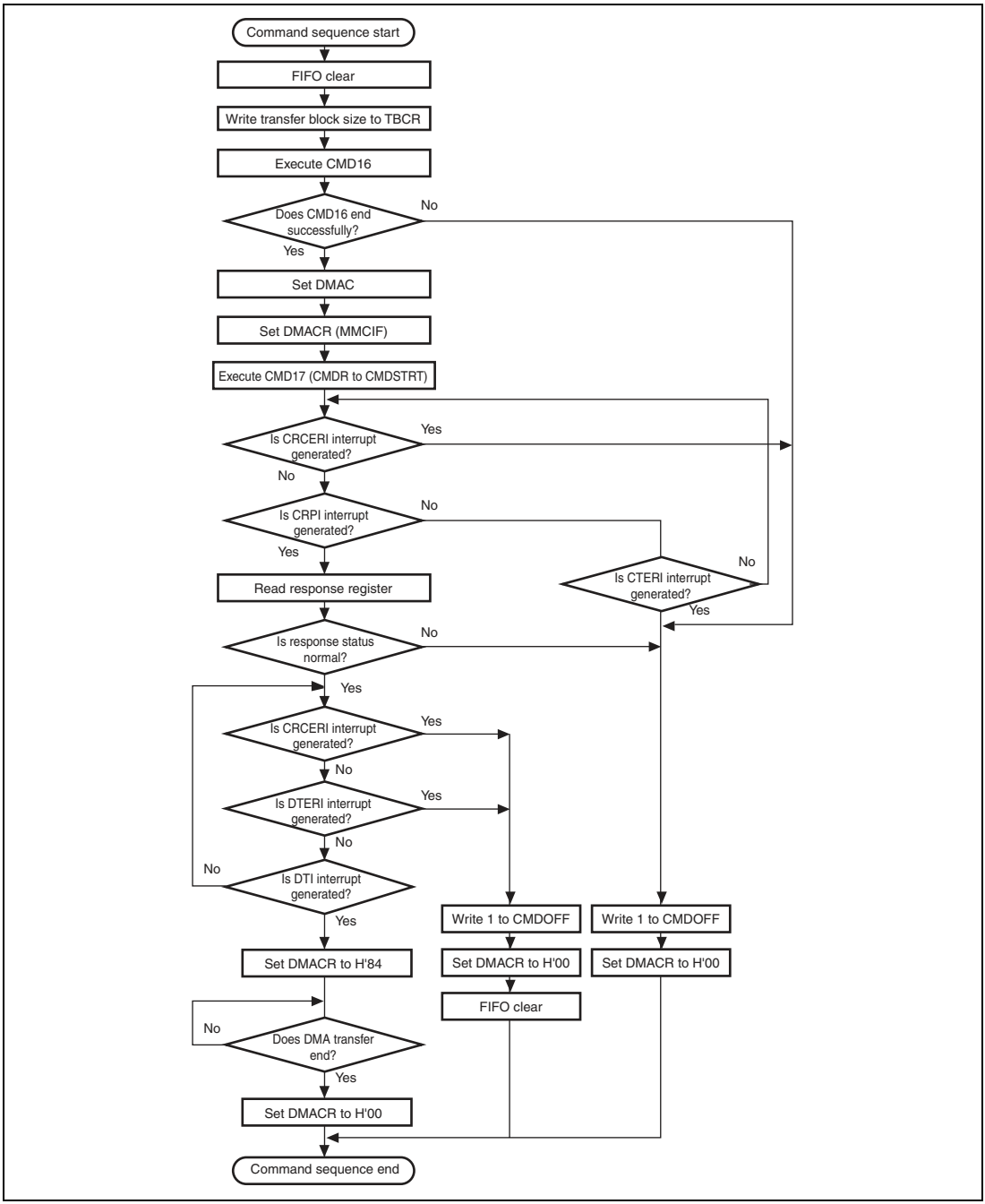


Figure 31.23 Operational Flowchart for Read Sequence (Single Block Transfer)

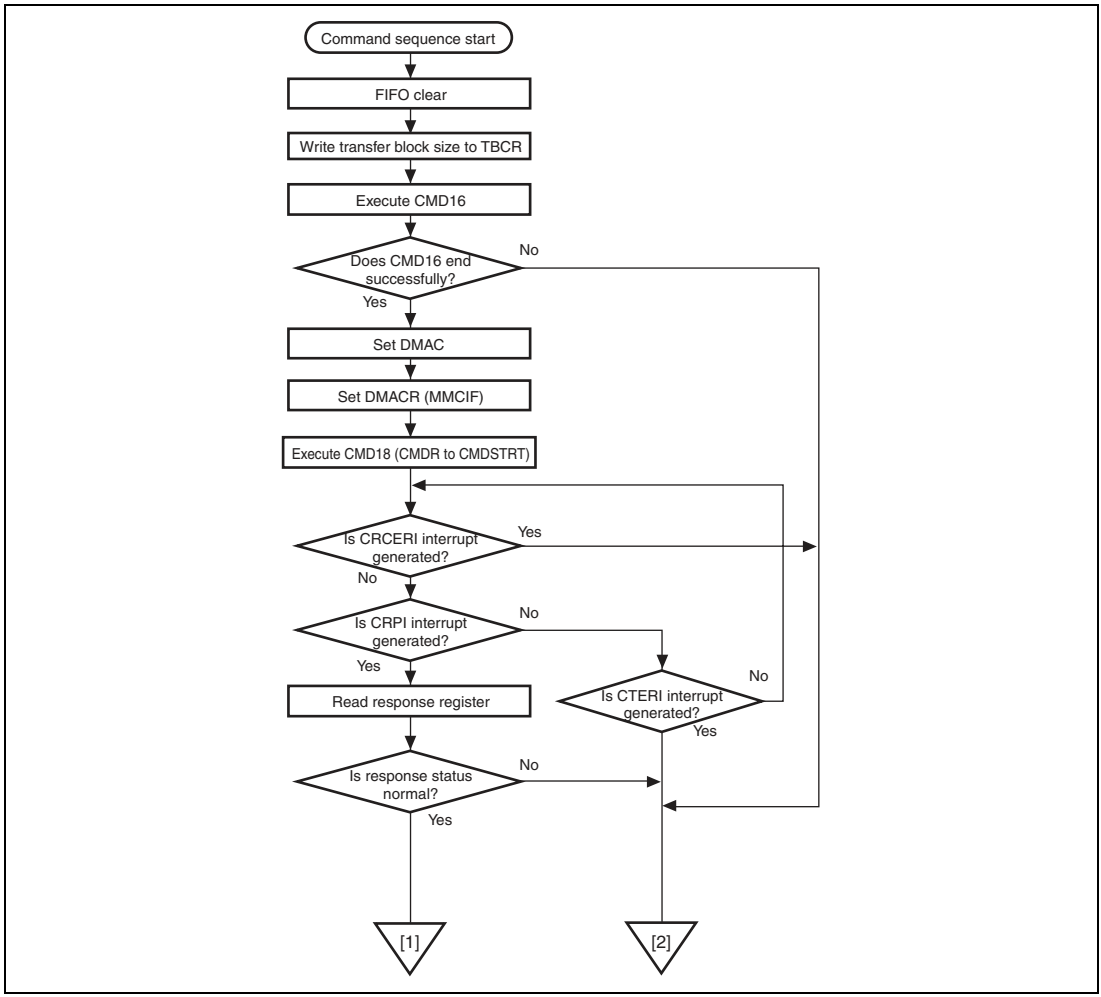


Figure 31.24 Operational Flowchart for Read Sequence (Open-ended Multiblock Transfer) (1)

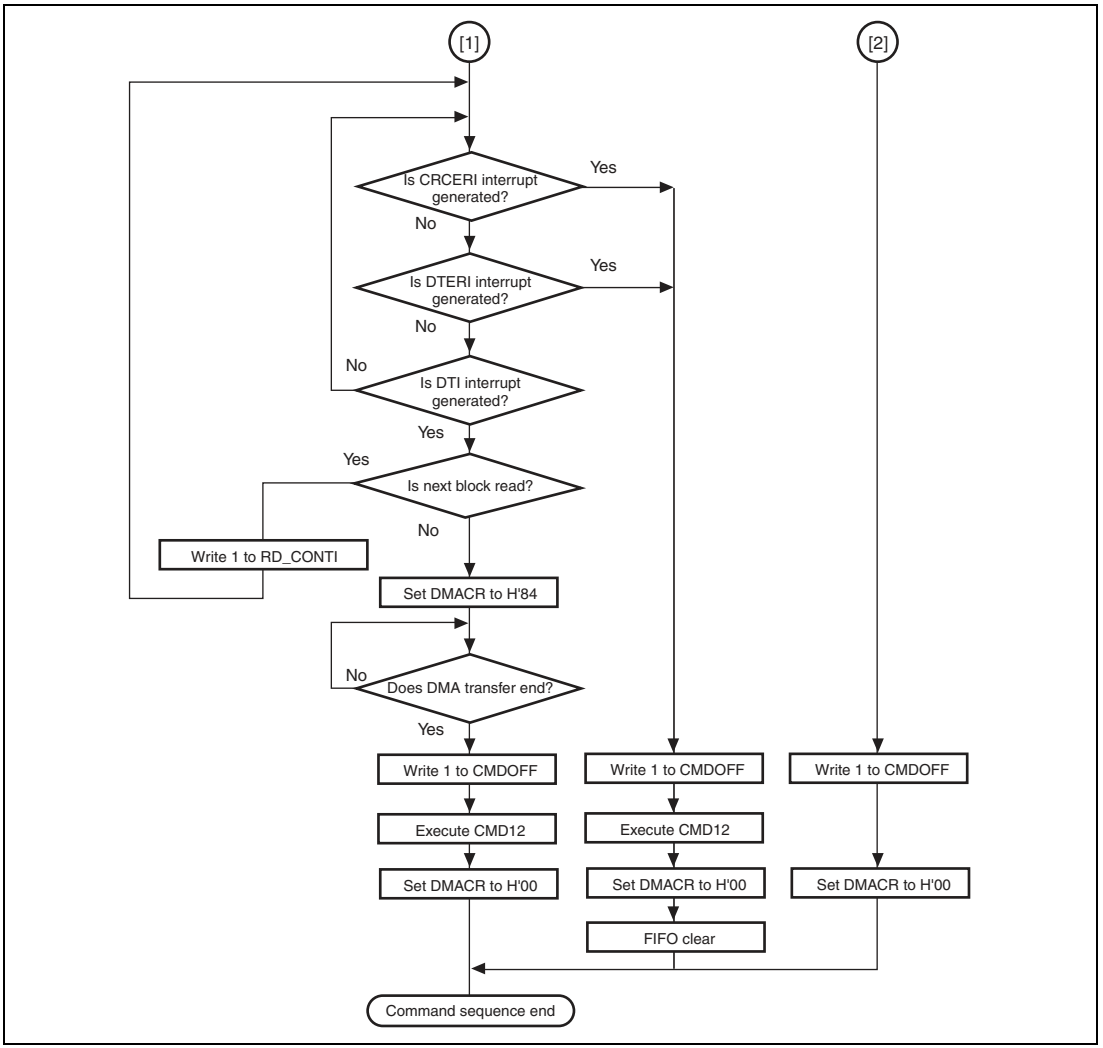


Figure 31.24 Operational Flowchart for Read Sequence (Open-ended Multiblock Transfer) (2)

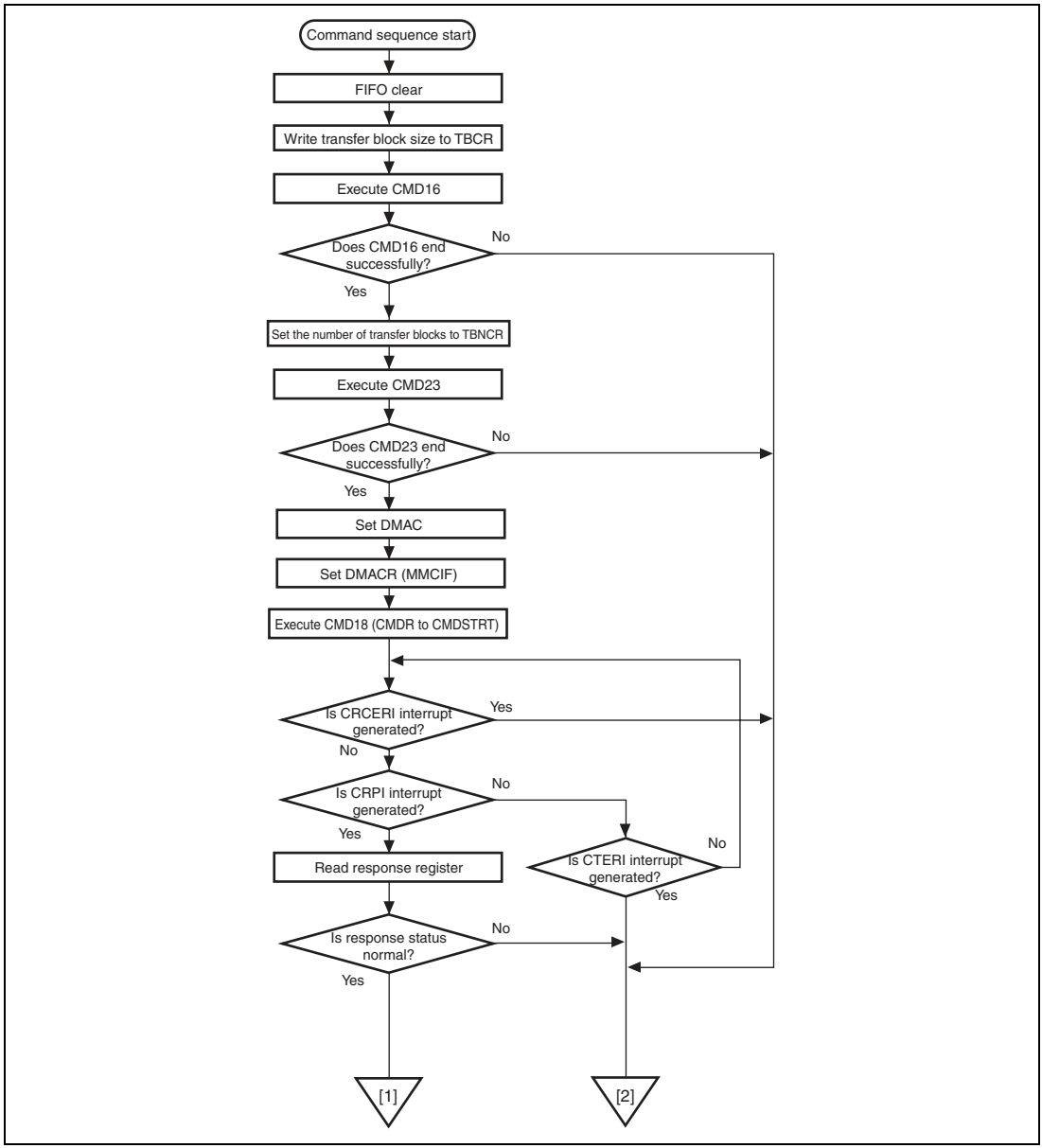


Figure 31.25 Operational Flowchart for Read Sequence (Pre-defined Multiblock Transfer) (1)

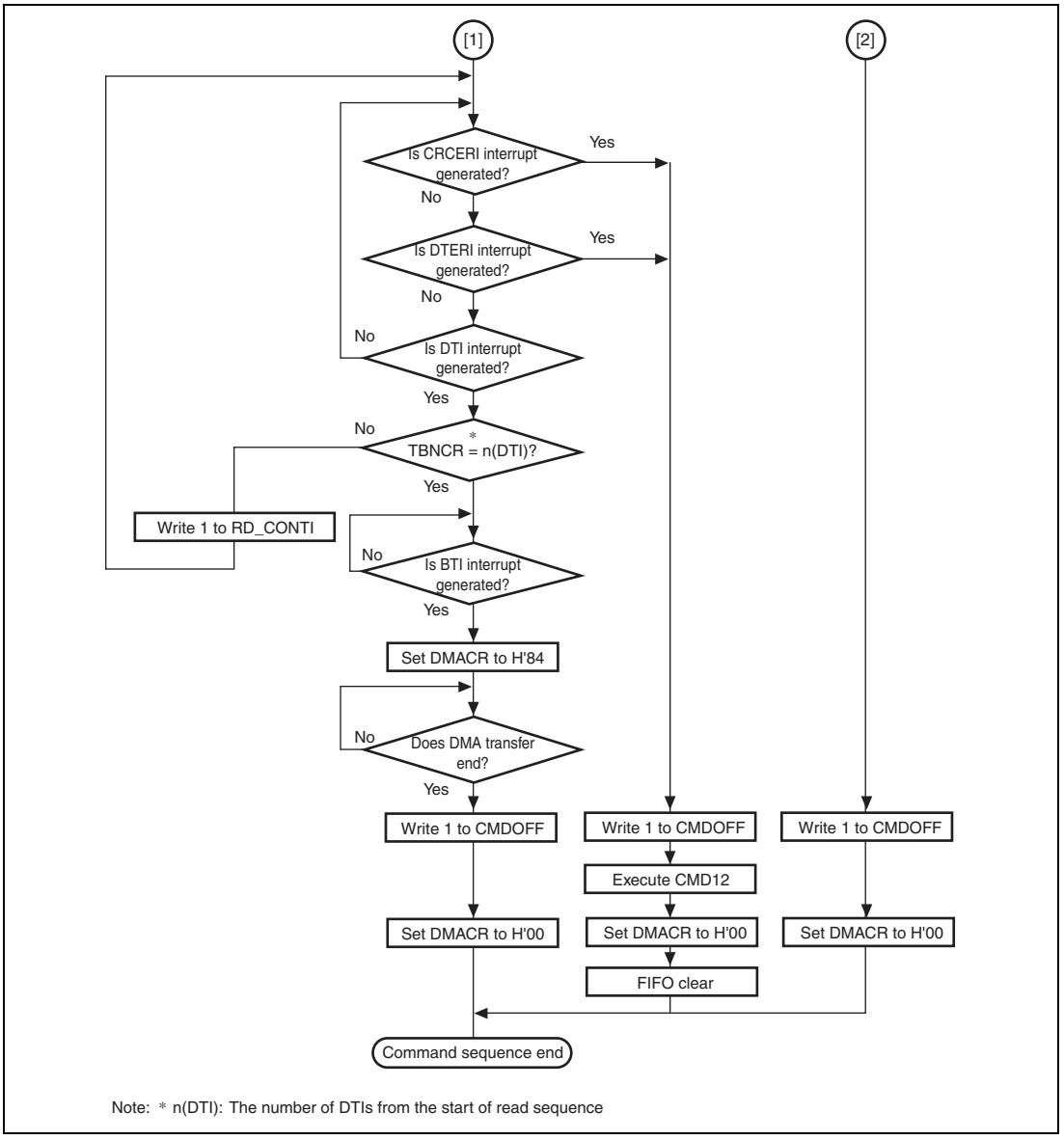


Figure 31.25 Operational Flowchart for Read Sequence (Pre-defined Multiblock Transfer) (2)

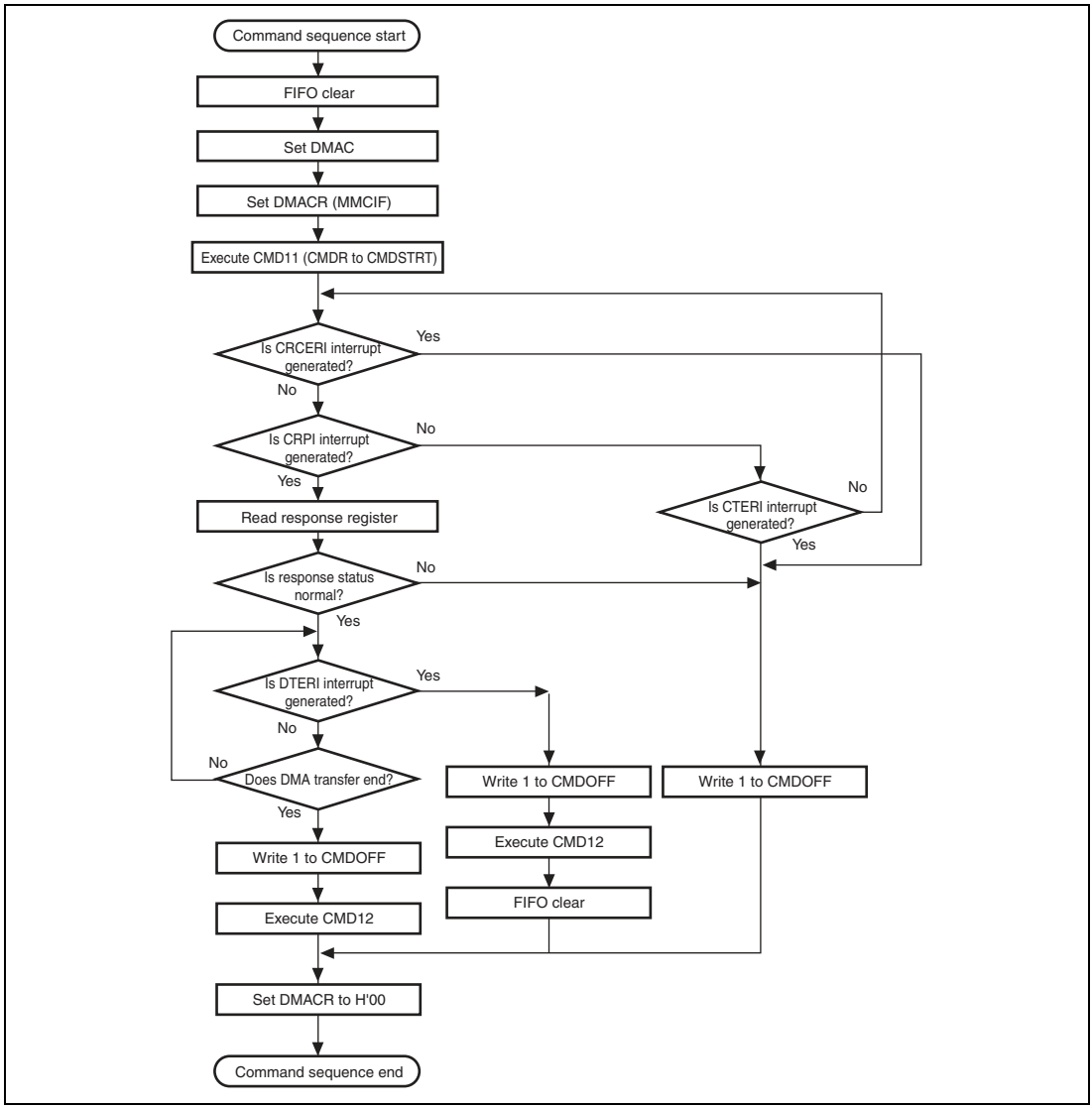


Figure 31.26 Operational Flowchart for Rear Sequence (Stream Read Transfer)

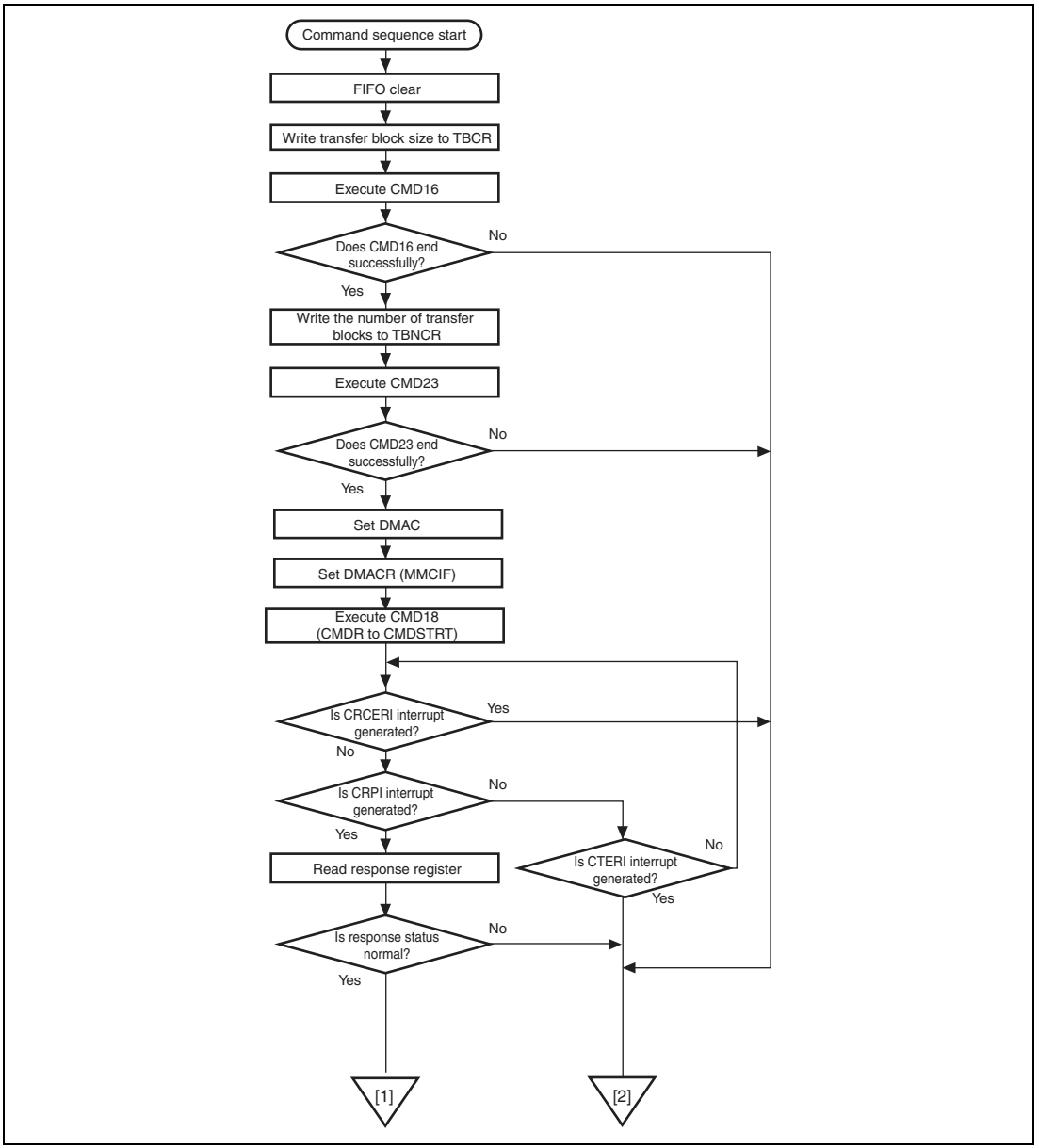


Figure 31.27 Operational Flowchart for Pre-defined Multiblock Read Transfer in Auto Mode (1)

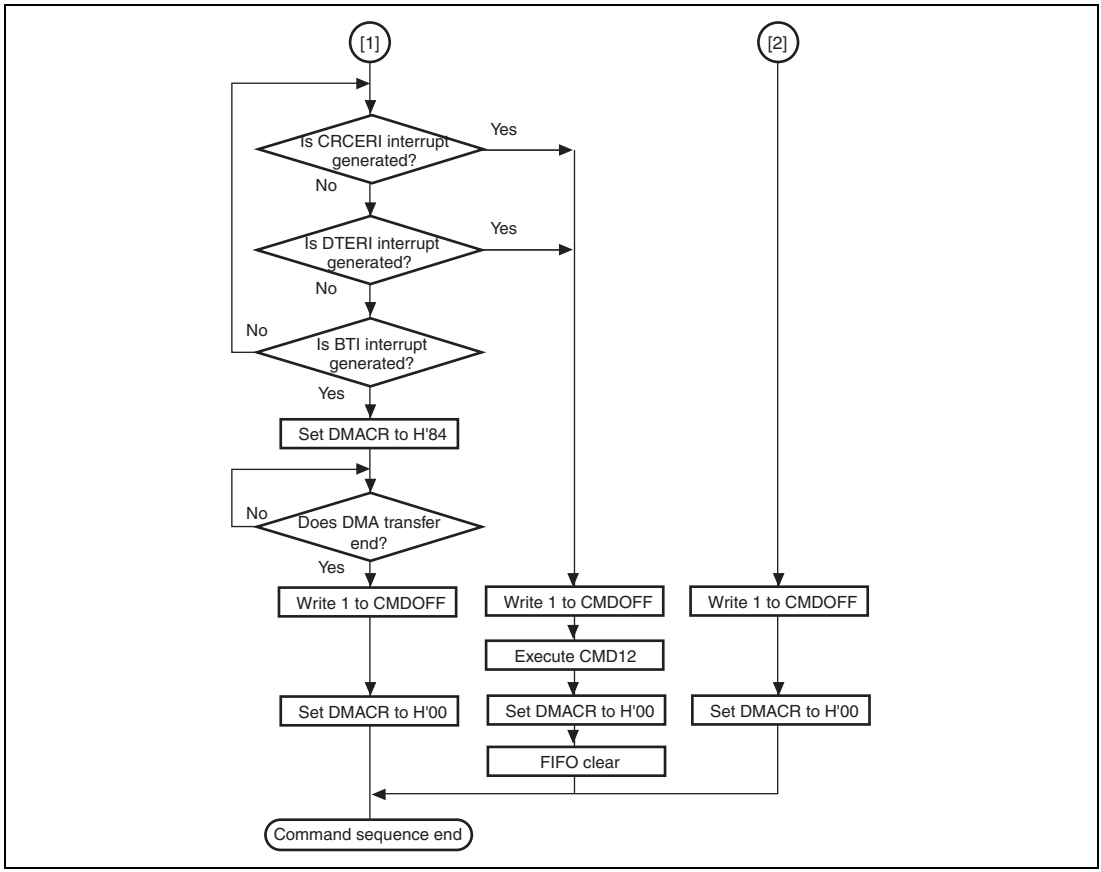


Figure 31.27 Operational Flowchart for Pre-defined Multiblock Read Transfer in Auto Mode (2)

31.5.2 Operation of Write Sequence

For transfer with DMAC, set MMCIF (DMACR) after setting DMAC. The FIFO ready flag is generated after DMACR is set and data more than threshold set in DMACR is written to the FIFO. Start transmission to the MMC after setting the flag. Figures 31.28 to 31.31 show the operational flowcharts for write sequence in MMC mode.

- FIFO is cleared.
- Write command is transmitted.
- DMACR is set and write data is set to the FIFO.
- Confirmed that the data more than DMACR setting condition is written to the FIFO by the FIFO ready flag (FRDYI), or that all data is written to the FIFO by the DMAC, and then the DATAEN bit in OPCR is set to 1 to start write data transmission.
- The end of the transfer with the DMAC is confirmed, and 0 is set to the DMAEN bit in DMACR.
- When the CRC error (CRCERI) or the command timeout error (CTERI) occurs during command response reception, write 1 to the CMDOFF bit.
- When the CRC error (CRCERI), write error (WRERI), or data timeout error (DTERI) occurs during write data transmission, 1 is written to the CMDOFF bit and DMACR is set to H'00 to clear the FIFO.

When the DMA is in use, an interrupt between blocks in pre-defined multiblock transfer can be processed by hard by setting the AUTO bit in DMACR to 1. Figure 31.32 shows the operational flowchart for pre-defined multi write sequence in MMC mode.

- FIFO is cleared.
- The number of blocks is set to TBNCR.
- The START bit in CMDSTRT is set to 1 and command transmission is started.
- Command response is received from the MMC.
- If the MMC does not return the command response, it detected by the command timeout error flag (CTERI).
- DMACR is set and write data is set to the FIFO.
- The end of the transfer with the DMAC is confirmed, and 0 is set to the DMAEN bit in DMACR.
- Command sequence end is detected by polling the BUSY flag in CSTR or the multiblock transfer (pre-defined) end flag (BTI).
- Errors during command sequence (data transmission) are detected by the CRC error flag (CRCERI) or the data timeout error flag. When interrupts are detected, set the CMDOFF bit in OPCR to 1 to issue the CMD12 command and suspend the command sequence.

- Not in the data busy state is confirmed. If the data busy state is entered, the data busy state is detected by the data busy end flag (DBSYI).
- After data transfer (after DRPI is detected), check whether the data busy state is entered. If the data busy state is entered, the end of the data busy state is detected by the data busy end flag (DBSYI).
- The CMDOFF bit is set to 1 and command sequence is ended.
- When the CRC error (CRCERI) or the command timeout error (CTERI) occurs during command response reception, write 1 to the CMDOFF bit.
- When the CRC error (CRCERI), write error (WRERI), or the data timeout error (DTERI) occurs during write data transmission, write 1 to the CMDOFF bit and set DMACR to H'00 to clear the FIFO.

Note: Access from the DMAC to the FIFO should be performed by byte or longword data.

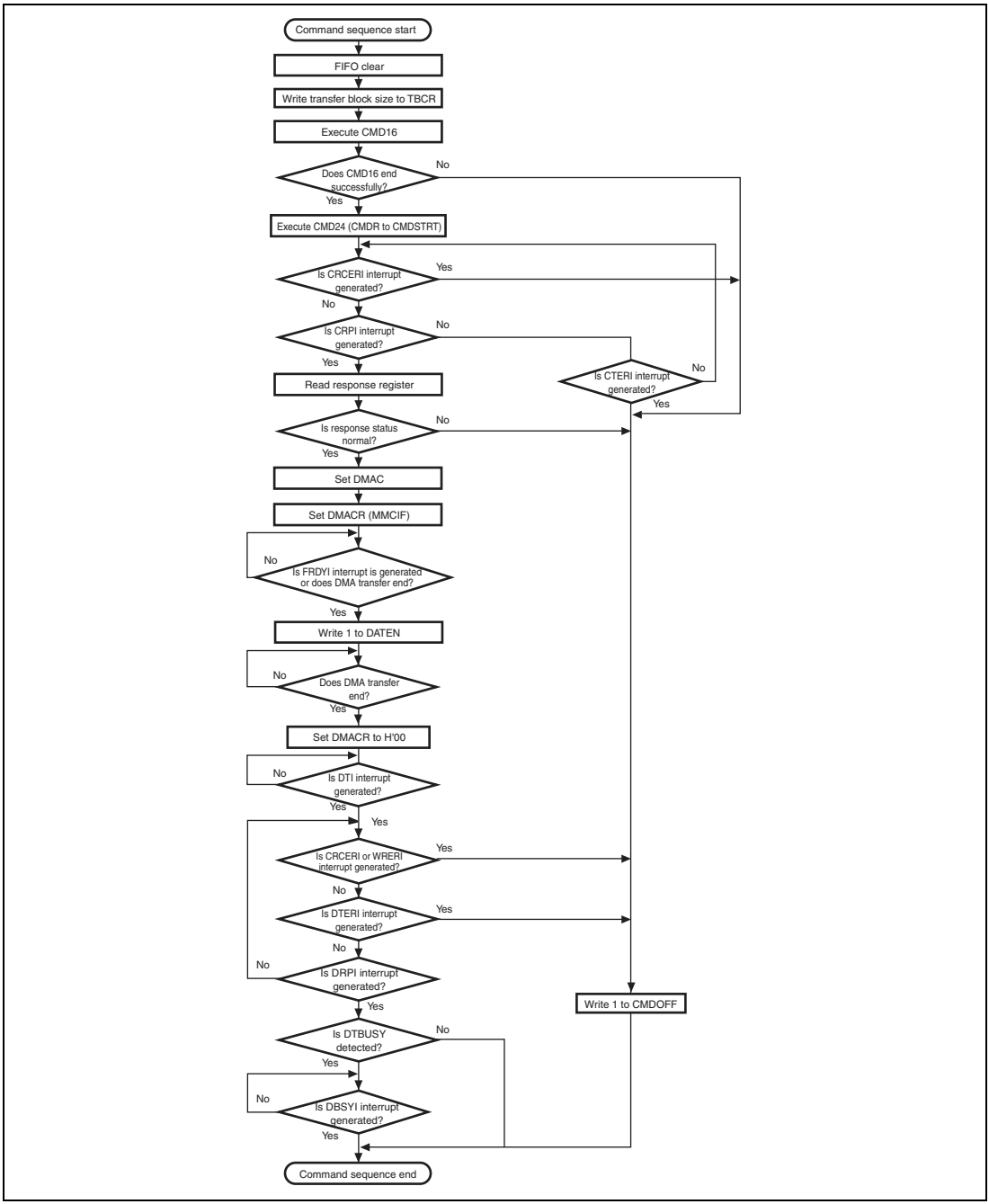


Figure 31.28 Operational Flowchart for Write Sequence (Single Block Transfer)

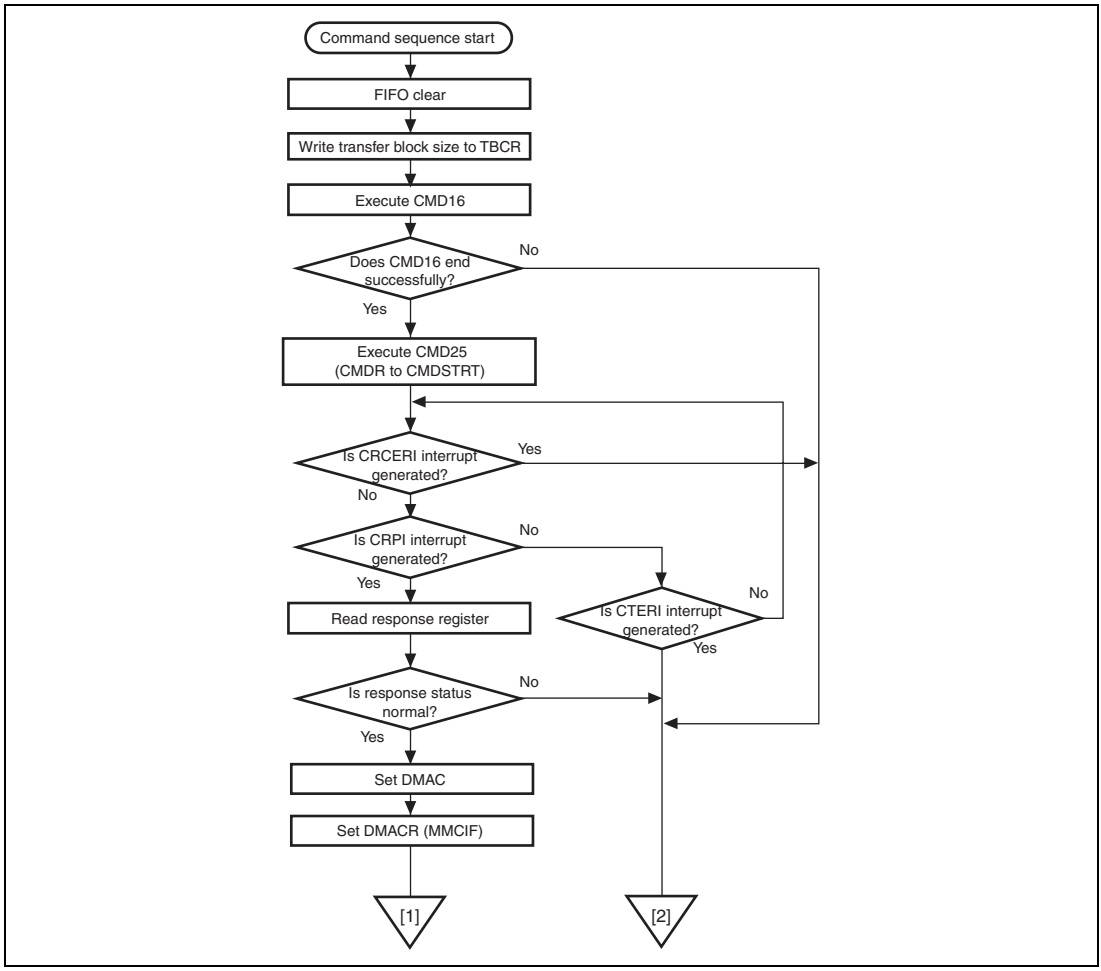


Figure 31.29 Operational Flowchart for Write Sequence (Open-ended Multiblock Transfer) (1)

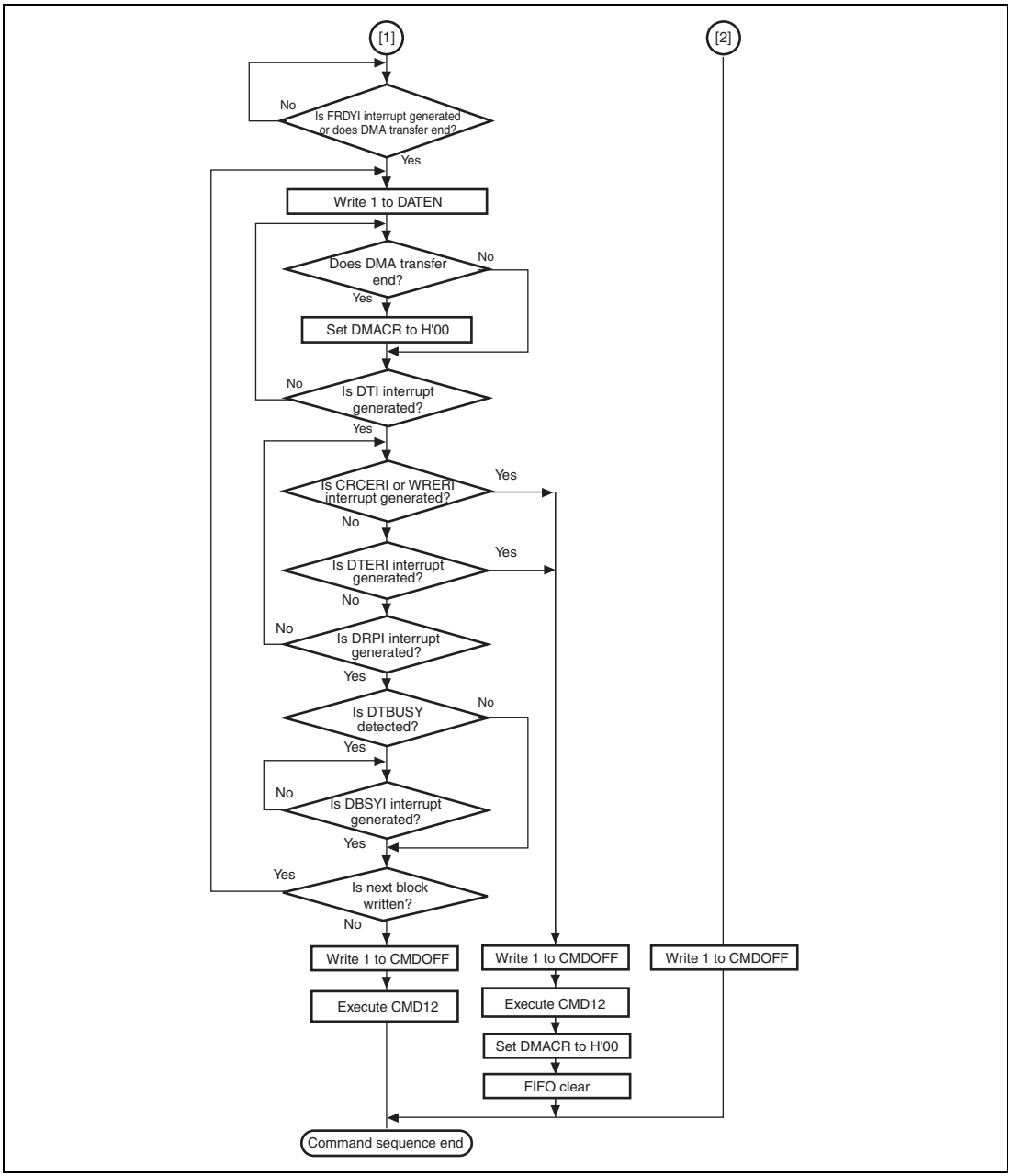


Figure 31.29 Operational Flowchart for Write Sequence (Open-ended Multiblock Transfer) (2)

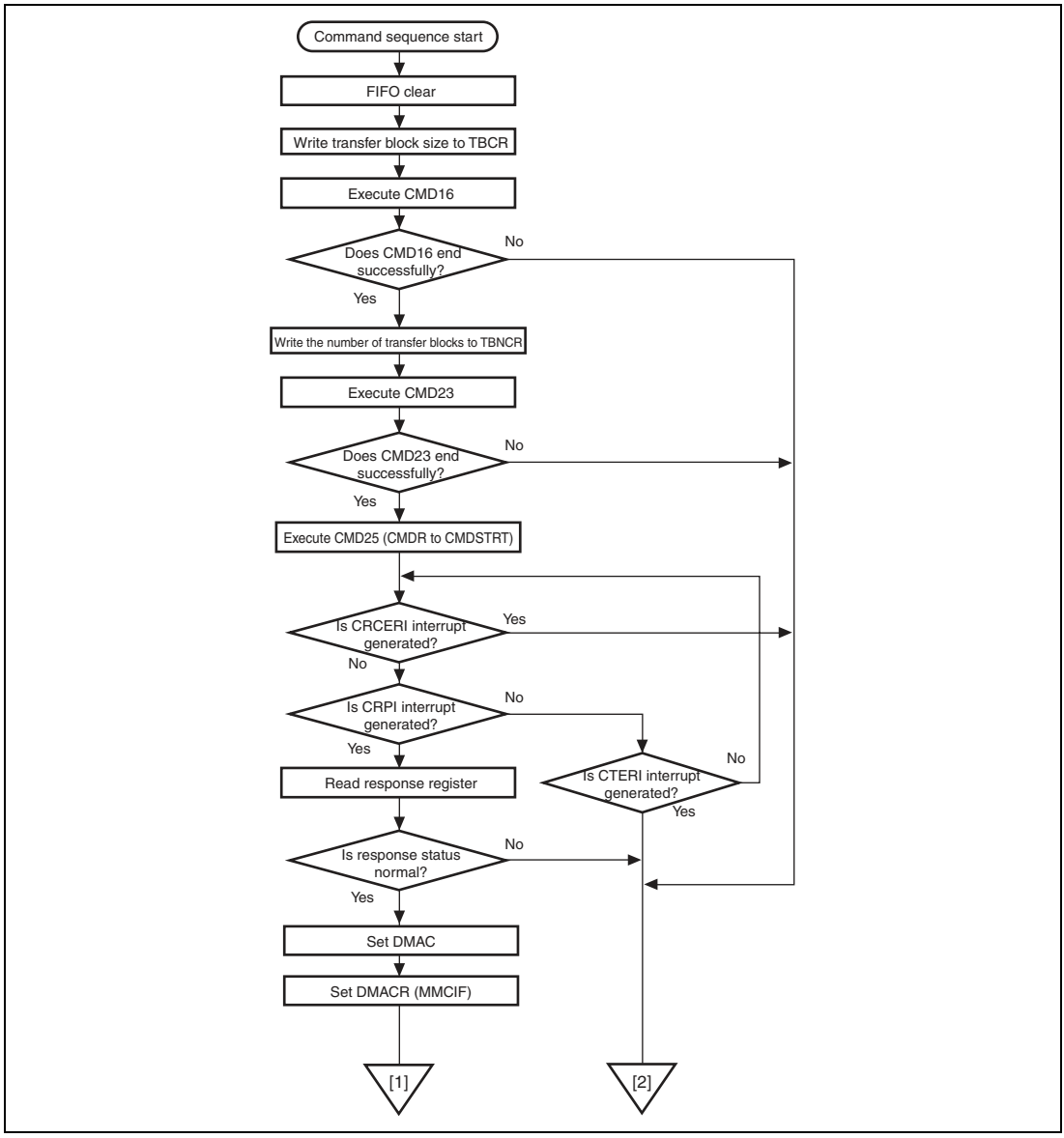


Figure 31.30 Operational Flowchart for Write Sequence (Pre-defined Multiblock Transfer) (1)

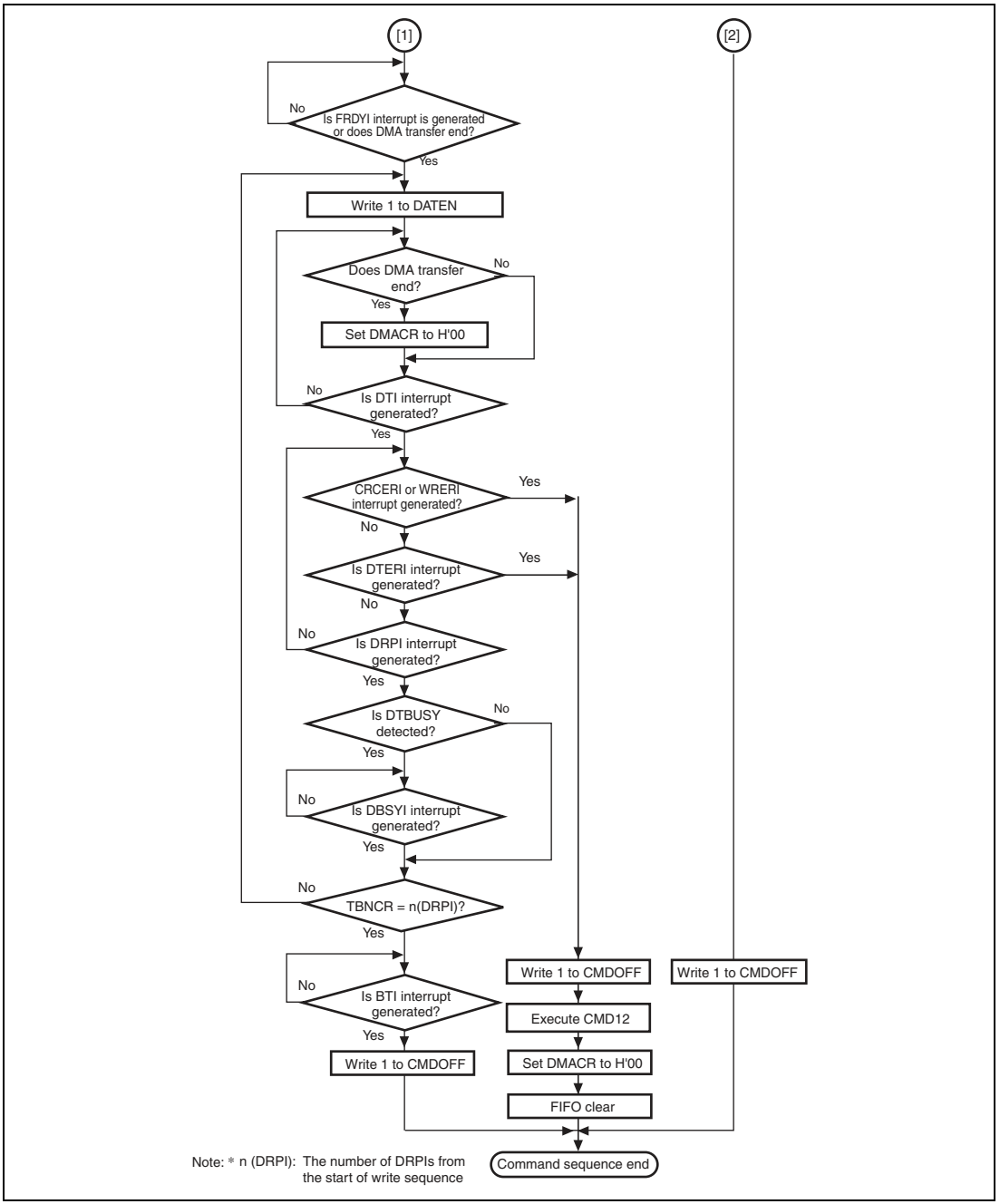


Figure 31.30 Operational Flowchart for Write Sequence (Pre-defined Multiblock Transfer) (2)

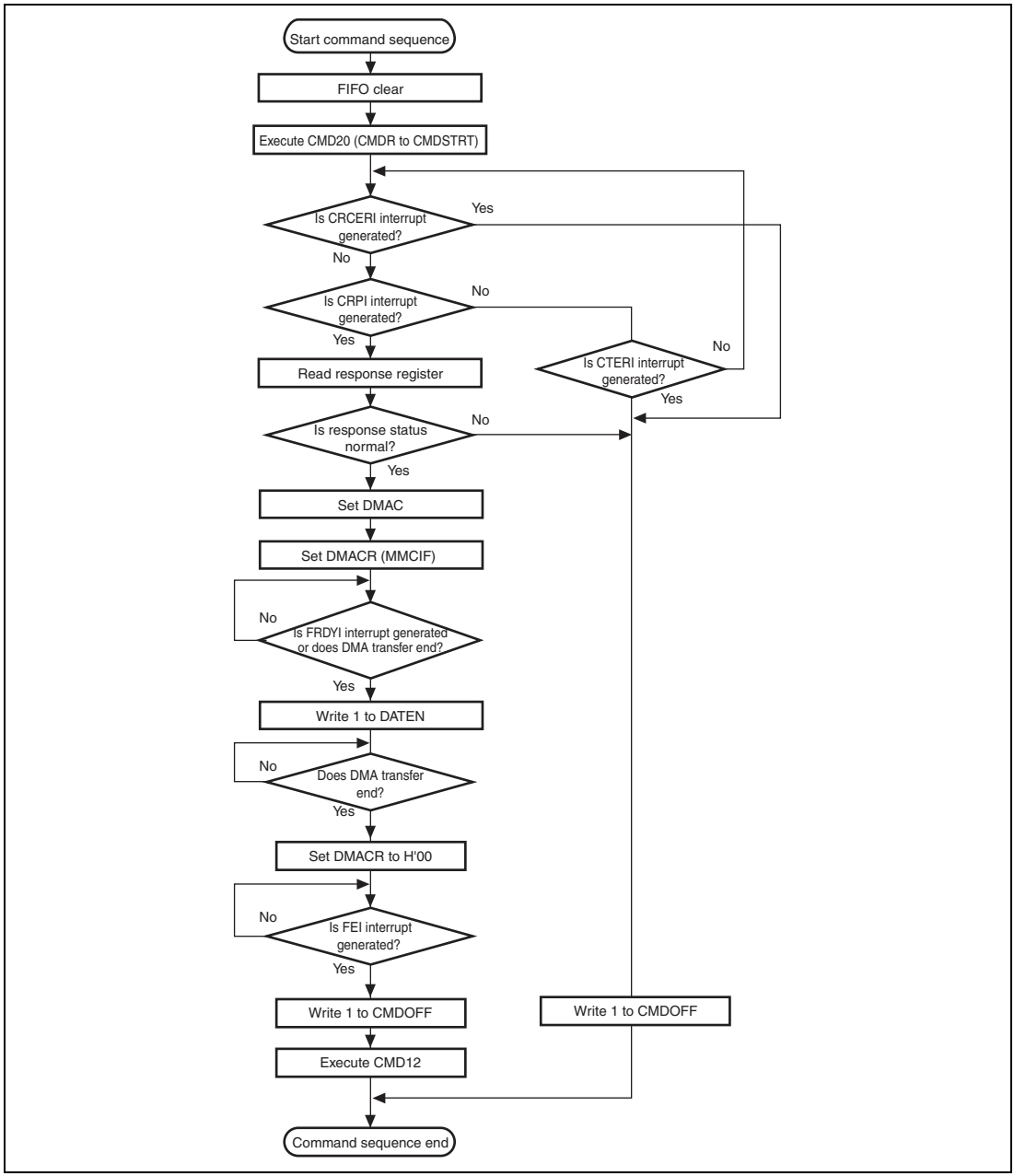


Figure 31.31 Operational Flowchart for Write Sequence (Stream Write Transfer)

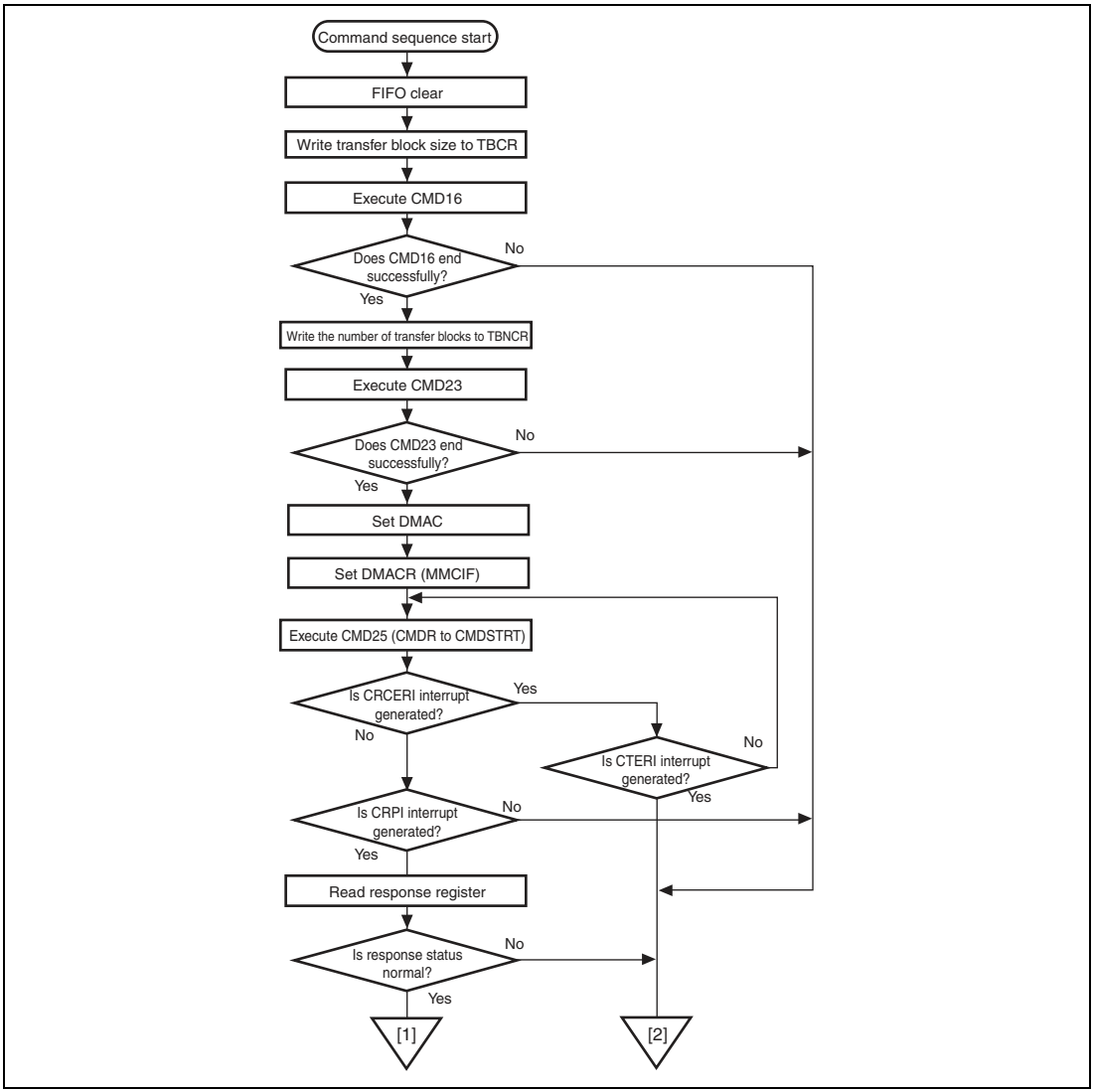


Figure 31.32 Operational Flowchart for Pre-defied Multiblock Write Transfer in Auto Mode (1)

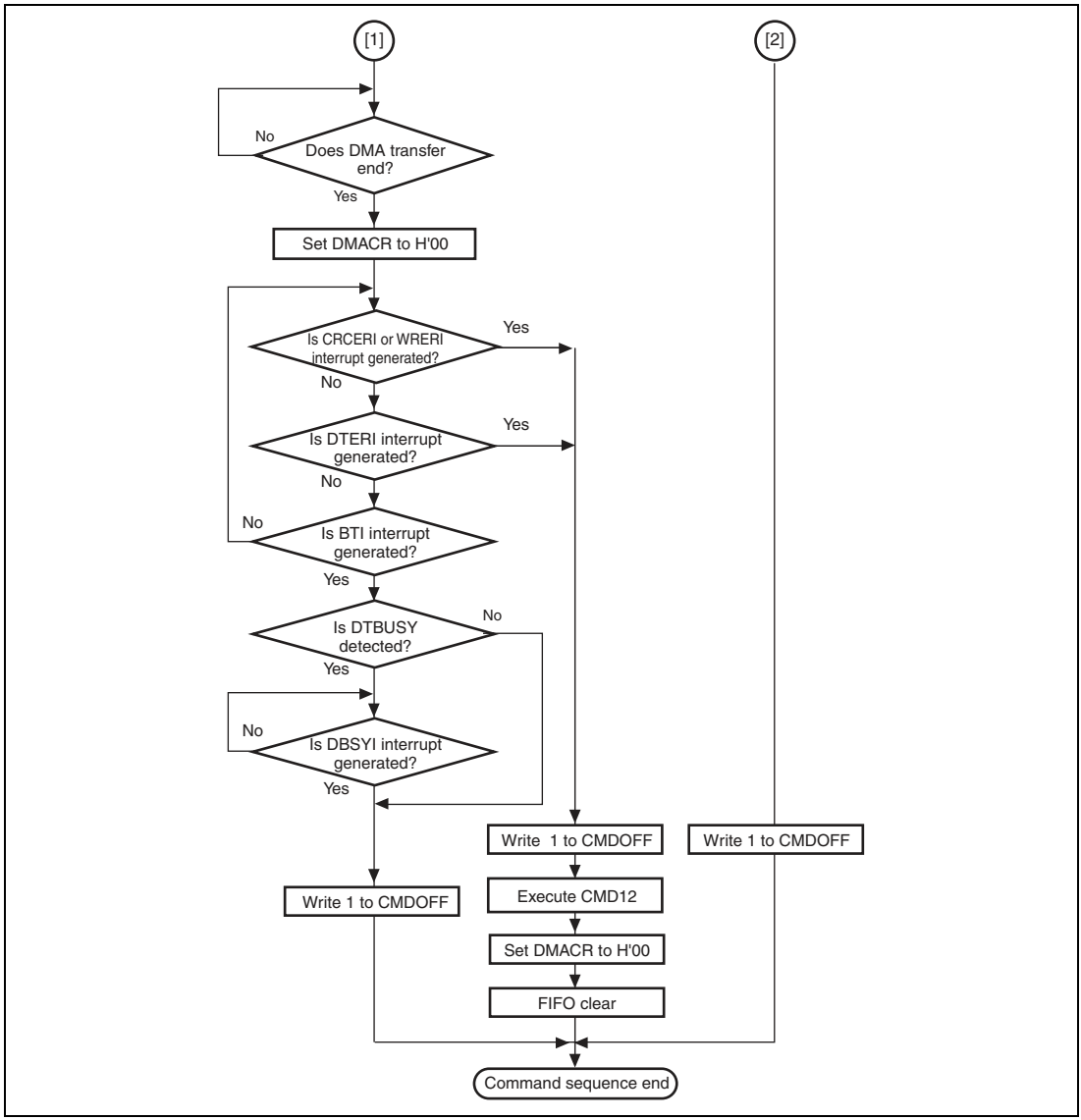


Figure 31.32 Operational Flowchart for Pre-defied Multiblock Write Transfer in Auto Mode (2)

31.6 MMCIF Interrupt Sources

Table 31.5 lists the MMCIF interrupt sources. The interrupt sources are classified into four groups, and four interrupt vectors are assigned. Each interrupt source can be individually enabled by the enable bits in INTCR0 to INTCR2. The disabled interrupt source does not set the flag.

Table 31.5 MMCIF Interrupt Sources

Name	Interrupt Source	Interrupt Flag
MMCI0	Write error	WRERI
	CRC error*	CRCERI*
	Data timeout error	DTERI
	Command timeout error	CTERI
MMCI2	FIFO empty	FEI
	FIFO full	FFI
MMCI1	Data response	DPRI
	Data transfer end	DTI
	Command response end	CRPI
	Command output end	CMDI
	Data busy end	DBSYI
	Block transfer end	BTI
MMCI3	FIFO ready	FRDYI

Note: * Excluding the CRC error in the R2 command response.

Section 32 SSL Accelerator (SSL)

SSL accelerator (SSL: Security Socket Layer) performs the RSA operation (RSA: Rivest Shamir Adleman) with public key which is used to sign data with a digital signature, and encrypts or decrypts the common key, DES (Data Encryption Standard), and Triple-DES that are used to preserve secrecy of data in the network to perform efficient encryption communication.

With the RSA operation circuit, 32 to 512-bit width of addition, subtraction, and multiplication and 512-bit width of macro operations as well as 512-bit width of RSA operation (modular multiplication using multiple precision integers) are executed.

The SSL accelerator can use 56 bits or more encryption key. If to be exported or the like, the necessary procedures must be taken according to the foreign exchange law. Please contact your nearest Renesas sales office when you require the detailed functional specification for the SSL accelerator.

Section 33 User Break Controller (UBC)

The user break controller (UBC) provides functions that simplify program debugging. These functions make it easy to design an effective self-monitoring debugger, enabling the chip to debug programs without using an in-circuit emulator. Break conditions that can be set in the UBC are instruction fetch or data read/write access, data size, data contents, address value, and stop timing in the case of instruction fetch.

33.1 Features

1. The following break comparison conditions can be set.

   Number of break channels: two channels (channels A and B)

   User break can be requested as either the independent or sequential condition on channels A and B (sequential break setting: channel A and then channel B match with break conditions, but not in the same bus cycle).

- Address

  Compares 40 bits configured of the ASID and addresses 32 bits: the ASID can be selected either all-bit comparison or all-bit mask. Comparison bits are maskable in 1-bit units; user can mask addresses at lower 12 bits (4-k page), lower 10 bits (1-k page), or any size of page, etc.

  One of the four address buses (logic address bus (LAB), internal address bus (IAB),

  One of the four data buses (L-bus data (LDB), I-bus data (IDB), X-memory data bus (XDB) and Y-memory data bus (YDB)) can be selected.

- Data

  Only on channel B, 32-bit maskable.

  One of the four data buses (L-bus data (LDB), I-bus data (IDB), X-memory data bus (XDB) and Y-memory data bus (YDB)) can be selected.

- Bus cycle

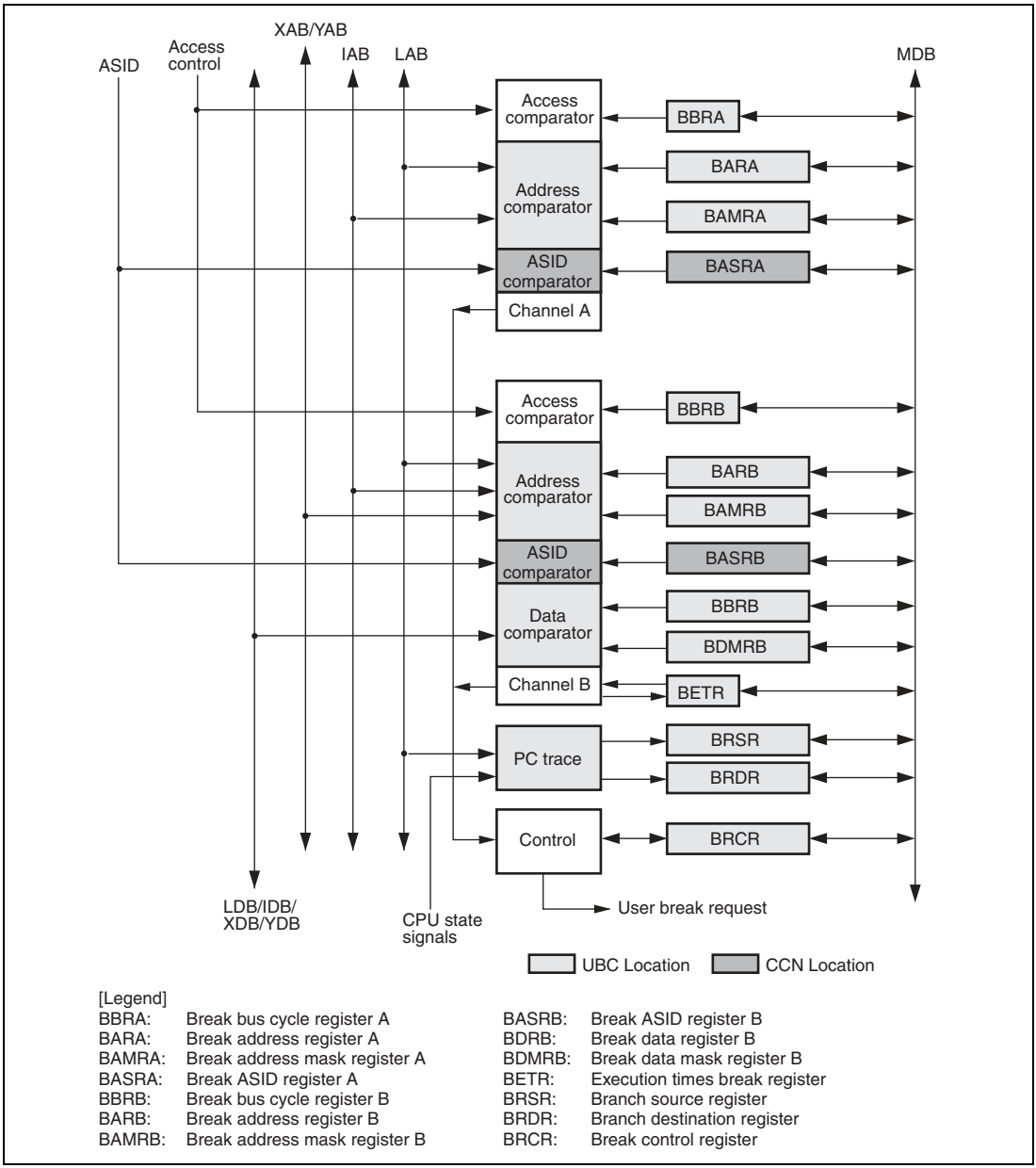
  Instruction fetch or data access

- Read/write
- Operand size

  Byte, word, and longword

2. A user-designed user-break condition exception processing routine can be run.
3. In an instruction fetch cycle, it can be selected that a break is set before or after an instruction is executed.
4. Maximum repeat times for the break condition (only for channel B): $2^{12} - 1$ times.
5. Eight pairs of branch source/destination buffers.

Figure 33.1 shows a block diagram of the UBC.



[Legend]

BBRA:	Break bus cycle register A	BASRB:	Break ASID register B
BARA:	Break address register A	BDRB:	Break data register B
BAMRA:	Break address mask register A	BDMRB:	Break data mask register B
BASRA:	Break ASID register A	BETR:	Execution times break register
BBRB:	Break bus cycle register B	BRSR:	Branch source register
BARB:	Break address register B	BRDR:	Branch destination register
BAMRB:	Break address mask register B	BRCR:	Break control register

Figure 33.1 Block Diagram of UBC

33.2 Register Descriptions

The user break controller has the following registers. Refer to section 37, List of Registers, for more details on the addresses and access size of these registers.

- Break address register A (BARA)
- Break address mask register A (BAMRA)
- Break bus cycle register A (BBRA)
- Break address register B (BARB)
- Break address mask register B (BAMRB)
- Break bus cycle register B (BBRB)
- Break data register B (BDRB)
- Break data mask register B (BDMRB)
- Break control register (BRCR)
- Execution times break register (BETR)
- Branch source register (BRSR)
- Branch destination register (BRDR)
- Break ASID register A (BASRA)
- Break ASID register B (BASRB)

33.2.1 Break Address Register A (BARA)

BARA is a 32-bit readable/writable register. BARA specifies the address used as a break condition in channel A.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAA31 to BAA0	All 0	R/W	Break Address A Store the address on the LAB or IAB specifying break conditions of channel A.

33.2.2 Break Address Mask Register A (BAMRA)

BAMRA is a 32-bit readable/writable register. BAMRA specifies bits masked in the break address specified by BARA.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAMA31 to BAMA 0	All 0	R/W	Break Address Mask A Specify bits masked in the channel A break address bits specified by BARA (BAA31 to BAA0). 0: Break address bit BAAn of channel A is included in the break condition 1: Break address bit BAAn of channel A is masked and is not included in the break condition Note: n = 31 to 0

33.2.3 Break Bus Cycle Register A (BBRA)

BBRA is a 16-bit readable/writable register, which specifies (1) L bus cycle or I bus cycle, (2) instruction fetch or data access, (3) read or write, and (4) operand size as the break conditions of channel A.

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	CDA1	0	R/W	L Bus Cycle/I Bus Cycle Select A
6	CDA0	0	R/W	Select the L bus cycle or I bus cycle as the bus cycle of the channel A break condition. 00: Condition comparison is not performed 01: The break condition is the L bus cycle 10: The break condition is the I bus cycle 11: The break condition is the L bus cycle

Bit	Bit Name	Initial Value	R/W	Description
5	IDA1	0	R/W	Instruction Fetch/Data Access Select A
4	IDA0	0	R/W	Select the instruction fetch cycle or data access cycle as the bus cycle of the channel A break condition. 00: Condition comparison is not performed 01: The break condition is the instruction fetch cycle 10: The break condition is the data access cycle 11: The break condition is the instruction fetch cycle or data access cycle
3	RWA1	0	R/W	Read/Write Select A
2	RWA0	0	R/W	Select the read cycle or write cycle as the bus cycle of the channel A break condition. 00: Condition comparison is not performed 01: The break condition is the read cycle 10: The break condition is the write cycle 11: The break condition is the read cycle or write cycle
1	SZA1	0	R/W	Operand Size Select A
0	SZA0	0	R/W	Select the operand size of the bus cycle for the channel A break condition. 00: The break condition does not include operand size 01: The break condition is byte access 10: The break condition is word access 11: The break condition is longword access

33.2.4 Break Address Register B (BARB)

BARB is a 32-bit readable/writable register. BARB specifies the address used as a break condition in channel B. Control bits CDB1, CDB0, XYE, and XYS in BBRB select one of the four address buses for break condition B.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAB31 to BAB0	All 0	R/W	Break Address B Stores an address which specifies a break condition in channel B. If the I bus or L bus is selected in BBRB, an IAB or LAB address is set in BAB31 to BAB0. If the X memory is selected in BBRB, the values in bits 15 to 1 in XAB are set in BAB31 to BAB17. In this case, the values in BAB16 to BAB0 are arbitrary. If the Y memory is selected in BBRB, the values in bits 15 to 1 in YAB are set in BAB15 to BAB1. In this case, the values in BAB31 to BAB16 are arbitrary.

Table 33.1 Specifying Break Address Register

Bus Selection in BBRB	BAB31 to BAB17	BAB16	BAB15 to BAB1	BAB0
L bus	LAB31 to LAB0			
I bus	IAB31 to IAB0			
X bus	XAB15 to XAB1	Don't care	Don't care	Don't care
Y bus	Don't care	Don't care	YAB15 to YAB1	Don't care

33.2.5 Break Address Mask Register B (BAMRB)

BAMRB is a 32-bit readable/writable register. BAMRB specifies bits masked in the break address specified by BARB.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAMB31 to BAMB0	All 0	R/W	Break Address Mask B Specifies bits masked in the break address of channel B specified by BARB (BAB31 to BAB0). 0: Break address BABn of channel B is included in the break condition 1: Break address BABn of channel B is masked and is not included in the break condition Note: n = 31 to 0

33.2.6 Break Data Register B (BDRB)

BDRB is a 32-bit readable/writable register. The control bits CDB1, CDB0, XYE, and XYS in the break bus cycle register (BBRB) select one of the four data buses for break condition B.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDB31 to BDB0	All 0	R/W	Break Data Bit B Stores data which specifies a break condition in channel B. If the I bus is selected in BBRB, the break data on IDB is set in BDB31 to BDB0. If the L bus is selected in BBRB, the break data on LDB is set in BDB31 to BDB0. If the X memory is selected in BBRB, the break data in bits 15 to 0 in XDB is set in BDB31 to BDB16. In this case, the values in BDB15 to BDB0 are arbitrary. If the Y memory is selected in BBRB, the break data in bits 15 to 0 in YDB are set in BDB15 to BDB0. In this case, the values in BDB31 to BDB16 are arbitrary.

Table 33.2 Specifying Break Data Register

Bus Selection in BBRB	BDB31 to BDB16	BDB15 to BDB0
L bus	LDB31 to LDB0	
I bus	IDB31 to IDB0	
X bus	XDB15 to XDB0	Don't care
Y bus	Don’t care	YDB15 to YDB0

Notes: 1. Specify an operand size when including the value of the data bus in the break condition.
2. When the byte size is selected as a break condition, the same byte data must be set in bits 15 to 8 and 7 to 0 in BDRB as the break data.
3. Set the data in bits 31 to 16 when including the value of the data bus as an L-bus break condition for the MOVS.W @-As,Ds, MOVS.W @As,Ds, MOVS.W @As+,Ds, or MOVS.W @As+Ix,Ds instruction.

33.2.7 Break Data Mask Register B (BDMRB)

BDMRB is a 32-bit readable/writable register. BDMRB specifies bits masked in the break data specified by BDRB.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDMB31 to BDMB0	All 0	R/W	Break Data Mask B Specifies bits masked in the break data of channel B specified by BDRB (BDB31 to BDB0). 0: Break data BDBn of channel B is included in the break condition 1: Break data BDBn of channel B is masked and is not included in the break condition Note: n = 31 to 0

Notes: 1. Specify an operand size when including the value of the data bus in the break condition.
2. When the byte size is selected as a break condition, the same byte data must be set in bits 15 to 8 and 7 to 0 in BDRB as the break mask data in BDMRB.
3. Set the mask data in bits 31 to 16 when including the value of the data bus as an L-bus break condition for the MOVS.W @-As,Ds, MOVS.W @As,Ds, MOVS.W @As+,Ds, or MOVS.W @As+Ix,Ds instruction.

33.2.8 Break Bus Cycle Register B (BBRB)

BBRB is a 16-bit readable/writable register, which specifies (1) X bus or Y bus, (2) L bus cycle or I bus cycle, (3) instruction fetch or data access, (4) read or write, and (5) operand size as the break conditions of channel B.

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	XYE	0	R/W	Selects the X memory bus or Y memory bus as the channel B break condition. Note that this bit setting is enabled only when the L bus is selected in the CDB1 and CDB0 bits. Selection between the X memory bus and Y memory bus is done by the XYS bit. 0: Selects L bus for the channel B break condition 1: Selects X/Y memory bus for the channel B break condition
8	XYS	0	R/W	Selects the X bus or the Y bus as the bus of the channel B break condition. 0: Selects the X bus for the channel B break condition 1: Selects the Y bus for the channel B break condition
7	CDB1	0	R/W	L Bus Cycle/I Bus Cycle Select B
6	CDB0	0	R/W	Select the L bus cycle or I bus cycle as the bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the L bus cycle 10: The break condition is the I bus cycle 11: The break condition is the L bus cycle
5	IDB1	0	R/W	Instruction Fetch/Data Access Select B
4	IDB0	0	R/W	Select the instruction fetch cycle or data access cycle as the bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the instruction fetch cycle 10: The break condition is the data access cycle 11: The break condition is the instruction fetch cycle or data access cycle

Bit	Bit Name	Initial Value	R/W	Description
3	RWB1	0	R/W	Read/Write Select B
2	RWB0	0	R/W	Select the read cycle or write cycle as the bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the read cycle 10: The break condition is the write cycle 11: The break condition is the read cycle or write cycle
1	SZB1	0	R/W	Operand Size Select B
0	SZB0	0	R/W	Select the operand size of the bus cycle for the channel B break condition. 00: The break condition does not include operand size 01: The break condition is byte access 10: The break condition is word access 11: The break condition is longword access

33.2.9 Break Control Register (BRCR)

BRCR sets the following conditions:

1. Channels A and B are used in two independent channel conditions or under the sequential condition.
2. A break is set before or after instruction execution.
3. Specify whether to include the number of execution times on channel B in comparison conditions.
4. Determine whether to include data bus on channel B in comparison conditions.
5. Enable PC trace.
6. Enable ASID check.

BRCR is a 32-bit readable/writable register that has break conditions match flags and bits for setting a variety of break conditions.

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	BASMA	0	R/W	Break ASID Mask A Specifies whether bits in channel A break ASID7 to ASID0 (BASA7 to BASA0) which are set in BASRA are masked or not. 0: All BASRA bits are included in the break conditions and the ASID is checked 1: All BASRA bits are not included in the break conditions and the ASID is not checked
20	BASMB	0	R/W	Break ASID Mask B Specifies whether bits in channel B break ASID7 to ASID0 (BASB7 to BASB0) which are set in BASRB are masked or not. 0: All BASRB bits are included in the break conditions and the ASID is checked 1: All BASRB bits are not included in the break conditions and the ASID is not checked
19 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	SCMFCA	0	R/W	L Bus Cycle Condition Match Flag A When the L bus cycle condition in the break conditions set for channel A is satisfied, this flag is set to 1. In order to clear this flag, write 0 into this bit. 0: The L bus cycle condition for channel A does not match 1: The L bus cycle condition for channel A matches
14	SCMFCB	0	R/W	L Bus Cycle Condition Match Flag B When the L bus cycle condition in the break conditions set for channel B is satisfied, this flag is set to 1. In order to clear this flag, write 0 into this bit. 0: The L bus cycle condition for channel B does not match 1: The L bus cycle condition for channel B matches

Bit	Bit Name	Initial Value	R/W	Description
13	SCMFDA	0	R/W	I Bus Cycle Condition Match Flag A When the I bus cycle condition in the break conditions set for channel A is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The I bus cycle condition for channel A does not match 1: The I bus cycle condition for channel A matches
12	SCMFDB	0	R/W	I Bus Cycle Condition Match Flag B When the I bus cycle condition in the break conditions set for channel B is satisfied, this flag is set to 1. In order to clear this flag, write 0 into this bit. 0: The I bus cycle condition for channel B does not match 1: The I bus cycle condition for channel B matches
11	PCTE	0	R/W	PC Trace Enable 0: Disables PC trace 1: Enables PC trace
10	PCBA	0	R/W	PC Break Select A Selects the break timing of the instruction fetch cycle for channel A as before or after instruction execution. 0: PC break of channel A is set before instruction execution 1: PC break of channel A is set after instruction execution
9, 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	DBEB	0	R/W	Data Break Enable B Selects whether or not the data bus condition is included in the break condition of channel B. 0: No data bus condition is included in the condition of channel B 1: The data bus condition is included in the condition of channel B

Bit	Bit Name	Initial Value	R/W	Description
6	PCBB	0	R/W	PC Break Select B Selects the break timing of the instruction fetch cycle for channel B as before or after instruction execution. 0: PC break of channel B is set before instruction execution 1: PC break of channel B is set after instruction execution
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	SEQ	0	R/W	Sequence Condition Select Selects two conditions of channels A and B as independent or sequential conditions. 0: Channels A and B are compared under independent conditions 1: Channels A and B are compared under sequential conditions (channel A, then channel B)
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ETBE	0	R/W	Number of Execution Times Break Enable Enables the execution-times break condition only on channel B. If this bit is 1 (break enable), a user break is issued when the number of break conditions matches with the number of execution times that is specified by BETR. 0: The execution-times break condition is disabled on channel B 1: The execution-times break condition is enabled on channel B

33.2.10 Execution Times Break Register (BETR)

BETR is a 16-bit readable/writable register. When the execution-times break condition of channel B is enabled, this register specifies the number of execution times to make the break. The maximum number is $2^{12} - 1$ times. When a break condition is satisfied, it decreases BETR. A break is issued when the break condition is satisfied after BETR becomes H'0001.

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 to 0	BET11 to BET0	All 0	R/W	Number of Execution Times

33.2.11 Branch Source Register (BRSR)

BRSR is a 32-bit read-only register. BRSR stores bits 27 to 0 in the address of the branch source instruction. BRSR has the flag bit that is set to 1 when a branch occurs. This flag bit is cleared to 0 when BRSR is read, the setting to enable PC trace is made, or BRSR is initialized by a power-on reset. Other bits are not initialized by a power-on reset. The eight BRSR registers have a queue structure and a stored register is shifted at every branch.

Bit	Bit Name	Initial Value	R/W	Description
31	SVF	0	R	BRSR Valid Flag Indicates whether the branch source address is stored. When a branch source address is fetched, this flag is set to 1. This flag is cleared to 0 by reading from BRSR. 0: The value of BRSR register is invalid 1: The value of BRSR register is valid
30 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 0	BSA27 to BSA0	—	R	Branch Source Address Store bits 27 to 0 of the branch source address.

33.2.12 Branch Destination Register (BRDR)

BRDR is a 32-bit read-only register. BRDR stores bits 27 to 0 in the address of the branch destination instruction. BRDR has the flag bit that is set to 1 when a branch occurs. This flag bit is cleared to 0 when BRDR is read, the setting to enable PC trace is made, or BRDR is initialized by a power-on reset. Other bits are not initialized by a power-on reset. The eight BRDR registers have a queue structure and a stored register is shifted at every branch.

Bit	Bit Name	Initial Value	R/W	Description
31	DVF	0	R	BRDR Valid Flag Indicates whether a branch destination address is stored. When a branch destination address is fetched, this flag is set to 1. This flag is cleared to 0 by reading BRDR. 0: The value of BRDR register is invalid 1: The value of BRDR register is valid
30 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 0	BDA27 to BDA0	—	R	Branch Destination Address Store bits 27 to 0 of the branch destination address.

33.2.13 Break ASID Register A (BASRA)

BASRA is an 8-bit readable/writable register that specifies ASID which becomes the break condition for channel A. BASRA is in CCN.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BASA7 to BASA0	—	R/W	Break ASID A Store ASID (bits 7 to 0) which is the break condition for channel A.

33.2.14 Break ASID Register B (BASRB)

BASRB is an 8-bit readable/writable register that specifies ASID which becomes the break condition for channel B. BASRB is in CCN.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BASB7 to BASB0	—	R/W	Break ASID B Store ASID (bits 7 to 0) which is the break condition for channel B.

33.3 Operation

33.3.1 Flow of the User Break Operation

The flow from setting of break conditions to user break exception processing is described below:

1. The break addresses and corresponding ASID are set in the break address registers (BARA or BARB) and break ASID registers (BASRA or BASRB in CNN). The masked addresses are set in the break address mask registers (BAMRA or BAMRB). The break data is set in the break data register (BDRB). The masked data is set in the break data mask register (BDMRB). The bus break conditions are set in the break bus cycle registers (BBRA or BBRB). Three groups of BBRA or BBRB (L bus cycle/I bus cycle select, instruction fetch/data access select, and read/write select) are each set. No user break will be generated if even one of these groups is set with 00. The respective conditions are set in the bits of the break control register (BRCR). Make sure to set all registers related to breaks before setting BBRA or BBRB.
2. When the break conditions are satisfied, the UBC sends a user break request to the CPU and sets the L bus condition match flag (SCMFCA or SCMFCB) and the I bus condition match flag (SCMFDA or SCMFDB) for the appropriate channel. When the X/Y memory bus is specified for channel B, SCMFCB is used for the condition match flag.
3. The appropriate condition match flags (SCMFCA, SCMFDA, SCMFCB, and SCMFDB) can be used to check if the set conditions match or not. The matching of the conditions sets flags, but they are not reset. 0 must first be written to them before they can be used again.
4. There is a chance that the break set in channel A and the break set in channel B occur around the same time. In this case, there will be only one break request to the CPU, but these two break channel match flags could be both set.
5. When selecting the I bus as the break condition, note the following:
   - Several bus masters, including the CPU and DMAC, are connected to the I bus. The UBC monitors bus cycles generated by all bus masters, and determines the condition match.
   - Physical addresses are used for the I bus. Set a physical address in break address registers (BARA and BARB). The bus cycles for virtual addresses issued on the L bus by the CPU are converted to physical addresses before being output to the I bus. (If the address translation function is enabled, address translation by the MMU is carried out.)
   - For data access cycles issued on the L bus by the CPU, if their virtual addresses are not to be cached, they are issued with the data size specified on the L bus and their addresses are not rounded.
   - For instruction fetch cycles issued on the L bus by the CPU, even though their virtual addresses are not to be cached, they are issued in longwords and their addresses are rounded to match longword boundaries.

— If a virtual address issued on the L bus by the CPU is an address to be cached and a cache miss occurs, its bus cycle is issued as a cache fill cycle on the I bus. In this case, it is issued in longwords and its address is rounded to match longword boundaries. However note that cache fill is not performed for a write miss in write through mode. In this case, the bus cycle is issued with the data size specified on the L bus and its address is not rounded. In write back mode, a write back cycle may be issued in addition to a read fill cycle. It is a longword bus cycle whose address is rounded to match longword boundaries.

— I bus cycles (including read fill cycles) resulting from instruction fetches on the L bus by the CPU are defined as instruction fetch cycles on the I bus, while other bus cycles are defined as data access cycles.

— The DMAC only issues data access cycles for I bus cycles.

— If a break condition is specified for the I bus, even when the condition matches in an I bus cycle resulting from an instruction executed by the CPU, at which instruction the break is to be accepted cannot be clearly defined.

6. While the block bit (BL) in the CPU status register (SR) is set to 1, no breaks can be accepted. However, condition determination will be carried out, and if the condition matches, the corresponding condition match flag is set to 1.

33.3.2 Break on Instruction Fetch Cycle

1. When L bus/instruction fetch/read/word or longword is set in the break bus cycle register (BBRA or BBRB), the break condition becomes the L bus instruction fetch cycle. Whether it breaks before or after the execution of the instruction can then be selected with the PCBA or PCBB bit in the break control register (BRCR) for the appropriate channel. If an instruction fetch cycle is set as a break condition, clear LSB in the break address register (BARA or BARB) to 0. A break cannot be generated as long as this bit is set to 1.
2. An instruction set for a break before execution breaks when it is confirmed that the instruction has been fetched and will be executed. This means this feature cannot be used on instructions fetched by overrun (instructions fetched at a branch or during an interrupt transition, but not to be executed). When this kind of break is set for the delay slot of a delayed branch instruction, the break is generated prior to execution of the delayed branch instruction.

   Note: If a branch does not occur at a delay condition branch instruction, the subsequent instruction is not recognized as a delay slot.

3. When the condition is specified to be occurred after execution, the instruction set with the break condition is executed and then the break is generated prior to the execution of the next instruction. As with pre-execution breaks, this cannot be used with overrun fetch instructions. When this kind of break is set for a delayed branch instruction and its delay slot, a break is not generated until the first instruction at the branch destination.

4. When an instruction fetch cycle is set for channel B, the break data register B (BDRB) is ignored. Therefore, break data cannot be set for the break of the instruction fetch cycle.
5. If the I bus is set for a break of an instruction fetch cycle, the condition is determined for the instruction fetch cycles on the I bus. For details, see No.5 in section 33.3.1, Flow of the User Break Operation.

33.3.3 Break on Data Access Cycle

1. If the L bus is specified as a break condition for data access break, condition comparison is performed for the virtual addresses (and data) accessed by the executed instructions, and a break occurs if the condition is satisfied. If the I bus is specified as a break condition, condition comparison is performed for the physical addresses (and data) of the data access cycles that are issued on the I bus by all bus masters including the CPU, and a break occurs if the condition is satisfied. For details on the CPU bus cycles issued on the I bus, see No.5 in section 33.3.1, Flow of the User Break Operation.
2. The relationship between the data access cycle address and the comparison condition for each operand size is listed in table 33.3.

Table 33.3 Data Access Cycle Addresses and Operand Size Comparison Conditions

Access Size	Address Compared
Longword	Compares break address register bits 31 to 2 to address bus bits 31 to 2
Word	Compares break address register bits 31 to 1 to address bus bits 31 to 1
Byte	Compares break address register bits 31 to 0 to address bus bits 31 to 0

This means that when address H'00001003 is set in the break address register (BARA or BARB), for example, the bus cycle in which the break condition is satisfied is as follows (where other conditions are met).

Longword access at H'00001000

Word access at H'00001002

Byte access at H'00001003

3. When the data value is included in the break conditions on channel B:

   When the data value is included in the break conditions, either longword, word, or byte is specified as the operand size of the break bus cycle register B (BBRB). When data values are included in break conditions, a break is generated when the address conditions and data conditions both match. To specify byte data for this case, set the same data in two bytes at bits 15 to 8 and bits 7 to 0 of the break data register B (BDRB) and break data mask register B (BDMRB). When word or byte is set, bits 31 to 16 of BDRB and BDMRB are ignored. Set the word data in bits 31 to 16 in BDRB and BDMRB when including the value of the data bus as a break condition for the MOVS.W @-As,Ds, MOVS.W @As,Ds, MOVS.W @As+,Ds, or MOVS.W @As+Ix,Ds instruction (bits 15 to 0 are ignored).
4. Access by a PREF instruction is handled as read access in longword units without access data. Therefore, if including the value of the data bus when a PREF instruction is specified as a break condition, a break will not occur.
5. If the L bus is selected, a break occurs on ending execution of the instruction that matches the break condition, and immediately before the next instruction is executed. However, when data is also specified as the break condition, the break may occur on ending execution of the instruction following the instruction that matches the break condition. If the I bus is selected, the instruction at which the break will occur cannot be determined. When this kind of break occurs at a delayed branch instruction or its delay slot, the break may not actually take place until the first instruction at the branch destination.

33.3.4 Break on X/Y-Memory Bus Cycle

1. The break condition on an X/Y-memory bus cycle is specified only in channel B. If the XYE bit in BBRB is set to 1, the break address and break data on X/Y-memory bus are selected. At this time, select the X-memory bus or Y-memory bus by specifying the XYS bit in BBRB. The break condition cannot include both X-memory and Y-memory at the same time. The break condition is applied to an X/Y-memory bus cycle by specifying L bus/data access/read or write/word or no specified operand size in the break bus cycle register B (BBRB).
2. When an X-memory address is selected as the break condition, specify an X-memory address in the upper 16 bits in BARB and BAMRB. When a Y-memory address is selected, specify a Y-memory address in the lower 16 bits. Specification of X/Y-memory data is the same for BDRB and BDMRB.
3. The timing of a data access break for the X memory or Y memory bus to occur is the same as a data access break of the L bus. For details, see No.5 in section 33.3.3, Break on Data Access Cycle.

33.3.5 Sequential Break

1. By setting the SEQ bit in BRCR to 1, the sequential break is issued when a channel B break condition matches after a channel A break condition matches. A user break is not generated even if a channel B break condition matches before a channel A break condition matches. When channels A and B conditions match at the same time, the sequential break is not issued. To clear the channel A condition match when a channel A condition match has occurred but a channel B condition match has not yet occurred in a sequential break specification, clear the SEQ bit in BRCR to 0.
2. In sequential break specification, the L/I/X/Y bus can be selected and the execution times break condition can be also specified. For example, when the execution times break condition is specified, the break condition is satisfied when a channel B condition matches with BETR = H'0001 after a channel A condition has matched.

33.3.6 Value of Saved Program Counter

When a break occurs, the address of the instruction from where execution is to be resumed is saved in the SPC, and the exception handling state is entered. If the L bus is specified as a break condition, the instruction at which the break should occur can be clearly determined (except for when data is included in the break condition). If the I bus is specified as a break condition, the instruction at which the break should occur cannot be clearly determined.

1. When instruction fetch (before instruction execution) is specified as a break condition:

   The address of the instruction that matched the break condition is saved in the SPC. The instruction that matched the condition is not executed, and the break occurs before it. However when a delay slot instruction matches the condition, the address of the delayed branch instruction is saved in the SPC.
2. When instruction fetch (after instruction execution) is specified as a break condition:

   The address of the instruction following the instruction that matched the break condition is saved in the SPC. The instruction that matches the condition is executed, and the break occurs before the next instruction is executed. However when a delayed branch instruction or delay slot matches the condition, these instructions are executed, and the branch destination address is saved in the SPC.
3. When data access (address only) is specified as a break condition:

   The address of the instruction immediately after the instruction that matched the break condition is saved in the SPC. The instruction that matches the condition is executed, and the break occurs before the next instruction is executed. However when a delay slot instruction matches the condition, the branch destination address is saved in the SPC.

4. When data access (address + data) is specified as a break condition:

   When a data value is added to the break conditions, the address of an instruction that is within two instructions of the instruction that matched the break condition is saved in the SPC. At which instruction the break occurs cannot be determined accurately.

   When a delay slot instruction matches the condition, the branch destination address is saved in the SPC. If the instruction following the instruction that matches the break condition is a branch instruction, the break may occur after the branch instruction or delay slot has finished. In this case, the branch destination address is saved in the SPC.

33.3.7 PC Trace

1. Setting PCTE in BRCR to 1 enables PC traces. When branch (branch instruction, and interrupt exception) is generated, the branch source address and branch destination address are stored in BRSR and BRDR, respectively.
2. The values stored in BRSR and BRDR are as given below due to the kind of branch.
   - If a branch occurs due to a branch instruction, the address of the branch instruction is saved in BRSR and the address of the branch destination instruction is saved in BRDR.
   - If a branch occurs due to an interrupt or exception, the value saved in SPC due to exception occurrence is saved in BRSR and the start address of the exception handling routine is saved in BRDR.

   When a repeat loop of the DSP extended function is used, control being transferred from the repeat end instruction to the repeat start instruction is not recognized as a branch, and the values are not stored in BRSR and BRDR.
3. BRSR and BRDR have eight pairs of queue structures. The top of queues is read first when the address stored in the PC trace register is read. BRSR and BRDR share the read pointer. Read BRSR and BRDR in order, the queue only shifts after BRDR is read. After switching the PCTE bit (in BRCR) off and on, the values in the queues are invalid.

33.3.8 Usage Examples

(1) Break Condition Specified for L Bus Instruction Fetch Cycle

(Example 1-1)

- Register specifications

  BARA = H'00000404, BAMRA = H'00000000, BBRA = H'0054, BARB = H'00008010, BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00300400

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00000404, Address mask: H'00000000

  Bus cycle: L bus/instruction fetch (after instruction execution)/read (operand size is not included in the condition)

  The ASID check is not included.

  <Channel B>

  Address: H'00008010, Address mask: H'00000006

  Data: H'00000000, Data mask: H'00000000

  Bus cycle: L bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

  The ASID check is not included.

  A user break occurs after an instruction of address H'00000404 is executed or before instructions of addresses H'00008010 to H'00008016 are executed.

(Example 1-2)

- Register specifications

  BARA = H'00037226, BAMRA = H'00000000, BBRA = H'0056, BARB = H'0003722E, BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000008, BASRA = H'80, BASRB = H'70

  Specified conditions: Channel A/channel B sequential mode

  <Channel A>

  Address: H'00037226, Address mask: H'00000000, ASID = H'80

  Bus cycle: L bus/instruction fetch (before instruction execution)/read/word

<Channel B>

Address: H'0003722E, Address mask: H'00000000, ASID = H'70

Data: H'00000000, Data mask: H'00000000

Bus cycle: L bus/instruction fetch (before instruction execution)/read/word

After an instruction with ASID = H'80 and address H'00037226 is executed, a user break occurs before an instruction with ASID = H'70 and address H'0003722E is executed.

(Example 1-3)

- Register specifications

  BARA = H'00027128, BAMRA = H'00000000, BBRA = H'005A, BARB = H'00031415, BAMRB = H'00000000, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00300000

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00027128, Address mask: H'00000000

  Bus cycle: L bus/instruction fetch (before instruction execution)/write/word

  The ASID check is not included.

  <Channel B>

  Address: H'00031415, Address mask: H'00000000

  Data: H'00000000, Data mask: H'00000000

  The ASID check is not included.

  Bus cycle: L bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

  On channel A, no user break occurs since instruction fetch is not a write cycle. On channel B, no user break occurs since instruction fetch is performed for an even address.

(Example 1-4)

- Register specifications

  BARA = H'00037226, BAMRA = H'00000000, BBRA = H'005A, BARB = H'0003722E, BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000008, BASRA = H'80, BASRB = H'70

  Specified conditions: Channel A/channel B sequential mode

  <Channel A>

  Address: H'00037226, Address mask: H'00000000, ASID = H'80

  Bus cycle: L bus/instruction fetch (before instruction execution)/write/word

<Channel B>

Address: H'0003722E, Address mask: H'00000000, ASID = H'70

Data: H'00000000, Data mask: H'00000000

Bus cycle: L bus/instruction fetch (before instruction execution)/read/word

Since instruction fetch is not a write cycle on channel A, a sequential condition does not match. Therefore, no user break occurs.

(Example 1-5)

- Register specifications

  BARA = H'00000500, BAMRA = H'00000000, BBRA = H'0057, BARB = H'00001000, BAMRB = H'00000000, BBRB = H'0057, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00300001, BETR = H'0005

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00000500, Address mask: H'00000000

  Bus cycle: L bus/instruction fetch (before instruction execution)/read/longword

  The ASID check is not included.

  <Channel B>

  Address: H'00001000, Address mask: H'00000000

  Data: H'00000000, Data mask: H'00000000

  Bus cycle: L bus/instruction fetch (before instruction execution)/read/longword

  The number of execution-times break enable (5 times)

  The ASID check is not included.

  On channel A, a user break occurs before an instruction of address H'00000500 is executed. On channel B, a user break occurs after the instruction of address H'00001000 are executed four times and before the fifth time.

(Example 1-6)

- Register specifications

  BARA = H'00008404, BAMRA = H'00000FFF, BBRA = H'0054, BARB = H'00008010, BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000400, BASRA = H'80, BASRB = H'70

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00008404, Address mask: H'00000FFF, ASID = H'80

Bus cycle: L bus/instruction fetch (after instruction execution)/read (operand size is not included in the condition)

<Channel B>

Address: H'00008010, Address mask: H'00000006, ASID = H'70

Data: H'00000000, Data mask: H'00000000

Bus cycle: L bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

A user break occurs after an instruction with ASID = H'80 and addresses H'00008000 to H'00008FFE is executed or before an instruction with ASID = H'70 and addresses H'00008010 to H'00008016 are executed.

Break Condition Specified for L Bus Data Access Cycle:

(Example 2-1)

- Register specifications

  BARA = H'00123456, BAMRA = H'00000000, BBRA = H'0064, BARB = H'000ABCDE, BAMRB = H'000000FF, BBRB = H'006A, BDRB = H'0000A512, BDMRB = H'00000000, BRCR = H'00000080, BASRA = H'80, BASRB = H'70

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00123456, Address mask: H'00000000, ASID = H'80

  Bus cycle: L bus/data access/read (operand size is not included in the condition)

  <Channel B>

  Address: H'000ABCDE, Address mask: H'000000FF, ASID = H'70

  Data: H'0000A512, Data mask: H'00000000

  Bus cycle: L bus/data access/write/word

  On channel A, a user break occurs with longword read from ASID = H'80 and address H'00123454, word read from address H'00123456, or byte read from address H'00123456. On channel B, a user break occurs when word H'A512 is written in ASID = H'70 and addresses H'000ABC00 to H'000ABCFE.

(Example 2-2)

- Register specifications

  BARA = H'01000000, BAMRA = H'00000000, BBRA = H'0066, BARB = H'0000F000, BAMRB = H'FFFF0000, BBRB = H'036A, BDRB = H'00004567, BDMRB = H'00000000, BRCR = H'00300080

  Specified conditions: Channel A/channel B independent mode

<Channel A>

Address: H'01000000, Address mask: H'00000000

Bus cycle: L bus/data access/read/word

The ASID check is not included.

<Channel B>

Y Address: H'0000F000, Address mask: H'FFFF0000

Data: H'00004567, Data mask: H'00000000

Bus cycle: Y bus/data access/write/word

The ASID check is not included.

On channel A, a user break occurs during word read from address H'01000000 in the memory space. On channel B, a user break occurs when word data H'4567 is written in address H'0000F000 in the Y memory space. The X/Y-memory space is changed by a mode setting.

Break Condition Specified for I Bus Data Access Cycle:

(Example 3-1)

- Register specifications

  BARA = H'00314156, BAMRA = H'00000000, BBRA = H'0094, BARB = H'00055555, BAMRB = H'00000000, BBRB = H'00A9, BDRB = H'00007878, BDMRB = H'00000F0F, BRCR = H'00000080, BASRA = H'80, BASRB = H'70

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00314156, Address mask: H'00000000, ASID = H'80

  Bus cycle: I bus/instruction fetch/read (operand size is not included in the condition)

  <Channel B>

  Address: H'00055555, Address mask: H'00000000, ASID = H'70

  Data: H'00000078, Data mask: H'0000000F

  Bus cycle: I bus/data access/write/byte

  On channel A, a user break occurs when instruction fetch is performed for ASID = H'80 and address H'00314156 in the memory space.

  On channel B, a user break occurs when ASID = H'70 and byte data H'7* is written in address H'00055555 on the I bus.

33.4 Usage Notes

1. The CPU can read from or write to the UBC registers via the I bus. Accordingly, during the period from executing an instruction to rewrite the UBC register till the new value is actually rewritten, the desired break may not occur. In order to know the timing when the UBC register is changed, read from the last written register. Instructions after then are valid for the newly written register value.
2. UBC cannot monitor access to the L bus and I bus in the same channel.
3. Note on specification of sequential break:

   A condition match occurs when a B-channel match occurs in a bus cycle after an A-channel match occurs in another bus cycle in sequential break setting. Therefore, no break occurs even if a bus cycle, in which an A-channel match and a B-channel match occur simultaneously, is set.
4. When a user break and another exception occur at the same instruction, which has higher priority is determined according to the priority levels defined in table 7.1 in section 7, Exception Handling. If an exception with higher priority occurs, the user break is not generated.
   - Pre-execution break has the highest priority.
   - When a post-execution break or data access break occurs simultaneously with a re-execution-type exception (including pre-execution break) that has higher priority, the re-execution-type exception is accepted, and the condition match flag is not set (see the exception in the following note). The break will occur and the condition match flag will be set only after the exception source of the re-execution-type exception has been cleared by the exception handling routine and re-execution of the same instruction has ended.
   - When a post-execution break or data access break occurs simultaneously with a completion-type exception (TRAPA) that has higher priority, though a break does not occur, the condition match flag is set.
5. Note the following exception for the above note.

   If a post-execution break or data access break is satisfied by an instruction that generates a CPU address error (or TLB related exception) by data access, the CPU address error (or TLB related exception) is given priority to the break. Note that the UBC condition match flag is set in this case.
6. Note the following when a break occurs in a delay slot.

   If a pre-execution break is set at the delay slot instruction of the RTE instruction, the break does not occur until the branch destination of the RTE instruction.
7. User breaks are disabled during UBC module standby mode. Do not read from or write to the UBC registers during UBC module standby mode; the values are not guaranteed.

8. When the repeat loop of the DSP extended function is used, even though a break condition is satisfied during execution of the entire repeat loop or several instructions in the repeat loop, the break may be held. For details, see section 7, Exception Handling.

Section 34 Pin Function Controller (PFC)

The pin function controller (PFC) consists of registers to select the functions and I/O directions of multiplex pins. Pin functions and I/O directions can be individually selected for every pin regardless of the LSI operating mode. Table 34.1 lists the multiplex pins of this LSI.

Note: The signals related to the SDHI should be selected only on the models that include it.

Table 34.1 Multiplexed Pins

Port	Port Function (Related Module)	Other Function (Related Module)
A	PTA7 input/output (port)	D23 input/output (BSC)
	PTA6 input/output (port)	D22 input/output (BSC)
	PTA5 input/output (port)	D21 input/output (BSC)
	PTA4 input/output (port)	D20 input/output (BSC)
	PTA3 input/output (port)	D19 input/output (BSC)
	PTA2 input/output (port)	D18 input/output (BSC)
	PTA1 input/output (port)	D17 input/output (BSC)
	PTA0 input/output (port)	D16 input/output (BSC)
B	PTB7 input/output (port)	D31 input/output (BSC)
	PTB6 input/output (port)	D30 input/output (BSC)
	PTB5 input/output (port)	D29 input/output (BSC)
	PTB4 input/output (port)	D28 input/output (BSC)
	PTB3 input/output (port)	D27 input/output (BSC)
	PTB2 input/output (port)	D26 input/output (BSC)
	PTB1 input/output (port)	D25 input/output (BSC)
	PTB0 input/output (port)	D24 input/output (BSC)
C	PTC7 input/output (port)	LCD_DATA7 output (LCDC)
	PTC6 input/output (port)	LCD_DATA6 output (LCDC)
	PTC5 input/output (port)	LCD_DATA5 output (LCDC)
	PTC4 input/output (port)	LCD_DATA4 output (LCDC)
	PTC3 input/output (port)	LCD_DATA3 output (LCDC)
	PTC2 input/output (port)	LCD_DATA2 output (LCDC)
	PTC1 input/output (port)	LCD_DATA1 output (LCDC)
	PTC0 input/output (port)	LCD_DATA0 output (LCDC)

Port	Port Function (Related Module)	Other Function (Related Module)
D	PTD7 input/output (port)/ PINT15 input (INTC)	LCD_DATA15 output (LCDC)
	PTD6 input/output (port)/ PINT14 input (INTC)	LCD_DATA14 output (LCDC)
	PTD5 input/output (port)/ PINT13 input (INTC)	LCD_DATA13 output (LCDC)
	PTD4 input/output (port)/ PINT12 input (INTC)	LCD_DATA12 output (LCDC)
	PTD3 input/output (port)	LCD_DATA11 output (LCDC)
	PTD2 input/output (port)	LCD_DATA10 output (LCDC)
	PTD1 input/output (port)	LCD_DATA9 output (LCDC)
	PTD0 input/output (port)	LCD_DATA8 output (LCDC)
E	PTE6 input (port)	AFE_RXIN input (AFEIF)/IIC_SCL input/output (IIC)
	PTE5 input (port)	$\overline{\text{AFE\_RDET}}$ input (AFEIF)/IIC_SDA input/output (IIC)
	PTE4 input/output (port)	LCD_M_DISP output (LCDC)
	PTE3 input/output (port)	LCD_CL1 output (LCDC)
	PTE2 input/output (port)	LCD_CL2 output (LCDC)
	PTE1 input/output (port)	LCD_DON output (LCDC)
	PTE0 input/output (port)	LCD_FLM output (LCDC)
F	PTF6 input (port)	DA1 output (DAC)
	PTF5 input (port)	DA0 output (DAC)
	PTF4 input (port)	AN3 input (ADC)
	PTF3 input (port)	AN2 input (ADC)
	PTF2 input (port)	AN1 input (ADC)
	PTF1 input (port)	AN0 input (ADC)
	PTF0 input (port)	$\overline{\text{ADTRG}}$ input (ADC)
G	PTG6 input/output (port)	USB1d_RCV input (USB)/IRQ5 input (INTC)/ AFE_FS input (AFEIF)/$\overline{\text{PCC\_REG}}$ output (PCC)
	PTG5 input/output (port)	USB1d_TXSE0 output (USB)/IRQ4 input (INTC)/ AFE_TXOUT output (AFEIF)/$\overline{\text{PCC\_DRV}}$ output (PCC)
	PTG4 input/output (port)	USB1d_TXDPLS output (USB)/ AFE_SCLK input (AFEIF)/$\overline{\text{IOIS16}}$ input (BSC)/ $\overline{\text{PCC\_IOIS16}}$ input (PCC)

Port	Port Function (Related Module)	Other Function (Related Module)
G	PTG3 input/output (port)/PINT11 input (INTC)	USB1d_DMNS input (USB)/ AFE_RLYCNT output (AFEIF)/PCC_BVD2 input (PCC)
	PTG2 input/output (port)/PINT10 input (INTC)	USB1d_DPLS input (USB)/AFE_HC1 output (AFEIF)/ PCC_BVD1 input (PCC)
	PTG1 input/output (port)/PINT9 input (INTC)	USB1d_SPEED output (USB)/$\overline{\text{PCC\_CD2}}$ input (PCC)
	PTG0 input/output (port)/PINT8 input (INTC)	USB1d_TXENL output (USB)/$\overline{\text{PCC\_CD1}}$ input (PCC)
H	PTH6 input/output (port)	$\overline{\text{RAS}}$ output (BSC)
	PTH5 input/output (port)	$\overline{\text{CAS}}$ output (BSC)
	PTH4 input/output (port)	CKE output (BSC)
	PTH3 input/output (port)	STATUS1 output
	PTH2 input/output (port)	STATUS0 output
	PTH1 input/output (port)	USB2_pwr_en output (USB)
	PTH0 input/output (port)	USB1_pwr_en output (USB)/USBF_UPLUP (USB)
J	PTJ6 input/output (port)	AUDCK output (HUDI)
	PTJ5 input/output (port)	$\overline{\text{ASEBRKAK}}$ output (HUDI)
	PTJ4 input/output (port)	AUDATA3 output (HUDI)
	PTJ3 input/output (port)	AUDATA2 output (HUDI)
	PTJ2 input/output (port)	AUDATA1 output (HUDI)
	PTJ1 input/output (port)	AUDATA0 output (HUDI)
	PTJ0 input/output (port)	$\overline{\text{AUDSYNC}}$ output (HUDI)
K	PTK3 input/output (port)/PINT7 input (INTC)	PCC_RESET output (PCC)
	PTK2 input/output (port)/PINT6 input (INTC)	PCC_RDY input (PCC)
	PTK1 input/output (port)/PINT5 input (INTC)	$\overline{\text{PCC\_VS2}}$ input (PCC)
	PTK0 input/output (port)/PINT4 input (INTC)	$\overline{\text{PCC\_VS1}}$ input (PCC)
L	PTL7 input/output (port)	$\overline{\text{TRST}}$ input (HUDI)
	PTL6 input/output (port)	TMS input (HUDI)
	PTL5 input/output (port)	TDO output (HUDI)
	PTL4 input/output (port)	TDI input (HUDI)
	PTL3 input/output (port)	TCK input (HUDI)
M	PTM7 input/output (port)	$\overline{\text{DREQ1}}$ input (DMAC)
	PTM6 input/output (port)/PINT0 input (INTC)	$\overline{\text{DREQ0}}$ input (DMAC)

Port	Port Function (Related Module)	Other Function (Related Module)
M	PTM5 input/output (port)	$\overline{\text{DACK1}}$ output (DMAC)
	PTM4 input/output (port)/PINT1 input (INTC)	$\overline{\text{DACK0}}$ output (DMAC)
	PTM3 input/output (port)/PINT3 input (INTC)	TEND1 output (DMAC)
	PTM2 input/output (port)/PINT2 input (INTC)	TEND0 output (DMAC)
	PTM1 input/output (port)	$\overline{\text{CS5B}}$ output (BSC)/$\overline{\text{CE1A}}$ output (BSC)
	PTM0 input/output (port)	$\overline{\text{CS6B}}$ output (BSC)/$\overline{\text{CE1B}}$ output (BSC)
P	PTP4 input/output (port)	USB1d_SUSPEND output (USB)/ $\overline{\text{REFOUT}}$ output (BSC)/$\overline{\text{IRQOUT}}$ output (BSC)
	PTP3 input/output (port)	IRQ3 input (INTC)/$\overline{\text{IRL3}}$ input (INTC)
	PTP2 input/output (port)	IRQ2 input (INTC)/$\overline{\text{IRL2}}$ input (INTC)
	PTP1 input/output (port)	IRQ1 input (INTC)/$\overline{\text{IRL1}}$ input (INTC)
	PTP0 input/output (port)	IRQ0 input (INTC)/$\overline{\text{IRL0}}$ input (INTC)
R	PTR7 input/output (port)	A25 output (BSC)
	PTR6 input/output (port)	A24 output (BSC)
	PTR5 input/output (port)	A23 output (BSC)
	PTR4 input/output (port)	A22 output (BSC)
	PTR3 input/output (port)	A21 output (BSC)
	PTR2 input/output (port)	A20 output (BSC)
	PTR1 input/output (port)	A19 output (BSC)
	PTR0 input/output (port)	A0 output (BSC)
S	PTS4 input/output (port)	SIOF0_SYNC input/output (SIOF)
	PTS3 input/output (port)	SIOF0_MCLK input (SIOF)
	PTS2 input/output (port)	SIOF0_TxD output (SIOF)
	PTS1 input/output (port)	SIOF0_RxD input (SIOF)
	PTS0 input/output (port)	SIOF0_SCK input/output (SIOF)
T	PTT4 input/output (port)	$\overline{\text{SCIF0\_CTS}}$ input (SCIF)/TPU_TO1 output (TPU)
	PTT3 input/output (port)	$\overline{\text{SCIF0\_RTS}}$ output (SCIF)/TPU_TO0 output (TPU)
	PTT2 input/output (port)	SCIF0_TxD output (SCIF)/IrTX output (IrDA)
	PTT1 input/output (port)	SCIF0_RxD input (SCIF)/IrRX input (IrDA)
	PTT0 input/output (port)	SCIF0_SCK input/output (SCIF)

Port	Port Function (Related Module)	Other Function (Related Module)
U	PTU4 input/output (port)	SIOF1_SYNC input/output (SIOF)/SD_DAT2 input/output (SDHI)
	PTU3 input/output (port)	SIOF1_MCLK input (SIOF)/SD_DAT1 input/output (SDHI)/ TPU_TI3B input (TPU)
	PTU2 input/output (port)	MMC_DAT input/output (MMC)/ SIOF1_TxD output (SIOF)/SD_DAT0 input/output (SDHI)/ TPU_TI3A input (TPU)
	PTU1 input/output (port)	MMC_CMD input/output (MMC)/ SIOF1_RxD input (SIOF)/SD_CMD input/output (SDHI)/ TPU_TI2B input (TPU)
	PTU0 input/output (port)	MMC_CLK output (MMC)/ SIOF1_SCK input/output (SIOF)/SD_CLK output (SDHI)/ TPU_TI2A input (TPU)
V	PTV4 input/output (port)	MMC_VDDON output (MMC)/$\overline{\text{SCIF1\_CTS}}$ input (SCIF)/ LCD_VEPWC output (LCDC)/TPU_TO3 output (TPU)
	PTV3 input/output (port)	$\overline{\text{MMC\_ODMOD}}$ output (MMC)/$\overline{\text{SCIF1\_RTS}}$ output (SCIF)/ LCD_VCPWC output (LCDC)/TPU_TO2 output (TPU)
	PTV2 input/output (port)	SIM_D input/output (SIM)/SCIF1_TxD output (SCIF)/ SD_CD input (SDHI)
	PTV1 input/output (port)	SIM_RST output (SIM)/SCIF1_RxD input (SCIF)/ SD_WP input (SDHI)
	PTV0 input/output (port)	SIM_CLK output (SIM)/SCIF1_SCK input/output (SCIF)/ SD_DAT3 input/output (SDHI)

34.1 Register Descriptions

The PFC has the following registers. Refer to section 37, List of Registers, for more details on the addresses and access size of these registers.

- Port A control register (PACR)
- Port B control register (PBCR)
- Port C control register (PCCR)
- Port D control register (PDCR)
- Port E control register (PECR)
- Port F control register (PFCR)
- Port G control register (PGCR)
- Port H control register (PHCR)
- Port J control register (PJCR)
- Port K control register (PKCR)
- Port L control register (PLCR)
- Port M control register (PMCR)
- Port P control register (PPCR)
- Port R control register (PRCR)
- Port S control register (PSCR)
- Port T control register (PTCR)
- Port U control register (PUCR)
- Port V control register (PVCR)
- Pin select register A (PSELA)
- Pin select register B (PSELB)
- Pin select register C (PSELC)
- Pin select register D (PSELD)

34.1.1 Port A Control Register (PACR)

PACR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15	PA7MD1	0	R/W	PA7 Mode
14	PA7MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Port output
				10: Port input (Pull-up MOS: On)
				11: Port input (Pull-up MOS: Off)
13	PA6MD1	0	R/W	PA6 Mode
12	PA6MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Port output
				10: Port input (Pull-up MOS: On)
				11: Port input (Pull-up MOS: Off)
11	PA5MD1	0	R/W	PA5 Mode
10	PA5MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Port output
				10: Port input (Pull-up MOS: On)
				11: Port input (Pull-up MOS: Off)
9	PA4MD1	0	R/W	PA4 Mode
8	PA4MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Port output
				10: Port input (Pull-up MOS: On)
				11: Port input (Pull-up MOS: Off)
7	PA3MD1	0	R/W	PA3 Mode
6	PA3MD0	0	R/W	00: Other functions (See table 34.1.)
				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	PA2MD1	0	R/W	PA2 Mode
4	PA2MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3	PA1MD1	0	R/W	PA1 Mode
2	PA1MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1	PA0MD1	0	R/W	PA0 Mode
0	PA0MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

34.1.2 Port B Control Register (PBCR)

PBCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15	PB7MD1	0	R/W	PB7 Mode
14	PB7MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
13	PB6MD1	0	R/W	PB6 Mode
12	PB6MD0	0	R/W	00: Other functions (See table 34.1.) 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
11	PB5MD1	0	R/W	PB5 Mode
10	PB5MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
9	PB4MD1	0	R/W	PB4 Mode
8	PB4MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7	PB3MD1	0	R/W	PB3 Mode
6	PB3MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5	PB2MD1	0	R/W	PB2 Mode
4	PB2MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3	PB1MD1	0	R/W	PB1 Mode
2	PB1MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1	PB0MD1	0	R/W	PB0 Mode
0	PB0MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

34.1.3 Port C Control Register (PCCR)

PCCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15	PC7MD1	1	R/W	PC7 Mode
14	PC7MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
13	PC6MD1	1	R/W	PC6 Mode
12	PC6MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
11	PC5MD1	1	R/W	PC5 Mode
10	PC5MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
9	PC4MD1	1	R/W	PC4 Mode
8	PC4MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7	PC3MD1	1	R/W	PC3 Mode
6	PC3MD0	0	R/W	00: Other functions (See table 34.1.) 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	PC2MD1	1	R/W	PC2 Mode
4	PC2MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3	PC1MD1	1	R/W	PC1 Mode
2	PC1MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1	PC0MD1	1	R/W	PC0 Mode
0	PC0MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

34.1.4 Port D Control Register (PDCR)

PDCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15	PD7MD1	1	R/W	PD7 Mode
14	PD7MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
13	PD6MD1	1	R/W	PD6 Mode
12	PD6MD0	0	R/W	00: Other functions (See table 34.1.) 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
11	PD5MD1	1	R/W	PD5 Mode
10	PD5MD0	0	R/W	00: Other functions (See table 34.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
9	PD4MD1	1	R/W	PD4 Mode
8	PD4MD0	0	R/W	00: Other functions (See table 34.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7	PD3MD1	1	R/W	PD3 Mode
6	PD3MD0	0	R/W	00: Other functions (See table 34.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5	PD2MD1	1	R/W	PD2 Mode
4	PD2MD0	0	R/W	00: Other functions (See table 34.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3	PD1MD1	1	R/W	PD1 Mode
2	PD1MD0	0	R/W	00: Other functions (See table 34.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1	PD0MD1	1	R/W	PD0 Mode
0	PD0MD0	0	R/W	00: Other functions (See table 34.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

34.1.5 Port E Control Register (PECR)

PECR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	PE6MD1	1	R/W	PE6 Mode 0: Other functions (See table 34.1.) 1: Port input (Pull-up MOS: Off)
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11	PE5MD1	1	R/W	PE5 Mode 0: Other functions (See table 34.1.) 1: Port input (Pull-up MOS: Off)
10	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
9	PE4MD1	1	R/W	PE4 Mode
8	PE4MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7	PE3MD1	1	R/W	PE3 Mode
6	PE3MD0	0	R/W	00: Other functions (See table 34.1.) 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	PE2MD1	1	R/W	PE2 Mode
4	PE2MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3	PE1MD1	1	R/W	PE1 Mode
2	PE1MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1	PE0MD1	1	R/W	PE0 Mode
0	PE0MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

34.1.6 Port F Control Register (PFCR)

PFCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	PF6MD1	0	R/W	PF6 Mode
12	PF6MD0	0	R/W	00: Other functions (See table 34.1.) 01: Reserved 1X: Port input (Pull-up MOS: Off)

Bit	Bit Name	Initial Value	R/W	Description
11	PF5MD1	0	R/W	PF5 Mode
10	PF5MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Reserved
				1X: Port input (Pull-up MOS: Off)
9	PF4MD1	0	R/W	PF4 Mode
8	PF4MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Reserved
				1X: Port input (Pull-up MOS: Off)
7	PF3MD1	0	R/W	PF3 Mode
6	PF3MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Reserved
				1X: Port input (Pull-up MOS: Off)
5	PF2MD1	0	R/W	PF2 Mode
4	PF2MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Reserved
				1X: Port input (Pull-up MOS: Off)
3	PF1MD1	0	R/W	PF1 Mode
2	PF1MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Reserved
				1X: Port input (Pull-up MOS: Off)
1	PF0MD1	1	R/W	PF0 Mode
0	PF0MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Reserved
				10: Port input (Pull-up MOS: On)
				11: Port input (Pull-up MOS: Off)

Note: X: Don't care

34.1.7 Port G Control Register (PGCR)

PGCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	PG6MD1	1	R/W	PG6 Mode
12	PG6MD0	1	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
11	PG5MD1	1	R/W	PG5 Mode
10	PG5MD0	1	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
9	PG4MD1	1	R/W	PG4 Mode
8	PG4MD0	1	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7	PG3MD1	1	R/W	PG3 Mode
6	PG3MD0	1	R/W	00: Other functions (See table 34.1) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5	PG2MD1	1	R/W	PG2 Mode
4	PG2MD0	1	R/W	00: Other functions (See table 34.1.) 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	PG1MD1	1	R/W	PG1 Mode
2	PG1MD0	1	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1	PG0MD1	1	R/W	PG0 Mode
0	PG0MD0	1	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

34.1.8 Port H Control Register (PHCR)

PHCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	PH6MD1	0	R/W	PH6 Mode
12	PH6MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
11	PH5MD1	0	R/W	PH5 Mode
10	PH5MD0	0	R/W	00: Other functions (See table 34.1.) 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	PH4MD1	0	R/W	PH4 Mode
8	PH4MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Port output
				10: Port input (Pull-up MOS: On)
				11: Port input (Pull-up MOS: Off)
7	PH3MD1	0	R/W	PH3 Mode
6	PH3MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Port output
				10: Port input (Pull-up MOS: On)
				11: Port input (Pull-up MOS: Off)
5	PH2MD1	0	R/W	PH2 Mode
4	PH2MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Port output
				10: Port input (Pull-up MOS: On)
				11: Port input (Pull-up MOS: Off)
3	PH1MD1	1	R/W	PH1 Mode
2	PH1MD0	1	R/W	00: Other functions (See table 34.1.)
				01: Port output
				10: Port input (Pull-up MOS: On)
				11: Port input (Pull-up MOS: Off)
1	PH0MD1	1	R/W	PH0 Mode
0	PH0MD0	1	R/W	00: Other functions (See table 34.1.)
				01: Port output
				10: Port input (Pull-up MOS: On)
				11: Port input (Pull-up MOS: Off)

34.1.9 Port J Control Register (PJCR)

PJCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	PJ6MD1	0/1*	R/W	PJ6 Mode
12	PJ6MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
11	PJ5MD1	0/1*	R/W	PJ5 Mode
10	PJ5MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
9	PJ4MD1	0/1*	R/W	PJ4 Mode
8	PJ4MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7	PJ3MD1	0/1*	R/W	PJ3 Mode
6	PJ3MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5	PJ2MD1	0/1*	R/W	PJ2 Mode
4	PJ2MD0	0	R/W	00: Other functions (See table 34.1.) 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	PJ1MD1	0/1*	R/W	PJ1 Mode
2	PJ1MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1	PJ0MD1	0/1*	R/W	PJ0 Mode
0	PJ0MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

Note: * When $\overline{\text{ASEMD0}}$ = 1 (normal mode) at power-on reset, initial value is 1. When $\overline{\text{ASEMD0}}$ = 0 (ASE mode) at power-on reset, initial value is 0.

34.1.10 Port K Control Register (PKCR)

PKCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	PK3MD1	1	R/W	PK3 Mode
6	PK3MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5	PK2MD1	1	R/W	PK2 Mode
4	PK2MD0	0	R/W	00: Other functions (See table 34.1.) 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	PK1MD1	1	R/W	PK1 Mode
2	PK1MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1	PK0MD1	1	R/W	PK0 Mode
0	PK0MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

34.1.11 Port L Control Register (PLCR)

PLCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15	PL7MD1	0	R/W	PL7 Mode
14	PL7MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
13	PL6MD1	0	R/W	PL6 Mode
12	PL6MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
11	PL5MD1	0	R/W	PL5 Mode
10	PL5MD0	0	R/W	00: Other functions (See table 34.1.) 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	PL4MD1	0	R/W	PL4 Mode
8	PL4MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7	PL3MD1	0	R/W	PL3 Mode
6	PL3MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

34.1.12 Port M Control Register (PMCR)

PMCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15	PM7MD1	1	R/W	PM7 Mode
14	PM7MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
13	PM6MD1	1	R/W	PM6 Mode
12	PM6MD0	0	R/W	00: Other functions (See table 34.1.) 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
11	PM5MD1	1	R/W	PM5 Mode
10	PM5MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
9	PM4MD1	1	R/W	PM4 Mode
8	PM4MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7	PM3MD1	1	R/W	PM3 Mode
6	PM3MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5	PM2MD1	1	R/W	PM2 Mode
4	PM2MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3	PM1MD1	0	R/W	PM1 Mode
2	PM1MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1	PM0MD1	0	R/W	PM0 Mode
0	PM0MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

34.1.13 Port P Control Register (PPCR)

PPCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	PP4MD1	1	R/W	PP4 Mode
8	PP4MD0	1	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7	PP3MD1	1	R/W	PP3 Mode
6	PP3MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5	PP2MD1	1	R/W	PP2 Mode
4	PP2MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3	PP1MD1	1	R/W	PP1 Mode
2	PP1MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1	PP0MD1	1	R/W	PP0 Mode
0	PP0MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

34.1.14 Port R Control Register (PRCR)

PRCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15	PR7MD1	0	R/W	PR7 Mode
14	PR7MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
13	PR6MD1	0	R/W	PR6 Mode
12	PR6MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
11	PR5MD1	0	R/W	PR5 Mode
10	PR5MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
9	PR4MD1	0	R/W	PR4 Mode
8	PR4MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7	PR3MD1	0	R/W	PR3 Mode
6	PR3MD0	0	R/W	00: Other functions (See table 34.1.) 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	PR2MD1	0	R/W	PR2 Mode
4	PR2MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Port output
				10: Port input (Pull-up MOS: On)
				11: Port input (Pull-up MOS: Off)
3	PR1MD1	0	R/W	PR1 Mode
2	PR1MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Port output
				10: Port input (Pull-up MOS: On)
				11: Port input (Pull-up MOS: Off)
1	PR0MD1	0	R/W	PR0 Mode
0	PR0MD0	0	R/W	00: Other functions (See table 34.1.)
				01: Port output
				10: Port input (Pull-up MOS: On)
				11: Port input (Pull-up MOS: Off)

34.1.15 Port S Control Register (PSCR)

PSCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	PS4MD1	1	R/W	PS4 Mode
8	PS4MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7	PS3MD1	1	R/W	PS3 Mode
6	PS3MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5	PS2MD1	1	R/W	PS2 Mode
4	PS2MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3	PS1MD1	1	R/W	PS1 Mode
2	PS1MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1	PS0MD1	1	R/W	PS0 Mode
0	PS0MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

34.1.16 Port T Control Register (PTCR)

PTCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	PT4MD1	1	R/W	PT4 Mode
8	PT4MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7	PT3MD1	1	R/W	PT3 Mode
6	PT3MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5	PT2MD1	1	R/W	PT2 Mode
4	PT2MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3	PT1MD1	1	R/W	PT1 Mode
2	PT1MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1	PT0MD1	1	R/W	PT0 Mode
0	PT0MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

34.1.17 Port U Control Register (PUCR)

PUCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	PU4MD1	1	R/W	PU4 Mode
8	PU4MD0	1	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7	PU3MD1	1	R/W	PU3 Mode
6	PU3MD0	1	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5	PU2MD1	1	R/W	PU2 Mode
4	PU2MD0	1	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3	PU1MD1	1	R/W	PU1 Mode
2	PU1MD0	1	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1	PU0MD1	1	R/W	PU0 Mode
0	PU0MD0	1	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

34.1.18 Port V Control Register (PVCR)

PVCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	PV4MD1	0	R/W	PV4 Mode
8	PV4MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
7	PV3MD1	0	R/W	PV3 Mode
6	PV3MD0	0	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
5	PV2MD1	1	R/W	PV2 Mode
4	PV2MD0	1	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
3	PV1MD1	1	R/W	PV1 Mode
2	PV1MD0	1	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)
1	PV0MD1	1	R/W	PV0 Mode
0	PV0MD0	1	R/W	00: Other functions (See table 34.1.) 01: Port output 10: Port input (Pull-up MOS: On) 11: Port input (Pull-up MOS: Off)

34.1.19 Pin Select Register A (PSELA)

PSELA is a 16-bit readable/writable register that selects the pin functions multiplexing two or more other functions.

To use one of other functions in the pin multiplexing two or more other functions, the port control register should be set as other functions after setting the corresponding bit in PSELA.

Bit	Bit Name	Initial Value	R/W	Description
15	PSELA15	1	R/W	USB1d_TXENL/$\overline{\text{PCC\_CD1}}$ Select as PTG0 Other Functions
14	PSELA14	0	R/W	00: Select USB1d_TXENL 01: Reserved 10: Select $\overline{\text{PCC\_CD1}}$ 11: Reserved
13	PSELA13	1	R/W	USB1d_SPEED/$\overline{\text{PCC\_CD2}}$ Select as PTG1 Other Functions
12	PSELA12	0	R/W	00: Select USB1d_SPEED 01: Reserved 10: Select $\overline{\text{PCC\_CD2}}$ 11: Reserved
11	PSELA11	1	R/W	USB1d_DPLS/AFE_HC1/PCC_BVD1 Select as PTG2 Other Functions
10	PSELA10	0	R/W	00: Select USB1d_DPLS 01: Select AFE_HC1 10: Select PCC_BVD1 11: Reserved
9	PSELA9	1	R/W	USB1d_DMNS/AFE_RLYCNT/PCC_BVD2 Select as PTG3 Other Functions
8	PSELA8	0	R/W	00: Select USB1d_DMNS 01: Select AFE_RLYCNT 10: Select PCC_BVD2 11: Reserved

Bit	Bit Name	Initial Value	R/W	Description
7	PSELA7	1	R/W	USB1d_TXDPLS/AFE_SCLK/$\overline{\text{IOIS16}}$/$\overline{\text{PCC\_IOIS16}}$ Select as PTG4 Other Functions
6	PSELA6	0	R/W	00: Select USB1d_TXDPLS 01: Select AFE_SCLK 10: Select $\overline{\text{IOIS16}}$/$\overline{\text{PCC\_IOIS16}}$ 11: Reserved
5	PSELA5	1	R/W	USB1d_TXSE0/AFE_TXOUT/$\overline{\text{PCC\_DRV}}$ /IRQ4 Select as PTG5 Other Functions
4	PSELA4	0	R/W	00: Select USB1d_TXSE0 01: Select AFE_TXOUT 10: Select $\overline{\text{PCC\_DRV}}$ 11: Select IRQ4
3	PSELA3	1	R/W	USB1d_RCV/AFE_FS/$\overline{\text{PCC\_REG}}$ /IRQ5 Select as PTG5 Other Functions
2	PSELA2	0	R/W	00: Select USB1d_RCV 01: Select AFE_FS 10: Select $\overline{\text{PCC\_REG}}$ 11: Select IRQ5
1	PSELA1	0	R/W	USB1d_SUSPEND/$\overline{\text{REFOUT}}$/$\overline{\text{IRQOUT}}$ Select as PTP4 Other Functions
0	PSELA0	0	R/W	00: Select USB1d_SUSPEND 01: Select $\overline{\text{REFOUT}}$/$\overline{\text{IRQOUT}}$; select $\overline{\text{REFOUT}}$ as $\overline{\text{REFOUT}}$/$\overline{\text{IRQOUT}}$ output source 10: Select $\overline{\text{REFOUT}}$/$\overline{\text{IRQOUT}}$; select $\overline{\text{IRQOUT}}$ as $\overline{\text{REFOUT}}$/$\overline{\text{IRQOUT}}$ output source 11: Select $\overline{\text{REFOUT}}$/$\overline{\text{IRQOUT}}$; select OR of $\overline{\text{REFOUT}}$ and $\overline{\text{IRQOUT}}$ as $\overline{\text{REFOUT}}$/$\overline{\text{IRQOUT}}$ output source

34.1.20 Pin Select Register B (PSELB)

PSELB is a 16-bit readable/writable register that selects the pin functions multiplexing two or more other functions.

To use one of other functions in the pin multiplexing two or more other functions, the port control register should be set as other functions after setting the corresponding bit in PSELB.

Bit	Bit Name	Initial Value	R/W	Description
15	PSELB15	0	R/W	$\overline{\text{SCIF0\_RTS}}$/TPU_TO0 Select as PTT3 Other Functions 0: Select $\overline{\text{SCIF0\_RTS}}$ 1: Select TPU_TO0
14	PSELB14	0	R/W	$\overline{\text{SCIF0\_CTS}}$/TPU_TO1 Select as PTT4 Other Functions 0: Select $\overline{\text{SCIF0\_CTS}}$ 1: Select TPU_TO1
13 12	PSELB13 PSELB12	1 1	R/W R/W	$\overline{\text{MMC\_ODMOD}}$/$\overline{\text{SCIF1\_RTS}}$/LCD_VCPWC/TPU_TO2 Select as PTV3 Other Functions 00: Select $\overline{\text{SCIF1\_RTS}}$ 01: Select TPU_TO2 10: Select $\overline{\text{MMC\_ODMOD}}$ 11: Select LCD_VCPWC
11 10	PSELB11 PSELB10	1 1	R/W R/W	MMC_VDDON/$\overline{\text{SCIF1\_CTS}}$/LCD_VEPWC/TPU_TO3 Select as PTV4 Other Functions 00: Select $\overline{\text{SCIF1\_CTS}}$ 01: Select TPU_TO3 10: Select MMC_VDDON 11: Select LCD_VEPWC
9	PSELB9	0	R/W	$\overline{\text{AFE\_RDET}}$/IIC_SDA Select as PTE5 Other Functions 0: Select IIC_SDA 1: Select $\overline{\text{AFE\_RDET}}$
8	PSELB8	0	R/W	AFE_RXIN/IIC_SCL Select as PTE6 Other Functions 0: Select IIC_SCL 1: Select AFE_RXIN
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	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PSELB0	0	R/W	SD Host Interface Select as PTU4 to PTU0 and PTV2 to PTV0 Other Functions 0: Not select SD host interface 1: Select SD host interface

34.1.21 Pin Select Register C (PSELC)

PSELC is a 16-bit readable/writable register that selects the pin functions multiplexing two or more other functions.

To use one of other functions in the pin multiplexing two or more other functions, the port control register should be set as other functions after setting the corresponding bit in PSELC.

Bit	Bit Name	Initial Value	R/W	Description
15	PSELC15	0	R/W	MMC_CLK/SIOF1_SCK/SD_CLK/TPU_TI2A Select as PTU0 Other Functions 00: Select SIOF1_SCK 01: Select TPU_TI2A 10: Select MMC_CLK 11: Select according to PSELB0 setting Reserved when PSELB0 = 0 Select SD_CLK when PSELB0 = 1
14	PSELC14	0	R/W	
13	PSELC13	0	R/W	MMC_CMD/SIOF1_RxD/SD_CMD/TPU_TI2B Select as PTU1 Other Functions 00: Select SIOF1_RxD 01: Select TPU_TI2B 10: Select MMC_CMD 11: Select according to PSELB0 setting Reserved when PSELB0 = 0 Select SD_CMD when PSELB0 = 1
12	PSELC12	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
11	PSELC11	0	R/W	SIM_RST/SCIF1_RxD/SD_WP Select as PTV1 Other Functions 00: Select SCIF1_RxD 01: Reserved 10: Select SIM_RST 11: Select according to PSELB0 setting Reserved when PSELB0 = 0 Select SD_WP when PSELB0 = 1
10	PSELC10	0	R/W	
9	PSELC9	0	R/W	SIM_D/SCIF1_TxD/SD_CD Select as PTV2 Other Functions 00: Select SCIF1_TxD 01: Reserved 10: Select SIM_D 11: Select according to PSELB0 setting Reserved when PSELB0 = 0 Select SD_CD when PSELB0 = 1
8	PSELC8	0	R/W	
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

34.1.22 Pin Select Register D (PSELD)

PSELD is a 16-bit readable/writable register that selects the pin functions multiplexing two or more other functions.

To use one of other functions in the pin multiplexing two or more other functions, the port control register should be set as other functions after setting the corresponding bit in PSELD.

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 13	PSELD14 PSELD13	0 0	R/W R/W	MMC_DAT/SIOF1_TxD/SD_DAT0/TPU_TI3A Select as PTU2 Other Functions 00: Select SIOF1_TxD 01: Select TPU_TI3A 10: Select MMC_DAT 11: Select according to PSELB0 setting Reserved when PSELB0 = 0 Select SD_DAT0 when PSELB0 = 1
12	PSELD12	0	R/W	Pin SD_DAT0 Control when PSELD[14:13] = B'11 0: Pins are not controlled 1: Pins are controlled SD_DAT0: Pulled up
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	PSELD10	0	R/W	SIOF1_MCLK/SD_DAT1/TPU_TI3B Select as PTU3
9	PSELD9	0	R/W	Other Functions 00: Select SIOF1_MCLK 01: Select TPU_TI3B 10: Reserved 11: Select according to PSELB0 setting Reserved when PSELB0 = 0 Select SD_DAT1 when PSELB0 = 1
8	PSELD8	0	R/W	Pin SD_DAT1 Control when PSELD[10:9] = B'11 0: Pins are not controlled 1: Pins are controlled SD_DAT1: Pulled up
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	PSELD6	0	R/W	SIOF1_SYNC/SD_DAT2 Select as PTU4 Other
5	PSELD5	0	R/W	Functions 00: Select SIOF1_SYNC 01: Reserved 10: Reserved 11: Select according to PSELB0 setting Reserved when PSELB0 = 0 Select SD_DAT2 when PSELB0 = 1
4	PSELD4	0	R/W	Pin SD_DAT2 Control when PSELD[6:5] = B'11 0: Pins are not controlled 1: Pins are controlled SD_DAT2: Pulled up
3	—	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
2	PSELD2	0	R/W	SIM_CLK/SCIF1_SCK/SD_DAT3 Select as PTV0
1	PSELD1	0	R/W	Other Functions 00: Select SCIF1_SCK 01: Reserved 10: Select SIM_CLK 11: Select according to PSELB0 setting Reserved when PSELB0 = 0 Select SD_DAT3 when PSELB0 = 1
0	PSELD0	0	R/W	Pin SD_DAT3 Control when PSELD[2:1] = B'11 0: Pins are not controlled 1: Pins are controlled SD_DAT3: Pulled up

34.1.23 USB Transceiver Control Register (UTRCTL)

UTRCTL controls 1.8 V/3.3 V I/O buffer drivability.

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved
8	DRV	0	R/W	I/O Buffer Drive Control 0: 1.8 V/3.3 V I/O buffer high drivability 1: 1.8 V/3.3 V I/O buffer low drivability Power supply pin, VccQ1, can be applied 1.65 to 1.95 V or 2.7 to 3.6 V. When 1.65 to 1.95 V is applied to VccQ1, setting the drivability high (DRV = 0) is recommended. When 2.7 to 3.6 V is applied to VccQ1, setting the drivability low (DRV = 1) is recommended.
7 to 2	—	All 0	R/W	Reserved
1	USB_ TRANS	0	R/W	See section 23, USB Pin Multiplex Controller.
0	USB_SEL	1	R/W	

Section 35 I/O Ports

This LSI has 18 I/O ports (ports A to H, J to M, and R to V). All I/O port pins are multiplexed with other pin functions (the pin function controller (PFC) handles the selection of pin functions and pull-up MOS control). Each I/O port has a data register, which stores data for the pins.

35.1 Port A

Port A is an input/output port with the pin configuration shown in figure 35.1. Each pin has an input pull-up MOS, which is controlled by the port A control register (PACR) in the PFC.

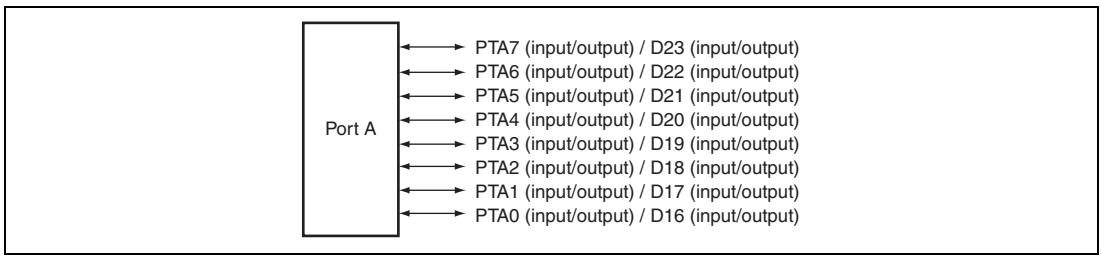


Figure 35.1 Port A

35.1.1 Register Description

Port A has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port A data register (PADR)

35.1.2 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. When the pin function is general output port, if the port is read, the value of the corresponding PADR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DT	0	R/W	Table 35.1 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 35.1 Port A Data Register (PADR) Read/Write Operations

PACR State				
PAnMD1	PAnMD0	Pin State	Read	Write
0	0	Other function	PADR value	Value is written to PADR, but does not affect pin state.
	1	Output	PADR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PADR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PADR, but does not affect pin state.

Note: n = 7 to 0

35.2 Port B

Port B is an input/output port with the pin configuration shown in figure 35.2. Each pin has an input pull-up MOS, which is controlled by the port B control register (PBCR) in the PFC.

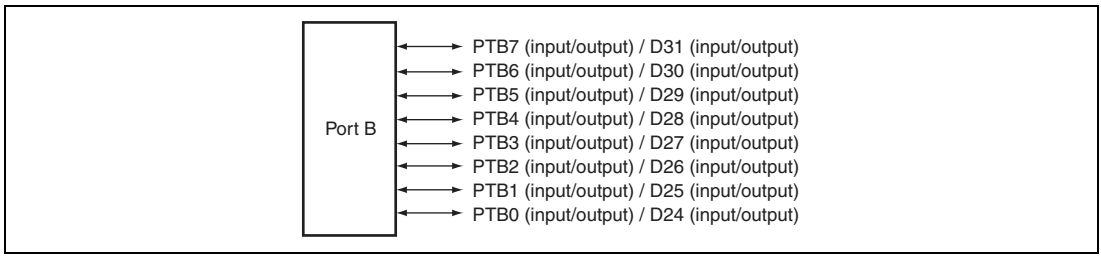


Figure 35.2 Port B

35.2.1 Register Description

Port B has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port B data register (PBDR)

35.2.2 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. When the pin function is general output port, if the port is read, the value of the corresponding PBDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DT	0	R/W	Table 35.2 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 35.2 Port B Data Register (PBDR) Read/Write Operations

PBCR State				
PBnMD1	PBnMD0	Pin State	Read	Write
0	0	Other function	PBDR value	Value is written to PBDR, but does not affect pin state.
	1	Output	PBDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PBDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PBDR, but does not affect pin state.

Note: n = 7 to 0

35.3 Port C

Port C is an input/output port with the pin configuration shown in figure 35.3. Each pin has an input pull-up MOS, which is controlled by the port C control register (PCCR) in the PFC.

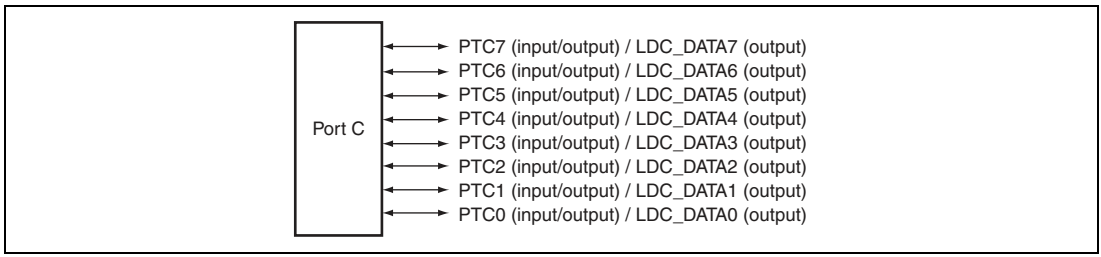


Figure 35.3 Port C

35.3.1 Register Description

Port C has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port C data register (PCDR)

35.3.2 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. When the pin function is general output port, if the port is read, the value of the corresponding PCDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DT	0	R/W	Table 35.3 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 35.3 Port C Data Register (PCDR) Read/Write Operations

PCCR State				
PCnMD1	PCnMD0	Pin State	Read	Write
0	0	Other function	PCDR value	Value is written to PCDR, but does not affect pin state.
	1	Output	PCDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PCDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PCDR, but does not affect pin state.

Note: n = 0 to 7

35.4 Port D

Port D is an input/output port with the pin configuration shown in figure 35.4. Each pin has an input pull-up MOS, which is controlled by the port D control register (PDCR) in the PFC.

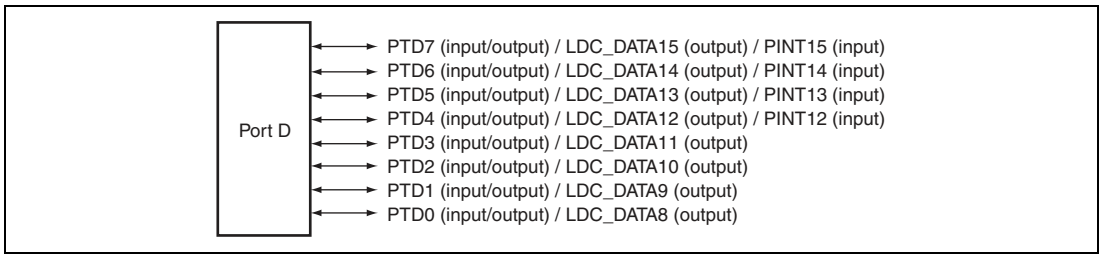


Figure 35.4 Port D

35.4.1 Register Description

Port D has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port D data register (PDDR)

35.4.2 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. When the pin function is general output port, if the port is read, the value of the corresponding PDDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DT	0	R/W	Table 35.4 shows the function of PDDR.
6	PD6DT	0	R/W	
5	PD5DT	0	R/W	
4	PD4DT	0	R/W	
3	PD3DT	0	R/W	
2	PD2DT	0	R/W	
1	PD1DT	0	R/W	
0	PD0DT	0	R/W	

Table 35.4 Port D Data Register (PDDR) Read/Write Operations

PDCR State				
PDnMD1	PDnMD0	Pin State	Read	Write
0	0	Other function	PDDR value	Value is written to PDDR, but does not affect pin state.
	1	Output	PDDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PDDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PDDR, but does not affect pin state.

Note: n = 6 and 7

35.5 Port E

Port E is an input/output port with the pin configuration shown in figure 35.5. Each pin has an input pull-up MOS, which is controlled by the port E control register (PECR) in the PFC.

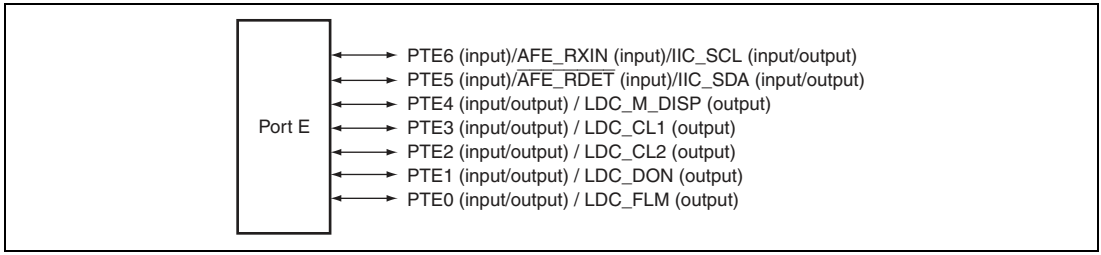


Figure 35.5 Port E

35.5.1 Register Description

Port E has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port E data register (PEDR)

35.5.2 Port E Data Register (PEDR)

PEDR is a register that stores data for pins PTE6 to PTE0. Bits PE6DT to PE0DT correspond to pins PTE6 to PTE0. When the pin function is general output port, if the port is read, the value of the corresponding PEDR bit is returned directly. When the function is general input port, if the port is read the corresponding pin level is read.

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	PE6DT	0	R/W	Table 35.5 shows the function of PEDR.
5	PE5DT	0	R/W	
4	PE4DT	0	R/W	
3	PE3DT	0	R/W	
2	PE2DT	0	R/W	
1	PE1DT	0	R/W	
0	PE0DT	0	R/W	

Table 35.5 Port E Data Register (PEDR) Read/Write Operations

PECR State				
PEnMD1	PEnMD0	Pin State	Read	Write
0	0	Other function	PEDR value	Value is written to PEDR, but does not affect pin state.
	1	Output	PEDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PEDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PEDR, but does not affect pin state.

Note: n= 0 to 4

PECR State

PEnMD1	Pin State	Read	Write
0	Other function	PEDR value	Value is written to PEDR, but does not affect pin state.
1	Input (Pull-up MOS off)	Pin state	Value is written to PEDR, but does not affect pin state.

Note: n= 5 or 6

35.6 Port F

Port F is an input/output port with the pin configuration shown in figure 35.6. Each pin has an input pull-up MOS, which is controlled by the port F control register (PFCR) in the PFC.

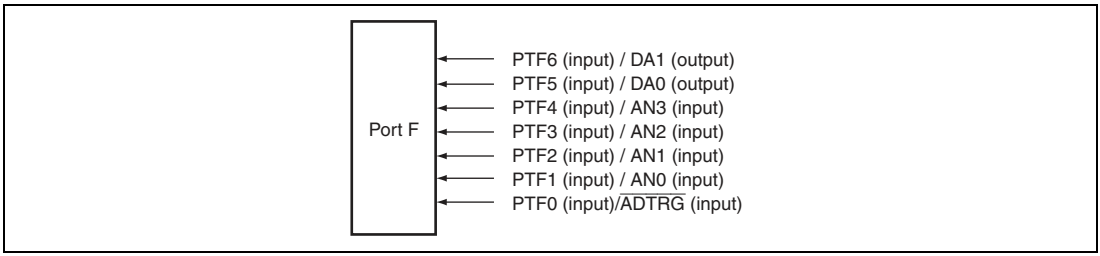


Figure 35.6 Port F

35.6.1 Register Description

Port F has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port F data register (PFDR)

35.6.2 Port F Data Register (PFDR)

PFDR is a register that stores data for pins PTF6 to PTF0. Bits PF6DT to PF0DT correspond to pins PTF6 to PTF0. When the function is general input port, if the port is read, the corresponding pin level is read.

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	PF6DT	0	R	Table 35.6 shows the function of PFDR.
5	PF5DT	0	R	
4	PF4DT	0	R	
3	PF3DT	0	R	
2	PF2DT	0	R	
1	PF1DT	0	R	
0	PF0DT	0	R	

Table 35.6 Port F Data Register (PFDR) Read/Write Operations

PFCR State				
PFnMD1	PFnMD0	Pin State	Read	Write
0	0	Other function	PFDR value	Value is written to PFDR, but does not affect pin state.
	1	Reserved	—	—
1	—	Input (Pull-up MOS off)	Pin state	Value is written to PFDR, but does not affect pin state.

Note: n = 1 to 6

PFCR State

PFnMD1	PFnMD0	Pin State	Read	Write
0	0	Other function	PFDR value	Value is written to PFDR, but does not affect pin state.
	1	Reserved	—	—
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PFDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PFDR, but does not affect pin state.

Note: n = 0

35.7 Port G

Port G is an input/output port with the pin configuration shown in figure 35.7. Each pin has an input pull-up MOS, which is controlled by the port G control register (PGCR) in the PFC.

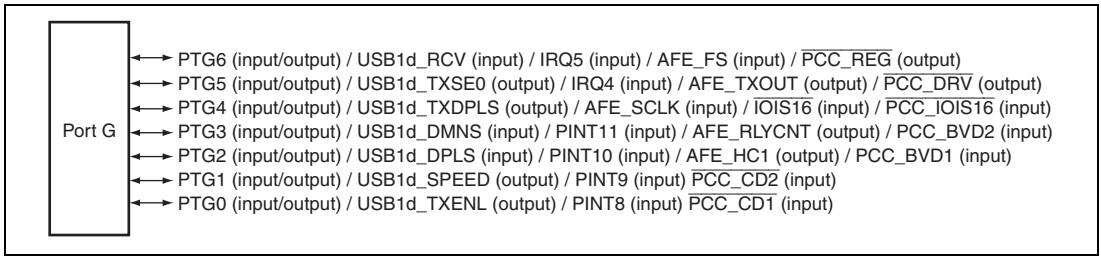


Figure 35.7 Port G

35.7.1 Register Description

Port G has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port G data register (PGDR)

35.7.2 Port G Data Register (PGDR)

PGDR is a register that stores data for pins PTG6 to PTG0. Bits PG6DT to PG0DT correspond to pins PTG6 to PTG0. When the pin function is general output port, if the port is read, the value of the corresponding PGDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

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	PG6DT	0	R/W	Table 35.7 shows the function of PGDR.
5	PG5DT	0	R/W	
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 35.7 Port G Data Register (PGDR) Read/Write Operations

PGCR State				
PGnMD1	PGnMD0	Pin State	Read	Write
0	0	Other function	PGDR value	Value is written to PGDR, but does not affect pin state.
	1	Output	PGDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PGDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PGDR, but does not affect pin state.

Note: n = 0 to 6

35.8 Port H

Port H is an input/output port with the pin configuration shown in figure 35.8. Each pin has an input pull-up MOS, which is controlled by the port H control register (PHCR) in the PFC.

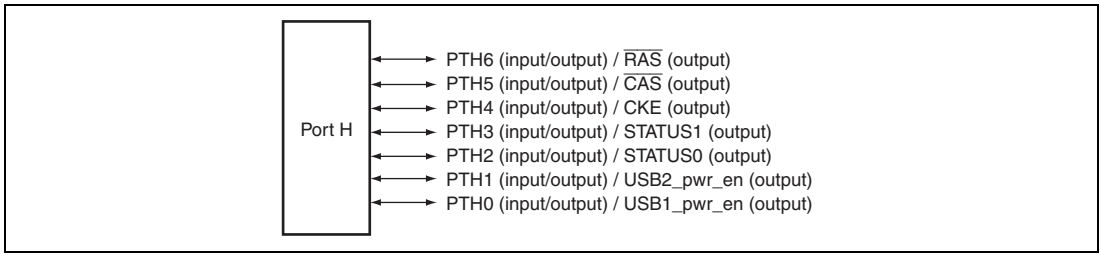


Figure 35.8 Port H

35.8.1 Register Description

Port H has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port H data register (PHDR)

35.8.2 Port H Data Register (PHDR)

PHDR is a register that stores data for pins PTH6 to PTH0. Bits PH6DT to PH0DT correspond to pins PTH6 to PTH0. When the pin function is general output port, if the port is read, the value of the corresponding PHDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

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	PH6DT	0	R/W	Table 35.8 shows the function of PHDR.
5	PH5DT	0	R/W	
4	PH4DT	0	R/W	
3	PH3DT	0	R/W	
2	PH2DT	0	R/W	
1	PH1DT	0	R/W	
0	PH0DT	0	R/W	

Table 35.8 Port H Data Register (PHDR) Read/Write Operations

PHCR State				
PHnMD1	**PHnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PHDR value	Value is written to PHDR, but does not affect pin state.
	1	Output	PHDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PHDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PHDR, but does not affect pin state.

Note: n = 0 to 6

35.9 Port J

Port J is an input/output port with the pin configuration shown in figure 35.9. Each pin has an input pull-up MOS, which is controlled by the port J control register (PJCR) in the PFC.

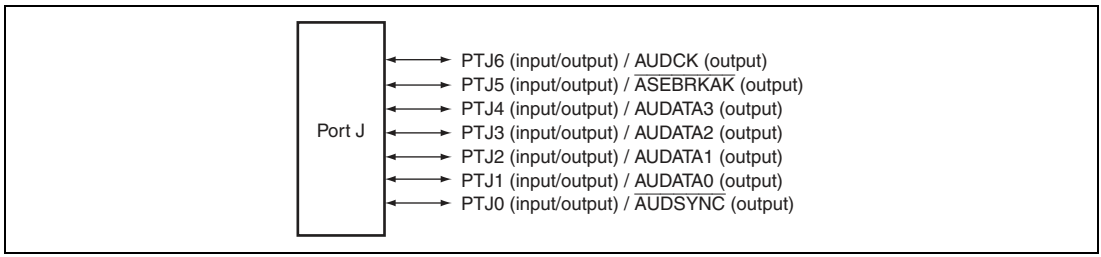


Figure 35.9 Port J

35.9.1 Register Description

Port J has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port J data register (PJDR)

35.9.2 Port J Data Register (PJDR)

PJDR is a register that stores data for pins PTJ6 to PTJ0. Bits PJ6DT to PJ0DT correspond to pins PTJ6 to PTJ0. When the pin function is general output port, if the port is read, the value of the corresponding PJDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

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	PJ6DT	0	R/W	Table 35.9 shows the function of PJDR.
5	PJ5DT	0	R/W	
4	PJ4DT	0	R/W	
3	PJ3DT	0	R/W	
2	PJ2DT	0	R/W	
1	PJ1DT	0	R/W	
0	PJ0DT	0	R/W	

Table 35.9 Port J Data Register (PJDR) Read/Write Operations

PJCR State				
PJnMD1	PJnMD0	Pin State	Read	Write
0	0	Other function	PJDR value	Value is written to PJDR, but does not affect pin state.
	1	Output	PJDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PJDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PJDR, but does not affect pin state.

Note: n = 0 to 6

35.10 Port K

Port K is an input/output port with the pin configuration shown in figure 35.10. Each pin has an input pull-up MOS, which is controlled by the port K control register (PKCR) in the PFC.

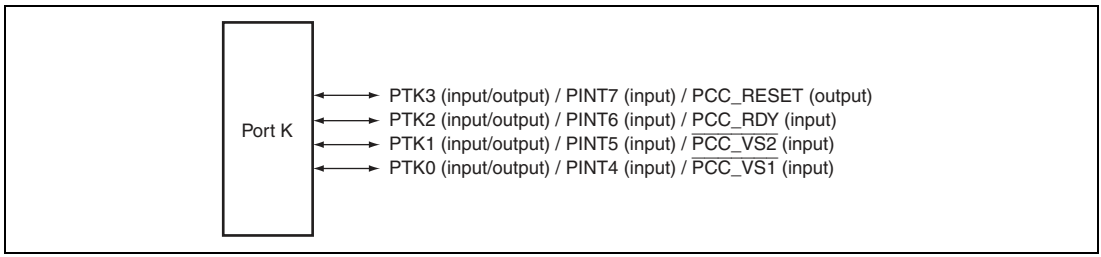


Figure 35.10 Port K

35.10.1 Register Description

Port K has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port K data register (PKDR)

35.10.2 Port K Data Register (PKDR)

PKDR is a register that stores data for pins PTK3 to PTK0. Bits PK3DT to PK0DT correspond to pins PTK3 to PTK0. When the pin function is general output port, if the port is read, the value of the corresponding PKDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

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	PK3DT	0	R/W	Table 35.10 shows the function of PKDR.
2	PK2DT	0	R/W	
1	PK1DT	0	R/W	
0	PK0DT	0	R/W	

Table 35.10 Port K Data Register (PKDR) Read/Write Operations

PKCR State				
PKnMD1	PKnMD0	Pin State	Read	Write
0	0	Other function	PKDR value	Value is written to PKDR, but does not affect pin state.
	1	Output	PKDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PKDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PKDR, but does not affect pin state.

Note: n = 0 to 3

35.11 Port L

Port L is an input/output port with the pin configuration shown in figure 35.11. Each pin has an input pull-up MOS, which is controlled by the port L control register (PLCR) in the PFC.

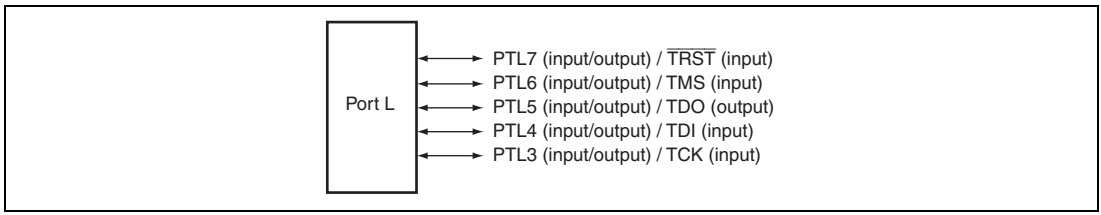


Figure 35.11 Port L

35.11.1 Register Description

Port L has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port L data register (PLDR)

35.11.2 Port L Data Register (PLDR)

PLDR is a register that stores data for pins PTL7 to PTL3. Bits PL7DT to PL3DT correspond to pins PTL7 to PTL3. When the function is general output port, if the port is read, the value of the corresponding PLDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

Bit	Bit Name	Initial Value	R/W	Description
7	PL7DT	0	R/W	Table 35.11 shows the function of PLDR.
6	PL6DT	0	R/W	
5	PL5DT	0	R/W	
4	PL4DT	0	R/W	
3	PL3DT	0	R/W	
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Table 35.11 Port L Data Register (PLDR) Read/Write Operations

PLCR State				
PLnMD1	PLnMD0	Pin State	Read	Write
0	0	Other function	PLDR value	Value is written to PLDR, but does not affect pin state.
	1	Output	PLDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PLDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PLDR, but does not affect pin state.

Note: n = 3 to 7

35.12 Port M

Port M is an input/output port with the pin configuration shown in figure 35.12. Each pin has an input pull-up MOS, which is controlled by the port M control register (PMCR) in the PFC.

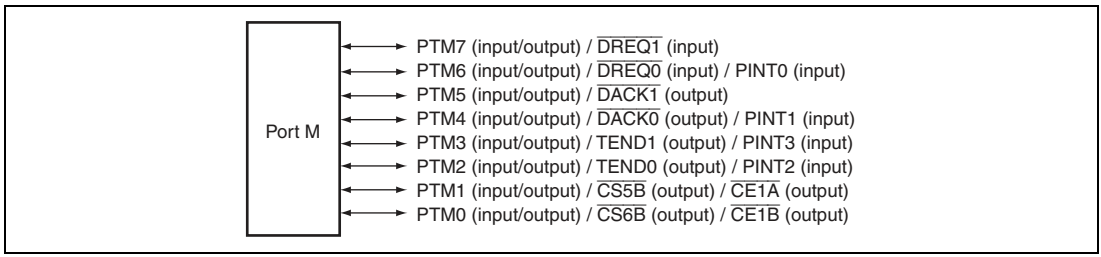


Figure 35.12 Port M

35.12.1 Register Description

Port M has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port M data register (PMDR)

35.12.2 Port M Data Register (PMDR)

PMDR is a register that stores data for pins PTM7 to PTM0. Bits PM7DT to PM0DT correspond to pins PTM7 to PTM0. When the pin function is general output port, if the port is read, the value of the corresponding PMDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

Bit	Bit Name	Initial Value	R/W	Description
7	PM7DT	0	R/W	Table 35.12 shows the function of PMDR.
6	PM6DT	0	R/W	
5	PM5DT	0	R/W	
4	PM4DT	0	R/W	
3	PM3DT	0	R/W	
2	PM2DT	0	R/W	
1	PM1DT	0	R/W	
0	PM0DT	0	R/W	

Table 35.12 Port M Data Register (PMDR) Read/Write Operations

PMCR State				
PMnMD1	PMnMD0	Pin State	Read	Write
0	0	Other function	PMDR value	Value is written to PMDR, but does not affect pin state.
	1	Output	PMDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PMDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PMDR, but does not affect pin state.

Note: n = 0 to 7

35.13 Port P

Port P is an input/output port with the pin configuration shown in figure 35.13. Each pin has an input pull-up MOS, which is controlled by the port P control register (PPCR) in the PFC.

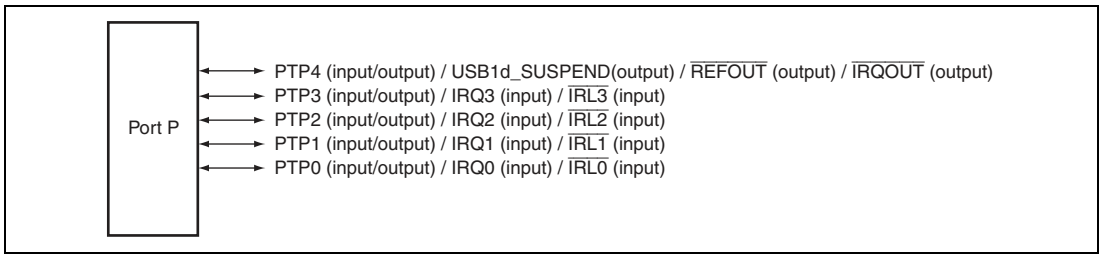


Figure 35.13 Port P

35.13.1 Register Description

Port P has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port P data register (PPDR)

35.13.2 Port P Data Register (PPDR)

PPDR is a register that stores data for pins PTP4 to PTP0. Bits PP4DT to PP0DT correspond to pins PTP4 to PTP0. When the pin function is general output port, if the port is read, the value of the corresponding PPDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

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	PP4DT	0	R/W	Table 35.13 shows the function of PPDR.
3	PP3DT	0	R/W	
2	PP2DT	0	R/W	
1	PP1DT	0	R/W	
0	PP0DT	0	R/W	

Table 35.13 Port P Data Register (PPDR) Read/Write Operations

PPCR State				
PPnMD1	PPnMD0	Pin State	Read	Write
0	0	Other function	PPDR value	Value is written to PPDR, but does not affect pin state.
	1	Output	PPDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PPDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PPDR, but does not affect pin state.

Note: n = 0 to 4

35.14 Port R

Port R is an input/output port with the pin configuration shown in figure 35.14. Each pin has an input pull-up MOS, which is controlled by the port R control register (PRCR) in the PFC.

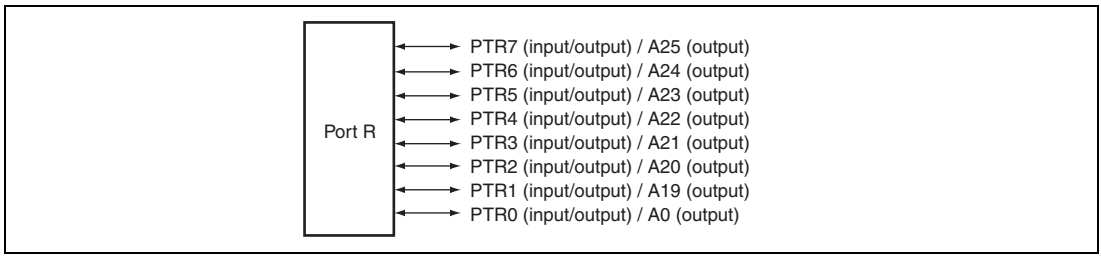


Figure 35.14 Port R

35.14.1 Register Description

Port R has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port R data register (PRDR)

35.14.2 Port R Data Register (PRDR)

PRDR is a register that stores data for pins PTR7 to PTR0. Bits PR7DT to PR0DT correspond to pins PTR7 to PTR0. When the pin function is general output port, if the port is read, the value of the corresponding PRDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

Bit	Bit Name	Initial Value	R/W	Description
7	PR7DT	0	R/W	Table 35.14 shows the function of PRDR.
6	PR6DT	0	R/W	
5	PR5DT	0	R/W	
4	PR4DT	0	R/W	
3	PR3DT	0	R/W	
2	PR2DT	0	R/W	
1	PR1DT	0	R/W	
0	PR0DT	0	R/W	

Table 35.14 Port R Data Register (PRDR) Read/Write Operations

PRCR State				
PRnMD1	PRnMD0	Pin State	Read	Write
0	0	Other function	PRDR value	Value is written to PRDR, but does not affect pin state.
	1	Output	PRDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PRDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PRDR, but does not affect pin state.

Note: n = 0 to 7

35.15 Port S

Port S is an input/output port with the pin configuration shown in figure 35.15. Each pin has an input pull-up MOS, which is controlled by the port S control register (PSCR) in the PFC.

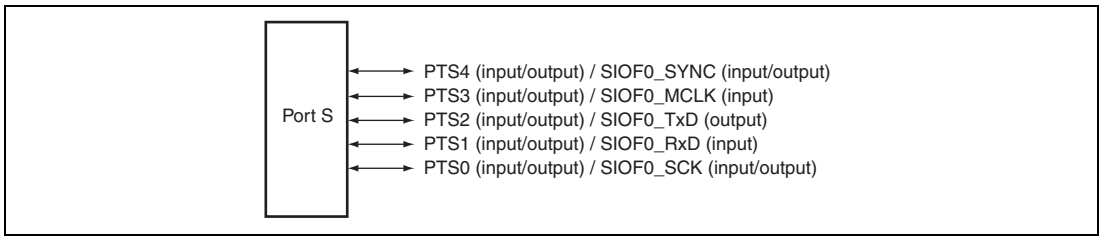


Figure 35.15 Port S

35.15.1 Register Description

Port S has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port S data register (PSDR)

35.15.2 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. When the pin function is general output port, if the port is read, the value of the corresponding PSDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

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 35.15 shows the function of PSDR.
3	PS3DT	0	R/W	
2	PS2DT	0	R/W	
1	PS1DT	0	R/W	
0	PS0DT	0	R/W	

Table 35.15 Port S Data Register (PSDR) Read/Write Operations

PSCR State				
PSnMD1	**PSnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PSDR value	Value is written to PSDR, but does not affect pin state.
	1	Output	PSDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PSDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PSDR, but does not affect pin state.

Note: n = 0 to 4

35.16 Port T

Port T is an input/output port with the pin configuration shown in figure 35.16. Each pin has an input pull-up MOS, which is controlled by the port T control register (PTCR) in the PFC.

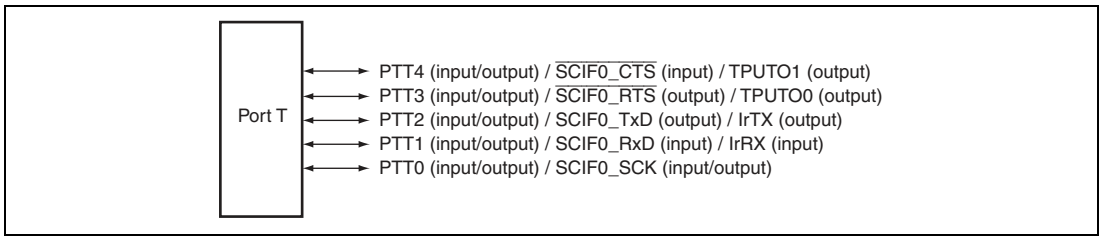


Figure 35.16 Port T

35.16.1 Register Description

Port T has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port T data register (PTDR)

35.16.2 Port T Data Register (PTDR)

PTDR is a register that stores data for pins PTT4 to PTT0. Bits PT4DT to PT0DT correspond to pins PTT4 to PTT0. When the pin function is general output port, if the port is read, the value of the corresponding PTDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

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	PT4DT	0	R/W	Table 35.16 shows the function of PTDR.
3	PT3DT	0	R/W	
2	PT2DT	0	R/W	
1	PT1DT	0	R/W	
0	PT0DT	0	R/W	

Table 35.16 Port T Data Register (PTDR) Read/Write Operations

PTCR State				
PTnMD1	**PTnMD0**	**Pin State**	**Read**	**Write**
0	0	Other function	PTDR value	Value is written to PTDR, but does not affect pin state.
	1	Output	PTDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PTDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PTDR, but does not affect pin state.

Note: n = 0 to 4

35.17 Port U

Port U is an input/output port with the pin configuration shown in figure 35.17. Each pin has an input pull-up MOS, which is controlled by the port U control register (PUCR) in the PFC.

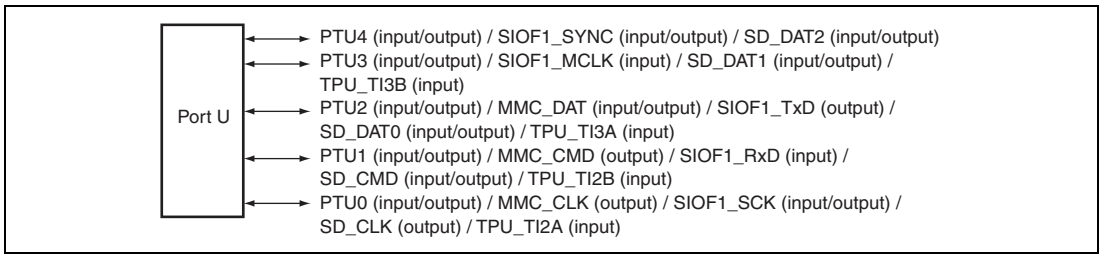


Figure 35.17 Port U

35.17.1 Register Description

Port U has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port U data register (PUDR)

35.17.2 Port U Data Register (PUDR)

PUDR is a register that stores data for pins PTU4 to PTU0. Bits PU4DT to PU0DT correspond to pins PTU4 to PTU0. When the pin function is general output port, if the port is read, the value of the corresponding PUDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

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	PU4DT	0	R/W	Table 35.17 shows the function of PUDR.
3	PU3DT	0	R/W	
2	PU2DT	0	R/W	
1	PU1DT	0	R/W	
0	PU0DT	0	R/W	

Table 35.17 Port U Data Register (PUDR) Read/Write Operations

PUCR State				
PUnMD1	PUnMD0	Pin State	Read	Write
0	0	Other function	PUDR value	Value is written to PUDR, but does not affect pin state.
	1	Output	PUDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PUDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PUDR, but does not affect pin state.

Note: n = 0 to 4

35.18 Port V

Port V is an input/output port with the pin configuration shown in figure 35.18. Each pin has an input pull-up MOS, which is controlled by the port V control register (PVCR) in the PFC.

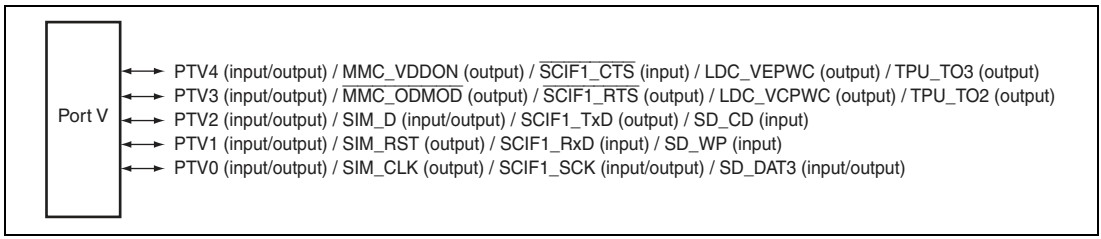


Figure 35.18 Port V

35.18.1 Register Description

Port V has the following register. Refer to section 37, List of Registers, for the address and access size for this register.

- Port V data register (PVDR)

35.18.2 Port V Data Register (PVDR)

PVDR is a register that stores data for pins PTV4 to PTV0. Bits PV4DT to PV0DT correspond to pins PTV4 to PTV0. When the pin function is general output port, if the port is read, the value of the corresponding PVDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read.

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	PV4DT	0	R/W	Table 35.18 shows the function of PVDR.
3	PV3DT	0	R/W	
2	PV2DT	0	R/W	
1	PV1DT	0	R/W	
0	PV0DT	0	R/W	

Table 35.18 Port V Data Register (PVDR) Read/Write Operations

PVCR State				
PVnMD1	PVnMD0	Pin State	Read	Write
0	0	Other function	PVDR value	Value is written to PVDR, but does not affect pin state.
	1	Output	PVDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PVDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PVDR, but does not affect pin state.

Note: n = 0 to 4

Section 36 User Debugging Interface (H-UDI)

This LSI incorporates a user debugging interface (H-UDI) and advanced user debugger (AUD) for a boundary scan function and emulator support.

This section describes the H-UDI. The AUD is a function exclusively for use by an emulator. Refer to the User's Manual for the relevant emulator for details of the AUD.

36.1 Features

The H-UDI is a serial I/O interface which supports JTAG (Joint Test Action Group, IEEE Standard 1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture) specifications.

The H-UDI in this LSI supports a boundary scan mode, and is also used for emulator connection.

When using an emulator, H-UDI functions should not be used. Refer to the emulator manual for the method of connecting the emulator.

Figure 36.1 shows a block diagram of the H-UDI.

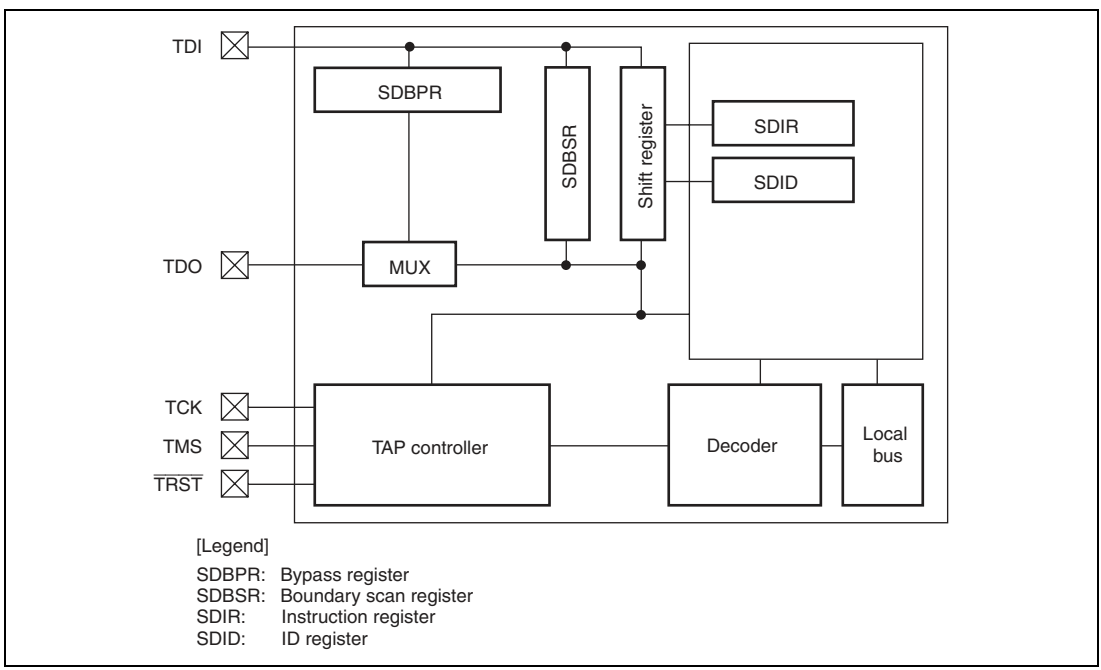


Figure 36.1 Block Diagram of H-UDI

36.2 Input/Output Pins

Table 36.1 shows the pin configuration of the H-UDI.

Table 36.1 Pin Configuration

Pin Name	I/O	Description
TCK	Input	Serial Data Input/Output Clock Pin Data is serially supplied to the H-UDI from the data input pin (TDI), and output from the data output pin (TDO), in synchronization with this clock.
TMS	Input	Mode Select Input Pin The state of the TAP control circuit is determined by changing this signal in synchronization with TCK. The protocol supports the JTAG standard (IEEE Std.1149.1).
$\overline{\text{TRST}}$	Input	Reset Input Pin Input is accepted asynchronously with respect to TCK, and when low, the H-UDI is reset. $\overline{\text{TRST}}$ must be low for a constant period when power is turned on regardless of using the H-UDI function. This is different from the JTAG standard. See section 36.4.2, Reset Configuration, for more information.
TDI	Input	Serial Data Input Pin Data transfer to the H-UDI is executed by changing this signal in synchronization with TCK.
TDO	Output	Serial Data Output Pin Data read from the H-UDI is executed by reading this pin in synchronization with TCK. The data output timing depends on the command type set in the SDIR. See section 36.4.3, TDO Output Timing, for more information.
$\overline{\text{ASEMD0}}$	Input	ASE Mode Select Pin If a low level is input at the $\overline{\text{ASEMD0}}$ pin while the $\overline{\text{RESETP}}$ pin is asserted, ASE mode is entered; if a high level is input, normal mode is entered. When the $\overline{\text{ASEMD0}}$ pin is used by the user system alone without using the emulator and H-UDI, fix the $\overline{\text{ASEMD0}}$ pin high. In ASE mode, dedicated emulator function can be used. The input level at the $\overline{\text{ASEMD0}}$ pin should be held for at least one cycle after $\overline{\text{RESETP}}$ negation.
$\overline{\text{ASEBRKAK}}$ $\overline{\text{AUDSYNC}}$ AUDATA3 to AUDATA0 AUDCK	Output	Dedicated Emulator Pin

36.3 Register Descriptions

The H-UDI has the following registers. Refer to section 37, List of Registers, for more details on the addresses and states of these registers in each operating mode.

- Bypass register (SDBPR)
- Instruction register (SDIR)
- Boundary scan register (SDBSR)
- ID register (SDID)
- Shift register

36.3.1 Bypass Register (SDBPR)

SDBPR is a 1-bit register that cannot be accessed by the CPU. When SDIR is set to the bypass mode, SDBPR is connected between H-UDI pins TDI and TDO. The initial value is undefined.

36.3.2 Instruction Register (SDIR)

SDIR is a 16-bit read-only register. The register is in JTAG IDCODE in its initial state. It is initialized by $\overline{\text{TRST}}$ assertion or in the TAP test-logic-reset state, and can be written to by the H-UDI irrespective of the CPU mode. Operation is not guaranteed if a reserved command is set in this register.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	TI7 to TI5	All 1	R	Test Instruction 7 to 0 The H-UDI instruction is transferred to SDIR by a serial input from TDI. For commands, see table 36.2.
12	TI4	0	R	
11 to 8	TI3 to TI0	All 1	R	
7 to 2	—	All 1	R	Reserved These bits are always read as 1.
1	—	0	R	Reserved This bit is always read as 0.
0	—	1	R	Reserved This bit is always read as 1.

Table 36.2 H-UDI Commands

Bits 15 to 8								
TI7	**TI6**	**TI5**	**TI4**	**TI3**	**TI2**	**TI1**	**TI0**	**Description**
0	0	0	0	—	—	—	—	JTAG EXTEST
0	0	1	0	—	—	—	—	JTAG CLAMP
0	0	1	1	—	—	—	—	JTAG HIGHZ
0	1	0	0	—	—	—	—	JTAG SAMPLE/PRELOAD
0	1	1	0	—	—	—	—	H-UDI reset, negate
0	1	1	1	—	—	—	—	H-UDI reset, assert
1	0	1	—	—	—	—	—	H-UDI interrupt
1	1	1	0	—	—	—	—	JTAG IDCODE (Initial value)
1	1	1	1	—	—	—	—	JTAG BYPASS
Other than the above								Reserved

36.3.3 Shift Register

Shift register is a 32-bit register. The upper 16 bits are set in SDIR at Update-IR.

If shifted in, the shift-in value is shift-out after the value of the 32-bit shift register is shifted out.

36.3.4 Boundary Scan Register (SDBSR)

SDBSR is a 434-bit shift register, located on the PAD, for controlling the input/output pins of this LSI. The initial value is undefined. This register cannot be accessed by the CPU.

Using the EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ commands, a boundary scan test supporting the JTAG standard can be carried out. Table 36.3 shows the correspondence between this LSI's pins and boundary scan register bits.

Table 36.3 Pins and Boundary Scan Register Bits

Bit	Pin Name	I/O	Bit	Pin Name	I/O
from TDI			404	D24/PTB0	OUT
433	MD2	IN	403	D23/PTA7	OUT
432	MD1	IN	402	D22/PTA6	OUT
431	MD0	IN	401	D21/PTA5	OUT
430	D31/PTB7	IN	400	D20/PTA4	OUT
429	D30/PTB6	IN	399	D19/PTA3	OUT
428	D29/PTB5	IN	398	D18/PTA2	OUT
427	D28/PTB4	IN	397	D17/PTA1	OUT
426	D27/PTB3	IN	396	D16/PTA0	OUT
425	D26/PTB2	IN	395	RD/$\overline{WR}$	OUT
424	D25/PTB1	IN	394	$\overline{CAS}$/PTH5	OUT
423	D24/PTB0	IN	393	$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	OUT
422	D23/PTA7	IN	392	$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	OUT
421	D22/PTA6	IN	391	CKE/PTH4	OUT
420	D21/PTA5	IN	390	$\overline{RAS}$/PTH6	OUT
419	D20/PTA4	IN	389	$\overline{WE1}$/$\overline{DQMLU}$/$\overline{WE}$	OUT
418	D19/PTA3	IN	388	$\overline{WE0}$/DQMLL	OUT
417	D18/PTA2	IN	387	$\overline{CS2}$	OUT
416	D17/PTA1	IN	386	$\overline{CS3}$	OUT
415	D16/PTA0	IN	385	A17	OUT
414	$\overline{CAS}$/PTH5	IN	384	A16	OUT
413	CKE/PTH4	IN	383	A15	OUT
412	$\overline{RAS}$/PTH6	IN	382	A14	OUT
411	D31/PTB7	OUT	381	A13	OUT
410	D30/PTB6	OUT	380	A12	OUT
409	D29/PTB5	OUT	379	A11	OUT
408	D28/PTB4	OUT	378	A10	OUT
407	D27/PTB3	OUT	377	A9	OUT
406	D26/PTB2	OUT	376	A8	OUT
405	D25/PTB1	OUT	375	A7	OUT

Bit	Pin Name	I/O	Bit	Pin Name	I/O
374	A6	OUT	342	A15	Control
373	A5	OUT	341	A14	Control
372	A4	OUT	340	A13	Control
371	A3	OUT	339	A12	Control
370	D31/PTB7	Control	338	A11	Control
369	D30/PTB6	Control	337	A10	Control
368	D29/PTB5	Control	336	A9	Control
367	D28/PTB4	Control	335	A8	Control
366	D27/PTB3	Control	334	A7	Control
365	D26/PTB2	Control	333	A6	Control
364	D25/PTB1	Control	332	A5	Control
363	D24/PTB0	Control	331	A4	Control
362	D23/PTA7	Control	330	A3	Control
361	D22/PTA6	Control	329	A0/PTR0	IN
360	D21/PTA5	Control	328	D15	IN
359	D20/PTA4	Control	327	D14	IN
358	D19/PTA3	Control	326	D13	IN
357	D18/PTA2	Control	325	D12	IN
356	D17/PTA1	Control	324	D11	IN
355	D16/PTA0	Control	323	D10	IN
354	RD/$\overline{WR}$	Control	322	D9	IN
353	$\overline{CAS}$/PTH5	Control	321	D8	IN
352	$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	Control	320	D7	IN
351	$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	Control	319	D6	IN
350	CKE/PTH4	Control	318	D5	IN
349	$\overline{RAS}$/PTH6	Control	317	D4	IN
348	$\overline{WE1}$/DQMLU/$\overline{WE}$	Control	316	D3	IN
347	$\overline{WE0}$/DQMLL	Control	315	D2	IN
346	$\overline{CS2}$	Control	314	D1	IN
345	$\overline{CS3}$	Control	313	D0	IN
344	A17	Control	312	$\overline{CS6B}$/$\overline{CE1B}$/PTM0	IN
343	A16	Control	311	$\overline{CS5B}$/$\overline{CE1A}$/PTM1	IN

Bit	Pin Name	I/O	Bit	Pin Name	I/O
310	$\overline{\text{BREQ}}$	IN	278	D1	OUT
309	$\overline{\text{WAIT}}$/$\overline{\text{PCC\_WAIT}}$	IN	277	D0	OUT
308	A19/PTR1	IN	276	$\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$/PTM0	OUT
307	A20/PTR2	IN	275	$\overline{\text{CS6A}}$/$\overline{\text{CE2B}}$	OUT
306	A21/PTR3	IN	274	$\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$/PTM1	OUT
305	A22/PTR4	IN	273	$\overline{\text{CS5A}}$/$\overline{\text{CE2A}}$	OUT
304	A23/PTR5	IN	272	$\overline{\text{BACK}}$	OUT
303	A24/PTR6	IN	271	$\overline{\text{CS0}}$	OUT
302	A25/PTR7	IN	270	$\overline{\text{CS4}}$	OUT
301	$\overline{\text{DREQ0}}$/PINT0/PTM6	IN	269	$\overline{\text{BS}}$	OUT
300	$\overline{\text{DACK0}}$/PINT1/PTM4	IN	268	$\overline{\text{RD}}$	OUT
299	TEND0/PINT2/PTM2	IN	267	A18	OUT
298	$\overline{\text{DREQ1}}$/PTM7	IN	266	A19/PTR1	OUT
297	$\overline{\text{DACK1}}$/PTM5	IN	265	A20/PTR2	OUT
296	TEND1/PINT3/PTM3	IN	264	A21/PTR3	OUT
295	A2	OUT	263	A22/PTR4	OUT
294	A1	OUT	262	A23/PTR5	OUT
293	A0/PTR0	OUT	261	A24/PTR6	OUT
292	D15	OUT	260	A25/PTR7	OUT
291	D14	OUT	259	$\overline{\text{DREQ0}}$/PINT0/PTM6	OUT
290	D13	OUT	258	$\overline{\text{DACK0}}$/PINT1/PTM4	OUT
289	D12	OUT	257	TEND0/PINT2/PTM2	OUT
288	D11	OUT	256	$\overline{\text{DREQ1}}$/PTM7	OUT
287	D10	OUT	255	$\overline{\text{DACK1}}$/PTM5	OUT
286	D9	OUT	254	TEND1/PINT3/PTM3	OUT
285	D8	OUT	253	A2	Control
284	D7	OUT	252	A1	Control
283	D6	OUT	251	A0/PTR0	Control
282	D5	OUT	250	D15	Control
281	D4	OUT	249	D14	Control
280	D3	OUT	248	D13	Control
279	D2	OUT	247	D12	Control

Bit	Pin Name	I/O	Bit	Pin Name	I/O
246	D11	Control	216	$\overline{\text{DACK0}}$/PINT1/PTM4	Control
245	D10	Control	215	TEND0/PINT2/PTM2	Control
244	D9	Control	214	$\overline{\text{DREQ1}}$/PTM7	Control
243	D8	Control	213	$\overline{\text{DACK1}}$/PTM5	Control
242	D7	Control	212	TEND1/PINT3/PTM3	Control
241	D6	Control	211	$\overline{\text{PCC\_VS1}}$/PINT4/PTK0	IN
240	D5	Control	210	$\overline{\text{PCC\_VS2}}$/PINT5/PTK1	IN
239	D4	Control	209	PCC_RDY/PINT6/PTK2	IN
238	D3	Control	208	PCC_RESET/PINT7/PTK3	IN
237	D2	Control	207	$\overline{\text{ASEBRKAK}}$/PTJ5	IN
236	D1	Control	206	$\overline{\text{AUDSYNC}}$/PTJ0	IN
235	D0	Control	205	AUDCK/PTJ6	IN
234	$\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$/PTM0	Control	204	AUDATA0/PTJ1	IN
233	$\overline{\text{CS6A}}$/$\overline{\text{CE2B}}$	Control	203	AUDATA1/PTJ2	IN
232	$\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$/PTM1	Control	202	AUDATA2/PTJ3	IN
231	$\overline{\text{CS5A}}$/$\overline{\text{CE2A}}$	Control	201	AUDATA3/PTJ4	IN
230	$\overline{\text{BACK}}$	Control	200	NMI	IN
229	$\overline{\text{CS0}}$	Control	199	IRQ0/$\overline{\text{IRL0}}$/PTP0	IN
228	$\overline{\text{CS4}}$	Control	198	IRQ1/$\overline{\text{IRL1}}$/PTP1	IN
227	$\overline{\text{BS}}$	Control	197	IRQ2/$\overline{\text{IRL2}}$/PTP2	IN
226	$\overline{\text{RD}}$	Control	196	IRQ3/$\overline{\text{IRL3}}$/PTP3	IN
225	A18	Control	195	SCIF0_SCK/PTT0	IN
224	A19/PTR1	Control	194	SCIF0_RxD/IrRX/PTT1	IN
223	A20/PTR2	Control	193	SCIF0_TxD/IrTX/PTT2	IN
222	A21/PTR3	Control	192	$\overline{\text{SCIF0\_RTS}}$/TPU_TO0/PTT3	IN
221	A22/PTR4	Control	191	$\overline{\text{SCIF0\_CTS}}$/TPU_TO1/PTT4	IN
220	A23/PTR5	Control	190	MMC_CLK/SIOF1_SCK/SD_CLK/ TPU_TI2A/PTU0	IN
219	A24/PTR6	Control	189	MMC_CMD/SIOF1_RxD/SD_CMD/ TPU_TI2B/PTU1	IN
218	A25/PTR7	Control	188	MMC_DAT/SIOF1_TxD/SD_DAT0/ TPU_TI3A/PTU2	IN
217	$\overline{\text{DREQ0}}$/PINT0/PTM6	Control			

Bit	Pin Name	I/O	Bit	Pin Name	I/O
187	SIOF1_MCLK/SD_DAT1/TPU_TI3B/ PTU3	IN	168	PCC_RESET/PINT7/PTK3	OUT
186	SIOF1_SYNC/SD_DAT2/PTU4	IN	167	$\overline{\text{ASEBRKAK}}$/PTJ5	OUT
185	SIM_CLK/SCIF1_SCK/SD_DAT3/ PTV0	IN	166	$\overline{\text{AUDSYNC}}$/PTJ0	OUT
184	SIM_RST/SCIF1_RxD/SD_WP/PTV1	IN	165	AUDCK/PTJ6	OUT
183	SIM_D/SCIF1_TxD/SD_CD/PTV2	IN	164	AUDATA0/PTJ1	OUT
182	$\overline{\text{MMC\_ODMOD}}$/$\overline{\text{SCIF1\_RTS}}$/ LCD_VCPWC/TPU_TO2/PTV3	IN	163	AUDATA1/PTJ2	OUT
181	MMC_VDDON/$\overline{\text{SCIF1\_CTS}}$/ LCD_VEPWC/TPU_TO3/PTV4	IN	162	AUDATA2/PTJ3	OUT
180	USB1d_TXENL/PINT8/ $\overline{\text{PCC\_CD1}}$/PTG0	IN	161	AUDATA3/PTJ4	OUT
179	USB1d_SPEED/PINT9/ $\overline{\text{PCC\_CD2}}$/PTG1	IN	160	IRQ0/$\overline{\text{IRL0}}$/PTP0	OUT
178	USB1d_DPLS/PINT10/AFE_HC1/ PCC_BVD1/PTG2	IN	159	IRQ1/$\overline{\text{IRL1}}$/PTP1	OUT
177	USB1d_DMNS/PINT11/AFE_RLYCNT/ PCC_BVD2/PTG3	IN	158	IRQ2/$\overline{\text{IRL2}}$/PTP2	OUT
176	USB1d_TXDPLS/AFE_SCLK/$\overline{\text{IOIS16}}$/ $\overline{\text{PCC\_IOIS16}}$/PTG4	IN	157	IRQ3/$\overline{\text{IRL3}}$/PTP3	OUT
175	USB1d_TXSE0/IRQ4/AFE_TXOUT/ $\overline{\text{PCC\_DRV}}$/PTG5	IN	156	SCIF0_SCK/PTT0	OUT
174	USB1d_RCV/IRQ5/AFE_FS/ $\overline{\text{PCC\_REG}}$/PTG6	IN	155	SCIF0_RxD/IrRX/PTT1	OUT
173	USB1d_SUSPEND/$\overline{\text{REFOUT}}$/ $\overline{\text{IRQOUT}}$/PTP4	IN	154	SCIF0_TxD/IrTX/PTT2	OUT
172	$\overline{\text{USB1\_ovr\_current}}$/USBF_VBUS	IN	153	$\overline{\text{SCIF0\_RTS}}$/TPU_TO0/PTT3	OUT
171	$\overline{\text{PCC\_VS1}}$/PINT4/PTK0	OUT	152	$\overline{\text{SCIF0\_CTS}}$/TPU_TO1/PTT4	OUT
170	$\overline{\text{PCC\_VS2}}$/PINT5/PTK1	OUT	151	MMC_CLK/SIOF1_SCK/SD_CLK/ TPU_TI2A/PTU0	OUT
169	PCC_RDY/PINT6/PTK2	OUT	150	MMC_CMD/SIOF1_RxD/SD_CMD/ TPU_TI2B_PTU1	OUT

Bit	Pin Name	I/O	Bit	Pin Name	I/O
149	MMC_DAT/SIOF1_TxD/SD_DAT0/ TPU_TI3A/PTU2	OUT	130	PCC_RESET/PINT7/PTK3	Control
148	SIOF1_MCLK/SD_DAT1/TPU_TI3B/ PTU3	OUT	129	$\overline{\text{ASEBRKAK}}$/PTJ5	Control
147	SIOF1_SYNC/SD_DAT2/PTU4	OUT	128	$\overline{\text{AUDSYNC}}$/PTJ0	Control
146	SIM_CLK/SCIF1_SCK/SD_DAT3/ PTV0	OUT	127	AUDCK/PTJ6	Control
145	SIM_RST/SCIF1_RxD/SD_WP/PTV1	OUT	126	AUDATA0/PTJ1	Control
144	SIM_D/SCIF1_TxD/SD_CD/PTV2	OUT	125	AUDATA1/PTJ2	Control
143	$\overline{\text{MMC\_ODMOD}}$/$\overline{\text{SCIF1\_RTS}}$/ LCD_VCPWC/TPU_TO2/PTV3	OUT	124	AUDATA2/PTJ3	Control
142	MMC_VDDON/$\overline{\text{SCIF1\_CTS}}$/ LCD_VEPWC/TPU_TO3/PTV4	OUT	123	AUDATA3/PTJ4	Control
141	USB1d_TXENL/PINT8 $\overline{\text{PCC\_CD1}}$/PTG0	OUT	122	IRQ0/$\overline{\text{IRL0}}$/PTP0	Control
140	USB1d_SPEED/PINT9/ $\overline{\text{PCC\_CD2}}$/PTG1	OUT	121	IRQ1/$\overline{\text{IRL1}}$/PTP1	Control
139	USB1d_DPLS/PINT10/AFE_HC1/ PCC_BVD1/PTG2	OUT	120	IRQ2/$\overline{\text{IRL2}}$/PTP2	Control
138	USB1d_DMNS/PINT11/AFE_RLYCNT/ PCC_BVD2/PTG3	OUT	119	IRQ3/$\overline{\text{IRL3}}$/PTP3	Control
137	USB1d_TXDPLS/AFE_SCLK/$\overline{\text{IOIS16}}$/ $\overline{\text{PCC\_IOIS16}}$/PTG4	OUT	118	SCIF0_SCK/PTT0	Control
136	USB1d_TXSE0/IRQ4/AFE_TXOUT/ $\overline{\text{PCC\_DRV}}$/PTG5	OUT	117	SCIF0_RxD/IrRX/PTT1	Control
135	USB1d_RCV/IRQ5/AFE_FS/ $\overline{\text{PCC\_REG}}$/PTG6	OUT	116	SCIF0_TxD/IrTX/PTT2	Control
134	USB1d_SUSPEND/$\overline{\text{REFOUT}}$/ $\overline{\text{IRQOUT}}$/PTP4	OUT	115	$\overline{\text{SCIF0\_RTS}}$/TPU_TO0/PTT3	Control
133	$\overline{\text{PCC\_VS1}}$/PINT4/PTK0	Control	114	$\overline{\text{SCIF0\_CTS}}$/TPU_TO1/PTT4	Control
132	$\overline{\text{PCC\_VS2}}$/PINT5/PTK1	Control	113	MMC_CLK/SIOF1_SCK/SD_CLK/TPU _TI2A/PTU0	Control
131	PCC_RDY/PINT6/PTK2	Control	112	MMC_CMD/SIOF1_RxD/SD_CMD/ TPU_TI2B_PTU1	Control

Bit	Pin Name	I/O	Bit	Pin Name	I/O
111	MMC_DAT/SIOF1_TxD/SD_DAT0/ TPU_TI3A/PTU2	Control	91	SIOF0_SCK/PTS0	IN
110	SIOF1_MCLK/SD_DAT1/TPU_TI3B/ PTU3	Control	90	SIOF0_RxD/PTS1	IN
109	SIOF1_SYNC/SD_DAT2/PTU4	Control	89	SIOF0_TxD/PTS2	IN
108	SIM_CLK/SCIF1_SCK/SD_DAT3/ PTV0	Control	88	SIOF0_MCLK/PTS3	IN
107	SIM_RST/SCIF1_RxD/SD_WP/PTV1	Control	86	SIOF0_SYNC/PTS4	IN
106	SIM_D/SCIF1_TxD/SD_CD/PTV2	Control	87	LCD_CLK	IN
105	$\overline{\text{MMC\_ODMOD}}$/$\overline{\text{SCIF1\_RTS}}$/ LCD_VCPWC/TPU_TO2/PTV3	Control	85	LCD_M_DISP/PTE4	IN
104	MMC_VDDON/$\overline{\text{SCIF1\_CTS}}$/ LCD_VEPWC/TPU_TO3/PTV4	Control	84	LCD_CL1/PTE3	IN
103	USB1d_TXENL/PINT8/ $\overline{\text{PCC\_CD1}}$/PTG0	Control	83	LCD_CL2/PTE2	IN
102	USB1d_SPEED/PINT9/ $\overline{\text{PCC\_CD2}}$/PTG1	Control	82	LCD_DON/PTE1	IN
101	USB1d_DPLS/PINT10/AFE_HC1/ PCC_BVD1/PTG2	Control	81	LCD_FLM/PTE0	IN
100	USB1d_DMNS/PINT11/AFE_RLYCNT/ PCC_BVD2/PTG3	Control	80	LCD_DATA0/PTC0	IN
99	USB1d_TXDPLS/AFE_SCLK/$\overline{\text{IOIS16}}$/ $\overline{\text{PCC\_IOIS16}}$/PTG4	Control	79	LCD_DATA1/PTC1	IN
98	USB1d_TXSE0/IRQ4/AFE_TXOUT/ $\overline{\text{PCC\_DRV}}$/PTG5	Control	78	LCD_DATA2/PTC2	IN
97	USB1d_RCV/IRQ5/AFE_FS/ $\overline{\text{PCC\_REG}}$/PTG6	Control	77	LCD_DATA3/PTC3	IN
96	USB1d_SUSPEND/$\overline{\text{REFOUT}}$/ $\overline{\text{IRQOUT}}$/PTP4	Control	76	LCD_DATA4/PTC4	IN
95	$\overline{\text{ADTRG}}$/PTF0	IN	75	LCD_DATA5/PTC5	IN
94	USB1_pwr_en/USBF_UPLUP/PTH0	IN	74	LCD_DATA6/PTC6	IN
93	$\overline{\text{USB2\_ovr\_current}}$	IN	73	LCD_DATA7/PTC7	IN
92	USB2_pwr_en/PTH1	IN	72	LCD_DATA8/PTD0	IN

Bit	Pin Name	I/O	Bit	Pin Name	I/O
71	LCD_DATA9/PTD1	IN	39	LCD_DATA8/PTD0	OUT
70	LCD_DATA10/PTD2	IN	38	LCD_DATA9/PTD1	OUT
69	LCD_DATA11/PTD3	IN	37	LCD_DATA10/PTD2	OUT
68	LCD_DATA12/PINT12/PTD4	IN	36	LCD_DATA11/PTD3	OUT
67	LCD_DATA13/PINT13/PTD5	IN	35	LCD_DATA12/PINT12/PTD4	OUT
66	LCD_DATA14/PINT14/PTD6	IN	34	LCD_DATA13/PINT13/PTD5	OUT
65	LCD_DATA15/PINT15/PTD7	IN	33	LCD_DATA14/PINT14/PTD6	OUT
64	STATUS0/PTH2	IN	32	LCD_DATA15/PINT15/PTD7	OUT
63	STATUS1/PTH3	IN	31	STATUS0/PTH2	OUT
62	MD5	IN	30	STATUS1/PTH3	OUT
61	MD4	IN	29	USB1_pwr_en/USBF_UPLUP/PTH0	Control
60	MD3	IN	28	USB2_pwr_en/PTH1	Control
59	USB1_pwr_en/USBF_UPLUP/PTH0	OUT	27	SIOF0_SCK/PTS0	Control
58	USB2_pwr_en/PTH1	OUT	26	SIOF0_RxD/PTS1	Control
57	SIOF0_SCK/PTS0	OUT	25	SIOF0_TxD/PTS2	Control
56	SIOF0_RxD/PTS1	OUT	24	SIOF0_MCLK/PTS3	Control
55	SIOF0_TxD/PTS2	OUT	23	SIOF0_SYNC/PTS4	Control
54	SIOF0_MCLK/PTS3	OUT	22	LCD_M_DISP/PTE4	Control
53	SIOF0_SYNC/PTS4	OUT	21	LCD_CL1/PTE3	Control
52	LCD_M_DISP/PTE4	OUT	20	LCD_CL2/PTE2	Control
51	LCD_CL1/PTE3	OUT	19	LCD_DON/PTE1	Control
50	LCD_CL2/PTE2	OUT	18	LCD_FLM/PTE0	Control
49	LCD_DON/PTE1	OUT	17	LCD_DATA0/PTC0	Control
48	LCD_FLM/PTE0	OUT	16	LCD_DATA1/PTC1	Control
47	LCD_DATA0/PTC0	OUT	15	LCD_DATA2/PTC2	Control
46	LCD_DATA1/PTC1	OUT	14	LCD_DATA3/PTC3	Control
45	LCD_DATA2/PTC2	OUT	13	LCD_DATA4/PTC4	Control
44	LCD_DATA3/PTC3	OUT	12	LCD_DATA5/PTC5	Control
43	LCD_DATA4/PTC4	OUT	11	LCD_DATA6/PTC6	Control
42	LCD_DATA5/PTC5	OUT	10	LCD_DATA7/PTC7	Control
41	LCD_DATA6/PTC6	OUT	9	LCD_DATA8/PTD0	Control
40	LCD_DATA7/PTC7	OUT	8	LCD_DATA9/PTD1	Control

Bit	Pin Name	I/O	Bit	Pin Name	I/O
7	LCD_DATA10/PTD2	Control	3	LCD_DATA14/PINT14/PTD6	Control
6	LCD_DATA11/PTD3	Control	2	LCD_DATA15/PINT15/PTD7	Control
5	LCD_DATA12/PINT12/PTD4	Control	1	STATUS0/PTH2	Control
4	LCD_DATA13/PINT13/PTD5	Control	0	STATUS1/PTH3	Control
to TDO					

Note: * Control means a low active signal.
The corresponding pin is driven with an OUT value when the Control is driven low.

36.3.5 ID Register (SDID)

SDID is a 32-bit read-only register in which SDIDH and SDIDL are connected. Each register is a 16-bit that can be read by CPU.

The IDCODE command is set from the H-UDI pin. This register can be read from the TDO when the TAP state is Shift-DR. Writing is disabled.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DID31 to DID0	Refer to description	R	Device ID31 to ID0 Device ID register that is stipulated by JTAG. • H'002F200F (initial value) for this SH7720 Group. • H'002F2447 (initial value) for this SH7721 Group. Upper four bits may be changed by the chip version. SDIDH corresponds to bits 31 to 16. SDIDL corresponds to bits 15 to 0.

36.4 Operation

36.4.1 TAP Controller

Figure 36.2 shows the internal states of the TAP controller. State transitions support the JTAG standard.

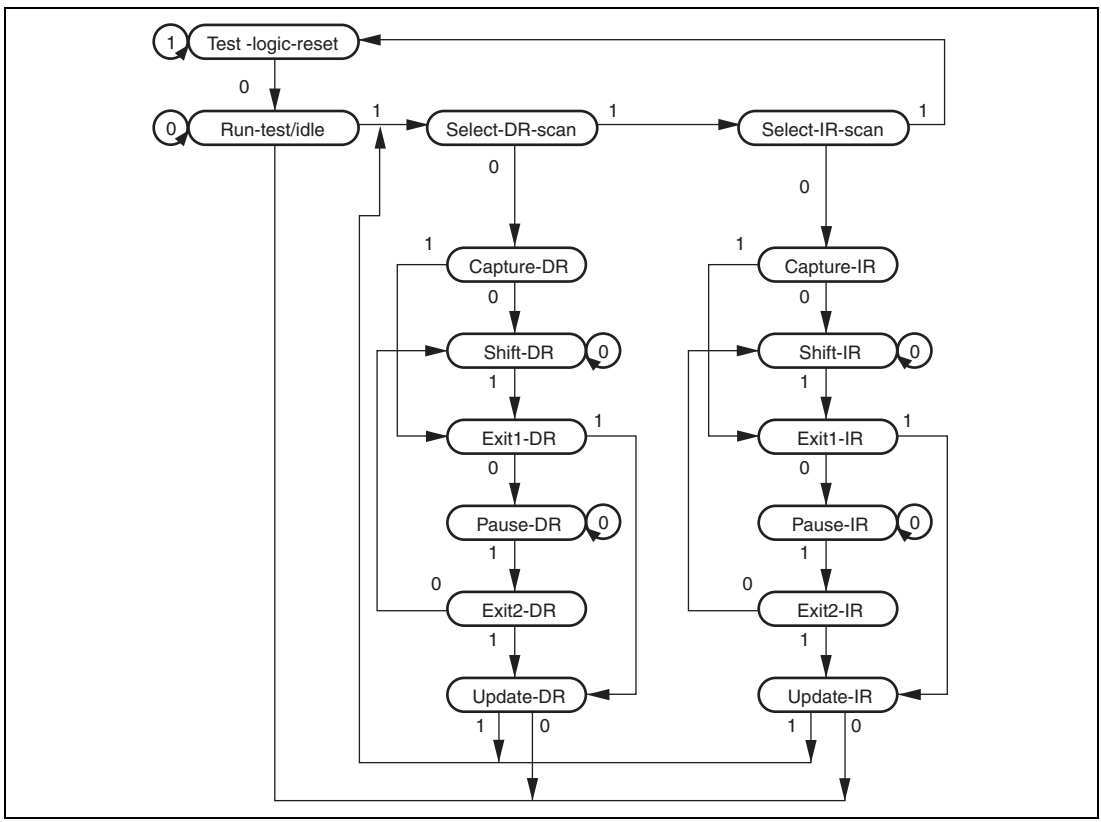


Figure 36.2 TAP Controller State Transitions

Note: The transition condition is the TMS value at the rising edge of TCK. The TDI value is sampled at the rising edge of TCK; shifting occurs at the falling edge of TCK. For details on change timing of the TDO value, see section 36.4.3, TDO Output Timing. The TDO is at high impedance, except with shift-DR and shift-IR states. During the change to $\overline{\text{TRST}}$ = 0, there is a transition to test-logic-reset asynchronously with TCK.

36.4.2 Reset Configuration

Table 36.4 Reset Configuration

$\overline{\text{ASEMD0}}$*1	$\overline{\text{RESETP}}$	$\overline{\text{TRST}}$*4	Chip State
H	L	L	Normal reset and H-UDI reset
		H	Normal reset
	H	L	H-UDI reset only
		H	Normal operation
L	L	L	Reset hold*2
		H	Normal reset*3
	H	L	H-UDI reset only
		H	Normal operation

Notes: 1. Performs normal mode and ASE mode settings
$\overline{\text{ASEMD0}}$ = H, normal mode
$\overline{\text{ASEMD0}}$ = L, ASE mode
2. In ASE mode, reset hold is enabled by driving the $\overline{\text{RESETP}}$ and $\overline{\text{TRST}}$ pins low for a constant cycle. In this state, the CPU does not start up, even if $\overline{\text{RESETP}}$ is driven high. When $\overline{\text{TRST}}$ is driven high, H-UDI operation is enabled, but the CPU does not start up. The reset hold state is canceled by the following:
- Another $\overline{\text{RESETP}}$ assert (power-on reset)
- $\overline{\text{TRST}}$ reassert

3. In ASE mode, reset may not be enabled. When the emulator is not being connected, set $\overline{\text{ASEMD0}}$ to high.
4. When using this LSI in normal mode, it is recommended that the $\overline{\text{TRST}}$ pin is fixed low.

36.4.3 TDO Output Timing

The timing of data output from the TDO is switched by the command type set in the SDIR. The timing changes at the TCK falling edge when JTAG commands (EXTEST, CLAMP, HIGHZ, SAMPLE/PRELOAD, IDCODE, and BYPASS) are set. This is a timing of the JTAG standard. When the H-UDI commands (H-UDI reset negate, H-UDI reset assert, and H-UDI interrupt) are set, TDO is output at the TCK rising edge earlier than the JTAG standard by a half cycle.

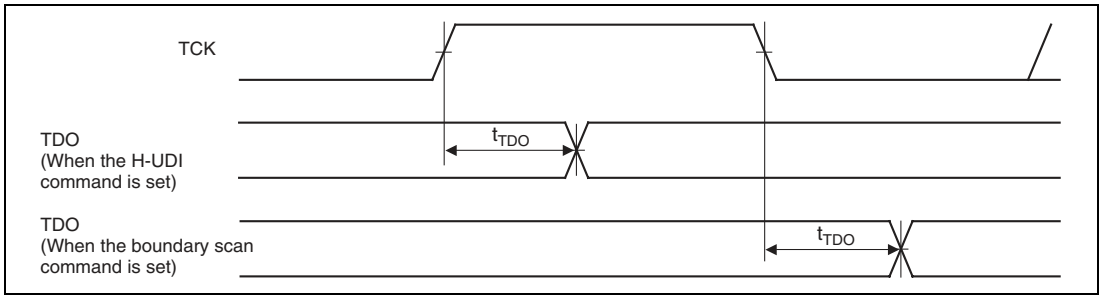


Figure 36.3 H-UDI Data Transfer Timing

36.4.4 H-UDI Reset

An H-UDI reset is executed by inputting an H-UDI reset assert command in SDIR. An H-UDI reset is of the same kind as a power-on reset. An H-UDI reset is released by inputting an H-UDI reset negate command. The required time between the H-UDI reset assert command and H-UDI reset negate command is the same as time for keeping the $\overline{\text{RESETP}}$ pin low to apply a power-on reset.

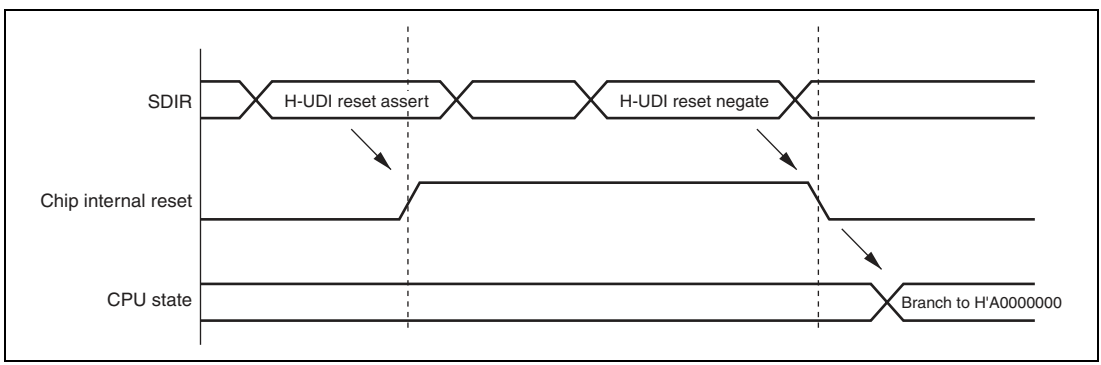


Figure 36.4 H-UDI Reset

36.4.5 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting a command from the H-UDI in the SDIR. An H-UDI interrupt is a general exception or an interrupt operation, resulting in a branch to an address based on the VBR value plus offset, and with return by the RTE instruction. This interrupt request has a fixed priority level of 15.

H-UDI interrupts are accepted in sleep mode, but not in standby mode.

36.5 Boundary Scan

A command can be set in SDIR by the H-UDI to place the H-UDI pins in the boundary scan mode stipulated by JTAG.

36.5.1 Supported Instructions

This LSI supports the three essential instructions defined in the JTAG standard (BYPASS, SAMPLE/PRELOAD, and EXTEST) and three option instructions (IDCODE, CLAMP, and HIGHZ).

(1) BYPASS

The BYPASS instruction is an essential standard instruction that operates the bypass register. This instruction shortens the shift path to speed up serial data transfer involving other chips on the printed circuit board. While this instruction is executing, the test circuit has no effect on the system circuits. The upper four bits of the instruction code are B'1111.

(2) SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction inputs values from this LSI's internal circuitry to the boundary scan register, outputs values from the scan path, and loads data onto the scan path. When this instruction is executing, this LSI's input pin signals are transmitted directly to the internal circuitry, and internal circuit values are directly output externally from the output pins. This LSI's system circuits are not affected by execution of this instruction. The upper four bits of the instruction code are B'0100.

In a SAMPLE operation, a snapshot of a value to be transferred from an input pin to the internal circuitry, or a value to be transferred from the internal circuitry to an output pin, is latched into the boundary scan register and read from the scan path. Snapshot latching is performed in synchronization with the rise of TCK in the Capture-DR state. Snapshot latching does not affect normal operation of this LSI.

In a PRELOAD operation, an initial value is set in the parallel output latch of the boundary scan register from the scan path prior to the EXTEST instruction. Without a PRELOAD operation, when the EXTEST instruction was executed an undefined value would be output from the output pin until completion of the initial scan sequence (transfer to the output latch) (with the EXTEST instruction, the parallel output latch value is constantly output to the output pin).

(3) EXTEST

This instruction is provided to test external circuitry when this LSI is mounted on a printed circuit board. When this instruction is executed, output pins are used to output test data (previously set by the SAMPLE/PRELOAD instruction) from the boundary scan register to the printed circuit board, and input pins are used to latch test results into the boundary scan register from the printed circuit board. If testing is carried out by using the EXTEST instruction N times, the Nth test data is scanned-in when test data (N-1) is scanned out.

Data loaded into the output pin boundary scan register in the Capture-DR state is not used for external circuit testing (it is replaced by a shift operation).

The upper four bits of the instruction code are B'0000.

(4) IDCODE

A command can be set in SDIR by the H-UDI pins to place the H-UDI pins in the IDCODE mode stipulated by JTAG. When the H-UDI is initialized ($\overline{\text{TRST}}$ is asserted or TAP is in the Test-Logic-Reset state), the IDCODE mode is entered.

(5) CLAMP, HIGHZ

A command can be set in SDIR by the H-UDI pins to place the H-UDI pins in the CLAMP or HIGHZ mode stipulated by JTAG.

36.5.2 Points for Attention

1. Boundary scan mode does not cover the following signals:
   - Clock-related signals (EXTAL, XTAL, EXTAL_USB, XTAL_USB, EXTAL_RTC, XTAL_RTC, CKIO)
   - System- and E10A-related signals ($\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, CA, $\overline{\text{ASEMD0}}$)
   - H-UDI-related signals (TCK, TDI, TDO, TMS, $\overline{\text{TRST}}$)
   - IIC-related signals (IIC_SCL/PTE6, IIC_SDA/PTE5)
   - Analog-related signals (AN0/PTF1, AN1/PTF2, AN2/PTF3, AN3/PTF4, DA0/PTF5, DA1/PTF6, USB1_P, USB1_M, USB2_P, USB2_M)
2. When the EXTEST, CLAMP, and HIGHZ commands are set, fix the $\overline{\text{RESETP}}$ pin low.
3. Fix the CA pin high, during boundary scan.
4. When a boundary scan test for other than BYPASS and IDCODE is carried out, fix the $\overline{\text{ASEMD0}}$ pin high.

36.6 Usage Notes

1. An H-UDI command, once set, will not be modified as long as another command is not re-issued from the H-UDI. If the same command is given continuously, the command must be set after a command (BYPASS, etc.) that does not affect chip operations is once set.
2. Because chip operations are suspended in standby mode, H-UDI commands are not accepted. To keep the TAP state constant before and after standby mode, TCK must be high during standby mode transition.
3. The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot be used when using an emulator.

36.7 Advanced User Debugger (AUD)

The AUD is a function only for an emulator. For details on the AUD, refer to each emulator's User's Manual.

Section 37 List of Registers

The address list gives information on the on-chip I/O registers and is configured as described below.

1. Register Addresses (by functional module, in order of the corresponding section numbers)

- Descriptions by functional module, in order of the corresponding section numbers
- Access to reserved addresses which are not described in this list is prohibited.
- When registers consist of 16 or 32 bits, the addresses of the MSBs are given, on the presumption of a big-endian system.

2. Register Bits

- Bit configurations of the registers are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- Reserved bits are indicated by — in the bit name column.
- No entry in the bit-name column indicates that the whole register is allocated as a counter or for holding data.
- When registers consist of 16 or 32 bits, bits are described from the MSB side. The order in which bytes are described is on the presumption of a big-endian system.

3. Register States in Each Operating Mode

- Register states are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- For the initial state of each bit, refer to the description of the register in the corresponding section.
- The register states described here are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, refer to the section on that on-chip module.

37.1 Register Addresses

Entries under Access Size indicate number of bits.

Note: Access to undefined or reserved address is prohibited. Since operation or continued operation is not guaranteed when these registers are accessed, do not attempt such access.

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
MMU control register	MMUCR	32	H'FFFF FFE0	MMU	32
Page table entry register high	PTEH	32	H'FFFF FFF0		32
Page table entry register low	PTEL	32	H'FFFF FFF4		32
Translation table base register	TTB	32	H'FFFF FFF8		32
Cache control register 2	CCR2	32	H'A400 00B0	Cache	32
Cache control register 3	CCR3	32	H'A400 00B4		32
Cache control register 1	CCR1	32	H'FFFF FFEC		32
Interrupt event register 2	INTEVT2	32	H'A400 0000	Exception handling	32
TRAPA exception register	TRA	32	H'FFFF FFD0		32
Exception event register	EXPEVT	32	H'FFFF FFD4		32
Interrupt event register	INTEVT	32	H'FFFF FFD8		32
Exception address register	TEA	32	H'FFFF FFFC		32
Interrupt priority register F	IPRF	16	H'A408 0000	INTC	16
Interrupt priority register G	IPRG	16	H'A408 0002		16
Interrupt priority register H	IPRH	16	H'A408 0004		16
Interrupt priority register I	IPRI	16	H'A408 0006		16
Interrupt priority register J	IPRJ	16	H'A408 0008		16
Interrupt request register 5	IRR5	8	H'A408 0020		8
Interrupt request register 6	IRR6	8	H'A408 0022		8
Interrupt request register 7	IRR7	8	H'A408 0024		8
Interrupt request register 8	IRR8	8	H'A408 0026		8
Interrupt request register 9	IRR9	8	H'A408 0028		8
Interrupt request register 0	IRR0	8	H'A414 0004		8

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Interrupt request register 1	IRR1	8	H'A414 0006	INTC	8
Interrupt request register 2	IRR2	8	H'A414 0008		8
Interrupt request register 3	IRR3	8	H'A414 000A		8
Interrupt request register 4	IRR4	8	H'A414 000C		8
Interrupt control register 1	ICR1	16	H'A414 0010		16
Interrupt control register 2	ICR2	16	H'A414 0012		16
PINT interrupt enable register	PINTER	16	H'A414 0014		16
Interrupt priority register C	IPRC	16	H'A414 0016		16
Interrupt priority register D	IPRD	16	H'A414 0018		16
Interrupt priority register E	IPRE	16	H'A414 001A		16
Interrupt control register 0	ICR0	16	H'A414 FEE0		16
Interrupt priority register A	IPRA	16	H'A414 FEE2		16
Interrupt priority register B	IPRB	16	H'A414 FEE4		16
Common control register	CMNCR	32	H'A4FD 0000	BSC	32
Bus control register for CS0	CS0BCR	32	H'A4FD 0004		32
Bus control register for CS2	CS2BCR	32	H'A4FD 0008		32
Bus control register for CS3	CS3BCR	32	H'A4FD 000C		32
Bus control register for CS4	CS4BCR	32	H'A4FD 0010		32
Bus control register for CS5A	CS5ABCR	32	H'A4FD 0014		32
Bus control register for CS5B	CS5BBCR	32	H'A4FD 0018		32
Bus control register for CS6A	CS6ABCR	32	H'A4FD 001C		32
Bus control register for CS6B	CS6BBCR	32	H'A4FD 0020		32
Wait control register for CS0	CS0WCR	32	H'A4FD 0024		32
Wait control register for CS2	CS2WCR	32	H'A4FD 0028		32
Wait control register for CS3	CS3WCR	32	H'A4FD 002C		32
Wait control register for CS4	CS4WCR	32	H'A4FD 0030		32
Wait control register for CS5A	CS5AWCR	32	H'A4FD 0034		32
Wait control register for CS5B	CS5BWCR	32	H'A4FD 0038		32
Wait control register for CS6A	CS6AWCR	32	H'A4FD 003C		32
Wait control register for CS6B	CS6BWCR	32	H'A4FD 0040		32
SDRAM control register	SDCR	32	H'A4FD 0044		32

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Refresh timer control/status register	RTCSR	32	H'A4FD 0048	BSC	32
Refresh timer counter	RTCNT	32	H'A4FD 004C		32
Refresh time constant register	RTCOR	32	H'A4FD 0050		32
SDRAM mode register	SDMR2	—	H'A4FD 4xxx		16
SDRAM mode register	SDMR3	—	H'A4FD5xxx		16
DMA source address register_0	SAR_0	32	H'A401 0020	DMAC	16, 32
DMA destination address register_0	DAR_0	32	H'A401 0024		16, 32
DMA transfer count register_0	DMATCR_0	32	H'A401 0028		16, 32
DMA channel control register_0	CHCR_0	32	H'A401 002C		8, 16, 32
DMA source address register_1	SAR_1	32	H'A401 0030		16, 32
DMA destination address register_1	DAR_1	32	H'A401 0034		16, 32
DMA transfer count register_1	DMATCR_1	32	H'A401 0038		16, 32
DMA channel control register _1	CHCR_1	32	H'A401 003C		8, 16, 32
DMA source address register_2	SAR_2	32	H'A401 0040		16, 32
DMA destination address register_2	DAR_2	32	H'A401 0044		16, 32
DMA transfer count register_2	DMATCR_2	32	H'A401 0048		16, 32
DMA channel control register_2	CHCR_2	32	H'A401 004C		8, 16, 32
DMA source address register_3	SAR_3	32	H'A401 0050		16, 32
DMA destination address register_3	DAR_3	32	H'A401 0054		16, 32
DMA transfer count register_3	DMATCR_3	32	H'A401 0058		16, 32
DMA channel control register_3	CHCR_3	32	H'A401 005C		8, 16, 32
DMA operation register	DMAOR	16	H'A401 0060		16
DMA source address register_4	SAR_4	32	H'A401 0070		16, 32
DMA destination address register_4	DAR_4	32	H'A401 0074		16, 32
DMA transfer count register_4	DMATCR_4	32	H'A401 0078		16, 32
DMA channel control register_4	CHCR_4	32	H'A401 007C		8, 16, 32
DMA source address register_5	SAR_5	32	H'A401 0080		16, 32
DMA destination address register_5	DAR_5	32	H'A401 0084		16, 32
DMA transfer count register_5	DMATCR_5	32	H'A401 0088		16, 32
DMA channel control register_5	CHCR_5	32	H'A401 008C		8, 16, 32

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
DMA extended resource selector 0	DMARS0	16	H'A409 0000	DMAC	16
DMA extended resource selector 1	DMARS1	16	H'A409 0004		16
DMA extended resource selector 2	DMARS2	16	H'A409 0008		16
USBH/USBF clock control register	UCLKCR	8	H'A40A 0008	CPG	8, 16*[2]
Frequency control register	FRQCR	16	H'A415 FF80		16
Watchdog timer counter	WTCNT	8	H'A415 FF84	WDT	8, 16*[2]
Watchdog timer control/status register	WTCSR	8	H'A415 FF86		8, 16*[2]
Standby control register 3	STBCR3	8	H'A40A 0000	Power-down modes	8
Standby control register 4	STBCR4	8	H'A40A 0004		8
Standby control register 5	STBCR5	8	H'A40A 0010		8
Standby control register	STBCR	8	H'A415 FF82		8
Standby control register 2	STBCR2	8	H'A415 FF88		8
Timer start register	TSTR	8	H'A412 FE92	TMU	8
Timer constant register_0	TCOR_0	32	H'A412 FE94		32
Timer counter_0	TCNT_0	32	H'A412 FE98		32
Timer control register_0	TCR_0	16	H'A412 FE9C		16
Timer constant register_1	TCOR_1	32	H'A412 FEA0		32
Timer counter_1	TCNT_1	32	H'A412 FEA4		32
Timer control register_1	TCR_1	16	H'A412 FEA8		16
Timer constant register_2	TCOR_2	32	H'A412 FEAC		32
Timer counter_2	TCNT_2	32	H'A412 FEB0		32
Timer control register_2	TCR_2	16	H'A412 FEB4		16
Timer start register	TSTR	16	H'A448 0000	TPU	16
Timer control register_0	TCR_0	16	H'A448 0010		16
Timer mode register_0	TMDR_0	16	H'A448 0014		16
Timer I/O control register_0	TIOR_0	16	H'A448 0018		16
Timer interrupt enable register_0	TIER_0	16	H'A448 001C		16
Timer status register_0	TSR_0	16	H'A448 0020		16
Timer counter_0	TCNT_0	16	H'A448 0024		16

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Timer general register A_0	TGRA_0	16	H'A448 0028	TPU	16
Timer general register B_0	TGRB_0	16	H'A448 002C		16
Timer general register C_0	TGRC_0	16	H'A448 0030		16
Timer general register D_0	TGRD_0	16	H'A448 0034		16
Timer control register_1	TCR_1	16	H'A448 0050		16
Timer mode register_1	TMDR_1	16	H'A448 0054		16
Timer I/O control register_1	TIOR_1	16	H'A448 0058		16
Timer interrupt enable register_1	TIER_1	16	H'A448 005C		16
Timer status register_1	TSR_1	16	H'A448 0060		16
Timer counter_1	TCNT_1	16	H'A448 0064		16
Timer general register A_1	TGRA_1	16	H'A448 0068		16
Timer general register B_1	TGRB_1	16	H'A448 006C		16
Timer general register C_1	TGRC_1	16	H'A448 0070		16
Timer general register D_1	TGRD_1	16	H'A448 0074		16
Timer control register_2	TCR_2	16	H'A448 0090		16
Timer mode register_2	TMDR_2	16	H'A448 0094		16
Timer I/O control register_2	TIOR_2	16	H'A448 0098		16
Timer interrupt enable register_2	TIER_2	16	H'A448 009C		16
Timer status register_2	TSR_2	16	H'A448 00A0		16
Timer counter_2	TCNT_2	16	H'A448 00A4		16
Timer general register A_2	TGRA_2	16	H'A448 00A8		16
Timer general register B_2	TGRB_2	16	H'A448 00AC		16
Timer general register C_2	TGRC_2	16	H'A448 00B0		16
Timer general register D_2	TGRD_2	16	H'A448 00B4		16
Timer control register_3	TCR_3	16	H'A448 00D0		16
Timer mode register_3	TMDR_3	16	H'A448 00D4		16
Timer I/O control register_3	TIOR_3	16	H'A448 00D8		16
Timer interrupt enable register_3	TIER_3	16	H'A448 00DC		16
Timer status register_3	TSR_3	16	H'A448 00E0		16
Timer counter_3	TCNT_3	16	H'A448 00E4		16

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Timer general register A_3	TGRA_3	16	H'A448 00E8	TPU	16
Timer general register B_3	TGRB_3	16	H'A448 00EC		16
Timer general register C_3	TGRC_3	16	H'A448 00F0		16
Timer general register D_3	TGRD_3	16	H'A448 00F4		16
Compare match timer start register	CMSTR	16	H'A44A 0000	CMT	16
Compare match timer control/status register_0	CMCSR_0	16	H'A44A 0010		16
Compare match timer counter_0	CMCNT_0	32	H'A44A 0014		32
Compare match timer constant register_0	CMCOR_0	32	H'A44A 0018		32
Compare match timer control/status register_1	CMCSR_1	16	H'A44A 0020		16
Compare match timer counter_1	CMCNT_1	32	H'A44A 0024		32
Compare match timer constant register_1	CMCOR_1	32	H'A44A 0028		32
Compare match timer control/status register_2	CMCSR_2	16	H'A44A 0030		16
Compare match timer counter_2	CMCNT_2	32	H'A44A 0034		32
Compare match timer constant register_2	CMCOR_2	32	H'A44A 0038		32
Compare match timer control/status register_3	CMCSR_3	16	H'A44A 0040		16
Compare match timer counter_3	CMCNT_3	32	H'A44A 0044		32
Compare match timer constant register_3	CMCOR_3	32	H'A44A 0048		32
Compare match timer control/status register_4	CMCSR_4	16	H'A44A 0050		16
Compare match timer counter_4	CMCNT_4	32	H'A44A 0054		32
Compare match timer constant register_4	CMCOR_4	32	H'A44A 0058		32
64-Hz counter	R64CNT	8	H'A413 FEC0	RTC	8
Second counter	RSECCNT	8	H'A413 FEC2		8
Minute counter	RMINCNT	8	H'A413 FEC4		8
Hour counter	RHRCNT	8	H'A413 FEC6		8

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Day of week counter	RWKCNT	8	H'A413 FEC8	RTC	8
Date counter	RDAYCNT	8	H'A413 FECA		8
Month counter	RMONCNT	8	H'A413 FECC		8
Year counter	RYRCNT	16	H'A413 FECE	SCIF	16
Second alarm register	RSECAR	8	H'A413 FED0		8
Minute alarm register	RMINAR	8	H'A413 FED2		8
Hour alarm register	RHRAR	8	H'A413 FED4		8
Day of week alarm register	RWKAR	8	H'A413 FED6		8
Date alarm register	RDAYAR	8	H'A413 FED8		8
Month alarm register	RMONAR	8	H'A413 FEDA		8
RTC control register 1	RCR1	8	H'A413 FEDC		8
RTC control register 2	RCR2	8	H'A413 FEDE		8
Year alarm register	RYRAR	16	H'A413 FEE0		16
RTC control register 3	RCR3	8	H'A413 FEE4		8
Serial mode register_0	SCSMR_0	16	H'A443 0000		16
Bit rate register_0	SCBRR_0	8	H'A443 0004		8
Serial control register_0	SCSCR_0	16	H'A443 0008		16
Transmit data stop register_0	SCTDSR_0	8	H'A443 000C		8
FIFO error count register_0	SCFER_0	16	H'A443 0010		16
Serial status register_0	SCSSR_0	16	H'A443 0014		16
FIFO control register_0	SCFCR_0	16	H'A443 0018		16
FIFO data count register_0	SCFDR_0	16	H'A443 001C		16
Transmit FIFO data register_0	SCFTDR_0	8	H'A443 0020		8
Receive FIFO data register_0	SCFRDR_0	8	H'A443 0024		8
Serial mode register_0	SCSMR_1	16	H'A443 8000		16
Bit rate register_1	SCBRR_1	8	H'A443 8004		8
Serial control register_1	SCSCR_1	16	H'A443 8008		16
Transmit data stop register_1	SCTDSR_1	8	H'A443 800C		8
FIFO error count register_1	SCFER_1	16	H'A443 8010		16
Serial status register_1	SCSSR_1	16	H'A443 8014		16
FIFO control register_1	SCFCR_1	16	H'A443 8018		16

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
FIFO data count register_1	SCFDR_1	16	H'A443 801C	SCIF	16
Transmit FIFO data register_1	SCFTDR_1	8	H'A443 8020		8
Receive FIFO data register_1	SCFRDR_1	8	H'A443 8024		8
IrDA mode register	SCIMR	16	H'A444 0000	IrDA	16
I^2C bus control register 1	ICCR1	8	H'A447 0000	IIC	8
I^2C bus control register 2	ICCR2	8	H'A447 0004		8
I^2C bus mode register	ICMR	8	H'A447 0008		8
I^2C bus interrupt enable register	ICIER	8	H'A447 000C		8
I^2C bus status register	ICSR	8	H'A447 0010		8
Slave address register	SAR	8	H'A447 0014		8
I^2C bus transmit data register	ICDRT	8	H'A447 0018		8
I^2C bus receive data register	ICDRR	8	H'A447 001C		8
I^2C bus master transfer clock select register	ICCKS	8	H'A447 0020		8
Mode register_0	SIMDR_0	16	H'A441 0000	SIOF	16
Clock select register_0	SISCR_0	16	H'A441 0002		16
Transmit data assign register_0	SITDAR_0	16	H'A441 0004		16
Receive data assign register_0	SIRDAR_0	16	H'A441 0006		16
Control data assign register_0	SICDAR_0	16	H'A441 0008		16
Control register_0	SICTR_0	16	H'A441 000C		16
FIFO control register_0	SIFCTR_0	16	H'A441 0010		16
Status register_0	SISTR_0	16	H'A441 0014		16
Interrupt enable register_0	SIIER_0	16	H'A441 0016		16
Transmit data register_0	SITDR_0	32	H'A441 0020		32
Receive data register_0	SIRDR_0	32	H'A441 0024		32
Transmit control data register_0	SITCR_0	32	H'A441 0028		32
Receive control data register_0	SIRCR_0	32	H'A441 002C		32
Mode register_1	SIMDR_1	16	H'A441 8000		16
Clock select register_1	SISCR_1	16	H'A441 8002		16
Transmit data assign register_1	SITDAR_1	16	H'A441 8004		16
Receive data assign register_1	SIRDAR_1	16	H'A441 8006		16

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Control data assign register_1	SICDAR_1	16	H'A441 8008	SIOF	16
Control register_1	SICTR_1	16	H'A441 800C		16
FIFO control register_1	SIFCTR_1	16	H'A441 8010		16
Status register_1	SISTR_1	16	H'A441 8014		16
Interrupt enable register_1	SIIER_1	16	H'A441 8016		16
Transmit data register_1	SITDR_1	32	H'A441 8020		32
Receive data register_1	SIRDR_1	32	H'A441 8024		32
Transmit control data register_1	SITCR_1	32	H'A441 8028		32
Receive control data register_1	SIRCR_1	32	H'A441 802C		32
AFEIF control register 1	ACTR1	16	H'A44E 0180	AFEIF	16
AFEIF control register 2	ACTR2	16	H'A44E 0182		16
AFEIF status register 1	ASTR1	16	H'A44E 0184		16
AFEIF status register 2	ASTR2	16	H'A44E 0186		16
Make ratio count register	MRCR	16	H'A44E 0188		16
Minimum pose count register	MPCR	16	H'A44E 018A		16
Dial number queue	DPNQ	16	H'A44E 018C		16
Ringing pulse counter	RCNT	16	H'A44E 018E		16
AFE control data register	ACDR	16	H'A44E 0190		16
AFE status data register	ASDR	16	H'A44E 0192		16
Transmit data FIFO port	TDFP	16	H'A44E 0194		16, 32
Receive data FIFO port	RDFP	16	H'A44E 0198		16, 32
USB transceiver control register	UTRCTL	16	H'A405 012C	USBPMC	16
Hc Revision register	USBHR	32	H'A442 8000		32
Hc Control register	USBHC	32	H'A442 8004		32
Hc Command Status register	USBHCS	32	H'A442 8008		32
Hc Interrupt Status register	USBHIS	32	H'A442 800C		32
Hc Interrupt Enable register	USBHIE	32	H'A442 8010		32
HcInterruptDisable register	USBHID	32	H'A442 8014		32
HcHCCA register	USBHHCCA	32	H'A442 8018		32
Hc Period Current ED register	USBHPCED	32	H'A442 801C		32
Hc Control Head ED	USBHCHED	32	H'A442 8020		32

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Hc Control Current ED register	USBHCCED	32	H'A442 8024	USBPMC	32
Hc Bulk Head ED register	USBHBHED	32	H'A442 8028		32
Hc Bulk Current ED register	USBHBCED	32	H'A442 802C		32
Hc Done Head ED register	USBHDHED	32	H'A442 8030		32
Hc Fm Interval register	USBHFI	32	H'A442 8034		32
Hc Fm Remaining register	USBHFR	32	H'A442 8038		32
Hc Fm Number register	USBHFN	32	H'A442 803C		32
Hc Periodic Start register	USBHPS	32	H'A442 8040		32
Hc LS Threshold register	USBHLST	32	H'A442 8044		32
Hc Rh Descriptor A register	USBHRDA	32	H'A442 8048		32
Hc Rh Descriptor B register	USBHRDB	32	H'A442 804C		32
Hc Rh Status register	USBHRS	32	H'A442 8050		32
Hc Rh Port Status 1register	USBHRPS1	32	H'A442 8054		32
Hc Rh Port Status 2 register	USBHRPS2	32	H'A442 8058		32
Interrupt flag register 0	IFR0	8	H'A442 0000	USBF	8
Interrupt flag register 1	IFR1	8	H'A442 0001		8
Interrupt flag register 2	IFR2	8	H'A442 0002		8
Interrupt flag register 3	IFR3	8	H'A442 0003		8
Interrupt enable register 0	IER0	8	H'A442 0004		8
Interrupt enable register 1	IER1	8	H'A442 0005		8
Interrupt enable register 2	IER2	8	H'A442 0006		8
Interrupt enable register 3	IER3	8	H'A442 0007		8
Interrupt select register 0	ISR0	8	H'A442 0008		8
Interrupt select register 1	ISR1	8	H'A442 0009		8
Interrupt select register 2	ISR2	8	H'A442 000A		8
Interrupt select register 3	ISR3	8	H'A442 000B		8
EP0i data register	EPDR0i	8	H'A442 000C		8
EP0o data register	EPDR0o	8	H'A442 000D		8
EP0s data register	EPDR0s	8	H'A442 000E		8
EP1 data register	EPDR1	8	H'A442 0010		8
EP2 data register	EPDR2	8	H'A442 0014		8

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
EP3 data register	EPDR3	8	H'A442 0018	USBF	8
EP4 data register	EPDR4	8	H'A442 001C		8
EP5 data register	EPDR5	8	H'A442 0020		8
EP0o receive data size register	EPSZ0o	8	H'A442 0024		8
EP1 receive data size register	EPSZ1	8	H'A442 0025		8
EP4 receive data size register	EPSZ4	8	H'A442 0026		8
Data status register	DASTS	8	H'A442 0027		8
FIFO clear register 0	FCLR0	8	H'A442 0028		8
FIFO clear register 1	FCLR1	8	H'A442 0029		8
Endpoint stall register 0	EPSTL0	8	H'A442 002A		8
Endpoint stall register 1	EPSTL1	8	H'A442 002B		8
Trigger register	TRG	8	H'A442 002C		8
DMA transfer setting register	DMA	8	H'A442 002D		8
Configuration value register	CVR	8	H'A442 002E		8
Control register 0	CTLR0	8	H'A442 002F		8
Time stamp register H	TSRH	8	H'A442 0030		8
Time stamp register L	TSRL	8	H'A442 0031		8
Endpoint information register	EPIR	8	H'A442 0032		8
Interrupt flag register 4	IFR4	8	H'A442 0034		8
Interrupt enable register 4	IER4	8	H'A442 0035		8
Interrupt select register 4	ISR4	8	H'A442 0036		8
Control register 1	CTLR1	8	H'A442 0037		8
Timer register H	TMRH	8	H'A442 0038		8
Timer register L	TMRL	8	H'A442 0039		8
Set time out register H	STOH	8	H'A442 003A		8
Set time out register L	STOL	8	H'A442 003B		8
Palette data register 00 to Palette data register FF	LDPR00 to LDPRFF	32	H'A440 0000 to H'A440 03FC	LCDC	32
LCDC input clock register	LDICKR	16	H'A440 0400		16
LCDC module type register	LDMTR	16	H'A440 0402		16

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
LCDC data format register	LDDFR	16	H'A440 0404	LCDC	16
LCDC scan mode register	LDSMR	16	H'A440 0406		16
LCDC data fetch start address register for upper display panel	LDSARU	32	H'A440 0408		32
LCDC data fetch start address register for lower display panel	LDSARL	32	H'A440 040C		32
LCDC fetch data line address offset register for display panel	LDLAOR	16	H'A440 0410		16
LCDC palette control register	LDPALCR	16	H'A440 0412		16
LCDC horizontal character number register	LDHCNR	16	H'A440 0414		16
LCDC horizontal synchronization signal register	LDHSYNR	16	H'A440 0416		16
LCDC vertical displayed line number register	LDVDLNR	16	H'A440 0418		16
LCDC vertical total line number register	LDVTLNR	16	H'A440 041A		16
LCDC vertical synchronization signal register	LDVSYNR	16	H'A440 041C		16
LCDC AC modulation signal toggle line number register	LDACLNR	16	H'A440 041E		16
LCDC interrupt control register	LDINTR	16	H'A440 0420		16
LCDC power management mode register	LDPMMR	16	H'A440 0424		16
LCDC power supply sequence period register	LDPSPR	16	H'A440 0426		16
LCDC control register	LDCNTR	16	H'A440 0428		16
LCDC user specified interrupt control register	LDUINTR	16	H'A440 0434		16
LCDC user specified interrupt line number register	LDUINTLNR	16	H'A440 0436		16
LCDC memory access interval number register	LDLIRNR	16	H'A440 0440		16

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
A/D data register A	ADDRA	16	H'A44C 0000	ADC	16
A/D data register B	ADDRB	16	H'A44C 0002		16
A/D data register C	ADDRC	16	H'A44C 0004		16
A/D data register D	ADDRD	16	H'A44C 0006		16
A/D control/status register	ADCSR	16	H'A44C 0008		16
D/A data register 0	DADR0	8	H'A44D 0000	DAC	8
D/A data register 1	DADR1	8	H'A44D 0002		8
D/A control register	DACR	8	H'A44D 0004		8
Area 6 interface status register	PCC0ISR	8	H'A44B 0000	PCC	8
Area 6 general control register	PCC0GCR	8	H'A44B 0002		8
Area 6 card status change register	PCC0CSCR	8	H'A44B 0004		8
Area 6 card status change interrupt enable register	PCC0CSCIER	8	H'A44B 0006		8
Serial mode register	SCSMR	8	H'A449 0000	SIM	8
Bit rate register	SCBRR	8	H'A449 0002		8
Serial control register	SCSCR	8	H'A449 0004		8
Transmit data register	SCTDR	8	H'A449 0006		8
Serial status register	SCSSR	8	H'A449 0008		8
Receive data register	SCRDR	8	H'A449 000A		8
Smart card mode register	SCSCMR	8	H'A449 000C		8
Serial control 2 register	SCSC2R	8	H'A449 000E		8
Wait time register	SCWAIT	16	H'A449 0010		16
Guard extension register	SCGRD	8	H'A449 0012		8
Sampling register	SCSMPL	16	H'A449 0014		16
Command register 0	CMDR0	8	H'A444 8000	MMC	8
Command register 1	CMDR1	8	H'A444 8001		8
Command register 2	CMDR2	8	H'A444 8002		8
Command register 3	CMDR3	8	H'A444 8003		8
Command register 4	CMDR4	8	H'A444 8004		8
Command register 5	CMDR5	8	H'A444 8005		8
Command start register	CMDSTRT	8	H'A444 8006		8

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Operation control register	OPCR	8	H'A444 800A	MMC	8
Card status register	CSTR	8	H'A444 800B		8
Interrupt control register 0	INTCR0	8	H'A444 800C		8
Interrupt control register 1	INTCR1	8	H'A444 800D		8
Interrupt status register 0	INTSTR0	8	H'A444 800E		8
Interrupt status register 1	INTSTR1	8	H'A444 800F		8
Transfer clock control register	CLKON	8	H'A444 8010		8
Command timeout control register	CTOCR	8	H'A444 8011		8
VDD/open drain control register	VDCNT	8	H'A444 8012		8
Transfer byte number count register	TBCR	8	H'A444 8014		8
Mode register	MODER	8	H'A444 8016		8
Command type register	CMDTYR	8	H'A444 8018		8
Response type register	RSPTYR	8	H'A444 8019		8
Transfer block number counter	TBNCR	16	H'A444 801A		16
Response register 0	RSPR0	8	H'A444 8020		8
Response register 1	RSPR1	8	H'A444 8021		8
Response register 2	RSPR2	8	H'A444 8022		8
Response register 3	RSPR3	8	H'A444 8023		8
Response register 4	RSPR4	8	H'A444 8024		8
Response register 5	RSPR5	8	H'A444 8025		8
Response register 6	RSPR6	8	H'A444 8026		8
Response register 7	RSPR7	8	H'A444 8027		8
Response register 8	RSPR8	8	H'A444 8028		8
Response register 9	RSPR9	8	H'A444 8029		8
Response register 10	RSPR10	8	H'A444 802A		8
Response register 11	RSPR11	8	H'A444 802B		8
Response register 12	RSPR12	8	H'A444 802C		8
Response register 13	RSPR13	8	H'A444 802D		8
Response register 14	RSPR14	8	H'A444 802E		8
Response register 15	RSPR15	8	H'A444 802F		8
Response register 16	RSPR16	8	H'A444 8030		8

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Response register D	RSPRD	8	H'A444 8031	MMC	8
Data timeout register	DTOUTR	16	H'A444 8032		16
Data register	DR	16	H'A444 8040		16
FIFO pointer clear register	FIFOCLR	8	H'A444 8042		8
DMA control register	DMACR	8	H'A444 8044		8
Interrupt control register 2	INTCR2	8	H'A444 8046		8
Interrupt status register 2	INTSTR2	8	H'A444 8048		8
Break data register B	BDRB	32	H'A4FF FF90	UBC	32
Break data mask register B	BDMRB	32	H'A4FF FF94		32
Break control register	BRCR	32	H'A4FF FF98		32
Execution times break register	BETR	16	H'A4FF FF9C		16
Break address register B	BARB	32	H'A4FF FFA0		32
Break address mask register B	BAMRB	32	H'A4FF FFA4		32
Break bus cycle register B	BBRB	16	H'A4FF FFA8		16
Branch source register	BRSR	32	H'A4FF FFAC		32
Break address register A	BARA	32	H'A4FF FFB0		32
Break address mask register A	BAMRA	32	H'A4FF FFB4		32
Break bus cycle register A	BBRA	16	H'A4FF FFB8		16
Branch destination register	BRDR	32	H'A4FF FFBC		32
Break ASID register A	BASRA	8	H'FFFF FFE4		8
Break ASID register B	BASRB	8	H'FFFF FFE8		8
Port A control register	PACR	16	H'A405 0100	PFC	16
Port B control register	PBCR	16	H'A405 0102		16
Port C control register	PCCR	16	H'A405 0104		16
Port D control register	PDCR	16	H'A405 0106		16
Port E control register	PECR	16	H'A405 0108		16
Port F control register	PFCR	16	H'A405 010A		16
Port G control register	PGCR	16	H'A405 010C		16
Port H control register	PHCR	16	H'A405 010E		16
Port J control register	PJCR	16	H'A405 0110		16
Port K control register	PKCR	16	H'A405 0112		16

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Port L control register	PLCR	16	H'A405 0114	PFC	16
Port M control register	PMCR	16	H'A405 0116		16
Port P control register	PPCR	16	H'A405 0118		16
Port R control register	PRCR	16	H'A405 011A		16
Port S control register	PSCR	16	H'A405 011C		16
Port T control register	PTCR	16	H'A405 011E		16
Port U control register	PUCR	16	H'A405 0120		16
Port V control register	PVCR	16	H'A405 0122		16
Pin select register A	PSELA	16	H'A405 0124		16
Pin select register B	PSELB	16	H'A405 0126		16
Pin select register C	PSELC	16	H'A405 0128		16
Pin select register D	PSELD	16	H'A405 012A		16
Port A data register	PADR	8	H'A405 0140	I/O port	8
Port B data register	PBDR	8	H'A405 0142		8
Port C data register	PCDR	8	H'A405 0144		8
Port D data register	PDDR	8	H'A405 0146		8
Port E data register	PEDR	8	H'A405 0148		8
Port F data register	PFDR	8	H'A405 014A		8
Port G data register	PGDR	8	H'A405 014C		8
Port H data register	PHDR	8	H'A405 014E		8
Port J data register	PJDR	8	H'A405 0150		8
Port K data register	PKDR	8	H'A405 0152		8
Port L data register	PLDR	8	H'A405 0154		8
Port M data register	PMDR	8	H'A405 0156		8
Port P data register	PPDR	8	H'A405 0158		8
Port R data register	PRDR	8	H'A405 015A		8
Port S data register	PSDR	8	H'A405 015C		8
Port T data register	PTDR	8	H'A405 015E		8
Port U data register	PUDR	8	H'A405 0160		8
Port V data register	PVDR	8	H'A405 0162		8

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Instruction register	SDIR	16	H'A410 0200	H-UDI	16
ID register	SDIDH	16	H'A410 0214		16, 32
ID register	SDIDL	16	H'A410 0216		16

Notes: 1. Entries under Access Size indicate the size for accessing (reading/writing) the control registers. Specifying the sizes other than listed ones results in the wrong operation.
2. 8 bits when reading and 16 bits when writing.

37.2 Register Bits

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

Each line covers eight bits, and 16-bit and 32-bit registers are shown as 2 or 4 lines, respectively.

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
MMUCR	——	—	—	—	—	—	—	—	MMU
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	SV	
	—	—	RC	RC	—	TF	IX	AT	
PTEH	VPN	VPN	VPN	VPN	VPN	VPN	VPN	VPN	
	VPN	VPN	VPN	VPN	VPN	VPN	VPN	VPN	
	VPN	VPN	VPN	VPN	VPN	VPN	—	—	
	ASID	ASID	ASID	ASID	ASID	ASID	ASID	ASID	
PTEL	—	—	—	PPN	PPN	PPN	PPN	PPN	
	PPN	PPN	PPN	PPN	PPN	PPN	PPN	PPN	
	PPN	PPN	PPN	PPN	PPN	PPN	—	V	
	—	PR	PR	SZ	C	D	SH	—	
TTB									
CCR2	—	—	—	—	—	—	—	—	Cache
	—	—	—	—	—	—	—	LE	
	—	—	—	—	—	—	W3LOAD	W3LOCK	
	—	—	—	—	—	—	W2LOAD	W2LOCK	
CCR3	—	—	—	—	—	—	—	—	
	CSIZE7	CSIZE6	CSIZE5	CSIZE4	CSIZE3	CSIZE2	CSIZE1	CSIZE0	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CCR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	CF	CB	WT	CE	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
INTEVT2	—	—	—	—	—	—	—	—	Exception handling
	—	—	—	—	—	—	—	—	
	—	—	—	—	INTEVT2	INTEVT2	INTEVT2	INTEVT2	
	INTEVT2	INTEVT2	INTEVT2	INTEVT2	INTEVT2	INTEVT2	—	—	
TRA	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	TRA	TRA	
	TRA	TRA	TRA	TRA	TRA	TRA	—	—	
EXPEVT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	EXPEVT	EXPEVT	EXPEVT	EXPEVT	
	EXPEVT	EXPEVT	EXPEVT	EXPEVT	EXPEVT	EXPEVT	EXPEVT	EXPEVT	
INTEVT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	INTEVT	INTEVT	INTEVT	INTEVT	
	INTEVT	INTEVT	INTEVT	INTEVT	INTEVT	INTEVT	INTEVT	INTEVT	
TEA	TEA	TEA	TEA	TEA	TEA	TEA	TEA	TEA	
	TEA	TEA	TEA	TEA	TEA	TEA	TEA	TEA	
	TEA	TEA	TEA	TEA	TEA	TEA	TEA	TEA	
	TEA	TEA	TEA	TEA	TEA	TEA	TEA	TEA	
IPRF		ADC				DMAC(2)			INTC
		USBF				CMT			
IPRG		SCIF0				SCIF1			
	—	—	—	—	—	—	—	—	
IPRH		PINTA				PINTB			
		TPU				IIC			
IPRI		SIOF0				SIOF1			
		MMC				PCC			
IPRJ	—	—	—	—		USBH			
		SDHI				AFEIF			
IRR5	ADCIR	—	DEI5R	DEI4R	—	—	SCIF1IR	SCIF0IR	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
IRR6	—	—	SIOF1IR	SIOF0IR	—	—	PINTBR	PINTAR	INTC
IRR7	—	—	—	IICIR	TPI3R	TPI2R	TPI1R	TPI0R	
IRR8	MMCI3R	MMCI2R	MMCI1R	MMCI0R	AFECIR	—	—	SDIR	
IRR9	PCCIR	USBHIR	—	CMIR	—	USBFI1R	USBFI0R	—	
IRR0	—	TMU_ SUNIR	IRQ5R	IRQ4R	IRQ3R	IRQ2R	IRQ1R	IRQ0R	
IRR1	—	—	—	—	DEI3R	DEI2R	DEI1R	DEI0R	
IRR2	—	—	—	SSLIR	—	—	—	LCDCIR	
IRR3	TENDIR	TXIR	RXIR	ERIR	—	CUIR	PRIR	ATIR	
IRR4	—	TUNI2R	TUNI1R	TUNI0R	ITIR	—	—	RCMIR	
ICR1	MAI	IRQLVL	BLMSK	—	IRQ51S	IRQ50S	IRQ41S	IRQ40S	
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S	
ICR2	PINT15S	PINT14S	PINT13S	PINT12S	PINT11S	PINT10S	PINT9S	PINT8S	
	PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S	PINT0S	
PINTER	PINT15E	PINT14E	PINT13E	PINT12E	PINT11E	PINT10E	PINT9E	PINT8E	
	PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT1E	PINT0E	
IPRC		IRQ3				IRQ2			
		IRQ1				IRQ0			
IPRD		—				TMU (TMU_SUN1)			
		IRQ5				IRQ4			
IPRE		DMAC(1)			—	—	—	—	
		LCDC	—	—		SSL			
ICR0	NMIL	—	—	—	—	—	—	NMIE	
	—	—	—	—	—	—	—	—	
IPRA		TMU0				TMU1			
		TMU2				RTC			
IPRB		WDT				REF			
		SIM			—	—	—	—	
CMNCR	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	BSD	—	MAP	BLOCK	DPRTY1	DPRTY0	DMAIW2	
	DMAIW1	DMAIW0	DMAIWA	—	ENDIAN	—	HIZMEM	HIZCNT	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
CS0BCR	—	IWW2	IWW1	IWW0	IWRWD	IWRWD	IWRWD	IWRWS2	BSC
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS2BCR	—	IWW2	IWW1	IWW0	IWRWD	IWRWD	IWRWD	IWRWS2	
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS3BCR	—	IWW2	IWW1	IWW0	IWRWD	IWRWD	IWRWD	IWRWS2	
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS4BCR	—	IWW2	IWW1	IWW0	IWRWD	IWRWD	IWRWD	IWRWS2	
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS5ABCR	—	IWW2	IWW1	IWW0	IWRWD	IWRWD	IWRWD	IWRWS2	
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS5BBCR	—	IWW2	IWW1	IWW0	IWRWD	IWRWD	IWRWD	IWRWS2	
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS6ABCR	—	IWW2	IWW1	IWW0	IWRWD	IWRWD	IWRWD	IWRWS2	
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS6BBCR	—	IWW2	IWW1	IWW0	IWRWD	IWRWD	IWRWD	IWRWS2	
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
CS6BBCR	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	BSC
	—	—	—	—	—	—	—	—	
CS0WCR*[1]	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	—	—	—	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS0WCR*[2]	—	—	—	—	—	—	—	—	
	—	—	—	BEN	—	—	BW1	BW0	
	—	—	—	—	—	W3	W2	W1	
	W0	WM	—	—	—	—	—	—	
CS0WCR*[3]	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	BW1	BW0	
	—	—	—	—	—	W3	W2	W1	
	W0	WM	—	—	—	—	—	—	
CS2WCR*[1]	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	—	—	—	
	—	—	—	—	—	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	—	—	
CS2WCR*[4]	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	A2CL1	
	A2CL0	—	—	—	—	—	—	—	
CS3WCR*[1]	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	—	—	—	
	—	—	—	—	—	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	—	—	
CS3WCR*[4]	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	TRP1	TRP0	—	TRCD1	TRCD0	—	A3CL1	
	A3CL0	—	—	TRWL1	TRWL0	—	TRC1	TRC0	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
CS4WCR*1	—	—	—	—	—	—	—	—	BSC
	—	—	—	BAS	—	WW2	WW1	WW0	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS4WCR*2	—	—	—	—	—	—	—	—	
	—	—	—	BEN	—	—	BW1	BW0	
	—	—	—	SW1	SW0	W3	W2	W1	
	W0	WM	—	—	—	—	HW1	HW0	
CS5AWCR*1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	WW2	WW1	WW0	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS5BWCR*1	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	WW2	WW1	WW0	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS5BWCR*5	—	—	—	—	—	—	—	—	
	—	—	SA1	SA0	—	—	—	—	
	—	TED3	TED2	TED1	TED0	PCW3	PCW2	PCW1	
	PCW0	WM	—	—	TEH3	TEH2	TEH1	TEH0	
CS6AWCR*1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS6BWCR*1	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	—	—	—	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS6BWCR*5	—	—	—	—	—	—	—	—	
	—	—	SA1	SA0	—	—	—	—	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
CS6BWCR*5	—	TED3	TED2	TED1	TED0	PCW3	PCW2	PCW1	BSC
	PCW0	WM	—	—	TEH3	TEH2	TEH1	TEH0	
SDCR	—	—	—	—	—	—	—	—	
	—	—	—	A2ROW1	A2ROW0	—	A2COL1	A2COL0	
	—	—	DEEP	—	RFSH	RMODE	PDOWN	BACTV	
	—	—	—	A3ROW1	A3ROW0	—	A3COL1	A3COL0	
RTCSR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	CMF	CMIE	CKS2	CKS1	CKS0	RRC2	RRC1	RRC0	
RTCNT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
RTCOR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
SAR_0									DMAC
DAR_0									
DMATCR_0									

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
CHCR_0	—	—	—	—	—	—	—	—	DMAC
	DO	TL	—	—	—	—	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	DL	DS	TB	TS1	TS0	IE	TE	DE	
SAR_1									
DAR_1									
DMATCR_1									
CHCR_1	—	—	—	—	—	—	—	—	
	DO	TL	—	—	—	—	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	DL	DS	TB	TS1	TS0	IE	TE	DE	
SAR_2									
DAR_2									
DMATCR_2									

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
DMATCR_2									DMAC
CHCR_2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	—	TB	TS1	TS0	IE	TE	DE	
SAR_3									
DAR_3									
DMATCR_3									
CHCR_3	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	—	TB	TS1	TS0	IE	TE	DE	
SAR_4									
DAR_4									

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
DMATCR_4									DMAC
CHCR_4	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	—	TB	TS1	TS0	IE	TE	DE	
SAR_5									
DAR_5									
DMATCR_5									
CHCR_5	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	—	TB	TS1	TS0	IE	TE	DE	
DMAOR	—	—	CMS1	CMS0	—	—	PR1	PR0	
	—	—	—	—	—	AE	NMIF	DME	
DMARS0	C1MID5	C1MID4	C1MID3	C1MID2	C1MID1	C1MID0	C1RID1	C1RID0	
	C0MID5	C0MID4	C0MID3	C0MID2	C0MID1	C0MID0	C0RID1	C0RID0	
DMARS1	C3MID5	C3MID4	C3MID3	C3MID2	C3MID1	C3MID0	C3RID1	C3RID0	
	C2MID5	C2MID4	C2MID3	C2MID2	C2MID1	C2MID0	C2RID1	C2RID0	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
DMARS2	C5MID5	C5MID4	C5MID3	C5MID2	C5MID1	C5MID0	C5RID1	C5RID0	DMAC
	C4MID5	C4MID4	C4MID3	C4MID2	C4MID1	C4MID0	C4RID1	C4RID0	
UCLKCR	USSCS1	USSCS0	—	—	—	—	—	—	CPG
FRQCR	PLL2EN	—	—	CKOEN	—	—	STC1	STC0	
	—	—	IFC1	IFC0	—	PFC2	PFC1	PFC0	
WTCNT									WDT
WTCSR	TME	WT/IT	RSTS	WOVF	IOVF	CKS2	CKS1	CKS0	
STBCR3	MSTP37	MSTP36	MSTP35	—	MSTP33	MSTP32	MSTP31	MSTP30	Power-down modes
STBCR4	—	—	MSTP45	MSTP44	MSTP43	MSTP42	MSTP41	MSTP40	
STBCR5	—	MSTP56	—	MSTP54	—	MSTP52	MSTP51	MSTP50	
STBCR	STBY	—	—	STBXTL	—	MSTP2	MSTP1	—	
STBCR2	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	—	MSTP3	
TSTR	—	—	—	—	—	STR2	STR1	STR0	TMU
TCOR_0									
TCNT_0									
TCR_0	—	—	—	—	—	—	—	UNF	
	—	—	UNIE	—	—	TPSC2	TPSC1	TPSC0	
TCOR_1									
TCNT_1									

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
TCR_1	—	—	—	—	—	—	—	UNF	TMU
	—	—	UNIE	—	—	TPSC2	TPSC1	TPSC0	
TCOR_2									
TCNT_2									
TCR_2	—	—	—	—	—	—	—	UNF	
	—	—	UNIE	—	—	TPSC2	TPSC1	TPSC0	
TSTR	—	—	—	—	—	—	—	—	TPU
	—	—	—	—	CST3	CST2	CST1	CST0	
TCR_0	—	—	—	—	—	—	—	—	
	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TMDR_0	—	—	—	—	—	—	—	—	
	—	BFWT	BFB	BFA	—	MD2	MD1	MD0	
TIOR_0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	IOA2	IOA1	IOA0	
TIER_0	—	—	—	—	—	—	—	—	
	—	—	TC1EU	TC1EV	TG1ED	TG1EC	TG1EB	TG1EA	
TSR_0	—	—	—	—	—	—	—	—	
	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0									
TGRA_0									
TGRB_0									

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
TGRC_0									TPU
TGRD_0									
TCR_1	—	—	—	—	—	—	—	—	
	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TMDR_1	—	—	—	—	—	—	—	—	
	—	BFWT	BFB	BFA	—	MD2	MD1	MD0	
TIOR_1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	IOA2	IOA1	IOA0	
TIER_1	—	—	—	—	—	—	—	—	
	—	—	TC1EU	TC1EV	TG1ED	TG1EC	TG1EB	TG1EA	
TSR_1	—	—	—	—	—	—	—	—	
	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_1									
TGRA_1									
TGRB_1									
TGRC_1									
TGRD_1									
TCR_2	—	—	—	—	—	—	—	—	
	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TMDR_2	—	—	—	—	—	—	—	—	
	—	BFWT	BFB	BFA	—	MD2	MD1	MD0	
TIOR_2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	IOA2	IOA1	IOA0	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
TIER_2	—	—	—	—	—	—	—	—	TPU
	—	—	TC1EU	TC1EV	TG1ED	TG1EC	TG1EB	TG1EA	
TSR_2	—	—	—	—	—	—	—	—	
	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_2									
TGRA_2									
TGRB_2									
TGRC_2									
TGRD_2									
TCR_3	—	—	—	—	—	—	—	—	
	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TMDR_3	—	—	—	—	—	—	—	—	
	—	BFWT	BFB	BFA	—	MD2	MD1	MD0	
TIOR_3	—	—	—	—	—	—	—	—	
	—	—	—	—	—	IOA2	IOA1	IOA0	
TIER_3	—	—	—	—	—	—	—	—	
	—	—	TC1EU	TC1EV	TG1ED	TG1EC	TG1EB	TG1EA	
TSR_3	—	—	—	—	—	—	—	—	
	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_3									
TGRA_3									
TGRB_3									

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
TGRC_3									TPU
TGRD_3									
CMSTR	—	—	—	—	—	—	—	—	CMT
	—	—	—	STR4	STR3	STR2	STR1	STR0	
CMCSR_0	CMF	OVF	—	—	—	—	CMS	CMM	
	—	—	CMR1	CMR0	—	CKS2	CKS1	CKS0	
CMCNT_0									
CMCOR_0									
CMCSR_1	CMF	OVF	—	—	—	—	CMS	CMM	
		—	CMR1	CMR0	—	CKS2	CKS1	CKS0	
CMCNT_1									
CMCOR_1									
CMCSR_2	CMF	OVF	—	—	—	—	CMS	CMM	
	—	—	CMR1	CMR0	—	CKS2	CKS1	CKS0	
CMCNT_2									

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
CMCNT_2									CMT
CMCOR_2									
CMCSR_3	CMF	OVF	—	—	—	—	CMS	CMM	
	—	—	CMR1	CMR0	—	CKS2	CKS1	CKS0	
CMCNT_3									
CMCOR_3									
CMCSR_4	CMF	OVF	—	—	—	—	CMS	CMM	
	—	—	CMR1	CMR0	—	CKS2	CKS1	CKS0	
CMCNT_4									
CMCOR_4									
R64CNT	—	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz	RTC
RSECCNT	—	Ten's position of seconds				One's position of seconds			
RMINCNT	—	Ten's position of minutes				One's position of minutes			

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
RHRCNT	—	—	Ten's position of hours			One's position of hours			RTC
RWKCNT	—	—	—	—	—		Day of week		
RDAYCNT	—	—	Ten's position of dates			One's position of dates			
RMONCNT	—	—	—	Ten's position of months					
RYRCNT	Thousand's position of years					Hundred's position of years			
	Ten's position of years					One's position of years			
RSECAR	ENB	Ten's position of seconds				One's position of seconds			
RMINAR	ENB	Ten's position of minutes				One's position of minutes			
RHRAR	ENB	—	Ten's position of hours			One's position of hours			
RWKAR	ENB	—	—	—	—		Day of week		
RDAYAR	ENB	—	Ten's position of dates			One's position of dates			
RMONAR	ENB	—	—	Ten's position of months					
RCR1	CF	—	—	CIE	AIE	—	—	AF	
RCR2	PEF	PES2	PES1	PES0	RTCEN	ADJ	RESET	START	
RYRAR	Thousand's position of years					Hundred's position of years			
	Ten's position of years					One's position of years			
RCR3	YAEN	—	—	—	—	—	—	—	
SCSMR_0	—	—	—	—	—	SRC2	SRC1	SRC0	SCIF
	C/A	CHR	PE	O/E	STOP	—	CKS1	CKS0	
SCBRR_0	SCBRD7	SCBRD6	SCBRD5	SCBRD4	SCBRD3	SCBRD2	SCBRD1	SCBRD0	
SCSCR_0	TDRQE	RDRQE	—	—	TSIE	ERIE	BRIE	DRIE	
	TIE	RIE	TE	RE	—	—	CKE1	CKE0	
SCTDSR_0									
SCFER_0	—	—	PER5	PER4	PER3	PER2	PER1	PER0	
	—	—	FER5	FER4	FER3	FER2	FER1	FER0	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
SCSSR_0	—	—	—	—	—	—	ORER	TSF	SCIF
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFCR_0	TSE	TCRST	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	
SCFDR_0	—	T6	T5	T4	T3	T2	T1	T0	
	—	R6	R5	R4	R3	R2	R1	R0	
SCFTDR_0	SCFTD7	SCFTD6	SCFTD5	SCFTD4	SCFTD3	SCFTD2	SCFTD1	SCFTD0	
SCFRDR_0	SCFRD7	SCFRD6	SCFRD5	SCFRD4	SCFRD3	SCFRD2	SCFRD1	SCFRD0	
SCSMR_1	—	—	—	—	—	SRC2	SRC1	SRC0	
	C/A	CHR	PE	O/E	STOP	—	CKS1	CKS0	
SCBRR_1	SCBRD7	SCBRD6	SCBRD5	SCBRD4	SCBRD3	SCBRD2	SCBRD1	SCBRD0	
SCSCR_1	TDRQE	RDRQE	—	—	TSIE	ERIE	BRIE	DRIE	
	TIE	RIE	TE	RE	—	—	CKE1	CKE0	
SCTDSR_1									
SCFER_1	—	—	PER5	PER4	PER3	PER2	PER1	PER0	
	—	—	FER5	FER4	FER3	FER2	FER1	FER0	
SCSSR_1	—	—	—	—	—	—	ORER	TSF	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFCR_1	TSE	TCRST	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	
SCFDR_1	—	T6	T5	T4	T3	T2	T1	T0	
	—	R6	R5	R4	R3	R2	R1	R0	
SCFTDR_1	SCFTD7	SCFTD6	SCFTD5	SCFTD4	SCFTD3	SCFTD2	SCFTD1	SCFTD0	
SCFRDR_1	SCFRD7	SCFRD6	SCFRD5	SCFRD4	SCFRD3	SCFRD2	SCFRD1	SCFRD0	
SCIMR	—	—	—	—	—	—	—	—	IrDA
	IRMOD	ICK3	ICK2	ICK1	ICK0	PSEL	—	—	
ICCR1	ICE	RCVD	MST	TRS	—	—	—	—	IIC
ICCR2	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—	
ICMR	MLS	—	—	—	BCWP	BC2	BC1	BC0	
ICIER	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
SAR	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	IIC
ICDRT	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	
SIMDR_0	TRMD1	TRMD0	SYNCAT	REDG	FL3	FL2	FL1	FL0	SIOF
	TXDIZ	RCIM	SYNCAC	SYNCDL	—	—	—	—	
SISCR_0	MSSEL	MSIMM	—	BRPS4	BRPS3	BRPS2	BRPS1	BRPS0	
	—	—	—	—	—	BRDV2	BRDV1	BRDV0	
SITDAR_0	TDLE	—	—	—	TDLA3	TDLA2	TDLA1	TDLA0	
	TDRE	TLREP	—	—	TDRA3	TDRA2	TDRA1	TDRA0	
SIRDAR_0	RDLE	—	—	—	RDLA3	RDLA2	RDLA1	RDLA0	
	RDRE	—	—	—	RDRA3	RDRA2	RDRA1	RDRA0	
SICDAR_0	CD0E	—	—	—	CD0A3	CD0A2	CD0A1	CD0A0	
	CD1E	—	—	—	CD1A3	CD1A2	CD1A1	CD1A0	
SICTR_0	SCKE	FSE	—	—	—	—	TXE	RXE	
	—	—	—	—	—	—	TXRST	RXRST	
SIFCTR_0	TFWM2	TFWM1	TFWM0	TFUA4	TFUA3	TFUA2	TFUA1	TFUA0	
	RFWM2	RFWM1	RFWM0	RFUA4	RFUA3	RFUA2	RFUA1	RFUA0	
SISTR_0	—	TCRDY	TFEMP	TDREQ	—	RCRDY	RFFUL	RDREQ	
	—	—	SAERR	FSERR	TFOVF	TFUDF	RFUDF	RFOVF	
SIIER_0	TDMAE	TCRDYE	TFEMPE	TDREQE	RDMAE	RCRDYE	RFFULE	RDREQE	
	—	—	SAERRE	FSERRE	TFOVFE	TFUDFE	RFUDFE	RFOVFE	
SITDR_0	SITDL15	SITDL14	SITDL13	SITDL12	SITDL11	SITDL10	SITDL9	SITDL8	
	SITDL7	SITDL6	SITDL5	SITDL4	SITDL3	SITDL2	SITDL1	SITDL0	
	SITDR15	SITDR14	SITDR13	SITDR12	SITDR11	SITDR10	SITDR9	SITDR8	
	SITDR7	SITDR6	SITDR5	SITDR4	SITDR3	SITDR2	SITDR1	SITDR0	
SIRDR_0	SIRDL15	SIRDL14	SIRDL13	SIRDL12	SIRDL11	SIRDL10	SIRDL9	SIRDL8	
	SIRDL7	SIRDL6	SIRDL5	SIRDL4	SIRDL3	SIRDL2	SIRDL1	SIRDL0	
	SIRDR15	SIRDR14	SIRDR13	SIRDR12	SIRDR11	SIRDR10	SIRDR9	SIRDR8	
	SIRDR7	SIRDR6	SIRDR5	SIRDR4	SIRDR3	SIRDR2	SIRDR1	SIRDR0	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
SITCR_0	SITC015	SITC014	SITC013	SITC012	SITC011	SITC010	SITC09	SITC08	SIOF
	SITC07	SITC06	SITC05	SITC04	SITC03	SITC02	SITC01	SITC00	
	SITC115	SITC114	SITC113	SITC112	SITC111	SITC110	SITC19	SITC18	
SITCR_0	SITC17	SITC16	SITC15	SITC14	SITC13	SITC12	SITC11	SITC10	
SIRCR_0	SIRC015	SIRC014	SIRC013	SIRC012	SIRC011	SIRC010	SIRC09	SIRC08	
	SIRC07	SIRC06	SIRC05	SIRC04	SIRC03	SIRC02	SIRC01	SIRC00	
	SIRC115	SIRC114	SIRC113	SIRC112	SIRC111	SIRC110	SIRC19	SIRC18	
	SIRC17	SIRC16	SIRC15	SIRC14	SIRC13	SIRC12	SIRC11	SIRC10	
SIMDR_1	TRMD1	TRMD0	SYNCAT	REDG	FL3	FL2	FL1	FL0	
	TXDIZ	RCIM	SYNCAC	SYNCDL	—	—	—	—	
SISCR_1	MSSEL	MSIMM	—	BRPS4	BRPS3	BRPS2	BRPS1	BRPS0	
	—	—	—	—	—	BRDV2	BRDV1	BRDV0	
SITDAR_1	TDLE	—	—	—	TDLA3	TDLA2	TDLA1	TDLA0	
	TDRE	TLREP	—	—	TDRA3	TDRA2	TDRA1	TDRA0	
SIRDAR_1	RDLE	—	—	—	RDLA3	RDLA2	RDLA1	RDLA0	
	RDRE	—	—	—	RDRA3	RDRA2	RDRA1	RDRA0	
SICDAR_1	CD0E	—	—	—	CD0A3	CD0A2	CD0A1	CD0A0	
	CD1E	—	—	—	CD1A3	CD1A2	CD1A1	CD1A0	
SICTR_1	SCKE	FSE	—	—	—	—	TXE	RXE	
	—	—	—	—	—	—	TXRST	RXRST	
SIFCTR_1	TFWM2	TFWM1	TFWM0	TFUA4	TFUA3	TFUA2	TFUA1	TFUA0	
	RFWM2	RFWM1	RFWM0	RFUA4	RFUA3	RFUA2	RFUA1	RFUA0	
SISTR_1	—	TCRDY	TFEMP	TDREQ	—	RCRDY	RFFUL	RDREQ	
	—	—	SAERR	FSERR	TFOVF	TFUDF	RFUDF	RFOVF	
SIIER_1	TDMAE	TCRDYE	TFEMPE	TDREQE	RDMAE	RCRDYE	RFFULE	RDREQE	
	—	—	SAERRE	FSERRE	TFOVFE	TFUDFE	RFUDFE	RFOVFE	
SITDR_1	SITDL15	SITDL14	SITDL13	SITDL12	SITDL11	SITDL10	SITDL9	SITDL8	
	SITDL7	SITDL6	SITDL5	SITDL4	SITDL3	SITDL2	SITDL1	SITDL0	
	SITDR15	SITDR14	SITDR13	SITDR12	SITDR11	SITDR10	SITDR9	SITDR8	
	SITDR7	SITDR6	SITDR5	SITDR4	SITDR3	SITDR2	SITDR1	SITDR0	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
SIRDR_1	SIRDL15	SIRDL14	SIRDL13	SIRDL12	SIRDL11	SIRDL10	SIRDL9	SIRDL8	SIOF
	SIRDL7	SIRDL6	SIRDL5	SIRDL4	SIRDL3	SIRDL2	SIRDL1	SIRDL0	
	SIRDR15	SIRDR14	SIRDR13	SIRDR12	SIRDR11	SIRDR10	SIRDR9	SIRDR8	
	SIRDR7	SIRDR6	SIRDR5	SIRDR4	SIRDR3	SIRDR2	SIRDR1	SIRDR0	
SITCR_1	SITC015	SITC014	SITC013	SITC012	SITC011	SITC010	SITC09	SITC08	
	SITC07	SITC06	SITC05	SITC04	SITC03	SITC02	SITC01	SITC00	
	SITC115	SITC114	SITC113	SITC112	SITC111	SITC110	SITC19	SITC18	
SITCR_1	SITC17	SITC16	SITC15	SITC14	SITC13	SITC12	SITC11	SITC10	
SIRCR_1	SIRC015	SIRC014	SIRC013	SIRC012	SIRC011	SIRC010	SIRC09	SIRC08	
	SIRC07	SIRC06	SIRC05	SIRC04	SIRC03	SIRC02	SIRC01	SIRC00	
	SIRC115	SIRC114	SIRC113	SIRC112	SIRC111	SIRC110	SIRC19	SIRC18	
	SIRC17	SIRC16	SIRC15	SIRC14	SIRC13	SIRC12	SIRC11	SIRC10	
ACTR1	HC	—	—	—	—	—	—	—	AFEIF
	DLB	—	—	FFSZ2	FFSZ1	FFSZ0	TE	RE	
ACTR2	—	—	—	—	—	—	—	—	
	—	—	—	DPST	PPS	RCEN	—	RLYC	
ASTR1	—	—	—	—	TFEM	RFFM	THEM	RHFM	
	—	—	—	—	TFE	RFF	THE	RHF	
ASTR2	—	—	—	—	—	—	DPEM	RDETM	
	—	—	—	—	—	—	DPE	RDET	
MRCR	—	—	—	—	—	—	MRCR9	MRCR8	
	MRCR7	MRCR6	MRCR5	MRCR4	MRCR3	MRCR2	MRCR1	MRCR0	
MPCR	MPCR15	MPCR14	MPCR13	MPCR12	MPCR11	MPCR10	MPCR9	MPCR8	
	MPCR7	MPCR6	MPCR5	MPCR4	MPCR3	MPCR2	MPCR1	MPCR0	
DPNQ	DN03	DN02	DN01	DN00	DN13	DN12	DN11	DN10	
	DN23	DN22	DN21	DN20	DN33	DN32	DN31	DN30	
RCNT	RCNT15	RCNT14	RCNT13	RCNT12	RCNT11	RCNT10	RCNT9	RCNT8	
	RCNT7	RCNT6	RCNT5	RCNT4	RCNT3	RCNT2	RCNT1	RCNT0	
ACDR	ACDR15	ACDR14	ACDR13	ACDR12	ACDR11	ACDR10	ACDR9	ACDR8	
	ACDR7	ACDR6	ACDR5	ACDR4	ACDR3	ACDR2	ACDR1	ACDR0	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
ASDR	ASDR15	ASDR14	ASDR13	ASDR12	ASDR11	ASDR10	ASDR9	ASDR8	AFEIF
	ASDR7	ASDR6	ASDR5	ASDR4	ASDR3	ASDR2	ASDR1	ASDR0	
TDFP	TDFP15	TDFP14	TDFP13	TDFP12	TDFP11	TDFP10	TDFP9	TDFP8	
	TDFP7	TDFP6	TDFP5	TDFP4	TDFP3	TDFP2	TDFP1	TDFP0	
RDFP	RDFP15	RDFP14	RDFP13	RDFP12	RDFP11	RDFP10	RDFP9	RDFP8	
	RDFP7	RDFP6	RDFP5	RDFP4	RDFP3	RDFP2	RDFP1	RDFP0	
UTRCTL	—	—	—	—	—	—	—	DRV	USB PMC
	—	—	—	—	—	—	USB_ TRANS	USB_SEL	
USBHR	—	—	—	—	—	—	—	—	USBH
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	Rev7	Rev6	Rev5	Rev4	Rev3	Rev2	Rev1	Rev0	
USBHC	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	RWE	RWC	IR	
	HCFS1	HCFS0	BLE	CLE	IE	PLE	CBSR1	CBSR0	
USBHCS	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	SOC1	SOC0	
	—	—	—	—	—	—	—	—	
	—	—	—	—	OCR	BLF	CLF	HCR	
USBHIS	—	OC	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	RHSC	FNO	UE	RD	SF	WDH	SO	
USBHIE	MIE	OC	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	RHSC	FNO	UE	RD	SF	WDH	SO	
USBHID	MIE	OC	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
USBHID	—	—	—	—	—	—	—	—	USBH
	—	RHSC	FNO	UE	RD	SF	WDH	SO	
USBHHCCA	HCCA23	HCCA22	HCCA21	HCCA20	HCCA19	HCCA18	HCCA17	HCCA16	
	HCCA15	HCCA14	HCCA13	HCCA12	HCCA11	HCCA10	HCCA9	HCCA8	
	HCCA7	HCCA6	HCCA5	HCCA4	HCCA3	HCCA2	HCCA1	HCCA0	
	—	—	—	—	—	—	—	—	
USBHPCED	PCED27	PCED26	PCED25	PCED24	PCED23	PCED22	PCED21	PCED20	
	PCED19	PCED18	PCED17	PCED16	PCED15	PCED14	PCED13	PCED12	
	PCED11	PCED10	PCED9	PCED8	PCED7	PCED6	PCED5	PCED4	
	PCED3	PCED2	PCED1	PCED0	—	—	—	—	
USBHCHED	CHED27	CHED26	CHED25	CHED24	CHED23	CHED22	CHED21	CHED20	
	CHED19	CHED18	CHED17	CHED16	CHED15	CHED14	CHED13	CHED12	
	CHED11	CHED10	CHED9	CHED8	CHED7	CHED6	CHED5	CHED4	
	CHED3	CHED2	CHED1	CHED0	—	—	—	—	
USBHCCED	CCED27	CCED26	CCED25	CCED24	CCED23	CCED22	CCED21	CCED20	
	CCED19	CCED18	CCED17	CCED16	CCED15	CCED14	CCED13	CCED12	
	CCED11	CCED10	CCED9	CCED8	CCED7	CCED6	CCED5	CCED4	
	CCED3	CCED2	CCED1	CCED0	—	—	—	—	
USBHBHED	BHED27	BHED26	BHED25	BHED24	BHED23	BHED22	BHED21	BHED20	
	BHED19	BHED18	BHED17	BHED16	BHED15	BHED14	BHED13	BHED12	
	BHED11	BHED10	BHED9	BHED8	BHED7	BHED6	BHED5	BHED4	
	BHED3	BHED2	BHED1	BHED0	—	—	—	—	
USBHBCED	BCED27	BCED26	BCED25	BCED24	BCED23	BCED22	BCED21	BCED20	
	BCED19	BCED18	BCED17	BCED16	BCED15	BCED14	BCED13	BCED12	
	BCED11	BCED10	BCED9	BCED8	BCED7	BCED6	BCED5	BCED4	
	BCED3	BCED2	BCED1	BCED0	—	—	—	—	
USBHDHED	DH27	DH26	DH25	DH24	DH23	DH22	DH21	DH20	
	DH19	DH18	DH17	DH16	DH15	DH14	DH13	DH12	
	DH11	DH10	DH9	DH8	DH7	DH6	DH5	DH4	
	DH3	DH2	DH1	DH0	—	—	—	—	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
USBHFI	FIT	FSMPS14	FSMPS13	FSMPS12	FSMPS11	FSMPS10	FSMPS9	FSMPS8	USBH
	FSMPS7	FSMPS6	FSMPS5	FSMPS4	FSMPS3	FSMPS2	FSMPS1	FSMPS0	
	—	—	FI13	FI12	FI11	FI10	FI9	FI8	
	FI7	FI6	FI5	FI4	FI3	FI2	FI1	FI0	
USBHFR	FRT	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	FR13	FR12	FR11	FR10	FR9	FR8	
	FR7	FR6	FR5	FR4	FR3	FR2	FR1	FR0	
USBHFN	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	FN15	FN14	FN13	FN12	FN11	FN10	FN9	FN8	
	FN7	FN6	FN5	FN4	FN3	FN2	FN1	FN0	
USBHPS	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	PS13	PS12	PS11	PS10	PS9	PS8	
	PS7	PS6	PS5	PS4	PS3	PS2	PS1	PS0	
USBHLST	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	LST11	LST10	LST9	LST8	
	LST7	LST6	LST5	LST4	LST3	LST2	LST1	LST0	
USBHRDA	POTPGT7	POTPGT6	POTPGT5	POTPGT4	POTPGT3	POTPGT2	POTPGT1	POTPGT0	
	—	—	—	—	—	—	—	—	
	—	—	—	NOCP	OCPM	DT	NPS	PSM	
	NDP7	NDP6	NDP5	NDP4	NDP3	NDP2	NDP1	NDP0	
USBHRDB	PPCM15	PPCM14	PPCM13	PPCM12	PPCM11	PPCM10	PPCM9	PPCM8	
	PPCM7	PPCM6	PPCM5	PPCM4	PPCM3	PPCM2	PPCM1	PPCM0	
	DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8	
	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0	
USBHRS	CRWE	—	—	—	—	—	—	—	
	—	—	—	—	—	—	OCIC	LPSC	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
USBHRS	DRWE	—	—	—	—	—	—	—	USBH
	—	—	—	—	—	—	OCI	LPS	
USBHRPS1	—	—	—	—	—	—	—	—	
	—	—	—	PRSC	OCIC	PSSC	PESC	CSC	
	—	—	—	—	—	—	LSDA	PPS	
	—	—	—	PRS	POCI	PSS	PES	CCS	
USBHRPS2	—	—	—	—	—	—	—	—	
	—	—	—	PRSC	OCIC	PSSC	PESC	CSC	
	—	—	—	—	—	—	LSDA	PPS	
	—	—	—	PRS	POCI	PSS	PES	CCS	
IFR0	BRST	EP1FULL	EP2TR	EP2EMPTY	SETUPTS	EP0oTS	EP0iTR	EP0iTS	USBF
IFR1	—	—	—	—	VBUSMN	EP3TR	EP3TS	VBUSF	
IFR2	—	—	SURSS	SURSF	CFDN	SOF	SETC	SETI	
IFR3	—	—	—	—	EP5TR	EP5TS	EP4TF	EP4TS	
IER0	BRST IE	EP1FULL IS	EP2TR IE	EP2 EMPTY IE	SETUPTS IE	EP0oTS IE	EP0iTR IE	EP0iTS IE	
IER1	—	—	—	—	—	EP3TR IE	EP3TS IE	VBUSF IE	
IER2	—	—	—	SURSE IE	CFDN IE	SOFE IE	SETCE IE	SETIE IE	
IER3	—	—	—	—	EP5TR IE	EP5TS IE	EP4TF IE	EP4TS IE	
ISR0	BRST IS	EP1FULL IS	EP2TR IS	EP2 EMPTY IS	SETUPTS IS	EP0oTS IS	EP0iTR IS	EP0iTS IS	
ISR1	—	—	—	—	—	EP3TR IS	EP3TS IS	VBUSF IS	
ISR2	—	—	—	SURSE IS	CFDN IS	SOFE IS	SETCE IS	SETIE IS	
ISR3	—	—	—	—	EP5TR IS	EP5TS IS	EP4TF IS	EP4TS IS	
EPDR0i	D7	D6	D5	D4	D3	D2	D1	D0	
EPDR0o	D7	D6	D5	D4	D3	D2	D1	D0	
EPDR0s	D7	D6	D5	D4	D3	D2	D1	D0	
EPDR1	D7	D6	D5	D4	D3	D2	D1	D0	
EPDR2	D7	D6	D5	D4	D3	D2	D1	D0	
EPDR3	D7	D6	D5	D4	D3	D2	D1	D0	
EPDR4	D7	D6	D5	D4	D3	D2	D1	D0	
EPDR5	D7	D6	D5	D4	D3	D2	D1	D0	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
EPSZ0o									USBF
EPSZ1									
EPSZ4									
DASTS	—	—	EP3DE	EP2DE	—	—	—	EP0iDE	
FCLR0	—	EP3CLR	EP1CLR	EP2CLR	—	—	EP0oCLR	EP0iCLR	
FCLR1	—	—	—	EP5CCLR	—	—	EP5CLR	EP4CLR	
EPSTL0	—	—	—	—	EP3STL	EP2STL	EP1STL	EP0STL	
EPSTL1	—	—	—	—	—	—	EP5STL	EP4STL	
TRG	—	EP3PKTE	EP1RDFN	EP2PKTE	—	EP0sRDFN	EP0oRDFN	EP0iPKTE	
DMA	—	—	—	—	—	PULLUPE	EP2DMAE	EP1DMAE	
CVR	CNFV1	CNFV0	INTV1	INTV0	—	ALTV2	ALTV1	ALTV0	
CTLR0	—	—	—	RWUPS	RSME	—	ASCE	—	
TSRH	—	—	—	—	—	D10	D9	D8	
TSRL	D7	D6	D5	D4	D3	D2	D1	D0	
EPIR	D7	D6	D5	D4	D3	D2	D1	D0	
IFR4	—	—	—	—	—	—	—	TMOUT	
IER4	—	—	—	—	—	—	—	TMOUT IE	
ISR4	—	—	—	—	—	—	—	TMOUT IS	
CTLR1	—	—	—	—	—	—	TMRACLR	TMREN	
TMRH	D15	D14	D13	D12	D11	D10	D9	D8	
TMRL	D7	D6	D5	D4	D3	D2	D1	D0	
STOH	D15	D14	D13	D12	D11	D10	D9	D8	
STOL	D7	D6	D5	D4	D3	D2	D1	D0	
LDPRnn (nn:H'00 to H'FF)	—	—	—	—	—	—	—	—	LCDC
	PALDnn23	PALDnn22	PALDnn21	PALDnn20	PALDnn19	PALDnn18	PALDnn17	PALDnn16	
	PALDnn15	PALDnn14	PALDnn13	PALDnn12	PALDnn11	PALDnn10	PALDnn9	PALDnn8	
	PALDnn7	PALDnn6	PALDnn5	PALDnn4	PALDnn3	PALDnn2	PALDnn1	PALDnn0	
LDICKR	—	—	ICKSEL1	ICKSEL0	—	—	—	—	
	—	—	DCDR5	DCDR4	DCDR3	DCDR2	DCDR1	DCDR0	
LDMTR	FLMPOL	CL1POL	DISPPOL	DPOL	—	MCNT	CL1CNT	CL2CNT	
	—	—	MIFTYP5	MIFTYP4	MIFTYP3	MIFTYP2	MIFTYP1	MIFTYP0	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
LDDFR	—	—	—	—	—	—	—	PABD	LCDC
	—	DSP COLOR6	DSP COLOR5	DSP COLOR4	DSP COLOR3	DSP COLOR2	DSP COLOR1	DSP COLOR0	
LDSMR	—	—	ROT	—	—	—	AU1	AU0	
	—	—	—	—	—	—	—	—	
LDSARU	—	—	—	—	—	—	SAU25	SAU24	
	SAU23	SAU22	SAU21	SAU20	SAU19	SAU18	SAU17	SAU16	
	SAU15	SAU14	SAU13	SAU12	SAU11	SAU10	SAU9	SAU8	
	SAU7	SAU6	SAU5	SAU4	—	—	—	—	
LDSARL	—	—	—	—	—	—	SAL25	SAL24	
	SAL23	SAL22	SAL21	SAL20	SAL19	SAL18	SAL17	SAL16	
	SAL15	SAL14	SAL13	SAL12	SAL11	SAL10	SAL9	SAL8	
	SAL7	SAL6	SAL5	SAL4	—	—	—	—	
LDLAOR	LAO15	LAO14	LAO13	LAO12	LAO11	LAO10	LAO9	LAO8	
	LAO7	LAO6	LAO5	LAO4	LAO3	LAO2	LAO1	LAO0	
LDPALCR	—	—	—	—	—	—	—	—	
	—	—	—	PALS	—	—	—	PALEN	
LDHCNR	HDCN7	HDCN6	HDCN5	HDCN4	HDCN3	HDCN2	HDCN1	HDCN0	
	HTCN7	HTCN6	HTCN5	HTCN4	HTCN3	HTCN2	HTCN1	HTCN0	
LDHSYNR	HSYNW3	HSYNW2	HSYNW1	HSYNW0	—	—	—	—	
	HSYNP7	HSYNP6	HSYNP5	HSYNP4	HSYNP3	HSYNP2	HSYNP1	HSYNP0	
LDVDLNR	—	—	—	—	—	VDLN10	VDLN9	VDLN8	
	VDLN7	VDLN6	VDLN5	VDLN4	VDLN3	VDLN2	VDLN1	VDLN0	
LDVTLNR	—	—	—	—	—	VTLN10	VTLN9	VTLN8	
	VTLN7	VTLN6	VTLN5	VTLN4	VTLN3	VTLN2	VTLN1	VTLN0	
LDVSYNR	VSYNW3	VSYNW2	VSYNW1	VSYNW0	—	VSYNP10	VSYNP9	VSYNP8	
	VSYNP7	VSYNP6	VSYNP5	VSYNP4	VSYNP3	VSYNP2	VSYNP1	VSYNP0	
LDACLNR	—	—	—	—	—	—	—	—	
	—	—	—	ACLN4	ACLN3	ACLN2	ACLN1	ACLN0	
LDINTR	MINTEN	FINTEN	VSINTEN	VEINTEN	MINTS	FINTS	VSINTS	VEINTS	
	—	—	—	—	—	—	—	—	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
LDPMMR	ONC3	ONC2	ONC1	ONC0	OFFD3	OFFD2	OFFD1	OFFD0	LCDC
	—	VCPE	VEPE	DONE	—	—	LPS1	LPS0	
LDPSPR	ONA3	ONA2	ONA1	ONA0	ONB3	ONB2	ONB1	ONB0	
	OFFE3	OFFE2	OFFE1	OFFE0	OFFF3	OFFF2	OFFF1	OFFF0	
LDCNTR	—	—	—	—	—	—	—	—	
	—	—	—	DON2	—	—	—	DON	
LDUINTR	—	—	—	—	—	—	—	UINTEN	
	—	—	—	—	—	—	—	UINTS	
LDUINTLNR	—	—	—	—	—	UINTLN10	UINTLN9	UINTLN8	
	UINTLN7	UINTLN6	UINTLN5	UINTLN4	UINTLN3	UINTLN2	UINTLN1	UINTLN0	
LDLIRNR	—	—	—	—	—	—	—	—	
	LIRN7	LIRN6	LIRN5	LIRN4	LIRN3	LIRN2	LIRN1	LIRN0	
ADDRA	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	ADC
	AD1	AD0	—	—	—	—	—	—	
ADDRB	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRC	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRD	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	MULTI1	CKS	CH2	CH1	CH0	
ADCR	TRG1	TRG0	SCN	TESVD1	RESVD2	—			
DADR0									DAC
DADR1									
DACR	DAOE1	DAOE0	—	—	—	—	—	—	
PCC0ISR	P0RDY/ IREQ	P0MWP	P0VS2	P0VS1	P0CD2	P0CD1	P0BVD2/ P0SPKR	P0BVD1/ P0STSCHG	PCC
PCC0GCR	P0DRVE	P0PCCR	P0PCCT	P0USE	P0MMOD	P0PA25	P0PA24	P0REG	
PCC0CSCR	P0SCDI	—	P0IREQ	P0SC	P0CDC	P0RC	P0BW	P0BD	
PCC0CSCIER	P0CRE	IREQE1	IREQE0	P0SCE	P0CDE	P0RE	P0BWE	P0BDE	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
SCSMR	—	—	—	O/E	—	—	—	—	SIM
SCBRR	—	—	—	—	—	BRR2	BRR1	BRR0	
SCSCR	TIE	RIE	TE	RE	WAIT_IE	TEIE	CKE1	CKE0	
SCTDR	SCTD7	SCTD6	SCTD5	SCTD4	SCTD3	SCTD2	SCTD1	SCTD0	
SCSSR	TDRE	RDRF	ORER	ERS	PER	TEND	WAIT_ER	—	
SCRDR	SCRD7	SCRD6	SCRD5	SCRD4	SCRD3	SCRD2	SCRD1	SCRD0	
SCSCMR	—	LCB	PB	—	SDIR	SINV	RST	SMIF	
SCSC2R	EIO	—	—	—	—	—	—	—	
SCGRD	SCGRD7	SCGRD6	SCGRD5	SCGRD4	SCGRD3	SCGRD2	SCGRD1	SCGRD0	
SCWAIT	SCWAIT15	SCWAIT14	SCWAIT13	SCWAIT12	SCWAIT11	SCWAIT10	SCWAIT9	SCWAIT8	
	SCWAIT7	SCWAIT6	SCWAIT5	SCWAIT4	SCWAIT3	SCWAIT2	SCWAIT1	SCWAIT0	
SCSMPL	—	—	—	—	—	SCSMPL10	SCSMPL9	SCSMPL8	
	SCSMPL7	SCSMPL6	SCSMPL5	SCSMPL4	SCSMPL3	SCSMPL2	SCSMPL1	SCSMPL0	
CMDR0	START	HOST	INDEX5	INDEX4	INDEX3	INDEX2	INDEX1	INDEX0	MMC
CMDR1	CMDR17	CMDR16	CMDR15	CMDR14	CMDR13	CMDR12	CMDR11	CMDR10	
CMDR2	CMDR27	CMDR26	CMDR25	CMDR24	CMDR23	CMDR22	CMDR21	CMDR20	
CMDR3	CMDR37	CMDR36	CMDR35	CMDR34	CMDR33	CMDR32	CMDR31	CMDR30	
CMDR4	CMDR47	CMDR46	CMDR45	CMDR44	CMDR43	CMDR42	CMDR41	CMDR40	
CMDR5	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0	END	
CMDSTRT	—	—	—	—	—	—	—	START	
OPCR	CMDOFF	—	RD_CONTI	DATAEN	—	—	—	—	
CSTR	BUSY	FIFO_ FULL	FIFO_ EMPTY	CWRE	DTBUSY	DTBUSY_ TU	—	REQ	
INTCR0	FEIE	FFIE	DRPIE	DTIE	CRPIE	CMDIE	DBSYIE	BTIE	
INTCR1	INTQ2E	INTQ1E	INTQ0E	—	—	CRCERIE	DTERIE	CTERIE	
INTSTR0	FEI	FFI	DRPI	DTI	CRPI	CMDI	DBSYI	BTI	
INTSTR1	—	—	—	—	WRERI	CRCERI	DTERI	CTERI	
CLKON	CLKON	—	—	—	CSEL3	CSEL2	CSEL1	CSEL0	
CTOCR	—	—	—	—	—	—	—	CTSEL0	
VDCNT	VDDON	ODMOD	—	—	—	—	—	—	
TBCR	—	—	—	—	C3	C2	C1	C0	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
MODER	—	—	—	—	—	—	—	—	MMC
CMDTYR	—	TY6	TY5	TY4	TY3	TY2	TY1	TY0	
RSPTYR	—	—	RTY5	RTY4	RTY3	RTY2	RTY1	RTY0	
TBNCR	TBNCR15	TBNCR14	TBNCR13	TBNCR12	TBNCR11	TBNCR10	TBNCR9	TBNCR8	
	TBNCR7	TBNCR6	TBNCR5	TBNCR4	TBNCR3	TBNCR2	TBNCR1	TBNCR0	
RSPR0	RSPR07	RSPR06	RSPR05	RSPR04	RSPR03	RSPR02	RSPR01	RSPR00	
RSPR1	RSPR17	RSPR16	RSPR15	RSPR14	RSPR13	RSPR12	RSPR11	RSPR10	
RSPR2	RSPR27	RSPR26	RSPR25	RSPR24	RSPR23	RSPR22	RSPR21	RSPR20	
RSPR3	RSPR37	RSPR36	RSPR35	RSPR34	RSPR33	RSPR32	RSPR31	RSPR30	
RSPR4	RSPR47	RSPR46	RSPR45	RSPR44	RSPR43	RSPR42	RSPR41	RSPR40	
RSPR5	RSPR57	RSPR56	RSPR55	RSPR54	RSPR53	RSPR52	RSPR51	RSPR50	
RSPR6	RSPR67	RSPR66	RSPR65	RSPR64	RSPR63	RSPR62	RSPR61	RSPR60	
RSPR7	RSPR77	RSPR76	RSPR75	RSPR74	RSPR73	RSPR72	RSPR71	RSPR70	
RSPR8	RSPR87	RSPR86	RSPR85	RSPR84	RSPR83	RSPR82	RSPR81	RSPR80	
RSPR9	RSPR97	RSPR96	RSPR95	RSPR94	RSPR93	RSPR92	RSPR91	RSPR90	
RSPR10	RSPR107	RSPR106	RSPR105	RSPR104	RSPR103	RSPR102	RSPR101	RSPR100	
RSPR11	RSPR117	RSPR116	RSPR115	RSPR114	RSPR113	RSPR112	RSPR111	RSPR110	
RSPR12	RSPR127	RSPR126	RSPR125	RSPR124	RSPR123	RSPR122	RSPR121	RSPR120	
RSPR13	RSPR137	RSPR136	RSPR135	RSPR134	RSPR133	RSPR132	RSPR131	RSPR130	
RSPR14	RSPR147	RSPR146	RSPR145	RSPR144	RSPR143	RSPR142	RSPR141	RSPR140	
RSPR15	RSPR157	RSPR156	RSPR155	RSPR154	RSPR153	RSPR152	RSPR151	RSPR150	
RSPR16	RSPR167	RSPR166	RSPR165	RSPR164	RSPR163	RSPR162	RSPR161	RSPR160	
RSPRD	—	—	—	RSPRD4	RSPRD3	RSPRD2	RSPRD1	RSPRD0	
DTOUTR	DTOUTR15	DTOUTR14	DTOUTR13	DTOUTR12	DTOUTR11	DTOUTR10	DTOUTR9	DTOUTR8	
	DTOUTR7	DTOUTR6	DTOUTR5	DTOUTR4	DTOUTR3	DTOUTR2	DTOUTR1	DTOUTR0	
DR	DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8	
	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0	
FIFOCLR	FIFOCLR7	FIFOCLR6	FIFOCLR5	FIFOCLR4	FIFOCLR3	FIFOCLR2	FIFOCLR1	FIFOCLR0	
DMACR	DMAEN	AUTO	—	—	—	SET2	SET1	SET0	
INTCR2	INTRQ3E	—	—	—	—	—	—	FRDYIE	
INTSTR2	—	—	—	—	—	—	FRDY_TU	FRDYI	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
BDRB	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24	UBC
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16	
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8	
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0	
BDMRB	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24	
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16	
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8	
	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0	
BRCR	—	—	—	—	—	—	—	—	
	—	—	BASWA	BASWB	—	—	—	—	
	SCMFCA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—	—	
	DBEB	PCBB	—	—	SEQ	—	—	ETBE	
BETR	—	—	—	—	BET11	BET10	BET9	BET8	
	BET7	BET6	BET5	BET4	BET3	BET2	BET1	BET0	
BARB	BAB31	BAB30	BAB29	BAB28	BAB27	BAB26	BAB25	BAB24	
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16	
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0	
BAMRB	BAMB31	BAMB30	BAMB29	BAMB28	BAMB27	BAMB26	BAMB25	BAMB24	
	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB16	
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8	
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0	
BBRB	—	—	—	—	—	—	XYE	XYS	
	CDB1	CDB0	IDB1	IDB0	RWB1	RWB0	SZB1	SZB0	
BRSR	SVF	—	—	—	BSA27	BSA26	BSA25	BSA24	
	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17	BSA16	
	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BSA8	
	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BSA0	
BARA	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24	
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
BARA	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	UBC
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	
BAMRA	BAMA31	BAMA30	BAMA29	BAMA28	BAMA27	BAMA26	BAMA25	BAMA24	
	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA17	BAMA16	
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9	BAMA8	
	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1	BAMA0	
BBRA	—	—	—	—	—	—	—	—	
	CDA1	CDA0	IDA1	IDA0	RWA1	RWA0	SZA1	SZA0	
BRDR	DVF	—	—	—	BDA27	BDA26	BDA25	BDA24	
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16	
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8	
	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0	
BASRA	BASA7	BASA6	BASA5	BASA4	BASA3	BASA2	BASA1	BASA0	
BASRB	BASB7	BASB6	BASB5	BASB4	BASB3	BASB2	BASB1	BASB0	
PACR	PA7MD1	PA7MD0	PA6MD1	PA6MD0	PA5MD1	PA5MD0	PA4MD1	PA4MD0	PFC
	PA3MD1	PA3MD0	PA2MD1	PA2MD0	PA1MD1	PA1MD0	PA0MD1	PA0MD0	
PBCR	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0	
	PB3MD1	PB3MD0	PB2MD1	PB2MD0	PB1MD1	PB1MD0	PB0MD1	PB0MD0	
PCCR	PC7MD1	PC7MD0	PC6MD1	PC6MD0	PC5MD1	PC5MD0	PC4MD1	PC4MD0	
	PC3MD1	PC3MD0	PC2MD1	PC2MD0	PC1MD1	PC1MD0	PC0MD1	PC0MD0	
PDCR	PD7MD1	PD7MD0	PD6MD1	PD6MD0	PD5MD1	PD5MD0	PD4MD1	PD4MD0	
	PD3MD1	PD3MD0	PD2MD1	PD2MD0	PD1MD1	PD1MD0	PD0MD1	PD0MD0	
PECR	—	—	PE6MD1	—	PE5MD1	—	PE4MD1	PE4MD0	
	PE3MD1	PE3MD0	PE2MD1	PE2MD0	PE1MD1	PE1MD0	PE0MD1	PE0MD0	
PFCR	—	—	PF6MD1	PF6MD0	PF5MD1	PF5MD0	PF4MD1	PF4MD0	
	PF3MD1	PF3MD0	PF2MD1	PF2MD0	PF1MD1	PF1MD0	PF0MD1	PF0MD0	
PGCR	—	—	PG6MD1	PG6MD0	PG5MD1	PG5MD0	PG4MD1	PG4MD0	
	PG3MD1	PG3MD0	PG2MD1	PG2MD0	PG1MD1	PG1MD0	PG0MD1	PG0MD0	
PHCR	—	—	PH6MD1	PH6MD0	PH5MD1	PH5MD0	PH4MD1	PH4MD0	
	PH3MD1	PH3MD0	PH2MD1	PH2MD0	PH1MD1	PH1MD0	PH0MD1	PH0MD0	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
PJCR	—	—	PJ6MD1	PJ6MD0	PJ5MD1	PJ5MD0	PJ4MD1	PJ4MD0	PFC
	PJ3MD1	PJ3MD0	PJ2MD1	PJ2MD0	PJ1MD1	PJ1MD0	PJ0MD1	PJ0MD0	
PKCR	—	—	—	—	—	—	—	—	
	PK3MD1	PK3MD0	PK2MD1	PK2MD0	PK1MD1	PK1MD0	PK0MD1	PK0MD0	
PLCR	PL7MD1	PL7MD0	PL6MD1	PL6MD0	PL5MD1	PL5MD0	PL4MD1	PL4MD0	
	PL3MD1	PL3MD0	—	—	—	—	—	—	
PMCR	PM7MD1	PM7MD0	PM6MD1	PM6MD0	PM5MD1	PM5MD0	PM4MD1	PM4MD0	
	PM3MD1	PM3MD0	PM2MD1	PM2MD0	PM1MD1	PM1MD0	PM0MD1	PM0MD0	
PPCR	—	—	—	—	—	—	PP4MD1	PP4MD0	
	PP3MD1	PP3MD0	PP2MD1	PP2MD0	PP1MD1	PP1MD0	PP0MD1	PP0MD0	
PRCR	PR7MD1	PR7MD0	PR6MD1	PR6MD0	PR5MD1	PR5MD0	PR4MD1	PR4MD0	
	PR3MD1	PR3MD0	PR2MD1	PR2MD0	PR1MD1	PR1MD0	PR0MD1	PR0MD0	
PSCR	—	—	—	—	—	—	PS4MD1	PS4MD0	
	PS3MD1	PS3MD0	PS2MD1	PS2MD0	PS1MD1	PS1MD0	PS0MD1	PS0MD0	
PTCR	—	—	—	—	—	—	PT4MD1	PT4MD0	
	PT3MD1	PT3MD0	PT2MD1	PT2MD0	PT1MD1	PT1MD0	PT0MD1	PT0MD0	
PUCR	—	—	—	—	—	—	PU4MD1	PU4MD0	
	PU3MD1	PU3MD0	PU2MD1	PU2MD0	PU1MD1	PU1MD0	PU0MD1	PU0MD0	
PVCR	—	—	—	—	—	—	PV4MD1	PV4MD0	
	PV3MD1	PV3MD0	PV2MD1	PV2MD0	PV1MD1	PV1MD0	PV0MD1	PV0MD0	
PSELA	PSELA15	PSELA14	PSELA13	PSELA12	PSELA11	PSELA10	PSELA9	PSELA8	
	PSELA7	PSELA6	PSELA5	PSELA4	PSELA3	PSELA2	PSELA1	PSELA0	
PSELB	PSELB15	PSELB14	PSELB13	PSELB12	PSELB11	PSELB10	PSELB9	PSELB8	
	—	—	—	—	—	—	—	PSELB0	
PSELC	PSELC15	PSELC14	PSELC13	PSELC12	PSELC11	PSELC10	PSELC9	PSELC8	
	—	—	—	—	—	—	—	—	
PSELD	—	PSELD14	PSELD13	PSELD12	—	PSELD10	PSELD9	PSELD8	
	—	PSELD6	PSELD5	PSELD4	—	PSELD2	PSELD1	PSELD0	
PADR	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	PA1DT	PA0DT	I/O port
PBDR	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB0DT	
PCDR	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT	

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	Module
PDDR	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT	I/O port
PEDR	—	PE6DT	PE5DT	PE4DT	PE3DT	PE2DT	PE1DT	PE0DT	
PFDR	—	PF6DT	PF5DT	PF4DT	PF3DT	PF2DT	PF1DT	PF0DT	
PGDR	—	PG6DT	PG5DT	PG4DT	PG3DT	PG2DT	PG1DT	PG0DT	
PHDR	—	PH6DT	PH5DT	PH4DT	PH3DT	PH2DT	PH1DT	PH0DT	
PJDR	—	PJ6DT	PJ5DT	PJ4DT	PJ3DT	PJ2DT	PJ1DT	PJ0DT	
PKDR	—	—	—	—	PK3DT	PK2DT	PK1DT	PK0DT	
PLDR	PL7DT	PL6DT	PL5DT	PL4DT	PL3DT	—	—	—	
PMDR	PM7DT	PM6DT	PM5DT	PM4DT	PM3DT	PM2DT	PM1DT	PM0DT	
PPDR	—	—	—	PP4DT	PP3DT	PP2DT	PP1DT	PP0DT	
PRDR	PR7DT	PR6DT	PR5DT	PR4DT	PR3DT	PR2DT	PR1DT	PR0DT	
PSDR	—	—	—	PS4DT	PS3DT	PS2DT	PS1DT	PS0DT	
PTDR	—	—	—	PT4DT	PT3DT	PT2DT	PT1DT	PT0DT	
PUDR	—	—	—	PU4DT	PU3DT	PU2DT	PU1DT	PU0DT	
PVDR	—	—	—	PV4DT	PV3DT	PV2DT	PV1DT	PV0DT	
SDIR	T17	T16	T15	T14	T13	T12	T11	T10	H–UDI
	—	—	—	—	—	—	—	—	
SDIDH	DID31	DID30	DID29	DID28	DID27	DID26	DID25	DID24	
	DID23	DID22	DID21	DID20	DID19	DID18	DID17	DID16	
SDIDL	DID15	DID14	DID13	DID12	DID11	DID10	DID9	DID8	
	DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0	

Notes: 1. Specified memory type is normal area or byte selection SRAM.
2. Specified memory type is burst ROM (asynchronous).
3. Specified memory type is burst ROM (synchronous).
4. Specified memory type is SDRAM.
5. Specified memory type is PCMCIA.

37.3 Register States in Each Operating Mode

Register Abbreviation	Power-On Reset*[1]	Manual Reset*[1]	Software Standby	Module Standby	Sleep	Module
MMUCR	Initialized	Initialized	Retained	Retained	Retained	MMU
PTEH	Initialized	Initialized	Retained	Retained	Retained	
PTEL	Initialized	Initialized	Retained	Retained	Retained	
TTB	Initialized	Initialized	Retained	Retained	Retained	
CCR2	Initialized	Initialized	Retained	Retained	Retained	Cache
CCR3	Initialized	Initialized	Retained	Retained	Retained	
CCR1	Initialized	Initialized	Retained	Retained	Retained	
INTEVT2	Initialized	Initialized	Retained	—	Retained	Exception handling
TRA	Initialized	Initialized	Retained	—	Retained	
EXPEVT	Initialized	Initialized	Retained	—	Retained	
INTEVT	Initialized	Initialized	Retained	—	Retained	
TEA	Initialized	Initialized	Retained	—	Retained	
IPRF	Initialized	Initialized	Retained	—	Retained	INTC
IPRG	Initialized	Initialized	Retained	—	Retained	
IPRH	Initialized	Initialized	Retained	—	Retained	
IPRI	Initialized	Initialized	Retained	—	Retained	
IPRJ	Initialized	Initialized	Retained	—	Retained	
IRR5	Initialized	Initialized	Retained	—	Retained	
IRR6	Initialized	Initialized	Retained	—	Retained	
IRR7	Initialized	Initialized	Retained	—	Retained	
IRR8	Initialized	Initialized	Retained	—	Retained	
IRR9	Initialized	Initialized	Retained	—	Retained	
IRR0	Initialized	Initialized	Retained	—	Retained	
IRR1	Initialized	Initialized	Retained	—	Retained	
IRR2	Initialized	Initialized	Retained	—	Retained	
IRR3	Initialized	Initialized	Retained	—	Retained	
IRR4	Initialized	Initialized	Retained	—	Retained	
ICR1	Initialized	Initialized	Retained	—	Retained	

Register Abbreviation	Power-On Reset*1	Manual Reset*1	Software Standby	Module Standby	Sleep	Module
ICR2	Initialized	Initialized	Retained	—	Retained	INTC
PINTER	Initialized	Initialized	Retained	—	Retained	
IPRC	Initialized	Initialized	Retained	—	Retained	
IPRD	Initialized	Initialized	Retained	—	Retained	
IPRE	Initialized	Initialized	Retained	—	Retained	
ICR0	Initialized	Initialized	Retained	—	Retained	
IPRA	Initialized	Initialized	Retained	—	Retained	
IPRB	Initialized	Initialized	Retained	—	Retained	
CMNCR	Initialized	Retained	Retained	—	Retained	BSC
CS0BCR	Initialized	Retained	Retained	—	Retained	
CS2BCR	Initialized	Retained	Retained	—	Retained	
CS3BCR	Initialized	Retained	Retained	—	Retained	
CS4BCR	Initialized	Retained	Retained	—	Retained	
CS5ABCR	Initialized	Retained	Retained	—	Retained	
CS5BBCR	Initialized	Retained	Retained	—	Retained	
CS6ABCR	Initialized	Retained	Retained	—	Retained	
CS6BBCR	Initialized	Retained	Retained	—	Retained	
CS0WCR	Initialized	Retained	Retained	—	Retained	
CS2WCR	Initialized	Retained	Retained	—	Retained	
CS3WCR	Initialized	Retained	Retained	—	Retained	
CS4WCR	Initialized	Retained	Retained	—	Retained	
CS5AWCR	Initialized	Retained	Retained	—	Retained	
CS5BWCR	Initialized	Retained	Retained	—	Retained	
CS6AWCR	Initialized	Retained	Retained	—	Retained	
CS6BWCR	Initialized	Retained	Retained	—	Retained	
SDCR	Initialized	Retained	Retained	—	Retained	
RTCSR	Initialized	Retained	Retained	—	Retained	
RTCNT	Initialized	Retained	Retained	—	Retained	
RTCOR	Initialized	Retained	Retained	—	Retained	
SAR_0	Initialized	Initialized	Retained	Retained	Retained	DMAC
DAR_0	Initialized	Initialized	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset*[1]	Manual Reset*[1]	Software Standby	Module Standby	Sleep	Module
DMATCR_0	Initialized	Initialized	Retained	Retained	Retained	DMAC
CHCR_0	Initialized	Initialized	Retained	Retained	Retained	
SAR_1	Initialized	Initialized	Retained	Retained	Retained	
DAR_1	Initialized	Initialized	Retained	Retained	Retained	
DMATCR_1	Initialized	Initialized	Retained	Retained	Retained	
CHCR_1	Initialized	Initialized	Retained	Retained	Retained	
SAR_2	Initialized	Initialized	Retained	Retained	Retained	
DAR_2	Initialized	Initialized	Retained	Retained	Retained	
DMATCR_2	Initialized	Initialized	Retained	Retained	Retained	
CHCR_2	Initialized	Initialized	Retained	Retained	Retained	
SAR_3	Initialized	Initialized	Retained	Retained	Retained	
DAR_3	Initialized	Initialized	Retained	Retained	Retained	
DMATCR_3	Initialized	Initialized	Retained	Retained	Retained	
CHCR_3	Initialized	Initialized	Retained	Retained	Retained	
SAR_4	Initialized	Initialized	Retained	Retained	Retained	
DAR_4	Initialized	Initialized	Retained	Retained	Retained	
DMATCR_4	Initialized	Initialized	Retained	Retained	Retained	
CHCR_4	Initialized	Initialized	Retained	Retained	Retained	
SAR_5	Initialized	Initialized	Retained	Retained	Retained	
DAR_5	Initialized	Initialized	Retained	Retained	Retained	
DMATCR_5	Initialized	Initialized	Retained	Retained	Retained	
CHCR_5	Initialized	Initialized	Retained	Retained	Retained	
DMAOR	Initialized	Initialized	Retained	Retained	Retained	
DMARS0	Initialized	Initialized	Retained	Retained	Retained	
DMARS1	Initialized	Initialized	Retained	Retained	Retained	
DMARS2	Initialized	Initialized	Retained	Retained	Retained	
UCLKCR	Initialized	Retained	Retained	—	Retained	CPG
FRQCR	Initialized*[6]	Retained	Retained	—	Retained	
WTCNT	Initialized*[6]	Retained	Retained	—	Retained	WDT
WTCSR	Initialized*[6]	Retained	Retained	—	Retained	

Register Abbreviation	Power-On Reset*1	Manual Reset*1	Software Standby	Module Standby	Sleep	Module
STBCR3	Initialized	Retained	Retained	—	Retained	Power-down modes
STBCR4	Initialized	Retained	Retained	—	Retained	
STBCR5	Initialized	Retained	Retained	—	Retained	
STBCR	Initialized	Retained	Retained	—	Retained	
STBCR2	Initialized	Retained	Retained	—	Retained	
TSTR	Initialized	Initialized	Initialized*2	Initialized	Retained	TMU
TCOR_0	Initialized	Initialized	Retained	Retained	Retained	
TCNT_0	Initialized	Initialized	Retained	Retained	Retained	
TCR_0	Initialized	Initialized	Retained	Retained	Retained	
TCOR_1	Initialized	Initialized	Retained	Retained	Retained	
TCNT_1	Initialized	Initialized	Retained	Retained	Retained	
TCR_1	Initialized	Initialized	Retained	Retained	Retained	
TCOR_2	Initialized	Initialized	Retained	Retained	Retained	
TCNT_2	Initialized	Initialized	Retained	Retained	Retained	
TCR_2	Initialized	Initialized	Retained	Retained	Retained	
TSTR	Initialized	Initialized	Retained	Retained	Retained	TPU
TCR_0	Initialized	Initialized	Retained	Retained	Retained	
TMDR_0	Initialized	Initialized	Retained	Retained	Retained	
TIOR_0	Initialized	Initialized	Retained	Retained	Retained	
TIER_0	Initialized	Initialized	Retained	Retained	Retained	
TSR_0	Initialized	Initialized	Retained	Retained	Retained	
TCNT_0	Initialized	Initialized	Retained	Retained	Retained	
TGRA_0	Initialized	Initialized	Retained	Retained	Retained	
TGRB_0	Initialized	Initialized	Retained	Retained	Retained	
TGRC_0	Initialized	Initialized	Retained	Retained	Retained	
TGRD_0	Initialized	Initialized	Retained	Retained	Retained	
TCR_1	Initialized	Initialized	Retained	Retained	Retained	
TMDR_1	Initialized	Initialized	Retained	Retained	Retained	
TIOR_1	Initialized	Initialized	Retained	Retained	Retained	
TIER_1	Initialized	Initialized	Retained	Retained	Retained	
TSR_1	Initialized	Initialized	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset*[1]	Manual Reset*[1]	Software Standby	Module Standby	Sleep	Module
TCNT_1	Initialized	Initialized	Retained	Retained	Retained	TPU
TGRA_1	Initialized	Initialized	Retained	Retained	Retained	
TGRB_1	Initialized	Initialized	Retained	Retained	Retained	
TGRC_1	Initialized	Initialized	Retained	Retained	Retained	
TGRD_1	Initialized	Initialized	Retained	Retained	Retained	
TCR_2	Initialized	Initialized	Retained	Retained	Retained	
TMDR_2	Initialized	Initialized	Retained	Retained	Retained	
TIOR_2	Initialized	Initialized	Retained	Retained	Retained	
TIER_2	Initialized	Initialized	Retained	Retained	Retained	
TSR_2	Initialized	Initialized	Retained	Retained	Retained	
TCNT_2	Initialized	Initialized	Retained	Retained	Retained	
TGRA_2	Initialized	Initialized	Retained	Retained	Retained	
TGRB_2	Initialized	Initialized	Retained	Retained	Retained	
TGRC_2	Initialized	Initialized	Retained	Retained	Retained	
TGRD_2	Initialized	Initialized	Retained	Retained	Retained	
TCR_3	Initialized	Initialized	Retained	Retained	Retained	
TMDR_3	Initialized	Initialized	Retained	Retained	Retained	
TIOR_3	Initialized	Initialized	Retained	Retained	Retained	
TIER_3	Initialized	Initialized	Retained	Retained	Retained	
TSR_3	Initialized	Initialized	Retained	Retained	Retained	
TCNT_3	Initialized	Initialized	Retained	Retained	Retained	
TGRA_3	Initialized	Initialized	Retained	Retained	Retained	
TGRB_3	Initialized	Initialized	Retained	Retained	Retained	
TGRC_3	Initialized	Initialized	Retained	Retained	Retained	
TGRD_3	Initialized	Initialized	Retained	Retained	Retained	
CMSTR	Initialized	Initialized	Retained	Retained	Retained	CMT
CMCSR_0	Initialized	Initialized	Retained	Retained	Retained	
CMCNT_0	Initialized	Initialized	Retained	Retained	Retained	
CMCOR_0	Initialized	Initialized	Retained	Retained	Retained	
CMCSR_1	Initialized	Initialized	Retained	Retained	Retained	
CMCNT_1	Initialized	Initialized	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset*1	Manual Reset*1	Software Standby	Module Standby	Sleep	Module
CMCOR_1	Initialized	Initialized	Retained	Retained	Retained	CMT
CMCSR_2	Initialized	Initialized	Retained	Retained	Retained	
CMCNT_2	Initialized	Initialized	Retained	Retained	Retained	
CMCOR_2	Initialized	Initialized	Retained	Retained	Retained	
CMCSR_3	Initialized	Initialized	Retained	Retained	Retained	
CMCNT_3	Initialized	Initialized	Retained	Retained	Retained	
CMCOR_3	Initialized	Initialized	Retained	Retained	Retained	
CMCSR_4	Initialized	Initialized	Retained	Retained	Retained	
CMCNT_4	Initialized	Initialized	Retained	Retained	Retained	
CMCOR_4	Initialized	Initialized	Retained	Retained	Retained	
R64CNT	Retained	Retained	Retained	Retained	Retained	RTC
RSECCNT	Retained	Retained	Retained	Retained	Retained	
RMINCNT	Retained	Retained	Retained	Retained	Retained	
RHRCNT	Retained	Retained	Retained	Retained	Retained	
RWKCNT	Retained	Retained	Retained	Retained	Retained	
RDAYCNT	Retained	Retained	Retained	Retained	Retained	
RMONCNT	Retained	Retained	Retained	Retained	Retained	
RYRCNT	Retained	Retained	Retained	Retained	Retained	
RSECAR	Retained*3	Retained	Retained	Retained	Retained	
RMINAR	Retained*3	Retained	Retained	Retained	Retained	
RHRAR	Retained*3	Retained	Retained	Retained	Retained	
RWKAR	Retained*3	Retained	Retained	Retained	Retained	
RDAYAR	Retained*3	Retained	Retained	Retained	Retained	
RMONAR	Retained*3	Retained	Retained	Retained	Retained	
RCR1	Initialized	Initialized	Retained	Retained	Retained	
RCR2	Initialized	Initialized*4	Retained	Retained	Retained	
RYRAR	Retained	Retained	Retained	Retained	Retained	
RCR3	Initialized	Retained	Retained	Retained	Retained	
SCSMR_0	Initialized	Initialized	Retained	Retained	Retained	SCIF
SCBRR_0	Initialized	Initialized	Retained	Retained	Retained	
SCSCR_0	Initialized	Initialized	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset*[1]	Manual Reset*[1]	Software Standby	Module Standby	Sleep	Module
SCTDSR_0	Initialized	Initialized	Retained	Retained	Retained	SCIF
SCFER_0	Initialized	Initialized	Retained	Retained	Retained	
SCSSR_0	Initialized	Initialized	Retained	Retained	Retained	
SCFCR_0	Initialized	Initialized	Retained	Retained	Retained	
SCFDR_0	Initialized	Initialized	Retained	Retained	Retained	
SCFTDR_0	Initialized	Initialized	Retained	Retained	Retained	
SCFRDR_0	Initialized	Initialized	Retained	Retained	Retained	
SCSMR_1	Initialized	Initialized	Retained	Retained	Retained	
SCBRR_1	Initialized	Initialized	Retained	Retained	Retained	
SCSCR_1	Initialized	Initialized	Retained	Retained	Retained	
SCTDSR_1	Initialized	Initialized	Retained	Retained	Retained	
SCFER_1	Initialized	Initialized	Retained	Retained	Retained	
SCSSR_1	Initialized	Initialized	Retained	Retained	Retained	
SCFCR_1	Initialized	Initialized	Retained	Retained	Retained	
SCFDR_1	Initialized	Initialized	Retained	Retained	Retained	
SCFTDR_1	Initialized	Initialized	Retained	Retained	Retained	
SCFRDR_1	Initialized	Initialized	Retained	Retained	Retained	
SCIMR	Initialized	Initialized	Retained	Retained	Retained	IrDA
ICCR1	Initialized	Initialized	Retained	Retained	Retained	IIC
ICCR2	Initialized	Initialized	Retained	Retained	Retained	
ICMR	Initialized	Initialized	Retained	Retained	Retained	
ICIER	Initialized	Initialized	Retained	Retained	Retained	
ICSR	Initialized	Initialized	Retained	Retained	Retained	
SAR	Initialized	Initialized	Retained	Retained	Retained	
ICDRT	Initialized	Initialized	Retained	Retained	Retained	
ICDRR	Initialized	Initialized	Retained	Retained	Retained	
ICCKS	Initialized	Initialized	Retained	Retained	Retained	
SIMDR_0	Initialized	Initialized	Retained	Retained	Retained	SIOF0
SISCR_0	Initialized	Initialized	Retained	Retained	Retained	
SITDAR_0	Initialized	Initialized	Retained	Retained	Retained	
SIRDAR_0	Initialized	Initialized	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset*1	Manual Reset*1	Software Standby	Module Standby	Sleep	Module
SICDAR_0	Initialized	Initialized	Retained	Retained	Retained	SIOF0
SICTR_0	Initialized	Initialized	Retained	Retained	Retained	
SIFCTR_0	Initialized	Initialized	Retained	Retained	Retained	
SISTR_0	Initialized	Initialized	Retained	Retained	Retained	
SIIER_0	Initialized	Initialized	Retained	Retained	Retained	
SITDR_0	Initialized	Initialized	Retained	Retained	Retained	
SIRDR_0	Initialized	Initialized	Retained	Retained	Retained	
SITCR_0	Initialized	Initialized	Retained	Retained	Retained	
SIRCR_0	Initialized	Initialized	Retained	Retained	Retained	
SIMDR_1	Initialized	Initialized	Retained	Retained	Retained	SIOF1
SISCR_1	Initialized	Initialized	Retained	Retained	Retained	
SITDAR_1	Initialized	Initialized	Retained	Retained	Retained	
SIRDAR_1	Initialized	Initialized	Retained	Retained	Retained	
SICDAR_1	Initialized	Initialized	Retained	Retained	Retained	SIOF1
SICTR_1	Initialized	Initialized	Retained	Retained	Retained	
SIFCTR_1	Initialized	Initialized	Retained	Retained	Retained	
SISTR_1	Initialized	Initialized	Retained	Retained	Retained	
SIIER_1	Initialized	Initialized	Retained	Retained	Retained	
SITDR_1	Initialized	Initialized	Retained	Retained	Retained	
SIRDR_1	Initialized	Initialized	Retained	Retained	Retained	
SITCR_1	Initialized	Initialized	Retained	Retained	Retained	
SIRCR_1	Initialized	Initialized	Retained	Retained	Retained	
ACTR1	Initialized	Initialized	Retained	Retained	Retained	AFEIF
ACTR2	Initialized	Initialized	Retained	Retained	Retained	
ASTR1	Initialized	Initialized	Retained	Retained	Retained	
ASTR2	Initialized	Initialized	Retained	Retained	Retained	
MRCR	Initialized	Initialized	Retained	Retained	Retained	
MPCR	Initialized	Initialized	Retained	Retained	Retained	
DPNQ	Initialized	Initialized	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset*[1]	Manual Reset*[1]	Software Standby	Module Standby	Sleep	Module
RCNT	Initialized	Initialized	Retained	Retained	Retained	AFEIF
ACDR	Initialized	Initialized	Retained	Retained	Retained	
ASDR	Initialized	Initialized	Retained	Retained	Retained	
TDFP	Initialized	Initialized	Retained	Retained	Retained	
RDFP	Initialized	Initialized	Retained	Retained	Retained	
UTRCTL	Initialized	Retained	Retained	—	Retained	USB PMC
USBHR	Initialized	Initialized	Retained	Retained	Retained	USBH
USBHC	Initialized	Initialized	Retained	Retained	Retained	
USBHCS	Initialized	Initialized	Retained	Retained	Retained	
USBHIS	Initialized	Initialized	Retained	Retained	Retained	
USBHIE	Initialized	Initialized	Retained	Retained	Retained	
USBHID	Initialized	Initialized	Retained	Retained	Retained	
USBHHCCA	Initialized	Initialized	Retained	Retained	Retained	
USBHPCED	Initialized	Initialized	Retained	Retained	Retained	
USBHCHED	Initialized	Initialized	Retained	Retained	Retained	
USBHCCED	Initialized	Initialized	Retained	Retained	Retained	
USBHBHED	Initialized	Initialized	Retained	Retained	Retained	
USBHBCED	Initialized	Initialized	Retained	Retained	Retained	
USBHDHED	Initialized	Initialized	Retained	Retained	Retained	
USBHFI	Initialized	Initialized	Retained	Retained	Retained	
USBHFR	Initialized	Initialized	Retained	Retained	Retained	
USBHFN	Initialized	Initialized	Retained	Retained	Retained	
USBHPS	Initialized	Initialized	Retained	Retained	Retained	
USBHLST	Initialized	Initialized	Retained	Retained	Retained	
USBHRDA	Initialized	Initialized	Retained	Retained	Retained	
USBHRDB	Initialized	Initialized	Retained	Retained	Retained	
USBHRS	Initialized	Initialized	Retained	Retained	Retained	
USBHRPS1	Initialized	Initialized	Retained	Retained	Retained	
USBHRPS2	Initialized	Initialized	Retained	Retained	Retained	
IFR0	Initialized	Initialized	Retained	Retained	Retained	USBF
IFR1	Initialized	Initialized	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset*1	Manual Reset*1	Software Standby	Module Standby	Sleep	Module
IFR2	Initialized	Initialized	Retained	Retained	Retained	USBF
IFR3	Initialized	Initialized	Retained	Retained	Retained	
IER0	Initialized	Initialized	Retained	Retained	Retained	
IER1	Initialized	Initialized	Retained	Retained	Retained	
IER2	Initialized	Initialized	Retained	Retained	Retained	
IER3	Initialized	Initialized	Retained	Retained	Retained	
ISR0	Initialized	Initialized	Retained	Retained	Retained	
ISR1	Initialized	Initialized	Retained	Retained	Retained	
ISR2	Initialized	Initialized	Retained	Retained	Retained	
ISR3	Initialized	Initialized	Retained	Retained	Retained	
EPDR0i	Initialized	Initialized	Retained	Retained	Retained	
EPDR0o	Initialized	Initialized	Retained	Retained	Retained	
EPDR0s	Initialized	Initialized	Retained	Retained	Retained	
EPDR1	Initialized	Initialized	Retained	Retained	Retained	
EPDR2	Initialized	Initialized	Retained	Retained	Retained	
EPDR3	Initialized	Initialized	Retained	Retained	Retained	
EPDR4	Initialized	Initialized	Retained	Retained	Retained	
EPDR5	Initialized	Initialized	Retained	Retained	Retained	
EPSZ0o	Initialized	Initialized	Retained	Retained	Retained	
EPSZ1	Initialized	Initialized	Retained	Retained	Retained	
EPSZ4	Initialized	Initialized	Retained	Retained	Retained	
DASTS	Initialized	Initialized	Retained	Retained	Retained	
FCLR0	Initialized	Initialized	Retained	Retained	Retained	
FCLR1	Initialized	Initialized	Retained	Retained	Retained	
EPSTL0	Initialized	Initialized	Retained	Retained	Retained	
EPSTL1	Initialized	Initialized	Retained	Retained	Retained	
TRG	Initialized	Initialized	Retained	Retained	Retained	
DMA	Initialized	Initialized	Retained	Retained	Retained	
CVR	Initialized	Initialized	Retained	Retained	Retained	
CTLR0	Initialized	Initialized	Retained	Retained	Retained	
TSRH	Initialized	Initialized	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset*[1]	Manual Reset*[1]	Software Standby	Module Standby	Sleep	Module
TSRL	Initialized	Initialized	Retained	Retained	Retained	USBF
EPIR	Initialized	Initialized	Retained	Retained	Retained	
IFR4	Initialized	Initialized	Retained	Retained	Retained	
IER4	Initialized	Initialized	Retained	Retained	Retained	
ISR4	Initialized	Initialized	Retained	Retained	Retained	
CTLR1	Initialized	Initialized	Retained	Retained	Retained	
TMRH	Initialized	Initialized	Retained	Retained	Retained	
TMRL	Initialized	Initialized	Retained	Retained	Retained	
STOH	Initialized	Initialized	Retained	Retained	Retained	
STOL	Initialized	Initialized	Retained	Retained	Retained	
LDPRnn (nn:00 to FF)	Initialized	Initialized	Retained	Retained	Retained	LCDC
LDICKR	Initialized	Initialized	Retained	Retained	Retained	
LDMTR	Initialized	Initialized	Retained	Retained	Retained	
LDDFR	Initialized	Initialized	Retained	Retained	Retained	
LDSMR	Initialized	Initialized	Retained	Retained	Retained	
LDSARU	Initialized	Initialized	Retained	Retained	Retained	
LDSARL	Initialized	Initialized	Retained	Retained	Retained	
LDLAOR	Initialized	Initialized	Retained	Retained	Retained	
LDPALCR	Initialized	Initialized	Retained	Retained	Retained	
LDHCNR	Initialized	Initialized	Retained	Retained	Retained	
LDHSYNR	Initialized	Initialized	Retained	Retained	Retained	
LDVDLNR	Initialized	Initialized	Retained	Retained	Retained	
LDVTLNR	Initialized	Initialized	Retained	Retained	Retained	
LDVSYNR	Initialized	Initialized	Retained	Retained	Retained	
LDACLNR	Initialized	Initialized	Retained	Retained	Retained	
LDINTR	Initialized	Initialized	Retained	Retained	Retained	
LDPMMR	Initialized	Initialized	Retained	Retained	Retained	
LDPSPR	Initialized	Initialized	Retained	Retained	Retained	
LDCNTR	Initialized	Initialized	Retained	Retained	Retained	
LDUINTR	Initialized	Initialized	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset*[1]	Manual Reset*[1]	Software Standby	Module Standby	Sleep	Module
LDUINTLNR	Initialized	Initialized	Retained	Retained	Retained	LCDC
LDLIRNR	Initialized	Initialized	Retained	Retained	Retained	
ADDRA	Initialized	Initialized	Initialized	Initialized	Retained	ADC
ADDRB	Initialized	Initialized	Initialized	Initialized	Retained	
ADDRC	Initialized	Initialized	Initialized	Initialized	Retained	
ADDRD	Initialized	Initialized	Initialized	Initialized	Retained	
ADCSR	Initialized	Initialized	Initialized	Initialized	Retained	
DADR0	Initialized	Initialized	Retained	Retained	Retained	DAC
DADR1	Initialized	Initialized	Retained	Retained	Retained	
DACR	Initialized	Initialized	Retained	Retained	Retained	
PCC0ISR	*[5]	*[5]	*[5]	*[5]	*[5]	PCC
PCC0GCR	Initialized	Retained	Retained	Retained	Retained	
PCC0CSCR	Initialized	Retained	Retained	Retained	Retained	
PCC0CSCIER	Initialized	Retained	Retained	Retained	Retained	
SCSMR	Initialized	Initialized	Retained	Retained	Retained	SIM
SCBRR	Initialized	Initialized	Retained	Retained	Retained	
SCSCR	Initialized	Initialized	Retained	Retained	Retained	
SCTDR	Initialized	Initialized	Retained	Retained	Retained	
SCSSR	Initialized	Initialized	Retained	Retained	Retained	
SCRDR	Initialized	Initialized	Retained	Retained	Retained	
SCSCMR	Initialized	Initialized	Retained	Retained	Retained	
SCSC2R	Initialized	Initialized	Retained	Retained	Retained	
SCGRD	Initialized	Initialized	Retained	Retained	Retained	
SCWAIT	Initialized	Initialized	Retained	Retained	Retained	
SCSMPL	Initialized	Initialized	Retained	Retained	Retained	
CMDR0	Initialized	Initialized	Retained	Retained	Retained	MMC
CMDR1	Initialized	Initialized	Retained	Retained	Retained	
CMDR2	Initialized	Initialized	Retained	Retained	Retained	
CMDR3	Initialized	Initialized	Retained	Retained	Retained	
CMDR4	Initialized	Initialized	Retained	Retained	Retained	
CMDR5	Initialized	Initialized	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset*[1]	Manual Reset*[1]	Software Standby	Module Standby	Sleep	Module
CMDSTRT	Initialized	Initialized	Retained	Retained	Retained	MMC
OPCR	Initialized	Initialized	Retained	Retained	Retained	
CSTR	Initialized	Initialized	Retained	Retained	Retained	
INTCR0	Initialized	Initialized	Retained	Retained	Retained	
INTCR1	Initialized	Initialized	Retained	Retained	Retained	
INTSTR0	Initialized	Initialized	Retained	Retained	Retained	
INTSTR1	Initialized	Initialized	Retained	Retained	Retained	
CLKON	Initialized	Initialized	Retained	Retained	Retained	
CTOCR	Initialized	Initialized	Retained	Retained	Retained	
VDCNT	Initialized	Initialized	Retained	Retained	Retained	
TBCR	Initialized	Initialized	Retained	Retained	Retained	
MODER	Initialized	Initialized	Retained	Retained	Retained	
CMDTYR	Initialized	Initialized	Retained	Retained	Retained	
RSPTYR	Initialized	Initialized	Retained	Retained	Retained	
TBNCR	Initialized	Initialized	Retained	Retained	Retained	
RSPR0	Initialized	Initialized	Retained	Retained	Retained	
RSPR1	Initialized	Initialized	Retained	Retained	Retained	
RSPR2	Initialized	Initialized	Retained	Retained	Retained	
RSPR3	Initialized	Initialized	Retained	Retained	Retained	
RSPR4	Initialized	Initialized	Retained	Retained	Retained	
RSPR5	Initialized	Initialized	Retained	Retained	Retained	
RSPR6	Initialized	Initialized	Retained	Retained	Retained	
RSPR7	Initialized	Initialized	Retained	Retained	Retained	
RSPR8	Initialized	Initialized	Retained	Retained	Retained	
RSPR9	Initialized	Initialized	Retained	Retained	Retained	
RSPR10	Initialized	Initialized	Retained	Retained	Retained	
RSPR11	Initialized	Initialized	Retained	Retained	Retained	
RSPR12	Initialized	Initialized	Retained	Retained	Retained	
RSPR13	Initialized	Initialized	Retained	Retained	Retained	
RSPR14	Initialized	Initialized	Retained	Retained	Retained	
RSPR15	Initialized	Initialized	Retained	Retained	Retained	

Register Abbreviation	Power-On Reset*1	Manual Reset*1	Software Standby	Module Standby	Sleep	Module
RSPR16	Initialized	Initialized	Retained	Retained	Retained	MMC
RSPRD	Initialized	Initialized	Retained	Retained	Retained	
DTOUTR	Initialized	Initialized	Retained	Retained	Retained	
DR	Initialized	Initialized	Retained	Retained	Retained	
FIFOCLR	Initialized	Initialized	Retained	Retained	Retained	
DMACR	Initialized	Initialized	Retained	Retained	Retained	
INTCR2	Initialized	Initialized	Retained	Retained	Retained	
INTSTR2	Initialized	Initialized	Retained	Retained	Retained	
RDTIMSEL	Initialized	Initialized	Retained	Retained	Retained	
BDRB	Initialized	Retained	Retained	Retained	Retained	UBC
BDMRB	Initialized	Retained	Retained	Retained	Retained	
BRCR	Initialized	Retained	Retained	Retained	Retained	
BETR	Initialized	Retained	Retained	Retained	Retained	
BARB	Initialized	Retained	Retained	Retained	Retained	
BAMRB	Initialized	Retained	Retained	Retained	Retained	
BBRB	Initialized	Retained	Retained	Retained	Retained	
BRSR	Initialized	Retained	Retained	Retained	Retained	
BARA	Initialized	Retained	Retained	Retained	Retained	
BAMRA	Initialized	Retained	Retained	Retained	Retained	
BBRA	Initialized	Retained	Retained	Retained	Retained	
BRDR	Initialized	Retained	Retained	Retained	Retained	
BASRA	Initialized	Retained	Retained	Retained	Retained	
BASRB	Initialized	Retained	Retained	Retained	Retained	
PACR	Initialized	Retained	Retained	—	Retained	PFC
PBCR	Initialized	Retained	Retained	—	Retained	
PCCR	Initialized	Retained	Retained	—	Retained	
PDCR	Initialized	Retained	Retained	—	Retained	
PECR	Initialized	Retained	Retained	—	Retained	
PFCR	Initialized	Retained	Retained	—	Retained	
PGCR	Initialized	Retained	Retained	—	Retained	
PHCR	Initialized	Retained	Retained	—	Retained	

Register Abbreviation	Power-On Reset*[1]	Manual Reset*[1]	Software Standby	Module Standby	Sleep	Module
PJCR	Initialized	Retained	Retained	—	Retained	PFC
PKCR	Initialized	Retained	Retained	—	Retained	
PLCR	Initialized	Retained	Retained	—	Retained	
PMCR	Initialized	Retained	Retained	—	Retained	
PPCR	Initialized	Retained	Retained	—	Retained	
PRCR	Initialized	Retained	Retained	—	Retained	
PSCR	Initialized	Retained	Retained	—	Retained	
PTCR	Initialized	Retained	Retained	—	Retained	
PUCR	Initialized	Retained	Retained	—	Retained	
PVCR	Initialized	Retained	Retained	—	Retained	
PSELA	Initialized	Retained	Retained	—	Retained	
PSELB	Initialized	Retained	Retained	—	Retained	
PSELC	Initialized	Retained	Retained	—	Retained	
PSELD	Initialized	Retained	Retained	—	Retained	
PADR	Initialized	Retained	Retained	—	Retained	I/O port
PBDR	Initialized	Retained	Retained	—	Retained	
PCDR	Initialized	Retained	Retained	—	Retained	
PDDR	Initialized	Retained	Retained	—	Retained	
PEDR	Initialized	Retained	Retained	—	Retained	
PFDR	Initialized	Retained	Retained	—	Retained	
PGDR	Initialized	Retained	Retained	—	Retained	
PHDR	Initialized	Retained	Retained	—	Retained	
PJDR	Initialized	Retained	Retained	—	Retained	
PKDR	Initialized	Retained	Retained	—	Retained	
PLDR	Initialized	Retained	Retained	—	Retained	
PMDR	Initialized	Retained	Retained	—	Retained	
PPDR	Initialized	Retained	Retained	—	Retained	
PRDR	Initialized	Retained	Retained	—	Retained	
PSDR	Initialized	Retained	Retained	—	Retained	
PTDR	Initialized	Retained	Retained	—	Retained	

Register Abbreviation	Power-On Reset*1	Manual Reset*1	Software Standby	Module Standby	Sleep	Module
PUDR	Initialized	Retained	Retained	—	Retained	I/O port
PVDR	Initialized	Retained	Retained	—	Retained	
SDIR	Retained	Retained	Retained	Retained	Retained	H-UDI
SDIDH	Retained	Retained	Retained	Retained	Retained	
SDIDL	Retained	Retained	Retained	Retained	Retained	

Notes: 1. For the initial value, see the corresponding section on each module. Since the values of registers of which initial values are undefined are not retained, described as initialized.
2. Initialized when the multiplication ratio of PLL1 is changed.
3. Some bits are initialized by a power-on reset. For details, see section 17, Realtime Clock (RTC).
4. Some bits are initialized by a manual reset. For details, see section 17, Realtime Clock (RTC).
5. Changes according to the status of the PC card.
6. Not initialized by a power-on reset due to the WDT.

Section 38 Electrical Characteristics

38.1 Absolute Maximum Ratings

Table 38.1 shows the absolute maximum ratings.

Table 38.1 Absolute Maximum Ratings

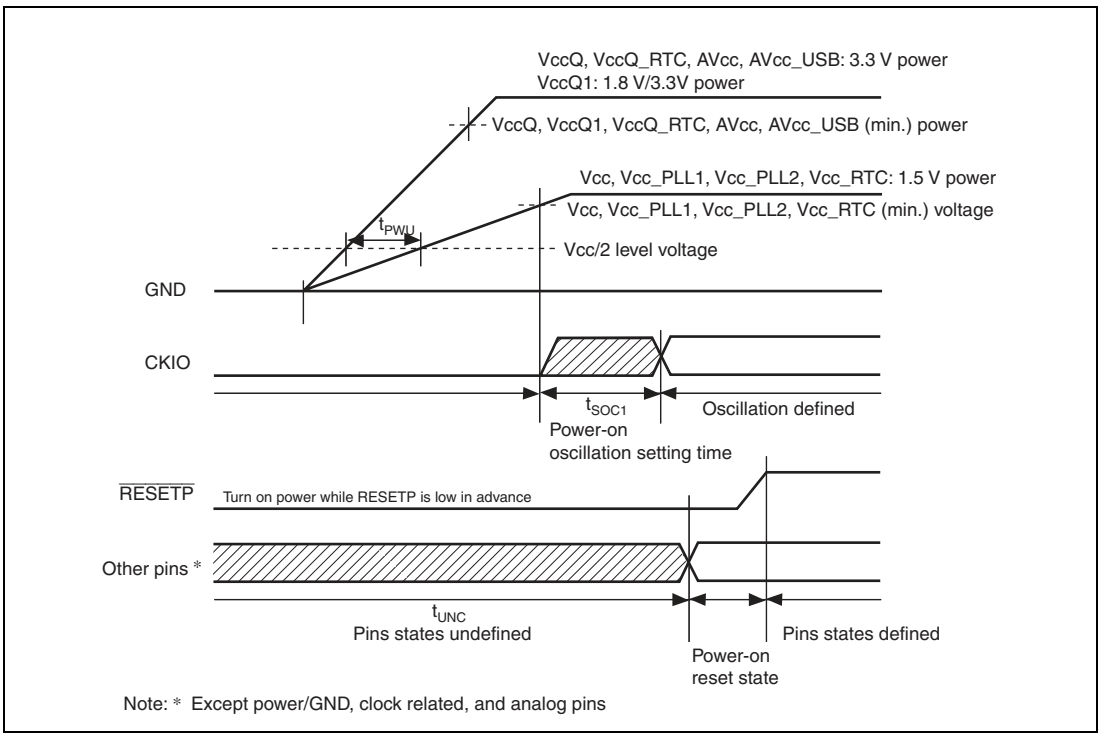
Item	Symbol	Value	Unit
Power supply voltage (I/O)	$V_{CC}Q$, $V_{CC}Q1$, $V_{CC}Q\_RTC$	–0.3 to 4.6	V
Power supply voltage (internal)	V_{CC}, V_{CC}_PLL1, V_{CC}_PLL2, V_{CC}_RTC	–0.3 to 2.1	V
Input voltage	V_{in}	–0.3 to $V_{CC}Q + 0.3$ –0.3 to $V_{CC}Q1 + 0.3$ –0.3 to $V_{CC}Q\_RTC + 0.3$	V
Analog power supply voltage	AV_{CC}	–0.3 to 4.6	V
USB power supply voltage	AV_{CC}_USB	–0.3 to 4.6	V
Analog input voltage	V_{AN}	–0.3 to $AV_{CC} + 0.3$	V
Operating temperature	T_{opr}	–20 to 75	°C
Storage temperature	T_{stg}	–55 to 125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

38.2 Power-On and Power-Off Order

- Order of turning on 1.5 V power (V_{CC}, V_{CC}_PLL1, V_{CC}_PLL2, and V_{CC}_RTC), 1.8 V/3.3 V power (V_{CC}Q1), and 3.3V power (V_{CC}Q, V_{CC}_RTC, AV_{CC}, AV_{CC}_USB
  - First turn on the 3.3 V power and 3.3 V/1.8 V power, then turn on the 1.5 V power within 1 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.
  - Until voltage is applied to all power supplies and a low level is input to the $\overline{\text{RESETP}}$ pin, 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: Handling of CA pin

The CA pin must be ensured to go high before power is turned on. When it is not ensured, through current flows to the I/O buffer, etc. and it may cause the LSI damage even if a clock is not input.

Table 38.2 Recommended Timing in Power-On

Item	Symbol	Maximum Value	Unit
Time difference between the power-on of (V_{CC}Q, V_{CC}Q1, V_{CC}Q_RTC, AV_{CC}, AV_{CC}_USB) and (V_{CC}, V_{CC}_PLL1, V_{CC}_PLL2, V_{CC}_RTC) levels	t_{PWU}	1	ms
Time over which the internal state is undefined	t_{UNC}	100	ms

Note: * The table shown above is recommended values, so they represent guidelines rather than strict requirements.

The time over which the internal state is undefined means the time taken to reach Vcc (min.).

The pin states become defined when V_{CC}Q, V_{CC}Q1, V_{CC}Q_RTC, AV_{CC}, and AV_{CC}_USB (min.) are reached. The period of power-on reset ($\overline{\text{RESETP}}$) is, however, normally accepted as meaning the time taken for oscillation to become stable (when using the on-chip oscillator) after Vcc (min.) is reached.

Ensure that the period over which the internal state is undefined is less than or equal to 100 ms.

- Power-off order
  - In the reverse order of powering-on, first turn off the 1.5 V power, then turn off the 3.3 V/1.8 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.
  - 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.

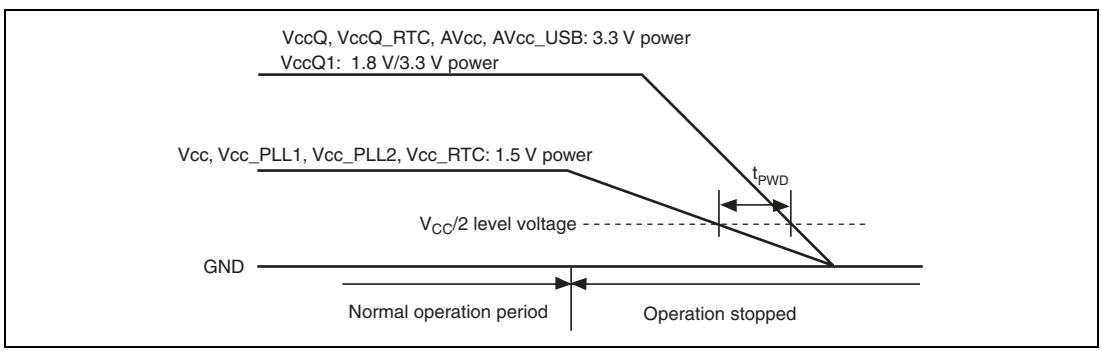


Table 38.3 Recommended Timing in Power-Off

Item	Symbol	Maximum Value	Unit
Time difference between the power-off of ($V_{CC}Q$, $V_{CC}Q1$, $V_{CC}Q\_RTC$, AV_{CC}, AV_{CC}_USB) and (V_{CC}, V_{CC}_PLL1, V_{CC}_PLL2, V_{CC}_RTC) levels	t_{PWD}	10	ms

Note: * The table shown above is recommended values, so they represent guidelines rather than strict requirements.

- Power on and off in hardware standby mode
  - — Hardware standby mode can be used while an RTC clock is in operation.
  - — Apply a low level on the CA pin. Confirm that the level on the STATUS0 pin and STATUS1 pin have become high and low, respectively. V_{CC}, V_{CC}_PLL1, V_{CC}_PLL2, $V_{CC}Q$, $V_{CC}Q1$, AV_{CC}, AV_{CC}_USB can then be turned off. Power supplies $V_{CC}Q\_RTC$ and V_{CC}_RTC must remain on. The CA pin must be ensured to go low.
  - — $V_{CC}Q$, $V_{CC}Q1$, AV_{CC}, AV_{CC}_USB, V_{CC}, V_{CC}_PLL1, and V_{CC}_PLL2 must be turned on while $\overline{\text{RESETP}}$ is low. After the power supply is stable, apply a high level to the CA pin. $\overline{\text{RESETP}}$ must then be canceled to high.

38.3 DC Characteristics

Tables 38.4 and 38.5 show the DC characteristics.

Table 38.4 DC Characteristics (1) [Common]

Conditions: T_a = –20°C to +75°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage		$V_{cc}Q$ $V_{cc}Q\_RTC$	2.7	3.3	3.6	V	
		$V_{cc}Q1$	2.7/1.65	3.3/1.8	3.6/1.95	V	
		V_{cc} V_{cc}_PLL1*1 V_{cc}_PLL2*1 V_{cc}_RTC*1	1.4	1.5	1.6	V	
Analog (A/D, D/A) power supply voltage		AV_{cc}*2	3.0	3.3	3.6	V	When not in use, connect to $V_{cc}Q$.
Analog USB power supply voltage		AV_{cc}_USB	3.0	3.3	3.6	V	When not in use, connect to $V_{cc}Q$.
Analog (A/D, D/A) power supply current	During A/D conversion	AI_{cc}	—	0.8	2	mA	
	During A/D and D/A conversion		—	2.4	6	mA	
	Idle		—	—	20	μA	T_a = 25°C
Analog USB power supply current		AI_{cc}_USB*2	—	4	8	mA	AV_{cc}_USB = 3.3 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Current consumption*[3]	Normal operation	I_{CC}	—	230	300	mA	V_{CC} = 1.5V Iϕ = 133 MHz
		$I_{CC}Q$	—	60	80	mA	$V_{CC}Q$, $V_{CC}Q1$ = 3.3 V Bϕ = 66 MHz
	Sleep mode	I_{CC}	—	35	50	mA	When sleep mode is entered after a power-on reset: $V_{CC}Q$, $V_{CC}Q1$ = 3.3 V Bϕ = 33 MHz
		$I_{CC}Q$	—	15	20		
	Standby mode	I_{CC}	—	80	250	µA	T_a = 25°C, RTC off $V_{CC}Q$, $V_{CC}Q1$ = 3.3 V V_{CC} = 1.5 V
		$I_{CC}Q$	—	10	20		
	Hardware standby mode (state when only V_{CC}_RTC and $V_{CC}Q$_RTC are on)	I_{ustby}	—	—	50	µA	T_a = 25°C $V_{CC}Q$_RTC = 3.3 V V_{CC}_RTC = 1.5 V RTC clock = 32 kHz
Input leakage current	All input pins	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to $V_{CC}Q$ – 0.5 V V_{in} = 0.5 to $V_{CC}Q1$ – 0.5 V
Three-state leakage current	Input/output pins, all output pins (off state)	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to $V_{CC}Q$ – 0.5 V V_{in} = 0.5 to $V_{CC}Q1$ – 0.5 V
Pull-up resistance	I/O port pins	P_{pull}	20	50	120	kΩ	
Pin capacitance	All pins	C	—	—	10	pF	

Notes: 1. When the PLL and RTC are not used, the V_{CC}_PLL1, V_{CC}_PLL2, V_{CC}_RTC, $V_{CC}Q$_RTC, V_{SS}_PLL1, V_{SS}_PLL2, and V_{SS}_RTC should be power supplied.

2. AV_{CC} and AV_{CC}_USB should satisfy the condition $V_{CC}Q - 0.3\ V \leq AV_{CC}$ and AV_{CC}_USB $\leq V_{CC}Q + 0.3$ V.
Even when the A/D converter, D/A converter, and USB are not used, AV_{CC}, AV_{SS}, AV_{CC}_USB, and AV_{SS}_USB should not be open. Connect AV_{CC} and AV_{CC}_USB to $AV_{CC}Q$, and AV_{SS} and AV_{SS}_USB to $V_{SS}Q$.

3. Current consumption values are for V_{IH}min = V_{CC} – 0.5 V, V_{IH}min = $V_{CC}Q1$ – 0.5 V, and V_{IL}max = 0.5 V with all output pins unloaded.

Table 38.4 DC Characteristics (2-a) [Except USB Transceiver, I²C, ADC, DAC Analog Related Pins]

Condition: Ta = –20 to 75°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	Group 1 input pins*	V_{IH}	$V_{CC}Q \times 0.9$	—	$V_{CC}Q + 0.3$	V	
	CA, EXTAL_RTC, $\overline{\text{RESETP}}$		$V_{CC}Q\_RTC \times 0.9$	—	$V_{CC}Q\_RTC + 0.3$	V	
	Group 2 input pins*		$V_{CC}Q1 \times 0.85$	—	$V_{CC}Q1 + 0.3$	V	$V_{CC}Q1$ = 1.65 to 1.95 V
			2.2	—	$V_{CC}Q1 + 0.3$	V	$V_{CC}Q1$ = 2.7 to 3.6 V
	PTF5 to PTF6		2.2	—	$AV_{CC} + 0.3$	V	
	PTF1 to PTF4		2.0	—	$AV_{CC} + 0.3$	V	
	Other input pins		2.2	—	$V_{CC}Q + 0.3$	V	
Input low voltage	Group 1 input pins*	V_{IL}	–0.3	—	$V_{CC}Q \times 0.1$	V	
	CA, EXTAL_RTC, $\overline{\text{RESETP}}$		–0.3	—	$V_{CC}Q\_RTC \times 0.1$	V	
	Group 2 input pins*		–0.3	—	$V_{CC}Q1 \times 0.15$	V	$V_{CC}Q1$ = 1.65 to 1.95 V
			–0.3	—	$V_{CC}Q1 \times 0.2$	V	$V_{CC}Q1$ = 2.7 to 3.6 V
	PTF5 to PTF6		–0.3	—	$AV_{CC} \times 0.2$	V	
	PTF1 to PTF4		–0.3	—	$AV_{CC} \times 0.2$	V	
	Other input pins		–0.3	—	$V_{CC}Q \times 0.2$	V	
Output high voltage	Group 2 output pins*	V_{OH}	$V_{CC}Q1 \times 0.85$	—	—	V	$V_{CC}Q1$ = 1.65 to 1.95 V I_{OH} = –0.2 mA
			2.4	—	—	V	$V_{CC}Q1$ = 3.0 to 3.6 V I_{OH} = –0.2 mA
			2.2	—	—	V	$V_{CC}Q1$ = 2.7 to 3.6 V I_{OH} = –2 mA
	Other output pins		2.2	—	—	V	I_{OH} = –2mA
Output low voltage	Group 2 output pins*	V_{OL}	—	—	$V_{CC}Q1 \times 0.15$	V	$V_{CC}Q1$ = 1.65 to 1.95 V I_{OL} = 0.2 mA
			—	—	0.5	V	$V_{CC}Q1$ = 2.7 to 3.6 V I_{OL} = 1.6 mA
	Other output pins		—	—	0.5	V	I_{OL} = 1.6 mA

Note: *

Group 1: NMI, ASEMD0, MD0 to MD5, TRST/PTL7, IRQ0/IRL0/PTP0 to IRQ3/IRL3/PTP3, USB1d_TXSE0/IRQ4/AFE_TXOUT/PCC_DRV/PTG5, USB1d_RCV/IRQ5/AFE_FS/PCC_REG/PTG6, EXTAL, RESETM, AFE_RDET/IIC_SDA/PTE5, and AFE_RXIN/IIC_SCL/PTE6

Group 2: A0/PTR0, A1 to A18, A19/PTR1 to A25/PTR7, CKIO, RD/WR, CAS/PTH5, WE3/DQMUU/ICIOWR, WE2/DQMUL/ICIORD, CKE/PTH4, RAS/PTH6, WE1/DQMLU/WE, WE0/DQMLL, CS2, CS3, CS6B/CE1B/PTM0, CS6A/CE2B, CS5B/CE1A/PTM1, CS5A/CE2A, BACK, CS0, BREQ, CS4, BS, RD, WAIT/PCC_WAIT, DREQ0/PINT0/PTM6, DACK0/PINT1/PTM4, TEND0/PINT2/PTM2, DREQ1/PTM7, DACK1/PTM5, TEND1/PINT3/PTM3, D0 to D15, D16/PTA0 to D23/PTA7, and D24/PTB0 to D31/PTB7

Table 38.4 DC Characteristics (2-b) [I²C Related Pins*]

Conditions: $V_{cc}Q$ = –2.7 to 3.6 V, V_{cc} = 1.4 to 1.6 V, Ta = –20 to 75°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage	VCCQ	2.7	3.3	3.6	V	
Input high voltage	VIH	$V_{cc}Q \times 0.7$	—	$V_{cc}Q + 0.3$	V	
Input low voltage	VIL	–0.3	—	$V_{cc}Q \times 0.3$	V	
Output low voltage	VOL	—	—	0.4	V	IOL = 1.6 mA
Permissible output low current	IOL	—	—	10	mA	

Note: * The IIC_SCL and IIC_SDA pins (open-drain pins).

Table 38.4 DC Characteristics (2-c) [USB Transceiver Related Pins*[1]]

Condition: Ta = –20 to 75°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage*[2]	AV_{CC}_USB	3.0	3.3	3.6	V	
Differential input sensitivity	V_{DI}	0.2	—	—	V	|(DP) – (DM)|
Differential common mode range	V_{CM}	0.8	—	2.5	V	
Single ended receiver threshold voltage	V_{SE}	0.8	—	2.0	V	
Output high voltage	V_{OH}	2.5	—	AV_{CC}_USB	V	
Output low voltage	V_{OL}	—	—	0.3	V	
Tray state leakage voltage	I_{LO}	–10	—	10	μA	$0V < V_{IN} < 3.3V$

Notes: 1. D+ and D– pins.
2. AV_{CC}_USB should satisfy the condition $V_{CC}Q \leq AV_{CC}$_USB and be supplied AV_{CC}_USB and AV_{SS}_USB.

Table 38.5 Permissible Output Current Values

Conditions: $V_{CC}Q = V_{CC}$Q_RTC = V_{CC}_Q1 = 2.7 to 3.6 V,
$V_{CC} = V_{CC}$_PLL1 = V_{CC}_PLL2 = V_{CC}_RTC = 1.4 to 1.6 V, AV_{CC} = 3.0 to 3.6 V,
AV_{CC}_USB = 3.0 to 3.6 V, Ta = –20 to 75°C

Item	Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	I_{OL}	—	—	2.0	mA
Permissible output low current (total)	ΣI_{OL}	—	—	120	mA
Permissible output high current (per pin)	$-I_{OH}$	—	—	2.0	mA
Permissible output high current (total)	$\Sigma (-I_{OH})$	—	—	40	mA

Note: * To ensure chip reliability, do not exceed the output current values given in table 38.5.

38.4 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.

$V_{CC}Q1$ can be set to 2.7 to 3.6 V or 1.65 to 1.95 V. When $V_{CC}Q1$ is set to 1.65 to 1.95 V, the drivability of the I/O buffer will be its highest specifications. For the change of the drivability of the I/O buffer, see section 34.1.23, USB Transceiver Control Register (UTRCTL).

Table 38.6 Maximum Operating Frequencies

Conditions: $V_{CC}Q$ = 2.7 to 3.6 V, $V_{CC}Q1$ = 2.7 to 3.6 V or 1.65 to 1.95 V,
V_{CC} = 1.4 to 1.6 V, Ta = −20 to 75°C

Item		Symbol	Min.	Typ.	Max.	Unit	Remarks
Operating frequency	CPU, cache (Iϕ)	f	24	—	133.34	MHz	133 MHz version
	External bus (Bϕ)*		24	—	66.67		
	Peripheral module (Pϕ)		8.34	—	33.34		

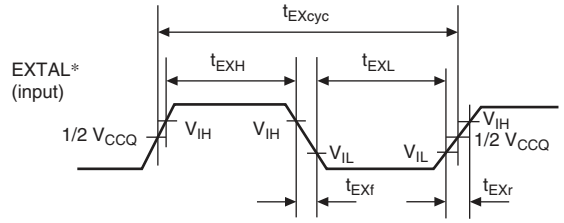
Note: * When using the USB host controller, the external bus frequency (Bϕ) should be set to 32 MHz or higher.

38.4.1 Clock Timing

Table 38.7 Clock Timing

Conditions: $V_{CC}Q$ = 2.7 to 3.6 V, $V_{CC}Q1$ = 2.7 to 3.6 V or 1.65 to 1.95 V, V_{CC} = 1.4 to 1.6 V,
Ta = –20 to 75°C,
Maximum external bus operating frequency: 66.67 MHz

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency	f_{EX}	10	66.67	MHz	38.1
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	
CKIO clock output frequency	f_{OP}	20	66.67	MHz	38.2
CKIO clock output cycle time	t_{cyc}	15	50	ns	
CKIO clock output low pulse width	t_{CKOL}	3	—	ns	
CKIO clock output high pulse width	t_{CKOH}	3	—	ns	
CKIO clock output rise time	t_{CKOr}	—	3	ns	
CKIO clock output fall time	t_{CKOf}	—	3	ns	
CKIO clock input frequency	f_{CKI}	20	66.67	MHz	38.3
CKIO clock input cycle time	t_{CKICYC}	15	50	ns	
CKIO clock input low pulse width	t_{CKIL}	3	—	ns	
CKIO clock input high pulse width	t_{CKIH}	3	—	ns	
CKIO clock input rise time	t_{CKIR}	—	3	ns	
CKIO clock input fall time	t_{CKIF}	—	3	ns	
$\overline{\text{RESETP}}$ setup time	t_{RESPS}	20	—	ns	38.4
$\overline{\text{RESETP}}$ assert time	t_{RESPW}	20	—	t_{cyc}	38.4, 38.5
$\overline{\text{RESETM}}$ assert time	t_{RESMW}	20	—	t_{cyc}	38.5
Power-On Oscillation Settling Time	t_{SOC1}	10	—	ms	38.4
Oscillation Settling Time on Return from Standby 1	t_{SOC2}	10	—	ms	38.5
Oscillation Settling Time on Return from Standby 2	t_{SOC3}	10	—	ms	38.6
PLL synchronization settling time	t_{PLL}	100	—	μs	38.7



Note: * When clock is input from EXTAL pin

Figure 38.1 EXTAL Clock Input Timing

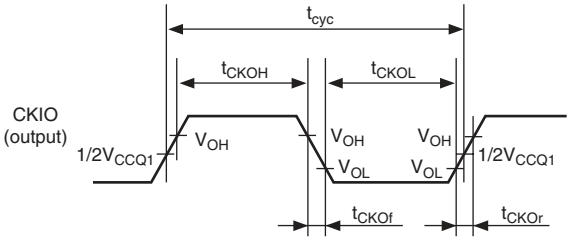


Figure 38.2 CKIO Clock Output Timing

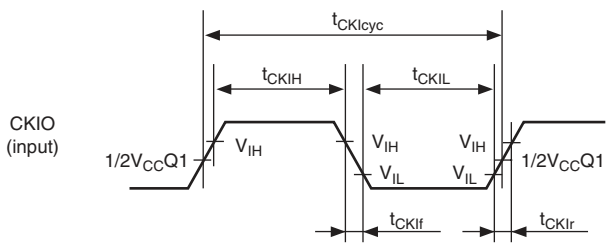
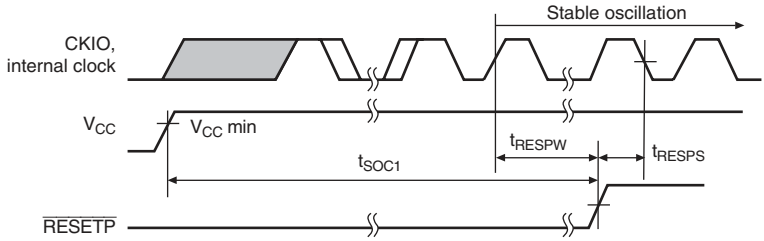
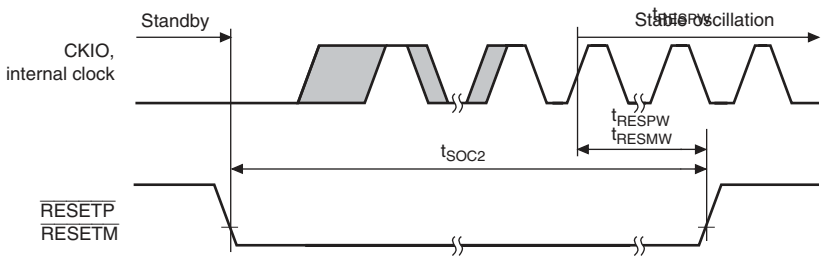


Figure 38.3 CKIO Clock Input Timing



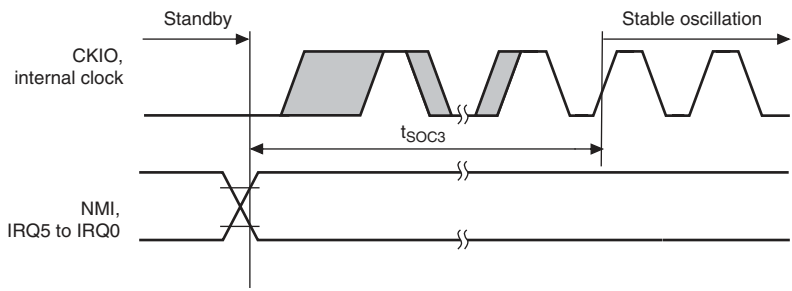
Note: Oscillation settling time when on-chip oscillator is used

Figure 38.4 Power-On Oscillation Settling Time



Note: Oscillation settling time when on-chip oscillator is used

Figure 38.5 Oscillation Settling Time on Return from Standby (Return by Reset)



Note: Oscillation settling time when on-chip oscillator is used

Figure 38.6 Oscillation Settling Time on Return from Standby (Return by NMI or IRQ)

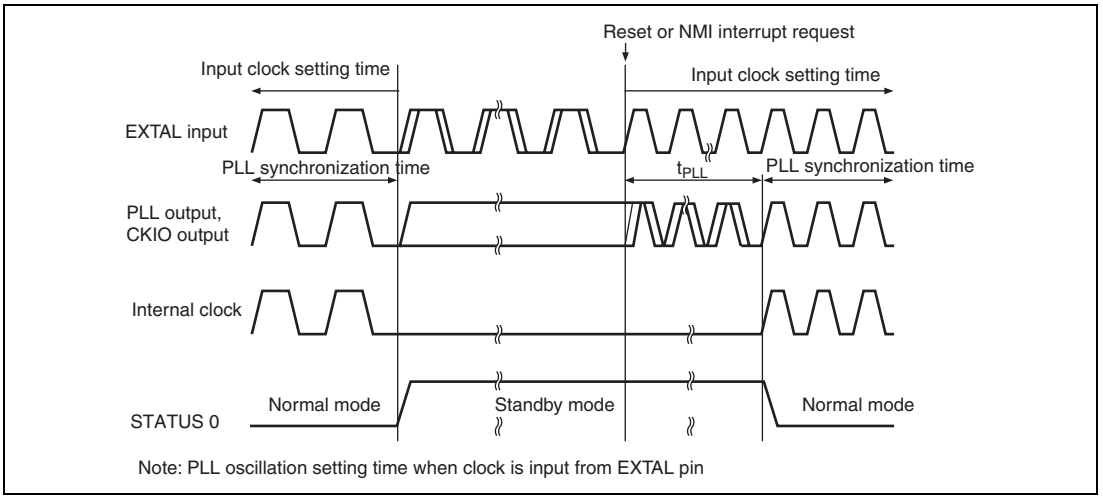


Figure 38.7 PLL Synchronization Settling Time by Reset, NMI or IRQ Interrupts

38.4.2 Control Signal Timing

Table 38.8 Control Signal Timing

Conditions: $V_{CC}Q$ = 2.7 to 3.6 V, $V_{CC}Q1$ = 2.7 to 3.6 V or 1.65 to 1.95 V,
V_{CC} = 1.4 to 1.6 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{RESETP}}$ pulse width	t_{RESPW}	20*[3]	—	tcyc*[2*4]	38.8, 38.9
$\overline{\text{RESETP}}$ setup time*[1]	t_{RESPS}	23	—	ns	
$\overline{\text{RESETP}}$ hold time	t_{RESPH}	2	—	ns	
$\overline{\text{RESETM}}$ pulse width	t_{RESPW}	20*[3]	—	tcyc*[2*4]	
$\overline{\text{RESETM}}$ setup time*[1]	t_{RESPS}	23	—	ns	
$\overline{\text{RESETM}}$ hold time	t_{RESPH}	2	—	ns	
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	$1/2t_{cyc}$ + 7	—	ns	38.10
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	$1/2t_{cyc}$ + 2	—	ns	
NMI setup time*[1]	t_{NMIS}	8	—	ns	38.9
NMI hold time	t_{NMIH}	3	—	ns	
IRQ5 to IRQ0 setup time*[1]	t_{IRQS}	8	—	ns	
IRQ5 to IRQ0 hold time	t_{IRQH}	3	—	ns	
$\overline{\text{BACK}}$ delay time	t_{BACKD}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 13	ns	38.10, 38.11
STATUS0 delay time	t_{STD}	—	18	ns	
Bus tri-state delay time 1	t_{BOFF1}	0	30	ns	
Bus tri-state delay time 2	t_{BOFF2}	0	30	ns	
Bus buffer-on time 1	t_{BON1}	0	30	ns	38.10, 38.11
Bus buffer-on time 2	t_{BON2}	0	30	ns	

Notes: 1. $\overline{\text{RESETP}}$, NMI, and IRQ5 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. The upper limits of the external bus clock are 66.67 MHz (133 MHz version).
3. In standby mode, $t_{RESPW} = t_{SOC2}$ (10 ms). When the clock multiplication ratio is changed, $t_{RESPW} = t_{PLL}$ (100 µs).
4. t_{cyc} means the external bus clock cycle (B clock cycle).

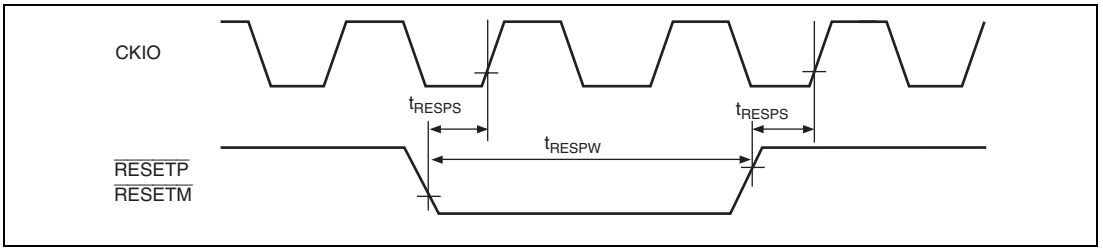


Figure 38.8 Reset Input Timing

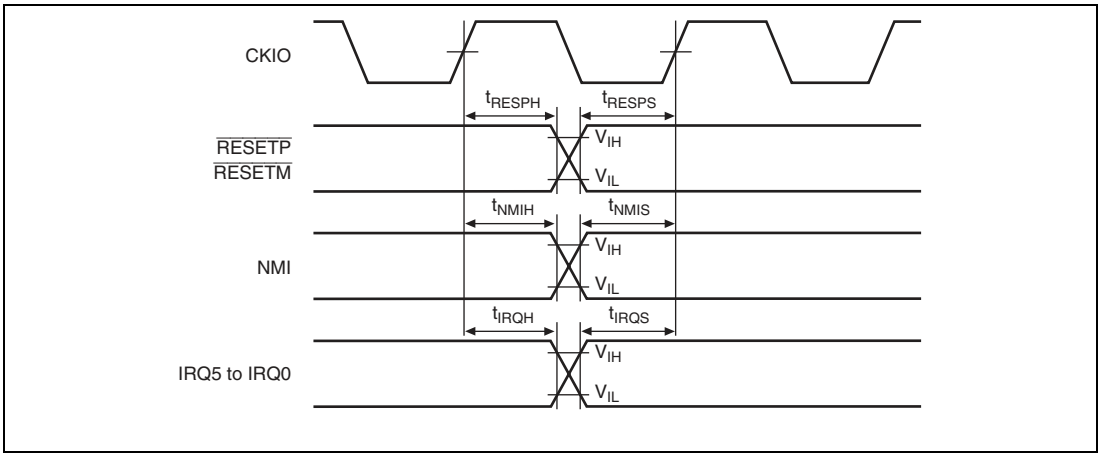


Figure 38.9 Interrupt Signal Input Timing

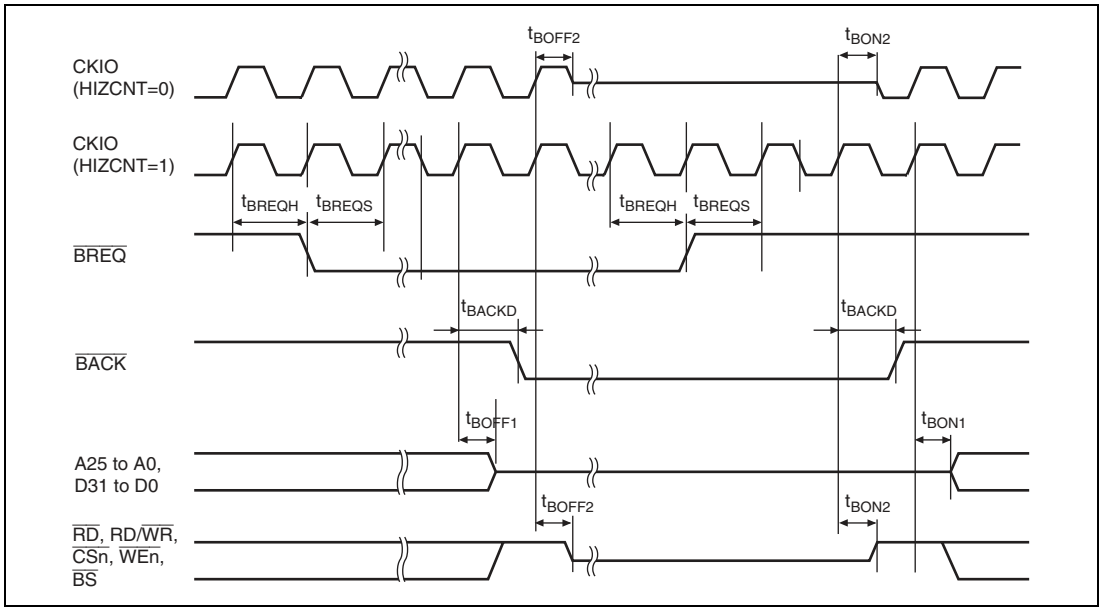


Figure 38.10 Bus Release Timing

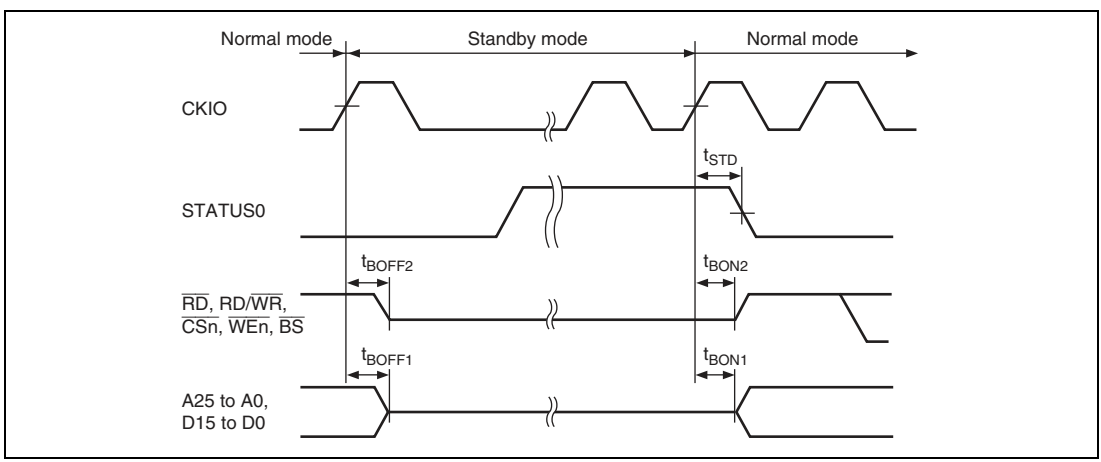


Figure 38.11 Pin Drive Timing at Standby

38.4.3 AC Bus Timing

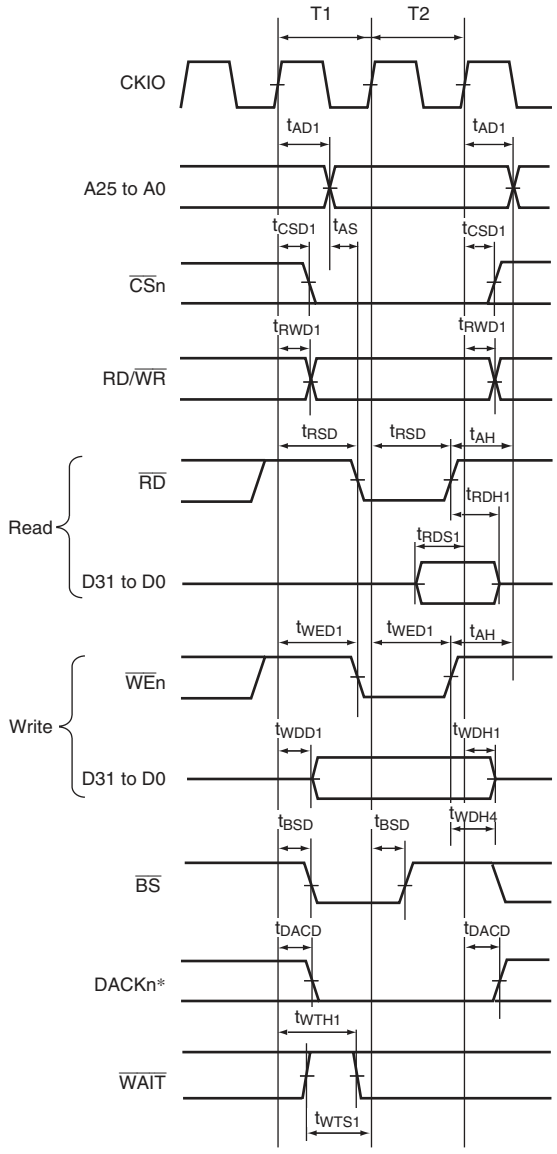
Table 38.9 Bus Timing

Conditions: Clock Mode 0, $V_{CC}Q$ = 2.7 to 3.6 V, $V_{CC}Q1$ = 2.7 to 3.6 V or 1.65 to 1.95 V, V_{CC} = 1.4 to 1.6 V, Ta = –20 to 75°C

		66.67 MHz			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
Address delay time 1	t_{AD1}	1	13	ns	38.12 to 38.42
Address delay time 2	t_{AD2}	$1/2t_{cyc}$	$1/2t_{cyc} + 13$	ns	38.19
Address setup time	t_{AS}	0	—	ns	38.12 to 38.19
Address hold time	t_{AH}	0	—	ns	38.12, 38.13
$\overline{BS}$ delay time	t_{BSD}	—	13	ns	38.12 to 38.36, 38.37, 38.38
$\overline{CS}$ delay time 1	t_{CSD1}	1	13	ns	38.12 to 38.36, 38.37 to 38.42
Read/write delay time 1	t_{RWD1}	1	13	ns	38.12 to 38.36, 38.37 to 38.42
Read strobe delay time	t_{RSD}	$1/2t_{cyc}$	$1/2t_{cyc} + 13$	ns	38.12 to 38.19, 38.39, 38.40
Read data setup time 1	t_{RDS1}	$1/2t_{cyc} + 10$	—	ns	38.12 to 38.18, 38.37 to 38.42
Read data setup time 2	t_{RDS2}	7	—	ns	38.20 to 38.23, 38.28 to 38.30, 38.37, 38.38
Read data setup time 3	t_{RDS3}	$1/2t_{cyc} + 10$	—	ns	38.19
Read data hold time 1	t_{RDH1}	0	—	ns	38.12 to 38.18, 38.37 to 38.42
Read data hold time 2	t_{RDH2}	2	—	ns	38.20 to 38.23, 38.28 to 38.30, 38.37, 38.38
Read data hold time 3	t_{RDH3}	0	—	ns	38.19
Write enable delay time 1	t_{WED1}	$1/2t_{cyc}$	$1/2t_{cyc} + 13$	ns	38.12 to 38.17, 38.39, 38.40
Write enable delay time 2	t_{WED2}	—	13	ns	38.18

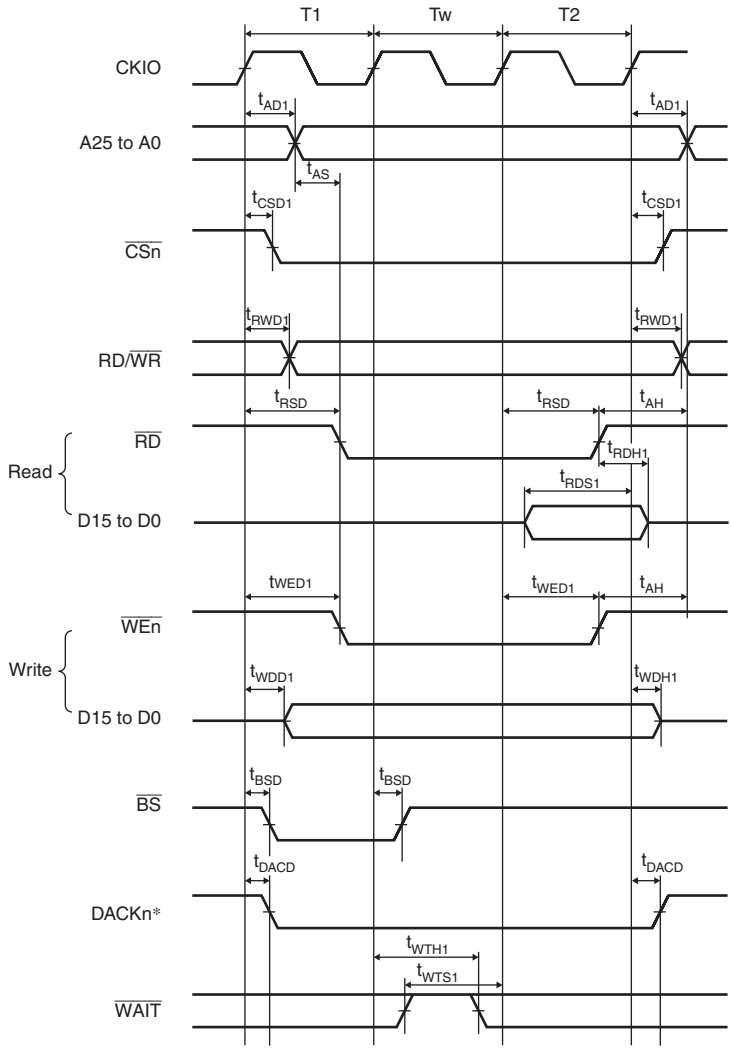
Item	Symbol	66.67 MHz Min.	66.67 MHz Max.	Unit	Figure
Write data delay time 1	t_{WDD1}	—	13	ns	38.12 to 38.18, 38.39 to 38.42
Write data delay time 2	t_{WDD2}	—	13	ns	38.24 to 38.27, 38.31 to 38.33, 38.37, 38.38
Write data hold time 1	t_{WDH1}	1	—	ns	38.12 to 38.18, 38.37 to 38.42
Write data hold time 2	t_{WDH2}	1	—	ns	38.24 to 38.27, 38.31 to 38.33, 38.37, 38.38
Write data hold time 4	t_{WDH4}	0	—	ns	38.12
Write data hold time 5	t_{WDH5}	1	—	ns	38.39 to 38.42
$\overline{WAIT}$ setup time 1	t_{WTS1}	$1/2t_{cyc} + 7$	—	ns	38.12 to 38.19, 38.40, 38.42
$\overline{WAIT}$ hold time 1	t_{WTH1}	$1/2t_{cyc} + 2$	—	ns	38.12 to 38.19, 38.40, 38.42
$\overline{RAS}$ delay time 1	t_{RASD1}	1	13	ns	38.20 to 38.36, 38.37, 38.38
$\overline{CAS}$ delay time 1	t_{CASD1}	1	13	ns	38.20 to 38.36, 38.37, 38.38
DQM delay time 1	t_{DQMD1}	1	13	ns	38.20 to 38.36, 38.37, 38.38
CKE delay time 1	t_{CKED1}	1	13	ns	38.35, 38.36, 38.37, 38.38
DACK delay time	t_{DACD}	—	13	ns	38.12 to 38.36, 38.37
$\overline{ICIORD}$ delay time	t_{ICRSD}	—	$1/2t_{cyc} + 13$	ns	38.39, 38.40
$\overline{ICIOWR}$ delay time	t_{ICWSD}	—	$1/2t_{cyc} + 13$	ns	38.41, 38.42
$\overline{IOIS16}$ setup time	t_{IO16S}	$1/2t_{cyc} + 6$	—	ns	38.42
$\overline{IOIS16}$ hold time	t_{IO16H}	$1/2t_{cyc} + 4$	—	ns	38.42
$\overline{REFOUT}$, $\overline{IRQOUT}$ delay time	t_{REFOD}	—	$1/2t_{cyc} + 13$	ns	38.43

38.4.4 Basic Timing



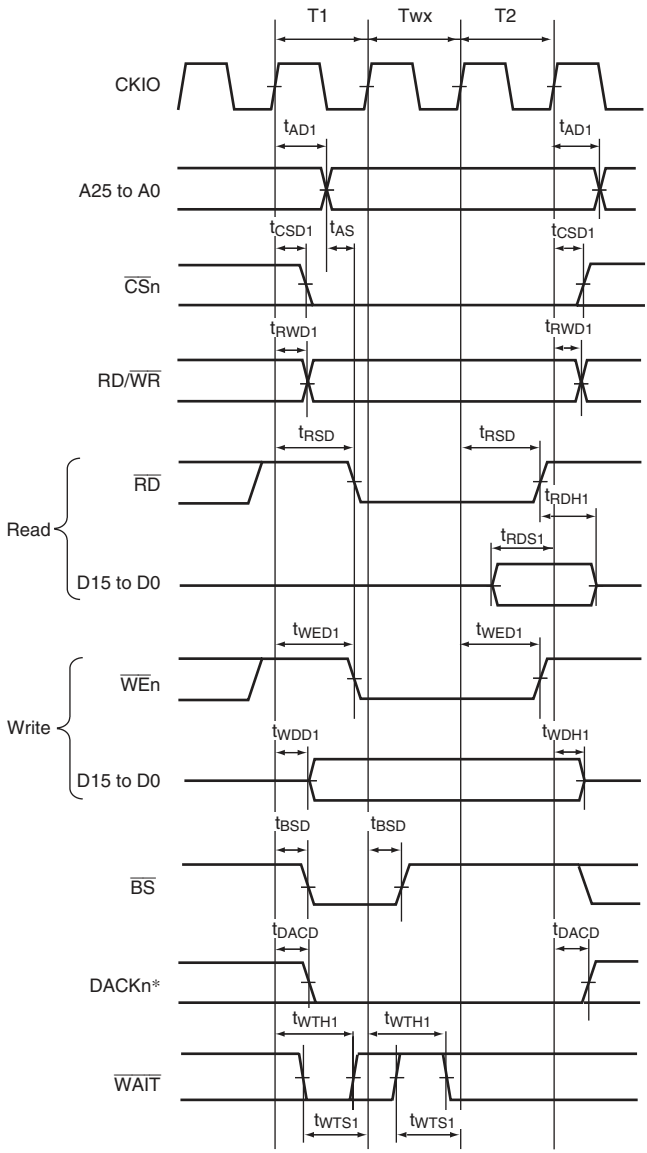
Note: * Waveform when active low is specified for DACKn.

Figure 38.12 Basic Bus Cycle in Normal Space (No Wait)



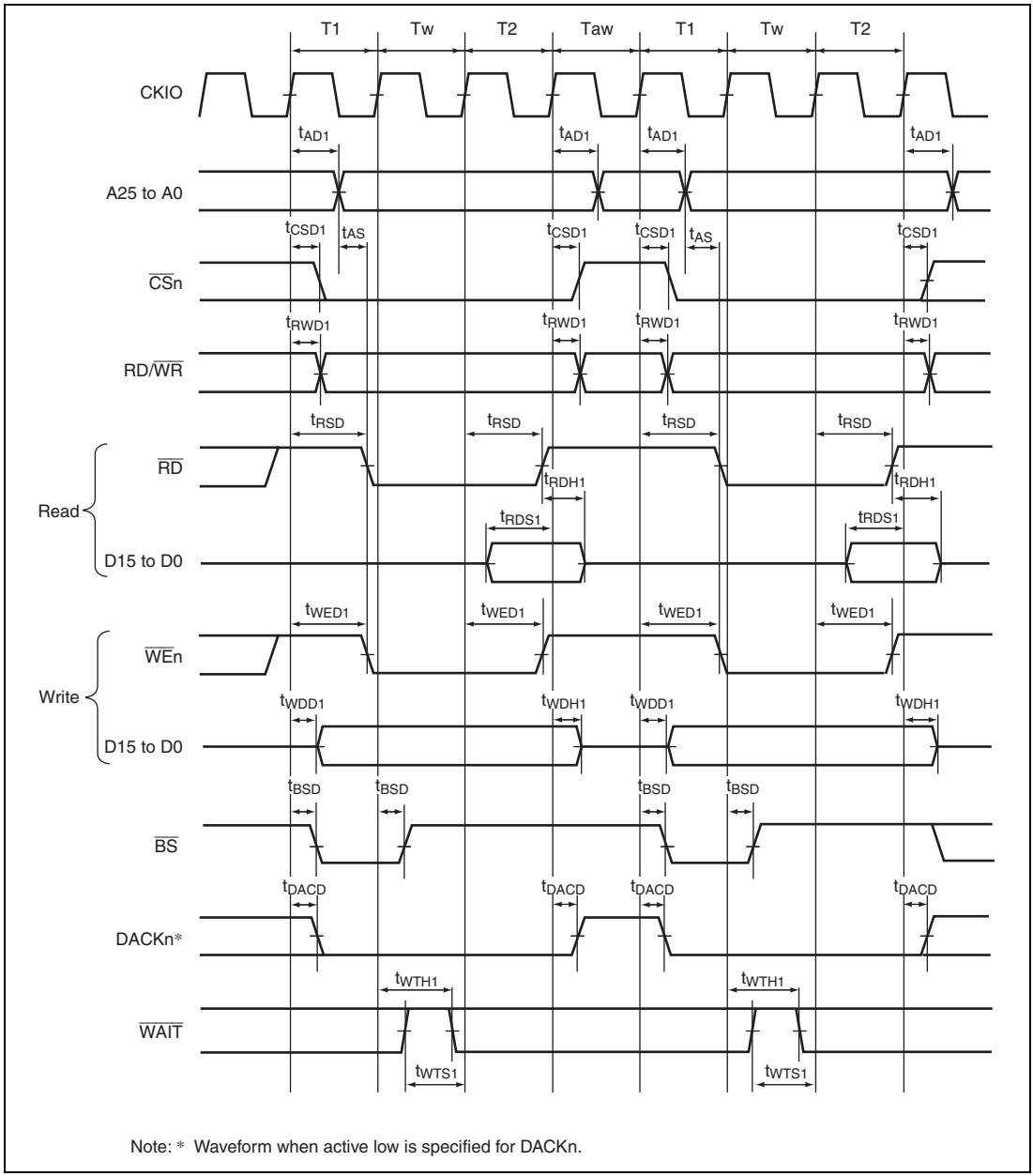
Note: * Waveform when active low is specified for DACKn.

Figure 38.13 Basic Bus Cycle in Normal Space (Software Wait 1)



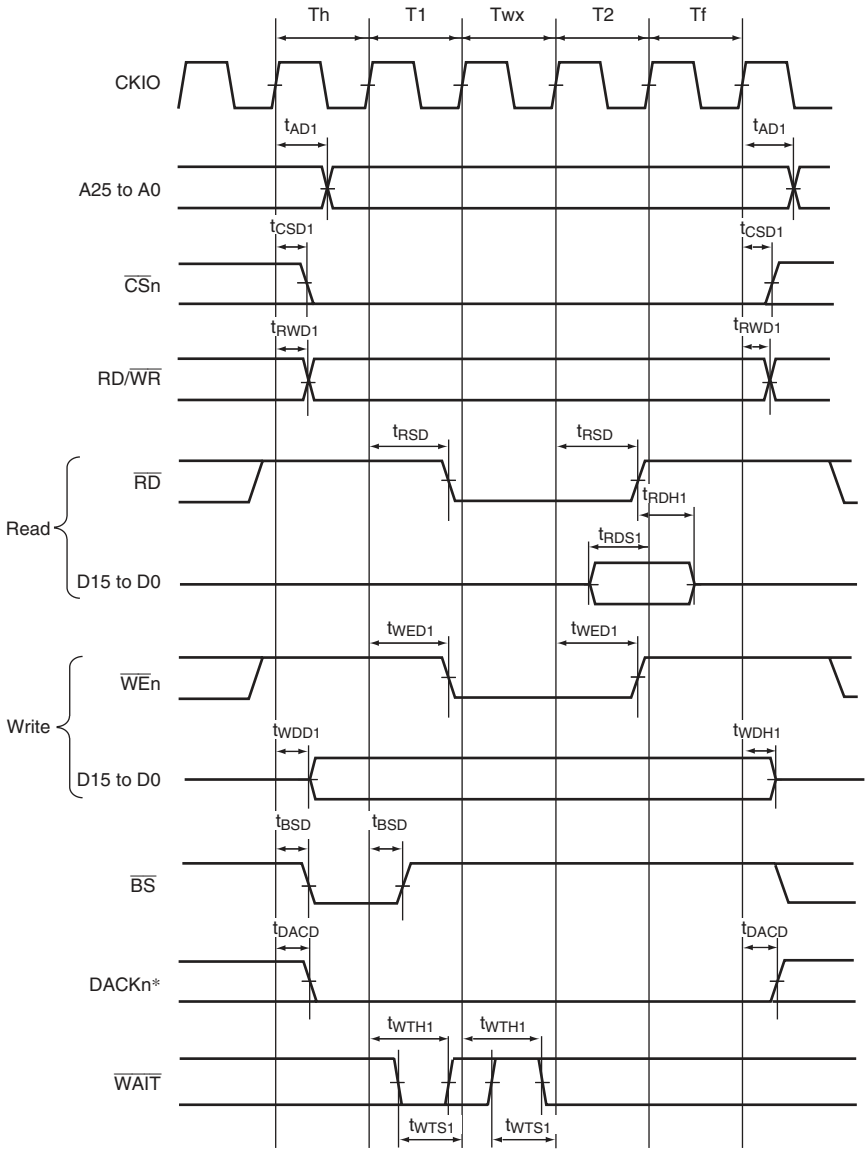
Note: * Waveform when active low is specified for DACKn.

Figure 38.14 Basic Bus Cycle in Normal Space (External Wait 1 Input)



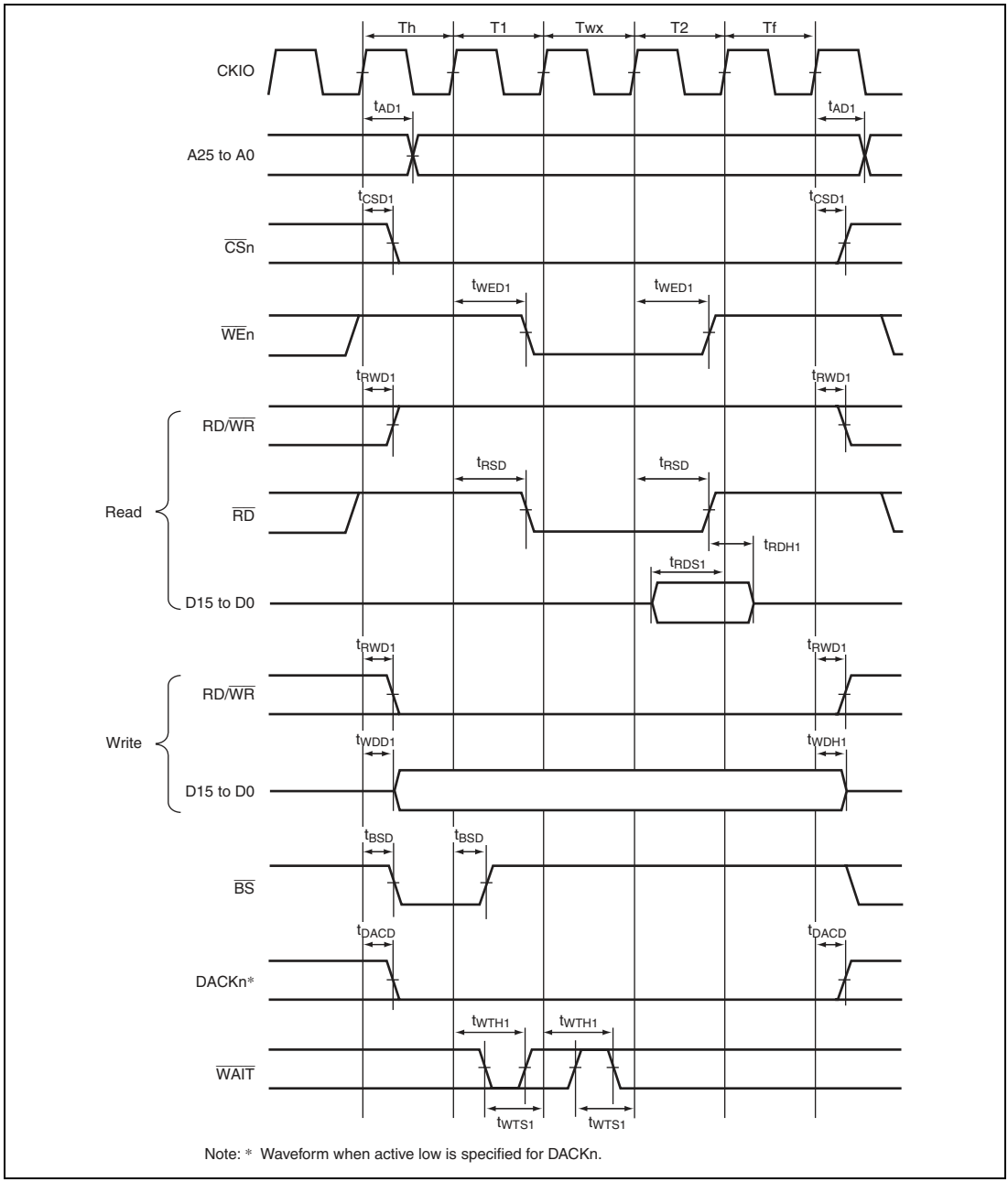
Note: * Waveform when active low is specified for DACKn.

Figure 38.15 Basic Bus Cycle in Normal Space (Software Wait 1, External Wait Valid (WM Bit = 0), No Idle Cycle)



Note: * Waveform when active low is specified for DACKn.

Figure 38.16 CS Extended Bus Cycle in Normal Space (SW = 1 Cycle, HW = 1 Cycle, External Wait 1 Input)



Note: * Waveform when active low is specified for DACKn.

Figure 38.17 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, External Wait 1 Input, BAS = 0 (UB and LB in Write Cycle Controlled))

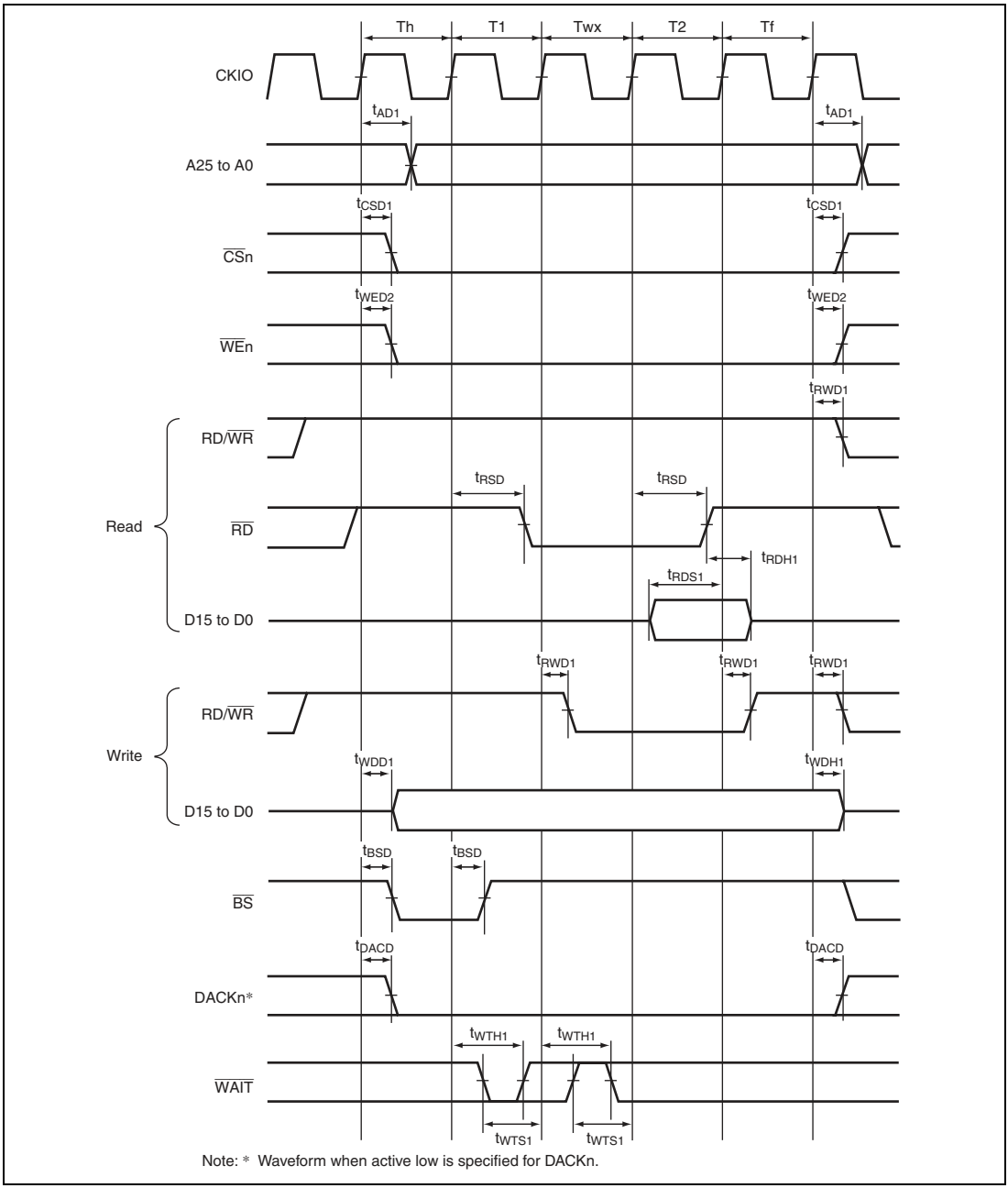
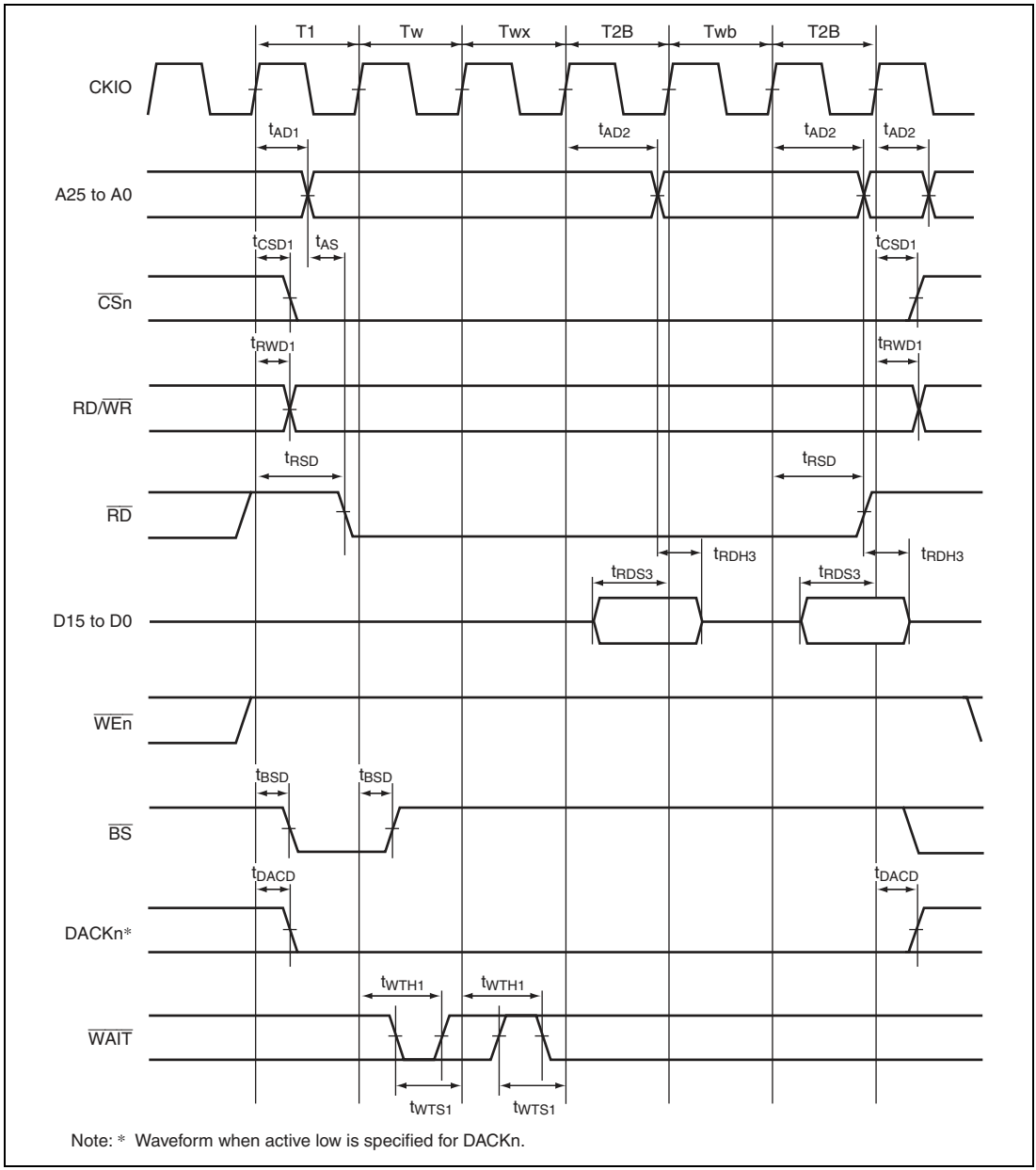


Figure 38.18 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, External Wait 1 Input, BAS = 1 (WE in Write Cycle Controlled))

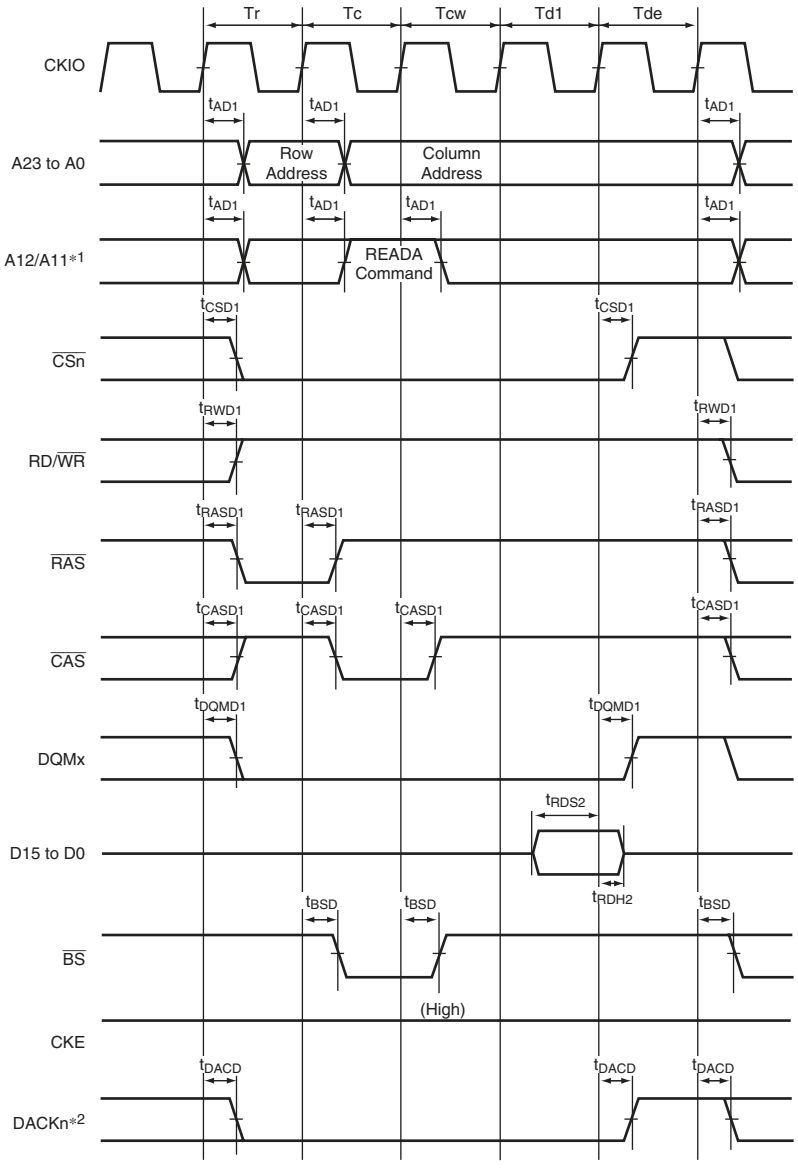
38.4.5 Burst ROM Timing



Note: * Waveform when active low is specified for DACKn.

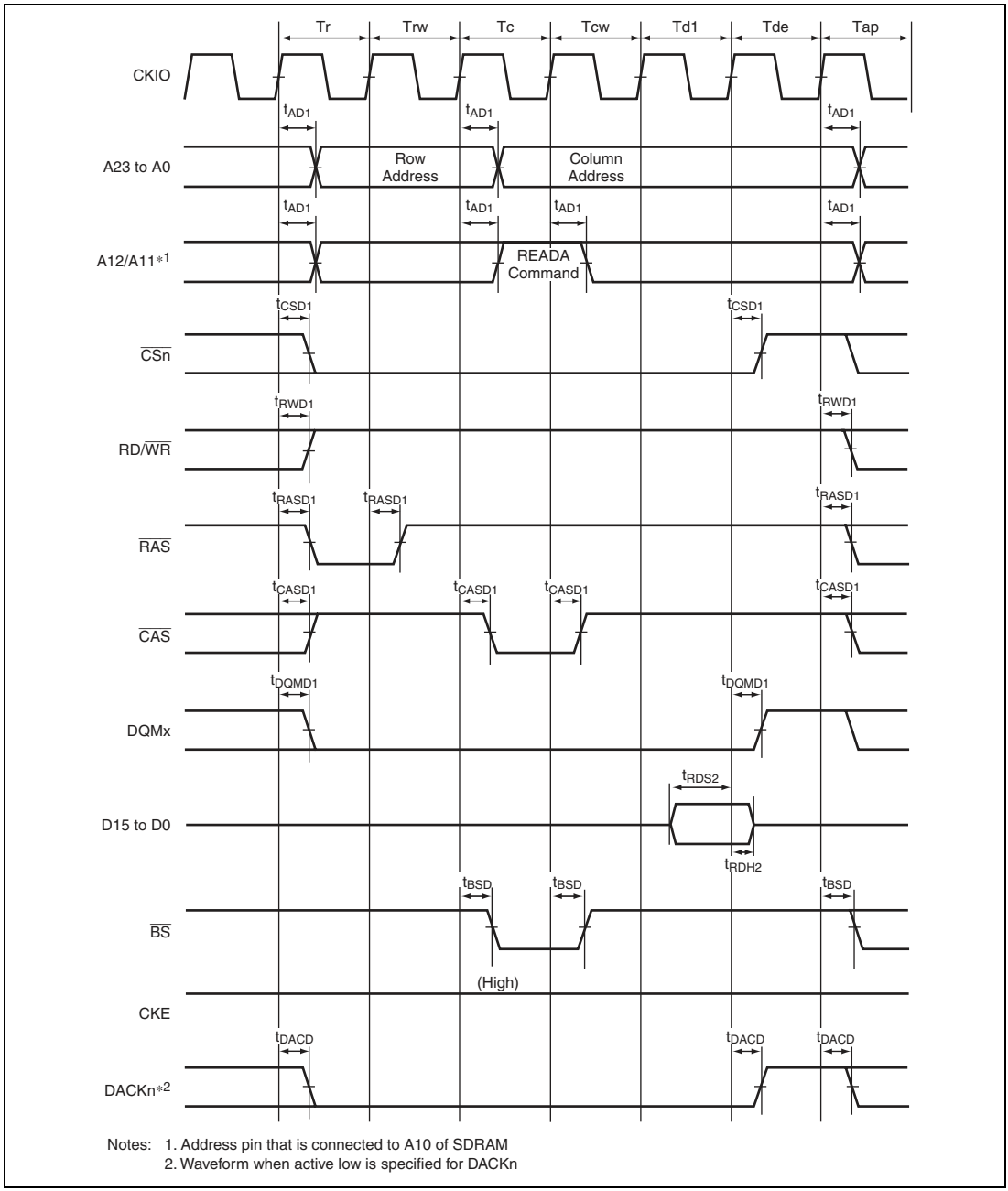
Figure 38.19 Read Bus Cycle of Burst ROM
(Software Wait 1, External Wait 1 Input, Burst Wait 1, Number of Burst 2)

38.4.6 SDRAM Timing



Notes: 1. Address pin that is connected to A10 of SDRAM
2. Waveform when active low is specified for DACKn

Figure 38.20 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 38.21 Single Read Bus Cycle of SDRAM (Auto Precharge Mode, CAS Latency 2, TRCD = 2 Cycles, TRP = 2 Cycles)

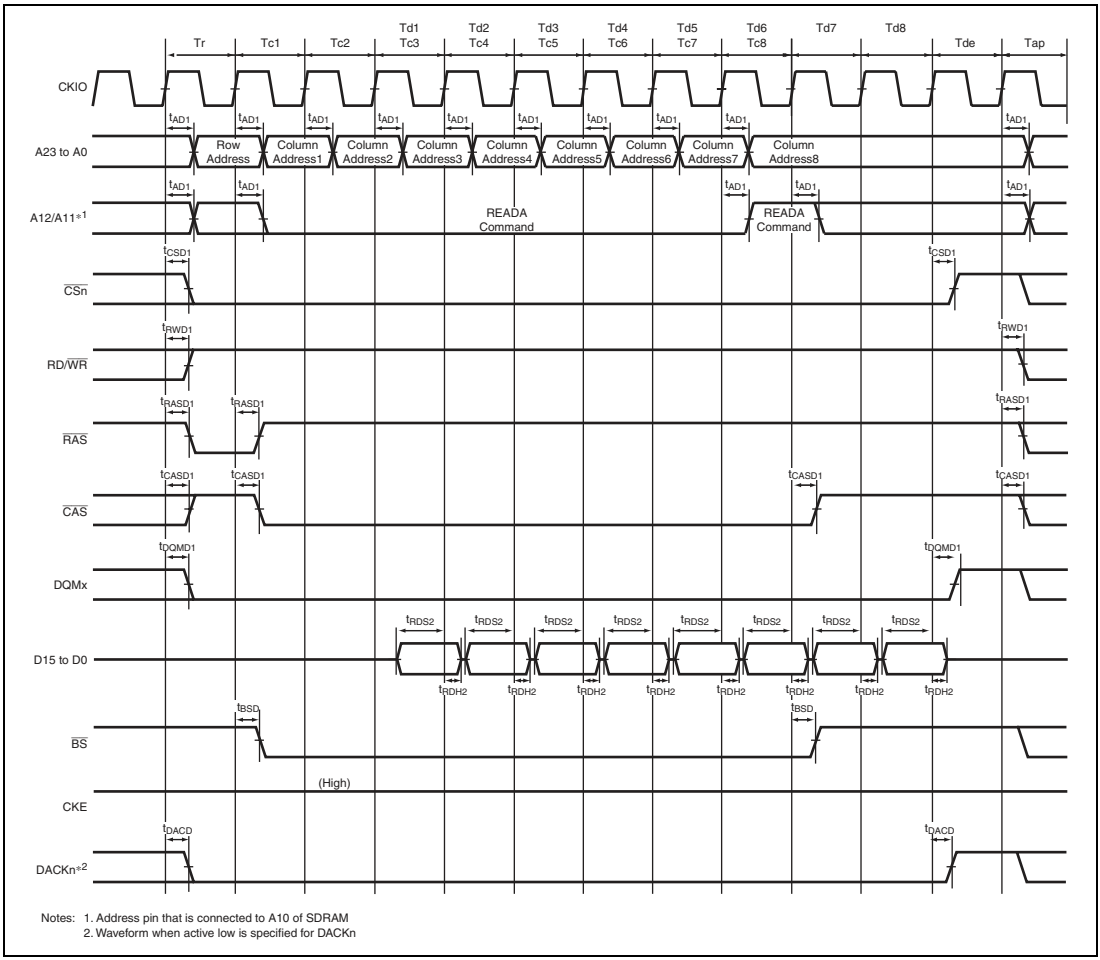
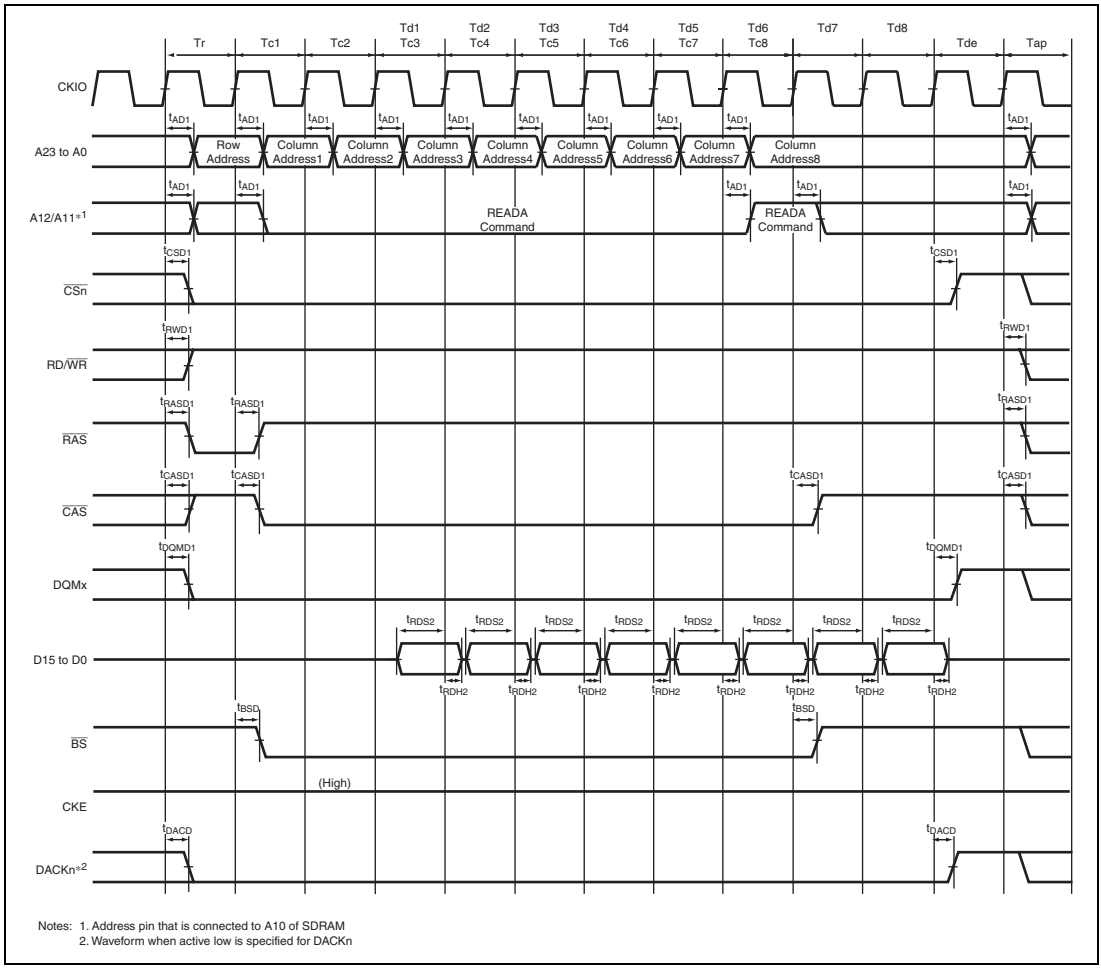
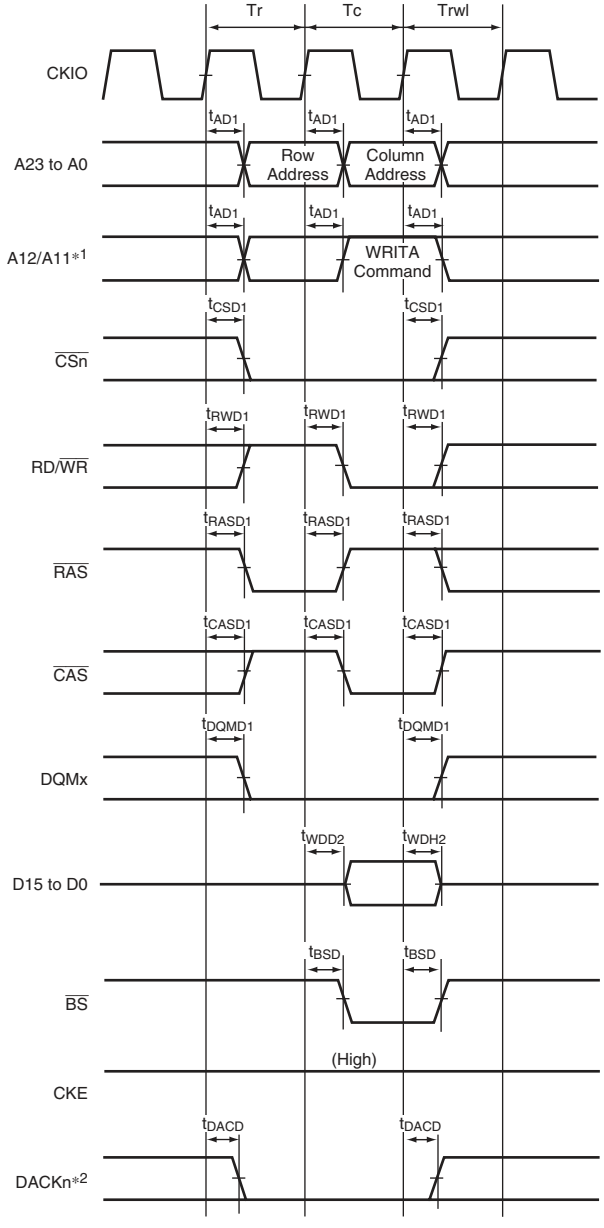


Figure 38.22 Burst Read Bus Cycle of SDRAM (Single Read × 8)
(Auto Precharge Mode, CAS Latency 2, TRCD = 1 Cycle, TRP = 2 Cycles)



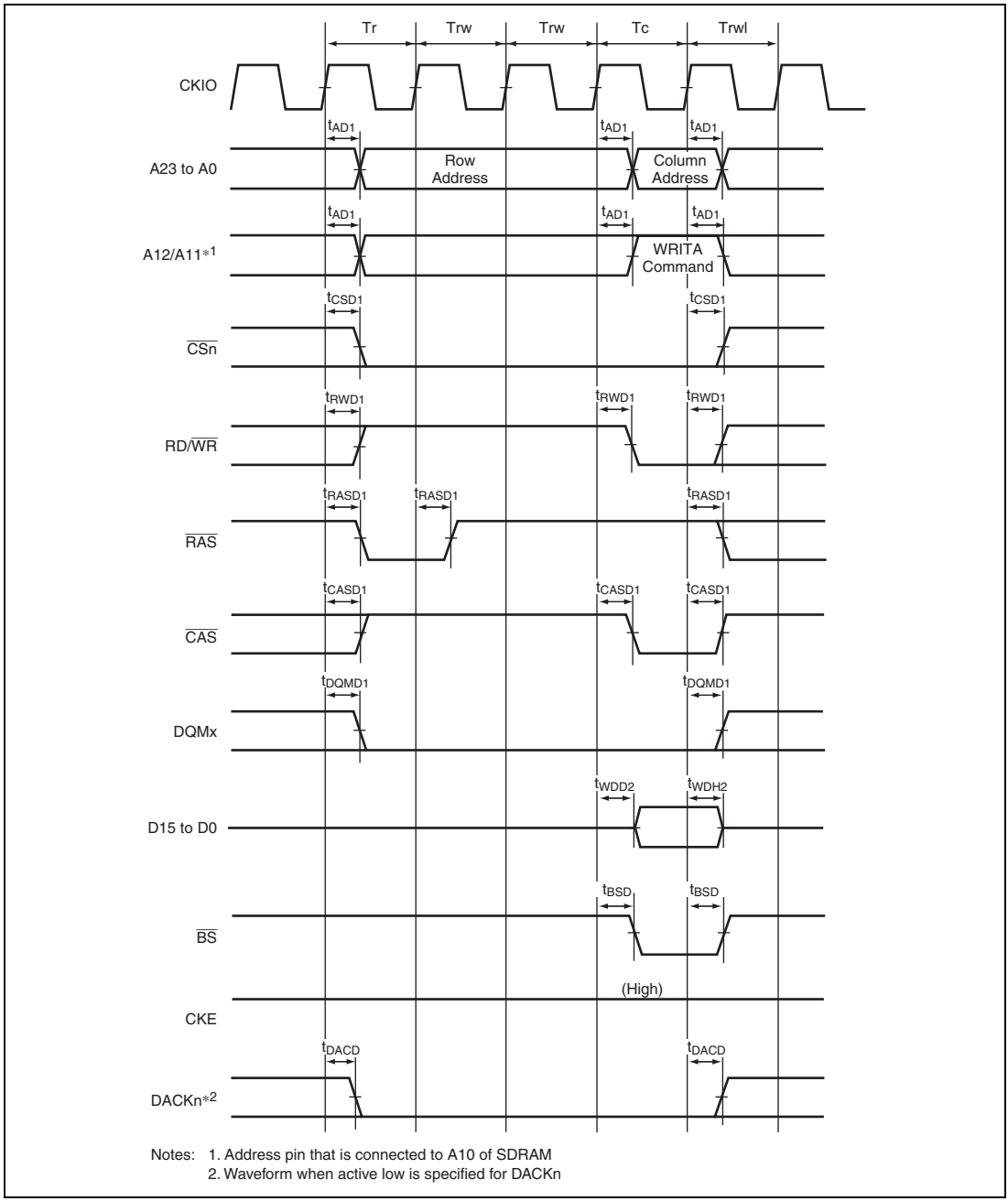
Notes: 1. Address pin that is connected to A10 of SDRAM
2. Waveform when active low is specified for DACKn

Figure 38.23 Burst Read Bus Cycle of SDRAM (Single Read × 8)
(Auto Precharge Mode, CAS Latency 2, TRCD = 2 Cycles, TRP = 1 Cycle)



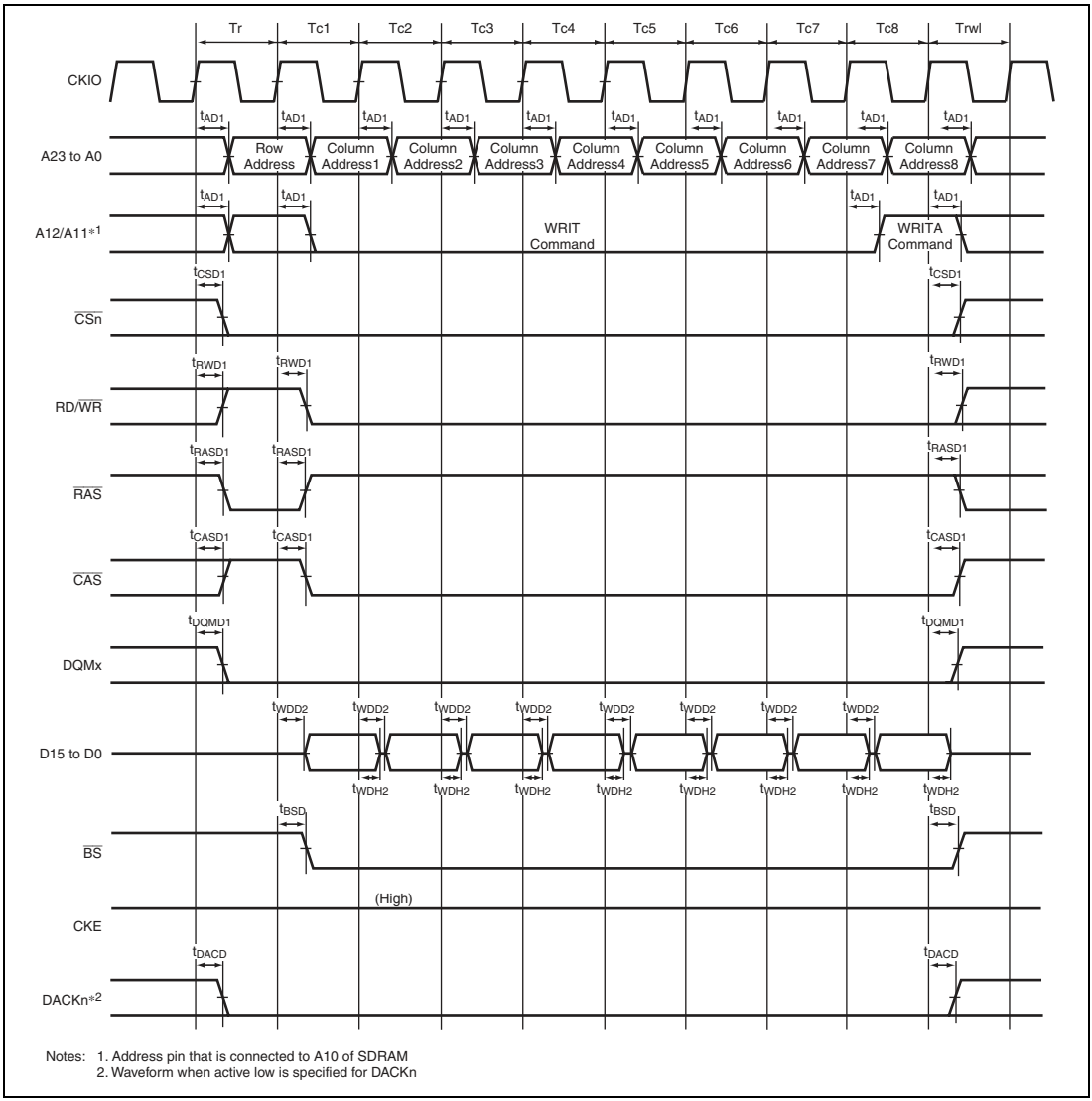
Notes: 1. Address pin that is connected to A10 of SDRAM
2. Waveform when active low is specified for DACKn

Figure 38.24 Single Write Bus Cycle of SDRAM (Auto Precharge Mode, TRWL = 1 Cycle)



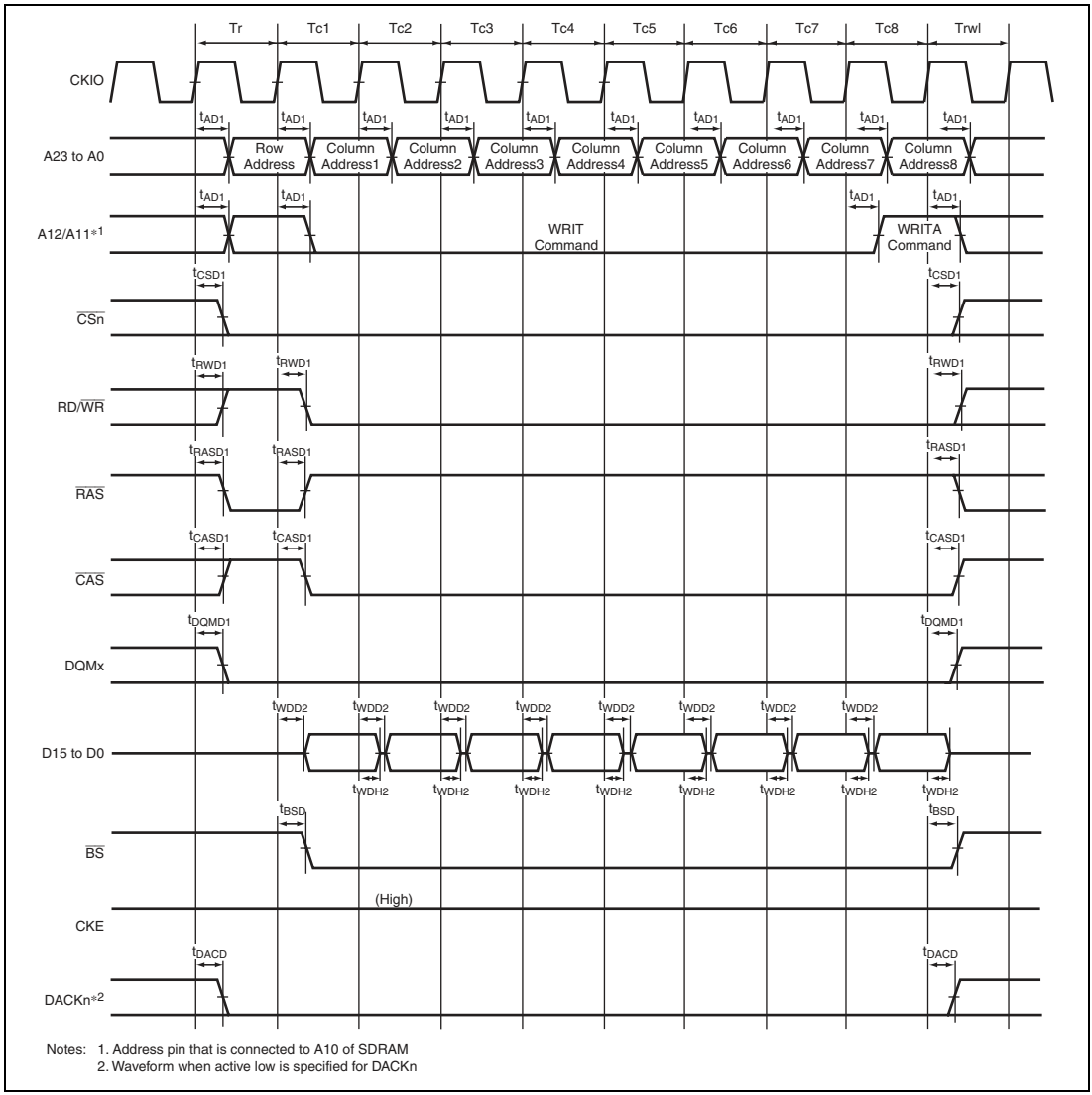
Notes: 1. Address pin that is connected to A10 of SDRAM
2. Waveform when active low is specified for DACKn

Figure 38.25 Single Write Bus Cycle of SDRAM (Auto Precharge Mode, TRCD = 3 Cycles, TRWL = 1 Cycle)



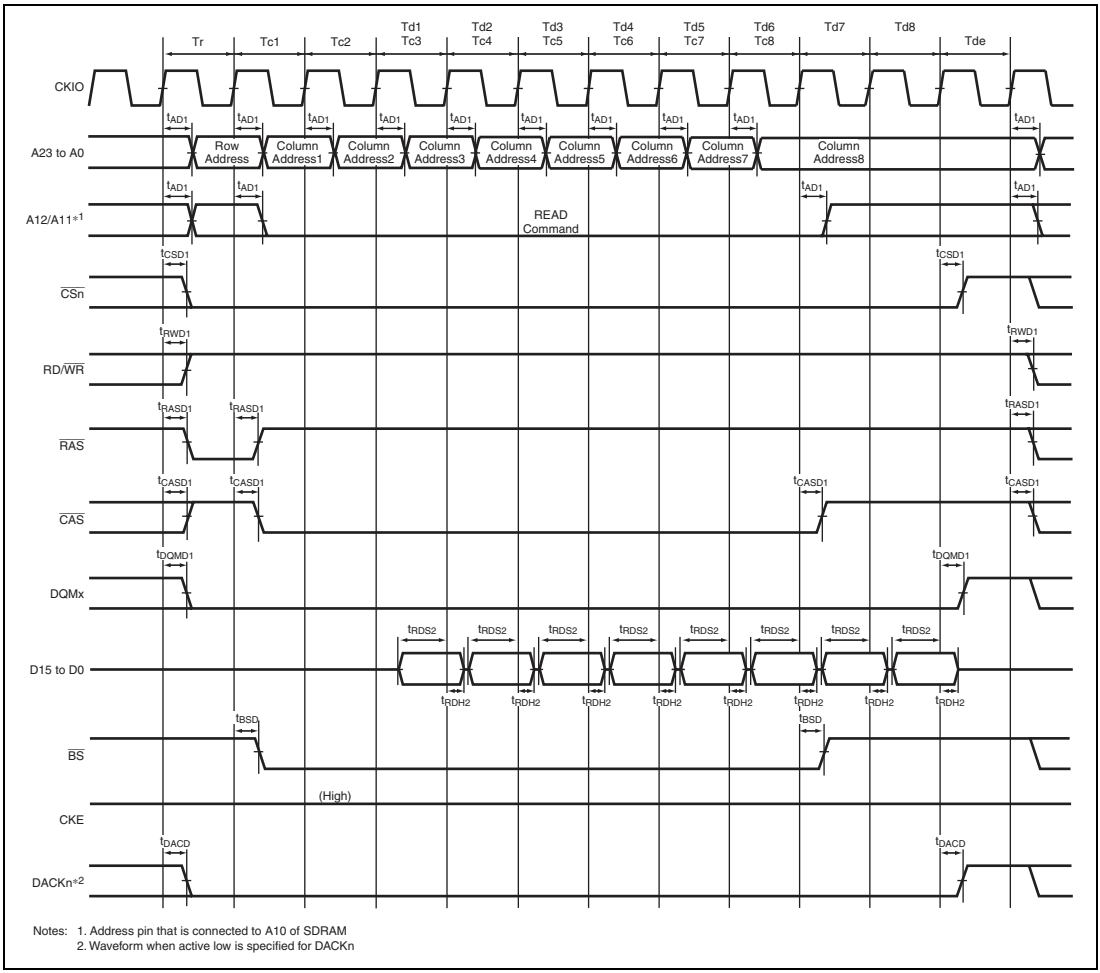
Notes: 1. Address pin that is connected to A10 of SDRAM
2. Waveform when active low is specified for DACKn

Figure 38.26 Burst Write Bus Cycle of SDRAM (Single Write × 8) (Auto Precharge Mode, TRCD = 1 Cycle, TRWL = 1 Cycle)



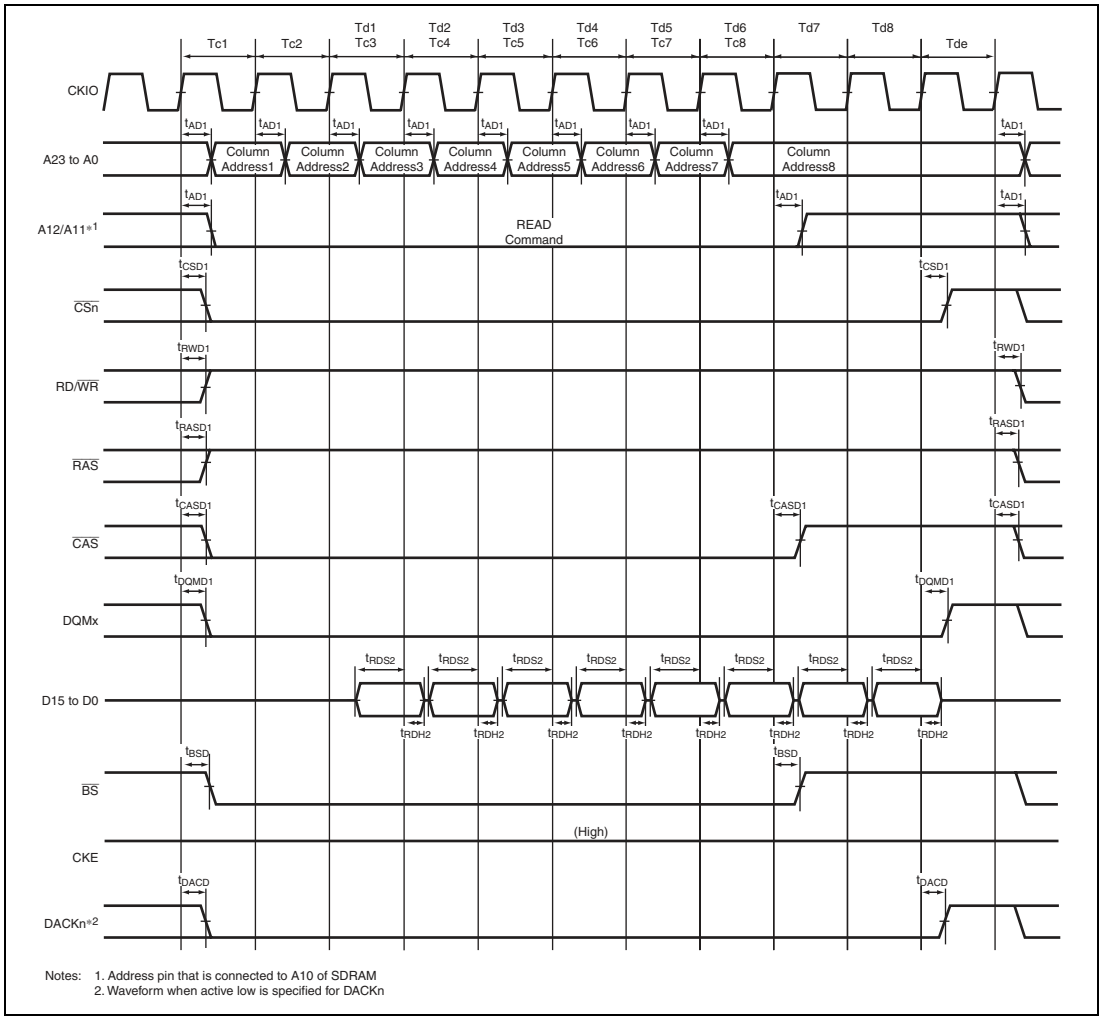
Notes: 1. Address pin that is connected to A10 of SDRAM
2. Waveform when active low is specified for DACKn

Figure 38.27 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 38.28 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 38.29 Burst Read Bus Cycle of SDRAM (Single Read × 8)
(Bank Active Mode: READ Command,
Same Row Address, CAS Latency 2, TRCD = 1 Cycle)

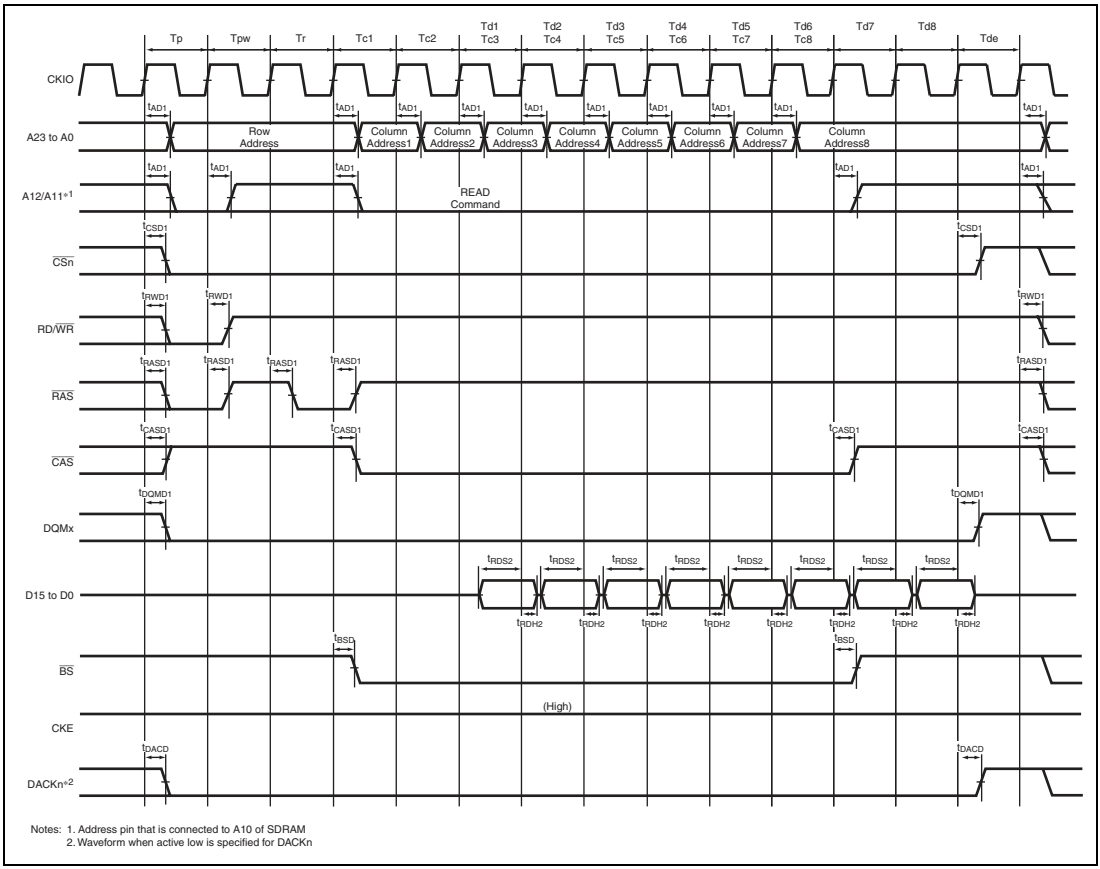
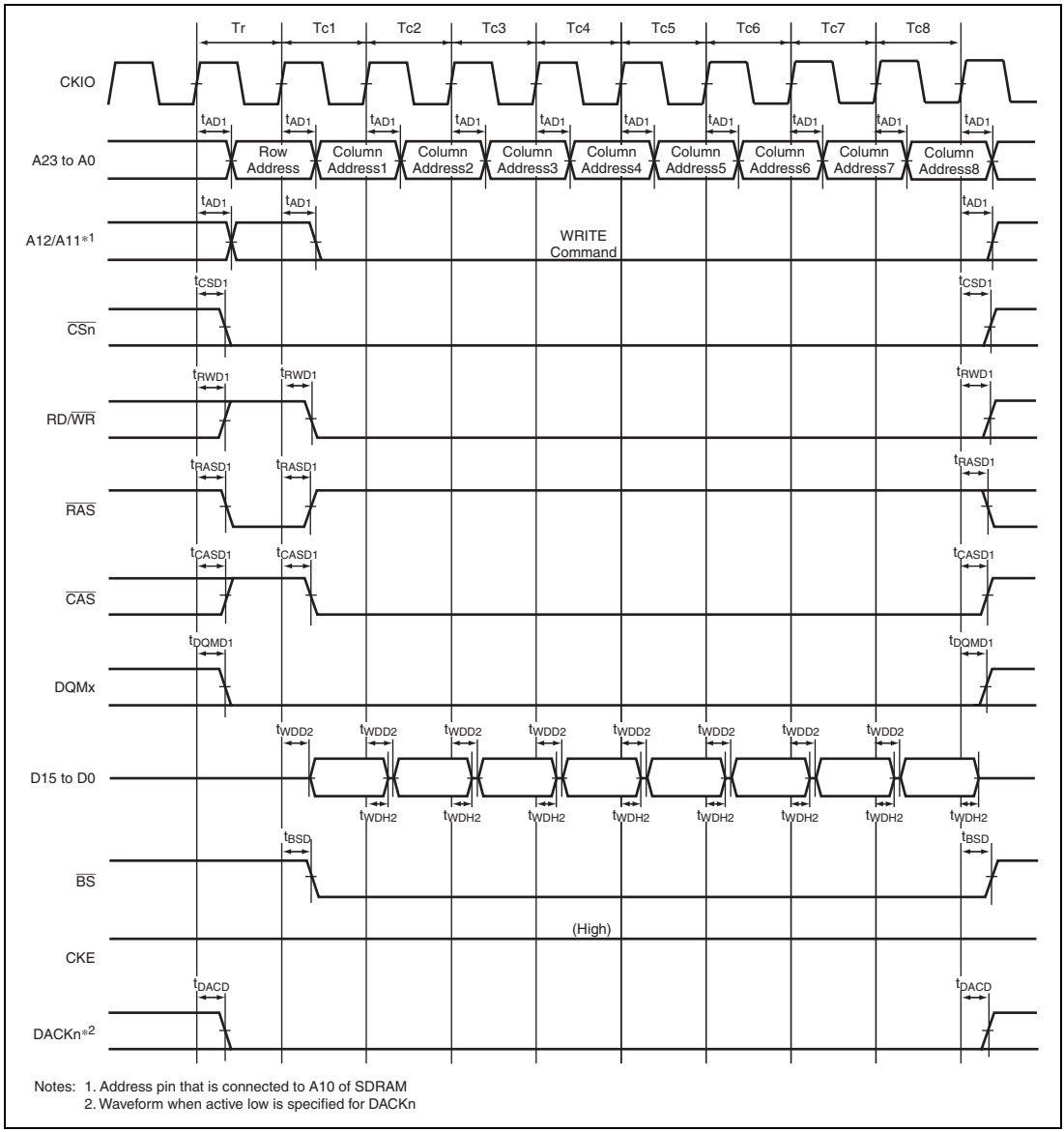
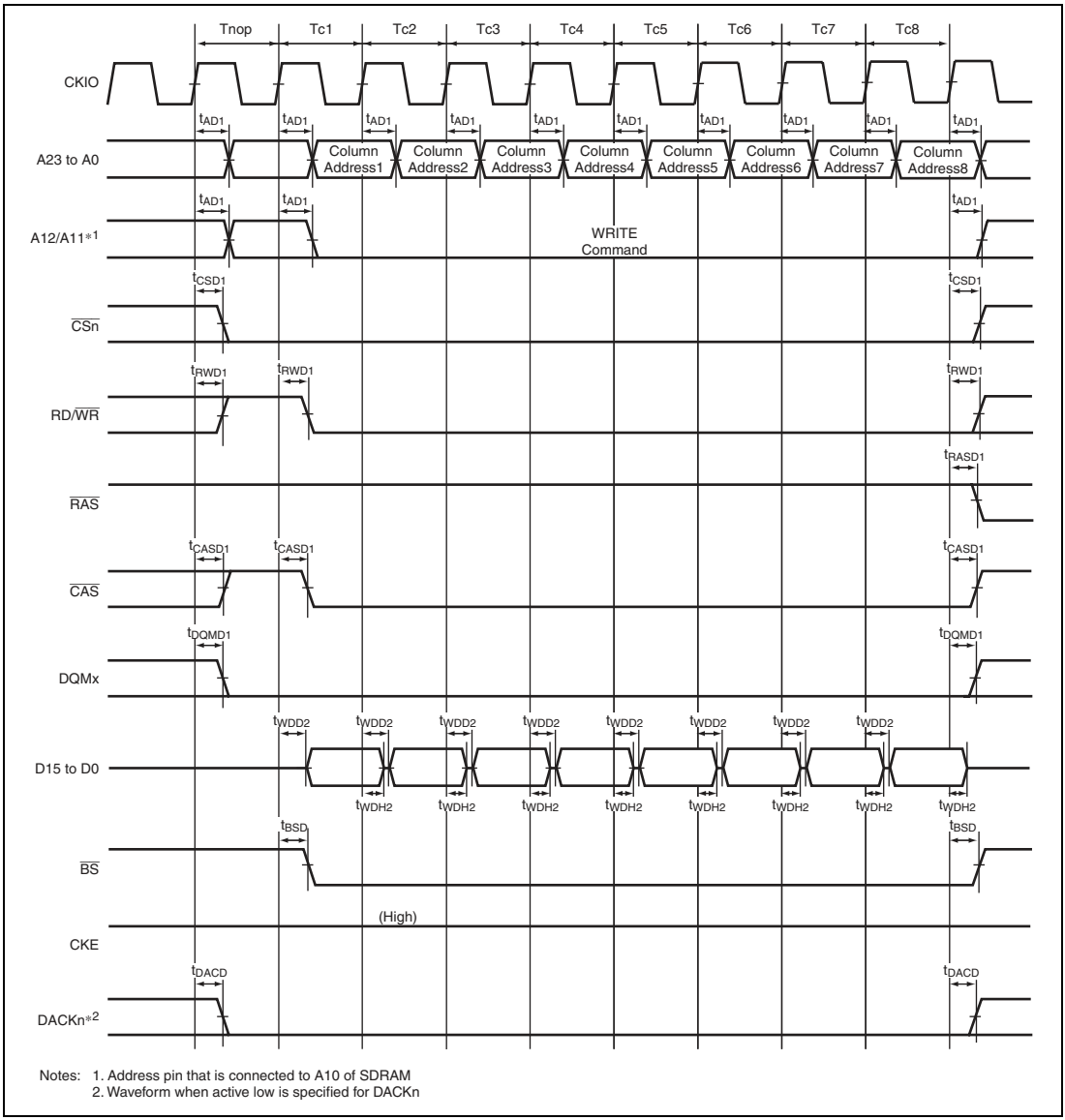


Figure 38.30 Burst Read Bus Cycle of SDRAM (Single Read × 8)
(Bank Active Mode: PRE + ACTV + READ Command,
Different Row Address, CAS Latency 2, TRCD = 1 Cycle)



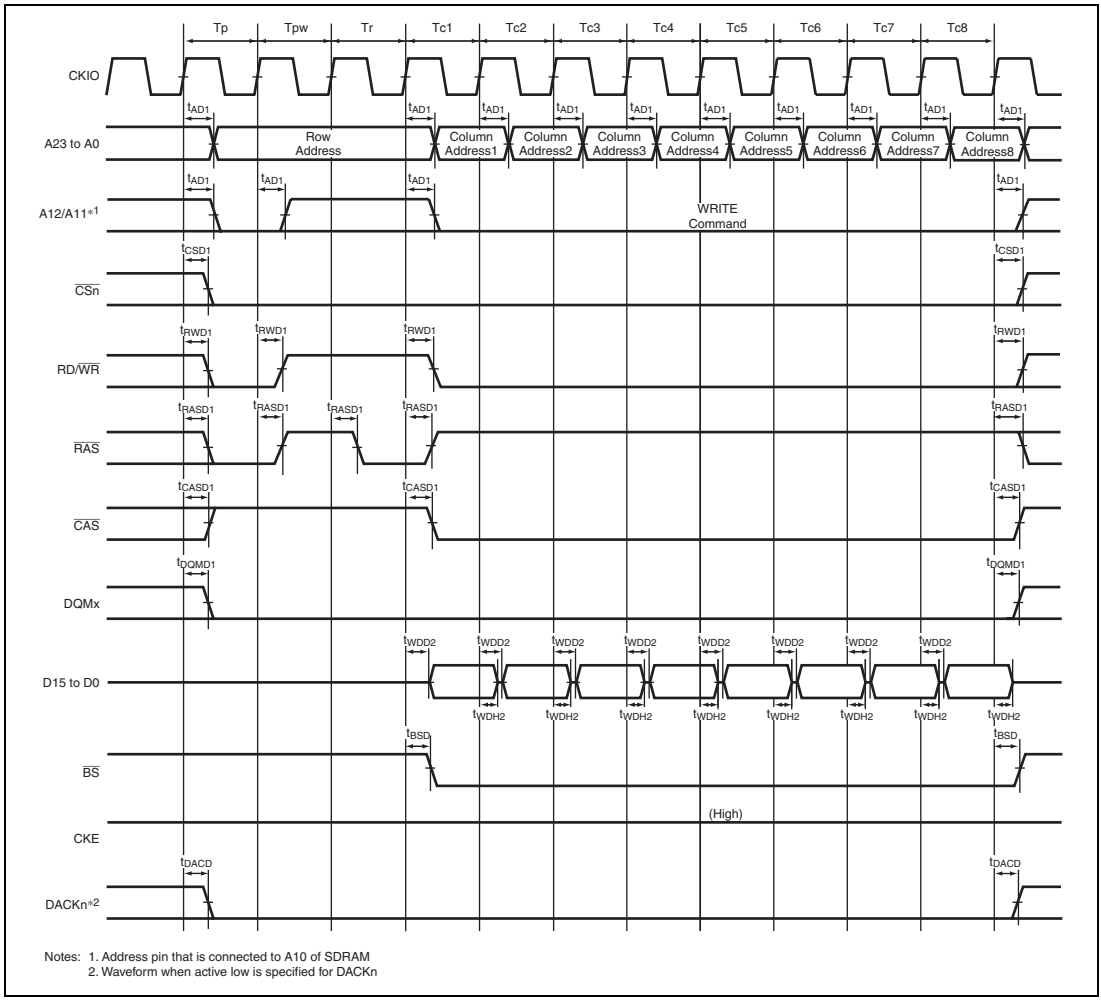
Notes: 1. Address pin that is connected to A10 of SDRAM
2. Waveform when active low is specified for DACKn

Figure 38.31 Burst Write Bus Cycle of SDRAM (Single Write × 8)
(Bank Active Mode: ACTV + WRIT Command, TRCD = 1 Cycle)



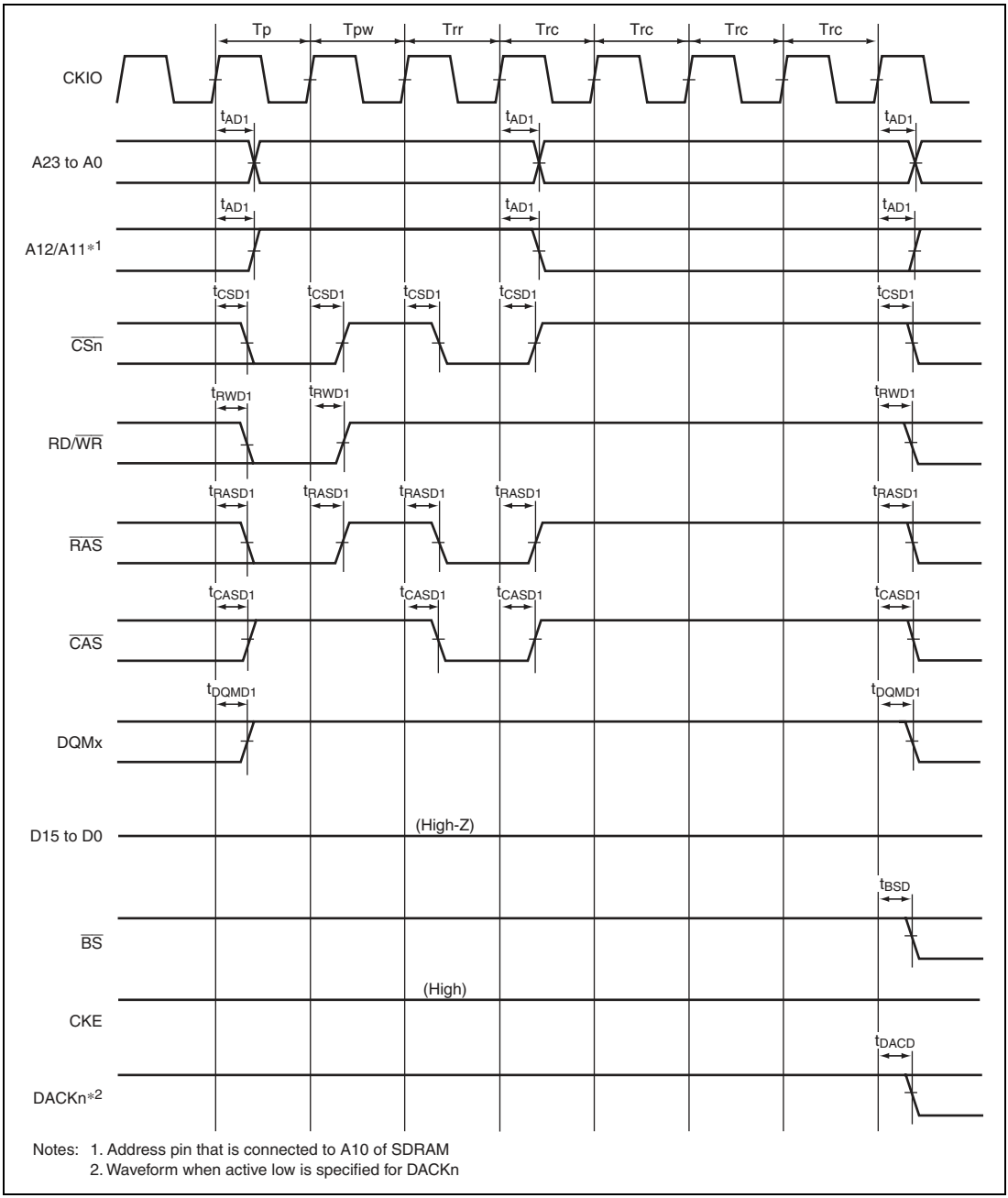
Notes: 1. Address pin that is connected to A10 of SDRAM
2. Waveform when active low is specified for DACKn

Figure 38.32 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 38.33 Burst Write Bus Cycle of SDRAM (Single Write × 8)
(Bank Active Mode: PRE + 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 38.34 Auto Refresh Timing of SDRAM (TRP = 2 Cycles)

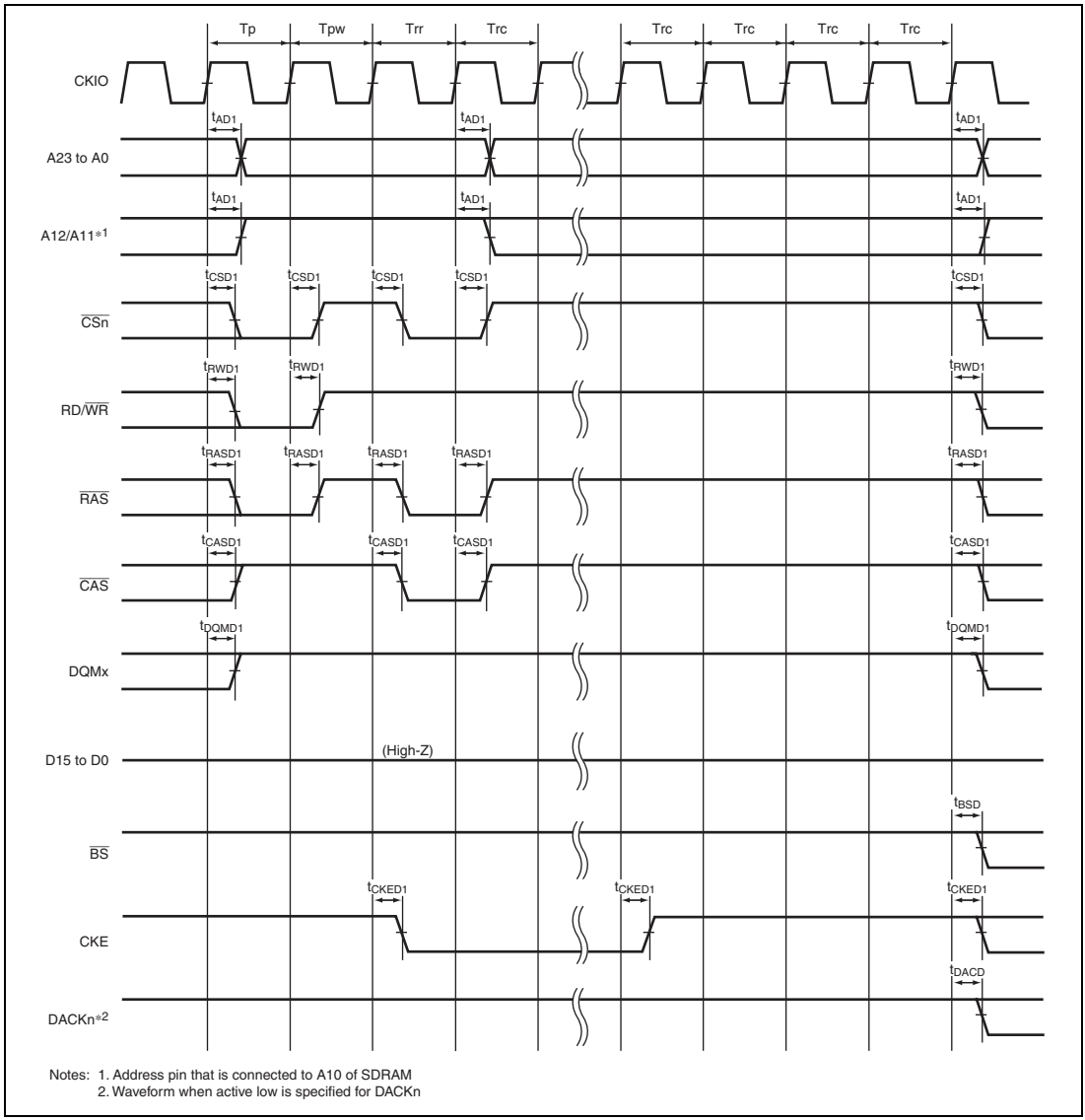


Figure 38.35 Self Refresh Timing of SDRAM (TRP = 2 Cycles)

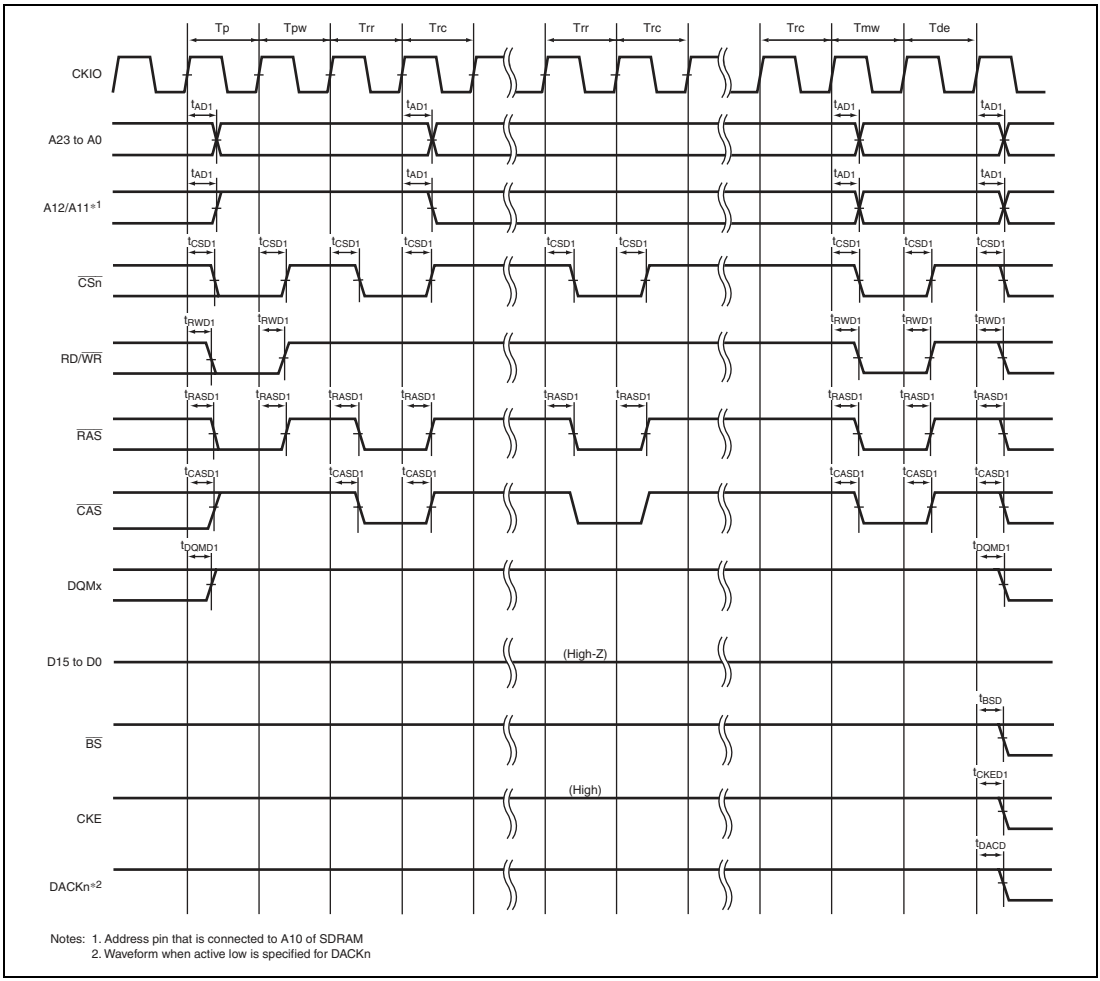
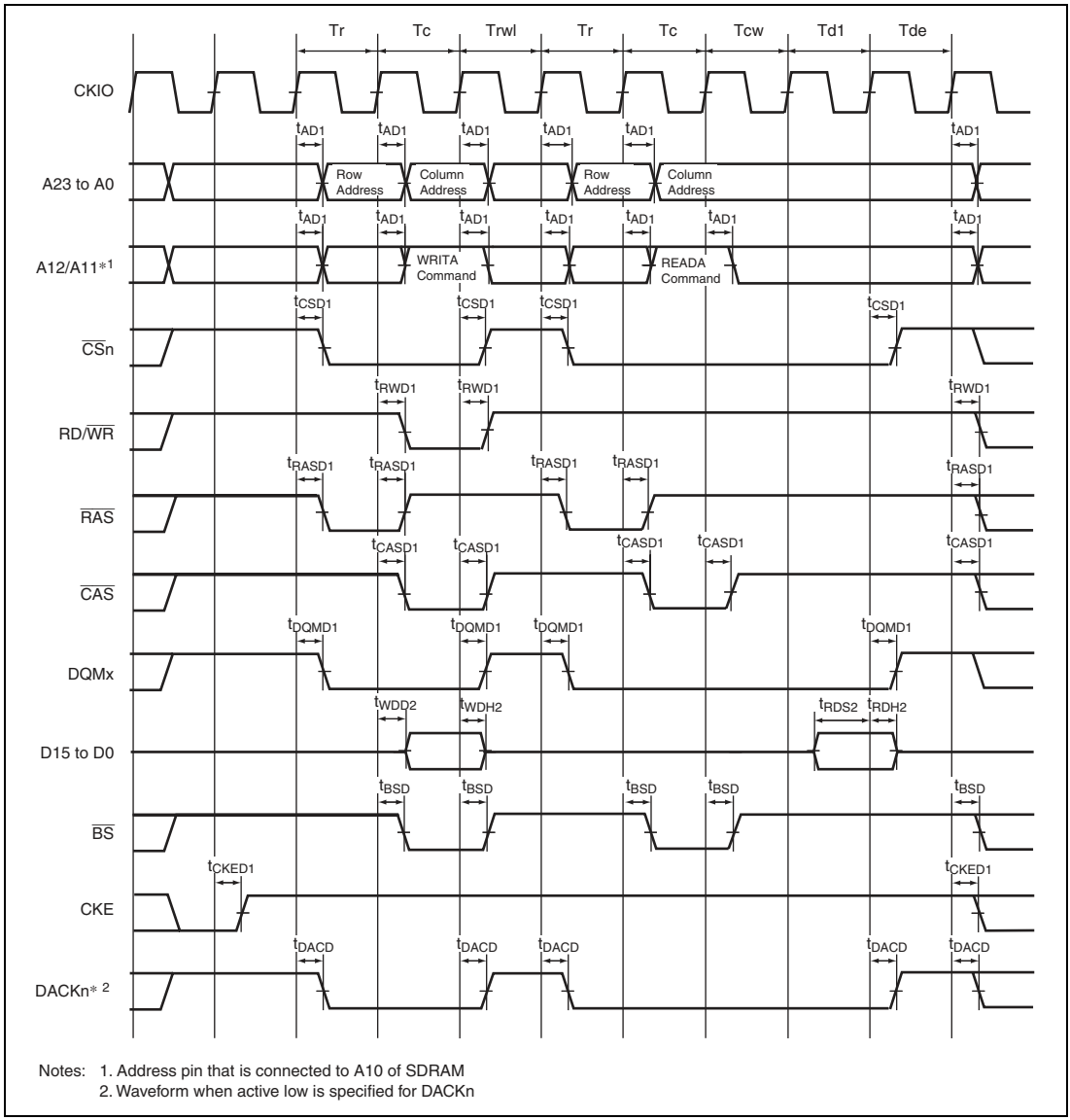
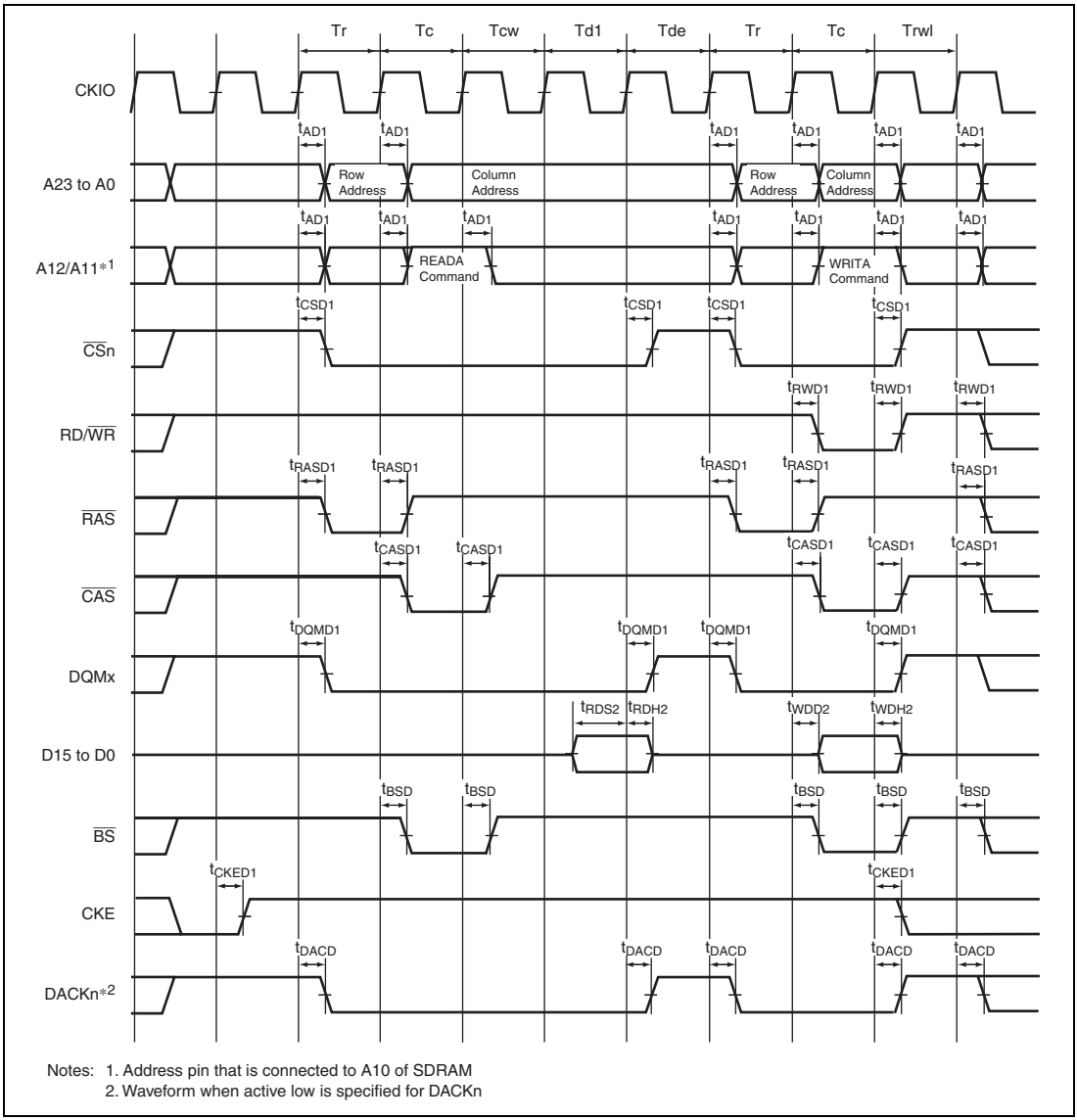


Figure 38.36 Power-On Sequence of SDRAM (Mode Write Timing, TRP = 2 Cycles)



Notes: 1. Address pin that is connected to A10 of SDRAM
2. Waveform when active low is specified for DACKn

Figure 38.37 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 38.38 Read to Write Bus Cycle in Power-Down Mode of SDRAM (Auto Precharge Mode, TRCD = 1 Cycle, TRP = 1 Cycle, TRWL = 1 Cycle)

38.4.7 PCMCIA Timing

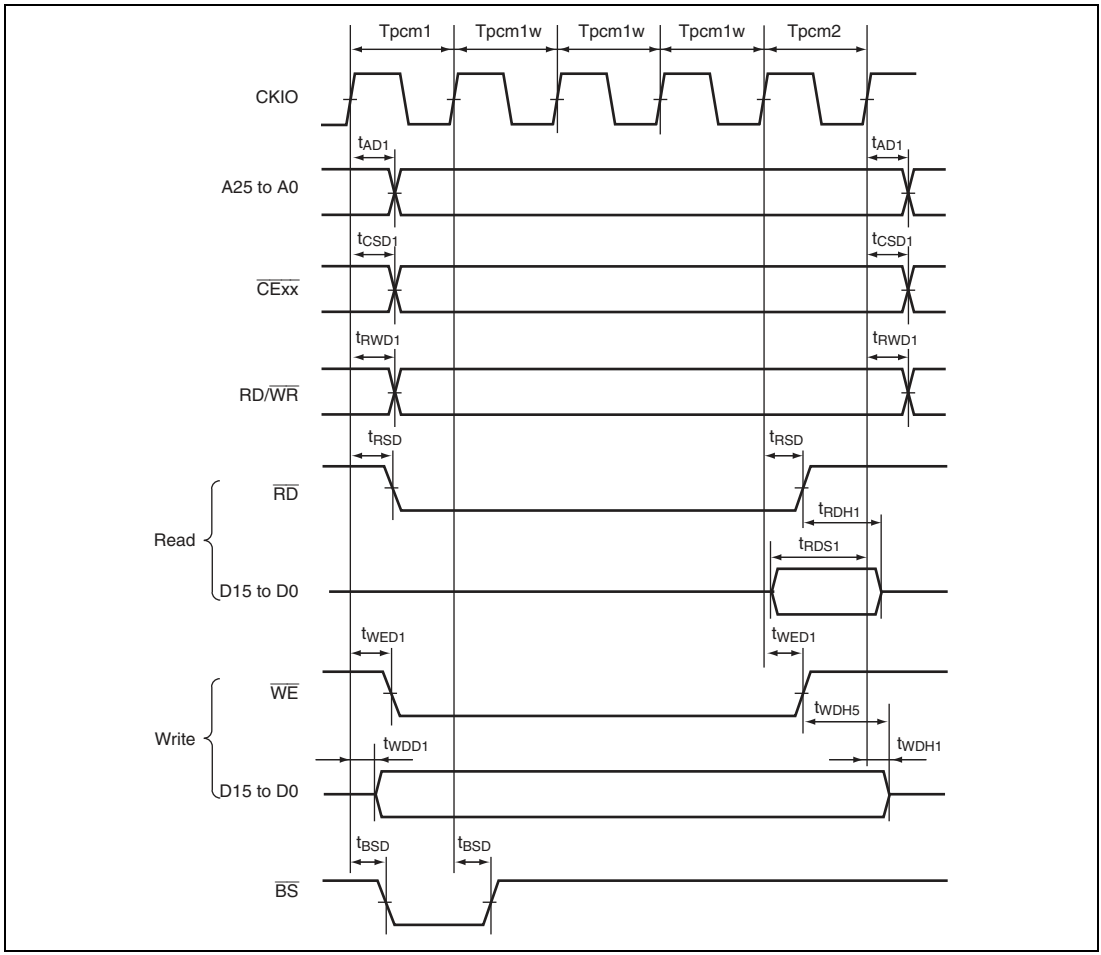


Figure 38.39 PCMCIA Memory Card Interface Bus Timing

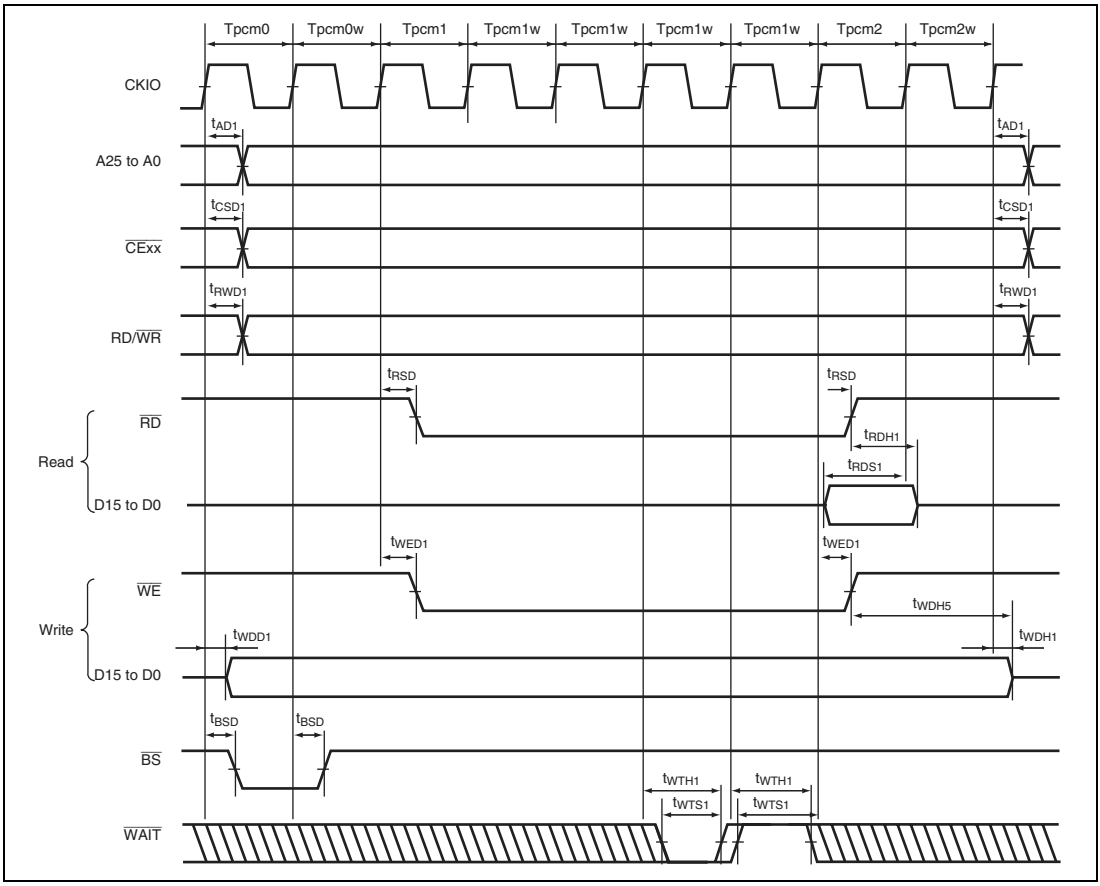


Figure 38.40 PCMCIA Memory Card Interface Bus Timing (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait 1, Hardware Wait 1)

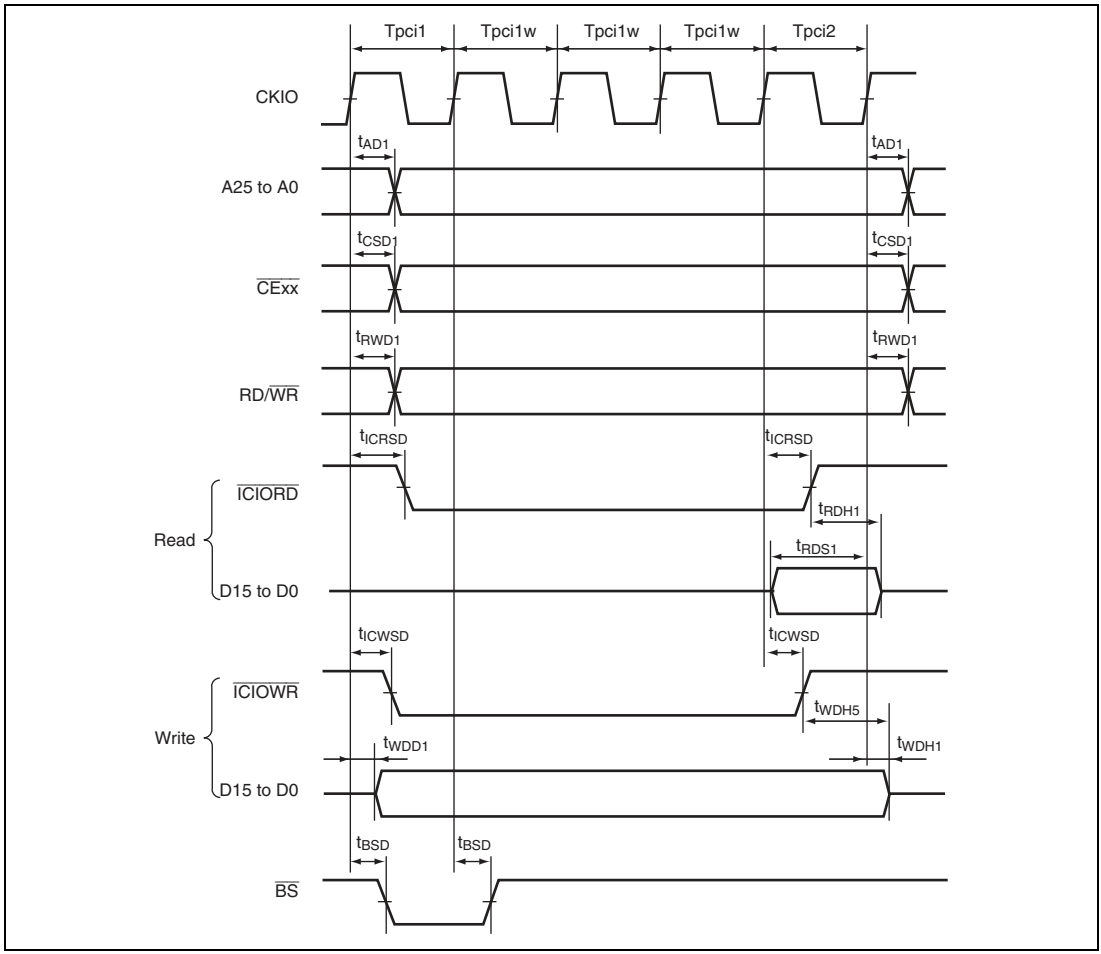


Figure 38.41 PCMCIA I/O Card Interface Bus Timing

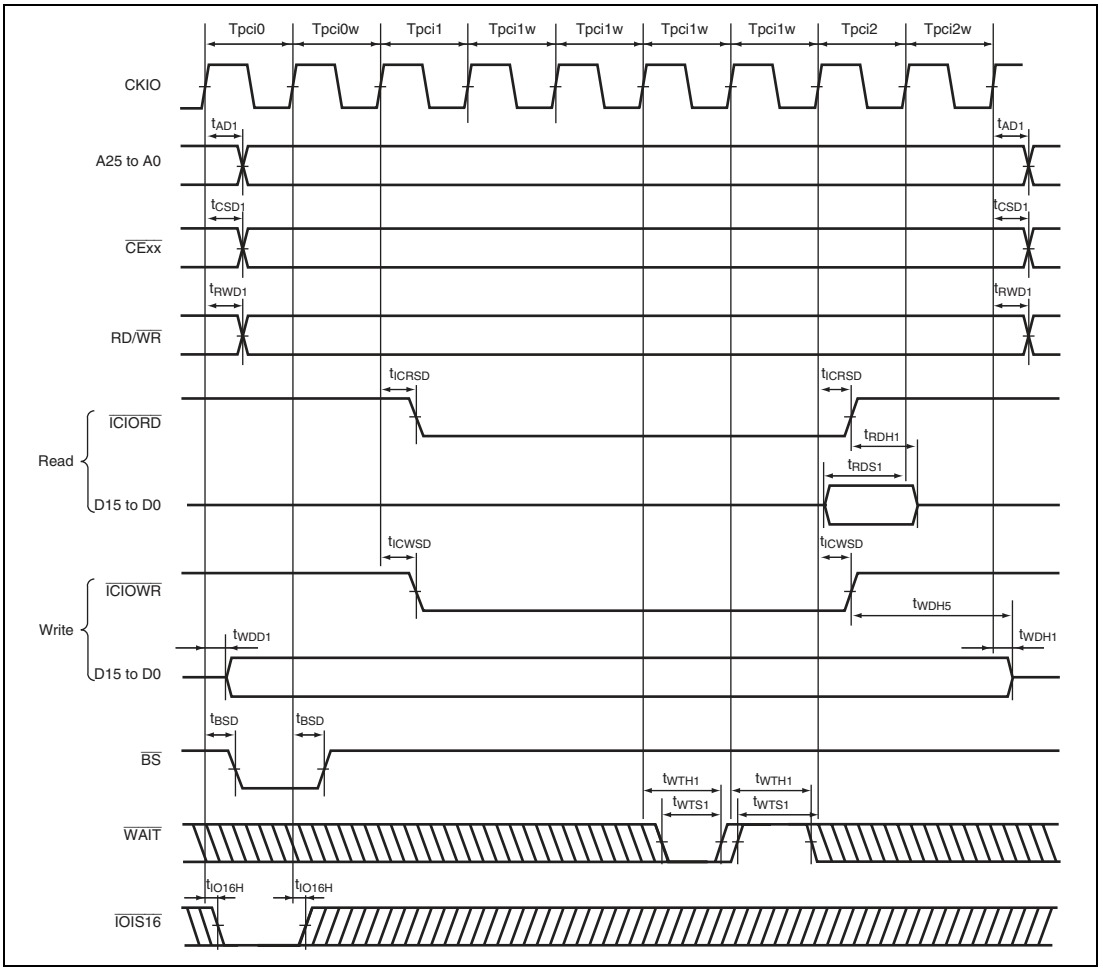


Figure 38.42 PCMCIA I/O Card Interface Bus Timing (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait 1, Hardware Wait 1)

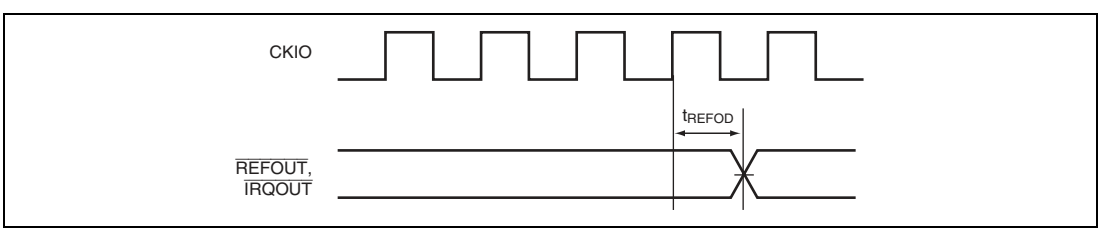


Figure 38.43 $\overline{\text{REFOUT}}$, $\overline{\text{IRQOUT}}$ Delay Time

38.4.8 Peripheral Module Signal Timing

Table 38.10 Peripheral Module Signal Timing

Conditions: $V_{cc}Q$ = 2.7 to 3.6 V, $V_{cc}Q1$ = 2.7 to 3.6 V or 1.65 to 1.95 V,
V_{cc} = 1.4 to 1.6 V, AV_{cc} = 3.0 to 3.6 V, Ta = –20 to 75°C

Module	Item	Symbol	Min.	Max.	Unit	Figure
I/O port	Output data delay time	t_{PORTD}	—	17	ns	38.44
	Input data setup time	t_{PORTS}	15	—		
	Input data hold time	t_{PORTH}	8	—		
DMAC	$\overline{\text{DREQ}}$n setup time	t_{DREQS}	6	—	ns	38.45
	$\overline{\text{DREQ}}$n hold time	t_{DREQH}	4	—		
	$\overline{\text{DACK}}$n, $\overline{\text{TEND}}$n delay time	t_{DACD}	—	13		38.46

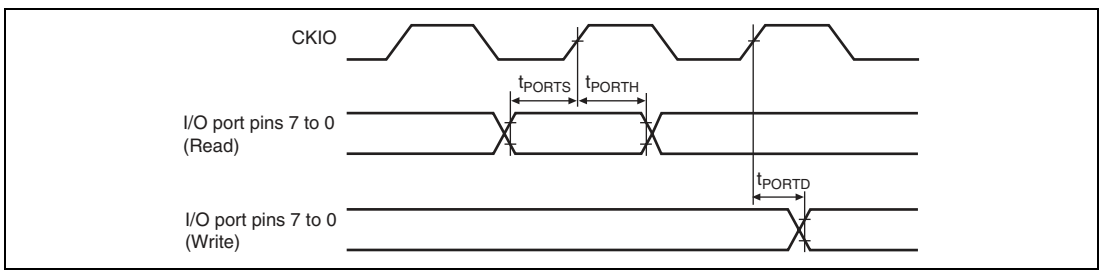


Figure 38.44 I/O Port Timing

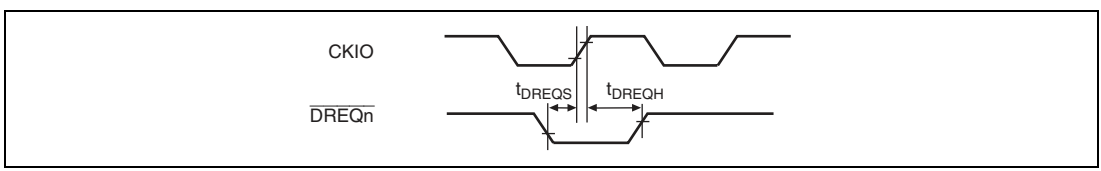


Figure 38.45 DREQ Input Timing (DREQ Low Level is Detected)

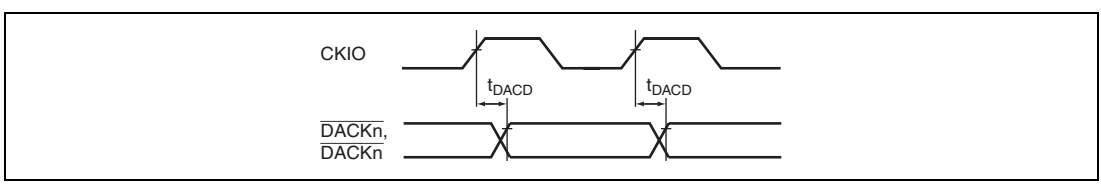


Figure 38.46 DACK Output Timing

38.4.9 16-Bit Timer Pulse Unit (TPU)

Table 38.11 16-Bit Timer Pulse Unit

Conditions: $V_{CC}Q$ = 2.7 to 3.6 V, $V_{CC}Q1$ = 2.7 to 3.6 V or 1.65 to 1.95 V,
V_{CC} = 1.4 to 1.6 V, AV_{CC} = 3.0 to 3.6 V, Ta = –20 to 75°C

Item		Symbol	Min.	Max.	Unit	Figure
Timer output delay time		t_{TOD}	—	15	ns	38.47
Timer clock input setup time		t_{TCKS}	15	—	ns	38.48
Timer clock pulse width	Single-edge setting	t_{TCKWH}, t_{TCKWL}	2	—	Pcyc*	38.48
	Both-edge setting	t_{TCKWH}, t_{TCKWL}	3	—		

Note: * Peripheral clock (Pϕ) cycle.

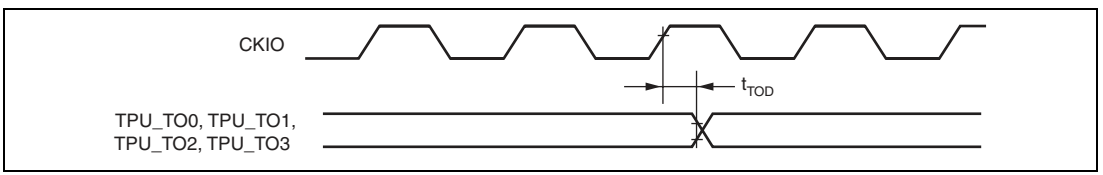


Figure 38.47 TPU Output Timing

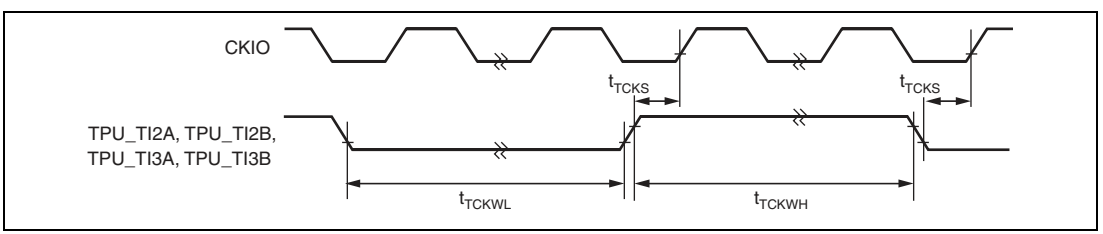


Figure 38.48 TPU Clock Input Timing

38.4.10 RTC Signal Timing

Table 38.12 RTC Signal Timing

Conditions: $V_{cc}Q$ = 2.7 to 3.6 V, $V_{cc}Q1$ = 2.7 to 3.6 V or 1.65 to 1.95 V,
V_{cc} = 1.4 to 1.6 V, AV_{cc} = 3.0 to 3.6 V, Ta = –20 to 75°C

Module	Item	Symbol	Min.	Max.	Unit	Figure
RTC	Oscillation settling time	t_{ROSC}	3	—	s	38.49

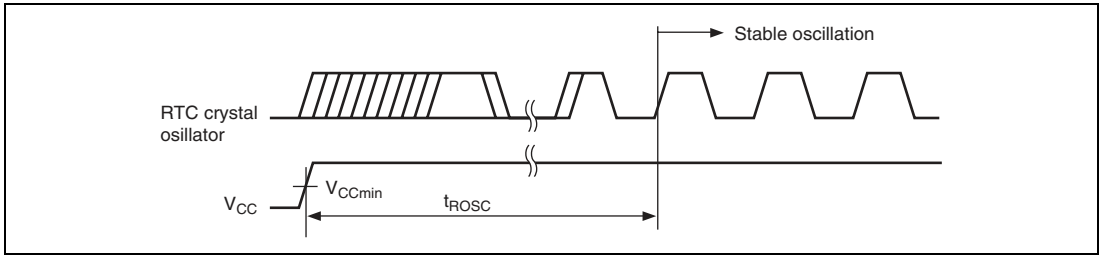


Figure 38.49 Oscillation Settling Time when RTC Crystal Oscillator is Turned On

38.4.11 SCIF Module Signal Timing

Table 38.13 SCIF Module Signal Timing

Conditions: $V_{CC}Q$ = 2.7 to 3.6 V, $V_{CC}Q1$ = 2.7 to 3.6 V or 1.65 to 1.95 V,
V_{CC} = 1.4 to 1.6 V, AV_{CC} = 3.0 to 3.6 V, Ta = –20 to 75°C

Module	Item		Symbol	Min.	Max.	Unit	Figure
SCIF	Input clock cycle	Asynchronous	t_{Scyc}	12	—	t_{Pcyc}	38.50
		Synchronous		4	—		38.51
	Input clock rise time		t_{SCKr}	—	1.5		38.50
	Input clock fall time		t_{SCKf}	—	1.5		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Transmit data delay time		t_{TXD}	—	3 t_{Pcyc} *+ 50	ns	38.51
	Receive data setup time (synchronous)		t_{RXS}	2 t_{Pcyc} *	—		
	Receive data hold time (synchronous)		t_{RXH}	2 t_{Pcyc} *	—		
	$\overline{RTS}$ delay time		t_{RTSD}	—	100		
	$\overline{CTS}$ setup time		t_{CTSS}	100	—		
	$\overline{CTS}$ hold time		t_{CTSH}	100	—		

Note: * t_{Pcyc} is a cycle time of a peripheral clock (Pϕ).

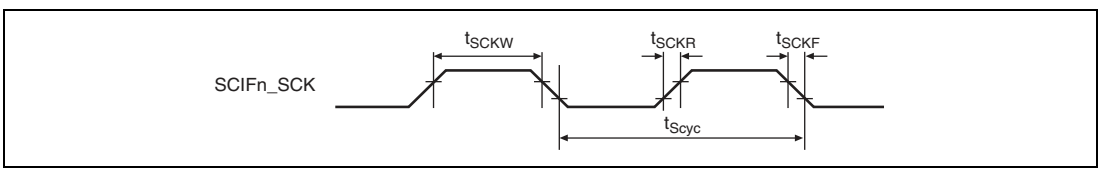


Figure 38.50 SCK Input Clock Timing

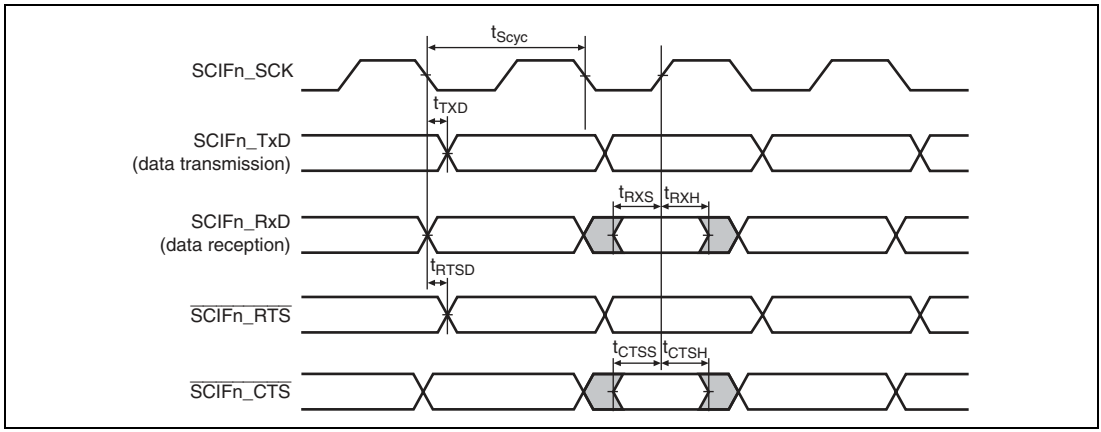


Figure 38.51 SCIF Input/Output Timing in Synchronous Mode

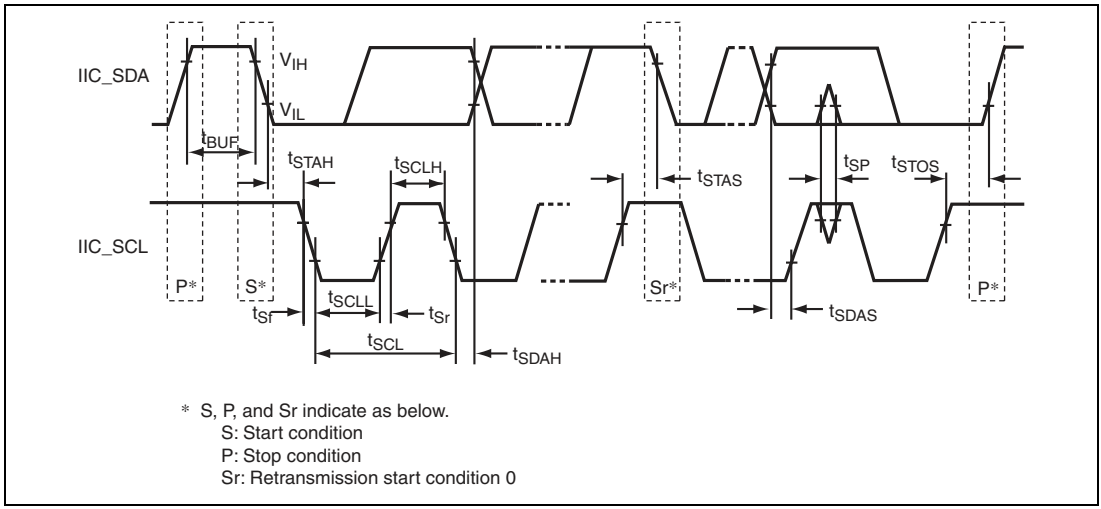
38.4.12 I²C Bus Interface Timing

Table 38.14 I²C Bus Interface Timing

Conditions: $V_{cc}Q$ = 2.7 to 3.6 V, $V_{cc}Q1$ = 2.7 to 3.6 V, V_{cc} = 1.4 to 1.6 V, Ta = –20 to 75°C

			Value				
Item	**Symbol**	**Test Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**	**Figure**
SCL input cycle time	t_{SCL}		12 t_{Pcyc} + 600	—	—	ns	38.52
SCL input high pulse width	t_{SCLH}		3 t_{Pcyc} + 300	—	—	ns	
SCL input low pulse width	t_{SCLL}		5 t_{Pcyc} + 300	—	—	ns	
SCL, SDA input fall time	t_{Sf}		—	—	300	ns	
SCL, SDA input spike pulse removal time	t_{SP}		—	—	1 t_{cyc}	ns	
SDA input bus free time	t_{BUF}		5 t_{Pcyc}	—	—	ns	
Start condition input hold time	t_{STAH}		3 t_{Pcyc}	—	—	ns	
Retransmission start condition input setup time	t_{STAS}		3 t_{Pcyc}	—	—	ns	
Stop condition input setup time	t_{STOS}		3 t_{Pcyc}	—	—	ns	
Data input setup time	t_{SDAS}		1 t_{Pcyc} + 20	—	—	ns	
Data input hold time	t_{SDAH}		0	—	—	ns	
Capacitive load of SCL, SDA	Cb		0	—	400	pF	
SCL, SDA output fall time	t_{Sf}	VccQ = 3.0 V	—	—	250	ns	
			—	—	300	ns	

Note: * t_{Pcyc} is a cycle time of a peripheral clock (Pϕ).



* S, P, and Sr indicate as below.
S: Start condition
P: Stop condition
Sr: Retransmission start condition 0

Figure 38.52 I²C Bus Interface Input/Output Timing

38.4.13 SIOF Module Signal Timing

Table 38.15 SIOF Module Signal Timing

Conditions: $V_{CC}Q$ = 2.7 to 3.6 V, $V_{CC}Q1$ = 2.7 to 3.6 V or 1.65 to 1.95 V,
V_{CC} = 1.4 to 1.6 V, AV_{CC} = 3.0 to 3.6 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
SIOF_MCLK clock input cycle time	t_{MCYC}	t_{Pcyc} *1	—	ns	38.53
SIOF_MCLK input high level width	t_{MWH}	0.4 × t_{MCYC}	—	ns	38.53
SIOF_MCLK input low level width	t_{MWL}	0.4 × t_{MCYC}	—	ns	38.53
SIOF_SCK clock cycle time	t_{SICYC}	t_{Pcyc} *1	—	ns	38.54 to 38.58
SIOF_SCK output high level width	t_{SWHO}	0.4 × t_{SICYC}	—	ns	38.54 to 38.57
SIOF_SCK output low level width	t_{SWLO}	0.4 × t_{SICYC}	—	ns	38.54 to 38.57
SIOF_SYNC output delay time	t_{FSD}	—	20	ns	38.54 to 38.57
SIOF_SCK input high level width	t_{SWHI}	0.4 × t_{SICYC}	—	ns	38.58
SIOF_SCK input low level width	t_{SWLI}	0.4 × t_{SICYC}	—	ns	38.58
SIOF_SYNC input setup time	t_{FSS}	20	—	ns	38.58
SIOF_SYNC input hold time	t_{FSH}	20	—	ns	38.58
SIOF_TXD output delay time	t_{STDD}	—	20	ns	38.54 to 38.58
SIOF_RXD input setup time	t_{SRDS}	20	—	ns	38.54 to 38.58
SIOF_RXD input hold time	t_{SRDH}	20	—	ns	38.54 to 38.58

Note: t_{Pcyc} is a cycle time of a peripheral clock (Pϕ).

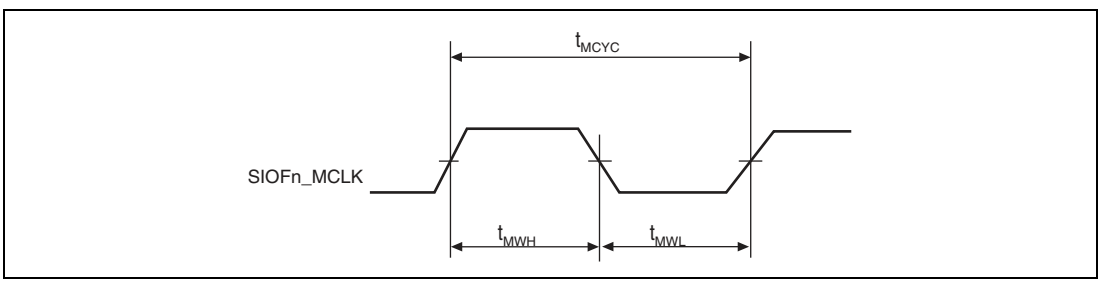


Figure 38.53 SIOF_MCLK Input Timing

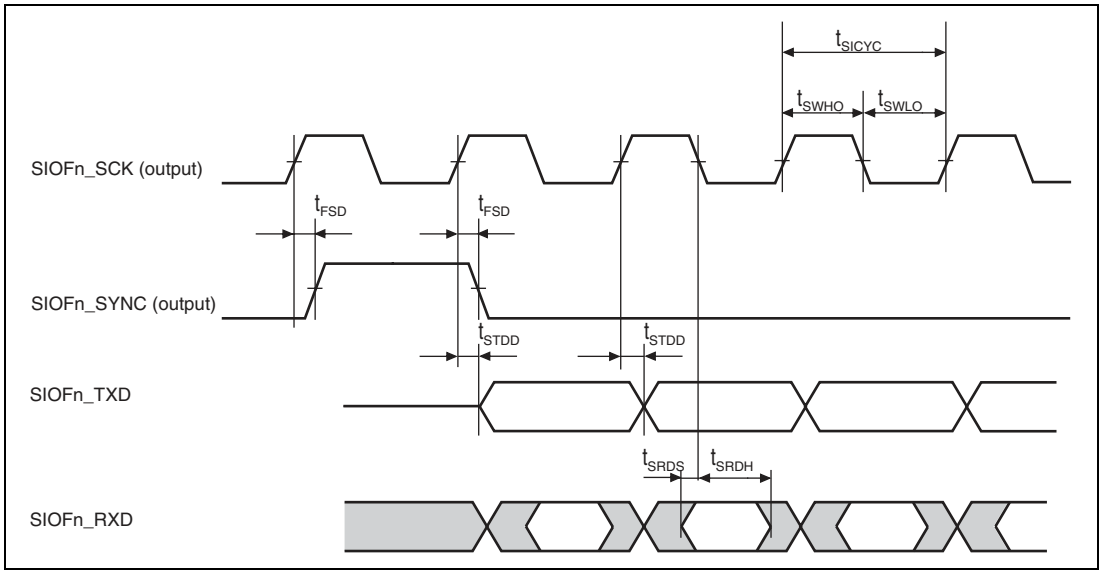


Figure 38.54 SIOF Transmission/Reception Timing (Master Mode 1, Fall Sampling)

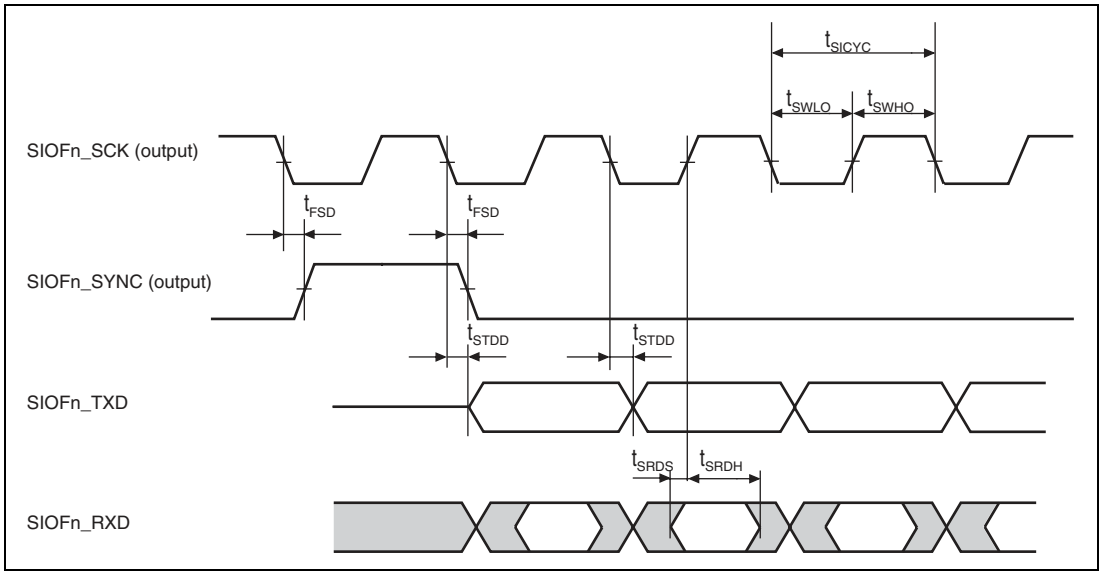


Figure 38.55 SIOF Transmission/Reception Timing (Master Mode 1, Rise Sampling)

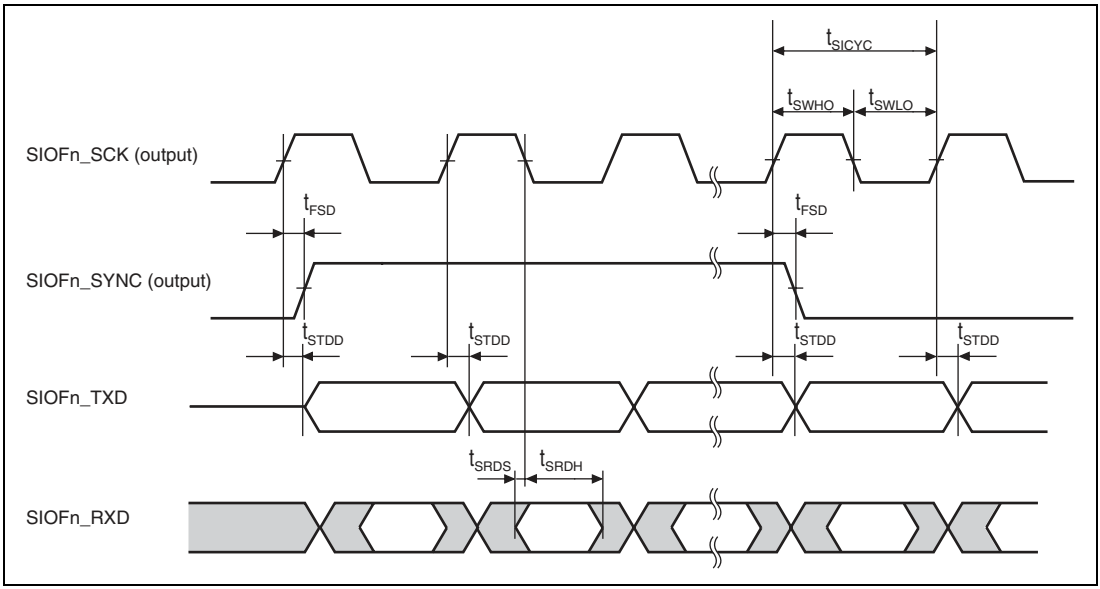


Figure 38.56 SIOF Transmission/Reception Timing (Master Mode 2, Fall Sampling)

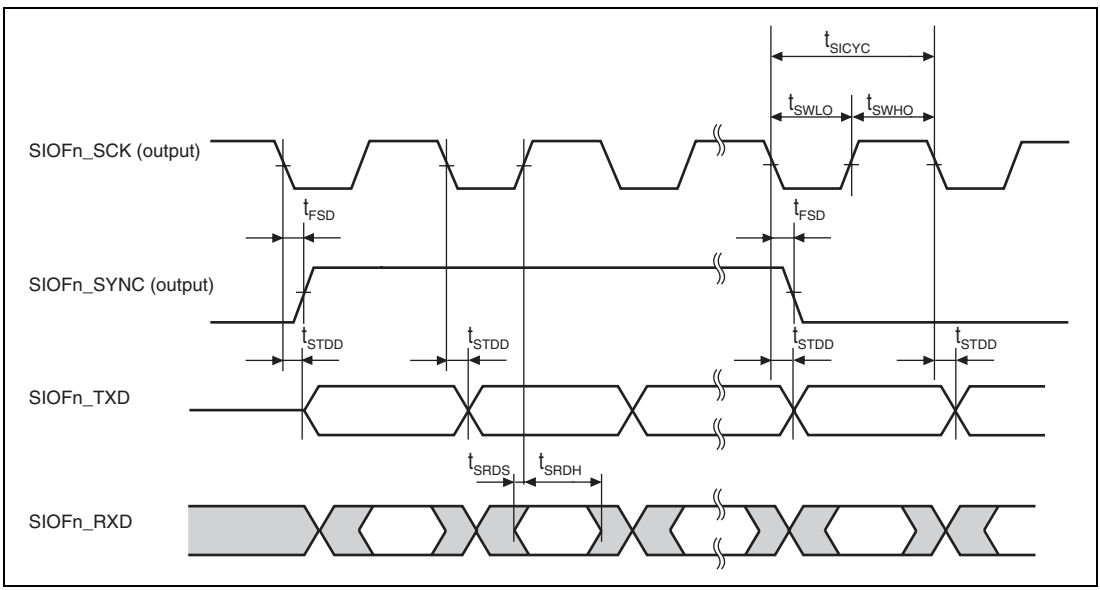


Figure 38.57 SIOF Transmission/Reception Timing (Master Mode 2, Rise Sampling)

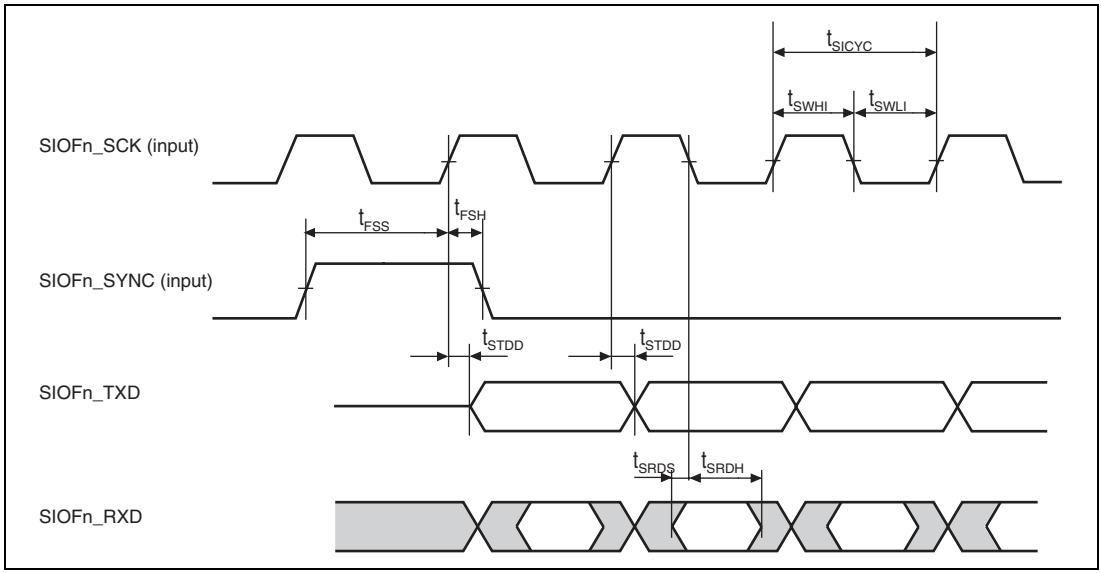


Figure 38.58 SIOF Transmission/Reception Timing (Slave Mode 1, Slave Mode 2)

38.4.14 AFEIF Module Signal Timing

Table 38.16 AFEIF Module Signal Timing

Conditions: $V_{CC}Q$ = 2.7 to 3.6 V, $V_{CC}Q1$ = 2.7 to 3.6 V or 1.65 to 1.95 V,
V_{CC} = 1.4 to 1.6 V, AV_{CC} = 3.0 to 3.6 V, Ta = –20 to 75°C

Item	Symbol	min	max	Unit
AFE_SCLK clock input cycle time	t_{ASCYC}	$8 \times t_{Pcyc}$	—	ns
AFE_SCLK input high level width	t_{ASWH}	$0.4 \times t_{ASCYC}$	—	ns
AFE_SCLK input low level width	t_{ASWL}	$0.4 \times t_{ASCYC}$	—	ns
AFE_FS input time	t_{AFSD}	0	50	ns
AFE_TXOUT output delay time	t_{ATDD}	—	t_{Pcyc} + 20	ns
AFE_RXIN input setup time	t_{ARDS}	20	—	ns
AFE_RXIN input hold time	t_{ARDH}	$2 \times t_{Pcyc}$ + 20	—	ns
AFE_HC1 output delay time	t_{AHCD}	—	$3 \times t_{Pcyc}$ + 20	ns
AFE_RLYC output delay time	t_{ARLYD}	—	t_{Pcyc} + 20	ns

Note: t_{Pcyc} is a cycle time (ns) of a peripheral clock (Pϕ).

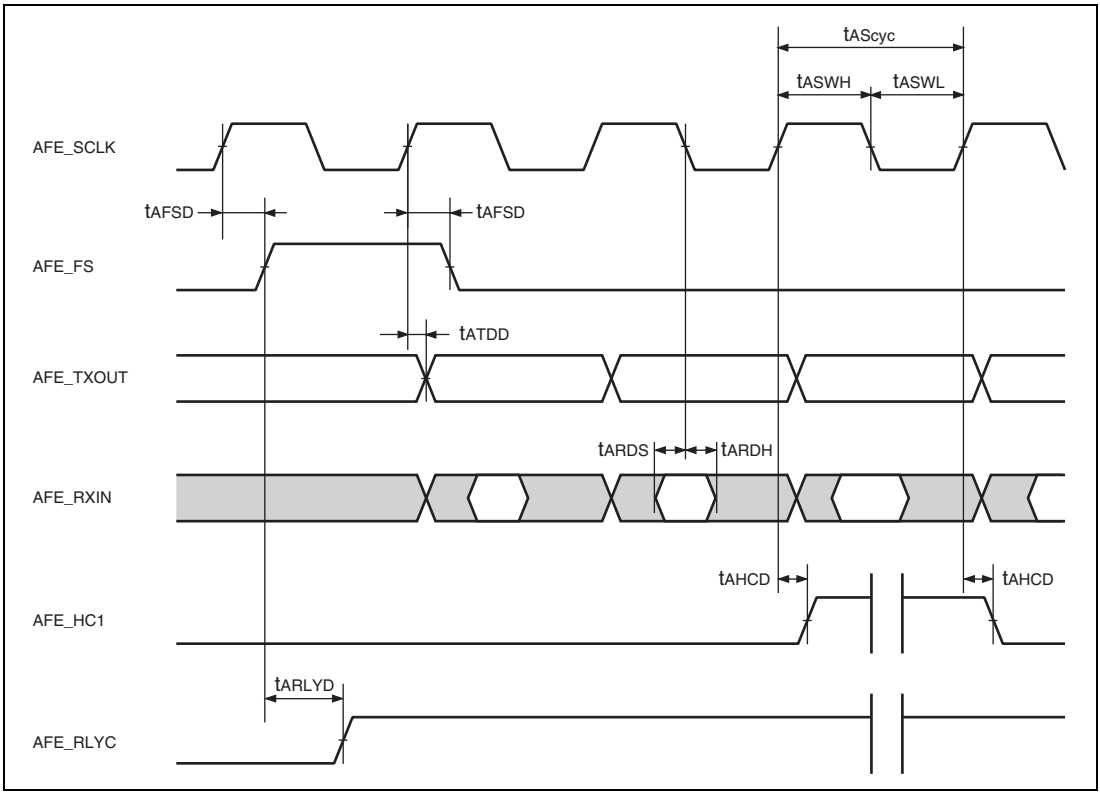


Figure 38.59 AFEIF Module AC Timing

38.4.15 USB Module Signal Timing

Table 38.17 USB Module Clock Timing

Conditions: $V_{CC}Q$ = 2.7 to 3.6 V, $V_{CC}Q1$ = 2.7 to 3.6 V or 1.65 to 1.95 V,
V_{CC} = 1.4 to 1.6 V, AV_{CC}_USB= 3.0 to 3.6 V, Ta = –20 to 75°C

Item		Symbol	Min.	Max.	Unit	Figure
EXTAL_USB clock frequency (48 MHz)	(USBF)	t_{FREQ}	47.9	48.1	MHz	38.60
	(USBH)		47.976	48.024		
Clock rise time		t_{R48}	—	6	ns	
Clock fall time		t_{F48}	—	6	ns	

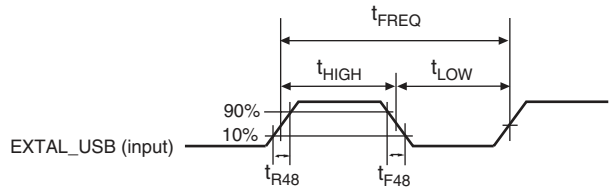


Figure 38.60 USB Clock Timing

Table 38.18 USB Electrical Characteristics (Full-Speed)

Item	Symbol	Min.	Max.	Unit	Figure
Transition time (rise)*[2]	t_R	4	20	ns	CL = 50 pF
Transition time (fall)*[2]	t_F	4	20	ns	CL = 50 pF
Rise/fall time matching	t_{RFM}	85	111	%	(TR/TF)
Output signal crossover power supply voltage	V_{CRS}	1.3	2.0	V	—

Notes: Measured with edge control C_{EDGE} = 47 pF and connection of direct resister Rs = 27 Ω.
1. Value when CL = 50 pF unless specified.
2. Value within 10% to 90% of the signal power supply voltage.

Table 38.19 USB Electrical Characteristics (Low-Speed)

Item	Symbol	Min.	Max.	Unit	Figure
Transition time (rise)*	t_R	75	—	ns	CL = 200 pF
		—	300	ns	CL = 600 pF
Transition time (fall)*	t_F	75	—	ns	CL = 200 pF
		—	300	ns	CL = 600 pF
Rise/fall time matching	t_{RFM}	80	125	%	(TR/TF)
Output signal crossover power supply voltage	V_{CRS}	1.3	2.0	V	—

Notes: Measured with edge control C_{EDGE} = 47 pF and connection of direct resister Rs = 27 Ω.
* Value within 10% to 90% of the signal power supply voltage.

38.4.16 LCDC Module Signal Timing

Table 38.20 LCDC Module Signal Timing

Conditions: $V_{CC}Q$ = 2.7 to 3.6 V, $V_{CC}Q1$ = 2.7 to 3.6 V or 1.65 to 1.95 V,
V_{CC} = 1.4 to 1.6 V, AV_{CC} = 3.0 to 3.6 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
LCD_CLK input clock frequency	t_{FREQ}	—	66	MHz	
LCD_CLK input clock rise time	t_r	—	3	ns	
LCD_CLK input clock fall time	t_f	—	3	ns	
LCD_CLK input clock duty	t_{DUTY}	90	110	%	
Clock (LCD_CL2) cycle time	t_{CC}	25	—	ns	38.61
Clock (LCD_CL2) high level pulse width	t_{CHW}	7	—	ns	
Clock (LCD_CL2) low level pulse width	t_{CLW}	7	—	ns	
Clock (LCD_CL2) transition time (rise/fall)	t_{CT}	—	3	ns	
Data (LCD_DATA) delay time	t_{DDdo}	–3.5	3	ns	
Display enable (LCD_M_DISP) delay time	t_{IDdo}	–3.5	3	ns	
Horizontal synchronous signal (LCD_CL1) delay time	t_{HDdo}	–3.5	3	ns	
Vertical synchronous signal (LCD_FLM) delay time	t_{VDdo}	–3.5	3	ns	

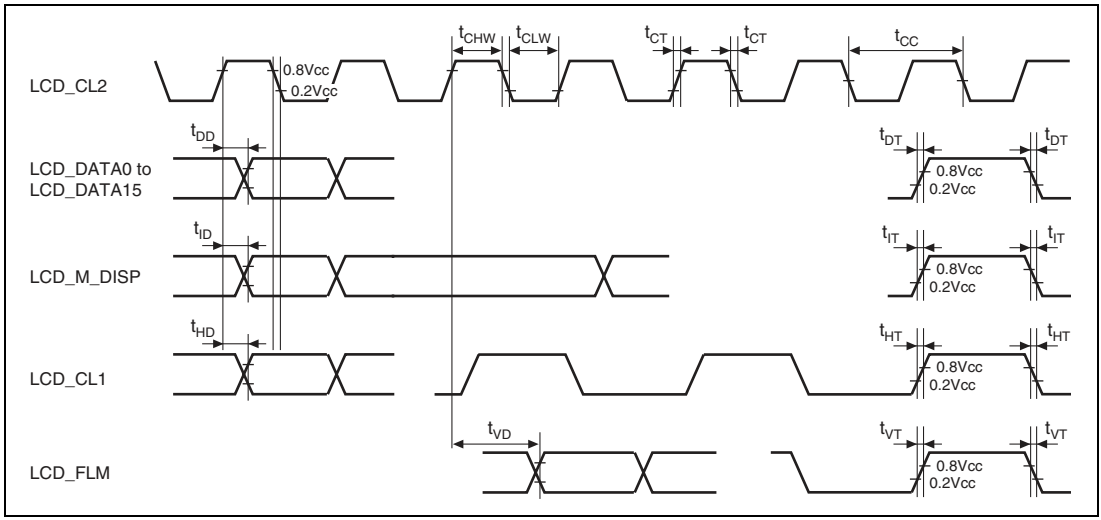


Figure 38.61 LCDC Module Signal Timing

38.4.17 SIM Module Signal Timing

Table 38.21 SIM Module Signal Timing

Conditions: $V_{cc}Q$ = 2.7 to 3.6 V, $V_{cc}Q1$ = 2.7 to 3.6 V or 1.65 to 1.95 V,
V_{cc} = 1.4 to 1.6 V, AV_{cc} = 3.0 to 3.6 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
SIM_CLK clock cycle	t_{SMCYC}	2 x tpcyc	16 x tpcyc	ns	38.62
SIM_CLK clock high level width	t_{SMCWH}	0.4 × t_{SMCYC}	—	ns	
SIM_CLK clock low level width	t_{SMCWL}	0.4 × t_{SMCYC}	—	ns	
SIM_RST reset output delay	t_{SMRD}	0	20	ns	

Note: t_{Pcyc} is a cycle time of a peripheral clock (Pϕ).

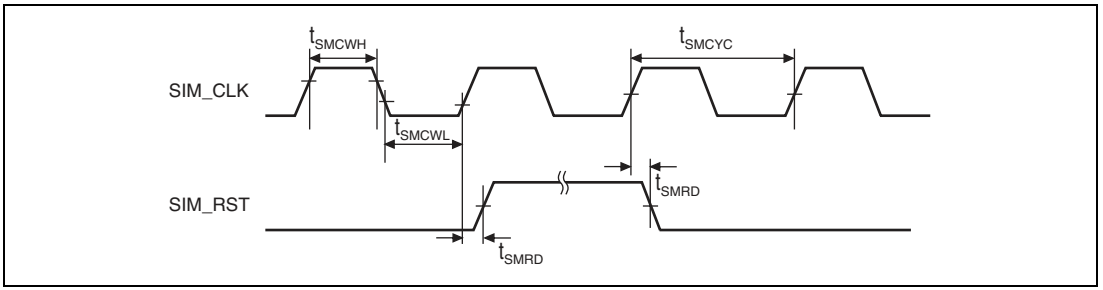


Figure 38.62 SIM Module Signal Timing

38.4.18 MMCIF Module Signal Timing

Table 38.22 MMCIF Module Signal Timing

Conditions: $V_{CC}Q$ = 2.7 to 3.6 V, $V_{CC}Q1$ = 2.7 to 3.6 V or 1.65 to 1.95 V,
V_{CC} = 1.4 to 1.6 V, AV_{CC} = 3.0 to 3.6 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
MMC_CLK clock cycle	t_{MMCYC}	60	—	ns	38.63, 38.64
MMC_CLK clock high level width	t_{MMWH}	0.4 × t_{MMCYC}	—	ns	
MMC_CLK clock low level width	t_{MMWL}	0.4 × t_{MMCYC}	—	ns	
MMC_CMD output data delay	t_{MMCD}	—	10	ns	
MMC_CMD input data hold	t_{MMRCS}	10	—	ns	
MMC_CMD input data setup	t_{MMRCH}	10	—	ns	
MMC_DAT output data delay	t_{MMTDD}	—	10	ns	
MMC_DAT input data setup	t_{MMRDS}	10	—	ns	
MMC_DAT input data hold	t_{MMRDH}	10	—	ns	

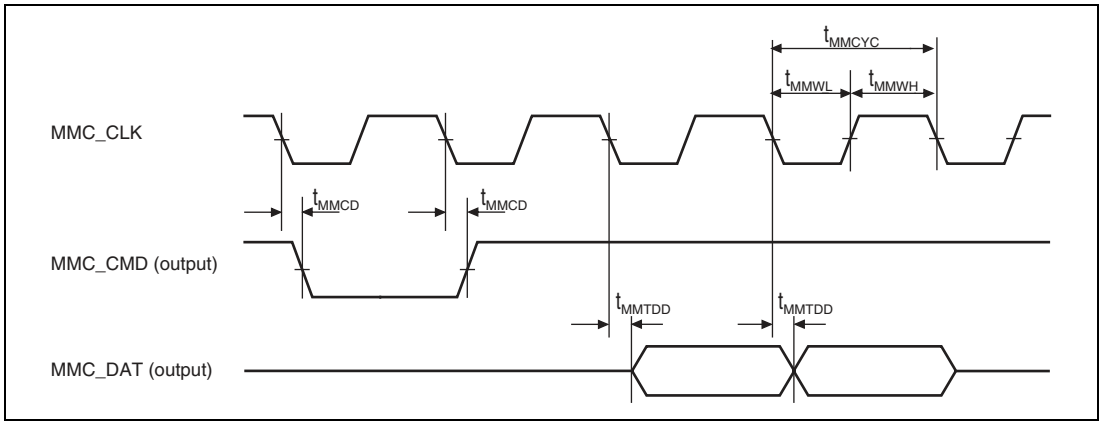


Figure 38.63 MMCIF Transmit Timing

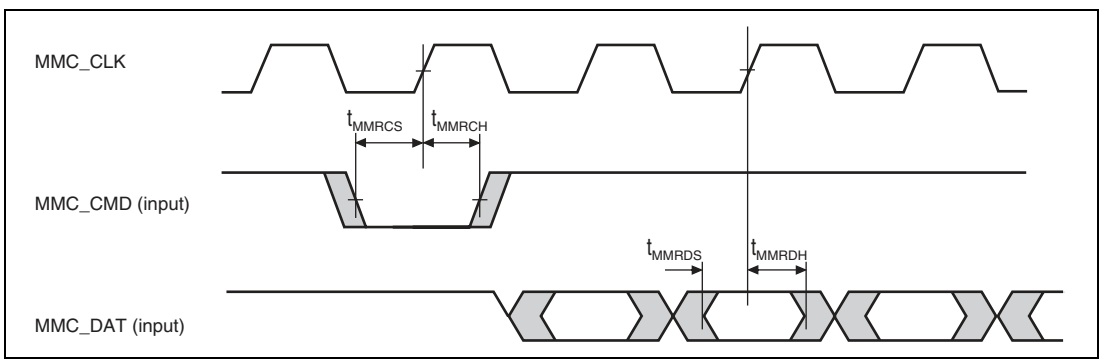


Figure 38.64 MMCIF Receive Timing (Rise Sampling)

38.4.19 H-UDI Related Pin Timing

Table 38.23 H-UDI Related Pin Timing

Conditions: $V_{CC}Q$ = VccQ_RTC = 2.7 to 3.6 V, VccQ1 = 2.7 to 3.6 V or 1.65 to 1.95 V,
V_{CC} = V_{CC}_PLL1 = V_{CC} _PLL2 = Vcc_RTC = 1.4 to 1.6 V,
AVcc = AVcc_USB = 3.0 to 3.6 V, T_a = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
TCK cycle time	t_{TCKcyc}	50	—	ns	38.65
TCK high level pulse width	t_{TCKH}	12	—	ns	
TCK low level pulse width	t_{TCKL}	12	—	ns	
TCK rise/fall time	t_{TCKf}	—	4	ns	
$\overline{\text{TRST}}$ setup time	t_{TRSTS}	12	—	ns	38.66
$\overline{\text{TRST}}$ hold time	t_{TRSTH}	50	—	t_{cyc}	
TDI setup time	t_{TDIS}	10	—	ns	38.67
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	
$\overline{\text{ASEMD0}}$ setup time	$t_{ASEMD0S}$	12	—	ns	38.68
$\overline{\text{ASEMD0}}$ hold time	$t_{ASEMD0H}$	12	—	ns	

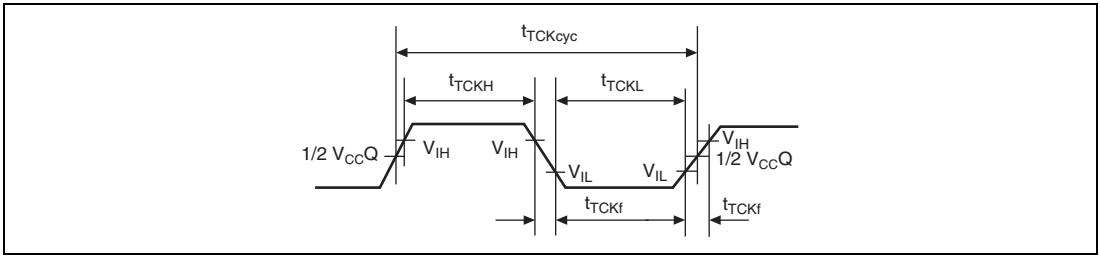


Figure 38.65 TCK Input Timing

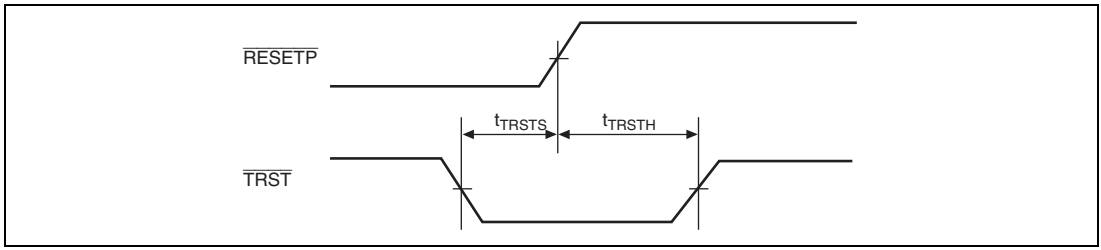


Figure 38.66 $\overline{\text{TRST}}$ Input Timing (Reset Hold)

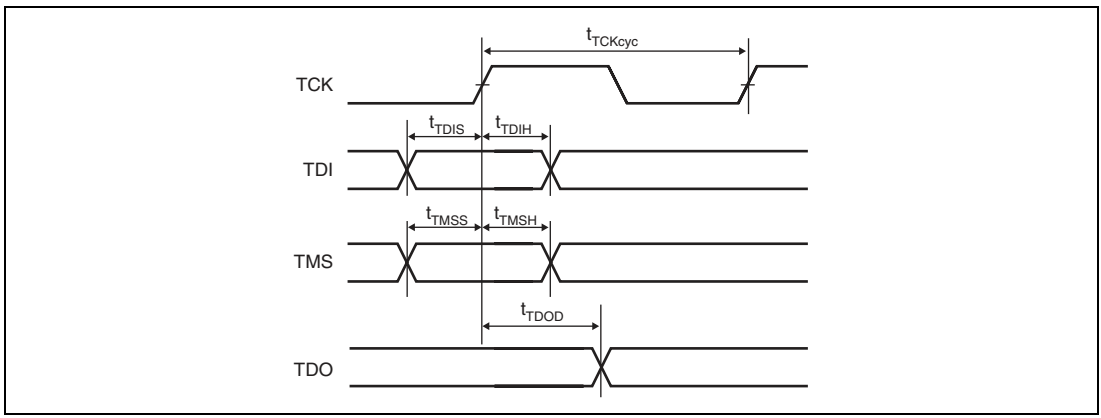


Figure 38.67 H-UDI Data Transfer Timing

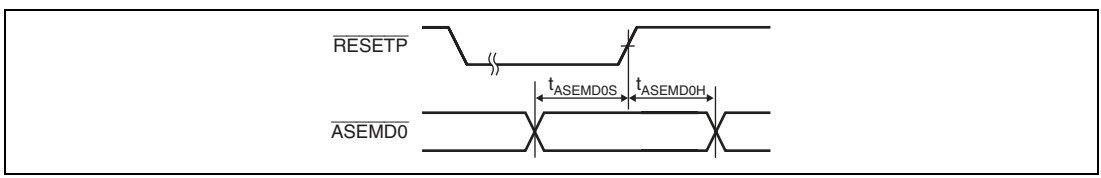


Figure 38.68 $\overline{\text{ASEMD0}}$ Input Timing

38.5 A/D Converter Characteristics

Table 38.24 lists the A/D converter characteristics.

Table 38.24 A/D Converter Characteristics

Conditions: $V_{cc}Q$ = 2.7 to 3.6 V, V_{cc} = 1.4 to 1.6 V, AV_{cc} = 3.0 to 3.6 V, Ta = –20 to 75°C

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

38.6 D/A Converter Characteristics

Table 38.25 lists D/A converter characteristics.

Table 38.25 D/A Converter Characteristics

Conditions: $V_{cc}Q$ = 2.7 to 3.6 V, V_{cc} = 1.4 to 1.6 V, AV_{cc} = 3.0 to 3.6 V, Ta = –20 to 75°C

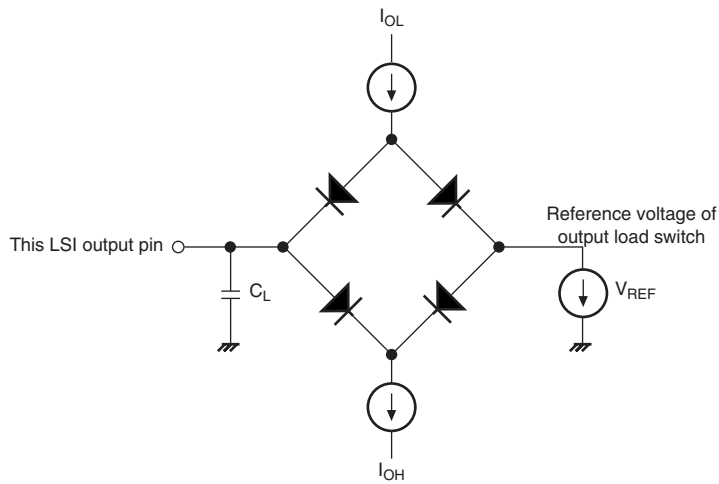
Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	8	8	8	bits	
Conversion time	—	—	10.0	μs	20 pF capacitive load
Absolute accuracy	—	±2.5	±4.0	LSB	2 MΩ resistance load

38.7 AC Characteristic Test Conditions

- I/O signal reference level:

  $\frac{V_{CC}Q}{2}$, $\frac{V_{CC}Q1}{2}$

  ($V_{cc}Q$ = 2.7 to 3.6 V, $V_{cc}Q1$ = 2.7 to 3.6 V or 1.65 to 1.95 V, V_{cc} = 1.4 to 1.6 V)
- Input pulse level: VccQ to VssQ, VccQ1 to VssQ1
- 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: CKIO, $\overline{CS0}$, $\overline{CS2}$ to $\overline{CS6B}$
50 pF: All other pins
2. I_{OL} = 0.2 mA, I_{OH} = -0.2 mA

Figure 38.69 Output Load Circuit

Appendix

A. Pin States

Table A.1 Pin States

Category									
PLBG 0256 GA-A	PLBG 0256 KA-A	Pin Name	Power-On Reset	Manual Reset	Software Standby	Hardware Standby	Bus Release	I/O	Handling of Unused Pins
A1	A2	VssQ	—	—	—	—	—	—	
A2	D5	VccQ	—	—	—	—	—	—	
A3	D6	STATUS1/PTH3	H	H/P	L/K	L/Z	L/P	O/IO	Open
A4	D7	LCD_DATA13/ PINT13/PTD5	V	O/I/P	O/I/P	Z/Z/Z	O/I/P	O/I/IO	Open
A5	E6	VssQ	—	—	—	—	—	—	
A6	D8	VccQ	—	—	—	—	—	—	
A7	E8	LCD_DATA5/ PTC5	V	O/P	O/K	Z/Z	O/P	O/IO	Open
A8	E9	LCD_DATA1/ PTC1	V	O/P	O/K	Z/Z	O/P	O/IO	Open
A9	D10	LCD_CL2/PTE2	V	O/P	O/K	Z/Z	O/P	O/IO	Open
A10	A11	VssQ	—	—	—	—	—	—	
A11	E12	VccQ	—	—	—	—	—	—	
A12	E13	LCD_CLK	I	I	I	Z	I	I	Pull-up
A13	D12	VssQ	—	—	—	—	—	—	
A14	E15	VccQ	—	—	—	—	—	—	
A15	D13	USB1_pwr_en/ USBF_UPLUP/ PTH0	Z	O/O/P	O/O/K	Z/Z/Z	O/O/P	O/O/IO	Pull-up
A16	A15	AVss	—	—	—	—	—	—	
A17	A16	AN0/PTF1	Z	Z/I	Z/Z	Z/Z	I/I	I/I	Pull-up
A18	B18	AVcc_USB	—	—	—	—	—	—	
A19	D17	AVss_USB	—	—	—	—	—	—	
A20	B21	VssQ	—	—	—	—	—	—	

PLBG0256GA-A	PLBG0256KA-A	Pin Name	Category: Power-On Reset	Manual Reset	Software Standby	Hardware Standby	Bus Release	I/O	Handling of Unused Pins
B1	E4	Vcc_PLL2	—	—	—	—	—	—	
B2	B1	MD2	I	i	i	Z	i	I	Must be used
B3	B2	XTAL	O	O	O	O	O	O	Open
B4	A5	$\overline{\text{RESETM}}$	I	I	I	I	I	I	Pull-up
B5	A4	MD4	I	i	Z	Z	i	I	Must be used
B6	C1	LCD_DATA15/ PINT15/PTD7	V	O/I/P	O/I/P	Z/Z/Z	O/I/P	O/I/IO	Open
B7	B3	LCD_DATA11/ PTD3	V	O/P	O/K	Z/Z	O/P	O/IO	Open
B8	E7	LCD_DATA7/ PTC7	V	O/P	O/K	Z/Z	O/P	O/IO	Open
B9	D9	LCD_DATA3/ PTC3	V	O/P	O/K	Z/Z	O/P	O/IO	Open
B10	E10	LCD_FLM/PTE0	V	O/P	O/K	Z/Z	O/P	O/IO	Open
B11	D11	LCD_M_DISP/ PTE4	V	O/P	O/K	Z/Z	O/P	O/IO	Open
B12	E14	SIOF0_MCLK/ PTS3	V	I/P	Z/K	Z/Z	I/P	I/IO	Open
B13	E16	USB2_pwr_en/ PTH1	Z	O/P	O/K	Z/Z	O/P	O/IO	Pull-up
B14	B16	DA1/PTF6	Z	Z/I	Z/Z	Z/Z	O/I	O/I	Open
B15	B17	AN2/PTF3	Z	Z/I	Z/Z	Z/Z	I/I	I/I	Pull-up
B16	A17	USB2_M	Z*2	L	Z	Z	I	IO	Pull-down
B17	A18	USB1_P	Z*1	Z*1	Z	Z	I	IO	Open
B18	A21	USB1_M	Z*1	Z*1	Z	Z	I	IO	Open
B19	A20	AVcc_USB	—	—	—	—	—	—	
B20	E20	VccQ	—	—	—	—	—	—	
C1	D2	Vcc_PLL1	—	—	—	—	—	—	
C2	A1	MD1	I	i	i	Z	i	I	Must be used

Category									
PLBG 0256 GA-A	PLBG 0256 KA-A	Pin Name	Power-On Reset	Manual Reset	Software Standby	Hardware Standby	Bus Release	I/O	Handling of Unused Pins
C3	B5	MD5	I	i	i	Z	i	I	Must be used
C4	A3	EXTAL	I	I	I	I	I	I	Pull-up
C5	B4	MD3	I	i	Z	Z	i	I	Must be used
C6	B7	LCD_DATA12/ PINT12/PTD4	V	O/I/P	O/I/P	Z/Z/Z	O/I/P	O/I/IO	Open
C7	B8	LCD_DATA9/ PTD1	V	O/P	O/K	Z/Z	O/P	O/IO	Open
C8	B9	LCD_DATA6/ PTC6	V	O/P	O/K	Z/Z	O/P	O/IO	Open
C9	B10	LCD_DATA2/ PTC2	V	O/P	O/K	Z/Z	O/P	O/IO	Open
C10	B11	LCD_DON/PTE1	V	O/P	O/K	Z/Z	O/P	O/IO	Open
C11	A12	SIOF0_SYNC/ PTS4	V	O/P	Z/K	Z/Z	IO/P	IO/IO	Open
C12	A13	SIOF0_TxD/PTS2	V	O/P	Z/K	Z/Z	O/P	O/IO	Open
C13	A14	SIOF0_SCK/ PTS0	V	O/P	Z/K	Z/Z	IO/P	IO/IO	Open
C14	E17	$\overline{\text{ADTRG}}$/PTF0	V	I/P	Z/K	Z/Z	I/P	I/I	Open
C15	D18	AN3/PTF4	Z	Z/I	Z/Z	Z/Z	I/I	I/I	Pull-up
C16	D16	USB2_P	Z*2	L	Z	Z	I	IO	Pull-down
C17	B19	AVcc	—	—	—	—	—	—	
C18	E18	USB1d_TXDPLS/ AFE_SCLK/$\overline{\text{IOIS16}}$/ $\overline{\text{PCC\_IOIS16}}$/PTG4	Z	O/I/I/I/P	O/Z/Z/Z/ K	Z/Z/Z/Z/Z	O/I/I/I/P	O/I/I/I/IO	Pull-up
C19	B20	$\overline{\text{USB1\_ovr\_current}}$/ USBF_VBUS	I	I/I	I/I	I/I	I/I	I/I	Pull-down
C20	E21	EXTAL_USB	I	I	I	I	I	I	Pull-up
D1	F1	VssQ1	—	—	—	—	—	—	
D2	D1	MD0	I	i	I	Z	i	I	Must be used

Category									
PLBG 0256 GA-A	PLBG 0256 KA-A	Pin Name	Power-On Reset	Manual Reset	Software Standby	Hardware Standby	Bus Release	I/O	Handling of Unused Pins
D3	C2	D31/PTB7	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
D4	B6	STATUS0/PTH2	H	H/P	H/K	H/Z	L/P	O/IO	Open
D5	A6	LCD_DATA14/ PINT14/PTD6	V	O/I/P	O/I/P	Z/Z/Z	O/I/P	O/I/IO	Open
D6	A7	LCD_DATA10/ PTD2	V	O/P	O/K	Z/Z	O/P	O/IO	Open
D7	A8	LCD_DATA8/ PTD0	V	O/P	O/K	Z/Z	O/P	O/IO	Open
D8	A9	LCD_DATA4/ PTC4	V	O/P	O/K	Z/Z	O/P	O/IO	Open
D9	A10	LCD_DATA0/ PTC0	V	O/P	O/K	Z/Z	O/P	O/IO	Open
D10	E11	LCD_CL1/PTE3	V	O/P	O/K	Z/Z	O/P	O/IO	Open
D11	B12	Vss	—	—	—	—	—	—	
D12	B13	Vcc	—	—	—	—	—	—	
D13	B14	SIOF0_RxD/PTS1	V	I/P	Z/K	Z/Z	I/P	I/IO	Open
D14	B15	$\overline{\text{USB2\_ovr\_current}}$	I	I	I	I	I	I	Pull-up
D15	D14	DA0/PTF5	Z	Z/I	Z/Z	Z/Z	O/I	O/I	Open
D16	D15	AN1/PTF2	Z	Z/I	Z/Z	Z/Z	I/I	I/I	Pull-up
D17	A19	USB1d_DMNS/ PINT11/ AFE_RLYCNT/ PCC_BVD2/PTG3	Z	I/I/O/I/P	I/I/O/Z/P	Z/Z/Z/Z/Z	I/I/O/I/P	I/I/O/I/IO	Pull-up
D18	C21	USB1d_SUSPEND/ $\overline{\text{REFOUT}}$/IRQOUT/ PTP4	Z	O/O/O/ P	O/Z/Z/K	Z/Z/Z/Z	O/O/O/P	O/O/O/ IO	Pull-up
D19	F18	XTAL_USB	O	O	O	O	O	O	Open
D20	F21	USB1d_TXENL/ $\overline{\text{PCC\_CD1}}$/ PINT8/PTG0	Z	O/I/I/P	O/Z/I/P	Z/Z/Z/Z	O/I/I/P	O/I/I/IO	Pull-up
E1	G1	VccQ1	—	—	—	—	—	—	
E2	E1	Vss_PLL2	—	—	—	—	—	—	
E3	F4	Vss_PLL1	—	—	—	—	—	—	

Category									
PLBG 0256 GA-A	PLBG 0256 KA-A	Pin Name	Power-On Reset	Manual Reset	Software Standby	Hardware Standby	Bus Release	I/O	Handling of Unused Pins
E4	G4	D30/PTB6	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
E17	G18	USB1d_SPEED/ $\overline{\text{PCC\_CD2}}$/ PINT9/PTG1	Z	O/I/I/P	O/Z/I/P	Z/Z/Z/Z	O/I/I/P	O/I/I/IO	Pull-up
E18	D20	USB1d_RCV/ AFE_FS/$\overline{\text{PCC\_REG}}$/ IRQ5/PTG6	Z	I/I/O/I/P	I/Z/O/I/K	Z/Z/Z/Z/Z	I/I/O/I/P	I/I/O/I/IO	Pull-up
E19	D21	USB1d_TXSE0/ AFE_TXOUT/ $\overline{\text{PCC\_DRV}}$/IRQ4/ PTG5	Z	O/O/O/ I/P	O/Z/O/I/K	Z/Z/Z/Z/Z	O/O/O/I/ P	O/O/O/I/ IO	Pull-up
E20	G21	VssQ	—	—	—	—	—	—	
F1	G2	D24/PTB0	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
F2	E2	D29/PTB5	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
F3	D4	D28/PTB4	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
F4	H4	D27/PTB3	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
F17	F17	MMC_VDDON/ $\overline{\text{SCIF1\_CTS}}$/ LCD_VEPWC/ TPU_TO3/PTV4	O	O/I/O/ O/P	Z/Z/O/O/ K	Z/Z/Z/Z/Z	O/I/O/O/ P	O/I/O/O/I O	Open
F18	C20	$\overline{\text{AFE\_RDET}}$/ IIC_SDA/PTE5	I	I/I/I	I/I/I	I/I/I	I/IO/I	I/IO/I	Pull-up
F19	F20	USB1d_DPLS/ PINT10/AFE_HC1/ PCC_BVD1/PTG2	Z	I/I/O/I/P	I/I/Z/Z/P	Z/Z/Z/Z/Z	I/I/O/I/P	I/I/O/I/IO	Pull-up
F20	H20	VccQ	—	—	—	—	—	—	
G1	H2	VssQ1	—	—	—	—	—	—	
G2	F2	D26/PTB2	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
G3	E5	D25/PTB1	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
G4	J4	Vcc	—	—	—	—	—	—	
G17	G17	Vss	—	—	—	—	—	—	

Category									
PLBG 0256 GA-A	PLBG 0256 KA-A	Pin Name	Power-On Reset	Manual Reset	Software Standby	Hardware Standby	Bus Release	I/O	Handling of Unused Pins
G18	H18	$\overline{\text{MMC\_ODMOD}}$/ $\overline{\text{SCIF1\_RTS}}$/ LCD_VCPWC/ TPU_TO2/PTV3	O	O/O/O/ O/P	Z/Z/O/O/ K	Z/Z/Z/Z/Z	O/O/O/ O/P	O/O/O/ O/IO	Open
G19	G20	AFE_RXIN/ IIC_SCL/PTE6	I	I/I/I	I/I/I	I/I/I	I/IO/I	I/IO/I	Pull-up
G20	J20	SIM_CLK/ SCIF1_SCK/ SD_DAT3/PTV0	Z	O/Z/I/P	Z/Z/Z/K	Z/Z/Z/Z	O/I/IO/P	O/IO/IO/ IO	Pull-up
H1	J1	VccQ1	—	—	—	—	—	—	
H2	H1	D23/PTA7	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
H3	F5	D22/PTA6	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
H4	G5	Vss	—	—	—	—	—	—	
H17	J18	Vcc	—	—	—	—	—	—	
H18	H17	SIM_RST/ SCIF1_RxD/ SD_WP/PTV1	Z	O/Z/I/P	Z/Z/Z/K	Z/Z/Z/Z	O/I/I/P	O/I/I/IO	Pull-up
H19	H21	SIM_D/ SCIF1_TxD/ SD_CD/PTV2	Z	I/Z/I/P	Z/Z/I/K	Z/Z/Z/Z	IO/O/I/P	IO/O/I/ IO	Pull-up
H20	K20	MMC_DAT/ SIOF1_TxD/ SD_DAT0/ TPU_TI3A/PTU2	Z	I/O/I/I/P	Z/Z/Z/Z/K	Z/Z/Z/Z/Z	IO/O/IO/ I/P	IO/O/IO/ I/IO	Pull-up
J1	K1	VssQ1	—	—	—	—	—	—	
J2	J2	D20/PTA4	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
J3	K4	D21/PTA5	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
J4	H5	D19/PTA3	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
J17	K17	MMC_CMD/ SIOF1_RxD/ SD_CMD/ TPU_TI2B/PTU1	Z	I/I/I/I/ P	Z/Z/Z/Z/K	Z/Z/Z/Z/ Z	IO/I/IO/ I/P	IO/I/IO/ I/IO	Pull-up

Category									
PLBG 0256 GA-A	PLBG 0256 KA-A	Pin Name	Power-On Reset	Manual Reset	Software Standby	Hardware Standby	Bus Release	I/O	Handling of Unused Pins
J18	J17	SIOF1_MCLK/ SD_DAT1/ TPU_TI3B/PTU3	Z	I/I/I/P	Z/Z/Z/K	Z/Z/Z/Z	I/IO/I/P	I/IO/I/IO	Pull-up
J19	J21	SIOF1_SYNC/ SD_DAT2/PTU4	Z	O/I/P	Z/Z/K	Z/Z/Z	IO/IO/P	IO/IO/IO	Pull-up
J20	L17	$\overline{\text{SCIF0\_RTS}}$/ TPU_TO0/PTT3	V	O/O/P	Z/Z/K	Z/Z/Z	O/O/P	O/O/IO	Open
K1	L1	VccQ1	—	—	—	—	—	—	
K2	K2	D17/PTA1	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
K3	J5	D18/PTA2	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
K4	L4	D16/PTA0	Z	Z/P	Z/K	Z/Z	Z/P	IO/IO	Pull-up
K17	L20	SCIF0_TxD/IrTX/ PTT2	V	Z/Z/P	Z/Z/K	Z/Z/Z	O/O/P	O/O/IO	Open
K18	K18	$\overline{\text{SCIF0\_CTS}}$/ TPU_TO1/PTT4	V	I/O/P	Z/Z/K	Z/Z/Z	I/O/P	I/O/IO	Open
K19	K21	MMC_CLK/ SIOF1_SCK/ SD_CLK/ TPU_TI2A/PTU0	Z	O/O/O/ I/P	O/Z/Z/Z/ K	Z/Z/Z/Z/ Z	O/IO/O/ I/P	O/IO/O/ I/IO	Pull-up
K20	M17	VssQ	—	—	—	—	—	—	
L1	K5	CKIO	IO	ZIO	ZIO	Z	ZIO	IO	Open
L2	M1	$\overline{\text{WE2}}$/DQMUL/ $\overline{\text{ICIORD}}$	H	H/H/H	HZ/HZ/ HZ	Z/Z/Z	Z/Z/Z	O/O/O	Open
L3	M4	$\overline{\text{WE3}}$/DQMUU/ $\overline{\text{ICIOWR}}$	H	H/H/H	HZ/HZ/ HZ	Z/Z/Z	Z/Z/Z	O/O/O	Open
L4	L5	RD/$\overline{\text{WR}}$	H	H	HZ	Z	Z	O	Open
L17	L21	SCIF0_RxD/IrRX/ PTT1	V	Z/Z/P	Z/Z/K	Z/Z/Z	I/I/P	I/I/IO	Open
L18	M20	IRQ3/$\overline{\text{IRL3}}$/PTP3	V	I/I/P	I/I/K	Z/Z/Z	I/I/P	I/I/IO	Open
L19	N17	SCIF0_SCK/PTT0	V	Z/P	Z/K	Z/Z	I/P	IO/IO	Open
L20	L18	VccQ	—	—	—	—	—	—	

Category									
PLBG 0256 GA-A	PLBG 0256 KA-A	Pin Name	Power-On Reset	Manual Reset	Software Standby	Hardware Standby	Bus Release	I/O	Handling of Unused Pins
M1	L2	$\overline{\text{CAS}}$/PTH5	H	H/P	HZ/K	Z/Z	HZ/P	O/IO	Open
M2	N1	$\overline{\text{WE0}}$/DQMLL	H	H/H	HZ/HZ	Z/Z	Z/Z	O/O	Open
M3	N5	$\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$	H	H/H/H	HZ/HZ/HZ	Z/Z/Z	Z/Z/Z	O/O/O	Open
M4	M5	CKE/PTH4	Z	O/P	HZ/K	Z/Z	OZ/P	O/IO	Open
M17	M21	IRQ1/$\overline{\text{IRL1}}$/PTP1	V	I/I/P	I/I/K	Z/Z/Z	I/I/P	I/I/IO	Open
M18	N20	NMI	I	I	I	I	I	I	Pull-up
M19	M18	IRQ0/$\overline{\text{IRL0}}$/PTP0	V	I/I/P	I/I/K	Z/Z/Z	I/I/P	I/I/IO	Open
M20	P17	IRQ2/$\overline{\text{IRL2}}$/PTP2	V	I/I/P	I/I/K	Z/Z/Z	I/I/P	I/I/IO	Open
N1	M2	$\overline{\text{RAS}}$/PTH6	H	H/P	HZ/K	Z/Z	HZ/P	O/IO	Open
N2	P1	$\overline{\text{CS3}}$	H	H	HZ	Z	Z	O	Open
N3	P5	$\overline{\text{CS2}}$	H	H	HZ	Z	Z	O	Open
N4	N4	Vcc	—	—	—	—	—	—	
N17	N21	Vss	—	—	—	—	—	—	
N18	P20	AUDATA2/PTJ3	X	O/P	O/K	Z/Z	O/P	O/IO	Open
N19	N18	AUDATA1/PTJ2	X	O/P	O/K	Z/Z	O/P	O/IO	Open
N20	R17	AUDATA3/PTJ4	X	O/P	O/K	Z/Z	O/P	O/IO	Open
P1	N2	VssQ1	—	—	—	—	—	—	
P2	W2	A14	O	O	OZ	Z	Z	O	Open
P3	P2	A17	O	O	OZ	Z	Z	O	Open
P4	R5	Vss	—	—	—	—	—	—	
P17	P21	Vcc	—	—	—	—	—	—	
P18	R20	AUDATA0/PTJ1	X	O/P	O/K	Z/Z	O/P	O/IO	Open
P19	P18	AUDCK/PTJ6	V	O/P	O/K	Z/Z	O/P	O/IO	Open
P20	T17	VssQ	—	—	—	—	—	—	
R1	P4	VccQ1	—	—	—	—	—	—	
R2	T2	A11	O	O	OZ	Z	Z	O	Open
R3	R2	A13	O	O	OZ	Z	Z	O	Open
R4	R1	A15	O	O	OZ	Z	Z	O	Open
R17	T20	$\overline{\text{AUDSYNC}}$/PTJ0	X	O/P	O/K	Z/Z	O/P	O/IO	Open

Category									
PLBG 0256 GA-A	PLBG 0256 KA-A	Pin Name	Power-On Reset	Manual Reset	Software Standby	Hardware Standby	Bus Release	I/O	Handling of Unused Pins
R18	R21	$\overline{\text{ASEMD0}}$	I	I	I	I	I	I	Pull-up
R19	R18	$\overline{\text{TRST}}$/PTL7	I	I/P	Z/K	Z/Z	I/P	I/IO	Pull-down
R20	U17	VccQ	—	—	—	—	—	—	
T1	T5	A16	O	O	OZ	Z	Z	O	Open
T2	V1	A6	O	O	OZ	Z	Z	O	Open
T3	V2	A5	O	O	OZ	Z	Z	O	Open
T4	T1	A12	O	O	OZ	Z	Z	O	Open
T17	U20	TMS/PTL6	I	I/P	Z/K	Z/Z	I/P	I/IO	Pull-up
T18	T18	TCK/PTL3	I	I/P	Z/K	Z/Z	I/P	I/IO	Pull-up
T19	U21	PCC_RESET/ PINT7/PTK3	V	O/I/P	O/I/P	Z/Z/Z	O/I/P	O/I/IO	Open
T20	V18	$\overline{\text{ASEBRKAK}}$/PTJ5	V	O/P	O/K	Z/Z	O/P	O/IO	Open
U1	R4	VssQ1	—	—	—	—	—	—	
U2	T4	A9	O	O	OZ	Z	Z	O	Open
U3	W1	A4	O	O	OZ	Z	Z	O	Open
U4	AA3	A10	O	O	OZ	Z	Z	O	Open
U5	Y5	D11	Z	Z	Z	Z	Z	IO	Pull-up
U6	Y6	D8	Z	Z	Z	Z	Z	IO	Pull-up
U7	AA8	D4	Z	Z	Z	Z	Z	IO	Pull-up
U8	AA9	D1	Z	Z	Z	Z	Z	IO	Pull-up
U9	AA10	Vcc	—	—	—	—	—	—	
U10	V11	Vss	—	—	—	—	—	—	
U11	U11	$\overline{\text{BACK}}$	O	O	O	Z	L	O	Open
U12	U12	$\overline{\text{BS}}$	H	H	HZ	Z	Z	O	Open
U13	V13	A19/PTR1	O	O/P	OZ/K	Z/Z	Z/P	O/IO	Open
U14	U15	A22/PTR4	O	O/P	OZ/K	Z/Z	Z/P	O/IO	Open
U15	U16	A24/PTR6	O	O/P	OZ/K	Z/Z	Z/P	O/IO	Open
U16	V15	$\overline{\text{DACK0}}$/PINT1/ PTM4	V	O/I/P	O/I/P	Z/Z/Z	O/I/P	O/I/IO	Open
U17	W21	$\overline{\text{DREQ1}}$/PTM7	V	I/P	Z/K	Z/Z	I/P	I/IO	Open

Category									
PLBG 0256 GA-A	PLBG 0256 KA-A	Pin Name	Power-On Reset	Manual Reset	Software Standby	Hardware Standby	Bus Release	I/O	Handling of Unused Pins
U18	T21	TDI/PTL4	I	I/P	Z/K	Z/Z	I/P	I/IO	Pull-up
U19	V21	PCC_RDY/PINT6/ PTK2	V	I/I/P	Z/I/P	Z/Z/Z	I/I/P	I/I/IO	Open
U20	W20	TDO/PTL5	Z	O/P	OZ/K	Z/Z	O/P	O/IO	Open
V1	U1	VccQ1	—	—	—	—	—	—	
V2	Y2	A3	O	O	OZ	Z	Z	O	Open
V3	U4	A7	O	O	OZ	Z	Z	O	Open
V4	AA6	D12	Z	Z	Z	Z	Z	IO	Pull-up
V5	Y4	D14	Z	Z	Z	Z	Z	IO	Pull-up
V6	AA7	D9	Z	Z	Z	Z	Z	IO	Pull-up
V7	Y7	D6	Z	Z	Z	Z	Z	IO	Pull-up
V8	Y8	D2	Z	Z	Z	Z	Z	IO	Pull-up
V9	Y9	D0	Z	Z	Z	Z	Z	IO	Pull-up
V10	Y10	$\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$/PTM1	H	H/H/P	HZ/HZ/K	Z/Z/Z	Z/Z/P	O/O/IO	Open
V11	V12	$\overline{\text{BREQ}}$	Z	I	I	Z	I	I	Pull-up
V12	U13	$\overline{\text{WAIT}}$/$\overline{\text{PCC\_WAIT}}$	I	I/I	I/I	Z/Z	Z/Z	I/I	Pull-up
V13	U14	A20/PTR2	O	O/P	OZ/K	Z/Z	Z/P	O/IO	Open
V14	V14	A23/PTR5	O	O/P	OZ/K	Z/Z	Z/P	O/IO	Open
V15	Y19	$\overline{\text{DREQ0}}$/PINT0/ PTM6	V	I/I/P	Z/I/P	Z/Z/Z	I/I/P	I/I/IO	Open
V16	Y18	EXTAL_RTC	I	I	I	I	I	I	Pull-up
V17	AA19	XTAL_RTC	O	O	O	O	O	O	Open
V18	V17	$\overline{\text{RESETP}}$	I	I	I	I	I	I	Must be used
V19	AA21	$\overline{\text{PCC\_VS2}}$/PINT5/ PTK1	V	I/I/P	Z/I/P	Z/Z/Z	I/I/P	I/I/IO	Open
V20	V20	VssQ	—	—	—	—	—	—	
W1	U2	A8	O	O	OZ	Z	Z	O	Open
W2	AA2	A2	O	O	OZ	Z	Z	O	Open
W3	AA1	A1	O	O	OZ	Z	Z	O	Open

Category									
PLBG 0256 GA-A	PLBG 0256 KA-A	Pin Name	Power-On Reset	Manual Reset	Software Standby	Hardware Standby	Bus Release	I/O	Handling of Unused Pins
W4	AA4	A0/PTR0	O	O/P	OZ/K	Z/Z	Z/P	O/IO	Open
W5	AA5	D15	Z	Z	Z	Z	Z	IO	Pull-up
W6	V7	D10	Z	Z	Z	Z	Z	IO	Pull-up
W7	V8	D7	Z	Z	Z	Z	Z	IO	Pull-up
W8	V9	D3	Z	Z	Z	Z	Z	IO	Pull-up
W9	V10	$\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$/PTM0	H	H/H/P	HZ/HZ/K	Z/Z/Z	Z/Z/P	O/O/IO	Open
W10	U9	$\overline{\text{CS5A}}$/$\overline{\text{CE2A}}$	H	H/H	HZ/HZ	Z/Z	Z/Z	O/O	Open
W11	AA12	$\overline{\text{CS4}}$	H	H	HZ	Z	Z	O	Open
W12	AA13	A18	O	O	OZ	Z	Z	O	Open
W13	AA14	A21/PTR3	O	O/P	OZ/K	Z/Z	Z/P	O/IO	Open
W14	Y15	A25/PTR7	O	O/P	OZ/K	Z/Z	Z/P	O/IO	Open
W15	Y16	TEND0/PINT2/ PTM2	V	O/I/P	O/I/P	Z/Z/Z	O/I/P	O/I/IO	Open
W16	AA18	VccQ_RTC	—	—	—	—	—	—	
W17	V16	TEND1/PINT3/ PTM3	V	O/I/P	O/I/P	Z/Z/Z	O/I/P	O/I/IO	Open
W18	Y20	Vss_RTC	—	—	—	—	—	—	
W19	Y21	$\overline{\text{PCC\_VS1}}$/PINT4/ PTK0	V	I/I/P	Z/I/P	Z/Z/Z	I/I/P	I/I/IO	Open
W20	U18	VccQ	—	—	—	—	—	—	
Y1	Y1	VssQ1	—	—	—	—	—	—	
Y2	V5	VccQ1	—	—	—	—	—	—	
Y3	V6	D13	Z	Z	Z	Z	Z	IO	Pull-up
Y4	Y3	VssQ1	—	—	—	—	—	—	
Y5	V4	VccQ1	—	—	—	—	—	—	
Y6	U5	D5	Z	Z	Z	Z	Z	IO	Pull-up
Y7	U6	VssQ1	—	—	—	—	—	—	
Y8	U7	VccQ1	—	—	—	—	—	—	
Y9	U8	$\overline{\text{CS6A}}$/$\overline{\text{CE2B}}$	H	H/H	H/Z	Z/Z	Z/Z	O/O	Open
Y10	AA11	VssQ1	—	—	—	—	—	—	

PLBG0256GA-A	PLBG0256KA-A	Pin Name	Category: Power-On Reset	Manual Reset	Software Standby	Hardware Standby	Bus Release	I/O	Handling of Unused Pins
Y11	U10	VccQ1	—	—	—	—	—	—	
Y12	Y11	$\overline{CS0}$	H	H	HZ	Z	Z	O	Open
Y13	Y12	$\overline{RD}$	H	H	HZ	Z	Z	O	Open
Y14	Y13	VssQ1	—	—	—	—	—	—	
Y15	Y14	VccQ1	—	—	—	—	—	—	
Y16	AA15	VssQ1	—	—	—	—	—	—	
Y17	AA16	VccQ1	—	—	—	—	—	—	
Y18	AA17	$\overline{DACK1}$/PTM5	V	O/P	O/K	Z/Z	O/P	O/IO	Open
Y19	Y17	CA	I	I	I	I	I	I	Pull-up
Y20	AA20	Vcc_RTC	—	—	—	—	—	—	

Notes: *1 The conditions for setting USB1_P and USB1_M to Z (open) are as follows:
(1) Pull the $\overline{USB1\_ovr\_current}$/USBF_VBUS pin down.
(2) Clear the USB_TRANS bit in UTRCTR to 0 (initial value).
Set the USB_SEL bit in UTRCTR to 1 (initial value).

*2 After negation of $\overline{RESETP}$, USB2_P and USB2_M go low after tens of EXTAL_USB clock cycles have been input.

1. Handlings of unused pins in this table are handling examples with the pin functions set to the initial values of the pin function controller (PFC) and cannot be guaranteed in some cases.
2. Controlled by software when an input buffer (PAD) is not enabled.
3. Normal input pin specification.
4. A schmitt characteristic is provided.
5. A board with which the emulator can be used must be designed according to the emulator specifications.

[Legend]

I: Input (input buffer on, output buffer off)
i: Input (input buffer on, output buffer off, fixed input in internal logic)
O: Output (input buffer off, output buffer on, unidentified level)
L: Low output (input buffer off, output buffer on)
H: High output (input buffer off, output buffer on)
Z: High-impedance (input buffer off, output buffer off)
V: Input buffer off, output buffer off, pull-up on
M: Input buffer on, output buffer off, pull-up on

K: Input buffer off/output buffer off (pull-up on), input buffer off/output buffer off (pull-up off), or input buffer off/output buffer on according to the register settings

P: Input buffer on/output buffer off (pull-up on), input buffer on/output buffer off (pull-up off), or input buffer off/output buffer on according to the register settings

X: Undefined

B. Product Lineup

(1) SH7720 Group

Model	Power Supply Voltage I/O	Power Supply Voltage Internal	Operating Frequency	Product Code	Package	SSL	SDHI
SH7720	3.3 V ±0.3V	1.5 V ±0.1V	133.34 MHz	HD6417720BP133C	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	O	—
				HD6417720BP133CV	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	O	—
				HD6417720BL133C	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	O	—
				HD6417720BL133CV	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	O	—
SH7320				HD6417320BP133C	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	O	O
				HD6417320BP133CV	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	O	O
				HD6417320BL133C	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	O	O
				HD6417320BL133CV	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	O	O

[Legend] O: Provided; —: Not provided

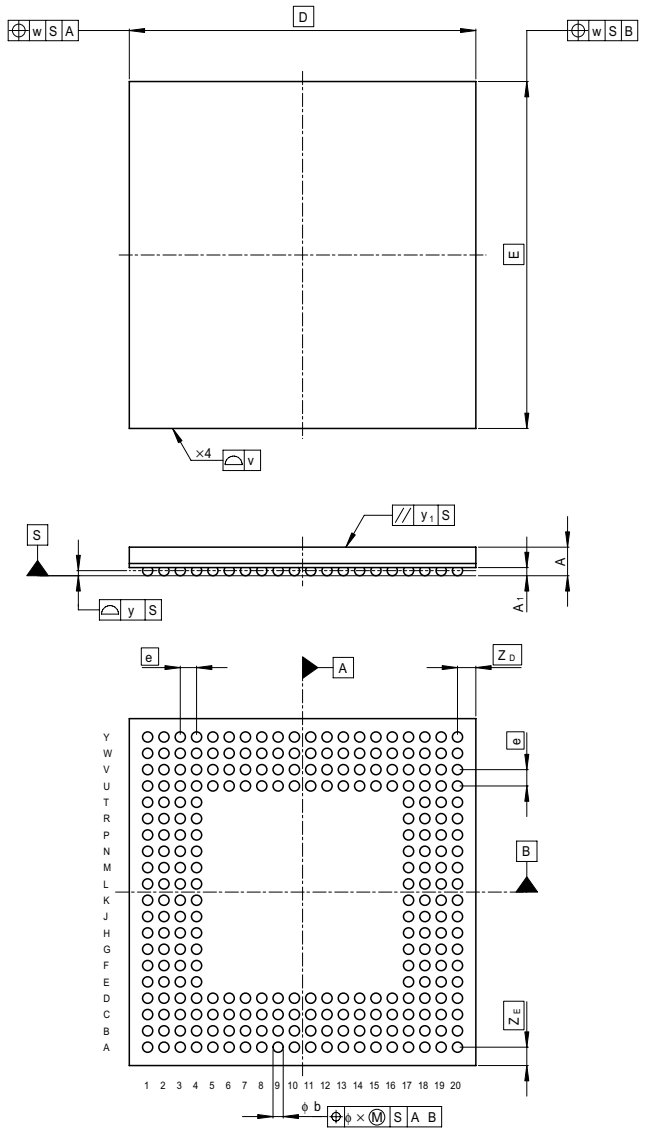
(2) SH7721 Group

Model	Power Supply Voltage I/O	Power Supply Voltage Internal	Operating Frequency	Product Code	Package	SSL	SDHI
SH7721	3.3 V ±0.3V	1.5 V ±0.1V	133.34 MHz	R8A77210C133BG	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	—	—
				R8A77210C133BGV	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	—	—
				R8A77210C133BA	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	—	—
				R8A77210C133BAV	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	—	—
				R8A77211C133BG	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	—	O
				R8A77211C133BGV	256-pin 17mm x 17mm CSP (PLBG0256GA-A)	—	O
				R8A77211C133BA	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	—	O
				R8A77211C133BAV	256-pin 11mm x 11mm CSP (PLBG0256KA-A)	—	O

[Legend] O: Provided; —: Not provided

C. Package Dimensions

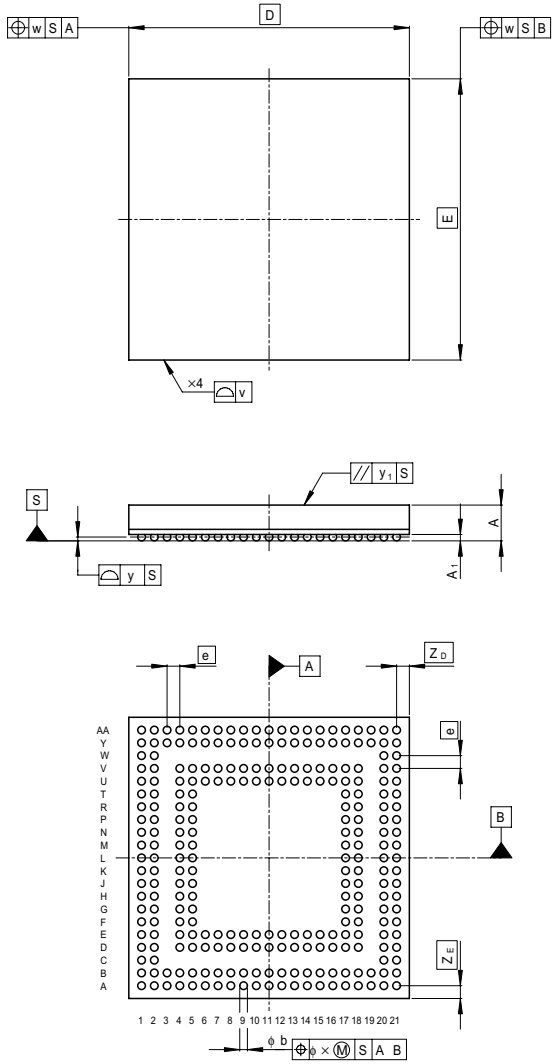
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFBGA256-17x17-0.80	PLBG0256GA-A	BP-256H/BP-256HV	0.6g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	17.0	—
E	—	17.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.40
A_1	0.35	0.40	0.45
e	—	0.80	—
b	0.45	0.50	0.55
x	—	—	0.08
y	—	—	0.10
y_1	—	—	0.2
S_D	—	—	—
S_E	—	—	—
Z_D	—	0.9	—
Z_E	—	0.9	—

Figure C.1 Package Dimensions (PLBG0256GA-A (BP-256H/HV))

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFBGA256-11x11-0.50	PLBG0256KA-A	BP-256C/BP-256CV	0.3g

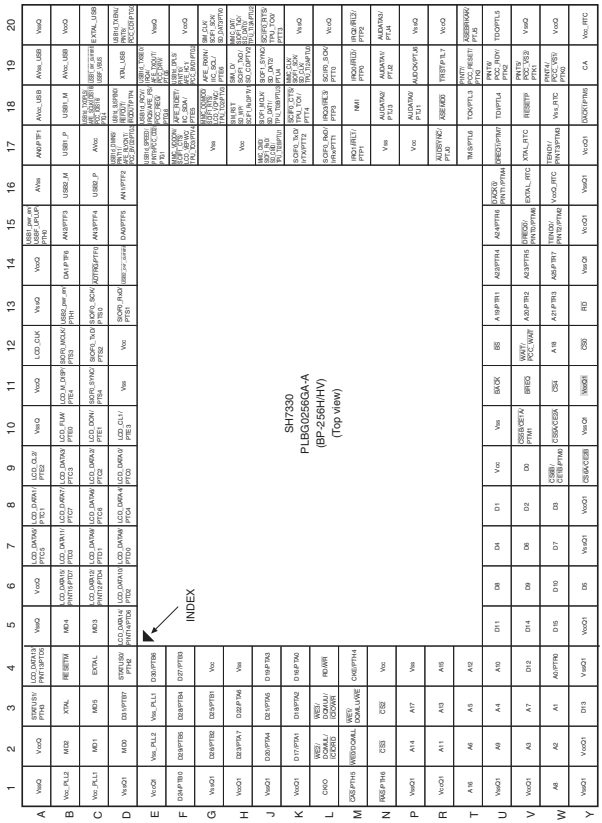


Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	——	11.00	——
E	——	11.00	——
v	——	——	0.15
w	——	——	0.20
A	——	——	1.40
A_1	0.20	0.25	0.30
e	——	0.50	——
b	0.25	0.30	0.35
x	——	——	0.05
y	——	——	0.08
y_1	——	——	0.2
S_D	——	——	——
S_E	——	——	——
Z_D	——	0.5	——
Z_E	——	0.5	——

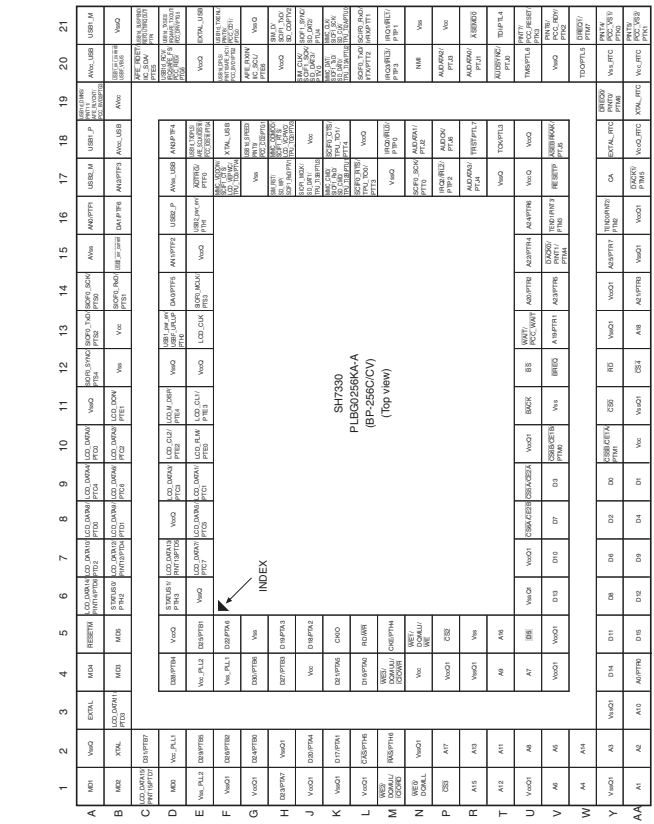
Figure C.2 Package Dimensions (PLBG0256KA-A (BP-256C/CV))

Main Revisions for This Edition

Item	Page	Revision (See Manual for Details)
1.3.1 Pin Assignments Figure 1.2 Pin Assignments (PLBG0256GA-A (BP-256H/HV))	11	Figure amended



Item	Page	Revision (See Manual for Details)
1.3.1 Pin Assignments Figure 1.3 Pin Assignments (PLBG0256KA-A (BP-256C/CV))	12	Figure amended



8.3.7 Interrupt Request Register 3 (IRR3)	255	Table amended

Bit	Bit Name	Initial Value	R/W	Description
7	TENDIR	0	R/W	TEI Interrupt Request Indicates whether the TEI (SIM) interrupt is generated. 0: TEI interrupt request is not generated 1: TEI interrupt request is generated

Item	Page	Revision (See Manual for Details)
8.3.9 Interrupt Request Register 5 (IRR5)	258	Table amended

Bit	Bit Name	Initial Value	R/W	Description
7	ADCIR	0	R/W	ADI Interrupt Request Indicates whether the ADI (ADC) interrupt request is generated. 0: ADI interrupt request is not generated 1: ADI interrupt request is generated
1	SCIF1IR	0	R/W	SCIFI1 Interrupt Request Indicates whether the SCIFI1 (SCIF1) interrupt request is generated. 0: SCIFI1 interrupt request is not generated 1: SCIFI1 interrupt request is generated
0	SCIF0IR	0	R/W	SCIFI0 Interrupt Request Indicates whether the SCIFI0 (SCIF0) interrupt request is generated. 0: SCIFI0 interrupt request is not generated 1: SCIFI0 interrupt request is generated

8.3.10 Interrupt Request Register 6 (IRR6)	259	Table amended

Bit	Bit Name	Initial Value	R/W	Description
5	SIOF1IR	0	R/W	SIOFI1 Interrupt Request Indicates whether the SIOFI1 (SIOF1) interrupt request is generated. 0: SIOFI1 interrupt request is not generated 1: SIOFI1 interrupt request is generated
4	SIOF0IR	0	R/W	SIOFI0 Interrupt Request Indicates whether the SIOFI0 (SIOF0) interrupt request is generated. 0: SIOFI0 interrupt request is not generated 1: SIOFI0 interrupt request is generated

8.4.6 Interrupt Exception Handling and Priority	270	Table amended

Table 8.3 Interrupt Exception Handling Sources and Priority (IRQ Mode)

Interrupt Source		Interrupt Code *1	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
DMAC (2)	DEI4	H'B80*3	0 to 15 (0)	IPRF (11 to 8)	High	
	DEI5	H'BA0*3			Low	
ADC	ADI	H'BE0*3	0 to 15 (0)	IPRF (15 to 12)	—	
SCIF0	SCIFI0	H'C00*3	0 to 15 (0)	IPRG (15 to 12)	—	

	271	Table amended

Interrupt Source		Interrupt Code *1	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
SIM	ERI	H'4E0*2	0 to 15 (0)	IPRB (7 to 4)	High	
	RXI	H'500*2			↕	
	TXI	H'520*2				
	TEI	H'540*2			Low	

	273	Table amended

Table 8.4 Interrupt Exception Handling Sources and Priority (IRL Mode)

Interrupt Source		Interrupt Code *1	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
ADC	ADI	H'BE0*3	0 to 15 (0)	IPRF (15 to 12)	—	
SCIF0	SCIFI0	H'C00*3	0 to 15 (0)	IPRG (15 to 12)	—	
SCIF1	SCIFI1	H'C20*3	0 to 15 (0)	IPRG (11 to 8)	—	

Item	Page	Revision (See Manual for Details)
8.4.6 Interrupt Exception Handling and Priority Table 8.4 Interrupt Exception Handling Sources and Priority (IRL Mode)	274	Table amended

Interrupt Source		Interrupt Code *1	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
SIM	ERI	H'4E0*2	0 to 15 (0)	IPRB (7 to 4)	High	
	RXI	H'500*2			↕	
	TXI	H'520*2				
	TEI	H'540*2			Low	

9.4.5 Refresh Timer Control/Status Register (RTCSR)	328	Table amended		

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.

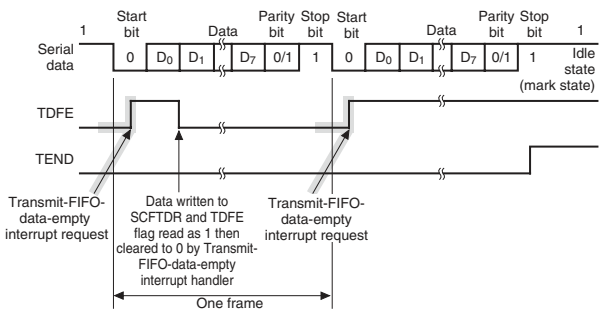
9.4.6 Refresh Timer Counter (RTCNT)	329	Table amended		

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.

9.4.7 Refresh Time Constant Register (RTCOR)	330	Table amended		

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.

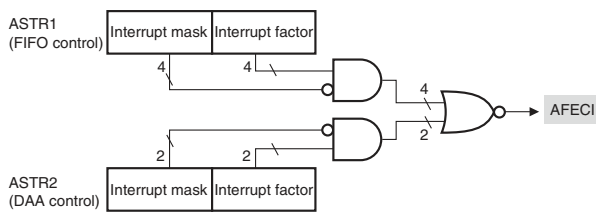
Item	Page	Revision (See Manual for Details)
10.5.3 Other Notes	453	Description added 4. When reading these following flag bits while they are just setting to 1, the read out value of the corresponding flag is 0, but the internal state of this operation may become same as read out 1. If writing 0 to the 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. (1) DMA Channel Control Registers (CHCR_0 to CHCR_5) TE bit. (2) DMA Operation Register (DMAOR) AE bit and NMIF bit. 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).
12.3.4 Using Interval Timer Mode	478	Description amended 3. When the counter overflows, the WDT sets the IOVF flag in WTCSR to 1 and an interval timer interrupt request (ITI) is sent to the INTC. The counter then resumes counting.
15.1 Features	513	Description amended • An interrupt request for each channel (TPIn (n = 0, 1, 2, 3))
16.1 Features	549	Description amended • Allows selection of compare match or overflow for the interrupt source. A common interrupt vector (CMI) is assigned to each interrupt source.

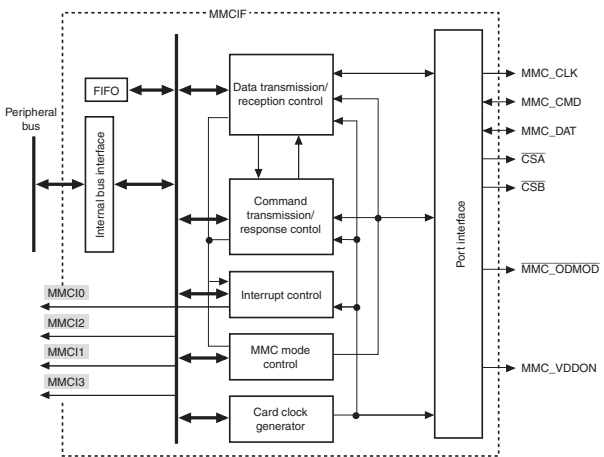
Item	Page	Revision (See Manual for Details)
17.1 Features	561	Description amended • Alarm interrupt (ATI): Frame comparison of seconds, minutes, hours, date, day of the week, month, and year can be used as conditions for the alarm interrupt • Periodic interrupts (PRI): the interrupt cycle may be 1/256 second, 1/64 second, 1/16 second, 1/4 second, 1/2 second, 1 second, or 2 seconds • Carry interrupt (CUI): a carry interrupt indicates when a carry occurs during a counter read
18.1 Features	587	Description amended • Six types of interrupts (SCIFIn (n = 0, 1)) (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 (SCIFIn (n = 0, 1)) (synchronous mode) A common interrupt vector is assigned to each interrupt source.
18.4.2 Serial Operation Figure 18.4 Example of Transmit Operation (Example with 8-Bit Data, Parity, One Stop Bit)	621	Figure amended 
20.3.10 I²C Bus Master Transfer Clock Select Register (ICCKS)	660	Table amended

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 1. The write value should always be 0.

Item	Page	Revision (See Manual for Details)
20.5 Interrupt Request	676	Description added There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK receive, STOP recognition, and arbitration lost/overrun error. Table 20.3 shows the contents of each interrupt request. A common interrupt vector (IICI) is assigned to each interrupt source.
21.1 Features	679	Description amended • Interrupts: One type (SIOFIn (n = 0, 1))
21.3.1 Mode Register (SIMDR)	685	Table amended

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 or reception in slave mode. 0: No bit delay 1: 1-bit delay

22.4.1 Interrupt Timing (4) Interrupt Generator Circuit	749	Description amended Interrupt is generated as is shown in figure 22.4. That is, AFEIFI signal is generated by performing OR operation on the four signals from ASTR1 in FIFO interrupt control and the two signals from ASTR2 in DAA interrupt control, and then sent out to INTC as one interrupt signal (AFECI).
Figure 22.4 Interrupt Generator		Figure amended 
26.3.1 LCDC Input Clock Register (LDICKR)	867	Description amended For a TFT panel, LCD_CL2 = DOTCLK, and for an STN or DSTN monochrome panel, LCD_CL2 = a clock with a frequency of (DOTCLK/data bus width of output to LCD panel). For a color panel, LCD_CL2 = a clock with a frequency of (3 × DOTCLK/data bus width of output to LCD panel). The LDICKR must be set so that the clock input to the LCDC is 66 MHz or less regardless of the LCD_CL2.

Item	Page	Revision (See Manual for Details)
27.3.2 A/D Control/Status Registers (ADCSR)	934	Table amended **Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 11 / TRGE1 / 0 / R / Trigger Enable 10 / TRGE0 / 0 / R / 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 (ADTRG)
27.7.3 Notes on the ADCSR.ADST	948	Newly added
28.4 Operation	956	Description amended 2. When the DAOE0 bit in DACR is set to 1, D/A conversion starts. The results are output after the conversion has ended. The output value will be (DADR0 contents/255) × AVcc.
29.3.3 Area 6 Card Status Change Register (PCC0CSCR)	969	Description added PCC0CSCR is an 8-bit readable/writable able register. PCC0CSCR bits are set to 1 by interrupt sources of the PC card connected to area 6 (only bit 7 can be set to 1 as required). PCC0CSCR is initialized by a power-on reset but retains its value in a manual reset and in software standby mode. A common interrupt vector (PCCI) is assigned to each interrupt source.
31.1 Features Figure 31.1 Block Diagram of MMCIF	1028	Figure amended 

Item	Page	Revision (See Manual for Details)
31.3.13 Interrupt Control Registers 0 and 1 (INTCR0 and INTCR1) • INTCR1	1048	Table amended

Bit	Bit Name	Initial Value	R/W	Description
7	INTRQ2E	0	R/W	MMCI0 Interrupt Enable 0: Disables int_err_n interrupt 1: Enables int_err_n interrupt
6	INTRQ1E	0	R/W	MMCI1 Interrupt Enable 0: Disables int_tran_n interrupt 1: Enables int_tran_n interrupt
5	INTRQ0E	0	R/W	MMCI2 Interrupt Enable 0: Disables int_fstat_n interrupt 1: Enables int_fstat_n interrupt

31.3.20 Interrupt Control Register 2 (INTCR2)	1056	Table amended

Bit	Bit Name	Initial Value	R/W	Description
7	INTRQ3E	0	R/W	MMCI3 Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled

31.6 MMCIF Interrupt Sources Table 31.5 MMCIF Interrupt Sources	1108	Table amended

Name	Interrupt Source	Interrupt Flag
MMCI0	Write error	WRERI
	CRC error*	CRCERI*
	Data timeout error	DTERI
	Command timeout error	CTERI
MMCI2	FIFO empty	FEI
	FIFO full	FFI
MMCI1	Data response	DPRI
	Data transfer end	DTI
	Command response end	CRPI
	Command output end	CMDI
	Data busy end	DBSYI
	Block transfer end	BTI
MMCI3	FIFO ready	FRDYI

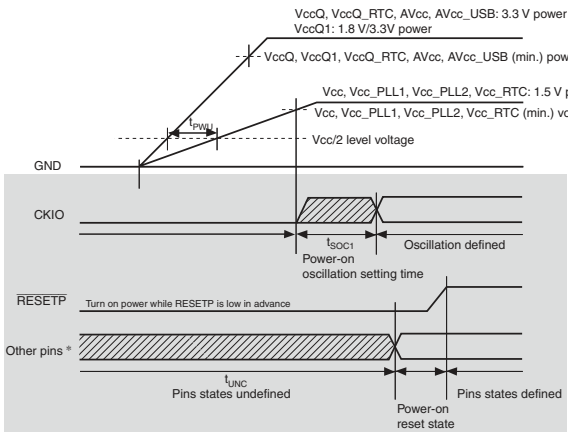
37.3 Register States in Each Operating Mode	1290	Table amended

Register Abbreviation	Power-On Reset*1	Manual Reset*1	Software Standby	Module Standby	Sleep	Module
CMNCR	Initialized	Retained	Retained	—	Retained	BSC
CS0BCR	Initialized	Retained	Retained	—	Retained	
CS2BCR	Initialized	Retained	Retained	—	Retained	
CS3BCR	Initialized	Retained	Retained	—	Retained	
CS4BCR	Initialized	Retained	Retained	—	Retained	
CS5ABCR	Initialized	Retained	Retained	—	Retained	
CS5BBCR	Initialized	Retained	Retained	—	Retained	
CS6ABCR	Initialized	Retained	Retained	—	Retained	
CS6BBCR	Initialized	Retained	Retained	—	Retained	
CS0WCR	Initialized	Retained	Retained	—	Retained	
CS2WCR	Initialized	Retained	Retained	—	Retained	
CS3WCR	Initialized	Retained	Retained	—	Retained	
CS4WCR	Initialized	Retained	Retained	—	Retained	
CS5AWCR	Initialized	Retained	Retained	—	Retained	
CS5BWCR	Initialized	Retained	Retained	—	Retained	
CS6AWCR	Initialized	Retained	Retained	—	Retained	
CS6BWCR	Initialized	Retained	Retained	—	Retained	
SDCR	Initialized	Retained	Retained	—	Retained	
RTCSR	Initialized	Retained	Retained	—	Retained	
RTCNT	Initialized	Retained	Retained	—	Retained	
RTCOR	Initialized	Retained	Retained	—	Retained	

Item	Page	Revision (See Manual for Details)
37.3 Register States in Each Operating Mode	1302	Table amended

Register Abbreviation	Power-On Reset*1	Manual Reset*1	Software Standby	Module Standby	Sleep	Module
BDRB	Initialized	Retained	Retained	Retained	Retained	UBC
BDMRB	Initialized	Retained	Retained	Retained	Retained	
BRCR	Initialized	Retained	Retained	Retained	Retained	
BETR	Initialized	Retained	Retained	Retained	Retained	
BARB	Initialized	Retained	Retained	Retained	Retained	
BAMRB	Initialized	Retained	Retained	Retained	Retained	
BBRB	Initialized	Retained	Retained	Retained	Retained	
BRSR	Initialized	Retained	Retained	Retained	Retained	
BARA	Initialized	Retained	Retained	Retained	Retained	
BAMRA	Initialized	Retained	Retained	Retained	Retained	
BBRA	Initialized	Retained	Retained	Retained	Retained	
BRDR	Initialized	Retained	Retained	Retained	Retained	
BASRA	Initialized	Retained	Retained	Retained	Retained	
BASRB	Initialized	Retained	Retained	Retained	Retained	

38.2 Power-On and Power-Off Order	1306	Figure amended



38.3 DC Characteristics	1309	Table amended

Table 38.4 DC Characteristics (1) [Common]

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Analog (A/D, D/A) power supply current	During A/D conversion	AI_{CC}	—	0.8	2	mA	
	During A/D and D/A conversion		—	2.4	6	mA	
	Idle		—	—	20	µA	T_a = 25°C

Item	Page	Revision (See Manual for Details)
38.4.3 AC Bus Timing Table 38.9 Bus Timing	1322	Table amended

		66.67 MHz			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
Address delay time 1	t_{AD1}	1	13	ns	38.12 to 38.42
Address delay time 2	t_{AD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 13	ns	38.19
Address setup time	t_{AS}	0	—	ns	38.12 to 38.19
Address hold time	t_{AH}	0	—	ns	38.12, 38.13
$\overline{BS}$ delay time	t_{BSD}	—	13	ns	38.12 to 38.36, 38.37, 38.38
$\overline{CS}$ delay time 1	t_{CSD1}	1	13	ns	38.12 to 38.36, 38.37 to 38.42
Read/write delay time 1	t_{RWD1}	1	13	ns	38.12 to 38.36, 38.37 to 38.42
Read strobe delay time	t_{RSD}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 13	ns	38.12 to 38.19, 38.39, 38.40
Read data setup time 1	t_{RDS1}	$1/2t_{cyc}$ + 10	—	ns	38.12 to 38.18, 38.37 to 38.42
Read data setup time 2	t_{RDS2}	7	—	ns	38.20 to 38.23, 38.28 to 38.30, 38.37, 38.38
Read data setup time 3	t_{RDS3}	$1/2t_{cyc}$ + 10	—	ns	38.19
Read data hold time 1	t_{RDH1}	0	—	ns	38.12 to 38.18, 38.37 to 38.42
Read data hold time 2	t_{RDH2}	2	—	ns	38.20 to 38.23, 38.28 to 38.30, 38.37, 38.38
Read data hold time 3	t_{RDH3}	0	—	ns	38.19
Write enable delay time 1	t_{WED1}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 13	ns	38.12 to 38.17, 38.39, 38.40
Write enable delay time 2	t_{WED2}	—	13	ns	38.18

	1323	Table amended

		66.67 MHz			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
Write data delay time 1	t_{WDD1}	—	13	ns	38.12 to 38.18, 38.39 to 38.42
Write data delay time 2	t_{WDD2}	—	13	ns	38.24 to 38.27, 38.31 to 38.33, 38.37, 38.38
Write data hold time 1	t_{WDH1}	1	—	ns	38.12 to 38.18, 38.37 to 38.42
Write data hold time 2	t_{WDH2}	1	—	ns	38.24 to 38.27, 38.31 to 38.33, 38.37, 38.38
Write data hold time 4	t_{WDH4}	0	—	ns	38.12
Write data hold time 5	t_{WDH5}	1	—	ns	38.39 to 38.42
$\overline{WAIT}$ setup time 1	t_{WTS1}	$1/2t_{cyc}$ + 7	—	ns	38.12 to 38.19, 38.40, 38.42
$\overline{WAIT}$ hold time 1	t_{WTH1}	$1/2t_{cyc}$ + 2	—	ns	38.12 to 38.19, 38.40, 38.42
$\overline{RAS}$ delay time 1	t_{RASD1}	1	13	ns	38.20 to 38.36, 38.37, 38.38
$\overline{CAS}$ delay time 1	t_{CASD1}	1	13	ns	38.20 to 38.36, 38.37, 38.38
DQM delay time 1	t_{DQMD1}	1	13	ns	38.20 to 38.36, 38.37, 38.38
$\overline{CKE}$ delay time 1	t_{CKED1}	1	13	ns	38.35, 38.36, 38.37, 38.38
DACK delay time	t_{DACD}	—	13	ns	38.12 to 38.36, 38.37

38.4.15 USB Module Signal Timing Table 38.17 USB Module Clock Timing	1366	Table amended

Item		**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
EXTAL_USB clock frequency (48 MHz)	(USBF)	t_{FREQ}	47.9	48.1	MHz	38.60
	(USBH)		47.976	48.024		
Clock rise time		t_{R48}	—	6	ns	
Clock fall time		t_{F48}	—	6	ns	

Index

R

S

T

U

V

W

X

Renesas 32-Bit RISC Microcomputer
SH7720 Group, SH7721 Group
User's Manual: Hardware

Publication Date: Rev.1.00, June 28, 2007
Rev.4.00, September 21, 2010
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.
7F, 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 1.0

SH7720 Group, SH7721 Group
User's Manual: Hardware

Renesas Electronics Corporation

R01UH0083EJ0400
(Previous Number: REJ09B0033-0300)